usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

Gradients Look Alike: Sensitivity is Often Overestimated in DP-SGD

Anvith Thudi and Hengrui Jia, *University of Toronto and Vector Institute;* Casey Meehan, *University of California, San Diego;* Ilia Shumailov, *University of Oxford;* Nicolas Papernot, *University of Toronto and Vector Institute*

https://www.usenix.org/conference/usenixsecurity24/presentation/thudi

This paper is included in the Proceedings of the
33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Gradients Look Alike: Sensitivity is Often Overestimated in DP-SGD

Anvith Thudi[‡§], *Hengrui Jia*[‡§], *Casey Meehan*[†], *Ilia Shumailov*[*], *Nicolas Papernot*[‡§]
University of Toronto[‡], *Vector Institute*[§], *University of California, San Diego*[†], *University of Oxford*[*]

Abstract

Differentially private stochastic gradient descent (DP-SGD) is the canonical approach to private deep learning. While the current privacy analysis of DP-SGD is known to be tight in some settings, several empirical results suggest that models trained on common benchmark datasets leak significantly less privacy for many datapoints. Yet, despite past attempts, a rigorous explanation for why this is the case has not been reached. Is it because there exist tighter privacy upper bounds when restricted to these dataset settings, or are our attacks not strong enough for certain datapoints? In this paper, we provide the first per-instance (i.e., “data-dependent") DP analysis of DP-SGD. Our analysis captures the intuition that points with similar neighbors in the dataset enjoy better data-dependent privacy than outliers. Formally, this is done by modifying the per-step privacy analysis of DP-SGD to introduce a dependence on the distribution of model updates computed from a training dataset. We further develop a new composition theorem to effectively use this new per-step analysis to reason about an entire training run. Put all together, our evaluation shows that this novel DP-SGD analysis allows us to now *formally* show that DP-SGD leaks significantly less privacy for many datapoints (when trained on common benchmarks) than the current data-independent guarantee. This implies privacy attacks will necessarily fail against many datapoints if the adversary does not have sufficient control over the possible training datasets.

1 Introduction

Differential Privacy (DP) is the standard framework for private data analysis [10]. Making an algorithm differentially private limits the success any attack can have in knowing whether any datapoint was or was not an input to the algorithm given just the outputs of the algorithm. To obtain this notion of indistinguishability the algorithm needs to perform a noisy analysis of the data. In the case of deep learning, the canonical private training algorithm is DP-SGD [1], where Gaussian noise is added to the gradients computed on training examples. Much work has gone into improving the privacy analysis of DP-SGD for a given amount of noise [17, 33, 34] in an effort to minimize the impact of noise on performance.

To reiterate, this current privacy analysis for DP-SGD is *data independent*: it assumes an upper-bound on how much *any* individual datapoint from *any* dataset can have their privacy leaked. It is furthermore now known to be tight; there exist specific pairs of datasets and models for which a privacy attack can match the upper-bound of DP-SGD [35]. Yet, when training on common benchmark datasets like CIFAR10, Carlini et al. [9] empirically saw that even strong privacy attacks perform significantly worse for many datapoints than the guarantees associated with DP-SGD. That is, when training on real-world data, there is a gap between what our strongest attacks can achieve and what our current data-independent privacy analysis of DP-SGD can tell us. Hence the question, why is there a gap? Is it because our attacks are still too weak for many datapoints, or is it because there exist tighter privacy upper bounds when restricted to these dataset settings? Past work attempted to answer this by analyzing specific privacy attacks [19, 31, 40], or studying a weakened notion of DP [44]. But all past work either have bounds limited in scope or with unproven assumptions. No work has yet derived tighter indistinguishability guarantees that are specific to the data being analyzed, analogous to how DP gives indistinguishability guarantees to prevent privacy attacks (for all possible datasets).

Our work provides the first per-instance DP analysis of DP-SGD, i.e., bounds on the distinguishability of outputted models that are specific to training on a given dataset or the dataset plus a point. This analysis bridges the theoretical gap between the tight data-independent analysis of DP-SGD and what is achievable when training on common deep-learning datasets. These new guarantees follow from an exploration of the role of *sensivity* in privacy analysis. Currently, to obtain data-independent privacy guarantees, a model trainer needs to bound how much *any* individual datapoint from *any* dataset can contribute to a gradient update—a quantity known as the algorithm’s sensitivity. This is currently done by setting an upper-bound ahead of time which is enforced during training by clipping the gradient computed on each datapoint to a norm below this preset sensitivity value. However, we highlight that this overestimates the sensitivity of DP-SGD to a *specific* datapoint in a *given* dataset when that datapoint's update is similar to the update given by many other datapoints in this dataset. In deep learning, many mini-batches in a dataset do produce similar gradients [25, 37, 39], hence such a case of overestimating sensitivity is common. Our per-instance (i.e., "data-dependent") DP analysis of DP-SGD leverages this phenomenon.

Let us first focus on a single update of DP-SGD. Intuitively, if many of the datapoints produced almost the same gradient, then with high probability we would have obtained the same updated model with or without one of these datapoints. Making this intuition rigorous, we introduce a class of distributions we call *sensitivity distributions*: broadly they capture the difference between updates computed from a given mini-batch to sampling another mini-batch. From this, we derive new bounds on the privacy leakage of a single DP-SGD update that incorporates how concentrated these distributions are at small values, i.e., have many mini-batches that produce almost the same gradient. Using this bound we can show that for many datapoints in common benchmark datasets, the individual per-step guarantee for that point can be magnitudes lower than the data-independent guarantee.

Building on our analysis for a single DP-SGD update, we give a per-instance bound on the *overall* privacy leakage of a *full DP-SGD run*. The current analysis considers the model that leaks the most privacy at every step (the worst-case model) and notes that summing the maximum per-step leakages bounds the overall privacy leakage of a DP-SGD run. Yet, the sensitivity distributions that we introduce are heavily dependent on the model being updated: e.g., there is a difference between gradients computed using a partially-trained model and a randomly-initialized model. Towards not relying on analyzing worst-case models, intuitively it should not matter what the privacy leakage of the worst-case models is if they are unlikely to be reached. More rigorously, we develop a new composition theorem which allows us to upper-bound the overall per-instance privacy leakage of using DP-SGD by the expected privacy leakage at each step during training.

These analytic results give a new framework to understand the privacy guarantees of DP-SGD for individual datapoints. However, it remains to verify whether this analysis is tight enough to show that many datapoints have better privacy when training on benchmark datasets. We thus turn to experimentation.[1] The crux of implementing our results is to repeat training several times to compute the expected per-step privacy leakage. Because this one-dimensional statistic is bounded by the existing worst-case privacy analysis, one achieves non-trivial estimates with few samples. Doing this:

1. We show that when training on common benchmark datasets, many data points have better per-instance privacy than what the current data-independent guarantee associated with DP-SGD tells us. For some datapoints, we observe more than a magnitude improvement in the privacy guarantee ε. This explains the prior results we motivated our work with: for many datapoints, attacks that can only observe the outputs from training with or without the datapoint will fail.
2. In our framework, we observe a disparity where correctly-classified points obtain better privacy guarantees than misclassified points. In other words, training algorithms that lead to high-performing models quantifiably leak less per-instance privacy for many points. This is as they reach states that have similar updates for large clusters of datapoints. We hypothesize that designing model architectures to be more performative may also make them more private.
3. In classical privacy analysis, training with higher mini-batch sampling rates leaks more privacy. However we find that for certain update rules, training with higher sampling rates can give better per-instance privacy because mini-batch updates concentrate on the dataset mean; this leads to many mini-batches with similar updates.

The consequences of our work are far reaching: having better per-instance DP guarantees has implications for unlearning, generalization, memorization and privacy auditing because of how DP formulates privacy by preventing distinguishability between the models trained with or without a datapoint. For unlearning, a strong per-instance DP guarantee implies that the models coming from training with a datapoint are indistinguishable to the models trained without it. We further discuss in the paper how per-instance DP guarantees can still be used to satisfy a private notion of unlearning, and provide a naive first algorithm which we hope motivates future work. For generalization and memorization, a strong per-instance DP guarantee implies that the models coming from training without a datapoint perform similarly to those that had trained with it. For privacy auditing, our work provides empirical upper-bounds to complement previous work on lower-bounds established by strong privacy attacks, allowing future work to test if such attacks are tight. We note however that privacy auditing can affect the data-independent privacy guarantee, and discuss mitigations and paths for future work in the paper. With our framework, one can now say a specific datapoint does not need to be unlearned, or that a datapoint will not be memorized. However, our work leaves open how to apply our analysis to obtain better data-independent privacy guarantees when possible.

2 Background

Here we describe the current data-independent privacy analysis of DP-SGD (Section 2.1) and the relevance of per-instance DP in explaining empirical privacy attacks in contrast to past

[1] The code is at https://github.com/cleverhans-lab/Gradients-Look-Alike-Sensitivity-is-Often-Overestimated-in-DP-SGD.

approaches (Section 2.2). We also discuss the implication of per-instance DP for unlearning and memorization in Section 2.2. Later, in Section 5.1, we describe past work on generalizing composition theorems and how they are not applicable for a better per-instance analysis of DP-SGD.

Machine Learning Notation We consider a learning setup where we have a dataset $X = \{x_1, \cdots x_n\}$ with datapoints from some space $\mathfrak{X}$ (e.g., images and their labels). Given a loss function $\mathcal{L} : \mathbb{R}^d \times \mathfrak{X} \to \mathbb{R}$, our objective is to minimize the loss $\frac{1}{n}\sum_{x_i \in X} \mathcal{L}(\theta, x_i)$ with respect to the parameters $\theta \in \mathbb{R}^d$ of some model. The canonical approach to do this for deep learning models is to use stochastic gradient descent (SGD). However, we consider having an additional requirement that the models we obtain should not leak the “privacy" of individual datapoints

2.1 DP-SGD Analysis

DP [10] is the de-facto definition of privacy used in ML. The typical definition used in machine learning is given below, where one thinks of M as the training algorithm:

Definition 2.1 ((ε,δ)-DP)**.** An algorithm M is said to be (ε,δ)-DP if for all neighbouring datasets X, X' (i.e. Hamming distance 1 apart), we have that

$$\mathbb{P}(M(X) \in S) \leq e^{\varepsilon}\mathbb{P}(M(X') \in S) + \delta$$

DP-SGD [1, 3, 38] is an (ε,δ)-DP version of stochastic gradient descent (SGD) which clips the individual gradients and adds Gaussian noise to the mini-batch update. Formally, given a dataset X, DP-SGD repeatedly computes the following deterministic update rule $U(X_B = \{x : x \sim X \textit{ with probability } \frac{L}{|X|}\}) = \sum_{x \in X_B} \nabla_\theta \mathcal{L}(\theta, x) / \max(1, \frac{||\nabla_\theta \mathcal{L}(\theta,x)||_2}{C})$ and then updates $\theta \to \theta - \eta \frac{1}{L}(U(X_B) + N(0, \sigma^2 C^2))$.

The current tightest privacy analysis of DP-SGD uses Rényi-DP (RDP) [33] which implies (ε,δ)-DP; the merits of first working with RDP is that it provides a tighter privacy analysis for releasing the composition of multiple steps in DP-SGD – where each step is an update computed on a different mini-batch X_B. An algorithm M is (α,ε)-Rényi DP if for all neighbouring datasets X, X' we have $D_\alpha(M(X)||M(X')) \leq \varepsilon$ where for two probability distributions P, Q we define the α-Rényi divergence as

$$D_\alpha(P||Q) := \frac{1}{\alpha - 1} \ln \mathbb{E}_{x \sim Q}(\frac{P}{Q})^\alpha$$

The RDP analysis follows two steps:

1. Per Step: Analyzing the privacy guarantee of each training step $\eta \frac{1}{L}(U(X_B) + N(0, \sigma^2 C^2))$, which is the same as $U(X_B) + N(0, \sigma^2 C^2)$ by the post-processing property of RDP (the output of a DP algorithm can be post-processed without degrading the DP guarantee provided).
2. Composition: Understanding the accumulated RDP guarantee of releasing all the updates.

The first part was analytically studied in Mironov et al. [34] and is called the sampled Gaussian mechanism. The accumulation step follows from the composition theorem for RDP [33]. In this paper, we provide new per-step and composition privacy analyses for DP-SGD that are *specific* to a pair of neighbouring datasets X, X'.

2.2 Motivation for Studying Per-Instance DP

In contrast to the classical analysis of DP-SGD, we will analyze its per-instance Rényi DP guarantees [43] – also known as "Individual Rényi DP" [15]. That is, the RDP guarantee specific to a *given* pair of neighbouring datasets.

Definition 2.2 (Per-Instance Rényi DP)**.** We say an algorithm M is (α,ε) per-instance Rényi DP for a pair of datasets $X, X' = X \cup x^*$ if

$$\max\{D_\alpha(M(X)||M(X')), D_\alpha(M(X')||M(X))\} \leq \varepsilon$$

Colloquially, when X is understood from context, we will specify the per-instance guarantee by saying an algorithm is (α,ε)-Rényi DP for a point x^* (which determines X').

Per-instance DP guarantees provide a privacy upper bound for an adversary trying to distinguish between a specific pair of datasets X, X' given the ouput of M on one of them, and not a bound for all neighbouring datasets like classical DP. However, this granularity allows for tighter analysis of each X, X' case. The tighter per-instance DP bounds we derive will allow us to say that for specific pairs of datasets $X, X' = X \cup x^*$, privacy attacks against x^* will fail when the adversary can only observe the outputs from X or X'. More generally, if there are strong per-instance guarantees for all the neighbouring datasets the adversary can observe, then they will still fail. Instantiating our analysis, we will show that on common benchmark datasets, an adversary trying to distinguish if a specific point was added or not will fail for many points.

