

# Latent Dirichlet Allocation Model Training with Differential Privacy

Fangyuan Zhao, Xuebin Ren, Shusen Yang, Qing Han, Peng Zhao, and Xinyu Yang

**Abstract**—Latent Dirichlet Allocation (LDA) is a popular topic modeling technique for hidden semantic discovery of text data and serves as a fundamental tool for text analysis in various applications. However, the LDA model as well as the training process of LDA may expose the text information in the training data, thus bringing significant privacy concerns. To address the privacy issue in LDA, we systematically investigate the privacy protection of the main-stream LDA training algorithm based on Collapsed Gibbs Sampling (CGS) and propose several differentially private LDA algorithms for typical training scenarios. In particular, we present the first theoretical analysis on the inherent differential privacy guarantee of CGS based LDA training and further propose a centralized privacy-preserving algorithm (HDP-LDA) that can prevent data inference from the intermediate statistics in the CGS training. Also, we propose a locally private LDA training algorithm (LP-LDA) on crowdsourced data to provide local differential privacy for individual data contributors. Furthermore, we extend LP-LDA to an online version as OLP-LDA to achieve LDA training on locally private mini-batches in a streaming setting. Extensive analysis and experiment results validate both the effectiveness and efficiency of our proposed privacy-preserving LDA training algorithms.

**Index Terms**—Topic model, Latent Dirichlet Allocation, collapsed Gibbs sampling, differential privacy

## I. INTRODUCTION

**L**ATENT Dirichlet Allocation (LDA) [1] is a basic building block widely used in many machine learning (ML) applications. In essence, LDA works by mapping the high dimensional word space to a low dimensional topic space while preserving the implicit probabilistic relationship. Therefore, LDA is often used as a dimension reduction tool for extracting main features from massive text datasets, thereby simplifying the subsequent text processing tasks like classification and similarity judgment. With such great benefits, a series of large-scale LDA platforms have been developed in the era of big data, including light LDA [2] for Microsoft, Peacock [3] and LDA\* [4] for Tencent, and Yahoo!LDA [5] for Yahoo!.

As shown in Fig. 1, with a broad spectrum of application fields, LDA may be trained on information-sensitive datasets from both large enterprises/organizations and massive crowdsourcing users. For example, patients' electronic health records can be fed in an LDA model to help doctors diagnose possible diseases [6]. Social media profiles can be utilized with LDA models to perform fine-grained community discovery [7]. However, similar to other ML techniques, the LDA model improves its utility by constantly exploring the raw data, thus inevitably memorizing some knowledge about the dataset. By utilizing this characteristic, several attack models have been proposed to distill private information of the training data from machine learning models. For example, membership

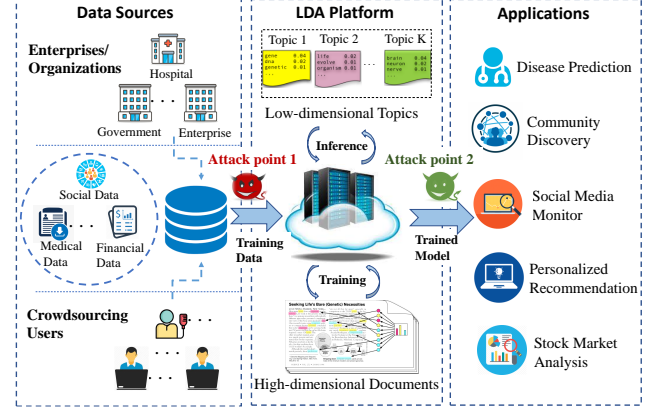


Fig. 1. Application Scenarios of LDA Model Training

inference attack [8] has been proved to be able to extract the membership of training samples. Model inversion attack [9] can be launched to recover the training data by observing the model predictions. As a typical ML model, LDA may also suffer from these attacks and cause severe privacy risks.

Differential privacy (DP) [10], as a rigorous paradigm for privacy preservation, provides not only a mathematical framework for quantifying the privacy risks of existing algorithms, but also an efficient guidance for customizing privacy-preserving algorithms. As a result, it has become the de-facto standard of privacy preservation and has been adopted in a broad wide of applications like data publication [11–13] and machine learning [14, 15]. Similarly, DP for LDA has also attracted lots of research interests [16–18]. For example, Park et al. [16] provided DP for variational Bayesian algorithm based LDA training. Zhu et al. [17] introduced DP into the collapsed Gibbs sampling (CGS) [19] training process, by adding noise to the word counts statistics in the last iteration of the sampling process. Wang et al. [18] proposed a local private solution to LDA training in a federated setting.

However, there are several limitations regarding the existing studies. On one hand, few works consider strong adversaries with full knowledge of the training mechanism and access to intermediate statistics for CGS-based LDA training (a common adversary model similar to [15]). In particular, in the training process, both the word-count information and the sampled topics for each word can reveal the information about specific training samples. Nevertheless, the existing work [17] simply focused on protecting the former but neglected the latter. On the other hand, these methods mainly prevent third-party adversaries from stealing the trained model (e.g., Attack Point

2 in Fig. 1). They implicitly assume the LDA model is trained on centralized datasets by a trustworthy server. However, the central server may act as an “honest but curious” adversary and steal training data silently (e.g., Attack Point 1 in Fig. 1). Then, users may be reluctant to share their data directly. Furthermore, most of the existing works focus on batched LDA for static text data, thus cannot efficiently cope with many practical scenarios where training data comes in a streaming fashion.

To address the issues, in this paper, we not only investigate to provide more comprehensive protection for the whole training process by utilizing the inherent privacy of the CGS algorithm on centralized datasets, but also present a solution to LDA training with local differential privacy (LDP). In addition, we further extend the LDP solution to an efficient online LDA scenario. Our contributions are summarized as follows:

- 1) We present the first study on the inherent privacy of CGS-based LDA by discovering the consistency between the topic sampling of CGS and the exponential mechanism of DP. Based on the study, we propose HDP-LDA, the first privacy-preserving LDA algorithm to protect the whole training process of CGS-based LDA on centralized datasets.
- 2) We propose LP-LDA, a novel privacy-preserving mechanism that supports training an LDA model on crowd-sourced datasets. LP-LDA can provide local differential privacy for individual data contributors.
- 3) We propose OLP-LDA, a privacy-preserving online LDA algorithm for crowd-sourced data streams. In particular, we first provide a baseline online LDA framework O-LDA, and then propose to utilize prior knowledge to refine model accuracy on locally private data streams.
- 4) We conduct extensive experiments on several real-world datasets to validate the effectiveness of the proposed algorithms. Experiment results show that the proposed algorithms can achieve much higher model utility while providing sufficient privacy guarantees.

This paper is an extension of our preliminary work [20], which focuses on the privacy preservation of batch LDA training on static datasets. Compared with [20], more extensive analysis and experiments have been added to present a comprehensive study. Besides, the current paper further proposes a novel online algorithm OLP-LDA to achieve efficient local differential privacy for LDA training on crowd-sourced data streams. In OLP-LDA, we first propose an online LDA training framework O-LDA, and then present a novel Bayesian denoising mechanism to enhance the utility of O-LDA on privacy-preserved data batches by leveraging prior knowledge. Finally, we perform extensive experiments to validate the performance of OLP-LDA.

The remainder of this paper is organized as follows. We discuss the related studies in Section II. We review the background of LDA and differential privacy in Section III. We present the intrinsic privacy study of CGS based LDA in Section IV. Then, we give the LDP solutions for the batch and online modes in Sections V and VI, respectively. The

experiments and simulation results are described and explained in Section VII. Finally, we conclude this paper in Section VIII.

## II. RELATED WORK

### A. Machine Learning with Differential Privacy

Many studies [21–26] have adopted DP in the privacy preservation of ML models. The basic idea is to perturb the ML models in different parts. *Output perturbation* on the final model result is the most straightforward solution. However, many ML models incur unbounded sensitivities, which makes it difficult to implement. To mitigate this issue, *sample-and-aggregate* [27–29] framework is proposed to first train on disjointedly sample partitions and then aggregate the trained results with DP. Still, this technique applies to ML on relatively small datasets. *Objective perturbation* randomizes the cost function of ML models [14, 30]. *Intermediate perturbation* aims to randomize the intermediate parameters during iterative training, which is effective for deep learning [15, 16, 31]. Recently, *input perturbation* that trains ML models on the perturbed datasets has attracted extensive research interests. A relevant notion in DP is called *local differential privacy* [32, 33], which shows that meaningful statistics could be obtained from massive randomized crowdsourced data. Both *input perturbation* and *local differential privacy* aim to eliminate the assumption for trustworthy servers, thus providing a stronger privacy guarantee.

### B. LDA Training with Differential Privacy

Similar to other ML models, LDA can also achieve DP protection by the aforementioned strategies except for objective perturbation, since it has no explicit cost function. To the best of our knowledge, most of the current works achieve privacy protection by perturbing the intermediate parameters during the training process. For instance, Park et al. [16] proposed to obtain DP guarantee for LDA models by perturbing the expected sufficient statistics in each iteration of the variational Bayesian method, which is a parameter estimation algorithm for LDA.

Zhu et al. [17] presented a differentially private LDA algorithm by perturbing the sampling distribution in the last iteration of the CGS process [19], which is another typical training algorithm for LDA. Decarolis et al. [34] decomposed the LDA training into a workflow based on the spectral algorithm (a parameter estimation algorithm for LDA) and then perturbed those intermediate statistics located in the cut of the workflow. There is other efficient differentially private parameter estimation algorithm using Bayesian inference with a Gibbs sampler [35]. However, the Gibbs sampler in it only considers a single conjugate structure and cannot be directly extended to LDA model training, which involves a coupled conjugate structure. Besides, the above DP methods [17, 19, 34] cannot defend against the untrustworthy data curators by design. Wang et al. [18] presented a locally private LDA training algorithm for the federated setting, in which the data are perturbed before uploading in each iteration. However, it is not a general method for traditional batch or online LDA learning. In this paper, we aim to present DP solutions to both

batch and online LDA training which can defend against the adversaries with full knowledge of the training process, even the untrustworthy data curators.

### C. Intrinsic Privacy of Randomized Algorithm

Several recent studies [36–39] have begun to look into the intrinsic privacy guarantee provided by the randomized algorithms. For example, Wang et al. [36] and Dimitrakakis et al. [38] first demonstrated that, when meeting certain conditions of the loss function, the posterior Bayesian sampling and the stochastic gradient Markov Chain Monte Carlo (MCMC) techniques could possess some inherent privacy guarantee without the introduction of extra noise. Foulds et al. [40] further extended this conclusion to the general MCMC methods. By utilizing the inherent randomness of MCMC, they achieved a certain level of privacy protection equivalent to that assured by a Laplace mechanism. Minami et al. [41] then relaxed the required conditions for the loss function in [36]. Nevertheless, these works investigated the inherent privacy guarantee of randomized algorithms in theory on the basis of some ideal assumptions about the parameters, such as the bounds of their sensitivities are known. However, in the LDA model training, it is often infeasible to compute or bound the parameters. In such a case, accurately measuring the inherent privacy guarantee remains a great challenge.

## III. PRELIMINARIES

### A. Collapsed Gibbs Sampling based LDA

1) *LDA Generative Model*: LDA is a generative model, which describes the hidden semantic architecture of document corpus generation. In the view of LDA, each document  $d_m$  containing  $N_m$  words in the corpus (or text dataset<sup>1</sup>)  $D$ , is a mixture of  $K$  different topics and can be represented by a  $K$ -dimensional “document-topic” distribution  $\theta_m$ . Each topic  $k$  is characterized by a mixture of  $V$  words, represented by a  $V$ -dimensional “topic-word” distribution  $\phi_k$ .

As shown in Fig. 2, LDA defines the generative process of a given corpus as follows:

- 1) For each topic  $k$ , draw a “topic-word” distribution  $\phi_k \sim \text{Dir}(\beta)$  over all  $V$  words, where  $\beta$  is the hyperparameter describing the prior observation for the “topic-word” count.
- 2) For each document  $d_m$ , draw a “document-topic” distribution  $\theta_m \sim \text{Dir}(\alpha)$  over all  $K$  topics, where  $\alpha$  is the hyperparameter describing the prior observation for the “document-topic” count.
- 3) For each word  $w_i$  in  $d_m$ ,  $1 \leq i \leq N_m$ , sample a topic  $k \sim \theta_m$  and a word  $t \sim \phi_k$ .

