Differentially Private Deep Learning with ModelMix

***Abstract*—Training large neural networks with meaningful/usable differential privacy security guarantees is a demanding challenge. In this paper, we tackle this problem by revisiting the two key operations in Differentially Private Stochastic Gradient Descent (DP-SGD): 1) iterative perturbation and 2) gradient clipping. We propose a generic optimization framework, called *ModelMix*, which performs random aggregation of intermediate model states. It strengthens the composite privacy analysis utilizing the entropy of the training trajectory and improves the (ϵ, δ) DP security parameters by an order of magnitude.**

We provide rigorous analyses for both the utility guarantees and privacy amplification of ModelMix. In particular, we present a formal study on the effect of gradient clipping in DP-SGD, which provides theoretical instruction on how hyper-parameters should be selected. We also introduce a refined gradient clipping method, which can further sharpen the privacy loss in private learning when combined with ModelMix.

Thorough experiments with significant privacy/utility improvement are presented to support our theory. We train a Resnet-20 network on CIFAR10 with 70.4% accuracy via ModelMix given $(\epsilon = 8, \delta = 10^{-5})$ DP-budget, compared to the same performance but with $(\epsilon = 145.8, \delta = 10^{-5})$ using regular DP-SGD; assisted with additional public low-dimensional gradient embedding, one can further improve the accuracy to 79.1% with $(\epsilon = 6.1, \delta = 10^{-5})$ DP-budget, compared to the same performance but with $(\epsilon = 111.2, \delta = 10^{-5})$ without ModelMix.

***Index Terms*—Differential Privacy; Rényi Differential Privacy; Clipped Stochastic Gradient Descent; Deep Learning;**

1. Introduction

Privacy concerns when learning with sensitive data are receiving increasing attention. Many practical attacks have shown that without proper protection, the model's parameters [1], [2], leakage on gradients during training [3], or just observations on the prediction results [4] may enable an adversary to successfully distinguish and even reconstruct the private samples used for learning. As an emergent canonical definition, Differential Privacy (DP) [5], [6] provides a semantic privacy metric to quantify how hard it is for an adversary to infer the participation of an individual in an aggregate statistic. As one of the most popular approaches, Differentially-Private Stochastic Gradient Descent (DP-SGD) [7], [8] and its variants [9], [10], [11], [12], [13], [14] have been widely studied over the last decade. DP-SGD can be applied to almost all optimization problems in machine learning to produce rigorous DP guarantees without additional assumptions regarding the objective function or dataset. However, despite its broad applicability, DP-SGD also suffers notoriously large utility loss especially when training cutting-edge deep models. Its practical implementation is also known to be sensitive to hyper-parameter selections [15], [16], [17]. Indeed, even in theory, the effects of the two artificial privatization operations applied in DP-SGD, *iterative gradient perturbation* and *gradient clipping*, are still not fully-understood.

To understand why these two artificial modifications are the key to differentially privatize iterative methods, we need to first introduce the concept of *sensitivity*, which plays a key role in DP. The sensitivity captures the maximum impact an individual sample from an input dataset may have on an algorithm's output. It is the foundation of almost all DP mechanisms, including the Laplace/Gaussian and Exponential Mechanisms [18], where the sensitivity determines how much randomization is needed to to hide any individual amongst the population with desired privacy. Unfortunately, in many practical optimization problems, the sensitivity is intractable or can only be loosely bounded.

To this end, DP-SGD proposes an alternative solution by assuming a more powerful adversary. In most private (centralized) learning applications, the standard black-box adversary can only observe the final model revealed. DP-SGD, on the other hand, assumes an adversary who can observe the intermediate updates during training. For convenience, we will call such an adversary a white-box adversary. Provided such an empowered adversary, DP-SGD clips the gradient evaluated by each individual sample and adds random noises to the updates in each iteration. Clipping guarantees that the sensitivity is bounded within each iteration. The total privacy loss is then upper bounded by a composition of the leakage from all iterations.

For convex optimization with Lipschitz continuity, where the norms of gradients are uniformly bounded by some given constant, DP-SGD is known to produce an asymptotically tight privacy-utility tradeoff [8]. However, it is, in general, impractical to assume Lipschitz continuity in tasks such as deep learning. Either asymptotically or non-asymptotically, the study of practical implementations of DP-SGD with more realistic and specific assumptions remains active [13], [15], [17], [19] and demanding. Much research effort has been dedicated to tackling the following two fundamental questions. First, *provided that we do not need to publish the intermediate computation results, how conservative is the privacy claim offered by DP-SGD?* Second, during practical implementation, *how to properly select the training model and hyper-parameters?*

Regarding the first question, many prior works [4], [20], [21] tried to empirically simulate what the adversary can infer from models trained by DP-SGD. In particular, [20] examined the respective power of "black-box" and "white-box" adversaries, and suggested that a substantial gap between the DP-SGD privacy bound and the actual privacy guarantee may exist. Unfortunately, beyond DP-SGD, there are few known ways to produce, let along improve, rigorous DP analysis for a training process. Most existing analyses need to assume either access to additional public data [13], [22], [23], or strongly-convex loss functions to enable objective perturbation [24]. Thus, for general applications, we still have to adopt the conservative DP-SGD analysis for the worst-case DP guarantees.

The second question is of particular interest to practitioners. The implementation of DP-SGD is tricky as the performance of DP-SGD is highly sensitive to the selection of the training model and hyper-parameters. The lack of theoretical analysis on gradient clipping makes it hard to find good parameters and optimize model architectures instructively, though many heuristic observations and optimizations on these choices are reported. [16], [19], [25] showed how to find proper model architectures to balance learning capability and utility loss when the dataset is given. [1] Recent work [27] also demonstrated empirical improvements through selecting the clipping threshold adaptively. However, even with these efforts, there is still a long way to go if we want to practically train large neural networks with rigorous and usable privacy guarantees. The biggest bottlenecks include

- the huge model dimension, which may be even larger than the size of the training dataset, and
- the long convergence time, which implies a massive composition of privacy loss and also forces DP-SGD to add formidable noise resulting in intolerable accuracy loss.

To this end, Tramer and Boneh in [19] argued that within the current framework, for most medium datasets ($<$ 500K datapoints) such as CIFAR10/100 and MNIST, the utility loss caused by DP-SGD will offset the powerful learning capacity offered by deep models. Therefore, simple linear models usually outperform the modern Convolutional Neural Network (CNN) for these datasets. How to privately train a model while still being able to enjoy the state-of-the-art success in modern machine learning is one of the key problems in the application of DP.

1.1. Our Strategy and Results

In this paper, we set out to provide a systematic study of DP-SGD from both theoretical and empirical perspectives to understand the two important but artificial operations: (1) iterative gradient perturbation and (2) gradient clipping. We propose a generic technique, *ModelMix*, to significantly sharpen the utility-privacy tradeoff. Our theoretical analysis also provides instruction on how to select the clipping parameter and quantify privacy amplification from other practical randomness.

We will stick to the worst-case DP guarantee without any relaxation, but view the private iterative optimization process from a different angle. In most practical deep learning tasks, with proper use of randomness, we will have a good chance of finding some reasonable (local) minimum via SGD regardless of the initialization (starting point) and the subsampling [28]. In particular, for convex optimization, we are guaranteed to approach the global optimum with a proper step size. In other words, *there are an infinite number of potential training trajectories* [2] *pointing to some (local) minimum of good generalization, and we are free to use any one of them to find a good model.* Thus, even without DP perturbation, the training trajectory has potential entropy if we are allowed to do random selection.

From this standpoint, a slow convergence rate when training a large model is not always bad news for privacy. This might seem counter-intuitive. But, in general, slow convergence means that the intermediate updates wander around a relatively large domain for a longer time before entering a satisfactory neighborhood of (global/local) optimum. Training a larger model may produce a more fuzzy and complicated convergence process, which could compensate the larger privacy loss composition caused in DP-SGD. We have to stress that our ultimate goal is to privately publish a good model, while DP-SGD with exposed updates is merely a tool to find a trajectory with analyzable privacy leakage. The above observation inspires a way to find a better DP guarantee even under the conservative "whitebox" adversary model: *can we utilize the potential entropy of the training trajectory while still bounding the sensitivity to produce rigorous DP guarantees?*

To be specific, different from standard DP-SGD which randomizes a particular trajectory with noise, we aim to privately construct an *envelope* of *training trajectories*, spanned by the many trajectories converging to some (global/local) minimum, and randomly generate one trajectory to amplify privacy. To achieve this, we must carefully consider the tradeoff between (1) controlling the worst-case sensitivity in the trajectory generalization and (2) the learning bias resultant from this approach. We summarize our contributions as follows.

(a) We present a generic optimization framework, called *ModelMix*, which iteratively builds an envelope of training trajectories through post-processing historical updates, and randomly aggregates those model states before applying gradient descent. We provide rigorous convergence and privacy analysis for *ModelMix*, which enables us to quantify (ϵ, δ)-DP budget of our protocol. The refined privacy analysis framework proposed can also be used to capture the privacy amplification of a

1. Most prior works report the best model and parameter selection by grid searching, where the private data is reused multiple times and the selection of parameters itself is actually sensitive. This additional privacy leakage, partially determined by the prior knowledge on the training data is in general very hard to quantify, though in practice it might be small [19], [26].

2. We will use *training trajectory* in the following to represent the sequences of intermediate updates produced by SGD.

large class of *training-purpose-oriented* operations commonly used in deep learning. This class of operations include data augmentation [29] and stochastic gradient Langevin dynamics (SGLD) [30], which cannot produce reasonable worst-case DP guarantees by themselves.

(b) We study the influence of gradient clipping in private optimization and present the first generic convergence rate analysis of clipped DP-SGD in deep learning. To our best knowledge, this is the first analysis of non-convex optimization via clipped DP-SGD with only mild assumptions on the concentration of stochastic gradient. We show that the key factor in clipped DP-SGD is the *sampling noise* [3] of the stochastic gradient. We then demonstrate why implementation of DP-SGD by clipping individual sample gradients can be unstable and sensitive to the selection of hyper-parameters. Those analyses can be used to instruct how to select hyper-parameters and improve network architecture in deep learning with DP-SGD.

(c) ModelMix is a fundamental improvement to DP-SGD, which can be applied to almost all applications together with other advances in DP-SGD, such as low-rank or low-dimensional gradient embedding [31], [32] and fine-tuning based transfer learning [33], [19] (if additional public data is provided). In our experiments, we focus on computer vision tasks, a canonical domain for private deep learning. We evaluate our methods on CIFAR-10, FMNIST and SVHN datasets using various neural network models and compare with the state-of-the-art results. Our approach improves the privacy/utility tradeoff significantly. For example, provided a privacy budget $(\epsilon = 8, \delta = 10^{-5})$, we are able to train Resnet20 on CIFAR10 with accuracy 70.4% compared to 56.1% when applying regular DP-SGD. As for private transfer learning on CIFAR10, we can improve the $(\epsilon = 2, \delta = 10^{-5})$-DP guarantee in [19] to $(\epsilon = 0.64, \delta = 10^{-5})$ producing the same 92.7% accuracy.

The remainder of this paper is organized as follows. In Section 2, we introduce background on statistical learning, differential privacy and DP-SGD. In Section 3, we formally present the ModelMix framework, whose utility in both convex and non-convex optimizations is studied in Theorem 3.1 and Theorem 3.2, respectively. In Section 4, we show how to efficiently compute the amplified (ϵ, δ) DP security parameters in Theorem 4.1 and a non-asymptotic amplification analysis is given in Theorem 4.2. Further experiments with detailed comparisons to the state-of-the-art works are included in Section 5. Finally, we conclude and discuss future work in Section 6.

1.2. Related Works

Theoretical (Clipped) DP-SGD Analysis: When DP-SGD was first proposed [7], and in most theoretical studies afterwards [8], [10], the objective loss function is assumed to be L-Lipschitz continuous, where the L_2 norm of the gradient is uniformly bounded by L. This enables a straightforward privatization on SGD by simply perturbing the gradients. In particular, for convex optimization on a dataset of n samples, a training loss of $\Theta(\sqrt{d\log(1/\delta)}/(\epsilon n))$ is known to be tight under an (ϵ, δ) DP guarantee [8].

However, it is hard to get a (tight) Lipschitz bound for general learning tasks. A practical version of DP-SGD was then presented in [9], where the Lipschitz assumption is replaced by gradient clipping to ensure bounded sensitivity. This causes a disparity between the practice and the theory as classic results [8], [10] assuming bounded gradients cannot be directly generalized to clipped DP-SGD. Some existing works tried to narrow this gap by providing new analysis. [34] presented a convergence analysis of smooth optimization with clipped SGD when the sampling noise in stochastic gradient is bounded. But [34] requires the clipping threshold c to be $\Omega(T)$ where T is the total number of iterations. This could be a strong requirement, as in practice, the iteration number T can be much larger than the constant clipping threshold c selected. [35] relaxed the requirement with an assumption that the sampling noise is symmetric. [17] studied the special case where clipped DP-SGD is applied to generalized linear functions. In this paper, we give the first generic analysis of clipped DP-SGD with only mild assumptions on the concentration property of stochastic gradients.

Assistance with Additional Public Data: When additional unrestricted (unlabeled) public data is available, an alternative model-agnostic approach is *Private Aggregation of Teacher Ensembles* (PATE) [22], [23], [36]. PATE builds a teacher-student framework, where private data is first split into multiple (usually hundreds) disjoint sets, and a teacher model is trained over each set separately. Then, one can apply those teacher models to privately label public data via a private majority voting. Those privately labeled samples are then used to train a student model, as a postprocessing of labeled samples. Another line of works considers improving the noise added in DP-SGD with public data. For example, in private transfer learning, we can first pretrain a large model with public data and then apply DP-SGD with private data to fine-tune a small fraction of the model parameters [19], [33]. However, both PATE and private transfer learning have to assume a large amount of public data. Another idea considers the projection of the private gradient into a low-rank/dimensional subspace, approximated by public samples, to reduce the magnitude of noise [31], [32], [37], [38]. When the public samples are limited, DP-SGD with low-rank gradient representation usually outperforms the former methods.

Except for PATE, our methods can be, in general, used to further enhance those state-of-the-art DP-SGD improvements with public data. For example, using 2K ImageNet public samples, the low-rank embedding method in [32] can train Resnet20 on CIFAR10 with 79.1% accuracy at a cost of $(\epsilon = 111.2, \delta = 10^{-5})$-DP; while ModelMix can improve the DP guarantee to $(\epsilon = 6.1, \delta = 10^{-5})$-DP with the same accuracy as shown in Section 5.3.

3. The noise corresponds to using a minibatch of samples to estimate the true full-batch gradient.

2. Preliminaries

Empirical Risk Minimization: In statistical learning, the model to be trained is commonly represented by a parameterized function $f(w,x) : (\mathcal{W},\mathcal{X}) \to \mathbb{R}$, mapping feature x from input domain $\mathcal{X}$ into an output (prediction/ classification) domain. In the following, we will always use d to represent the dimensionality of the parameter w, i.e., $w \in \mathbb{R}^d$. For example, one may consider $f(w,x)$ as a neural network with a sequence of linear layers connected by non-linear activation layers, and w represents the weights to be trained. Given a set $\mathcal{D}$ of n samples $\{(x_i,y_i), i = 1,2,...,n\}$, we define the problem of Empirical Risk Minimization (ERM) for some loss function $l(\cdot,\cdot)$ as follows,

$$\min_w F(w) = \min_w \frac{1}{n} \cdot \sum_{i=1}^{n} l(f(w,x_i),y_i). \tag{1}$$

For convenience, we simply use $f(w,x_i,y_i)$ to denote the objective loss function $l(f(w,x_i),y_i)$ in the rest of the paper. Below, we formally introduce the definitions of *Lipschitz continuity, smoothness* and *convexity*, which are commonly used in optimization research.

Definition 2.1 (Lipschitz Continuity)**.** *A function g is L-Lipschitz if for all $w,w' \in \mathcal{W}$, $|g(w)-g(w')| \leq L\|w-w'\|_2$.*

Definition 2.2 (Smoothness)**.** *A function g is β-smooth on $\mathcal{W}$ if for all $w,w' \in \mathcal{W}$, $g(w') \leq g(w)+\langle \nabla g(w), w'-w\rangle + \frac{\beta}{2}\|w'-w\|_2^2$.*

Definition 2.3 (Convexity)**.** *A function g is convex on $\mathcal{W}$ if for all $w,w' \in \mathcal{W}$ and $t \in (0,1)$, $f(tw+(1-t)w') \leq tg(w)+(1-t)g(w')$.*

In the following, we will simply use $\|\cdot\|$ to denote the l_2 norm unless specified otherwise.

Differential Privacy (DP): We first formally define (ϵ,δ)-DP and (α,ϵ)-Rényi DP as follows.

Definition 2.4 (Differential Privacy)**.** *Given a data universe $\mathcal{X}^*$, we say that two datasets $\mathcal{D},\mathcal{D}' \subseteq \mathcal{X}^*$ are neighbors, denoted as $\mathcal{D} \sim \mathcal{D}'$, if $\mathcal{D} = \mathcal{D}' \cup s$ or $\mathcal{D}' = \mathcal{D} \cup s$ for some additional datapoint s. A randomized algorithm $\mathcal{A}$ is said to be (ϵ,δ)-differentially private (DP) if for any pair of neighboring datasets $\mathcal{D},\mathcal{D}'$ and any event S in the output space of $\mathcal{A}$, it holds that*

$$\mathbb{P}(\mathcal{A}(\mathcal{D}) \in S) \leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{A}(\mathcal{D}') \in S) + \delta.$$

Definition 2.5 (Rényi Differential Privacy [39])**.** *A randomized algorithm $\mathcal{A}$ satisfies (α,ϵ)-Rényi Differential Privacy (RDP) if for any pair of neighboring datasets $\mathcal{D} \sim \mathcal{D}'$,*

$$\epsilon \geq \mathsf{D}_\alpha(\mathcal{M}(\mathcal{D})\|\mathcal{M}(\mathcal{D}')).$$

Here,

$$\mathsf{D}_\alpha(\mathsf{P}\|\mathsf{Q}) = \frac{1}{\alpha-1}\log\int \mathsf{q}(o)\left(\frac{\mathsf{p}(o)}{\mathsf{q}(o)}\right)^\alpha do, \tag{2}$$

represents α-Rényi Divergence between two distributions P and Q whose density functions are p and q, respectively.

In Definition 2.4 and 2.5, if two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ are defined in a form that $\mathcal{D}$ can be obtained by arbitrarily replacing an datapoint in $\mathcal{D}'$, then they become the definitions of bounded DP [5], [6] and RDP, respectivaly. In this paper, we adopt the unbounded DP version to match existing DP deep learning works [9], [19], [32] with a fair comparison.

