Rényi Differential Privacy of Propose-Test-Release and Applications to Private and Robust Machine Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Propose-Test-Release (PTR) is a differential privacy framework that works with local sensitivity of functions, instead of their global sensitivity. This framework is typically used for releasing robust statistics such as median or trimmed mean in a differentially private manner. While PTR is a common framework introduced over a decade ago, using it in applications such as robust SGD where we need many adaptive robust queries is challenging. This is mainly due to the lack of Rényi Differential Privacy (RDP) analysis, an essential ingredient underlying the moments accountant approach for differentially private deep learning. In this work, we generalize the standard PTR and derive the first RDP bound for it. We show that our RDP bound for PTR yields tighter DP guarantees than the directly analyzed (ε, δ)-DP. We also derive the algorithm-specific privacy amplification bound of PTR under subsampling. We show that our bound is much tighter than the general upper bound and close to the lower bound. Our RDP bounds enable tighter privacy loss calculation for the composition of many adaptive runs of PTR. As an application of our analysis, we show that PTR and our theoretical results can be used to design differentially private variants for byzantine robust training algorithms that use robust statistics for gradients aggregation. We conduct experiments on the settings of label, feature, and gradient corruption across different datasets and architectures. We show that PTR-based private and robust training algorithm significantly improves the utility compared with the baseline.

1 Introduction

Privacy is a major concern for deploying machine learning (ML). In response, differential privacy (DP) [DMNS06] has become the de-facto measure of privacy. For a differentially private mechanism, the probability distribution of the mechanism's outputs on a dataset should be close to the distribution of its outputs on the same dataset with any single individual's data replaced. A general recipe for releasing the value of a function f on dataset S in a differentially private way is adding random noise to $f(S)$ (output perturbation), where noise magnitude should scale with f's global sensitivity.

However, it might be over-conservative to add noise scaled with global sensitivity. There is a line of research on whether we can do better (e.g., [NRS07, DL09, TS13, KNRS13]). Propose-Test-Release (PTR) [DL09] is a framework that improves the general recipe with the notion of *local sensitivity*. The main idea of PTR is as follows: instead of adding noise with respect to global sensitivity, we propose an amount of noise that is tolerable for many common queries. When we receive the actual query, we test (in a differentially private way) whether answering the query with the proposed amount of noise is enough for privacy. If the proposed noise is too small for limiting the privacy loss from the actual query, we may refuse to answer the query or respond with a larger noise. PTR works especially well for releasing robust statistics such as median or trimmed mean, as the robust statistics usually have small local sensitivity on most common inputs.

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

While PTR is a basic framework that ages back to the early days after the introduction of DP, it has not been used for differentially private optimization before. DP-SGD [ACG+16] is the general backbone for differentially private deep learning and optimization. One major challenge of augmenting SGD with PTR and training DP models is the calculation of privacy parameters after a large number of adaptive compositions. Rényi differential privacy (RDP) and Moment Accountant [Mir17, ACG+16] enable us to calculate tighter privacy parameters for training DP models. Without the RDP bound of PTR, we need to calculate the privacy parameters using the advanced composition theorems [DR+14] that can lead to significantly looser privacy bounds. Besides, we also need the bound of *privacy amplification by subsampling* for PTR, which is the other important support for training DP models. It allows us to exploit the stochasticity of SGD for the interest of stronger privacy guarantees.

Technical Overview. In this work, we derive the Rényi DP bound for PTR, as well as for its Poisson subsampled variant. Our bounds make it possible for us to use PTR framework in augmenting private SGD. PTR could be characterized by three mechanisms. The first mechanism is $\mathcal{M}_1$ that determines which mechanism to run next. Depending on the outcome of $\mathcal{M}_1$ we then either run $\mathcal{M}_2$ or $\mathcal{M}_2'$. It is often the case that the worst-case privacy loss of one of $\mathcal{M}_2$ or $\mathcal{M}_2'$ is much larger than the other. Our bound exploits the fact that the worst-case will happen with small probability. Specifically, instead of considering the worst-case privacy loss between $\mathcal{M}_2$ and $\mathcal{M}_2'$, and naively composing it with $\mathcal{M}_1$, we show that for RDP we can tighten the bound by the *average* privacy loss of $\mathcal{M}_2$ and $\mathcal{M}_2'$ under the distribution imposed by $\mathcal{M}_1$. Direct (ε, δ)-DP analysis does not enjoy this benefit; compared with directly analyzing the (ε, δ)-DP bound of PTR, we show that by first bounding the RDP of PTR and then convert it to (ε, δ)-DP can lead to better privacy guarantee. Our proof could serve as a general recipe for analyzing DP/RDP guarantees for composed mechanisms where the privacy loss of each mechanism is adaptively determined. Additionally, we extend our analysis to the RDP of *subsampled* PTR. Our algorithm-specific analysis (the "white-box bound") allows us to get tighter privacy amplification bounds, compared with the one obtained by general subsampled RDP bound that supports any mechanisms (the "black-box bound") [ZW19]. The proof tackled several additional difficulties compared with the analysis for simple Gaussian mechanism [MTZ19]. By numerical verification, we show that our RDP bound for subsampled PTR is much tighter than the black-box bound, and is close to the lower bound (Figure 2.).

Applications of PTR in ML. PTR is especially suitable for improving the utility of privatizing robust statistics such as trimmed mean, as these functions usually have a much smaller local sensitivity compared to their global sensitivity. A critical application of robust statistics for machine learning is defending against corrupted data. The learning algorithm should be robust in the presence of corrupted data (referred to as *Byzantine failure* [LSP82]). Since privacy and robustness are two major concerns for ML training, developing techniques that achieve both goals simultaneously is desirable. We demonstrate the application of PTR in incorporating differential privacy with robust SGD methods. A popular technique for robustifying SGD is to replace the mean with some robust statistics (e.g., trimmed mean) for gradient aggregation [YCKB18, AHJ+21, GLV21]. We use trimmed mean as an example of showing how to augment robust SGD with DP through PTR. We show that the augmented SGD still maintains the robustness guarantee. We conduct extensive experiments on defending against three kinds of attacks: label, feature, and gradient corruptions, and we show that PTR-based robust SGD achieves much better utility than naively privatizing the robust SGD with global sensitivity.

2 Related Work

Since the seminal work of [DL09], the Propose-Test-Release has become a common DP framework mainly used in statistical inference [BAM20, LKO21]. To the best of our knowledge, this work is the first application of PTR in machine learning.

Several prior works (e.g., [LQS12, BBG18, WBK19, ZW19]) focus the general subsampled DP/RDP bound that supports any mechanisms. [WBK19] derives a general RDP privacy amplification for "sampling without replacement" scheme, and [ZW19] obtains a similar result under Poisson subsampling (i.e., including each data point independently at random with certain probability). Only a small body of recent works study algorithm-specific privacy amplification bound by subsampling. However, most of them focus on subsampled Gaussian mechanism [ACG+16, BDRS18, MTZ19]. This work derives the first subsampled RDP bound specific for flexible algorithms such as PTR.

Designing machine learning and optimization algorithms that achieve both privacy and Byzantine-robustness is certainly an important direction. Nevertheless, there are only few works on this line so far. [GGP+21] considered the problem of achieving privacy and byzantine resilience in distributed

SGD with an untrusted server. The privacy level they considered is essentially local differential privacy, which is orthogonal to our focus. [HKJ20] and [SGA20] aim for both robustness and secure multiparty computation instead of differential privacy. [MZH19] naively applies differential privacy to defend against data poisoning attack. However, they find that DP alone cannot defend adversaries that poison a large fraction of training examples. Aside from the setting of training ML models, [LKKO21] propose a polynomial time algorithm that achieves both goals for mean estimation via privatizing filter-based robust mean estimator [DKK+17]. [EMN21] develop a robust and differentially private mean estimator based on exponential mechanism. However, both of their approaches become inefficient (even in polynomial time asymptotically) in high dimensional settings.

3 Background

In this section, we introduce some background on differential privacy, Rényi differential privacy, and privacy-amplification by subsampling. We will also introduce notations as we proceed.

Differential Privacy. Differential privacy is a framework for protecting privacy when performing statistical releases on a dataset with sensitive information about individuals (see the surveys [DR+14, Vad17]). Specifically, for a differentially private mechanism, the probability distribution of the mechanism's outputs of a dataset should be close to the distribution of its outputs on the same dataset with any single individual's data replaced. To formalize this, we call two datasets S, S', each multisets over a data universe $\mathcal{X}$, *adjacent* if one can be obtained from the other by *adding or removing* a single element of $\mathcal{X}$. Further, we use $d(S, S')$ to denote the number of times of adding/removing of data points to transform S to S'. So S and S' are adjacent if and only if $d(S, S') = 1$.

Definition 3.1 (Differential Privacy [DMNS06])**.** For $\varepsilon, \delta \geq 0$, a randomized algorithm $\mathcal{M} : \mathrm{MultiSets}(\mathcal{X}) \to \mathcal{Y}$ is (ε, δ)-*differentially private* if for every dataset pair $S, S' \in \mathrm{MultiSets}(\mathcal{X})$ such that $d(S, S') = 1$, we have:

$$\forall\, T \subseteq \mathcal{Y}\ \Pr[\mathcal{M}(S) \in T] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(S') \in T] + \delta \tag{1}$$

where the randomness is over the coin flips of $\mathcal{M}$.

Rényi Differential Privacy (RDP). Rényi differential privacy (RDP) is a variant of the standard (ε, δ)-DP that uses Rényi-divergence as a distance metric between the output distributions of $\mathcal{M}(S)$ and $\mathcal{M}(S')$, which is particularly useful in training differentially private machine learning models.

Definition 3.2 (Rényi Differential Privacy [Mir17])**.** We say that a mechanism $\mathcal{M}$ is (α, ε)-RDP with order $\alpha \in (1, \infty)$ if for every dataset pair $S, S' \in \mathrm{MultiSets}(\mathcal{X})$ such that $d(S, S') = 1$, we have:

$$D_\alpha\left(\mathcal{M}(S) \| \mathcal{M}\left(S'\right)\right) := \frac{1}{\alpha - 1} \log \mathbb{E}_{o \sim \mathcal{M}(S')}\left[\left(\frac{\mu_{\mathcal{M}(S)}(o)}{\mu_{\mathcal{M}(S')}(o)}\right)^{\alpha}\right] \leq \varepsilon \tag{2}$$

where $\mu_{\mathcal{M}}(\cdot)$ denotes the density function of $\mathcal{M}$'s distribution. Further, we denote the moment $E_\alpha\left(\mathcal{M}(S) \| \mathcal{M}\left(S'\right)\right) := \mathbb{E}_{o \sim \mathcal{M}(S')}\left[\left(\frac{\mu_{\mathcal{M}(S)}(o)}{\mu_{\mathcal{M}(S')}(o)}\right)^{\alpha}\right]$ and function $f_\alpha(\varepsilon) := \exp((\alpha - 1)\varepsilon)$.

As we can see, (α, ε)-RDP is essentially an upper bound for the moment $E_\alpha\left(\mathcal{M}(S) \| \mathcal{M}\left(S'\right)\right) \leq f_\alpha(\varepsilon)$ for all adjacent S, S', where ε can be viewed as a degree of the privacy loss incurred by running $\mathcal{M}$. A different α typically leads to a different privacy bound ε. Following the convention of literature [ZW19], we view ε as a function of α, and the notation $\varepsilon_{\mathcal{M}}(\alpha)$ means the algorithm $\mathcal{M}$ obeys $(\alpha, \varepsilon_{\mathcal{M}}(\alpha))$-RDP. We note that RDP reduces to $(\varepsilon, 0)$-DP when we take $\alpha = \infty$. The current tightest (ε, δ)-DP/RDP transformation we are aware of is by [ALC+21]. A central property of DP/RDP is its behavior under composition. If we run multiple distinct differentially private algorithms on the same dataset, the resulting composed algorithm is also differentially private, with some degradation in the privacy parameters (ε, δ). Specifically, if we run k sequentially chosen (ε, δ)-DP algorithm on a dataset, the overall composed privacy parameter is $\left(\tilde{O}(\sqrt{k}\varepsilon), k\delta + \delta'\right)$-DP by Advanced composition theorem [DR+14]. However, the Advanced composition theorem for (ε, δ)-DP is loose. On the contrary, the composition is trivial for RDP as $\varepsilon_{\mathcal{M}_1 \circ \mathcal{M}_2}(\cdot) = \varepsilon_{\mathcal{M}_1}(\cdot) + \varepsilon_{\mathcal{M}_2}(\cdot)$. The Moment Accountant technique, which composes RDP and then transforms to DP, is a much simpler approach and often produces much more favorable privacy parameters than directly composing (ε, δ)-DP. Therefore, RDP and Moment Accountant are widely used to calculate the privacy guarantee in training differentially private deep learning models.

Privacy amplification by subsampling. "Privacy amplification by subsampling" is the other booster besides RDP / moments accountant that drives much of the recent progress in differentially private deep learning. Most of the existing works focus on Poisson subsampling [BBG18], which samples each data point independently with a given sampling probability q. The tightest privacy amplification bound we are aware of for general mechanism under Poisson Subsampling is from [ZW19].

4 Propose-Test-Release and Rényi Differential Privacy

In this section, we introduce a general version of the Propose-Test-Release framework, present our main results on its RDP bound, and its privacy amplification bound under Poisson subsampling.

In our presentation, we use $\mathbb{GS}_f = \sup_{S,S':d(S,S')=1} \|f(S) - f(S')\|$ to denote the global sensitivity of target function f (in ℓ_2 distance), and we use $\mathbb{LS}_f(S) = \sup_{S,S':d(S,S')=1} \|f(S) - f(S')\|$ to denote the local sensitivity of function f on dataset S. Before we present our main results, we would like to remind the readers about the definition of Laplace and Gaussian mechanisms, as well as their DP/RDP bound, which will be referred to later in the main results.

Laplace Mechanism: If f's output is 1-dimensional, the Laplace mechanism $\mathcal{M}_{\mathrm{Lap},b}(S) = f(S) + \mathrm{Lap}(0, b)$ obeys $(\varepsilon_{\mathrm{Lap}}^{(\tilde{b})}, 0)$-DP for $\varepsilon_{\mathrm{Lap}}^{(\tilde{b})} = e^{1/\tilde{b}}$, and obeys $(\alpha, \varepsilon_{\mathrm{R-Lap}}^{(\tilde{b})}(\alpha))$-RDP for $\varepsilon_{\mathrm{R-Lap}}^{(\tilde{b})}(\alpha) = \frac{1}{\alpha-1} \log\left(\frac{\alpha}{2\alpha-1} \exp\left(\frac{\alpha-1}{\tilde{b}}\right) + \frac{\alpha-1}{2\alpha-1} \exp\left(-\frac{\alpha}{\tilde{b}}\right)\right)$, where $\tilde{b} = b/\mathbb{GS}_f$ (the "noise-to-sensitivity" ratio).

Gaussian Mechanism: If f's output is d-dimensional, the Gaussian mechanism $\mathcal{M}_{\mathcal{N},\sigma}(S) = f(S) + \mathcal{N}(0, \sigma^2 \mathbf{1}_d)$ obeys $(\varepsilon_{\mathcal{N}}^{(\tilde{\sigma})}(\delta), \delta)$-DP for $\varepsilon_{\mathcal{N}}^{(\tilde{\sigma})}(\delta) = \tilde{\sigma}\sqrt{2\log(1.25/\delta)}$, and obeys $(\alpha, \varepsilon_{\mathrm{R-}\mathcal{N}}^{(\tilde{\sigma})}(\alpha))$-RDP for $\varepsilon_{\mathrm{R-}\mathcal{N}}^{(\tilde{\sigma})}(\alpha) = \frac{\alpha}{2\tilde{\sigma}^2}$, where $\tilde{\sigma} = \sigma/\mathbb{GS}_f$ (the "noise-to-sensitivity" ratio).

4.1 Propose-Test-Release

Naive use of Laplace/Gaussian mechanism may result in the poor utility of function output, as the global sensitivity of the target function may be intolerably large due to some extreme cases. Meanwhile, robust statistics such as median, mode, Inter-Quantile Range (IQR) are quite insensitive to single data addition/removal for datasets that are i.i.d. drawn from natural distributions. This means that robust statistics may have a small local sensitivity for most input datasets. The seminal work of [DL09] introduced Propose-Test-Release framework to reduce the noise addition when the target function has an approximation that is a robust statistic[1]. Here, we introduce a more general and useful version of PTR instantiated by Laplace and Gaussian mechanism. Given a target function f_1 (e.g., mean) and its robust variant f_2 (e.g., median), the PTR framework proceeds as follows: **(1) Propose:** a local sensitivity bound of the target function, τ, is proposed; **(2) Test:** a safety margin $\Delta(S)$, which is the minimum amount of data points that we need to replace for S to have local sensitivity larger than τ, is computed. A private version (via Laplace mechanism) of the safety margin, $\widehat{\Delta}$, is compared with a threshold; **(3) Release:** if the safety margin is large enough, then the algorithm releases $f_2(S)$ via Gaussian mechanism with a smaller noise σ_2 (usually scaled with τ). Otherwise, the algorithm release $f_1(S)$ with a larger noise σ_1 (usually scaled with $\mathbb{GS}_{f_1}$). The pseudocode is outlined in Algorithm 1.

Algorithm 1: Propose-Test-Release with Laplace and Gaussian mechanism.

input : S – dataset, f_1 – target function, f_2 – robust statistic, τ – proposed local sensitivity bound of f_2, σ_1, σ_2, b – Gaussian/Laplace noise scales ($\sigma_1 > \sigma_2$), δ_0 – failure probability.

$\Delta \leftarrow \min_{\tilde{S} \in \{\tilde{S}: \mathbb{LS}_{f_2}(\tilde{S}) > \tau\}} d\left(S, \tilde{S}\right)$.

