Communication-Efficient Differentially Private Federated Learning Using Second-Order Information

Mounssif Krouka
University of Oulu, Finland

Antti Koskela & Tejas Kulkarni
Nokia Bell Labs

Abstract

Training machine learning models with differential privacy (DP) is commonly done using first-order methods such as DP-SGD. In the non-private setting, second-order methods try to mitigate the slow convergence of first-order methods. The DP methods that use second-order information still provide faster convergence, however the existing methods cannot be easily turned into federated learning (FL) algorithms without an excessive communication cost required by the exchange of the Hessian or feature covariance information between the nodes and the server. In this paper we propose DP-FedNew, a DP method for FL that uses second-order information and results in per-iteration communication cost similar to first-order methods such as DP Federated Averaging.

1 Introduction

The goal of this work is communication efficient DP federated learning applicable to realistic settings. To this end, we focus on methods that use existing secure summation protocols for the implementation of the DP model training methods. In the context of FL (Kairouz et al., 2021b), combining secure aggregation with DP (Ullah et al., 2023; Hartmann & Kairouz, 2023; Kairouz et al., 2021a) reduces the trustworthiness assumptions on a central server. Specifically, when the DP noise in the model updates is additive and the model updates are sums of user-wise updates, DP perturbations can be offloaded to clients to obtain the global model under cryptographic guarantees (Truex et al., 2019) in addition to DP's usual statistical privacy guarantees.

With the performance gap between private and non-private training shrinking rapidly, communication costs in FL can easily become a bottleneck in adoption of these protocols. Several works including (Ullah et al., 2023; Chen et al., 2023b; 2022b; 2023a; 2022a) employ various compression and sketching tools to design communication efficient DP mechanisms for distributed mean estimation compatible with secure aggregation.

In this work, we take an optimization perspective and rely on existing DP secure aggregation primitives. In DP non-FL setting, methods that use second-order information have been recently developed for private convex problems and show impressive improvements in the privacy-utility tradeoffs. For example, Mehta et al. (2023) consider a method called DP-FC where the DP gradients are preconditioned with a noisy feature covariance matrix. The work by Ganesh et al. (2023) gives a DP second-order method with rigorous convergence analysis, with utility bounds matching the lower bounds of private empirical risk minimization Bassily et al. (2014).

Unfortunately, neither of these methods seem to be easily transferrable to the FL setting. For d_x features, the distributed version of (Mehta et al., 2023) requires a one time aggregation of a noisy covariance matrix of size $O(d_x^2)$ from users. The $O(d_x^2)$ term can still dominate in the total communication cost when $d_x \gg T$ (training length T). The method by Ganesh et al. (2023) does not seem to be easily transferrable to the FL setting due to the inverse of a non-private Hessian in the model update. It is the main goal of this work to fill this gap in the private FL literature.

1.1 Our Contributions

This paper proposes a DP optimization method for convex problems in FL that leverages the benefits of fast convergence of second-order methods and the communication cost of first-order methods. In particular, we build upon the work of Elgabli et al. (2022) where the Newton update step is approximated using one ADMM pass. The major contributions of this work are summarized as follows.

- To the best of our knowledge, we propose the first DP optimization method in the context of FL that uses second-order information via Hessians and has a model size communication cost. We do this by successfully building upon the work of Elgabli et al. (2022) where the Newton update step is approximated using one ADMM pass.
- We carry out comprehensive experiments for convex problems where we show that our proposed algorithm copes with various privacy budgets and excels in terms of test accuracy, outperforming the baseline methods. To mitigate the excessive compute and memory requirements for large Hessian matrices, we suggest a variant of DP-FedNew where we replace the Hessian with a certain approximation that uses the feature covariance matrices.
- We provide an asymptotic convergence analysis for the proposed method.

2 Background on Differential Privacy

An input dataset containing N data points is denoted as $D = (x_1, \ldots, x_N) \in \mathcal{D}$, where $\mathcal{D}$ denotes the set of datasets of all sizes. We say that two datasets D and D' are neighbors if we get one by adding or removing one element to/from the other (denoted $D \sim D'$). We say that a mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{O}$ is (ε, δ)-DP if the outputs for neighboring datasets are always (ε, δ)-indistinguishable.

Definition 1. *Let $\varepsilon \geq 0$ and $\delta \in [0,1]$. Mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{O}$ is (ε, δ)-DP if for every pair of neighboring datasets D, D', every measurable set $E \subset \mathcal{O}$,*

$$\mathbb{P}(\mathcal{M}(D) \in E) \leq \mathrm{e}^{\varepsilon} \mathbb{P}(\mathcal{M}(D') \in E) + \delta.$$

We call $\mathcal{M}$ tightly (ε, δ)-DP, if there does not exist $\delta' < \delta$ such that $\mathcal{M}$ is (ε, δ')-DP.

We refer to Definition (1) as *record-level DP*. In case we have n users and x_i's correspond to the whole local dataset owned by user i, $i \in [n]$, we call the corresponding DP definition *user-level DP*. (McMahan et al., 2018c;b).

In this work, we provide an accurate (ε, δ)-analysis for our methods using the hockey-stick divergence. When analyzing our DP-FL training methods, we model them as adaptive compositions such that the adversary has a view on all the intermediate global models. This means that we analyze mechanisms of the form

$$\mathcal{M}^{(T)}(D) = \big(\mathcal{M}_1(D), \mathcal{M}_2(\mathcal{M}_1(D), D), \ldots, \mathcal{M}_T(\mathcal{M}_1(D), \ldots, \mathcal{M}_{T-1}(D), D)\big). \tag{2.1}$$

In the methods we consider, each $\mathcal{M}_i$, $i \in [T]$, will correspond to a Gaussian mechanism and thus the analysis is equivalent to that of the Gaussian mechanism.

Lemma 2. *Consider an adaptive composition of T Gaussian mechanisms, each with L_2-sensitivity Δ and noise scale parameter σ. The adaptive composition is (ε, δ)-DP for*

$$\delta(\varepsilon) = \Phi\left(-\frac{\varepsilon\sigma}{\sqrt{T}\cdot\Delta} + \frac{\sqrt{T}\cdot\Delta}{2\sigma}\right) - e^{\varepsilon}\Phi\left(-\frac{\varepsilon\sigma}{\sqrt{T}\cdot\Delta} - \frac{\sqrt{T}\cdot\Delta}{2\sigma}\right).$$

3 FedNew

We consider as a starting point the single pass ADMM-method called FedNew as given in (Elgabli et al., 2022). We simply list here the method, more details are given in Appendix Section C.

Let T denote the total number of training iterations. Denote by θ^k are the global model parameters at iteration k, $H_i^k = \nabla^2 f_i(\theta^k)$ and $g_i^k = \nabla f_i(\theta^k)$. Also, denote the primal and dual variables of user i, $i \in [n]$, at iteration k, $k \in [T]$, as y_i^k and λ_i^k, respectively, and the global primal and dual variables at iteration k as y^k and λ^k. Then, the FedNew algorithm is described by the following steps.

1. At user i, at round k, the update of the primal variable is obtained from the local minimization problem
$$y_i^k = \arg\min_y \left[\tfrac{1}{2} y^T (H_i^k + \alpha I) y + \langle \lambda_i^{k-1}, y - y^{k-1} \rangle - y^T g_i^k + \tfrac{\rho}{2} \| y - y^{k-1} \|_2^2 \right]$$
for which the solution can be written as
$$y_i^k = (H_i^k + \alpha I + \rho I)^{-1} (g_i^k - \lambda_i^{k-1} + \rho y^{k-1}). \tag{3.1}$$
2. The primal variable update at the server is obtained by solving the problem
$$y^k = \arg\min_y \left[\sum_{i=1}^n \langle \lambda_i^{k-1}, y_i^k - y \rangle + \tfrac{\rho}{2} \sum_{i=1}^n \| y - y_i^k \|_2^2 \right]$$
which gives the solution
$$y^k = \tfrac{1}{n} \sum\nolimits_{i=1}^n (y_i^k + \tfrac{1}{\rho} \lambda_i^{k-1}). \tag{3.2}$$
3. The dual variables are updated locally: $\lambda_i^k = \lambda_i^{k-1} + \rho(y_i^k - y^k)$.
4. The global model parameters are updated as $\theta^{k+1} = \theta^k - \eta \cdot y^k$, where $\eta > 0$ denotes the learning rate hyperparameter.

Since $\sum_{i=1}^n \lambda_i^k = 0$, the update (3.2) can be written as an average of the primal variables: $y^k = \frac{1}{n} \sum_{i=1}^n y_i^k$.

4 DP-FedNew

There the global primal variable y^{k-1} and the dual variable λ_i^{k-1} are results of previous iterations and therefore do not incur additional per-iteration privacy cost. The only data-dependent objects are the gradients $g_i^k = \nabla f_i(\theta^k)$ which are functions of data and the previous iterations primal variables $y^{k-1}, y^{k-2}, \ldots$. We consider separately the user and record-level DP versions of DP-FedNew. The User-Level Algorithm is described in Appendix Section E. In both cases, the only modification to the FedNew algorithm happens in the update (3.1) of local primal variables where we add noise. In the record level case, instead of only limiting the sensitivity of the gradients $\nabla f_i(\theta)$ by clipping and adding normally distributed noise as in DP-SGD (Abadi et al., 2016), we need to consider the potential privacy leakage via the Hessians $\nabla^2 f_i(\theta)$ which are data-dependent. We use the additive Gaussian noise, however, remark that our algorithm is compatible with any suitable DP secure aggregation primitive (e.g. (Chen et al., 2022b; Kairouz et al., 2021a)) closed under summation and other noise distributions could be considered.

