SeqPATE: Differentially Private Text Generation via Knowledge Distillation

Zhiliang Tian[1]*, **Yingxiu Zhao**[2], **Ziyue Huang**[2], **Yuxiang Wang** [3], **Nevin L. Zhang** [2], **He He** [4]
[1] National University of Defense Technology,
[2] The Hong Kong University of Science and Technology,
[3] UC Santa Barbara,
[4] New York University
tianzhilianghit@gmail.com, yzhaocx@connect.ust.hk, zyhuang94@gmail.com,
yuxiangw@cs.ucsb.edu, lzhang@cse.ust.hk, hhe@nyu.edu

Abstract

Protecting the privacy of user data is crucial for text generation models, which can leak sensitive information during generation. Differentially private (DP) learning methods provide guarantees against identifying the existence of a training sample from model outputs. PATE is a recent DP learning algorithm that achieves high utility with strong privacy protection on training samples. However, text generation models output tokens sequentially in a large output space; the classic PATE algorithm is not customized for this setting. Furthermore, PATE works well to protect sample-level privacy, but is not designed to protect phrases in samples. In this paper, we propose SeqPATE, an extension of PATE to text generation that protects the privacy of individual training samples and sensitive phrases in training data. To adapt PATE to text generation, we generate pseudo-contexts and reduce the sequence generation problem to a next-word prediction problem. To handle the large output space, we propose a candidate filtering strategy to dynamically reduce the output space, and refine the teacher aggregation of PATE to avoid low agreement due to voting for a large number of candidates. To further reduce privacy losses, we use knowledge distillation to reduce the number of teacher queries. The experiments verify the effectiveness of SeqPATE in protecting both training samples and sensitive phrases.

1 Introduction

Recent work has shown that sensitive user information in training corpora, such as addresses and names, can be extracted from text generation models [6]. Providing privacy guarantees to the training corpora of text generation models has become a critical problem. Differential privacy (DP) provides provable guarantees against detecting individuals in datasets. Deep learning models with DP guarantees ensure that the existence of a specific training sample cannot be detected.

NoisySGD [42, 3, 1] is a popular DP algorithm for deep learning that adds noise to the gradients. PATE [31] is another type of DP learning algorithm that transfers knowledge from teachers trained on private data to a student model, where noises are added to teacher predictions to satisfy DP. PATE is model-agnostic, and its privacy cost derives from the knowledge distillation process instead of the model gradients in NoisySGD [42, 24]. Therefore, the noises required by PATE do not scale with model size. Given this benefit, PATE has great potential for text generation, since large language

*This paper was partially done when Zhiliang Tian was a Ph.D. student at HKUST and a visiting scholar at NYU.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

models (e.g., GPT-2 [35]) have become the backbone of most text generation models. However, NoisySGD and PATE are used to protect sample-level privacy [52, 24] and not customized to protect sensitive phrases in the data with a low privacy cost [22, 39, 51]. Additionally, PATE, originally designed for classification tasks, is not customized for sequential generation on a large output space (i.e., the natural language vocabulary), which is very common in text generation.

In this paper, we propose SeqPATE, a DP learning algorithm for text generation to protect the privacy of training corpora. By satisfying DP, SeqPATE has the guarantee of preventing the existence of training samples and sensitive phrases in the training corpora from being detected. Similarly to PATE, SeqPATE employs a teacher-student framework: (i) a student model learns to generate text from non-sensitive samples; and (ii) a number of teacher models, trained on sensitive text, supervise the student through noised outputs of aggregated teachers. The calibrated noise added to the output ensures that SeqPATE satisfies the DP requirements. This framework still faces some challenges in text generation. First, it suffers from the high costs of GPU memory and time. To obtain sentence-level supervision for text generation, the model needs to roll out all teachers to produce a sentence (i.e. all teachers vote to generate a word, which is then used as the input for the next word prediction). It results in a high inference cost with a large number of teachers (e.g. $2k$ teachers which are common in PATE). Second, the large output space (i.e., the vocabulary) in text generation leads to (i) low agreement rates among teachers and (ii) large noises required by DP, both of which significantly hurt the task performance.

To address the challenges, we generate pseudo-data using a pre-trained language model so that teachers only need to provide token-level supervision given the pseudo inputs. To handle the large output space and reduce the noise, we propose to dynamically filter the candidate words and select only words with high probabilities. Also, we aggregate teachers' outputs by interpolating their output distributions instead of voting with argmax predictions. DP learning methods provide privacy protection by adding noise, which also reduces the utility of the model. To reduce utility loss, we avoid unnecessary knowledge distillation by selectively applying knowledge distillation to generation steps where the student struggles. Most DP learning methods, including SeqPATE, prevent samples from being extracted. SeqPATE has further advantages in protecting users' secret phrases that occur multiple times in the corpora. We evaluate SeqPATE on a sentence completion task, which demonstrates its advantage in protecting samples and phrases compared to the baselines.

Our contribution is twofold: (i) We propose SeqPATE that provides privacy at both the sample level and the phrase level with theoretical analyses. (ii) We propose several strategies for SeqPATE to handle autoregressive text generation models with a large vocabulary.

2 Problem Setup

Our goal is to achieve the privacy protection quantified by DP in text generation to prevent attackers from inferring whether a sample or an n-gram appears in the training set. Our setting contains two types of textual datasets: (1) a private set $\mathcal{D}^{\text{pri}}$ from a corpus with sensitive information, (2) a public set $\mathcal{D}^{\text{pub}}$ that contains no sensitive information or comes from data contributors (e.g., volunteers) who have no objection to publishing their data. We aim to protect the privacy on the private set and can ignore the privacy protection on the public set.

Our application, sentence completion, aims to complete the whole sentence given the prefix. We train a language model to accomplish the task. The public set $\mathcal{D}^{\text{pub}}$ consists of prefixes, which can hardly contain sensitive information. The private set $\mathcal{D}^{\text{pri}}$ consists of whole sentences. Such a setting fits some real-world text generation applications: in dialog systems, the training samples from online services consist of questions and responses. The questions from customer service staff or service robots can be public, and the response from users carrying individual information should be private.

3 Background on DP and PATE

Definition 3.1. [Differential privacy (DP) [13, 14]] For any two neighboring datasets $\mathcal{D}, \mathcal{D}'$ (differ in only one individual), a randomized algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-differentially private if,

$$\mathsf{Pr}[\mathcal{M}(\mathcal{D}) \in S] \leq e^{\varepsilon} \cdot \mathsf{Pr}[\mathcal{M}(\mathcal{D}') \in S] + \delta, \quad \forall S \subseteq \mathcal{Y}, \quad \text{where } \varepsilon > 0,\ \delta \geq 0. \tag{1}$$

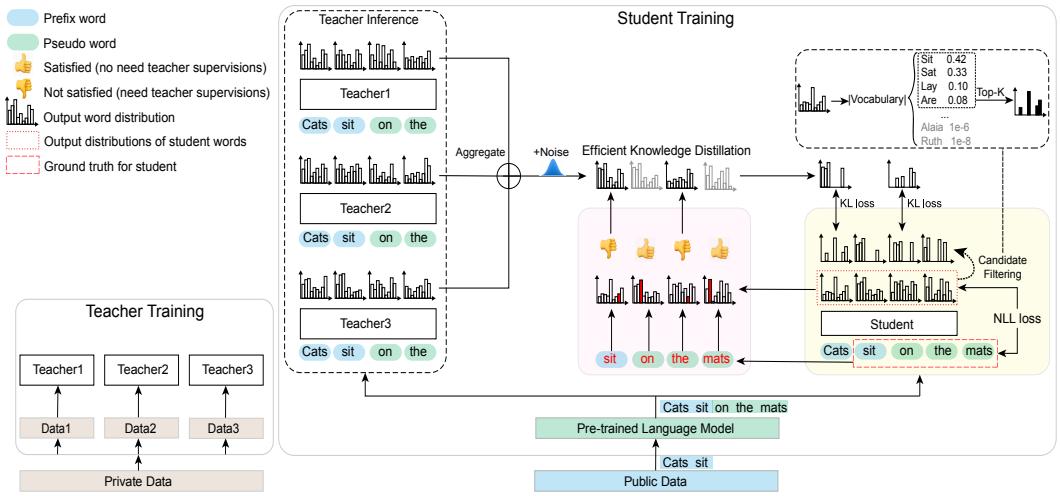

Figure 1: Overview of SeqPATE. SeqPATE trains teachers on private data. Student models are trained on pseudo-sentences generated by a pre-trained language model given the public prefixes. The student is supervised by aggregated teacher output distributions. SeqPATE benefits from candidate filtering (white block in the top right corner) and efficient knowledge distillation that determines whether teacher supervision is needed (pink block).

By definition, DP is a quantifiable definition of privacy that provides guarantees on identifications of individual data (preventing an adversary from inferring whether the input is $\mathcal{D}$ or $\mathcal{D}'$). ML models with DP ensure that each training sample has a degree of *plausible deniability*, i.e., the trained model is *just as likely as* to be trained on an alternative dataset *without* that sample. In SeqPATE, $\mathcal{M}$ is the entire training and inference process, S is the vocabulary, and $Pr[\cdot]$ denotes the output distribution of generating a word. Attackers cannot tell whether a sample is in the training set or not, since the output distributions of the datasets *with or without that sample* are very similar (bounded by Eq. 1).

PATE [31], designed for classification tasks, takes advantage of an unlabeled public dataset $\mathcal{D}^{\text{pub}}$ and also trains on a labeled private set $\mathcal{D}^{\text{pri}}$ in a semi-supervised scenario. PATE achieves DP through a teacher-student framework with M teacher models and a student model, where the student learns from the private set via knowledge distillation through teachers. PATE has three parts: (i) **The teacher models** are trained on the private set $\mathcal{D}^{\text{pri}}$, which is shuffled and divided into M disjoint subsets. Each teacher is trained on one subset. (ii) **Teacher aggregation** merges the teachers' outputs. Each of the trained teachers then provides supervision to the student's unlabeled public set $\mathcal{D}^{\text{pub}}$. We use noised majority votes from teachers as labels to supervise the student. (iii) **A student model** is trained on the public set $\mathcal{D}^{\text{pub}}$ with the supervision of the aggregated teachers.