2) *Collapsed Gibbs Sampling*: The objective of LDA training is to learn the topic-word distribution  $\phi_k$  for each topic  $k$ , which can be used to infer the document-topic distribution  $\theta_m$  for any unseen document  $d_m$ . Collapsed Gibbs Sampling (CGS) is the most popular training algorithm for LDA. As a special MCMC method, CGS works by generating topic samples alternatively for all the words in  $D$ , and then conducting

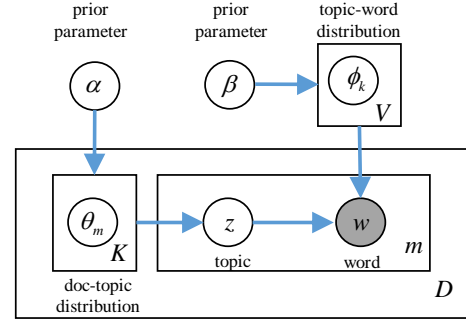


Fig. 2. LDA Graph Model

Bayesian estimation for the topic-word distribution based on the generated topic samples. Three main procedures of CGS are summarized as follows:

- **Initialization.** In the beginning, each word  $w \in D$  is randomly assigned with a topic  $k \in \mathcal{K}$ , and the word-count information  $n_m^k$  and  $n_k^t$  is counted<sup>2</sup>.  $n_k^t$  denotes the number of times that word  $t$  has been assigned with the topic  $k$ , and  $n_m^k$  refers to the number of times that topic  $k$  has been assigned to a word of the document  $d_m$ .
- **Burn-in.** In each iteration, the topic assignment for each word  $w \in D$  is updated alternatively by sampling from a multinomial distribution  $\mathbf{P} = [p_1, \dots, p_k, \dots, p_K]$ . Each component of  $\mathbf{P}$  can be computed by

$$p_k \propto \frac{n_k^t + \beta}{\sum_{t=1}^V (n_k^t + \beta)} \cdot \frac{n_m^k + \alpha}{\sum_{k=1}^K (n_m^k + \alpha)} \quad (1)$$

where  $p_k$  refers to the probability that topic  $k$  is sampled. The word-count information  $n_m^k$  and  $n_k^t$  is updated in each sampling. After the given  $T$  iterations, the burn-in process stops and the topic samples  $\mathbf{z}$  can be obtained.

- **Estimation.** The topic-word distribution  $\phi_k$  for each topic  $k$  is estimated based on the topic samples  $\mathbf{z}$  and the words  $\mathbf{w} \in D$ . In particular, each component of  $\phi_k$  can be estimated by

$$\mathbb{E}[\phi_k^t | \mathbf{z}, \mathbf{w}] = \frac{n_k^t + \beta}{\sum_{t=1}^V (n_k^t + \beta)}, \quad (2)$$

where  $\phi_k^t$  refers to the probability that word  $t$  is generated by topic  $k$  (corresponds to the assumption of LDA that words are generated by topics).

The detailed algorithm of CGS can be referred to in [42].

### B. Differential Privacy and Exponential Mechanism

Differential privacy (DP) provides a rigorous framework to quantify the privacy guarantee by analyzing the statistical difference between the algorithm outputs on neighboring datasets.

**Definition 1. (Differential Privacy [10])** A randomized mechanism  $\mathcal{M} : D \rightarrow Y$  satisfies  $\epsilon$ -differential privacy if for

<sup>2</sup>For simplicity,  $n_m^k$  is called document-topic count and  $n_k^t$  is called topic-word count in this paper.

<sup>1</sup>For simplicity, we use both terms interchangeably.

any neighboring datasets  $D, D'$  that differ by one record (i.e.,  $|D \oplus D'| = 1$ ) and any output  $S \subseteq Y$ , there is

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(D') \in S],$$

where  $\Pr[\cdot]$  is the probability and  $\epsilon$  is the privacy level of  $\mathcal{M}$ .

Exponential mechanism is a fundamental technique to achieve  $\epsilon$ -DP for the situations where a query requires an approximately "best" answer returned privately without any perturbation. The core idea of the exponential mechanism is to return an answer sampled from the answer set based on a certain distribution.

**Definition 2. (Exponential Mechanism [10])** A mechanism  $\mathcal{M}_E(x, u, \mathcal{R}) : D \rightarrow R$  satisfies  $\epsilon$ -DP if  $\mathcal{M}_E(x, u, \mathcal{R})$  outputs an element  $r \in R$  with probability  $p_r$  satisfies that

$$p_r \propto \exp\left(\frac{\epsilon}{2\Delta u} u(x, r)\right)$$

where  $u(x, r)$  is the utility function and  $\Delta u$  is its sensitivity.

### C. Local Differential Privacy

As a variant of DP, local differential privacy (LDP) describes a new privacy paradigm that any two different inputs will be mapped to the same value with similar probability.

**Definition 3. (Local Differential Privacy [43])** A randomized function  $f$  satisfies  $\epsilon$ -local differential privacy if and only if for any two input tuples  $t$  and  $t'$  in the domain of  $f$ , and for any output  $t^*$  in the range of  $f$ , there is:

$$\Pr[f(t) = t^*] \leq \exp(\epsilon) \cdot \Pr[f(t') = t^*],$$

where  $\Pr[\cdot]$  is the probability and  $\epsilon$  is the local differential privacy parameter.

LDP can be implemented via randomized response mechanism and well adapted to the crowd-sourcing scenario where the central server may not be trustworthy to end users who prefer to protect their own data individually.

## IV. HDP-LDA: A HYBRID PRIVATE LDA TRAINING ALGORITHM

In this section, we first point out the limitations of existing methods in protecting the intermediate statistics in LDA training. Then we present a systematical study on the inherent DP guarantee of CGS-based LDA training on centralized datasets. Finally, we propose a hybrid privacy-preserving algorithm HDP-LDA, which can protect all the intermediate statistics of the whole CGS-based LDA training.

### A. Limitations of the Existing Methods

A direct method to achieve DP in the CGS-based LDA algorithm is to add noise to the inner statistics [40][17], e.g., word counts  $n_k^t$  and  $n_m^k$ , based on which the sampling probability would be computed to perform topic sampling. In [40], the authors provide a general method to achieve DP in Gibbs Sampling, i.e., adding Laplace noise to the sufficient statistics,  $n_k^t$  and  $n_m^k$  in LDA at the beginning of the Gibbs Sampling process. [17] proposes to add Laplace noise to  $n_k^t$

and  $n_m^k$  in the final iteration. Actually, both methods cannot really protect the training process against strong adversaries with full knowledge of the training mechanism and access to intermediate statistics in LDA. The reasons are as follows:

- **Insufficient protection on word-counts.** In some training scenarios, e.g., in the distributed training of CGS-based LDA [44], the topic-word counts  $n_k^t$  would be released frequently to synchronize the training progress among all parties participating in training. In such a scenario, simply adding noise in the first or final iteration cannot sufficiently prevent privacy leakage in the word-count information<sup>3</sup> released in the training process.
- **No protection on the sampled topics.** Before each topic sampling, the CGS algorithm would first access to the word  $w$  to be sampled on and then compute the sampling distribution based on the word-count information  $n_k^t$  and  $n_m^k$  according to Equation (1). Therefore, the sampling process is not a post-process after sanitizing the word-counts since it depends not only on the sanitized word-counts but also on the word  $w$  in the raw dataset. That is to say, the sampled topics would cause additional privacy cost. In some practical scenarios [3], the topic assignments might be also exchanged among different parties to collaboratively train the model.

As analyzed above, we need a privacy-preserving algorithm which can prevent privacy leakage from both the word-count information and the sampled topics. For the former, we can introduce noise in each iteration to protect the word-counts. For the latter, we propose to utilize the inherent privacy of the CGS algorithm to achieve protection.

### B. Model Assumptions

Here we give some assumptions on the adversary model and the neighboring datasets.

1) *Adversary Model:* We assume that the data curator is trustworthy and has no interest in inferring data privacy. But there exists a strong external adversary who can observe the sampled topic assignments  $\mathbf{K}$  and the word-count matrices  $N_k^t$  and  $N_m^k$  in each iteration of the training process. Based on the observed statistics, the adversary attempts to infer the private information of the training data.

2) *Neighboring Datasets:* We construct the neighboring dataset  $D'$  by replacing a single word  $w_r = t \in D$  by  $t'$  and call this process as *word replacement*. And we expect to prevent the adversary from detecting the impact of this *word replacement* on the CGS algorithm and thus stealing the sensitive information.

### C. Inherent Privacy of CGS Training Algorithm

In the following, we will systemically study the inherent privacy of CGS-based LDA training algorithm.

<sup>3</sup>For simplicity, we also say "protect the word-count information".

TABLE I  
NOTATIONS

$\alpha, \beta$	hyper-parameters for LDA
$\mathcal{W}$	word space
$\mathcal{K}$	topic space
$\phi_k$	topic-word distribution for topic $k$
$\phi_k^t$	probability that word $t$ is generated by topic $k$
$n_m^k$	count of words with topic $k$ in document $d_m$
$n_k^t$	count of word $t$ with topic $k$ in corpus $D$
$n_t$	total word count of $t \in D$
$p_k$	probability that topic $k$ is sampled
$w_r$	replaced word
$\mathbf{w}^+$	set of related words
$\mathbf{w}^-$	set of unrelated words
$\epsilon_i^r$	inherent privacy loss on $w_r$ in the $i$ -th iteration
$\epsilon_i^t, \epsilon_i^{t'}$	inherent privacy loss on related words $t$ or $t'$

1) *Basic Idea*: It has been shown that, Gibbs sampling process has some degree of inherent DP for free [40] since it works in the same way as an exponential mechanism for DP. As a variant, Collapsed Gibbs Sampling (CGS) naturally inherits this property. Intuitively, CGS conducts random sampling iteratively to learn a topic-word distribution, which acts as a mechanism that probabilistically outputs a topic from the topic set. As described in Section III-B, the exponential mechanism works by probabilistically sampling an answer from the answer set. This intuitive consistency motivates us to analyze CGS process in the view of the exponential mechanism.

Without loss of generality, we consider the sampling process of any word  $w$  in the  $i$ -th iteration. Suppose its sampling distribution on  $K$  topics is  $\mathbf{P} = (p_1, p_2, \dots, p_K)^\top$ , where  $p_k$  represents the probability that topic  $k$  is sampled. Define a function  $u(w, k) = \log p_k$  with sensitivity  $\Delta u_k \neq 0$ , then  $p_k$  could be written as

$$\begin{aligned} p_k &= \exp(\log p_k) = \exp\left(\frac{\Delta u_k}{\Delta u_k} \log p_k\right) \\ &= \exp\left(\frac{\Delta u_k \cdot u(w, k)}{\Delta u_k}\right). \end{aligned} \quad (3)$$

According to Definition 2, Equation (3) can be viewed as the probability that an exponential mechanism  $\mathcal{M}_E(w, u, \mathcal{K}) : \mathcal{W} \rightarrow \mathcal{K}$  outputs the topic  $k \in \mathcal{K}$  according to its utility  $u(w, k) = \log p_k$ . And  $\mathcal{M}_E(w, u, \mathcal{K})$  provides  $(\Delta u_k)$ -DP. Notably, if taking the unnormalized probability  $r_k$ , e.g.,  $p_k \propto r_k$ , to define the utility function as  $u(w, k) = \log r_k$ , then  $\mathcal{M}_E(w, u, \mathcal{K})$  preserves  $2(\Delta u_k)$ -DP.

This observation enables us to calculate the inherent privacy of the CGS algorithm. To do so, we first investigate the inherent privacy loss in each iteration of the CGS training algorithm and then compose the privacy in total iterations by utilizing the composition theorem of DP.

2) *Inherent Privacy in Each Iteration*: In the  $i$ -th iteration, to bound the inherent privacy, it requires analyzing the impact of the *word replacement* on each sampling. According to the relationship with the replaced word, we consider the following three cases for each word in  $D$  respectively.

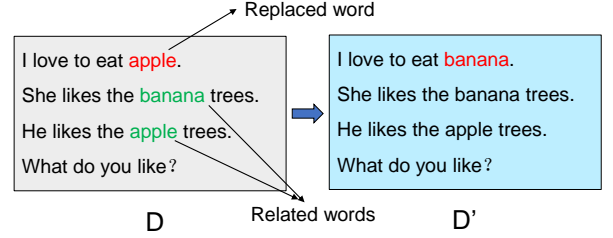


Fig. 3. An Example of Neighboring Datasets

- *Replaced word* is defined as the word  $w_r = t \in D$  which is replaced by  $t'$  in the neighboring dataset  $D'$ . For example, in Fig. 3, the word *apple* marked in red is the *Replaced word* in  $D$ .
- *Related words* refer to the words in  $\mathbf{w}^+$  which satisfies  $\{w = t \text{ or } w = t', \forall w \in \mathbf{w}^+\}$ .  $\mathbf{w}^+$  doesn't contain the *Replaced word*. The words *apple* and *banana* marked in green shown in Fig. 3 are *related words* in  $D$ .
- *Unrelated words* are the words in  $\mathbf{w}^-$  which satisfies  $\{w \neq t \text{ and } w \neq t', \forall w \in \mathbf{w}^-\}$ . All the rest words in  $D$  marked in black are unrelated words in the example shown in Fig. 3.

To analyze the privacy loss on the three types of words, the basic idea is to utilize the consistency between each topic sampling process and the exponential mechanism of DP as observed in Section IV-C1.

First, we analyze the privacy loss incurred by the topic sampling on the *replaced word*  $w_r$ . In particular, in  $D$ , this sampling is performed on  $w_r = t$  while in  $D'$ , this sampling is performed on  $w_r = t'$ . Here, we present a proposition to compute the incurred privacy loss in this sampling.

