ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230590

Fingerprint Attack: Client De-Anonymization in Federated Learning

Qiongka Xu[*], Trevor Cohn and Olga Ohrimenko

The University of Melbourne, Carlton, VIC, Australia
ORCiD ID: Qiongka Xu https://orcid.org/0000-0003-3312-6825,
Trevor Cohn https://orcid.org/0000-0003-4363-1673, Olga Ohrimenko https://orcid.org/0000-0002-9735-0538

Abstract. Federated Learning allows collaborative training without data sharing in settings where participants do not trust the central server and one another. Privacy can be further improved by ensuring that communication between the participants and the server is anonymized through a shuffle; decoupling the participant identity from their data. This paper seeks to examine whether such a defense is adequate to guarantee anonymity, by proposing a novel *fingerprinting attack* over gradients sent by the participants to the server. We show that clustering of gradients can easily break the anonymization in an empirical study of learning federated language models on two language corpora. We then show that training with differential privacy can provide a practical defense against our fingerprint attack.

1 Introduction

Federated Learning (FL) is a machine learning paradigm that enables collaborative distributed model training without the need to share training data [26]. Federated learning usually involves a central server (analyzer) which coordinates training by i) collecting gradient updates from local clients, ii) aggregating the updates; and iii) synchronizing the latest model parameters and sending them back to the clients. In such a way, the data from clients never leaves their system, and thus less of their sensitive information is available to the potentially untrusted parties, including the server and other clients.

Although the training data does not leave clients' local devices, they are still required to communicate to the server key information about the model, namely gradients over their local data from clients to the server. The implicit information in model parameters and their updates have been shown to leak private information through attacks such as membership inference [27] and data reconstruction [15].

Linking data from the same client enables the adversary to perform stronger attacks such as i) combining additional information, *e.g.,* user name, home address, and phone numbers that appeared in different batches; and ii) boosting the attack performance by employing multiple gradients from the same source. Anonymizing client identity is believed to defend against such linkage attacks and amplify the privacy guarantees in distributed and federated learning settings, as data of an individual client is "concealed" among the data of other clients [21, 11, 3]. Random data shuffling, performed by a third party other than the server or clients, can be seen as a simple method to anonymize the identities of clients and, hence, enhance privacy in FL, as illustrated in Figure 1. A trusted shuffler is placed between clients and the server and operates as follows i) it collects the data packages with model gradients from clients, ii) it removes identities thus anonymizing the providers' identity, and iii) it shuffles data and sends them to the central server. By delinking data from the same client the Shuffle module provides a defense against attacks that use multiple gradients over time. Intuitively, shuffling in FL limits the effectiveness of many attacks, as the server can only exploit single gradients rather than a full sequence of client updates from a training run.

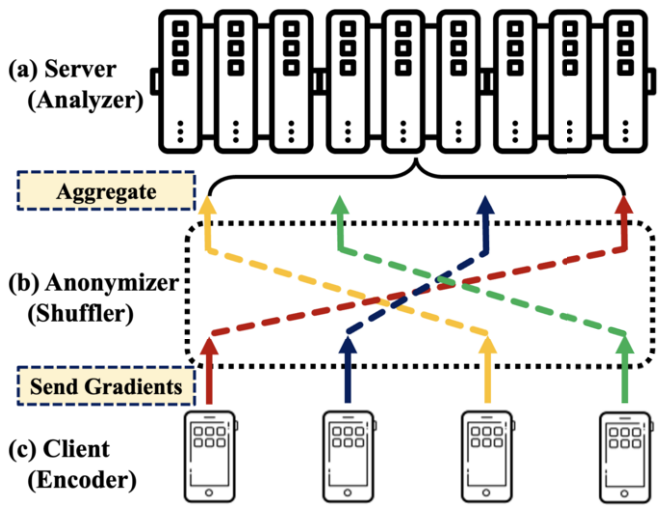


Figure 1: Framework of Federated Learning equipped with Encoder, Shuffler and Analyzer (ESA) [3], which are correlated to Server, Anonymizer and Clients. The data collected from clients are gradients of the (language) model indicated by the colored arrows.

Our work attempts to challenge the anonymization guarantees in the Shuffle-FL algorithm through a novel fingerprinting attack. In fingerprinting attack, a curious-but-honest[1] service provider records the gradients from all clients in the training process. We posit that the gradients from a client contain substantial information that is unique to that client, and thus provides a unique *fingerprint*. We propose an attack based on clustering and greedy match algorithms over pairs of gradients, in order to recover which data updates came from the same clients. We evaluate the effectiveness of our fingerprinting at-

* ✉: qiongkai.xu@unimelb.edu.au

[1] The server honestly follows the FL protocol but it is curious to learn the composition of the clients' datasets.

tack through extensive experiments on FL language modeling, showing substantially above-chance performance, and in some settings, perfect linking. As a defense, we apply differential privacy on the gradients before they are collected by the shuffler or analyzer. Our study shows that deferentially private gradients reduce the performance of fingerprinting attacks, although at a cost to model utility and training efficiency.

We summarize our contributions as follows:

- To the best of our knowledge, we are the first to propose a fingerprinting attack against the shuffler in the federated learning setting. The shuffled gradients could be grouped by greedy matching and clustering algorithms and thus traced to the same clients.
- We empirically demonstrate the feasibility of fingerprinting attacks on federated training when training a language model.
- We explore differential privacy as a defense and empirically show its effectiveness in defending against fingerprint attack, while providing a privacy-utility trade-off.

2 Related Work

Federated Learning Federated learning is a framework for collaboratively training machine learning models [16, 26, 17]. The general federated learning framework is composed of i) clients, who train local models using their data and periodically communicate the parameter updates to the server, and ii) a server, which aggregates received model updates and synchronizes the new parameters among the clients across several rounds of training. Federated learning has many applications [26] including those where (1) training of advanced deep neural network (DNN) requires a high volume of data [39, 14] that is unlikely to be owned by a single party; (2) the data cannot leave client's devices, for example, when training a diagnostic model across multiple institutions to predict clinical outcomes in patients with COVID-19 while maintaining data anonymity [8]. Language modeling is one of the fundamental tasks in Natural Language Processing and FL for language modeling recently attracted attention in academia and industry [29, 37, 6].

Attacks and Defenses in FL Several security and privacy challenges have been identified in adapting federated learning.