In practice, to achieve meaningful privacy guarantees, ϵ is usually selected as some small one-digit constant and δ is asymptotically $O(1/|\mathcal{D}|) = O(1/n)$. To randomize an algorithm, the most common approaches in DP are Gaussian or Laplace Mechanisms [18], where a Gaussian or Laplace noise proportional to the sensitivity is added to perturb the algorithm's output. In many applications, including the DP-SGD analysis, we need to quantify the cumulative privacy loss across sequential queries of some differentially private mechanism on one dataset. The following theorem provides an upper bound on the overall privacy leakage.

Theorem 2.1 (Advanced Composition [40])**.** *For any $\epsilon > 0$ and $\delta \in (0,1)$, the class of (ϵ,δ)-differentially private mechanisms satisfies $(\tilde{\epsilon}, T\delta+\tilde{\delta})$-differential privacy under T-fold adaptive composition for any $\tilde{\epsilon}$ and $\tilde{\delta}$ such that*

$$\tilde{\epsilon} = \sqrt{2T\log(1/\tilde{\delta})}\cdot\epsilon + T\epsilon(e^\epsilon - 1).$$

Theorem 2.2 (Advanced Composition via RDP [39])**.** *For any $\alpha > 1$ and $\epsilon > 0$, the class of (α,ϵ)-RDP mechanisms satisfies $(\tilde{\epsilon},\tilde{\delta})$-differential privacy under T-fold adaptive composition for any $\tilde{\epsilon}$ and $\tilde{\delta}$ such that*

$$\tilde{\epsilon} = T\epsilon - \log(\tilde{\delta})/(\alpha-1).$$

Theorem 2.1 provides a good characterization on how the privacy loss increases with composition. For small (ϵ,δ), we still have an $\tilde{O}(\sqrt{T}\epsilon, T\delta)$ DP guarantee after a T composition. In practice using RDP, Theorem 2.2 usually produces tighter constants in the privacy bound.

DP-SGD: (Stochastic) Gradient Descent ((S)GD) is a very popular approach to optimize a function. Suppose we try to solve the ERM problem and minimize some function $F(w) = \frac{1}{n}\sum_{i=1}^n f(w,x_i,y_i)$. SGD can be described as the following iterative protocol. In the $(k+1)$-th iteration, we apply Poisson sampling, i.e., each datapoint is i.i.d. sampled by a constant rate q, and a minibatch of B_k samples is produced from the dataset $\mathcal{D}$, denoted as S_k. We calculate the stochastic gradient as

$$G_k \leftarrow \sum_{(x_i,y_i)\in S_k} \nabla f(w_k,x_i,y_i). \tag{3}$$

Then, a gradient descent update is applied using

$$w_{k+1} = w_k - \eta \cdot G_k, \tag{4}$$

for some stepsize η. In particular, if the minibatch is selected to be the full batch, i.e., $S_k = \mathcal{D}$, then Equation (4) becomes the standard gradient descent procedure.

We make the following assumption regarding the sampling noise $\|\nabla f(w,x,y) - \nabla F(w)\|$ when the minibatch size equals 1. This assumption, which will be shown to be

necessary in Example 3.1, will be used in Theorem 3.2 when we derive the concrete convergence rate for clipped SGD.

Assumption 2.1 (Stochastic Gradient of Sub-exponential Tail)**.** *There exists some constant $\kappa > 0$ such that for any w, if we randomly select a datapoint (x, y) from $\mathcal{D}$, then*

$$\Pr(\|\nabla f(w, x, y) - \nabla F(w)\| \geq t) \leq e^{-t/\kappa}.$$

In Assumption 2.1, a larger κ implies stronger concentration, i.e., a faster decaying tail of the stochastic gradient. The modification from GD/SGD to its corresponding DP version is straightforward. When the loss function f is assumed to be L-Lipschitz [7], [8], i.e., $\|\nabla f(w, x_i, y_i)\| \leq L$ for any w, the worst-case sensitivity in Equation (4) is bounded by ηL in each iteration. One can derive a tighter bound [8] using existing results on the privacy amplification from sampling [41], [42]. Thus, SGD can be made private via iterative perturbation by replacing Equation (4) with the following:

$$w_{k+1} = w_k - \eta \cdot (G_k + \Delta_{k+1}), \tag{5}$$

where Δ_k is the noise for the k-th iteration. For example, if we want to use the Gaussian Mechanism to ensure (ϵ, δ)-DP when running T iterations, then Δ_{k+1} can be selected to be i.i.d. generated from

$$\Delta_{k+1} \leftarrow \mathcal{N}(\mathbf{0}, O(\frac{L^2 T \log(1/\delta)}{\epsilon^2}) \cdot \boldsymbol{I}_d).$$

Here, $\boldsymbol{I}_d$ represents the $d \times d$ identity matrix.

However, when we do not have the Lipschitz assumption, an alternative is to force a limited sensitivity through gradient clipping. Following the same notations as before, we describe DP-SGD with per-sample gradient clipping [9] as follows,

$$\begin{aligned} G_k &\leftarrow \sum_{(x_i, y_i) \in S_k} \mathsf{CP}\big(\nabla f(w_k, x_i, y_i), c\big); \\ w_{k+1} &= w_k - \eta \cdot (G_k + \Delta_{k+1}). \end{aligned} \tag{6}$$

Here, $\mathsf{CP}(\cdot, c)$ represents a clipping function of threshold c,

$$\mathsf{CP}(\nabla f(w, x, y), c) = \nabla f(w, x, y) \cdot \min\{1, \frac{c}{\|\nabla f(w, x, y)\|}\}.$$

With clipping, the l_2 norm of each per-sample gradient is bounded by c. Thus, the clipping threshold c virtually plays the role of the Lipschitz constant L in clipped SGD for privacy analysis.

3. ModelMix

In this section, we formally introduce ModelMix, and explain how it sharpens the utility-privacy trade-off in DP-SGD.

3.1. Intuition

We begin with the following observation. Suppose we run SGD twice on a least square regression $F(w) = 1/n \cdot \sum_{i=1}^{n} \|\langle w, x_i \rangle - y_i\|^2$ for T iterations and obtain two training trajectories $\boldsymbol{w} = (w_1, w_2, ..., w_T)$ and $\boldsymbol{w'} = (w'_1, w'_2, ..., w'_T)$. Suppose both w_T and w'_T are σ-close to the optimum $w^* = \arg\min_w F(w)$, i.e.,

$$\|w_T - w^*\| \leq \sigma \text{ and } \|w'_T - w^*\| \leq \sigma.$$

Due to the linearity of gradients in least square regression, if we mix $\boldsymbol{w}$ and $\boldsymbol{w'}$ to get $w''_k = \alpha w_k + (1 - \alpha) w'_k$ ($k = 1, 2, ..., T$) for some weight $\alpha \in (0, 1)$, then we produce a new SGD trajectory $\boldsymbol{w''}$ where w''_T is also σ-close to w^*.

This simple example gives us two inspirations. First, as mentioned earlier, with different randomness in initialization and subsampling, the training trajectory to find an optimum is not unique. Even in DP-SGD where we need to virtually publish the trajectory for analytical purpose, we are not restricted to expose a particular one. Second, and more important, this means that we have more freedom to randomize the SGD process. In the k-th iteration, instead of following the regular SGD rule in Equation (4) where we simply start from the previously updated state w_{k-1}, we can randomly mix w_{k-1} with some other w'_{k-1} from another reasonable training trajectory, and then *move to a new trajectory to proceed*.

However, the above idea to randomly mix the trajectories cannot be directly implemented as we do not have any prior knowledge on what good trajectories look like. Any training trajectory generated by the private dataset is sensitive and potentially creates privacy leakage. Thus, we need to be careful about how we generate the needed envelope. Recall that we use *envelope* to describe the space spanned by the mixtures of training trajectories. Provided the property that DP is immune to post-processing, we consider approximating the trajectory envelope using intermediate states already published. This allows us to privately construct the envelope and advance the optimization, simultaneously.

3.2. Algorithm and Observations

We start with a straw-man solution where we virtually run DP-SGD to alternately train two models in turns. We initialize two states $\tilde{w}_0^1$ and $\tilde{w}_0^2$ with respect to (w.r.t) the parameterized function we aim to optimize.

At any odd iteration, i.e., iteration $2k + 1$ ($k \geq 0$), we randomly generate $\boldsymbol{\alpha}_{2k+1} \in (0, 1)^d$ whose coordinate is i.i.d. uniform in $(0, 1)$. We then update the state of the first model $\tilde{w}^1$ as

$$\bar{w}_k^1 \leftarrow \boldsymbol{\alpha}_{2k+1} \circ \bar{w}_{k-1}^1 + (\mathbf{1} - \boldsymbol{\alpha}_{2k+1}) \circ \bar{w}_{k-1}^2 - \eta\big(\nabla F(\tilde{w}_{k-1}^1) + \Delta_{2k+1}\big), \tag{7}$$

where $\circ$ represents the Hadamard product and Δ represents the noise added. Essentially, we mix the two states $\bar{w}_{k-1}^1$ and $\bar{w}_{k-1}^2$ coordinate-wise. Similarly, at any even iteration, i.e., iteration $2(k + 1)$, we randomly generate some $\boldsymbol{\alpha}_{2k+2}$ and update the state of the second model $\tilde{w}^2$ as

$$\bar{w}_k^2 \leftarrow \boldsymbol{\alpha}_{2k+2} \circ \bar{w}_k^1 + (\mathbf{1} - \boldsymbol{\alpha}_{2k+2}) \circ \bar{w}_{k-1}^2 - \eta\big(\nabla F(\tilde{w}_{k-1}^2) + \Delta_{2k+2}\big). \tag{8}$$

We take turns mixing and updating two training trajectories privately using (7) and (8). At the high level, this is similar to a distributed GD, where two agents collaboratively train the model.

However, in a centralized scenario, it is unnecessary to artificially create two models and train them in a distributed manner, meaning that the efficiency of the above mixture method is not optimal. In the following, we propose an improved method where in the k-th iteration, we simply post-process on w_{k-1} and w_{k-2}, the updates already privately generated from the last two iterations, instead of two virtual models to approximate an envelope of training trajectory. The formal description of ModelMix is shown in Algorithm 1. We provide an illustration of how Algorithm 1 works in Fig. 1.

Algorithm 1 Differentially Private Stochastic Gradient Descent with ModelMix

1: **Input:** Objective function $F(w) = \frac{1}{n} \cdot \sum_{i=1}^{n} f(w, x_i, y_i)$, dataset $\mathcal{D} = \{(x_i, y_i), i = 1, 2, ..., n\}$, sampling rate q, step size η, clipping threshold c, total number of iterations T, mixing thresholds $\tau_{[1:T]}$, initialized model states $w_{-1}, w_0 \in \mathbb{R}^d$ and noise sequence $\Delta_{[1:T]}$.
2: **for** $k = 1, 2, ..., T$ **do**
3: Through i.i.d. sampling with a constant rate q, produce a minibatch S_k of B_k samples from $\mathcal{D}$ and calculate the stochastic gradient

$$G_{k-1} = \sum_{(x_i, y_i) \in S_k} \mathsf{CP}\big(\nabla f(w_{k-1}, x_i, y_i), c\big).$$

4: **for** $j = 1, 2, ..., d$ **do**
5: $\boldsymbol{\alpha}_k(j) \leftarrow \mathcal{U}[0, 1]$, a uniform distribution in [0,1].
6: **if** $|w_{k-1}(j) - w_{k-2}(j)| < \tau_k$ **then**
7: $w_{k-1}(j) \leftarrow w_{k-1}(j) + \text{sign}(w_{k-1}(j) - w_{k-2}(j)) \cdot \tau_k/2$;
8: $w_{k-2}(j) \leftarrow w_{k-2}(j) - \text{sign}(w_{k-1}(j) - w_{k-2}(j)) \cdot \tau_k/2$.
9: **end if**
10: Update the weight as follows:

$$\begin{aligned} w_k(j) =& \boldsymbol{\alpha}_k(j) \cdot w_{k-1}(j) + (1 - \boldsymbol{\alpha}_k(j)) \cdot w_{k-2}(j) \\ & - \eta \cdot (G_{k-1}(j) + \Delta_k(j)). \end{aligned} \tag{9}$$

11: **end for**
12: **end for**
13: **Output**: w_T.

From a more concise optimization standpoint, the mixed state in expectation is the average of the last two iterations' states $(w_{k-1} + w_{k-2})/2$. Therefore, in the expectation of Equation (9), we are essentially optimizing the original objective function $F(w)$ plus a proximal term in a form

$$F(w) + \|w - w_{k-2}\|^2/4, \tag{10}$$

whose gradient at w_{k-1} is $\nabla F(w_{k-1}) + (w_{k-1} - w_{k-2})/2$. Thus, ModelMix can also be viewed as introducing a randomized proximal term into the original objective function $F(w)$. ModelMix operation $\boldsymbol{\alpha}_k w_{k-1} + (\mathbf{1} - \boldsymbol{\alpha}_{k-1}) w_{k-2}$ averages out the noise added in the previous iterations, which makes the convergence more stable. This enables us to apply a larger step size during training, as shown later in Fig. 2.

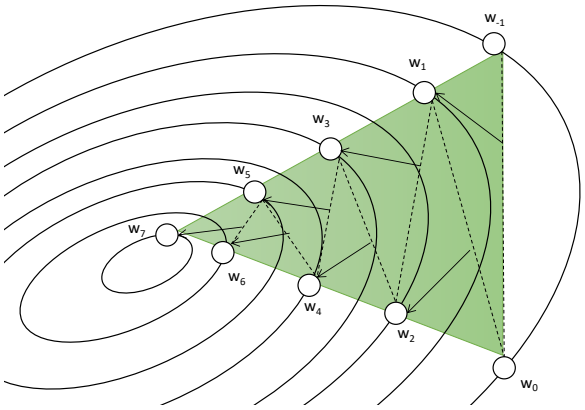

Figure 1: An illustration of how ModelMix works. Each node represents an intermediate state. The dashed lines represent the mixing and the arrows represent gradient descent. The green area represents the approximated trajectory envelope.

As will be shown later in Theorem 4.2, the privacy amplification is determined by the distance between w_{k-1} and w_{k-2}. Therefore, for a privacy analysis purpose, in steps 5-8 of Algorithm 1, we artificially ensure that the coordinate-wise distance between the states w_{k-1} and w_{k-2} is at least some parameter τ_k. In practice, the gap between the two states w_{k-1} and w_{k-2} already exists even without these operations, meaning that ModelMix naturally enjoys a privacy amplification. We only add those steps to enforce a worst-case lower bound on the coordinate-wise distance so that we can quantify the privacy amplification in a clean way.

A natural question that follows is *how do the parameters $\tau_{[1:T]}$ across T iterations affect privacy and utility*? τ_k captures the distance between the two training trajectories, or in other words, the volume of the approximated envelope. Ideally, we would like $\tau_{[1:T]}$ to match the size of the true envelope so that the impact of those artificial steps is minimized. Empirically, a harder learning task often has a slower convergence rate, and thus also has an envelope of larger volume. This allows us to select a larger τ_k and provide stronger privacy amplification. On the other hand, if the learning task is simple and we overestimate the size of the envelope with a large τ_k, then the learning rate might be compromised. This analysis is supported by Fig. 2, where we test the efficiency of ModelMix in two different tasks with various setups.

In Fig. 2 (a) and (b), we implement non-private SGD with or without ModelMix to train Resnet20 [43] on CIFAR10 [4] and SVHN [5], the two benchmark datasets. We set the sampling rate q to be 0.05 and run for 2,000 iterations. Compared to CIFAR10, classification on SVHN is an easier task. We fix the mixing threshold $\tau_k = \tau$ to be proportional to the step size η in gradient descent (Equation (9)), specifically, 0.1η and 0.05η.

4. https://www.cs.toronto.edu/ kriz/cifar.html
5. http://ufldl.stanford.edu/housenumbers/

In both Fig. 2 (a) and (b), different setups perform similarly in the initial stage (the first 400 iterations). However, in later epochs, as the training trajectories approach the (local) optimum, setups with larger τ suffer heavier losses. This phenomenon is clearer in Fig. 2 (b) which trains on the easier SVHN task. This matches our previous analysis: the size of the trajectory envelope gets smaller when (1) the learning task is easier or (2) we approach the optimum. On the other hand, with proper selection of $\tau = 0.05\eta$, in the non-private case ModelMix comes with almost no additional utility loss.

In Fig. 2 (c) and (d), we implement the private case where we will see how ModelMix strengthens the robustness of DP-SGD. We clip the per-sample gradient norm down to 8 and add the same amount of Gaussian noise to both DP-SGD cases, with or without ModelMix. This ensures an $(\epsilon = 8, \delta = 10^{-5})$ guarantee for regular DP-SGD. *As shown later in Theorem 4.2, ModelMix can achieve a much better DP guarantee under the same setup. But here, we focus on the performance of ModelMix given the same noise as regular DP-SGD to provide a clear comparison.* With the virtual proximal term in Equation (10), ModelMix allows us to use a larger stepsize resulting in a faster learning rate. In Fig. 2 (c) and (d), we set the stepsize $\eta = 0.2$ for ModelMix, and compare to the regular DP-SGD with $\eta = 0.1$ (black line) and $\eta = 0.2$ (red line). The larger stepsize $\eta = 0.2$ worsens the performance of regular DP-SGD, whereas with proper selection of $\tau = 0.05\eta$ (green line), ModelMix allows the application of larger stepsize and slightly out-performs regular DP-SGD.

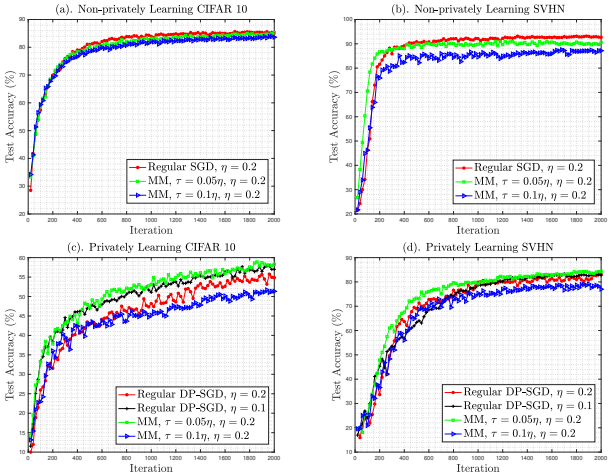

Figure 2: Performance comparison between DP-SGD with/without ModelMix (MM) provided with the same noise.

