Artificial Intelligence 341 (2025) 104293

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

No free lunch theorem for privacy-preserving LLM inference

Xiaojin Zhang [a,1], Yahao Pang [a], Yan Kang [b], Wei Chen [a], Lixin Fan [b], Hai Jin [a], Qiang Yang [b,c,*]

[a] Huazhong University of Science and Technology, China
[b] WeBank, China
[c] Hong Kong University of Science and Technology, China

ARTICLE INFO

Keywords:
Privacy
LLM inference
No free lunch theory

ABSTRACT

Individuals and businesses have been significantly benefited by Large Language Models (LLMs) including PaLM, Gemini and ChatGPT in various ways. For example, LLMs enhance productivity, reduce costs, and enable us to focus on more valuable tasks. Furthermore, LLMs possess the capacity to sift through extensive datasets, uncover underlying patterns, and furnish critical insights that propel the frontiers of technology and science. However, LLMs also pose privacy concerns. Users' interactions with LLMs may expose their sensitive personal or company information. A lack of robust privacy safeguards and legal frameworks could permit the unwarranted intrusion or improper handling of individual data, thereby risking infringements of privacy and the theft of personal identities. To ensure privacy, it is essential to minimize the dependency between shared prompts and private information. Various randomization approaches have been proposed to protect prompts' privacy, but they may incur utility loss compared to unprotected LLMs prompting. Therefore, it is essential to evaluate the balance between the risk of privacy leakage and loss of utility when conducting effective protection mechanisms. The current study develops a framework for inferring privacy-protected Large Language Models (LLMs) and lays down a solid theoretical basis for examining the interplay between privacy preservation and utility. The core insight is encapsulated within a theorem that is called as the NFL (abbreviation of the word No-Free-Lunch) Theorem.

1. Introduction

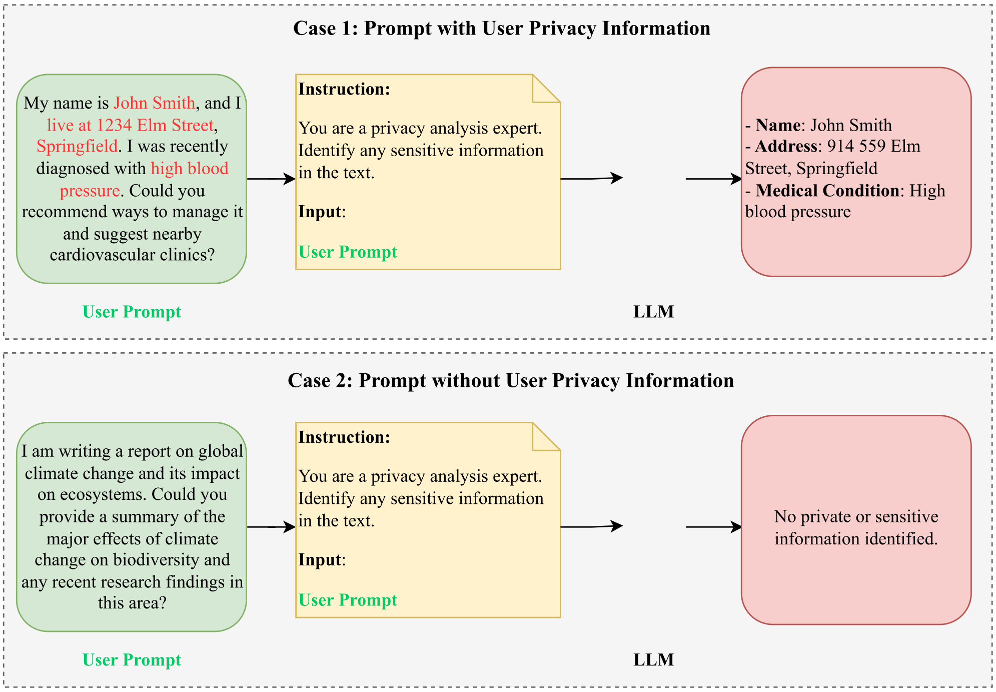
The advent of sophisticated Large Language Models, including PaLM [1] and ChatGPT [2] have brought substantial benefits to both individuals and enterprises. These models are equipped to facilitate our endeavors across a diverse spectrum of domains, from synthesizing information to generating new content and data analysis. By doing so, they enhance our productivity, reduce costs, and free us from tedious work, allowing us to focus on more valuable tasks [3]. Moreover, LLMs can assist in generating ideas, designing solutions, and facilitating research and development. For instance, in domains such as healthcare, finance, and science, LLMs can analyze massive volumes of data, identify patterns, and offer valuable insights that can propel technical and scientific breakthroughs and advancements.

[*] Corresponding author at: WeBank, China.
E-mail addresses: xiaojinzhang@hust.edu.cn (X. Zhang), qyang@cse.ust.hk (Q. Yang).

[1] *Present address:* National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China.

https://doi.org/10.1016/j.artint.2025.104293
Received 30 May 2024; Received in revised form 13 January 2025; Accepted 28 January 2025

Available online 4 February 2025


Case 1: Prompt with User Privacy Information
User Prompt
LLM
Case 2: Prompt without User Privacy Information
User Prompt
LLM

My name is John Smith, and I live at 1234 Elm Street, Springfield. I was recently diagnosed with high blood pressure. Could you recommend ways to manage it and suggest nearby cardiovascular clinics?

Instruction:

You are a privacy analysis expert. Identify any sensitive information in the text.

Input:

User Prompt

- **Name**: John Smith
- **Address**: 914 559 Elm Street, Springfield
- **Medical Condition**: High blood pressure

I am writing a report on global climate change and its impact on ecosystems. Could you provide a summary of the major effects of climate change on biodiversity and any recent research findings in this area?

Instruction:

You are a privacy analysis expert. Identify any sensitive information in the text.

Input:

User Prompt

No private or sensitive information identified.

Fig. 1. Privacy Leakage in Prompts. In Case 1, the prompt contains sensitive user information, including: name (John Smith), address (1234 Elm Street, Springfield), and medical condition (high blood pressure). This enables the LLM to identify and extract private data, potentially leading to privacy leakage. Such information risks exposing the user's identity, location, and health details, which could be misused for identity theft, targeted advertising, or discrimination. In contrast, the prompt in Case 2 excludes sensitive information, significantly reducing the risk of privacy leakage. This comparison highlights how sensitive information in user prompts can be easily identified by LLMs. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

While LLMs offer significant benefits, they also raise important privacy concerns. User prompts, which are central to interactions with LLMs, often include sensitive information such as personal identity, background details, and other confidential data, as illustrated in Fig. 1. When users do not employ privacy protection measures, any prompt containing sensitive information can be easily recognized and extracted by the server. Even with certain privacy protection techniques in place, servers may still deploy various attack methods—such as embedding inversion or model inference—to infer sensitive user data. These threats have been substantiated in numerous studies (see Section 2.1). However, overly stringent privacy protections can compromise the utility of LLMs, while insufficient protection leaves user privacy vulnerable. Therefore, our research aims to strike a balance by ensuring privacy protection for user prompts during the LLM inference phase, with minimal impact on model utility.

Therefore, preserving privacy is crucial when prompting LLMs for inference. In this context, a natural approach to safeguarding the privacy of prompts is to introduce randomization.

This study focuses on a typical LLM inference setting where the LLM operates as a black box, concealing its architecture, model parameters, and inference details within commercial APIs and user interfaces. To address this issue, several randomization techniques [4–6] have been proposed, demonstrating empirical effectiveness in mitigating privacy leakage resulting from exposed prompts. However, these approaches inevitably incur a certain degree of utility loss compared to prompting LMs without any protection. Moreover, existing research lacks the theoretical analysis necessary to quantify the compromise of the reduction in utility and exposure of privacy. The tension between reducing the exposure of private information and the detriment to utility propels our exploration to investigate the research inquiry: "*Theoretically, is there possibility of developing some protective methods that can achieve minimal exposure of private information and detriment to utility at the same time when prompting an LLM?*" Our primary finding is encapsulated in the Theorem 4.4 within context of our proposed framework for privacy-preserving LM inference, presenting a counterargument to this question. We named it as No-Free-Lunch Theorem.

Another pertinent research [7] similarly introduces a theorem that try to address the equilibrium between the utility and privacy. However, the scope of that study primarily focuses on training models in the context of horizontal federated learning, where privacy leakage is defined based on model gradients transmitted between clients and the server. In contrast, our work concentrates on LLM inference, where interactions involving prompts and LLM outputs occur between the client and server as detailed in section 3, and we define privacy leakage as the extent to which an adversary can deduce the original input from an protected one. These distinct definitions of privacy leakage provide novel theoretical insights for analyzing the balance between utility and private information. Our study's contributions are encapsulated in the following points:

- A framework aimed at privacy-preserving LLM inference, wherein we provide formal definitions for privacy leakage (as outlined in Definition 3.1) and utility loss (as outlined in Definition 3.2), has been proposed. Furthermore, we formulate the privacy-preserving LLM inference problem as a constrained optimization problem.
- A No-Free-Lunch theorem (as stated in Theorem 4.4) aimed at privacy-preserving LLM inference, which provides a quantitative characterization of the trade-off between detriment to utility and exposure of private information, has been established. Specifically, it demonstrates that weighted summation of utility reduction and privacy exposure is bounded below by a constant contingent upon the specific problem, and is not zero, thereby implying that a certain degree of model performance must be sacrificed in order to ensure a desired level of privacy preservation.

2. Related work

During the LLM inference, both the client and the LLM server have the potential to act as adversaries, compromising each other's privacy [8,9]. In this study, we focus on a scenario where the LLM server is the adversary and can infer the client's private information by analyzing the prompt provided by the client (see Section 3.1). In this section, we provide a concise overview of existing research concerning attack of privacy, safeguarding measures, and the balance of preserving utility against protecting privacy within the realm of large language model (LLM) inference.

2.1. *Privacy attacks in LLM inference*

During the inference phase of LLM, the LLM server may attempt to infer a client's private information by analyzing the prompts sent by the client. This can be achieved by inferring privacy information from a single prompt or by manipulating an otherwise innocuous conversation with the user to elicit prompts that contain private and sensitive information [10]. A multitude of strategies for launching attacks have been outlined in such contexts. For example, the server has the capability to initiate attacks that invert embeddings [11–14], aiming to reconstruct the original prompt based on the provided embedding. In addition, the server may employ these kinds of attacks for extracting sensitive information such as ethnicity, sex, and age from the provided embedding [15–19]. Moreover, the server hosting the LLM can exploit its model to discern the client's original prompt from a modified version [5,10].

2.2. *Privacy protections in LLM inference*

Secure Multi-Party Computation (SMPC) and Randomization are two mainstream protection mechanisms leveraged to protect the privacy of clients' sensitive information. SMPC facilitates the collaborative computation of a function among several entities, ensuring the privacy of their individual inputs to be confidential. In the context of LLMs, SMPC could be employed for executing computations on encrypted data, which ensures that no single party gains access to other parties' confidential inputs. These methods prioritize enhancing the efficiency of LLM architectures and SMPC protocols to minimize the substantial computation and communication expenses incurred by SMPC protocols [20–26]. Although SMPC can ensure the confidentiality of client data, it requires the LLM to collaborate tightly with the client, which can limit its application to LLMs that are only accessible through commercial APIs. Randomization is an alternative privacy-preserving mechanism that adds random noise to the prompt's embedding to protect the privacy of the prompt. For instance, the InferDPT approach [5] utilizes a differential privacy (DP) mechanism to alter the user's input text, thereby hindering any potential eavesdropping by a malevolent large language model (LLM) server that could deduce sensitive user information. In the context of the InferDPT system, users initially introduce minor, yet semantically coherent, variations to the tokens within their input. Afterward, this modified input is forwarded to an LLM, which formulates a reply and relays it back to the user. The user then utilizes their own pre-trained model to produce the conclusive output by integrating the initial input and the LLM's reply. The DP-OPT technique [27], on the other hand, harnesses advanced Deep Language Networks (as described in DLN [28]) under the guidance of a local model to autonomously refine input prompts. In this process, DP-OPT implements a privacy-preserving aggregation technique to safeguard the confidentiality of the prompts. In a separate study, Staab et al. [10] explored the application of text anonymization techniques for safeguarding user data privacy and concluded through empirical evidence that such methods alone are not adequate for ensuring robust privacy protection.