4 Approach

Fig. 1 shows an overview of SeqPATE. Given the public prefix (e.g., "Cats sit"), we first obtain the pseudo-inputs by completing the sentence (e.g., "Cats sit on the mats") using a pre-trained language model (Sec. 4.1). At each word, we then aggregate the teachers' prediction of the next word as supervision for training the student model (Sec. 4.2). To reduce the noise required by DP for a large output space of the size of the vocabulary, we reduce the output space by dynamically filtering unimportant words. To reduce the number of teacher queries that incur privacy losses, we propose an efficient knowledge distillation strategy that only queries teacher labels on uncertain examples (Sec. 4.3). We show the training algorithm in App. B and a running example in App. K.

4.1 Pseudo Input Generation

Conventional text generation models generate words sequentially from left to right. Thus, naively applying PATE to text generation requires rolling out all teachers word by word, i.e., iteratively sampling the next word from the aggregated teacher prediction. This is costly in both computation (running inference for hundreds of teacher models) and privacy costs (querying teachers at every step).

To tackle this challenge, we use a pre-trained language model to complete the public prefixes into pseudo sentences; thus, we only need to query teachers on the next word given a (pseudo) context.

4.2 Teacher Aggregation

PATE aggregates teacher predictions by majority vote. While it works for classification problems with a relatively small number of classes, the output space of text generation models contains all words in the vocabulary. As a result, the number of votes for each candidate word may be very low without a clear winner. For example, multiple candidates may tie for the top-1 prediction.

Inspired by Chen et al. [9, 17], we aggregate teacher results by averaging their output distributions. We first train M teacher models on disjoint subsets of the private data. To produce the aggregated next word distribution given a context c, we average the teachers' output distributions, add calibrated noises, and then renormalize the results into a proper distribution. Following Papernot et al. [32], we apply the Gaussian mechanism. Formally, let $p_\phi^m(\cdot \mid c)$ be the prediction of the m-th teacher. The aggregated distribution is $p_{\text{agg}}(\cdot \mid c) \propto \frac{1}{M}\sum_{m=1}^{M}(p_\phi^m(\cdot \mid c) + \mathcal{N}(0, \sigma^2))$, [2] where the Gaussian noise is added to the aggregated output distribution. The way of SeqPATE satisfies DP guarantee (Eq. 1) is to add that calibrated noise to the teachers' output as mentioned above (detailed analyses in Sec. 5).

4.3 Training of the Student Model

The student model is trained on public pseudo-data and also supervised by the aggregated teachers.

Training objectives. The student model is a language model that predicts the next word given prior contexts. Given contexts from the (public) pseudo-data autocompleted by a pre-trained language model (GPT-2), the student is supervised by both the aggregated teacher predictions and the next word in the pseudo-data (i.e. pseudo label). The pseudo-data acts as a prior for the student given that the number of teacher queries is limited due to privacy concerns. The student's loss function has two parts:

- $\mathcal{L}_{\text{teacher}}$ denotes the loss with respect to teacher supervision. Note that the aggregated teacher output is a distribution over words. Therefore, we minimize the forward KL divergence between the aggregated teacher distribution p_{agg} and the student output distribution p_θ:

$$\mathcal{L}_{\text{teacher}}(c, p_{\text{agg}}) = \text{KL}\left(p_{\text{agg}}(\cdot \mid c) \,\|\, p_\theta(\cdot \mid c)\right). \tag{2}$$

- $\mathcal{L}_{\text{pseudo}}$ denotes the loss with respect to the pseudo-labels w from $\tilde{\mathcal{D}}^{\text{pub}}$ (i.e. next words generated by a generic language model). Similar to standard language modeling, we use the negative log-likelihood:

$$\mathcal{L}_{\text{pseudo}}(c, w) = -\log p_\theta(w \mid c). \tag{3}$$

Eq. 4 shows the complete loss. (λ balances the two terms and we discuss the noise scale σ in Sec. 5.)

$$\mathcal{L}(p_{\text{agg}}, \tilde{\mathcal{D}}^{\text{pub}}) = \sum_{(c,w)\in\tilde{\mathcal{D}}^{\text{pub}}} \mathcal{L}_{\text{pseudo}}(c, w) + \lambda\mathcal{L}_{\text{teacher}}(c, p_{\text{agg}}), \tag{4}$$

Reducing the output space via candidate filtering. The high-dimensionality of the output of text generation models results in large noise (which is added to each coordinate). To reduce the output dimension (hence the amount of noise), we filter words on the tail of the distribution of the student model (i.e. set their probability to zero), and renormalize the teacher's aggregated distribution and the student output distribution over the rest words.

Note that the candidate filtering is based on the student's outputs on public or already released inputs, thus it does not affect the privacy guarantee. This choice improves the privacy-utility tradeoff by adaptively allocating the privacy budget to release the information most helpful to the task.

We experiment with two filtering strategies: top-k and top-p. In top-k filtering, we retain only the top-k most likely candidates and filter the rest according to the student model. In top-p filtering [18],

[2] Mathematically, the aggregated distribution with noises may be negative. If so, we renormalize the negative value to 0. Practically, we observed that being negative is an extremely rare event, since the M is usually very large (e.g., $2k$) and the first term dominates the above equation.

k is chosen dynamically such that the top-k words are the minimum set whose cumulative probability is at least p. The strategy seldom loses good candidates because the student usually does well on top-k predictions since the beginning of the training. [3]

Reducing the number of teacher queries via efficient knowledge distillation. While the aggregated teacher model satisfies DP, each query from the student incurs some privacy loss. Therefore, we obtain teacher supervision only on "hard" examples when training the student. Note that the student is trained on both the pseudo-data and local supervision from the teachers. We consider an example to be hard if the student cannot imitate the pseudo-label, in which case distilling knowledge from the teachers that are trained on large private data is helpful.

Concretely, we query teachers only when the rank of the pseudo-label is below a certain threshold among words ordered by descending probabilities under the student model. If we query the teachers, the student is trained via complete loss $\mathcal{L}(p_{\text{agg}}, \tilde{\mathcal{D}}^{\text{pub}})$ (Eq. 4); otherwise, the student is trained via the $\mathcal{L}_{\text{pseudo}}$ (Eq. 3). We note that the selection of tokens relies only on the student and is independent of the teachers; thus, the selection does not cause any additional privacy loss.

5 Privacy Analyses

5.1 Preliminary of Differential Privacy

Lemma 5.1 (Analytical Gaussian mechanism [2])**.** *For a numeric query $f : \mathcal{X}^n \to \mathbb{R}^d$ over a dataset $\mathcal{D}$, the randomized algorithm that outputs $f(\mathcal{D}) + Z$ where $Z \sim \mathcal{N}(0, \sigma^2 I_d)$ satisfies $(\varepsilon, \delta(\varepsilon))$-DP for all $\varepsilon \geq 0$ and $\delta(\varepsilon) = \Phi(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}) - e^{\varepsilon}\Phi(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta})$. where $\Delta := \Delta_2^{(f)} = \max_{\mathcal{D}\sim\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$ is the global L2 sensitivity of f and Φ is the CDF function of $\mathcal{N}(0, 1)$.*

We can use the same result for an adaptive composition of a sequence of Gaussian mechanisms.

Lemma 5.2 (Composition of Gaussian mechanisms [11])**.** *The adaptive composition of a sequence of Gaussian mechanisms with a noise level $\sigma_1, \sigma_2, \ldots$ and global L2 sensitivity $\Delta_1, \Delta_2, \ldots$ satisfies $(\varepsilon, \delta(\varepsilon))$-DP for all $\varepsilon \geq 0$ and $\delta(\varepsilon) \leq \delta_{\mathcal{M}}(\varepsilon)$ where $\mathcal{M}$ is a Gaussian mechanism with noise multiplier $\sigma/\Delta = \left(\sum_i (\Delta_i/\sigma_i)^2\right)^{-1/2}$.*

Specifically, the adaptive composition of a k identical Gaussian mechanism with a noise multiplier σ satisfies the same privacy guarantee of a single Gaussian mechanism with a noise multiplier $\sigma/\sqrt{k}$. By fixing k and ε, we can calibrate the noise by choosing an appropriate σ in Sec. 4.2.

5.2 Differential Privacy for Language Models at the Sample Level

Recall that we partition the private dataset into M disjoint subsets, and train each teacher model on one of the subsets. Let vector $x_i \in \mathbb{R}^{|\mathcal{V}|}$ denote the probability distribution predicted by the i-th teacher model given some context, where $|\mathcal{V}|$ is the vocabulary size. The aggregation function $f(\mathcal{D}) := \sum_{i=1}^{M} x_i$ is the sum of the probability distributions predicted by all teachers. Since the datasets are disjoint, changing one sample affects only one teacher model. For neighboring datasets $\mathcal{D}$, $\mathcal{D}'$, let j denote the index of each teacher model; the probability distributions x_j and x'_j (derived from $\mathcal{D}$ and $\mathcal{D}'$ respectively) are different. Then, the sensitivity Δ in Lemma 5.1 & 5.2 is (See detailed deductions in App. C),

$$\Delta := \Delta_2^{(f)} = \|f(\mathcal{D}) - f(\mathcal{D}')\|_2 \leq \|x_j - x'_j\|_2 \leq \sqrt{2}.$$

Adding the noises given by Lemma 5.2 to each coordinate (each candidate at each generation step of SeqPATE) preserves (ε, $\delta(\varepsilon)$)-DP for $f(\mathcal{D})$. Finally, when we extract top-k coordinates by top-k candidate filtering (Sec. 4.3), the privacy guarantee also holds due to the post-processing property [14]. Therefore, the fact about whether a sample is in SeqPATE's private sets is protected (satisfying (ε, $\delta(\varepsilon)$)-DP).

[3] In the first 10 training batches, the top-50 predictions of the student cover 94% "true" labels of pseudo samples.