$\widehat{\Delta} \leftarrow \Delta + \mathrm{Lap}(0, b)$.

if $\widehat{\Delta} \leq \log(1/(2\delta_0))b$ **then**

  return $f_1(S) + \mathcal{N}(0, \sigma_1^2 \mathbf{1}_d)$

else

  return $f_2(S) + \mathcal{N}(0, \sigma_2^2 \mathbf{1}_d)$

We make several remarks on Algorithm 1.

Remark 4.1. **(1)** The traditional version of PTR in the textbooks [DR+14, Vad17] simply refuse to output anything when the noisy safety margin $\widehat{\Delta}$ is small. Algorithm 1 is a more general version

[1] Sometimes the robust statistic itself is the target function, i.e., $f_1 = f_2$ in Algorithm 1.

which output $f_1(S) + \mathcal{N}(0, \sigma_1^2 \mathbf{1}_d)$ when the sensitivity test is failed. The textbook PTR can be thought of as a special case of Algorithm 1 where we set $\sigma_1 \to \infty$. **(2)** The mechanism used in the **Test** and **Release** step can be other mechanisms instead of Laplace and Gaussian. In this paper, we present our results for the PTR instantiated by these two mechanisms since we intend to apply it for differentially private SGD later. Our proof can be easily extended to other mechanisms as well. **(3)** The threshold $\log(1/(2\delta_0))b$ in Line 3 is chosen so that $\Pr[\mathrm{Lap}(0, b) > \log(1/(2\delta_0))b] = \delta_0$. **(4)** The global sensitivity of $\Delta(\cdot)$ is 1 for any functions. This could be seen by noticing that, for any pair of adjacent S, S', we have $d(S, \tilde{S}) \leq d(S', \tilde{S}) + 1$ for any dataset $\tilde{S}$.

Direct DP Analysis of Propose-Test-Release. We show the DP guarantee for the Propose-Test-Release framework. To prove privacy, we need to find the worst pair of adjacent datasets S and S' that incurs the largest privacy loss. It is clear that the worst possible scenario of PTR is when $f_2(S) - f_2(S') > \tau$, while the algorithm still releases $f_2(S)$ with noise scaled with τ. However, note that this worst possible scenario will only happen when *both* $\mathbb{LS}(S)$ *and* $\mathbb{LS}(S')$ *are greater than* τ. In this case, however, Δ for both S and S' are 0 since there are no data points we need to change for them to have local sensitivity larger than τ, and thus there is no privacy loss from $\widehat{\Delta}$. Moreover, when $\Delta = 0$, the probability that PTR will release $f_2(S) + \mathcal{N}(0, \sigma_2^2 \mathbf{1}_d)$ is at most δ_0 by construction, which could be simply added to the δ term in (ε, δ)-DP. If one of $\mathbb{LS}(S)$ and $\mathbb{LS}(S')$ is smaller than τ, then we know that $f_2(S) - f_2(S') \leq \tau$, and the overall privacy parameters could be computed by Basic Composition Theorem [DR+14]. We obtain the following differential privacy guarantee for PTR based on these observations. We defer more details of the proof to the Appendix A.

Theorem 4.2 (Direct DP analysis for PTR)**.** *Suppose* $\mathbb{GS}_{f_1} = \mathbb{GS}_{f_2} = 1$ *and* $\sigma_1 = \sigma_2/\tau$*, then Algorithm 1 is* $(\varepsilon_{\mathrm{Lap}}^{(b)} + \varepsilon_{\mathcal{N}}^{(\sigma_1)}(\delta), \delta_0 + \delta)$*-DP, where* $\varepsilon_{\mathrm{Lap}}^{(b)} = e^{1/b}$ *is the DP guarantee for Laplace mechanism, and* $\varepsilon_{\mathcal{N}}^{(\sigma_1)}(\delta) = \sigma_1 \sqrt{2\log(1.25/\delta)}$.

4.2 RDP of Propose-Test-Release

We then derive the RDP of Propose-Test-Release framework. The major differences between the proofs of (ε, δ)-DP and RDP bound is that, for (ε, δ)-DP we can move the probability of running into the bad scenario to the δ term, while for RDP we need to consider the "average-case" privacy loss. This is in fact an advantage of RDP, which we illustrate it in the following simple example.

Comparison between (ε, δ)-DP and RDP Analysis: A motivating example.[2] Suppose we have two mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ who are $(\varepsilon_1, \delta_1)$-DP and $(\varepsilon_2, \delta_2)$-DP, respectively. Consider a simple PTR-like mechanism $\mathcal{M}$ that randomly picks one of mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ to execute, each with probability $1 - \delta_0$ and δ_0, respectively[3]. We can only claim that $\mathcal{M}$ is $(\max(\varepsilon_1, \varepsilon_2), (1 - \delta_0)\delta_1 + \delta_0\delta_2)$-DP, or if we know $\varepsilon_2 \gg \varepsilon_1$ we can also move the "bad case" probability δ_0 to the δ term and obtain $(\varepsilon_1, \delta_0 + \delta_1)$-DP. However, if we know the RDP guarantee of $\mathcal{M}_1$ and $\mathcal{M}_2$ as $E_\alpha(\mathcal{M}_1(S) \| \mathcal{M}_1(S')) \leq f_\alpha(\varepsilon_1)$ and $E_\alpha(\mathcal{M}_2(S) \| \mathcal{M}_2(S')) \leq f_\alpha(\varepsilon_2)$[4], then $E_\alpha(\mathcal{M}(S) \| \mathcal{M}(S'))$ can be simply bounded by $(1 - \delta_0) f_\alpha(\varepsilon_1) + \delta_0 f_\alpha(\varepsilon_2)$. Compared with (ε, δ)-DP analysis, there are no extra inequalities used in RDP analysis of $\mathcal{M}$ except for the RDP guarantee of $\mathcal{M}_1$ and $\mathcal{M}_2$. Thus, RDP is more favorable in for PTR's privacy analysis, especially when δ_0 is close to the target δ.

Theorem 4.3 (RDP analysis of PTR)**.** *Suppose* $\mathbb{GS}_{f_1} = \mathbb{GS}_{f_2} = 1$ *and* $\sigma_1 = \sigma_2/\tau$*. Then for any* $\alpha > 1$*, Algorithm 1 is* $(\alpha, \varepsilon_{\mathrm{PTR}}(\alpha))$*-RDP for*

$$\varepsilon_{\mathrm{PTR}}(\alpha) \leq \max\left(f_\alpha^{-1}\left((1-\delta_0) f_\alpha\left(\varepsilon_{\mathrm{R-}\mathcal{N}}^{(\sigma_1)}(\alpha)\right) + \delta_0 f_\alpha\left(\varepsilon_{\mathrm{R-}\mathcal{N}}^{(\sigma_2)}(\alpha)\right)\right), \varepsilon_{\mathrm{R-}\mathcal{N}}^{(\sigma_1)}(\alpha) + \varepsilon_{\mathrm{R-Lap}}^{(b)}(\alpha)\right)$$

The above result implies that, with appropriately chosen δ_0, the privacy loss of PTR is the same as directly adding up the privacy loss from $\widehat{\Delta}$ and from releasing $f_1(S) + \mathcal{N}(0, \sigma_1^2 \mathbf{1}_d)$ regardless of the value of $\widehat{\Delta}$. Compared with naively releasing $f_1(S) + \mathcal{N}(0, \sigma_1^2 \mathbf{1}_d)$, PTR will only pay an extra cost for privatizing Δ, independent from the privacy cost for the case of releasing $f_2(S) + \mathcal{N}(0, \sigma_2^2 \mathbf{1}_d)$! Thus, the algorithm could potentially add much smaller noise while introducing just a little extra privacy loss. In Figure 1, we convert this RDP bound to (ε, δ)-DP via the RDP-DP conversion

[2] Details of derivation can be found in Appendix A.

[3] For the actual PTR, the δ_0 is not fixed but depends on the input dataset.

[4] Recall that $f_\alpha(\varepsilon) = \exp((\alpha - 1)\varepsilon)$ where if $\mathcal{M}$ is (α, ε)-RDP then $E_\alpha(\mathcal{M}(S) \| \mathcal{M}(S')) \leq f_\alpha(\varepsilon)$.

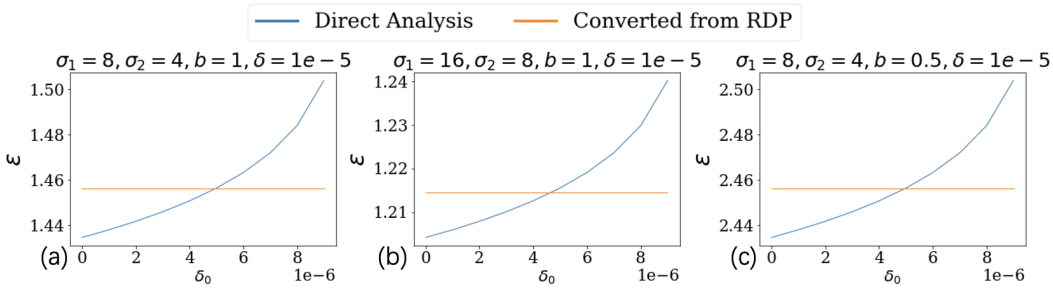

Figure 1: The ε parameter of the (ε, δ)-DP guarantee of PTR when $\delta = 10^{-5}$ for different noise scales. We convert the RDP bound in Theorem 4.3 to (ε, δ)-DP by the RDP-DP conversion formula from [BBG+20], and compare it with the ε obtained from the direct analysis in Theorem 4.2. For the bound converted from RDP, we search for the optimal $\alpha \in [1, 200]$. The bound is constant across different δ_0 since when δ_0 is small, the RDP for PTR will take the second term in (3).

formula by [BBG+20], and compare with the (ε, δ)-DP bound obtained through direct analysis in Theorem 4.2. Although due to the loss in RDP-DP conversion, the ε parameter by RDP is worse than the one by direct analysis when δ_0 is far smaller than the target δ, it is tighter when δ_0 is close to δ. A larger δ_0 means PTR has a greater chance to release $f_2(S) + \mathcal{N}(0, \sigma_2^2 \mathbf{1}_d)$, which leads to a better utility. Thus, the privacy parameter converted from the RDP bound is more favorable.

We would like to stress that in the theorem, most of the conditions on the parameters such as $\mathbb{GS}$ or σ_1, σ_2 can be easily relaxed. The conditions in the theorems are mainly used to make the presented bound more clean and interpretable.

4.3 RDP for Poisson Subsampled Propose-Test-Release

We further derive the RDP for Poisson subsampled PTR. Here, "subsampled" means the input dataset will be subsampled first before feeding to PTR. Poisson subsampling means each data point x will be included with probability q independently. One way of obtaining the RDP for any subsampled mechanism is by simply plugging in the RDP bound of the original algorithm into the privacy amplification formula for general mechanisms (e.g., [WBK19, ZW19]). Here, we directly derive the RDP for Poisson subsampled PTR in a white-box manner. Denote $\mathcal{M}_0$ as the random variable for PTR's output on dataset S, and $\mathcal{M}_1$ for PTR's output on dataset $S' = S \cup \{x\}$, and $\mathcal{M} = (1-q)\mathcal{M}_0 + q\mathcal{M}_1$. Compared with the analysis for subsampled Gaussian mechanism by [MTZ19], there are two main difficulties for extending the arguments of quasi-convexity of Rényi divergence for PTR: **(1)** the distribution of PTR's output may not be centrally symmetric, thus we need to bound both $D_\alpha(\mathcal{M}||\mathcal{M}_0)$ and $D_\alpha(\mathcal{M}_0||\mathcal{M})$. **(2)** the conditional distribution $\mathcal{M}|\widehat{\Delta}$ cannot be decomposed as $(1-q)\mathcal{M}_0|\widehat{\Delta} + q\mathcal{M}_1|\widehat{\Delta}$. A big part of our novelty in the proof is about addressing these two challenges. We defer details of the proof to Appendix A.

Theorem 4.4 (RDP analysis of sub-sampled PTR (the *"white-box bound"*))**.** *Let q be the subsampling probability. Suppose $\mathbb{GS}_{f_1} = \mathbb{GS}_{f_2} = 1$ and $\sigma_1 = \sigma_2/\tau$. When q, σ_1, σ_2, and α satisfy certain conditions, we have*

$$\varepsilon_{\mathrm{PTR} \circ \mathrm{PoissonSample}}(\alpha) \leq f_\alpha^{-1}\left(\max(\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2)\right)$$

where $\mathcal{B}_0 = 1 + 2q^2\alpha(\alpha-1)(\frac{1-\delta_0}{\sigma_1^2} + \frac{\delta_0}{\sigma_2^2})$, $\mathcal{B}_1 = \mathrm{R}_q^{(\alpha)} + 2\alpha(\alpha-1)[\mathrm{R}_q^{(\alpha)} - 2(1-q)\mathrm{R}_q^{(\alpha-1)} + (1-q)^2\mathrm{R}_q^{(\alpha-2)}]$, $\mathcal{B}_2 = \widetilde{\mathrm{R}}_q^{(\alpha)} + 2\alpha(\alpha-1)[\widetilde{\mathrm{R}}_q^{(\alpha)} - 2(1-q)\widetilde{\mathrm{R}}_q^{(\alpha+1)} + (1-q)^2\widetilde{\mathrm{R}}_q^{(\alpha+2)}]$, *with* $\mathrm{R}_q^{(\alpha)} = \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha\right]$ *and* $\widetilde{\mathrm{R}}_q^{(\alpha)} = \mathbb{E}_{s\sim\mu}\left[\left(\frac{\mu_0(s)}{\mu(s)}\right)^\alpha\right]$ *for* $\mu_0 \sim \mathrm{Lap}(0, b)$ *and* $\mu \sim (1-q)\mathrm{Lap}(0, b) + q\mathrm{Lap}(1, b)$.

Remark 4.5. Both $\mathrm{R}_q^{(\alpha)}$ and $\widetilde{\mathrm{R}}_q^{(\alpha)}$ can be computed easily since they either have closed form or can be accurately computed via numerical integration in bounded range.

This bound may not be very interpretable, which is a typical feature for the privacy amplification bounds as they are meant to be implemented in practice. After all, constant matters for differential privacy practitioners! To show the tightness of our bound, we plug in the RDP of original PTR

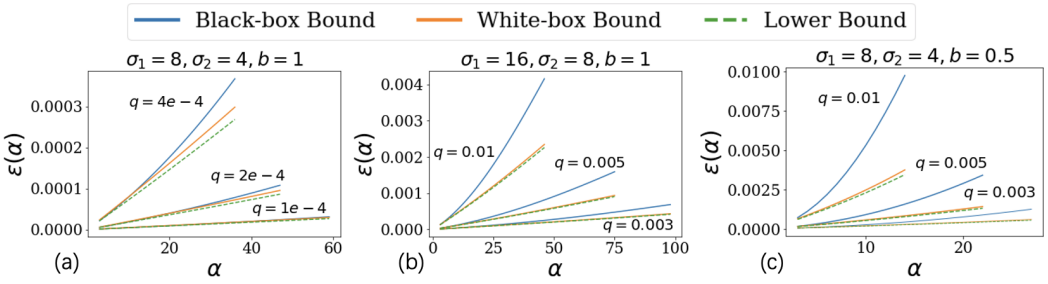

Figure 2: The RDP parameter $\varepsilon(\alpha)$ of the subsampled PTR as a function of order α. We compare the white-box bound from Theorem 4.4, and the black-box bound as well as the lower bound by plugging in the RDP of PTR in Theorem 4.3 to the privacy amplification upper/lower bounds for general mechanisms from [ZW19].

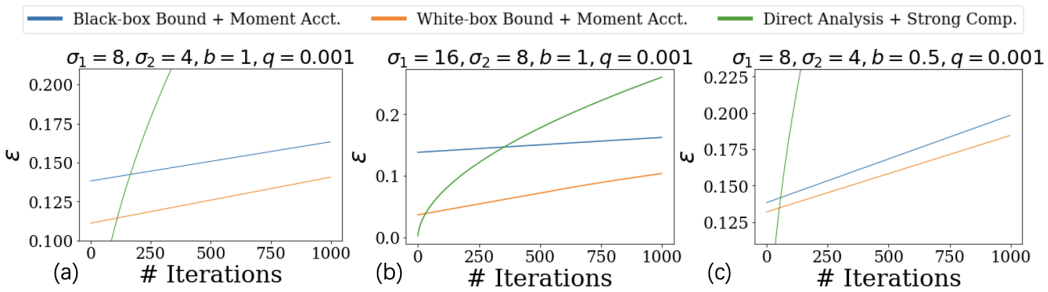

Figure 3: Illustration of the use of our Theorem 4.4 in moments accountant. We plot the the privacy loss ε for $\delta = 10^{-5}$ after different rounds of composition. We set $\delta_0 = 10^{-8}$ here to allow more iterations for Strong Composition of (ε, δ)-DP.

(Theorem 4.3) to the current tightest privacy amplification formula for general mechanisms derived in [ZW19] (the *"black-box bound"*), and compare it with our white-box bound in Theorem 4.4. As we can see from Figure 2, Theorem 4.4 (orange curve) is much tighter than the black-box bound (blue curve). Moreover, it is very close to the lower bound of privacy amplification by [ZW19] (green line), which means that Theorem 4.4 is near optimal. In Figure 3, we illustrate the application of subsampled RDP bound in Moment Accountant. We can see that the privacy parameters for the composed mechanism obtained based on our white-box bound (Theorem 4.4) is tighter than the one by black-box bound, as well as the one by directly composing Theorem 4.2 with strong composition theorem [KOV15]. We remark that Direct Analysis achieves lower privacy loss with very few iterations since we set $\delta_0 = 10^{-8}$ to allow more iterations for Strong Composition of (ε, δ)-DP, which leads to a better ε for a single iteration (see Figure 1).

