SubMix: Practical Private Prediction for Large-scale Language Models

Antonio A. Ginart[*]
Department of Electrical Engineering
Stanford University
tginart@stanford.edu

Laurens van der Maaten
Facebook AI Research
lvdmaaten@fb.com

James Zou
Department of Biomedical Data Science
Stanford University
jamesz@stanford.edu

Chuan Guo
Facebook AI Research
chuanguo@fb.com

Abstract

Recent data-extraction attacks have exposed that language models can memorize some training samples verbatim. This is a vulnerability that can compromise the privacy of the model's training data. In this work, we introduce SubMix: a practical protocol for private next-token prediction designed to prevent privacy violations by language models that were fine-tuned on a private corpus after pre-training on a public corpus. We show that SubMix limits the leakage of information that is unique to any individual user in the private corpus via a relaxation of group differentially private prediction. Importantly, SubMix admits a tight, data-dependent privacy accounting mechanism, which allows it to thwart existing data-extraction attacks while maintaining the utility of the language model. SubMix is the first protocol that maintains privacy even when publicly releasing tens of thousands of next-token predictions made by large transformer-based models such as GPT-2.

1 Introduction

The advent of transformers (Vaswani et al., 2017) has fostered a dramatic advancement in the capabilities of generative neural language models (LMs), enabling large-scale models such as GPT (Radford et al., 2019; Brown et al., 2020) to generate realistic, human-like text. Unfortunately, these impressive capabilities also come at a cost to privacy, as the amount of excess parameters in the LM enables it to memorize certain training samples. Consequently, several recent works have demonstrated practical training-data extraction attacks that reproduce entire sentences from the training dataset verbatim by querying the LM as an API (Carlini et al., 2019; 2020). These attacks expose the privacy risks of large-scale LMs, especially when their training data contains sensitive information such as addresses and personal ID numbers.

Existing solutions to data extraction attacks focus on using differential privacy (DP; Dwork et al. (2014)), which provably protects against privacy attacks (Yeom et al., 2018). Techniques such as DP-SGD (Abadi et al., 2016) have been applied to train differentially private neural networks on both vision and language tasks (McMahan et al., 2017; Papernot et al., 2018). However, the threat model in DP-SGD implicitly assumes that the adversary has full access to the private model's parameters and gradients during training, which results in pessimistic information leakage bounds that are unreasonable for most models. Indeed, existing work only performs DP-SGD training of small feedforward networks (Kerrigan et al., 2020) and RNNs (McMahan et al., 2017; Ramaswamy et al., 2020), often with an unsatisfactory privacy-utility trade-off. Training large machine-learning models with DP-SGD remains an open challenge (Jayaraman & Evans, 2019; Tramèr & Boneh, 2020).

Our study deviates from prior work by, instead, considering the problem of *private prediction* (Dwork & Feldman, 2018) using non-private language models fine-tuned on a private corpus. We propose SubMix, a novel private prediction mechanism for answering next-token queries. Focusing on private prediction affords SubMix three notable advantages: (1) Private prediction does not require modification of the training algorithm, which makes use of large-scale LMs feasible. (2) Private

[*] Work done while interning at Facebook AI Research.

prediction allows us to leverage the probabilistic nature of next-token sampling for highly efficient privacy accounting. (3) Private prediction allows us to leverage public pre-trained LMs[1] to obtain private predictive distributions that do not require noise addition to privatize the model's predictions.

SubMix utilizes an ensemble of LMs fine-tuned on disjoint parts of the private corpus and privatizes predictions by mixing the next-token distribution with that of a public pre-trained LM. The mixing weight is adaptively tuned based on the degree of consensus among models in the ensemble. If all models predict the same next-token distribution, then it is impossible for the next token to leak sensitive information about any unique individual so no mixing is required. By contrast, if models in the ensemble have high disagreement, SubMix will mix predictions with those of the public pre-trained model to minimize privacy leakage. This allows SubMix to perform accurate next-token prediction for most queries while preserving the privacy of the private corpus.

For any sequence of next-token queries issued to SubMix, we measure the amount of privacy leakage in the response using Rényi divergence (Rényi, 1961). Our privacy notion, which we refer to as *operational privacy*, is a sufficient condition for preventing samples that are unique to any user from being generated by the SubMix mechanism. Importantly, operational privacy allows us to perform tight data-dependent privacy accounting to upper bound the privacy loss of SubMix when answering a variable-length query sequence. Concretely, when answering up to $1,024$ next-token queries, SubMix realizes nearly 75% of the perplexity improvement that non-private fine-tuning would have achieved on GPT-2 models (Radford et al., 2019), with privacy leakage as small as $\epsilon = 2$. We also show that SubMix can effectively prevent existing data extraction attacks against GPT-2.

2 Problem Formulation

We begin by setting up the problem of private next-token prediction and reviewing existing literature on differential privacy. We then define and discuss the notion of operational privacy for SubMix.

2.1 Preliminaries

Let Σ denote a fixed finite vocabulary set. We use lower-case letters to denote single tokens (such as $x \in \Sigma$) and use bold font to denote contexts or strings of tokens (such as $\mathbf{x} \in \Sigma^*$). A (causal) language model h is a mapping from *context strings* to a distribution over next tokens: $h : \Sigma^* \to \mathbf{\Delta}^{|\Sigma|}$, where $\mathbf{\Delta}^{|\Sigma|}$ is the $|\Sigma|$-dimensional probability simplex. For a particular context $\mathbf{x} \in \Sigma^*$, let $h(\mathbf{x})$ denote the next-token distribution vector obtained from evaluating h on context $\mathbf{x}$, and let $h(z|\mathbf{x}) \in [0, 1]$ denote the probability mass on a token $z \in \Sigma$.

User-level Corpus Let $\mathcal{D}$ denote a dataset of unstructured text, which is a set of token sequences $\mathbf{x} \in \Sigma^*$. We assume that $\mathcal{D}$ is generated by a set of n distinct users, each holding a subset $\mathcal{D}_i$ of the full dataset $\mathcal{D}$, *i.e.*, $\mathcal{D} = \bigcup_{i=1}^n \mathcal{D}_i$ with $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ for $i \neq j$. We refer to each $\mathcal{D}_i$ as a *user-level corpus* for user i. As a concrete example, in the context of social media posts, $\mathcal{D}_i$ would contain *all* of the non-public posts made by user i. We aim to provide privacy guarantees for a model that is non-privately fine-tuned on the dataset $\mathcal{D}$.

Next-token Prediction One popular use case for language models is to perform *next-token prediction*, that is, return a token z when queried with a context $\mathbf{x}$. Such a query-answering API is useful for applications such as smart keyboard for auto-correction and text completion (Mirowski & Vlachos, 2015; Hertel, 2019). Typical approaches for next-token prediction involve sampling z from the next-token distribution vector $h(\mathbf{x})$; see Holtzman et al. (2019). Large transformers trained on unstructured internet text have achieved remarkable success for this task, producing natural-looking sentences via sequentially generating next tokens from a given prompt (Brown et al., 2020).

Text Extraction Attacks Recent studies have shown that it is possible for next-token prediction APIs to reveal sensitive private information contained in the training dataset. Carlini et al. (2019) defined *κ-eidetic memorization* to formalize the notion that extraction of strings that are uncommon in the corpus can lead to violations of user privacy.

[1] We do not provide privacy guarantees or text extraction protection for the public corpus on which the LMs are pre-trained; our privacy guarantees only on apply to the private corpus on which the LM is fine-tuned.

Definition 2.1 (κ-eidetic memorization (Carlini et al., 2019))**.** A string s is κ-eidetic memorized by an LM h if s is *extractable*[2] from h and s appears in at most κ examples in the training data $\mathcal{D}$.

Carlini et al. (2019) showed that if the training dataset contains token sequences of the form: "`My social security number is □□□-□□-□□□□`" where □ represents a digit of a user's social security number (SSN), then it is subtantially more likely for the LM trained on $\mathcal{D}$ to generate the exact SSN appearing in $\mathcal{D}$ compared to a random SSN. As a result, it is possible to design an efficient *extraction attack* that reproduces such unique sequences in the training dataset. Carlini et al. (2020) further extended this attack to large transformer-based LMs such as GPT2 (Radford et al., 2019), extracting memorized personal information such as name and address contained in the model's training dataset. Motivated by these shortcomings, this paper studies notions of privacy that can prevent such text-extraction attacks while preserving the model's utility.

2.2 Differential Privacy

Differential privacy (Dwork et al., 2014) is a powerful mathematical framework for privacy-preserving data analysis. The underlying principle in differential privacy and all its variants is the notion of *indistinguishability*. Informally, a mechanism $\mathcal{M}$ is private if, given two adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, the mechanism's outputs $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$ are approximately indistinguishable. Hence by observing the output of $\mathcal{M}$, it is difficult for an adversary to discern the difference between $\mathcal{D}$ and $\mathcal{D}'$. The above informal definition of privacy can be made mathematically precise by specifying: (1) the notion of adjacency between datasets $\mathcal{D}$ and $\mathcal{D}'$, and (2) the notion of approximate indistinguishability.