2.3. *Trade-off between utility and privacy in LLM inference*

In addition, beyond introducing methods to safeguard personal data, several investigations have scrutinized the equilibrium between utility and private information when interacting with LLMs. These investigations primarily focus on two scenarios for LLM inference.

In the first scenario, the LLM operates behind a commercial API, solely relying on textual prompts as inputs. Protection mechanisms typically involve introducing random noise to the prompts or their corresponding embeddings. Representative works in this scenario include DP-OPT [27] and InferDPT [5]. These studies have demonstrated a decline in task performance as the privacy budget decreases. Furthermore, DP-OPT highlights that leveraging larger LLMs can significantly mitigate performance-private information trade-off. LLM is divided into two parts in the second scenario: a larger portion deployed on the server and a smaller portion deployed on the client. The server and client collaborate in training these two portions of the LLM, incorporating random noise into the embedding vectors to protect privacy. Representative examples of research falling under this scenario include TextObfuscator [4]

Table 1
Used notation in this study.

Notation	Meaning
$\widetilde{d}^{(m)}$	The m-th token in client's protected prompt $\widetilde{d}$
ϵ_p	Leakage of private information (Definition 3.1)
$d^{(m)}$	The m-th token in client's prompt d
ϵ_u	Loss of utility (Definition 3.2)
d, $\widetilde{d}$	The client's original and protected prompts, respectively
w	Undistorted embedding of the client's prompt
$\widetilde{w}$	Distorted embedding of the client's prompt
P	Distribution of undistorted embedding w
$\widetilde{P}$	Distribution of distorted embedding $\widetilde{w}$
$\breve{P}$	Distribution of embedding that is independent of the embedding of the client's prompt
P_0	Distribution of test data
$\text{TV}(\cdot\|\|\cdot)$	Two distributions' total variation distance

and SAP [29]. Both studies also exhibited the delicate balance between privacy and utility. Specifically, as randomization increases, utility decreases while privacy increases, and vice versa.

The primary emphasis of these studies lies in the empirical assessment of the balance between confidentiality and performance, lacking a theoretical framework or a numerical evaluation of the utility-privacy equilibrium during LLMs' inference. Our study aims to address this gap by providing a theoretical analysis and quantification in this regard. The notation used in this study is presented in Table 1.

3. A framework of privacy-preserving LLM inference

Our framework considers a typical LLM inference setting in which a client sends prompts to query the *black-box* LLM hosted by a server. The LLM is black-box in the sense that the server of the LLM hides the LLM architecture and parameters as well as inference details, and it only exposes the query and prediction commercial APIs and interface for the client to make inferences. We also assume that the server may mount a privacy attack during the inference to infer client's privacy based on observed prompts, which necessitates client-side privacy protection.

We commence with an introduction of threat model in this section. Subsequently, we expound upon representative protection mechanisms and attacking methods, based on which we subsequently provide formal definitions of utility loss and privacy leakage. Finally, we formulate objective of privacy-preserving LLM inference as a constrained optimization problem.

3.1. Threat model

We assume the LLM server is the attacker. We discuss its threat to the client's private data considering objective, capability, and knowledge of attacker.

Attacker's objective. We regard the LLM server as a potential adversary intent on deducing the client's confidential details with considerable accuracy by analyzing the client's exposed embedding. To this end, the server tries to obtain tokens or words in the prompt as many as possible.

Attacker's capability. We classify the adversary as semi-honest, adhering to the LLM inference rules by ensuring the production of the output, yet potentially seeking to deduce sensitive client data based on client's prompt.

Attacker's knowledge. We assume that the server is aware that the client may apply a randomization mechanism to protect its uploaded prompts. The server may launch attacks using all available information (e.g., its hosted LLM) to reconstruct each word or token from the original prompt based on the observed perturbed embedding.

Additionally, while our primary focus is on scenarios where the LLM server acts as the adversary, we also consider the potential risks posed by malicious clients. Malicious clients may attempt to infer other clients' private data by exploiting shared inference results, model updates, or interactions within collaborative learning frameworks. Our framework is designed to mitigate such risks by implementing a robust isolation mechanism, where each client's prompt undergoes random perturbation before being processed by the LLM. This approach ensures that the LLM server, acting as an intermediary, does not directly access unprotected client data, thereby preventing malicious clients from inferring sensitive information about other users. As a result, our framework is adaptable to scenarios involving both a potentially adversarial server and malicious clients, ensuring comprehensive protection of client privacy.

3.2. Attacking methods

Considering the attacker's knowledge and capability, it is expected that they will employ a privacy breach technique, designated as Attack $\mathcal{A}$, to deduce the sequence of the original prompt's elements from the observed prompt. Herein, we give three representative attacks.

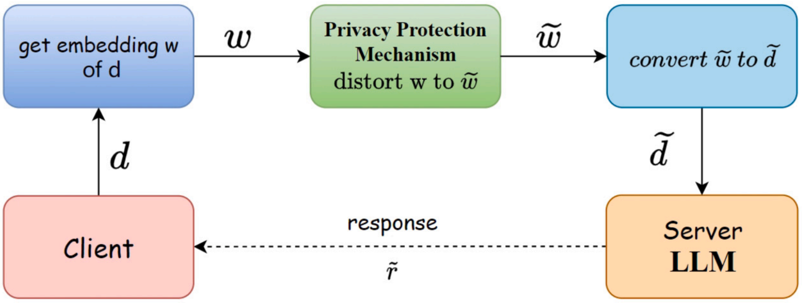

get embedding w of d
w
Privacy Protection Mechanism distort w to $\widetilde{w}$
$\widetilde{w}$
convert $\widetilde{w}$ to $\widetilde{d}$
d
$\widetilde{d}$
Client
response
$\tilde{r}$
Server LLM

Fig. 2. Illustration of the privacy-preserving LLM inference. The LLM inference procedure involves four steps: (1) the client designs a prompt d; (2) the client applies a privacy protection mechanism to the original prompt d to obtain a protected prompt $\widetilde{d}$; (3) the client sends $\widetilde{d}$ the server; (4) The LLM takes $\widetilde{d}$ as input and sends the generated response $\widetilde{r}$ back to the client.

- Input inference attack [19]: the server leverages a pre-trained BERT model to reconstruct tokens of the original prompt from the perturbed prompt. Specifically, the server substitutes each token in the perturbed prompt sequentially with the special one "[MASK]" and lets the BERT model predict the token at the "[MASK]" position. The accuracy of the attack is computed by comparing the predicted tokens to the original tokens.
- Embedding inversion attack [11]: the server deduces the original tokens by referencing the embeddings of the protected tokens. It utilizes an algorithm for nearest neighbor identification to accomplish this objective. Upon receiving a embedding of an altered token, the server identifies its closest counterpart within the vector space, which is then considered the inferred original token. The effectiveness of this process is gauged by the precision of the recovered tokens.
- LLM-assisted inference attack [5,10]: the server exploits the capitalizes of large language models like GPT-4 to reconstruct the original prompt from the perturbed version. To elaborate, the server introduces the perturbed prompt to the GPT-4 system and commands it to regenerate each token. The operation is deemed a success when the regenerated tokens align precisely with those of the original prompt.

In this study, we investigate the trade-off between loss of utility and leakage of privacy when client introduces randomization to the prompt used to query a black-box LLM for protecting the prompt's confidentiality. In the next section, we elaborate on the randomization protection mechanism and the associated privacy-preserving LLM inference process.

3.3. Protection mechanisms

The user employs *privacy preserving mechanism* $\mathcal{M} : \mathbb{R}^m \to \mathbb{R}^m$ that convert the embedding w of original prompt d to a distorted counterpart $\widetilde{w}$. w and $\widetilde{w}$ follow distributions P and $\widetilde{P}$, respectively. The client then converts the distorted embeddings $\widetilde{w}$ back to the protected prompt $\widetilde{d}$, which will be shared with the server for LLM inference. In this way, the chance that the attacker can infer the original prompt d based on $\widetilde{w}$ is reduced. The distribution $\widetilde{P}$ is termed *protected distribution* pertaining to client's prompt.

Let us consider the process of adding randomness as an illustrative case. We posit that w follows the distribution P, which is a normal distribution with mean μ_0 and covariance matrix Σ_0, and the perturbative noise ϵ is also normally distributed with a zero mean and noise covariance Σ_ϵ. Here, Σ_0 is specified as a diagonal matrix with elements $\{\sigma_1^2, \dots, \sigma_m^2\}$, and Σ_ϵ is similarly a diagonal matrix with identical elements $\{\sigma_\epsilon^2, \dots, \sigma_\epsilon^2\}$. Subsequently, original prompt is safeguarded through the addition of noise to its embedding: $\widetilde{w} = w + \epsilon \sim \mathcal{N}(\mu_0, \Sigma_0 + \Sigma_\epsilon)$ with distribution $\widetilde{P} = \mathcal{N}(\mu_0, \Sigma_0 + \Sigma_\epsilon)$. The randomization mechanisms can defend against a broad range of privacy attacks, for it directly replaces original tokens by sampling noise.

With the protection mechanism, the LLM inference procedure is illustrated in Fig. 2 and described as follows:

① The client designs a prompt d to accomplish a specific task.
② The client leverages a privacy protection mechanism $\mathcal{M}$ to transform d to protected version $\widetilde{d}$, aiming to prevent the server (i.e., the attacker) from inferring private information through investigating $\widetilde{d}$.
③ The client sends the prompt of $\widetilde{d}$ to the server to query the LLM.
④ The LLM at the server takes the protected prompt $\widetilde{d}$ as input and generates the corresponding response $\widetilde{r}$. Then, the server sends $\widetilde{r}$ to the client.

In our study, black-box LLM is mainly considered by us, whose architecture, model parameters, and inference details are hidden behind commercial APIs and user interfaces. Therefore, protection mechanisms involving encryption and secure multi-party computation do not apply to our framework. To protect the privacy of prompts querying a black-box LLM, randomization is the representative privacy protection mechanism investigated in the literature. In the next subsection, we elucidate the randomization protection mechanism.

