SAP: Privacy-Preserving Fine-Tuning on Language Models with Split-and-Privatize Framework

Xicong Shen[1], **Yang Liu**[1,*], **Yi Liu**[2,*], **Peiran Wang**[3], **Huiqi Liu**[1], **Jue Hong**[1], **Bing Duan**[1], **Zirui Huang**[4], **Yunlong Mao**[4], **Ye Wu**[1] and **Sheng Zhong**[4]
[1]Bytedance
[2]City University of Hong Kong
[3]Tsinghua University
[4]Nanjing University
{shenxicong, liuyang.fromthu, liuhuiqi.7, tanzhuo.107, duanbing.0, wuye.2020}@bytedance.com, yiliu247-c@my.cityu.edu.hk, whilebug@gmail.com, huangzirui@smail.nju.edu.cn, {maoyl, zhongsheng}@nju.edu.cn

Abstract

Pre-trained Language Models (PLM) have enabled a cost-effective approach to handling various downstream applications via Parameter-Efficient-Fine-Tuning (PEFT) techniques. In this context, service providers have introduced a popular fine-tuning-based product service known as Model-as-a-Service (MaaS). This service offers users access to extensive PLMs and training resources. With MaaS, users can fine-tune, deploy, and utilize their customized models seamlessly, leveraging a one-stop platform that allows them to work with their private datasets efficiently. However, this service paradigm has recently been exposed to the possibility of leaking users private data. To this end, we identify the data privacy leakage risks in MaaS-based PEFT and propose a Split-and-Privatize (SAP) framework, mitigating the privacy leakage by integrating split learning and differential privacy into MaaS PEFT. Furthermore, we propose Contributing-Token-Identification (CTI), a novel method to balance model utility degradation and privacy leakage. As a result, the proposed framework is comprehensively evaluated, demonstrating a 65% improvement in empirical privacy with only a 1% degradation in model performance on the Stanford Sentiment Treebank dataset, outperforming existing state-of-the-art baselines.

1 Introduction

In recent years, Pre-trained Language Models (PLMs) represented by BERT [Kenton and Toutanova, 2019] and GPT [Brown *et al.*, 2020] have demonstrated powerful text learning capabilities and have been widely used in various fields such as law [Jiang and Yang, 2023], finance [Arslan *et al.*, 2021], and healthcare [Arora and Arora, 2023]. To improve the adaptability of a PLM on these downstream applications, it is necessary to fine-tune it on datasets related to the downstream tasks. Considering that PLMs contain hundreds of millions of parameters, researchers have proposed several Parameter-Efficient-Fine-Tuning (PEFT) algorithms to reduce the cost of secondary training [Ding *et al.*, 2023], such as Low-Rank Adaptation (LoRA) [Hu *et al.*, 2021]. In practice, most users are unable to independently acquire the PLM and perform fine-tuning due to resource or technical constraints, which has given rise to a new business direction known as Model-as-a-Service (MaaS). In MaaS, enterprises (called service providers) with sufficient resources and technical capabilities release PLMs in the form of cloud services and provide customers (i.e., users) with a fine-tuning API so that they can customize their own PLM based on private data.

In the context of utilizing PLM APIs provided by service providers (e.g., OpenAI) for fine-tuning, users are often required to upload private data to the cloud [Chen *et al.*, 2024]. However, this data, such as text and images, frequently contains sensitive information, including personal identifiers and demographic attributes (e.g., age). Directly transmitting raw data to service providers poses a significant risk of privacy leakage, thereby hindering privacy-conscious users from engaging with customized services [Pan *et al.*, 2020]. Therefore, there is an urgent need for a privacy-preserving fine-tuning framework to alleviate privacy concerns and promote the development of PLM customization services.

To address this challenge, existing research has made preliminary attempts in three main directions: text privatization [Qu *et al.*, 2021], differentially private fine-tuning [Wang *et al.*, 2024; Du *et al.*, 2023], and split learning-based fine-tuning [Hong *et al.*, 2024]. While the first two approaches offer different levels of user data protection, they face challenges in effectively balancing the trade-off between utility and privacy due to the introduction of (local) Differential Privacy (DP) noise. For instance, Qu *et al.* in [Qu *et al.*, 2021] proposed a text privatization mechanism based on DP noise, requiring users to locally perturb individual data entries before sharing them with the service provider. However, this approach inevitably degrades the performance of downstream tasks, highlighting the inherent tension between pri-

*Yi Liu and Yang Liu are corresponding authors.

vacy preservation and model utility. In addition, fine-tuning schemes based on split learning are vulnerable to privacy inference attacks. For example, the attackers can utilize text embeddings to infer user privacy information via attribute inference attacks [Du *et al.*, 2023] and embedding inversion attacks [Song and Raghunathan, 2020].

To achieve a better privacy-utility trade-off, we propose a Split-and-Privatize (SAP) privacy-preserving fine-tuning framework based on the split learning architecture. Specifically, to address privacy concerns, inspired by split learning [Vepakomma *et al.*, 2018; Ceballos *et al.*, 2020; Wang *et al.*, 2023b], we first divide the entire PLM into a bottom model and a top model and then send the bottom model to the users while keeping the confidentiality of most parts of the PLM. During fine-tuning, the user feeds locally sensitive data into the bottom model and privatizes the outputs by applying $d\chi$-privacy mechanisms before sending them to the service provider. Furthermore, to maintain downstream tasks' utility performance, we propose a Contributing-Token-Identification (CTI) method to identify the important token representations. By reducing the perturbation to a small number of token representations that are strongly related to the utility task, we significantly improve the utility performance while maintaining a similar level of empirical privacy.