3.3. Utility Guarantee in (non)-convex Optimization and Effect of Gradient Clipping

In this subsection, we will provide formal utility analysis of Algorithm 1. As a warm-up, in Theorem 3.1, we start with convex optimization with a Lipschitz and smooth assumption, and show Algorithm 1 enjoys an $1/\sqrt{T}$ convergence rate. Specifically, after T iterations, if we set the average

$$\bar{w} = \frac{\sum_{k=1}^{T}(w_{k-1} + w_{k-2})}{2T},$$

where w_k represents the weight in the k-th iteration, then the utility loss $F(\bar{w}) - F(w^*)$ is bounded by $O(1/\sqrt{T})$. Based on Theorem 3.1, we then move to the non-convex case without a Lipschitz assumption in Theorem 3.2, and show a generic convergence rate.

Theorem 3.1 (Utility of ModelMix in Convex Optimization)**.** *Suppose $f(w, x, y)$ is L-Lipschitz and β-smooth convex. We set the clipping threshold $c = L$. For any $\gamma > 0$, if we set $\eta = \gamma/(nq\sqrt{T})$, then Algorithm 1 satisfies*

$$\begin{aligned}
&\mathbb{E}\big[F(\bar{w}) - F(w^*)\big] \\
&\leq \frac{3\mathcal{W}_0^2 + \sum_{k=1}^{T} d\tau_i^2/12}{2\gamma\sqrt{T}} + \frac{\gamma(L^2/q^2 + \mathbb{E}[\|\Delta\|^2]/(n^2q^2))}{\sqrt{T}} + \beta\cdot \\
&\big(\frac{12\mathcal{W}_0^2}{8T} + \frac{2\gamma\mathcal{W}_0(L + \mathbb{E}[\|\Delta\|]/n)}{qT^{3/2}} + \frac{11\gamma^2(L^2 + \mathbb{E}[\|\Delta\|^2]/n^2)}{8Tq^2}\big) \\
&= O\big(\frac{\mathcal{W}_0^2 + \sum_{k=1}^{T} d\tau_k^2}{\gamma\sqrt{T}} + \frac{\gamma(L^2 + \mathbb{E}[\|\Delta\|^2]/n^2)}{q^2\sqrt{T}}\big)
\end{aligned} \tag{11}$$

where $\mathcal{W}_0 = \sup_w \|w - w^*\|$ denotes the initial divergence and $\mathbb{E}[\|\Delta\|^2] = \mathbb{E}[\|\Delta_k\|^2]$ denotes the variance of noise added in each iteration.

Proof. See Appendix A. □

From Equation (11), one can see that with ModelMix, Algorithm 1 still enjoys an $O(1/\sqrt{T})$ convergence rate when $\tau_k = O(\eta) = O(1/\sqrt{T})$. Using Theorem 3.1, we can upper bound the utility loss of Algorithm 1. Since ModelMix can be seen as post-processing operations on DP-SGD, its privacy guarantee is at least as good as the privacy guarantee of DP-SGD. If we set Δ_k to be Gaussian noise generated from $\mathcal{N}(0, O(q^2L^2T\log(1/\delta)/\epsilon^2)\cdot\boldsymbol{I}_d)$ [9], then by Theorem 3.1, Algorithm 1 satisfies an (ϵ, δ)-DP guarantee and its utility loss is upper bounded by $O(\sqrt{d}\log(1/\delta)L/(n\epsilon))$. This asymptotically matches the classic results in [8]. However, we will present a more fine-tuned analysis in Section 4 to show the randomness in ModelMix can significantly sharpen the composite privacy loss.

In the following, we no longer assume the objective loss function to be either convex or Lipschitz continuous, and we try to understand the effect of gradient clipping. The following theorem gives a generic convergence analysis of ModelMix in non-convex optimization with clipped gradient.

Theorem 3.2 (Utility of ModelMix in non-Convex Optimization with Gradient Clipping)**.** *Suppose the objective loss function $F(w)$ is β-smooth and satisfies Assumption 2.1, then there exists some constant $\psi > 0$ such that when the clipping threshold c satisfies*

$$c \geq \max\{4\kappa\log(10), -\psi\kappa\log(\kappa)\log(\frac{\sqrt{d\log(1/\delta)}}{n\epsilon})\}, \tag{12}$$

then the convergence rate of Algorithm 1, which applies per-sample gradient clipping up to c and enjoys an (ϵ, δ)-DP guarantee, satisfies

$$\mathbb{E}\Big[\frac{\sum_{k=0}^{T-1} \min\left\{9/20 \cdot \|\nabla F(w_k)\|^2, c/20 \cdot \|\nabla F(w_k)\|\right\}}{T}\Big] \leq (\frac{v}{2}+\frac{5}{2})\frac{c\sqrt{\mathcal{R}_F \frac{101}{12}\beta d \log(1/\delta)}}{n\epsilon} + \frac{28c\beta d\log(1/\delta)\tilde{\mathcal{W}}_0}{12q(n\epsilon)^2} + \frac{cd\log(1/\delta)\sqrt{\frac{101}{12}}\beta^{3/2}}{qn\epsilon\sqrt{\mathcal{R}_F}}(\frac{\sum_{k=1}^{T} d\tau_k^2}{12}+\frac{21\tilde{\mathcal{W}}_0^2}{24}), \tag{13}$$

where $\mathcal{R}_F = \sup_w F(w) - \inf_w F(w)$, v is some constant determined by the noise mechanism and $\tilde{\mathcal{W}}_0 = \|w_0 - w_{-1}\|$ is the initial divergence. When $\tau_k = O(\eta)$, the right hand of (13) is $O(\frac{c\sqrt{d\log(1/\delta)}}{n\epsilon})$.

Proof. See Appendix B. □

In Theorem 3.2, we do not assume the objective function $F(w)$ to be convex or Lipschitz, but only smooth with concentrated stochastic gradients (Assumption 2.1).[6] The utility loss is measured by the norm of the gradient $\|\nabla F(w_k)\|$ in Equation (13), commonly considered in non-convex optimization [10], [34]. There are several interesting observations from Theorem 3.2. First, in clipped DP-SGD, we should not simply consider the clipping threshold c as a virtual Lipschitz constant. When the gradient norm $\|\nabla F(w_k)\|$ is large, Equation (13) suggests that

$$\sum_{k=0}^{T-1} \|\nabla F(w_k)\| = O(T \cdot \frac{\sqrt{\log(1/\delta)d}}{n\epsilon}),$$

which is independent of the clipping threshold c as long as c satisfies Equation (12). Therefore, when the objective function is hard to optimize, or when the gradient is large during the initial epochs, we should select a large clipping threshold to minimize the effect of clipping and let the state w_k approach some neighborhood domain of a (local) minimum fast. When $\|\nabla F(w_k)\|$ becomes small, c will reappear in the utility bound. In this case, the equation becomes similar to the classic DP-SGD privacy-utility tradeoff [8] but replacing the Lipschitz constant L by c, where

$$\frac{\sum_{k=0}^{T-1} \|\nabla F(w_k)\|}{T} = O(\frac{c\sqrt{\log(1/\delta)d}}{n\epsilon}).$$

Therefore, we want to select some properly small c.

The second and also more important issue worth mentioning is the restriction on c in Equation (12). We require the norm of the sampling noise in the stochastic gradient to be smaller than the clipping threshold c with sufficient probability. *Indeed, this is actually a necessary condition for clipped DP-SGD to work.* In the following example, we show that even without any noise perturbation, clipped SGD could fail when c is much smaller than the magnitude of sampling noise.

6. In the proof of Theorem 3.2, essentially we only need a high-probability bound of the sampling noise in the stochastic gradient, which is also necessary (see Example 3.1). Therefore, one can relax Assumption 2.1 of a subexponential tail to any proper concentration assumption that allows the derivation of a high probability bound of the sampling noise.

Example 3.1. *Suppose the loss function is $f(w, x) = (w-x)^2/2$ and we have three samples $x_1 = -20, x_2 = -10$ and $x_3 = 90$. The overall loss function is thus $F(w) = 1/3 \cdot \sum_{i=1}^{3}(w - x_i)^2/2$. Thus, $\nabla F(w) = w - 20$ and the minimum is achieved when $w = 20$. Moreover, the standard deviation of the per-sample stochastic gradient, where we randomly sample $x' \leftarrow \{x_1, x_2, x_3\}$ and compute the gradient of $f(w, x')$, is 49.7 for any w.*

We will show that if the clipping threshold is relatively small, for example, $c = 1$, then the clipped gradient is dramatically different from the true gradient $\nabla F(w)$. We first consider the case when the magnitude of true gradient is small, say at the optimum $w = 20$ where $\nabla F(20) = 0$. At $w = 20$, the expectation of the clipped stochastic gradient is $1/3$ rather than 0. On the other hand, when the magnitude of the true gradient is large, say $w = 0$ and $\nabla F(0) = -20$, the expectation of the clipped stochastic gradient is still $1/3$. But now it is in the opposite direction of the true gradient.

From Example 3.1, we know that if c is not properly selected, clipped SGD could fail to converge regardless of the magnitude of the full-batch gradient. As a summary, Theorem 3.2 and Example 3.1 suggest the following two key observations on clipped DP-SGD:

1) On hyper-parameter selection, to ensure stable convergence in general, we need the clipping threshold c to be sufficiently larger than the sampling noise of stochastic gradient.
2) On the other hand, to improve the performance of clipped DP-SGD, one promising direction is to reduce the sampling noise via network architecture optimization and data normalization. Indeed, many approaches supporting this goal are proposed and studied in deep learning research, for example, batchnorm layer [44], where the gradient is evaluated and normalized by a group of samples. We defer a systematical study on improvement via gradient variance reduction to our future work.

4. Privacy Guarantees and Amplifications

4.1. Numerical Calculation and Asymptotic Amplification

In this section, we first show how to numerically compute the (ϵ, δ) bound via RDP and asymptotically analyze the amplification of ModelMix. We use Gaussian Mechanism to produce the noise $\Delta_k \sim \mathcal{N}(0, \sigma^2 \cdot \boldsymbol{I}_d)$ across iterations. The following Theorem 4.1 shows an efficiently-calculable upper bound of (ϵ, δ) by measuring an α-Rényi divergence between two one-dimensional distributions. For simplicity, we fix $\tau_k = \tau$ in the following.

Theorem 4.1 (Privacy Calculation of ModelMix via RDP)**.** *Suppose Δ_k is Gaussian noise generated from $\mathcal{N}(0, \sigma^2 \cdot \boldsymbol{I}_d)$. We define two distributions P_0 and P_1,*

$$\mathsf{P}_0 = \mathcal{N}(0, \sigma) * \mathcal{U}[-\tau/(2\eta), \tau/(2\eta)],$$

$$\mathsf{P}_1 = \mathcal{N}(c, \sigma) * \mathcal{U}[-\tau/(2\eta), \tau/(2\eta)],$$

where $*$ represents the convolution of two probability distributions. In other words, sampling from P_0 is equivalent to sampling from Gaussian distribution $\mathcal{N}(0, \sigma)$ and uniform distribution $\mathcal{U}[-\tau/(2\eta), \tau/(2\eta)]$ independently, then summing the results together. Similarly, P_1 is the convolution of $\mathcal{N}(c, \sigma)$ and $\mathcal{U}[-\tau/(2\eta), \tau/(2\eta)]$.

Algorithm 1 for T iterations with sampling rate q satisfies (ϵ, δ) DP for any ϵ and δ such that

$$\epsilon \leq \min_{\alpha \in \mathbb{Z}, \alpha > 1} T\mathsf{D}_\alpha\big((1-q)\mathsf{P}_0 + q\mathsf{P}_1 \| \mathsf{P}_0\big) + \frac{\log(1/\delta)}{(\alpha - 1)}.$$

Proof. See Appendix C. □

Let $\mathcal{D}$ and $\mathcal{D}'$ be two neighboring datasets, where $(\tilde{x}, \tilde{y}) = \mathcal{D} - \mathcal{D}'$ is the differing element. Theorem 4.1 shows that the worst output divergence between $\mathcal{D}$ and $\mathcal{D}'$ occurs when the gradient $\nabla f(w_k, \tilde{x}, \tilde{y})$ is a vector whose Hamming weight is 1, such as $(c, 0, ..., 0)$. In other words, given an l_2-norm budget of c, the worst-case D_α divergence happens when the gradient of $(\tilde{x}, \tilde{y})$ is concentrated on a single dimension. This is different from standard (subsampled) Gaussian Mechanism, where the randomization is only due to the Gaussian noise and subsampling [42]. We will exploit this property to design more fine-tuned gradient clipping for ModelMix in Section 4.2. In the following, we give an asymptotic analysis on the privacy amplification of ModelMix.

Theorem 4.2 (Asymptotic Privacy Amplification from ModelMix)**.** *If the sampling rate q is some constant, the noise Δ_k is selected to be Gaussian noise $\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_d)$ and τ is sufficiently large, then Algorithm 1 with T iterations satisfies (ϵ, δ)-DP guarantee for any ϵ and δ such that*

$$\epsilon = \tilde{O}\Big(\frac{\eta T(c+\sigma)}{\tau n} + \sqrt{\frac{\eta T \log(1/\tilde{\delta})}{\tau} \cdot \Big(\frac{c^4}{n^4\sigma^3} + \frac{c^2}{n^2\sigma} + \sigma\Big)}\Big),$$
$$\delta = \tilde{O}\big(T \cdot \frac{(L/n) + \sigma}{\tau} + \tilde{\delta}\big).^7 \tag{14}$$

where $\tilde{\delta}$ can be any value within $(0, 1)$.

In addition, if in Algorithm 1, the per-sample gradient is clipped up to c in l_1 norm and the noise Δ_k is selected to be Laplace noise, then Algorithm 1 satisfies (ϵ, δ)-DP, where

$$\epsilon = \tilde{O}\Big(\frac{\eta T \epsilon_0 (e^{\epsilon_0} - 1)}{\tau} + \epsilon_0 \sqrt{\frac{\eta T \log(1/\delta)}{\tau}}\Big), \tag{15}$$

where ϵ_0 represents the ϵ privacy loss of a single iteration of (9) without ModelMix (set $\boldsymbol{\alpha}_k$ to be constant).

Proof. See Appendix D. □

We can compare between Theorem 4.2 and Theorem 2.1 (the case of regular DP-SGD without ModelMix). If a single iteration of a DP-SGD is ϵ_0-differentially-private, then classic advanced composition (Theorem 2.1) suggests that the composition of T iterations produces $O(\epsilon_0\sqrt{T\log(1/\delta)}, \delta)$-DP when ϵ_0 is small. Theorem 4.2 states that under the same setup, if the DP-SGD is further incorporated with ModelMix, then this composition becomes $\tilde{O}(\epsilon_0\sqrt{T\log(1/\delta)/\tau}, \delta)$, where the ϵ term decreases by a factor of $\tilde{O}(1/\sqrt{\tau})$.

Algorithm 2 l_2 and l_∞ Norm Gradient Clipping

Input: Individual gradient $\nabla f(w, x, y)$, l_2 norm clipping threshold c, l_∞ norm truncation parameter $p \in \mathbb{Z}^+$.
Clip $\nabla f(w, x, y) \leftarrow \mathsf{CP}(\nabla f(w, x, y), c)$
for $j = 1, 2, ..., d$ **do**
  if $|\nabla f(w, x, y)(j)| > \frac{c}{\sqrt{p}}$ **then**
    $\nabla f(w, x, y)(j) \leftarrow \text{sign}(\nabla f(w, x, y)(j)) \frac{c}{\sqrt{p}}$
  end if
end for
Output: $\nabla f(w, x, y)$.

4.2. Clipping with l_2 and l_∞ Sensitivity Guarantee

In this subsection, we show that in the framework of ModelMix with Gaussian Mechanism, one can further strengthen the privacy amplification if both the l_2 and l_∞ norm sensitivity can be guaranteed. We first present a gradient clipping algorithm, described in Algorithm 2. Algorithm 2 is a simple generalization of standard l_2-norm clipping operator $\mathsf{CP}(\cdot, c)$. For an individual gradient $\nabla f(w, x, y)$, we first clip the gradient up to c in l_2 norm. After that, we truncate each coordinate such that its absolute value does not exceed $c/\sqrt{p}$ for some constant integer $p \geq 1$. As a consequence, Algorithm 2 ensures that the l_2 and l_∞ norm of clipped gradient is upper bounded by c and $c/\sqrt{p}$, respectively. From our empirical observations, compared to l_2-norm clipping on c, DP-SGD is much less sensitive to l_∞-norm truncation on p. The reason behind this is that in practice the gradients obtained are rarely concentrated on few coordinates and thus l_∞ truncation with large p, even in the hundreds, will hardly change the geometry of gradients. Roughly speaking, p captures the number of significant coordinates in the gradient, where the gradient dimension could be millions in deep learning. However, such truncation enables us to derive a stronger amplification bound with ModelMix, as shown in Corollary 4.1.

Corollary 4.1. *Under the same setup of Theorem 4.1, if we further ensure that the l_∞ norm of each individual gradient in Algorithm 1 is bounded by $c/\sqrt{p}$ for some $p \in \mathbb{Z}^+$, then it satisfies (ϵ, δ) DP for any ϵ and δ such that*

$$\epsilon \leq \min_{\alpha \in \mathbb{Z}, \alpha > 1} \frac{\log\big(\sum_{k=0}^{\alpha} \binom{\alpha}{k}(1-q)^{\alpha-k} q^k \mathcal{A}_k\big) + \log(1/\delta)}{1 - \alpha},$$

where $\mathcal{A}_k = \big(\mathbb{E}_{z \sim \mathsf{P}_0}\big[(\mathsf{P}_1'(z)/\mathsf{P}_0(z))^k\big]\big)^p$. Here, $\mathsf{P}_0 = \mathcal{N}(0, \sigma) * \mathcal{U}[-\tau/(2\eta), \tau/(2\eta)]$ and $\mathsf{P}_1' = \mathcal{N}(c/\sqrt{p}, \sigma) * \mathcal{U}[-\tau/(2\eta), \tau/(2\eta)]$, i.e., P_0 shifted by $c/\sqrt{p}$.

Proof. See Appendix C. □

Corollary 4.1 generalizes Theorem 4.1 and recomputes the worst case divergence when we have both l_2 and l_∞ norm sensitivity guarantees. We prove that the worst case happens when the gradient $\nabla f(w_k, \tilde{x}, \tilde{y})$ of the differing element $(\tilde{x}, \tilde{y})$ is in a form whose Hamming weight is p and each non-zero coordinate is equal to $\pm c/\sqrt{p}$. Corollary 4.1 also shows an efficient way to numerically compute

the privacy loss by measuring the divergence between two one-dimensional distributions.

4.3. Privacy Amplification Examples

In Fig. 3, we provide concrete examples of DP-SGD with ModelMix under various setups. We set $n = 50,000$, $\delta = 10^{-5}$, the clipping threshold $c = 20$, and run DP-SGD under the Gaussian Mechanism [18]. We measure the cumulative privacy loss in terms of ϵ under various sampling rates q, mixing thresholds τ and l_∞ truncation parameter p. In all subfigures, we set the total number of iterations $T = 5,000$, and the budget $\epsilon = 200$ for regular DP-SGD.