**Proposition 1.** Suppose  $D'$  is constructed from  $D$  by replacing  $w_r = t \in D$  by  $t'$ , then the privacy loss  $\epsilon_i^r$  incurred by the topic sampling on the *replaced word*  $w_r$  in the  $i$ -th iteration can be bounded by

$$\epsilon_i^r \leq 2 \max_k \left\{ \left| \log \frac{n_k^{t'} + \beta}{n_k^t + \beta} \right| \right\}, \quad (4)$$

where  $n_k^{t'}$  and  $n_k^t$  represent the counts of topic  $k$  assigned to  $t'$  and  $t$  in  $D$ , respectively.

*Proof.* Shown in Appendix A.  $\square$

Then, we consider the privacy loss incurred by the topic sampling on the *related words*  $\mathbf{w}^+$ . For each word  $w \in \mathbf{w}^+$ , the *word replacement* impacts the topic sampling process through the topic-word counts  $n_k^t$  and  $n_k^{t'}$ , which would be used to compute the sampling distribution.

In particular, for some specific topic  $k$ , suppose the topic-word count for  $t$  and  $t'$  are  $n_k^t$  and  $n_k^{t'}$  respectively in  $D$ , the corresponding topic-word count would be  $n_k^t - 1$  and  $n_k^{t'} + 1$  in  $D'$ . Here, we also give a proposition to compute the inherent privacy loss.

**Proposition 2.** The privacy loss in the CGS sampling process on each related word  $w = t \in \mathbf{w}^+$  or  $w = t' \in \mathbf{w}^+$  in the



$i$ -th iteration can be bounded by

$$\epsilon_i^t \leq 2 \log(1 + \frac{1}{\beta}), \quad \epsilon_i^{t'} \leq 2 \log(1 + \frac{1}{\beta}),$$

where  $\epsilon_i^{t'}$  and  $\epsilon_i^t$  represent the privacy loss on  $w = t'$  and  $w = t$  in  $\mathbf{w}^+$  respectively.

*Proof.* The proof logic is similar to that of Proposition 1.  $\square$

Finally, we consider the *unrelated words*  $\mathbf{w}^-$ . Actually, for each word  $w = \hat{t} \in \mathbf{w}^-$ , the *word replacement* could not impact the sampling process on  $w = \hat{t}$ .

So far, we have analyzed the impacts of *word replacement* on the sampling process on all three types of words involved in the  $i$ -th iteration. Based on the impact analysis, the following theorem can bound the total privacy loss in the  $i$ -th iteration.

**Theorem 1.** Given the *replaced word*  $w_r = t \in D$ , the total privacy loss  $\epsilon_i$  incurred by CGS algorithm  $\mathcal{S}_i$  in the  $i$ -th iteration could be bounded by

$$\begin{aligned} \epsilon_i &\leq \epsilon_i^r + (n_t - 1)\epsilon_i^t + n_{t'}\epsilon_i^{t'} \\ &\leq 2\{\log(\frac{\max_{k,t'}\{n_k^{t'}\}}{\beta} + 1) \\ &\quad + (\max_{t'}\{n_{t'}\} + n_t - 1) \cdot \log(1 + \frac{1}{\beta})\}, \end{aligned} \quad (5)$$

where  $n_t$  and  $n_{t'}$  denote the total word-counts of  $t$  and  $t'$  appearing in  $D$  respectively.  $n_k^{t'}$  denotes the total count of topic  $k$  assigned to  $t'$  in  $D$  when performing sampling on the *replaced word*  $w_r$ .

*Proof.* Shown in Appendix B.  $\square$

Theorem 1 demonstrates that the inherent privacy loss in a single CGS iteration is the sum of the privacy losses on the *replaced word* and all the *related words*. And as we can see, the inherent privacy loss can be adjusted by the hyper parameter  $\beta$ . The larger  $\beta$ , the less privacy loss, and vice versa. That is reasonable since  $\beta$  represents the prior information for the topic-word count and a larger  $\beta$  means that the sampling process is mainly determined by the prior information instead of the training data. Then, less data information can be inferred in the sampling process.

3) *Inherent Privacy in Total CGS Training:* So far, we have identified an upper bound for the privacy loss in a single iteration. Now, based on the composition theorem of DP, we can bound the total privacy loss in the whole training process with multiple iterations.

**Theorem 2.** Suppose the privacy loss of each word  $w$  in the  $i$ -th iteration is  $\epsilon_i^w$ , then the total privacy loss  $\epsilon$  in the whole training process with  $n$  iterations could be bounded by

$$\epsilon \leq \max_{w \in D} \left\{ \sum_{i=1}^n \epsilon_i^w \right\}. \quad (6)$$

*Proof.* Shown in Appendix C.  $\square$

4) *Limitations of Inherent Privacy:* We have systemically analyzed the inherent privacy of the CGS training algorithm. Here, we present two limitations of the inherent privacy of CGS regarding the privacy protection.

- **Rapid Accumulation of Privacy Loss.** As shown in the second term of Equation (5), the privacy loss accumulates almost linearly with respect to the word count  $n_t$  and  $n_{t'}$ . This is mainly because the *word replacement* impacts not only the replaced word, but also all the related words in the sampling process.
- **Lack of Protection on Word-count Information.** Obviously, the inherent privacy cannot protect the word-count information, e.g.,  $n_k^t$ , which also reflects the raw data information. Since there is no noise introduced, the *word replacement* would incur an unmaskable change on the word-counts, e.g.,  $n_k^t - 1$  and  $n_k^{t'} + 1$ , thus further causing the privacy leakage if we assume an adversary who can observe that.

#### D. Hybrid Privacy-preserving LDA Training

In Section IV-A, we summarized the limitations of the existing works in protecting the sampled topics, while in Section IV-C4, we identified that the inherent privacy is lack of protection on the word-count information. To address these issues, we propose a hybrid privacy-preserving algorithm named HDP-LDA that combines the inherent privacy of CGS approach and external privacy based on noise injection. The basic idea is to introduce a proper noise in each iteration of CGS to protect the word-count information while mitigating the rapid privacy loss accumulation of inherent privacy. The detailed algorithm process is shown in Algorithm 1. Two main operations in the algorithm are explained as follows.

- **Adding noise:** As analyzed, both limitations of inherent privacy are caused by the difference on the topic-word counts  $n_k^t$  between  $D$  and  $D'$ . Therefore, we introduce some noise to obfuscate the difference on  $n_k^t$  in each iteration (Line 9 in Algorithm 1). Theorem 3 proves that Algorithm 1 can mitigate the rapid privacy loss accumulation of inherent privacy.
- **Clipping:** Quantifying the inherent privacy requires the knowledge of the upper bound of topic-word counts  $\max_{k,t}\{n_k^t\}$ , as shown in Equation (5). A natural bound for  $n_k^t$  is the total count  $n_t$  of word  $t$ , which, however, might be too loose if there are too many  $K$  topics in total. Therefore, we resort to a clipping method to limit the inherent privacy in each iteration. Notably, this clipping is only performed on a copy of  $n_k^t$  for the computation of sampling distribution (Line 12), but does not impact the subsequent updating of  $n_k^t$  in CGS (Lines 15 ~ 16).

**Theorem 3.** Algorithm 1 satisfies  $(\epsilon_L + \epsilon_I)$ -DP in each iteration, in which

$$\epsilon_I = 2 \log\left(\frac{C}{\beta} + 1\right) \quad (7)$$

denotes the inherent privacy loss,  $\epsilon_L$  denotes the privacy loss incurred by the Laplace noise, and  $C$  denotes clipping bound for  $n_k^t$ .

*Proof.* Shown in Appendix D.  $\square$

As shown in Theorem 3, the privacy loss in HDP-LDA consists of two parts: privacy loss  $\epsilon_L$  incurred by the Laplace

**Algorithm 1: HDP-LDA**


---

**Input:** Document corpus  $D$ , Prior parameters  $\alpha, \beta$ , Topic number  $K$ , Clipping bound  $C$

**Output:** Trained topic-word distribution  $\Phi$ , Privacy loss  $\epsilon = T \cdot (\epsilon_L + \epsilon_I)$

// **Initialization**

```

1 for  $d_m \in D$  do
2   for  $w = t \in d$  do
3     Sample topic:  $k \sim \text{Mult}(\frac{1}{K} \cdot \mathbf{I}_K)$ ;
4     Initialize word counts  $n_k^t$  and  $n_m^k$ ;
5   end
6 end

```

// **Collapsed Gibbs Sampling**

```

7 Set  $iter = 0$ ;
8 while  $iter < T$  do
9   Add noise to each  $n_k^t$  independently:
      $n_k^t \leftarrow n_k^t + \eta, \quad \eta \sim \text{Lap}(2/\epsilon_L)$ ;
10  for  $d \in D$  do
11    for  $w = t \in d$  do
12      Clip:  $(n_k^t)^{temp} \leftarrow \min\{n_k^t, C\}$ ;
13      Compute sampling distribution  $\mathbf{p}$ :
          $p_k \propto \frac{(n_k^t)^{temp} + \beta}{\sum_{t=1}^V (n_k^t + \beta)} \cdot \frac{n_m^k + \alpha}{\sum_{k=1}^K (n_m^k + \alpha)}$ ;
14      Compute inherent privacy loss:
          $\epsilon_I \leftarrow 2 \log(\frac{C}{\beta} + 1)$ ;
15      Sample topic and update word count  $n_k^t$ ;
16    end
17  end
18   $iter \leftarrow iter + 1$ ;
19 end
20 Compute the trained model  $\Phi$ ;

```

---

noise and the inherent privacy loss  $\epsilon_I$  of CGS algorithm. Equation (7) presents the inherent privacy loss. Comparing with Equation (5), we can see that the rapid accumulation of inherent privacy loss has been mitigated since the second term of Equation (5) has been removed.

## V. LP-LDA: LDA MODEL TRAINING WITH LDP

We have developed a comprehensive privacy protection approach for protecting the CGS training process on a centralized curated dataset, where the central server is assumed trustworthy. Nevertheless, in many distributed applications, the data curator may not be reliable and the individual data owners may be reluctant to share their sensitive data. In this case, we propose an LDP solution of LP-LDA for LDA that can train on crowdsourced data with LDP. LP-LDA is constituted by two parts: local perturbation at the user side and training on reconstructed dataset at the server side.

### A. Local Perturbation

The local perturbation at the user side includes the following steps:

- *Step 1.* Each document  $m$  is encoded as a binary vector  $\mathbf{V}_m$ , in which each bit  $\mathbf{V}_m[j]$  represents the presence of the  $j$ -th word in the word bag of the corpus.

- *Step 2.* Each bit  $\mathbf{V}_m[j]$  of the binary vector  $\mathbf{V}_m$  is then randomly flipped according to the following randomized response rule:

$$\hat{\mathbf{V}}_m[j] = \begin{cases} \mathbf{V}_m[j], & \text{with probability of } 1 - f \\ 1, & \text{with probability of } f/2 \\ 0, & \text{with probability of } f/2 \end{cases}$$

where  $f \in [0, 1]$  is a parameter that specifies the randomness of flipping and adjusts the local privacy level.

- *Step 3.* Then the noisy binary vector  $\hat{\mathbf{V}}_m[j]$  is sent to the central server by each user. Obviously,  $\mathbf{V}_m[j]$  is locally sanitized without concerning user's privacy.

### B. Training on the Reconstructed Dataset

After receiving the flipped binary vectors from a large number of data contributors, the central server can aggregate the vectors, reconstruct the dataset, and then perform training on the reconstructed dataset. The rationale behind this is that the training result of topic-word distribution is insensitive to the document partitions and only depends on the total word counts in the corpus.

- *Step 1.* For each bit in the noisy binary vectors, the server counts the number of 1's as  $n_t = \sum_{m=1}^M \hat{\mathbf{V}}_m[t]$ .
- *Step 2.* The server then estimates the true count  $N_t$  of each bit in the original binary vectors  $\mathbf{V}_m$  as  $\hat{N}_t = (2n_t - fM)/2(1 - f)$ .
- *Step 3.* For each bit, the server first computes the difference  $\delta_t = \hat{N}_t - n_t$ .
- *Step 4.* For each bit  $t$ , if  $\delta_t > 0$ , the server randomly samples  $\delta_t$  binary vectors with the  $t$ -th bit as 0 and sets the  $t$ -th bit as 1; if  $\delta_t < 0$ , then the server randomly samples  $|\delta_t|$  binary vectors with the  $t$ -th bit as 1 and sets the  $t$ -th bit as 0; otherwise, keeps the noisy bit vectors as received.
- *Step 5.* Based on the noisy bit vectors, the server reconstructs a dataset and performs the CGS process on it.

Algorithm ?? presents the detailed procedures of LP-LDA on both the user side and the server side.

### C. Privacy Analysis of LP-LDA

**Theorem 4.** LP-LDA satisfies  $\log \frac{1-f/2}{f/2}$ -LDP for each word, and  $V \cdot \log \frac{1-f/2}{f/2}$ -LDP for each document.