Our work on upper bounding per-instance DP guarantees is contrasted with past work on rigorously explaining when privacy attacks against DP-SGD will perform worse than what is implied by the current (tight) data-independent analysis. One line of work has been to upper-bound the performance of specific attacks. Putting aside the limitation in only upper-bounding specific attacks, this line of work either lacks an individualized guarantee to explain the difficulty for individual points [31] or relies on a particular threat model to explain better privacy [40]. In the case improved individual upper-bounds were achieved [18], this was with bounds that can fail due to assumptions. In short, this line of work lacks the generality/strength of Definition 2.2 in explaining why *any* empirical privacy attack will perform worse in some data settings.

A more recent line of work has been to attempt to do individual (i.e., per-instance) DP accounting for DP-SGD [44].

However, Yu et al. [44] could not analyze the per-instance guarantees of DP-SGD and instead relied on a weaker guarantee that holds if intermediate models were not random (which is not true for DP-SGD). The main technical bottleneck to extend their approach to analyze DP-SGD, as also noted by Yu et al. [44], was how to effectively analyze composition when the intermediate models are random variables. Our work provides a new composition theorem to handle this technical issue, and in doing so provides proper per-instance DP guarantees without the assumptions present in Yu et al. [44].

Per-instance DP guarantees are also important beyond privacy. Memorization [13] is a per-instance quantity (only reasoning about a particular pair X, X'), and hence is bounded by Definition 2.2. Similarly, unlearning is a per-instance quantity, and a growing section of the literature uses per-instance DP guarantees to quantify unlearning [18]. Hence in providing per-instance DP bounds for DP-SGD, we have also quantified a set of points that will not be memorized nor need to be unlearned (as they are already unlearnt). We however remark that care must be taken to user per-instance guarantees without voiding other privacy guarantees, and discuss this for unlearning where the privacy guarantee is to be agnostic to the order of unlearning requests in Section 5.4. We refer the reader to Kulynych et al. [28] for a more general discussion on the utility of DP inequalities in studying properties of deep learning.

3 A Per-Instance Analysis of DP-SGD

We now present our new analysis of DP-SGD which removes the data-independent nature of the per-step and composition analyses currently used for DP-SGD. The impact of this new analysis is presented in Section 4, where we show that many datapoints have much better privacy than suggested by the current analysis of DP-SGD, explaining the failure of many privacy attacks in practice.

The technical contributions that led to this are two-fold. At the per-step level, we generalize the notion of sensitivity to what we term *sensitivity distributions*; given two datasets, sensitivity distributions capture how similar the updates between mini-batches from either dataset are. At the composition step, we generalize RDP composition to do accounting by the "expected" intermediate privacy losses during training as opposed to the largest possible intermediate privacy losses. Together, we can now study the data-dependent behaviour of DP-SGD.

3.1 Sensitivity Distribution Generalize the (ε,δ)-DP Analysis

We first turn to (ε,δ)-DP, which is not used to analyze DP-SGD for composition reasons, but allows for simpler expressions to demonstrate the improvements afforded by particular data-dependent random variables we call *sensitivity distributions*. In particular, in this section we will first consider the classical data-independent (ε,δ)-DP analysis of the sampled Gaussian mechanism M and show how one can generalize this analysis and obtain tighter per-instance (ε,δ)-DP guarantees.

Recall that for an update rule U, the Gaussian mechanism is defined as $A(X) = U(X) + N(0,\sigma)$. The sampled Gaussian mechanism is then defined as $M(X) = A(\mathbf{X_B})$ where $\mathbf{X_B}$ is a mini-batch constructed from a dataset X by sampling each datapoint $x \in X$ independently with probability $\mathbb{P}_x(1)$ (unless otherwise stated we think of X_B, not bold-face, as a specific mini-batch). Note, one assumes the sampling probability $\mathbb{P}_x(1)$ is only a function of x and not the full dataset X, e.g., some fixed constant. The classical data-independent (ε,δ)-DP analysis of the sampled Gaussian mechanism follows two steps. First, we derive the guarantee for just the Gaussian mechanism. To do so, one first assumes a data-independent sensitivity bound C_U on U: for all $X, X' = X \cup \{x^*\}$ we have $||U(X) - U(X')||_2 \leq C_U$. This can be achieved by clipping the output values of U to have a small norm. With this constant C_U one has that the Gaussian mechanism A gives the (ε,δ)-DP guarantee $\varepsilon = C_{\delta,\sigma}C_U$ for some constant $C_{\delta,\sigma}$ depending on δ and σ where σ is the standard deviation of the added Gaussian noise [2]. To then analyze the sampled Gaussian mechanism one would incorporate the privacy gain from not sampling x^* sometimes [4][24] to get the privacy guarantees of M as (ε',δ')-DP where $\varepsilon' = \ln(\mathbb{P}_{x^*}(1)e^{C_{\delta,\sigma}\, C_U} + \mathbb{P}_{x^*}(0))$ and $\delta' = \mathbb{P}_{x^*}(1)\delta$. Here $\mathbb{P}_{x^*}(0) = 1 - \mathbb{P}_{x^*}(1)$, and this gain in privacy by sometimes not using the datapoint is called privacy amplification by sampling.

Towards tightening this analysis into a per-instance analysis, let

$$\Delta_{U,x^*}(X_B) := ||U(X_B) - U(X_B \cup \{x^*\})||_2$$

then $\Delta_{U,x^*}(\mathbf{X_B})$ is a data-dependent random variable which we will call a *sensitivity distribution*: it captures the change in the distribution of mini-batches updates caused by adding a point x^* to the mini-batch. The classical data-independent analysis only (implicitly) uses sensitivity distributions via the data-independent bound $|\Delta_{U,x^*}(X_B)| \leq C_U \ \forall X_B$. Instead, we will show how to directly use the L_p norms $||\Delta_{U,x^*}(\mathbf{X_B})||_p = (\mathbb{E}_{X_B}(\Delta_{U,x^*}(X_B)^p))^{1/p}$ (or generally the L_p norm of some monotonic transformation of $\Delta_{U,x^*}(\mathbf{X_B})$) to obtain tighter per-instance privacy guarantees. Furthermore, when using $p < \infty$, this analysis will be able to translate the phenomenon that many mini-batches produce similar updates into better privacy guarantees (as the sensitivity distribution concentrates at smaller values and hence has smaller p-norms). To emphasize this ability, past work that studied sampling relied mainly on the intuition that by sampling a datapoint with low probability, we have any given step often does not leak privacy for that point as it was not used. This translates to better privacy gurantes. By using the L_p norms of sensitivity distributions with $p < \infty$ we make an additional observation, which is that if many of the other mini-batches produce the same update, then

[2] For example, one can take $C_{\delta,\sigma} = \frac{\sqrt{2\ln(1.25/\delta)}}{\sigma}$ [11].

effectively we have an even lower probability of an attacker observing a noticeable shift due solely to that point.

In particular, recall that to prove per-instance (ε,δ)-DP for a pair of datasets $X, X' = X \cup \{x^*\}$ we need to bound $\mathbb{P}(M(X') \in S) \leq e^{\varepsilon}\mathbb{P}(M(X) \in S) + \delta$ and $\mathbb{P}(M(X) \in S) \leq e^{\varepsilon}\mathbb{P}(M(X') \in S) + \delta$. As a proof-of-concept on the role of sensitivity distributions, we present an analysis for the first inequality in Corollary 3.1 [3]. Inspecting Corollary 3.1, we see that it approximately follows the formula given by the classical analysis except the role of C_U is replaced with a dependency on how concentrated $\Delta_{U,x^*}(X_B)$ is at small values (the L_p norm of an exponential applied to $\Delta_{U,x^*}(X_B)$). When enough mini-batches provide updates more similar than the upper-bound C_U, the per-instance guarantee of Corollary 3.1 will significantly beat the classical data-independent analysis, as demonstrated for MNIST and CIFAR10 in Appendix B.

Corollary 3.1. *For $p \in (1,\infty)$, let $a_p = \mathbb{P}_{x^*}(1)(\mathbb{E}_{x_B}(e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)p}))^{1/p}$, $\varepsilon' = \ln(a_p^{\frac{1}{1-1/p}}\delta'^{\frac{-1}{p-1}} + \mathbb{P}_{x^*}(0))$ and $\delta'' = \mathbb{P}_{x^*}(1)\delta + \delta'$. Then, for $X' = X \cup \{x^*\}$ we have the following per-instance guarantee*

$$\mathbb{P}(M(X') \in S) \leq e^{\varepsilon'}\mathbb{P}(M(X) \in S) + \delta''$$

Proof Sketch: The proof of Corollary 3.1 follows two stages. First by expanding mini-batch sampling and applying Holder's inequality, we can show

$$\begin{aligned}&\mathbb{P}(M(X') \in S)\\ &\quad \leq \mathbb{P}_{x^*}(1)\mathbb{E}_{X_B}(e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)p})^{1/p}\mathbb{P}(M(X) \in S)^{1-1/p}\\ &\qquad\qquad + \mathbb{P}_{x^*}(1)\delta + \mathbb{P}_{x^*}(0)\mathbb{P}(M(X) \in S) \qquad (1)\end{aligned}$$

This is stated as Lemma A.1. One then analyzes the previous inequality in cases (first case is δ is upper-bounded, and else it is lower-bounded) to obtain an (ε,δ)-DP inequality. The full proof of Corollary 3.1 is in Appendix A.2.

3.2 Per-Instance Rényi-DP Analysis for DP-SGD

With now an understanding of the power of incorporating L_p norms of sensitivity distributions (upto some transformations) into DP analyses, we turn to analyzing the Rényi-DP guarantees of DP-SGD. Rényi-DP is more suited to compose the guarantees of each step of DP-SGD to obtain the guarantees for an entire training run. We first present per-step analyses for the sampled Gaussian mechanism, and then a new composition theorem to reason about the entire training run. We then discuss how to analyze DP-SGD for general update rules, i.e., not just the sum of gradients.

Our per-step analyses will focus on integer values of α for Rényi-DP. This is for simplicity, as Rényi divergences $D_\alpha(P||Q) := \frac{1}{\alpha-1}\ln\mathbb{E}_{x\sim Q}(\frac{P}{Q})^\alpha$ are increasing in their order α, hence we can bound the guarantee for any α by the guarantee for $\lceil\alpha\rceil$. In terms of notation, we will use ${X_B}^{\tilde{\alpha}} = ({X_B}^1, \cdots, {X_B}^\alpha)$ to denote α mini-batches from X (sampled independently if random). Analogously we use ${X'_B}^{\tilde{\alpha}}$ and X'_B for X'.

3.2.1 Per-Instance Rényi DP for the Sum Update Rule

In Section 3.1 we introduced the sensitivity distribution $\Delta_{U,x^*}(\mathbf{X_B}) = ||U(\mathbf{X_B}) - U(\mathbf{X_B} \cup \{\mathbf{x}^*\})||_2$ and showed how directly leveraging its L_p norms gives better per-instance DP analysis. In particular, how $p < \infty$ allows one to take advantage of expected sensitivity over mini-batches. However, for update rules of the form $U(X_B) = \sum_{x_i \in X_B} g(x_i)$ (i.e., the sum update rule typically used in DP-SGD) we have $\Delta_{U,x^*}(\mathbf{X_B})$ is always a constant: $\Delta_{U,x^*}(\mathbf{X_B}) = ||g(x^*)||_2$. Hence an analysis of the sampled Gaussian mechanism that used $\Delta_{U,x^*} := \sup_{X_B \sim X}\Delta_{U,x^*}(X_B)$ would effectively capture all L_p norms of the sensitivity distribution $\Delta_{U,x^*}(X_B)$ for the sum update rule. We state such a per-instance version of the classical RDP analysis of the sampled Gaussian mechanism below.

Theorem 3.2. *For integer $\alpha > 1$, the sampled Gaussian mechanism with noise σ and sampling probability $\mathbb{P}_{x^*}(1)$ for x^* is (α,ε) per-instance Rényi DP for $X, X' = X \cup x^*$ with:*

$$\varepsilon = \frac{1}{\alpha-1}\ln(\sum_{k=0}^{\alpha}\binom{\alpha}{k}(1-\mathbb{P}_{x^*}(1))^{\alpha-k}\mathbb{P}_{x^*}(1)^k\exp\frac{\Delta_{U,x^*}^2(k^2-k)}{2\sigma^2})$$

Note that some key variables in Theorem 3.2 are the sampling rate $\mathbb{P}_{x^*}(1)$ (increasing it typically increases the bound), the standard deviation of noise σ (increasing it typically decreases the bound), and the sensitivity upper-bound over mini-batches Δ_{U,x^*} (increasing it typically increases the bound). The proof strategy is analogous to Mironov et al. [34] and replaces their sensitivity upper-bound with the per-instance bound Δ_{U,x^*} on the mini-batches.

Proof Sketch: The proof follows by noting the density function for $M(X')$ can be written as a convex combination of $M(X)$ and a translated version of $M(X)$. One then proceeds to apply the quasi-convexity of Rényi divergences, and direct calculations with the Gaussian density function and the symmetry between the terms (due to translation). The full proof of Theorem 3.2 is in Appendix A.3.