5.3 Differential Privacy of Users' Secret Phrases

The above analyses show that we can protect the privacy of each sample (i.e., one *occurrence* of a sentence). However, in practice, we may want to protect all occurrences of some *secret phrases* specific to a user (e.g., names and addresses).[4] Consider a secret phrase s that occurs n_s times ($n_s \geq 1$) in the private set. According to group privacy [14], the protection on phrase s satisfies $(n\varepsilon, \frac{e^{n\varepsilon}-1}{e^{\varepsilon}-1}\delta)$-DP [22], where the privacy loss scales linearly with the number of occurrences of s (We discuss and analyze a better strategy to reduce the privacy loss of baselines in App. M).

Naively applying a DP algorithm requires larger noise to protect phrases that may occur multiple times. SeqPATE enjoys a stronger guarantee by assigning all data of a single user to one or a few teachers, such that any user-specific phrase occurs in the training data of only one or a few teachers. We denote $\tilde{n}_s$ as the number of teachers whose data contain the phrase s. Since adding or removing the phrase s affects only $\tilde{n}_s$ teachers ($\tilde{n}_s$ is usually 1 or 2) and thus results in a sensitivity of $\sqrt{2}\tilde{n}_s$ (See App. D for details). In this way, the strength of protection on secret phrases is roughly equal to that we have derived for sample-level DP. The exact $(\varepsilon, \delta(\varepsilon, \tilde{n}_s))$-DP for the phrase s can be obtained according to Lemma 5.1 & 5.2, where $\delta(\varepsilon, \tilde{n}_s) = \Phi(\frac{\tilde{n}_s}{\sqrt{2}\sigma} - \frac{\varepsilon\sigma}{\sqrt{2}\tilde{n}_s}) - e^{\varepsilon}\Phi(-\frac{\tilde{n}_s}{\sqrt{2}\sigma} - \frac{\varepsilon\sigma}{\sqrt{2}\tilde{n}_s})$. Unlike other generic DP algorithms such as NoisySGD, SeqPATE avoids a linear increase in privacy loss (i.e., a linear increase in ε) on user phrases by careful partitioning of the private data.

This effect is complimentary to other generic, but more intrusive, techniques such as *redaction* and *deduplication* [51] for addressing the same issue. Finally, a user-specific partitioning with SeqPATE also protects multiple secret phrases of the same user (e.g., a combination of SSN, credit card numbers, address, day of birth) *jointly* without incurring a larger privacy loss — a benefit that deduplication does not provide.

5.4 How does DP prevent memorization in SeqPATE?

In practice, the privacy of the language model is usually interpreted as not generating a secret phrase in the training data *as-is* during inference. Thus, one may wonder how DP prevents such unintended memorization of the training data. We remark that the protection against memorization follows the definition of DP. Consider the attack by Carlini et al. [6], which uses a language model to predict a secret phrase s given a prefix. By the closure to post-processing [14], the prediction also satisfies DP. We denote $\mathcal{W}$ as the undesirable event where SeqPATE generates the phrase s verbatim. The DP definition implies that the probability of $\mathcal{W}$ to happen when s is in the SeqPATE's private sets is at most e^{ε} larger than the probability of an alternative SeqPATE model trained without s in those sets. The chances for the latter model to generate text with s are astronomically small. Hence, DP implies that the probability of $\mathcal{W}$ under the former model (i.e. any SeqPATE model in general) is small.

6 Experiments

6.1 Experimental Settings

Datasets. We evaluate our model on two datasets. AirDialog [48] consists of 1M utterances from customer service dialog on flight booking; Europarl_v6 consists of 2M English sentences collected from European Parliament.[5] (See details about datasets in App. E.)

Baselines. We compare SeqPATE with two DP baselines: (1) standard **NoisySGD** trained on the private data with calibrated noise on clipped gradients [1, 22] and further trained on public set $\mathcal{D}^{\text{pub}}$ without protection; (2) based on **NoisySGD**, **NoisySGD+GC** [24] applies a ghost clipping which enables large batch size with memory saving techniques.

Additionally, we use two non-DP methods as reference: (1) **Pri-GPT** trained on the private set without any privacy protection; (2) the public pre-trained GPT-2 model **Pub-GPT** without access to private data. For all methods, we can optionally fine-tune on the generated pseudo-data as a warm-up, and the operation is denoted as $+\tilde{\mathcal{D}}^{\text{pub}}$.

[4] A formal definition of this is called personalized differential privacy, first seen in [16].

[5] www.statmt.org/europarl

Implementation details. All models are fine-tuned from the (public) pre-trained GPT-2 model [35]. The batch size is 32 for all comparing methods except the GC [24] (GC [24] requires 2048). We use Adam [23] and adjust the initial learning rate with a range of 10^{-3} to 10^{-6} for all methods. The δ mentioned in Sec. 5 for all DP methods is 10^{-6}.

For SeqPATE, before training the student model with teacher supervision, we first fine-tune it on the public pseudo-data $\tilde{\mathcal{D}}^{\text{pub}}$ as a warm-up. The coefficient λ that balances supervision for the teacher and the pseudo-data (Eq. 4) is set to 20, where we have tuned it on the validation set of the public pseudo-data. The default number of teacher models is $2k$, where our model works well according to the experiments in App. H. We designed some strategies [6] to reduce memory and disk usage (See strategies and the computational cost in App. I). We run SeqPATE with $2k$ teachers on a single GPU in 3 days. Our code is publicly accessible. [7]. (See details about hyperparameters in App. G.)

Evaluation Metrics. We evaluate the generated text by perplexity (PPL) and Bleu (Bleu-n) [33].

6.2 Overall Performance

Table 1: The performance on the two datasets with sample-level protections (mentioned in Sec. 5.2). All SeqPATE results are statistically significant compared to the strongest baseline under paired sample t-test ($p < 0.05$).

		AirDialog			Europarl_v6		
		PPL ↓	Bleu-3 ↑	Bleu-4 ↑	PPL ↓	Bleu-3 ↑	Bleu-4 ↑
Non-DP	Pri-GPT	3.88	21.51	17.16	23.25	1.77	0.86
	Pub-GPT	63.16	0.31	0.10	57.40	1.02	0.35
	Pub-GPT+$\tilde{\mathcal{D}}^{\text{pub}}$	19.39	0.71	0.25	45.40	1.38	0.52
DP (sample) $\varepsilon = 3$	NoisySGD	17.49	1.97	0.96	37.31	1.28	0.46
	NoisySGD+$\tilde{\mathcal{D}}^{\text{pub}}$	16.78	2.21	1.09	37.69	1.31	0.42
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	11.17	3.15	1.54	35.77	1.56	0.57
	SeqPATE	**8.00**	**5.09**	**3.24**	**33.92**	**1.60**	**0.61**

Protection at the sample level. Tab. 1 show the performance on the two datasets. Among the non-DP baselines, Pri-GPT acts as an upper bound on the performance, since it can fully utilize the private set by discarding privacy protection. Pub-GPT+$\tilde{\mathcal{D}}^{\text{pub}}$ outperforms Pub-GPT on both datasets, showing that the pseudo data is helpful (additional ablation study on the pseudo data in App. J also verifies this). NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ surpasses the above two methods, since it uses a much larger batch size (2048 vs 32) than NoisySGD. Our method, SeqPATE, significantly outperforms NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ (+59% in Bleu4 on AirDialog and +7.0% in Bleu4 on Europarl_v6) while ensuring the same level of privacy protection in terms of ε.

Protection on the user's secret phrases. We evaluate our method for privacy protection of secret phrases mentioned in Sec 5.3. The key step is to partition the data such that each phrase only occurs in the training data of very few teachers, which is straightforward given the user ID associated with the private data. In general, SeqPATE works with any set of secret phrases. In our experiments, we consider a user's full name as their secret phrase since it can be easily recognized from the data. We partition AirDialog's private data according to the accompanying user IDs. As a result, there are 96.6% users whose data are assigned to a single teacher (details about the data partition in App. F).

As described in Sec. 5.3, standard DP methods incur larger privacy loss on secret phrases. In Tab. 13, we see that NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ needs large noise to achieve a satisfactory level of protection on phrases, because ε increases linearly with the frequency of the phrase (group privacy [14]). "Batching users" indicates partitioning data into batches according to users, which helps NoisySGD protect users' phrases (more analyses in App. M). For SeqPATE, the number of teachers trained on data containing the phrase $\tilde{n}_s$ is close to 1 on average after our partition. Thus, SeqPATE provides the same level of protection on users' secret phrases with a smaller noise and thus achieves better performance (+70% and +36% in Bleu4) (see more about the protection level on users' secret phrases in App. F).

[6] We train and conduct the inference on the teachers one-by-one and cache the teachers' outputs.

[7] https://github.com/tianzhiliang/SeqPATE

Table 2: The performance on AirDialog with the protections of users' secret phrases (mentioned in Sec. 5.3). ε_{avg} is the average ε over all secret phrases, as ε of each phrase varies with the frequency of the phrase and the number of teachers (see App. F for detailed analyses about ε_{avg}). All results of SeqPATE are statistically significant compared to the strongest baseline under paired sample t-test ($p < 0.05$).