4.1 DP-FedNew with Record-Level Privacy

In the record-level case, to obtain the DP guarantees, we need to bound the sensitivity of y_i^k w.r.t. changes of data elements. Suppose the user i has the dataset D_i. Then, the data-dependent function that needs to be randomized is
$$F(D_i, \alpha, \rho, \lambda_i^{k-1}, y^{k-1}) = (H_i^k + \alpha I + \rho I)^{-1} (g_i^k - \lambda_i^{k-1} + \rho y^{k-1}).$$
Here α and ρ are pre-defined constant, and λ_i^{k-1} and y^{k-1} are auxiliary variables that are outputs of previous iterations. H_i^k stands for the Hessian and g_i^k for the gradient of user i. For limiting the sensitivity of the function F w.r.t. change of a single data element in D_i, we have the following result which justifies our record-level clipping procedure described in Algorithm 1.

Lemma 3. *Let D_i' and D_i be neighboring datasets such that $D_i' = D_i \cup \{x'\}$ for some data-element x'. Let Δ_i be defined as*
$$\Delta_i := F(D_i', \alpha, \rho, \lambda_i^{k-1}, y^{k-1}) - F(D_i, \alpha, \rho, \lambda_i^{k-1}, y^{k-1}).$$
Denote $\gamma = \rho + \alpha$. Assume
$$\begin{aligned} \| \nabla^2 f(x, \theta^k) \|_2 &\leq \Delta_H \quad \textit{for all } x \in D_i, \\ \| \nabla f(x, \theta^k) \|_2 &\leq C_1 \quad \textit{for all } x \in D_i, \\ \| g_i^k - \lambda_i^{k-1} + \rho y^{k-1} \| &\leq C_2, \end{aligned} \tag{4.1}$$

and $\gamma > \frac{\Delta_H}{|D_i|}$. *Then, we have:*

$$\|\Delta_i\|_2 \leq \frac{1}{\gamma \cdot |D_i|} \cdot C_1 + \frac{\Delta_H}{\gamma^2 \cdot |D_i| - \gamma \cdot \Delta_H} \cdot C_2,$$

where $|D_i|$ *is the size of the local dataset* D_i*.*

Algorithm 1 Record-level DP-FedNew algorithm to compute private y_i^k.

```
Input: clipping constants $C_1, C_2, \Delta_H > 0$, noise parameter $\sigma > 0$, regularization parameters $\alpha$ and $\rho$.
for iteration $k = 1, \ldots, T$ do
  for user $i = 1, \ldots, n$ do
      $g_i^k = \frac{1}{|D_i|} \sum_{x \in D_i} \text{clip}_{C_1}\big(\nabla f(x, \theta^k)\big)$
      $H_i^k = \frac{1}{|D_i|} \sum_{x \in D_i} \text{clip}_{\Delta_H}\big(\nabla^2 f(x, \theta^k)\big)$
    Scale the auxiliary variables with $C_3$: $g_{\text{sum}} = g_i^k - \lambda_i^{k-1} + \rho y^{k-1}$.
    if $\|g_{\text{sum}}\|_2 > C_2$ then
      $g_{\text{sum}} = g_i^k + \xi \cdot (-\lambda_i^{k-1} + \rho y^{k-1})$, where the scalar $\xi$ is chosen using Lemma 4 such that $\|g_i^k + \xi \cdot (-\lambda_i^{k-1} + \rho y^{k-1})\|_2 \leq C_2$.
    end if
    Compute the non-DP update of the primal variable:
      $\widehat{y}_i^k = (H_i^k + \gamma I)^{-1} g_{\text{sum}},$
    where $\gamma = \alpha + \rho$.
    Clip and perturb the primal variable:
      $\widetilde{y}_i^k \leftarrow \widehat{y}_i^k + E_i^k, E_i^k \sim \mathcal{N}(0, \frac{C^2\sigma^2}{n} I_d),$
    where $C = \frac{C_1}{\gamma \cdot |D_i|} + \frac{\Delta_H \cdot C_2}{\gamma^2 \cdot |D_i| - \gamma \cdot \Delta_H}$.
  end for
end for
```

For scaling the auxiliary variables in Algorithm 1, we can use the following analytical formula.

Lemma 4. *Let* $a, b \in \mathbb{R}^n$ *and* $C > 0$*. If we set*

$$\xi = \frac{-2\langle a, \frac{b}{\|b\|_2}\rangle + \sqrt{4\langle a, \frac{b}{\|b\|_2}\rangle^2 + 4(C^2 - \|a\|_2^2)}}{2\|b\|_2},$$

we have that $\|a + \xi \cdot b\|_2 = C$*.*

4.2 Memory Efficient Hessian Approximation

In our experiments and convergence analysis we focus on generalized linear models such as the logistic regression. This class of problems has recently turned out an attractive approach for private fine-tuning of large models pre-trained using public data (see, e.g., De et al., 2022; Mehta et al., 2023). Therein, a well-justified approximation of the Hessian using the covariance matrix of the feature vectors can be derived as follows (Mehta et al., 2023). Assume the loss function is of the form $f\big((x, y), \theta\big) = \ell(\theta^T x - y)$ for some twice differentiable function ℓ, where $x \in d_x$, $y \in c$, $\theta \in \mathbb{R}^{d_x \times c}$. Then, for a vectorized $d = d_x \cdot c$-dimensional model variables, the Hessian is a $(d_x \cdot c \times d_x \cdot c)$ matrix. We simply make the block-diagonal approximation $H((x, y), \theta) \approx I_c \otimes x x^T$. When the variables and gradients are expressed in $d_x \times c$ matrix form, we can replace the scaled Hessian times the gradient product with the product $\left(\frac{1}{|D_i|} X_i X_i^T + \gamma I\right)^{-1} g_{\text{sum}}$, where then $g_{\text{sum}} \in \mathbb{R}^{d_x \times c}$. This reduces both the compute and memory requirement considerably. For example, when $d_x = 512$ and output dimension $c = 10$, we shrink the memory consumption to 1%. We note that this approximation affect the privacy nor our convergence analysis and our experiments show that we still outperform the first order baseline. We call the resulting method DP FedNew Feature Covariance method (DP-FedNew-FC).

5 Convergence Analysis

The analysis of FedNew as given in (Elgabli et al., 2022) is an asymptotic analysis such that the primal variables y_i^k get closer to the optimal primal variables $y_i^{k,*}$ which would be the result of running the inner loop until convergence. I.e., they show that

$$\lim_{k \to \infty} \| y_i^k - y^{k,*} \|_2^2 = 0. \tag{5.1}$$

With $y^{k,*}$'s the outer loop corresponds to a single step of Newton's iteration.

In case of DP-noise perturbed local updates, we cannot have the asymptotic convergence of the form (5.1). However, under the assumptions of the analysis by Elgabli et al. (2022) and some weak assumptions related to the DP version of the algorithm, we obtain an asymptotic limit of the form

$$\lim_{k \to \infty} \| y^k - y^{k,*} \|_2^2 = \mathcal{O}\left(\frac{d\sigma^2}{n} \right),$$

where $y^k = \sum_{i=1}^n \widetilde{y}_i^k$ is a sum of the local noisy updates and $y^{k,*}$ again corresponds to a Newton update. Asymptotically, we can think of the iteration as a noisy Newton iteration where the additional noise matches the amount of local noise that one has, e.g., in DP gradient descent.

6 Experimental Results

For a full description of our datasets and experimental setup, please refer to Section J in the Appendix.

Datasets. As IID datasets, we use CIFAR10 (Krizhevsky & Hinton, 2009), EMNIST (Cohen et al., 2017b), and FashionMnist (Xiao et al., 2017). We extract features of sizes 64 and 2048 from the last layer of pretrained resnets mentioned in Table 3. We pick synthetic and Federated EMNIST as non-IID datasets, which are also used in (Noble et al., 2022). For experiments on non-IID datasets, we train linear layers from scratch.

Baselines. Noble et al. (2022) proposed a DP variant of the seminal Scaffold Karimireddy et al. (2020) method designed specifically to tackle data heterogeneity. In each global iteration, DP-Scaffold requires clients and server to exchange the parameter and control variable information, making the communication cost $2 \times d$. We treat a federated version of DP-GD (DP-FedGD) as a second first-order baseline method (depicted in Algorithms 5 and 6) because it has the same privacy and communication cost as DP-FedNew. In the centralized case, Mehta et al. (2023) came up with the DP-FC method which involves pre-multiplying the noisy full batch mean gradients with the inverse of a noisy feature covariance matrix. We present federated versions of DP-FedFC in Algorithms 3 and 4. Similar to DP-FedAVG, both DP-FedNew and DP-FedFC can be modified to perform multiple local client-side updates. However, full exploration of these variations justifies a separate work.

6.1 Experiments of Record-Level DP: IID Data

We train a linear layer and compare the performance of our algorithm, DP-FedNew with DP-FedGD. The dimension (d_x) of the features extracted from the pre-trained model is 64, and with the number of classes $c = 10$. We here tune only the learning rate η for DP-FedGD, and additionally tune the constants Δ_H, α, and ρ for DP-FedNew. Figure 1 shows the mean test accuracy for both models. We observe that DP-FedNew outperforms DP-FedGD by a large margin on all datasets for all ϵ levels. The results indicate a behavior observed before for private non-FL methods that use second order information Ganesh et al. (2023): our methods benefit from the second-order information such that they approach the optimal privacy-utility-tradeoff faster than first-order methods. The figures in Appendix Section J show more results, for both user and record-level DP.