The first concern is the impact of malicious participants on the model learning who can backdoor the model to have a specific prediction when a trigger is given in the input [2]. A series of strategies are proposed to eliminate the confounding contributions from malicious clients, such as certifiably robust models against backdoor in FL [38] and Krum [4] against Byzantine generals problem [19].

A second critical concern is *membership inference* [35] that the attackers can determine if data was utilized in the federated model training or not [27]. A second critical concern is whether the client's local data will be disclosed to other parties in training. More ambitious are *data reconstruction* attacks, which aim to recover samples used in training. Methods for inverting gradients were proposed to reconstruct the exact training image from the first linear layer of deep neural models [12]. The following work recovered the private texts by first identifying the set of used words and then directly reconstructing sentences based on beam search [15]. These attacks utilize the model parameters and their recent updates to infer the training data.

Common methods of defense are to perturb the parameters or model updates using differential privacy [1, 23]. Shuffle models were proposed to enhance the privacy protection in FL [13, 21], as individual data items are shuffled and thus anonymously hidden in a larger batch of data which increases the difficulty of discriminating their usage [3].

Our work belongs to the second challenge and the proposed fingerprinting attack serves as a new threat to FL, specifically aiming at disabling the privacy amplification of the shuffle module.

3 Fingerprinting Attack

In this section, we first formulate the federated learning framework with a shuffle module. Then, we describe our proposed fingerprinting attack.

3.1 Preliminaries

Federated Learning. Federated Learning (FL) trains a machine learning model $f(\boldsymbol{x};\Theta)$ using data of multiple clients or silos. For each iteration $t \in [\![1..T]\!]$, the participating clients $k \in [\![1..K]\!]$ calculate the gradients of the model based on subset samples of their own data $\{\boldsymbol{x}_t^k\}$,

$$\theta_t^k \leftarrow \nabla\mathcal{L}(\boldsymbol{x}_t^k;\Theta_t) \tag{1}$$

which often involves performing several iterations of mini-batch SGD locally on the client. Then, the server aggregates the gradients by averaging the updates from the clients,

$$\Theta_{t+1} \leftarrow \Theta_t - \lambda \cdot \text{Avg}(\{\theta_t^k\}). \tag{2}$$

The procedures for averaging parameters can vary, and we primarily use FedAvg [25]. The updated parameters are then distributed to the clients, and the process is repeated for several epochs, until convergence.

Shuffle Module. Encode, Shuffle and Analyze (ESA) [3] is a framework proposed for amplifying privacy protection by adding a shuffle module as an anonymizer between client-server communication. Shuffle module was also proved to provide a better privacy guarantee in the federated learning setting when combined with differential privacy [21]. The shuffle model $\mathbb{S}$ anonymizes the client identities by permuting the data for analysis, *i.e.,* gradients sent from clients to the server in our case:

$$\mathbb{S}(\langle\theta^k\rangle) = \langle\theta^{p(k)}\rangle, \tag{3}$$

where $\langle\cdot\rangle$ is an ordered sequence and $p(\cdot)$ indicates a permutation function on $[\![1..K]\!]$. The shuffle breaks the link between individual clients and their data. Moreover, the data is mixed with data from other clients. Accordingly, the server should no longer be able to exploit the information that several gradients come from the same source, and their link to the individual client.

The shuffle has no effect on the 'honest' computation of the server, *i.e.,* the server can still perform the aggregated update in Equation 2 based on the shuffled results,

$$\Theta_{t+1} \leftarrow \Theta_t - \lambda \cdot \text{Avg}(\{\theta_t^{p(k)}\}) \tag{4}$$

to produce an identical result as permutation does not affect the average value of a set,

$$\text{Avg}(\{\theta_t^{p(k)}\}) = \text{Avg}(\{\theta_t^k\}). \tag{5}$$

3.2 Fingerprinting Attack against Anonymization

We hypothesize that the linear layer gradients from the same clients should be "similar" to each other as i) each client possesses data in specific domains which decides the distribution of hidden representation, and ii) these vectors are the main factors in gradient calculation of corresponding linear layers. If that is the case, the attacker can use gradients to group data that comes from the same client. In this section, we first provide an analysis on the intuition of fingerprinting attacks. Then, we describe our fingerprinting attack methods, based on standard clustering techniques and a simple greedy match algorithm. We note that though clustering methods have been used to enhance the training of FL[7, 34], we use clustering as an adversarial tool to perform fingerprinting attacks.

3.2.1 Attack Intuition

Inspired by a data reconstruction attack [12], we note that the gradients with respect to the parameters of a linear layer in a neural network can be used to recover the inputs to that layer.

Definition 1 *Given a linear layer,*

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}, \tag{6}$$

where $\boldsymbol{W} \in \mathbb{R}^{M\times N}$, $\boldsymbol{x} = (x_1, \cdots, x_N)^\top$ *and* $\boldsymbol{y} = (y_1, \cdots, y_M)^\top$ *are the weight matrix, the inputs and the outputs respectively. The gradient of* $\boldsymbol{W}$ *with regard to loss* $\mathcal{L}$ *is defined as*

$$\Delta \boldsymbol{W} \triangleq \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}. \tag{7}$$

Proposition 1 *The gradient* $\Delta\boldsymbol{W}$ *of a linear layer is associated with its input* $\boldsymbol{x}$*,*

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{b}} \cdot \boldsymbol{x}^\top. \tag{8}$$

The proposition is derived by the chain rule of derivation and the fact that

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{b}}. \tag{9}$$

Due to the connection between $\Delta\boldsymbol{W}$ and $\boldsymbol{x}$, we hypothesize that the gradients $\Delta\boldsymbol{W}$ from the same clients are similar to each other, given that their training data possesses similar textual patterns, *e.g.,* topics or writing styles. We will show how to measure the similarity of the gradients in the following discussion.

3.2.2 Distance Measurement

We rely on a distance metric $\mathbb{D}(\theta, \theta')$ to capture the relation between the model gradients θ and θ'. Note that gradients from linear layers $\Delta\boldsymbol{W}$ are a subset of overall gradients θ. Inspired by Gradient Inversion [12], we consider the linear layers to conduct fingerprinting attacks. As the Transformer [36] is the current dominant model architecture in NLP, we focus our experiments on this architecture, for FL fine-tuning of the GPT-2 [31] language model. All parameters are concatenated into a vector and then normalized. Euclidean distance is considered in clustering algorithms and negative cosine similarity is utilized in Greedy Match. The rationality of negative cosine similarity is that it is proportional to the Euclidean distance between the normalized vectors.

