# SeqPATE: Differentially Private Text Generation via Knowledge Distillation

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Protecting the privacy of user data is crucial for text generation models, which may 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 protections on training samples. However, text generations conduct sequential generations on a large output space, and PATE is not customized for that paradigm. Besides, PATE does well in protecting sample-level privacy but requires a high privacy cost on protecting phrases in samples. In this paper, we propose SeqPATE, an extension of PATE on text generation, which aims to protect the privacy of both training samples and sensitive phrases in samples. To adapt to text generations, we generate pseudo inputs and reduce the sequence generation problem to next word predictions. To handle the large output space, we propose a candidate filtering strategy to dynamically reduce the space, and refine the teacher aggregation of PATE to avoid low agreement due to voting for a large number of candidates. To reduce privacy losses, we design an efficient knowledge distillation to decrease the frequency of querying teachers. The experiments verify the effectiveness of SeqPATE in protecting both samples and sensitive phrases.

## 1 Introduction

Recent work showed that sensitive user information in the training corpora, such as addresses and names, can be extracted from text generation models [5]. Providing privacy guarantees to the training corpora of text generation models becomes 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 [39, 3, 1] is a popular DP algorithm for deep learning, which adds noise to the gradients. PATE [29] 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 the teacher predictions to satisfy DP. PATE is model-agnostic, and its privacy cost derives from the knowledge distillation instead of the gradients of the whole model in NoisySGD [39, 22]. Hence, noises required by PATE do not scale with model size. With this benefit, PATE has a great potential for being adapted to text generations, since large models (e.g., GPT-2 [33]) become the mainstream in text generation. However, NoisySGD and PATE are used to protect sample-level privacy [46, 22] but require a high privacy cost on protecting sensitive phrases in samples [20, 36]. PATE, designed for classification tasks, is not customized for sequential generation on a large output space (i.e., vocabulary), which is very common in text generation.

In this paper, we propose SeqPATE, a DP learning algorithm for text generations to protect the privacy of both samples and sensitive phrases in training corpora with a moderate privacy cost. As an extension of PATE, SeqPATE employs a teacher-student framework: (i) a student model learns to

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

generate texts on non-sensitive samples; and (ii) a number of teacher models, trained on sensitive texts, supervise the student through the outputs of aggregated teachers. This framework still faces some challenges in text generations. Firstly, it suffers from 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 must 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). Secondly, the large output space (i.e., vocabulary) in text generation leads to (i) low agreement rates among teachers and (ii) large noises required by DP, both of which would 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 large probabilities. Also, we aggregate teachers' outputs by interpolating their output distributions instead of voting with argmax predictions. To reduce the privacy losses, selectively apply the knowledge distillation to the generation step only if the student has trouble with the generation.

Most DP learning methods, including SeqPATE, prevent the existence of individual training samples from being detected. SeqPATE has further advantages in protecting users' secret phrases (e.g., addresses) that may occur multiple times in the corpus. We evaluate our method on a sentence completion task. The comparison with a recent NoisySGD baseline [22] on the AirDialog dataset shows that SeqPATE improves Bleu4 by 59% on protecting samples and by 391% on protecting users' secret phrases with a meaningful strength of privacy protection.

Our contribution is twofold: (i) We propose SeqPATE to achieve the protections on both sample level and phrase level and also provide theoretical analyses on the two levels. The experiments show SeqPATE's advantages on the two levels compared to the baselines. (ii) We propose several strategies for SeqPATE to handle text generations with a sequence of classifications over large spaces.

## 2 Problem Setup

Our goal is to achieve the privacy protection quantified by DP in the text generation: preventing 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 an autoregressive language model to achieve the completion. The public set $\mathcal{D}^{\text{pub}}$ consists of the prefixes, which can hardly contain sensitive information. The private set $\mathcal{D}^{\text{pri}}$ consists of the 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. In writing assistants where a sample has an input prompt and an essay body, the prompt can be public but the body written by authors is private.