To comprehensively evaluate the effectiveness of the proposed framework, we conduct extensive experiments on four benchmark datasets encompassing both text classification and generation tasks. To assess privacy performance, we employed two state-of-the-art privacy attacks, i.e., embedding inversion attacks [Song and Raghunathan, 2020] and attribute inference attacks [Du *et al.*, 2023], to validate the framework's ability to safeguard data privacy. The experiment results demonstrate that the proposed SAP framework effectively achieves a better privacy-utility trade-off. The contributions of our work can be summarized as follows:

- We design a privacy-preserving fine-tuning framework based on the split learning, called SAP, for PLMs.
- We propose a CTI method to achieve a better privacy-utility trade-off.
- We demonstrate the effectiveness of SAP on four benchmark datasets, where it outperforms state-of-the-art baselines.

2 Related Work

Privacy-Preserving Language Model Fine Tuning. Previous works have discussed privacy concerns and their protection in PLM fine-tuning. Privacy protection methods can be categorized into two main types: post-fine-tuning protection and pre-fine-tuning protection. Post-fine-tuning protection aims to protect the fine-tuned corpus dataset's privacy against fine-tuned model users. The work [Sun *et al.*, 2024] has discussed certain attacks that extract sensitive information from fine-tuned PLMs. For the post-fine-tuning privacy protection, the most common approach is to utilize DP in the process of fine-tuning by adding noise to the gradients [Li *et al.*, 2024a; Charles *et al.*, 2024; Tang *et al.*, 2024; Li *et al.*, 2024b; Pan *et al.*, 2020]. While pre-fine-tuning considers fine-tuning service providers as an additional threat adversary. In this context, to protect the users' privacy, we generally utilize text privatization to alleviate privacy concerns. Common text privatization includes using generated data for fine-tuning [Akkus *et al.*, 2024], and adding DP-based noise to fine-tune the corpus [Qu *et al.*, 2021; Li *et al.*, 2023]. Another type is to utilize Low-Rank Adoption (LoRA) for privacy, including incorporating fully homomorphic encryption into LoRA [Li *et al.*, 2024b] and using the lightweight adapter with a compression emulator [Ji *et al.*, 2024].

Unlike the above works, some works focusing on split learning [Zmushko *et al.*, 2023; Lyu *et al.*, 2020; Wang *et al.*, 2024; Wang *et al.*, 2023b] split the PLM into two parts: the bottom one including an embedding layer for users and the top one including the rest parts for providers. In this way, users do not need to share data with service providers and can easily customize fine-tuning services. Furthermore, to protect the privacy of the text representation of the interaction between users and service providers, DP noise is generally introduced to protect privacy [Du *et al.*, 2023]. However, this line of methods has difficulty achieving a good trade-off due to the introduction of expensive DP noise and ignorance of the importance of tokens. To this end, we propose the CTI method to save the extra resources required for plain token transmission and reconstructing the model training.

Key Token Identification. In traditional classification tasks, token importance can be determined by analyzing word frequency [Schneider, 2004; Chen and Meurers, 2016] within each class, like TF-IDF [Aizawa, 2003; Ramos and others, 2003] and TF-ICF [Reed *et al.*, 2006] or token vector length, like Word2Vec [Church, 2017]. However, this approach is unsuitable for generation tasks, such as question-answering, where classification labels are unavailable. Instead, we leverage a technique called *attention-based token pruning*, originally designed to accelerate transformer inference by reducing the computational cost of attention blocks [Fu *et al.*, 2024; Guo *et al.*, 2024; Kim *et al.*, 2022]. However, previous works have also used this to identify those tokens with higher attention scores as key tokens [Wang *et al.*, 2023a; Liu *et al.*, 2023]. We use attention-based token pruning to identify key tokens since it suits current Large Language Models (LLMs) nature better than previous methods.

3 Background and Threat Model

Split Learning. Split learning (SL) [Vepakomma *et al.*, 2018] is a distributed learning technique that partitions the model into multiple segments distributed across different users, enabling collaborative model training without sharing raw data. In the simplest split learning configuration, known as SplitNN [Romanini *et al.*, 2021], each user trains a partial deep network, referred to as the *bottom model*, up to a designated layer called the *cut layer*. The output from the cut layer, representing the feature representation, is transmitted to the server. The server then completes the remaining training using another partial deep network, referred to as the *top model*, without accessing the users' raw data. This approach allows a complete round of forward propagation to

be performed without sharing the raw data. Subsequently, the gradients are backpropagated in a similar manner: starting from the final layer of the top model, they are propagated back to the cut layer. At this point, only the gradients of the cut layer are sent back to the user to update the bottom models. This process is repeated iteratively, with forward and backward propagations, until the distributed split learning network converges. While SL reduces the risk of exposing raw data, studies have uncovered potential vulnerabilities to privacy leakage through the intermediate representations transmitted by users [Dosovitskiy and Brox, 2016; He *et al.*, 2020]. For instance, [He *et al.*, 2020] introduced attack methods effective in both white-box and black-box scenarios, showing that original inputs can be partially reconstructed from these transmitted representations.

$d\chi$-Privacy Mechanism. Then, we introduce the background and definition of the commonly used text privatization method, namely the $d\chi$-privacy mechanism [Chatzikokolakis *et al.*, 2013]. Specifically, $d\chi$-privacy is a generalization of the concept of Differential Privacy (DP) over some metric space χ, aiming to protect the privacy of data on this space. Formally, $d\chi$-privacy is defined as follows:

Definition 1. *A randomized mechanism $\mathcal{M}$ satisfies $\eta d\chi$-privacy if for any two inputs $x, x' \in \mathcal{X}$,*

$$\frac{\Pr[M(x) = y]}{\Pr[M(x') = y]} \leq e^{\eta d(x,x')}, \forall y \in \mathcal{Y}, \tag{1}$$

where $\eta > 0$ is a privacy parameter and $d(x, x')$ is a distance function (e.g., Euclidean distance).