3.2.2 A Generalized Rényi-DP Composition

With now an analysis for the per-step guarantees from DP-SGD (which as currently implemented uses the sum update rule), we now resolve how to obtain a per-instance RDP bound for a full training run with DP-SGD without the limitations of past composition theorem (see Section 5.1 for a discussion on past composition bounds). In particular, we provide a composition theorem that bounds the overall per-instance privacy

[3] We will later turn to Rényi-DP which provides both inequalities.

leakage by the "expected" per-instance privacy guarantee at each step when training on a given dataset. This is presented in Theorem 3.3.

More technically, we once again generalize the classical analysis to look at arbitrary L_p norms, but now for the composition step. The classical Rényi DP composition theorem implicity uses the L_∞ norm of the distribution of per-step guarantees at each step (coming from the distribution of possible models at each step as training is random), and Theorem 3.3 generalizes this to arbitrary L_p norms of the exponential of the per-step guarantees (with some constants to scale). By using L_p norms with $p < \infty$ we take advantage of cases where many models have better privacy guarantees than the worst model.

Theorem 3.3. *Let $p \in (1,\infty)$ and consider a sequence of functions $X_1(x_1)$, $X_2(x_1,x_2)$, $X_3(x_2,x_3)$, $\cdots X_n(x_{n-1},x_n)$ where X_i is a density function in the second argument for any fixed value of the first argument, except X_1 which is a density function in x_1. Consider an analogous sequence $Y_1(x_1),\cdots,Y_n(x_{n-1},x_n)$. Then letting $X = \prod_{j=1}^n X_j$ be the density function for a sequence $x_1,\cdots,x_n$ generated according to the Markov chain defined by X_i, and similarly Y, we have*

$$D_\alpha(X||Y) \leq \frac{1}{\alpha-1}(\sum_{i=0}^{n-2}\frac{(p-1)^i}{p^{i+1}} \ln(\mathbb{E}_{X_1,\cdots X_{n-(i+1)}}(e^{(g_p^i(\alpha)-1)D_{g_p^i(\alpha)}(X_{n-i}||Y_{n-i})p}))) + \frac{1}{\alpha-1}(\frac{p-1}{p})^{n-1}(g_p^{n-1}(\alpha)-1)D_{g_p^{n-1}(\alpha)}(X_1||Y_1) \quad (2)$$

where $g_p(\alpha) = \frac{p}{p-1}\alpha - \frac{1}{p}$ and g_p^i is g_p composed i times, where we defined $g_p^0(\alpha) = \alpha$.

Note some key variables in Theorem 3.3 are a flexible parameter p (which we'll soon describe leads to blow-up as it gets smaller), and the distribution of per-step guarantees $D_{g_p^i(\alpha)}(X_{n-i}||Y_{n-i})$ (the more concentrated at 0 they are, the smaller the upper-bound). The proof relies on using an induction argument to continually break up the composition and is presented below.

Proof. The proof follows by repeating a similar reduction as Theorem A.2. First note

$$\begin{aligned}\int (X_1\cdots X_n)^\alpha (Y_1\cdots Y_n)^{1-\alpha}dx_1\cdots dx_n \\ = \int (X_1\cdots X_{n-1})^{\alpha-1/p}(Y_1\cdots Y_{n-1})^{1-\alpha} \\ (\int X_n^\alpha Y_n^{1-\alpha}dx_n)(X_1\cdots X_{n-1})^{1/p}dx_1\cdots dx_{n-1} \\ \leq (\int (X_1\cdots X_n)^{\frac{p}{p-1}\alpha-\frac{1}{p-1}}(Y_1\cdots Y_n)^{\frac{p}{p-1}(1-\alpha)}dx_1\cdots dx_{n-1})^{\frac{p-1}{p}} \\ (\int(\int X_n^\alpha Y_n^{1-\alpha}dx_n)^p(X_1\cdots X_{n-1})dx_1\cdots dx_{n-1})^{1/p} \quad (3)\end{aligned}$$

where the first equality was from using the markov property, and the last inequality was from Holder's inequality with Holder constant p. Do note that, defining $g_p(\alpha) = \frac{p}{p-1}\alpha - \frac{1}{p-1}$, we have $\frac{p}{p-1}(1-\alpha) = 1 - g_p(\alpha)$. So now looking at the first term of the upper-bound we got, we are back to the original expression but with $\alpha \to g_p(\alpha)$ and $n \to n-1$, and an exponent to $\frac{p-1}{p}$. Note the second term is an expectation over the $n-1$ model state of the Markov chain. Do note $\int X_n^\alpha Y_n^{1-\alpha}dx_n$ is $e^{(\alpha-1)D_\alpha(X_{n-i}||Y_{n-i})}$ for a fixed $n-1$ model state (i.e., fixed x_{n-1}). So repeating this step on the first term until we are left only with an integral over x_1 we have

$$\begin{aligned}\int (X_1\cdots X_n)^\alpha (Y_1\cdots Y_n)^{1-\alpha}dx_1\cdots dx_n \\ \leq (\prod_{i=0}^{n-2}(\mathbb{E}_{X_1,\cdots X_{n-(i+1)}}((e^{(g_p^i(\alpha)-1)D_{g_p^i(\alpha)}(X_{n-i}||Y_{n-i})})^p))^{\frac{(p-1)^i}{p^{i+1}}}) \\ ((e^{(g_p^{n-1}(\alpha)-1)D_{g_p^{n-1}(\alpha)}(X_1||Y_1)})^p)^{\frac{(p-1)^{n-1}}{p^n}} \quad (4)\end{aligned}$$

So now noting

$$D_\alpha(X||Y) = \frac{1}{\alpha-1}\ln(\int (X_1\cdots X_n)^\alpha (Y_1\cdots Y_n)^{1-\alpha}dx_1\cdots dx_n)$$

we conclude by the previous expression that

$$\begin{aligned}D_\alpha(X||Y) \\ \leq \frac{1}{\alpha-1}(\sum_{i=0}^{n-2}\frac{(p-1)^i}{p^{i+1}} \\ \ln(\mathbb{E}_{X_1,\cdots X_{n-(i+1)}}((e^{(g_p^i(\alpha)-1)D_{g_p^i(\alpha)}(X_{n-i}||Y_{n-i})p}))) \\ + \frac{1}{\alpha-1}((\frac{(p-1)^{n-1}}{p^n})\ln((e^{(g_p^{n-1}(\alpha)-1)D_{g_p^{n-1}(\alpha)}(X_1||Y_1)})^p)) \\ (5)\end{aligned}$$

which completes the proof as the last term simplifies to the term stated in the theorem. □

Applying to DP-SGD. To interpret Theorem 3.3 in the context of DP-SGD, we can let X_i be the distribution of the $i'th$ model update (for a fixed $(i-1)'th$ model) when training on one dataset D, and similarly Y_i when training on a neighbouring dataset D'. Letting $Train_{DP-SGD}$ denote the Markov chain of the intermediate model updates when using DP-SGD, we have the maximum over the bound given by Theorem 3.3 on $D_\alpha(Train_{DP-SGD}(D)||Train_{DP-SGD}(D'))$ and $D_\alpha(Train_{DP-SGD}(D')||Train_{DP-SGD}(D))$ provides our per-instance RDP guarantee for DP-SGD.

Balancing the value of p. To understand the dependence on p in Theorem 3.3, consider for a moment $p = 2$. In this case, we observe that at the i'th step, we need to compute a Rényi divergence of order $\sim 2^i\alpha$. It is known that the Rényi divergence $D_c(P||Q)$ grows with c [42], and in the case of the Gaussian mechanism, this growth is linear with c [33].

Hence this exponential growth in the Rényi divergence order can prove impractical as a useful tool to analyze DP-SGD. However, as $p \to \infty$ we see that the growth on the order of the divergence shrinks.

Yet, by taking larger p values we are effectively taking larger L_p-norms of the per-step guarantees seen in training and so effectively turn to worst-case per-step analysis as $p \to \infty$. Hence it is desirable to choose p just sufficient for there to not be a significant blow-up in the order of the divergences for a given n. This can be done by analyzing how $g_p^i(\alpha)$ grows.

Fact 3.4. *If $p = O(n)$ then $g_p^i(\alpha) \leq 2\alpha \ \forall i \leq n$. In particular, $p = 3n$ works for sufficiently large n.*

The proof follows from direct calculations with the formula for $g_p(\alpha)$.

Proof. Note that $g_p(\alpha) \leq \frac{p}{p-1}\alpha$ hence $g_p^i(\alpha) \leq (\frac{p}{p-1})^i\alpha$. From this we see showing $\frac{p}{p-1}^n \leq 2$ for $p = O(n)$ will imply $g_p^i(\alpha) \leq 2\alpha \ \forall 1 \leq n$.

Note we can equivalently show $ln(\frac{p}{p-1}) = \ln(p) - \ln(p-1) \leq \frac{\ln(2)}{n}$. But if we take $p = 3n$ note $\ln(3n) - \ln(3n-1) \leq \frac{1}{3n-1}$ by the derivative of $\ln(x) \leq \frac{1}{3n-1}$ for $x \geq 3n-1$. So it suffices to show $\frac{1}{3n-1} \leq \frac{\ln(2)}{n}$, but this is true for sufficiently large n.

□

Estimating Theorem 3.3

In cases where one does not know the expectations used in Theorem 3.3 analytically, as is the case with DP-SGD when it is applied to deep learning, one can resort to empirically estimating the means. Our goal is to understand how much better our data-dependent guarantees are than the data-independent baseline for DP-SGD on common datasets. Hence, we wish to estimate the expression of Theorem 3.3 (or specifically the per-step contributions) with an error $c\varepsilon$ for $c < 1$ where ε is the data-independent guarantee (per-step).

The following fact focuses on estimating the $i'th$ per-step guarantee with an error relative to the worst-case per-step guarantee when $p = 3n$ as is used in our experiments. In particular, letting $f := (e^{(g_p^i(\alpha)-1)D_{g_p^i(\alpha)}(X_{n-i}||Y_{n-i})})^p$ we have the $i'th$ per-step guarantee is $\frac{1}{\alpha-1}\frac{(p-1)^i}{p^{i+1}}\ln(\mathbb{E}_{X_1,\cdots,X_{n-(i+1)}}f)$ and is less than the data-independent per-step privacy guarantee ε/n if $\mathbb{E}_{X_1,\cdots X_{n-(i+1)}}f \leq e^{(\alpha-1)3\varepsilon}$ for $p = 3n$. Hence we describe the number of samples needed to estimate $\mathbb{E}f$ with precision relative to $e^{(\alpha-1)3\varepsilon}$ (with high probability), which can be done in a constant number of samples relative to the data-independent bound.

Fact 3.5. *Let ε/n be the classical α-Rényi DP guarantee for the $i'th$ step, and ε'/n be the analogous 2α-Rényi DP guarantee for the $i'th$ step. Then for $l \geq \frac{-\ln(J)}{c^2}e^{6(\alpha-1)\varepsilon-3(2\alpha-1)\varepsilon'}$ and $p = 3n$ with n s.t $g_p^{n-1} \leq 2\alpha$, we have $\mathbb{P}(|\mathbb{E}^l f - \mathbb{E}f| \geq ce^{(\alpha-1)3\varepsilon}) \leq J$. Here $\mathbb{E}^l$ denotes the empirical mean over l samples.*

The proof follows from Hoeffding's inequality.

Proof. For the given choice of p and α we have $g_p^i \leq 2\alpha$ hence $D_{g_p^i(\alpha)}(X_{n-i}||Y_{n-i}) \leq D_{2\alpha}(X_{n-i}||Y_{n-i}) \leq \varepsilon'/n$ where ε' is determined by ε (when accounting for the increase due to the α-order). Hence we have that $f \leq e^{3(2\alpha-1)\varepsilon'}$.

By Hoeffding's inequality we can hence conclude $\mathbb{P}(|\mathbb{E}^l f - \mathbb{E}f| \geq ce^{3(\alpha-1)\varepsilon}) \leq e^{-\frac{e^{6(\alpha-1)\varepsilon}c^2 l}{e^{3(2\alpha-1)\varepsilon'}}}$. Now upper-bounding the right-hand side by J and rearranging to isolate for l, we can conclude the stated condition on l.

□

3.2.3 Per-Instance Rényi DP for General Updates

The results of Section 3.2.1 and Section 3.2.2 provide a complete per-instance RDP analysis of the current implementation of DP-SGD. In particular, with the per-step update rule being the sum of gradients. In this section we ask, how should we analyze per-step guarantees (and hence DP-SGD given our composition theorem) if the update rule is not the sum? In general, the worst-case sensitivity over mini-batches may be far higher than the expected sensitivity over mini-batches (unlike the sum update rule), meaning the analysis from Theorem 3.2 may be as bad as a data-independent analysis. For example, the typical update rule used in normal SGD is the mean update rule. However, $\Delta_{U,x^*}(X_B)$ for the mean update rule is the difference between the update for the datapoint x^* and the mean of the updates on X_B; this difference is not the same for all minibatches X_B and hence would be overestimated with the analysis of Theorem 3.2. One could resolve this issue of overestimating sensitivity by using the L_p norms $||\Delta_{U,x^*}(\mathbf{X_B})||_p = (\mathbb{E}_{X_B}(\Delta_{U,x^*}(X_B)^p))^{1/p}$ with $p < \infty$ in the RDP analysis of the sampled Gaussian mechanism, as was done in the (ε,δ)-DP case. However, we are not aware of an approach to do this for Rényi DP.