		PPL ↓	Bleu-3 ↑	Bleu-4 ↑
DP (phrase) $\varepsilon_{\text{avg}} = 3$	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	16.75	1.71	0.57
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ (batching users)	13.42	3.25	1.45
	SeqPATE	**10.10**	**4.20**	**2.46**
DP (phrase) $\varepsilon_{\text{avg}} = 5$	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	16.49	1.89	0.69
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ (batching users)	10.56	4.60	2.87
	SeqPATE	**8.06**	**6.10**	**3.90**

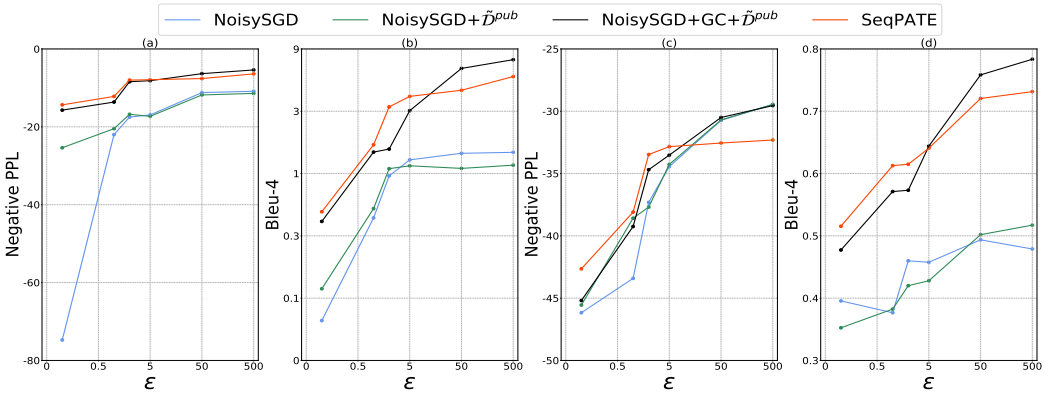

Figure 2: The private-utility tradeoff in Bleu-4 and PPL on a different ε. All the results are under sample level protections. Subfigure a & b show the results on AirDialog; c & d show the results on Europarl_v6. The grey lines show the "lower bound" since the method does not access the private set.

Privacy-utility tradeoff. In Fig. 2, we show the private-utility tradeoff curve of all DP algorithms. [8] Typically, DP with $\varepsilon \in [0.1, 10]$ is considered to provide a meaningful protection [45]. We observe that SeqPATE outperforms NoisySGD and NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ in this range. However, SeqPATE does not work better than the two methods when $\varepsilon > 10$. The reason is that NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ approaches Pri-GPT as ε approaches infinity (i.e. the noise approaches 0). However, SeqPATE with an infinite ε is still weaker than Pri-GPT because distillation still incurs performance loss: the teachers cannot completely transfer knowledge from the private data to the student. Therefore, we suggest using SeqPATE if strong privacy protection is desirable.

Table 3: Ablation studies. "−" means not using that strategy.

	AirDialog			Europarl_v6		
	PPL ↓	Bleu-3 ↑	Bleu-4 ↑	PPL ↓	Bleu-3 ↑	Bleu-4 ↑
SeqPATE	8.00	5.09	3.24	33.92	1.60	0.61
−Merge_P	11.96	3.14	1.85	39.19	1.40	0.47
−KL	12.08	3.26	1.81	39.81	1.41	0.52
$-\mathcal{L}_{\text{pseudo}}$	8.11	4.74	3.17	33.81	1.58	0.60
−Effi KD	9.37	4.45	3.02	34.10	1.57	0.57
−Gaussian	9.54	4.33	2.78	35.31	1.54	0.55
−All	13.21	2.95	1.69	42.74	1.32	0.44

[8] For the models without protections, we consider ε to be zero for baselines using the public data and ε to be infinity for baselines using the private data.

6.3 Ablation Studies

There are several design choices in SeqPATE and we study the importance of each of them. In Tab. 3, we consider the following variants of SeqPATE: (1) −Merge_P: aggregating the teachers by voting instead of averaging their output distributions; (2) −KL: training the student using the cross-entropy loss with respect to teachers' top-1 prediction instead of KL divergence; (3) $-\mathcal{L}_{\text{pseudo}}$: not learning from the pseudo label (Eq. 3); (4) −Effi KD: querying teachers on all samples without selection; (5) −Gaussian: using the Laplace mechanism as the original PATE algorithm instead of the Gaussian mechanism; and (6) −All: using none of the above strategies, which is similar (although not equivalent) to the original PATE (the difference is that PATE needs to roll out all teachers (Sec. 4.1)).

Aggregating the teachers by voting and training with KL loss are the most important strategies for SeqPATE. The poor performance on −Merge_P shows that voting is not suitable for text generation. The reason is that voting over a large output space leads to low agreement rates. The results show that the $\mathcal{L}_{\text{pseudo}}$ loss makes little contribution to SeqPATE. The reason is that we have pre-trained on the student's training set via $\mathcal{L}_{\text{pseudo}}$ before the student's training. The promotion caused by efficient knowledge distillation (Effi KD) on AirDialog is larger than that on Europarl_v6, which shows that the "clever" student (e.g., models on AirDialog with low PPL and high Bleu) benefits more from this strategy. This is because the "clever" student can dramatically save the privacy cost and transfer it to where it would benefit the student most. The poor performance of −All verifies that the original PATE is not suitable for text generation.

Table 4: Analyses about the candidate filtering strategies.

	AirDialog			Europarl_v6		
	PPL ↓	Bleu-3 ↑	Bleu-4 ↑	PPL ↓	Bleu-3 ↑	Bleu-4 ↑
top-p	8.00	5.09	3.24	33.92	1.60	0.61
top-k=1	18.23	0.89	0.38	45.15	1.40	0.53
top-k=10	12.47	3.47	1.95	35.94	1.55	0.54
top-k=50	7.89	4.96	3.35	33.74	1.59	0.59
top-k=100	8.78	4.64	3.17	34.48	1.60	0.62
top-k=200	9.24	3.77	2.94	34.63	1.57	0.55

6.4 Analyses on Candidate Filtering and Teacher Numbers

To analyze candidate filtering with different filtering strategies, we conduct experiments on top-p and top-k filtering. As shown in Tab. 4, our full model employs the top-p filtering (the threshold p is 0.95) surpasses most variants with manually chosen k. Top-k filtering (k =50 or 100) also works well. Filtering with a too small k ($k = 1$ or $k = 10$) implies discarding too much useful information from the supervision ($k = 1$ is different from − KL in Tab. 3, which uses the Top-1 of teachers' results). Filtering with oversize k results in unnecessarily large noises. Candidates with very small probabilities should be filtered during generation; however, random noises may increase their probabilities, so models may generate those words that are misled by the noise.

The results in App. H show that more teachers lead to better results when the number of teachers is in the range of $1 \sim 2k$. This is because the noise assigned to each teacher drops linearly as the number of teachers increases. Note that SeqPATE cannot always benefit from increasing the teacher numbers, because the scale of each teacher's data is linearly decreased as the teacher numbers go up. We choose $\varepsilon = 3$ on the sample level protection for all results in Tabs. 3 and 4.

Additionally, we conduct empirical comparisons and analyses of SeqPATE versus the original PATE in App. N. We show the effects of protections on users' secret phrases in App. O. We compare SeqPATE with another non-DP based baseline (i.e. blacklist based filtering) in App. P. We also conduct a case study in App. Q.

7 Related Work

Text generation models may leak user information through the generated texts [19, 7]. One direction of privacy protection is to protect author-level (user-level) information. The methods prevent attackers from inferring the author attributes (e.g., gender, age) [25] and the relationship between information

and authors [29]. Some researchers [40, 41] infer the membership (whether samples from a given author are used to train the model) given a black-box model. Some papers protect user privacy of training data against untrusted servers via federated learning [27, 10]. Another direction is to prevent attackers from extracting sensitive information in training sets by analyzing the outputs [30, 22], which is urgently needed [7]. Our SeqPATE focuses on this direction. In this direction, regularization methods [6, 43, 20] restrict the model capacity and prevent the model from memorizing exact training samples. Anonymization methods [26, 44] detect sensitive text and replace it with non-sensitive text. Unlike DP [14] methods, the above methods do not provide a quantifiable guarantee for privacy protection. Some researchers focus on protecting user privacy against untrusted servers via federated learning [27, 10].

Some researchers apply DP to text generation. For user-level privacy, ER-AE [4] augments the semantic information in the generated text to hide authors' writing styles from attackers. McMahan et al. [28] propose a recurrent language model with a DP guarantee against the identification of users. Note that the user-level privacy (relationships between users and their information) is different from the privacy of users' secret phrases in our model: Our model prevents individual user phrases from being detected. Some researchers apply NoisySGD to text generation to prevent sensitive training samples from being extracted: some of them [37, 39, 51] employ DP to protect a part of selected tokens; others [22, 50, 24] apply DP to protect both samples and all tokens, but the privacy cost on tokens is very high (Sec. 5.3). Our model falls into the latter category and reduces the privacy cost of tokens. Kerrigan et al. [22] apply NoisySGD [1] to text generation. Yu et al. [50] investigate fine-tuning strategies on pre-trained language models with NoisySGD. Li et al. [24] apply ghost clipping to pre-trained language models with NoisySGD and reduce memory usage. Shi et al. [38] apply DP to particular generation steps instead of training samples or n-grams. Brown et al. [5] analyze DP based method versus data sanitization of text generation models. Brown et al. [12] propose a efficient NoisySGD to speed up model training.

Differential privacy (DP) [13, 14] formally defines and quantifies privacy. ML models with DP guarantee [47, 15, 53] prevent the existence of individual training examples from being detected [6]. Some researchers protect the privacy of empirical risk minimization classifiers [8] and SVM [36] with DP. Following Song et al. [42], NoisySGD [1] achieves DP on deep learning models by adding noises to gradients. Pichapati et al. [34] adaptively clip the gradient in NoisySGD. PATE [31, 32] transfers the knowledge from teacher models trained on private sets with noises to a student model. KNN-PATE [52] refines PATE by accessing only the k-nearest neighbors from the private set. Jordon et al. [21] adversarially learn to generate synthetic data with discriminators trained by PATE. These methods are not customized for text generation models. Xie et al. [49] propose DPGAN to adversarially learn with a generator and a discriminator.

8 Conclusion

In this paper, we propose a novel framework, SeqPATE, to protect the privacy of the training data for text generation models with DP guarantees. SeqPATE achieves a good privacy-utility trade-off by leveraging both private and public data. As an extension of PATE, SeqPATE can handle the sequential generation paradigm with large output space at each step and is therefore adaptive to text generation models. We avoid rolling out teachers by providing pseudo-inputs for the teacher's inference and the student's training. We further reduce the output space by candidate filtering and limit privacy losses via efficient knowledge distillation. SeqPATE achieves a better performance with the sample-level protection and further provides much stronger protection on users' secret phrases. The limitations, ethical considerations, and social impacts of this paper are in App. A and L.