3.2.3 Naive Clustering

The inputs to the clustering are the complete collection of $K \times T$ gradient vectors as computed by K clients performing FL for T epochs. We use gradients of linear layers $\Delta\boldsymbol{W}^k$ as a subset of θ^k to construct the features. Clustering aims to assign the close vectors to the same group. To verify our design, we consider two representative clustering methods, K-means Clustering (**K-means**) [22] and Spectral Clustering (**Spectral**) [28]. **K-means** finds cluster centers that minimize the intra-class variance, which is iteratively optimized by calculating cluster centroids and data assignments to clusters. **Spectral** performs dimensionality reduction on the similarity matrix of the data before clustering.

3.2.4 Step-wise Greedy Match

Intuitively, gradients from the nearest steps possess the most signal, as the model has fewer parameter updates between neighboring steps. We propose to trace the alignments of $t \in [\![1..T]\!]$ training steps with $t+1$ steps to group the data from clients. For two neighboring epochs in the training sequence, we find the optimal alignment between the K clients' data from step t and step $t+1$. The similarity between the client gradients from two steps $\theta^{(t)}$ and $\theta^{(t+1)}$ is measured by the distance matrix $\boldsymbol{D} \in \mathbb{R}^{K\times K}$, $\boldsymbol{D}_{ij} = \mathbb{D}(\theta_i^{(t)}, \theta_j^{(t+1)})$ where $i, j \in [\![1..K]\!]$ are party identifiers. Greedy selection attempts to find the best pairings, by minimizing the distance for each adjacent time step. This formulation can be solved over the time series of parameter vectors to find a step-wise globally optimal alignment through solving a Linear Sum Assignment Problem using the Hungarian algorithm [18], where $\boldsymbol{M}$ is a matrix with each value $\boldsymbol{M}_{i,j}$ indicating a 0-1 assignment between gradients i and j.

$$\begin{aligned} &\min_{\boldsymbol{M}} \sum_i \sum_j \boldsymbol{D}_{i,j}\boldsymbol{M}_{i,j} \\ &\text{s.t. } \forall i, \sum_j \boldsymbol{M}_{i,j} = 1; \forall j, \sum_i \boldsymbol{M}_{i,j} = 1; \\ &\forall i, j, \boldsymbol{M}_{i,j} \in \{0, 1\}. \end{aligned} \tag{10}$$

The distance is defined as the negative cosine similarity of two vectors,

$$\mathbf{D}_{i,j} = 1 - \text{Sim}(\theta_i^{(t)}, \theta_j^{(t+1)}), \tag{11}$$

which is proportional to the Euclidean distance between two normalized gradients.

4 Attack Experiments

We conduct experiments on language modeling to show the effect of fingerprinting attacks.[2]

4.1 Experimental Settings

Datasets. We evaluate the fingerprinting attack on the language modeling task using two datasets: **20NewsGroup** (**News**) [20] and **EmpatheticDialogue** (**Dial**) [33]. These datasets are most commonly used for text classification, rather than language modeling, but we use them here as they contain natural data divisions that are a good match for typical FL scenarios. For **News**, we distributed the data to 20 clients, where each client accesses text samples in a single topic. For **Dial**, we include the 70 speakers with more than 288

[2] The code and its guideline are available at https://github.com/xuqiongkai/FingerprintAttack_on_FL.

utterances from the original dataset, with each speaker comprising a client. The statistics of the datasets are shown in Table 1.

Table 1: Statistic of **20NewsGroup** (**News**) and **EmpatheticDialogue** (**Dial**), with number of samples in train and valid sets. The total number of used samples equals to (#Train+#Valid)×#Clients.

Dataset	#Train	#Valid	#Clients	Total
News	512	64	20	11,520
Dial	256	32	70	20,160

Federated Learning. We simulate Federated Learning on an Nvidia A100 server based on FLSim[3]. All language models are initialized by loading a pre-trained GPT-2 model, the learning rate of the server is selected based on preliminary experiments, and the learning rate of clients' local training is set to 0.1 in all our experiments. We set the maximum sentence length to 40 tokens due to the limitation of our computational resources. Please see Appendices A and B for further details. We use stochastic gradient descent (SGD) optimizer without momentum to update the model parameters for each client.[4]

Language Model. As the Transformer [36] is the current dominant model architecture in NLP, we focus our experiments on this architecture, for FL fine-tuning of the GPT-2 [31] language model. We customized a smaller GPT model with 4 Transformer layers and pre-trained it on WikiText101. Please see more details about the model in Appendix A. Our experiments, focus on the linear layers in feed-forward modules, which are denoted as fully connected layers (FC) and projection layers (Proj). All parameters are concatenated into a vector and then normalized.

Evaluation. We evaluate the attack performance using standard evaluation metrics for clustering [24].

- **Purity Score** (Pur.) [24] measures the proportion of the dominant class over all clusters.
- **Rand Index** (RI) [32] measures the percentage of the correct decision pairs between all data points.
- **Mutual Information** (MI) [30] is a measurement of the information shared between a clustering result and the ground truth.

4.2 Results

Comparison of fingerprint attack methods. We run 10 epochs of federated learning of language model on **20NewsGroup** and **EmpatheticDialogue**, involving all 20 and 70 clients respectively. We report the performance of fingerprinting attacks in Tables 2 and 3. The proposed fingerprinting attacks achieve better clustering results than random baselines. **Spectral** consistently works better than **K-means** on both datasets, confirming the utility of its additional dimensionality reduction step. **Greedy** works the best among all our methods, achieving perfect grouping results on **20NewsGroup**. We attribute this to its explicit formulation enforcing a balanced partitioning of the data. With more clients in FL, the fingerprinting attack becomes more difficult as the attacker needs to consider more combinations of potential client-gradient mappings. This is reflected by the decrease in clustering performance with the growth of client numbers. Nonetheless, **Spectral** and **Greedy** still maintain high attack performance on both datasets.