In existing work, $d\chi$-privacy can be generalized from DP and its variant local DP (LDP) [Dwork, 2006], aiming to protect the privacy of data by introducing a random mechanism. Usually, researchers utilize $d\chi$-privacy generalized from LDP for various applications [Yang *et al.*, 2022]. Compared with the definition of LDP, $d\chi$-privacy replaces the exponent term of the inequality (1) from ϵ to $\eta d(x, x')$, so it is a relaxation of LDP. Furthermore, $d\chi$-privacy allows the indistinguishability of the output distributions to be scaled by the distance between inputs, which enables the randomized mechanism to retain more information about the input. Thus, $d\chi$-privacy can provide a better privacy-utility trade-off in the privacy-preserving LLM fine-tuning task.

Threat Model and Design Goals. In this paper, we assume that the service provider is honest but curious; that is, it always follows the designed fine-tuned protocol but is curious about the private information of participants (i.e., users' private data). Specifically, the adversary can access the intermediate representation transmitted by users and the bottom model parameters of users (i.e., white box setting). It then uses them to infer private information by adopting embedding inversion attacks [Qu *et al.*, 2021] and attribute inference attacks [Song and Raghunathan, 2020]. Given the above threat model, aligned with the previous work [Wang *et al.*, 2023b; Qu *et al.*, 2021], our goal is to achieve 1) the service provider cannot recover the original input text from the transmitted text representation; 2) the proposed framework should maintain comparable performance compared to centralized fine-tuning methods (i.e., non-private methods).

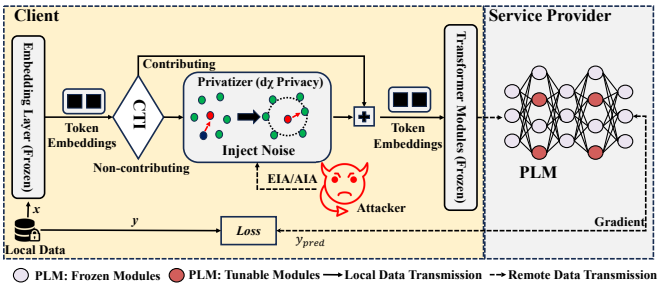


Figure 1: An overview of the SAP framework, where the PLM is split into a bottom model (embedding layer) and a top model.

4 Method

4.1 SAP Workflow

System Model. We consider a PLM fine-tuning scenario where a service provider $\mathcal{S}$ holds a PLM parameterized as $\boldsymbol{w}$ and provides customized fine-tuning services for users $\mathcal{U}$, where each user holds a private fine-tuning text dataset $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)|i = 1, 2, \ldots, |\mathcal{D}|\}$. See Fig. 1 for more details. To achieve optimal adaptation on the downstream task, the service provider and user need to collaboratively fine-tune the PLM, which can be formulated as:

$$\arg\min_{\boldsymbol{\delta}} \mathcal{L}(\boldsymbol{w} + \boldsymbol{\delta}, \mathcal{D}), \tag{2}$$

where $\boldsymbol{\delta}$ is the trainable parameters of the parameter efficient fine-tuning methods, e.g., low-rank adaptation (LoRA) [Hu *et al.*, 2021]. Due to privacy constraints, one party cannot perform the above fine-tuning process in a centralized manner. Specifically, the privacy constraints include that the service provider cannot share the PLM $\boldsymbol{w}$ with the user, and the user cannot share the private dataset $\mathcal{D}$ with the service provider.

Workflow Overview. To protect user privacy, we follow the SplitNN [Vepakomma *et al.*, 2018] architecture to implement fine-tuning. Specifically, the service provider sends the first m layers of the l-layer PLM to the user as the bottom model (including the embedding layer and several encoder blocks) and retains the remaining $(l - m)$ layers as the top model (including the rest of the layer following the head layers). The bottom model was sent to the user before fine-tuning. During the fine-tuning process, the user first computes the forward process using the bottom model to generate an intermediate representation. Next, the user identifies the importance of tokens using our proposed CTI (§4.3) and adds the noise to the representation (§4.2). Then, the representation was sent to the service providers to complete the forward process using the remaining rear layers. Since the sample labels remain with the user, updating trainable parameters in the PLM, such as the LoRA module, requires collaboration. The service provider sends the output back to the user, who computes the gradients of the output layer and returns them to the service provider for parameter updates. The above process is iteratively executed until the PLM converges.

Split Layer Selection Problem. Choosing the split layer in SAP is critical. A bottom model with only an embedding layer reduces user computation but allows easy input recovery via nearest neighbor search [Qu *et al.*, 2021]. Adding

more encoder blocks increases privacy, as higher-layer representations are more abstract [Song and Raghunathan, 2020]. However, since PLM weights are valuable assets, the provider may prefer to limit weight exposure. Thus, selecting the split layer requires balancing privacy, computational burden, and asset protection, discussed further in §5.

4.2 Text Privatization

In the SAP workflow, the user needs to first utilize the bottom model (i.e., the embedding layer) provided by the service provider to extract the text representation of the local private text dataset. However, if plain representations are directly released, the service provider might be able to accurately recover the original input text [Song and Raghunathan, 2020]. Therefore, to achieve stronger privacy protection, it is necessary for the user to employ privatization mechanisms to perturb text representations. We take the case where the bottom model is a frozen embedding layer as an example to demonstrate how to combine the SAP framework with the privatization mechanism proposed in [Feyisetan *et al.*, 2020] to guarantee $\eta d\chi$-privacy.