9 Acknowledgement

Research in this paper was supported by Hong Kong Research Grants Council under grand No. 16204920. HH is partly supported by the Samsung Advanced Institute of Technology (Next Generation Deep Learning: From Pattern Recognition to AI). YW is partially supported by NSF Award #2048091. The authors thank Mr. Wei Dong and Dr. Yiping Song for their help and insights on this paper.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, pages 308–318, 2016.

[2] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *ICLM*, pages 394–403. PMLR, 2018.

[3] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473, 2014.

[4] Haohan Bo, Steven HH Ding, Benjamin CM Fung, and Farkhund Iqbal. Er-ae: Differentially private text generation for authorship anonymization. In *NAACL*, pages 3997–4007, 2021.

[5] Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *ACM FAccT*, 2022.

[6] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security*, pages 267–284, 2019.

[7] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security*, pages 2633–2650, 2021.

[8] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. In *JMLR*, volume 12, 2011.

[9] Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. Distilling knowledge learned in bert for text generation. In *ACL*, pages 7893–7905, 2020.

[10] Jieren Deng, Chenghong Wang, Xianrui Meng, Yijue Wang, Ji Li, Sheng Lin, Shuo Han, Fei Miao, Sanguthevar Rajasekaran, and Caiwen Ding. A secure and efficient federated learning framework for nlp. In *EMNLPs*, 2021.

[11] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. In *Journal of RSS, Series B*, 2019.

[12] Christophe Dupuy, Radhika Arava, Rahul Gupta, and Anna Rumshisky. An efficient dp-sgd mechanism for large scale nlu models. In *ICASSP*, pages 4118–4122. IEEE, 2022.

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, 2006.

[14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. In *TCS*, volume 9, pages 211–407, 2014.

[15] James R Foulds, Mijung Park, Kamalika Chaudhuri, and Max Welling. Variational bayes in private settings (vips). In *IJCAI*, pages 5050–5054, 2021.

[16] Arpita Ghosh and Aaron Roth. Selling privacy at auction. In *ACM-EC*, pages 199–208, 2011.

[17] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. In *IJCV*, volume 129, pages 1789–1819, 2021.

[18] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2019.

[19] Dirk Hovy and Shannon L Spruit. The social impact of natural language processing. In *ACL*, pages 591–598, 2016.

[20] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. In *JMLR*, volume 18, pages 6869–6898. JMLR.org, 2017.

[21] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *ICLR*, 2018.

[22] Gavin Kerrigan, Dylan Slack, and Jens Tuyls. Differentially private language models benefit from public pre-training. In *Workshop in EMNLP*, pages 39–45, 2020.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[24] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *ICLR*, 2022.

[25] Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. In *ACL*, pages 25–30, 2018.

[26] Wakana Maeda, Yu Suzuki, and Satoshi Nakamura. Fast text anonymization using k-anonyminity. In *iiWAS*, pages 340–344, 2016.

[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *PMLR*, pages 1273–1282, 2017.

[28] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *ICLR*, 2018.

[29] Fatemehsadat Mireshghallah, Huseyin Inan, Marcello Hasegawa, Victor Rühle, Taylor Berg-Kirkpatrick, and Robert Sim. Privacy regularization: Joint privacy-utility optimization in languagemodels. In *NAACL*, pages 3799–3807, 2021.

[30] Yuta Nakamura, Shouhei Hanaoka, Yukihiro Nomura, Naoto Hayashi, Osamu Abe, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. Kart: Privacy leakage framework of language models pre-trained with clinical records. In *arXiv preprint arXiv:2101.00036*, 2020.

[31] Nicolas Papernot, Martın Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*, 2017.

[32] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with pate. In *ICLR*, 2018.

[33] Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.

[34] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X Yu, Sashank J Reddi, and Sanjiv Kumar. Adaclip: Adaptive clipping for private sgd. In *arXiv preprint arXiv:1908.07643*, 2019.

[35] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[36] Benjamin Rubinstein, Peter Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. In *JPC*, volume 4, pages 65–100, 2012.

[37] Taisho Sasada, Masataka Kawai, Yuzo Taenaka, Doudou Fall, and Youki Kadobayashi. Differentially-private text generation via text preprocessing to reduce utility loss. In *ICAIIC*, pages 042–047. IEEE, 2021.

[38] Weiyan Shi, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. Just fine-tune twice: Selective differential privacy for large language models. *EMNLP*, 2022.

[39] Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. Selective differential privacy for language modeling. In *NAACL*.

[40] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *S&P*, pages 3–18. IEEE, 2017.

[41] Congzheng Song and Vitaly Shmatikov. Auditing data provenance in text-generation models. In *KDD*, pages 196–206, 2019.

[42] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *GlobalSIP*, pages 245–248. IEEE, 2013.

[43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *JMLR*, volume 15, pages 1929–1958. JMLR.org, 2014.

[44] Yu Suzuki, Koichiro Yoshino, and Satoshi Nakamura. A k-anonymized text generation method. In *NBiS*, pages 1018–1026. Springer, 2017.

[45] Aleksei Triastcyn and Boi Faltings. Bayesian differential privacy for machine learning. In *ICML*, pages 9583–9592. PMLR, 2020.

[46] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *AISTATS*, pages 1226–1235. PMLR, 2019.

[47] Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *ICML*, pages 2493–2502. PMLR, 2015.

[48] Wei Wei, Quoc Le, Andrew Dai, and Jia Li. Airdialogue: An environment for goal-oriented dialogue research. In *EMNLP*, pages 3844–3854, 2018.

[49] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. In *arXiv preprint arXiv:1802.06739*, 2018.

[50] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. In *ICLR*, 2022.

[51] Xuandong Zhao, Lei Li, and Yu-Xiang Wang. Provably confidential language modelling. In *NAACL*, pages 943–955, 2022.

[52] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. In *CVPR*, pages 11854–11862, 2020.

[53] Alexander Ziller, Dmitrii Usynin, Rickmer Braren, Marcus Makowski, Daniel Rueckert, and Georgios Kaissis. Medical imaging deep learning with differential privacy. In *Scientific Reports*, volume 11, pages 1–8. Nature Publishing Group, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] It can be found in the supplemental materials.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] It can be found in the supplemental materials.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] It can be found in the supplemental materials.
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] It can be found in the supplemental materials.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] It can be found in the body paper or the supplemental materials.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] It can be found in the body paper or the supplemental materials.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [Yes]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] It can be found in the supplemental materials.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Ethical Considerations and Potential Negative Societal Impacts

This paper aims to protect the privacy of the training data for text generation models. Hence, our paper can tackle some ethical issues about privacy concerns in the existing text generation models (e.g., GPT-2). In terms of motivation and the algorithm, our paper would not cause ethical issues.

However, we should also consider some extreme situations where someone intentionally applies our model to illegal applications. Someone may employ text generation models to create fake news or misinformation and protect himself or herself from being detected. In this way, our model may be used for illegal applications, which is a kind of potential negative societal impact. In the future, we will add some constraints to our model so that our model cannot generate texts for illegal applications (e.g., fake news generation).

In addition, we know large ε (e.g., $\varepsilon = 500$) cannot provide a meaningful protection. We should carefully use our model and cannot assume that the model is perfect no matter what parameters (ε and δ) we use. One possible unethical application is to collect the data from users who believe our model can fully protect their privacy. It means the users may ignore the strength of privacy protection (in terms of the value of ε and δ). That may result in a negative impact on society. Hence, we kindly remind the researchers, who will use this model, to pay more attention to the strength of privacy protection. Further, we should prevent some researchers from collecting data from users who do not have a correct understanding of our algorithm. We suggest that the researchers should ensure the users, who contribute their data, fully understand the risks in our model.

As for the two datasets in the experiments, the Europarl_v6 does not contain the personally identifiable information of the real user. The Airdialog dataset contains some personally identifiable information of users, which enables us to conduct experiments to verify the performance of privacy protection. Note that the dataset had been already published to the public. So, our work in this paper does not further release the user's personal information.

B Algorithm for the Training of SeqPATE

The pseudo code of SeqPATE's training procedure is shown in Algorithm 1.

C Detailed Deduction of the Sensitivity in Sample Level DP

We obtain the Equations in Sec. 5.2 of the paper body since x_j and x'_j are the probability distributions over the vocabulary $\mathcal{V}$.

$$\Delta_2^{(f)} = \|f(\mathcal{D}) - f(\mathcal{D}')\|_2 \leq \|x_j - x'_j\|_2 = \left(\sum_{v=1}^{|\mathcal{V}|}(x_{jv} - x'_{jv})^2\right)^{1/2} \tag{5}$$

We know $(x_{jv} - x'_{jv})^2$ is smaller than $|x_{jv} - x'_{jv}|$ since $|x_{jv} - x'_{jv}| \in (0, 1)$ for each v. Hence, we have,

$$\left(\sum_{v=1}^{|\mathcal{V}|}(x_{jv} - x'_{jv})^2\right)^{1/2} \leq \left(\sum_{v=1}^{|\mathcal{V}|}|x_{jv} - x'_{jv}|\right)^{1/2} \leq \left(\sum_{v=1}^{|\mathcal{V}|}|x_{jv} + x'_{jv}|\right)^{1/2}$$

We know $|a + b| = a + b$ when $a, b \in (0, 1)$, so we have,

$$\left(\sum_{v=1}^{|\mathcal{V}|}|x_{jv} + x'_{jv}|\right)^{1/2} = \left(\sum_{v=1}^{|\mathcal{V}|}x_{jv} + \sum_{v=1}^{|\mathcal{V}|}x'_{jv}\right)^{1/2} = \left(1 + 1\right)^{1/2} \leq \sqrt{2},$$

In summary, the upper bound of the sensitivity is,

$$\Delta_2^{(f)} = \|f(\mathcal{D}) - f(\mathcal{D}')\|_2 \leq \|x_j - x'_j\|_2 = \sqrt{2},$$

Algorithm 1 Training procedure of SeqPATE

Require: $\mathcal{D}^{\text{pri}}, \mathcal{D}^{\text{pub}}$: datasets, GPT: a pre-trained GPT-2 model.
$\{f_\phi^m\}_{m=1}^M$: M teacher models, f_θ: a student model, f_Θ: a student model for self pre-training,
$\{\phi^m\}_{m=1}^M \leftarrow GPT, \Theta \leftarrow GPT$ # Initialize teachers and the student for self pre-training.
GPT generates a pseudo dataset $\tilde{\mathcal{D}}^{\text{pub}}$ based on $\mathcal{D}^{\text{pub}}$.
$\{\mathcal{D}_m^{\text{pri}}\}_{m=1}^M \leftarrow \mathcal{D}^{\text{pri}}$ # Divide private dataset into m subsets.
for all m in M **do**
    Train teacher f_ϕ^m on $\mathcal{D}_m^{\text{pri}}$
end for
Teachers $\{\phi^m\}_{m=1}^M$ conduct inference on $\tilde{\mathcal{D}}^{\text{pub}}$ to get $p_\phi^m(\cdot \mid c)$ required in Sec. 4.2 for all samples.