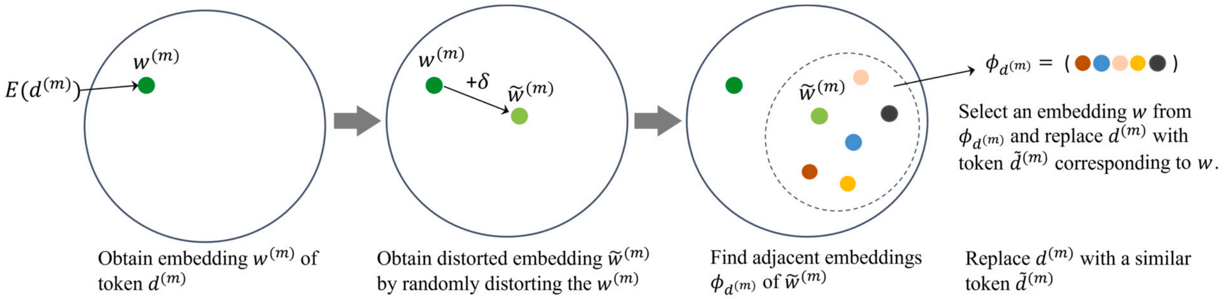

$E(d^{(m)})$
$w^{(m)}$
$w^{(m)}$
$+\delta$
$\widetilde{w}^{(m)}$
$\widetilde{w}^{(m)}$
$\phi_{d^{(m)}} = ($●●●●●$)$
Select an embedding w from $\phi_{d^{(m)}}$ and replace $d^{(m)}$ with token $\tilde{d}^{(m)}$ corresponding to w.
Obtain embedding $w^{(m)}$ of token $d^{(m)}$
Obtain distorted embedding $\widetilde{w}^{(m)}$ by randomly distorting the $w^{(m)}$
Find adjacent embeddings $\phi_{d^{(m)}}$ of $\widetilde{w}^{(m)}$
Replace $d^{(m)}$ with a similar token $\tilde{d}^{(m)}$

Fig. 3. Illustration of randomization protection mechanism. The randomization protection mechanism aims to replace each token in a prompt with a semantically similar token. It typically involves four steps: (1) obtain the embedding $w^{(m)}$ of a given token $d^{(m)}$; (2) add random noise δ to the embedding $w^{(m)}$ to obtain a distorted embedding $\widetilde{w}^{(m)}$; (3) find adjacent embeddings of $\widetilde{w}^{(m)}$, denoted as $\phi_{d^{(m)}}$; (4) Select an embedding w from $\phi_{d^{(m)}}$ and replace $d^{(m)}$ with token $\widetilde{d}^{(m)}$ corresponding to w.

3.3.1. Randomization protection mechanism

Given a tokenizer, a token vocabulary V, and $E \in \mathbb{R}^{|V| \times M}$, an embedding model, where $|V|$ is vocabulary's size and M is embedding's dimension, respectively, we first use tokenizer to turn prompt d into tokens $\{d^{(m)}\}_{m=1}^{|V|}$, where $d^{(m)} \in V$, and then we employ E to map a token $d^{(m)}$ into an embedding $w^{(m)} = E(d^{(m)})$. The random distortion of a token $d^{(m)}$ can be done by the addition of a random noise δ to $w^{(m)}$: $\widetilde{w}^{(m)} = w^{(m)} + \delta$ and replace $d^{(m)}$ with a token $\widetilde{d}^{(m)}$ whose embedding is close to $\widetilde{w}^{(m)}$, the procedure of which is illustrated in Fig. 3.

The literature has proposed various methods to compute δ and choose a semantic similar token $\widetilde{d}^{(m)}$ for $d^{(m)}$ based on the distorted embedding $\widetilde{w}^{(m)}$ of $w^{(m)}$. Herein, we give two examples:

- Applying d_χ-privacy to perturb token $d^{(m)}$ [6]. Given dimension π and privacy parameter η, the random noise δ is computed by $\delta = lv$, where the scalar $l \sim$ Gamma distribution $\Gamma(\pi, \frac{1}{\eta})$ and the uniform vector v is selected at random in accordance with a uniform distribution across the entire unit ball $\mathbb{B}^\pi$. We first compute its embedding $w^{(m)} = E(d^{(m)})$ and random noise embedding $\widetilde{w}^{(m)} = w^{(m)} + \delta$. A token $d^{(m)}$ is replaced with a semantically similar token $\widetilde{d}^{(m)}$ by solving the optimization problem: $\widetilde{d}^{(m)} = \text{argmin}_k \, ||w^{(m)} - \widetilde{w}^{(m)}||$. By repetitively replacing each token $d^{(m)}$ in d with $\widetilde{d}^{(m)}$, we obtain a privatized textual version $\widetilde{d}$ of the original prompt d.
- Applying the random adjacent list to perturb $d^{(m)}$ [5]. Given a token $d^{(m)}$, we first compute its embedding $w^{(m)} = E(d^{(m)})$ and random noise embedding $\widetilde{w}^{(m)} = w^{(m)} + \delta$. The random adjacency list of $d^{(m)}$ is composed of any token $d^{(k)} \in V, k \neq m$ whose embedding $w^{(k)}$ has a Euclidean distance to $w^{(m)}$ that is shorter than Euclidean distance between $w^{(m)}$ and $\widetilde{w}^{(m)}$. We denote the random adjacent list of $d^{(m)}$ as $\phi_{d^{(m)}}$. By repetitively replacing each token $d^{(m)}$ of d with a token in $\phi_{d^{(m)}}$, we obtain a privatized textual version $\widetilde{d}$ of d.

We now proceed to provide formal definitions for the fundamental concepts imposed by the privacy-preserving mechanism $\mathcal{M}$ and the attacking mechanism $\mathcal{A}$.

3.4. Utility loss and privacy leakage

The original prompt is protected by protection mechanism $\mathcal{M}$ perturbing its embedding, which affects both the loss of utility and leakage of privacy of the prompt (see Section 3.3), and the attacking method $\mathcal{A}$ typically reconstructs tokens of the original prompt based on the observed (i.e., protected) prompt (see Section 3.2), which determines the privacy leakage. In this section, we formally defined privacy leakage and utility loss.

Definition 3.1 *(Privacy Leakage).* Let $\widetilde{P}$ denote protected embedding's distribution of the client's prompt, and $\check{P}$ denote the distribution of embedding that is independent of the client's prompt. We define privacy leakage as the measurement that quantifies the gap between the recovery extent of the private information when the distorted prompt is provided and the random prompt is provided to the attacker:

$$\epsilon_p = R(\widetilde{P}) - R(\check{P}) \tag{1}$$

where $R(\widetilde{P}) = \mathbb{E}_{w \sim \widetilde{P}}[R(w)]$ and $R(\check{P}) = \mathbb{E}_{w \sim \check{P}}[R(w)]$. The $R(w)$ is the recovery extent R on the embedding w of prompt d and we define it as

$$R(w) = 1 - \frac{1}{I} \sum_{i=1}^{I} \frac{|| \frac{1}{|d|} \sum_{m=1}^{|d|} (d_i^{(m)}(w) - d^{(m)}) ||}{\Omega}, \tag{2}$$

where $d^{(m)}$ represents prompt d's original m-th token, $d_i^{(m)}(w)$ represents the m-th token inferred by the attacker at iteration i upon observing the embedding w, and I represents the total number of the iterative process of the attacking approach launched by the attacker.

Remark. (1) $R(\check{P})$ acts as the baseline scenario where the attacker attempts to recover the original prompt through random guessing. In essence, if the attacker achieves a recovery extent close to $R(\check{P})$, it indicates that the privacy leakage incurred by the attacker is approaching 0.
(2) We assume that $||d_i^{(m)} - d^{(m)}|| \in [0, \Omega]$, where Ω is the upper bound of the distance between tokens and it is a positive constant. Therefore, $\epsilon_p \in [0, 1]$.
(3) When all estimated tokens $d_i^{(m)}$ are equal to the original token $d^{(m)}$ for every index m ranging from 1 to the length of the token sequence $|d|$, it indicates that there is no differential privacy protection applied, and the privacy leakage is at its maximum (quantified as 1).
(4) Without loss of generality, we posit privacy parameter I is greater than 0. If I is equal to 0, it implies that no privacy protection is enforced, and the privacy loss for any token w will be 0.

Assumption 3.1. We assume that $R(\widetilde{w}) - R(w) \geq R(\widetilde{w})/c$ $\forall w \in \mathcal{W}$ and $\widetilde{w} \in \widetilde{\mathcal{W}}$, where $c > 0$ represents a constant.

This following definition captures the discrepancy in utility between using unprotected model information and using model information with privacy-preserving mechanisms, and it serves as a quantitative measure of the impact of privacy protection on the utility of the model inference.

Definition 3.2 *(Utility Loss)*. The decrement in utility reflects the contrast between the utility achieved through the model's information in the absence of privacy-preserving measures, adhering to the original distribution P, and the utility achieved when privacy safeguards are implemented, conforming to the secured distribution $\widetilde{P}$. Mathematically, it is defined as:

$$\epsilon_u = U(P) - U(\widetilde{P}),$$

where P and $\widetilde{P}$ represent the distributions of models without and with a privacy protection mechanism, respectively. The expected utility $U(P) = \mathbb{E}_{s \sim P_0} \mathbb{E}_{w \sim P} U(w, s)$ is calculated with respect to a test dataset $s \sim P_0$, where P_0 is the distribution of the test dataset. The utility function $U(w, s)$ quantifies the usefulness or effectiveness of the model w on the test dataset s.

3.5. Privacy-preserving LLM inference optimization

During inference of privacy-protecting LLM, the client aims to find a prompt protected by a protection mechanism, with the goal of minimizing utility loss given a privacy budget. Simultaneously, the server wants to infer confidential details of the client from protected prompt. We formulate the optimization problem of the privacy-preserving LLM inference as follows.

$$\begin{aligned} d^* &= \underset{d}{\arg\min}\, \epsilon_u(d|\mathcal{M}) \\ \text{s.t. } & \epsilon_p(d|\mathcal{M}, \mathcal{A}) \leq \xi, \end{aligned} \tag{3}$$

where $\mathcal{M}$ is the privacy protection mechanism applied by the client to protect its original prompt d, $\mathcal{A}$ is the attacking approach leveraged by the server to reconstruct d based on the observed prompt $\widetilde{d}$ sent by the client, and ξ is the constraint of leakage of privacy; ϵ_u is utility loss of d, affected by the $\mathcal{M}$; ϵ_p is the privacy leakage on d, which is affected by both $\mathcal{M}$ and $\mathcal{A}$.

The LLM inference problem formulated in Eq. (3) aims to find a prompt d^* that achieves the minimal utility loss while keeping the privacy leakage under an acceptable constraint ξ. Within the privacy-preserving LLM inference's framework, the problem to achieve minimal privacy leakage and utility loss simultaneously raises a fundamental question: Are there any possibility of designing protective mechanism to fulfill both objectives? In the forthcoming section, we address this question by presenting a counterargument in specific scenarios. We are thus motivated to introduce the notion of no free lunch in the realm of privacy-protected large language model inference, as detailed in Theorem 4.4. The theorem establishes the inherent equilibrium between private information and utility in LLM inference, highlighting challenges in simultaneously minimizing utility loss and privacy leakage. By examining the trade-off, valuable insights into the limitations and complexities associated with achieving optimal privacy and utility guarantees are derived in LLM inference scenarios.

4. No free lunch theorem for privacy-preserving LLM inference

The fundamental concept behind privacy-preserving LLM inference revolves around perturbing the embedding of the original prompt d and transmitting the perturbed embedding $\widetilde{w}$ as the prompt $\widetilde{d}$ to the server, ensuring the privacy of d. In this process, the utility loss is employed to quantify the increase in the expected average loss caused by the perturbed embedding $\widetilde{w}$ compared to the utility provided by w. It is worth noting that $\widetilde{w}$ and w follow the distributions $\widetilde{P}$ and P respectively. By considering these factors, we delve into the No Free Lunch Theorem for Privacy-Preserving LLM Inference, which is the focus of this article. This theorem sheds light on the inherent equilibrium between utility and private information in LLM inference, emphasizing challenges associated with simultaneously minimizing utility loss and privacy leakage. By exploring this trade-off, we gain a deeper understanding of the intricacy of securing optimal standard of privacy with compromising least utility of LLM inference mechanisms.