*Proof.* Suppose a word  $t$  appears in a noisy bit vector, then the probability of it being kept from the original bit vector is  $\Pr(\hat{\mathbf{V}}_m[t] = 1 | \mathbf{V}_m[t] = 1) = 1 - f/2$  and the probability of it being flipped from the original bit vector is  $\Pr(\hat{\mathbf{V}}_m[t] = 1 | \mathbf{V}_m[t] = 0) = f/2$ . Then, according to the definition, it guarantees the local differential privacy of

$$\epsilon = \left| \log \frac{\Pr(\hat{\mathbf{V}}_m[t] = 1 | \mathbf{V}_m[t] = 1)}{\Pr(\hat{\mathbf{V}}_m[t] = 1 | \mathbf{V}_m[t] = 0)} \right| = \log \frac{1 - f/2}{f/2}.$$

The analysis holds for any bit  $t$  that  $\hat{\mathbf{V}}_m[t] = 0$ .

Each bit of the document  $\mathbf{V}_m$  is perturbed independently. Then according to the sequential composition theorem of DP, LP-LDA guarantees  $V \cdot \epsilon$ -local DP for the entire document.  $\square$

Since the reconstruction and training process are essentially post-processes on the noisy bit vectors, the LDP guarantee remains unchanged for all the documents.

#### D. Utility Analysis of LP-LDA

**Theorem 5.** Let  $N_t$  and  $n_t$  denote the counts of word  $t$  in the original and perturbed datasets, respectively, then

$$\hat{N}_t = \frac{2n_t - fM}{2(1-f)} \quad (8)$$

is an unbiased estimator of  $N_t$  with the variance of

$$\text{Var}(\hat{N}_t) = \frac{(2-f)fM}{4(1-f)^2} + \frac{(M-N_t)N_t}{M}. \quad (9)$$

*Proof.* Let  $n_1$  denote the count of word  $t$  retained from the real datasets and  $n_2$  denote the noisy part, then  $n_1$  and  $n_2$  follow two Binomial distributions, i.e.,  $n_1 \sim B(N_t, 1-f/2)$ ,  $n_2 \sim B(M-N_t, f/2)$ . Let  $X = n_1 + n_2$ , then its first theoretical moment  $\mathbb{E}(X) = N_t(1-f/2) + (M-N_t) \cdot (f/2)$  and its first sample moment  $\bar{X} = n_t$ . Therefore,

$$\hat{N}_t = \frac{2n_t - fM}{2(1-f)}$$

is the moment estimator as well as an unbiased estimator. Its variance is then

$$\begin{aligned} \text{Var}(\hat{N}_t) &= \frac{\text{Var}(n_t)}{(1-f)^2} = \frac{\text{Var}(n_1 + n_2)}{(1-f)^2} \\ &= \frac{\text{Var}(n_1) + \text{Var}(n_2) + 2\text{Cov}(n_1, n_2)}{(1-f)^2} \\ &= \frac{(2-f)fM}{4(1-f)^2} + \frac{(M-N_t)N_t}{M}. \end{aligned}$$

□

## VI. OLP-LDA: ONLINE LDA MODEL TRAINING WITH LDP

### A. Motivations

In LP-LDA, we consider a static scenario that one-time LDA is trained on the locally sanitized dataset. Here, we consider the following two practical issues that may encounter in LDA model training.

- **Online Training.** Many practical applications require that ML models can be continuously trained on streaming datasets, which are contributed by users in mini-batches and accumulated over time. However, it is usually infeasible to store all streaming batches and perform batch model training such as LP-LDA.
- **Prior Knowledge.** In many real-world scenarios, the training server may not begin model training from zero but possess some prior datasets for building an initial model. These prior datasets may be from the publicly available datasets or the purchased high-quality crowd-sourced datasets, which are often non-private.

Considering the above issues, we further propose OLP-LDA, a privacy-preserving online LDA training algorithm. OLP-LDA aims to not only realize efficient online training on mini-batches with LDP, but also greatly improve the model utility by extracting knowledge from the prior dataset.

OLP-LDA consists of two components: the baseline online LDA training framework O-LDA, and the Bayesian denoising technique for the continuous reconstruction of noisy batches.

### B. O-LDA: Framework of Online LDA Model Training

1) *Basic Idea of Online LDA Training:* Given the prior information  $P(\Theta|D_0)$  of LDA model parameter  $\Theta$  from the prior dataset  $D_0$ , the Online LDA training aims to update the LDA model  $P(\Theta|D)$  with the evolving mini-batch sequence  $D_{1:L} = \{D_1, \dots, D_l, \dots, D_L\}$  where  $D_l$  represents the  $l$ -th mini-batch. Considering the correlations between mini-batches, the online training process could be regarded as a Bayesian learning process, in which  $P(\Theta|D)$  is updated based on a recurrence relationship:

$$P(\Theta|D_{0:l}) \propto P(\theta|D_{0:l-1})P(D_l|\Theta),$$

where the posterior  $P(\Theta|D_{0:l-1})$  learned from  $D_{0:l-1}$  would be used as the prior when learning from  $D_l$ .

2) *O-LDA Generative Model:* To capture the correlations between consecutive mini-batches  $D_{l-1}$  and  $D_l$ , we introduce a correlation factor  $\lambda$ . And the prior parameters  $\beta^l$  for  $D_l$  is represented as the combination of  $\beta^{l-1}$  for  $D_{l-1}$  and the topic-word matrix  $N_{k,t}^{l-1}$  learned from  $D_{l-1}$ . That is

$$\beta^l = \beta^{l-1} + \lambda N_{k,t}^{l-1}. \quad (10)$$

In Equation (10), a larger  $\lambda$  means a stronger dependency between  $D_{l-1}$  and  $D_l$ . When  $\lambda = 0$ , the training result of  $D_{l-1}$  will not influence the training process of  $D_l$ . Apparently,  $\beta^l$  could also be written as

$$\beta^l = \beta \mathbf{1}_{K,V} + \sum_{i=1}^{l-1} \lambda N_{k,t}^i, \quad (11)$$

where  $\beta$  denotes a hyperparameter fixed for the whole corpus.

As a result, we redefine the corpus generative process of the O-LDA model as follows:

- 1) Generate the first mini-batch  $D_1$  according to the standard LDA with given hyperparameters  $\alpha$  and  $\beta$ , and draw a correlation factor  $\lambda \sim U(0, 1)$  for the corpus.
- 2) For each mini-batch  $D_l, l \geq 2$ , generate the topic-word distribution  $\Phi^l \sim \text{Dir}(\beta^l)$  for  $D_l$ .
- 3) For each document  $m$  in mini-batch  $D_l$ , generate all the words according to the standard LDA with hyperparameter  $\alpha$  and topic-word distribution  $\Phi^l$ .

Then, the sampling distribution for word  $w = t$  in mini-batch  $D_l$  shown in Equation (1) should be replaced as

$$p(z_w = k) \propto \frac{n_{k,t}^l + \beta_{k,t}^l}{\sum_{t=1}^V (n_{k,t}^l + \beta_{k,t}^l)} \cdot \frac{n_m^k + \alpha}{\sum_{k=1}^K (n_m^k + \alpha)}, \quad (12)$$

where  $\beta_{k,t}^l$  denotes the element in the  $k$ -th row and  $t$ -th column of  $\beta^l$ . Similar to the standard LDA, the training result  $\Phi$  of topic word distribution is updated by

$$\mathbb{E}[\phi_k^t | D_{1:l}] = \frac{n_{k,t}^l + \beta_{k,t}^l}{\sum_{t=1}^V (n_{k,t}^l + \beta_{k,t}^l)}. \quad (13)$$

The parameter  $\lambda$  could be optimized with each mini-batch to reach a better performance using the MCMC method [45].



### C. Bayesian Denoising Scheme

1) *Basic Idea of Bayesian Denoising:* Recall that the locally sanitized dataset can be reconstructed based on the moment estimation of the word counts in LP-LDA (Section V-B). Bayesian denoising scheme aims to further improve this estimation, i.e., reducing the variance in Equation (9), with the knowledge in the prior dataset. The denoising process can be also viewed as a Bayesian estimation problem, where the word count  $N_t$  is the underlying parameter for a prior distribution  $\pi(N_t)$  and the observation  $n_t$ . The objective is to find an estimation function  $B(n_t)$  of  $n_t$  to minimize the following Bayes risk:

$$R(\pi(N_t)) = \min_{B(n_t)} \mathbb{E}_{N_t} [\mathbb{E}_{n_t|N_t} [\|B(n_t) - N_t\|^2 | N_t]].$$

According to [46], the optimal estimator can be calculated as the posterior expectation

$$\mathbb{E}[N_t | n_t] = \arg \min_{B(n_t)} R(\pi(N_t))$$

which depends on the prior distribution  $\pi(N_t)$  and the likelihood function  $P(n_t | N_t)$ .

2) *Prior Distribution:* The prior distribution  $\pi(N_t)$  can be assumed to follow a Gaussian distribution  $N(\mu_t, \sigma^2)$  for simplicity, where  $\mu_t$  denotes the word count information extracted from the prior dataset and  $\sigma^2$  represents the level of belief to the prior information. Gaussian distribution is a common assumption and  $\sigma^2$  could be adjusted according to the practical requirement.

3) *Likelihood Function:* The likelihood function  $P(n_t | N_t)$  can be computed as follows. As did in LP-LDA, each bit of the binary word vector transformed from the document has a probability of  $f/2$  to be flipped at the user side, so the likelihood function of the observed noisy word count could be written as

$$\begin{aligned} P(X = n_t | N_t) &= \sum_{i=0}^{\min\{N_t, n_t\}} p(X_1 = i, X_2 = n_t - i) \\ &= \sum_{i=0}^{\min\{N_t, n_t\}} \binom{N_t}{i} \left(1 - \frac{f}{2}\right)^i \left(\frac{f}{2}\right)^{N_t - i} \times \\ &\quad \binom{M - N_t}{n_t - i} \left(1 - \frac{f}{2}\right)^{M - N_t - n_t + i} \left(\frac{f}{2}\right)^{n_t - i}, \end{aligned} \quad (14)$$

where  $X_1$  denotes the count of word  $t$  retained from the real datasets,  $X_2$  denotes the noisy part and  $X = X_1 + X_2$ , and  $M$  denotes the document number in this mini-batch. Note that  $X_1$  and  $X_2$  follow two Binomial distributions

$$X_1 \sim B(N_t, 1 - f/2) \quad X_2 \sim B(M - N_t, f/2). \quad (15)$$

It is still intractable to present the posterior  $\mathbb{E}[N_t | n_t]$  in a closed form due to the complicated likelihood function in Equation (14). To tackle this problem, we consider deriving an approximate results based on the following Gaussian conjugate property.

**Lemma 1.** (Gaussian Conjugate Property [47]) Given a marginal Gaussian distribution  $p(X)$  and a conditional Gaus-

sian distribution  $p(Y|X)$ , the conditional distribution  $p(X|Y)$  satisfies that

$$p(X|Y) = N(X | w\mu_0 + (1-w)\frac{Y-b}{a}, \frac{\sigma_0^2\sigma_1^2}{\sigma_1^2 + a^2\sigma_0^2}),$$

if it holds that,

$$p(X) = N(X | \mu_0, \sigma_0^2), \quad p(Y|X) = N(Y | aX + b, \sigma_1^2),$$

where  $w = \frac{\sigma_1^2}{\sigma_1^2 + a^2\sigma_0^2}$ .

Lemma 1 implies that the posterior of a parameter  $\theta$  will be also a Gaussian distribution when the prior distribution of  $\theta$  and the likelihood function of the observed data conditioned on  $\theta$  are based on Gaussian distribution. Based on Lemma 1, we consider approximating the binomial distributions shown in Equation (15) using Gaussian distributions. The following lemma provides theoretical support for such approximation.

**Lemma 2.** (De Moivre-Laplace Central Limit Theorem) Suppose that in a series of  $n$  independent Bernoulli trials, event A has a probability  $p$  of occurrence ( $0 < p < 1$ ) in each trial. Denote  $S_n$  as the occurrence time of A, and let

$$Y_n^* = \frac{S_n - np}{\sqrt{npq}},$$

where  $q = 1 - p$ . Then for each real number  $y$ , it holds that

$$\lim_{n \rightarrow \infty} P(Y_n^* \leq y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^y e^{-t^2/2} dt.$$

Lemma 2 states that a binomially distributed random variable approximates the Gaussian random variable with mean  $np$  and standard deviation  $\sqrt{npq}$  as  $n$  grows larger. It has been validated that when  $np > 5$  or  $n(1-p) > 5$ , this approximation is reasonable and effective. Based on Lemma 2, both binomial distributions shown in Equation (15) could be approximated as:

$$\begin{aligned} X_1 &\sim N(N_t(1 - \frac{f}{2}), N_t \cdot \frac{f}{2}(1 - \frac{f}{2})) \\ X_2 &\sim N((M - N_t)\frac{f}{2}, (M - N_t)\frac{f}{2}(1 - \frac{f}{2})). \end{aligned} \quad (16)$$

Furthermore, according to the additive property of Gaussian distributions, the word count variable  $X$  should follow a Gaussian distribution

$$X \sim N((M - N_t)\frac{f}{2} + N_t(1 - \frac{f}{2}), \sigma_p^2), \quad (17)$$

where  $\sigma_p^2 = M\frac{f}{2}(1 - \frac{f}{2}) + 2cov(X_1, X_2)$  and then the likelihood function shown in Equation (14) could be approximated as:

$$P(X = n_t | N_t) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{[2X - Mf - 2N_t(1 - f)]^2}{4\sigma_p^2}\right). \quad (18)$$

**Theorem 6.** Suppose the prior distribution of count  $N_t$  of word  $t$  is set as  $N(\mu_t, \sigma^2)$ , and the likelihood function of the noisy count  $n_t$  is approximated by Gaussian distribution, then

the posterior distribution of  $N_t$  should be as follows:

$$N(\omega \cdot \mu_t + (1 - \omega) \frac{2n_t - fM}{2(1 - f)}, \frac{\sigma^2 \sigma_p^2}{\sigma_p^2 + (1 - f)^2 \sigma^2}),$$

and then the Bayesian estimator

$$B(n_t) = \omega \cdot \mu_t + (1 - \omega) \frac{2n_t - fM}{2(1 - f)}, \quad (19)$$

where  $\omega = \frac{Mf(2-f)}{Mf(2-f) + 4\sigma^2(1-f)^2}$ .