Impact of the number of training epochs. During training, the parameters of the language model are continuously updated and synchronized for each epoch. We investigate the influence of the model dynamic by varying the training epochs in our FL experiments. The fingerprinting attack performance on **News** and **Dial** is illustrated in Figure 2. There is a clear trend of the attack becoming more difficult with the increasing number of epochs. We note that larger epoch gaps mean more difference in model states by the same client; this lead to more divergent intermediate representations $\boldsymbol{x}$ given as inputs to the attacked linear layer. The **20NewsGroup** dataset shows a clear difference in efficacy of the methods, with the random performance from **K-means** and perfect performance for **Greedy**, uniformly across all experiment sizes.

Comparison of feature construction. We are interested in the impact of feature construction on the success of the fingerprinting attack. We compare $\Delta \boldsymbol{W}$ from the feedforward modules in Transformer, using the fully connected layers (FC) and the projection layers (Proj). All the variations of feature combination and selection of layers are effective in our attack. Combining both features achieves the best performance, as demonstrated in Table 4. The results for layer selection are reported in Table 5. We have not found a layer that outperforms others by a significant margin; accordingly we use only the first layer in our primary experiments.

5 Defense on Fingerprinting Attack

In this section, we investigate differential privacy on gradients as a defense method against fingerprinting attacks. Specifically, we discuss client-side differential privacy in federated learning.

5.1 Differential Privacy

Differential Privacy (DP) is a framework for capturing privacy-preserving properties of a mechanism [9].

Definition 2 (Differential Privacy) *A mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with range $\mathcal{R}$ and domain $\mathcal{D}$ satisfies (ε, δ) differentially privacy, if for any two neighboring datasets $d, d' \in \mathcal{D}$ and for any subsets $\mathcal{S} \subseteq \mathcal{D}$ it holds that*

$$\mathbb{P}[(\mathcal{M}(d) \in \mathcal{S})] \leq e^{\varepsilon} \cdot \mathbb{P}[(\mathcal{M}(d') \in \mathcal{S})] + \delta \tag{12}$$

Informally, the definition captures that changes in a dataset (*e.g.,* presence or absence of an individual) do not significantly change the output of a DP mechanism, and the changes are bounded by parameters ϵ and δ [10].[5]

We adopt Differential Privacy for Stochastic Gradient Decent (DP-SGD) [1] at every client. That is each client performs DP-SGD locally on their dataset. The algorithm includes two main steps:
1) **Clipping the gradients:**

$$\bar{\theta}(\boldsymbol{s}_i) \leftarrow \theta(\boldsymbol{s}_i) / \max\left(1, \frac{\|\theta(\boldsymbol{s}_i)\|}{\mathcal{C}}\right) \tag{13}$$

[3] https://github.com/facebookresearch/FLSim

[4] Momentum or other gradient smoothing methods would necessitate special treatment in the attack, and would otherwise cause false positive matches.

[5] Parameter δ is preferably smaller than $1/|\mathcal{D}|$, where $|\mathcal{D}|$ indicates the size of the dataset. In the experiments we set δ to 10^{-4}.

Table 2: The comparison of fingerprint attack on federated learning with various number of clients (# Clients) on **20NewsGroup** based on purity, rand-index and mutual information (higher means better attack success).

	Random	**K-means**	**Spectral**	**Greedy**
# Clients	Pur./ RI/ MI	Pur./ RI/ MI	Pur./ RI/ MI	Pur./ RI/ MI
3	0.458 / 0.563 / 0.074	0.667 / 0.699 / 0.547	1.000 / 1.000 / 1.099	1.000 / 1.000 / 1.099
5	0.351 / 0.689 / 0.192	0.400 / 0.481 / 0.500	0.960 / 0.971 / 1.501	1.000 / 1.000 / 1.609
10	0.260 / 0.827 / 0.484	0.200 / 0.264 / 0.325	0.850 / 0.960 / 2.031	1.000 / 1.000 / 2.303
20	0.206 / 0.909 / 0.933	0.100 / 0.138 / 0.199	0.245 / 0.785 / 0.766	1.000 / 1.000 / 2.996

Table 3: The comparison of fingerprint attack on federated learning with various number of clients (# Clients) on **EmpatheticDialogue** based on Pur./RI/MI. The number of clients is between 5 to 70.

	Random	**K-means**	**Spectral**	**Greedy**
# Clients	Pur./ RI/ MI	Pur./ RI/ MI	Pur./ RI/ MI	Pur./ RI/ MI
3	0.458 / 0.563 / 0.074	0.400 / 0.359 / 0.074	0.633 / 0.607 / 0.342	0.500 / 0.600 / 0.181
5	0.351 / 0.689 / 0.192	0.280 / 0.282 / 0.132	0.400 / 0.676 / 0.341	0.580 / 0.770 / 0.584
10	0.260 / 0.827 / 0.484	0.300 / 0.528 / 0.503	0.500 / 0.853 / 1.121	0.510 / 0.881 / 1.123
20	0.206 / 0.909 / 0.933	0.305 / 0.776 / 0.976	0.450 / 0.911 / 1.592	0.500 / 0.939 / 1.761
40	0.170 / 0.954 / 1.486	0.280 / 0.875 / 1.382	0.425 / 0.959 / 2.143	0.445 / 0.967 / 2.305
70	0.148 / 0.973 / 1.984	0.250 / 0.903 / 1.728	0.449 / 0.976 / 2.745	0.497 / 0.983 / 2.983

Table 4: The comparison of features for fingerprinting attacks on **20NewsGroup** and **EmpatheticDialogue**.