In this context, our objective is to investigate the limitations of randomization-based privacy-preserving mechanisms. To accomplish our goal, it is essential to measure the unavoidable decrement in utility inherent in the process of safeguarding privacy. Logically, as we apply greater changes to the original prompt w, we enhance the level of privacy preservation, but at the same time, we compromise the accuracy of the resulting output. To assess the degree of distortion, we employ Euclidean distance of tokens of the original prompt and protected prompt's tokens. This distance serves as a crucial component that connects the notions of utility loss and privacy leakage. By incorporating this distance metric, we establish a framework for understanding the equilibrium of utility loss and privacy preservation in context of randomization-based privacy-preserving LLM inference. This framework forms the basis for the formulation and exploration of the No Free Lunch Theorem for Privacy-Preserving LLM Inference, which is the central focus of our article.

Definition 4.1 *(Optimal Embedding)*. Let w^* denote the embedding of the prompt d^* that maximizes the utility. Specifically,

$$w^* = \underset{w \in \mathcal{W}}{\operatorname{argmax}}\, U(w),$$

where $U(w) = \mathbb{E}_{s \sim P_o} U(w, s)$ is the expected utility computed on a test dataset s sampled from distribution P_0, and $\mathcal{W}$ is the union of the support $\widetilde{P}$ and the support of P.

Subsequently, we define the notion of near-optimal embedding, denoted by $\mathcal{W}_c$. Here, $\widetilde{\mathcal{W}}$ represents the set of possible protected embeddings, and c is a non-negative constant. The near-optimal prompt embedding consists of those embeddings from $\widetilde{\mathcal{W}}$ for which the difference in utility, denoted by $U(w^*) - U(w)$, is no greater than c for all possible optimal embeddings w^* in the set $\mathcal{W}^*$. In other words, the near-optimal embedding captures the embeddings that result in a utility loss within the tolerance specified by c when compared to all possible optimal embeddings.

Definition 4.2 *(Near-optimal Embedding)*. Suppose $\widetilde{\mathcal{W}}$ is the support of the protected embedding's distribution. Let c be a constant which is no smaller than 0, we define the *near-optimal prompt embedding* $\mathcal{W}_c$ as the subset of $\widetilde{\mathcal{W}}$ that satisfies the following condition: for any optimal embedding w^* in the set $\mathcal{W}^*$, the difference in utility between w^* and any embedding w in $\mathcal{W}_c$ is no greater than c, i.e.,

$$\mathcal{W}_c = \left\{ w \in \widetilde{\mathcal{W}} : \left|U(w^*) - U(w)\right| \le c, \forall w^* \in \mathcal{W}^* \right\}.$$

In other words, near-optimal prompt embedding $\mathcal{W}_c$ consists of the embeddings from $\widetilde{\mathcal{W}}$ that exhibit a utility difference of at most c compared to any optimal embedding w^* in $\mathcal{W}^*$. This definition allows us to identify a set of embeddings that are close in utility to the optimal embeddings while providing a level of privacy protection.

Assumption 4.1 ensures that there is a limit on the density of embeddings within a certain distance from the optimal embeddings. This prevents situations where the utility function lacks variability and fails to distinguish between optimal embeddings and a subset of embeddings.

Assumption 4.1. Let c in $\mathcal{W}_c$ (see Definition 4.2) be α, the *maximum* constant s.t.

$$\int_{\widetilde{\mathcal{W}}} \widetilde{p}(w) \mathbb{1}\{w \in \mathcal{W}_\alpha\} dw \le \frac{\mathrm{TV}(P || \widetilde{P})}{2}, \tag{4}$$

where $\widetilde{p}$ is distorted embedding $\widetilde{w}$'s probability density, α is a positive constant.

Remark. In the above assumption, we state that α is the maximum constant that satisfies the inequality involving total variation distance between the true distribution P and the distorted distribution $\widetilde{P}$. Such inequality gives a bound to near-optimal parameters' cumulative density, as delineated in Definition 4.2. We assume that α is positive, indicating that there exists a non-zero tolerance for the utility loss.

Definition 4.3 *(Distortion Extent)*. Let $d^{(m)}$ and $\widetilde{d}^{(m)}$ be original prompt d's and protected prompt $\widetilde{d}$'s m-th token, respectively. Then, the distortion extent is defined as

$$\Delta = || \frac{1}{N} \sum_{m=1}^{N} g(d^{(m)}) - \frac{1}{N} \sum_{m=1}^{N} g(\widetilde{d}^{(m)}) ||, \tag{5}$$

where $N = |d|$ and $|| \cdot ||$ is the Euclidean distance.

Remark. Assuming general applicability, we posit that $\Delta \le 1$.

Assumption 4.2 *(Bi-Lipschitz Condition).* For any two prompts d_1 and d_2, it is assumed that there exist constants c_a and c_b s.t.:

$$c_a \cdot ||g(d_1) - g(d_2)|| \leq ||d_1 - d_2|| \leq c_b \cdot ||g(d_1) - g(d_2)||,$$

where $g(d)$ represents the encoding of prompt d, and $|| \cdot ||$ denotes a norm (e.g., Euclidean norm).

The above assumption establishes a bi-Lipschitz condition between the distance in the prompt space and the distance in the encoded space. It implies that the encoding function $g(\cdot)$ preserves the relative distances between prompts up to a scaling factor within a certain range determined by the constants c_a and c_b.

Based on the aforementioned context, we introduce Assumption 4.3, which assumes that the cumulative regret in the privacy-preserving LLM inference process is self-bounded. Specifically, a polynomial function of number of learning rounds I bounds cumulative regret. The bound is defined as $\Theta(I^p)$, where p is a positive exponent. Difference between utility function evaluations of reconstructed data points $d_i^{(m)}$ and original data points $d^{(m)}$ is measured as regret. The constants c_0 and c_2 represent positive values that determine the bounds of the regret.

Assumption 4.3 *(Self-bounded Regret).* Let I represent the overall count of learning iterations conducted by the semi-honest adversary. We suppose cumulative regret satisfies the following condition:

$$c_0 \cdot I^p \leq \sum_{i=1}^{I} ||g(d_i^{(m)}) - g(d^{(m)})|| \triangleq \Theta(I^p) \leq c_2 \cdot I^p, \tag{6}$$

where c_0 and c_2 are positive constants, $d^{(m)}$ represents the m-th data point in the dataset d, and $d_i^{(m)}$ denotes the m-th recovered data point generated at iteration i through the application of the optimization algorithm.

This assumption states that the cumulative regret is bounded by a polynomial function of I. The polynomial function is characterized by the exponent p, and the bounds are determined by the positive constants c_0 and c_2. The assumption implies that the attacker achieves regret that scales polynomially with I, indicating that the attacker exploits an asymptotically optimal regret for gradient-matching.

The provided assumption establishes a performance guarantee for the optimization algorithm used by the attacker and suggests that many classical gradient-matching optimizers satisfy this self-bounded regret condition.

Remark. In general, Assumption 4.3 indicates that the attacker will exploit asymptotically optimal regret for gradient-matching. This assumption is reasonable in practice as many classical gradient-matching optimizers satisfy it. The following examples illustrate this point:

Example 1. The AdaGrad algorithm [30] achieves an optimal regret bound of $\Theta(\sqrt{I})$ given by $O(\max \log h, h^{1-\beta/2}\sqrt{I})$, where $\beta \in (1, 2)$ and h is the dimension of the data. In this case, $p = 1/2$.

Example 2. The Adam algorithm [31] achieves an optimal regret bound of $\Theta(\sqrt{I})$ given by $O(\log h\sqrt{I})$ with an improved constant, where h is the dimension of the data. In this case, $p = 1/2$.

These examples demonstrate that well-known optimization algorithms like AdaGrad and Adam satisfy Assumption 4.3 and achieve optimal regret bounds with specific values of p. This supports the practicality and applicability of the assumption in various scenarios.

In the following Lemma 4.1, we establish a theoretical relationship between privacy leakage and the extent of distortion in the embedding process. Assuming the Lipschitz continuity property and certain bounds on the optimization algorithm, and that a function of the number of rounds bounds expected regret of attacker's optimization algorithm. We propose a lower bound on the privacy leakage, indicating that the reconstruction of the original prompt from the protected prompt is limited. This result offers significant understanding of the trade-off between privacy and distortion in presence of a semi-honest attacker.

Lemma 4.1 *(Theoretical Relationship between Recovery Extent and Distortion Extent). Let Assumption 4.2 hold. The semi-honest attacker employs an optimization algorithm to conduct attack in order to infer the original prompt d belonging to client from $\widetilde{d}$. Suppose embedding of d is w. Let $d^{(m)}$ be m-th token of d, $\widetilde{d}^{(m)}$ represent m-th token of $\widetilde{d}$. Let the distortion extent of the embedding be introduced in Definition 4.3. Let Assumption 4.3 hold. That is, the total number of rounds I of the optimization algorithm is the expected regret, denoted as $\Theta(I^p)$. We have that*

$$R(w) \geq 1 - \frac{c_b \Delta + c_2 \cdot c_b I^{p-1}}{\Omega}, \tag{7}$$

where c_2 is introduced in Assumption 4.3, c_b is introduced in Assumption 4.2, and Ω is introduced in Definition 3.1.

The following Lemma 4.2 establishes a theoretical relationship between privacy leakage and total variation distance in the context of prompt distortion. By considering prompt embedding's distributions before and after distortion, we derive a lower bound on leakage of the privacy measured by ϵ_p. This is expressed with respect to these distributions' total variance distance. This result provides a theoretical interconnection between privacy and TV distance, indicating that as the total variation distance increases, the privacy leakage also increases. The derived bound involves constants related to the Lipschitz continuity property, the optimization algorithm, and the overall distortion extent. This analysis contributes to the understanding of the interplay between privacy and utility in the presence of prompt distortion.

Lemma 4.2 *(Theoretical Relationship between Privacy Leakage and Total Variation Distance). Let ϵ_p be defined in Eq. (1). Let $\widetilde{P}$ and $\breve{P}$ represent the distorted prompt embedding distribution and the independent random prompt embedding distribution. Given the preceding conditions, it follows*

$$\epsilon_p \geq C_1 \cdot TV(\widetilde{P}||\breve{P}),$$

where $C_1 = \frac{\left(1-\frac{c_b+c_2\cdot c_b I^{p-1}}{\Omega}\right)}{c}$, c_2 is introduced in Assumption 4.3, c_b is introduced in Assumption 4.2, and c is introduced in Assumption 3.1. For an in-depth examination, consult Appendix B on the nexus between privacy and total variation.

In the following Lemma 4.3 establishes a theoretical relationship between utility loss and the total variation distance in the context of prompt distortion. Considering prompt embedding's distributions before and after distortion, we derive a lower bound on the utility loss measured by ϵ_u. The bound is expressed with respect to these distributions' total variation distance. This result reveals that as the total variation distance increases, the utility loss also increases. The bound is characterized by the constant α, which is related to the assumptions regarding the distorted embedding. This theorem provides valuable insights into understanding the impact of prompt distortion on the utility of the embedding, highlighting the trade-off between utility preservation and the level of distortion incurred.

Lemma 4.3 *(Theoretical Relationship between Utility Loss and Total Variation Distance). Suppose Assumption 4.1 is valid, and ϵ_u be defined in Definition 3.2. Let P and $\widetilde{P}$ represent embedding's distribution before and after being distorted. Given the preceding conditions, it follows*

$$\epsilon_u \geq \frac{\alpha}{2} \cdot TV(P||\widetilde{P}),$$

where α is introduced in Assumption 4.1. For an in-depth examination, consult Appendix C for detailed analysis.

Upon combining Lemma 4.2 with Lemma 4.3, we deduce a quantitative relation between decrement of utility and leakage of privacy, as articulated in Theorem 4.4.