In Fig. 3 (a-c), we set the sampling rate $q = 0.02$ (a minibatch of size 1000 in expectation), and examine the privacy loss under $\tau = 0.075\eta, 0.15\eta, 0.3\eta$, where η is the stepsize, respectively. With ModelMix, especially with further help of Algorithm 2 to ensure both l_2 and l_∞ norm sensitivity, we achieve orders of magnitude improvement. For example, in Fig. 3 (c) where $\tau = 0.3\eta$, compared to $(\epsilon = 200, \delta = 10^{-5})$ using regular DP-SGD, under the same setup, ModelMix, ModelMix with further l_∞ truncation of $p = 25$ and $p = 100$ produce $\epsilon = 31.7, 5.4$ and 4.8, respectively, with the same $\delta = 10^{-5}$. The corresponding ϵ numbers produced in Fig. 3 (a) and (b) are (57.2, 17.9, 15.3) and (40.4, 9.0, 7.9), repectively. The amplification factor of ModelMix matches our theoretical results, which is $\tilde{O}(1/\sqrt{\tau})$. On the other hand, the additional amplification from the l_∞ norm truncation will reach some limit as p increases. For large enough p, such combined amplification is empirically $O(1/\tau)$. In Fig. 3 (e-f), we change the sampling rate q to be 0.04 while keeping the remaining parameters the same as those in Fig. 3 (a-c). In general, given larger privacy budget and mixing threshold τ, ModelMix can produce stronger privacy amplification.

4.4. More Insights on Privacy Amplification

In the following, we provide more technical insights on how ModelMix improves the privacy. A quick intuition is that, compared to standard DP-SGD (Equation (6)) where w_k is only randomized by the noise Δ_k, the sources of randomness in Algorithm 1 are enriched, containing both the randomness in ModelMix and the noise perturbation. More randomness often implies stronger privacy. However, *though privacy relies on randomness, not all kinds of randomness can produce worst-case DP guarantees.* ModelMix is an interesting example. ModelMix must be applied with other noise mechanisms to produce meaningful DP guarantees, as explained below.

In ModelMix, the mixed state is within a bounded hull determined by earlier updates. This means that the randomness is bounded and localized, which cannot produce reasonable worst-case DP guarantees, in contrast to Laplace/Gaussian noise. To better illustrate this, consider the following example. Suppose $\nabla F(w_k, \mathcal{D}) = 0$ and $\nabla F(w_k, \mathcal{D}') = 1$ are two gradients evaluated by adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$. We randomize the gradients by introducing $\mathcal{U}[0, 5]$, the uniform

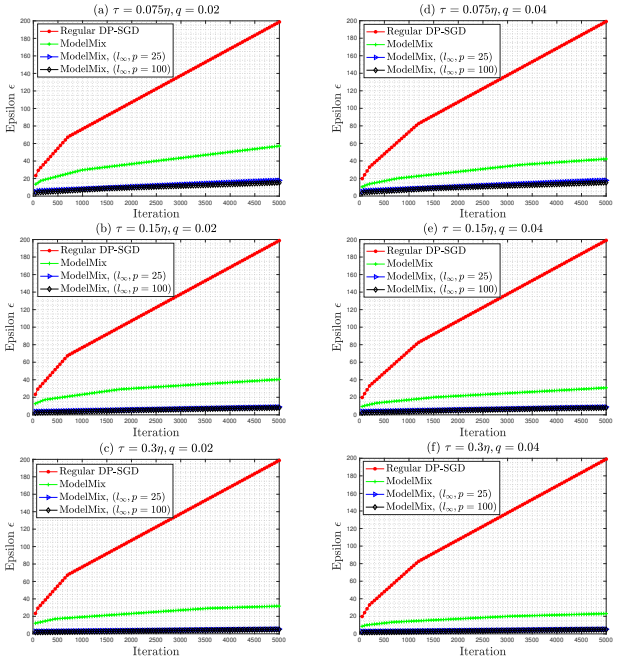

Figure 3: Privacy amplification from ModelMix on running DP-SGD with $n = 50K$ samples under Gaussian Mechanism.

distribution between $[0, 5]$. The perturbed gradients now become equivalent to being uniformly sampled from $\mathcal{U}[0, 5]$ and $\mathcal{U}[1, 6]$ respectively. When we compare the divergence between the two distributions $\mathcal{U}[0, 5]$ and $\mathcal{U}[1, 6]$, we cannot derive a worst-case ϵ bound ($\epsilon = \infty$ in this example) or an efficiently-composable (ϵ, δ) DP guarantee [8]. To this end, additional Laplace or Gaussian noise is needed.

However, even with additional noise, we cannot improve the privacy loss of a single iteration significantly. Continuing with the above example, in a single iteration of DP-SGD, if we add some Laplace noise $Lap(0, \lambda)$, which ensures $(\epsilon_0, 0)$-DP, and further perturb the update by a uniform noise within $\mathcal{U}(0, 5)$, the distributions of the two perturbed gradients $\nabla F(w_k, \mathcal{D}) = 0$ and $\nabla F(w_k, \mathcal{D}') = 1$ now become $Lap(0, \lambda) * \mathcal{U}(0, 5)$ and $Lap(0, \lambda) * \mathcal{U}(1, 6)$, respectively. Here, $*$ represents the convolution of two distributions. Still, the worst-case is not improved where we can only claim ϵ_0-DP or still some not efficiently-composable (ϵ, δ)-DP for a single iteration.

So how does the localized randomness in ModelMix amplify the privacy? While ModelMix cannot significantly improve the worst-case utility-privacy tradeoff in a single iteration [8], as shown in Theorems 4.1 and 4.2, it smoothens the divergence between the output distributions from two arbitrary adjacent datasets. *Such an amplification is limited in a single iteration, but when we consider the composition of the privacy loss, the amplification accumulates to produce a strong improvement.* To be specific, for the DP-SGD update protocol (Equation (6)), denoted by $\mathcal{M}$, we consider the

8. To achieve a meaningful privacy guarantee in DP-SGD for T iterations, we need the failure probability $\delta = o(1/(nT))$ in a single iteration.

pointwise $\epsilon(w)$*-loss* at a particular output w [9],

$$\epsilon_{\mathcal{D},\mathcal{D}',\mathsf{Aux}}(w) = \log \frac{\mathbb{P}(\mathcal{M}(\mathcal{D},\mathsf{Aux}) = w)}{\mathbb{P}(\mathcal{M}(\mathcal{D}',\mathsf{Aux}) = w)},$$

where $\mathcal{D}, \mathcal{D}'$ are two adjacent datasets and Aux represents the other auxiliary inputs used in $\mathcal{M}$. When we take the supremum $\epsilon = \sup_{\mathcal{D},\mathcal{D}',\mathsf{Aux}} \epsilon_{\mathcal{D},\mathcal{D}',\mathsf{Aux}}(w)$, it gives an equivalent ϵ-DP guarantee. As explained before, incorporation of ModelMix (Equation (9)) cannot improve this (supremum) worst-case loss. What we proved in Theorem 4.2 is that, for any two datasets $\mathcal{D}, \mathcal{D}'$ and Aux, the expectation $\mathbb{E}[\epsilon(w)]$ and the variance of $\mathrm{Var}[\epsilon(w)]$ for $w \sim \mathcal{M}(\mathcal{D}, \mathsf{Aux})$ will scale by $\tilde{O}(1/\tau)$ when ModelMix is further applied. Provided such an $\epsilon(w)$ loss of smaller mean and stronger concentration, we can derive a stronger high probability bound in a composite (ϵ, δ) form when we measure the cumulative ϵ loss across the states $w_{[1:T]}$ from T iterations.

4.5. Randomization beyond Noise

The analysis framework of ModelMix also sheds light on how to quantify the privacy amplification from a large class of "training-oriented" randomization commonly applied in deep learning. There are many reasons to introduce randomness in optimization and learning other than privacy preservation. For example, in stochastic gradient Langevin dynamics (SGLD) [30], [45] for nonconvex optimization, it is common to utilize noisy gradient descent to escape saddle points [46]. Randomization can also strengthen the learning performance, e.g., random dropout [47] and data augmentation [29]. In particular, data augmentation plays an important role in modern computer vision. Generally speaking, data augmentation represents a large class of methods to improve robustness and reduce memorization (instead of generalization) by generating virtual samples through random cropping [48], erasing [49] or mixing [50] the raw samples.

All of the above-mentioned randomnesses are localized, which cannot produce meaningful DP guarantees for the same reason as ModelMix. For example, consider random erasing [49], where a rectangle region of an image is randomly erased and replaced with random values. When we process private images using the above mechanisms, the random transformation is multiplicative over the private input, meaning that the output is restricted to a bounded domain determined by the specific input processed. Given two different images with at least one differentiating pixel, one can still distinguish them after random erasing as long as at least one differentiating pixel is not erased. A similar argument holds for dropout, where a node in a neural network is ignored independently with some fixed rate, and the saddle point escaping algorithm [46], where the gradient is perturbed by a bounded noise uniformly selected from a sphere. Our privacy analysis framework shows a way to analyze these randomizations in practical learning algorithms combined with DP-SGD to produce sharpened composition bounds.

5. Further Experiments

ModelMix is a generic technique, which can be implemented in almost all applications of DP-SGD without further assumptions. In this section, we provide further experiments to measure its performance combined with other state-of-the-art advances in DP-SGD. Below, when we report the improvement of ModelMix over existing works, we mostly follow the optimal hyper-parameters each work suggests. We repeat each simulation five times and report the median of the results. Our code can be found in the attached anonymous Github link [9], which is mainly built upon those provided in [19] and [32]. Details of the hyper-parameter selections we used can be found in Appendix E. Based on the experiments conducted by previous works, to have a clear comparison, we will also test the proposed algorithms on the following three benchmark datasets, CIFAR10, SVHN and FMNIST[10]. CIFAR10 consists of 60,000 color images in 10 classes, where 50,000 are for training and 10,000 for test. The Street View House Numbers (SVHN) dataset has 73,257 images of real world house digits for training and 26,032 for test.[11] Fashion MNIST (FMNIST) contains 70,000 greyscale images of fashion products from 10 categories with 60,000 for training and 10,000 for test. In the following experiments, we will assume that all the training samples in the above-mentioned datasets are private.

5.1. Shallow Network

Since the magnitude of gradient perturbation is adversely dependent on the model size, instead of using the cutting-edge deep models, a large number of works are devoted to building small models to carefully balance model capacity and utility loss caused by DP-SGD [9], [16], [19], [25]. One of the best existing results is given by [19]. In [19], Tramer and Boneh showed that handcrafted features extracted from raw data can significantly strengthen the performance of training shallow models with DP-SGD. They proposed to first privately use ScatterNet [51] to process the dataset and to then apply DP-SGD on the ScatterNet features. With a DP budget of $(\epsilon = 3, \delta = 10^{-5})$, [19] successfully trained a five-layer CNN with ScatterNet, which achieves 66.9% and 87.2% accuracy on CIFAR10 and FMNIST, respectively. In the following, we consider further improving their results by using ModelMix. With the same setup, we spend $\epsilon = 0.695$ budget to privately estimate the statistics of datasets to apply ScatterNet. In Table 1, we compare the utility-privacy tradeoff when we run DP-SGD with/without ModelMix on training the same CNN suggested by [19] on the ScatterNet features, where δ is always fixed to be 10^{-5} in all cases. Table 1 shows that ModelMix can bring significant improvement even in simple model training with very low privacy budget.

9. https://anonymous.4open.science/r/ModelMix-and-BatchClipping-A7CD

10. https://deepobs.readthedocs.io/en/stable/api/datasets/fmnist.html.

11. We do not use the 600,000 auxiliary samples provided in SVHN in our experiments.

CIFAR10						
Method\Privacy	$\epsilon = 0.2$	0.4	0.6	0.8	1.0	∞
[19]	32.8	45.9	56.1	61.2	64.1	74.6
DP-SGD +MM	58.5	63.3	65.0	66.9	68.3	74.6
FMNIST						
Method\Privacy	$\epsilon = 0.2$	0.4	0.6	0.8	1.0	∞
[19]	53.4	74.0	81.5	84.5	85.6	91.2
DP-SGD+MM	83.9	85.7	86.1	87.9	88.8	91.2

TABLE 1: **Test Accuracy of DP-SGD with/out ModelMix (MM) on training small CNN with ScatterNet Features.**

CIFAR10						
Method\Privacy	$\epsilon = 4$	5	6	7	8	∞
Regular DP-SGD	47.3	51.1	53.2	54.7	56.1	90.7
DP-SGD+MM	59.8	64.6	67.5	69.3	70.4	90.7
SVHN						
Method\Privacy	$\epsilon = 2$	4	5	6	8	∞
Regular DP-SGD	41.2	67.5	78.4	81.9	84.9	92.3
DP-SGD+MM	78.2	86.3	87.8	89.0	90.1	92.3

TABLE 2: **Test Accuracy of DP-SGD with/out ModelMix (MM) on training Resnet20**.

5.2. Deep Models

In this subsection, we consider applying both ModelMix to help train large neural networks with DP-SGD. We take CIFAR10 and SVHN as examples. We will further apply Algorithm 2 to enhance privacy amplification. In particular, according to our analysis on the clipping threshold and sampling noise, we select $c = 20$ and the l_∞-norm truncation parameter $p = 100$ in all experiments. In Table 2, we show the tradeoff between privacy and learning accuracy when training Resnet20 on CIFAR10 and SVHN with DP-SGD, respectively. Still, in all the cases, δ is fixed to be 10^{-5}. With an $(\epsilon = 8, \delta = 10^{-5})$-DP budget, combined with ModelMix, we train Resnet20 which achieves 70.4% and 90.1% accuracy on CIFAR10 and SVHN, respectively. In comparison, regular DP-SGD can only produce 56.1% and 84.9% accuracy. In Table 3, we also include results for particular test accuracy the privacy budget required. In general, ModelMix roughly brings $20\times$ improvement on ϵ to produce usable security parameters.

5.3. Assistance with Public Data

With access to additional public data, many elegant ideas have been proposed to significantly improve the performance of DP-SGD. In this subsection, we present results where we use ModelMix to further improve two representative works [19] and [32]. [19] shows a private transfer learning method on CIFAR10, where a SimCLR model [52] is first pretrained on unlabeled ImageNet and a linear model is then trained on the features extracted from the penultimate layer of the SimCLR model using DP-SGD. With a privacy budget $(\epsilon = 2, \delta = 10^{-5})$, one can achieve 92.7% on CIFAR10 [19]. Similarly, we can apply ModelMix to improve this DP guarantee to $(\epsilon = 0.64, \delta = 10^{-5})$ with the same performance. On the other hand, if we assume the same privacy budget $(\epsilon = 2, \delta = 10^{-5})$ as [19], we can instead achieve 93.6%, close to the non-private optimal performance 94.3%.

CIFAR10					
Method\Accuracy(%)	64	66	68	70	72
Regular DP-SGD	$\epsilon = 95.8$	108.3	118.2	140.6	182.9
DP-SGD+MM	$\epsilon = 4.7$	5.3	6.3	7.5	9.2
SVHN					
Method\Accuracy(%)	88	88.5	89	89.5	90.1
Regular DP-SGD	$\epsilon = 93.6$	101.3	112.4	128.9	144.5
DP-SGD+MM	$\epsilon = 5.1$	5.6	6.0	6.8	8

TABLE 3: **Privacy Loss of DP-SGD with/out ModelMix (MM) on training Resnet20**.

We also compare to [32], which takes 2,000 ImageNet samples as public data and estimates a low-dimensional embedding of the private gradient when applying DP-SGD. Using the hyper-parameters suggested by [32], we reproduce their experiments to privately train Resnet20 on CIFAR10, which achieves 73.2% accuracy with an $(\epsilon = 8, \delta = 10^{-5})$ budget, and 79.1% with $(\epsilon = 111.2, \delta = 10^{-5})$. With ModelMix, a sharpened tradeoff is produced, where with a budget $(\epsilon = 2.9, \delta = 10^{-5})$ and $(\epsilon = 6.1, \delta = 10^{-5})$, we achieve accuracy of 74.2% and 79.1%, respectively. This is also close to the non-private optimal performance 82.3% of [32].

6. Conclusion and Prospects

In this paper, we present a formal study on the privacy amplification from the trajectory entropy and the influence of gradient clipping in DP-SGD. We show fundamental improvements over DP-SGD, especially for deep learning, without assistance of other assumptions and additional data resources. This is a first step to consider the amplification from the potential entropy underlying the intermediate computation in DP-SGD and there are still many interesting directions for further generalization of ModelMix. For example, one may consider an adaptive selection of the envelope radius τ using techniques in [27] or approximate it with public data after a projection to a low-rank subspace [31]. Moreover, the privacy analysis presented can also be generalized to quantify amplification from a large class of practical randomness used in deep learning, as explained in Section 4.5; or to formalize the privacy guarantees of many heuristic privacy protections, such as Instahide [53] and Datamix [54]. Another issue we pointed out, which is of more interest to practitioners, is the important role of sampling noise in clipped DP-SGD and the connection to the clipping threshold. The theory and the empirical study shown here could be meaningful to instruct further architecture-level improvement, for example the application of batch norm for the noise variation reduction. We leave this to future work.

References

[1] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[2] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[3] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32:14774–14784, 2019.

[4] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33:22205–22216, 2020.

[5] 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.

[6] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[7] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[8] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

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

[10] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2722–2731, 2017.

[11] Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. On differentially private stochastic convex optimization with heavy-tailed data. In *International Conference on Machine Learning*, pages 10081–10091. PMLR, 2020.

[12] Lijie Hu, Shuo Ni, Hanshen Xiao, and Di Wang. High dimensional differentially private stochastic optimization with heavy-tailed data. *ACM SIGMOD PODS*, 2021.

[13] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 521–532. IEEE, 2018.

[14] Borja Balle, Peter Kairouz, Brendan McMahan, Om Thakkar, and Abhradeep Guha Thakurta. Privacy amplification via random check-ins. *Advances in Neural Information Processing Systems*, 33:4623–4634, 2020.

[15] Alexey Kurakin, Steve Chien, Shuang Song, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022.

[16] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9312–9321, 2021.

[17] Shuang Song, Om Thakkar, and Abhradeep Thakurta. Characterizing private clipped gradient descent on convex generalized linear problems. *arXiv preprint arXiv:2006.06783*, 2020.

[18] 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.

[19] Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2020.

[20] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papemoti, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 866–882. IEEE, 2021.

[21] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. Membership inference attack against differentially private deep learning model. *Trans. Data Priv.*, 11(1):61–79, 2018.