Instead, we show how a new sensitivity distribution comparing all mini-batches X_B in X to all mini-batches X'_B in $X' = X \cup \{x^*\}$, as opposed to just a single point x^* as done with $\Delta_{U,x^*}(X_B)$, is amenable to a Rényi-DP analysis of the sampled Gaussian mechanism that does not look at the maximum privacy leakage over mini-batches. If the distribution of all updates given by X is similar to the distribution of all updates given by X', then analysis with this new sensitivity distribution can be expected to beat the current data-independent analysis.

Specifically, given α minibatches sampled from X, $X_B{}^{\tilde{\alpha}} \sim X$, and a particular minibatch sampled from X', $X'_B \sim X'$, we define a new sensitivity distribution for α-Rényi DP as follows:

$$\Delta_{U,\alpha}(X_B{}^{\tilde{\alpha}}, X'_B)$$
$$:= \sum_i ||U(X_B{}^i)||_2^2 - (\alpha-1)||U(X'_B)||_2^2 - ||\Delta_\alpha(X_B{}^{\tilde{\alpha}}, X'_B)||_2^2$$

where $\Delta_\alpha({X_B}^{\tilde{\alpha}}, X'_B) = (\sum_i U({X_B}^i)) - (\alpha-1)U(X'_B)$. When letting ${X_B}^{\tilde{\alpha}}$ and X'_B be random variables, $\Delta_{U,\alpha}$ effectively compares all the mini-batches in X' to all the α-tuples of mini-batches in X. The α-tuples appear here due to their equivalence with an expectation over mini-batches to the power of α which appears when analyzing α-Rényi divergences. As described earlier, comparing this to the previous sensitivity distribution $\Delta_{U,x^*}(X_B)$, we see that this new sensitivity will compare all mini-batches in X to all mini-batches in X' (and not just to a point x^*) and hence captures more "global" changes in updates due a datapoint x^*.

Theorem 3.6 states the Rényi diveregence of the sampled Gaussian mechanism M between two arbitrary datasets using $\Delta_{U,\alpha}$ through applying a transformation on its fixed X'_B marginal values and taking its expectation over X'_B. Taking the maximum of the bounds for $D_\alpha(M(X)||M(X'))$ and $D_\alpha(M(X')||M(X))$ from Theorem 3.6 where $X' = X \cup \{x^*\}$ gives a per-instance guarantee of M for X, X'.

Theorem 3.6. *Let $\alpha > 1$ be an integer. Given two arbitrary datasets X, X', the sampled Gaussian mechanism M with noise σ satisfies:*

$$D_\alpha(M(X')||M(X)) \leq \frac{1}{(\alpha-1)} \mathbb{E}_{X_B}(\ln(\mathbb{E}_{{X'_B}^{\tilde{\alpha}}}(e^{\frac{-1}{2\sigma^2}\Delta_{U,\alpha}({X'_B}^{\tilde{\alpha}}, X_B)})))$$

Some key variables in Theorem 3.6 is the standard deviation of noise σ (increasing it decreases the upper-bound) and the sensitivity distribution $\Delta_{U,\alpha}({X_B}^{\tilde{\alpha}}, X'_B)$ (the more concentrated at 0 it is, the smaller the upper-bound). The proof relies on convexity, which is always true for the second argument of the Rényi divergence $D_\alpha(A||B)$, and then direct calculations involving Gaussians.

Proof. For simpler notation, we use $\mu_X = U(X)$. We proceed by taking α to be an integer (to use an expansion similar to Section 3.3 in Mironov et al. [34]) and utilizing Theorem 12 in Van Erven and Harremos [42]. We will let $N_{X_B} = N(\mu_{X_B}, \sigma^2)$ where $\mu_{X_B} = U(X_B)$ as stated earlier.

We proceed to bound $D_\alpha(M(X')||M(X))$ for arbitrary X', X. Hence a completely analogous argument will allow us to also bound $D_\alpha(M(X)||M(X'))$ when X' is specifically $X \cup \{x^*\}$. First note

$$\begin{aligned} D_\alpha(M(X')||M(X)) &= D_\alpha(\sum_{X'_B} \mathbb{P}(X'_B) N_{X'_B} || \sum_{X_B} \mathbb{P}(X_B) N_{X_B}) \\ &\leq \sum_{X_B} \mathbb{P}(X_B) D_\alpha(\sum_{X'_B} \mathbb{P}(X'_B) N_{X'_B} || N_{X_B}) \end{aligned} \tag{6}$$

where the last inequality is from the fact the divergence is convex in the second argument (Theorem 12 in Van Erven and Harremos [42]).

Now note

$$\begin{aligned} & e^{(\alpha-1)D_\alpha(\sum_{X'_B} \mathbb{P}(X'_B) N_{X'_B} || N_{X_B})} \\ &= \int (\sum_{X'_B} \mathbb{P}(X'_B) \frac{1}{(\sigma\sqrt{2\pi})^d} e^{\frac{-1}{2\sigma^2}|x-\mu_{X'_B}|^2})^\alpha \\ &\quad (\frac{1}{(\sigma\sqrt{2\pi})^d} e^{\frac{-1}{2\sigma^2}|x-\mu_{X_B}|^2})^{1-\alpha} dx \\ &= \sum_{{X'_B}^{\tilde{\alpha}}} \mathbb{P}({X'_B}^{\tilde{\alpha}}) \frac{1}{(\sigma\sqrt{2\pi})^d} \int e^{\frac{-1}{2\sigma^2}((\sum_{{X'_B}^i} |x-\mu_{{X'_B}^i}|^2) - (\alpha-1)|x-\mu_{X_B}|^2)} \end{aligned} \tag{7}$$

where we expanded $(\sum_{X'_B} \mathbb{P}(X'_B) \frac{1}{(\sigma\sqrt{2\pi})^d} e^{\frac{-1}{2\sigma^2}|x-\mu_{X'_B}|^2})^\alpha$ by noting each term in the product is just iterating through all α tuples of mini-batches from X'.

Note we can for now consider the integral in each dimension, as the overall integral is the product of each dimension. Also recall from the theorem statement that we define

$$\Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B) = (\sum_i \mu_{{X'_B}^i}) - (\alpha-1)\mu_{X_B}$$

Hence (letting everything be one dimensional for now) we have

$$\begin{aligned} & (\sum_{{X'_B}^i} |x-\mu_{{X'_B}^i}|^2) - (\alpha-1)|x-\mu_{X_B}|^2 \\ &= x^2 - 2\Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B)x + \sum_i \mu^2_{{X'_B}^i} - (\alpha-1)\mu^2_{X_B} \\ &= (x-\Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B))^2 + \sum_i \mu^2_{{X'_B}^i} - (\alpha-1)\mu^2_{X_B} - \Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B)^2 \end{aligned} \tag{8}$$

Hence, we have

$$\begin{aligned} & \int e^{\frac{-1}{2\sigma^2}((\sum_{{X'_B}^i} |x-\mu_{{X'_B}^i}|^2) - (\alpha-1)|x-\mu_{X_B}|^2)} \\ &= e^{\frac{-1}{2\sigma^2}(\sum_i {\mu_{{X'_B}^i}}^2 - (\alpha-1)\mu^2_{X_B} - \Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B)^2)} \int e^{\frac{-1}{2\sigma^2}(x-\Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B))^2} \\ &= \sigma\sqrt{2\pi} e^{\frac{-1}{2\sigma^2}(\sum_i \mu^2_{{X'_B}^i} - (\alpha-1)\mu^2_{X_B} - \Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B)^2)} \end{aligned} \tag{9}$$

Note going back to the integral over all dimensions we get $= (\sigma\sqrt{2\pi})^d e^{\frac{-1}{2\sigma^2}(\sum_i ||\mu_{{X'_B}^i}||_2{}^2 - (\alpha-1)||\mu_{X_B}||_2^2 - ||\Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B)||_2^2)}$.

Thus to conclude we get

$$\begin{aligned} D_\alpha(M(X')||M(X)) &\leq \sum_{X_B} \mathbb{P}(X_B) D_\alpha(\sum_{X'_B} \mathbb{P}(X'_B) N_{X'_B} || N_{X_B}) \\ &= \sum_{X_B} \mathbb{P}(X_B) \frac{1}{(\alpha-1)} \\ & \ln(\sum_{{X'_B}^{\tilde{\alpha}}} \mathbb{P}({X'_B}^{\tilde{\alpha}}) e^{\frac{-1}{2\sigma^2}(\sum_i ||\mu_{{X'_B}^i}||_2^2 - (\alpha-1)||\mu_{X_B}||_2^2 - ||\Delta_\alpha({X'_B}^{\tilde{\alpha}}, X_B)||_2^2)}) \end{aligned} \tag{10}$$

A completely analogous calculation gives the same bound with just X_B replaced with X'_B (and vice-versa) for $D_\alpha(M(X)||M(X'))$. Taking the max over both these divergences gives a bound on the per-step per-instance Rényi-DP guarantee.

□

Hence we now have a per-step RDP analysis for DP-SGD that takes advantage of when expected minibatch sensitivity to x^* is much better than the worst cast minibatch sensitivity. While this phenomenon is not useful for studying the sum update rule (what is currently used for DP-SGD) as every mini-batch has the same sensitivity to x^*, in Section 4.2 we show this analysis allows us to provide a tighter analysis of the mean update rule. Hence, this opens the possibility of future work deploying DP-SGD with different update rules.

4 Empirical Results

In Section 3 we provided the first framework to analyze DP-SGD's per-instance privacy guarantees. This followed by providing new per-step analyses (Theorem 3.2 and 3.6), and a new composition theorem that relies on summing "expected" per-step guarantees (Theorem 3.3). We now highlight several conclusions our framework allows us to make about per-instance privacy when using DP-SGD. For conciseness, we defer a subset of the experimental results to Appendix B.

Experimental Setup. In the subsequent experiments, we apply our analysis on MNIST [29] and CIFAR-10 [27]. Unless otherwise specified, LeNet-5 [30] and ResNet-20 [21] were trained on the two datasets for 10 and 200 epochs respectively using DP-SGD, with a mini-batch size equal to 128, $\varepsilon = 10$, $\delta = 10^{-5}$, $\alpha = 8$ (in cases of Rényi DP), and clipping norm $C = 1.0$. All the experiments are repeated 100 times by sampling 100 data points to obtain a distribution/confidence interval if not otherwise stated. Regarding hardware, we used NVIDIA T4 to accelerate our experiments.

Data Access Assumptions We now clarify the data access assumptions needed to run our methods. Theorem 3.2 for the sum update rule only needs the individual datapoint x and the model, and the composition theorem only additionally needs the checkpoints obtained during training. Hence, as one only needs to compute Theorem 3.2 and then plug those values in our composition to obtain per-instance guarantees, computing the per-instance DP guarantee for DP-SGD does not require access to the underlying dataset but only the checkpoints and the point x^* in question (applicable for the results in Section 4.1). However, Theorem 3.6 requires sampling minibatches from the datasets, hence our approach to analyze the mean update rule requires further access to the whole dataset (applicable for the results in Section 4.2).

4.1 Many Datapoints have Better Privacy

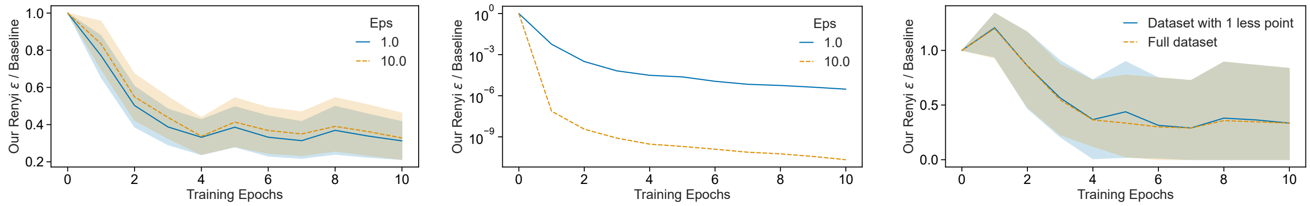
Here we describe how our per-instance RDP analysis of DP-SGD, using Theorem 3.2 for the per-step analysis (with the update rule being the sum of gradients as is typically used) and Theorem 3.3 for the composition analysis, allows us to explain why per-instance privacy attacks will fail for many datapoints: many points have better per-instance privacy than the data-independent analysis. We further investigate the distribution of the per-instance privacy guarantees, and which points exhibit better per-instance privacy with our analysis.

Improved Per-Instance Analysis for Most Points We compare the guarantees given by Theorem 3.2 for the per-step guarantee in DP-SGD to the guarantee given by the data-independent analysis (see Section 3.3 in Mironov et al. [34]), and plot per-step contribution coming from our composition theorem. In particular, we take X to be the full MNIST training set, and randomly sample a data point x^* from the test set to create $X' = X \cup x^*$ (as mentioned earlier, we repeat the sampling of x^* 100 times to obtain a confidence interval). We train 10 different models on X with the same initialization and compute the per-step contribution from Theorem 3.3 between X and X' (using Theorem 3.2 to analyze the per-step guarantee from a given model) over the training run, shown in Figure 1a. We can see that our analysis of the per-step contribution decreases with respect to the baseline as we progress through training. This persists regardless of the expected mini-batch size, the strength of DP used during training, and model architectures; see Figure 6 in Appendix B. By Theorem 3.3 we conclude that $D_\alpha(Train_{DP-SGD}(X)||Train_{DP-SGD}(X'))$ is significantly less than the baseline for many data points.