Theorem 4.4 *(No Free Lunch Theorem for Privacy-Preserving LLM Inference). Suppose ϵ_p be defined in Definition 3.1, and define ϵ_u as in Definition 3.2, given Assumption 4.1, it follows*

$$\frac{C_2}{C_1} \cdot \epsilon_p + \epsilon_u \geq C_2 \cdot TV(P||\breve{P}), \tag{8}$$

in which

- *$C_1 = \frac{\left(1-\frac{c_b+c_2\cdot c_b I^{p-1}}{\Omega}\right)}{c}$, where c_2 is introduced in Assumption 4.3, c_b is introduced in Assumption 4.2, and c is introduced in Assumption 3.1.*
- *$C_2 = \frac{\alpha}{2}$, where α is introduced in Assumption 4.1.*
- *$TV(P||\breve{P})$ is a constant representing the distribution's total variation distance of the undistorted embedding and the distribution's total variation distance of embedding independent of the client's prompt embedding. This constant is independent of the protection mechanisms.*

For an in-depth analysis, see Appendix D.

Theorem 4.4 demonstrates that sum of utility loss and privacy leakage incurred by protecting the prompt is lower bounded by a constant that is contingent on specific problem. This theorem essentially asserts that when interacting with an LLM using protected prompts, the client cannot attain infinitesimal levels of privacy leakage and utility loss simultaneously. Instead, *a trade-off must be made, wherein a reduction in privacy leakage (ϵ_p) is accompanied by a corresponding increase in utility loss (ϵ_u) and vice versa.*

This principle is similar to the No-Free-Lunch theorem for federated learning proposed by Zheng et al. [7], which demonstrated that the clients have to trade-off between privacy and utility when they mutually train a global model in privacy-preserving setting. Our study differs from Zhang et al.'s work in that we study the privacy-utility trade-off during LLM inference, whereby prompts and LLM responses are exchanged between a client and a server. In contrast, Zhang et al. [7] focused on federated learning, during which the clients and the server exchange model parameters and gradients. This difference requires a fundamentally distinct definition of privacy leakage, thereby entailing a novel theoretical analysis of the privacy-utility trade-off.

5. Experiment

In this section, we first introduce InferDPT, a privacy-preserving algorithm for LLM inference proposed by [5], which safeguards user prompt privacy by injecting controlled noise into the embedding vectors of the original prompts, with the aim of exploring the inherent trade-off between privacy leakage and utility loss. Next, we outline the experimental setup and apply our definitions of privacy leakage and utility loss to evaluate how these factors evolve as the level of privacy protection is reduced.

5.1. *InferDPT algorithm overview*

The InferDPT framework is an innovative application of differential privacy [32] in LLM inference, designed to enable privacy-preserving text generation, particularly for remote black-box LLMs. The framework comprises two main components: a perturbation module and an extraction module.

Perturbation Module: The perturbation module protects privacy by introducing noise to the embedding vectors of each token in the original document, leveraging a differential privacy mechanism. The noise level is controlled by the privacy budget, ϵ, with smaller values of ϵ providing stronger privacy guarantees. Additionally, the module generates a random adjacency list for each token, selecting replacement tokens to further strengthen privacy protection. This approach maintains the semantic coherence of the text while ensuring user privacy. The final perturbed document is created by applying these steps to each token in the original text. Subsequently, the user combines the perturbed document with task instructions to form a complete prompt, which is submitted to the remote black-box LLM. The LLM then generates a privacy-preserving output based on the perturbed input.

Extraction Module: The extraction module employs a local LLM, which is lightweight and deployable, while less capable than the remote black-box LLM. This local LLM generates text based on the original document and the privacy-preserving output as a reference.

The InferDPT algorithm offers several key advantages, including effective user prompt privacy protection, resistance to embedding inversion attacks, and its high-quality text generation. Furthermore, the algorithm is highly adaptable to various local LLM and deployment environments. Its modular design—comprising a perturbation module and an extraction module—enhances scalability, accommodating diverse privacy requirements and application scenarios. Specifically, InferDPT enables differential privacy across a wide range of text data scales and complexities. InferDPT incurs lower computational overhead compared to homomorphic encryption approaches (e.g., CipherGPT [24]), effectively balancing computational costs through dynamic sampling. Experimental results show that InferDPT maintains text generation quality on par with non-private GPT-4 across multiple datasets and a range of ϵ values.

However, the InferDPT algorithm also has some limitations. The quality of the generated text depends on the choice of the local LLM and the design of prompt. More importantly, it requires a trade-off between privacy protection and text generation quality, especially when strong privacy guarantees are applied. Additionally, deploying InferDPT requires certain hardware resources to run the local LLM, which may not be user-friendly for ordinary users.

Based on the above analysis, we believe that InferDPT is highly suitable for validating our theory. We will build upon this algorithm to explore the trade-off between privacy leakage and utility loss.

5.2. *Experiment setup*

Dataset: We used the CNN/Daily Mail dataset [33] to perform an open-ended text generation task. Following previous works [5][34][35], perturbed document containing 50 tokens is provided to the remote black-box LLM or the local LLM, with additional task descriptions appended to form a complete prompt (all prompts can be found in 5.3). LLM then generates the following 100 tokens based on this prompt.

Model: We used GPT-3.5-turbo as the remote black-box LLM and Vicuna-7b-4bit as the local LLM. Both models were configured with a temperature setting of 0.5 and a maximum token generation limit (*max_tokens*) of 150.

Quantifying Privacy Leakage ϵ_p: We first randomly select 50 tokens from the vocabulary to simulate the remote black-box LLM's attempt to recover the perturbed document through random guessing. However, we find that directly calculating the recovery extent R as defined in Definition 3.1 is problematic because the black-box LLM's responses do not have a direct correspondence with the perturbation tokens. To address this issue, we use cosine similarity as a measure of the recovery extent. Cosine similarity is defined as $cos(\theta) = \frac{A \cdot B}{|A||B|}$, which is consistent with the definition of recovery extent in Definition 3.1, where $cos(\theta)$ closer to 0 indicates a higher recovery extent, and $cos(\theta)$ closer to 1 indicates a lower recovery extent.

We calculate the cosine similarity $cos(\theta))_1$ between the randomly guessed document and the original document, denoted as $R(\check{P})$, which represents the recovery extent under random guessing. Then, we send the perturbed document to the remote black-box LLM, which attempts to recover the original document, and calculate the cosine similarity $cos(\theta))_2$ between the recovered document and the original document, denoted as $R(\widetilde{P})$, which represents the recovery extent of the remote black-box LLM. According to Definition 3.1, we can obtain ϵ_p. Our expected result is that as the privacy budget ϵ increases, $R(\check{P})$ remains stable, while $R(\widetilde{P})$ gradually increases, indicating that the remote black-box LLM's ability to recover the user's privacy increases, leading to an increase in privacy leakage ϵ_p.

Quantifying Utility Loss ϵ_u: According to our definition of utility loss in 3.2, Utility loss quantifies the difference in performance between LLM inference with and without privacy protection. Specifically, we first send the original document to the remote black-box

LLM for a text continuation task, obtaining 100 generated tokens. We calculate the utility of this output as $U(P)$. Next, we perturb the original document to create a protected version and send it to the remote black-box LLM again, generating another set of 100 tokens. These 100 tokens, along with the original prompt, are then fed into the local LLM to extract information and calculate its utility metrics, denoted as $U(\widetilde{P})$. The utility loss of InferDPT is quantified as the difference between these two utilities.

To evaluate utility, we employ several metrics commonly used in open-ended text generation tasks, including BERTScore [36], BLEU, Keyword Coverage, Semantic Similarity, Diversity, Coherence, and ROUGE (ROUGE-1, ROUGE-2, and ROUGE-L). These metrics are described in detail as follows:

- **BERTScore**: Measures the semantic similarity between the generated and reference texts using BERT embeddings, reflecting content consistency. It is particularly effective for evaluating semantic fidelity, as it captures the contextual meaning of words through pre-trained BERT models.
- **BLEU**: Evaluates the precision of n-grams in the generated text compared to the reference text, providing a measure of fluency and faithfulness. BLEU is widely used in tasks such as machine translation and text generation.
- **Keyword Coverage**: Assesses the extent to which the generated text retains key concepts or terms from the reference text. This ensures that essential information or keywords are accurately represented in the output.
- **Semantic Similarity**: Beyond BERTScore, this metric captures deeper contextual and conceptual alignment between the generated and reference texts, providing a complementary perspective on semantic fidelity.
- **Diversity**: Measures the variety and richness of the generated text, ensuring that the model avoids repetitive outputs. This is particularly important under privacy constraints, which can sometimes reduce variability in the content.
- **Coherence**: Evaluates the logical flow and structural consistency of the generated text, ensuring that the output aligns with the expected narrative or topic and maintains readability.
- **ROUGE Scores (ROUGE-1, ROUGE-2, and ROUGE-L)**: Measure the overlap of n-grams between the generated and reference texts. ROUGE-1 focuses on unigram overlap, ROUGE-2 on bigrams, and ROUGE-L considers the longest common subsequences. These metrics are widely used for assessing textual accuracy, coverage, and relevance.

We divided privacy protection into 24 distinct levels and calculated both privacy leakage and utility loss for each level. By plotting privacy leakage on the x-axis and utility loss on the y-axis, we mapped points corresponding to different privacy budgets or noise intensities. As expected, utility loss decreases gradually as privacy leakage increases, indicating a reduction in privacy protection strength.

5.3. Design of prompts for the InferDPT framework

In our study, we employed different quantification strategies for measuring privacy leakage and utility loss, which required distinct prompt designs. Specifically, for quantifying privacy leakage, we tasked the remote black-box LLM with attempting to recover the perturbed documents. For quantifying utility loss, we instructed the remote black-box LLM to complete continuation tasks based on the input text, and then used the local LLM to extract relevant content from the original document.

5.3.1. Prompts for quantifying privacy leakage

We designed the following prompt to instruct the remote black-box LLM to act as an attacker attempting to recover the perturbed documents:

You are a text recovery expert. Your task is to recover the original tokens that were replaced by their semantically similar alternatives.
Examples:

- *Original: The company announced record profits in the third quarter.*
  Perturbed: Business speak happy money three time part.
  Recovered: The business reported positive financial results in the third period.
- *Original: Scientists discover new species in Amazon rainforest.*
  Perturbed: People find thing green tree place.
  Recovered: Researchers identified a new organism in the tropical forest.

Instructions:

- *Perturbed:*
- *You should only respond with the recovered tokens.*
- *Output format: [token1] [token2] [token3] ...*

5.3.2. Prompts for quantifying utility loss

To quantify utility loss, we first provide the original document to the remote black-box LLM to generate a continuation. Next, the remote black-box LLM is tasked with generating a continuation for the perturbed document. To evaluate the utility loss, we design prompts for the local LLM to extract the continuation content from the remote black-box LLM's responses, using the original document

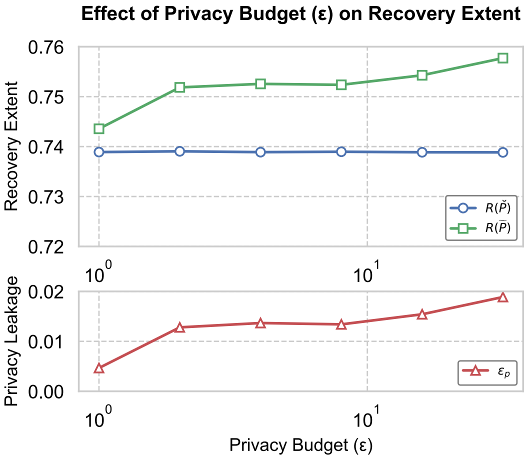


Fig. 4. Privacy Leakage Analysis with Varying Privacy Budget (ϵ). The figure illustrates the relationship between the privacy budget ϵ and two key metrics: recovery extent and privacy leakage. The top plot displays the recovery extents of the random document $R(\breve{P})$ and the perturbed document $R(\widetilde{P})$, while the bottom plot shows the privacy leakage, both as functions of the privacy budget ϵ.

as a reference. Accordingly, this section presents the prompts designed for both the remote black-box LLM's continuation task and the local LLM's extraction task.