Differentially Private Training Prior work on private LM training (McMahan et al., 2017; Ramaswamy et al., 2020) adopted the definition of *user-level adjacency*: $\mathcal{D}$ and $\mathcal{D}'$ are adjacent if they differ in a single user's data. Approximate indistinguishability is defined in terms of divergences and is applied to the trained model: The private training algorithm $\mathcal{M}(\mathcal{D})$ induces a distribution over models, and indistinguishability requires that $D(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}')) < \epsilon$ for some divergence D and small constant $\epsilon > 0$. Popular choices include the *max divergence* (Dwork et al., 2014) and the *Rényi divergence of order* α (Rényi, 1961):

$$D_\infty(P||Q) = \sup_{x \in \text{supp}(Q)} \log P(x) - \log Q(x), \qquad D_\alpha(P||Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \left[P(x)/Q(x) \right]^\alpha .$$

Specializing to the choice of Rényi divergence, we define *user-level Rényi differential privacy* (RDP; Mironov (2017)) for private training as follows.

Definition 2.2 (User-level RDP for private training)**.** For $\alpha > 1$, let D_α denote the order-α Rényi divergence. A private training algorithm $\mathcal{M}$ is an (α, ϵ)-RDP mechanism if for any $\mathcal{D}$ and $\mathcal{D}'$ that differ in only one user's data $\mathcal{D}_i$, we have $D_\alpha(\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}')) \leq \epsilon$.

In order to satisfy the criteria in Definition 2.2 for neural language models, the standard approach is to use DP-SGD (Abadi et al., 2016) to inject noise into the gradients computed at every iteration of SGD training, and use composition theorems to bound the total privacy leakage across iterations.

Differentially Private Prediction Private prediction differs from private training in that the notion of approximate indistinguishability applies to a sequence of predictions made by a *private prediction protocol* $\mathcal{P}$, rather than to a privately trained model. Formally, at each time step t, an adversary Adv (potentially adaptively) issues a context string $\mathbf{x}_t$, and the private prediction protocol $\mathcal{P}$ responds by generating a next token $y_t \in \Sigma$. We let $\mathcal{P} \leftrightharpoons_T \mathsf{Adv}$ denote the sequence of query-response pairs between P and Adv up until time T: $\mathcal{P} \leftrightharpoons_T \mathsf{Adv} = \{\mathbf{x}_t, y_t\}_{t=1}^T$. For a query sequence of length T, approximate indistinguishability requires that for adjacent datasets $\mathcal{D}, \mathcal{D}'$:

$$D\left(\mathcal{P}(\mathcal{D}) \underset{T}{\leftrightharpoons} \mathsf{Adv} \;||\; \mathcal{P}(\mathcal{D}') \underset{T}{\leftrightharpoons} \mathsf{Adv}\right) \leq \epsilon, \tag{1}$$

for some divergence D and $\epsilon > 0$. We summarize the above discussion in the following Rényi-DP variant of the definition for private prediction by Dwork & Feldman (2018).

[2] Carlini et al. (2019) define *text extraction* informally. In the supplement, we formalize it within the framework of statistical hypothesis testing and show that differential privacy is sufficient to prevent eidetic memorization.

Definition 2.3 (User-level RDP for private prediction)**.** Let $\alpha > 1$, $\epsilon > 0$, and $T \in \mathbb{Z}_+$. A prediction protocol $\mathcal{P}$ is (α, ϵ, T)-RDP if for any adversary Adv and any $\mathcal{D}$ and $\mathcal{D}'$ that differ in only one user's data $\mathcal{D}_i$, we have that Equation 1 holds.

It is well-known that differentially private models can be used for private prediction via the *post-processing theorem* (Mironov, 2017): If $h \leftarrow \mathcal{M}(\mathcal{D})$ is a model obtained from an (α, ϵ)-RDP training mechanism $\mathcal{M}$, then $\mathcal{M}'(\mathbf{x}; \mathcal{D}) = h(\mathbf{x})$ is an $(\alpha, \epsilon, \infty)$-RDP private prediction mechanism for any sequence of queries (regardless of length). However, DP-SGD (Abadi et al., 2016)—the primary mechanism for training private neural networks—makes an implicit assumption that the adversary also observes additional information that is not accessible if h is used as a prediction API, and in practice, it causes the accounting mechanism in DP-SGD to vastly overestimate the privacy leakage parameter ϵ (Nasr et al., 2021). One alternative is the general-purpose *subsample-and-aggregate* mechanism, which adds noise to an ensemble's output in order to privatize it. This results in a trade-off between the information leakage, ϵ, and the number of queries that can be answered, T (van der Maaten & Hannun, 2020). For smaller T, the mechanism needs less noise to achieve a particular ϵ. Conceptually, this is a step in the right direction, but the added noise greatly reduces utility and is superfluous if we can leverage pre-trained public LMs to privatize the predictive distribution.

2.3 Operational Privacy for Private Prediction

To remedy the problems in user-level differentially private training and prediction, we propose a different notion of privacy that is sufficient for preventing text extraction attacks, but admits more specialized privacy mechanisms with tighter privacy accounting.

Let $\mathbb{P}(\mathcal{D})$ denote the power set of $\mathcal{D}$. A *partition* $\Pi \in \mathbb{P}(\mathcal{D})$ of $\mathcal{D}$ is a collection of sets π that satisfies $\bigcup_{\pi \in \Pi} \pi = \mathcal{D}$ and that satisfies $\pi \cap \pi' = \emptyset$ for distinct $\pi, \pi' \in \Pi$. We refer to the elements of Π as *parts*. For some fixed ordering, we let Π_i denote the i-th part. As a minor abuse of notation, we let $\mathcal{D} \setminus \pi$ denote the usual element-wise subtraction and write $\Pi \setminus \pi$ instead of $\Pi \setminus \{\pi\}$ for brevity. Recall the notion of the private prediction protocol $\mathcal{P}$. We augment the protocol $\mathcal{P}$ with the capability to terminate the query-response sequence at any time. With a slight abuse of notation, we denote by $T(\mathcal{P})$ the sequence length produced by $\mathcal{P}$. We define ***Rényi operational privacy*** (ROP) as follows.

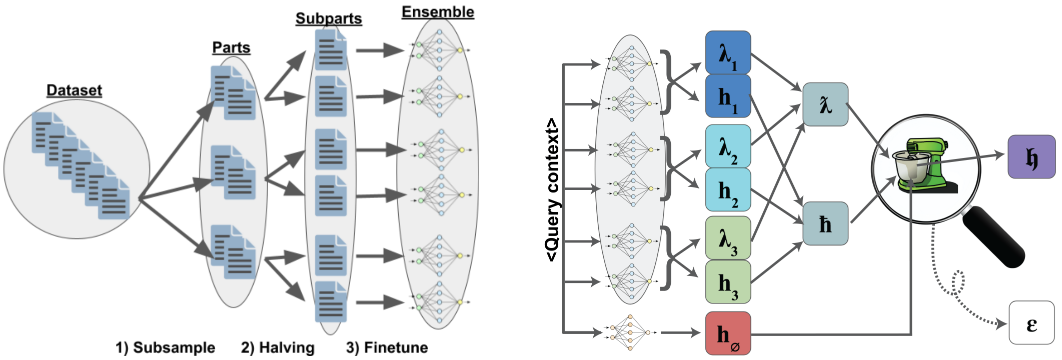
Definition 2.4 (Rényi operational privacy for private prediction)**.** Let $\mathcal{D}$ be a dataset of user-level corpora and let Π be a partition so that each user-level corpus $\mathcal{D}_i$ is contained in some part $\Pi_j \in \Pi$. For $\alpha > 1$ and $\epsilon > 0$, a prediction protocol $\mathcal{P}$ is (α, ϵ)-ROP for partition Π of dataset $\mathcal{D}$ if for any part $\Pi_i \in \Pi$ and adversary Adv, we have:

$$D_\alpha^{\text{sym}} \left(\mathcal{P}(\Pi) \underset{T(\mathcal{P})}{\leftrightharpoons} \mathsf{Adv} \;||\; \mathcal{P}(\Pi \setminus \Pi_i) \underset{T(\mathcal{P})}{\leftrightharpoons} \mathsf{Adv} \right) \leq \epsilon.$$

Where $D_\alpha^{\text{sym}}(P||Q) = \max\{D_\alpha(P||Q), D_\alpha(Q||P)\}$. ROP differs from user-level RDP in Definition 2.3 in two aspects:

1. We substitute user-level adjacency with partition-level adjacency. By definition, the partition is constructed so that any user's data belongs to a single part. Readers familiar with *group privacy* (Dwork et al., 2014) may recognize partition-level adjacency as a formal relaxation of the group-level adjacency used in Rényi group differential privacy. Partition-level RDP is neither strictly weaker nor stronger than user-level RDP, and, at a cost, conversion is possible (Appendix C).
2. We allow *variable-length* query-response sequences by enabling the private-prediction protocol $\mathcal{P}$ to terminate[3] at will. ROP accounts for the privacy leakage in the responses made throughout the prediction protocol's operation lifetime, which is why we refer to it as *operational*. By allowing for a variable-length sequence, we provide the mechanism with additional flexibility without increasing susceptibility to text extraction. Non-sensitive queries often can be answered without much privacy leakage, whereas sensitive queries may quickly exhaust the privacy budget, causing the protocol to terminate early. The protocol's decision to terminate at time $T(\mathcal{P})$ may leak some information about how sensitive the queried contexts are. However, this leakage is relatively insignificant and can be upper bounded (allowing us to convert variable-length ϵ into fixed-length; see Appendix C).