To see our improvement over the max of $D_\alpha(Train_{DP-SGD}(X)||Train_{DP-SGD}(X'))$ and $D_\alpha(Train_{DP-SGD}(X')||Train_{DP-SGD}(X))$, i.e., the Rényi-DP guarantee, we computed the expectation when training on X and $X' = X \cup \{x^*\}$ for 10 training points x^* where X is now the training set of MNIST with one point removed and X' is the full training set. Our results are shown in Figure 1c where we see a similar decreasing trend relative to the baseline over training: we conclude by Theorem 3.3 that many datapoints have better per-instance Rényi DP than the baseline. In other words, we conclude many datapoints have stronger per-instance RDP guarantees than can be demonstrated through the classical data-independent analysis.

Long-Tail of Better Per-Instance Privacy. However, the previous figures only show the average effect over datapoints. In Figures 2a and 2b we plot the distribution of per-step guarantees over 500 data points in CIFAR10. The key observations are (1) there exists a long tail of data points with significantly better per-instance privacy than the baseline illustrated by the log-scale in Figures 2a and 2b, (2) such improvements mostly exist in the later half of the training process, and (3) such improvements are mostly independent of mini-batch size.

Correct Points are More Per-Instance Private. Next, we



(a) Training with the datapoint $(X \cup \{x^*\})$

(b) Training with the datapoint $(X \cup \{x^*\})$ (10^{th} percentile)

(c) Training with and without the datapont ($X \cup \{x^*\}$ and X)

Figure 1: Per-step privacy contribution from our composition theorem (Theorem 3.3) using the per-step guarantee for the sum update rule (Theorem 3.2) as needed for DP-SGD, plotted as a fraction of the baseline data-independent per-step DP-SGD guarantee (Section 3.3 in Mironov et al. [34]). The x-axis represents the release of the intermediate models up to a given step in training. The y-axis represents the per-instance privacy leakage for a point given by the release of the model at that training step (relative to the data-independent guarantee); summing all the steps gives the overall privacy leakage of training. The different lines represent changing the Gaussian noise to train with different data-independent ε, δ-DP values. The expectations for Theorem 3.3 are computed over 10 trials. Figure 1a plots the average relative per-step contribution of 100 random points in MNIST for different strengths of the DP guarantee (i.e., different upper bounds ε) used when training on $X' = X \cup \{x^*\}$. The 10^{th} *percentile* is plotted in Figure 1b. Figure 1c plots expectation over 10 random points in MNIST when training on X' and X. We see from both subfigures our per-step contribution more tightly captures the per-instance privacy than the baseline as training progresses: using Theorem 3.3 one can conclude that many datapoints have better overall data-dependent privacy guarantees than expected by classical analysis. Note our analysis does worse at the first few steps of training as our composition theorem has a blow up in the order of the Rényi divergence for the per-step guarantee for early steps of training; if the sensitivity does not drop quickly enough our composition theorem accounts higher privacy leakage to early steps than the data-independent bounds.

turn to understanding what datapoints are experiencing better privacy when using DP-SGD. In Figure 3, we plot the per-step guarantees given by Theorem 3.2 for correctly and incorrectly classified data points at the beginning, middle, and end of training on CIFAR10 (and for MNIST in Figure 7 in Appendix B). We see that, on average, correctly classified data points have better per-step privacy guarantees than incorrectly classified data points across training. This disparity holds most strongly towards the end of training.

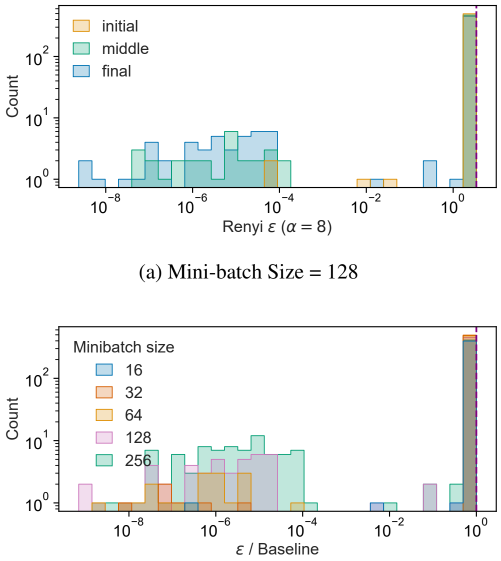
4.2 Higher Sampling Rates can give Better Privacy

We now highlight how our analysis, if it uses Theorem 3.6 for the per-step analysis, allows us to better analyze DP-SGD with other update rules (not the sum of gradients which is what the current implementation of DP-SGD uses and Section 4.1 analyzed). In particular, we will analyze the mean update rule and show how it has a privacy trade-off with sampling rate that is opposite to the trade-off for the sum update rule.

In normal SGD (with gradient clipping), one computes a mean for the per-step update $U(X_B) = \frac{1}{|X_B|}\sum_{x \in X_B} \nabla_\theta \mathcal{L}(\theta,x)/\max(1, \frac{||\nabla_\theta \mathcal{L}(\theta,x)||_2}{C})$. However, DP-SGD computes a weighted sum $U(X_B) = \frac{1}{L}\sum_{x \in X_B} \nabla_\theta \mathcal{L}(\theta,x)/\max(1, \frac{||\nabla_\theta \mathcal{L}(\theta,x)||_2}{C})$. Note the subtle difference between dividing by a fixed constant L (typically the expected mini-batch size when Poisson sampling datapoints) and by the mini-batch size $|X_B|$. This means for the sum the upper-bound on sensitivity is $\frac{C}{L}$, while for the mean the upper-bound on sensitivity is only C (consider neighbouring mini-batches of size 1 and 2). Hence using the mean update rule requires far more noise and so is not practical to use. We highlight how our per-instance analysis by sensitivity distributions provides better guarantees for the mean update rule.

Better Analysis of the Mean Update Rule. Letting M now be the sampled Gaussian mechanism with the mean update rule, we compute the bound on $D_\alpha(M(X')||M(X))$ and $D_\alpha(M(X)||M(X'))$ given by Theorem 3.6, where we estimated the inner and outer expectation using 20 samples, i.e., 20 random $X_B'^{\alpha}$ (or X_B^{α}) for each of the 20 random X_B (or X_B'). We obtain Figure 4a and 4c by repeating this for 500 data points in CIFAR10 while varying the training stage. We observe that for both divergences, we beat the baseline analysis by more than a magnitude at the middle and end of training. We conclude Theorem 3.6 gives us better per-instance Rényi DP guarantees for the mean update rule.

Per-Instance Privacy Improves with Higher Sampling Rate. Furthermore, counter-intuitively to typical subsample privacy amplification, in Figure 4c we see that our bound decreases with increasing expected mini-batch size: we attribute this to the law of large numbers, whereby increasing the expected mini-batch size leads to sampled mini-batches having similar updates more often and hence the sensitivity distribution concentrates at smaller values. An analogous result is shown for MNIST in Figure 8 (in Appendix B).



(a) Mini-batch Size = 128

(b) Varying Mini-batch Size

Figure 2: Distribution plots of our per-step guarantees for the sum update rule given by Theorem 3.2 for 500 datapoints in CIFAR10 with respect to: (a) different stages of training, and (b) varying mini-batch size. The x-axis represents the per-instance guarantee relative to the data-independent guarantee: i.e., the further the mass is to the left, the more our data-dependent guarantees improves upon the data-independent baseline. The purple dashed line represents the data-independent baseline. We observe a "long tail" of datapoints with magnitudes better privacy than expected in both plots, illustrated by the log scale on the x-axis.

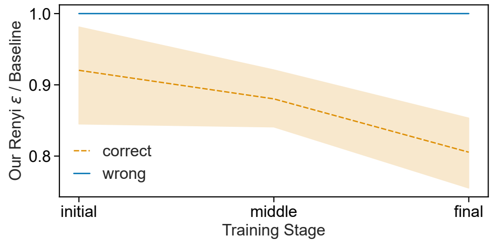
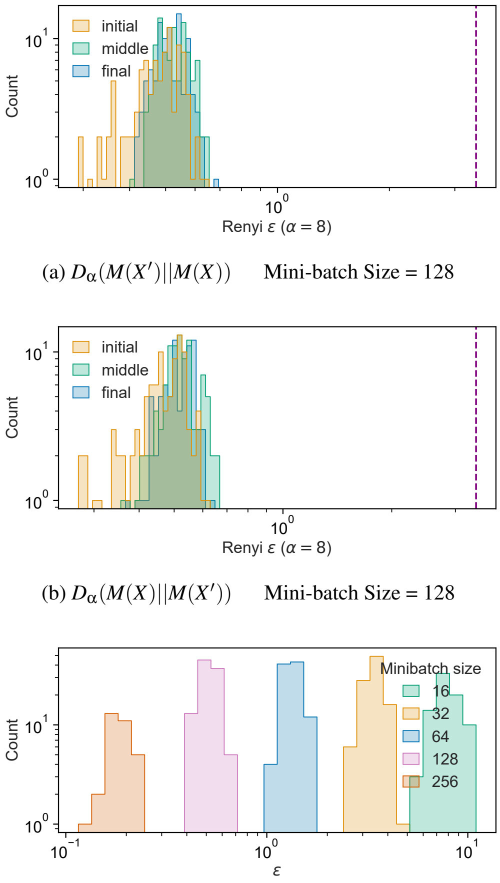


Figure 3: Per-step guarantees given by Theorem 3.2 for 500 datapoints in CIFAR10 across training stages with respect to correct or incorrect classifications. It can be seen that correctly classified datapoints are on average more private than incorrectly classified ones.

5 Discussion

Here we first discuss past work on composition theorems (Section 5.1) and the current computational trade-offs of our analysis which future work could improve (Section 5.2). We then discuss some theoretical questions based on observations from our analysis (Section 5.3). Lastly, we describe several applications of our analysis (Section 5.4).



(a) $D_\alpha(M(X')||M(X))$ Mini-batch Size = 128

(b) $D_\alpha(M(X)||M(X'))$ Mini-batch Size = 128

(c) $D_\alpha(M(X')||M(X))$ Varying Mini-batch Size

Figure 4: Distribution plots (log scale) of our per-step guarantees for the mean update rule (Theorem 3.6) for 500 datapoints in CIFAR10 with respect to different training stages and mini-batch sizes. Bounds on both $D_\alpha(M(X)||M(X'))$ and $D_\alpha(M(X')||M(X))$ are shown (the maximum of both is the per-instance Rényi-DP guarantee) for an expected mini-batch size of 128. From Figures 4a,4b, we conclude our per-step guarantees for the mean update rule (Theorem 3.6) gives better data-dependent guarantees for the mean update rule than classical analysis, and from Figure 4c that increasing the expected mini-batch size decreases our bound for this update rule (counter-intuitive to privacy amplification by subsampling).

5.1 Fully Adaptive Composition Theorems

One of the main technical contributions of this paper is generalizing the normal Rényi DP composition theorem (Proposition 1 in Mironov [33]), which sums worst-case per-step guarantees, to allow for better per-instance analysis. Other work have also generalized the composition theorem to have better per-instance analysis [15, 26], and called these new theorems Fully Adaptive Composition. For Rényi DP, Feldman and Zrnic [15] showed that composition can be done by considering the worst-case sum of the per-step guarantees from

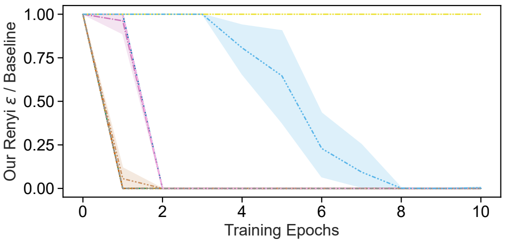


Figure 5: The expected reweighted per-step contributions which are summed for our composition theorem (Theorem 3.3) using Theorem 3.2 for the unweighted per-step guarantee for 10 different points in MNIST. The guarantees are computed once each epoch when training with the datapoint (i.e., on $X' = X \cup \{x^*\}$). The shaded region is the 95% confidence interval over 10 trials. As seen by the confidence intervals having a width a small fraction of the baseline value, with just 10 trials we are very confident in the estimates of the per-step contributions for most points.

a DP-SGD training run (Theorem 3.1 in Feldman and Zrnic [15]), as opposed to summing the worst-case guarantee at each step. Koskela et al. [26] state an analogous composition theorem for Gaussian DP. However, for DP-SGD, the degree of improvement provided by the worse-case sum compared to the normal composition is not clear. It could be that the worst-case sum is equal to the sum of the worst-case per-step guarantees, which is the case if the training run goes to worst-case states with non-zero probability at each step. Furthermore, it is hard to measure the worst-case sum to show this is not the case. Specifically, it requires measuring the L_∞ norm of a distribution, which without further assumptions beyond boundedness is much harder than the p-norms needed for our method. In short, we are the first to provide a per-instance composition theorem that can be used to determine better per-instance guarantees for DP-SGD.

5.2 Computational Limitations and Future Improvements

As explained in Section 3.2.2, there is a tension between preventing blow-up in our composition theorem (Theorem 3.3) and estimating the per-step contributions with few samples: both require manipulating a parameter p, with the former requiring large p and the latter requiring small p. We showed in Section 3.2.2 how the value of p we chose for our experiments strikes a balance where we can limit blow-up while still estimating the per-step contribution to the composition with few samples. This balance is further backed up by the confidence intervals for our estimates of the per-step contributions (see Figure 5). Nevertheless, we still require several training runs to compute the per-instance guarantee for a specific point.