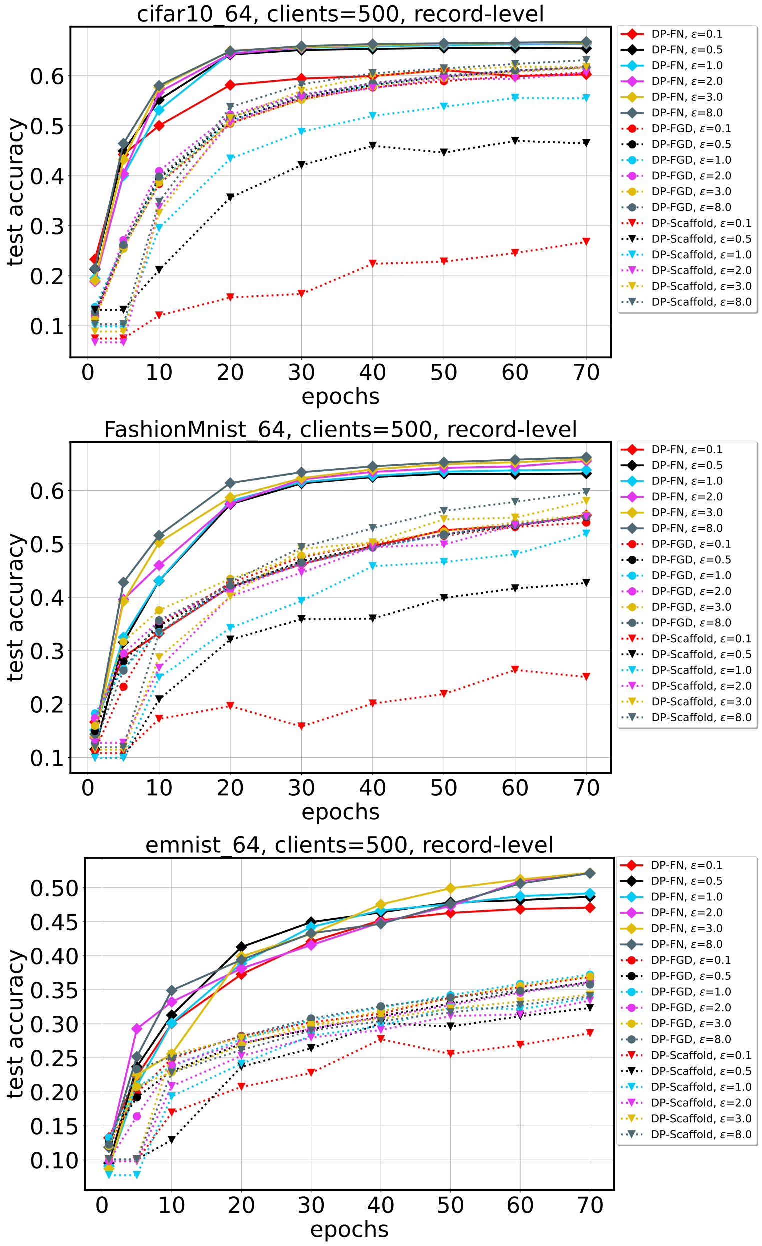

Figure 1: IID data: Record-level results for DP-FedNew (DP-FN in plots), DP-FedGD (DP-FGD in plots), and DP-Scaffold. For each ϵ, we plot the test accuracies of the best model obtained after hyperparameter tuning. The model size is 64×10.

6.2 Experiments of Record-Level DP: Non-IID Data

Figure 2 compare all three solutions on Federated EMNIST and synthetic datasets for record-level DP. We can see in Figure 2 that DP-FedNew-FC performs at par with DP-FedGD for low ε-values and outperforms it for higher ε even when each client has a dataset for a single label.

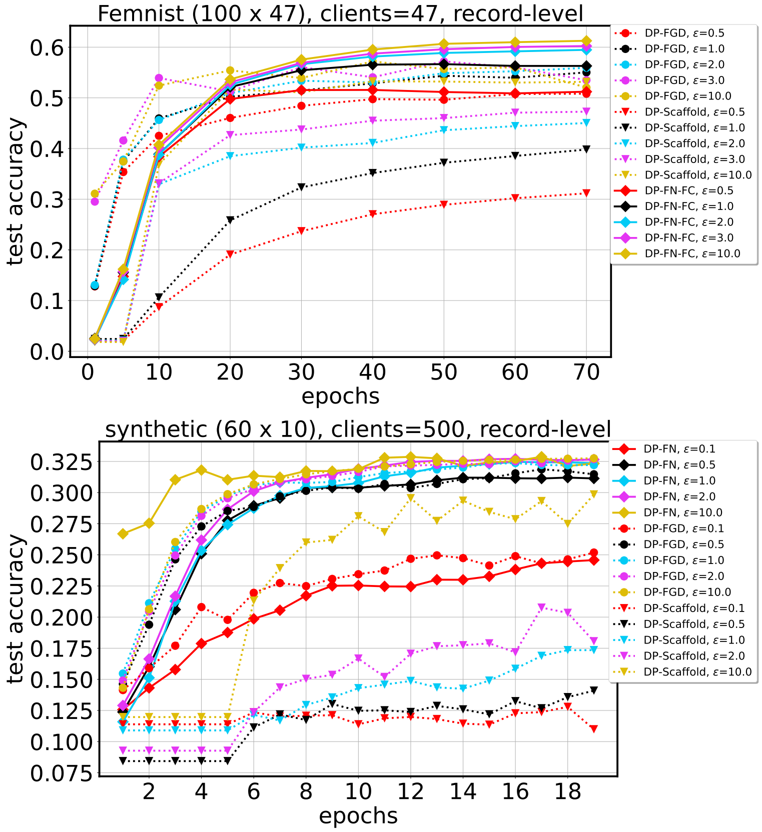

Figure 2: Non-IID data: Record-level results for DP-FedNew (DP-FN in plots) and DP-FedGD (DP-FGD in plots). For each ϵ, we plot the average test accuracies of the best model obtained after hyperparameter tuning. The model sizes are 100×47 and 60×10.

7 Concluding Remarks

We propose the first DP distributed optimization with model-sized communication overhead that uses the curvature information via the Hessian matrix of the loss function. Our approach nicely complements an orthogonal line of research dedicated to the development of resource efficient DP primitives for secure aggregation.

The experimental results indicate a behavior shown before for private non-FL methods that use second order information (Ganesh et al., 2023): our methods benefit from the speed up given by the second-order information such that they approach the optimal privacy-utility-tradeoff faster than first-order methods. This shows up as better privacy-utility-tradeoffs in reasonable-length training runs compared to first-order methods. One interesting line of future work is to consider modifications where multiple local steps are run and also experimental comparisons against DP-FedAve with multiple local iterations.

REFERENCES

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

Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pp. 394–403. PMLR, 2018.

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

Wei-Ning Chen, Christopher A. Choquette-Choo, Peter Kairouz, and Ananda Theertha Suresh. The fundamental price of secure aggregation in differentially private federated learning. In *International Conference on Machine Learning, ICML 2022*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3056–3089. PMLR, 2022a.

Wei-Ning Chen, Ayfer Özgür, and Peter Kairouz. The poisson binomial mechanism for unbiased federated learning with secure aggregation. In *International Conference on Machine Learning, ICML 2022*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3490–3506. PMLR, 2022b.

Wei-Ning Chen, Peter Kairouz, and Ayfer Özgür. Breaking the communication-privacy-accuracy trilemma. *IEEE Trans. Inf. Theory*, 69(2):1261–1281, 2023a.

Wei-Ning Chen, Dan Song, Ayfer Özgür, and Peter Kairouz. Privacy amplification via compression: Achieving the optimal privacy-accuracy-communication trade-off in distributed mean estimation. *Advances in Neural Information Processing Systems*, 36, 2023b.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017a.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, 2017b.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Anis Elgabli, Chaouki Ben Issaid, Amrit Singh Bedi, Ketan Rajawat, Mehdi Bennis, and Vaneet Aggarwal. Fednew: A communication-efficient and privacy-preserving newton-type method for federated learning. In *International Conference on Machine Learning*, pp. 5861–5877. PMLR, 2022.

Arun Ganesh, Mahdi Haghifam, Thomas Steinke, and Abhradeep Thakurta. Faster differentially private convex optimization via second-order methods. *arXiv preprint arXiv:2305.13209*, 2023.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems*, 2021.

Florian Hartmann and Peter Kairouz. Distributed differential privacy for large-scale data analysis. Blog, March 2023.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5201–5212. PMLR, 2021a.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021b.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Antti Koskela and Tejas Kulkarni. Practical differentially private hyperparameter tuning with subsampling. *37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, abs/2301.11989, 2023. URL `https://openreview.net/pdf?id=OeLInnFKUK`.

Antti Koskela, Joonas Jälkö, Lukas Prediger, and Antti Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using FFT. In *International Conference on Artificial Intelligence and Statistics*, pp. 3358–3366. PMLR, 2021.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020*. mlsys.org, 2020. URL `https://proceedings.mlsys.org/book/316.pdf`.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS*, Proceedings of Machine Learning Research, 2017. URL `http://proceedings.mlr.press/v54/mcmahan17a.html`.

Brendan McMahan, Galen Andrew, Ilya Mironov, Nicolas Papernot, Peter Kairouz, Steve Chien, and Úlfar Erlingsson. A general approach to adding differential privacy to iterative training procedures. *NeurIPS 2018 workshop on Privacy Preserving Machine Learning (PPML)*, 2018a.

H. Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *CoRR*, abs/1812.06210, 2018b.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *6th International Conference on Learning Representations, ICLR 2018,*. OpenReview.net, 2018c.

Harsh Mehta, Walid Krichene, Abhradeep Guha Thakurta, Alexey Kurakin, and Ashok Cutkosky. Differentially private image classification from features. *Transactions on Machine Learning Research*, 2023.

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

Maxence Noble, Aurélien Bellet, and Dieuleveut Aymeric. Differentially private federated learning on heterogeneous data. In *25th International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

David M Sommer, Sebastian Meiser, and Esfandiar Mohammadi. Privacy loss classes: The central limit theorem in differential privacy. *Proceedings on Privacy Enhancing Technologies*, 2019(2): 245–269, 2019.

Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pp. 1–11, 2019.

Enayat Ullah, Christopher A. Choquette-Choo, Peter Kairouz, and Sewoong Oh. Private federated learning with autotuned compression. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pp. 34668–34708. PMLR, 2023.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. Technical report, 2017.