Let $[x_i^1, x_i^2, \ldots, x_i^n]$ represent a sequence of tokens for the input text $\boldsymbol{x}_i$. The user first obtains the embedding vector $\phi(x_i^j)$ for each token x_i^j in the sample $\boldsymbol{x}_i$ based on the embedding layer. The independent random noise $\boldsymbol{n}$ is added to each embedding vector:

$$\hat{\phi}(x_i^j) = \phi(x_i^j) + \boldsymbol{n}, \quad p(\boldsymbol{n}) \propto \exp(-\eta\|\boldsymbol{n}\|), \tag{3}$$

where $\phi(\cdot)$ is the output of the bottom model. To achieve DP, the perturbed vector should be replaced by its nearest neighbor in the embedding space:

$$\bar{\phi}(x_i^j) = \arg\min_{\boldsymbol{w}_m} \left\| \hat{\phi}(x_i^j) - \boldsymbol{w}_m \right\|, \tag{4}$$

where $\boldsymbol{w}_m$ represents the vector in the embedding space. Finally, the user sends $|\mathcal{D}|$ perturbed sequences $\bar{\phi}(\boldsymbol{x_i}) = [\bar{\phi}(x_i^1), \bar{\phi}(x_i^2), \ldots, \bar{\phi}(x_i^n)]$ to the service provider.

4.3 Contributing Token Identification (CTI)

Although text privatization strengthens the protection of data privacy, fine-tuning PLM on the perturbed representations will inevitably lead to performance degradation on the downstream task [Qu *et al.*, 2021], so there is a trade-off between utility and privacy. To improve the utility-privacy trade-off of the SAP framework, we propose a Contributing Token Identification (CTI) method. Considering that text tasks generally involve two aspects: classification and generation, we design token importance calculation methods for the two types of tasks, respectively. For text classification tasks, important tokens can be identified by analyzing their statistical contributions to each category. However, in generation tasks without categorical labels, such an analysis is not applicable. To address this, we propose using attention scores to determine the importance of tokens in each input sequence. By doing so, we enhance utility performance while maintaining comparable levels of privacy protection, achieved by reducing the perturbations applied to these important tokens. The following is a detailed description of this method.

Utility Importance w/ Classification. In natural language processing, term frequency-inverse document frequency (TF-IDF) [Salton and Buckley, 1988] is a widely used metric to measure the importance of a word within a document relative to its occurrence across a collection or corpus. Inspired by the concept of TF-IDF, we propose a metric that measures the importance of each token in relation to the utility target for text classification tasks. Let $p(t = t_m|y = c)$ represent the frequency of a token t_m appearing in the c-th class of samples; then the utility importance (UI) of a token t_m to a class c is defined as:

$$\mathrm{UI}_{mc} = \frac{1}{N-1} \sum_{c', c' \neq c} \ln \frac{p(t = t_m|y = c)}{p(t = t_m|y = c')}, \tag{5}$$

where $\ln \frac{p(t=t_m|y=c)}{p(t=t_m|y=c')}$ can be regarded as the difference between the probability distribution of tokens in the c-th class of samples and that in the c'-th class of samples specifically at token t_m, and N is the number of categories. Intuitively, tokens that appear frequently in the c-th class of samples while being low-frequency in other classes of samples will be considered to contribute significantly to distinguishing the c-th class from other classes and thus will be assigned a larger UI.

Utility Importance w/o Classification. For text generation tasks, we leverage attention scores to assess the importance of tokens. The key insight is that more important tokens tend to have higher attention scores within the Multi-head Attention (MHA) mechanism [Vaswani *et al.*, 2017]. To formalize this, we first define the relationship between the query, key, and value in MHA as follows:

$$\mathrm{Att}_{\mathbf{W}_{k,q,v,o}}(x) = \mathbf{W}_l \sum_{i=1}^{n} \mathbf{W}_v x_i \, softmax(\frac{x^\top \mathbf{W}_q^\top \mathbf{W}_k x_i}{\sqrt{d}}), \tag{6}$$

where n is the number of independent heads, d is the feature dimension, and $\mathbf{W}$ represents the linear parameters in the attention block. Unlike token importance in text classification, which is computed in classes, attention-based token importance is computed in sequence. For an input sequence, we denote the *attention probability* [Kim *et al.*, 2022] of the head h in the layer l between tokens x_i and x_j in sequence c with length n as:

$$A^{(h,l)}(x_i, x_j) = softmax(\frac{x^\top \mathbf{W}_q^\top \mathbf{W}_k x_i}{\sqrt{d}})_{(i,j)}. \tag{7}$$

The attention-based CTI score of a token x_i is then defined as:

$$s^{(l)}(x_i) = \frac{1}{N_h} \frac{1}{n} \sum_{h=1}^{N_h} A^{(h,l)}(x_i, x_j), \tag{8}$$

where N_h refers to the total head numbers. However, merely computing importance through attention score has drawbacks. Attention blocks may sometimes give high scores to simple words with small L_1 norms, such as punctuation marks or conjunctions [Guo *et al.*, 2024]. To improve the accuracy of token importance, we multiply the attention score $s^{(l)}(x_i)$ with the L_1 norm of the corresponding token value:

$$\bar{s}^{(l)}(x_i) = s^{(l)}(x_i) \times \|\mathbf{W}_v x_i\|_1. \tag{9}$$

In this way, the attention scores of punctuation marks or conjunctions will be constrained by the L_1 norm and will not get high scores. Furthermore, the transformer blocks may come from either several bottom layers of the target large model or from a smaller model that has been fine-tuned on the target dataset. Assume there are L layers for CTI computation; then the raw UI for tokens m in sequence c is defined as:

$$\mathrm{UI}_m^{raw} = \frac{1}{L}\sum_{l=1}^{L} \bar{s}^{(l)}(x_m), \tag{10}$$

where $\bar{s}^{(l)}$ denotes averages across layers. Finally, we assign the scores of each token in the sequence c to a normal distribution such as $\mathrm{UI}_m = (\mathrm{UI}_m^{raw} - \mu_c)/\sigma_c$, where μ_c and σ_c are the mean and standard deviation of raw UIs in the sequence c. Intuitively, in a given sequence, important tokens generally play a decisive role in the task and thus are generally given a larger attention probability in MHA.