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

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

[24] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[25] Nicolas Papernot, Steve Chien, Shuang Song, Abhradeep Thakurta, and Ulfar Erlingsson. Making the shoe fit: Architectures, initializations, and tuning for learning with privacy. 2019.

[26] Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 298–309, 2019.

[27] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34, 2021.

[28] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

[29] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

[30] Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. *Proceedings of Machine Learning Research vol*, 75:1–34, 2018.

[31] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning*, pages 12208–12218. PMLR, 2021.

[32] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2020.

[33] Zelun Luo, Daniel J Wu, Ehsan Adeli, and Li Fei-Fei. Scalable differential privacy with sparse network finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5059–5068, 2021.

[34] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2019.

[35] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33:13773–13782, 2020.

[36] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11854–11862, 2020.

[37] Yingxue Zhou, Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private sgd with gradient subspace identification. In *International Conference on Learning Representations*, 2020.

[38] Hilal Asi, John Duchi, Alireza Fallah, Omid Javidbakht, and Kunal Talwar. Private adaptive gradient methods for convex optimization. In *International Conference on Machine Learning*, pages 383–392. PMLR, 2021.

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

[40] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[41] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, pages 6277–6287, 2018.

[42] Ilya Mironov, Kunal Talwar, and Li Zhang. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[43] 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.

[44] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

[45] Bai Li, Changyou Chen, Hao Liu, and Lawrence Carin. On connecting stochastic gradient mcmc and differential privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 557–566, 2019.

[46] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1724–1732. JMLR. org, 2017.

[47] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[49] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020.

[50] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[51] Edouard Oyallon, Sergey Zagoruyko, Gabriel Huang, Nikos Komodakis, Simon Lacoste-Julien, Matthew Blaschko, and Eugene Belilovsky. Scattering networks for hybrid representation learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2208–2221, 2018.

[52] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[53] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International Conference on Machine Learning*, pages 4507–4518. PMLR, 2020.

[54] Zhijian Liu, Zhanghao Wu, Chuang Gan, Ligeng Zhu, and Song Han. Datamix: Efficient privacy-preserving edge-cloud inference. In *Computer Vision – ECCV 2020*, pages 578–595. Springer International Publishing, 2020.

[55] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Nonconvex finite-sum optimization via scsg methods. *Advances in Neural Information Processing Systems*, 30, 2017.

[56] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

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

Appendix A. Proof of Theorem 3.1

For a β-smooth function $F(w)$, we have the following fact [55] that for any w and w',

$$-\frac{\beta}{2}\|w-w'\|^2 \leq F(w)-F(w')-\langle\nabla F(w'), w-w'\rangle \leq \frac{\beta}{2}\|w-w'\|^2. \tag{16}$$

Equation (16) will be constantly applied in the following proof. It is noted that $\mathbb{E}[G_k] = nq\nabla F(w_k)$ and we use η' to denote $\eta\cdot(nq)$ in the following. Thus, $\mathbb{E}[\eta G_k] = \eta'\nabla F(w_k)$.

For each $k \in [0:T-1]$, conditional on w_k and w_{k-1}, with the updating rule defined in Algorithm 1, we have

$$\begin{aligned}
&\mathbb{E}\big[\|w_{k+1}-w^*\|^2\big]\\
=&\mathbb{E}\big[\|\alpha_{k+1}w_k+(1-\alpha_{k+1})w_{k-1}-w^*-\eta(G_k+\Delta_{k+1})+e_{\tau_{k+1}}\|^2\big]\\
=&\mathbb{E}\big[\|\alpha_{k+1}(w_k-w^*)+(1-\alpha_{k+1})(w_{k-1}-w^*)\|^2\\
&\qquad -2\eta'\langle\frac{w_k+w_{k-1}}{2}-w^*,\nabla F(w_k)\rangle\rangle\\
&\qquad +\frac{\eta'^2}{n^2q^2}\big(\mathbb{E}[\|G_k\|^2]+\mathbb{E}[\|\Delta_{k+1}\|^2]\big)+\mathbb{E}[\|e_{\tau_{k+1}}\|^2].
\end{aligned} \tag{17}$$

Here, $e_{\tau_{k+1}}$ represents the additional bias caused by the modification which ensures the per coordinate distance between w_k and w_{k-1} is at least τ_{k+1}, whose mean is zero. Therefore, the variance $\mathbb{E}[\|e_{\tau_{k+1}}\|^2]$ of $e_{\tau_{k+1}}$ is bounded by $d\tau^2/12$. Furthermore, we have that for each coordinate

$$\begin{aligned}
&\|\alpha_{k+1}(j)(w_k(j)-w^*(j))+(1-\alpha_{k+1}(j))(w_{k-1}(j)-w^*(j))\|^2\\
\leq&\alpha_{k+1}(j)\|w_k(j)-w^*(j)\|^2+(1-\alpha_{k+1}(j))\|w_{k-1}(j)-w^*(j)\|^2,
\end{aligned}$$

and thus,

$$\begin{aligned}
&\mathbb{E}[\|\alpha_{k+1}(w_k-w^*)+(1-\alpha_{k+1})(w_{k-1}-w^*)\|^2]\\
\leq&\frac{\|w_k-w^*\|^2+\|w_{k-1}-w^*\|^2}{2}.
\end{aligned}$$

In addition, due to the convexity and the smoothness assumption,

$$\begin{aligned}
&\langle\frac{w_k+w_{k-1}}{2}-w^*,\nabla F(w_k)\rangle\\
=&\langle w_k-w^*+\frac{w_{k-1}-w_k}{2},\nabla F(w_k)-\nabla F(w^*)\rangle\\
\geq&F(w_k)-F(w^*)+\langle\frac{w_{k-1}-w_k}{2},\nabla F(w_k)-\nabla F(w^*)\rangle\\
\geq&\frac{F(w_k)+F(w_{k-1})}{2}-F(w^*)-\frac{\beta}{4}\|w_k-w_{k-1}\|^2.
\end{aligned} \tag{18}$$

In (18), we use (16) and the following fact that $\langle w-w',\nabla F(w)-\nabla F(w')\rangle \geq F(w)-F(w')$ for any w, w' and a convex function $F(\cdot)$. With the Lipschitz assumption, we know $\|G_k\|^2 \leq n^2L^2$. Therefore,

$$\begin{aligned}
&\mathbb{E}\big[\frac{F(w_k)+F(w_{k-1})}{2}-F(w^*)\big]\\
\leq&\frac{1}{2\eta'}\big(\frac{\|w_k-w^*\|^2+\|w_{k-1}-w^*\|^2}{2}-\|w_{k+1}-w^*\|^2+\frac{d\tau_{k+1}^2}{12}\big)\\
&+\frac{\eta'}{2n^2q^2}(n^2L^2+\mathbb{E}[\|\Delta_{k+1}\|^2])+\frac{\beta}{4}\|w_k-w_{k-1}\|^2.
\end{aligned} \tag{19}$$

Summing up both sides of Equation (19) from $k=0,1,...,T-1$, we take expectation across $w_{[-1:T]}$ and have

$$\begin{aligned}
&2\mathbb{E}\big[F(\frac{\sum_{k=1}^T w_{k-1}+w_{k-2}}{2T})-F(w^*)\big]\\
\leq&\frac{2}{T}\cdot\sum_{k=1}^T\mathbb{E}\big[\frac{F(w_{k-1})+F(w_{k-2})}{2}-F(w^*)\big]\\
\leq&\eta'^{-1}\big(\frac{\|w_{-1}-w^*\|^2+2\|w_0-w^*\|^2}{2T}+\frac{\sum_{k=0}^{T-1}d\tau_{k+1}^2}{12T}\big)+\frac{\eta'L^2}{q^2}\\
&+\frac{\eta'}{n^2q^2}(\mathbb{E}[\sum_{k=0}^{T-1}\|\Delta_{k+1}\|^2/T])+\frac{\beta\mathbb{E}[\sum_{k=0}^{T-1}\|w_k-w_{k-1}\|^2]}{2T}.
\end{aligned}$$

It is noted that the updating rule can be written as,

$$\begin{aligned} w_{k+1} - w_k &= -(1-\alpha_{k+1})(w_k - w_{k-1}) - \eta(G_k + \Delta_{k+1}) \\ &= (1-\alpha_{k+1})(1-\alpha_k)(w_{k-1} - w_{k-2}) + \\ &\quad (1-\alpha_{k+1})\eta(G_{k-1} + \Delta_k) - \eta(G_k + \Delta_{k+1}) \end{aligned} \tag{20}$$

By the recursion and the independce of different α_k, we have a closed-form upper bound of $\|w_{k+1} - w_k\|^2$ conditional on the initialization $\|w_0 - w_{-1}\|^2$, where

$$\begin{aligned} &\mathbb{E}[\|w_{k+1} - w_k\|^2] \\ &\leq (\frac{1}{3})^{k+1}\|w_0 - w_{-1}\|^2 + \frac{11}{2}\eta^2(n^2L^2 + \mathbb{E}[\|\Delta\|^2]) \\ &\quad + 2(k+1)(1/2)^{k+1}\eta\|w_0 - w_{-1}\|(nL + \mathbb{E}[\|\Delta\|]). \end{aligned} \tag{21}$$

Here, in (21), we use the following fact that

$$\mathbb{E}[\|\sum_{i=1}^{k}\prod_{j=1}^{i-1}\alpha_j\|^2] = \sum_{i=1}^{k}(\frac{1}{3})^i + 2\sum_{i=1}^{k}\sum_{j=1}^{i}(\frac{1}{2})^j(\frac{1}{3})^{i-j} \leq \frac{3}{2} + 4 = \frac{11}{2},$$

Now putting all the above together, we have that $\mathbb{E}\big[F(\frac{\sum_{k=1}^{T} w_{k-1}+w_{k-2}}{2T}) - F(w^*)\big]$ is upper bounded by

$$\begin{aligned} &\frac{\|w_{-1} - w^*\|^2 + 2\|w_0 - w^*\|^2 + \sum_{k=1}^{T} d\tau_i^2/12}{2\gamma\sqrt{T}} \\ &+ \frac{\gamma(L^2/q^2 + \mathbb{E}[\|\Delta\|^2]/(n^2q^2))}{2\sqrt{T}} + \frac{\beta\big(\frac{11}{2}\eta^2(n^2L^2 + \mathbb{E}[\|\Delta\|^2])\big)}{T} \\ &+ \frac{\beta\big(\frac{3}{2}\|w_0 - w_{-1}\|^2 + 4\eta\|w_0 - w_{-1}\|(nL + \mathbb{E}[\|\Delta\|])}{4T} \\ &= \frac{\|w_{-1} - w^*\|^2 + 2\|w_0 - w^*\|^2 + \sum_{k=1}^{T} d\tau_i^2/12}{2\gamma\sqrt{T}} \\ &\quad + \frac{3\beta\|w_0 - w_{-1}\|^2 + 11\gamma^2(\frac{L^2}{q^2} + \mathbb{E}[\frac{\|\Delta\|^2}{n^2q^2}])}{8T} \\ &\quad + \frac{\beta\gamma\|w_0 - w_{-1}\|(\frac{L}{q} + \mathbb{E}[\frac{\|\Delta\|}{nq}])}{T^{3/2}} + \frac{\gamma(L^2 + \mathbb{E}[\|\Delta\|^2])}{2\sqrt{T}}. \end{aligned}$$

Appendix B. Proof of Theorem 3.2

We first prove the following theorem, which analyzes the standard clipped DP-SGD without ModelMix.

Theorem B.1 (Convergence of Clipped DP-SGD)**.** *Suppose the objective loss function $F(w)$ is β-smooth and satisfies Assumption 2.1, then there exists some constant $\psi > 0$ such that when the clipping threshold c satisfies*

$$c \geq \max\{4\kappa\log(10), -\psi\kappa\log(\kappa)\log(\frac{\sqrt{d\log(1/\delta)}}{n\epsilon})\}, \tag{22}$$

then the convergence rate of per-sample clipped DP-SGD with an (ϵ, δ)-DP guarantee satisfies

$$\begin{aligned} &\mathbb{E}\Big[\frac{\sum_{k=0}^{T-1}\min\big\{9/20\cdot\|\nabla F(w_k)\|^2, c/20\cdot\|\nabla F(w_k)\|\big\}}{T}\Big] \\ &\leq (\frac{v}{2} + \frac{5}{2})\cdot\frac{c\sqrt{\mathcal{R}_F\beta d\log(1/\delta)}}{n\epsilon}, \end{aligned} \tag{23}$$

where $\mathcal{R}_F = \sup_w F(w) - \inf_w F(w)$ and v is some constant determined by noise Mechanism.

Proof. At the k-th iteration, based on the smoothness assumption, we have

$$F(w_{k+1}) \leq F(w_k) + \langle\nabla F(w_k), w_{k+1} - w_k\rangle + \frac{\beta}{2}\cdot\|w_{k+1} - w_k\|^2. \tag{24}$$

We consider the following equivalent updating rule, where the step size η is scaled by $1/(nq)$ in (6),

$$w_{k+1} - w_k = \eta\cdot\Big(\frac{1}{nq}\cdot(\sum_{j=1}^{B_k}\mathsf{CP}(g_k^j, c) + \Delta_{k+1})\Big).$$

Here $g_k^j = \nabla f(w_k, x_{(j)}, y_{(j)})$, where $(x_{(j)}, y_{(j)})$ is the j-th sample selected in the minibatch S_k. For simplicity, we use g_k to denote the random variable

$$g_k = \nabla f(w_k, x, y),$$

where (x, y) is randomly selected from the dataset $\mathcal{D}$. Thus, we have the following observation that

$$\mathbb{E}[\mathsf{CP}(g_k, c)] = \mathbb{E}[\frac{1}{nq}\cdot\sum_{j=1}^{B_k}\mathsf{CP}(g_k^j, c)],$$

due to the i.i.d. sampling of rate q. Thus, with expectation we have

$$\mathbb{E}[w_{k+1} - w_k] = \eta\Big(\frac{1}{B_k}\sum_{j=1}^{B_k}\mathbb{E}[\mathsf{CP}(g_k^j, c)] + \mathbb{E}[\Delta_{k+1}]\Big) = \eta\mathbb{E}[\mathsf{CP}(g_k, c)],$$

$$\mathbb{E}[\|w_{k+1} - w_k\|^2] \leq \eta^2\cdot(c^2/q^2 + \mathbb{E}[\|\Delta_{k+1}\|^2]/(n^2q^2)).$$

Taking this into Equation (24), we have

$$\begin{aligned} \mathbb{E}[F(w_{k+1})] &\leq F(w_k) - \mathbb{E}[\langle\nabla F(w_k), w_{k+1} - w_k\rangle] \\ &\quad + \frac{\beta}{2}\cdot\mathbb{E}[\|w_{k+1} - w_k\|^2] \\ &\leq F(w_k) - \eta\cdot\mathbb{E}[\langle\nabla F(w_k), \mathsf{CP}(g_k, c)\rangle] \\ &\quad + \frac{\beta\eta^2}{2}\cdot(c^2/q^2 + \mathbb{E}[\|\Delta_{k+1}\|^2/(n^2q^2)]). \end{aligned} \tag{25}$$

Let $\gamma_k = \min\{1, \frac{c}{\|g_k\|}\}$, i.e., $\mathsf{CP}(g_k, c) = \gamma_k g_k$. From previous works [9] on the standard Gaussian DP mechanism, we know that the variance of the noise Δ_k is bounded by $O(q^2dT\log(1/\delta)/\epsilon^2)$. Therefore, there must exist some constant v such that we can rewrite Equation (25) as

$$\begin{aligned} \mathbb{E}[F(w_{k+1})] &\leq F(w_k) - \eta\cdot\mathbb{E}[\langle\nabla F(w_k), \gamma_k g_k\rangle] \\ &\quad + \frac{\beta\eta^2}{2}\cdot(\mathbb{E}[\|\gamma_k g_k\|^2] + \mathbb{E}[\|\Delta_{k+1}]\|^2) \\ &\leq F(w_k) - \eta\cdot\mathbb{E}[\langle\nabla F(w_k), \gamma_k g_k\rangle] \\ &\quad + \frac{\beta\eta^2}{2}\cdot(c^2/q^2 + v\cdot\frac{dT\log(1/\delta)}{n^2\epsilon^2}). \end{aligned} \tag{26}$$

In the following Lemma, we lower bound $\mathbb{E}[\langle\nabla F(w_k), \gamma_k g_k\rangle]$. We use $\xi_k = g_k - \nabla F(x_k)$ to denote the stochastic gradient noise.

Lemma B.1. *For any $p > 2$ and $c_0 > 0$ such that $c = p\cdot c_0$, it holds that*

$$\mathbb{E}[\langle\nabla F(w_k), \gamma_k g_k\rangle] \geq \min\{a_1\|\nabla F(x_k)\|^2 - a_2, a_3\|\nabla F(x_k)\|\},$$

where a_1, a_2 and a_3 are constants define as below

$$a_1 = \frac{(1 - e^{-\frac{c_0}{\kappa}})}{2}, a_2 = \Big(\frac{(c_0 + \kappa)\cdot e^{-\frac{c_0}{\kappa}}}{1 - e^{-\frac{c_0}{\kappa}}}\Big)^2,$$

$$a_3 = c\Big((1 - e^{-\frac{c_0}{\kappa}})(1 - \frac{1}{p-1} - \frac{1}{p-2}) - e^{-\frac{c_0}{\kappa}}\Big).$$

Proof. The proof contains two parts.