	Spectral	**Greedy**
Feature	Pur./ RI/ MI	Pur./ RI/ MI
FC	0.270 / 0.905 / 1.102	1.000 / 1.000 / 2.996
Proj	0.245 / 0.795 / 0.699	1.000 / 1.000 / 2.996
Both	0.275 / 0.900 / 1.034	1.000 / 1.000 / 2.996

(a) **20NewsGroup**

	Spectral	**Greedy**
Feature	Pur./ RI/ MI	Pur./ RI/ MI
FC	0.390 / 0.916 / 1.469	0.465 / 0.935 / 1.690
Proj	0.405 / 0.917 / 1.504	0.445 / 0.935 / 1.698
Both	0.465 / 0.912 / 1.610	0.500 / 0.939 / 1.761

(b) **EmpatheticDialogue**

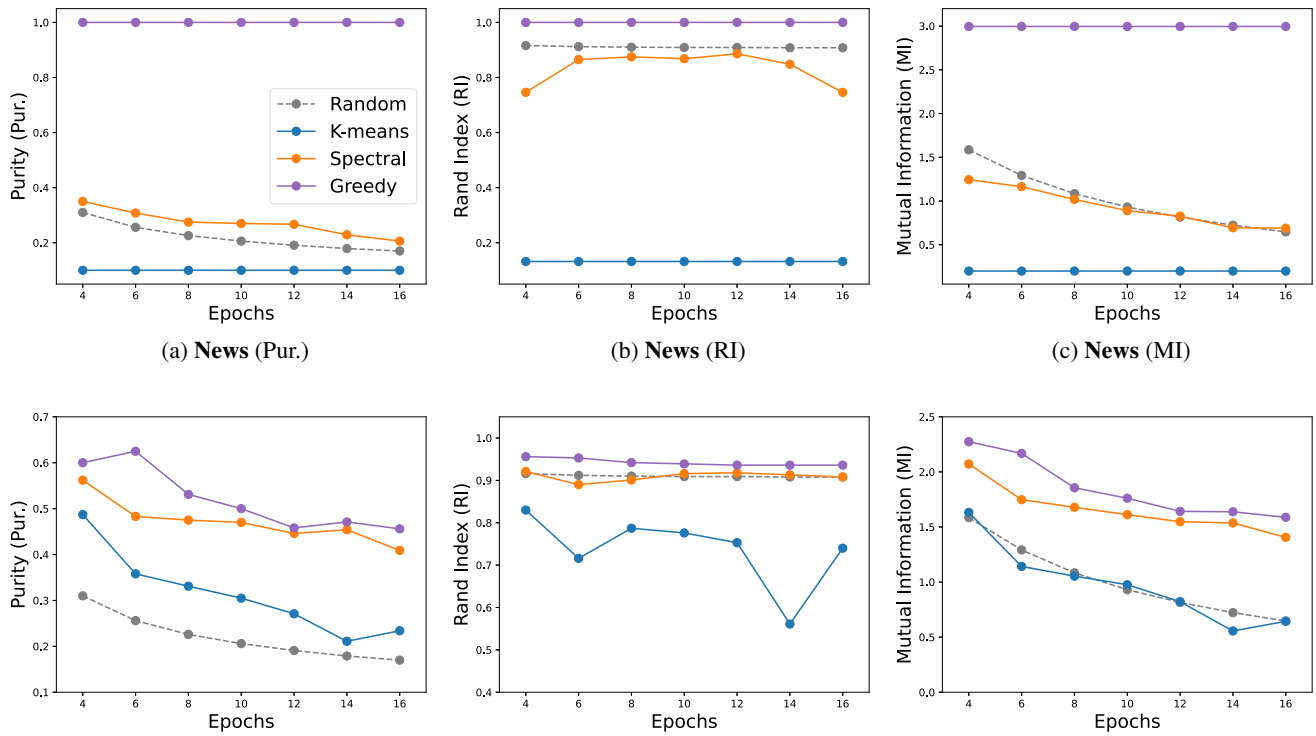
Table 5: The comparison of fingerprinting attacks on different layers $\{1, 2, 3, 4\}$ in transformer models. Both gradients of the connected layer (FC) and projection layer (Proj) are used. Purity (Pur.), Rand Index (RI) and Mutual Information (MI) are reported.

	Spectral	**Greedy**
Layer	Pur./ RI/ MI	Pur./ RI/ MI
1	0.275 / 0.900 / 1.034	1.000 / 1.000 / 2.996
2	0.280 / 0.883 / 0.908	1.000 / 1.000 / 2.996
3	0.255 / 0.874 / 0.881	1.000 / 1.000 / 2.996
4	0.285 / 0.881 / 0.962	1.000 / 1.000 / 2.996

(a) **20NewsGroup**

	Spectral	**Greedy**
Layer	Pur./ RI/ MI	Pur./ RI/ MI
1	0.465 / 0.912 / 1.610	0.500 / 0.939 / 1.761
2	0.465 / 0.922 / 1.588	0.535 / 0.941 / 1.813
3	0.465 / 0.927 / 1.628	0.520 / 0.942 / 1.869
4	0.445 / 0.928 / 1.583	0.490 / 0.939 / 1.777

(b) **EmpatheticDialogue**



(a) **News** (Pur.) (b) **News** (RI) (c) **News** (MI)

(d) **Dial** (Pur.) (e) **Dial** (RI) (f) **Dial** (MI)

Figure 2: The comparison of fingerprinting methods (**K-means**, **Spectral**, and **Greedy**) with 4 to 16 epochs on **News** and **Dial** using Purity (Pur.). Rand Index (RI) and Mutual Information (MI).

Table 6: Clustering performance (Pur./RI/MI) on gradients with DP-SGD on **20NewsGroup**, using different clipping bounds $\mathcal{C}$ and noise multipliers σ. Target delta $\delta = 10^{-4}$ and corresponding ε are used to demonstrate the DP budget. We highlight the results according to the attack performance. The baseline performance of FL without DP-SGD (No-DP) is reported in the captions of sub-tables.

$\mathcal{C}$	$\sigma = 0.5$ ($10.87, 10^{-4}$)-DP	$\sigma = 1.0$ ($2.216, 10^{-4}$)-DP	$\sigma = 1.5$ ($0.935, 10^{-4}$)-DP
50	0.685/0.959/2.283	0.220/0.915/1.033	0.205/0.913/0.930
100	0.215/0.914/0.990	0.225/0.914/1.007	0.210/0.914/0.988
200	0.210/0.914/0.971	0.245/0.916/1.087	0.245/0.915/1.054

(a) **Greedy** w. No-DP: 1.000 / 1.000 / 2.996

$\mathcal{C}$	$\sigma = 0.5$ ($10.87, 10^{-4}$)-DP	$\sigma = 1.0$ ($2.216, 10^{-4}$)-DP	$\sigma = 1.5$ ($0.935, 10^{-4}$)-DP
50	0.235/0.879/0.816	0.190/0.876/0.846	0.190/0.883/0.733
100	0.180/0.887/0.775	0.200/0.874/0.784	0.205/0.864/0.818
200	0.215/0.858/0.822	0.200/0.876/0.802	0.180/0.879/0.835

(b) **Spectral** w. No-DP: 0.245 / 0.785 / 0.766

2) **Adding noise to gradients:**

$$\bar{\theta} \leftarrow \frac{1}{L} \sum_i \bar{\theta}(\boldsymbol{s}_i) + \mathcal{N}(0, \sigma^2 \mathcal{C}^2 \boldsymbol{I}) \tag{14}$$

where clipping bound $\mathcal{C}$ and noise multiplier σ can be chosen to balance the extent of privacy conferred versus the efficacy of the trained model. The overall parameters of DP-SGD are calculated according to [1]. Note that the DP-SGD step is performed at each client after performing iterative local updates. Clipped and noisy gradients are then communicated to the server or the anonymizer. Note that since the server is not trusted, differential privacy is performed at the client.