Prompt for Remote Black-box LLM:

You are a creative writer skilled in continuing stories and texts in a natural and engaging way. Your task is to generate a coherent continuation that matches the style and context of the original text.

Continue this text in a natural and engaging way, maintaining the same style and tone. Write 2-3 sentences that flow naturally from the original text.

——Original Text:

——[your continuation]

Prompt for Local LLM:

Analyze the coherence between the "Prefix Text" and "Perturbed Generation", then follow these rules:
1. If the "Perturbed Generation" meets ALL these criteria:
- Continues the story/topic from the "Prefix Text" naturally
- Maintains consistent facts and details
- Uses similar tone and style
→ Output the "Perturbed Generation" exactly as is
2. Otherwise, if ANY criteria are not met:
- Write a new continuation that:
- Directly follows from the last sentence of "Prefix Text"
- Matches the style and tone of "Perturbed Generation"
- Maintains factual consistency with "Prefix Text"
OUTPUT ONLY the continuation text, with no explanations or additional text.
——"Prefix Text":
——"Perturbed Generation":
——"Your Continuation":

5.4. Results

According to our definition of privacy leakage Definition 3.1, a recovery extent R closer to 0 indicates a higher amount of recovered private information. As shown in Fig. 4, $R(\widetilde{P})$ increases as the privacy budget ϵ increases, suggesting that a higher ϵ reduces the difficulty of recovering protected prompts, allowing the remote black-box LLM to retrieve more private information from them. In contrast, the $R(\breve{P})$ rate remains relatively stable across different ϵ values, indicating that random guessing does not affect the amount of recovered private information with changes in the privacy budget. This stability serves as a baseline, highlighting the sensitivity of protected prompt recovery to changes in the privacy budget ϵ in remote black-box LLMs.

Additionally, privacy leakage (ϵ_p) increases as ϵ rises, consistent with the expectation that a higher privacy budget allows for greater information leakage. At lower values of ϵ, privacy leakage is minimal, and in some cases even negative, indicating strong privacy protection, where the remote black-box LLM's ability to recover protected prompts is worse than random guessing. However, as ϵ approaches higher values, privacy leakage increases, suggesting that with a larger privacy budget, the remote black-box LLM is able to recover more private information from the protected prompts, thereby increasing the privacy risk.

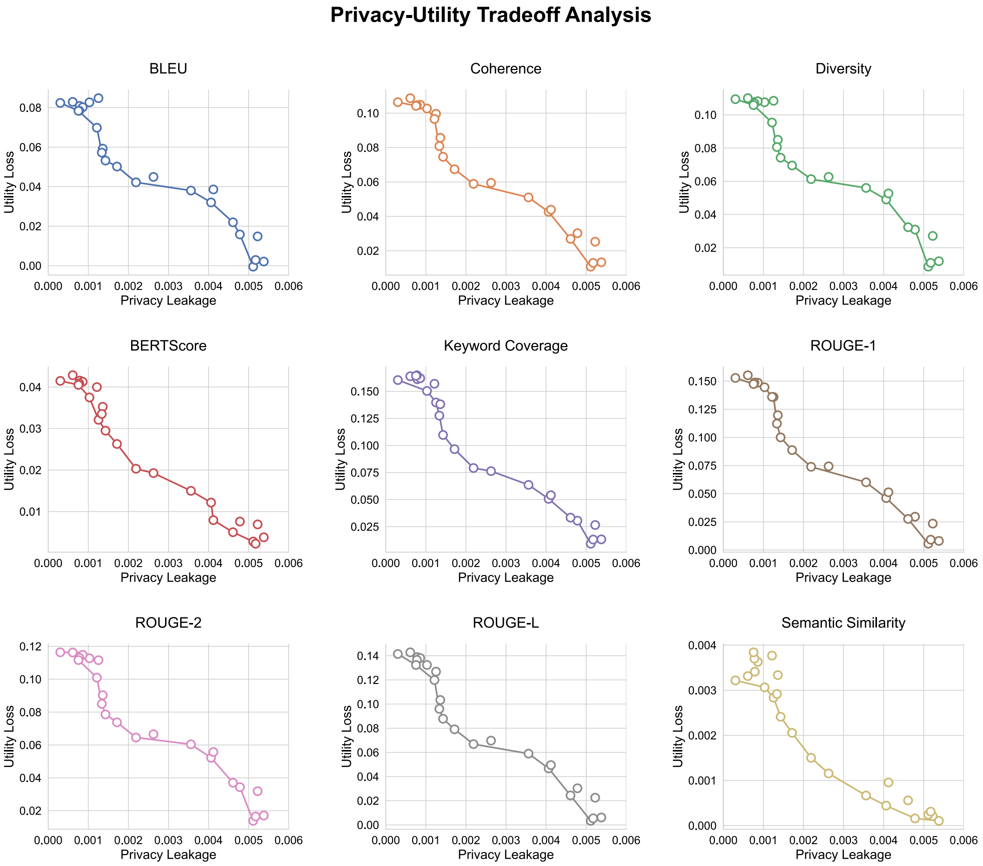


Fig. 5. Privacy and Utility Tradeoff. This figure illustrates the relationship between nine utility loss metrics (vertical axis) and one privacy leakage metric (horizontal axis). Each subplot shows the results for 24 different privacy budgets ϵ, reflecting the corresponding utility loss and privacy leakage. As ϵ decreases, privacy leakage increases while utility loss decreases. This clearly demonstrates the tradeoff between privacy leakage and utility loss.

Although this result is consistent with our defined relationship between privacy leakage and ϵ, it is important to note that we did not use a specialized recovery algorithm for the experiment. Instead, recovery was performed by sending instructions to the remote black-box LLM. We believe that employing a dedicated iterative recovery algorithm could further improve the accuracy of privacy leakage detection.

Moreover, as shown in Fig. 5, this experimental result illustrates the relationship between privacy leakage and utility loss, providing a clear visualization of the trade-off between the two. The figure reveals that as privacy leakage increases—indicating a decrease in the level of privacy protection—all utility metrics (e.g., BERTScore and semantic similarity) exhibit a declining trend. This demonstrates that InferDPT requires a balance between privacy leakage and utility loss: while higher levels of privacy protection effectively reduce privacy leakage, they come at the cost of significant utility loss. Conversely, lower levels of privacy protection improve utility but compromise privacy. Only at an optimal level of privacy protection can both privacy and utility be reasonably balanced. These findings strongly validate our theory that there is an inherent trade-off between privacy leakage and utility loss, aligning well with the “no free lunch” theorem. This discovery offers valuable insights for optimizing privacy-preserving strategies in LLM inference.

6. Conclusion

While LLMs offer tremendous benefits in terms of productivity enhancement and data analysis, their usage also raises concerns about the security and privacy of user queries. Protecting privacy is, therefore, crucial when prompting LLMs for inference, and minimizing privacy leakage is a fundamental requirement. Various randomization approaches have been suggested for safeguarding prompts’ privacy. However, adoption of protection mechanisms may introduce utility loss. This conflict between minimizing privacy leakage and utility loss motivated us to investigate whether developing a protection mechanism that simultaneously achieves utility loss and minimal privacy leakage is plausible.

To answer this question, we propose a privacy-preserving LLM inference framework. Within this framework, we propose and experimentally verified a No-Free-Lunch Theorem for privacy-preserving LLM inference, demonstrating that the weighted summation of these factors exceeds a constant which is contingent on specific problem and non-zero. This highlights inevitable loss of utility when leakage budget of privacy is excessively constrained.

Moving forward, it is essential to continue exploring novel approaches and solutions to strike an optimal balance between privacy and utility. Our future study may concentrate on refining and expanding privacy-preserving LLM inference framework, investigating additional protection mechanisms (e.g., encryption), and evaluating their practical and theoretical effectiveness.

CRediT authorship contribution statement

Xiaojin Zhang: Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Yahao Pang:** Writing – review & editing. **Yan Kang:** Writing – review & editing. **Wei Chen:** Writing – review & editing. **Lixin Fan:** Writing – review & editing. **Hai Jin:** Writing – review & editing. **Qiang Yang:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank Yulin Fei for helpful comments. This work was supported by the National Science and Technology Major Project under Grant 2022ZD0115301.

Appendix A. Analysis for Lemma 4.1

Lemma A.1 *(Theoretical Relationship between Recovery Extent and Distortion Extent). Let Assumption 4.2 hold. The semi-honest attacker uses an optimization algorithm to infer the original prompt d of the client based on the protected prompt $\widetilde{d}$. Let w and $\widetilde{w}$ represent the embeddings of d and $\widetilde{d}$, respectively. Let $d^{(m)}$ represent m-th token of d, $\widetilde{d}^{(m)}$ represent m-th token of $\widetilde{d}$. Let $\Delta = \|\frac{1}{|d|}\sum_{m=1}^{|d|} g(d^{(m)}) - \frac{1}{|d|}\sum_{m=1}^{|d|} g(\widetilde{d}^{(m)})\|$ represent the distortion of the embedding. Let Assumption 4.3 hold. That is, the expected regret of the optimization algorithm in a total of I, $I > 0$, rounds is $\Theta(I^p)$. We have that*

$$R(w) \geq 1 - \frac{c_b\Delta + c_2 \cdot c_b I^{p-1}}{\Omega}, \tag{A.1}$$

where c_2 is introduced in Assumption 4.3, and c_b is introduced in Assumption 4.2.

Proof. Recall that

$$R(\widetilde{w}) = 1 - \frac{1}{I}\sum_{i=1}^{I} \frac{||\frac{1}{|d|}\sum_{m=1}^{|d|}(d_i^{(m)}(\widetilde{w}) - d^{(m)})||}{\Omega}. \tag{A.2}$$

To protect privacy, the client selects a protection mechanism, which maps the original parameter w to a protected parameter $\widetilde{w}$. After observing the protected parameter, a semi-honest adversary infers the private information using the optimization approaches. Let $d_i^{(m)}$ represent the reconstructed data at iteration i using the optimization algorithm. Therefore the cumulative regret over I rounds

$$\begin{aligned}\sum_{i=1}^{I}[||g(d_i^{(m)}(\widetilde{w})) - w|| - ||g(d) - w||] &= \sum_{i=1}^{I}[||g(d_i^{(m)}(\widetilde{w})) - g(d)||] \\ &= \Theta(I^p).\end{aligned}$$

Therefore, we have

$$c_0 \cdot I^p \leq \sum_{i=1}^{I} ||g(d_i^{(m)}(\widetilde{w})) - g(d)|| = \Theta(I^p) \leq c_2 \cdot I^p, \tag{A.3}$$

where c_0 and c_2 are constants independent of I.