(1). First, we consider the case when the gradient is small where $\|\nabla F(x_k)\| \leq (p-1)c_0$.

$$\begin{aligned}
&\mathbb{E}[\langle \nabla F(w_k), \gamma_k g_k\rangle] \\
&= \mathbb{E}[\langle \nabla F(w_k), \gamma_k(\nabla F(w_k) + \xi_k)\rangle] \\
&= \|\nabla F(w_k)\|^2 \cdot \mathbb{E}[\gamma_k] + \mathbb{E}[\langle \nabla F(w_k), \xi_k\rangle \cdot (1-\gamma_k)] \\
&\geq \|\nabla F(w_k)\|^2 \Pr(\|g_k\| \leq c) - \|\nabla F(w_k)\| \mathbb{E}[\|\xi_k\| \cdot \mathbf{1}_{\|g_k\|>c}] \\
&\geq \|\nabla F(w_k)\|^2 \Pr(\|\xi_k\| \leq c_0) - \|\nabla F(w_k)\| \mathbb{E}[\|\xi_k\| \cdot \mathbf{1}_{\|g_k\|>c}]
\end{aligned} \tag{27}$$

Here, the second equality is because $\mathbb{E}[\langle \nabla F(w_k), \xi_k] = 0$. In the third inequality, we use the following facts. It is not hard to see that

$$\|\nabla F(w_k)\|^2 \cdot \mathbb{E}[\gamma_k] \geq \|\nabla F(w_k)\|^2 \cdot \Pr(\|g_k\| \leq c).$$

Since $\|F(w_k)\| \leq (p-1)c_0$ is assumed, we have the probability of clipping

$$\Pr(\|g_k\| \leq c) \geq \Pr(\|\xi_k\| \leq c_0).$$

As for the second term, when $\|g_k\| \leq c$, i.e., no clipping is performed. In this case, $\gamma_k = 1$ and we have

$$\mathbb{E}[\langle \nabla F(w_k), \xi_k\rangle \cdot (1-\gamma_k) \cdot \mathbf{1}_{\|g_k\|\leq c}] = 0.$$

Thus, we only need to consider $\mathbb{E}[\langle \nabla F(w_k), \xi_k\rangle \cdot (1-\gamma_k) \cdot \mathbf{1}_{\|g_k\|\leq c}]$ when clipping is performed. By Cauchy-Schwartz inequality and $1-\gamma_k \leq 1$, we have the bound in Equation (27). Now, we turn to upper bound $\mathbb{E}[\|\xi_k\| \cdot \mathbf{1}_{\|g_k\|>c}]$. By the *Integrated Tail Probability Expectation Formula*, we can derive the following variant,

$$\begin{aligned}
\mathbb{E}[\|\xi_k\|\mathbf{1}_{\|g_k\|>c}] &\leq \mathbb{E}[\|\xi_k\|\mathbf{1}_{\|\xi_k\|>c_0}] \\
&= c_0(1 - \Pr(\|\xi_k\| < c_0)) + \int_{c_0}^{+\infty} \Pr(\|\xi_k\| > t) dt \\
&\leq c_0 e^{-c_0/\kappa} + \int_{c_0}^{+\infty} e^{-t/\kappa} dt = (c_0 + \kappa) e^{-c_0/\kappa}.
\end{aligned}$$

Putting things together, we have

$$\begin{aligned}
&\mathbb{E}[\langle \nabla F(w_k), \gamma_k g_k\rangle] \\
&\geq (1 - e^{-c_0/\kappa}) \|\nabla F(w_k)\|^2 - (c_0+\kappa) e^{-c_0/\kappa} \|\nabla F(x_k)\| \\
&\geq \frac{(1 - e^{-c_0/\kappa})}{2} \|\nabla F(w_k)\|^2 - (\frac{(c_0+\kappa)e^{-c_0/\kappa}}{1 - e^{-c_0/\kappa}})^2.
\end{aligned}$$

Here, the second inequality we use the following fact: $ax^2 - bx \geq (a/2)x^2 - (b/a)^2$, for any positive constants a.

(2). Second, we consider the case when the gradient is larger where $\|\nabla F(x_k)\| > (p-1)c_0$.

$$\begin{aligned}
&\mathbb{E}[\langle \nabla F(w_k), \gamma_k g_k\rangle] \\
=&\|\nabla F(w_k)\|^2 \cdot \mathbb{E}[\gamma_k] + \mathbb{E}[\gamma_k \langle \nabla F(w_k), \xi_k\rangle] \\
\geq&\|\nabla F(w_k)\|^2 \mathbb{E}[\gamma_k \cdot \mathbf{1}_{\|\xi_k\|\leq c_0}] - \|\nabla F(w_k)\| \mathbb{E}[\gamma_k \|\xi_k\| \cdot \mathbf{1}_{\|\xi_k\|\leq c_0}] \\
&+ \mathbb{E}[\langle \nabla F(w_k), \gamma_k g_k\rangle \cdot \mathbf{1}_{\|\xi_k\|>c_0}] \\
\geq&c\|\nabla F(w_k)\| \Pr(\|\xi_k\| \leq c_0) \cdot \Big(\frac{\|\nabla F(w_k)\|}{\|\nabla F(w_k)\| + c_0} - \frac{c_0}{\|\nabla F(w_k)\| - c_0}\Big) \\
&+ \mathbb{E}[\langle \nabla F(w_k), \gamma_k g_k\rangle \cdot \mathbf{1}_{\|\xi_k\|>c_0}].
\end{aligned} \tag{28}$$

Here, the second inequality is because the term

$$\Big(\frac{\|\nabla F(w_k)\|}{\|\nabla F(w_k)\| + \|\xi_k\|} - \frac{\|\xi_k\|}{\|\nabla F(w_k)\| - \|\xi_k\|}\Big)$$

reaches the minimal when $\|x_{i_k}\| = c_0$ for $\|x_{i_k}\| \in [0, c_0]$. On one hand,

$$\frac{\|\nabla F(w_k)\|}{\|\nabla F(w_k)\| + c_0} - \frac{c_0}{\|\nabla F(w_k)\| - c_0} \geq \frac{(p-1)}{p} - \frac{1}{p-2}.$$

Moreover, we have

$$\mathbb{E}[\langle \nabla F(w_k), \gamma_k g_k\rangle \cdot \mathbf{1}_{\|\xi_k\|>c_0}] \geq -c\|\nabla F(w_k)\| \Pr(\|\xi_k\| > c_0),$$

since $\|\gamma_k g_k\| < c$. Thus, with the assumption $\Pr(\|\xi_k\| > c_0) \leq e^{-c_0/\kappa}$, we have

$$\begin{aligned}
&\mathbb{E}[\langle \nabla F(w_k), \gamma_k g_k\rangle] \\
\geq&c\|\nabla F(w_k)\| \big((1 - e^{-c_0/\kappa})(1 - \frac{1}{p-1} - \frac{1}{p-2}) - e^{-c_0/\kappa}\big).
\end{aligned}$$

□

Now, we pick $p = 4$, and we have the following corollary, i.e.,

$$\mathbb{E}[\langle \nabla F(w_k), \gamma_k g_k\rangle] + a_2 \geq \min\{a_1\|\nabla F(x_k)\|^2, a_3\|\nabla F(x_k)\|\},$$

where

$$a_1 = \frac{(1 - e^{-c/(4\kappa)})}{2}, a_2 = (\frac{(c/4+\kappa)e^{-c/(4\kappa)}}{1 - e^{-c/(4\kappa)}})^2$$

and

$$a_3 = c\big(1/6 - 7/6 \cdot e^{-c/(4\kappa)})\big).$$

Now, we return to Equation (26) and sum up across $k = 0, 1, ..., T-1$, and we have

$$\begin{aligned}
&\mathbb{E}[\frac{\sum_{k=0}^{T-1} \min\{a_1\|\nabla F(w_k)\|^2, a_3\|\nabla F(w_k)\|\}}{T}] \\
&\leq \frac{F(w_0) - F(w_T)}{T\eta} + \frac{\beta\eta}{2}\big(\frac{c^2}{q^2} + v\frac{dT\log(1/\delta)}{n^2\epsilon^2}\big) + a_2 \\
&= (\frac{v}{2} + \frac{3}{2})\frac{c\sqrt{\mathcal{R}_F \beta d \log(1/\delta)}}{n\epsilon q^2} + a_2.
\end{aligned} \tag{29}$$

Here, we set $\eta = \frac{\sqrt{\mathcal{R}_F d\log(1/\delta)}}{n\epsilon(c/q)\sqrt{\beta}}$ and $T = \frac{(n\epsilon)^2}{d\log(1/\delta)}$. Finally, we want to pick c to ensure a_2 is at most in the same order of $O(\frac{\sqrt{d\log(1/\delta)}}{n\epsilon})$, while a_1 and a_3 are positive. To this end, we select c such that

$$c \geq 4\kappa\log(10), \frac{c}{2\kappa} - \log\frac{(c/4+\kappa)^2}{c} \geq 0.4 - \log(\frac{\sqrt{\mathcal{R}_F \beta d\log(1/\delta)}}{n\epsilon}),$$

then the right hand of (29) is further bounded by $(\frac{v}{2} + \frac{5}{2})\frac{c\sqrt{\mathcal{R}_F\beta d\log(1/\delta)}}{n\epsilon}$ while $a_1 = 9/20$ and $a_3 = c/20$. □

We still consider the case where the step size η is scaled by a factor $1/(nq)$ for simplicity. For each $k \in [1:T]$, from the updating rule in Algorithm 1, we have that conditional on w_k and w_{k-1},

$$\begin{aligned}
&\mathbb{E}\big[F(w_{k+1})\big] \\
&\leq \mathbb{E}\big[F(w_k) + \langle \nabla F(w^k), w^{k+1} - w^k\rangle + \frac{\beta}{2}\|w^{k+1} - w^k\|^2\big] \\
&= F(w_k) + \langle \nabla F(w_k), \frac{w_{k-1} - w_k}{2} - \eta\mathbb{E}[\mathsf{CP}(g_k, c)]\rangle \\
&+ \frac{\beta}{2}\mathbb{E}\big[\|(1-\alpha_{k+1})(w_{k-1} - w_k) - \eta/(np)(G_k + \Delta_{k+1}) + e_{\tau_{k+1}}\|^2\big] \\
&\leq F(w_k) + \frac{\langle \nabla F(w_k), w_{k-1} - w_k\rangle}{2} - \eta\langle \nabla F(w_k), \mathbb{E}[\mathsf{CP}(g_k, c)]\rangle \\
&+ \beta\big(\frac{\|w_k - w_{k-1}\|^2}{3} + \eta^2(c^2/q^2 + \|\Delta_{k+1}\|^2/(n^2q^2)) + \|e_{\tau_{k+1}}\|^2]\big).
\end{aligned} \tag{30}$$

Here, we use the fact that $\mathbb{E}(1-\alpha)^2 = 1/3$ for a random $\alpha \in (0,1)$ and the AM-GM inequality. Still, $e_{\tau_{k+1}}$ represents the additional bias caused by the modification which ensures the per coordinate distance between w_k and w_{k-1} is at least τ_{k+1}. We have that the variance $\mathbb{E}[\|e_{\tau_{k+1}}\|^2]$ of e_τ is bounded by $d\tau_{k+1}^2/12$.

Now we apply (16) again on the term $\frac{\langle \nabla F(w^k), w_k - w_{k-1}\rangle}{2}$, we have that

$$\frac{\langle \nabla F(w^k), w_{k-1} - w_k\rangle}{2} \leq \frac{1}{2}\big(F(w_{k-1}) - F(w_k) + \frac{\beta}{2}\|w_k - w_{k-1}\|^2\big). \tag{31}$$

Now, substitute (31) back to (30), we have

$$\begin{aligned}\mathbb{E}[F(w_{k+1})] \leq \frac{F(w_k)+F(w_{k-1})}{2} - \eta\langle \nabla F(w_k), \mathbb{E}[\mathsf{CP}(g_k, c)]\rangle \\ + \beta\big(\frac{7\|w_k - w_{k-1}\|^2 + \tau_{k+1}^2}{12} + \frac{\eta^2}{q^2}\cdot(c^2 + \mathbb{E}[\|\Delta_{k+1}\|^2]/n^2)\big).\end{aligned} \tag{32}$$

In the above inequality, we use the fact that $\|\mathsf{CP}(g_k^j, c)\| \leq c$ due to the clipping. We sum up both sides of (32) and have

$$\begin{aligned}\mathbb{E}\big[\frac{\eta\sum_{k=0}^{T-1}\langle \nabla F(w_k), \mathsf{CP}(g_k, c)\rangle}{T}\big] \leq \frac{3\mathcal{R}_F}{2T} + \beta\eta^2\big(\frac{c^2}{q^2} + \frac{\mathbb{E}[\|\Delta\|^2]}{(nq)^2}\big) \\ + \frac{\beta\sum_{k=0}^{T-1}\big(d\tau_{k+1}^2 + 7\|w_k - w_{k-1}\|^2\big)}{12T}\end{aligned} \tag{33}$$

where $\mathcal{R}_F = \sup_w F(w) - \inf_w F(w)$. With a similar reasoning as (21) in Appendix A, we have that given w_0 and w_{-1},

$$\begin{aligned}\sum_{k=0}^{T-1}\|w_k - w_{k-1}\|^2 \leq \frac{3}{2}\|w_0 - w_{-1}\|^2 + 4\eta c\|w_0 - w_{-1}\| \\ + \frac{11T}{2}\eta^2(c^2/q^2 + \mathbb{E}[\|\Delta\|^2]/(nq)^2).\end{aligned} \tag{34}$$

The rest of the analysis for the term $\mathbb{E}\big[\frac{\eta\sum_{k=0}^{T-1}\langle \nabla F(w_k), \mathsf{CP}(g_k, c)\rangle}{T}\big]$ is the same as the proof of Theorem B.1 where virtually we handle a function whose smooth parameter becomes $\frac{101}{12}\beta$. Therefore, we still select $\eta = \frac{\sqrt{\mathcal{R}_F d\log(1/\delta)}}{n\epsilon(c/q)\sqrt{\frac{101}{12}\beta}}$ and $T = \frac{(n\epsilon)^2}{d\log(1/\delta)}$, and substitute them into (34), and then the right hand of (23) in the ModelMix case becomes

$$\begin{aligned}(\frac{v}{2} + \frac{5}{2})\frac{c\sqrt{\mathcal{R}_F\frac{101}{12}\beta d\log(1/\delta)}}{n\epsilon} + \frac{28c\beta d\log(1/\delta)\|w_0 - w_{-1}\|}{12q(n\epsilon)^2} \\ + \frac{cd\log(1/\delta)\sqrt{\frac{101}{12}}\beta^{3/2}}{qn\epsilon\sqrt{\mathcal{R}_F}}\big(\frac{\sum_{k=1}^T d\tau_k^2}{12} + \frac{21\|w_0 - w_{-1}\|^2}{24}\big).\end{aligned} \tag{35}$$

Appendix C. Proof of Theorem 4.1 and Corollary 4.1

For simplicity, we normalize $\bar{\tau} = \tau/\eta$. We consider the following equivalent aggregation model $\mathcal{M}$. Let $\mathcal{D} = (a_1, a_2, ..., a_n)$ and $\mathcal{D}' = (a_1, a_2, ..., a_{n-1})$ be two adjacent datasets. $\mathcal{M}(\mathcal{D})$ ($\mathcal{M}(\mathcal{D}')$) implements as follows. First, we apply i.i.d. sampling of rate q to generate a subset S from $\mathcal{D}$ ($\mathcal{D}'$) and output its sum $\Sigma(S) = \sum_{a_i\in S} a_i$ perturbed by a sum of Gaussian and independent ModelMix, i.e., $\mathcal{U}[-\frac{\bar{\tau}}{2}, \frac{\bar{\tau}}{2}]^d \times \mathcal{N}(\mathbf{0}, \sigma^2\cdot \boldsymbol{I}_d)$, where we denote this kind of mixture distribution as $\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})$. In the above model, $a_i = \mathsf{CP}(\nabla f(w, x_i, y_i), c)$ represents the individual gradient of each datapoint clipped by c for any i, where we model the distribution of DP-SGD with ModelMix procedure equivalently as

$$\mathcal{U}^d[-\frac{\tau}{2}, \frac{\tau}{2}] + \eta(\sum_{a_i\in S} a_i + \Delta). \tag{36}$$

Now, we can rewrite the distributions of $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$ as follows,

$$\mathcal{M}(\mathcal{D}') = \sum_S p_S \mathcal{MG}_{\bar{\tau},\sigma}^d(\Sigma(S)),$$

for any $S \subset \mathcal{D}\cap\mathcal{D}'$ and P_S represents the probability that S gets sampled from $\mathcal{D}'$ with i.i.d. sampling of rate q. Correspondingly,

$$\mathcal{M}(\mathcal{D}) = \sum_S p_S\big((1-q)\mathcal{MG}_{\bar{\tau},\sigma}^d(\Sigma(S)) + q\mathcal{MG}_{\bar{\tau},\sigma}^d(\Sigma(S) + a_n)\big).$$

Therefore, when we measure the α-Rényi divergence, we have

$$\begin{aligned}&\mathsf{D}_\alpha(\mathcal{M}(\mathcal{D})\|\mathcal{M}(\mathcal{D}')) \\ &\leq \max_S \mathsf{D}_\alpha\big(\mathcal{MG}_{\bar{\tau},\sigma}^d(\Sigma(S))\|(1-q)\mathcal{MG}_{\bar{\tau},\sigma}^d(\Sigma(S)) + q\mathcal{MG}_{\bar{\tau},\sigma}^d(\Sigma(S, a_n))\big) \\ &\leq \max_{v,\|v\|\leq c} \mathsf{D}_\alpha\big(\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})\|(1-q)\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0}) + q\mathcal{MG}_{\bar{\tau},\sigma}^d(v)\big).\end{aligned} \tag{37}$$

In the first and the second inequality of (37), we use the quasi-convexity and translation invariance properties of Rényi divergence, respectively. Thus, in the second inequality we subtract $\Sigma(S)$ on both side and we know $\|v = \Sigma(S\cup a_n) - \Sigma(S) = a_n\|$ is upper bounded by c.