*Proof.* Based on Equation (18) and Lemma 2, the result could be derived directly.  $\square$

Recall the moment estimator of the word count  $N_t$  shown in Equation (8), we can observe that the Bayesian estimator of  $N_t$  is a linear combination of the moment estimator and the prior  $\mu_t$ . And the weight  $\omega$  of  $\mu_t$  is controlled by the prior parameter  $\sigma^2$  which is adjustable according to practical requirement. Intuitively, as  $\sigma^2$  grows larger, the belief to the prior information becomes weaker, and the weight of  $\mu_t$  becomes smaller.

**Theorem 7.** The Bayesian estimator  $B(n_t)$  of  $N_t$  in Equation (19) has a variance of

$$\text{Var}(B(n_t)) = (1 - \omega)^2 \left( \frac{(2 - f)fM}{4(1 - f)^2} + \frac{(M - N_t)N_t}{M} \right), \quad (20)$$

where  $\omega = \frac{Mf(2-f)}{Mf(2-f) + 4\sigma^2(1-f)^2}$ .

*Proof.* Since  $\mu_t$  is a constant, then

$$\text{Var}(\omega \cdot \mu_t + (1 - \omega)\hat{N}_t) = (1 - \omega)^2 \cdot D(\hat{N}_t),$$

where  $\hat{N}_t$  denotes the estimator of  $N_t$  in Equation (8).  $\square$

Theorem 7 reveals that if we replace the moment estimator in Equation (8) by the Bayesian estimator in Equation (19) to implement the dataset reconstruction process, the variance term could be reduced to  $(1 - \omega)^2$  of that of the moment estimator.

## D. OLP-LDA Algorithm

We present the online LDA algorithm on locally private mini-batches by embedding the Bayesian denoising process into the training procedures of O-LDA.

Let  $D_0$  be the prior dataset,  $D_l$  be the  $l$ -th locally sanitized mini-batch arriving at the central server,  $N_t^l$  be the real count of word  $t$  in  $D_l$ ,  $\mu_t^l$  be the prior parameter for  $N_t^l$ , shown as the mean of the prior distribution, and  $\eta_t^l$  be the Bayesian estimation  $B(n_t^l)$  of  $N_t^l$  in Equation (19). We give the updating rule of  $\mu_t^l$  as follows:

$$\mu_t^l = \frac{\mu_t^{l-1} + \eta_t^{l-1}}{2} \times \frac{|D_l|}{|D_{l-1}|}, \quad (21)$$

where  $|D_l|$  refers to the documents number in mini-batch  $D_l$ . Especially, it holds that  $\mu_t^0 = \eta_t^0$  since  $D_0$  is possessed by the central server as the clean dataset and no need to be reconstructed. Algorithm 2 shows the details of our proposed OLP-LDA.

## E. Analysis of OLP-LDA

We discuss the impacts of the mini-batch size and the prior data size on the model utility in terms of Bayesian denoising.

1) *Impact of Mini-batch Size:* Given a mini-batch  $D_l$ , the Bayesian estimation  $B(n_t^l) = \frac{B(n_t^l)}{|D_l|} |D_l|$  of  $N_t^l$  is used to perform the denoising. Obviously, more accurate frequency estimation  $\frac{B(n_t^l)}{|D_l|}$  would lead to more accurate denoising.

According to Equation (20), the variance of  $\frac{B(n_t^l)}{|D_l|}$  should be:

$$\text{Var}\left(\frac{B(n_t^l)}{|D_l|}\right) = (1 - \omega)^2 \left( \frac{(2 - f)f}{4(1 - f)^2 |D_l|} + \frac{(|D_l| - N_t^l)N_t^l}{|D_l|^3} \right), \quad (22)$$

where

$$(1 - \omega)^2 = \left( \frac{4\sigma^2(1 - f)^2}{f(2 - f)|D_l| + 4\sigma^2(1 - f)^2} \right)^2.$$

As seen, the larger  $|D_l|$ , the smaller variance of  $\frac{B(n_t^l)}{|D_l|}$ , thus the more accurate denoising.

2) *Impact of Size of Prior Dataset:* Given a prior dataset  $D_0$ , if we take a random variable  $X_t^i \in \{0, 1\}$  to represent whether a given word  $t$  appears in the  $i$ -th document in  $D_0$ , then according to the law of large numbers, the mean of  $X_t^i$  in the prior dataset should tend to the real frequency statistic  $p_t$  in the whole corpus as the size  $D_0$  keeps increasing, that is

$$\lim_{|D_0| \rightarrow \infty} \Pr\left\{ \left| \frac{1}{|D_0|} \sum_{i=0}^{|D_0|} X_t^i - p_t \right| < \epsilon \right\} = 1, \forall \epsilon > 0, \quad (23)$$

where  $\frac{1}{|D_0|} \sum_{i=0}^{|D_0|} X_t^i = \frac{N_t^0}{|D_0|}$ . That means that the larger prior dataset can extract more accurate word-count information. Then, according to Equation (21) and  $\mu_t^0 = \eta_t^0$ , the more accurate prior information would be generated for the subsequent noisy mini-batches.

Therefore, we can conclude that both the larger size of the prior dataset and mini-batches could further improve the model utility.

## VII. EVALUATION

In this section, we conducted extensive simulation experiments on four real-world datasets to evaluate the effectiveness of our proposed algorithms.

### A. Experiment Setup

**Datasets:** Four real-world datasets were used in our experiments:

- KOS<sup>4</sup> contains 3,430 blog entries from dailykos website.
- NIPS<sup>5</sup> contains 1,500 research papers from NIPS conference.
- Enron<sup>6</sup> contains email messages from about 150 users. The first 10,000 documents are extracted for experiments.

<sup>4</sup><http://archive.ics.uci.edu/ml/datasets/KOS>

<sup>5</sup><http://archive.ics.uci.edu/ml/datasets/NIPS>

<sup>6</sup><http://archive.ics.uci.edu/ml/datasets/Enron>

**Algorithm 2: OLP-LDA**


---

**Input:** Flipping probability  $f$ , Hyperparameters  $\alpha, \beta$ , Topic number  $K$ , Correlation factor  $\lambda$ , Reconstruction factor  $\omega$ , Prior dataset  $D_0$ , Local dataset  $D$

**Output:** Training results  $\Phi^{0:\infty}$

// On the user side

- 1 **foreach** document  $d \in D$  **do**
- 2      $\hat{d} = RR(d)$ ;     // Randomized response
- 3     Upload  $\hat{d}$  to the server;
- 4 **end**

// On the server side

- 1 Train initial model:  $\Phi^0, N_{k,t}^0 = LDA(D_0, \alpha, \beta, K)$ ;     // Standard LDA training algorithm
- 2 **foreach** word  $t$  **do**
- 3     Initialize  $\mu_t^0 = \eta_t^0 = \sum_k N_{k,t}^0$ ;
- 4 **end**
- 5 **for**  $l=1 : \infty$  **do**
- 6     **for**  $\hat{d}$  uploaded in the  $l$ -th time window **do**
- 7         Aggregate  $\hat{d}$ :  $D_l = \{\hat{d}_1, \hat{d}_2, \dots, \hat{d}_{N_l}\}$ ;
- 8         Generate prior for  $D_l$  acc. to Equation (21);
- 9         Reconstruct  $D_l$  by Bayesian denoising scheme acc. to Equation (19);
- 10         Update  $\beta = \beta + \lambda N_{k,t}^{l-1}$ ;
- 11         Train model:  $\Phi^l, N_{k,t}^l = LDA(\alpha, \beta, K)$ ;
- 12         **foreach** word  $t$  **do**
- 13             Compute  $\mu_t^l = \eta_t^{l-1}$ ;
- 14             Compute  $\eta_t^l = B(n_t)$  acc. to Equation (19);
- 15         **end**
- 16     **end**
- 17 **end**

---

TABLE II  
DETAILS OF THE DATASETS

Dataset	#. words	#. training docs	#. test docs
KOS	209169	3000	430
NIPS	410753	1350	150
Enron	356363	8000	2000
FVENA	386061	1350	450

- FVENA<sup>7</sup> contains books written by 50 authors in the 19th century. 1,800 documents from the first and eighth authors (900 for each) are extracted for experiments.

We extracted part of these datasets as our training datasets and the rest as the testsets. For simplicity, we conducted a pre-processing on these datasets, in which, all stop words were removed, and 1,000 most frequent words in each dataset were chosen as the corresponding vocabulary list. Details about the datasets after pre-processing can be found in Table II.

**Simulation Methodology:**

For HDP-LDA, we designed a simple *Topic-based Attack* algorithm to validate its performance on protecting the sampled

**Algorithm 3: Topic-based Attack Algorithm**


---

**Input:** The attacked word  $w_{mn}$

**Output:** The inferred result  $\bar{t}$

- 1 Set  $i = 0$ ;
- 2 **while**  $i < T$  **do**
- 3     Add noise:  $(\hat{n}_k^t)_i \leftarrow n_k^t + \eta$ ,  $\eta \sim Lap(2/\epsilon_L)$ ;
- 4     Record  $(\hat{n}_k^t)_i$ ;
- 5     **for**  $w \in D$  **do**
- 6         Sample topic:  $k_i \sim \mathbf{P}$ ;
- 7         **if**  $w = w_{mn}$  **then**
- 8             Record  $k_i$ ;
- 9         **end**
- 10     **end**
- 11      $i \leftarrow i + 1$ ;
- 12 **end**
- 13 Compute  $\bar{t} = \arg \max_t P[w_{mn} = t | k_1, \dots, k_i, \dots, k_T]$  acc. to Equation (24);

---

topics. Specifically, for the  $n$ -th word  $w_{mn} = t$  in document  $d_m$ , we utilized all the sampled topics  $\{k_1, \dots, k_i, \dots, k_T\}$  on  $w_{mn}$  in  $T$  iterations and the sanitized word-counts  $\mathbf{s} = \{(\hat{n}_k^t)_i, i \leq T\}$  to infer  $w_{mn}$ . The detailed algorithm is shown in Algorithm 3. In our experiment, for simplicity, we selected the last word in  $D$  as the attacked word and took the inferred probability  $P[w_{mn} = t | k_1, \dots, k_i, \dots, k_T]$  computed by Equation (24) as the attack accuracy.

$$\begin{aligned}
 & Pr(w_{mn} = t | k_1, \dots, k_i, \dots, k_T) \\
 &= \frac{Pr(w_{mn} = t, k_1, \dots, k_i, \dots, k_T)}{Pr(k_1, \dots, k_i, \dots, k_T)} \\
 &= \frac{\prod_i Pr(w_{mn} = t, k_i)}{\prod_i Pr(k_i)} = \prod_i \frac{Pr(w_{mn} = t, k_i)}{Pr(k_i)} \quad (24) \\
 &\approx \prod_i \frac{(\hat{n}_{k_i}^t)_i / \sum_{k,t} (\hat{n}_k^t)_i}{\sum_t (\hat{n}_{k_i}^t)_i / \sum_{k,t} (\hat{n}_k^t)_i} = \prod_i \frac{(\hat{n}_{k_i}^t)_i}{\sum_t (\hat{n}_{k_i}^t)_i}.
 \end{aligned}$$

For both LDP solutions LP-LDA and OLP-LDA, all documents in the training datasets  $D_{\text{train}}$  were perturbed with the randomized response technique described in Section V-A in a centralized manner to simulate the procedures of crowdsourcing users. Then, for LP-LDA, the reconstruction process and model training were implemented. For OLP-LDA, the prior dataset owned by the server was generated by randomly sampling from  $D_{\text{train}}$  and used for constructing the initial model. The arriving mini-batches were generated by randomly drawing from the sanitized dataset, and then reconstructed according to the Bayesian denoising scheme. Finally, OLP-LDA was trained on those reconstructed mini-batches.