An exciting byproduct during our research on subsampled PTR is the privacy amplification bound for a $\mathcal{M}$ that outputs $\mathcal{M}_1(S), \mathcal{M}_2(S)$ sequentially and *independently*. This result is important since such a $\mathcal{M}$ is instantiated in many existing techniques in improving differentially private deep learning. We defer this result to the Appendix.

5 Differentially Private and Robust SGD with Propose-Test-Release

In this section, we demonstrate the application of the RDP for (subsampled) PTR algorithm in privatizing robust SGD algorithms.

Attack Model. Mini-batch SGD is a common method for training deep neural networks. Despite its strong convergence properties in the standard settings, it is well known that even a small fraction of corrupted gradients can lead SGD to an arbitrarily poor solution [BTN00, BBC11]. An important attack model is called Byzantine contamination framework or *Byzantine failure* [LSP82]. Consider an optimization problem with n stochastic gradient oracles; at each iteration there are up to F gradient oracles are corrupted (usually referred as *Byzantine agents* in the context of distributed SGD). The identity of corrupted oracles is a priori unknown. As the corrupted gradients can be arbitrarily skewed, this attack model is able to capture many important scenarios including **corruption in feature** (e.g., existence of outliers), **corruption in gradients** (e.g., hardware failure, unreliable communication channels during distributed training) and **corruption in labels** (e.g. label flip (backdoor) attacks).

A popular class of defense strategies against Byzantine failure is to use robust gradient aggregation rules such as trimmed-mean [GLV21], coordinate-wise median [YCKB18], and geometric median [AHJ+21]. Since both privacy and robustness are essential for ML applications, it is important to develop learning algorithms that achieve differential privacy and robustness simultaneously.

Baseline: privatizing robust SGD with Gaussian Mechanism. A natural (but naive) way to accomplish such a goal for SGD-based algorithms is to privatize each iteration of robust SGD through Gaussian mechanism. Specifically, in each iteration, we add Gaussian noise to the aggregated gradient, where noise magnitude scale with the global sensitivity of the robust aggregation function. The privacy parameter for the entire process is computed through Moment Accountant. The global sensitivity can be obtained by either upper bounding the Lipschitz constant of the objective function, or clipping each gradient according to a certain threshold. However, this method could potentially add over-conservative noise to the aggregated gradients, as the local sensitivity of robust aggregation functions is usually very low for most input gradient sets.

Privatizing robust SGD with PTR. We improve the naive method with the PTR framework, where we instantiate it by showing how to privatize trimmed-mean SGD method from [GLV21]. We first define trimmed-sum[5] aggregation function, and show that the Δ for trimmed-sum could be computed efficiently. Given a dataset $S = \{x_1, \ldots, x_m\}$, let $x_{(k)}$ denote the kth smallest data point among S in ℓ_2 norm, i.e., $\|x_{(1)}\| \leq \|x_{(2)}\| \leq \ldots \leq \|x_{(m)}\|$. The *$F$-trimmed sum of S* is defined as $\mathrm{TSUM}_F(S) = \sum_{i=1}^{m-F} x_{(i)}$ if $m > F$, or simply 0 if $m \leq F$. Define $\mathbb{LS}_f^{(r)}(S) = \sup_{S,\tilde{S}:d(S,\tilde{S})=r} \mathbb{LS}_f(\tilde{S})$, which is the maximum local sensitivity that can be achieved by adding/removing r elements from S. Thus, $\mathbb{LS}_f^{(0)}(S) = \mathbb{LS}_f(S)$, and the safety margin $\Delta(S) = \min\{r : \mathbb{LS}_f^{(r)}(S) > \tau\}$. Therefore, as long as we can efficiently compute $\mathbb{LS}_f^{(r)}(S)$, we can efficiently compute $\Delta(S)$ in linear time by simply enumerate all r from 0 to m and terminate once $\mathbb{LS}_f^{(r)}(S) > \tau$. In fact, $\mathbb{LS}_{\mathrm{TSUM}_F}^{(r)}(S)$ can be computed in $O(1)$ time.

Theorem 5.1. $\mathbb{LS}_{\mathrm{TSUM}_F}^{(r)}(S) = \|x_{(m-F+1+r)}\|$ *if* $r \leq F-1$, *or* $\mathbb{GS}_{\mathrm{TSUM}_F}$ *if* $r > F-1$.

Back to the goal of designing differentially private and robust SGD. Fix a positive integer F which is a potential upper bound for the number of corrupted gradients. We instantiate a PTR-based gradient aggregation algorithm by instantiating $f_1(\cdot)$ as $\mathrm{SUM}(S) = \sum_{x \in S} x$, and $f_2(\cdot)$ as $\mathrm{TSUM}_F(\cdot)$. We simply plugging in this PTR algorithm as the gradient aggregation function for regular SGD. We show the convergence guarantee of this algorithm in the presence of at most F gradients being corrupted for the case that the loss function is Lipschitz.

Theorem 5.2 (Informal)**.** *When the loss function is Lipschitz in model parameters, SGD with PTR instantiated by* SUM *and* TSUM *provides convergence guarantee for update to F gradients being corrupted arbitrarily for any $F < m/2$.*

We make two remarks of our specific design choices for implementing PTR-based SGD in practice.

Remark 5.3. **(1)** In practice, F could be adjusted dynamically during the model training, based on the value of $\widehat{\Delta}$[6]. That is, if $\widehat{\Delta} \leq B$, it means that the current gradient batch has many outliers, thus we increase F for the next iteration; if $\widehat{\Delta} > B$, it means that gradients are relatively concentrated and few of them are corrupted, thus we can decrease F afterwards. **(2)** We set f_1 as SUM instead of TSUM since F is usually set to be larger than the actual corrupted gradients; in this case, $\widehat{\Delta} \leq B$ means that there are many benign gradients also have large norms. These benign gradients usually correspond to the data points that are misclassified by the partially trained models, which is important for improving model performance. The pseudocode for the algorithm is deferred to the Appendix A.

5.1 Evaluation

In this section, we empirically evaluate the performance of PTR-based private and robust SGD. Specifically, we compare the utility-privacy tradeoff between the algorithm that privatize trimmed mean-based SGD with PTR (TSGD + PTR), and the baseline algorithm that privatize trimmed mean-based SGD with Gaussian mechanism (TSGD + Gaussian).

Experiment Settings. We evaluate the utility-privacy tradeoff of the two robust and differentially private SGD algorithms under three common types of attack: label, feature, and communicated

[5] We use trimmed-sum instead of trimmed-mean for the ease of analysis, as trimmed-mean essentially just scales the learning rate.

[6] This does not introduce extra privacy leakage as DP is closed under post-processing.

Dataset	Corruption Type	CR	$\varepsilon = 3.0$ TSGD+Gaussian	$\varepsilon = 3.0$ TSGD+PTR	$\varepsilon = 5.0$ TSGD+Gaussian	$\varepsilon = 5.0$ TSGD+PTR
MNIST	**Label**	**0**	87.53%	91.43% (+3.9%)	81.31%	92.8% (+11.49%)
		0.1	90.15%	93.29% (+3.138%)	89.15%	94.608% (+5.456%)
		0.2	91.33%	92.704% (+1.374%)	92.47%	94.084% (+1.614%)
	Feature	**0.1**	89.47%	92.28% (+2.812%)	86.53%	93.388% (+6.854%)
		0.2	91.71%	92.29% (+0.582%)	86.99%	92.864% (+5.872%)
	Gradient	**0.1**	90.27%	91.7% (+1.43%)	88.41%	91.68% (+3.27%)
		0.2	91.13%	91.45% (+0.32%)	88.70%	90.31% (+1.61%)
			$\varepsilon = 7.0$ TSGD+Gaussian	$\varepsilon = 7.0$ TSGD+PTR	$\varepsilon = 10.0$ TSGD+Gaussian	$\varepsilon = 10.0$ TSGD+PTR
CIFAR10	**Label**	**0**	57.94%	58.92% (+0.98%)	58.82%	60.13% (+1.31%)
		0.1	55.48%	56.262% (+0.78%)	57.42%	58.218% (+0.8%)
		0.2	49.69%	50.946% (+1.26%)	52.95%	53.106% (+0.152%)
	Feature	**0.1**	55.70%	56.628% (+0.93%)	57.36%	58.678% (+1.318%)
		0.2	55.13%	55.526% (+0.398%)	56.93%	57.336% (+0.404%)
	Gradient	**0.1**	55.81%	57.29% (+1.48%)	57.21%	58.96% (+1.75%)
		0.2	54.45%	56.31% (+1.86%)	56.11%	57.48% (+1.37%)
			$\varepsilon = 2.0$ TSGD+Gaussian	$\varepsilon = 2.0$ TSGD+PTR	$\varepsilon = 2.5$ TSGD+Gaussian	$\varepsilon = 2.5$ TSGD+PTR
EMNIST	**Label**	**0**	72.94%	76.44% (+3.5%)	79.15%	81.02% (+1.87%)
		0.1	72.60%	75.66% (+3.06%)	79.38%	80.63% (+1.25%)
		0.2	70.03%	72.62% (+2.59%)	77.48%	79.19% (+1.71%)
	Feature	**0.1**	69.60%	74.01% (+4.41%)	77.80%	81.04% (+3.24%)
		0.2	70.04%	74.76% (+4.72%)	78.99%	80.74% (+1.75%)
	Gradient	**0.1**	71.75%	76.19% (+4.44%)	77.32%	77.73% (+0.41%)
		0.2	70.76%	74.65% (+3.89%)	76.68%	77.17% (+0.49%)

Table 1: Model Accuracy under different privacy budgets and corruption settings. Every statistic is averaged over 5 runs with different random seed. The improvement of $\mathrm{TSGD} + \mathrm{PTR}$ over $\mathrm{TSGD} + \mathrm{Gaussian}$ is highlighted in the red text. εs are chosen differently for different datasets since the best accuracy-privacy tradeoff point varied for datasets. 'CR' means corruption ratio.

gradient corruption. For label corruption, we randomly flip of label of certain amount of data points. For feature corruption, we add Gaussian noise from $\mathcal{N}(0, 100)$ directly to the corrupted images. For gradient corruption, we add Gaussian noise from $\mathcal{N}(0, 100)$ to the true gradients. We experiment on three classic datasets, MNIST, CIFAR10, and EMNIST. We vary different corruption ratios and privacy parameter ε. Experiment details are deferred to Appendix.

Results. The comparison of model test accuracy under given corruption settings and privacy parameters are summarized in Table 1. We highlight the improvement of $\mathrm{TSGD} + \mathrm{PTR}$ over the baseline $\mathrm{TSGD} + \mathrm{Gaussian}$ in red texts. As we can see, for all settings, $\mathrm{TSGD} + \mathrm{PTR}$ outperforms and often works significantly better than $\mathrm{TSGD} + \mathrm{Gaussian}$. This demonstrates that, while $\mathrm{TSGD} + \mathrm{PTR}$ may introduce extra privacy loss in the **test** step, the performance gain from adding smaller noise in the **release** step overshadows it.

We also observe two interesting phenomena in the experiment: **(1)** a higher corruption ratio may not necessarily lead to worse model performance for trimmed mean-based robust SGD, especially for MNIST dataset. This is because the high norm gradients can be either corrupted, or benign but the partially-trained model misclassifies the corresponding data points. The latter case is extremely important for improving model performance compared with the gradients of data points are already being classified correctly. When the corruption ratio is small, more benign gradients are being trimmed, which may lead to worse model performance. **(2)** For $\mathrm{TSGD} + \mathrm{Gaussian}$, a larger privacy budget may not lead to better model performance on MNIST dataset. This is because when the training accuracy reaches the peak, there are many trimmed gradients whose corresponding training data points are already correctly classified; continuing training without those data points and with large noise may result in catastrophic forgetting.

6 Conclusion and Limitation

This work derives the Rényi Differential Privacy for propose-test-release framework as well as its subsampled version. With the RDP bound for the PTR framework, this work demonstrate the application of PTR in training differentially private and robust models. One limitation of PTR is that it does not work well in privatizing coordinate-wise median in high-dimensional space. The global sensitivity of coordinate-wise median is far greater than the one for mean, which results in huge privacy loss.

References

[ACG+16] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318. ACM, 2016.

[AHJ+21] Anish Acharya, Abolfazl Hashemi, Prateek Jain, Sujay Sanghavi, Inderjit S Dhillon, and Ufuk Topcu. Robust training in high dimensions via block coordinate geometric median descent. *arXiv preprint arXiv:2106.08882*, 2021.

[ALC+21] Shahab Asoodeh, Jiachun Liao, Flavio P Calmon, Oliver Kosut, and Lalitha Sankar. Three variants of differential privacy: Lossless conversion and applications. *IEEE Journal on Selected Areas in Information Theory*, 2(1):208–222, 2021.

[BAM20] Victor-Emmanuel Brunel and Marco Avella-Medina. Propose, test, release: Differentially private estimation with high probability. *arXiv preprint arXiv:2002.08774*, 2020.

[BBC11] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[BBG18] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in Neural Information Processing Systems*, 31, 2018.

[BBG+20] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2496–2506. PMLR, 2020.

[BDRS18] Mark Bun, Cynthia Dwork, Guy N Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated cdp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–86, 2018.

[BTN00] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3):411–424, 2000.

[CATVS17] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[DKK+17] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, pages 999–1008. PMLR, 2017.

[DL09] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[EMN21] Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Tight and robust private mean estimation with few users. *arXiv preprint arXiv:2110.11876*, 2021.

[GGP+21] Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, Sebastien Rouault, and John Stephan. Combining differential privacy and byzantine resilience in distributed sgd. *arXiv preprint arXiv:2110.03991*, 2021.

[GLV21] Nirupam Gupta, Shuo Liu, and Nitin Vaidya. Byzantine fault-tolerant distributed machine learning with norm-based comparative gradient elimination. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 175–181. IEEE, 2021.

[HD18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

[HKJ20] Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Secure byzantine-robust machine learning. *arXiv preprint arXiv:2006.04747*, 2020.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[KNRS13] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*, pages 457–476. Springer, 2013.

[KOV15] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[Kri09] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[LeC98] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[LKKO21] Xiyang Liu, Weihao Kong, Sham Kakade, and Sewoong Oh. Robust and differentially private mean estimation. *Advances in Neural Information Processing Systems*, 34, 2021.

[LKO21] Xiyang Liu, Weihao Kong, and Sewoong Oh. Differential privacy and robust statistics in high dimensions. *arXiv preprint arXiv:2111.06578*, 2021.

[LQS12] Ninghui Li, Wahbeh Qardaji, and Dong Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 32–33, 2012.

[LSP82] LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[Mir17] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[MTZ19] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[MZH19] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. *arXiv preprint arXiv:1903.09860*, 2019.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[SGA20] Jinhyun So, Başak Güler, and A Salman Avestimehr. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications*, 39(7):2168–2181, 2020.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TS13] Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, pages 819–850, 2013.

[Vad17] Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017.

[VEH14] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[WBK19] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

[XKG19] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*, pages 6893–6901. PMLR, 2019.

[YCKB18] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

[YZCL21] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. *arXiv preprint arXiv:2102.12677*, 2021.

[ZJW18] Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594*, 2018.

[ZW19] Yuqing Zhu and Yu-Xiang Wang. Poission subsampled rényi differential privacy. volume 97 of *Proceedings of Machine Learning Research*, pages 7634–7642, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 4 and 5, in particular Figure 1, 2 and Table 1.
   (b) Did you describe the limitations of your work? [Yes] See Section 6 for a discussion on limitations.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We do not see any potential negative societal impacts of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Appendix A
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide our code in supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix B for full details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report error bars for Table 1 in Appendix B.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B for full details.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite each dataset and model architecture we use, see full details in Appendix B.
   (b) Did you mention the license of the assets? [Yes] See Appendix B.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proofs, Additional Theoretical Results & Discussion

A.1 (ε, δ)-DP and Rényi DP for Propose-Test-Release

We consider two adjacent datasets S, S' where $S' = S \cup \{x\}$. We denote the threshold $B = \log(1/(2\delta_0))b$. Note that we have $\Pr[\mathrm{Lap}(0, b) > B] = \delta_0$. The output of Algorithm 1 on dataset S is a sample from a joint distribution $(\widehat{\Delta}, \mathcal{M})(S)$ where $\widehat{\Delta}(S) = \mathrm{Lap}(\Delta(S), b)$ and $\mathcal{M}(S)|_{\widehat{\Delta}} = \mathcal{N}(f_1(S), \sigma_1^2)\mathbb{1}[\widehat{\Delta} \le B] + \mathcal{N}(f_2(S), \sigma_2^2)\mathbb{1}[\widehat{\Delta} > B]$.

Theorem 4.2 (restated). Suppose $\mathbb{GS}_{f_1} = \mathbb{GS}_{f_2} = 1$ and $\sigma_1 = \sigma_2/\tau$, then Algorithm 1 is $(\varepsilon_{\mathrm{Lap}}^{(b)} + \varepsilon_{\mathcal{N}}^{(\sigma_1)}(\delta), \delta_0 + \delta)$-DP.

Proof. We consider two cases for $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$.

Case 1: both $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ are greater than τ. In this case, we have $\Delta(S) = \Delta(S') = 0$ (recall that Δ refers to the minimum amount of data addition/removal to make the local sensitivity $> \tau$). Therefore, there are no privacy loss in $\widehat{\Delta}$. Besides, the probability that PTR releases $f_2(S) + \mathcal{N}(0, \sigma_2^2)$ is at most $\Pr[\widehat{\Delta} > B] = \Pr[\mathrm{Lap}(0, b) > B] = \delta_0$. Therefore, with probability at least $1 - \delta_0$, the PTR is $(\varepsilon_{\mathcal{N}}^{(\sigma_1)}(\delta), \delta)$-DP, and overall it is $(\varepsilon_{\mathcal{N}}^{(\sigma_1)}(\delta), \delta + \delta_0)$-DP.