## 3 Background on DP and PATE

**Definition 3.1.** [Differential privacy (DP) [11, 12]] 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 $(\varepsilon, \delta)$-differentially private if,

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

As the definition, DP is a quantifiable definition of privacy that provides guarantees on identifications of individual data (prevent an adversary from inferring whether the input is $\mathcal{D}$ or $\mathcal{D}'$). ML models with DP guarantee ensure that each individual 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.

PATE [29], 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 via a

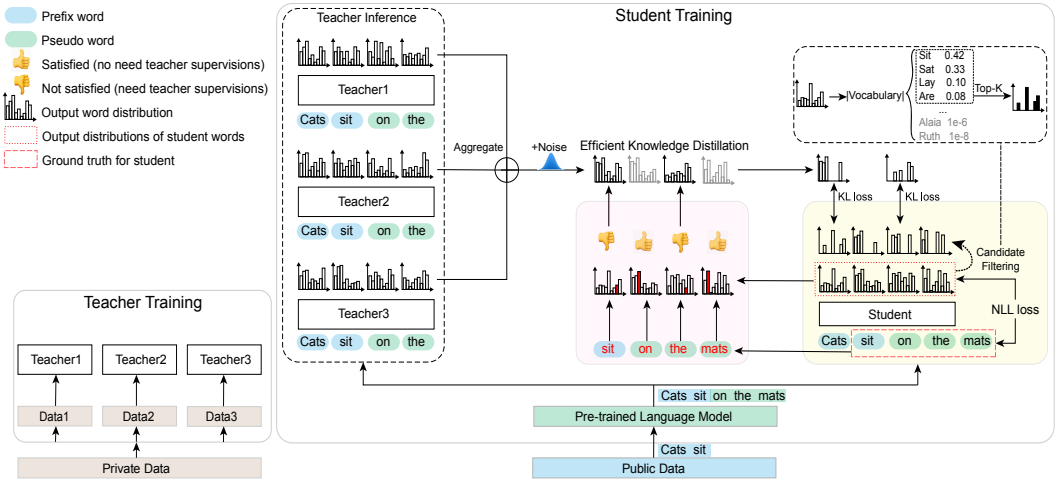


Figure 1: Overview of SeqPATE. SeqPATE trains teachers on the private data. It conducts the student's training and the teachers' inference on pseudo sentences generated by a pre-trained language model given a public prefix. The student is supervised by the aggregation of teacher output distributions. SeqPATEbs benefits from the candidate filtering (white block in the top right corner) and the efficient knowledge distillation that determines whether needs teacher supervision (pink block).

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 teachers' outputs. Given the trained teachers, we employ each teacher to provide supervision on 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. L.

## 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 teachers' predictions by majority vote. While it works for classification problems with a relatively small number of classes, the output space of text generation 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. [8, 15], we aggregate teachers' 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. [30], we apply the Gaussian mechanism. Formally, let $p_\phi^m(\cdot \mid c)$ be the $m$-th teacher's prediction. The aggregated distribution is $p_{\text{agg}}(\cdot \mid c) \propto \frac{1}{M}\sum_{m=1}^{M}(p_\phi^m(\cdot \mid c) + \mathcal{N}(0, \sigma^2))$, [1] where the Gaussian noise is added to the aggregated output distribution.

### 4.3 Training of the Student Model

The student model is trained on the 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 contexts. Given contexts from (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 act 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 the teachers' supervision. Note that the teachers' aggregated output is a distribution over words. Therefore, we minimize the forward KL divergence between the aggregated teacher distribution $p_{\text{agg}}$ and the student output distribution $p_\theta$:

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

- $\mathcal{L}_{\text{pseudo}}$ 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. ($\lambda$ balances the two terms and we discuss the noise scale $\sigma$ 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 available privacy budgets to release the information more 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 [16], $k$ 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. [2]

**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 the teacher supervision only on "hard" examples in student's training. 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.

[1] 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 big (e.g., $2k$) and the first term dominates the above equation.