To evaluate the practical performance on privacy protection, the membership inference attack (MIA) [8] was implemented to simulate the inference on trained models. Since MIA works on supervised learning models, LDA models for unlabelled datasets cannot be directly verified. Instead, the privately trained LDA models were incorporated into a classifier to derive an LDA-based classification model, which then acts as

<sup>7</sup><http://archive.ics.uci.edu/ml/datasets/Victorian+Era+Authorship+Attribution>

the target model for inference. In the simulation, the labeled dataset FVENA was used to train and test the target models. The detailed MIA attack process can be referred to [8, 48].

All our experiments were run on a laboratory-based workstation equipped with 10 cores of Intel(R) Xeon(R) E5-2640 v4@2.40GHz and 30GB memory. And the proposed algorithms were implemented with Python (version 3.7). In our experiments, for all datasets, the topic number was set as 50, and the default hyperparameters  $\alpha$ ,  $\beta$  were set as 1, and 0.01, respectively.

**Metrics:** We select **Perplexity** as the metric of LDA utility. Perplexity measures the likelihood that the test data is generated by the trained LDA model. A lower perplexity means a higher likelihood, and hence better model utility. Given a test set  $D_{\text{test}}$  with  $M$  documents, denote  $\phi_k^t$  as the learned parameters from  $D_{\text{train}}$  and  $\theta_m^k$  as the inferred parameters from  $D_{\text{test}}$ , the perplexity on  $D_{\text{test}}$  can be computed as

$$\text{per}(D_{\text{test}}) = \exp\left(-\frac{\sum_m \sum_i \log(\sum_k \theta_m^k \phi_k^{w_i})}{\sum_m |d_m|}\right)$$

where  $|d_m|$  and  $w_i$  denote the number of words and the  $i$ -th word in  $d_m$  respectively.

**Comparison:** To validate our proposed algorithm, we also compare with two algorithms:

- CDP-LDA [40], in which DP is achieved by perturbing the word count matrices  $N_k^t$  and  $N_m^k$  with Laplace noise  $\text{Lap}(1/\epsilon)$  in the first iteration.
- CDP-LDA+, which is an extended version of CDP-LDA, introduces Laplace noise into  $N_k^t$  and  $N_m^k$  in each iteration to protect the training process.

### B. Performance of HDP-LDA

We first validate the performance of our proposed HDP-LDA algorithm in terms of both privacy protection and model utility.

1) *Defense Against Topic-based Attack:* Fig. 4 reports the privacy protection of HDP-LDA by validating its defending ability against the *Topic-based attack*. In Fig. 4, HDP-LDA achieves DP by setting proper clipping bound  $C$  and  $\beta$  to limit the inherent privacy, according to Equation (7). While CDP-LDA+ achieves DP by adding Laplace noise. As shown, for a plain CGS algorithm without any intervention (referred to as Non-Private in Fig. 4) and CDP-LDA+, the attack accuracy curves show sharp increases with the iteration number even in the strong privacy regime ( $\epsilon = 1$ ) of CDP-LDA+. This is because both CDP-LDA+ and Non-Private can not limit the inherent privacy loss in the topic sampling process, which can accumulate rapidly with the iteration number. In contrast, we can see that HDP-LDA can effectively defend against the attack even when  $\epsilon = 10$ . This demonstrates that HDP-LDA which limits the inherent privacy can effectively prevent the privacy leakage from the sampled topics.

2) *Utility vs. DP:* Fig. 5 compares the perplexity of HDP-LDA and CDP-LDA+ under different privacy levels  $\epsilon$  incurred by the Laplace noise. In Fig. 5, HDP-LDA limits the inherent privacy level as 10 in each iteration under which the topic sampling process can be protected as shown in Fig. 4. The

curves marked by Limited Inherent present the perplexity of HDP-LDA with limited inherent privacy as 10 but no Laplace noise. Compared with Non Private (or plain CGS), Limited Inherent shows a utility degradation since it has stronger inherent privacy (through clipping the word-counts and setting the hyper-parameter  $\beta$  larger).

However, the model utility of HDP-LDA is no worse than CDP-LDA+, and much better on KOS dataset, even if CDP-LDA+ incurs more privacy loss than HDP-LDA when taking the inherent privacy loss into account. That is because in HDP-LDA, we set a larger  $\beta$  in the training process as the prior information which could improve the model robustness to the noise.

### C. Performance of LP-LDA

We then present the performance of our proposed LDP based LDA algorithm LP-LDA for static dataset.

1) *Utility vs. DP:* Fig. 6 shows the perplexity of LP-LDA in comparison with CDP-LDA under different levels of local differential privacy. As we can see, the perplexity curves of both LP-LDA and CDP-LDA show monotonous decrease as  $\epsilon$  increases, which illustrates the tradeoff between the model utility and privacy. In particular, for stronger privacy regime with smaller  $\epsilon$ , the compared CDP-LDA performs better than LP-LDA because the injected Laplace noise incurs less randomness than the randomized response technique of LP-LDA. However, for weaker privacy regime with larger  $\epsilon$ , LP-LDA performs much better than CDP-LDA, even close to the non-private LDA model as  $\epsilon$  keeps increasing.

2) *Defense Against MIA:* We also present the privacy guarantee of LP-LDA by verifying the defending ability of its LDA-based classification model against MIA. In particular, we compare the membership inference accuracy between LP-LDA and CDP-LDA. The baseline non-private model was implemented by performing the CGS training process for 300 iterations. For reference, we also present the performance of the target LDA-based classification models.

Fig. 7(a) shows the membership inference accuracy of MIA on the LDA-based classification model generated by LP-LDA. As shown, for both LP-LDA and CDP-LDA methods, the inference accuracy curves are below that of the CGS algorithm, which demonstrates that both privacy-preserving methods can effectively mitigate the inference of MIA. Furthermore, both inference accuracy curves show almost a monotonous increasing trend with the increase of  $\epsilon$ , which demonstrates that smaller  $\epsilon$ -DP could provide stronger defense ability.

Fig. 7(b) and Fig. 7(c) present the model utility (i.e., prediction accuracy and F1 score) of the LDA-based classification model generated by LP-LDA, versus the privacy levels  $\epsilon$ . As shown, both the prediction accuracy and F1-score for LP-LDA and CDP-LDA increase with  $\epsilon$ . This illustrates the general tradeoff between the privacy level and model utility. Similar to the results shown in Fig. 6, the LDA-based classification model of LP-LDA performs better than CDP-LDA in the weaker privacy regime. As  $\epsilon$  keeps increasing, LP-LDA can even approach to the plain CGS algorithm without extra noise.

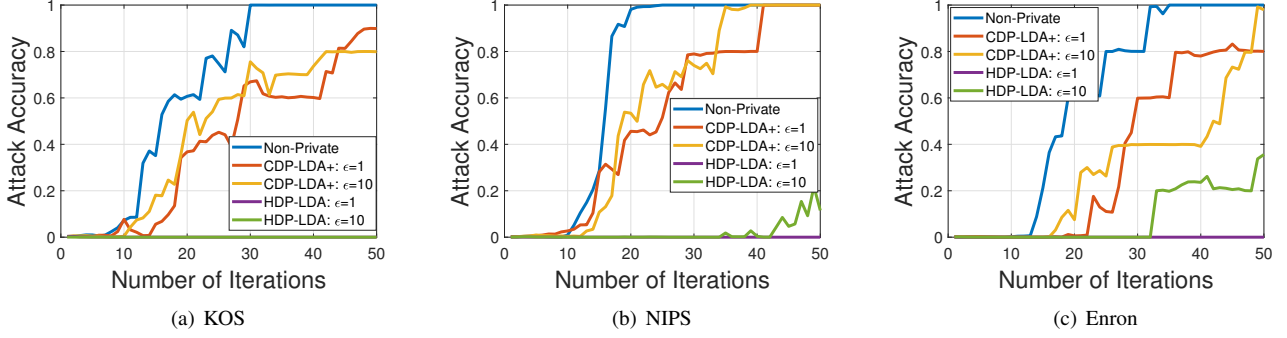


Fig. 4. Defense Against Topic-based Attack

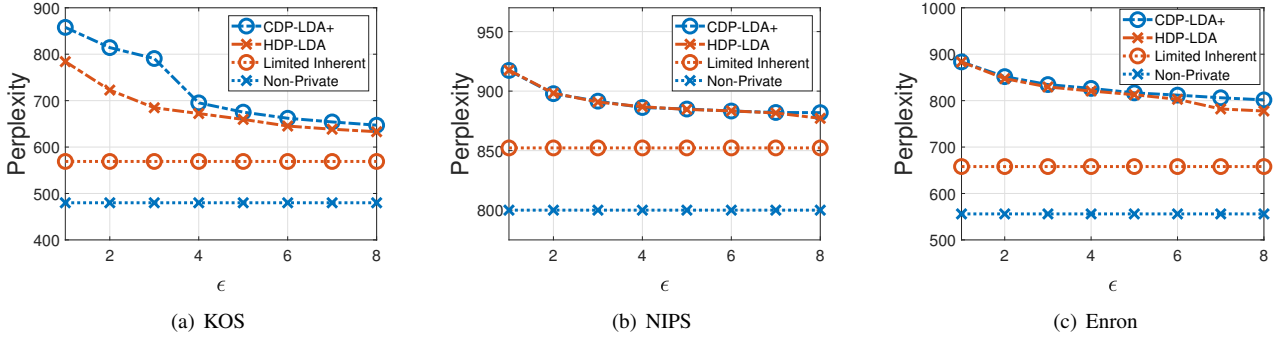


Fig. 5. Perplexity vs. Privacy level of HDP-LDA

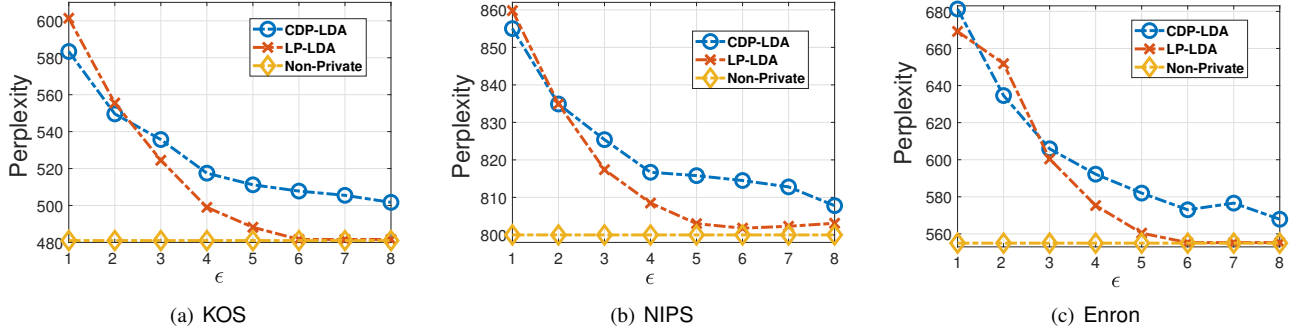


Fig. 6. Perplexity vs. Privacy Level of LP-LDA

#### D. Performance of OLP-LDA

We finally present the performance of our proposed OLP-LDA algorithm. In each experiment, the correlation factor  $\lambda$  was set to 0.5, the reconstruction factor  $\omega$  was set to 0.4, the mini-batch size was set to 160 and the sizes of the prior dataset for KOS, NIPS and Enron were set to 500, 200, and 1,000 respectively.

1) *Effect of Privacy Levels:* Fig. 8 shows the average perplexity (normalized by the number of mini-batches) of our proposed OLP-LDA versus the number of mini-batches, under different privacy levels. Particularly, the non-private solution O-LDA is also compared for reference. As we can see in all three subfigures, the perplexity curves for different privacy levels decrease monotonically with the increase of batch number, which demonstrates that OLP-LDA can iteratively

refine the model with the continuous training batches. And also we can see, the different privacy level leads to different model performance in terms of perplexity, which also illustrates the tradeoff between the model performance and the privacy level.

2) *Effect of the Prior Dataset Size:* Fig. 9 depicts the performance of OLP-LDA in comparison with its non-private version O-LDA, under different sizes of the prior dataset. The privacy level in OLP-LDA was set to 3.0 and the black horizontal dashed lines marks the perplexity obtained by the non-private batched CGS training algorithm. As expected, a larger scale of the prior dataset leads to a better model performance, that's because a larger prior dataset could provide more accurate prior knowledge for the word counts and reflect more representative information of the whole corpus.

