A APPENDIX

A.1 PROOF OF THEOREM 1

Following the proof of f-DP central limit theorem in (Bu et al., 2020), we have the following definitions given a function f:

$$\mathrm{kl}(f) \triangleq -\int_0^1 \log|f'(x)|\,\mathrm{d}x, \tag{17}$$

$$\widetilde{\mathrm{kl}}(f) \triangleq \int_0^1 |f'(x)|\log|f'(x)|\,\mathrm{d}x, \tag{18}$$

$$\kappa_2(f) \triangleq \int_0^1 \log^2|f'(x)|\,\mathrm{d}x, \tag{19}$$

$$\widetilde{\kappa}_2(f) \triangleq \int_0^1 |f'(x)|\log^2|f'(x)|\,\mathrm{d}x, \tag{20}$$

$$\kappa_3(f) \triangleq \int_0^1 |\log|f'(x)||^3\mathrm{d}x, \tag{21}$$

$$\widetilde{\kappa}_3(f) \triangleq \int_0^1 |f'(x)|\cdot|\log|f'(x)||^3\mathrm{d}x. \tag{22}$$

Let $\{f_{ni}: 1\leqslant i\leqslant n\}_{n=1}^{\infty}$ be a triangular array of trade-off functions and assume the following limits for some constants $K\geqslant 0$ and $s>0$ as $n\to\infty$:

1. $\sum_{i=1}^n \mathrm{kl}\left(f_{ni}\right)+\widetilde{\mathrm{kl}}\left(f_{ni}\right)\to K$,
2. $\max_{1\leqslant i\leqslant n}\mathrm{kl}\left(f_{ni}\right)\to 0, \quad \max_{1\leqslant i\leqslant n}\widetilde{\mathrm{k}}1\left(f_{ni}\right)\to 0$,
3. $\sum_{i=1}^n \kappa_2\left(f_{ni}\right)\to s^2, \quad \sum_{i=1}^n \widetilde{\kappa}_2\left(f_{ni}\right)\to s^2$,
4. $\sum_{i=1}^n \kappa_3\left(f_{ni}\right)\to 0, \quad \sum_{i=1}^n \widetilde{\kappa}_3\left(f_{ni}\right)\to 0$,

it is shown in (Bu et al., 2020) that

$$\lim_{n\to\infty} f_{n1}\otimes f_{n2}\otimes\cdots\otimes f_{nn}(\alpha)=G_{K/s}(\alpha) \tag{23}$$

uniformly for all $\alpha\in[0,1]$. Let $g_t(x)=-G'_{\mu_t}-1=\left|G'_{\mu_t}\right|-1$, where the second equation is due to the shape of the tradoff function. Then equations from (17) to (22) are reformulated as:

$$\mathrm{kl}\left(f_{p,t}\right) = -\int_0^1 \log(1+pg_t(x))\mathrm{d}x, \tag{24}$$

$$\widetilde{\mathrm{kl}}\left(f_{p,t}\right) = \int_0^1 (1+pg_t(x))\log(1+pg_t(x))\mathrm{d}x, \tag{25}$$

$$\kappa_2\left(f_{p,t}\right) = \int_0^1 [\log(1+pg_t(x))]^2\mathrm{d}x, \tag{26}$$

$$\widetilde{\kappa}_2\left(f_{p,t}\right) = \int_0^1 (1+pg_t(x))[\log(1+pg_t(x))]^2\mathrm{d}x, \tag{27}$$

$$\kappa_3\left(f_{p,t}\right) = \int_0^1 [\log(1+pg_t(x))]^3\mathrm{d}x, \tag{28}$$

$$\widetilde{\kappa}_3\left(f_{p,t}\right) = \int_0^1 (1+pg_t(x))[\log(1+pg_t(x))]^3\mathrm{d}x. \tag{29}$$

According to (23), we need to compute $\sum_{t=1}^{T}\left(\mathrm{kl}\left(f_{p,t}\right)+\widetilde{\mathrm{kl}}\left(f_{p,t}\right)\right)$ and $\sum_{t=1}^{T}\kappa_2\left(f_{p,t}\right)$ to obtain the CLT. Let $p\sqrt{T}\to v$ and $p\to 0^+$, we compute K by

$$\begin{aligned}
K &= \lim_{\substack{p\to 0^+\\ p\sqrt{T}\to v}}\sum_{t=1}^{T}\left(\mathrm{kl}\left(f_{p,t}\right)+\widetilde{\mathrm{kl}}\left(f_{p,t}\right)\right)\\
&= \lim_{\substack{p\to 0^+\\ p\sqrt{T}\to v}}\sum_{t=1}^{T} p\int_0^1 g_t(x)\cdot\log(1+pg_t(x))\mathrm{d}x\\
&= \sum_{t=1}^{T}\int_0^1 \frac{v^2}{T}\cdot g_t(x)\cdot\lim_{p\to 0^+}\frac{1}{p}\log(1+pg_t(x))\mathrm{d}x\\
&= \frac{v^2}{T}\sum_{t=1}^{T}\int_0^1 g_t(x)^2\mathrm{d}x\\
&= \frac{v^2}{T}\sum_{t=1}^{T}\left(\mathrm{e}^{\mu_t^2}-1\right).
\end{aligned} \tag{30}$$