Adaptive Privatization via UI. After obtaining the importance scores of tokens, we adaptively assign DP parameters to each token based on the scores to achieve better privacy-utility. The above process is formalized as follows:

$$\eta_m = \frac{2\eta_0}{1+\exp(-\mathrm{UI}_m + c_0)}, \tag{11}$$

where η_0 denotes the basic privacy budget and c_0 is a constant. Compared to a fixed privacy budget η_0 for all tokens, the user reduces the perturbation to the embedding vectors of tokens with larger UI values and increases the perturbation to those with smaller UI values, thereby achieving a better utility-privacy trade-off.

5 Experimental Results

5.1 Experiment Settings

Datasets. The effectiveness of the SAP framework is evaluated on both text classification and generation tasks. For the classification task, we use the classic sentiment analysis datasets, i.e., Financial Phrasebank (FP) [Malo *et al.*, 2014] and Stanford Sentiment Treebank (SST) [Wang *et al.*, 2019], and the topic classification dataset, i.e., Blog [Lyu *et al.*, 2020]. For the text generation task, we use the question-answering dataset SQuAD [Rajpurkar *et al.*, 2016].

Models. Roberta-Large, with 355 million parameters, and Llama-3, with 8 billion parameters, are used as pre-trained models, both of which are publicly available[1].

Baselines. SAP is compared to the following baselines:

DP-Forward [Du et al.*, 2023].* This approach aims to directly perturb the embedding matrix in the forward pass of LLMs to meet the strict DP requirements on both training and inference data.

SLDP-FT [Wang et al.*, 2024].* This approach achieves privacy-preserving fine-tuning by perturbing the forward-pass embedding via sequence LDP in split learning.

DP-OPT [Hong et al.*, 2024].* This approach utilized a split learning framework and adopted a local privacy setting, allowing users to privatize their data locally with DP.

[1]https://huggingface.co/

Dataset	Privacy Parameter η_0							
	40	45	50	55	60	65	70	None
FP	93.15	95.15	96.03	96.74	97.21	97.71	97.97	98.62
Blog	92.82	94.21	95.31	95.95	96.43	96.64	96.71	96.97
SST	82.77	87.33	91.25	93.56	94.55	95.07	95.29	95.63

Table 1: Classification performance evaluation results of the Roberta model under different η_0 on the FP, Blog, and SST datasets.

Privacy Attacks. To comprehensively evaluate the privacy performance of the SAP, we employ two types of state-of-the-art privacy inference attacks [Song and Raghunathan, 2020; Qu *et al.*, 2021] to evaluate SAP. To maximize the attacker's abilities, we consider a white-box setting, where the attacker can access the perturbed text representations transmitted by the user as well as the parameters of the bottom model. We elaborate on the above attack as follows:

Embedding Inversion Attacks (EIA). The attack is a token-level attack aimed at recovering the original input text from perturbed text representations. When the bottom model contains only the embedding layer, the attack predicts the original token by finding the nearest neighbor of each perturbed embedding in the embedding space [Qu *et al.*, 2021]. For bottom models with additional layers, a more sophisticated optimization-based attack [Song and Raghunathan, 2020] is used. This method iteratively refines word selection vectors by minimizing the distance between the predicted text's representations and the observed representations for each input sample.

Attribute Inference Attacks (AIA). This attack aims to infer sensitive user attributes from text representations. As described in [Du *et al.*, 2023], the attacker is assumed to have access to privacy attribute labels for a subset of samples, e.g., the author's gender in the Blog dataset. The privacy inference is treated as a downstream task, where a classifier is trained on the text representations and corresponding labels. Once trained, the attacker uses this classifier to predict privacy attributes of other samples based on their text representations.

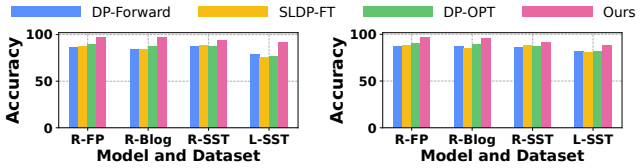
Implementation Details. Our experiments are implemented based on the Transformers library and the PEFT library of Huggingface. Specifically, the LoRA method is adopted to fine-tune the PLM, and the AdamW optimizer with a linear learning rate scheduler is used during fine-tuning, where the initial learning rate is set to 3e-4. Empirically, the constant c_0 in Eq. (11) is set to $(\max(\mathrm{UI}_m) + \min(\mathrm{UI}_m))/2$. We use utility classification accuracy (UA) and F1 scores to evaluate utility performance, respectively. Besides, empirical privacy (EP) [Li *et al.*, 2023] is used as a metric to evaluate privacy protection capability, where empirical privacy is defined as $1 - X$ and X represents the attack success rate.

5.2 Performance Evaluation

Since SAP uses a privacy protection mechanism different from DP techniques, it is difficult to directly compare it with baselines under different noise scales (i.e., ϵ in DP). For this reason, this section only shows the classification and genera-

Dataset	Privacy Parameter η_0							
	500	550	650	650	700	750	800	None
FP	92.16	93.64	95.21	95.89	96.47	97.34	97.58	97.89
Blog	91.55	93.19	93.97	94.47	95.87	96.11	96.45	97.01
SST	81.49	85.25	88.51	90.39	91.68	93.01	94.03	95.74
SQuAD	10.57	13.65	31.24	42.48	53.31	62.45	67.78	82.74

Table 2: The classification performance evaluation results of the Llama model under different η_0 on the FP, Blog, and SST datasets, as well as the generation performance results on the SQuAD dataset.



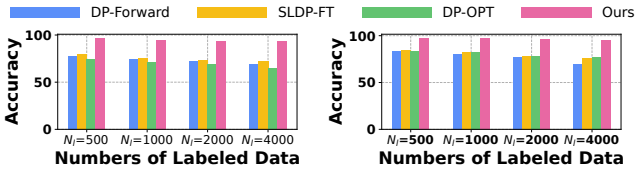
(a) $EP = 30\%$ (b) $EP = 35\%$

Figure 2: Comparison with the existing baselines on EIA.

tion performance of SAP under different privacy parameters η_0. We compare the privacy-utility performance with baselines in the following section.