Train f_Θ on $\tilde{\mathcal{D}}^{\text{pub}}$ # self pre-training for the student.
$\theta \leftarrow \Theta$ # Initialize the student model.

while not converge **do**
    for all batch of samples $\{S\}^{\text{batchsize}}$ in $\tilde{\mathcal{D}}^{\text{pub}}$ **do**
        Student f_θ conducts feed-forward on $\{S\}^{\text{batchsize}}$.
        for all sample S in the batch $\{S\}^{\text{batchsize}}$ **do**
            for all token w_i in sample S **do**
                $p_{\text{agg}}(\cdot \mid c) \propto \frac{1}{M}\sum_{m=1}^M (p_\phi^m(\cdot \mid c) + \mathcal{N}(0, \sigma^2))$ # Aggregate teachers' outputs
                Select only top-k or top-p predicted tokens as student's output.
                Obtain $\mathcal{L}_{\text{teacher}}$ and $\mathcal{L}_{\text{pseudo}}$ as Eq. 4 in the paper body. # Noise is added into $\mathcal{L}_{\text{teacher}}$ to protect the privacy.
                Get $\mathcal{L}$ by combining $\mathcal{L}_{\text{teacher}}$ and $\mathcal{L}_{\text{pseudo}}$ # Knowledge distillation with active learning.
            end for
        end for
        Update ϕ respect to $\mathcal{L}$.
    end for
end while

D Detailed Deduction of the Sensitivity of the Privacy on Users' Secret Phrases

Here, we show the detailed deduction of obtaining the sensitivity of the privacy on users' secret phrases mentioned in Sec. 5.3 (in the paper body). we treat each user's secret phrase s as a data point. As only $\tilde{n}_s$ teacher models can access the phrase s in the private set, changing the phrase s affects at most $\tilde{n}_s$ teacher models. We redefine the neighboring datasets $\mathcal{D}$, $\mathcal{D}'$ are two datasets differ at only one user's secret phrase s. It means the phrase s occurs in one dataset but not in another one. Let j denotes the index of each teacher model, and $\{x_1, ..., x_j, ..., x_a\}$ and $\{x'_1, ..., x'_j, ..., x'_b\}$ mean the teacher outputs affected by the phrase s (for the datasets $\mathcal{D}$ and $\mathcal{D}'$). We have $a \leq \tilde{n}_s$ and $b \leq \tilde{n}_s$ since changing a secret phrase s affects at most $\tilde{n}_s$ teacher models. We calculate the sensitivity of f as follows.

$$\Delta_2^{(f)} = \|f(\mathcal{D}) - f(\mathcal{D}')\|_2 \leq \|\sum_{j=1}^{a} x_j - \sum_{j=1}^{b} x'_j\|_2 \leq \|\sum_{j=1}^{\tilde{n}_s} x_j - \sum_{j=1}^{\tilde{n}_s} x'_j\|_2$$

For the above equation, we obtain the following equation (x_j and x'_j are the probability distributions over the vocabulary $\mathcal{V}$).

$$\|\sum_{j=1}^{\tilde{n}_s} x_j - \sum_{j=1}^{\tilde{n}_s} x'_j\|_2 = \left(\sum_{v=1}^{|\mathcal{V}|}(\sum_{j=1}^{\tilde{n}_s} x_{jv} - \sum_{j=1}^{\tilde{n}_s} x'_{jv})^2\right)^{1/2} = \left(\sum_{v=1}^{|\mathcal{V}|}(\frac{\tilde{n}_s}{\tilde{n}_s}\sum_{j=1}^{\tilde{n}_s} x_{jv} - \frac{\tilde{n}_s}{\tilde{n}_s}\sum_{j=1}^{\tilde{n}_s} x'_{jv})^2\right)^{1/2}$$
$$= \left(\sum_{v=1}^{|\mathcal{V}|}\tilde{n}_s^2(\frac{\sum_{j=1}^{\tilde{n}_s} x_{jv}}{\tilde{n}_s} - \frac{\sum_{j=1}^{\tilde{n}_s} x'_{jv}}{\tilde{n}_s})^2\right)^{1/2} = \tilde{n}_s\left(\sum_{v=1}^{|\mathcal{V}|}(\frac{\sum_{j=1}^{\tilde{n}_s}(x_{jv} - x'_{jv})}{\tilde{n}_s})^2\right)^{1/2}$$

We know $(\frac{\sum_{j=1}^{\tilde{n}_s}(x_{jv}-x'_{jv})}{\tilde{n}_s})^2 \in (0,1)$ since $|x_{jv}-x'_{jv}| \in (0,1)$. Then, we have $(\frac{\sum_{j=1}^{\tilde{n}_s}(x_{jv}-x'_{jv})}{\tilde{n}_s})^2 \leq |\frac{\sum_{j=1}^{\tilde{n}_s}(x_{jv}-x'_{jv})}{\tilde{n}_s}|$. Hence, we have,

$$\tilde{n}_s\Big(\sum_{v=1}^{|\mathcal{V}|}(\frac{\sum_{j=1}^{\tilde{n}_s}(x_{jv}-x'_{jv})}{\tilde{n}_s})^2\Big)^{1/2} \leq \tilde{n}_s\Big(\sum_{v=1}^{|\mathcal{V}|}|\frac{\sum_{j=1}^{\tilde{n}_s}(x_{jv}-x'_{jv})}{\tilde{n}_s}|\Big)^{1/2} = \tilde{n}_s\Big(\frac{1}{\tilde{n}_s}\sum_{v=1}^{|\mathcal{V}|}|\sum_{j=1}^{\tilde{n}_s}(x_{jv}-x'_{jv})|\Big)^{1/2}$$

$$\leq \tilde{n}_s\Big(\frac{1}{\tilde{n}_s}\sum_{v=1}^{|\mathcal{V}|}\sum_{j=1}^{\tilde{n}_s}|x_{jv}-x'_{jv}|\Big)^{1/2} = \tilde{n}_s\Big(\frac{1}{\tilde{n}_s}\sum_{j=1}^{\tilde{n}_s}\sum_{v=1}^{|\mathcal{V}|}|x_{jv}-x'_{jv}|\Big)^{1/2} \leq \tilde{n}_s\Big(\frac{1}{\tilde{n}_s}\sum_{j=1}^{\tilde{n}_s}\sum_{v=1}^{|\mathcal{V}|}|x_{jv}+x'_{jv}|\Big)^{1/2}$$

$$= \tilde{n}_s\Big(\frac{1}{\tilde{n}_s}\sum_{j=1}^{\tilde{n}_s}\sum_{v=1}^{|\mathcal{V}|}(x_{jv}+x'_{jv})\Big)^{1/2} = \tilde{n}_s\Big(\frac{1}{\tilde{n}_s}\sum_{j=1}^{\tilde{n}_s}(\sum_{v=1}^{|\mathcal{V}|}x_{jv}+\sum_{v=1}^{|\mathcal{V}|}x'_{jv})\Big)^{1/2} = \tilde{n}_s\Big(\frac{\sum_{j=1}^{\tilde{n}_s}(1+1)}{\tilde{n}_s}\Big)^{1/2} = \sqrt{2}\tilde{n}_s$$

In summary, the upper bound of the sensitivity is,

$$\Delta_2^{(f)} = \|f(\mathcal{D}) - f(\mathcal{D}')\|_2 \leq \|\sum_{j=1}^{a}x_j - \sum_{j=1}^{b}x'_j\|_2 \leq \sqrt{2}\tilde{n}_s,$$

E Descriptions about the Datasets

The AirDialog dataset [48] [9] consists of 402,038 dialogues. Each dialogue consists of more than two utterances. We treat each utterance as a sample (i.e. sentence) in our sentence completion task. The Airdialog dataset contains some personally identifiable information about users. It contains the users' names of the dialog speakers, which enables us to conduct experiments to verify the performance of privacy protection. Note that the dataset had been already published to the public. In this way, We obtain the Europarl_v6 dataset from a machine translation benchmark [10], where we only use the monolingual English dataset with 2,015,440 raw sentences. The Europarl_v6 does not contain the personally identifiable information of the real user.

For the above datasets, we filter the short sentence with less than eight tokens. Then, the first four tokens act as the prefix, and the rest of the tokens act as the output (ground-truth). We split each datasets into a private set $\mathcal{D}^{\text{pri}}$ and a public set $\mathcal{D}^{\text{pub}}$. For the AirDialog dataset, the private set contains 0.95M/5K/50K samples for training/validation/testing, and the public set contains 40K/5K for training/validation. For the Europarl_v6 dataset, the private set contains 1.72M/10K/50K samples for training/validation/testing, and the public set contains 40K/5K for training/validation. The vocabulary size for the two datasets is set to 50K. We replace the tokens out of the vocabulary with a special token.

F Dataset Partitions and Experiments about the Protection on Users' Secret Phrases

To achieve the protection of users' secret phrases mentioned in Sec. 5.3, we partition the original dataset into teachers' training data with the following principles: (1) the teacher number for each user should be small; (2) the scales of the data for every teacher are roughly balanced.