We further compute s following the same procedure:

$$\begin{aligned}
s^2 =& \sum_{t=1}^{T}\kappa_2\left(f_{p,t}\right)\\
=& \frac{v^2}{T}\sum_{t=1}^{T}\int_0^1\lim_{p\to 0^+}\left[\frac{1}{p}\log(1+pg_t(x))\right]^2\mathrm{d}x\\
=& \frac{v^2}{T}\sum_{t=1}^{T}\int_0^1 g_t(x)^2\mathrm{d}x\\
=& \frac{v^2}{T}\sum_{t=1}^{T}\left(\mathrm{e}^{\mu_t^2}-1\right).
\end{aligned} \tag{31}$$

Substituting (30) and (31) into (23), we have

$$\lim_{T\to\infty} f_{p,t}\otimes_{t=1}^{T} = G_\mu(\alpha) \tag{32}$$

with

$$\mu = \frac{K}{s} = p\sqrt{\sum_{t=1}^{T}\left(\mathrm{e}^{\mu_t^2}-1\right)}. \tag{33}$$

A.2 Data Sets and Settings

MNIST (Lecun & Bottou, 1998) that contains 60000 training samples and 10,000 testing samples with ten balanced classes. Each grayscale sample is stored as a 28×28 matrix. The classification model used in our experiments consists of two convolutional layers and two fully connected layers. Following each convolutional layer is a max-pooling layer with a pooling size of 2×2. For activations, we use ReLU and for classification, we use softmax. Cross-entropy loss is used.

In the experiment, initial clipping value is set to 1.5 and learning rate is set to 0.15. We set $p=\frac{250}{60000}$ for the subsampling with independent Bernoulli trial. All the results are the average over 5 times repeated experiments.

FashionMNIST is a dataset of Zalando's article images, serving as an alternative to MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits. As a result, we use the same experimental setup as MNIST. The only difference is that we specified a clipping value of 4 as the initial value.

IMDB For natural language processing or text analytics, the IMDB dataset contains 50K movie reviews. This is a binary sentiment classification dataset. We use a three-layer network with one embedding layer and two fully connected layers; this can be viewed as an MLP model because the embedding layer is a special implementation of fully connected layers. We train for 25 epochs with the Adam optimizer, with the initial clipping value set to be 2.

Please keep in mind that the DP-SGD and proposed dynamic DP-SGD can also be used to obtain the DP-Adam counterparts of Adam due to DP's postprocessing properties.

NAME is a name classification dataset containing person names from 18 countries. It is available on Pytorch NLP tutorials[4]. We train a LSTM model to determine which country the given name belongs to. The name is treated as a sequence, and the characters are fed into LSTM one by one. We use a one-layer LSTM with hidden size 128 and embedding size 64 followed by a fully connected layer, an SGD optimizer with learning rate 2 and we train 50 epochs for each experiment, and the initial clipping value is set to be 1.5.

InfiniteMNIST and Federated Learning InfiniteMNIST is a dataset consisting of massive training samples derived from origin MNIST by applying different types of transformations. Such large scale dataset is suitable for a federated setting. We extract 250K and 500K images as two training data sets and simulate identical number of clients. Thus each client is not enough to train the model, but they can cooperatively learn a model via federated learning. We use the same network structure as in the experiment for MNIST for each client.

Federated Learning Algorithm Fed-SGD algorithm is used for optimization. For each round, we randomly sample clients with sampling rate $p = 1 \times 10^{-3}$ for the case of MNIST-250K and $p = 5 \times 10^{-4}$ for the case of MNIST-500K. Gradients are computed on each selected client and then sent to the server for aggregation. In this setting, local DP mechanism is required to protect client side gradients, and each client apply clipping and additive noise on local gradients before transmission. The server will aggregate noised gradients, which achieves central DP. The details is provided in the following Algorithm 3.

Algorithm 3 Federated Dynamic DP Algorithm

Require: Clients set $\mathcal{S}$, DP budget (ϵ, δ), client sampling rate p, dataset $X = (X_1, X_2 ... X_{|\mathcal{S}|})$, training rounds T, hyper-parameters: ρ_μ, ρ_c and C_0.

Compute μ_0 in Algorithm 1
for $t = 0, \ldots, T-1$ **do**
  Compute $C_t = (\rho_c)^{-\frac{t}{T}} \cdot C_0$ according to (14)
  Calibrate noise : $\sigma_t = \frac{C_0}{\mu_0}(\rho_\mu \cdot \rho_c)^{-\frac{t}{T}}$
  Sample clients for the t-th iteration, i.e., $\mathcal{S}_t \in \mathcal{S}$ with Poisson sampling rate p.
  for $k \in S_t$ **do**
    $g_{k,t} = \frac{1}{|\mathcal{S}_t|}\left(\sum_{x \in \mathcal{S}_t}(g_x; C_t) + \xi_{k,t}\right)$ with $\xi_{k,t} \sim \mathcal{N}(0, \sigma^2 I)$
    Send $g_{k,t}$ to the server.
  end for
  Server computes: $\theta_t = \theta_{t-1} - \eta \frac{1}{|\mathcal{S}_t|}\left(\sum_{k \in \mathcal{S}_t} g_k\right)$
end for

[4]https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html