Performance Evaluation on Classification Tasks. We utilize Roberta and Llama as PLMs to evaluate classification task performance on the FP, Blog, and SST datasets under varying privacy parameters η_0. For the Roberta model, η_0 is set to $\{40, 45, 50, 55, 60, 65, 70\}$, while for Llama, it is set to $\{500, 550, 600, 650, 700, 750, 800\}$. It is important to note that η_0 is determined by $d(x, x')$, where a smaller η indicates stronger privacy protection. The numerical results, summarized in Tables 1 and 2, indicate that the classification performance of both models remains largely unaffected under appropriate privacy protection settings. For instance, with Roberta, the accuracy within the η_0 range of [50, 70] is comparable to the accuracy without privacy protection. This demonstrates that the proposed framework, SAP, can effectively balance privacy and utility in classification tasks. The evaluation of its privacy performance will be discussed in the subsequent section.

Performance Evaluation on Generation Tasks. In our evaluation, we use the Llama model as the PLM within SAP to assess its generation performance on the SQuAD dataset under various privacy parameters. We apply the same privacy parameters used in the classification task for consistency. The experimental results, presented in Table 2, reveal a sharp decline in generation performance when $\eta_0 = 500$. However, performance begins to gradually recover when η is within the range of [700, 800]. We attribute this to the unique nature of generation tasks, where the introduction of the $d\chi$-Privacy mechanism can cause slight semantic alterations in certain tokens, leading to significant changes in the generated content. It is worth noting that existing baselines, such as DP-Forward and DP-OPT, do not currently support generation tasks, highlighting the novelty of our approach in this area.

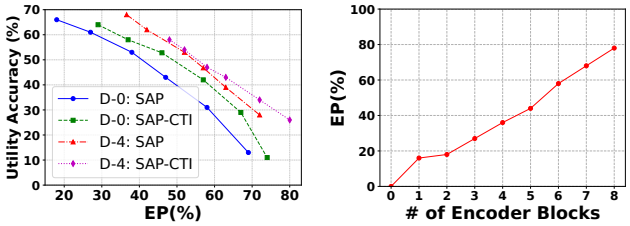


(a) $EP = 30\%$ (b) $EP = 35\%$

Figure 3: Comparison with the existing baselines on AIA.

# of Encoder Blocks	Metric	Privacy Parameter η_0						
		45	50	55	60	65	70	None
2	EP	55.28	46.03	37.46	31.18	25.79	21.42	19.68
	UA	87.21	91.08	93.24	94.38	95.30	95.41	95.84
4	EP	72.15	67.98	59.13	51.42	45.34	40.97	37.80
	UA	87.06	90.38	93.21	94.25	94.89	95.13	95.72
6	EP	80.45	75.22	71.58	68.06	64.83	61.41	59.19
	UA	86.79	90.51	93.16	93.61	94.77	95.06	95.53

Table 3: Empirical privacy against EIA (%) and utility accuracy (%) of SAP-CTI with different split positions and different privacy parameter settings on the Roberta model and SST dataset, where "None" represents the case without privatization.



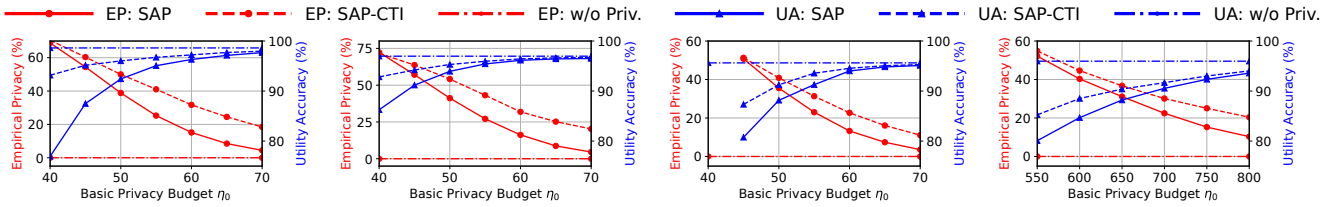
(a) (b)

Figure 4: (a) Utility accuracy versus empirical privacy against EIA on the Llama model and SQuAD dataset. (b) Empirical privacy against EIA of the SAP framework (without privatization) with different split positions on the Roberta model and SST dataset.

5.3 Privacy Evaluation

Comparison with SOTA Baselines. To ensure a fair comparison with the baselines, we evaluate the performance of different methods under the same EP conditions. Additionally, since existing baselines do not support generation tasks, our comparison focuses solely on their privacy-utility tradeoffs in classification tasks.

Defend Against EIA. We assume that the attacker employs an EIA [Song and Raghunathan, 2020] by iteratively optimizing the word selection vector to minimize the distance between the predicted text representation and the observed representation. To assess the privacy performance of SAP and the baselines in defending against EIA, we use the Roberta and Llama models as PLMs within SAP and conduct evaluations on the SST dataset. We set EP at 35% and 30% to examine the performance of SAP and the baselines. The experimental results, illustrated in Fig. 2, indicate that SAP significantly outperforms the baselines. For instance, when EP is 30%, SAP achieves a 12% higher performance than DP-Forward on the FP dataset by using the Roberta model, demonstrating its superior capability in miti-



(a) Roberta, FP (b) Roberta, Blog (c) Roberta, SST (d) Llama, SST

Figure 5: Impact of the privacy parameter η_0 on the EP against EIA and UA.

gating EIA threats.