Case 2: at least one of $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ are smaller than τ. In this case, we know that $\Pr[\mathcal{M}(S) \in T] \le e^{\varepsilon_{\mathcal{N}}^{(\sigma_1)}(\delta)} \Pr[\mathcal{M}(S') \in T] + \delta$ regardless of the value of $\widehat{\Delta}$. Thus, by basic composition theorem, PTR in this case is $\left(\varepsilon_{\mathrm{Lap}}^{(b)} + \varepsilon_{\mathcal{N}}^{(\sigma_1)}(\delta), \delta\right)$-DP.

Therefore, PTR is $(\varepsilon_{\mathrm{Lap}}^{(b)} + \varepsilon_{\mathcal{N}}^{(\sigma_1)}(\delta), \delta_0 + \delta)$-DP overall. □

Comparison between (ε, δ)-DP and RDP Analysis: A motivating example (expanded). Suppose we have two mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ who are $(\varepsilon_1, \delta_1)$-DP and $(\varepsilon_2, \delta_2)$-DP, respectively. Consider a simple PTR-like mechanism $\mathcal{M}$ that randomly picks one of mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ to run, each with probability $1 - \delta_0$ and δ_0[7]. A straightforward (ε, δ)-DP analysis for $\mathcal{M}$ can be given as follows: for any possible event T,

$$\begin{aligned}
\Pr[\mathcal{M}(S) \in T] &= (1 - \delta_0) \Pr[\mathcal{M}_1(S) \in T] + \delta_0 \Pr[\mathcal{M}_2(S) \in T] && (3)\\
&\le (1 - \delta_0)[e^{\varepsilon_1} \Pr[\mathcal{M}_1(S') \in T] + \delta_1] + \delta_0[e^{\varepsilon_2} \Pr[\mathcal{M}_2(S') \in T] + \delta_2] && (4)\\
&= e^{\varepsilon_1}(1 - \delta_0) \Pr[\mathcal{M}_1(S') \in T] + e^{\varepsilon_2}\delta_0 \Pr[\mathcal{M}_2(S') \in T] + (1 - \delta_0)\delta_1 + \delta_0\delta_2 && (5)\\
&\le e^{\max(\varepsilon_1, \varepsilon_2)} \Pr[\mathcal{M}(S) \in T] + (1 - \delta_0)\delta_1 + \delta_0\delta_2 && (6)
\end{aligned}$$

That is, $\mathcal{M}$ is $(\max(\varepsilon_1, \varepsilon_2), (1 - \delta_0)\delta_1 + \delta_0\delta_2)$-DP. Without further information, this bound is the best we can do since it is tight when there exists event T such that $\Pr[\mathcal{M}_1(S') \in T] = 0$ while $\Pr[\mathcal{M}_2(S') \in T] > 0$. Alternatively, if we know $\varepsilon_2 \gg \varepsilon_1$ we can also move the probability δ_0 to the δ term and obtain $(\varepsilon_1, \delta_0 + \delta_1)$ (which is the case for Theorem 4.2).

However, if we know the RDP guarantee of $\mathcal{M}_1$ and $\mathcal{M}_2$ as $E_\alpha(\mathcal{M}_1(S) \| \mathcal{M}_1(S')) \le f_\alpha(\varepsilon_1)$ and $E_\alpha(\mathcal{M}_2(S) \| \mathcal{M}_2(S')) \le f_\alpha(\varepsilon_2)$[8], then $E_\alpha(\mathcal{M}(S) \| \mathcal{M}(S'))$ can be simply bounded as

$$\begin{aligned}
\mathbb{E}_{\mathcal{M}(S)}\left[\left(\frac{\mu_{\mathcal{M}(S')}}{\mu_{\mathcal{M}(S)}}\right)^\alpha\right] &= (1 - \delta_0)\mathbb{E}_{\mathcal{M}_1(S)}\left[\left(\frac{\mu_{\mathcal{M}_1(S')}}{\mu_{\mathcal{M}_1(S)}}\right)^\alpha\right] + \delta_0\mathbb{E}_{\mathcal{M}_2(S)}\left[\left(\frac{\mu_{\mathcal{M}_2(S')}}{\mu_{\mathcal{M}_2(S)}}\right)^\alpha\right] && (7)\\
&\le (1 - \delta_0) f_\alpha(\varepsilon_1) + \delta_0 f_\alpha(\varepsilon_2) && (8)
\end{aligned}$$

Compared with (ε, δ)-DP analysis, there are no extra inequalities used in RDP analysis of $\mathcal{M}$ except for the RDP guarantee of $\mathcal{M}_1$ and $\mathcal{M}_2$. Thus, RDP is more favorable in for PTR's privacy analysis, especially when δ_0 is close to the target δ.

[7] For the actual PTR, the δ_0 is not fixed but depends on the input dataset.

[8] Recall that $f_\alpha(\varepsilon) = \exp((\alpha - 1)\varepsilon)$ where if $\mathcal{M}$ is (α, ε)-RDP then $E_\alpha(\mathcal{M}(S) \| \mathcal{M}(S')) \le f_\alpha(\varepsilon)$.

Theorem 4.3 (restated). Suppose $\mathbb{GS}_{f_1} = \mathbb{GS}_{f_2} = 1$ and $\sigma_1 = \sigma_2/\tau$. Then for any $\alpha > 1$, Algorithm 1 is $(\alpha, \varepsilon_{\mathrm{PTR}}(\alpha))$-RDP for

$$\varepsilon_{\mathrm{PTR}}(\alpha) \leq \max\left(f_\alpha^{-1}\left((1-\delta_0)f_\alpha\left(\varepsilon_{\mathrm{R-}\mathcal{N}}^{(\sigma_1)}(\alpha)\right) + \delta_0 f_\alpha\left(\varepsilon_{\mathrm{R-}\mathcal{N}}^{(\sigma_2)}(\alpha)\right)\right), \varepsilon_{\mathrm{R-}\mathcal{N}}^{(\sigma_1)}(\alpha) + \varepsilon_{\mathrm{R-Lap}}^{(b)}(\alpha)\right)$$

Proof. We will denote the density of $(\widehat{\Delta}, \mathcal{M})(S)$ as μ and that of $(\widehat{\Delta}, \mathcal{M})(S')$ as μ'. We will use $\mu(s,t)$ to denote the joint density on the pair of outputs (s,t), where $s \sim \widehat{\Delta}(S)$ and $t \sim \mathcal{M}(S)|_{\widehat{\Delta}}$. Furthermore, when we write $\mu(s)$ it refers to the marginal density of μ on s, and $\mu(t|s)$ refers to the conditional density on t given s.

In order to bound RDP of PTR with order α, it suffices to bound the moments $\mathbb{E}_{(s,t)\sim\mu}\left[\left(\frac{\mu'(s,t)}{\mu(s,t)}\right)^\alpha\right]$ and $\mathbb{E}_{(s,t)\sim\mu'}\left[\left(\frac{\mu(s,t)}{\mu'(s,t)}\right)^\alpha\right]$ then take the bigger of the two bounds. For readability, we may abbreviate the two quantities as $\mathbb{E}_\mu\left[\left(\frac{\mu'}{\mu}\right)^\alpha\right]$ and $\mathbb{E}_{\mu'}\left[\left(\frac{\mu}{\mu'}\right)^\alpha\right]$. We do the following to decompose $\mathbb{E}_{(s,t)\sim\mu}\left[\left(\frac{\mu'(s,t)}{\mu(s,t)}\right)^\alpha\right]$:

$$\mathbb{E}_{(s,t)\sim\mu}\left[\left(\frac{\mu'(s,t)}{\mu(s,t)}\right)^\alpha\right] \tag{9}$$

$$= \mathbb{E}_{(s,t)\sim\mu}\left[\left(\frac{\mu'(s)\mu'(t|s)}{\mu(s)\mu(t|s)}\right)^\alpha\right] \tag{10}$$

$$= \mathbb{E}_{s\sim\mu}\left[\left(\frac{\mu'(s)}{\mu(s)}\right)^\alpha \mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\right] \tag{11}$$

$$= \mathbb{E}_{s\sim\mu}\left[\left(\frac{\mu'(s)}{\mu(s)}\right)^\alpha \left(\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\mathbb{1}[s \leq B] + \mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\mathbb{1}[s > B]\right)\right] \tag{12}$$

$$\tag{13}$$

When $s \leq B$, we know that

$$\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right] = \mathbb{E}_{t\sim\mathcal{N}(f_1(S),\sigma_1^2\mathbf{1}_d)}\left[\left(\frac{\mathcal{N}(t; f_1(S'), \sigma_1^2\mathbf{1}_d)}{\mathcal{N}(t; f_1(S), \sigma_1^2\mathbf{1}_d)}\right)^\alpha\right] \tag{14}$$

$$= \mathbb{E}_{t\sim\mathcal{N}(\mathbf{0},\sigma_1^2\mathbf{1}_d)}\left[\left(\frac{\mathcal{N}(t; f_1(S') - f_1(S), \sigma_1^2\mathbf{1}_d)}{\mathcal{N}(t; \mathbf{0}, \sigma_1^2\mathbf{1}_d)}\right)^\alpha\right] \tag{15}$$

$$= \mathbb{E}_{t\sim\mathcal{N}(0,\sigma_1^2)}\left[\left(\frac{\mathcal{N}(t; \|f_1(S') - f_1(S)\|, \sigma_1^2)}{\mathcal{N}(t; 0, \sigma_1^2)}\right)^\alpha\right] \tag{16}$$

$$\leq \mathbb{E}_{t\sim\mathcal{N}(0,\sigma_1^2)}\left[\left(\frac{\mathcal{N}(t; 1, \sigma_1^2)}{\mathcal{N}(t; 0, \sigma_1^2)}\right)^\alpha\right] \tag{17}$$

$$= f_\alpha\left(\varepsilon_{\mathrm{R-}\mathcal{N}}^{(\sigma_1)}(\alpha)\right) \tag{18}$$

where (15) is due to the translation invariance of Rényi divergence, (16) is due to the rotation trick, (17) is because of $\|f_1(S') - f_1(S)\| \leq 1$.

We now analyze the upper bound of $\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]$ when $s > B$ by considering two separate cases: when both $\mathbb{LS}_{f_2}(S) > \tau$ and $\mathbb{LS}_{f_2}(S') > \tau$, and when there is at least one of $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ is greater than τ.

Case 1: both $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ are greater than τ. In this case, the only known upper bound of $\|f_2(S) - f_2(S')\|$ is the global sensitivity $\mathbb{GS}_{f_2} = 1$. Therefore, we only have

$\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right] \le f_\alpha\left(\varepsilon^{(\sigma_2)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)$ when $s > B$. Therefore, in this case we have

$$\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\mathbb{1}[s\le B] + \mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\mathbb{1}[s> B] \tag{19}$$

$$= f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\mathbb{1}[s\le B] + f_\alpha\left(\varepsilon^{(\sigma_2)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\mathbb{1}[s> B] \tag{20}$$

However, note that when both $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ is greater than τ, we have $\Delta(S) = \Delta(S') = 0$, which means that there is no privacy loss by releasing the result of $\widehat{\Delta}(S)$ or $\widehat{\Delta}(S')$. Therefore, we have $\mu(s) = \mu'(s) = \mathrm{Lap}(s; 0, b)$, and thus

$$\mathbb{E}_{(s,t)\sim\mu}\left[\left(\frac{\mu'(s,t)}{\mu(s,t)}\right)^\alpha\right] \tag{21}$$

$$= \mathbb{E}_{s\sim\mu}\left[f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\mathbb{1}[s\le B] + f_\alpha\left(\varepsilon^{(\sigma_2)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\mathbb{1}[s> B]\right] \tag{22}$$

$$= f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\Pr[\mathrm{Lap}(0,b)\le B] + f_\alpha\left(\varepsilon^{(\sigma_2)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\Pr[\mathrm{Lap}(0,b)> B] \tag{23}$$

$$= (1-\delta_0)f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right) + \delta_0 f_\alpha\left(\varepsilon^{(\sigma_2)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right) \tag{24}$$

Case 2: at least one of $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ are smaller than τ. In this case, we know that we have $\|f_2(S) - f_2(S')\| \le \tau$. Thus, when $s \ge B$, we have

$$\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right] = \mathbb{E}_{t\sim\mathcal{N}(0,\sigma_2^2)}\left[\left(\frac{\mathcal{N}(t;\|f_2(S')-f_2(S)\|,\sigma_2^2)}{\mathcal{N}(t;0,\sigma_2^2)}\right)^\alpha\right] \tag{25}$$

$$\le \mathbb{E}_{t\sim\mathcal{N}(0,\sigma_2^2)}\left[\left(\frac{\mathcal{N}(t;\tau,\sigma_2^2)}{\mathcal{N}(t;0,\sigma_2^2)}\right)^\alpha\right] \tag{26}$$

$$= f_\alpha\left(\varepsilon^{(\sigma_2/\tau)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right) \tag{27}$$

Thus, we have

$$\mathbb{E}_{(s,t)\sim\mu}\left[\left(\frac{\mu'(s,t)}{\mu(s,t)}\right)^\alpha\right] \tag{28}$$

$$\le \mathbb{E}_{s\sim\mu}\left[\left(\frac{\mu'(s)}{\mu(s)}\right)^\alpha\left(f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\mathbb{1}[s\le B] + f_\alpha\left(\varepsilon^{(\sigma_2/\tau)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\mathbb{1}[s> B]\right)\right] \tag{29}$$

$$= \mathbb{E}_{s\sim\mu}\left[\left(\frac{\mu'(s)}{\mu(s)}\right)^\alpha f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\right] \tag{30}$$

$$= f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\mathbb{E}_{s\sim\mu}\left[\left(\frac{\mu'(s)}{\mu(s)}\right)^\alpha\right] \tag{31}$$

$$\le f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right) f_\alpha(\varepsilon^{(b)}_{\mathrm{R-Lap}}(\alpha)) \tag{32}$$

where (30) is because by our condition, $\sigma_1 = \sigma_2/\tau$.

Therefore, we have

$$\begin{aligned}
&D_\alpha(\mu'\|\mu)\\
&\le \frac{1}{\alpha-1}\log\left(\max((1-\delta_0)f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right) + \delta_0 f_\alpha\left(\varepsilon^{(\sigma_2)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right), f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right) f_\alpha(\varepsilon^{(b)}_{\mathrm{R-Lap}}(\alpha)))\right)\\
&= \max\left(\frac{1}{\alpha-1}\log\left((1-\delta_0)f_\alpha\left(\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right) + \delta_0 f_\alpha\left(\varepsilon^{(\sigma_2)}_{\mathrm{R-}\mathcal{N}}(\alpha)\right)\right), \varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha) + \varepsilon^{(b)}_{\mathrm{R-Lap}}(\alpha)\right)
\end{aligned}$$

Since we did not use any condition that depends on the fact that $S' = S \cup \{x\}$, we know that $D_\alpha(\mu\|\mu')$ also has the exactly the same upper bound, which leads to the conclusion. □

A.1.1 Discussion: can we improve privacy analysis by not releasing $\widehat{\Delta}$?

One may wonder if we can further improve the privacy analysis of PTR by not releasing $\widehat{\Delta}$. However, releasing $\widehat{\Delta}$ is essential for the applications of PTR. The rationale behind PTR is to exploit the fact that, while a function's global sensitivity may be large, its local sensitivity may be much smaller for most of the "common inputs". Thus, such a mechanism will only be preferred over a regular output perturbation mechanism when the local sensitivity of data drawn from input data distribution rarely exceeds the threshold. Without knowing about $\widehat{\Delta}$, the user **cannot** know whether they are actually enjoying the benefits from PTR or simply wasting privacy budgets on private sensitivity tests. Furthermore, the user cannot adjust the hyperparameters or switch algorithms accordingly. Notably, in Section 5 (the application of PTR in privatizing robust SGD), we also use the information from $\widehat{\Delta}$ to dynamically adjust the number of gradients to be trimmed (note that this does not affect privacy analysis since the adjustment is post-processing of $\widehat{\Delta}$).

Besides, we gave an attempt to directly analyze the variant of PTR that does not release $\hat{\Delta}$, and we do not see an easy way to obtain a better privacy bound than we have in Theorem 4.3.

We follow the same notations as in the proof of Theorem 4.3: Given a pair of neighboring dataset S, S', we denote the density of $\mathcal{M}(S)$ as μ and that of $\mathcal{M}(S')$ as μ'. Given $s \sim \widehat{\Delta}(S)$, we denote $\mu(t|s \leq B)$ the density of $\mathcal{N}(f_1(S), \sigma_1^2)$, and $\mu(t|s > B)$ the density of $\mathcal{N}(f_2(S), \sigma_2^2)$. $\mu'(t|s \leq B)$ and $\mu'(t|s > B)$ are defined analogously.

Similar to the proof of Theorem 4.3, we consider two separate cases: when both $\mathbb{LS}_{f_2}(S) > \tau$ and $\mathbb{LS}_{f_2}(S') > \tau$, and when there is at least one of $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ is greater than τ.

Case 1. For the case that both $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ are greater than τ, from the proof of Theorem 4.3 we know that $\widehat{\Delta}(S)$ and $\widehat{\Delta}(S')$ has exactly the same distribution since $\Delta(S) = \Delta(S') = 0$. Thus, the exactly the same proof in Theorem 4.3 applies for the case of not releasing $\widehat{\Delta}$.