[3] After termination, the mechanism can technically continue to issue responses, but only in a way that is entirely independent of the private corpus, *e.g.*, by using a public pre-trained LM.



(a) **Training.** The corpus is a dataset comprised of private user text. Each document represents all of the text corresponding to a particular user. At training time, SUBMIX learns an ensemble by: (1) subsampling the dataset into non-overlapping parts, (2) halving each part into two subparts, and (3) fine-tuning the LM on each subpart using $\mathcal{L}$.

(b) **Prediction.** The bottom-most network in the figure represents the pre-trained public model; the other networks form the ensemble of model pairs obtained after SUBMIX training. SUBMIX prediction produces a mixing weight for each model. These weights are aggregated and used to mix the ensemble predictions with the predictions of the public model.

Figure 1: Overview of SUBMIX's training protocol (**left**) and prediction protocol (**right**).

3 SUBMIX

We introduce SUBMIX, a private next-token prediction protocol that satisfies the operational privacy definition introduced above (Figure 1). SUBMIX follows the design of the subsample-and-aggregate mechanism by first forming a random partition Π of the training dataset, with each user's data belonging to a single random part Π_i. For each part Π_i, the protocol further splits Π_i into two *subparts* π_i and π_i' by randomly assigning users in Π_i to the two halves.

Fine-tuning a Pre-trained Model Language models are often first trained on vast internet crawls to develop a general understanding of human language, and then fine-tuned on a more domain-specific dataset for the target task (Dai & Le, 2015; Howard & Ruder, 2018). We treat language model training as a black-box operation, and denote the training routine $\mathcal{L}(\cdot)$ as a function that returns a model $h_\mathcal{D} = \mathcal{L}(\mathcal{D})$. We assume access to a LM pre-trained on public data, and use routine $\mathcal{L}$ to fine-tune the LM on the private user-level corpora. Specifically, Algorithm 1 fine-tunes a public pre-trained LM on each subpart π_i and π_i' to produce LMs $h_{\pi_i} = \mathcal{L}(\pi_i)$ and $h_{\pi_i'} = \mathcal{L}(\pi_i')$. By convention, fine-tuning on the empty set returns the public pre-trained model: $h_\emptyset = \mathcal{L}(\emptyset)$.

Algorithm 1 SUBMIX Training.

```
Inputs: User-level private corpus $\mathcal{D}$, LM fine-tuning routine $\mathcal{L}$
Outputs: Fine-tuned LMs $h_{\pi_i}, h_{\pi_i'}$ for $i = 1, \ldots, k$
Hyperparameters: # of parts $k$
$\Pi \leftarrow$ Random $k$-fold partition of $\mathcal{D}$ with $|\Pi_i| = |\mathcal{D}|/k$
for $i \in \{1, \ldots, k\}$ do
    $(\pi_i, \pi_i') \leftarrow$ Randomly split part $\Pi_i$ into two subparts.
    $h_{\pi_i} \leftarrow \mathcal{L}(\pi_i)$, $h_{\pi_i'} \leftarrow \mathcal{L}(\pi_i')$
end for
```

Next-token Distribution Given a query context $\mathbf{x}_t$, each part Π_i is responsible for producing a next-token probability mass function (pmf) by combining $h_{\pi_i}(\mathbf{x}_t)$ and $h_{\pi_i'}(\mathbf{x}_t)$ into $\bar{h}_i(\mathbf{x}_t) = (h_{\pi_i}(\mathbf{x}_t) + h_{\pi_i'}(\mathbf{x}_t))/2$. SUBMIX mixes this pmf with the public pre-trained model $h_\emptyset$ to add noise to the prediction that hides private information. It does so by computing:

$$h_i(\mathbf{x}_t) = \lambda^* \bar{h}_i(\mathbf{x}_t) + (1 - \lambda^*) h_\emptyset(\mathbf{x}_t),$$

for a suitable choice of the mixing weight λ^*. A value of $\lambda^* = 0$ means the fine-tuned LMs h_{π_i} and $h_{\pi_i'}$ are not used (no privacy loss), and $\lambda^* = 1$ means no noise was added (no utility loss). We select λ^* based on how much information about the part Π_i is contained in the pmfs $h_{\pi_i}(\mathbf{x}_t)$ and $h_{\pi_i'}(\mathbf{x}_t)$.

Intuitively, since both π_i and π_i' are random samples from the same data distribution, if the models h_{π_i} and $h_{\pi_i'}$ did not memorize the query context $\mathbf{x}_t$ then $h_{\pi_i}(\mathbf{x}_t)$ and $h_{\pi_i'}(\mathbf{x}_t)$ will be similar. Hence, the selected value of λ^* should be close to 1. If either h_{π_i} or $h_{\pi_i'}$ memorized the context $\mathbf{x}_t$, then

Algorithm 2 SUBMIX Prediction.

Inputs: Fine-tuned LMs $h_{\pi_i}, h_{\pi'_i}$ for $i = 1, \ldots, k$, privacy parameters ϵ, time step t, query context $\mathbf{x}_t \in \Sigma^*$
Outputs: Next token response $y_t \in \Sigma$
Hyperparameters: Rényi divergence order α, target leakage β

```
 1: if $t = 1$ then
 2:     $\varepsilon_i \leftarrow \epsilon$ for $i = 1, \ldots, k$
 3: else if STOP has been issued then
 4:     return $y_t \sim h_\emptyset(\mathbf{x}_t)$
 5: end if
 6: for $i = 1, \ldots, k$ do
 7:     $\bar{h}_i(\mathbf{x}_t) \leftarrow \frac{1}{2}(h_{\pi_i}(\mathbf{x}_t) + h_{\pi'_i}(\mathbf{x}_t))$
 8:     Compute $\lambda_i$ using Equation 2.
 9: end for
10: $\lambda^* \leftarrow \frac{1}{k}\sum_{i=1}^{k}\lambda_i$
11: $\bar{h}(\mathbf{x}_t) \leftarrow \frac{1}{k}\sum_{i=1}^{k}\bar{h}_i(\mathbf{x}_t)$
12: $h(\mathbf{x}_t) \leftarrow \lambda^*\bar{h}(\mathbf{x}_t) + (1-\lambda^*)h_\emptyset(\mathbf{x}_t)$
13: for $i = 1, \ldots, k$ do
14:     $\lambda^*_{-i} \leftarrow \frac{1}{k-1}\sum_{j \neq i}\lambda_j$
15:     $\bar{h}_{-i} \leftarrow \frac{1}{k-1}\sum_{j \neq i}\bar{h}_j(\mathbf{x}_t)$
16:     $\mathfrak{h} \leftarrow \lambda^*\bar{h}(\mathbf{x}_t) + (1-\lambda^*)h_\emptyset(\mathbf{x}_t)$
17:     $\mathfrak{h}' \leftarrow \lambda^*_{-i}\bar{h}_{-i}(\mathbf{x}_t) + (1-\lambda^*_{-i})h_\emptyset(\mathbf{x}_t))$
18:     $\varepsilon_i \leftarrow \varepsilon_i - \max\{D_\alpha(\mathfrak{h}||\mathfrak{h}'), D_\alpha(\mathfrak{h}'||\mathfrak{h})\}$
19: end for
20: if $\forall i$: $\varepsilon_i > 0$ then
21:     $y_t \sim h(\mathbf{x}_t)$
22: else
23:     Issue STOP signal.
24:     $y_t \sim h_\emptyset(\mathbf{x}_t)$
25: end if
26: return $y_t$
```

$h_{\pi_i}(\mathbf{x}_t)$ and $h_{\pi'_i}(\mathbf{x}_t)$ are dissimilar as π_i and π'_i have no users in common. This suggests that mixing with the pre-trained LM $h_\emptyset$ is necessary for hiding the sensitive information in Π_i, so λ^* should be close to 0. SUBMIX balances between these two extremes by computing a separate λ_i for each part Π_i. Specifically, it sets a target privacy leakage $\beta > 0$ and optimizes:

$$\lambda_i \leftarrow \max_{\lambda \in [0,1]} \{\lambda : \mathsf{D}_i(\mathbf{x}_t, \lambda) \leq \beta\}, \tag{2}$$

where $\mathsf{D}_i(\mathbf{x}_t, \lambda) = D_\alpha\left(\lambda h_{\pi_i}(\mathbf{x}_t) + (1-\lambda)h_\emptyset(\mathbf{x}_t) \;||\; \lambda h_{\pi'_i}(\mathbf{x}_t) + (1-\lambda)h_\emptyset(\mathbf{x}_t)\right)$. The final value of λ^* is obtained by averaging the λ_i values for $i = 1, \ldots, k$, where k is the number of parts.