Let x and x' represent two data. From our assumption, we have that

$$c_a||g(x) - g(x')|| \leq ||x - x'|| \leq c_b||g(x) - g(x')||. \tag{A.4}$$

Let $d_i^{(m)}(\widetilde{w})$ represent the reconstructed m-th data at iteration i using the optimization algorithm. We have that

$$\begin{aligned}||\frac{1}{|d|}\sum_{m=1}^{|d|}(d_i^{(m)}(\widetilde{w}) - d^{(m)})|| &\leq ||\frac{1}{|d|}\sum_{m=1}^{|d|}(\widetilde{d}^{(m)} - d^{(m)})|| + ||\frac{1}{|d|}\sum_{m=1}^{|d|}(d_i^{(m)}(\widetilde{w}) - \widetilde{d}^{(m)})|| \\ &\leq c_b \cdot ||\frac{1}{|d|}\sum_{m=1}^{|d|}(g(\widetilde{d}^{(m)}) - g(d^{(m)}))|| + c_b||\frac{1}{|d|}\sum_{m=1}^{|d|}(g(d_i^{(m)}(\widetilde{w})) - g(\widetilde{d}^{(m)}))||\end{aligned}$$

where the second inequality is due to $||\frac{1}{|d|}\sum_{m=1}^{|d|}(\widetilde{d}^{(m)} - d^{(m)})|| \le c_b \cdot ||\frac{1}{|d|}\sum_{m=1}^{|d|}(g(\widetilde{d}^{(m)}) - g(d^{(m)}))||$ and $||\frac{1}{|d|}\sum_{m=1}^{|d|}(d_i^{(m)} - \widetilde{d}^{(m)})|| \le c_b||\frac{1}{|d|}\sum_{m=1}^{|d|}(g(d_i^{(m)}(\widetilde{w})) - g(\widetilde{d}^{(m)}))||$.

From the definition of distortion extent, we know that

$$\|\frac{1}{|d|}\sum_{m=1}^{|d|} g(d^{(m)}) - \frac{1}{|d|}\sum_{m=1}^{|d|} g(\widetilde{d}^{(m)})\| = \Delta. \tag{A.5}$$

Therefore, we have

$$\begin{aligned}\Omega(1 - R(\widetilde{w})) = \frac{1}{I}\sum_{i=1}^{I} ||\frac{1}{|d|}\sum_{m=1}^{|d|}(d_i^{(m)} - d)|| &\le c_b\Delta + c_b \cdot \frac{1}{I}\sum_{i=1}^{I} ||\frac{1}{|d|}\sum_{m=1}^{|d|}(g(d_i^{(m)}(\widetilde{w})) - g(\widetilde{d}^{(m)}))|| \\ &\le c_b\Delta + c_2 \cdot c_b I^{p-1}.\end{aligned}$$

Note that $c_b + c_b c_2 \le \Omega$. Therefore, we have that

$$R(\widetilde{w}) \ge 1 - \frac{c_b\Delta + c_2 \cdot c_b I^{p-1}}{\Omega}. \quad \square \tag{A.6}$$

Appendix B. Lemma 4.2 quantitative relationship between TV($\widetilde{P}||\check{P}$) and privacy leakage

Lemma B.1 *(Theoretical Relationship between Privacy Leakage and Total Variation Distance). Let ϵ_p be defined in Eq. (1). Let $\widetilde{P}$ and $\check{P}$ represent the distorted prompt embedding distribution and the independent random prompt embedding distribution. Then we have,*

$$\epsilon_p \ge C_1 \cdot TV(\widetilde{P}||\check{P}),$$

where $C_1 = \frac{\left(1 - \frac{c_b + c_2 \cdot c_b I^{p-1}}{\Omega}\right)}{c}$.

Proof. Let $\mathcal{U} = \{w \in \mathcal{W}_p : d\check{P}(w) - d\widetilde{P}(w) > 0\}$, and $\mathcal{V} = \{w \in \mathcal{W}_p : d\check{P}(w) - d\widetilde{P}(w) < 0\}$, where $\mathcal{W}_p$ represents the union of the supports of $\check{P}$ and $\widetilde{P}$.

For any $w \in \mathcal{V}$, the definition of $\mathcal{V}$ implies that $d\widetilde{P}(w) > d\check{P}(w) \ge 0$. Therefore, w belongs to the support of $\widetilde{P}$, which is denoted as $\widetilde{\mathcal{W}}$. Therefore we have that

$$\mathcal{V} \subset \widetilde{\mathcal{W}}. \tag{B.1}$$

Similarly, we have that

$$\mathcal{U} \subset \check{\mathcal{W}}. \tag{B.2}$$

Recall that w represents the embedding, $\check{P}$ represents the distribution of the embedding after being protected, and $\check{P}(w)$ represents the corresponding probability density function.

We define

$$R(w) = 1 - \frac{1}{I}\sum_{i=1}^{I}\frac{||d_i(w) - \check{d}||}{\Omega}. \tag{B.3}$$

The privacy leakage is defined as

$$\epsilon_p = R(\widetilde{P}) - R(\check{P}), \tag{B.4}$$

where $R(\widetilde{P}) = \mathbb{E}_{w\sim\widetilde{P}}[R(w)]$ and $R(\check{P}) = \mathbb{E}_{w\sim\check{P}}[R(w)]$.

Then we have that

$$\begin{aligned}\epsilon_p &= R(\widetilde{P}) - R(\check{P}) \\ &= \mathbb{E}_{w\sim\widetilde{P}}[R(w)] - \mathbb{E}_{w\sim\check{P}}[R(w)] \\ &= \int_{\mathcal{W}} R(w)d\widetilde{P}(w) - \int_{\mathcal{W}} R(w)d\check{P}(w) \\ &= \int_{\mathcal{V}} R(w)[d\widetilde{P}(w) - d\check{P}(w)] - \int_{\mathcal{U}} R(w)[d\check{P}(w) - d\widetilde{P}(w)]\end{aligned}$$

$$\geq \inf_{w\in\mathcal{V}} R(w) \int_{\mathcal{V}} [d\widetilde{P}(w) - d\check{P}(w)] - \sup_{w\in\mathcal{U}} R(w) \int_{\mathcal{U}} [d\check{P}(w) - d\widetilde{P}(w)]$$
$$= \left(\inf_{w\in\mathcal{V}} R(w) - \sup_{w\in\mathcal{U}} R(w)\right) \int_{\mathcal{V}} [d\widetilde{P}(w) - d\check{P}(w)].$$

From the definition of total variation distance, we have

$$\int_{\mathcal{U}} [d\widetilde{P}(w) - d\check{P}(w)] = ||\widetilde{P} - \check{P}||_{\mathrm{TV}}. \tag{B.5}$$

From Lemma A.1, we know that,

$$R(w) \geq 1 - \frac{c_b\Delta + c_2 \cdot c_b I^{p-1}}{\Omega}. \tag{B.6}$$

Therefore, we have that

$$\epsilon_p \geq \left(\inf_{w\in\mathcal{V}} R(w) - \sup_{w\in\mathcal{U}} R(w)\right) \int_{\mathcal{V}} [d\widetilde{P}(w) - d\check{P}(w)]$$
$$\geq \inf_{w\in\mathcal{V}} \frac{R(w)}{c} \int_{\mathcal{V}} [d\widetilde{P}(w) - d\check{P}(w)]$$
$$\geq \inf_{w\in\mathcal{V}} \frac{1}{c}\left(1 - \frac{c_b\Delta + c_2 \cdot c_b I^{p-1}}{\Omega}\right) \cdot \mathrm{TV}(\widetilde{P}||\check{P})$$
$$\geq C_1 \cdot \mathrm{TV}(\widetilde{P}||\check{P}),$$

where the second inequality is due to Assumption 3.1, the third inequality is due to Eq. (B.6), and the fourth inequality is due to $C_1 = \frac{\left(1-\frac{c_b+c_2\cdot c_b I^{p-1}}{\Omega}\right)}{c}$ and $\Delta \leq 1$. □

Appendix C. Analysis for Lemma 4.3: quantitative relationship between TV($P||\widetilde{P}$) and ϵ_u

Lemma C.1 *(Theoretical Relationship between Utility Loss and Total Variation Distance). Let Assumption 4.1 hold, and ϵ_u be defined in Definition 3.2. Let P and $\widetilde{P}$ represent the distribution of the aggregated parameter before and after being protected. Then we have,*

$$\epsilon_u \geq \frac{\Delta}{2} \cdot TV(P||\widetilde{P}).$$

Proof. Let $\mathcal{U} = \{w \in \mathcal{W}_u : d\widetilde{P}(w) - dP(w) > 0\}$, and $\mathcal{V} = \{w \in \mathcal{W}_u : d\widetilde{P}(w) - dP(w) < 0\}$, where $\mathcal{W}_u$ represents the union of the supports of $\widetilde{P}$ and P.

For any $w \in \mathcal{V}$, the definition of $\mathcal{V}$ implies that $dP(w) > d\widetilde{P}(w) \geq 0$. Therefore, w belongs to the support of P, which is denoted as $\mathcal{W}$. Therefore we have that

$$\mathcal{V} \subset \mathcal{W}_u. \tag{C.1}$$

Similarly, we have that

$$\mathcal{U} \subset \widetilde{\mathcal{W}}. \tag{C.2}$$

It is assumed that the utility of the unprotected model information achieves the maximal value at the convergence step. Therefore, we have that

$$\mathcal{W} \subset \mathcal{W}^*. \tag{C.3}$$

Notice that from the definition of $\mathcal{W}^*$, for any $w \in \mathcal{W}$ and $w^* \in \mathcal{W}^*$ we have that

$$U(w^*) \geq U(w). \tag{C.4}$$

Let Δ be a positive constant defined in Assumption 4.1, from Definition 4.2 we have

$$\mathcal{W}_\Delta = \left\{ w \in \widetilde{\mathcal{W}} : \left|U(w^*) - U(w)\right| \leq \Delta, \forall w^* \in \mathcal{W}^* \right\},$$

which implies that for any $w \in \widetilde{\mathcal{W}} \setminus \mathcal{W}_\Delta$ and $w^* \in \mathcal{W}^*$ it holds that

$$\left|U(w^*) - U(w)\right| > \Delta. \tag{C.5}$$

Combining Eq. (C.4) and Eq. (C.5), for any $w \in \widetilde{\mathcal{W}} \setminus \mathcal{W}_\Delta$ and $w^* \in \mathcal{W}^*$ we have

$$U(w^*) - U(w) > \Delta. \tag{C.6}$$

Recall that $\widetilde{P}$ represents the distribution of the aggregated parameter after being protected. Then we have

$$\begin{aligned}
\epsilon_u &= U(P) - U(\widetilde{P})\\
&= \mathbb{E}_{w\sim P}[U(w)] - \mathbb{E}_{w\sim \widetilde{P}}[U(w)]\\
&= \int_{\mathcal{W}} U(w)dP(w) - \int_{\mathcal{W}} U(w)d\widetilde{P}(w)\\
&= \int_{\mathcal{V}} U(w)[dP(w) - d\widetilde{P}(w)] - \int_{\mathcal{U}} U(w)[d\widetilde{P}(w) - dP(w)]\\
&\overset{\spadesuit}{=} \int_{\mathcal{V}} U(w)\mathbb{1}\{w \in \mathcal{W}^*\}[dP(w) - d\widetilde{P}(w)] - \int_{\mathcal{U}} U(w)[d\widetilde{P}(w) - dP(w)]\\
&\overset{\bigstar}{=} \int_{\mathcal{V}} U(w)\mathbb{1}\{w \in \mathcal{W}^*\}[dP(w) - d\widetilde{P}(w)] - \int_{\mathcal{U}} U(w)\mathbb{1}\{w \in \widetilde{\mathcal{W}}\}[d\widetilde{P}(w) - dP(w)]
\end{aligned}$$

where $\spadesuit$ is due to $\mathcal{V} \subset \mathcal{W}_u \subset \mathcal{W}^*$ from Eq. (C.1) and Eq. (C.3), and $\bigstar$ is due to $\mathcal{U} \subset \widetilde{\mathcal{W}}$ from Eq. (C.2).