Defend Against AIA. Fig. 3 presents the results of the SAP framework and baselines defending against AIA on the Blog dataset with different numbers of labeled data N_l under $EP = \{30\%, 35\%\}$. Specifically, the attacker fine-tunes the Roberta model using some auxiliary gender labels along with the corresponding text representations sent by the user to infer the gender labels of other samples. Experimental results demonstrate that SAP significantly outperforms existing baselines under various N_l conditions, underscoring its superior privacy-utility balance. This advantage is attributed to the innovative design of the CTI mechanism and the $d\chi$-Privacy mechanism. Unlike baselines that rely on DP noise to protect privacy and overlook the significance of tokens, SAP emphasizes preserving the features of important tokens while maintaining robust privacy through lightweight privacy mechanisms. Furthermore, the results indicate that the attack success rate of the attacker is positively related to the amount of labeled data it possesses.

Privacy Evaluation on Generation Tasks. SAP's privacy performance is also validated on the SQuAD dataset using the Llama-3 model for open-ended generative responses. Since generation tasks lack classification labels, an attention-based CTI method is employed. To calculate attention values, the client requires encoder or decoder blocks. For a bottom model with only an embedding layer, the client uses an open-source model, such as Roberta-Base, as a proxy. Fig. 4 (a) shows the trade-off between utility and privacy of SAP with different EP. We can observe that the proposed attention-based CTI method can effectively improve the utility-privacy trade-off of SAP on generation tasks.

5.4 Parameter Sensitivity Analysis

Impact of Split Position. In the SAP framework, the split position of the PLM is a crucial consideration. The Roberta model, with 24 encoder blocks, was split after the 1st to 8th blocks in experiments, compared to a setup where only the embedding layer constitutes the bottom model. Fig. 4 (b) illustrates the impact of different split positions on EP against EIA without text privatization. The results indicate that increasing the number of encoder blocks in the bottom model enhances privacy, making it harder for attackers to infer the input text from transmitted representations. Empirical privacy reaches approximately 80% with 8 encoder blocks, even without text privatization. Furthermore, we delve into the privacy protection capability and utility performance of the SAP framework with different split positions and different privacy parameter settings. Compared with centralized fine-tuning, the results in the last column of Table 3 indicate that as the number of layers included in the bottom model increases, the UA of the SAP framework without privatization decreases slightly while the EP increases significantly. In addition, we can observe that by applying text privatization and reducing the privacy parameter, EP is further strengthened, but at the same time, the UA also decreases.

5.5 Ablation Analysis

Impact of CTI in SAP. We evaluate the performance and security of the SAP framework under varying η_0 when the bottom model uses a frozen embedding layer. As shown in Fig. 5, splitting the model without text privatization does not affect performance compared to the centralized fine-tuning accuracy. However, without privatization, embedded vectors are vulnerable to EIA, with a 100% attack success rate. Introducing perturbations to ensure $\eta d\chi$-privacy enhances SAP's privacy protection. Lower η_0 values improve privacy but reduce utility, illustrating a trade-off between utility and privacy. For instance, on the Roberta model with the FP dataset, SAP achieves 38.85% empirical privacy with a 6.17% performance drop when η_0 is set to 50. Experimental results on the Roberta and Llama models show that the frequency-based CTI method enhances the utility-privacy trade-off in the SAP framework for classification tasks. By adaptively adjusting the privacy budget based on the utility importance of each token, the CTI method improves both UA and EP. Specifically, on the Roberta model with the FP dataset and η_0 set to 50, the SAP with the CTI algorithm scheme achieves 49.98% empirical privacy with only a 2.73% performance loss.

6 Conclusion

We propose a privacy-preserving fine-tuning framework, SAP, with a utility enhancement method called CTI. SAP splits the PLM into a top model on the vendor and a bottom model on the customer, using adaptive text privatization to perturb transmitted representations. This approach protects both model and data privacy while maintaining competitive performance. SAP is adaptable to various LLM customization scenarios. For customers with limited resources, a frozen embedding layer in the bottom model is recommended, enhancing privacy by 40% with a 4.6% performance loss on the SST dataset. For customers with more resources, a bottom model with 6 encoder blocks enhances privacy by 65% with only a 1% performance degradation.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (NSFC) Grant 62272222, the Jiangsu Province Outstanding Youth Fund Project (No. BK20230080), and the Fundamental Research Funds for the Central Universities (No. 2024300401). The authors would like to acknowledge these funding sources for their support.

Ethical Statement

There are no ethical issues.

References

[Aizawa, 2003] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[Akkus *et al.*, 2024] Atilla Akkus, Mingjie Li, Junjie Chu, Michael Backes, Yang Zhang, and Sinem Sav. Generated data with fake privacy: Hidden dangers of fine-tuning large language models on generated data, 2024.

[Arora and Arora, 2023] Anmol Arora and Ananya Arora. The promise of large language models in health care. *The Lancet*, 401(10377):641, 2023.

[Arslan *et al.*, 2021] Yusuf Arslan, Kevin Allix, Lisa Veiber, Cedric Lothritz, Tegawendé F Bissyandé, Jacques Klein, and Anne Goujon. A comparison of pre-trained language models for multi-class text classification in the financial domain. In *Proc. of WWW*, 2021.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. 2020.

[Ceballos *et al.*, 2020] Iker Ceballos, Vivek Sharma, Eduardo Mugica, Abhishek Singh, Alberto Roman, Praneeth Vepakomma, and Ramesh Raskar. Splitnn-driven vertical partitioning. *arXiv preprint arXiv:2008.04137*, 2020.

[Charles *et al.*, 2024] Zachary Charles, Arun Ganesh, Ryan McKenna, H. Brendan McMahan, Nicole Mitchell, Krishna Pillutla, and Keith Rush. Fine-tuning large language models with user-level differential privacy, 2024.

[Chatzikokolakis *et al.*, 2013] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *Proc. of PETS*, 2013.

[Chen and Meurers, 2016] Xiaobin Chen and Detmar Meurers. Characterizing text difficulty with word frequencies. In *Proceedings of the 11th workshop on innovative use of nlp for building educational applications*, pages 84–94, 2016.