To proceed, we first use the result (Theorem 5) in [42], which suggests that $\mathsf{D}_\alpha(\mathcal{M}(\mathcal{D})\|\mathcal{M}(\mathcal{D}')) \geq \mathsf{D}_\alpha(\mathcal{M}(\mathcal{D}')\|\mathcal{M}(\mathcal{D}))$ and it suffices to consider $\mathsf{D}_\alpha(\mathcal{M}(\mathcal{D})\|\mathcal{M}(\mathcal{D}'))$ to derive the RDP bound in the following. It is noted that

$$\begin{aligned}&(1-\alpha)\mathsf{D}_\alpha(\mathcal{M}(\mathcal{D})\|\mathcal{M}(\mathcal{D}')) \\ &\leq \log\max_v \mathbb{E}_{o\sim\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})}\big((1-q) + q\frac{\mathcal{MG}_{\bar{\tau},\sigma}^d(v)(o)}{\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})(o)}\big)^\alpha \\ &= \log\max_v\big\{\sum_{k=0}^{\alpha}\binom{\alpha}{k}(1-q)^{\alpha-k}q^k\mathbb{E}_{o\sim\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})}[\big(\frac{\mathcal{MG}_{\bar{\tau},\sigma}^d(v)(o)}{\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})(o)}\big)^k]\big\}.\end{aligned}$$

Thus, in the following, it is equivalent to consider $\mathcal{A}_k = \mathbb{E}_{o\sim\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})}[\big(\frac{\mathcal{MG}_{\bar{\tau},\sigma}^d(v)(o)}{\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})(o)}\big)^k]$, which has a semi-closed form

$$\mathcal{A}_k = \max_v \sum_{j=1}^d \log\int \frac{\big(\mathbb{P}(\mathcal{MG}_{\bar{\tau},\sigma}^1(v_j) = o_j)\big)^k}{\big(\mathbb{P}(\mathcal{MG}_{\bar{\tau},\sigma}^1(0) = o_j)\big)^{k-1}}do_j = \max_v\sum_{j=1}^d h(v_j). \tag{38}$$

Here, v_j and o_j represents the j-th coordinated of v and o, and we use the fact that the distribution of each coordinate of $\mathcal{MG}_{\bar{\tau},\sigma}^d(v)$ or $\mathcal{MG}_{\bar{\tau},\sigma}^d(\mathbf{0})$ is independent and thus its density function is a product of $\mathcal{MG}_{\bar{\tau},\sigma}^1(v_j)$ or $\mathcal{MG}_{\bar{\tau},\sigma}^1(0)$. To be specific, we have that $h(v_j)$ can be expressed as

$$h(v_j) = \log\int_o \frac{\big(\int_{-\frac{\bar{\tau}}{2}}^{\frac{\bar{\tau}}{2}}\frac{e^{-(o-\alpha-v_j)^2/(2\sigma^2)}}{\bar{\tau}\sqrt{2\pi\sigma^2}}d\alpha\big)^k}{\big(\int_{-\frac{\bar{\tau}}{2}}^{\frac{\bar{\tau}}{2}}\frac{e^{-(o-a)^2/(2\sigma^2)}}{\bar{\tau}\sqrt{2\pi\sigma^2}}d\alpha\big)^{k-1}}do.$$

Then, to characterize the worst-case divergence, it is equivalent to considering the following

$$\max_v\sum_{j=1}^d h(v_j) \quad \text{where} \quad \sum_{j=1}^d v_j^2 = \|v\|^2 \leq c^2. \tag{39}$$

Due to the symmetric property of $h(\cdot)$, i.e., $h(v_j) = h(-v_j)$, we only need to consider the case where $v_j \geq 0$. Define $g(x) = h(\sqrt{x})$, then solving Equation (39) is equivalent to solving

$$\max \sum_{j=1}^{d} g(\tilde{v}_j) \quad \text{where} \quad \forall j, \tilde{v}_j \in [0, c] \quad \text{and} \quad \sum_{j=1}^{d} \tilde{v}_j \leq c^2. \tag{40}$$

With some simple calculation on the second-order derivative of $g(\tilde{v})$, $g(\tilde{v})$ is a convex function and therefore the maximal of $\sum_{j=1}^{d} g(\tilde{v}_j)$ over a simplex constraint must be achieved at the vertices. Therefore, the maximum is achieved when $\tilde{v}$ is some one-hot vector, and we only need to consider the one-dimensional case by selecting $v = (c, 0, 0, ..., 0)$. As a straightforward proof of Corollary 4.1, when we further ensure the l_∞ norm sensitivity, then $\tilde{v}$ is within the intersection of an l_1 ball of radius c^2 and a hyper cube, where the length of each side is (c^2/p). Then, $\max_v \sum_{j=1}^{d} h(v_j)$ is achieved when $v = (c/\sqrt{p}, ..., c/\sqrt{p}, 0, ..., 0)$ whose Hamming weight is p. The rest proof is a straightforward application of Theorem 2.2 to concert RDP results to (ϵ, δ) DP to get the composition bound claimed.

Remark C.1. *When we apply Laplace mechanism to handle the case of l_1-norm sensitivity, i.e., $\sum_{j=1}^{d} |v_j| \leq c$, the corresponding $h(v_i)$ is still convex with respect to v_i. Therefore, the maximum of $\sum_{j=1}^{d} h(v_i)$ is still achieved when v is a one-hot vector and we can reduce the multi-dimensional problem to the single dimension scenario.*

Appendix D. Proof of Theorem 3.3

Proof of Sketch: Following the arguments of Theorem 4.1, still let $\mathcal{M}$ be the mechanism of one iteration of DP-SGD with ModelMix, and $w_1, w_2, ..., w_T$ be the outputs returned across T iterations. To derive an (ϵ, δ)-DP bound of the cumulative privacy loss, it suffices to consider the high probability bound of the sum of point-wise privacy loss

$$\sup_{\mathcal{D}, \mathcal{D}', \mathsf{Aux}} \sum_{k=1}^{T} \epsilon_{\mathcal{D}, \mathcal{D}', \mathsf{Aux}}(w_k | w_{[1:k-1]}),$$

for $\epsilon_{\mathcal{D}, \mathcal{D}', \mathsf{Aux}}(w_k | w_{[1:k-1]}) = \log \frac{\mathbb{P}(\mathcal{M}(\mathcal{D}, \mathsf{Aux}) = w_k | w_{[1:k-1]})}{\mathbb{P}(\mathcal{M}(\mathcal{D}', \mathsf{Aux}) = w_k | w_{[1:k-1]})}$. In other words, to ensure (ϵ, δ) DP of the cumulative privacy across T iterations, it suffices to ensure =

$$\Pr_{w_{[1:T]}} (\sup_{\mathcal{D}, \mathcal{D}', \mathsf{Aux}} \sum_{j=1}^{T} \epsilon_{\mathcal{D}, \mathcal{D}', \mathsf{Aux}}(w_k | w_{[1:k-1]}) \leq \epsilon) \geq 1 - \delta.$$

In our application, we may apply Bernstein-Azuma inequality [56] to derive a high probability bound of the sum of $\epsilon_{\mathcal{D}, \mathcal{D}', \mathsf{Aux}}(w_k | w_{[1:k-1]})$ for $k = 1, 2, ..., T$. Thus, we calculate the worst case expectation and the variance of $\epsilon_{\mathcal{D}, \mathcal{D}', \mathsf{Aux}}(w_k | w_{[1:k-1]})$, respectively, and then apply Bernstein-Azuma inequality to obtain the bound claimed. The details of calculation can be found below.

D.1. Gaussian mechanism case

Proof. Since we are interested in the asymptotic behavior of $\bar{\tau}$ and q is a constant, it suffices to consider the case when $q = 1$. When q = 1, we are essentially considering the full-batch GD and one can generalize the following analysis via the privacy amplification theorem by sampling [57]. Then, the two output distributions of (36) from $\mathcal{D}$ and $\mathcal{D}'$ we aim to compare are equivalent to

$$\mathbb{P}(o) = \int_{-\frac{\bar{\tau}}{2}}^{\frac{\bar{\tau}}{2}} \frac{1}{\bar{\tau}\sqrt{2\pi\sigma^2}} e^{-\frac{(o-\alpha-c)^2}{2\sigma^2}} d\alpha,$$

and

$$\mathbb{P}'(o) = \int_{-\frac{\bar{\tau}}{2}}^{\frac{\bar{\tau}}{2}} \frac{1}{\bar{\tau}\sqrt{2\pi\sigma^2}} e^{-\frac{(o-\alpha)^2}{2\sigma^2}} d\alpha.$$

We define the pointwise $\epsilon(o)$ loss at o as $\epsilon(o) = \log \frac{\mathbb{P}(o)}{\mathbb{P}'(o)}$.[12] Regarding $\epsilon(o)$, we have the following observation. First,

$$\mathbb{P}(o) = \frac{\Phi(\frac{o-c-\bar{\tau}/2}{\sigma}) - \Phi(\frac{o-c+\bar{\tau}/2}{\sigma})}{\bar{\tau}},$$

$$\mathbb{P}'(o) = P(o+c) = \frac{\Phi(\frac{o-\bar{\tau}/2}{\sigma}) - \Phi(\frac{o+\bar{\tau}/2}{\sigma})}{\bar{\tau}},$$

where $\Phi(t) = \int_t^\infty e^{-t^2/2}/\sqrt{2\pi}$. Regarding $\Phi(t)$, we have the following folk lemma that for any $t > 0$,

$$\frac{\sqrt{2}e^{-t^2/2}}{\sqrt{\pi}(t + \sqrt{t^2+4})} \leq \Phi(t) \leq \frac{\sqrt{2}e^{-t^2/2}}{\sqrt{\pi}(t + \sqrt{t^2+8/\pi})}. \tag{41}$$

For any $t < 0$, we can also bound $\Phi(t)$ since $\Phi(t) = 1 - \Phi(-t)$.

To derive the composition bound of (ϵ, δ), we provide the following lemma to capture the mean and variance of $\epsilon(o)$, respectively.

Lemma D.1. *When $\bar{\tau} > \max\{2\sigma \log \bar{\tau} + 2c, e^{3c/\sigma}, 2\}$ and $c/\sigma = \Theta(1)$, for any ρ such that*

$$\log \rho > \frac{c}{\sigma} + \sqrt{2\log(\bar{\tau} \cdot \max\{c, 1\})},$$

we have

$$\mathbb{E}_{o \sim P}\ \epsilon(o) = O(\frac{c \log^2 \rho}{\bar{\tau}}) + O(\frac{\sigma \log \rho}{\bar{\tau}}),$$

and the variance $\mathcal{V}_0 = Var(\epsilon(o))$ satisfies

$$\mathcal{V}_0 = \frac{1}{\bar{\tau}} \cdot O(\frac{c^4 \log \rho}{\sigma^3} + \frac{c^2 \log^3 \rho}{\sigma} + \sigma \log \rho).$$

Proof. Let us consider the mean first. Note that

$$\begin{aligned}
\mathbb{E}_{o \sim P}\ \epsilon(o) &= \int P(o) \log(\frac{P(o)}{P'(o)}) \\
&\leq \int_{P(o) > P'(o)} P(o) \log(\frac{P(o)}{P'(o)}) \\
&= \int_{c/2}^{+\infty} P(o) \log(\frac{P(o)}{P'(o)}).
\end{aligned}$$

Therefore, we only consider the case when $o > c/2$. Let ρ denote a failure probability parameter that will be specified later. We will divide the $[c/2, +\infty)$ integral into four parts (1) $[c/2, \bar{\tau}/2 - \sigma \log \rho)$, (2) $[\bar{\tau}/2 - \sigma \log \rho, \bar{\tau}/2 + 2c)$, (3) $[\bar{\tau}/2 + 2c, \bar{\tau}/2 + \sigma \log \rho)$ and (4) $[\bar{\tau}/2 + \sigma \log \rho, +\infty)$ and compute the integral of these four parts correspondingly. When $o \in [c/2, \bar{\tau}/2 - \sigma \log \rho]$,

$$\log \frac{P(o)}{P'(o)} = \log(1 + \frac{P(o) - P'(o)}{P'(o)}) < \frac{P(o) - P(o+c)}{P(o+c)}.$$

12. Due to the symmetry of both uniform and Gaussian distributions, it is sufficient to only consider the case by defining $\epsilon(o) = \log \frac{\mathbb{P}(o)}{\mathbb{P}'(o)}$.

By definition, we have

$$P(o+c) = \frac{1}{\bar{\tau}} \cdot (1 - \Phi(\frac{-o+\bar{\tau}/2}{\sigma}) - \Phi(\frac{o+\bar{\tau}/2}{\sigma})) > \frac{1 - 2 \cdot \Phi(\log \rho)}{\bar{\tau}},$$

$$\begin{aligned} P(o) - P(o+c) = & \frac{1}{\bar{\tau}} \cdot (\Phi(\frac{-o-c+\bar{\tau}/2}{\sigma}) - \Phi(\frac{-o+\bar{\tau}/2}{\sigma})) \\ & + \frac{1}{\bar{\tau}} \cdot (\Phi(\frac{o+\bar{\tau}/2}{\sigma}) - \Phi(\frac{o-c+\bar{\tau}/2}{\sigma})) \\ < & \frac{1}{\bar{\tau}} \cdot \Phi(\frac{-o-c+\bar{\tau}/2}{\sigma}) < \frac{1}{\bar{\tau}} \cdot \Phi(\log \rho - \frac{c}{\sigma}). \end{aligned}$$

Combining these with Equation (41), we have that for any $o \in [c/2, \bar{\tau}/2 - \sigma \log \rho)$,

$$\log \frac{P(o)}{P'(o)} < \frac{\Phi(\log \rho - \frac{c}{\sigma})}{1 - 2 \cdot \Phi(\log \rho)} < \frac{e^{-(\log \rho - c/\sigma)^2/2}}{\log \rho - c/\sigma}.$$

If we choose ρ such that

$$\log \rho > \frac{c}{\sigma} + \sqrt{2 \log \bar{\tau}}, \tag{42}$$

then

$$e^{-(\log \rho - c/\sigma)^2/2} < \frac{1}{\bar{\tau}} \implies \log \frac{P(o)}{P'(o)} < \frac{1}{\bar{\tau} \cdot \sqrt{2 \log \bar{\tau}}}.$$

Thus, the integral of the first part $[c/2, \bar{\tau}/2 - \sigma \log \rho]$ is,

$$\int_{c/2}^{\bar{\tau}/2 - \sigma \log \rho} P(o) \log \frac{P(o)}{P'(o)} < \frac{1}{\bar{\tau} \cdot \sqrt{2 \log \bar{\tau}}}.$$

Now, we consider the second part where $o \in [\tau/2 - \sigma \log \rho, \bar{\tau}/2 + 2c)$. In this case, utilizing Equation (41), we can show that

$$\log \frac{P(o)}{P'(o)} < \log \frac{1/\bar{\tau}}{P'(o)} \leq \log \frac{1}{\Phi(\frac{2c}{\sigma}) - \Phi(\frac{2c+\bar{\tau}/2}{\sigma})} = O(\frac{c^2}{\sigma^2}).$$

Therefore,

$$\begin{aligned} \int_{\tau/2 - \sigma \log \rho}^{\bar{\tau}/2 + 2c} P(o) \log \frac{P(o)}{P'(o)} do &= O(\frac{c^2}{\sigma^2}) \cdot \frac{2c + \sigma \log \rho}{\bar{\tau}} \\ &= O(\frac{c^2 \log \rho}{\sigma \bar{\tau}}). \end{aligned}$$

When $o \geq \tau/2 + 2c$, we have

$$\begin{aligned} \frac{P(o)}{P'(o)} &= \frac{\Phi(\frac{o-c-\bar{\tau}/2}{\sigma}) - \Phi(\frac{o-c+\bar{\tau}/2}{\sigma})}{\Phi(\frac{o-\bar{\tau}/2}{\sigma}) - \Phi(\frac{o+\bar{\tau}/2}{\sigma})} < 2 \cdot \frac{\Phi(\frac{o-c-\bar{\tau}/2}{\sigma})}{\Phi(\frac{o-\bar{\tau}/2}{\sigma})} \\ &< 4 \cdot e^{(2(o-\bar{\tau}/2)c - c^2)/(2\sigma^2)}. \end{aligned}$$

Here, the constant factor two in the first inequality is derived based on Equation (41) and $\bar{\tau} > 1$. When $\bar{\tau}$ is large, this factor will approximate 1. We use 2 here as a simple relaxation. Therefore,

$$\log \frac{P(o)}{P'(o)} < \frac{2(o - \bar{\tau}/2)c - c^2}{2\sigma^2} + \log(4).$$

For convenience, we will denote $z = \log(4) - c^2/2\sigma^2$ such that

$$\log \frac{P(o)}{P'(o)} < \frac{(o - \bar{\tau}/2)c}{\sigma^2} + z.$$

This allows us to compute the third and the fourth part as follows. When $o \in [\bar{\tau}/2 + 2c, \bar{\tau}/2 + \sigma \log \rho)$,

$$\begin{aligned} & \int_{\bar{\tau}/2+2c}^{\bar{\tau}/2 + \sigma \log \rho} P(o) \log \frac{P(o)}{P'(o)} do \\ & < \int_{\bar{\tau}/2+2c}^{\bar{\tau}/2 + \sigma \log \rho} P(o) \cdot (\frac{(o - \bar{\tau}/2)c}{\sigma^2} + z) do \\ & < (\frac{c \log \rho}{\sigma} + z) \cdot \int_{\bar{\tau}/2+2c}^{\bar{\tau}/2 + \sigma \log \rho} P(o) do \\ & < (\frac{c \log \rho}{\sigma} + z) \cdot (\sigma \log \rho - 2c) \cdot P(\bar{\tau}/2 + 2c) \\ & = O(\frac{c \log^2 \rho}{\bar{\tau}}) + O(\frac{\sigma \log \rho}{\bar{\tau}}). \end{aligned}$$

Finally, when $o \in [\bar{\tau}/2 + \sigma \log \rho, +\infty)$,

$$\begin{aligned} & \int_{\frac{\bar{\tau}}{2} + \sigma \log \rho}^{+\infty} P(o) \log \frac{P(o)}{P'(o)} do \\ & < \int_{\frac{\bar{\tau}}{2} + \sigma \log \rho}^{+\infty} P(o) \cdot (\frac{(o - \bar{\tau}/2)c}{\sigma^2} + z) do \\ & = O(\frac{1}{\bar{\tau}} \int_{\frac{\bar{\tau}}{2} + \sigma \log \rho}^{+\infty} e^{-(o - \bar{\tau}/2 - c)^2/(2\sigma^2)} \cdot \frac{(o - \bar{\tau}/2)c}{\sigma^2} do) \\ & = O(c e^{-(\sigma \log \rho - c)^2/(2\sigma^2)}). \end{aligned}$$

In order for the integral of this part to be negligible, ρ must be chosen such that

$$c e^{-(\sigma \log \rho - c)^2/(2\sigma^2)} < \frac{1}{\bar{\tau}} \implies \log \rho > \frac{c}{\sigma} + \sqrt{2 \log(\bar{\tau} c)}. \tag{43}$$

Combine this with the requirement in Equation (42), we have the final requirement for ρ that

$$\log \rho > \frac{c}{\sigma} + \sqrt{2 \log(\bar{\tau} \cdot \max\{c, 1\})}. \tag{44}$$

In conclusion, we can sum up the integral of the four parts to show that for any ρ satisfying Equation (44),

$$\mathbb{E}_{o \sim P}\ \epsilon(o) = O(\frac{c^2 \log \rho}{\sigma \bar{\tau}}) + O(\frac{c \log^2 \rho}{\bar{\tau}}) + O(\frac{\sigma \log \rho}{\bar{\tau}}) + O(\frac{1}{\bar{\tau}}).$$

Since $\log \rho > c/\sigma$, we can simplify the above equation to

$$\mathbb{E}_{o \sim P}\ \epsilon(o) = O(\frac{c \log^2 \rho}{\bar{\tau}}) + O(\frac{\sigma \log \rho}{\bar{\tau}}).$$

If we consider c or σ to be constant, then we can simply set $\rho = \Theta(\bar{\tau})$ to satisfy Equation (44) and show that

$$\mathbb{E}_{o \sim P}\ \epsilon(o) = O(\frac{c \log^2 \bar{\tau}}{\bar{\tau}}) + O(\frac{\sigma \log \bar{\tau}}{\bar{\tau}}).$$

Let us now consider the variance.