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

Concretely, we query the 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 $\Phi$ 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 [10]). *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 noise multiplier $\sigma$ satisfies the same privacy guarantee of a single Gaussian mechanism with a noise multiplier $\sigma/\sqrt{k}$. By fixing $k$ and $\varepsilon$, we can calibrate the noises by choosing an appropriate $\sigma$ in our model.

## 5.2 Differential Privacy for Language Models at the Sample Level

Recall that we partition the private dataset to $M$ 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 over all the teachers. Since the datasets are disjoint and have no duplicate samples, changing one sample only affects one teacher model. For neighboring datasets $\mathcal{D}$, $\mathcal{D}'$, let $j$ denote the index of each teacher model, then the probability distributions $x_j$ and $x'_j$ (deriving from $\mathcal{D}$ and $\mathcal{D}'$ respectively) are different. Then, the sensitivity $\Delta$ in Lemma 5.1 & 5.2 is (See detailed deductions in App. C),

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

Then, adding the noises calibrated in Lemma 5.2 to each coordinate is sufficient to preserve ($\varepsilon$, $\delta(\varepsilon)$)-DP for $f(\mathcal{D})$. Finally, when we extract top-$k$ coordinates of $f(\mathcal{D})$ (Sec. 4.3), the privacy guarantee also holds according to the post-processing property [12]. Intuitively, the information about whether a sample occurs in the private set is protected, and the protection satisfies ($\varepsilon$, $\delta(\varepsilon)$)-DP.

## 5.3 Differential Privacy of Users' Secret Phrases

The above analysis shows that we can protect the privacy of each sample (i.e., one *occurrence* of a sentence). However, in practical text generations, we may want to protect all occurrences of some *secret phrases* specific to a user (e.g., names and addresses).[3] Consider a secret phrase $s$ occurring $n_s$ times ($n_s \geq 1$) in the private set. According to group privacy [12], the protection on phrase $s$ satisfies $(n\varepsilon, \frac{e^{n\varepsilon}-1}{e^{\varepsilon}-1}\delta)$-DP [20, 36], where the privacy loss scales linearly with the number of occurrences of $s$.

While naively applying a DP algorithm requires larger noise to protect user 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 $\sqrt{2}\tilde{n}_s$

[3] A formal definition of this is called personalized differential privacy, first seen in [14].

(See App. D for detailed deductions). In this way, the strength of privacy guarantee on secret phrases is similar 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 on user phrases by careful partitioning of the private data.

### 5.4 How does DP prevent memorization?

In practice, the privacy of the language model is sometimes 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. [5], which uses a language model to predict a secret phrase $s$ given a prefix. By the closure to post-processing [12], the prediction also satisfies DP. We denote $\mathcal{S}$ as the undesirable event where the model generates the phrase $s$ verbatim. The definition of DP implies that the probability of $\mathcal{S}$ to happen when $s$ is in the training data is at most $e^{\varepsilon}$ larger than the probability of an alternative model trained without $s$ in the data. The chances for the latter model to generate a sequence containing $s$ are astronomically small; thus DP implies that the probability of $\mathcal{S}$ under the former model needs to be just as small.

## 6 Experiments

### 6.1 Experimental Settings

**Datasets.** We evaluate our model on two datasets. AirDialog [44] consists of 1M utterances from customer service dialog on flight booking; Europarl_v6 consists of 2M English sentences collected from European Parliament.[4] (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, 20] and further trained on public set $\mathcal{D}^{\text{pub}}$ without protection; (2) based on **NoisySGD**, **NoisySGD+GC** [22] 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 $+\tilde{\mathcal{D}}^{\text{pub}}$.