Future work may be able to improve (analytically) the trade-off between blow-up and sample complexity for Theorem 3.3, and hence make it cheaper to compute the per-instance DP guarantees. Future work may also be able to derive composition theorems analogous to Theorem 3.3 which are easier to estimate. Similarly, Theorem 3.6 is computationally expensive to compute; we require computing several mini-batch updates at every step to estimate the expectations. Future work may be able to derive alternative per-step guarantees for general update rules that are easier to empirically estimate.

5.3 Theoretical Questions

In this paper we applied our analysis of DP-SGD to deep learning. In particular, our analysis led to data-dependent guarantees but stopped short of data-independent guarantees; the technical problem for deriving data-independent guarantees is bounding the "expected per-step guarantee" for all datasets. However, one can still apply our analysis of DP-SGD to classical machine learning. For example, theoretical work has shown improved data-independent privacy guarantees for convex losses [2]; the proof relied on the update rule being a contraction for convex losses. In contrast to this approach, our analysis can directly translate smaller gradient norms (in expectation) during training to better privacy guarantees. Hence, we believe an interesting future direction is applying our analysis to learning settings that permit direct calculations of sensitivity distributions and hence potentially derive data-independent guarantees.

In this direction, a particular phenomenon we wish to highlight is that by training with less noise we sometimes see a disproportional decrease in per-instance privacy consumption. This is shown in Figure 1b where training with the noise level for $\varepsilon = 10$ (data-independent bound) resulted in disproportionately smaller per-step privacy consumption than training with the higher noise level for $\varepsilon = 1$ (data-independent bound) for the 10th percentile of points with the smallest sensitivity. Stated another way, our analysis shows that for some datapoints more noise is not always better for privacy. In settings where sensitivity distributions can be explicitly calculated, one may be able to compute what this noise vs. per-instance privacy trade-off is.

5.4 Applications

In this paper, we focused on explaining how privacy attacks will fail for many datapoints if the adversary only observes typical datasets common to deep learning. This was done by providing a new per-instance DP analysis for DP-SGD. We now highlight other applications of our analysis.

Estimating Privacy A growing theme in private deep learning is empirically “estimating" privacy in different data settings, and is broadly encapsulated by privacy auditing [22, 35, 36, 45]. However, here estimating means obtaining lower bounds on privacy leakage (e.g., the parameter ε used in differential privacy). Our work presents a shift in how we can go about estimating privacy. Our analysis provides per-instance *upper-bounds* that complement past lower-bounds, and when direct calculations of expected sensitivity distributions are not possible, one can still estimate our upper-bounds by repeating training. Future work can hence use our analysis to provide potentially matching empirical upper-bounds to previous empirical lower-bounds obtained with specific privacy attacks, and hence be able to conclude that these privacy attacks are optimal in more settings than just the worst-case.

However, it is possible that the result from auditing can be used in a way that increases the data-independent privacy leakage. Hence, to be more specific, we emphasize two use-cases of per-instance guarantees for auditing and when they retain data-independent differential privacy.

Internal Audit: The first is an internal audit, formally:

1. Input dataset D, auditing dataset D*, training algorithm T
2. Compute auditing statistics $S(D^*, T)$
3. Release $T(D)$

If T is a DP algorithm, we have the outputs of this protocol is private in the training dataset D. In our case, T is DP-SGD and our method provides tools to estimate S when S are per-instance guarantees for $x \in D^*$ (that are better than the data-independent guarantee).

Audit to Modify Training: However, if the audit affects the public release, then we can leak privacy. This leads to the second use-case, stated formally as:

1. Input D, reference algorithm T, final algorithm T'
2. Compute $S(D,T)$
3. Release $T'(D,S)$

An example of a possible T' is computing our per-instance guarantees as S, and then dropping all the least private points in the training set (in the hope of having better privacy guarantees and hence better utility vs. privacy); this is broadly captured by privacy filtering where one defines $T'(D,S) = T(D(S))$ where $D(S) \subset D$. However, note T' will now depend on the training dataset D through S. In this case, to preserve data-independent privacy, one needs to bound the sensitivity of T' to S and the sensitivity of S to D. How T' differs from T is not specified, giving a somewhat ill-posed problem. However, a specific privacy filtering algorithm was studied by Feldman and Zrnic [15] which relied on using per-step per-instance guarantees (not composition of steps). While in this paper we do not provide a filtering algorithm, we believe our composition theorem provides a new tool that future work may use in designing privacy filtering algorithms. However, we remark that at least for the filtering algorithm of Feldman and Zrnic [15], our composition theorem can be worse than their composition theorem; their algorithm enforces the almost every condition for their composition theorem which is then tighter than our composition theorem as it does not have divergence order blow-up for early steps (which ours does). This is in contrast to DP-SGD (see Section 5.1), highlighting the interdependence between algorithm and composition theorem choices.

Estimating Memorization Related to estimating privacy, a broad literature is concerned with measuring memorization [7, 8, 14, 41, 46]. The methodology for estimating memorization varies, but includes privacy attacks [8], or approximations of influence [7, 14]. Our work provides, to the best of our knowledge, the first approach to estimating memorization via upper bounds. Hence, our work may provide a complementary tool to past work on memorization.

Estimating Unlearning Unlearning a datapoint x^* is to obtain the model (distribution) coming from training on the dataset $D \setminus x^*$ given a model trained on the dataset D [6]. The only known methods to do this exactly for deep learning are variations of naively retraining on $D \setminus x^*$ [5]. Given the general intractability of exact unlearning, significant work has looked into *approximate unlearning*; approximate unlearning is to obtain the same model (distribution) as training with $D \setminus x^*$ up to some error in a predefined metric. A popular measure of approximate unlearning has been using per-instance DP guarantees [18], or only one of the per-instance DP inequalities [20][16] (which is implied by the former). However, the only known methods (to the best of our knowledge) to achieve this kind of guarantee for deep learning is to train with DP-SGD and use the data-independent DP bound as the unlearning guarantee. Our analysis allows for unlearning guarantees that are specific to individual points. While DP-SGD does not explicitly target specific points to have better unlearning guarantees, future work may be able to use our analysis to derive a modified version of DP-SGD that explicitly unlearns a subpopulation of the training set (hence future deletion requests for that subpopulation are already handled).

Another influential notion of unlearning is adaptive machine unlearning [20], which requires unlearning to also be private to the sequence of update requests; that is, regardless of the order in which people request for unlearning, one finally returns a model close to retraining without any of their data. Per-instance guarantees naturally leak information about the dataset, and so one might wonder if it is still possible to satisfy adaptive machine unlearning when using per-instance DP guarantees to unlearn. We now illustrate a specific unlearning algorithm using per-instance guarantees to not retrain when possible, which satisfies adaptive machine unlearning. However, this does not immediately give a more efficient unlearning algorithm than retraining. This is because checking the per-instance guarantees for each unlearning request is expensive with our current method. However, we hope this serves as motivation for future work on efficiently computing per-instance guarantees.

We now state a Rényi divergence version of adaptive machine unlearning; Rényi divergence implies the (ε,δ)-DP inequalities used in the original definition [20], while also being consistent with our paper. Note, we use u^i to denote the i^{th} unlearning request in a sequence (only deletion), $D^i = D \setminus \{\cup_{j=1}^{i} u^j\}$, and s for other hyperparameters.

Definition 5.1 (Rényi α, β, γ- adaptive unlearning)**.** We say R_A is a Rényi α, β, γ- adaptive unlearning algorithm for A if for all update request function $UpdReq$, initial datasets D, $t \geq 1$, and with probability $1 - \gamma$ over the draw of unlearning requests $u^1, \cdots, u^t$ from $UpdReq$ we have $D_\alpha(R_A(D^{t-1}, u^t, s^{t-1}) || A(D^t)) \leq \beta$

Our adaptive unlearning algorithm using per-instance guarantees, which we will call Naive Per-Instance Unlearning (NPIU), is defined recursively over the sequence of unlearning requests. Intuitively, it checks if the current distribution of models is far away from the retraining distribution (via a triangle inequality), and if so it retrains from scratch. Formally:

1. If $\frac{\alpha-1}{\alpha-2} D_{2\alpha}(R_A(D^{t-2}, u^{t-1}, s^{t-2}) || A(D^{t-1})) + D_{2\alpha-1}(A(D^{t-1}) || A(D^t)) \leq \beta$ then $R_A(D^{t-1}, u^t, s^{t-1}) = R_A(D^{t-2}, u^{t-1}, s^{t-2})$, i.e., keep same output as before
2. Else $R_A(D^{t-1}, u^t, s^{t-1}) = A(D^t)$, i.e., retrain from scratch

Fact 5.2. *NPIU satisfies* $(\alpha, \beta, 0)$*-adaptive unlearning.*

Proof. Consider any t, D, and update requests $u_1, \cdots, u_t$ (which defines the sequence of datasets). If the first condition is satisfied, then by Corollary 4 in [33] (the weak triangle inequality for Rényi Divergences) we have $D_\alpha(R_A(D^{t-2}, u^{t-1}, s^{t-2}) || A(D^t)) \leq \frac{\alpha-1/2}{\alpha-1} D_{2\alpha}(R_A(D^{t-2}, u^{t-1}, s^{t-2}) || A(D^{t-1})) + D_{2\alpha-1}(A(D^{t-1}) || A(D^t)) \leq \beta$ hence $D_\alpha(R_A(D^{t-1}, u^t, s^{t-1}) || A(D^t) \leq \beta$ and so the unlearning criteria is satisfied at step t.

If not we have $R_A(D^{t-1}, u^t, s^{t-1}) = A(D^t)$ and so $D_\alpha(R_A(D^{t-1}, u^t, s^{t-1}) || A(D^t)) = 0 \leq \beta$ meaning the unlearning criteria is once again satisfied at step t. Hence as we proved the condition holds for arbitrary t, D and sequence of update requests, the algorithm is $(\alpha, \beta, 0)$ adaptive machine unlearning. This completes the proof.

□

Interpreting the algorithm in more detail, we have it checks if the per-instance guarantee $D_{2\alpha-1}(A(D^{t-1}) || A(D^t))$ is small enough while recursively using information on what the per-instance guarantee was already with respect to D^{t-1}; if the sum of the guarantees is over the budget, it retrains from scratch. In particular, on accounting for the budget already used, note for each unlearning request one needs a divergence of two times the original order to implement the algorithm (see the first condition), which could be done by recursively applying Corollary 4 in [33] (as done in the proof) until one hits only divergences between $A(D^{i-1})$ and $A(D^i)$ for some i, where i is at least as large as the last time one hit the else condition and had to rerun A. Our results do not give efficient methods to measure these divergences (though computing different orders can reuse cached data such as checkpoints and gradients), but we hope unlearning motivates further work on efficiently computing per-instance guarantees.

An Alternative Framework for Forgeability However, underlying machine unlearning (as a legal requirement, e.g., the EU GDPR [32]) is the problem of whether an auditor can ever claim an entity did not unlearn a point. That is, can a model trainer claim to have trained without a point even if they in fact did? Forgeability [39] is a framework under which a model trainer can claim to have obtained their model by training on a dataset they did not in fact train on. To make a claim of training on a given dataset, forgeability relies on providing a valid Proof-of-Learning (PoL) [23] that uses the claimed dataset (different from what the model trainer originally used). Recall PoL is a sequence of checkpoints and minibatches for which the update from $i'th$ minibatch given the $i'th$ checkpoint leads to the $i+1'th$ checkpoint upto some error δ in a metric d. However, it is currently not known how to properly pick the threshold δ and metric d to define a "valid" update for a PoL (due to a lack of models for the backend noise during training), or how to make PoL efficient to verify without introducing additional security risks [12]. Hence the current framework for forgeability may not be robust until PoL is better understood.

As an alternative to using PoL, a per-instance DP guarantee tells us that it is very likely we would have obtained the same sequence of checkpoints with either of the two datasets. Hence, when an auditor claims a model trainer trained on a point (and the point has strong per-instance DP guarantees), the trainer can refute by submitting a dataset without the point and their original sequence of checkpoints and the details of their DP training implementation. Given this, an auditor that only has the information provided in the submitted proof can no longer distinguish between whether a trainer had or had not trained on the point.

6 Conclusion

Our work can be viewed as the first existential result showing better per-instance DP guarantees for deep learning when using DP-SGD. In doing so, we provided one resolution to an open problem in the field of privacy attacks against deep learning: why many privacy attacks fail in practice. However, further work is needed to convert our analysis into a fast algorithm to do privacy accounting. Our composition theorem requires computing several training runs, and the per-step analysis of Theorem 3.6 (which allows better analysis of the mean update rule) requires computing updates for many mini-batches at each step. Future work may be able to significantly reduce the cost associated with using these theorems, or propose alternative theorems that are more efficient to implement. Future work can also likely design algorithms that explicitly take advantage of sensitivity distributions, which we showed are implicit in DP-SGD in explaining its better per-instance privacy guarantees.

Acknowledgements

We would like to acknowledge our sponsors, who support our research with financial and in-kind contributions: Amazon, Apple, CIFAR through the Canada CIFAR AI Chair, DARPA through the GARD project, Intel, Meta, NSERC through the COHESA Strategic Alliance and a Discovery Grant, Ontario through an Early Researcher Award, and the Sloan Foundation. Anvith Thudi is supported by a Vanier Fellowship from the Natural Sciences and Engineering Research Council of Canada. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. We would further like to thank Relu Patrascu at the University of Toronto for providing the compute infrastructure needed to perform the experimentation outlined in this work. We would also like to thank members of the CleverHans lab, Mahdi Haghifam, and our shepherd for their feedback on drafts of the manuscript.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] J. Altschuler and K. Talwar. Privacy of noisy stochastic gradient descent: More iterations without more privacy loss. *Advances in Neural Information Processing Systems*, 35:3788–3800, 2022.