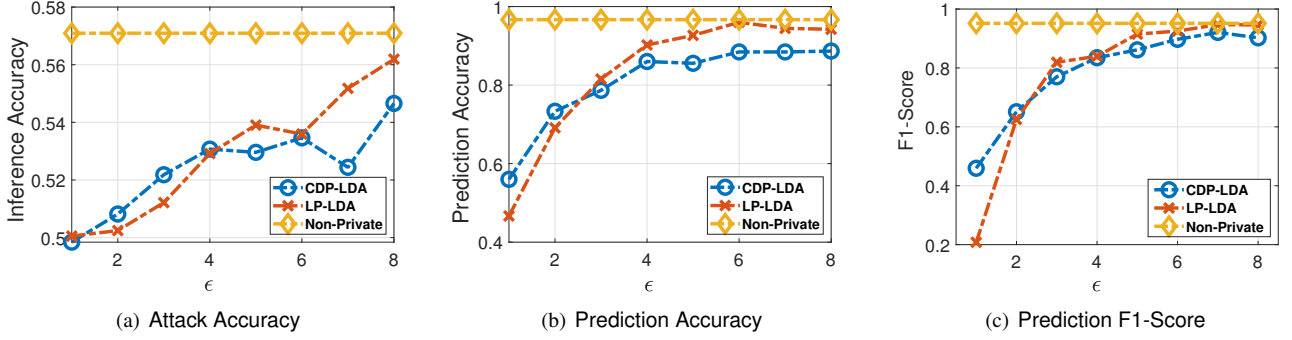


Fig. 7. Defense Against Membership Inference Attack

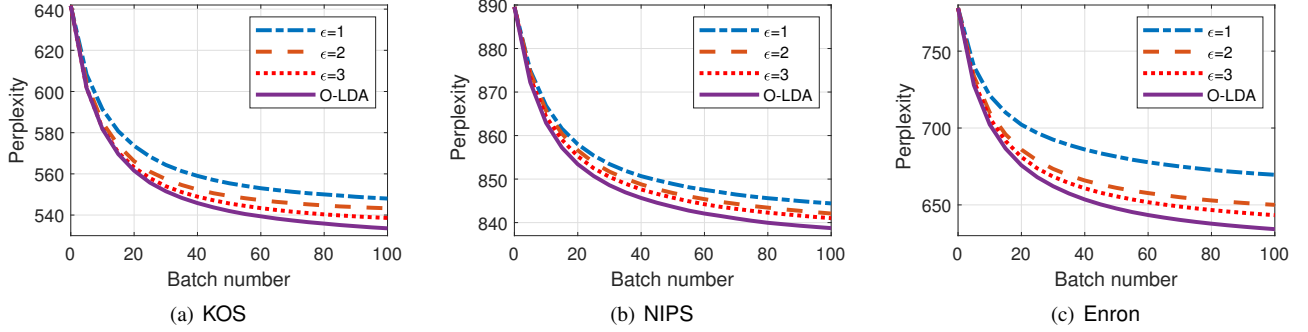


Fig. 8. Perplexity of OLP-LDA vs. Batch Number

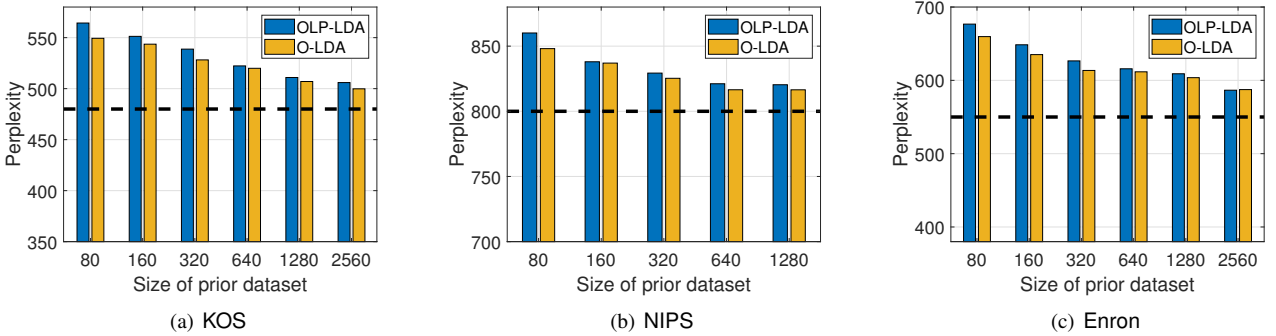


Fig. 9. Perplexity of OLP-LDA vs. Prior Data Size

3) *Effect of the Mini-batch Size*: Fig. 10 shows the impact of the mini-batch size on the model performance of OLP-LDA. For comparison, we also present the model performance of both OLP-LDA without the prior data and the non-private algorithm O-LDA. As shown, the perplexity of OLP-LDA without prior information is much larger than those of both O-LDA and OLP-LDA with prior dataset. Besides, the perplexity obtained by OLP-LDA with prior data can be very close to that in the non-private O-LDA. These observations show that the high effectiveness of our proposed OLP-LDA and validate that prior data can better enhance the performance of OLP-LDA. Furthermore, the perplexity of both OLP-LDA without prior data and O-LDA fall with the increase of mini-batch size respectively. However, for OLP-LDA, the perplexity does not show a clear decreasing trend with the growth of the mini-

batch size, and even slightly increases, which means OLP-LDA is relatively not sensitive to the mini-batch size. This is because the prior dataset has sufficiently optimized the model of OLP-LDA and achieved the approximate optimal model as O-LDA.

## VIII. CONCLUSION

This paper investigates the privacy protection of Latent Dirichlet Allocation model training in the framework of differential privacy. We present the first analysis on the intrinsic privacy-preserving effect possessed by Collapsed Gibbs Sampling based LDA training algorithm and further propose a centralized privacy-preserving algorithm (HDP-LDA) that can prevent data inference from the intermediate statistics in the CGS training. Concerning the privacy risks at the



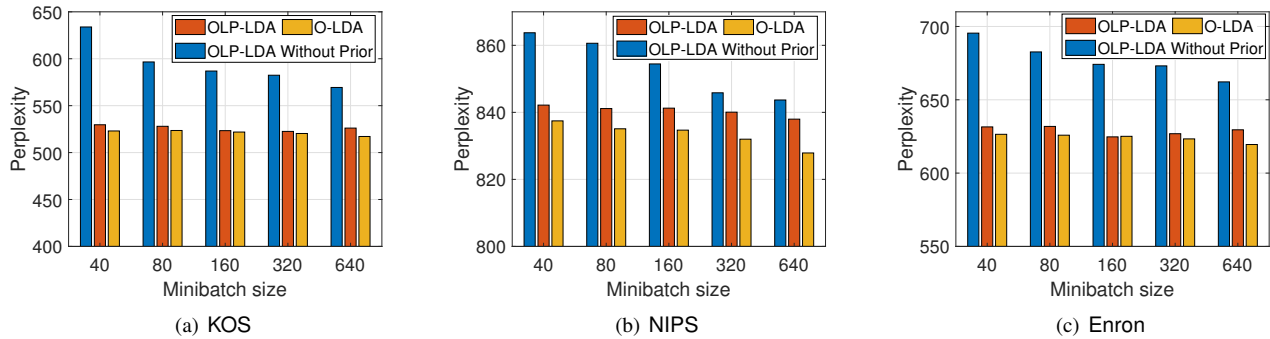


Fig. 10. Perplexity of OLP-LDA vs. Mini-batch Size.

central server, we also design an LDP solution of LP-LDA by sanitizing the individual documents at the local side and performing the model training on the reconstruction dataset at the server side. For LDA training in a stream setting, we further provide an online algorithm OLP-LDA that can efficiently refine the training model on continuously perturbed data batches. Both extensive analysis and experimental results on real-world datasets validate and demonstrate that our proposed algorithms can achieve high model utility under differential privacy.

## REFERENCES

- [1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [2] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T.-Y. Liu, and W.-Y. Ma, "Lightlda: Big topic models on modest computer clusters," in *Proc. WWW*, 2015, pp. 1351–1361.
- [3] Y. Wang, X. Zhao, Z. Sun, H. Yan, L. Wang, Z. Jin, L. Wang, Y. Gao, J. Zeng, Q. Yang *et al.*, "Towards topic modeling for big data," *arXiv preprint arXiv:1405.4402*, 2014.
- [4] L. Yut, C. Zhang, Y. Shao, and B. Cui, "Lda\*: a robust and large-scale topic modeling system," *Proc. VLDB Endowment*, vol. 10, no. 11, pp. 1406–1417, 2017.
- [5] A. Smola and S. Narayanamurthy, "An architecture for parallel topic models," *Proc. VLDB Endowment*, vol. 3, no. 1-2, pp. 703–710, 2010.
- [6] M. Hoogendoorn, P. Szolovits, L. M. Moons, and M. E. Numans, "Utilizing uncoded consultation notes from electronic medical records for predictive modeling of colorectal cancer," *Proc. AIM*, vol. 69, pp. 53–61, 2016.
- [7] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen, "An lda-based community structure discovery approach for large-scale social networks," in *Proc. IEEE ISI*, 2007, pp. 200–207.
- [8] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE S&P*, 2017, pp. 3–18.
- [9] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. USENIX Security*, 2014, pp. 17–32.
- [10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. TCC*. Springer, 2006, pp. 265–284.
- [11] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and S. Y. Philip, "LoPub: High-dimensional crowdsourced data publication with local differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2151–2166, 2018.
- [12] Y. Li, X. Ren, S. Yang, and X. Yang, "Impact of prior knowledge and data correlation on privacy leakage: A unified analysis," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2342–2357, 2019.
- [13] T. Zhu, P. Xiong, G. Li, and W. Zhou, "Correlated differential privacy: Hiding information in non-iid data set," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 229–242, 2014.
- [14] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, no. Mar, pp. 1069–1109, 2011.
- [15] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM CCS*, 2016, pp. 308–318.
- [16] M. Park, J. Foulds, K. Chaudhuri, and M. Welling, "Variational bayes in private settings (vips)," *arXiv preprint arXiv:1611.00340*, 2016.
- [17] T. Zhu, G. Li, W. Zhou, P. Xiong, and C. Yuan, "Privacy-preserving topic model for tagging recommender systems," *Knowl. Inf. Syst.*, vol. 46, no. 1, pp. 33–58, 2016.
- [18] Y. Wang, Y. Tong, and D. Shi, "Federated latent dirichlet allocation: A local differential privacy based framework," in *AAAI*, 2020, pp. 6283–6290.
- [19] T. Griffiths, "Gibbs sampling in the generative model of latent dirichlet allocation," 2002.
- [20] F. Zhao, X. Ren, S. Yang, and X. Yang, "On privacy protection of latent dirichlet allocation model training," in *Proc. IJCAI*, 2019, pp. 4860–4866.
- [21] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proc. NIPS*, 2009, pp. 289–296.
- [22] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near-optimal differentially private principal components," in *Proc. NIPS*, 2012, pp. 989–997.
- [23] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A practical differentially private random decision tree classifier," in *Proc. IEEE ICDM Workshops*, 2009, pp. 114–121.
- [24] C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin, and K. Ren, "Ganobfuscator: Mitigating information leakage under gan via differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2358–2371, 2019.
- [25] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "Dp-admm: Admm-based distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1002–1012, 2019.
- [26] T. Zhang and Q. Zhu, "Dynamic differential privacy for admm-based distributed classification learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 172–187, 2016.
- [27] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proc. ACM STOC*, 2007, pp. 75–84.
- [28] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proc. ACM STOC*, vol. 9, 2009, pp. 371–380.

- [29] J. Czerniak and H. Zarzycki, "Application of rough sets in the presumptive diagnosis of urinary system diseases," in *Proc. ACS*. Springer, 2003, pp. 41–51.
- [30] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proc. IEEE FOCS*, 2014, pp. 464–473.
- [31] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. ACM CCS*, 2015, pp. 1310–1321.
- [32] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM CCS*, 2014, pp. 1054–1067.
- [33] M. Sun and W. P. Tay, "On the relationship between inference and data privacy in decentralized iot networks," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 852–866, 2019.
- [34] C. DeCarolis, M. Ram, S. A. Esmaili, Y.-X. Wang, and F. Huang, "An end-to-end differentially private latent dirichlet allocation using a spectral algorithm," *arXiv preprint arXiv:1805.10341*, 2018.
- [35] G. Bernstein and D. R. Sheldon, "Differentially private bayesian inference for exponential families," in *Advances in Neural Information Processing Systems*, 2018, pp. 2919–2929.
- [36] Y. Wang, S. Fienberg, and A. Smola, "Privacy for free: Posterior sampling and stochastic gradient monte carlo," in *Proc. ICML*, 2015, pp. 2493–2502.
- [37] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: Private data release via bayesian networks," *ACM Trans. Dat. Syst.*, vol. 42, no. 4, p. 25, 2017.
- [38] C. Dimitrakakis, B. Nelson, A. Mitrokovska, and B. I. Rubinstein, "Robust and private bayesian inference," in *Proc. ALT*. Springer, 2014, pp. 291–305.
- [39] Z. Zhang, B. I. Rubinstein, and C. Dimitrakakis, "On the differential privacy of bayesian inference," in *Proc. AAAI*, 2016.
- [40] J. Foulds, J. Geumlek, M. Welling, and K. Chaudhuri, "On the theory and practice of privacy-preserving bayesian data analysis," *arXiv preprint arXiv:1603.07294*, 2016.
- [41] K. Minami, H. Arai, I. Sato, and H. Nakagawa, "Differential privacy without sensitivity," in *Proc. NIPS*, 2016, pp. 956–964.
- [42] G. Heinrich, "Parameter estimation for text analysis," Technical report, Tech. Rep., 2005.
- [43] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014.
- [44] D. Newman, A. Asuncion, P. Smyth, and M. Welling, "Distributed algorithms for topic models," *J. Mach. Learn. Res.*, vol. 10, no. 8, 2009.
- [45] H. Amouliyan, M. Clausel, E. Gaussier, and M.-R. Amini, "Streaming-lda: A copula-based approach to modeling topic dependencies in document streams," in *Proc. ACM SIGKDD*, 2016, pp. 695–704.
- [46] E. L. Lehmann and G. Casella, *Theory of point estimation*. Springer Science & Business Media, 2006.
- [47] C. M. Bishop, *Pattern recognition and machine learning*. Springer Science & Business Media, 2006.
- [48] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model," *Transactions on Data Privacy*, vol. 11, no. 1, pp. 61–79, 2018.