**Hyper-parameters and implementation details.** All models are fine-tuned from the (public) pre-trained GPT-2 model [33]. The batch size is 32 for all comparing methods except the GC [22] (GC [22] requires 2048). We use Adam [21] and adjust the initial learning rate with a range of $10^{-3}$ to $10^{-6}$ for all methods. The $\delta$ 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 pseudo public data $\tilde{\mathcal{D}}^{\text{pub}}$ as a warm-up. The coefficient $\lambda$ 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 [5] to reduce the usage of the memory and hard-disk (See more details about the strategies and the computational cost in App. I). We can run SeqPATE with $2k$ teachers on a single GPU within 3 days. (See more hyperparameters in App. G.)

**Metrics.** We evaluate the generated text by perplexity (PPL) and Bleu (Bleu-n) [31].

### 6.2 Overall Performance

**Protection at the sample level.** Tab. 1 show the performance on the two datasets (See more metrics (i.e., Bleu-1 and Bleu-2) in App. K). 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

[4] www.statmt.org/europarl

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

| | | 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** |

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

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 $\varepsilon$.

**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).

| | | PPL ↓ | Bleu-3 ↑ | Bleu-4 ↑ |
|---|---|---|---|---|
| DP (phrase) $\varepsilon_{\text{avg}} = 3$ | NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ | 16.75 | 1.71 | 0.57 |
| | 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 |
| | SeqPATE | **8.06** | **6.10** | **3.90** |

Table 2: The performance on AirDialog with the protections of users' secret phrases (mentioned in Sec. 5.3). $\varepsilon_{\text{avg}}$ is the average $\varepsilon$ over all secret phrases, as $\varepsilon$ of each phrase varies with the phrase frequency and the teacher number (See App. F for detailed analyses about the $\varepsilon_{\text{avg}}$). All results of SeqPATE are statistically significant under t-test ($p < 0.05$).

As described in Sec. 5.3, standard DP methods incur larger privacy loss on secret phrases. In Tab. 2, we see that NoisySGD+GC+$\tilde{\mathcal{D}}^{\text{pub}}$ needs a large noise to achieve a satisfactory level of protection on phrases, because $\varepsilon$ increases linearly with the phrase frequency (group privacy [12]). 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 a better performance (+431% and +565% in Bleu4) (See details about the level of protection on users' secret phrases in App. F).

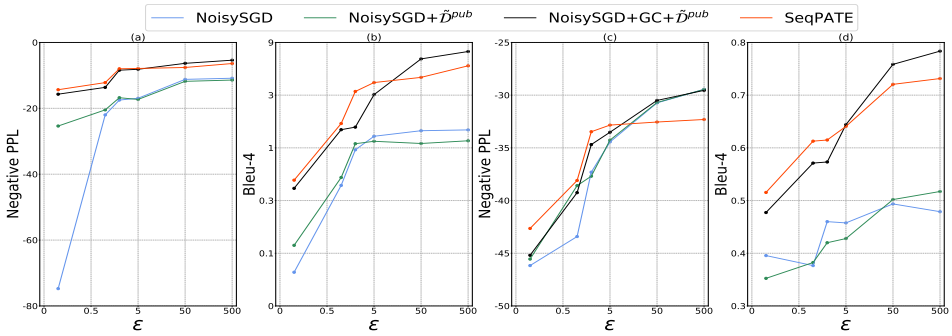


Figure 2: The private-utility tradeoff in Bleu-4 and negative PPL on a different $\varepsilon$. All the results are under sample level protections. Subfigure a & b show the results on AirDialog; subfigure c & d show the results on Europarl_v6. The grey lines show the "lower bound" performance since the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

**Privacy-utility tradeoff.** In Fig. 2, we show the private-utility tradeoff curve of all DP algorithms. [6] Typically, DP with $\varepsilon \in [0.1, 5]$ is considered to provide a meaningful protection [42]. 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 $\varepsilon$ approaches infinity (i.e. the noise approaches 0). However, SeqPATE with an infinite $\varepsilon$ 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.

| | 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 |

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

### 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 the 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 the text generation.

| | 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 |

Table 4: Analyses about the candidate filtering strategies.