5.2 Defense Experiments

We study the effects of DP on gradients by varying the noise multiplier σ and clipping bound C. The target δ is set to 10^{-4}. The main results on **News** are reported in Table 6 and full results on both datasets are provided in Appendix C. We vary the parameters of DP, up to the strongest setting $\varepsilon = 0.935$ which corresponds to noise multiplier $\sigma = 1.5$. As expected, the clustering performance of both **Spectral** and **Greedy** is negatively correlated with the privacy budget, indicating the strong correlation between the extent of differential privacy and the capability of preventing fingerprinting attacks on federated learning.

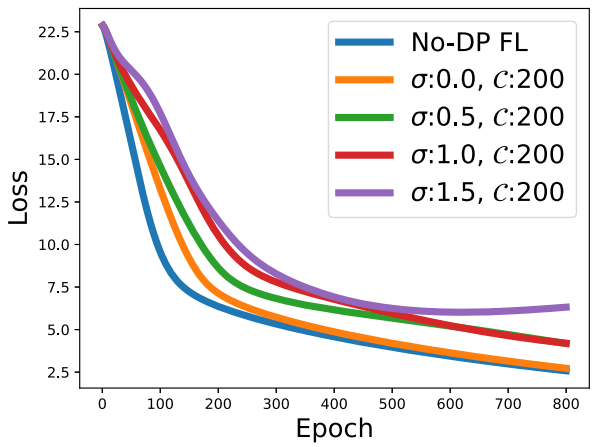


Figure 3: The comparison of various defense methods including clipping-only ($\sigma = 0$), DP-SGD (with varying noise parameters σ) and the baseline, No-DP FL, based on federated language model training on **20NewsGroup**. The loss on a holdout validation set is reported.

We further investigate the consequence of using DP-SGD in federated learning of language models. We study the vanilla FL (No-DP FL) and various DP-SGD settings on FL by comparing the model losses (log perplexity) on validation sets, as illustrated in Figure 3. We observe that some of these models have low utility, particularly the model with the strongest DP guarantee with $\sigma = 1.5$, which begins to diverge after 500 epochs. In the setting where only clipping is used and no noise is added, $\sigma = 0.0$, the model achieves a similar loss to the No-DP model, albeit with no DP guarantees. Increasing σ to 0.5 or 1.0 results in about 3 times slower convergence iterations and 4 times slower running speed for each epoch, but still attains reasonable generalization performance.

6 Conclusion

In this paper, we evaluate privacy guarantees provided by a shuffle module if deployed in Federated Learning setting. To this end, we design a new fingerprinting attack that can link shuffled data updates across training epochs back to the same user. Our experimental results show the feasibility of our attack when training language models on shuffled gradients. DP on gradients is examined to show its effectiveness in defending against the new fingerprinting attack.

Limitations

Our work proposes an attack technique on the Shuffle module in Federated Learning, which could be used for malicious purposes. However, the main purpose of our work is to expose the threat to the research community. We have also discussed DP as a possible defense method against the new attack. Though differential privacy can alleviate the fingerprinting attack, it decreases the performance of federated learning. Finding methods with a better privacy-utility trade-off is still an open question. Methods based on Multi-Party Computation (MPC) [5] can also be seen as defense mechanisms against fingerprinting attacks since clients send their updates in an encrypted form. However, these methods require other considerations in practice such as computational costs and participants strictly following the protocol.

Subsampling is also a widely used technology in federated learning. It may diminish the performance of **Greedy** as the one-to-one alignment constraint does not hold in this case. However, the clustering-based attacks, *i.e.,* **K-means** and **Spectral**, will have less of an effect, as they do not impose the requirement of an equal number of participants for each round.

The fingerprinting attack requires the attacker to record all gradient updates of all clients, which can lead to significant storage costs. The cost can be reduced with a few efforts: i) using Step-wise Greedy Match, which only requires storing gradients from the recent two epochs, and ii) using a single transformer layer instead of all layers or even a part of a layer, *i.e.,* FC and Proj layers, as demonstrated in Tables 4 and 5.

Acknowledgement

This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative. This work was supported in part by an Oracle Research Grant.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, 'Deep learning with differential privacy', in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, (2016).

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov, 'How to backdoor federated learning', in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948. PMLR, (2020).

[3] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld, 'Prochlo: Strong privacy for analytics in the crowd', in *Proceedings of the 26th symposium on operating systems principles*, pp. 441–459, (2017).

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer, 'Machine learning with adversaries: Byzantine tolerant gradient descent', *Advances in Neural Information Processing Systems*, **30**, (2017).

[5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth, 'Practical secure aggregation for privacy-preserving machine learning', in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, (2017).

[6] Mingqing Chen, Ananda Theertha Suresh, Rajiv Mathews, Adeline Wong, Cyril Allauzen, Françoise Beaufays, and Michael Riley, 'Federated learning of n-gram language models', in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 121–130, (2019).

[7] Marcos F Criado, Fernando E Casado, Roberto Iglesias, Carlos V Regueiro, and Senén Barro, 'Non-iid data and continual learning processes in federated learning: A long road ahead', *Information Fusion*, **88**, 263–280, (2022).

[8] Ittai Dayan, Holger R Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J Wood, Chien-Sung Tsai, et al., 'Federated learning for predicting clinical outcomes in patients with covid-19', *Nature medicine*, **27**(10), 1735–1743, (2021).

[9] Cynthia Dwork, 'A firm foundation for private data analysis', *Communications of the ACM*, **54**(1), 86–95, (2011).