Case 2. For the case that at least one of $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ is smaller than τ, here's our attempt:

$$E_\alpha(\mathcal{M}(S)\|\mathcal{M}(S')) = \mathbb{E}_{t\sim\mu}\left[\left(\frac{\mu'(t)}{\mu(t)}\right)^\alpha\right] \tag{33}$$

$$= \mathbb{E}_{t\sim\mu}\left[\left(\frac{\mu'(t|s \leq B)\Pr[\widehat{\Delta}(S') \leq B] + \mu'(t|s > B)\Pr[\widehat{\Delta}(S') > B]}{\mu(t|s \leq B)\Pr[\widehat{\Delta}(S) \leq B] + \mu(t|s > B)\Pr[\widehat{\Delta}(S) > B]}\right)^\alpha\right] \tag{34}$$

As we can see, while the distribution of $\mathcal{M}(S)$ is a Gaussian mixture, the probability for different components is also depending on S, which introduce more challenge in bounding $E_\alpha(\mathcal{M}(S)\|\mathcal{M}(S'))$. One relatively simple way to bound the above expression is by noticing that since $\widehat{\Delta} = \Delta + \mathrm{Lap}(0, b)$, by the privacy guarantee of Laplace mechanism we have $\Pr[\widehat{\Delta}(S') \leq B] \leq e^{1/b}\Pr[\widehat{\Delta}(S) \leq B]$ and $\Pr[\widehat{\Delta}(S') > B] \leq e^{1/b}\Pr[\widehat{\Delta}(S) > B]$. Thus, we have

$$(34) \leq \exp\left(\frac{\alpha}{b}\right)\mathbb{E}_{t\sim\mu}\left[\left(\frac{\mu'(t|s \leq B)\Pr[\widehat{\Delta}(S) \leq B] + \mu'(t|s > B)\Pr[\widehat{\Delta}(S) > B]}{\mu(t|s \leq B)\Pr[\widehat{\Delta}(S) \leq B] + \mu(t|s > B)\Pr[\widehat{\Delta}(S) > B]}\right)^\alpha\right] \tag{35}$$

$$\leq \exp\left(\frac{\alpha}{b}\right)\mathbb{E}_{t\sim\mu}\left[\left(\frac{\mu'(t|s \leq B)}{\mu(t|s \leq B)}\right)^\alpha\right] \tag{36}$$

where the last inequality is due to the quasi-convexity of Renyi divergence [VEH14] (note that $\mathbb{E}_{t\sim\mu}\left[\left(\frac{\mu'(t|s\leq B)}{\mu(t|s\leq B)}\right)^\alpha\right] = \mathbb{E}_{t\sim\mu}\left[\left(\frac{\mu'(t|s>B)}{\mu(t|s>B)}\right)^\alpha\right]$ by construction for this case). Thus, we have

$$R_\alpha(\mathcal{M}(S)\|\mathcal{M}(S')) \leq \varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha) + \frac{1}{\alpha - 1}\left(\frac{\alpha}{b}\right) \tag{37}$$

for the case of at least one of $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ are smaller than τ.

Now we show that this bound is not as good as the corresponding bound in Theorem 4.3. The corresponding bound in Theorem 4.3 for this case is $\varepsilon^{(\sigma_1)}_{\mathrm{R-}\mathcal{N}}(\alpha) + \varepsilon^{(b)}_{\mathrm{R-Lap}}(\alpha)$, so we only need to

show $\varepsilon^{(b)}_{\mathrm{R-Lap}}(\alpha) < \frac{1}{\alpha-1}\left(\frac{\alpha}{b}\right)$.

$$\varepsilon^{(b)}_{\mathrm{R-Lap}}(\alpha) = \frac{1}{\alpha-1}\log\left(\frac{\alpha}{2\alpha-1}\exp\left(\frac{\alpha-1}{b}\right) + \frac{\alpha-1}{2\alpha-1}\exp\left(-\frac{\alpha}{b}\right)\right) \tag{38}$$

$$< \frac{1}{\alpha-1}\log\left(\frac{\alpha}{2\alpha-1}\exp\left(\frac{\alpha}{b}\right) + \frac{\alpha-1}{2\alpha-1}\exp\left(\frac{\alpha}{b}\right)\right) \tag{39}$$

$$= \frac{1}{\alpha-1}\log\left(\exp\left(\frac{\alpha}{b}\right)\right) \tag{40}$$

$$= \frac{1}{\alpha-1}\left(\frac{\alpha}{b}\right) \tag{41}$$

where the first inequality is due to $\exp\left(\frac{\alpha-1}{b}\right) < \exp\left(\frac{\alpha}{b}\right)$ and $\exp\left(-\frac{\alpha}{b}\right) < \exp\left(\frac{\alpha}{b}\right)$.

Thus, we think at least there are no simple solution for improving the privacy bound for PTR by not releasing $\widehat{\Delta}$. However, even if there are a better way to derive the privacy bound, this variant of PTR may not be user-friendly as the counterpart who release $\widehat{\Delta}$.

A.2 Rényi DP for Subsampled Propose-Test-Release

We consider two adjacent datasets S, S' where $S' = S \cup \{x\}$. We denote the threshold $B = \log(1/(2\delta_0))b$. Note that we have $\Pr[\mathrm{Lap}(0,b) > B] = \delta_0$. The output of Algorithm 1 on dataset S is a sample from a joint distribution $(\widehat{\Delta}, \mathcal{M})(S)$ where $\widehat{\Delta}(S) = \mathrm{Lap}(\Delta(S), b)$ and $\mathcal{M}(S)|_{\widehat{\Delta}} = \mathcal{N}(f_1(S), \sigma_1^2)\mathbb{1}[\widehat{\Delta} \le B] + \mathcal{N}(f_2(S), \sigma_2^2)\mathbb{1}[\widehat{\Delta} > B]$.

Theorem 4.4 (full version). Let q be the subsampling probability. Suppose $\mathbb{GS}_{f_1} = \mathbb{GS}_{f_2} = 1$ and $\sigma_1 = \sigma_2/\tau$. If $q \le \frac{\exp(-1/b)}{4+\exp(-1/b)}$ and $\sigma_1 \ge \sigma_2 \ge 4$, and α satisfy $1 < \alpha \le \frac{1}{2}\sigma_2^2 L - 2\ln\sigma_2, \alpha \le \frac{\frac{1}{2}\sigma_2^2 L^2 - \ln 5 - 2\ln\sigma_2}{L + \ln(q'\alpha) + 1/\left(2\sigma_2^2\right)}$, where $L = \ln\left(1 + \frac{1}{q'(\alpha-1)}\right)$ and $q' = \frac{q}{q+(1-q)\exp(-1/b)}$, we have

$$\varepsilon_{\mathrm{PTR}\circ\mathrm{PoissonSample}}(\alpha) \le f_\alpha^{-1}\left(\max(\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2)\right)$$

where

$$\mathcal{B}_0 = 1 + 2q^2\alpha(\alpha-1)\left(\frac{1-\delta_0}{\sigma_1^2} + \frac{\delta_0}{\sigma_2^2}\right) \tag{42}$$

$$\mathcal{B}_1 = \mathrm{R}_q^{(\alpha)} + \frac{2\alpha(\alpha-1)}{\sigma_1^2}\left[\mathrm{R}_q^{(\alpha)} - 2(1-q)\mathrm{R}_q^{(\alpha-1)} + (1-q)^2\mathrm{R}_q^{(\alpha-2)}\right] \tag{43}$$

$$\mathcal{B}_2 = \widetilde{\mathrm{R}}_q^{(\alpha)} + \frac{2\alpha(\alpha-1)}{\sigma_1^2}\left[\widetilde{\mathrm{R}}_q^{(\alpha)} - 2(1-q)\widetilde{\mathrm{R}}_q^{(\alpha+1)} + (1-q)^2\widetilde{\mathrm{R}}_q^{(\alpha+2)}\right] \tag{44}$$

with $\mathrm{R}_q^{(\alpha)} = \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha\right]$ and $\widetilde{\mathrm{R}}_q^{(\alpha)} = \mathbb{E}_{s\sim\mu}\left[\left(\frac{\mu_0(s)}{\mu(s)}\right)^\alpha\right]$ for $\mu_0 \sim \mathrm{Lap}(0,b)$ and $\mu \sim (1-q)\mathrm{Lap}(0,b) + q\mathrm{Lap}(1,b)$.

Proof. Let T denote a set-valued random variable defined by taking a random subset of S, where each element of S is independently placed in T with probability q. Conditioned on T, the PTR outputs $(\widehat{\Delta}, \mathcal{M})(T)$. Thus,

$$\left(\widehat{\Delta}, \mathcal{M}\right)(S) = \sum_{T\subseteq S} p_T \cdot \left(\widehat{\Delta}, \mathcal{M}\right)(T) \tag{45}$$

$$\left(\widehat{\Delta}, \mathcal{M}\right)(S') = \sum_{T\subseteq S} p_T \cdot \left((1-q)\cdot(\widehat{\Delta}, \mathcal{M})(T) + q\cdot(\widehat{\Delta}, \mathcal{M})(T\cup\{x\})\right) \tag{46}$$

where p_T denotes the probabilty of sampling the subset T.

$$
\begin{aligned}
&D_\alpha\left(\left(\widehat{\Delta},\mathcal{M}\right)(S')\|\left(\widehat{\Delta},\mathcal{M}\right)(S)\right)\\
&=D_\alpha\left(\sum_T p_T\cdot\left((1-q)\cdot(\widehat{\Delta},\mathcal{M})(T)+q\cdot(\widehat{\Delta},\mathcal{M})(T\cup\{x\})\right)\|\sum_T p_T\cdot\left(\widehat{\Delta},\mathcal{M}\right)(T)\right)\\
&\le\sup_T D_\alpha\left((1-q)\cdot(\widehat{\Delta},\mathcal{M})(T)+q\cdot(\widehat{\Delta},\mathcal{M})(T\cup\{x\})\|\left(\widehat{\Delta},\mathcal{M}\right)(T)\right)
\end{aligned}
$$

where the last step is due to the quasi-convexity of Rényi divergence ([VEH14], Theorem 13). Symmetrically, we also have

$$
\begin{aligned}
&D_\alpha\left(\left(\widehat{\Delta},\mathcal{M}\right)(S')\|\left(\widehat{\Delta},\mathcal{M}\right)(S)\right) && (47)\\
&\le\sup_T D_\alpha\left(\left(\widehat{\Delta},\mathcal{M}\right)(T)\|(1-q)\cdot(\widehat{\Delta},\mathcal{M})(T)+q\cdot(\widehat{\Delta},\mathcal{M})(T\cup\{x\})\right) && (48)
\end{aligned}
$$

Fix a subset T and denote $T'=T\cup\{x\}$. We use μ_0 to denote the density function of $(\widehat{\Delta},\mathcal{M})(T)$, where $\mu_0(s,t)$ refers to the density on (s,t). We use μ_1 to denote the density function of $(\widehat{\Delta},\mathcal{M})(T')$, where $\mu_1(s,t)$ refers to the density on (s,t). Let $\mu=(1-q)\mu_0+q\mu_1$. We want to bound $\mathbb{E}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^\alpha\right]$ and $\mathbb{E}_{\mu}\left[\left(\frac{\mu_0}{\mu}\right)^\alpha\right]$.

We first bound $\mathbb{E}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^\alpha\right]$, which is usually considered as an easier one.

By decomposition, we have

$$
\begin{aligned}
\mathbb{E}_{s,t\sim\mu_0}\left[\left(\frac{\mu(s,t)}{\mu_0(s,t)}\right)^\alpha\right]&=\mathbb{E}_{s,t\sim\mu_0}\left[\left(\frac{\mu(s)\mu(t|s)}{\mu_0(s)\mu_0(t|s)}\right)^\alpha\right] && (49)\\
&=\mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha\mathbb{E}_{t\sim\mu_0(\cdot|s)}\left[\left(\frac{\mu(t|s)}{\mu_0(t|s)}\right)^\alpha\right]\right] && (50)
\end{aligned}
$$

For the density of conditional distribution $\mu(t|s)$, we have

$$
\begin{aligned}
\mu(t|s)&=\frac{\mu(s,t)}{\mu(s)} && (51)\\
&=\frac{(1-q)\mu_0(s,t)+q\mu_1(s,t)}{\mu(s)} && (52)\\
&=I[s\le B]\cdot\frac{(1-q)\mu_0(s)\cdot\mathcal{N}(t;f_1(T),\sigma_1^2)+q\mu_1(s)\cdot\mathcal{N}(t;f_1(T'),\sigma_1^2)}{\mu(s)} && (53)\\
&\quad+I[s>B]\cdot\frac{(1-q)\mu_0(s)\cdot\mathcal{N}(t;f_2(T),\sigma_2^2)+q\mu_1(s)\cdot\mathcal{N}(t;f_2(T'),\sigma_2^2)}{\mu(s)} && (54)
\end{aligned}
$$

Denote $A(s)=\frac{q\mu_1(s)}{\mu(s)}$. Recall that $\mu(s)=(1-q)\mu_0(s)+q\mu_1(s)$, so we have $\frac{(1-q)\mu_0(s)}{\mu(s)}=1-A$. Then we have

$$
\begin{aligned}
\mu(t|s)&=I[s\le B]\left((1-A)\cdot\mathcal{N}(t;f_1(T),\sigma_1^2)+A\cdot\mathcal{N}(t;f_1(T'),\sigma_1^2)\right) && (55)\\
&\quad+I[s>B]\left((1-A)\cdot\mathcal{N}(t;f_2(T),\sigma_2^2)+A\cdot\mathcal{N}(t;f_2(T'),\sigma_2^2)\right) && (56)
\end{aligned}
$$

and we know that

$$
\mu_0(t|s)=I[s\le B]\mathcal{N}(t;f_1(T),\sigma_1^2)+I[s>B]\mathcal{N}(t;f_2(T),\sigma_2^2) \quad (57)
$$

Therefore we have

$$\mathbb{E}_{t\sim\mu_0(\cdot|s)}\left[\left(\frac{\mu(t|s)}{\mu_0(t|s)}\right)^\alpha\right] \tag{58}$$

$$= \mathbb{E}_{t\sim\mu_0(\cdot|s)}\left[\left(\frac{\mu(t|s)}{\mu_0(t|s)}\right)^\alpha I[s\le B] + \left(\frac{\mu(t|s)}{\mu_0(t|s)}\right)^\alpha I[s>B]\right] \tag{59}$$

$$= \mathbb{E}_{t\sim\mu_0(\cdot|s)}\left[\left(\frac{\mu(t|s)}{\mu_0(t|s)}\right)^\alpha\right] I[s\le B] + \mathbb{E}_{t\sim\mu_0(\cdot|s)}\left[\left(\frac{\mu(t|s)}{\mu_0(t|s)}\right)^\alpha\right] I[s>B] \tag{60}$$

$$= I[s\le B]\mathbb{E}_{t\sim\mathcal{N}(f_1(T),\sigma_1^2)}\left[\left((1-A)+A\cdot\frac{\mathcal{N}(t;f_1(T'),\sigma_1^2)}{\mathcal{N}(t;f_1(T),\sigma_1^2)}\right)^\alpha\right] \tag{61}$$

$$+ I[s>B]\mathbb{E}_{t\sim\mathcal{N}(f_2(T),\sigma_2^2)}\left[\left((1-A)+A\cdot\frac{\mathcal{N}(t;f_2(T'),\sigma_2^2)}{\mathcal{N}(t;f_2(T),\sigma_2^2)}\right)^\alpha\right] \tag{62}$$

Note that $\mathbb{E}_{t\sim\mathcal{N}(f_1(T),\sigma_1^2)}\left[\left((1-A)+A\cdot\frac{\mathcal{N}(t;f_1(T'),\sigma_1^2)}{\mathcal{N}(t;f_1(T),\sigma_1^2)}\right)^\alpha\right]$ can be exactly bounded by the Rényi DP of subsampled RDP with sampling probability A.