We decompose $\int_{\mathcal{U}} U(w)\mathbb{1}\{w \in \widetilde{\mathcal{W}}\}[d\widetilde{P}(w) - dP(w)]$ as the summation of $\int_{\mathcal{U}} U(w)\mathbb{1}\{w \in \widetilde{\mathcal{W}}\}\mathbb{1}\{w \in \mathcal{W}_\Delta\}[d\widetilde{P}(w) - dP(w)]$ and $\int_{\mathcal{U}} U(w)\mathbb{1}\{w \in \widetilde{\mathcal{W}}\}\mathbb{1}\{w \notin \mathcal{W}_\Delta\}[d\widetilde{P}(w) - dP(w)]$. Then we have

$$\begin{aligned}
\epsilon_u &= \int_{\mathcal{V}} U(w)\mathbb{1}\{w \in \mathcal{W}^*\}[dP(w) - d\widetilde{P}(w)] - \int_{\mathcal{U}} U(w)\mathbb{1}\{w \in \widetilde{\mathcal{W}}\}[d\widetilde{P}(w) - dP(w)]\\
&\geq \Delta \cdot \left[\mathrm{TV}(P||\widetilde{P}) - \int_{\mathcal{U}} \mathbb{1}\{w \in \widetilde{\mathcal{W}}\}\mathbb{1}\{w \in \mathcal{W}_\Delta\}[d\widetilde{P}(w) - dP(w)]\right]\\
&\geq \Delta \cdot \mathrm{TV}(P||\widetilde{P}) - \Delta \cdot \int_{\widetilde{\mathcal{W}}} \mathbb{1}\{w \in \mathcal{W}_\Delta\}d\widetilde{P}(w)\\
&\geq \frac{\Delta}{2} \cdot \mathrm{TV}(P||\widetilde{P}),
\end{aligned}$$

in which

- the first inequality is due to $U(w) \leq U(w^*)$ for any $w \in \mathcal{W}_\Delta$ and $w^* \in \mathcal{W}^*$ according to Eq. (C.4), and $U(w^*) - U(w) > \Delta$ for any $w \in \widetilde{\mathcal{W}} \setminus \mathcal{W}_\Delta$ and $w^* \in \mathcal{W}^*$ from Eq. (C.6).
- the second inequality is due to $\int_{\mathcal{U}} \mathbb{1}\{w \in \widetilde{\mathcal{W}}\}\mathbb{1}\{w \in \mathcal{W}_\Delta\}[d\widetilde{P}(w) - dP(w)] \leq \int_{\mathcal{U}} \mathbb{1}\{w \in \widetilde{\mathcal{W}}\}\mathbb{1}\{w \in \mathcal{W}_\Delta\}d\widetilde{P}(w) \leq \int_{\widetilde{\mathcal{W}}} \mathbb{1}\{w \in \mathcal{W}_\Delta\}d\widetilde{P}(w)$.
- the third inequality is due to $\int_{\widetilde{\mathcal{W}}} \mathbb{1}\{w \in \mathcal{W}_\Delta\}d\widetilde{P}(w) \leq \frac{\mathrm{TV}(P||\widetilde{P})}{2}$. □

Appendix D. Analysis of Theorem 4.4

With Lemma B.1 and Lemma C.1, it is now natural to provide a quantitative relationship between the utility loss and the privacy leakage (Theorem D.1).

Theorem D.1 *(No free lunch theorem (NFL) for privacy and utility). Let ϵ_p be defined in Definition 3.1, and let ϵ_u be defined in Definition 3.2, with Assumption 4.1 we have that*

$$\frac{C_2}{C_1} \cdot \epsilon_p + \epsilon_u \geq C_2 \cdot TV(P||\breve{P}),$$

where $C_1 = 1 - \frac{c_b + c_2 \cdot c_b I^{p-1}}{\Omega}$ and $C_2 = \frac{\Delta}{2}$.

Proof. From Lemma B.1, we have

$$\epsilon_p \geq C_1 \cdot \mathrm{TV}(\widetilde{P}||\breve{P}) \tag{D.1}$$

Let $C_2 = \frac{\Delta}{2}$. From Lemma C.1, we have

$$\epsilon_u \geq C_2 \cdot \mathrm{TV}(P||\widetilde{P}). \tag{D.2}$$

Combining Eq. (D.1) and Eq. (D.2), we have that

$$\begin{aligned}\frac{C_2}{C_1} \cdot \epsilon_p + \epsilon_u &= C_2 \cdot (\mathrm{TV}(\widetilde{P}||\check{P}) + \mathrm{TV}(P||\widetilde{P})) \\ &\geq C_2 \cdot \mathrm{TV}(P||\check{P}).\end{aligned}$$

Therefore, we have that

$$\frac{C_2}{C_1} \cdot \epsilon_p + \epsilon_u \geq C_2 \cdot \mathrm{TV}(P||\check{P}),$$

where $C_1 = 1 - \frac{c_b + c_2 \cdot c_b I^{p-1}}{\Omega}$ and $C_2 = \frac{\Delta}{2}$. □

Data availability

No data was used for the research described in the article.

References

[1] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, et al., Palm: scaling language modeling with pathways, arXiv preprint, arXiv:2204.02311.

[2] OpenAI, Chatgpt.

[3] B. Zhang, J. Zhu, H. Su, Toward the third generation artificial intelligence, Sci. China Inf. Sci. 66 (2) (2023) 121101.

[4] X. Zhou, Y. Lu, R. Ma, T. Gui, Y. Wang, Y. Ding, Y. Zhang, Q. Zhang, X. Huang, TextObfuscator: making pre-trained language model a privacy protector via obfuscating word representations, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 5459–5473.

[5] M. Tong, K. Chen, Y. Qi, J. Zhang, W. Zhang, N. Yu, Inferdpt: privacy-preserving inference for black-box large language model, arXiv preprint, arXiv:2310.12214.

[6] Y. Li, Z. Tan, Y. Liu, Privacy-preserving prompt tuning for large language model services, arXiv preprint, arXiv:2305.06212.

[7] X. Zhang, H. Gu, L. Fan, K. Chen, Q. Yang, No free lunch theorem for security and utility in federated learning, arXiv preprint, arXiv:2203.05816.

[8] Y. Kang, T. Fan, H. Gu, L. Fan, Q. Yang, Grounding foundation models through federated transfer learning: a general framework, arXiv preprint, arXiv:2311.17431.

[9] Y. Zhu, X. Wang, E. Zhong, N. Liu, H. Li, Q. Yang, Discovering spammers in social networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 26, 2012, pp. 171–177.

[10] R. Staab, M. Vero, M. Balunović, M. Vechev, Beyond memorization: violating privacy via inference with large language models, in: The Twelfth International Conference on Learning Representations, 2024.

[11] C. Qu, W. Kong, L. Yang, M. Zhang, M. Bendersky, M. Najork, Natural language understanding with privacy-preserving bert, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 1488–1497.

[12] J. Morris, V. Kuleshov, V. Shmatikov, A. Rush, Text embeddings reveal (almost) as much as text, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 12448–12460.

[13] H. Li, M. Xu, Y. Song, Sentence embedding leaks more information than you expect: generative embedding inversion attack to recover the whole sentence, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 14022–14040, https://aclanthology.org/2023.findings-acl.881.

[14] K. Kugler, S. Münker, J. Höhmann, A. Rettinger, Invbert: reconstructing text from contextualized word embeddings by inverting the bert pipeline, arXiv preprint, arXiv:2109.10104.

[15] C. Song, A. Raghunathan, Information leakage in embedding models, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, CCS '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 377–390.

[16] H. Li, Y. Song, L. Fan, You don't know my favorite color: preventing dialogue representations from revealing speakers' private personas, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 5858–5870, https://aclanthology.org/2022.naacl-main.429.

[17] C. Song, V. Shmatikov, Overlearning reveals sensitive attributes, arXiv preprint, arXiv:1905.11742.

[18] I. Hayet, Z. Yao, B. Luo, Invernet: an inversion attack framework to infer fine-tuning datasets through word embeddings, in: Findings of the Association for Computational Linguistics: EMNLP 2022, 2022, pp. 5009–5018.

[19] X. Yue, M. Du, T. Wang, Y. Li, H. Sun, S.S.M. Chow, Differential privacy for text analytics via natural text sanitization, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 3853–3866.

[20] D. Li, H. Wang, R. Shao, H. Guo, E. Xing, H. Zhang, MPCFORMER: fast, performant and private transformer inference with MPC, in: The Eleventh International Conference on Learning Representations, 2023.

[21] Y. Dong, W.-j. Lu, Y. Zheng, H. Wu, D. Zhao, J. Tan, Z. Huang, C. Hong, T. Wei, W. Cheng, Puma: secure inference of llama-7b in five minutes, arXiv preprint, arXiv:2307.12533.

[22] X. Liu, Z. Liu, Llms can understand encrypted prompt: towards privacy-computing friendly transformers, arXiv preprint, arXiv:2305.18396.

[23] M. Zheng, Q. Lou, L. Jiang, Primer: fast private transformer inference on encrypted data, arXiv preprint, arXiv:2303.13679.

[24] X. Hou, J. Liu, J. Li, Y. Li, W. jie Lu, C. Hong, K. Ren, Ciphergpt: secure two-party gpt inference, Cryptology ePrint Archive, Paper 2023/1147, https://eprint.iacr.org/2023/1147, 2023.

[25] M. Hao, H. Li, H. Chen, P. Xing, G. Xu, T. Zhang, Iron: private inference on transformers, Adv. Neural Inf. Process. Syst. 35 (2022) 15718–15731.

[26] Y. Ding, H. Guo, Y. Guan, W. Liu, J. Huo, Z. Guan, X. Zhang, East: efficient and accurate secure transformer framework for inference, arXiv preprint, arXiv:2308.09923.

[27] J. Hong, J.T. Wang, C. Zhang, Z. Li, B. Li, Z. Wang, DP-OPT: make large language model your privacy-preserving prompt engineer, arXiv preprint, https://arxiv.org/abs/2312.03724, 2023.

[28] A. Sordoni, X. Yuan, M.-A. Côté, M. Pereira, A. Trischler, Z. Xiao, A. Hosseini, F. Niedtner, N.L. Roux, Joint prompt optimization of stacked LLMs using variational inference, in: Thirty-Seventh Conference on Neural Information Processing Systems, 2023.

[29] X. Shen, Y. Liu, H. Liu, J. Hong, B. Duan, Z. Huang, Y. Mao, Y. Wu, D. Wu, A split-and-privatize framework for large language model fine-tuning, 2023.

[30] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (7) (2011).

[31] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint, arXiv:1412.6980.

[32] C. Dwork, Differential privacy, in: International Colloquium on Automata, Languages, and Programming, Springer, 2006, pp. 1–12.

[33] K.M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, arXiv:1506.03340, https://arxiv.org/abs/1506.03340, 2015.

[34] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, J. Weston, Neural text generation with unlikelihood training, arXiv:1908.04319, https://arxiv.org/abs/1908.04319, 2019.

[35] J. Xu, X. Liu, J. Yan, D. Cai, H. Li, J. Li, Learning to break the loop: analyzing and mitigating repetitions for neural text generation, arXiv:2206.02369, https://arxiv.org/abs/2206.02369, 2022.

[36] T. Zhang, V. Kishore, F. Wu, K.Q. Weinberger, Y. Artzi, Bertscore: evaluating text generation with bert, arXiv:1904.09675, https://arxiv.org/abs/1904.09675, 2020.