Yuqing Zhu, Jinshuo Dong, and Yu-Xiang Wang. Optimal accounting of differential privacy via characteristic function. *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

A Communication Costs of the Methods Considered

Table 1: Our proposed methods DP-FedNew (Algorithm 1 and 4) and DP-FedNew-FC have both $\mathcal{O}(d)$ communication and use second-order information about the loss functions. Neither of the baseline methods DP-Scaffold 2, DP-FedGD (Algorithm 5 and 6) and DP-FedFC (Algorithm 3 and 4) do not have both of these properties. Here d, d_x denote the dimension of the model and features. T is the number of training iterations.

method	comm. cost	uses 2nd order info.	performs local updates
DP-FedNew	$d \times T$	yes	no
DP-FedNew-FC	$d \times T$	yes	no
DP-FedFC	$d_x^2 + d \times T$	yes	no
DP-Scaffold	$2d \times T$	no	yes
DP-FedGD	$d \times T$	no	no

B More Details on the Privacy Analysis

In this work, we provide an accurate (ε, δ)-analysis for our methods using the hockey-stick divergence. This way, we are able to get optimal privacy parameters for a given sensitivity analysis of the data-dependent functions and in particular we obtain lower bounds than using, e.g., the Rényi differential privacy (RDP) (Mironov, 2017) which is a commonly used alternative.

We next shortly describe the mathematical results needed for obtaining accurate (ε, δ)-DP bounds using the hockey-stick divergence.... The (ε, δ)-DP as defined in 1 can be characterized using the hockey-stick divergence as follows. For $\alpha > 0$ the hockey-stick divergence H_α from a distribution P to a distribution Q is defined as

$$H_\alpha(P||Q) = \int \left[P(t) - \alpha \cdot Q(t)\right]_+ \, \mathrm{d}t,$$

where for $t \in \mathbb{R}$, $[t]_+ = \max\{0, t\}$. Tight (ε, δ)-values for a given mechanism can be obtained using the hockey-stick-divergence:

Lemma B.1 (Zhu et al. 2022)**.** *For a given $\varepsilon \geq 0$, tight $\delta(\varepsilon)$ is given by the expression*

$$\delta(\varepsilon) = \max_{D \sim D'} H_{\mathrm{e}^\varepsilon}(\mathcal{M}(D)||\mathcal{M}(D')).$$

Thus, if we can bound the divergence $H_{\mathrm{e}^\varepsilon}(\mathcal{M}(D)||\mathcal{M}(D'))$ accurately, we also obtain accurate $\delta(\varepsilon)$-bounds. To this end we need to consider so-called dominating pairs of distributions:

Definition B.2 (Zhu et al. 2022)**.** *A pair of distributions (P, Q) is a* dominating pair *of distributions for mechanism $\mathcal{M}(D)$ if for all neighboring datasets D and D' and for all $\alpha > 0$,*

$$H_\alpha(\mathcal{M}(D)||\mathcal{M}(D')) \leq H_\alpha(P||Q).$$

If the equality holds for all α for some D, D', then (P, Q) is a tightly dominating pair of distributions.

When analyzing iterative DP-FL training methods, we model them as adaptive compositions such that the adversary has a view on the output of all intermediate outputs. This means that we analyze mechanisms of the form

$$\mathcal{M}^{(T)}(D) = \big(\mathcal{M}_1(D), \mathcal{M}_2(\mathcal{M}_1(D), D), \ldots, \mathcal{M}_T(\mathcal{M}_1(D), \ldots, \mathcal{M}_{T-1}(D), D)\big). \tag{B.1}$$

In the methods we propose, each $\mathcal{M}_i$, $i \in [T]$, will correspond to a Gaussian mechanism with a given sensitivity and noise scale. We next describe in detail how to obtain accurate bounds for compositions of Gaussian mechanisms.

We get upper bounds for adaptive compositions using the dominating pairs of distributions as follows:

Theorem B.3 (Zhu et al. 2022)**.** *If (P, Q) dominates $\mathcal{M}$ and (P', Q') dominates $\mathcal{M}'$, then $(P \times P', Q \times Q')$ dominates the adaptive composition $\mathcal{M} \circ \mathcal{M}'$.*

To convert the hockey-stick divergence from $P \times P'$ to $Q \times Q'$ into an efficiently computable form, we consider so called privacy loss random variables.

Definition B.4. *Let P and Q be probability density functions. We define the privacy loss random variable (PRV) $\omega_{P/Q}$ as*

$$\omega_{P/Q} = \log \frac{P(t)}{Q(t)}, \quad t \sim P(t).$$

PRVs can be utilized for obtaining accurate privacy guarantees via the following result.

Theorem B.5 (Gopi et al. 2021)**.** *The $\delta(\varepsilon)$-bounds can be represented using the following representation that involves the PRV:*

$$H_{\mathrm{e}^{\varepsilon}}(P||Q) = \mathop{\mathbb{E}}_{s \sim \omega_{P/Q}} \left[1 - \mathrm{e}^{\varepsilon - s}\right]_{+}. \tag{B.2}$$

Moreover, if $\omega_{P/Q}$ is the PRV for the pair of distributions (P, Q) and $\omega_{P'/Q'}$ the PRV for the pair of distributions (P', Q'), then the PRV for the pair of distributions $(P \times P', Q \times Q')$ is given by $\omega_{P/Q} + \omega_{P'/Q'}$.

Given a dominating pair of distributions (P, Q) for a mechanism $\mathcal{M}$, B.5 is all that is needed for obtaining (ε, δ)-bounds for $\mathcal{M}$. In some cases, such as in the case of the Gaussian mechanism, this expression leads to analytical bounds Balle & Wang (see, e.g., 2018). In the general case, Fast Fourier Technique-based methods (Koskela et al., 2021; Gopi et al., 2021) can be used to numerically evaluate the convolutions appearing when summing the PRVs and evaluating the expression B.2.

In this work, the methods we propose are based on additive Gaussian noise and the privacy analysis is equivalent to that of the Gaussian mechanism.

Hockey-stick divergence between two Gaussians. Let $x_0, x_1 \in \mathbb{R}$, $\sigma \geq 0$, and let P be the density function of $\mathcal{N}(x_0, \sigma^2)$ and Q the density function of $\mathcal{N}(x_1, \sigma^2)$. Then, the PRV $\omega_{P/Q}$ is distributed as (Lemma 11 by Sommer et al., 2019)

$$\omega_{P/Q} \sim \mathcal{N}\left(\frac{(x_0 - x_1)^2}{2\sigma^2}, \frac{(x_0 - x_1)^2}{\sigma^2}\right). \tag{B.3}$$

Thus, in particular: $H_\alpha(P||Q) = H_\alpha(Q||P)$ for all $\alpha > 0$. Plugging in PLD $\omega_{P/Q}$ to the expression (B.2), we find that for all $\varepsilon \geq 0$, the hockey-stick divergence $H_{\mathrm{e}^{\varepsilon}}(P||Q)$ is given by the expression

$$\delta(\varepsilon) = \Phi\left(-\frac{\varepsilon\sigma}{\Delta} + \frac{\Delta}{2\sigma}\right) - e^{\varepsilon}\Phi\left(-\frac{\varepsilon\sigma}{\Delta} - \frac{\Delta}{2\sigma}\right), \tag{B.4}$$

where Φ denotes the CDF of the standard univariate Gaussian distribution and $\Delta = |x_0 - x_1|$. This formula was originally given by Balle & Wang (2018).

If $\mathcal{M}$ is of the form $\mathcal{M}(D) = f(D) + Z$, where $f : \mathcal{D}^N \rightarrow \mathbb{R}^d$ and $Z \sim \mathcal{N}(0, \sigma^2 I_d)$, and $\Delta = \max_{D \simeq D'} \|f(D) - f(D')\|_2$ gives the L_2-sensitivity, then for $x_0 = 0$, $x_1 = \Delta$, (P, Q) of the above form gives a tightly dominating pair of distributions for $\mathcal{M}$ (Zhu et al., 2022). Subsequently, by Theorem B.5, $\mathcal{M}$ is (ε, δ)-DP for $\delta(\varepsilon)$ given by (B.4).

It also directly follows from Theorem B.5 and the form of the PRV (B.3) that the PRV for the adaptive composition of T Gaussian mechanisms is given by

$$\omega_{P/Q} \sim \mathcal{N}\left(\frac{T \cdot \Delta^2}{2\sigma^2}, \frac{T \cdot \Delta^2}{\sigma^2}\right)$$

and we obtain the following expression.

Lemma B.6. *Consider an adaptive composition of T Gaussian mechanisms, each with L_2-sensitivity Δ and noise scale parameter σ. The adaptive composition is (ε, δ)-DP for*

$$\delta(\varepsilon) = \Phi\left(-\frac{\varepsilon\sigma}{\sqrt{T} \cdot \Delta} + \frac{\sqrt{T} \cdot \Delta}{2\sigma}\right) - e^{\varepsilon}\Phi\left(-\frac{\varepsilon\sigma}{\sqrt{T} \cdot \Delta} - \frac{\sqrt{T} \cdot \Delta}{2\sigma}\right).$$

C More Details of Non-Private FedNew

We first shortly describe Newton's method which FedNew approximates with fast communication. Let n denote the number of users and $f_i(\theta)$ the empirical loss of user i, $i \in [n]$, where $\theta \in \mathbb{R}^d$ denotes the model parameters. We consider the minimization problem

$$\min_{\theta \in \mathbb{R}^d} f(\theta) := \sum\nolimits_{i=1}^n f_i(\theta).$$

The Newton iteration which is the basis for most of the second-order methods, is given as

$$\theta^{k+1} = \theta^k - \left(\sum\nolimits_{i=1}^n \nabla^2 f_i(\theta^k) \right)^{-1} \sum\nolimits_{i=1}^n \nabla f_i(\theta^k).$$

Straightforward FL approaches suffer from a very high communication cost due to the possible communication of the Hessians. To this end, we consider the FedNew method which has only $O(d)$ user-wise communication cost per iteration.

The update $(\sum_{i=1}^n \nabla^2 f_i(\theta^k))^{-1} \sum_{i=1}^n \nabla f_i(\theta^k)$ in the Newton iteration is approximated such that the ADMM algorithm is applied to the augmented Lagrangian

$$\mathcal{L}(\{y_i, \lambda_i\}_{i=1}^n, y) = \sum_{i=1}^n \frac{1}{2} y_i^T (H_i^k + \alpha I) y_i - y_i^T g_i^k + \sum_{i=1}^n \langle \lambda_i, y_i - y \rangle + \frac{\rho}{2} \sum_{i=1}^n \|y_i - y\|_2^2,$$

where y_i's denote the so-called primal variables and λ_i's the so-called dual variables, $H_i^k = \nabla^2 f_i(\theta^k)$, $g_i^k = \nabla f_i(\theta^k)$ and θ^k are the global model parameters at iteration k. We refer to (Elgabli et al., 2022) for more details on the derivation of the FedNew method, and simply list here the resulting algorithm.