Lemma A.1 ([MTZ19], Theorem 11)**.** *If $q\le\frac{1}{5}, \sigma\ge 4$, and α satisfy $1<\alpha\le\frac{1}{2}\sigma^2L-2\ln\sigma$, $\alpha\le\frac{\frac{1}{2}\sigma^2L^2-\ln 5-2\ln\sigma}{L+\ln(q\alpha)+1/(2\sigma^2)}$ where $L=\ln\left(1+\frac{1}{q(\alpha-1)}\right)$, then for any function f with ℓ_2-sensitivity τ satisfies*

$$\mathbb{E}_{t\sim\mathcal{N}(f(T),\sigma^2)}\left[\left((1-q)+q\cdot\frac{\mathcal{N}(t;f(T'),\sigma^2)}{\mathcal{N}(t;f_2(T),\sigma^2)}\right)^\alpha\right] \le 1+2q^2\tau^2\alpha(\alpha-1)/\sigma^2 \tag{63}$$

Similar to the proof of Theorem 4.3, we consider two cases:

Case 1: both $\mathbb{LS}_{f_2}(T)$ and $\mathbb{LS}_{f_2}(T')$ are greater than τ. In this case, the only known upper bound of $\|f_2(T)-f_2(T')\|$ is the global sensitivity $\mathbb{GS}_{f_2}=1$. Therefore, we only have $\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\le 1+2A^2\alpha(\alpha-1)/\sigma_2^2$ when $s>B$. Therefore, in this case we have

$$\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\mathbb{1}[s\le B] + \mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\mathbb{1}[s>B] \tag{64}$$

$$= (1+2A^2\alpha(\alpha-1)/(\sigma_1)^2)\mathbb{1}[s\le B] + (1+2A^2\alpha(\alpha-1)/(\sigma_2)^2)\mathbb{1}[s>B] \tag{65}$$

$$= 1+2A^2\alpha(\alpha-1)\left(\frac{\mathbb{1}[s\le B]}{\sigma_1^2}+\frac{\mathbb{1}[s>B]}{\sigma_2^2}\right) \tag{66}$$

However, note that when both $\mathbb{LS}_{f_2}(S)$ and $\mathbb{LS}_{f_2}(S')$ is greater than τ, we have $\Delta(S)=\Delta(S')=0$. Therefore, we have $\mu(s)=\mu'(s)=\mathrm{Lap}(s;0,b)$, $A=q$, and thus

$$\mathbb{E}_{(s,t)\sim\mu}\left[\left(\frac{\mu'(s,t)}{\mu(s,t)}\right)^\alpha\right] \tag{67}$$

$$= \mathbb{E}_{s\sim\mu}\left[1+2q^2\alpha(\alpha-1)\left(\frac{\mathbb{1}[s\le B]}{\sigma_1^2}+\frac{\mathbb{1}[s>B]}{\sigma_2^2}\right)\right] \tag{68}$$

$$= 1+2q^2\alpha(\alpha-1)\left(\frac{1-\delta_0}{\sigma_1^2}+\frac{\delta_0}{\sigma_2^2}\right) \tag{69}$$

Case 2: at least one of $\mathbb{LS}_{f_2}(T)$ and $\mathbb{LS}_{f_2}(T')$ are smaller than τ. In this case, we know that we have $\|f_2(T)-f_2(T')\|\le\tau$. Since $A=\frac{q\mu_1(s)}{(1-q)\mu_0(s)+q\mu_1(s)}\le\frac{q}{(1-q)+q\exp(-1/b)}$ which satisfy the conditions in Lemma A.1 by our assumption, when $s\ge B$, we have $\mathbb{E}_{t\sim\mu|s}\left[\left(\frac{\mu'(t|s)}{\mu(t|s)}\right)^\alpha\right]\le$

$1 + 2A^2\alpha(\alpha-1)/(\sigma_2/\tau)^2$. Thus we have

$$\mathbb{E}_{s,t\sim\mu_0}\left[\left(\frac{\mu(s,t)}{\mu_0(s,t)}\right)^\alpha\right] \tag{70}$$

$$= \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha \mathbb{E}_{t\sim\mu_0(\cdot|s)}\left[\left(\frac{\mu(t|s)}{\mu_0(t|s)}\right)^\alpha\right]\right] \tag{71}$$

$$\leq \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha \left(1 + 2A^2\alpha(\alpha-1)\left(\frac{\mathbb{1}[s\leq B]}{\sigma_1^2} + \frac{\mathbb{1}[s>B]}{(\sigma_2/\tau)^2}\right)\right)\right] \tag{72}$$

$$= \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha\right] + 2\alpha(\alpha-1)\mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha A^2\left(\frac{I[s\leq B]}{\sigma_1^2} + \frac{I[s>B]}{(\sigma_2/\tau)^2}\right)\right] \tag{73}$$

$$= \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha\right] + \frac{2\alpha(\alpha-1)}{\sigma_1^2}\mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha A^2\right] \tag{74}$$

Denote $\mathrm{R}_q^{(\alpha)} = \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha\right]$, which is the RDP of subsampled Laplace mechanism with sampling rate q (note that $\mu(s) = (1-q)\mu_0(s) + q\mu_1(s)$).

Since

$$A = \frac{q\mu_1(s)}{\mu(s)} = 1 - \frac{(1-q)\mu_0(s)}{\mu(s)} \tag{75}$$

$$A^2 = 1 - \frac{2(1-q)\mu_0(s)}{\mu(s)} + \frac{(1-q)^2\mu_0^2(s)}{\mu^2(s)} \tag{76}$$

Plug this back to the second term, we have

$$\mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha A^2\right] \tag{77}$$

$$= \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha \left(1 - \frac{2(1-q)\mu_0(s)}{\mu(s)} + \frac{(1-q)^2\mu_0^2(s)}{\mu^2(s)}\right)\right] \tag{78}$$

$$= \mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^\alpha\right] - 2(1-q)\mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^{\alpha-1}\right] + (1-q)^2\mathbb{E}_{s\sim\mu_0}\left[\left(\frac{\mu(s)}{\mu_0(s)}\right)^{\alpha-2}\right] \tag{79}$$

$$= \mathrm{R}_q^{(\alpha)} - 2(1-q)\mathrm{R}_q^{(\alpha-1)} + (1-q)^2\mathrm{R}_q^{(\alpha-2)} \tag{80}$$

Note that this bound is independent on T due to translation invariance, hence it is an upper bound for arbitrary T (that satisfy case 2). Thus, the overall bound becomes

$$\mathbb{E}_{s,t\sim\mu_0}\left[\left(\frac{\mu(s,t)}{\mu_0(s,t)}\right)^\alpha\right] \leq \mathrm{R}_q^{(\alpha)} + \frac{2\alpha(\alpha-1)}{\sigma_1^2}\left[\mathrm{R}_q^{(\alpha)} - 2(1-q)\mathrm{R}_q^{(\alpha-1)} + (1-q)^2\mathrm{R}_q^{(\alpha-2)}\right] \tag{81}$$

Denote $\widetilde{\mathrm{R}}_q^{(\alpha)} = \mathbb{E}_{s\sim\mu}\left[\left(\frac{\mu_0(s)}{\mu(s)}\right)^\alpha\right]$. Since we know that

$$\widetilde{\mathrm{R}}_q^{(\alpha)} = \mathbb{E}_\mu\left[\left(\frac{\mu_0}{\mu}\right)^\alpha\right] = \mathbb{E}_{\mu_0}\left[\left(\frac{\mu_0}{\mu}\right)^{\alpha-1}\right] = \mathbb{E}_{\mu_0}\left[\left(\frac{\mu}{\mu_0}\right)^{1-\alpha}\right] = \mathrm{R}_q^{(1-\alpha)} \tag{82}$$

Thus, by setting $\alpha \leftarrow 1-\alpha$, we have

$$\mathbb{E}_{s,t\sim\mu}\left[\left(\frac{\mu_0(s,t)}{\mu(s,t)}\right)^\alpha\right] \tag{83}$$

$$= \mathbb{E}_{s,t\sim\mu_0}\left[\left(\frac{\mu(s,t)}{\mu_0(s,t)}\right)^{1-\alpha}\right] \tag{84}$$

$$= \widetilde{\mathrm{R}}_q^{(1-\alpha)} + \frac{2\alpha(\alpha-1)}{\sigma_1^2}\left[\widetilde{\mathrm{R}}_q^{(1-\alpha)} - 2(1-q)\widetilde{\mathrm{R}}_q^{(-\alpha)} + (1-q)^2\widetilde{\mathrm{R}}_q^{(-\alpha-1)}\right] \tag{85}$$

$$= \widetilde{\mathrm{R}}_q^{(\alpha)} + \frac{2\alpha(\alpha-1)}{\sigma_1^2}\left[\widetilde{\mathrm{R}}_q^{(\alpha)} - 2(1-q)\widetilde{\mathrm{R}}_q^{(\alpha+1)} + (1-q)^2\widetilde{\mathrm{R}}_q^{(\alpha+2)}\right] \tag{86}$$

So overall, the RDP of subsampled PTR is

$$\frac{1}{\alpha-1}\ln\left(\max\left(1+2q^2\alpha(\alpha-1)\left(\frac{1-\delta_0}{\sigma_1^2}+\frac{\delta_0}{\sigma_2^2}\right),\right.\right. \tag{87}$$

$$\mathrm{R}_q^{(\alpha)}+\frac{2\alpha(\alpha-1)}{\sigma_1^2}\left[\mathrm{R}_q^{(\alpha)}-2(1-q)\mathrm{R}_q^{(\alpha-1)}+(1-q)^2\mathrm{R}_q^{(\alpha-2)}\right], \tag{88}$$

$$\left.\left.\widetilde{\mathrm{R}}_q^{(\alpha)}+\frac{2\alpha(\alpha-1)}{\sigma_1^2}\left[\widetilde{\mathrm{R}}_q^{(\alpha)}-2(1-q)\widetilde{\mathrm{R}}_q^{(\alpha+1)}+(1-q)^2\widetilde{\mathrm{R}}_q^{(\alpha+2)}\right]\right)\right) \tag{89}$$

□

A.3 Non-negativity of the Pearson-Vajda χ^α pseudo-divergence for independent joint distributions

The major difference between PTR and simple DP mechanisms such as Gaussian is that PTR is a flexible design which contains two data-dependent outputs. If a mechanism $\mathcal{M}$ comprises of two sub-mechanisms and output $\mathcal{M}(S)=(\mathcal{M}_1(S),\mathcal{M}_2(S))$, is there anything we can say about the privacy amplification bound of subsampled $\mathcal{M}$? A byproduct of our research on subsampled PTR is the privacy amplification bound for a $\mathcal{M}$ that outputs $\mathcal{M}_1(S),\mathcal{M}_2(S)$ independently. This result is important since such a $\mathcal{M}$ is instantiated in many existing techniques in improving differentially private deep learning. For example, the weight clustering methods in [ZJW18] is essentially composing subsampled Gaussian mechanism composition (one Gaussian mechanism for each gradient group); the gradient projection methods in [YZCL21] is also composing subsampled Gaussian mechanism composition (one Gaussian mechanism for subspace embedding and one Gaussian mechanism for residual gradients).

A lower bound for the RDP of subsampled mechanism is derived in [ZW19] (Theorem 6). Meanwhile, [ZW19] show that for many randomized algorithms (including the popular Gaussian mechanism and Laplace mechanism) that satisfy a special condition, the lower bound is also an upper bound (and thus a tight bound) for the subsampled mechanism for all α. The condition is the non-negativity of Pearson-Vajda χ^α pseudo divergence.

Theorem A.2 (Condition for Exact RDP Bound for Subsampled Mechanisms [ZW19])**.** *Let $\mathcal{M}$ be a randomized algorithm with up to αth order RDP $\varepsilon(\alpha)<\infty$. If for all adjacent data sets $S\sim S'$, and all odd $3\le\ell\le\alpha$,*

$$D_{\chi^\alpha}\left(\mathcal{M}(S)\|\mathcal{M}\left(S'\right)\right):=\mathbb{E}_{\mathcal{M}(S')}\left(\frac{\mathcal{M}(S)}{\mathcal{M}\left(S'\right)}-1\right)^\ell\ge 0 \tag{90}$$

then the lower bound in Theorem 6 of [ZW19] is also an upper bound.

$D_{\chi^\alpha}\left(\mathcal{M}(S)\|\mathcal{M}\left(S'\right)\right)$ is so-called as Pearson-Vajda χ^α pseudo divergence. When α is even, D_{χ^α} is trivially non-negative. When $\alpha=2$, D_{χ^2} is the well-known χ^2 divergence. When $\alpha=1$, D_{χ^1} is always 0. In [ZW19], they have shown that both Laplace and Gaussian mechanisms satisfy (90) for all adjacent datasets. Here, we show that if two mechanisms satisfy Condition (90), then so does the mechanisms that independently output the two mechanisms' outputs.

Theorem A.3. *If both of the two mechanisms $\mathcal{M}_1,\mathcal{M}_2$ satisfy (90), then the mechanism $\mathcal{M}(S)=(\mathcal{M}_1(S),\mathcal{M}_2(S))$ also satisfy (90), given that the randomness of $\mathcal{M}_1$ and $\mathcal{M}_2$ are independent in $\mathcal{M}$.*

This result means that we can further tighten the subsampled RDP bound for many existing techniques for improving differentially private deep learning, such as the weight clustering [ZJW18] and gradient projection [YZCL21]. We also note that, for PTR, since the Gaussian mechanism used in the **Release** step will depend on $\widehat{\Delta}$, it does not satisfy the independence assumption and thus Theorem A.3 does not apply to PTR.

Proof. We show the following claim: for any density function $\mu_1, \pi_1, \mu_2, \pi_2$, if for all integer $\alpha \geq 1$ we have

$$\mathbb{E}_{x\sim\mu_1}\left[\left(\frac{\pi_1(x)}{\mu_1(x)} - 1\right)^{\alpha}\right] \geq 0 \tag{91}$$

$$\mathbb{E}_{y\sim\mu_2}\left[\left(\frac{\pi_2(y)}{\mu_2(y)} - 1\right)^{\alpha}\right] \geq 0 \tag{92}$$

then we have

$$\mathbb{E}_{x\sim\mu_1, y\sim\mu_2}\left[\left(\frac{\pi_1(x)\pi_2(y)}{\mu_1(x)\mu_2(y)} - 1\right)^{\alpha}\right] \geq 0 \tag{93}$$

When α is even, the non-negativity is trivial. Therefore, we assume α is odd in the following text.

$$\mathbb{E}_{x\sim\mu_1, y\sim\mu_2}\left[\left(\frac{\pi_1(x)\pi_2(y)}{\mu_1(x)\mu_2(y)} - 1\right)^{\alpha}\right] \tag{94}$$

$$= \mathbb{E}_{x\sim\mu_1, y\sim\mu_2}\left[\left(\left(\frac{\pi_1(x)}{\mu_1(x)} - 1\right)\cdot\frac{\pi_2(y)}{\mu_2(y)} + \frac{\pi_2(y)}{\mu_2(y)} - 1\right)^{\alpha}\right] \tag{95}$$

$$= \mathbb{E}_{x\sim\mu_1, y\sim\mu_2}\left[\sum_{i=0}^{\alpha}\binom{\alpha}{i}\left(\left(\frac{\pi_1(x)}{\mu_1(x)} - 1\right)\cdot\frac{\pi_2(y)}{\mu_2(y)}\right)^{i}\left(\frac{\pi_2(y)}{\mu_2(y)} - 1\right)^{\alpha-i}\right] \tag{96}$$

$$= \sum_{i=0}^{\alpha}\binom{\alpha}{i}\mathbb{E}_{x\sim\mu_1}\left[\left(\frac{\pi_1(x)}{\mu_1(x)} - 1\right)^{i}\right] \tag{97}$$

$$\cdot\, \mathbb{E}_{y\sim\mu_2}\left[\left(\frac{\pi_2(y)}{\mu_2(y)}\right)^{i}\left(\frac{\pi_2(y)}{\mu_2(y)} - 1\right)^{\alpha-i}\right] \tag{98}$$

$$\tag{99}$$

where the last equality is due to the linearity of expectation and independence of x and y.

We already know that $\mathbb{E}_{x\sim\mu_1}\left[\left(\frac{\pi_1(x)}{\mu_1(x)} - 1\right)^{i}\right] \geq 0$ for arbitrary $i \geq 0$. We use the following lemma to show the second term is also non-negative for any i. The only interesting case is when i is even (if i is odd, then $\alpha - i$ is even so the second term is clearly non-negative).

Lemma A.4. *For any even a and odd b, and any non-negative random variable X, we have*

$$\mathbb{E}\left[X^a(X-1)^b\right] \geq \mathbb{E}\left[(X-1)^b\right] \tag{100}$$

Proof.

$$\mathbb{E}\left[X^a(X-1)^b\right] \tag{101}$$

$$= \mathbb{E}\left[X^a(X-1)^b I[X \geq 1] + X^a(X-1)^b I[X < 1]\right] \tag{102}$$

$$= \mathbb{E}\left[X^a(X-1)^b I[X \geq 1]\right] + \mathbb{E}\left[X^a(X-1)^b I[X < 1]\right] \tag{103}$$

$$\geq \mathbb{E}\left[(X-1)^b I[X \geq 1]\right] + \mathbb{E}\left[(X-1)^b I[X < 1]\right] \tag{104}$$

$$= \mathbb{E}\left[(X-1)^b I[X \geq 1] + (X-1)^b I[X < 1]\right] \tag{105}$$

$$= \mathbb{E}\left[(X-1)^b\right] \tag{106}$$

□

By the above lemma, we know that when i is even,

$$\mathbb{E}_{y\sim\mu_2}\left[\left(\frac{\pi_2(y)}{\mu_2(y)}\right)^{i}\left(\frac{\pi_2(y)}{\mu_2(y)} - 1\right)^{\alpha-i}\right] \tag{107}$$

$$\geq \mathbb{E}_{y\sim\mu_2}\left[\left(\frac{\pi_2(y)}{\mu_2(y)} - 1\right)^{\alpha-i}\right] \tag{108}$$

$$\geq 0 \tag{109}$$

This leads to the conclusion. □

Does PTR algorithm also satisfy Condition (90)? According to our numerical verification, the answer is unfortunately no. However, from Figure 2, we can see that the RDP for subsampled RDP derived from Theorem 4.4 is only slightly worse than the lower bound (green line), which means that Theorem 4.4 is near optimal.

A.4 Bound of the Local sensitivity after r adding/removal for Trimmed Sum

Recall that we denote a dataset $S = \{x_1, \ldots, x_m\}$, and $x_{(k)}$ denote the kth smallest data point among S in ℓ_2 norm, i.e., $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(m)}$. $\mathrm{TSUM}_F(S) = \sum_{i=1}^{m-F} x_{(i)}$ if $m > F$, or 0 if $m \leq F$.

Theorem 5.1 (Restate). $\mathbb{LS}^{(r)}_{\mathrm{TSUM}_F}(S) = \left\|x_{(m-F+1+r)}\right\|$ if $r \leq F-1$, or $\mathbb{GS}_{\mathrm{TSUM}_F}$ if $r > F-1$.

Proof. The $\mathbb{GS}_{\mathrm{TSUM}_F}$ for $r > F-1$ is trivial as the local sensitivity can never be larger than global sensitivity. When $r \leq F-1$, it is easy to see that the local sensitivity of TSUM_F is just $x_{(m-F+1)}$, as we can add the element with the maximum possible norm x_∞ in the data space to S, so that $\mathrm{TSUM}_F(S \cup \{x_\infty\}) = \sum_{i=1}^{m+1-F} x_{(i)}$, and $\|\mathrm{TSUM}_F(S \cup \{x_\infty\}) - \mathrm{TSUM}_F(S)\| = \left\|x_{(i)}\right\|$. If the added element has norm smaller than $\left\|x_{(m+1-F)}\right\|$, we will always have $\|\mathrm{TSUM}_F(S \cup \{x_\infty\}) - \mathrm{TSUM}_F(S)\| < \left\|x_{(i)}\right\|$. We can easily see that single element removal will also not change $\mathrm{TSUM}_F(S)$ that much. Thus $\mathbb{LS}^{(0)}_{\mathrm{TSUM}_F}(S) = x_{(m-F+1)}$.