### 6.4 Analyses on Candidate Filtering and Teacher Numbers

To analyze candidate filtering with different filtering strategies, we conduct the experiments on top-$p$ and the 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. The filtering with a too-small $k$ ($k = 1$ or $k = 10$) implies discarding too much useful information from the supervision ($k = 1$ is different to −KL in Tab. 3, which uses the Top-1 of teachers' results). Filtering with oversize $k$ results in unnecessarily large noises. The candidates

[6] For the models without protections, we consider $\varepsilon$ to be zero for baselines using the public data and $\varepsilon$ to be infinity for baselines using the private data.

with very small probabilities should be filtered during the 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 teacher numbers are in a range of $1 \sim 2k$. This is because the noise assigned to each teacher drops linearly as the teacher numbers go up. 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, 4, and App. H.

# 7 Related Work

Text generation models may leak user information through the generated texts [17, 6]. 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) [23] and the relationship between information and authors [27]. Some researchers [37, 38] infer the membership (whether samples from a given author are used to train the model) given a black-box model. Some papers protect users' privacy of training data against untrusted servers via federated learning [25, 9]. Another direction is to prevent attackers from extracting sensitive information in training sets by analyzing the outputs [28, 20], which is in urgent need since Carlini et al. [6] have achieved the extraction. Our SeqPATE focuses on this direction. In this direction, regularization methods [5, 40, 18] restrict the model capacity and avoid the model memorizing exact training samples. Anonymization methods [24, 41] detect sensitive text and replace it with non-sensitive texts. Unlike DP [12] methods, the above methods do not provide a quantifiable guarantee for privacy protection.

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. [26] propose a recurrent language model with a DP guarantee against the identification of the 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 users' phrases from being detected. Some researchers apply NoisySGD to text generation to prevent the sensitive training data (not the user-level) from being extracted: some of them [35, 36] employ DP to protect a part of selected tokens; others [20, 45, 22] 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. [20] apply NoisySGD [1] to text generation. Yu et al. [45] investigate fine-tuning strategies on pre-trained language models with NoisySGD. Li et al. [22] apply ghost clipping to pre-trained language models with NoisySGD and reduce memory usage.

Differential privacy (DP) [11, 12] formally defines and quantifies privacy. ML models with DP guarantee [43, 13, 47] prevents the existence of individual training examples from being detected [5]. Some researchers protect the privacy of empirical risk minimization classifiers [7] and SVM [34] with DP. Following Song et al. [39], NoisySGD [1] achieves DP on deep learning models by adding noises to gradients. Pichapati et al. [32] adaptively clip the gradient on NoisySGD. PATE [29, 30] transfers the knowledge from teacher models trained on private sets with noises to a student model. KNN-PATE [46] refines PATE by accessing only the k-nearest neighbors from the private set. Jordon et al. [19] adversarially learn to generate synthetic data with discriminators trained by PATE. Those methods are not customized for text generation models.

# 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 tradeoff 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 therefore is 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.

# 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 *International Conference on Machine Learning*, pages 394–403. PMLR, 2018.

[3] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, 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] 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.

[6] 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.

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

[8] 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.

[9] 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. *arXiv preprint arXiv:2201.11934*, 2022.

[10] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. In *Journal of the Royal Statistical Society: Series B*, 2019.

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

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

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

[14] Arpita Ghosh and Aaron Roth. Selling privacy at auction. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 199–208, 2011.

[15] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. In *International Journal of Computer Vision*, volume 129, pages 1789–1819, 2021.

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

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

[18] 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.

[19] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

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

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

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

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

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

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

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

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

[32] 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.

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

[34] Benjamin Rubinstein, Peter Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. In *Journal of Privacy and Confidentiality (JPC)*, volume 4, pages 65–100. Carnegie Mellon University, 2012.

[35] 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.

[36] Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. Selective differential privacy for language modeling. In *arXiv preprint arXiv:2108.12944*, 2021.

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

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

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

[40] 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.

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

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

[43] 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.

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

[45] 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 *arXiv preprint arXiv:2110.06500*, 2021.

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

[47] 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]