#### APPENDIX A

##### PROOF OF PROPOSITION 1

*Proof.* According to the observation in section IV-C1, we need to compute an upper bound of the sensitivity  $\Delta u = \max_k \{\Delta \log p_k\}$  of the utility function  $u = \log p$ .

As shown in Equation (1), the sampling probability on  $w_i = t \in D$  for any given topic  $k$  can be computed by

$$p_k \propto r_k = \frac{n_k^t + \beta}{\sum_{t=1}^V (n_k^t + \beta)} \cdot \frac{n_m^k + \alpha}{\sum_{k=1}^K (n_m^k + \alpha)} \quad (25)$$

And correspondingly, the sampling probability on  $w_i' = t' \in D'$  can be computed by

$$p_k' \propto r_k' = \frac{n_k^{t'} + \beta}{\sum_{t=1}^V (n_k^t + \beta)} \cdot \frac{n_m^k + \alpha}{\sum_{k=1}^K (n_m^k + \alpha)} \quad (26)$$

Notably, here we assume  $n_m^k | D = n_m^k | D'$  since if the *word replacement* causes any change on  $n_m^k$ , which also means some sampling process has been changed, then that change would be observed by the adversary.

Then taking into account the normalization constants,  $\Delta \log p_k$  can be bounded by

$$\begin{aligned} \Delta \log p_k &= \left| \log \frac{p_k'}{p_k} \right| = \left| \log \left( \frac{r_k}{r_k'} \cdot \frac{\sum_k r_k'}{\sum_k r_k} \right) \right| \\ &= \left| \log \left( \frac{r_k}{r_k'} \right) + \log \left( \frac{\sum_k r_k'}{\sum_k r_k} \right) \right| \\ &\leq \left| \log \left( \frac{r_k}{r_k'} \right) \right| + \left| \log \left( \frac{\sum_k r_k'}{\sum_k r_k} \right) \right| \\ &\leq \Delta \log r_k + \max_k \left\{ \left| \log \frac{r_k'}{r_k} \right| \right\} \\ &\leq 2 \max_k \{ \Delta \log r_k \} = 2 \max_k \left\{ \left| \log \frac{n_k^{t'} + \beta}{n_k^t + \beta} \right| \right\} \end{aligned} \quad (27)$$

Until now, proposition 1 has been proved.  $\square$

#### APPENDIX B

##### PROOF OF THEOREM 1

*Proof.* Let  $P(\cdot)$  and  $P'(\cdot)$  denote the probability on  $D$  and  $D'$  respectively, and for convenience of derivation, the conditional probability  $Pr[\cdot]$  are simply represented by  $Pr[\cdot]$ , then

$$\begin{aligned} P'(\mathcal{S}_i(D') = \mathbf{o}_i | \mathcal{S}_{i-1}(D') = \mathbf{o}_{i-1}) &= P'(\mathcal{S}_i(w_r = t) = o_r) \cdot \\ &\prod_{h=1}^{n_t-1} P'(\mathcal{S}_i(w_h = t) = o_h^t) \cdot \prod_{k=1}^{n_{t'}} P'(\mathcal{S}_i(w_k = t') = o_k^{t'}) \cdot \\ &P'(\mathcal{S}_i(\mathbf{w}^-) = \mathbf{o}^-) \\ &\leq e^{\epsilon_i} P(\mathcal{S}_i(w_r = t') = o_r) \cdot e^{\sum_{h=1}^{n_t-1} \epsilon_i} \prod_{h=1}^{n_t-1} P(\mathcal{S}_i(w_h = t) = o_h^t) \\ &\cdot e^{\sum_{k=1}^{n_{t'}} \epsilon_i} \prod_{k=1}^{n_{t'}} P(\mathcal{S}_i(w_k = t') = o_k^{t'}) \cdot P(\mathcal{S}_i(\mathbf{w}^-) = \mathbf{o}^-) \\ &= e^{\epsilon_i + (n_t-1)\epsilon_i + n_{t'}\epsilon_i} P(\mathcal{S}_i(D) = \mathbf{o}_i | \mathcal{S}_{i-1}(D) = \mathbf{o}_{i-1}) \end{aligned} \quad (28)$$

where  $n_t$  and  $n_{t'}$  denote the counts of word  $t$  and  $t'$  in  $D$  respectively and  $\mathbf{w}^-$  denotes all the *Unrelated words* in  $D$ .

Then, according to Proposition 1 and Proposition 2, the privacy loss in the  $i$ -th iteration can be bounded by

$$\epsilon_i \leq 2 \max_k \left\{ \left| \log \frac{n_k^{t'} + \beta}{n_k^t + \beta} \right| + (n_{t'} + n_t - 1) \cdot \log \left( 1 + \frac{1}{\beta} \right) \right\}$$

Due to the arbitrariness of  $t'$ ,  $\epsilon_i$  can be further bounded by

$$\begin{aligned} \epsilon_i &\leq 2 \left\{ \log \left( \frac{\max_{k,t'} \{n_k^{t'}\}}{\beta} + 1 \right) \right. \\ &\quad \left. + (\max_{t'} \{n_{t'}\} + n_t - 1) \cdot \log \left( 1 + \frac{1}{\beta} \right) \right\} \end{aligned}$$

Until now, Theorem 1 has been proved.  $\square$

APPENDIX C  
PROOF OF THEOREM 2

*Proof.* Given the replaced word  $w_r \in D$ , the privacy loss of  $w_r$  in the  $i$ -th iteration  $\epsilon_i^r$  can be bounded by Theorem 1. Now consider the privacy loss in multiple iterations. Let  $\mathcal{S}$  denotes the CGS algorithm throughout the overall training process,  $\mathcal{S}_i$  the algorithm in the  $i$ -th iteration.

$$\begin{aligned}
P'(\mathcal{S}(D') = \mathbf{o}) &= P'(\mathcal{S}_1(D') = \mathbf{o}_1, \dots, \mathcal{S}_i(D') = \mathbf{o}_i, \dots, \mathcal{S}_n(D') = \mathbf{o}_n) \\
&= \prod_{i=1}^n P'(\mathcal{S}_i(D') = \mathbf{o}_i | \mathcal{S}_1(D') = \mathbf{o}_1, \dots, \mathcal{S}_{i-1}(D') = \mathbf{o}_{i-1}) \\
&\leq \prod_{i=1}^n e^{\epsilon_i} \cdot P(\mathcal{S}_i(D) = \mathbf{o}_i | \mathcal{S}_1(D) = \mathbf{o}_1, \dots, \mathcal{S}_{i-1}(D) = \mathbf{o}_{i-1}) \\
&= e^{\sum_{i=1}^n \epsilon_i} P(\mathcal{S}_1(D) = \mathbf{o}_1, \dots, \mathcal{S}_i(D) = \mathbf{o}_i, \dots) \\
&= e^{\sum_{i=1}^n \epsilon_i} P(\mathcal{S}(D) = \mathbf{o})
\end{aligned}$$

Until now, we have found the privacy loss  $\epsilon_{w_r}$  of  $w_r$  over  $n$  iterations.

However, due to the arbitrariness of  $w_r$ , the upper bound of the privacy loss incurred by CGS algorithm should be found by traversing all the words in  $D$ . Therefore, the total privacy loss  $\epsilon$  can be bounded by

$$\epsilon \leq \max_{w \in D} \left\{ \sum_{j=1}^n \epsilon_j^w \right\}$$

So far, Theorem 2 has been proved.  $\square$

APPENDIX D  
PROOF OF THEOREM 3

*Proof.* In each iteration of Algorithm 1, the privacy loss should be divided into 2 parts: 1). privacy loss  $\epsilon_L$  when protecting the word counts information  $\mathbf{s} = \{n_k^t\}$ , let  $\mathcal{A}_1(D)$  be the protection process and  $\mathbf{s}'$  the sanitized word counts; 2). privacy loss  $\epsilon_I$  in the complete topic sampling process in this iteration, let  $\mathcal{A}_2(\mathbf{s}', D)$  be the sampling process and  $\mathbf{o}$  the outputs.

First consider the first part. Suppose  $D'$  is constructed by replacing  $w_r = t \in D$  by  $t'$ , then this replacement will cause  $n_k^t - 1$  and  $n_k^{t'} + 1$  on  $D'$  for some  $k$ , and the added noise will need to cover the two caused changes:

$$\begin{aligned}
Pr[\mathcal{A}_1(D) = \mathbf{s}'] &= \sum_{\mathbf{s}'} Pr[\dots, (n_k^t) + \eta_k^t = s_k^t, \dots, (n_k^{t'}) + \eta_k^{t'} = s_k^{t'}, \dots | D] \\
&= \sum_{\mathbf{s}'} \prod_{h,j} Pr[(n_h^j) + \eta_h^j = s_h^j | D] \\
&\leq \sum_{\mathbf{s}'} \prod_{(h,j) \neq (k,t), (k,t')} Pr[(n_h^j) + \eta_h^j = s_h^j | D'] \\
&\cdot e^{\frac{\epsilon_L}{2}} Pr[(n_k^t - 1) + \eta_k^t = s_k^t | D'] \cdot e^{\frac{\epsilon_L}{2}} Pr[(n_k^{t'} + 1) + \eta_k^{t'} = s_k^{t'} | D'] \\
&\leq \sum_{\mathbf{s}'} e^{\epsilon_L} Pr[\dots, (n_k^t - 1) + \eta_k^t = s_k^t, \dots, (n_k^{t'} + 1) + \eta_k^{t'} = s_k^{t'}, \dots | D'] \\
&= e^{\epsilon_L} Pr[\mathcal{A}_1(D') = \mathbf{s}']
\end{aligned} \tag{29}$$

Then consider the second part. According to Theorem 1, we focus on the privacy loss on the *Related words* and the *Replace*

*word*. The privacy loss on the *Related words* can be computed by

$$\begin{aligned}
\epsilon &\leq \max_k \{\Delta \log p_k\} = \max_k \left\{ \log \frac{p_k(w = t(t') | D)}{p_k(w = t(t') | D')} \right\} \\
&= \max_k \left\{ \log \frac{\frac{(n_k^{t(t')})^{temp} + \beta}{\sum_{t=1}^V ((n_k^t)' + \beta)} \cdot \frac{n_m^k + \alpha}{\sum_{k=1}^K (n_m^k + \alpha)} | D}{\frac{(n_k^{t(t')})^{temp} + \beta}{\sum_{t=1}^V ((n_k^t)' + \beta)} \cdot \frac{n_m^k + \alpha}{\sum_{k=1}^K (n_m^k + \alpha)} | D'} \right\} \\
&= \max_k \left\{ \log \frac{(n_k^{t(t')})^{temp} + \beta | D}{(n_k^{t(t')})^{temp} + \beta | D'} \right\} = 0
\end{aligned} \tag{30}$$

where  $t(t')$  means  $t$  or  $t'$  and the last step is because  $n_k^{t(t')}$  has been privatized by adding Laplace noise. Therefore, according to Theorem 1, the privacy loss  $\epsilon_I$  should be

$$\epsilon_I \leq 2 \cdot \log \left( \frac{\max_{k,t'} \{n_k^{t'}\}}{\beta} + 1 \right) \leq 2 \cdot \log \left( \frac{C}{\beta} + 1 \right) \tag{31}$$

where  $C$  denotes the clipping bound on  $n_k^t$ . Now, combine Equation (29) and Equation (31),

$$\begin{aligned}
Pr[\mathcal{A}_2(\mathcal{A}_1(D), D) = \mathbf{o}] &= \sum_{\mathbf{s}'} Pr[\mathcal{A}_1(D) = \mathbf{s}'] \cdot Pr[\mathcal{A}_2(\mathbf{s}', D) = \mathbf{o}] \\
&\leq \sum_{\mathbf{s}'} e^{\epsilon_L} Pr[\mathcal{A}_1(D') = \mathbf{s}'] \cdot e^{\epsilon_I} Pr[\mathcal{A}_2(\mathbf{s}', D') = \mathbf{o}] \\
&\leq e^{\epsilon_L + \epsilon_I} Pr[\mathcal{A}_2(\mathcal{A}_1(D'), D') = \mathbf{o}]
\end{aligned}$$

So far, Theorem 3 has been proved.  $\square$