[3] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

[4] A. Beimel, H. Brenner, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94:401–437, 2014.

[5] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[6] Y. Cao and J. Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[7] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019.

[8] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

[9] N. Carlini, M. Jagielski, C. Zhang, N. Papernot, A. Terzis, and F. Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[11] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9 (3–4):211–407, 2014.

[12] C. Fang, H. Jia, A. Thudi, M. Yaghini, C. A. Choquette-Choo, N. Dullerud, V. Chandrasekaran, and N. Papernot. Proof-of-learning is currently more broken than you think. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 797–816. IEEE, 2023.

[13] V. Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

[14] V. Feldman and C. Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

[15] V. Feldman and T. Zrnic. Individual privacy accounting via a renyi filter. *Advances in Neural Information Processing Systems*, 34:28080–28091, 2021.

[16] A. Ginart, M. Guan, G. Valiant, and J. Y. Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

[17] S. Gopi, Y. T. Lee, and L. Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642, 2021.

[18] C. Guo, T. Goldstein, A. Hannun, and L. Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

[19] C. Guo, B. Karrer, K. Chaudhuri, and L. van der Maaten. Bounding training data reconstruction in private (deep) learning. In *International Conference on Machine Learning*, pages 8056–8071. PMLR, 2022.

[20] V. Gupta, C. Jung, S. Neel, A. Roth, S. Sharifi-Malvajerdi, and C. Waites. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34:16319–16330, 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] M. Jagielski, J. Ullman, and A. Oprea. Auditing differentially private machine learning: How private is private SGD? In *Advances in Neural Information Processing Systems, NeurIPS 2020*, volume 33, pages 22205–22216. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fc4ddc15f9f4b4b06ef7844d6bb53abf-Abstract.html.

[23] H. Jia, M. Yaghini, C. A. Choquette-Choo, N. Dullerud, A. Thudi, V. Chandrasekaran, and N. Papernot. Proof-of-learning: Definitions and practice. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1039–1056. IEEE, 2021.

[24] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[25] Z. Kong, A. Roy Chowdhury, and K. Chaudhuri. Forgeability and membership inference attacks. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*, pages 25–31, 2022.

[26] A. Koskela, M. Tobaben, and A. Honkela. Individual privacy accounting with gaussian differential privacy. *arXiv preprint arXiv:2209.15596*, 2022.

[27] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[28] B. Kulynych, Y.-Y. Yang, Y. Yu, J. Błasiok, and P. Nakkiran. What you see is what you get: Principled deep learning via distributional generalization. *Advances in Neural Information Processing Systems*, 35:2168–2183, 2022.

[29] Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[30] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[31] S. Mahloujifar, A. Sablayrolles, G. Cormode, and S. Jha. Optimal membership inference bounds for adaptive composition of sampled gaussian mechanisms. *arXiv preprint arXiv:2204.06106*, 2022.

[32] A. Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

[33] I. Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[34] I. Mironov, K. Talwar, and L. Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[35] M. Nasr, S. Song, A. Thakurta, N. Papernot, and N. Carlini. Adversary instantiation: Lower bounds for differentially private machine learning, 2021.

[36] M. Nasr, J. Hayes, T. Steinke, B. Balle, F. Tramèr, M. Jagielski, N. Carlini, and A. Terzis. Tight auditing of differentially private machine learning. *arXiv preprint arXiv:2302.07956 [cs.LG]*, 2023. doi: 10.48550/arXiv.2302.07956.

[37] I. Shumailov, Z. Shumaylov, D. Kazhdan, Y. Zhao, N. Papernot, M. A. Erdogdu, and R. J. Anderson. Manipulating sgd with data ordering attacks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18021–18032. Curran Associates, Inc., 2021.

[38] S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, pages 245–248. IEEE, 2013.

[39] A. Thudi, H. Jia, I. Shumailov, and N. Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4007–4022, 2022.

[40] A. Thudi, I. Shumailov, F. Boenisch, and N. Papernot. Bounding membership inference. *arXiv preprint arXiv:2202.12232*, 2022.

[41] K. Tirumala, A. Markosyan, L. Zettlemoyer, and A. Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

[42] T. Van Erven and P. Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[43] Y.-X. Wang. Per-instance differential privacy. *Journal of Privacy and Confidentiality*, 9(1), 2019.

[44] D. Yu, G. Kamath, J. Kulkarni, T.-Y. Liu, J. Yin, and H. Zhang. Individual privacy accounting for differentially private stochastic gradient descent. 2022.

[45] S. Zanella-Béguelin, L. Wutschitz, S. Tople, A. Salem, V. Rühle, A. Paverd, M. Naseri, and B. Köpf. Bayesian estimation of differential privacy. *arXiv preprint arXiv:2206.05199 [cs.LG]*, 2022. doi: 10.48550/arXiv.2206.05199.

[46] C. Zhang, S. Bengio, M. Hardt, M. C. Mozer, and Y. Singer. Identity crisis: Memorization and generalization under extreme overparameterization. *arXiv preprint arXiv:1902.04698*, 2019.

A Proofs

A.1 Proof of Lemma A.1

Lemma A.1. *With the above notation, and $X' = X \cup \{x^*\}$, we have the per-instance inequality*

$$\begin{aligned}&\mathbb{P}(M(X') \in S)\\ &\quad \leq \mathbb{P}_{x^*}(1)\mathbb{E}_{X_B}(e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)p})^{1/p}\mathbb{P}(M(X) \in S)^{1-1/p}\\ &\qquad + \mathbb{P}_{x^*}(1)\delta + \mathbb{P}_{x^*}(0)\mathbb{P}(M(X) \in S) \end{aligned} \tag{11}$$

Proof. First note sampling mini-batches from X' is equivalent to sampling a mini-batch X_B from X, then sampling x^* with probability $\mathbb{P}_{x^*}(1)$. Hence we have

$$\begin{aligned}\mathbb{P}(M(X') \in S) = \sum_{x_B}(&\mathbb{P}_{x^*}(1)\mathbb{P}(A(X_B \cup x^*) \in S)\\ &+ \mathbb{P}_{x^*}(0)\mathbb{P}(A(X_B) \in S))\mathbb{P}(X_B)\end{aligned} \tag{12}$$

Now note we have $\mathbb{P}(A(X_B \cup x^*) \in S) \leq e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)}\mathbb{P}(A(X_B) \in S) + \delta$ by the (ε, δ)-DP guarantee of the Gaussian mechanism. So considering summing that over X_B we have $\sum_{X_B}\mathbb{P}(A(X_B \cup x^*) \in S)\mathbb{P}(X_B) \leq \sum_{X_B} e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)}\mathbb{P}(A(X_B) \in S)\mathbb{P}(X_B) + \delta$. Now we apply Holder's inequality to get $\sum_{X_B} e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)}\mathbb{P}(A(X_B) \in S)\mathbb{P}(X_B) \leq \mathbb{E}_{X_B}((e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)})^p)^{1/p}\mathbb{E}_{x_B}(\mathbb{P}(A(x_B) \in S)^q)^{1/q}$. Note that $\mathbb{P}(A(x_B) \in S)^q \leq \mathbb{P}(A(x_B) \in S)$ for $q \geq 1$ as $\mathbb{P}(A(x_B) \in S) \leq 1$.

So we have

$$\begin{aligned}&\sum_{X_B}\mathbb{P}(A(X_B \cup x^*) \in S)\\ &\leq \mathbb{E}_{x_B}((e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)})^p)^{1/p}\mathbb{E}_{x_B}(\mathbb{P}(A(x_B) \in S))^{1/q} + \delta\end{aligned} \tag{13}$$

So to conclude we have

$$\begin{aligned}&\mathbb{P}(M(X') \in S)\mathbb{P}(X_B)\\ &\leq \mathbb{P}_{x^*}(1)\mathbb{E}_{x_B}((e^{C_{\delta,\sigma}\Delta_{U,x^*}(X_B)})^p)^{1/p}\mathbb{E}_{x_B}(\mathbb{P}(A(x_B) \in S))^{1-1/p}\\ &\qquad + \delta + \mathbb{P}_{x^*}(0)\mathbb{E}_{X_B}(\mathbb{P}(A(X_B) \in S))\end{aligned} \tag{14}$$

Note $\mathbb{E}_{X_B}(\mathbb{P}(A(X_B) \in S)) = \mathbb{P}(M(X) \in S)$ which completes the proof.

□

A.2 Proof of Corollary 3.1

Proof. The following proof relies on independently analyzing two cases for what the value of δ could be to conclude the corollary statement (and is inspired by the proof of Proposition 3 in [33]).

Let $Q = \mathbb{P}(M(X) \in S)$ and note the first term in Lemma A.1 is then $(a_p^{\frac{1}{1-1/p}}Q)^{1-1/p}$. Now consider two cases: if $a_p^{\frac{1}{1-1/p}}Q > \delta'^{\frac{p}{p-1}}$ we have $(a_p^{\frac{1}{1-1/p}}Q)^{1-1/p} \leq a_p^{\frac{1}{1-1/p}}Q \cdot \delta'^{\frac{-1}{p-1}}$, and so the overall expression in Lemma A.1 is $\leq (a_p^{\frac{1}{1-1/p}}\delta'^{\frac{-1}{p-1}} + \mathbb{P}(0))Q + \mathbb{P}_{x^*}(1)\delta$.

Now else we have the first term is $\leq \delta'$ and the overall expression is $\leq \mathbb{P}_{\mathbf{x}^*}(0)Q + \mathbb{P}_{x^*}(1)\delta + \delta'$. Combining the two scenarios we see we always have $\mathbb{P}(M(X') \in S) \leq (a_p^{\frac{1}{1-1/p}}\delta'^{\frac{-1}{p-1}} + \mathbb{P}_{x^*}(0))\mathbb{P}(M(X) \in S) + \mathbb{P}_{x^*}(1)\delta + \delta'$, giving the stated condition.

□

A.3 Proof of Theorem 3.2

Proof. The proof is analogous to the results of Mironov et al. [34] with slight modifications to make it per-instance.

Recall that the density function for the sampled Gaussian mechanism $M(X)$ is

$$\sum_{X_B \subset D}\mathbb{P}(X_B)N(U(X_B), \sigma^2\mathbb{I}^d)$$

and for $X' = X \cup x^*$ it is

$$\begin{aligned}&\sum_{X'_B \subset X'}\mathbb{P}(X'_B)N(U(X'_B), \sigma^2\mathbb{I}^d)\\ &\quad = \sum_{X_B \subset D}\mathbb{P}(X_B)(\mathbb{P}_{x^*}(0)N(U(X_B), \sigma^2\mathbb{I}^d)\\ &\qquad + \mathbb{P}_{x^*}(1)N(U(X_B \cup x^*), \sigma^2\mathbb{I}^d))\end{aligned} \tag{15}$$

By quasi concavity of the Rényi divergence we then have

$$
\begin{aligned}
&D_\alpha(M(X)||M(X')) \\
&\leq sup_{X_B \subset X} D_\alpha(N(U(X_B), \sigma^2 \mathbb{I}^d)||\mathbb{P}_{x^*}(0)N(U(X_B), \sigma^2 \mathbb{I}^d) \\
&\qquad + \mathbb{P}_{x^*}(1)N(U(X_B \cup x^*), \sigma^2 \mathbb{I}^d)) \\
&\quad \leq sup_{X_B \subset X} D_\alpha(N(0, \sigma^2 \mathbb{I}^d)||\mathbb{P}_{x^*}(0)N(0, \sigma^2 \mathbb{I}^d) \\
&\qquad + \mathbb{P}_{x^*}(1)N(U(X_B \cup x^*) - U(X_B), \sigma^2 \mathbb{I}^d)) \qquad (16)
\end{aligned}
$$

where we also used that Rényi divergences are translationally invariant. Now by noting the covariances are symmetric, we can apply a change of variables such that $U(X_B \cup x^*) - U(X_B) \to ||U(X_B \cup x^*) - U(X_B)||_2 e_1$ without changing the divergence. As now the change is only along one dimension of the product distribution, by additivity of Rényi divergences we conclude

$$
\begin{aligned}
D_\alpha(M(X)||M(X')) &\leq sup_{X_B \subset D} D_\alpha(N(0, \sigma^2)||\mathbb{P}_{x^*}(0)N(0, \sigma^2) \\
&\quad + \mathbb{P}_{x^*}(1)N(||U(X_B \cup x^*) - U(X_B)||_2, \sigma^2)) \qquad (17)
\end{aligned}
$$

However as $||U(X_B \cup x^*) - U(X_B)||_2 \leq \Delta_{U,x^*}$ we can conclude (by change of variables and post-processing)

$$
\begin{aligned}
D_\alpha(M(X)||M(X')) &\leq D_\alpha(N(0, \sigma^2)||\mathbb{P}_{x^*}(0)N(0, \sigma^2) \\
&\quad + \mathbb{P}_{x^*}(1)N(\Delta_{U,x^*}, \sigma^2)) \qquad (18)
\end{aligned}
$$

Analogous calculations show

$$
\begin{aligned}
&D_\alpha(M(X')||M(X)) \\
&\leq D_\alpha(\mathbb{P}_{x^*}(0)N(0, \sigma^2) + \mathbb{P}_{x^*}(1)N(\Delta_{U,x^*}, \sigma^2)||N(0, \sigma^2)) \qquad (19)
\end{aligned}
$$

Now Theorem 5 in Mironov et al. [34] states that if P, Q are two differentiable distributions s.t $P(x) = Q(v(x))$ where $v(v(x)) = x$ and v is also differentiable, then for all $\alpha \geq 1$ and $q \in [0,1]$ $D_\alpha((1-q)P + qQ||Q) \geq D_\alpha(Q||(1-q)P + qQ)$. Now defining $Q = N(0, \sigma^2)$ and $P = N(\Delta_{U,x^*}, \sigma^2)$ we have $v(x) = \Delta_{U,x^*} - x$ which is differentiable and $v(v(x)) = x$, and so conclude

$$
\begin{aligned}
D(M(X)||M(X')) &\leq D_\alpha(N(0, \sigma^2)||\mathbb{P}_{x^*}(0)N(0, \sigma^2) + \\
&\qquad \mathbb{P}_{x^*}(1)N(\Delta_{U,x^*}, \sigma^2)) \\
&\leq D_\alpha(\mathbb{P}_{x^*}(0)N(0, \sigma^2) + \mathbb{P}_{x^*}(1)N(\Delta_{U,x^*}, \sigma^2)||N(0, \sigma^2)) \qquad (20)
\end{aligned}
$$

As the last expression already bounds $D(M(X')||M(X))$, we proceed to bound it to get our desired guarantee.