In case of convergence of the local iterations (repeating steps 1 to 3 until convergence), the following conditions are satisfied by the FedNew iteration for all $i \in [n]$ (see Elgabli et al., 2022, and the references therein):

$$\begin{aligned} & y_i^*(\theta^k) = y^*(\theta^k), \\ & (H_i^k + \alpha I) y_i^*(\theta^k) - g_i^k + \lambda_i^*(\theta^k) = 0, \end{aligned} \tag{C.1}$$

where $y_i^*(\theta^k)$ and $\lambda_i^*(\theta^k)$ denote the optimal values of y_i^k and λ_i^k, respectively, at iteration k, i.e., the results of running the ADMM steps until the end at the iteration k.

D Detailed Description of DP-FedFC and DP-FedGD Algorithms

For completeness, we describe here in detail the baseline algorithms.

Algorithm 2 Record-level full-batch DP-SCAFFOLD (modification of the user-level algorithm by Noble et al., 2022)

```
Input: dataset $D = \{D_i\}_{i=1}^n$, noise level $\sigma$, clipping constant $C$, training length $T$, number of local steps $M$, local and global learning rates $\eta_l$ and $\eta_g$, initial $c_i^0$.
for iteration $k = 1, \dots, T$ do
    Server: Send $(\theta^{k-1}, c^{k-1})$ to all users.
    for user $i = 1, \dots, n$ do
        Initialize model: $y_i^0 = \theta^{k-1}$.
        for local step $m = 1, \dots, M$ do
            Clients: Add DP noise to local gradients:
            $\tilde{g}_i^m = \frac{1}{|D_i|} \sum_{x \in D_i} \mathrm{clip}_C\big(\nabla f(x, \theta^m)\big) + \frac{2C}{|D_i|} \mathcal{N}(0, \sigma^2)$
            $y_i^m = y_i^{m-1} - \eta_l(\tilde{g}_i^m - c_i^{k-1} + c^{k-1})$
        end for
        Update user control variables:
        $\tilde{c}_i^k = c_i^{k-1} - c^{k-1} + \frac{(\theta^{k-1} - y_i^M)}{M\eta_l}$
        $(\Delta y_i^k, \Delta c_i^k) = (y_i^M - \theta^{k-1}, \tilde{c}_i^k - c_i^{k-1})$
        Share $(\Delta y_i^k, \Delta c_i^k)$ with server.
        $c_i^k = \tilde{c}_i^k$
    end for
    Server: $(\Delta\theta^k, \Delta c^k) = \frac{1}{n} \sum_{i=1}^n (\Delta y_i^k, \Delta c_i^k)$
    Server: Global model update, $\theta^k = \theta^{k-1} + \eta_g \Delta\theta^k$.
    Server: Update global control variable, $c^k = c^{k-1} + \Delta c^k$.
end for
```

Algorithm 3 Record-level DP-FedFC Algorithm (Mehta et al., 2023)

```
Input: dataset $D = \{D_i\}_{i=1}^n$, noise levels $\sigma_c, \sigma_g$, clipping constants $C_c, C_g$, training length $T$, learning rate $\eta$, regularization parameter $\gamma$.
for user $i = 1, \dots, n$ do
    Clip local user inputs: $\widetilde{D}_i = \big[\mathrm{clip}_{C_c}(x_1) \quad \dots \quad \mathrm{clip}_{C_c}(x_{|D_i|})\big]$.
    Compute local noisy feature covariance matrix: $\mathcal{C}_i = \widetilde{D}_i \widetilde{D}_i^T + E_i, E_i \sim \mathcal{N}\big(0, \frac{I_d (C_c^2 \sigma_c)^2}{n}\big)$
    Share $\mathcal{C}_i$ with server.
end for
Server aggregates $\{\mathcal{C}_i\}_{i=1}^n$, computes the global noisy convariance matrix $\mathcal{C} = \frac{\sum_i^n \mathcal{C}_i}{n} + \gamma I_d$.
for iteration $k = 1, \dots, T$ do
    for user $i = 1, \dots, n$ do
        Clip and perturb local gradients:
        $$u_i^k = \frac{\sum_{x \in D_i} \mathrm{clip}_{C_g}\big(\nabla f(\theta^k, x)\big) + E_i^k}{|D_i|}, E_i^k \sim \mathcal{N}\Big(0, \frac{I_d (C_g \sigma_g)^2}{n}\Big).$$
        Share local update $u_i^k$ with server.
    end for
    Server aggregates $\{u_i^k\}_{i=1}^n$, computes the global noisy update $U^k = \mathcal{C}^{-1} \cdot \Big(\frac{\sum_{i=1}^n u_i^k}{n}\Big)$.
    Update model parameters: $\theta^{k+1} = \theta^k - \eta \cdot U^k$.
end for
```

Algorithm 4 User-level DP-FedFC Algorithm (modification of the user-level algorithm by Mehta et al., 2023)

```
Input: dataset $D = \{D_i\}_{i=1}^n$, noise levels $\sigma_c, \sigma_g$, clipping constants $C_c, C_g$, training length $T$, learning rate $\eta$, regularization parameter $\gamma$.
for user $i = 1, \dots, n$ do
  Clip local covariance matrices: $\tilde{\mathcal{C}}_i = \text{clip}_{C_c}(D_i.D_i^T)$.
  Compute local noisy feature covariance matrix: $\mathcal{C}_i = \tilde{\mathcal{C}}_i + E_i, E_i \sim \mathcal{N}\big(0, \frac{I_d(C_c\sigma_c)^2}{n}\big)$
  Share $\mathcal{C}_i$ with server.
end for
Server aggregates $\{\mathcal{C}_i\}_{i=1}^n$, computes global noisy convariance matrix $\mathcal{C} = \frac{\sum_i^n \mathcal{C}_i}{n} + \gamma I_d$, and shares it back with clients.
for iteration $k = 1, \dots, T$ do
  for user $i = 1, \dots, n$ do
    Compute and average local gradients:
      $g_i^k = \frac{\sum_{x \in D_i} \nabla f(\theta^k, x)}{|D_i|}.$
    Clip and perturb local updates multiplied with a preconditioner:
      $u_i^k = \text{clip}_{C_g}(\mathcal{C}^{-1} \cdot g_i^k) + E_i^k, E_i^k \sim \mathcal{N}\Big(0, \frac{I_d(C_g\sigma_g)^2}{n}\Big).$
    Share noisy update $u_i^k$ with server.
  end for
  Server aggregates $\{u_i^k\}_{i=1}^n$, and computes global noisy update $U^k = \frac{\sum_{i=1}^n u_i^k}{n}$.
  Update model parameters: $\theta^{k+1} = \theta^k - \eta \cdot U^k$.
end for
```

Algorithm 5 Record-level DP-FedGD Algorithm

```
Input: dataset $D = \{D_i\}_{i=1}^n$, noise levels $\sigma_g$, clipping constants $C_g$, training length $T$, learning rate $\eta$.
for iteration $k = 1, \dots, T$ do
  for user $i = 1, \dots, n$ do
    Clip and perturb the avg. of local gradients:
      $u_i^k = \frac{\sum_{x \in D_i} \text{clip}_{C_g}\big(\nabla f(\theta^k, x)\big) + E_i^k}{|D_i|}, \quad E_i^k \sim \mathcal{N}\Big(0, \frac{I_d(C_g\sigma_g)^2}{n}\Big).$
    Share $u_i^k$ with server.
  end for
  Server aggregates $\{u_i^k\}_{i=1}^n$, computes the global noisy update $U^k = \frac{\sum_{i=1}^n u_i^k}{n}$.
  Update model parameters: $\theta^{k+1} = \theta^k - \eta \cdot U^k$.
end for
```

Algorithm 6 User-level DP-FedGD Algorithm

```
Input: dataset $D = \{D_i\}_{i=1}^n$, noise levels $\sigma_g$, clipping constants $C_g$, training length $T$, learning rate $\eta$.
for iteration $k = 1, \dots, T$ do
  for user $i = 1, \dots, n$ do
    Compute and average local gradients:
      $$g_i^k = \frac{\sum_{x \in D_i} \nabla f(\theta^k, x)}{|D_i|}.$$
    Clip and perturb local updates:
      $$u_i^k = \mathrm{clip}_{C_g}(g_i^k) + E_i^k, \quad E_i^k \sim \mathcal{N}\Big(0, \frac{I_d (C_g \sigma_g)^2}{n}\Big).$$
    Share noisy update $u_i^k$ with server.
  end for
  Server aggregates $\{u_i^k\}_{i=1}^n$, and computes global noisy update $U^k = \frac{\sum_{i=1}^n u_i^k}{n}$.
  Update model parameters: $\theta^{k+1} = \theta^k - \eta \cdot U^k$.
end for
```

E More Details of Private FedNew

E.1 DP-FedNew with User-level Privacy

For user-level privacy, we need to hide users i whole contribution. We can obtain this simply by clipping the user-wise updates and adding normally distributed noise (similar user-level algorithms considered, e.g., in McMahan et al., 2018a; Ponomareva et al., 2023). This means that we simply replace the local update (3.1) in FedNew by the pseudocode of Algorithm 7, where we clip $\widehat{y}_i^k$ with some constant $C > 0$ and add normally distributed noise with covariance $C^2 \sigma^2 I_d$, $\sigma > 0$, to the resulting clipped update $\mathrm{clip}_C(\widehat{y}_i^k)$, where the clipping function is defined for vectors and matrices as

$$\mathrm{clip}_C(\widehat{y}_i^k) = \begin{cases} \widehat{y}_i^k, & \text{if} \quad \|\widehat{y}_i^k\|_2 \leq C, \\ C \cdot \frac{\widehat{y}_i^k}{\|\widehat{y}_i^k\|_2}, & \text{else.} \end{cases}$$