There are 9131 users in AirDialog's private training set. As shown in Tab. 6, after the above processing, the number of users whose data are assigned to a single teacher is 8824; there are 263 users whose data occurs in 2 teachers' training data; there are 41 users whose data belong to 3 teachers; only 3 users' data are accessed by more than 3 teachers. On average, the number of teachers for each user is 1.039.

The users' secret phrases mentioned in Sec. 5.3 are often the phrases known by a few users (occurs in a few users' data). For a secret phrase s, the number of teachers accessing the phrase $\tilde{n}_s$ is very small,

[9] The dataset comes from https://github.com/google/airdialogue

[10] The description can be found at https://www.statmt.org/europarl. The data come from https://statmt.org/wmt11/training-monolingual.tgz

# Users	8824	263	41	3
# Teachers for each user	1	2	3	> 3

Table 5: The statistical information of the teacher numbers for all users in the AirDialog's private training.

and therefore the protections provided by SeqPATE on those phrases are naturally strong (if many users know s, the $\tilde{n}_s$ is large and the protection is naturally weak). The secret phrases may be phone number, address, name, or SSN number. In the AirDialog dataset, the description of each sample contains the "user's full name". With that information, we can easily check the existence and count the frequency of the "user's full name" in each sample. So, we can easily evaluate the protections on secret phrases by treating "user's full name" as the secret phrase.

Methods	ε_{avg}	min ε	max ε	% $\varepsilon \leq$ average ε
NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	3	0.12	731.58	52.8%
SeqPATE	3	2.85	25.64	95.4%
NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	5	0.20	1219.31	52.8%
SeqPATE	5	4.75	42.78	95.4%

Table 6: The statistical information of ε on all the users' secret phrases under the protection of different algorithms. The first three columns show the average/minimal/maximal ε over all secret phrases. The last column indicates the percentage of secret phrases whose ε is lower than the average ε.

In the experiments about protections on users' secret phrases, the ε of each phrase is different. In NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$, the ε of the phrase s relies on the phrase frequency n_s; in SeqPATE, the ε of the phrase s relies on the teacher number $\tilde{n}_s$. Hence, the ε_{avg} reported in "DP (phrase)" of the second Table in Sec. 6.2 means average ε over all secret phrases. There are 6705 secret phrases in the AirDialog dataset. In the model training, we apply a same scale of noises to the algorithm and then calculate the exact ε for all secret phrases. The rows 1 and 2 of Tab. 6 show the ε of the models in rows 1 and 2 of the second Table in Sec. 6.2. If we fix the average ε_{avg} at 3, the ε of phrases on NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ range from 0.12 to 731.58, and the ε of phrases on SeqPATE range from 2.85 to 25.64. The rows 3 and 4 of Tab. 6 show the ε of the models in rows 3 and 4 of the second Table in Sec. 6.2. If we fix the average ε_{avg} at 5, the ε of phrases on NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ range from 0.20 to 1219, and the ε of phrases on SeqPATE range from 4.75 to 42.74.

From the Tab. 6, we can observe that 95.4% phrases have a lower ε (stronger protection) than the average ε_{avg} (3 or 5). However, NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ can only ensure 52.8% phrases enjoy a stronger protection than the average level. Hence, compared to NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$, SeqPATE can provide strong protection on more secret phrases, even if we assign the same ε_{avg} to SeqPATE and NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$.

G Details about the Experimental Setting

All the comparing methods use the same base model, the GPT-small, which has 12 stacked layers as mentioned in the original paper [35]. The pre-trained GPT-2 model comes from the official website [11]. We truncate the sentences with the maximal sentence length of 40. In the top-p strategy, the threshold p is 0.95 (We have tried the threshold of $0.90 \sim 1.0$ and found $p = 0.95$ works well). The threshold in active learning mentioned in Sec. 4.3 is 10 or 5 (We also need to tune the parameter). The hyperparameter tuning is conducted on the validation set of the public pseudo data, so tuning does not introduce additional privacy losses. For all our experiments, we adopt *autodp* [46] — an open-source library that implements the analytical Gaussian mechanism for privacy accounting and calibration. We use the TESLA V100 GPU devices with 32GB memory on a Slurm HPC cluster.

[11] github.com/openai/gpt-2

H Analyses about the Number of Teacher Models

Tabs. 7 & 8 show the analyses about SeqPATE's performance with different teacher numbers on our two datasets. We evaluate the performance with the sample level protection ($\varepsilon = 3$).

	AirDialog				
	PPL ↓	B-1 ↑	B-2 ↑	B-3 ↑	B-4 ↑
#teacher=1	19.28	8.59	2.35	0.86	0.28
#teacher=10	16.57	7.97	2.24	0.85	0.30
#teacher=200	10.96	12.81	5.13	2.89	1.34
#teacher=2k	8.00	15.14	8.30	5.09	3.24

Table 7: SeqPATE's performance with different teacher numbers on the AirDialog dataset.

	Europarl_v6				
	PPL ↓	B-1 ↑	B-2 ↑	B-3 ↑	B-4 ↑
#teacher=1	41.56	12.39	3.71	1.13	0.39
#teacher=10	38.94	12.89	3.75	1.21	0.44
#teacher=200	34.55	13.25	4.18	1.36	0.51
#teacher=2k	33.92	13.75	4.69	1.60	0.61

Table 8: SeqPATE's performance with different teacher numbers on the Europarl_v6 dataset.

I The Computational Cost of SeqPATE

It seems that our model requires huge computational resources and a costly infrastructure to run. However, our model can train and infer on a single GPU machine. In this section, we introduce some simple strategies we used in our implementation and also introduce the total computational cost of our model.

Memory usage and hard-disk space usage. Since our method uses a large number of teachers, the naive implementation of loading all teachers into the memory for aggregation is impractical. However, note that our algorithm only needs to access the teachers' top-k predictions. Therefore, we train teacher models sequentially. Once a teacher model is trained, we obtain its top-k predictions (k=200 at most in our experiments) on the public training data and save the results (i.e. k probabilities). Then, we discard the teacher model. Finally, SeqPATE only needs the teacher's supervision on a small number of samples. In our experiments, training on 500∼1k teacher labeled samples is sufficient. Overall, saving teachers' inference results uses 8∼16GB. The memory usage is similar to that of a GPT2 model because we do not load all teacher models into the memory and instead run inference sequentially and merge teachers' predictions offline.

Training time. While we have a large number of teachers, each teacher is trained on only a small fraction of the entire dataset. Thus, the time it takes to train all teachers is roughly equal to the time of training a single GPT2 model on the full dataset (of 1∼2M samples in our experiments). In practice, the teachers' training time of SeqPATE on AirDialog dataset is roughly 1 or 2 days; their training time on Europarl_v6 dataset is 2 or 3 days. For both datasets, the student's training time is range from several minutes to half an hour. For the NoisySGD, the whole training takes 1 or 2 days. The running time of the inference for all methods is similar, which takes around 10 minutes.

In summary, with the simple strategies, the teacher training and aggregation steps are not much more expensive than training a GPT2 model. Compared to standard NLG model training, our algorithm does not require special hardware or distributed learning.

J The Contribution of the Pseudo Public Dataset $\tilde{\mathcal{D}}^{\text{pub}}$

The following experiments verify the contribution of the pseudo-public dataset $\tilde{\mathcal{D}}^{\text{pub}}$ to our task. We conduct the experiments on the AirDialog dataset. The results in Tab. 9 shows that using

$\tilde{\mathcal{D}}^{\text{pub}}$ can prompt the performance a lot, where the promotion can be found in both SeqPATE and NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ methods. SeqPATE relies highly on the pseudo data since the pseudo data provide the input text in the student's training.

Methods	Dataset	PPL ↓	B-1 ↑	B-2 ↑	B-3 ↑	B-4 ↑
NoisySGD+GC	w/ $\tilde{\mathcal{D}}^{\text{pub}}$	11.17	13.21	5.90	3.15	1.54
NoisySGD+GC	w/o $\tilde{\mathcal{D}}^{\text{pub}}$	12.05	12.94	5.97	2.96	1.36
SeqPATE	w/ $\tilde{\mathcal{D}}^{\text{pub}}$	8.00	15.14	8.30	5.09	3.24
SeqPATE	w/o $\tilde{\mathcal{D}}^{\text{pub}}$	16.12	10.86	3.92	2.13	0.96

Table 9: The comparisons between using and not using the pseudo dataset, $\tilde{\mathcal{D}}^{\text{pub}}$.

K The Illustration of a Running Example

Here, we will use an example to show our training processing. In this example, the prefix from the public dataset $\mathcal{D}^{\text{pub}}$ is "I want to book". We feed the prefix to a pre-trained GPT-2 model to generate a pseudo sentence "I want to book a flight from Tokyo to Hawaii". The pseudo sentence serves as an example in the pseudo-public dataset $\tilde{\mathcal{D}}^{\text{pub}}$. We feed the pseudo sentence to the teacher models to conduct the teachers' inference; we also feed it to the student model to conduct the feed-forward of the student's training. Teacher models output the probability distributions on all words (10 words, in total) of the sentence. Then, we aggregate all teachers' probability distributions and add the calibrated noises to the aggregated distributions. The student model also generates the corresponding probability distributions on those words. We conduct the knowledge distillation with active learning and the top-k or top-p filtering over the student's probability distributions. For example, if the student model can do well on the words ("I", "want", "book", "flight", and "Tokyo"), the student queries the teachers' output distributions only on the rest of the words ("to", "from", "to", and "Hawaii"). The student is supervised by the teachers' outputs via the KL loss mentioned in Sec. 4.3. Besides, the student is always supervised by the $\mathcal{L}_{\text{pseudo}}$ loss on the whole pseudo sentence ("I want to book a flight from Tokyo to Hawaii"). Finally, the student model conducts back-propagation according to the above losses.

L Limitation of This Paper

Even if our proposed method obtains remarkable performance, this work still has some limitations. We will continue to focus on this topic and try to address those limitations in future work.