Prediction and Privacy Accounting Given the next-token pmfs $h_i(\mathbf{x}_t)$ for $i = 1, \ldots, k$, SUBMIX computes the ensemble pmf, $h(\mathbf{x}_t) = 1/k \sum_{i=1}^{k} h_i(\mathbf{x}_t)$, and samples from it to obtain a next-token prediction. Our mechanism for selecting the mixing weight λ^* can be shown to limit the privacy loss of a sample from $h(\mathbf{x}_t)$ under the operational privacy notion: Since each λ_i is determined entirely by the part Π_i, the next-token pmf after removal of Π_i can be derived in closed form. This allows us to compute the Rényi divergence in $h(\mathbf{x}_t)$ for adjacent datasets $\Pi \setminus \Pi_i$. We present the SUBMIX prediction protocol in Algorithm 2, and give its formal privacy analysis in the following proposition.

Proposition 3.1. SUBMIX *is an* (α, ϵ)*-ROP prediction mechanism.*

Proof. We will use the adaptive sequential composition theorem for RDP filters (Feldman & Zrnic, 2021, Theorem 4.3). Lines 20-25 ensure that for all parts $i = 1, \ldots, k$, at stopping time $T(\mathcal{P})$, the sequence of query responses $y_1, \ldots, y_{T(\mathcal{P})-1}$ satisfies:

$$\sum_{t=1}^{T(\mathcal{P})-1} D_\alpha^{\text{sym}}\left(y_t \sim \mathsf{P}(\Pi) \;||\; y_t \sim \mathcal{P}(\Pi \setminus \Pi_i)\right) \leq \sum_{t=1}^{T(\mathcal{P})-1} \varepsilon_i(t) - \varepsilon_i(t+1),$$

where $\varepsilon_i(t)$ is the remaining privacy budget at the start of time t. Then by the RDP filter:

$$\begin{aligned}\max_i D_\alpha^{\text{sym}}\left(\mathsf{P}(\Pi) \leftrightharpoons \mathsf{Adv} \;||\; \mathsf{P}(\Pi \setminus \Pi_i) \leftrightharpoons \mathsf{Adv}\right) &\leq \max_i \sum_{t=1}^{T(\mathcal{P})-1} D_\alpha^{\text{sym}}\left(y_t \sim \mathsf{P}(\Pi) \;||\; y_t \sim \mathsf{P}(\Pi \setminus \Pi_i)\right) \\ &\leq \max_i \sum_{t=1}^{T(\mathcal{P})-1} \varepsilon_i(t) - \varepsilon_i(t+1) \leq \max_i \epsilon = \epsilon.\end{aligned}$$

To conclude the analysis, note that after SUBMIX issues the STOP signal, any subsequent queries are answered by $h_\emptyset$. Since $h_\emptyset$ is not a function of Π, this does not leak any additional information. □

4 Experiments

Datasets We evaluate SubMix by fine-tuning the pre-trained GPT-2 model from HuggingFace (Wolf et al., 2019) on two "private" datasets: (1) `Wikitext-103` (Merity et al., 2016), a collection of 103 million tokens scraped from Wikipedia; and (2) `BigPatent-G` (Sharma et al., 2019), a collection of over 200,000 patents. We split the `wikitext-103` corpus into blocks of length 512 tokens and define each block as a (synthetic) user $\mathcal{D}_i$. We split `BigPatent-G` by patent and define each user to be a single patent. This setup mimics settings in which users have distinct data distributions within the text corpus.

Fine-Tuning & Evaluation We use standard hyperparameters for fine-tuning; see Appendix B for details. To assess the quality of LM predictions, we measure predictive perplexity:

$$\mathbf{PP}_h = \mathbb{E}_{\mathbf{x}=x_1 \cdots x_L \sim \mathcal{D}_{\text{heldout}}} \left[\exp\left(-\frac{1}{L} \sum_{i=1}^{L} \log h(x_i | x_1 \cdots x_{i-1}) \right) \right],$$

where $\mathcal{D}_{\text{heldout}}$ is the private held-out set and $h(\cdot | x_1 \cdots x_{i-1})$ denotes the pmf for the next token given context $x_1 \cdots x_{i-1}$. In practice, we truncate the context window to a fixed maximum length $L = 512$. Since each held-out sample consists of a block of $L = 512$ tokens, computing $\mathbf{PP}_h$ for a single sample requires L queries in total. Hence, the total number of queries B is a multiple of 512.

Following Geumlek et al. (2017), we report privacy loss in terms of α-Rényi divergence. Note that we can convert from (α, ϵ)-RDP to (ϵ', δ)-DP via $\epsilon' = \epsilon + \frac{\log(1/\delta)}{\alpha - 1}$ (Mironov, 2017). In the paper, we measure ϵ using $\alpha = 2$ Rényi divergence; we present results for other values of α in Appendix A.

4.1 Privacy-Utility Trade-off

Baseline Comparisons We first compare SubMix to three privacy-preserving mechanisms as baselines: (1) DP-SGD (Abadi et al., 2016) for private training; and (2) subsample-and-aggregate (S&A, Dwork et al. (2014)) and (3) GNMax (Papernot et al., 2018) for private prediction. For DP-SGD, concurrent work (Li et al., 2021) proposed training with very large batch size and fixed number of updates to significantly improve privacy-utility trade-off. The drawback of this method is that it requires taking multiple passes of the data in order to reap the benefits (whereas only one pass is needed for SubMix and non-private training). See Appendix B for details on adapting these mechanisms for private next-token prediction. Figure 2 shows the predictive perplexity of the private mechanisms on the `Wikitext-103` dataset for ROP/RDP[4] parameters $\alpha = 2$ and $\epsilon = 2$. The pre-trained GPT-2 model has a perplexity of 37.5 on `Wikitext-103` and is trivially private on that corpus. Fine-tuning the LM non-privately achieves a perplexity of 20.0. SubMix achieves a perplexity of 26.9 at $B = 1,024$ queries, which is substantially below the perplexity of the pre-trained LM. By contrast, the other mechanisms do not improve over the pre-trained baseline, even for a *single* query ($B = 1$). subsection A.1 presents more detailed results on the privacy-utility trade-off of the baseline methods: all of them require extremely large ϵ to improve over the pre-trained baseline, with DP-SGD outperforming the private prediction baselines (S&A and GNMax).

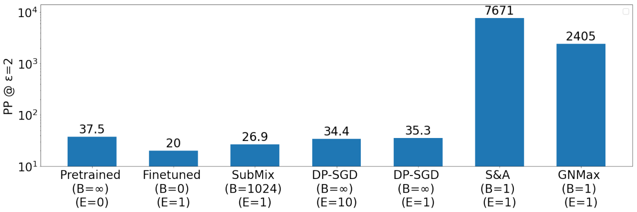


Figure 2: Perplexity for $\epsilon = 2$, $\alpha = 2$ of four privacy-preserving mechanisms (SubMix, DP-SGD, S&A, and GNMax) using GPT-2 on `Wikitext-103` with varying query budgets B. The number of epochs of training is denoted with E. For DP-SGD, we report results for both one epoch ($E = 1$) and ten epochs ($E = 10$). Other than DP-SGD ($E = 10$), all other fine-tuning methods only require one pass of training. Non-privately fine-tuning achieves a perplexity of 20.0 and the pre-trained public model achieves a perplexity of 37.5 . Lower perplexity is better.

Varying the Privacy Loss In Figure 3, we show the trade-off between perplexity and ROP privacy loss, ϵ, of SubMix at $\alpha = 2$. On both `Wikitext-103` (left plot) and `BigPatent-G` (right plot),

[4] Privacy loss is measured under RDP for DP-SGD, S&A, and GNMax; and under ROP for SubMix.

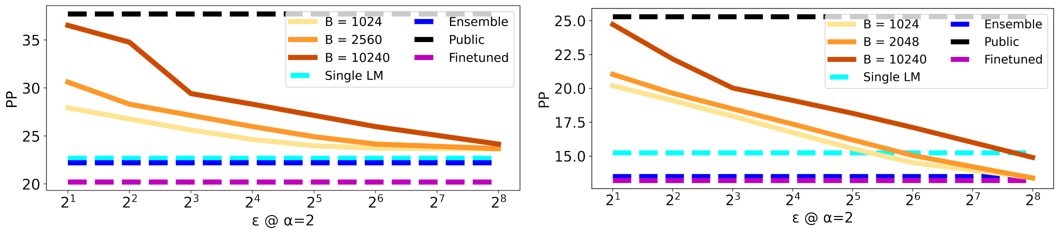


Figure 3: Perplexity of SUBMIX ($k = 8$) on `Wikitext-103` (**left**) and `BigPatent-G` (**right**) as a function of ROP privacy loss ϵ for three query budget values B. The perplexity of the pre-trained model and (non-private) single model, ensemble, and fully fine-tuned models are shown for reference.