[Chen *et al.*, 2024] Guanzhong Chen, Zhenghan Qin, Mingxin Yang, Yajie Zhou, Tao Fan, Tianyu Du, and Zenglin Xu. Unveiling the vulnerability of private fine-tuning in split-based frameworks for large language models: A bidirectionally enhanced attack. *arXiv preprint arXiv:2409.00960*, 2024.

[Church, 2017] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[Ding *et al.*, 2023] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

[Dosovitskiy and Brox, 2016] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proc. of CVPR*, 2016.

[Du *et al.*, 2023] Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proc. of CCS*, 2023.

[Dwork, 2006] Cynthia Dwork. Differential privacy. In *Proc. of ICALP*, 2006.

[Feyisetan *et al.*, 2020] Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proc. of WSDM*, 2020.

[Fu *et al.*, 2024] Qichen Fu, Minsik Cho, Thomas Merth, Sachin Mehta, Mohammad Rastegari, and Mahyar Najibi. Lazyllm: Dynamic token pruning for efficient long context llm inference, 2024.

[Guo *et al.*, 2024] Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters, 2024.

[He *et al.*, 2020] Zecheng He, Tianwei Zhang, and Ruby B Lee. Attacking and protecting data privacy in edge–cloud collaborative inference systems. *IEEE Internet of Things Journal*, 8(12):9706–9716, 2020.

[Hong *et al.*, 2024] Junyuan Hong, Jiachen T Wang, Chenhui Zhang, LI Zhangheng, Bo Li, and Zhangyang Wang. Dp-opt: Make large language model your privacy-preserving prompt engineer. In *Proc. of ICLR*, 2024.

[Hu *et al.*, 2021] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Proc. of ICLR*, 2021.

[Ji *et al.*, 2024] Lixia Ji, Shijie Xiao, Bingzhi Xu, and Han Zhang. Transferrable dp-adapter tuning: A privacy-preserving multimodal parameter-efficient fine-tuning framework. In *Proc. of QRS*, 2024.

[Jiang and Yang, 2023] Cong Jiang and Xiaolei Yang. Legal syllogism prompting: Teaching large language models for legal judgment prediction. In *Proc. of ICAIL*, 2023.

[Kenton and Toutanova, 2019] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, 2019.

[Kim *et al.*, 2022] Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In *Proc. of KDD*, New York, NY, USA, 2022. Association for Computing Machinery.

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

[Li *et al.*, 2024a] Xianzhi Li, Ran Zmigrod, Zhiqiang Ma, Xiaomo Liu, and Xiaodan Zhu. Fine-tuning language models with differential privacy through adaptive noise allocation, 2024.

[Li *et al.*, 2024b] Yang Li, Wenhan Yu, and Jun Zhao. Privtuner with homomorphic encryption and lora: A p3eft scheme for privacy-preserving parameter-efficient fine-tuning of ai foundation models. *arXiv preprint arXiv:2410.00433*, 2024.

[Liu *et al.*, 2023] Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting token pruning for object detection and instance segmentation. *arXiv preprint arXiv:2306.07050*, 2023.

[Lyu *et al.*, 2020] Lingjuan Lyu, Xuanli He, and Yitong Li. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. In *Proc. of EMNLP*, 2020.

[Malo *et al.*, 2014] Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796, 2014.

[Pan *et al.*, 2020] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. Privacy risks of general-purpose language models. In *Proc. of SP*, 2020.

[Qu *et al.*, 2021] Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. Natural language understanding with privacy-preserving bert. In *Proc. of CIKM*, 2021.

[Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[Ramos and others, 2003] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proc. of ICML*, 2003.

[Reed *et al.*, 2006] Joel W Reed, Yu Jiao, Thomas E Potok, Brian A Klump, Mark T Elmore, and Ali R Hurson. Tf-icf: A new term weighting scheme for clustering dynamic data streams. In *Proc. of ICMLA*, 2006.

[Romanini *et al.*, 2021] Daniele Romanini, Adam James Hall, Pavlos Papadopoulos, Tom Titcombe, Abbas Ismail, Tudor Cebere, Robert Sandmann, Robin Roehm, and Michael A Hoeh. Pyvertical: A vertical federated learning framework for multi-headed splitnn. *arXiv preprint arXiv:2104.00489*, 2021.

[Salton and Buckley, 1988] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[Schneider, 2004] Karl-Michael Schneider. On word frequency information and negative evidence in naive bayes text classification. In *Proc. of EsTAL*, 2004.

[Song and Raghunathan, 2020] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proc. of CCS*, 2020.

[Sun *et al.*, 2024] Qian Sun, Hanpeng Wu, and Xi Sheryl Zhang. On active privacy auditing in supervised fine-tuning for white-box language models, 2024.

[Tang *et al.*, 2024] Xinyu Tang, Ashwinee Panda, Milad Nasr, Saeed Mahloujifar, and Prateek Mittal. Private fine-tuning of large language models with zeroth-order optimization, 2024.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[Vepakomma *et al.*, 2018] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

[Wang *et al.*, 2019] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. Proc. of ICLR.

[Wang *et al.*, 2023a] Hongjie Wang, Bhishma Dedhia, and Niraj K Jha. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. *arXiv preprint arXiv:2305.17328*, 2023.

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

[Wang *et al.*, 2024] Teng Wang, Lindong Zhai, Tengfei Yang, Zhucheng Luo, and Shuanggen Liu. Selective privacy-preserving framework for large language models fine-tuning. *Information Sciences*, 678:121000, 2024.

[Yang *et al.*, 2022] Mengmeng Yang, Ivan Tjuawinata, and Kwok-Yan Lam. K-means clustering with local d-privacy for privacy-preserving data analysis. *IEEE Transactions on Information Forensics and Security*, 17:2524–2537, 2022.

[Zmushko *et al.*, 2023] Philip Zmushko, Marat Mansurov, Ruslan Svirschevski, Denis Kuznedelev, Max Ryabinin, and Aleksandr Beznosikov. Privacy preserving api fine-tuning for llms. 2023.