Firstly, compared to NoisySGD-based methods, our model is not good at handling "big ε" (the ε is very large, e.g., $\varepsilon = 50$). As mentioned in Sec. 6.2, the phenomenon is reasonable since the knowledge distillation in our method cannot completely transfer knowledge but a very large ε in NoisySGD results in a very small noise. Note that researchers usually treat that ε ranging from 0.1 to 5 provides meaningful protections. Too large ε can hardly provide sufficient protections for the data. $\varepsilon > 5$ says that an individual could be identified with the confidence of more than 99.33% [45]. Hence, this limitation is not a big issue for this paper.

Secondly, compared with other papers, our experiment processing may be complex since our default setting is to use $2k$ teacher models. It means that we need to train $2k$ teacher models to conduct the experiments. Fortunately, we design some strategies to enable all teacher models to train on a single GPU within 3 days (according to App. I),); thus the computational cost is not very high. In practice, we can use a shell script to run the $2k$ teachers automatically so the operations in the experiment are not so heavy to conduct.

Thirdly, we have not applied our method to other text generation applications, such as machine translation and summarization. The privacy concern in those tasks is also an urgent need. We may try to apply SeqPATE to some new applications in the future. Since the state-of-the-art models in those applications may have some sophisticated components , we believe some further works are needed to apply our model to other text generation tasks.

M More Explanation of the Experiments on Users' Secret Phrases

SeqPATE achieves strong privacy protections on users' secret phrases by carefully partitioning the private set according to users. Note that NoisySGD (DP-SGD) can also follow the similar idea and partition the private set to training batches according to users, that is, we try to ensure a user's data is in one or a few batches. We call this method "batching users ". We conducted experiments in this way and reported the results in Table 13 (row 2 and 4). The experimental results show that our method on protecting users' secret phrases still outperforms the NoisySGD baselines with batching users.

		PPL ↓	Bleu-3 ↑	Bleu-4 ↑
DP (phrase) $\varepsilon_{\text{avg}} = 3$	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	16.75	1.71	0.57
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ (batching users)	13.42	3.25	1.45
	SeqPATE	**10.10**	**4.20**	**2.46**
DP (phrase) $\varepsilon_{\text{avg}} = 5$	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	16.49	1.89	0.69
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ (batching users)	10.56	4.60	2.87
	SeqPATE	**8.06**	**6.10**	**3.90**

Table 10: The performance on AirDialog with the protections of users' secret phrases (mentioned in Sec. 5.3). This Table is the same as the table 2 in the paper body

We note that SeqPATE still has some advantages over NoisySGD (with batching users) in protecting users' secret phrases:

- The privacy loss of SeqPATE scales linear with the teacher number $\tilde{n}_s$ for a user's data as mentioned in Sec. 5.3. The average $\tilde{n}_s$ is 1.038 as mentioned in Appendix F. The privacy loss of NoisySGD (with batching users) scales with the root of training steps (number of batches the model trained) according to advanced composition [1]. Therefore, if the training phase consists of K epochs, a user's phrase contributes to the privacy loss for K times. Deep learning models usually require many epochs for training. In this paper, the epoch number is usually $10 \sim 20$. In short, NoisySGD's privacy loss on phrases is at least $3 \sim 4$ times larger than its sample level privacy loss; SeqPATE's privacy loss on phrases is roughly equal to its sample level privacy loss.
- It would be difficult to adjust the batch size in NoisySGD to satisfy the requirements to batching users, because (a) the performance of many deep learning models is sensitive to the batch size; (b) the batch size cannot be too large due to the limitations of GPU memory.

N Empirical Comparisons and Analyses of SeqPATE Versus the Original PATE

We claimed that the original PATE is hard to directly work on text generation tasks. Here, we provide more detailed analyses about it and verify this claim with some experimental results and estimations.

Firstly, the original PATE is required to roll out all teachers to collect all teachers' inference results. At each step, the input word of all the teachers and the student comes from the previous output of the teacher inference. It means that we need to online align all teachers' inference and student training at each step (conducting teacher inference and student training synchronously). Hence, we need to either (1) load all teachers and the student to a single program, or (2) run teachers and the student serially and merge teachers' results at each step. Given that the teacher number is usually more than 100 (2000 teachers obtain the best performance in our setting). We cannot load them into a program due to the GPU or CPU memory. If we conduct the training serially, the computational cost is extremely high so it is almost impossible to roll out the teachers.

Secondly, even if we do not consider the teachers' rolling out, the performance of the original PATE is also far from satisfactory. In the ablation study (Table 3 in the paper body), $-$All indicates that SeqPATE gives up all proposed strategies except conducting knowledge distillation on the pseudo data. $-$All underperforms our full model and the gap between $-$All and our full model is also quite large (Bleu4 of $-$All drops from 3.24 to 1.69).

In summary, considering the performance and computational cost, the original PATE almost cannot work in text generation. SeqPATE does make a great improvement to adapt PATE to the text generation.

O The Intuitive Effects of Protections on Users' Secret Phrases

DP-based methods usually measure the strength of privacy protection via the factor ε and δ according to the DP definition. As for the text generation application, we employ a more practical evaluation to show what and how the DP-based methods protect the privacy.

We define a metric R_{name} to measure the average percentage of generating users' names in the output text. This metric indicates the degree of leaking users' secret phrases (i.e. user name). A smaller R_{name} indicates better protection.

	Methods	$R_{\text{name}} \downarrow$
Non-DP	Pri-GPT	4.25%
DP (sample)	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ (batching users)	1.89%
$\varepsilon = 3$	SeqPATE	0.21%

Table 11: The average percentage of generating users' names in the output texts. The corresponding sample-level ε is 3.

	Methods	$R_{\text{name}} \downarrow$
Non-DP	Pri-GPT	4.25%
DP (phrase)	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ (batching users)	0.43%
$\varepsilon_{\text{avg}} = 3$	SeqPATE	0.20%

Table 12: The average percentage of generating users' names in the output texts. The corresponding ε users' on secret phrases is 3.

Table 11 shows the results on Pri-GPT and DP-based methods with the sample-level ε is 3. The results show that our SeqPATE significantly avoids generating trained users' names trained (avoids 95% of them). The Pri-GPT has no privacy protection and the percentage of generating users' names is high (4.25%), which demonstrates that information leakage is serious in the current pre-trained models (Pri-GPT). Under the same level of protection ($\varepsilon = 3$), SeqPATE provides stronger protection than NoisySGD. It verifies our claim (in Sec. 5.3) that SeqPATE is skilled at protecting users' secret phrase.

Table 12 shows the results on Pri-GPT and DP-based methods with ε of 3 in the users' phrases. Under the same ε of protections on users' phrases, the gap between SeqPATE and NoisySGD is not so large and SeqPATE is still better than NoisySGD. It shows the superiority of SeqPATE in protecting users' phrases.

In summary, SeqPATE shows its superiority in protecting both samples and users' phrases, and SeqPATE avoids leaking information significantly.

P Experiments about Protecting the Privacy by Filtering with a Blacklist

Though the blacklist-based methods and DP-based methods are not comparable, we did add a new experiment, where we create a blacklist with the user name, destinations, and some other sensitive words/phrases (e.g. dates).

the blacklist-based method cannot theoretically measure the "degree or strength" of privacy protection, so it is hard to compare DP-based methods with non-DP-based methods in the same (fair) level of protection.

Pri-GPT-blacklist indicates applying the blacklist to the results of the GPT model trained on private data. For each generated sentence, we replace the words in blacklist with a special token. Notice that, although Pri-GPT-blacklist outperforms SeqPATE, the blacklist-based methods have the following issues: (1) It only protects the privacy of the given types (i.e. user names, destinations, and dates); (2) Even for the given types, it can only protects a part of sensitive information since the blacklist is finite; (3) It cannot measure the strength of protection.

		PPL ↓	Bleu-3 ↑	Bleu-4 ↑
DP (phrase) $\varepsilon_{\text{avg}} = 5$	Pri-GPT-blacklist	6.84	11.40	8.13
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ (batching users)	10.56	4.60	2.87
	SeqPATE	8.06	6.10	3.90

Table 13: Comparison between DP-based method versus the private GPT with blacklist filtering

Good Case	Input	As of your requirements,
	Ground Truth	As of your requirements, there is only AA airline available. Shall I go for further process to book ?
	Pri-GPT	As of your requirements, there is a flight-1008 on UA airline, is it ok for you?
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	As of your requirements, I contact you about this issue? Thank you.
	SeqPATE	As of your requirements, I have found that you are willing to cancel your reservation. Please confirm your reservation booking.
Good Case	Input	Yes, I would like
	Ground Truth	Yes, I would like to prefer a connecting flight.
	Pri-GPT	Yes, I would like to travel from BOS to HOU.
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	Yes, I would like to help you. If you need any assistance please feel free to contact us.
	SeqPATE	Yes, I would like to attend a meeting with the team to discuss a list of all the team members.
Bad Case	Input	I am Janet Miller.
	Ground Truth	I am Janet Miller. I want to change my existing reservation , as my father already done a reservation for m.
	Pri-GPT	I am Janet Miller.
	NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$	I am Janet Miller. do you want to travel? If you are interested, please contact me. Thank you.
	SeqPATE	I am Janet Miller. I am a member of the International Association of American Airline flight attendants.

Table 14: Case studies with two good cases and one bad case.

Q Case Study

We report two good cases and a bad case in Table 14. The input is a prefix with the first four tokens of the sentence. Given the input, the three models generate the whole sentence. In the first case, Pri-GPT generates the flight number, which may leak sensitive information. NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ and SeqPATE success to hide the flight number. Compared to NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$, SeqPATE is more similar to the ground-truth (both of them are talking about booking). In the second case, Pri-GPT generates the locations, which is also quite sensitive. The outputs of NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ and SeqPATE do not contain so much individual information.

In the last case, the input text contains the user's name, which is a piece of sensitive information. NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ avoids talking about the individual information. Pri-GPT does not continue to generate, so it does not leak any information. However, SeqPATE generates the company name of the flight, which is a sensitive phrase for individuals. We note that such a bad case is very rare in the model outputs. DP-based methods aim to trade-off the privacy protection and the model performance. Sometimes, SeqPATE generates informative, appropriate, but too detailed texts, where the details may contain sensitive information. Nevertheless, most cases of SeqPATE achieve to generate sentences with a high quality and enough privacy protection.