[10] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor, 'Our data, ourselves: Privacy via distributed noise generation', in *Annual international conference on the theory and applications of cryptographic techniques*, pp. 486–503. Springer, (2006).

[11] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta, 'Amplification by shuffling: From local to central differential privacy via anonymity', in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479. SIAM, (2019).

[12] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller, 'Inverting gradients-how easy is it to break privacy in federated learning?', *Advances in Neural Information Processing Systems*, **33**, 16937–16947, (2020).

[13] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh, 'Shuffled model of differential privacy in federated learning', in *International Conference on Artificial Intelligence and Statistics*, pp. 2521–2529. PMLR, (2021).

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.

[15] Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen, 'Recovering private text in federated learning of language models', *Advances in Neural Information Processing Systems*, **35**, 8130–8143, (2022).

[16] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al., 'Advances and open problems in federated learning', *Foundations and Trends® in Machine Learning*, **14**(1–2), 1–210, (2021).

[17] Jakub Konecnỳ, H Brendan McMahan, and Daniel Ramage, 'Federated optimization: Distributed optimization beyond the datacenter'.

[18] Harold W Kuhn, 'The hungarian method for the assignment problem', *Naval research logistics quarterly*, **2**(1-2), 83–97, (1955).

[19] Leslie Lamport, Robert Shostak, and Marshall Pease, 'The byzantine generals problem', *ACM Transactions on Programming Languages and Systems*, **4**(3), 382–401, (1982).

[20] Ken Lang, 'Newsweeder: Learning to filter netnews', in *Machine Learning Proceedings 1995*, 331–339, Elsevier, (1995).

[21] Ruixuan Liu, Yang Cao, Hong Chen, Ruoyang Guo, and Masatoshi Yoshikawa, 'Flame: Differentially private federated learning in the shuffle model', in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8688–8696, (2021).

[22] Stuart Lloyd, 'Least squares quantization in pcm', *IEEE transactions on information theory*, **28**(2), 129–137, (1982).

[23] Lingjuan Lyu, Xuanli He, and Yitong Li, 'Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness', in *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2355–2365, (2020).

[24] Christopher D Manning, *Introduction to information retrieval*, Syngress Publishing,, 2008.

[25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, 'Communication-efficient learning of deep networks from decentralized data', in *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, (2017).

[26] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas, 'Federated learning of deep networks using model averaging', *arXiv preprint arXiv:1602.05629*, **2**, (2016).

[27] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov, 'Exploiting unintended feature leakage in collaborative learning', in *2019 IEEE symposium on security and privacy (SP)*, pp. 691–706. IEEE, (2019).

[28] Andrew Ng, Michael Jordan, and Yair Weiss, 'On spectral clustering: Analysis and an algorithm', *Advances in neural information processing systems*, **14**, (2001).

[29] Peyman Passban, Tanya Roosta, Rahul Gupta, Ankit Chadha, and Clement Chung, 'Training mixed-domain translation models via federated learning', in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2576–2586, Seattle, United States, (July 2022). Association for Computational Linguistics.

[30] Darius Pfitzner, Richard Leibbrandt, and David Powers, 'Characterization and evaluation of similarity measures for pairs of clusterings', *Knowledge and Information Systems*, **19**(3), 361–394, (2009).

[31] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, 'Language models are unsupervised multitask learners', (2019).

[32] William M Rand, 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical association*, **66**(336), 846–850, (1971).

[33] Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau, 'Towards empathetic open-domain conversation models: A new benchmark and dataset', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5370–5381, Florence, Italy, (July 2019). Association for Computational Linguistics.

[34] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek, 'Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints', *IEEE transactions on neural networks and learning systems*, **32**(8), 3710–3722, (2020).

[35] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov, 'Membership inference attacks against machine learning models', in *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, (2017).

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', *Advances in neural information processing systems*, **30**, (2017).

[37] Orion Weller, Marc Marone, Vladimir Braverman, Dawn Lawrie, and Benjamin Van Durme, 'Pretrained models for multilingual federated learning', in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1413–1421, Seattle, United States, (July 2022). Association for Computational Linguistics.

[38] Chulin Xie, Minghao Chen, Pin-Yu Chen, and Bo Li, 'Crfl: Certifiably robust federated learning against backdoor attacks', in *International Conference on Machine Learning*, pp. 11372–11382. PMLR, (2021).

[39] Qingchen Zhang, Laurence T Yang, Zhikui Chen, and Peng Li, 'A survey on deep learning for big data', *Information Fusion*, **42**, 146–157, (2018).

A Pre-trained Language Model Settings

We pre-train a local language model using WikiText101 for 20 epochs. A 4×V100 server is used for pre-training the language model for 8 hr 23 minutes, achieving a final evaluation loss of 4.03. The detailed model and training settings are provided in Table 7 and 8.

Table 7: Hyper-parameters for pre-training.

Optimizer	AdamW
Learning Rate	5e-05
Batch Size	24
Adam Beta1	0.9
Adam Beta2	0.999
Adam Epsilon	1e-08

Table 8: Hyper-parameters for language model.

Model Type	GPT-2
Embedding Dimension	192
Number of Heads	12
Number of Layer	4
Attention Dropout Rate	0.1
Embedding Dropout Rate	0.1

B Ablation Study on Federated Learning Settings

We compare the fingerprinting attack on federated learning with various learning rates (γ) for server training. All client learning rates are set to 0.1 in all our experiments. We involve 20 clients and train 10 epochs for both datasets. Although the gradients in **Dial** are noisier than **News**, the overall fingerprinting attack performance is consistently effective. A lower learning rate indicates a higher risk to the fingerprinting attack on both **Dial** and **News**, which means the risk of FL could increase towards the end of training when the learning rate is decayed to a very small value at this stage. The selected settings in our paper are highlighted with double daggers (‡).

Table 9: The comparison of fingerprint attacks, **K-means**, **Spectral**, and **Greedy**, on Federated Learning with various learning rate (γ). Purity (Pur.), Rand Index (RI) and Mutual Information (MI) are reported. * indicates **K-means** does not converge in the experiment.