From the differentially privacy point of view, at each iteration we release only the noisy sum of the local updates,

$$\mathcal{M}(D) \sim \sum_{i=1}^n \mathrm{clip}_C(\widehat{y}_i^k) + \mathcal{N}(0, n \cdot C^2 \sigma^2 I_d). \tag{E.1}$$

Algorithm 7 User-level DP-FedNew algorithm to compute private y_i^k.

```
Input: clipping constant $C > 0$, noise parameter $\sigma > 0$, number of iterations $T$, regularization parameter $\gamma$.
for iteration $k = 1, \dots, T$ do
  for user $i = 1, \dots, n$ do
    Compute the non-DP update of the primal variable:
      $$\widehat{y}_i^k = (H_i^k + \alpha I + \rho I)^{-1}(g_i^k - \lambda_i^{k-1} + \rho y^{k-1}).$$
    Clip and perturb the primal variable:
      $$\widetilde{y}_i^k \leftarrow \mathrm{clip}_C(\widehat{y}_i^k) + E_i, \quad E_i \sim \mathcal{N}\left(0, \frac{C^2 \sigma^2}{n} I_d\right).$$
  end for
end for
```

F Proof of Lemma 3 (Record-Level Sensitivity Bound)

For the proof of Lemma 3 we first need the following auxiliary lemma:

Lemma F.1. *Suppose $A, B \in \mathbb{R}^{d\times d}$ are positive definite, $\|\cdot\|$ is a matrix norm and $\|A-B\|\|A^{-1}\| < 1$. Then*

$$\|A^{-1} - B^{-1}\| \le \frac{\|A-B\|\|A^{-1}\|^2}{1 - \|A-B\|\|A^{-1}\|}.$$

Proof. The proof can be found in Section 5.8 of (Horn & Johnson, 2012) (see also Lemma C.4 by Ganesh et al. (2023)). □

Lemma F.2. *Let Δ_i be defined as in* (3). *Let Δ_H be an upper bound for the norm of H, i.e., an upper bound for the norm of data-sample-wise Hessian. Assume*

$$\begin{aligned} \|\nabla^2 f(x,\theta^k)\|_2 &\le \Delta_H \quad \textit{for all } x \in D_i, \\ \|\nabla f(x,\theta^k)\|_2 &\le C_1 \quad \textit{for all } x \in D_i, \\ \|g_i^k - \lambda_i^{k-1} + \rho y^{k-1}\| &\le C_2, \end{aligned} \tag{F.1}$$

and

$$\gamma > \frac{\Delta_H}{|D_i|}. \tag{F.2}$$

Then, we have:

$$\|\Delta_i\|_2 \le \frac{1}{\gamma\cdot|D_i|}\cdot C_1 + \frac{\Delta_H}{\gamma^2\cdot|D_i| - \gamma\cdot\Delta_H}\cdot C_2,$$

where $|D_i|$ is the size of the local dataset D_i.

Proof. For ease of notation, consider the function

$$f(D_i,\gamma,\theta) = (H_i^k + \gamma I)^{-1}(g_i^k + \theta),$$

where $\gamma > 0$ is a constant and θ stands for auxiliary variables. We need to bound the 2-norm of

$$\Delta_i = f(D_i', \gamma, \theta) - f(D_i, \gamma, \theta),$$

where $D_i' = D_i \bigcup \{x'\}$ for some data-element x'. Adding and subtracting $(H_i^k + H' + \gamma I)^{-1}(g_i^k + \theta)$ to Δ_i, we have:

$$\begin{aligned} \Delta_i &= (H_i^k + H' + \gamma I)^{-1}(g_i^k + g' + \theta) - (H_i^k + \gamma I)^{-1}(g_i^k + \theta) \\ &= (H_i^k + H' + \gamma I)^{-1}(g_i^k + g' + \theta) - (H_i^k + H' + \gamma I)^{-1}(g_i^k + \theta) \\ &+ (H_i^k + H' + \gamma I)^{-1}(g_i^k + \theta) - (H_i^k + \gamma I)^{-1}(g_i^k + \theta) \\ &= (H_i^k + H' + \gamma I)^{-1}g' + \big((H_i^k + H' + \gamma I)^{-1} - (H_i^k + \gamma I)^{-1}\big)(g_i^k + \theta). \end{aligned} \tag{F.3}$$

For the first term on the right-hand side of (F.3) we use the following fact: if a matrix A is positive definite with smallest eigenvalue $\lambda_{\min}$, then $\|A^{-1}\| = \lambda_{\min}^{-1}$. Clearly, since H_i^k and H' are positive semidefinite, $(H_i^k + H' + \gamma I)^{-1}$ is positive definite with smallest eigenvalue larger than γ, and we have that

$$\begin{aligned} \|(H_i^k + H' + \gamma I)^{-1}g'\|_2 &\le \|(H_i^k + H' + \gamma I)^{-1}\|_2\|g'\|_2 \\ &\le \frac{1}{\gamma}\|g'\|_2 \le \frac{1}{\gamma}\cdot\frac{C_1}{|D_i|}. \end{aligned}$$

By Lemma F.1 and the assumptions (F.1) and (F.2) we have that

$$\begin{aligned}
\|\big((H_i^k + H' + \gamma I)^{-1} - (H_i^k + \gamma I)^{-1}\big)(g_i^k + \theta)\|_2 &\leq \|(H_i^k + H' + \gamma I)^{-1} - (H_i^k + \gamma I)^{-1}\|_2 \|g_i^k + \theta\|_2 \\
&\leq \frac{\|H'\|_2 \|(H_i^k + H' + \gamma I)^{-2}\|_2}{1 - \|(H_i^k + H' + \gamma I)^{-1}\|_2 \|H'\|_2} \cdot C_2 \\
&\leq \frac{\|H'\|_2 \cdot \gamma^{-2}}{1 - \gamma^{-1}\|H'\|_2} \cdot C_2 \\
&\leq \frac{\frac{\Delta_H}{|D_i|}\gamma^{-2}}{1 - \gamma^{-1}\frac{\Delta_H}{|D_i|}} \cdot C_2 \\
&= \frac{\Delta_H}{\gamma^2 \cdot |D_i| - \gamma \Delta_H} \cdot C_2.
\end{aligned}$$

□

G Proof of Lemma 4

Lemma G.1. *Let $a, b \in \mathbb{R}^n$ and $C > 0$. If we set*

$$\xi = \frac{-2\langle a, \frac{b}{\|b\|_2}\rangle + \sqrt{4\langle a, \frac{b}{\|b\|_2}\rangle^2 + 4(C^2 - \|a\|_2^2)}}{2\|b\|_2},$$

we have that

$$\|a + \xi \cdot b\|_2 = C.$$

Proof. Denote by $\hat{b}$ a unit vector in the direction of b. Setting the right-hand side of

$$\|a + b\|_2^2 = \|a\|_2^2 + 2\|b\|_2\langle a, \hat{b}\rangle + \|b\|_2^2$$

equal to C^2 and solving the quadratic equation for $\|b\|_2$, we arrive at the claim.

□

H Asymptotic Convergence Analysis for DP-FedNew

For the convergence analysis of DP-FedNew, we assume that the clipping constants are chosen such that no clipping happens during the iteration. This is a natural assumption, e.g., in case the gradients are Lipschitz, the per-example Hessian is bounded in Frobenius norm (plausible assumption, e.g., for generalized linear models) and the auxiliary terms in the local update, i.e., $g_i^k - \lambda_i^{k-1} + \rho y^{k-1}$ stays bounded along the iteration. Then, the noisy update can be written as

$$\widetilde{y}_i^k = (H_i^k + \alpha I + \rho I)^{-1}(g_i^k - \lambda_i^{k-1} + \rho y^{k-1}) + E_i^k, \tag{H.1}$$

where $E_i^k \sim \mathcal{N}(0, \frac{C^2\sigma^2}{n} I_d)$ and C is the sensitivity parameter, i.e., either the clipping constant in case of user level algorithm or then the parameter given by Lemma 3 in case of record-level algorithm. Without loss of generality of our results, we also assume that $C = 1$.

The following auxiliary result applies for the noisy update rule (H.1) and is central in our analysis.

Lemma H.1. *Consider one iteration of DP-FedNew. Assume the per-example approximations of the Hessian at user i at iteration k, H_i^k, is positive semidefinite. Denote by $\lambda^{*,k}$ and $y^{k,*}$ the dual variables that are the results of running the non-DP FedNew inner iteration until the end (given the results of the DP iterations from $k-1$ iterations). Denote the dual residual $s^k := \rho(y^k - y^{k-1})$. We have:*

$$\begin{aligned}
\mathbb{E}\langle \lambda_i^k + s^k - \lambda^{k,*}, \widetilde{y}_i^k - y^{k,*}\rangle &\leq -\alpha \mathbb{E}\|y^{k,*} - \widetilde{y}_i^k\|^2 \\
&+ \sigma^2 \cdot \mathrm{Trace}(H_i^k) + \sigma^2 \cdot (\alpha + \rho) \cdot d,
\end{aligned}$$

where the expectation is taken over the randomness of E_i^k, the noise added by the user i at iteration k.

In the following result, we define a certain Lyapunov function for the DP-FedNew algorithm, and by using the auxiliary Lemma H.8, we obtain a stochastic inequality which leads us to the main result.

Lemma H.2. *Let the Lyapunov function V_k be defined as*

$$\begin{aligned} V_k := \frac{1}{\rho}\sum_{i=1}^{n} \|\lambda_i^k - \lambda^{k,*}\|_2^2 + 2\beta_1 \sum_{i=1}^{n} \|\widetilde{y}_i^k - y^{k,*}\|_2^2 \\ + \rho n \|y^k - y^{k,*}\|_2^2 + 2\rho n \|y^k - y^{k-1}\|_2^2, \end{aligned} \tag{H.2}$$

where $\widetilde{y}_i^k$ denotes the noisy update (H.1) *Denote $\widetilde{V}_k = \mathbb{E} V_k$, where the expectation is taken over all additive noises up to iteration k. Then, $\widetilde{V}_k$ satisfies*

$$\widetilde{V}_k \leq \widetilde{V}_{k-1} - \beta_2 \mathbb{E} \sum_{i=1}^{n} \|\widetilde{y}_i^k - y^{k,*}\|^2 + \sigma^2 \cdot (\alpha + \rho) \cdot d$$

for some constant $\beta_2 > 0$, where the expectation is taken over the noise added at iteration k.