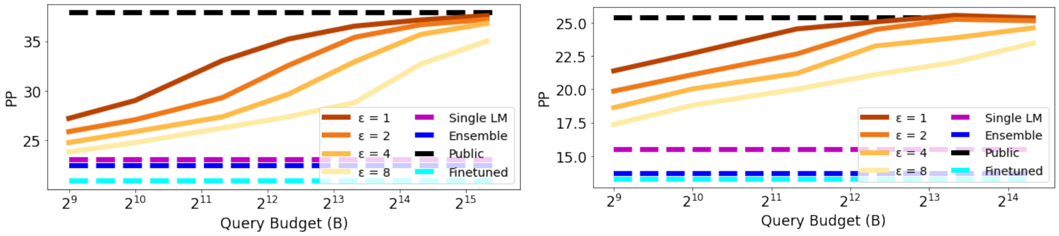


Figure 4: Perplexity of SUBMIX ($k = 8$) on `Wikitext-103` (**left**) and `BigPatent-G` (**right**) as a function of query budget B for different ROP privacy losses ϵ. The perplexity of the pre-trained model and (non-private) single model, ensemble, and fully fine-tuned models are shown for reference.

SUBMIX substantially improves over the pre-trained GPT-2 model, even when the query budget is increased to $B = 10,240$ queries. As expected, SUBMIX matches the perplexity of non-private LM at higher values of ϵ. Interestingly, SUBMIX's perplexity is even lower than that of a single non-privately fine-tuned LM at high ϵ. We surmise this is due to the performance gap between a single fine-tuned LM and an LM ensemble. The effect is less pronounced on `Wikitext-103` because that corpus was split into users by block, as a result of which many LMs contain text blocks from the same Wikipedia article. This reduces the positive effects of ensembling on predictive perplexity.

Varying the Query Budget Figure 4 shows the trade-off between perplexity and the number of queries B. The results in the figure were obtained by tuning the target leakage to obtain the desired budget. As expected, answering more queries using SUBMIX increases the average perplexity for each next-token query at a given ϵ. However, SUBMIX attains a surprisingly low perplexity for a moderate number of queries (*e.g.*, $B = 2^{10}$) at all ϵ values on both `Wikitext-103` and `BigPatent-G`.

Varying the Number of Parts A key hyperparameter of interest in SUBMIX is the size of the partition Π. Intuitively, a smaller number of partitions, allows each part to train a better quality LM at the cost of a larger Rényi divergence when a part is removed. Figure 5 shows the trade-off between perplexity and ROP privacy loss, ϵ, for varying partition sizes, k. We observe the key trend that as ϵ decreases, perplexity increases more rapidly for smaller k because the privacy budget is exhausted more quickly when each part has a greater relative contribution to the ensemble. The optimal value for k is generally around 16 for all ϵ values of interest, although this may depend on the data distribution and design choices such as model architecture and training hyperparameters.

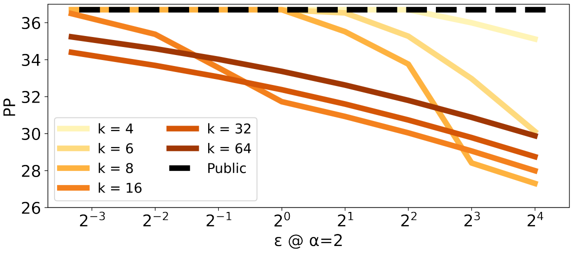


Figure 5: Perplexity of SUBMIX on `Wikitext-103` as a function of ROP privacy loss ϵ for different partition sizes k.

4.2 Text Extraction Attacks

To empirically validate that SubMix prevents text extraction attacks, we perform a random code text-extraction experiment in the style of Ramaswamy et al. (2020); Shi et al. (2021). We randomly generate m codes with each code being an ℓ-digit number (for example, representing a user's age, ZIP-code, phone number, SSN, *etc.*). The fine-tuning dataset is constructed so that each user's text is single sentence: $\mathcal{D}_i =$ “My number is: <random ℓ-digit number here>”. We then fine-tune on this dataset and make private predictions using SubMix for the query context “My number is:” to test whether SubMix prevents text extraction (and if so, at what ϵ). As a baseline, we fine-tune GPT-2 on this dataset for 1000 iterations. This results in the LM memorizing all ℓ codes, achieving a perplexity of less than 0.5 and $\geq 90\%$ recall when prompted with context. We apply SubMix with $k = {}^{m}/_{2}$ parts so that each model in the ensemble strongly memorizes one number, achieving near 100% recall when prompted with the context.

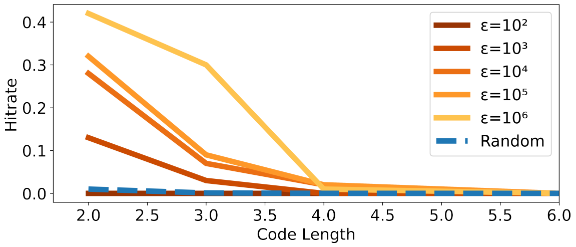


Figure 6: Hit rate of text extraction attacks on SubMix for varying lengths of code. The # of parts is $k = 3$ and the # of codes generated is $g = 100$. The non-privately fine-tuned LM has a hit rate of ≥ 0.9 for all lengths.

For the text extract attack, the figure-of-merit is the *hit rate* of the g generations, defined as the number of generated codes that exactly match a secret code divided by g. Figure 6 shows the hit rate of the text extraction attack. For all code lengths $\ell = 2, \ldots, 5$, SubMix succeeds in preventing the attack at $\epsilon = 10^2$. For longer code lengths, even higher values of ϵ suffice for preventing this random-sampling text extraction. Intuitively, extraction becomes more difficult as the code lengths increase. This experiment shows that the mechanism for limiting the release of sensitive information via solving Equation 2 is effective, and the privacy accounting in SubMix meaningfully measures the amount of privacy loss. We also ran these attacks for varying values of K and m, and made the same qualitative observations. In addition, we performed experiments in which we varied $g \in \{10, 100, 1000\}$. We found that this does not affect SubMix's hit rate.

5 Discussion & Related Work

Related Work McMahan et al. (2017) was the first to study the training of differentially private language models by using DP-SGD to train a small recurrent neural network. However, the resulting LMs have far fewer parameters than modern transformers and attain much higher perplexities. More recently, Shi et al. (2021) explored an alternative approach called *selective differential privacy*, where the privacy guarantee only applies to blocks of text in the training corpus that are deemed sensitive, *e.g.*, addresses and phone numbers. Unfortunately, this approach is difficult to scale to large unstructured text corpora because it requires annotating all text in the corpus with a privacy sensitivity level.

SubMix has conceptual similarities to PATE (Papernot et al., 2016; 2018) for private semi-supervised learning. Both SubMix and PATE make predictions using an ensemble of models trained (or fine-tuned, in the case of SubMix) on private data, and employ a data-dependent and query-dependent privacy accounting mechanism at prediction time. The central idea in both methods is that privacy loss is small when models trained on different parts of the data agree on a prediction. However, PATE is more natural in discriminative or classification tasks because it return a distribution's noisy argmax. On the other hand, SubMix is more natural in generative tasks because it returns a sample from the distribution.

SubMix is also related to prior work on private posterior sampling (Geumlek et al., 2017; Dimitrakakis et al., 2017), where the randomness in the privacy mechanism comes from releasing a sample from a distribution defined by the private data. In particular, SubMix uses a privacy accounting methodology based on Rényi divergences similar to that of Geumlek et al. (2017).

Mireshghallah et al. (2021) propose adding a privacy regularizer to language model training to reduce its memorization of sensitive text. They empirically showed that the regularizer reduces the effectiveness of text extraction attacks, but it does not satisfy any formal privacy guarantee such as DP. Other related works exploring privacy in natural language processing include (Gopi et al., 2020; Lyu et al., 2020; Xu et al., 2020; Li et al., 2018; Kim et al., 2021).

Limitations & Future Directions One limitation of the SUBMIX protocol as presented here is that it only supports decoding from the ensemble of LMs by sampling directly from the predicted pmf. This type of sampling is known to produce unnatural and incoherent text (Kulikov et al., 2018; Holtzman et al., 2019). Better decoding methods such as top-k sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2019) exist, but they require modifications to the protocol that may cause additional privacy leakage. However, we note that a close alternative to top-k and nucleus sampling is *temperature decoding* (Holtzman et al., 2019), which scales the predicted pmf by a temperature term to decrease its entropy. SUBMIX readily supports this decoding method by applying temperature scaling as a post-processing step. In future work, we aim to extend out work by designing protocols that can support different types of decoding strategies as well.

Another limitation of SUBMIX is that the use of an ensemble substantially increases the computational and storage requirements. Our experiments suggests that an overhead factor of 8 is needed to attain a non-vacuous trade-off between privacy and utility. One potential solution to reduce the computational requirements of SUBMIX may be to fine-tune only the top few transformer layers closest to the prediction head. This would allow the evaluation of the bottom transformer layers to be shared between models in the ensemble, thereby reducing both computation and storage requirements. We leave the exploration of such efficiency improvements for future work.

REFERENCES

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

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 267–284, 2019.

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. *arXiv preprint arXiv:2012.07805*, 2020.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28:3079–3087, 2015.

Christos Dimitrakakis, Blaine Nelson, Zuhe Zhang, Aikateirni Mitrokotsa, and Benjamin Rubinstein. Differential privacy for bayesian inference through posterior sampling. *Journal of machine learning research*, 18(11):1–39, 2017.

Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. In *Conference On Learning Theory*, pp. 1693–1702. PMLR, 2018.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a renyi filter. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Joseph Geumlek, Shuang Song, and Kamalika Chaudhuri. R\'enyi differential privacy mechanisms for posterior sampling. *arXiv preprint arXiv:1710.00892*, 2017.

Sivakanth Gopi, Pankaj Gulhane, Janardhan Kulkarni, Judy Hanwen Shen, Milad Shokouhi, and Sergey Yekhanin. Differentially private set union. In *International Conference on Machine Learning*, pp. 3627–3636. PMLR, 2020.

Matthias Hertel. Neural language models for spelling correction. Master's thesis, Albert-Ludwigs-Universität Freiburg im Breisgau, 2019.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 1895–1912, 2019.

Gavin Kerrigan, Dylan Slack, and Jens Tuyls. Differentially private language models benefit from public pre-training. *arXiv preprint arXiv:2009.05886*, 2020.

Kunho Kim, Sivakanth Gopi, Janardhan Kulkarni, and Sergey Yekhanin. Differentially private n-gram extraction. *arXiv preprint arXiv:2108.02831*, 2021.

Ilia Kulikov, Alexander H Miller, Kyunghyun Cho, and Jason Weston. Importance of search and evaluation strategies in neural dialogue modeling. *arXiv preprint arXiv:1811.00907*, 2018.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. *arXiv preprint arXiv:1805.06093*, 2018.

Lingjuan Lyu, Xuanli He, and Yitong Li. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. *arXiv preprint arXiv:2010.01285*, 2020.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Fatemehsadat Mireshghallah, Huseyin A Inan, Marcello Hasegawa, Victor Rühle, Taylor Berg-Kirkpatrick, and Robert Sim. Privacy regularization: Joint privacy-utility optimization in language models. *arXiv preprint arXiv:2103.07567*, 2021.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017.

Piotr Mirowski and Andreas Vlachos. Dependency recurrent neural language models for sentence completion. In *Proc. IJCNLP*, pp. 511—-517, 2015.

Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. *arXiv preprint arXiv:2101.04535*, 2021.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pp. 547–561. University of California Press, 1961.

Philippe Rigollet. 18. s997: High dimensional statistics. *Lecture Notes), Cambridge, MA, USA: MIT Open-CourseWare*, 2015.

Eva Sharma, Chen Li, and Lu Wang. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*, 2019.

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

Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). *arXiv preprint arXiv:2011.11660*, 2020.

Laurens van der Maaten and Awni Hannun. The trade-offs of private prediction. *arXiv preprint arXiv:2007.05089*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

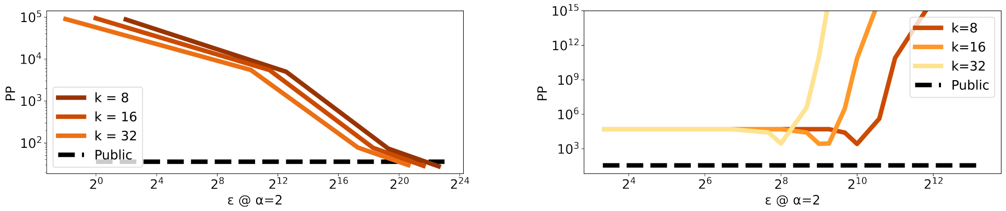
Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. A differentially private text perturbation method using a regularized mahalanobis metric. *arXiv preprint arXiv:2010.11947*, 2020.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282. IEEE, 2018.

Appendix

A Supplementary Figures

A.1 Baselines



(a) Subsample-and-aggregate

(b) GNMax

Figure 7: Perplexity of general-purpose private prediction mechanisms at different values of RDP privacy parameter ϵ on `Wikitext-103`.

We perform a more comprehensive evaluation of the privacy-utility trade-off for the baseline mechanisms. We observe that subsample-and-aggregate (S&A) in Figure 7a does not achieve a favorable trade-off. We also vary the number of parts k for the ensemble. Figure 7b shows the privacy-utility trade-off for the GNMax baseline. Unlike DP-SGD and S&A, GNMax achieves its minimal perplexity at some value of ϵ rather than perplexity being monotonically decreasing in ϵ. This is due to the fact that there is a reasonably strong consensus amongst the LM ensemble. When the noise magnitude is too small, the ensemble concentrates its prediction onto a single token, hence having an unbounded perplexity on other likely tokens. When the noise magnitude is too large, GNMax converges to the uniform distribution over all tokens. Nevertheless, even at the empirically optimal noise magnitude, GNMax achieves a worse perplexity than the public pre-trained LM.

A.2 Rényi Divergence Order

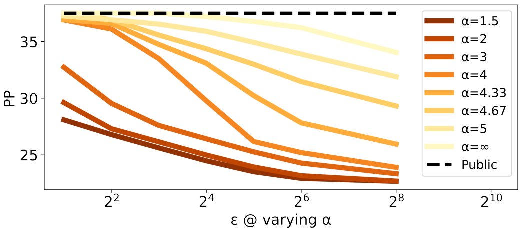


Figure 8: Perplexity of SubMix ($k = 8$) on `Wikitext-103` as a function of ROP privacy loss ϵ for different values of Rényi divergence order α. Budget $B = 2560$.

In our main experimental results, we used a Rényi divergence order of $\alpha = 2$. We explore how SubMix's privacy-utility trade-off varies under different choices of α. Note that Rényi divergence is monotonically increasing in α (Mironov, 2017), so the RDP guarantee further worsens for baseline mechanisms at higher α. Figure 8 shows the perplexity attained by SubMix at different ROP privacy parameters ϵ while varying the Rényi divergence order α. We observe that SubMix retains its desirable privacy-utility trade-off for $\alpha \leq 4$. At $\alpha > 4$, we observe a sharp decline in the performance of SubMix even at high values of ϵ. However, note that for all values of α, SubMix remains non-vacuous by achieving a non-negligible reduction in perplexity compared to the pre-trained LM at $\epsilon = 256$, although this is far from a reasonable guarantee.

B Experimental & Implementation Details

We provide additional details about SubMix and the baseline private prediction mechanisms.

B.1 SubMix

For SubMix training, on both `Wikitext-103` and `BigPatent-G`, we fine-tune GPT-2 using HuggingFace's AdamW optimizer at a learning rate of 0.0001 and batch size of 8. We use a linear warm-up schedule with the number of steps equal to the number of batches. At prediction time, we set the target privacy leakage parameter β using the simple yet solid heuristic $\beta = \epsilon/B$, where B is the number of queries to be answered.

B.2 DP-SGD

Concurrently with this work, Li et al. (2021) developed improved hyperparameter configurations for DP-SGD training of large-scale transformers. We have improved our DP-SGD baseline to reflect those findings. Of course, DP-SGD is a training algorithm, not a prediction protocol, and thus the released model can be used indefinitely. On the other hand, one of the major drawbacks of the DP-SGD is that it requires a large compute overhead. We find that for a single pass of the data, DP-SGD only marginally improves upon the pre-trained baseline at small ϵ on Wikitext-103, even using the newly discovered configurations. Allowing for a $10\times$ increase in computation (i.e. 10 total epochs), we do see modest improvements with DP-SGD. It is possible that more epochs of training could further improve DP-SGD, but we leave that exploration for future work.

We use the PyTorch Opacus[5] library for DP-SGD training. We swept in the range $[0.1, 1]$ for the noise multiplier to obtain the target ϵ. We set a small gradient clipping norm in of 0.1, as suggested by Li et al. (2021). We swept large batch sizes of $\{2048, 4096, 8192\}$, as suggested by Li et al. (2021). We used the AdamW solver with a learning rate of 0.0001.

B.3 Subsample-and-aggregate

Similar to SubMix, we train the LM ensemble for S&A by splitting the private corpus into k parts and training an LM on each part using identical hyperparameters. At prediction time, each LM outputs a next-token pmf $h_i(\mathbf{x}_t)$, and the mechanism aggregates these pmfs using a simple average: $h(\mathbf{x}_t) = {}^1\!/_k \sum_{i=1}^{k} h_i(\mathbf{x}_t)$. Since each $h_i(\mathbf{x}_t)$ is a probability vector with L_1-norm equal to 1, the global L_1-sensitivity of each LM's prediction is $1/K$. Based on this sensitivity, we apply the Laplace mechanism with a suitable scale to obtain Rényi differential privacy. Similar to SubMix, we compose the leakage over multiple queries using adaptive composition. Finally, we sweep the Laplace noise scale to obtain the optimal privacy-utility trade-off.

B.4 GNMax

GNMax trains an LM ensemble in the same manner as SubMix and S&A. At prediction time, GNMax produces a next-token histogram $\bar{\mathbf{n}}$, where $\bar{\mathbf{n}}_x$ is equal to the number of LMs in the ensemble that predict x as its top next token. The mechanism then adds Gaussian noise with scale σ to $\bar{\mathbf{n}}$ and outputs the token with the highest count after noise addition. Privacy accounting is done using the data-dependent and query-dependent bound in Theorem 6 of Papernot et al. (2018). Similar to SubMix and S&A, we accumulate privacy loss across multiple queries using adaptive composition.