	K-means	**Spectral**	**Greedy**
γ	Pur./ RI/ MI	Pur./ RI/ MI	Pur./ RI/ MI
1e-4	0.105 / 0.180 / 0.233	0.235 / 0.871 / 0.815	0.530 / 0.946 / 2.138
1e-5	N/A*	0.235 / 0.870 / 0.885	0.840 / 0.974 / 2.500
1e-6‡	0.100 / 0.138 / 0.199	0.270 / 0.896 / 0.958	1.000 / 1.000 / 2.996

(a) **20NewsGroup**

	K-means	**Spectral**	**Greedy**
γ	Pur./ RI/ MI	Pur./ RI/ MI	Pur./ RI/ MI
1e-5	0.190 / 0.650 / 0.568	0.290 / 0.879 / 1.120	0.330 / 0.921 / 1.299
1e-6	0.185 / 0.606 / 0.514	0.260 / 0.906 / 1.083	0.295 / 0.919 / 1.202
1e-7‡	0.305 / 0.776 / 0.976	0.450 / 0.911 / 1.592	0.500 / 0.939 / 1.761

(b) **EmpatheticDialogue**

C Ablation Study on Differential Privacy Settings

We compare the effect of varying settings of DP-SGD on fingerprinting attack performance. We choose clipping value $\mathcal{C} \in \{50, 100, 200\}$ and noise multiplier $\sigma \in \{0, 0.5, 1.0, 1.5\}$.

			K-means	**Spectral**	**Greedy**
Clip $\mathcal{C}$	Noise σ	(ϵ, δ)-DP	Pur./ RI/ MI	Pur./ RI/ MI	Pur./ RI/ MI
No-DP			0.100 / 0.138 / 0.199	0.245 / 0.785 / 0.766	1.000/ 1.000 / 2.996
50	0.0	$(\infty, 10^{-4})$	0.100 / 0.138 / 0.199	0.280 / 0.875 / 0.937	1.000 / 1.000 / 2.996
50	0.5	$(10.87, 10^{-4})$	0.145 / 0.211 / 0.293	0.235 / 0.879 / 0.816	0.685 / 0.959 / 2.283
50	1.0	$(2.216, 10^{-4})$	0.145 / 0.210 / 0.289	0.190 / 0.876 / 0.846	0.220 / 0.915 / 1.033
50	1.5	$(0.935, 10^{-4})$	0.160 / 0.578 / 0.326	0.190 / 0.883 / 0.733	0.205 / 0.913 / 0.930
100	0.0	$(\infty, 10^{-4})$	0.100 / 0.138 / 0.199	0.265 / 0.864 / 0.943	1.000 / 1.000 / 2.996
100	0.5	$(10.87, 10^{-4})$	0.145 / 0.210 / 0.289	0.180 / 0.887 / 0.775	0.215 / 0.914 / 0.990
100	1.0	$(2.216, 10^{-4})$	0.145 / 0.210 / 0.288	0.200 / 0.874 / 0.784	0.225 / 0.914 / 1.007
100	1.5	$(0.935, 10^{-4})$	0.170 / 0.538 / 0.359	0.205 / 0.864 / 0.818	0.210 / 0.914 / 0.988
200	0.0	$(\infty, 10^{-4})$	0.100 / 0.138 / 0.199	0.265 / 0.877 / 0.964	1.000 / 1.000 / 2.996
200	0.5	$(10.87, 10^{-4})$	0.145 / 0.210 / 0.289	0.215 / 0.858 / 0.822	0.210 / 0.914 / 0.971
200	1.0	$(2.216, 10^{-4})$	0.175 / 0.470 / 0.383	0.200 / 0.876 / 0.802	0.245 / 0.916 / 1.087
200	1.5	$(0.935, 10^{-4})$	0.165 / 0.584 / 0.373	0.180 / 0.879 / 0.835	0.245 / 0.915 / 1.054

Table 10: The comparison of various DP-SGD with No-DP on **20NewsGroup**.

			K-means	**Spectral**	**Greedy**
Clip $\mathcal{C}$	Noise σ	(ϵ, δ)-DP	Pur./ RI/ MI	Pur./ RI/ MI	Pur./ RI/ MI
No-DP			0.305 / 0.776 / 0.976	0.450 / 0.911 / 1.592	0.500 / 0.939 / 1.761
50	0.0	$(\infty, 10^{-4})$	1.000 / 1.000 / 2.996	1.000 / 1.000 / 2.996	1.000 / 1.000 / 2.996
50	0.5	$(12.67, 10^{-4})$	1.000 / 1.000 / 2.996	1.000 / 1.000 / 2.996	1.000 / 1.000 / 2.996
50	1.0	$(3.095, 10^{-4})$	0.325 / 0.793 / 1.144	0.740 / 0.940 / 2.239	0.910 / 0.987 / 2.812
50	1.5	$(1.426, 10^{-4})$	0.175 / 0.483 / 0.425	0.290 / 0.896 / 1.087	0.295 / 0.919 / 1.238
100	0.0	$(\infty, 10^{-4})$	0.950 / 0.993 / 2.913	0.510 / 0.860 / 1.972	1.000 / 1.000 / 2.996
100	0.5	$(12.67, 10^{-4})$	0.645 / 0.935 / 2.067	0.860 / 0.973 / 2.586	0.980 / 0.997 / 2.946
100	1.0	$(3.095, 10^{-4})$	0.150 / 0.235 / 0.315	0.205 / 0.895 / 0.854	0.240 / 0.915 / 1.054
100	1.5	$(1.426, 10^{-4})$	0.170 / 0.577 / 0.357	0.205 / 0.896 / 0.876	0.225 / 0.915 / 1.047
200	0.0	$(\infty, 10^{-4})$	0.715 / 0.940 / 2.457	0.795 / 0.961 / 2.632	0.950 / 0.995 / 2.926
200	0.5	$(12.67, 10^{-4})$	0.190 / 0.568 / 0.499	0.230 / 0.898 / 0.958	0.250 / 0.916 / 1.057
200	1.0	$(3.095, 10^{-4})$	0.190 / 0.638 / 0.561	0.225 / 0.889 / 0.934	0.255 / 0.916 / 1.097
200	1.5	$(1.426, 10^{-4})$	0.210 / 0.624 / 0.628	0.280 / 0.905 / 1.097	0.295 / 0.919 / 1.193

Table 11: The comparison of various DP-SGD with No-DP on **EmpatheticDialogue**.