Our main convergence result is of qualitative nature and states that the DP version inherits the stability of FedNew in a sense that the added DP noise does not make the solution to diverge from the non-private iterations. The result follows from Lemma H.2.

Theorem H.3. *Let $\sigma > 0$. For all $k \in \mathbb{Z}$, there exists $\ell > k$ such that*

$$\mathbb{E}\|y^\ell - y^{\ell,*}\|^2 \leq \frac{\sigma^2 \cdot (\alpha + \rho) \cdot d}{n\beta_2}$$

where the expectation is taken over the noise added at iteration ℓ.

H.1 Non-Private Analysis by Elgabli et al. (2022)

To make following the DP convergence analysis easier to follow, we review here the main results of (Elgabli et al., 2022) and depict the main story of their analysis.

Consider the non-private FedNew algorithm, i.e., the variables y and λ are those given by the algorithm described in Section 3.

The convergence analysis of (Elgabli et al., 2022) is starts with the following auxiliary lemma.

Lemma H.4. *Consider one iteration of FedNew. Assume the per-example approximations of the Hessian at user i, H_i^k, is positive semidefinite. Denote the dual residual $s^k := \rho(y^k - y^{k-1})$. We have:*

$$\mathbb{E}\langle \lambda_i^k + s^k - \lambda^{k,*}, y_i^k - y^{k,*} \rangle \leq -\alpha \mathbb{E}\|y^{k,*} - y_i^k\|^2 + \sigma^2 \cdot \mathrm{Trace}(H_i^k).$$

Next, the inequality given in Lemma H.4 is reformulated to obtain the inequality of Lemma H.5 below. This reformulation requires, however, two additional assumptions. First, it is assumed that for the function $Q_i(\theta, y) = \frac{1}{2} y (\nabla^2 f_i(\theta) + \alpha I) y - y^T \nabla f_i(\theta)$ we have that

$$\|\nabla_y Q_i(\theta_1, y_1) - \nabla_y Q_i(\theta_2, y_2)\|_2 \leq L_q \|y_1 - y_2\|_2 \tag{H.3}$$

for some constant $L_q > 0$. We remark that the condition (H.3) this is a fairly strong requirement. However, one can easily show that this holds for the linear regression, for example, since then $\nabla^2 f_i(\theta)$ independent of θ for all $i \in [n]$. Another requirement for Lemma H.5 is that the iterates of FedNew satisfy the inequality

$$\|y^k - y^{k-1}\|_2 \leq \|y^k - y^{k,*}\|_2. \tag{H.4}$$

As we show in detail below in Section H.2, this inequality is satisfied for large enough values of the regularization parameter ρ.

With these assumption the following technical result is shown next. This will lead us to formulating a Lyapunov function for the iteration.

Lemma H.5. *Assume the conditions* (H.3) *and* (H.4) *hold true for all* $k \in [T]$. *For any* $\beta \leq \alpha - 2.5\rho - \frac{8L_q^2 n}{\rho}$, *the iterates of FedNew satisfy the inequality*

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{k,*}\|_2^2 + 2\beta\sum_{i=1}^{n}\|y_i^k - y^{k,*}\|_2^2 + \rho n\|y^k - y^{k,*}\|_2^2 + 2\rho n\|y^k - y^{k-1}\|_2^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{k-1,*}\|_2^2 \\
&+ \frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1} - y^{k-1,*}\|_2^2 + \frac{4L_q^2 n}{\rho}\|y^{k-1} - y^{k-1,*}\|_2^2 \\
&+ 2\rho n\|y^{k-1} - y^{k-2}\|_2^2.
\end{aligned}$$

From Lemma (H.5) if follows that the Lyapunov function V_k defined as

$$\begin{aligned}
V_k := \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda^{k,*}\|_2^2 + 2\beta_1\sum_{i=1}^{n}\|y_i^k - y^{k,*}\|_2^2 \\
+ \rho n\|y^k - y^{k,*}\|_2^2 + 2\rho n\|y^k - y^{k-1}\|_2^2
\end{aligned} \tag{H.5}$$

satisfies the inequality

$$V_k \leq V_{k-1} - \beta_2\sum_{i=1}^{n}\|y_i^k - y^{k,*}\|^2 \tag{H.6}$$

for some constant $\beta_2 > 0$.

The main result of (Elgabli et al., 2022) follows from the inequality (H.6). We give an alternative proof for it, to also motivate the proof of our DP result.

Theorem H.6. *As the number of iterations* $k \to \infty$, *the local ADMM iterates approach the Newton updates, i.e.,*

$$\|y_i^k - y^{k,*}\|_2 \to 0.$$

Proof. From Lemma H.5 it follows that the Luapynov function V_k defined in (H.5) satisfies the inequality (H.6). Since V_k is non-negative and monotonously decreasing, by the monotone convergence theorem it has a limit as $k \to \infty$. Thus, $\beta_2\sum_{i=1}^{n}\|y_i^k - y^{k,*}\|^2 \to 0$ as $k \to \infty$ from which the claim follows. □

H.2 Assumption Used in Thm. H.12

In the proof of Theorem H.12 we need to assume that the iterates of the non-private FedNew algorithm satisfy

$$\|y^k - y^{k-1}\|_2 \leq \|y^k - y^{k,*}\|_2. \tag{H.7}$$

for all $k \in [T]$. As the following result shows, this in a reasonable assumption for a large enough ρ.

Lemma H.7. *There exists* $\rho_0 > 0$ *such that for all* $\rho \geq \rho_0$, *the assumption* (H.7) *holds true.*

Proof. Recall that

$$\begin{aligned}
y^k &= \frac{1}{n}\sum_{i=1}^{n} y_i^k \\
&= \frac{1}{n}\sum_{i=1}^{n}(H_i^k + \alpha I + \rho I)^{-1}(g_i^k - \lambda_i^{k-1} + \rho \cdot y^{k-1}).
\end{aligned}$$

Therefore

$$y^k - y^{k-1} = \frac{1}{n}\sum_{i=1}^{n}\Big[(H_i^k + \alpha I + \rho I)^{-1}\big(g_i^k - \lambda_i^{k-1} + \rho \cdot y^{k-1} \\ - (H_i^k + \alpha I + \rho I)y^{k-1}\big)\Big] \\ = \frac{1}{n}\sum_{i=1}^{n}(H_i^k + \alpha I + \rho I)^{-1}\big(g_i^k - \lambda_i^{k-1} - (H_i^k + \alpha I)y^{k-1}\big)$$

which shows that $\|y^k - y^{k-1}\|_2 \to 0$ as $\rho \to \infty$. On the other hand,

$$\begin{aligned} y^k - y^{k,*} &= \frac{1}{n}\sum_{i=1}^{n}\Big[(H_i^k + \alpha I + \rho I)^{-1}\big(g_i^k - \lambda_i^{k-1} + \rho \cdot y^{k-1}\big) \\ &\quad - (H_i^k + \alpha I)\big(g_i^k - \lambda_i^*(\theta^k)\big)\big)\Big] \\ &= \frac{1}{n}\sum_{i=1}^{n}\Big[y^{k-1} + (H_i^k + \alpha I + \rho I)^{-1}\big(g_i^k - \lambda_i^{k-1} - (H_i^k + \alpha I)y^{k-1}\big) \\ &\quad - (H_i^k + \alpha I)\big(g_i^k - \lambda_i^*(\theta^k)\big)\big)\Big]. \end{aligned} \tag{H.8}$$

since

$$y_i^{k,*} = (H_i^k + \alpha I)^{-1}\big(g_i^k - \lambda_i^*(\theta^k)\big).$$

We see from (H.8) that

$$y^k - y^{k,*} \to \frac{1}{n}\sum_{i=1}^{n}\Big[y^{k-1} - (H_i^k + \alpha I)\big(g_i^k - \lambda_i^*(\theta^k)\big)\big)\Big]$$

as $\rho \to \infty$. Thus, □

H.3 First Step of the DP Convergence Analysis: Proof of Lemma H.8

The following result is a stochastic version of (Lemma 1, Elgabli et al., 2022) and applies for the noisy update rule (H.1).

Lemma H.8. *Consider one iteration of DP-FedNew. Assume the per-example approximations of the Hessian at user i at iteration k, H_i^k, is positive semidefinite. Denote the dual residual $s^k := \rho(y^k - y^{k-1})$. Assume in the added noise $C = 1$. We have:*

$$\mathbb{E}\langle \lambda_i^k + s^k - \lambda^{k,*}, \widetilde{y}_i^k - y^{k,*}\rangle \le -\alpha\mathbb{E}\|y^{k,*} - \widetilde{y}_i^k\|^2 \\ + C^2 \cdot \sigma^2 \cdot \mathrm{Trace}(H_i^k) + \sigma^2 \cdot (\alpha + \rho) \cdot d,$$

where the expectation is taken over the randomness of E_i^k, the noise added by the user i at iteration k.