Computing the predictive perplexity of GNMax is not straightforward as the mechanism does not output a next-token pmf, and the induced next-token pmf via taking the argmax of $\bar{\mathbf{n}} + \mathcal{N}(0, \sigma^2)$ does not have a simple closed form. Instead, we *lower bound* the perplexity using the following inequality. Let $\mathbf{q}$ denote the induced next-token pmf for GNMax, and let x be the ground truth next token. The (log) perplexity for this prediction is equal to $-\log(\mathbf{q}_x)$. Let random variable N denote the argmax of $\bar{\mathbf{n}} + \mathcal{N}(0, \sigma^2)$. We can upper bound $\mathbf{q}_x$ with:

$$\mathbf{q}_x = \mathbf{Pr}[x = N] \leq \mathbf{Pr}[x = N | N \in \{z : \bar{\mathbf{n}}_z \geq \bar{\mathbf{n}}_x\}] \leq \frac{1}{|\{z : \bar{\mathbf{n}}_z \geq \bar{\mathbf{n}}_x\}|}$$

The first inequality follows from the fact that conditioning N onto a subset of outcomes does not decrease the probability of any outcome that satisfies the condition, and the second inequality follows from the fact that $\mathbf{Pr}[x = N]$ is monotonically increasing in $\bar{\mathbf{n}}_x$. We also have the upper bound:

$$\mathbf{q}_x \leq \mathbf{Pr}[\mathcal{N}(\bar{\mathbf{n}}_x - \bar{\mathbf{n}}_{z^*}, 2\sigma^2) \geq 0],$$

[5] `https://github.com/pytorch/opacus`

where $z^* = \operatorname{argmax}_{z \neq x} \bar{\mathbf{n}}_z$. Taking the negative log of the minimum of the two upper bounds above allows us to lower bound the perplexity. We reported this lower bound in all of our experiments instead of the true perplexity for $\mathbf{q}$.

C Differential Privacy Conversions

We give conversion for RDP mechanisms using partition-level adjacency compared to user-level adjacency. Theoretically speaking, partition-level adjacency is neither stronger nor weaker than user-level adjacency. Therefore, conversion is costly in both direction. With that being said, at larger α, the conversion is a modest factor of 2 for partition-to-user, whereas the conversion cost grows large for small α. The conversion cost for user-to-partition does not depend on α and is always a factor of n/k, which is generally intolerably large.

C.1 Conversion from Partition-level RDP to User-level RDP

Theorem C.1. *For $\alpha > 2$, if a mechanism is partition-level (α, ϵ)-RDP under any partition, then it is user-level $(\alpha/2, \frac{2\alpha-3}{\alpha-2}\epsilon)$-RDP*

Proof. This follows from the application of the weak triangle inequality for Renyi divergence (Corollary 4a in Mironov (2017)).

Let M be the mechanism's output, M' be the mechanism's output with arbitrary change to the text of the i-th user, and let M'' be the mechanism's output with the entire part containing the i-user removed. Recall that in the main text we defined the *symmetrized* Renyi divergence $D_\alpha^{\text{sym}}(P||Q) = \max\{D_\alpha(P||Q), D_\alpha(Q||P)\}$.

The proof proceeds by showing that M and M' are always in a *symmetric* Renyi divergence ball about M''. In turn, this upper bounds the Renyi divergence between M and M'.

The weak triangle inequality tells us that for any R:

$$D_\alpha(P||Q) \leq \frac{\alpha - 1/2}{\alpha - 1} D_{2\alpha}(P||R) + D_{2\alpha-1}(R||Q)$$

Plugging in $P \leftarrow M$, $Q \leftarrow M'$ and $R \leftarrow M''$ yields the result with little effort.

$$\begin{aligned}
D_{\alpha/2}(M||M') &\leq \frac{\alpha/2 - 1/2}{\alpha/2 - 1} D_\alpha(M||M'') + D_{\alpha-1}(M''||M') \\
&\leq \frac{\alpha/2 - 1/2}{\alpha/2 - 1} D_\alpha(M||M'') + D_\alpha(M''||M') \\
&\leq \frac{\alpha/2 - 1/2}{\alpha/2 - 1} D_\alpha^{\text{sym}}(M||M'') + D_\alpha^{\text{sym}}(M'||M'') \\
&\leq \frac{\alpha/2 - 1/2}{\alpha/2 - 1} D_\alpha^{\text{sym}}(M||M'') + D_\alpha^{\text{sym}}(M'||M'') \\
&\leq \frac{\alpha - 3/2}{\alpha/2 - 1} \epsilon
\end{aligned}$$

□

Depending on α and ϵ, the conversion can be tightened somewhat by using the more general version of the weak triangle inequality (Prop. 11 in Mironov (2017)).

Corollary C.1.1. *If a mechanism is partition-level ϵ-DP under any partition, then it is user-level (2ϵ)-DP.*

Proof. Follow the proof in Thm. C.1 but in the limit as $\alpha \to \infty$. Notice that in this case, D_∞^{sym} is a proper metric and there is no slack in the triangle inequality. □

C.2 Conversion from User-level RDP to Partition-level RDP

Theorem C.2. *If a mechanism is user-level (α, ϵ)-RDP then it is partition-level $(\alpha, \frac{n}{k}\epsilon)$-RDP for a uniform k-partition*

Proof. Follows from the conversion from user-level RDP to group-level RDP (see a standard refernce, for example Dwork et al. (2014)). For group size of n/k, group-level RDP implies partition-level RDP under a uniform partition since group-level RDP implies statistical indistinguishability up to any choice of n/k users. □

C.3 Implications for Correlation Attacks

One reason for using group (R)DP is protection again correlation attacks. In the case that private text might exist amongst more than one user, such as a shared secret, group differential privacy provides protection as long as the number of users sharing the secret is small than the group size parameter. In general, ϵ-DP and (α, ϵ)-RDP guarantees implicitly a group size of $k = 1$. Naively, such a guarantee can be converted to a larger group size for a cost *multiplicative* in the group size. A user-level (α, ϵ)-RDP mechanism is also group-level $(\alpha, \kappa\epsilon)$-RDP under group size κ.

Therefore, any RDP guarantee *does* provide some protection against correlation attacks, but the strength of the protection decreases rapidly as the number of correlated user increases.

In some sense, this is to be expected, since, obviously, privacy becomes infeasible when all or a majority of users are correlated. However, for a modest number of correlated users, the privacy guarantee can still be significant.

C.4 Conversion from Variable to Fixed Length Prediction Sequences

Our definition of (α, ϵ)-ROP assumes a variable-length query sequence. We discuss here how to convert any variable-length (α, ϵ)-ROP guarantee into a fixed-length one. To clarify this distinction, we write (α, ϵ, T)-ROP to imply that the sequence length is fixed at T. Note that fixed-length (α, ϵ, T)-ROP differs from (α, ϵ, T)-RDP only in the notion of adjacency.

In order to do this, we make use of a general-purpose *random stopping* (RS) mechanism. Recall that a private prediction protocol $\mathcal{P}$ can issue a termination signal at any time, under the condition that any future queries must be answered in a data-independent way. We will use the public pre-trained model $h_\emptyset$ to answer queries after termination. The random stopping mechanism is parameterized by a *fixed* response budget B and an expansion factor $C > 1/2$. The random stopping mechanism then uniformly at random selects some value $\tau \in \{1, ..., CB\}$ and issues the termination before the τ-th response is made, *if it has not been issued already*. We refer to such a mechanism as a (B, C)-random stopping (RS) mechanism.

The proposition below shows that the additional information leaked from $T(\mathcal{P})$ for a (B, C)-RS mechanism is at most $\log(CB)$. Furthermore, we show that if the queries are drawn iid from some test distribution, then the expected test perplexity for this fixed-length version of the private prediction mechanism can be derived exactly.

Proposition C.3. *Let $h_\emptyset$ be a public LM. The following are true:*

(1) The (B, C)-RS mechanism converts any (α, ϵ)-ROP prediction protocol $\mathcal{P}$ to an $(\alpha, \epsilon + \log(CB), B)$-RDP prediction protocol.

(2) Suppose that the public LM achieves an expected test perplexity of $p_\emptyset$. If $\mathcal{P}$ has a stationary pre-termination expected test perplexity of p, then the expected test perplexity post-conversion is upper bounded by $(1 - \frac{1}{2C})p + \frac{p_\emptyset}{2C}$

Proof. Part (1) Let $\mathcal{P}$ be an (α, ϵ)-ROP prediction prediction protocol. Let $\mathbf{T}$ be the set of all termination rules that may (causally) depend on $\mathcal{D}$, Π, and $\{\mathbf{x}_t\}$. Let $\mathcal{T}$ denote the particular choice of termination rule used by $\mathcal{P}$. Let $\mathsf{RS}(\mathcal{T})$ denote rule $\mathcal{T}$ wrapped with the (B, C)-RS mechanism.