Following the calculations of Section 3.3 in Mironov et al. [34] we use $\mu = \mathbb{P}_{x^*}(0)N(0, \sigma^2) + \mathbb{P}_{x^*}(1)N(\Delta_{U,x^*}, \sigma^2)$, $\mu_0 = N(0, \sigma^2)$, and $\mu_{\Delta_{U,x^*}} = N(\Delta_{U,x^*}, \sigma^2)$. We are thus interested in $D_\alpha(\mu||\mu_0) = \frac{1}{\alpha-1} \ln \int (\mu(w)/\mu_0(w))^\alpha \mu_0(w)$. Note $(\mu/\mu_0)^\alpha = \sum_{k=0}^{\alpha} \binom{\alpha}{k} (1 - \mathbb{P}_{x^*}(1))^{\alpha-k} (\frac{\mu_{\Delta_{U,x^*}}}{\mu_0})^k$ and so plugging that into the previous integral, and then completing the square in the exponent for the Gaussian density function, we get

$$
\begin{aligned}
D_\alpha(\mu||\mu_0) &= \frac{1}{\alpha-1} \ln \sum_{k=0}^{\alpha} \binom{\alpha}{k} (1 - \mathbb{P}_{x^*}(1))^{\alpha-k} \\
&\quad \frac{1}{\sigma\sqrt{2\pi}} \int \exp \frac{-(x-k)^2 + (k^2 \Delta_{U,x^*}^2 - k\Delta_{U,x^*}^2)}{2\sigma^2} \\
&= \frac{1}{\alpha-1} \ln \sum_{k=0}^{\alpha} \binom{\alpha}{k} (1 - \mathbb{P}_{x^*}(1))^{\alpha-k} \exp \frac{\Delta_{U,x^*}^2 (k^2 - k)}{2\sigma^2}
\end{aligned}
\qquad (21)
$$

This concludes the proof.

□

A.4 Proof of Theorem A.2

We first present a version of Theorem 3.3 that uses Cauchy-Schwarz, i.e., Holder's inequality with Holder constant $p = 2$. This we believe is easier to follow, and makes clearer the specific role of the Holder's constants in the proof of Theorem 3.3

Theorem A.2. *Consider a sequence of functions $X_1(x_1), X_2(x_1, x_2), X_3(x_2, x_3), \cdots X_n(x_{n-1}, x_n)$ where X_i is a density function in the second arugment for any fixed value of the first argument, except X_1 which is a densitiy function in x_1. Consider an analogous sequence $Y_1(x_1), \cdots, Y_n(x_{n-1})$. Then letting $X = \prod_{j=1}^n X_j$ be the density function for a sequence $x_1, \cdots, x_n$ generated according to the Markov chain defined by X_i, and similarly Y, we have*

$$
\begin{aligned}
&D_\alpha(X||Y) \\
&\leq \frac{1}{\alpha-1} (\sum_{i=0}^{n-2} \frac{1}{2^{i+1}} \ln(\mathbb{E}_{X_1, \cdots X_{n-(i+1)}}((e^{(g^i(\alpha)-1) D_{g^i(\alpha)}(X_{n-i}||Y_{n-i})})^2))) \\
&\qquad + \frac{1}{\alpha-1}((\frac{1}{2})^n \ln((e^{(g^{n-1}(\alpha)-1) D_{g^{n-1}(\alpha)}(X_1||Y_1)})^2)) \qquad (22)
\end{aligned}
$$

where $g(\alpha) = 2\alpha - 1$ and g^i means g composed i times, where $g^0(\alpha) = \alpha$

Proof. The proof relies on repeating the same reduction on the number of steps being considered. First note

$$
\begin{aligned}
&\int (X_1 \cdots X_n)^\alpha (Y_1 \cdots Y_n)^{1-\alpha} dx_1 \cdots dx_n \\
&\qquad = \int (X_1 \cdots X_{n-1})^{\alpha - 1/2} (Y_1 \cdots Y_{n-1})^{1-\alpha} \\
&\qquad (\int X_n^\alpha Y_n^{1-\alpha} dx_n)(X_1 \cdots X_{n-1})^{1/2} dx_1 \cdots dx_{n-1} \\
&\leq (\int (X_1 \cdots X_n)^{2\alpha-1} (Y_1 \cdots Y_n)^{1-(2\alpha-1)} dx_1 \cdots dx_{n-1})^{1/2} \\
&\quad (\int (\int X_n^\alpha Y_n^{1-\alpha} dx_n)^2 (X_1 \cdots X_{n-1}) dx_1 \cdots dx_{n-1})^{1/2} \qquad (23)
\end{aligned}
$$

where the first equality was from using the markov property, and the last inequality was from Cauchy-Schwarz. So now looking at the first term, we are back to the original expression but with $\alpha \to g(a) = 2\alpha - 1$ and $n \to n-1$, and an exponent to $1/2$. Note the second term is an expectation over the $n-1$ model state of the Markov chain. Do note $\int X_n^\alpha Y_n^{1-\alpha} dx_n$ is $e^{(\alpha-1) D_\alpha(X_{n-i}||Y_{n-i})}$ for a fixed $n-1$ model state (i.e., fixed

x_{n-1}). So repeating this step on the first term until we are left only with an integral over x_1 we have

$$\int (X_1\cdots X_n)^{\alpha}(Y_1\cdots Y_n)^{1-\alpha}dx_1\cdots dx_n$$

$$\le (\prod_{i=0}^{n-2}(\mathbb{E}_{X_1,\cdots X_{n-(i+1)}}((e^{(g^i(\alpha)-1)D_{g^i(\alpha)}(X_{n-i}||Y_{n-i})})^2))^{(\frac{1}{2})^{i+1}})$$

$$((e^{(g^{n-1}(\alpha)-1)D_{g^{n-1}(\alpha)}(X_1||Y_1)})^2)^{(\frac{1}{2})^n} \tag{24}$$

So now noting

$$D_\alpha(X||Y) = \frac{1}{\alpha-1}\ln(\int (X_1\cdots X_n)^{\alpha}(Y_1\cdots Y_n)^{1-\alpha}dx_1\cdots dx_n)$$

we conclude by the previous expression that

$$D_\alpha(X||Y)$$

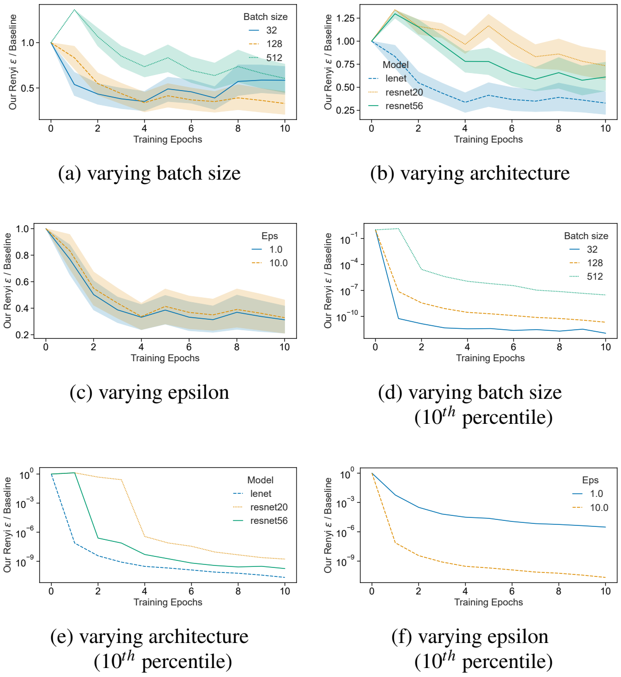
$$\le \frac{1}{\alpha-1}(\sum_{i=0}^{n-2}\frac{1}{2^{i+1}}\ln(\mathbb{E}_{X_1,\cdots X_{n-(i+1)}}((e^{(g^i(\alpha)-1)D_{g^i(\alpha)}(X_{n-i}||Y_{n-i})})^2)))$$

$$+\frac{1}{\alpha-1}((\frac{1}{2})^n\ln((e^{(g^{n-1}(\alpha)-1)D_{g^{n-1}(\alpha)}(X_1||Y_1)})^2)) \tag{25}$$

which completes the proof. □

B Additional Empirical Results

We present additional experiments here.



(a) varying batch size

(b) varying architecture

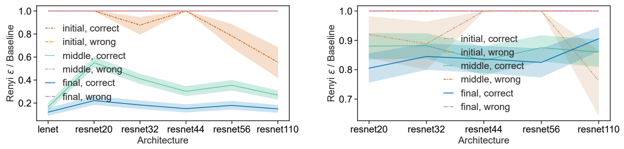
(c) varying epsilon

(d) varying batch size (10^{th} percentile)

(e) varying architecture (10^{th} percentile)

(f) varying epsilon (10^{th} percentile)

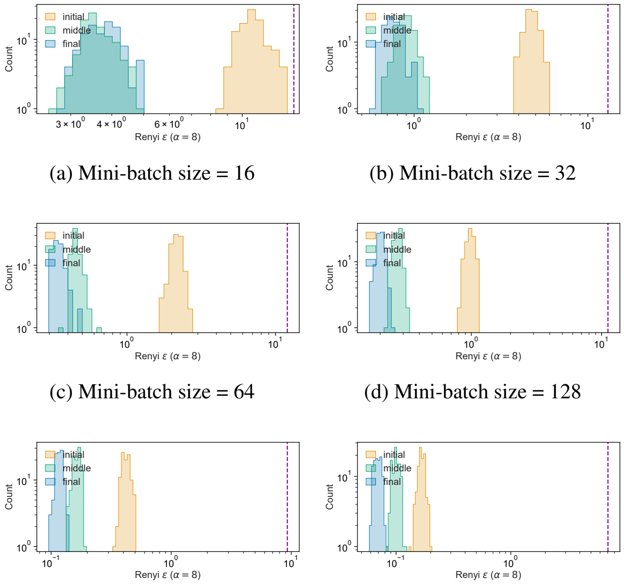
Figure 6: Expected privacy guarantees from Theorem 3.2 plotted as a fraction of the per-step DP-SGD guarantee over training. One can see that the ratio between our guarantee and the per-step DP-SGD guarantee (the baseline) decreases as training approaches the end, and this is consistent across different strengths of DP (i.e., ε set for the entire training), varying mini-batch size, and different model architectures.



(a) MNIST

(b) CIFAR10

Figure 7: Per-step Rényi-DP guarantee given by Theorem 3.2 (divided by the baseline guarantee) plotted with respect to different model architectures trained on MNIST and CIFAR-10, at 3 stages of training. Consistently across different architectures and datasets, data points at later stages of training that are correctly classified have significantly better privacy guarantees than the baseline.



(a) Mini-batch size = 16

(b) Mini-batch size = 32

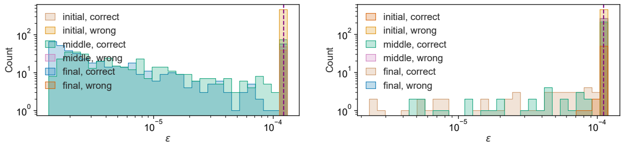
(c) Mini-batch size = 64

(d) Mini-batch size = 128

(e) Mini-batch size = 256

(f) Mini-batch size = 512

Figure 8: Distribution plots (log scale) of per-step guarantees given by Theorem 3.6 computed on LeNet-5 trained on MNIST with mean update rule and varying mini-batch sizes of 16, 32, 64, 128, 256, 512. As specified by the legend labels, we group the plotted guarantees by whether the model is at the initial, middle, or final stage of the training. It can be seen that in all settings our guarantee is better than the baseline, which is represented by the dashed purple line. However, the guarantee distributions of points at the initial stage of training are closer to the baseline compared to the other distributions. Additionally, since a mean update rule is used, the bounds depend on the mini-batch size, and better bounds are achieved when the mini-batch size is large.



(a) MNIST

(b) CIFAR10

Figure 9: Distribution plots of per-step guarantee given by Corollary 3.1 computed on MNIST and CIFAR10 with sum update rule with expected minibatch size 128, for different stages of training and correctly and incorrectly classified points. It can be seen that in all settings our guarantee is better than the baseline. However, the guarantee distributions of incorrectly classified points and points at the initial stage of training are closer to the baseline compared to the other settings.