To maximize the local sensitivity of S with r elements addition/removal, it's trivial to see that the best strategy is simply adding element with the maximum possible norm x_∞ in the data space to S, and the local sensitivity for the changed dataset $\tilde{S} = S \cup (\{x_\infty\} \times r)$ has local sensitivity $\left\|x_{(m+r)-F+1}\right\| = \|x_{m-F+1+r}\|$ as long as $r \leq F-1$. □

A.5 Convergence Guarantee of PTR-based Gradient Aggregation under Byzantine Failure

Settings of Robust Training. We denote the target loss function as $\mathcal{L}(w) = \mathbb{E}_{z\sim\mathcal{D}}\left[\ell(w, z)\right]$, where $w \in \mathbb{R}^d$ is the model parameters and z is a data point randomly drawn from some distribution $\mathcal{D}$. We assume ℓ is R-Lipschitz, β-smooth and α-strongly convex. We have n stochastic gradient oracles $g_1, \ldots, g_n$, where at each iteration t, for every non-corrupted gradient oracles i, it is an unbiased estimator $g_i^{(t)}$ for the gradient of the global expected loss function with respect to the current model parameters w_t, i.e., $\mathbb{E}[g_i^{(t)}] = \nabla\mathcal{L}(w^{(t)})$. We additionally assume that non-corrupted stochastic gradients have bounded variance, i.e., for some $\sigma > 0$ we have

$$\mathbb{E}_{g_i^{(t)}}\left[\left\|g_i^{(t)} - \mathbb{E}[g_i^{(t)}]\right\|^2\right] \leq \sigma^2 \tag{110}$$

at every step t.

Remark A.5. We do not consider the effect of subsampling here for clean presentation. The effect from subsampling could be easily handled by deriving a high probability upper bound for the number of corrupted gradient oracles that will be sampled.

SGD with PTR works as

$$\widehat{\Delta}, \tilde{g}^{(t)} \leftarrow \mathrm{PTR}(\{g_1, \ldots, g_n\}) \tag{111}$$

$$w^{(t+1)} \leftarrow w^{(t)} - \left(\eta_A I[\widehat{\Delta} < \log(1/(2\delta_0))] + \eta_B I[\widehat{\Delta} \geq \log(1/(2\delta_0))]\right)\tilde{g}^{(t)} \tag{112}$$

Further, we call it Routine A if $\widehat{\Delta}$ is small and $\mathrm{PTR}(\{g_1, \ldots, g_n\}) = \sum_{i=1}^{n} g_{(i)}^{(t)} + \mathcal{N}(0, \sigma_1 \mathbf{1}_d)$, and call it Routine B if $\widehat{\Delta}$ is large and $\mathrm{PTR}(\{g_1, \ldots, g_n\}) = \sum_{i=1}^{n-F} g_{(i)}^{(t)} + \mathcal{N}(0, \sigma_2 \mathbf{1}_d)$. It makes sense to use a smaller learning rate η_A when the PTR goes to Routine A, and use a larger learning rate η_B for Routine B, since the two routines use different amount of gradient information.

Theorem 5.2 (formal version). Let $w^* \in \mathrm{argmin}_w \mathcal{L}(w)$. If there are at most F gradients being corrupted at each iteration, and if we set $\eta_A = \frac{n-F}{n}\eta_B$ and $\sigma_2^2 \geq \frac{(n-F)(n+1)}{n^2}\left(\frac{(n-F)\sigma^2 + FR}{d} + \sigma_1^2\right)$, then as $t \to \infty$, we have

$$\mathbb{E}_{g,\xi}\left[\left\|w^{(t)} - w^*\right\|^2\right] \leq \frac{M_B}{1-\rho_B} \tag{113}$$

for

$$0 < \eta_B \leq \frac{2\alpha(n-2F)}{n^2 + (n-F+1)(n-F)\beta^2} \tag{114}$$

where

$$\rho_B = 1 - 2\eta_B\alpha(n-2F) + \eta_B^2(n^2 + (n-F+1)(n-F)\beta^2) \tag{115}$$

$$M_B = \eta_B^2(n-F+1)\left((n-F)\sigma^2 + d\sigma_2^2\right) + \left(\frac{F}{n}\right)^2 R^2 \tag{116}$$

Proof. Let $\eta = \eta_A I[\widehat{\Delta} < \log(1/(2\delta_0))] + \eta_B I[\widehat{\Delta} \geq \log(1/(2\delta_0))]$. By the update rule of parameters in SGD, we have

$$\left\|w^{(t+1)} - w^*\right\|^2 = \left\|w^{(t)} - w^*\right\|^2 - 2\eta\left\langle w^{(t)} - w^*, \tilde{g}^{(t)}\right\rangle + \eta^2\left\|\tilde{g}^{(t)}\right\|^2 \tag{117}$$

where

$$\tilde{g}^{(t)} = \mathrm{PTR}(\{g_1, \ldots, g_n\}) \tag{118}$$

If PTR runs Routine B, then

$$\tilde{g}^{(t)} = \sum_{i=1}^{n-F} g_{(i)}^{(t)} + \xi, \xi \sim \mathcal{N}(0, \sigma_2\mathbf{I}_d) \tag{119}$$

If PTR runs Routine A, then

$$\tilde{g}^{(t)} = \sum_{i=1}^{n} g_{(i)}^{(t)} + \xi, \xi \sim \mathcal{N}(0, \sigma_1\mathbf{I}_d) \tag{120}$$

In the following proof, we consider the two routines separately. We denote a set of $n-F$ non-corrupted gradients at step t as $\mathcal{H}^{(t)}$. We use $g_{(i)}^{(t)}$ the ith smallest gradient among the set of gradients at step t, i.e., $g_{(1)}^{(t)} \leq \ldots \leq g_{(n)}^{(t)}$.

Case of Running Routine B. In this case, $\mathbb{E}[\|\xi\|^2] = d\sigma_2^2$. We first upper bound $\left\|\tilde{g}^{(t)}\right\|$:

$$\left\|\tilde{g}^{(t)}\right\| = \left\|\sum_{i=1}^{n-F} g_{(i)}^{(t)} + \xi\right\| \tag{121}$$

$$\leq \sum_{i=1}^{n-F}\left\|g_{(i)}^{(t)}\right\| + \|\xi\| \tag{122}$$

$$\leq \sum_{i\in\mathcal{H}^{(t)}}\left\|g_i^{(t)}\right\| + \|\xi\| \tag{123}$$

Therefore

$$\left\|\tilde{g}^{(t)}\right\|^2 \leq (n-F+1)\left(\sum_{i\in\mathcal{H}^{(t)}}\left\|g_i^{(t)}\right\|^2 + \|\xi\|^2\right) \tag{124}$$

due to AM-QM inequality.

Now we lower bound $\left\langle w^t - w^*, \tilde{g}^{(t)}\right\rangle$. We denote the non-corrupted gradients that are not being trimmed in step t as $\tilde{\mathcal{H}}^{(t)} = \mathcal{H}^{(t)} \cap \{g_{(1)}^{(t)}, \ldots, g_{(n-F)}^{(t)}\}$, and we denote the corrupted gradients that are not being trimmed in step t as $\mathcal{B}^{(t)} = \{g_{(1)}^{(t)}, \ldots, g_{(n-F)}^{(t)}\} \setminus \tilde{\mathcal{H}}^{(t)}$. Since there are at most F corrupted gradients, we have $|\tilde{\mathcal{H}}^{(t)}| \geq n-2F$ and $|\mathcal{B}| \leq F$. Since

$$\tilde{g}^{(t)} = \sum_{i\in\tilde{\mathcal{H}}^{(t)}} g_i^{(t)} + \sum_{i\in\mathcal{B}^t} g_i^{(t)} + \xi \tag{125}$$

we have

$$\left\langle w^{(t)} - w^*, \tilde{g}^{(t)} \right\rangle = \sum_{i \in \tilde{\mathcal{H}}^{(t)}} \left\langle w^{(t)} - w^*, g_i^{(t)} \right\rangle + \sum_{i \in \mathcal{B}^t} \left\langle w^{(t)} - w^*, g_i^{(t)} \right\rangle + \left\langle w^{(t)} - w^*, \xi \right\rangle \tag{126}$$

$$\geq \sum_{i \in \tilde{\mathcal{H}}^{(t)}} \left\langle w^{(t)} - w^*, g_i^{(t)} \right\rangle - FR \left\| w^{(t)} - w^* \right\| + \left\langle w^{(t)} - w^*, \xi \right\rangle \tag{127}$$

Denote this lower bound as

$$\phi_t = \sum_{i \in \tilde{\mathcal{H}}^{(t)}} \left\langle w^{(t)} - w^*, g_i^{(t)} \right\rangle - FR \left\| w^{(t)} - w^* \right\| + \left\langle w^{(t)} - w^*, \xi \right\rangle \tag{128}$$

Then

$$\left\| w^{(t+1)} - w^* \right\|^2 \leq \left\| w^{(t)} - w^* \right\|^2 - 2\eta\phi_t + \eta^2(n - F + 1) \left(\sum_{i \in \mathcal{H}^{(t)}} \left\| g_i^{(t)} \right\|^2 + \|\xi\|^2 \right) \tag{129}$$

Now this upper bound only include quantities that are independent from corrupted gradients. Take expectation of both sides over $g^{(t)}$ and ξ, we have

$$\mathbb{E}_{g,\xi} \left[\left\| w^{(t+1)} - w^* \right\|^2 \right] \tag{130}$$

$$\leq \left\| w^{(t)} - w^* \right\|^2 - 2\eta \mathbb{E}_{g,\xi}[\phi_t] + \eta^2(n - F + 1) \mathbb{E}_{g,\xi} \left[\sum_{i \in \mathcal{H}^{(t)}} \left\| g_i^{(t)} \right\|^2 + \|\xi\|^2 \right] \tag{131}$$

For $\mathbb{E}_{g,\xi}[\phi_t]$, we can obtain its lower bound

$$\mathbb{E}_{g,\xi}[\phi_t] = \sum_{i \in \tilde{\mathcal{H}}^{(t)}} \left\langle w^{(t)} - w^*, \nabla \mathcal{L}(w^{(t)}) \right\rangle - FR \left\| w^{(t)} - w^* \right\| \tag{132}$$

$$\geq \alpha(n - 2F) \left\| w^{(t)} - w^* \right\|^2 - FR \left\| w^{(t)} - w^* \right\| \tag{133}$$

since $\mathcal{L}$ is α-strongly convex.

For $\mathbb{E}_{g,\xi} \left[\sum_{i \in \mathcal{H}^{(t)}} \left\| g_i^{(t)} \right\|^2 + \|\xi\|^2 \right]$, since $\mathbb{E}_g \left[\left\| g_i^{(t)} \right\|^2 \right] \leq \sigma^2 + \|\nabla \mathcal{L}(w^t)\|^2$ and $\mathbb{E}[\|\xi\|^2] = d\sigma_2^2$, we have

$$\mathbb{E}_{g,\xi} \left[\sum_{i \in \mathcal{H}^{(t)}} \left\| g_i^{(t)} \right\|^2 + \|\xi\|^2 \right] \tag{134}$$

$$\leq (n - F) \left(\sigma^2 + \left\| \nabla \mathcal{L}(w^{(t)}) \right\|^2 \right) + d\sigma_2^2 \tag{135}$$

$$\leq (n - F) \left(\sigma^2 + \beta^2 \left\| w^{(t)} - w^* \right\|^2 \right) + d\sigma_2^2 \tag{136}$$

since $\left\| \nabla \mathcal{L}(w^{(t)}) \right\| = \left\| \nabla \mathcal{L}(w^{(t)}) - \nabla \mathcal{L}(w^*) \right\| \leq \beta \left\| w^{(t)} - w^* \right\|$ by β-smoothness of $\mathcal{L}$.

Plugging in the lower and upper bounds, we have

$$\mathbb{E}_{g,\xi} \left[\left\| w^{(t+1)} - w^* \right\|^2 \right] \tag{137}$$

$$\leq \left(1 - 2\eta\alpha(n - 2F) + \eta^2\beta^2(n - F + 1)(n - F) \right) \left\| w^{(t)} - w^* \right\|^2 \tag{138}$$

$$+ 2\eta FR \left\| w^{(t)} - w^* \right\| \tag{139}$$

$$+ \eta^2(n - F + 1) \left((n - F)\sigma^2 + d\sigma_2^2 \right) \tag{140}$$

$$\leq \left(1 - 2\eta\alpha(n - 2F) + \eta^2(n^2 + (n - F + 1)(n - F)\beta^2) \right) \left\| w^{(t)} - w^* \right\|^2 \tag{141}$$

$$+ \eta^2(n - F + 1) \left((n - F)\sigma^2 + d\sigma_2^2 \right) + \left(\frac{F}{n} \right)^2 R^2 \tag{142}$$

where the last step is due to

$$2\eta FR\left\|w^{(t)}-w^*\right\| \leq \left(\frac{F}{n}\right)^2 R^2 + n^2\eta^2\left\|w^{(t)}-w^*\right\|^2 \tag{143}$$

Let

$$\rho_B = 1 - 2\eta\alpha(n-2F) + \eta^2(n^2 + (n-F+1)(n-F)\beta^2) \tag{144}$$

$$M_B = \eta^2(n-F+1)\left((n-F)\sigma^2 + d\sigma_2^2\right) + \left(\frac{F}{n}\right)^2 R^2 \tag{145}$$

Then we have

$$\mathbb{E}_{g,\xi}\left[\left\|w^{(t+1)}-w^*\right\|^2\right] \leq \rho_B\left\|w^{(t)}-w^*\right\|^2 + M_B \tag{146}$$

Therefore, as long as $\rho_B < 1$, $w^{(t)}$ will eventually

For $\rho_B < 1$, we need

$$0 < \eta \leq \frac{2\alpha(n-2F)}{n^2 + (n-F+1)(n-F)\beta^2} \tag{147}$$

Case of Running Routine A. We follow a similar analysis as for the case of Routine A. In this case, $\mathbb{E}[\|\xi\|^2] = d\sigma_1^2$.

We first upper bound $\left\|\tilde{g}^{(t)}\right\|$:

$$\left\|\tilde{g}^{(t)}\right\| = \left\|\sum_{i=1}^n g_{(i)}^{(t)} + \xi\right\| \tag{148}$$

$$\leq \sum_{i=1}^n \left\|g_{(i)}^{(t)}\right\| + \|\xi\| \tag{149}$$

Therefore

$$\left\|\tilde{g}^{(t)}\right\|^2 \leq (n+1)\left(\sum_{i\in\mathcal{H}^{(t)}}\left\|g_i^{(t)}\right\|^2 + \|\xi\|^2\right) \tag{150}$$

In this case, there are no gradients being corrupted, and thus we have $|\tilde{\mathcal{H}}^{(t)}| \geq n-F$. Therefore, for $\mathbb{E}_{g,\xi}[\phi_t]$ we have

$$\mathbb{E}_{g,\xi}[\phi_t] = \sum_{i\in\tilde{\mathcal{H}}^{(t)}}\left\langle w^{(t)}-w^*, \nabla\mathcal{L}(w^{(t)})\right\rangle - FR\left\|w^{(t)}-w^*\right\| \tag{151}$$

$$\geq \alpha(n-F)\left\|w^{(t)}-w^*\right\|^2 - FR\left\|w^{(t)}-w^*\right\| \tag{152}$$

For $\mathbb{E}_{g,\xi}\left[\sum_{i\in\mathcal{H}^{(t)}}\left\|g_{(i)}^{(t)}\right\|^2 + \sum_{i\in[n]\setminus\mathcal{H}^{(t)}}\left\|g_{(i)}^{(t)}\right\|^2 + \|\xi\|^2\right]$, since $\mathbb{E}_g\left[\left\|g_{(i)}^{(t)}\right\|^2\right] \leq \sigma^2 + \left\|\nabla\mathcal{L}(w^{(t)})\right\|^2$ and $\mathbb{E}[\|\xi\|^2] = d\sigma_1^2$, we have

$$\mathbb{E}_{g,\xi}\left[\sum_{i\in\mathcal{H}^{(t)}}\left\|g_{(i)}^{(t)}\right\|^2 + \sum_{i\in[n]\setminus\mathcal{H}^{(t)}}\left\|g_{(i)}^{(t)}\right\|^2 + \|\xi\|^2\right] \tag{153}$$

$$\leq (n-F)\left(\sigma^2 + \left\|\nabla\mathcal{L}(w^{(t)})\right\|^2\right) + FR + d\sigma_1^2 \tag{154}$$

$$\leq (n-F)\left(\sigma^2 + \beta^2\left\|w^{(t)}-w^*\right\|^2\right) + FR + d\sigma_1^2 \tag{155}$$

Plugging in the lower and upper bounds, we have

$$\mathbb{E}_{g,\xi}\left[\left\|w^{(t+1)}-w^*\right\|^2\right] \tag{156}$$

$$\leq \left(1-2\eta\alpha(n-F)+\eta^2(n+1)(n-F)\beta^2\right)\left\|w^{(t)}-w^*\right\|^2 \tag{157}$$

$$+2\eta FR\cdot\left\|w^{(t)}-w^*\right\| \tag{158}$$

$$+\eta^2(n+1)\left((n-F)\sigma^2+FR+d\sigma_1^2\right) \tag{159}$$

$$\leq \left(1-2\eta\alpha(n-F)+\eta^2(n^2+(n+1)(n-F)\beta^2)\right)\left\|w^{(t)}-w^*\right\|^2 \tag{160}$$

$$+\eta^2(n+1)\left((n-F)\sigma^2+FR+d\sigma_1^2\right)+\left(\frac{F}{n}\right)^2R^2 \tag{161}$$

Follow similar analysis, but we will have

$$\rho_A = 1-2\eta\alpha(n-F)+\eta^2(n^2+(n+1)(n-F)\beta^2) \tag{162}$$

$$M_A = \eta^2(n+1)\left((n-F)\sigma^2+FR+d\sigma_1^2\right)+\left(\frac{F}{n}\right)^2R^2 \tag{163}$$

So for every time we run Routine A (with full gradient sum), we have

$$\mathbb{E}_{g,\xi}\left[\left\|w^{(t)}-w^*\right\|^2\right] \leq \rho_A\left\|w^{(t-1)}-w^*\right\|^2+M_A \tag{164}$$

And we have $M_B < M_A$, so more routine B can improve the utility.