Consider the random length B sequence $\mathcal{P} \underset{B}{\leftrightharpoons} \mathsf{Adv}$. Recall that after termination is issued, $\mathcal{P}$ falls back to $h_\emptyset$ to answer queries, so no user information can be leaked after termination. Thus, $\mathcal{P} \underset{B}{\leftrightharpoons} \mathsf{Adv}$

is information-theoretically equivalent to $\left(\mathcal{P} \underset{\mathsf{RS}(\mathcal{T}))}{\leftrightharpoons} \mathsf{Adv}, \mathsf{RS}(\mathcal{T})\right)$. In other words, information leakage can only occur in the head of the sequence (determined by $\mathsf{RS}(\mathcal{T})$) and in the timing of termination $\mathsf{RS}(\mathcal{T})$ itself, but not in the tail in which queries are answered independently of data.

Let $\mathcal{P}'$ and $\mathcal{T}'$ denote adjacent protocol and termination rules, respectively.

We seek to upper bound:

$$D_\alpha\left(\mathcal{P} \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv}, \mathsf{RS}(\mathcal{T}) || \mathcal{P}' \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv}, \mathsf{RS}(\mathcal{T}')\right)$$

We must exercise a bit of caution, because $\mathsf{RS}(\mathcal{T})$ is not necessarily independent of $\mathcal{P} \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv}$, so we cannot split up the joint Rényi divergence into a sum without additional justification.

However, in general for any joint random variable pairs (X, Y) and (X', Y') of equal support:

$$D_\alpha(X, Y || X', Y') \leq D_\alpha(X || X') + \max_{x \in \mathcal{X}} D_\alpha(Y|x || Y'|x)$$

The above identity follows easily from replacing joint distributions $p(x, y)$ and $p'(x, y)$ with marginalized distributions $p(x)p(y|x)$ and $p'(x)p'(y|x)$ in the definition of Rényi divergence.

We apply the identity:

$$D_\alpha\left(\mathcal{P} \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv}, \mathsf{RS}(\mathcal{T}) || \mathcal{P}' \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv}, \mathsf{RS}(\mathcal{T}')\right) \leq \tag{3}$$

$$D_\alpha\left(\mathcal{P} \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv} || \mathcal{P}' \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv}\right) + \max_{\leftrightharpoons} D_\alpha\left(\mathsf{RS}(\mathcal{T})| \leftrightharpoons || \mathsf{RS}(\mathcal{T}')) \mid \leftrightharpoons\right) \leq \tag{4}$$

$$D_\alpha\left(\mathcal{P} \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv} || \mathcal{P}' \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv}\right) + \max_{\mathsf{T}, \mathsf{T}' \in \mathbf{T}} D_\alpha(\mathsf{RS}(\mathsf{T}) || \mathsf{RS}(\mathsf{T}')) \leq \tag{5}$$

$$D_\alpha\left(\mathcal{P} \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv} || \mathcal{P}' \underset{\mathsf{RS}(\mathcal{T})}{\leftrightharpoons} \mathsf{Adv}\right) + \max_{\mathsf{T}, \mathsf{T}' \in \mathbf{T}} D_\infty(\mathsf{RS}(\mathsf{T}) || \mathsf{RS}(\mathsf{T}')) \leq \tag{6}$$

$$\epsilon + \log(BC) \tag{7}$$

Eqn. 4 follows from the application of the identity. The symbol $\leftrightharpoons$ denotes a shorthand for the realized query-response sequence.

Eqn. 5 follows from upper bounding the maximal divergence conditioned over all sequences $\leftrightharpoons$ with the maximal divergence over all possible termination rules. This is immediate given that any the termination rule conditioned on a sequence is contained in $\mathbf{T}$.

Eqn. 6 follows from $D_\alpha \leq D_\infty$ for any α. The first term of Eqn. 7 follows from the (α, ϵ)-ROP assumption on $(\mathcal{P}, \mathcal{T})$, and the second term follows from noting that $\mathbf{Pr}[\mathsf{RS}(\mathsf{T}) = t] \geq 1/BC$ for all $t \in \{1, ...B\}$.

Part (2) By assumption, we have that:

$$p = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{test}}}[\mathbf{PP}_{\mathcal{P}}]$$

before termination. We also have that

$$p_\emptyset = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{test}}}[\mathbf{PP}_{h_\emptyset}]$$

Let $V_t = \mathbf{1}\{t < \mathsf{RS}(\mathcal{T})\}$. Due to the stationarity, the expected perplexity is:

$$\frac{1}{B}\sum_{t=1}^{B} pV_t + p_\emptyset \bar{V}_t$$

Computing the sum above yields the result. □

We walk through an example in order to make the Prop. above more concrete. In Fig. 2, we report SubMix achieves $\mathbf{PP} = 26.9$ at $\epsilon = 2$ for $B = 1000$ queries. Selecting a value of $C \leftarrow 10$ results in $\mathbf{PP}_{C\leftarrow 10} \leq 27.78$ at $\epsilon_{C\leftarrow 10} = 11.21$. Selecting $C \leftarrow 100$ results in $\mathbf{PP}_{C\leftarrow 100} \leq 26.988$ at $\epsilon_{C\leftarrow 100} = 13.51$. Selecting $C \leftarrow 1$ results in $\mathbf{PP}_{C\leftarrow 1} \leq 31.3$ at $\epsilon_{C\leftarrow 1} = 8.9$.

D Text Extraction and Eidetic Memorization

We formalize the notion of extraction through the framework of statistcal hypothesis testing.

Definition D.1. (Text Extraction Game) An unknown target string $S \in \Sigma^*$ is a substring appearing in the corpus $\mathcal{D}$. Suppose S has been narrowed down to one of m values: $S \in \{\mathfrak{s}_1, ..., \mathfrak{s}_m\}$. An adversary Adv observes (or interacts with) mechanism (or protocol) $\mathcal{M}$ and outputs a guess $\mathsf{s} = \mathsf{Adv}(\mathcal{M})$.

Definition D.2. ((β, m)-Extractibility) An unknown string S is (β, m)-extractable from mechanism (or protocol) $\mathcal{M}$ if for any choice of $\mathcal{D}$ and $(\mathfrak{s}_1, ..., \mathfrak{s}_m)$, there exists an adversary Adv such that:

$$\min_i \mathbf{Pr}(S = \mathsf{s} | S = \mathfrak{s}_i) > \beta$$

Remark D.1. *The reader may wonder why we use* $\min_i$ *rather than* $\max_i$*. The reason is because the attacker should be unbiased about which string they want to extract, so to maximize their success rate they should equalize this success rate across all strings* $\mathfrak{s}_i$*.*

Definition D.3. ((κ, β, m)-Eidetic Memorization) A string S appearing in at most κ examples in the training data $\mathcal{D}$ is (κ, β, m)-eidetic memorized if S is (β, m)-extractable.

Theorem D.2. *If a mechanism* $\mathcal{M}$ *is* (α, ϵ)*-RDP then* $\mathcal{M}$ *cannot* $(\kappa, \frac{\epsilon\kappa + \log(2)}{\log(m)}, m)$*-eidetically memorize any string* S*.*

Remark D.3. *This result does not depend on* α *and therefore incurs more slack when* α *is large.*

Proof. To begin, note that κ-eidetic memorization limits the occurrences of S in $\mathcal{D}$. Given that S appears at most κ times in $\mathcal{D}$, it follows that S appears in at most κ user texts:

$$|\{\mathcal{D}_i : S \in \mathcal{D}_i\}| \leq \kappa \tag{8}$$

Let $\mathcal{D}|\{S = \mathfrak{s}_i\}$ denote the dataset when S takes on value $\mathfrak{s}_i$.

Let $\mathcal{M}|\{S = \mathfrak{s}_i\}$ denote the output of the mechanism when S takes on value $\mathfrak{s}_i$.

Let $d_\mathcal{H}$ denote the user-level Hamming distance between datasets.

Based on (8) we know that for all $i, j \in [m]$:

$$d_\mathcal{H}\left(\mathcal{D}|\{S = \mathfrak{s}_i\}, \mathcal{D}|\{S = \mathfrak{s}_j\}\right) \leq \kappa \tag{9}$$

By assumption, $\mathcal{M}$ is (α, ϵ)-RDP, so from the above follows:

$$D_\alpha\left(\mathcal{M}|\{S = \mathfrak{s}_i\} || \mathcal{M}|\{S = \mathfrak{s}_j\}\right) \leq \kappa\epsilon \tag{10}$$

By the monotonicity of Renyi divergence in α:

$$D_{\text{KL}}\left(\mathcal{M}|\{S = \mathfrak{s}_i\} || \mathcal{M}|\{S = \mathfrak{s}_j\}\right) \leq D_\alpha\left(\mathcal{M}|\{S = \mathfrak{s}_i\} || \mathcal{M}|\{S = \mathfrak{s}_j\}\right) \leq \kappa\epsilon \tag{11}$$

With an upper bound on the KL-Divergence, the final step will be the application of Fano's inequality to multiple hypothesis testing (Rigollet, 2015):

$$\max_i \mathbf{Pr}(\mathsf{s} \neq S | S = \mathfrak{s}_i) \geq 1 - \frac{\kappa\epsilon + \log 2}{\log m} \tag{12}$$

□