$$\mathcal{V}_0(\epsilon(o)) = \mathbb{E}_{o \sim P} \epsilon^2(o) = \int_{-\infty}^{c/2} P(o) \epsilon^2(o) do + \int_{c/2}^{+\infty} P(o) \epsilon^2(o) do$$

Note that for any $o > c/2$,

$$\begin{aligned} \epsilon^2(o) &= \log^2 \frac{P(o)}{P'(o)} = \log^2 \frac{P'(o)}{P(o)} \\ &= \log^2 \frac{P'(-o)}{P(2c - o)} = \log^2 \frac{P(c - o)}{P'(c - o)} \\ &= \epsilon^2(c - o). \end{aligned}$$

However, due to property of the Gaussian function, for any $o > c/2$,

$$P(o) = P(2c - o) > P(c - o).$$

Therefore,

$$\begin{aligned}\int_{-\infty}^{c/2} P(o)\epsilon^2(o)do &= \int_{c/2}^{+\infty} P(c-o)\epsilon^2(c-o)do \\ &= \int_{c/2}^{+\infty} P(c-o)\epsilon^2(o)do \\ &< \int_{c/2}^{+\infty} P(o)\epsilon^2(o)do.\end{aligned}$$

Since we have already bounded $\epsilon(o)$ when computing the mean for any $o > c/2$, we have

$$\begin{aligned}\mathcal{V}_0(\epsilon(o)) &< 2\int_{c/2}^{+\infty} P(o)\epsilon^2(o)do \\ &= \frac{1}{\bar{\tau}} \cdot O(\frac{c^4 \log \rho}{\sigma^3} + \frac{c^2 \log^3 \rho}{\sigma} + \sigma \log \rho).\end{aligned}$$

□

The proof in cemma D.1 also implies that $\epsilon(o) < 1/(\bar{\tau} \cdot \sqrt{2\log \bar{\tau}})$ in the first part $o \in [c/2, \bar{\tau}/2 - \sigma \log \rho)$. Furthermore, the probability of o being in the rest three parts is bounded by

$$\Pr[o \geq \bar{\tau}/2 - \sigma \log \rho] < \frac{2\sigma \log \rho}{\bar{\tau}}.$$

We can choose ρ to be the minimum that satisfies Equation (44), which implies that

$$\Pr[o \geq \bar{\tau}/2 - \sigma \log \rho] < \frac{2c + 2\sigma\sqrt{2\log(\bar{\tau} \cdot \max\{c, 1\})}}{\bar{\tau}}.$$

Therefore,

$$\Pr\left[\epsilon(o) \geq \frac{1}{\bar{\tau} \cdot \sqrt{2\log \bar{\tau}}}\right] < \frac{2c + 2\sigma\sqrt{2\log(\bar{\tau} \cdot \max\{c, 1\})}}{\bar{\tau}}.$$

For simplicity, let us define

$$\epsilon_0 = \frac{1}{\bar{\tau} \cdot \sqrt{2\log \bar{\tau}}} \quad \text{and} \quad \delta_0 = \frac{2c + 2\sigma\sqrt{2\log(\bar{\tau} \cdot \max\{c, 1\})}}{\bar{\tau}}.$$

The conditional expectation and variance satisfy

$$\mathbb{E}_{o\sim P}\ [\epsilon(o) \mid \epsilon(o) < \epsilon_0] = \frac{\mathbb{E}_{o\sim P}\epsilon(o)}{1 - \delta_0},$$

$$\mathbb{E}_{o\sim P}\ [\epsilon^2(o) \mid \epsilon(o) < \epsilon_0] = \frac{\mathcal{V}_0}{1 - \delta_0}.$$

Now, we may apply the Bernstein inequality [56] to give a concentration bound. Let $o_1, o_2, ..., o_T$ be the intermediate updates from Algorithm 1 across T iterations. Provided fresh randomness in each iteration, $\epsilon(o_k)$ forms an martingale. Conditional on that $\epsilon(o_k)$ that $|\epsilon(o_k)| \leq \epsilon_0$ for $k = 1, 2, ..., T$, where we can apply Bernstein inequality, we have that with probability at least $1 - (T\delta_0 + \tilde{\delta})$,

$$\begin{aligned}\sum_{k=1}^{T} \epsilon(o_k) &= O\Big(\frac{T\mathbb{E}[\epsilon(o)] + \sqrt{\log(1/\tilde{\delta})(T\mathcal{V}_0 + \epsilon_0)}}{1 - \delta_0}\Big) \\ &= \tilde{O}\Big(\frac{T(c+\sigma)}{\bar{\tau}} + \sqrt{\log(1/\tilde{\delta})}\Big(\sqrt{\frac{c^4/\sigma^3 + c^2/\sigma + \sigma}{\bar{\tau}}} + \frac{1}{\sqrt{\tau}}\Big)\Big),\end{aligned} \tag{45}$$

for any $\tilde{\delta} \in (0, 1)$. Finally, we scale the parameters back to the case of Algorithm 1, where the sensitivity is c/B and $\bar{\tau} = \tau/\eta$, we have the bound claimed. □

D.2. Laplace mechanism case

Proof. In the rest of the proof, we turn to consider the Laplace scenario with bounded l_1-norm sensitivity c. Now we assume that the noise Δ is a Laplace distribution of parameter λ, i.e., $\mathbb{P}(\Delta = o) = \frac{\lambda}{2}e^{-\lambda|o|}$. With a similar reasoning and by Remark C.1, it is equivalent to considering the following distribution,

$$o \sim \mathcal{U}[0, \bar{\tau}] + \text{cap}(\lambda).$$

With some calculation, we know its probability density function (pdf) can be expressed as follows,

$$P(o) = \begin{cases} \dfrac{e^{-\lambda|o|}(1 - e^{-\lambda\bar{\tau}})}{2\bar{\tau}}, & o \leq 0; \\ \dfrac{2 - e^{-\lambda o} - e^{-\lambda(\bar{\tau} - o)}}{2\bar{\tau}}, & o \in (0, \bar{\tau}); \\ \dfrac{e^{-\lambda|o-\bar{\tau}|}(1 - e^{-\lambda\bar{\tau}})}{2\tau}, & o \geq \bar{\tau}. \end{cases}$$

Similarly, we use $P'(o)$ to denote the pdf of $\mathcal{U}[c, c + \tau] + \text{cap}(\lambda)$. We will use $\epsilon_0 = \lambda c$ in the following.

Lemma D.2. *When $\bar{\tau} > \max\{2\log \bar{\tau}/\lambda + c, 1\}$,*

$$\begin{aligned}\mathbb{E}_{o\sim P}\ \epsilon(o) \leq \epsilon_0(e^{\epsilon_0} - 1)\Big(\frac{1 - e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda} + \frac{2\log \bar{\tau}/\lambda + c}{\bar{\tau}}\Big) \\ + \frac{e^{\epsilon_0} - 1}{2(\bar{\tau} - 1)}\Big(e^{\frac{\epsilon_0}{2(\bar{\tau}-1)}} - 1\Big)\Big(1 - \frac{1 - e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda}\Big).\end{aligned}$$

and the variance $\mathcal{V}_0 = Var(\epsilon(o))$ satisfies

$$\mathcal{V}_0 \leq \epsilon_0^2\Big(\frac{1 - e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda} + \frac{2\log \bar{\tau}/\lambda + s}{\bar{\tau}}\Big) + \Big(\frac{e^{\epsilon_0} - 1}{2(\bar{\tau} - 1)}\Big)^2\Big(1 - \frac{1 - e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda}\Big).$$

Proof. We split the calculation of $\mathbb{E}_{o\sim P}\epsilon(o)$ into two parts, where

$$\begin{aligned}\mathbb{E}_{o\sim P}\ \epsilon(o) &= \int_{\mathbb{R}} P(o)\log\frac{P(o)}{P'(o)}\ do \\ &\leq \int_{\mathbb{R}} P(o)\log\frac{P(o)}{P'(o)}\ do + \int_{\mathbb{R}} P'(o)\log\frac{P'(o)}{P(o)}\ do \\ &= \int_{\mathbb{R}} \Big[P(o)\Big(\log\frac{P(o)}{P'(o)} \\ &\quad + \log\frac{P'(o)}{P(o)}\Big) + (P'(o) - P(o))\log\frac{P'(o)}{P(o)}\Big]do \\ &= \int_{\mathbb{R}} (P'(o) - P(o))\log\frac{P'(o)}{P(o)}do \\ &\leq \underbrace{\int_{\mathbb{R}/\mathcal{I}} \sup_{o\in\mathbb{R}/\mathcal{I}} |\log\frac{P'(o)}{P(o)}| \cdot |P'(o) - P(o)|do}_{(A)} \\ &\quad + \underbrace{\int_{\mathcal{I}} \sup_{o\in\mathcal{I}} |\log\frac{P'(o)}{P(o)}| \cdot |P'(o) - P(o)|do}_{(B)}.\end{aligned}$$

Here, $\mathcal{I} = [c + \log \bar{\tau}/\lambda, \bar{\tau} - \log \bar{\tau}/\lambda]$. We first handle (A), where it is easy to verify that ModelMix does not change the worst case ϵ bound and $\sup_{o\in\mathbb{R}/\mathcal{I}} |\log\frac{P'(o)}{P(o)}| \leq \epsilon_0 = \lambda c$. Thus, $|P'(o) - P(o)| \leq (e^{\epsilon_0} - 1)P(o)$. On the other hand, we can upper bound the probability $\Pr(o \in \mathbb{R}/\mathcal{I})$ as follows. It can be computed that $\Pr(o \leq 0) = \Pr(o \geq \bar{\tau}) = \frac{1 - e^{-\lambda\bar{\tau}}}{2\bar{\tau}\lambda}$. As for $o \in (0, \bar{\tau})$, $P(o)$ is indeed concentrated at its mean $\bar{\tau}/2$, and thus

$$\begin{aligned}\Pr(o \in \mathbb{R}/\mathcal{I}) &\leq 2 \cdot \frac{1 - e^{-\lambda\bar{\tau}}}{2\bar{\tau}\lambda} + (1 - \frac{|\mathcal{I}|}{\bar{\tau}}) \cdot (1 - 2\frac{1 - e^{-\lambda\bar{\tau}}}{2\bar{\tau}\lambda}) \\ &\leq \frac{1 - e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda} + \frac{2\log \bar{\tau}/\lambda + c}{\bar{\tau}}.\end{aligned} \tag{46}$$

Therefore, we have $(A) \leq \epsilon_0(e^{\epsilon_0}-1)(\frac{1-e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda_0}+\frac{2\log\bar{\tau}/\lambda+c}{\bar{\tau}})$.

Now, we turn to bound (B). for sufficiently large

$$\bar{\tau} > \max\Big\{c+\frac{2\log\bar{\tau}}{\lambda}, 1\Big\},$$

we have for $o \in \mathcal{I}$,

$$\begin{aligned}|\epsilon(o)| &= \big|\log\big(1+\frac{e^{-\lambda(o-c)}+e^{-\lambda(\bar{\tau}+c-o)}-e^{-\lambda(o)}-e^{-\lambda(\bar{\tau}-o)}}{2-e^{-\lambda(o-c)}-e^{-\lambda(\bar{\tau}+c-o)}}\big)\big| \\ &= \big|\log\big(1+\frac{e^{-\lambda o}(e^{\lambda c}-1)+e^{-\lambda(\bar{\tau}-o)}(e^{-\lambda c}-1)}{2-e^{-\lambda(o-c)}-e^{-\lambda(\bar{\tau}+c-o)}}\big)\big| \\ &\leq \frac{(e^{\lambda c}-1)/\bar{\tau}}{2-2/\bar{\tau}} = \frac{e^{\epsilon_0}-1}{2\bar{\tau}-2}.\end{aligned} \tag{47}$$

Since

$$\sup_{o\in\mathcal{I}}\epsilon(o) \leq \frac{e^{\epsilon_0}-1}{2(\bar{\tau}-1)}$$

and

$$\Pr(o\in\mathcal{I}) \leq 1-\Pr(o\in(-\infty,0)\cup(\bar{\tau},+\infty)) = 1-\frac{1-e^{-\lambda\bar{\tau}}}{\tau\lambda},$$

we have

$$\begin{aligned}\mathbb{E}_{o\sim P}\ \epsilon(o) \leq{}& \epsilon_0(e^{\epsilon_0}-1)(\frac{1-e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda}+\frac{2\log\bar{\tau}/\lambda+c}{\bar{\tau}}) \\ &+\frac{e^{\epsilon_0}-1}{2(\bar{\tau}-1)}(e^{\frac{\epsilon_0}{2(\bar{\tau}-1)}}-1)(1-\frac{1-e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda}).\end{aligned}$$

In the following, we set out to characterize the variance of $\epsilon(o)$. It is noted that

$$\begin{aligned}\mathrm{Var}(\epsilon(o)) &= \mathbb{E}\big[(\epsilon(o)-\mathbb{E}[\epsilon(o)])^2\big] \leq \mathbb{E}[(\epsilon(o)-0)^2] \\ &= \mathbb{E}[\epsilon(o)^2\cdot(\mathbf{1}_{o\in\mathcal{I}}+\mathbf{1}_{o\in\mathbb{R}/\mathcal{I}})].\end{aligned}$$

We use the upper bound $|\epsilon(o)| \leq \frac{e^{\epsilon_0}-1}{2(\bar{\tau}-1)}$ when $o\in\mathcal{I}$ and for the rest we simply bound $|\epsilon(o)|\leq\epsilon_0$, and we have

$$\begin{aligned}\mathcal{V}_0 = \mathrm{Var}(\epsilon(o)) \leq{}& \epsilon_0^2(\frac{1-e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda}+\frac{2\log\bar{\tau}/\lambda+s}{\bar{\tau}}) \\ &+(\frac{e^{\epsilon_0}-1}{2(\bar{\tau}-1)})^2(1-\frac{1-e^{-\lambda\bar{\tau}}}{\bar{\tau}\lambda}).\end{aligned}$$

□

Still, applying Bernstein inequality, to ensure an δ failure probability, we may select

$$\begin{aligned}t &= \sqrt{2}\cdot\sqrt{T\mathcal{V}_0\log(1/\delta)+\log(1/\delta)\epsilon_0^2/9}+\log(1/\delta)\epsilon_0^2/3 \\ &= \tilde{O}(\sqrt{T\log(1/\delta)}\cdot\big(\epsilon_0\sqrt{\frac{1}{\bar{\tau}\lambda}+\frac{s}{\bar{\tau}}}+\frac{e^{\epsilon_0}-1}{\bar{\tau}}\big),\end{aligned}$$

and correspondingly,

$$\begin{aligned}\epsilon &= \tilde{O}\Big(\epsilon_0\cdot\frac{T}{\bar{\tau}}\cdot(e^{\epsilon_0}-1)+(\epsilon_0+\frac{e^{\epsilon_0}-1}{\sqrt{\bar{\tau}}})\cdot\sqrt{\frac{T}{\bar{\tau}}\cdot\log(1/\delta)}\Big) \\ &= \tilde{O}(\frac{\eta T\epsilon_0(e^{\epsilon_0}-1)}{\tau}+\epsilon_0\sqrt{\frac{\eta T\log(1/\delta)}{\tau}}).\end{aligned} \tag{48}$$

□

Appendix E. Experiments Setups

Experiments in Section 5.1 (The comparisons to [19] where we use DP-SGD to privately train a CNN on ScatterNetwork features.) For CIFAR-10, we reproduce the results of [19] by choosing sampling rate $q = 8192/50000$, clipping threshold $c = 0.1$ with stepsize $\eta = 4$ for 700 iterations. For FMNIST, to reproduce the results of [19], we choose sampling rate $q = 8192/60000$, clipping threshold $c = 0.1$ with stepsize $\eta = 4$ for 800 iterations. When we apply ModelMix, for both CIFAR10 and FMNIST, we adopt the same sampling rate as they use above. But we use a larger clipping threshold $c = 1$ with $\eta = 0.4$ and run for the same number of iterations. As for the selection of τ_k, we select $\tau_k = 0.05\eta$ for the first half epochs and $\tau_k = 0.025\eta$ for the rest.

The CNN model that we use is the same as that applied in [19]. The parameters can be found in Table 8 and Table 9 in [19].

Experiments in Section 5.2 (The experiments on training Resnet20 with DP-SGD.) For CIFAR-10, in the application of regular DP-SGD with per-sample clipping, we select the sampling rate $q = 1500/50000$, clipping threshold $c = 20$, $\eta = 0.1$, and run for 3500 iterations. When we apply ModelMix, we select the same $q = 1500/50000$, $c = 20$, but a larger $\eta = 0.15$ and run for 3500 iterations. We set $\tau_k = 0.15\eta$ uniformly.

For SVHN, when we test regular DP-SGD, we select the sampling rate $q = 1830/73260$, clipping threshold $c = 20$, $\eta = 0.1$, and run for 4000 iterations. When we apply ModelMix, we select the same $q = 1830/73260$, $c = 20$, but a larger $\eta = 0.15$ and run for 4000 iterations. We set $\tau_k = 0.15\eta$ uniformly.

Experiments in Section 5.3 (The experiments on DP-SGD applications with further access to public data.) In the private transfer learning application, we reproduce the results of [19] on CIFAR-10 by applying DP-SGD to learn a linear model on the features transformed by pretrained SimCLRv2 network. The parameters are selected as they suggest, where $q = 1024/50000$, $c = 0.1$ and $\eta = 4$, and run for 2700 iterations. When we incorporate ModelMix, we adopt the same $q = 1024/50000$, $c = 0.1$, but a larger stepsize $\eta = 5$. τ is set as $0.0075\eta, 0.005\eta, 0.0025\eta$ for the first, second, and third 1/3 of the total 1500 iterations, respectively. Due to the privacy amplification of ModelMix, we use a smaller noise scaled by 0.85 compared to their selection. Consequently, we achieve $(\epsilon = 0.64, \delta = 10^{-5})$ with the same 92.7% accuracy.

When comparing to the low-dimensional embedding method presented in [32], we adopt $q = 1000/50000$, $c_0 = 5$ for the embedded gradient and $c_1 = 2$ for the residual gradient, and $\eta = 0.1$ for 10000 iterations, as suggested, to reproduce 73.2% accuracy on CIFAR-10 with Resnet20 with a budget $(\epsilon = 8, \delta = 10^{-5})$ under Gaussian Mechanism. When we apply ModelMix, we select $q = 2000/50000$, $\eta = 0.2$, $p = 25$, and with the same clipping thresholds as above and run for 5000 iterations. With the privacy amplification of ModelMix, we uniformly scale the noise added by 0.75 and select $\tau = 0.05$ uniformly for all iterations. We achieve an 74.2% accuracy at a budget $(\epsilon = 2.9, \delta = 10^{-5})$. With the same setup, we can also achieve an accuracy of 79.1% at a budget $(\epsilon = 6.1, \delta = 10^{-5})$.