Overall, we if there are at most F gradients being corrupted at each iteration, we have

$$\mathbb{E}\left[\left\|w^{(t)}-w^*\right\|^2\right] \leq \rho_A\left\|w^{(t-1)}-w^*\right\|^2+M_A \tag{165}$$

for

$$\rho_A = 1-2\eta_A\alpha(n-F)+\eta_A^2(n^2+(n+1)(n-F)\beta^2) \tag{166}$$

$$M_A = \eta_A^2(n+1)\left((n-F)\sigma^2+FR+d\sigma_1^2\right)+\left(\frac{F}{n}\right)^2R^2 \tag{167}$$

if PTR runs Routine A, and

$$\mathbb{E}_{g,\xi}\left[\left\|w^{(t+1)}-w^*\right\|^2\right] \leq \rho_B\left\|w^{(t)}-w^*\right\|^2+M_B \tag{168}$$

for

$$\rho_B = 1-2\eta_B\alpha(n-2F)+\eta_B^2(n^2+(n-F+1)(n-F)\beta^2) \tag{169}$$

$$M_B = \eta_B^2(n-F+1)\left((n-F)\sigma^2+d\sigma_2^2\right)+\left(\frac{F}{n}\right)^2R^2 \tag{170}$$

if PTR runs Routine B.

If we set $\eta_A = \frac{n-F}{n}\eta_B$, since

$$\sigma_2^2 \geq \frac{(n-F)(n+1)}{n^2}\left(\frac{(n-F)\sigma^2+FR}{d}+\sigma_1^2\right) \tag{171}$$

we have $M_A \leq M_B$, and we can also easily verify that $\rho_A \leq \rho_B$. Thus, as $t\to\infty$, we have

$$\mathbb{E}_{g,\xi}\left[\left\|w^{(t)}-w^*\right\|^2\right] \leq \frac{M_B}{1-\rho_B} \tag{172}$$

for

$$0 < \eta_B \leq \frac{2\alpha(n-2F)}{n^2+(n-F+1)(n-F)\beta^2} \tag{173}$$

□

A.6 Pseudo-code of TSGD+PTR

The pseudo-code of TSGD+PTR is shown in Algorithm 2, which uses Algorithm 3 (`PTR-TMEAN`) as a subroutine.

Algorithm 2: Private Trimmed-mean SGD with Propose-Test-Release

input : Dataset $\{\mathbf{z}^1, \ldots, \mathbf{z}^N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, \mathbf{z}^i)$, learning rate η, batch size B, sensitivity bound τ, Clipping threshold R, noise multiplier σ.

Initialize θ_0 randomly.
for $t \in [T]$ **do**
  Random Subsampling.
  Take a random batch $\mathcal{B}_t$ with sampling probability q in Poisson subsampling.
  Obtain Gradients.
  For each $i \in \mathcal{B}_t$, get (potentially faulty) $g_i^{(t)}$.
  Gradient Clipping.
  $g_i^{(t)} \leftarrow C \cdot g_i^{(t)}$ for $C = \min\left(1, R/\left\|g_i^{(t)}\right\|_2\right)$.
  Noisy Gradient Aggregation with PTR.
  $\tilde{g}^{(t)} \leftarrow$ `PTR-TMEAN` $\left(\{g_i^{(t)}\}\right)$.
  Descent.
  $\theta_{t+1}, \omega \leftarrow \theta_t - \eta\tilde{g}^{(t)}$.
  Adjust F**.**
  if ω *is* '+' **then**
    Increase F.
  else
    Decrease F.

Algorithm 3: `PTR-TMEAN`

input : S – Set of (clipped) gradient vectors at step t: $\{g_i^{(t)}\} \subseteq \mathbb{R}^d$,

$\Delta \leftarrow \min_{\tilde{S} \in \{\tilde{S}: \mathbb{LS}_{f_2}(\tilde{S}) > \tau\}} d\left(S, \tilde{S}\right)$.
$\widehat{\Delta} \leftarrow \Delta + \mathrm{Lap}(0, b)$.
if $\widehat{\Delta} \leq \log(1/(2\delta_0))b$ **then**
  return $\mathrm{SUM}(S) + \sigma R \cdot \mathcal{N}(0, \mathbf{1}_d)$, '+'
else
  return $\mathrm{TSUM}_F(S) + \sigma\tau \cdot \mathcal{N}(0, \mathbf{1}_d)$, '−'

A.6.1 Why not directly apply PTR to regular SGD?

As we discussed in the main text, PTR typically works with robust statistics such as trimmed mean. Regular SGD use mean as gradient aggregation function. Mean, however, does not have a low local sensitivity on most of the inputs. Therefore, we focus on the application of PTR in privatizing robust statistics.

B Experiment Settings & Additional Results

B.1 Experiment Settings for Table 1

B.1.1 Corruption Simulation.

Following the literature in Byzantine robustness [YCKB18, XKG19, AHJ+21, GLV21], we consider three possible sources of Byzantine failures: corruption in features, labels and communicated gradients. All experiments are repeated for 0% (i.e., clean), 10%, and 20% corruption ratio (CR).

Feature Corruption. Corruption in Features can arise from the process of data collection. Following [AHJ+21], we adopt the additive corruption introduced in [HD18]. Specifically, we add Gaussian noise from $\mathcal{N}(0, 100)$ directly to the corrupted images.

Gradient Corruption. Gradient can be corrupted in distributed SGD, e.g., due to hardware malfunction or malicious users. We consider the gradient corruption following [XKG19, AHJ+21], where we add Gaussian noise from $\mathcal{N}(0, 100)$ to the true gradients.

Label Corruption. Noisy labels are pervasive in the dataset. We randomly flip of label of certain amount of data points.

B.1.2 Datasets & Models.

MNIST [LeC98] is one of the most commonly used benchmark datasets in deep learning containing 70000 handwritten digit images. CIFAR-10 [Kri09] is another classic benchmark for image classification. It consists of 60000 images from 10 different classes with 6000 images each. EMNIST [CATVS17] is similar to MNIST but has a much larger size (145,600 character images and 26 balanced classes).

In Table 1, all models are trained entirely from scratch. For all datasets, we use a small CNN whose architecture is inherited from the official tutorial of tensorflow/privacy[9].

B.1.3 Hyperparameters.

For `TSGD+PTR`, we set $\delta_0 = 10^{-8}, b = 1$. For MNIST and EMNIST, we set gradient clipping bound $R = 1$, $\tau = 0.5$, noise multiplier $\sigma = 1.1$ for `TSGD+PTR`, and $\sigma = 0.7$ for `TSGD+Gaussian`. The noise multiplier for `TSGD+PTR` and `TSGD+Gaussian` are picked differently in order to align their privacy loss in each iteration. For CIFAR10, we set gradient clipping bound $R = 3$, $\tau = 2$, noise multiplier $\sigma = 1.1$ for `TSGD+PTR`, and $\sigma = 0.9$ for `TSGD+Gaussian`.

For MNIST and EMNIST dataset, we set the learning rate as 0.15, batch size as 256; for CIFAR10 dataset, we set the learning rate as 0.1, batch size as 1024.

We set F to be 25% of the batch size for `TSGD+Gaussian`. For `TSGD+PTR`, F is dynamically adjusted based on the value of $\widehat{\Delta}$. If sensitivity test is passed, we increase F by $0.02 \times \text{batchsize}$, and if sensitivity test is failed, we decrease F by the same amount.

All of our experiments are performed on Tesla P100-PCIE-16GB GPU.

B.2 Additional Results on More Architectures

We experiment with more architectures on CIFAR10 dataset. Specifically, we use two famous, moderately large architecture ResNet18 [HZRS16] and VGG11 [SZ14]. We follow the common procedure in prior works [ACG+16]: we use ResNet18 and VGG11 that are pretrained by ImageNet dataset. The pre-training weight is publicly available from PyTorch. We only finetune the last layer of the model.

We set gradient clipping bound $R = 5$, batch size as 2048, learning rate 0.01. For `TSGD+PTR`, we set $\delta_0 = 10^{-8}, b = 1$, and $\tau = 3$. We set noise multiplier $\sigma = 2.2$ for `TSGD+PTR`, and $\sigma = 1.8$ for `TSGD+Gaussian`. We set F to be 25% of the batch size for `TSGD+Gaussian`. F is dynamically adjusted in the same way as the experiments in the main text.

[9] https://github.com/tensorflow/privacy

Archi.	Corruption Type	CR	$\varepsilon = 3.0$ TSGD+Gaussian	$\varepsilon = 3.0$ TSGD+PTR	$\varepsilon = 5.0$ TSGD+Gaussian	$\varepsilon = 5.0$ TSGD+PTR
VGG11	Label	0	50.05%	52.09% (+2.04%)	51.75%	52.85% (+1.1%)
		0.1	44.03%	48.73% (+4.7%)	48.78%	50.24% (+1.46%)
		0.2	35.04%	43.63% (+8.59%)	43.17%	46.35% (+3.18%)
	Feature	0.1	45.59%	49.57% (+3.98%)	49.60%	50.74% (+1.14%)
		0.2	43.82%	47.95% (+4.13%)	48.08%	48.88% (+0.8%)
	Gradient	0.1	45.61%	50.15% (+4.54%)	50.10%	51.15% (+1.05%)
		0.2	45.40%	50.15% (+4.75%)	50.46%	50.82% (+0.36%)
			$\varepsilon = 3.0$ TSGD+Gaussian	$\varepsilon = 3.0$ TSGD+PTR	$\varepsilon = 5.0$ TSGD+Gaussian	$\varepsilon = 5.0$ TSGD+PTR
ResNet18	Label	0	38.97%	43.15% (+3.18%)	46.27%	47.15% (+0.88%)
		0.1	35.84%	42.55% (+6.71%)	42.05%	44.77% (+2.72%)
		0.2	26.15%	36.48% (+10.33%)	33.97%	39.71% (+5.74%)
	Feature	0.1	37.91%	43.15% (+5.24%)	42.54%	44.93% (+2.39%)
		0.2	36.30%	41.85% (+5.55%)	41.21%	43.84% (+2.63%)
	Gradient	0.1	38.42%	44.27% (+5.85%)	43.83%	46.4% (+2.57%)
		0.2	37.79%	44.34% (+6.55%)	43.22%	46.14% (+2.92%)

Table 2: Model Accuracy under different privacy budgets and corruption settings. Every statistic is averaged over 5 runs with different random seed. The improvement of $\mathrm{TSGD} + \mathrm{PTR}$ over $\mathrm{TSGD} + \mathrm{Gaussian}$ is highlighted in the red text.

Corruption Type	CR	$\varepsilon = 2.0$ TSGD+Gaussian	$\varepsilon = 2.0$ TSGD+PTR	$\varepsilon = 2.5$ TSGD+Gaussian	$\varepsilon = 2.5$ TSGD+PTR
Targeted Label Flip	0	72.94%	76.44% (+3.5%)	79.15%	81.02% (+1.87%)
	0.1	72.06%	75.11% (+3.05%)	76.25%	78.97% (+2.72%)
	0.2	70.72%	73.85% (+3.13%)	73.69%	76.67% (+2.98%)
Gradient Bit Flip	0.1	69.58%	75.67% (+6.09%)	75.39%	78.82% (+3.43%)
	0.2	65.13%	75.67% (+10.54%)	71.85%	78.85% (+7.0%)

Table 3: Model Accuracy on EMNIST dataset under different privacy budgets on two more severe types of failures.

The results are shown in Table 2. As we can see, `TSGD+PTR` consistently outperforms `TSGD+Gaussian` across different architectures.

B.3 Additional Results on More Corruption Types

Besides the three corruption types we considered in the maintext, we evaluate on two additional possible errors which are considered more severe kinds of failure.

1. **Gradient Bit-flipping failure** where the bits that control the sign of the floating numbers are flipped, e.g., due to some hardware failure. A faulty worker pushes the negative gradient instead of the true gradient to the servers.
2. **Targeted label flipping failure** where the labels are flipped in a "targeted" way, i.e., for any $\mathit{label} \in \{0, \ldots, 25\}$, is replaced by $25 - \mathit{label}$. Such failures/attacks can be caused by data poisoning or software failures.

We experiment on EMNIST dataset and the results are shown in Table 3. The experiment settings are exactly the same as the settings for Table 1. As we can see, `TSGD+PTR` once again outperform `TSGD+Gaussian` significantly.

B.4 Comparison between regular DPSGD and trimmed-mean based robust SGD

We additionally show the comparison between trimmed mean robust SGD with/without PTR and regular DPSGD in Table 4 on EMNIST dataset with the same experiment settings described before. As we can see, the robust SGD performs worse than non-robust counterpart on clean training data. This is because when the training data are clean, the outliers filtered out by robust SGD in the gradient batch are usually corresponding to the data points that are misclassified, which are important for

Corruption Type	CR	TSGD+Gaussian	$\varepsilon = 2.0$ DPSGD	TSGD+PTR	TSGD+Gaussian	$\varepsilon = 2.5$ DPSGD	TSGD+PTR
Label	**0**	72.94%	77.29%	76.44% (−0.85%)	79.15%	83.06%	81.02% (−2.04%)
	0.1	72.60%	74.80%	75.66% (+0.86%)	79.38%	79.30%	80.63% (+1.33%)
	0.2	70.03%	71.42%	72.62% (+1.2%)	77.48%	77.43%	79.19% (+1.76%)
Feature	**0.1**	69.60%	74.58%	74.01% (−0.57%)	77.80%	80.95%	81.04% (+0.09%)
	0.2	70.04%	73.79%	74.76% (+0.97%)	78.99%	79.43%	80.74% (+1.31%)
Gradient	**0.1**	71.75%	72.97%	76.19% (+3.22%)	77.32%	76.59%	77.73% (+1.14%)
	0.2	70.76%	71.22%	74.65% (+3.43%)	76.68%	75.69%	77.17% (+1.48%)

Table 4: Model accuracy comparison with regular DPSGD on EMNIST dataset. The improvement of TSGD + PTR over regular DPSGD is highlighted in the red text.

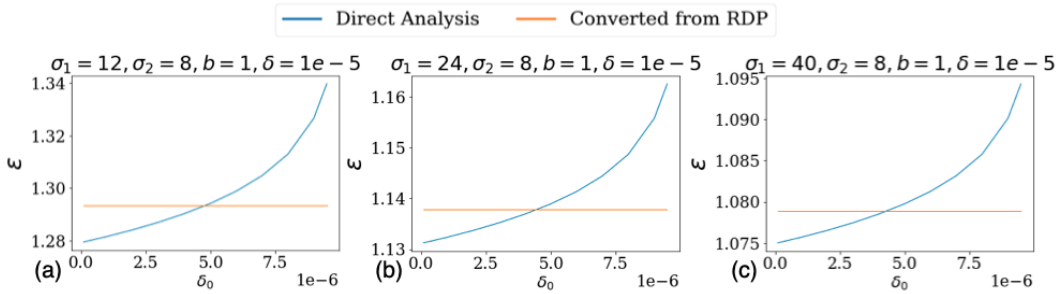

Figure 4: The ε parameter of the (ε, δ)-DP guarantee of PTR when $\delta = 10^{-5}$ for different noise scales. We convert the RDP bound in Theorem 4.3 to (ε, δ)-DP by the RDP-DP conversion formula from [BBG+20], and compare it with the ε obtained from the direct analysis in Theorem 4.2. For the bound converted from RDP, we search for the optimal $\alpha \in [1, 200]$. The bound is constant across different δ_0 since when δ_0 is small, the RDP for PTR will take the second term in (3).

improving model performance. However, TSGD+PTR achieves better performance on most of the corruption settings.

B.5 Additional Results on Privacy Analysis Comparison

In this section, we show more numerical results on privacy analysis comparison by varying $\tau = \sigma_2/\sigma_1$. In Figure 4, we numerically compute the privacy bound from direct analysis and the one converted from RDP, with $\tau \in [2/3, 1/3, 0.1]$. In Figure 5, we show the subsampled privacy bound composed with moment account, also with $\tau \in [2/3, 1/3, 0.1]$. As we can see, our Theorem 4.3 and Theorem 4.4 once again provide tighter bounds compared with the baseline.

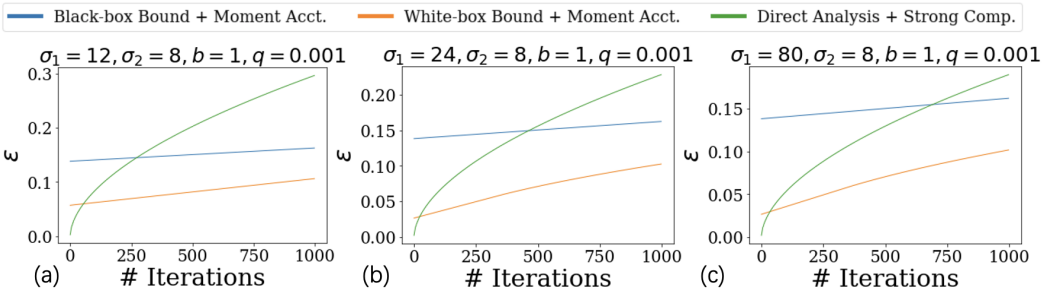

Figure 5: Illustration of the use of our Theorem 4.4 in moments accountant. We plot the the privacy loss ε for $\delta = 10^{-5}$ after different rounds of composition. We set $\delta_0 = 10^{-8}$ here to allow more iterations for Strong Composition of (ε, δ)-DP.