Proof. It follows from the noisy update rule (H.1) that

$$(H_i^k + \alpha I)(\widetilde{y}_i^k - E_i^k) - g_i^k + \lambda_i^{k-1} + \rho(\widetilde{y}_i^k - y^{k-1}) - \rho E_i^k = 0. \tag{H.9}$$

Substituting the update of the dual variable

$$\lambda_i^{k-1} = \lambda_i^k - \rho(\widetilde{y}_i^k - y^k)$$

to Eq. (H.9) gives

$$(H_i^k + \alpha I)(\widetilde{y}_i^k - E_i^k) - g_i^k + \lambda_i^k + \rho(y^k - y^{k-1}) - \rho E_i^k = 0.$$

Recall the dual residual $s^k = \rho(y^k - y^{k-1})$. We get

$$\lambda_i^k + s^k = g_i^k - (H_i^k + \alpha I)(\widetilde{y}_i^k - E_i^k) + \rho E_i^k. \tag{H.10}$$

Recall that for the optimal values of the primal and dual variables of the non-DP dual iteration at the global iteration k for user i, ${y_i^*}^k$ and ${\lambda_i^*}^k$, respectively, we have that

$$\begin{aligned}{\lambda_i^*}^k &= g_i^k - (H_i^k + \alpha I){y_i^*}^k \\ &= g_i^k - (H_i^k + \alpha I){y^*}^k,\end{aligned}$$

since ${y_i^*}^k = {y^*}^k$. Subtracting ${\lambda_i^*}^k$ from both sides of Eq. (H.10), we get

$$\begin{aligned}\lambda_i^k + s^k - {\lambda^*}^k &= (H_i^k + \alpha I)(E_i^k - \widetilde{y}_i^k) + (H_i^k + \alpha I){y^*}^k + \rho E_i^k \\ &= (H_i^k + \alpha I)({y^*}^k - \widetilde{y}_i^k + E_i^k) + \rho E_i^k.\end{aligned} \tag{H.11}$$

where ${y^*}^k$ is the converged result of the non-DP dual iteration at global iteration k with Hessian evaluated at θ^k. Taking inner product of between both sides of Eq. (H.11) and the vector $\widetilde{y}_i^k - {y^*}^k$, we get

$$\begin{aligned}&\langle \lambda_i^k + s^k - {\lambda^*}^k, \widetilde{y}_i^k - {y^*}^k \rangle \\ &= \langle (H_i^k + \alpha I)({y^*}^k - \widetilde{y}_i^k + E_i^k), \widetilde{y}_i^k - {y^*}^k \rangle + \rho \langle E_i^k, \widetilde{y}_i^k - {y^*}^k \rangle \\ &= -\langle (H_i^k + \alpha I)(\widetilde{y}_i^k - {y^*}^k), \widetilde{y}_i^k - {y^*}^k \rangle + \langle (H_i^k + \alpha I)E_i^k, \widetilde{y}_i^k - {y^*}^k \rangle \\ &+ \rho \langle E_i^k, \widetilde{y}_i^k - {y^*}^k \rangle \\ &\leq -\alpha \|{y^*}^k - \widetilde{y}_i^k\|^2 + \langle (H_i^k + \alpha I)E_i^k, \widetilde{y}_i^k - {y^*}^k \rangle + \rho \langle E_i^k, \widetilde{y}_i^k - {y^*}^k \rangle\end{aligned} \tag{H.12}$$

since H_i^k is positive semidefinite. Recall:

$$\widetilde{y}_i^k = \widehat{y}_i^k + E_i^k,$$

where $\widehat{y}_i^k$ denotes the non-perturbed update, i.e.

$$\widehat{y}_i^k = (H_i^k + \alpha I + \rho I)^{-1}(g_i^k - \lambda_i^{k-1} + \rho y^{k-1}).$$

Thus, we can write (H.12) as

$$\begin{aligned}\langle \lambda_i^k + s^k - {\lambda^*}^k, \widetilde{y}_i^k - {y^*}^k \rangle \leq -\alpha \|{y^*}^k - \widetilde{y}_i^k\|^2 + \langle (H_i^k + \alpha I)E_i^k, \widehat{y}_i^k \\ + E_i^k - {y^*}^k \rangle + \rho \langle E_i^k, {y^*}^k - \widetilde{y}_i^k \rangle.\end{aligned} \tag{H.13}$$

Since $\mathbb{E}_{x \sim \mathcal{N}(0, I_d)} x^T A x = \mathrm{Trace}(A)$ for any square matrix A and since $E_i^k \sim \mathcal{N}(0, \sigma^2 I_d)$, we have

$$\begin{aligned}&\mathbb{E}\langle \lambda_i^k + s^k - {\lambda^*}^k, \widetilde{y}_i^k - {y^*}^k \rangle \\ &\leq -\alpha \mathbb{E}\|{y^*}^k - \widetilde{y}_i^k\|^2 + \mathbb{E}\langle (H_i^k + \alpha I)E_i^k, E_i^k \rangle + \mathbb{E}\langle (H_i^k + \alpha I)E_i^k, \widehat{y}_i^k - {y^*}^k \rangle \\ &+ \rho \mathbb{E}\langle E_i^k, \widetilde{y}_i^k - {y^*}^k \rangle \\ &= -\alpha \mathbb{E}\|{y^*}^k - \widetilde{y}_i^k\|^2 + \mathbb{E}\langle (H_i^k + \alpha I)E_i^k, E_i^k \rangle + \rho \mathbb{E}\langle E_i^k, \widehat{y}_i^k + E_i^k - {y^*}^k \rangle \\ &= -\alpha \mathbb{E}\|{y^*}^k - \widetilde{y}_i^k\|^2 + \sigma^2 \cdot \mathrm{Trace}(H_i^k + \alpha I) + \rho \cdot d \\ &= -\alpha \|{y^*}^k - \widetilde{y}_i^k\|^2 + \sigma^2 \cdot \mathrm{Trace}(H_i^k) + \sigma^2 \cdot (\alpha + \rho) \cdot d,\end{aligned}$$

where the expectation is taken over the randomness of E_i^k. □

In case f is β-smooth, we have the following corollary.

Corollary H.9. *If the loss function f is β-smooth, then*

$$\mathbb{E}\langle \lambda_i^k + s^k - {\lambda^*}^k, \widetilde{y}_i^k - {y^*}^k \rangle \leq -\mathbb{E}\alpha \|{y^*}^k - \widetilde{y}_i^k\|^2 + \sigma^2 \cdot (\alpha + \rho + \beta) \cdot d,$$

where the expectation is taken over the randomness of the local additive noise E_i^k.

Proof. If f is β-smooth, since it is twice differentiable, we have that $\|H_i^k\|_2 \leq \beta$. Since the trace of a square matrix equals the sum of its singular values, $\mathrm{Trace}(H_i^k) \leq d \cdot \beta$ and the claim follows from Lemma H.8. □

H.4 Additional Auxiliary Results

In addition to the conditions (H.3), we need the assumption that the iterates of DP-FedNew satisfy the inequality

$$\mathbb{E}\|y^k - y^{k-1}\|_2^2 \leq \mathbb{E}\|y^k - y^{k,*}\|_2^2 \tag{H.14}$$

for all $k \in [T]$, where the expectation is taken over the additive normally distributed noises at iteration k. This condition is true in case the (H.3) for the non-private FedNew is true, since $\mathbb{E}_X\|Y + X\|_2^2 = \|Y\|_2^2 + \mathbb{E}\|X\|_2^2$ for all random variables X with $\mathbb{E}X = 0$.

Lemma H.10. *Assume the conditions* (H.3) *and* (H.14) *are satisfied for all* $k \in [T]$. *For any* $\beta \leq \alpha - 2.5\rho - \frac{8L_q^2 n}{\rho}$, *the iterates of FedNew satisfy the inequality*

$$\begin{aligned}
&\frac{1}{\rho}\mathbb{E}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{k,*}\|_2^2 + 2\beta\mathbb{E}\sum_{i=1}^n \|\widetilde{y}_i^k - y^{k,*}\|_2^2 + \rho n\mathbb{E}\|y^k - y^{k,*}\|_2^2 \\
&+ 2\rho n\mathbb{E}\|y^k - y^{k-1}\|_2^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{k-1,*}\|_2^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^n \|y_i^{k-1} - y^{k-1,*}\|_2^2 \\
&+ \frac{4L_q^2 n}{\rho}\|y^{k-1} - y^{k-1,*}\|_2^2 + 2\rho n\|y^{k-1} - y^{k-2}\|_2^2 + \sigma^2 \cdot (\alpha + \rho) \cdot d,
\end{aligned}$$

where the expectation is taken over the additive normally distributed noises at iteration k.

The proof of Lemma H.5 is obtained by carrying out a lengthy refactorization to the inequality of Lemma H.4 and can be found in (Elgabli et al., 2022). The proof of Lemma H.10 is given by exactly the same refactorization applied to the inequality given by H.8.

As a result of Lemma H.10 we define a Lyapunov function for the stochastic DP-FedNew iteration and show the following inequality for it.

Lemma H.11. *Let the Lyapunov function* V_k *be defined as*

$$\begin{aligned}
V_k := \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda^{k,*}\|_2^2 + 2\beta_1\sum_{i=1}^n \|\widetilde{y}_i^k - y^{k,*}\|_2^2 \\
+ \rho n\|y^k - y^{k,*}\|_2^2 + 2\rho n\|y^k - y^{k-1}\|_2^2,
\end{aligned} \tag{H.15}$$

where $\widetilde{y}_i^k$ *denotes the noisy update* (H.1) *Denote* $\widetilde{V}_k = \mathbb{E}V_k$, *where the expectation is taken over all additive noises up to iteration k. Then,* $\widetilde{V}_k$ *satisfies*

$$\widetilde{V}_k \leq \widetilde{V}_{k-1} - \beta_2\mathbb{E}\sum_{i=1}^n \|\widetilde{y}_i^k - y^{k,*}\|^2 + \sigma^2 \cdot (\alpha + \rho) \cdot d \tag{H.16}$$

for some constant $\beta_2 > 0$, *where the expectation is taken over the noise added at iteration k.*

H.5 Our Main Theorem

Theorem H.12. *For all* $k \in \mathbb{Z}$, *there exists* $\ell > k$ *such that*

$$\|y^\ell - y^{\ell,*}\|^2 \leq \frac{\sigma^2 \cdot (\alpha + \rho) \cdot d}{n\beta_2}. \tag{H.17}$$

Proof. At a given iteration k, either

$$\beta_2\sum_{i=1}^n \|\widetilde{y}_i^k - y^{k,*}\|^2 \leq \sigma^2 \cdot (\alpha + \rho) \cdot d \tag{H.18}$$

which implies the inequality (H.17) for $\ell = k$ (combining with the inequality $\sum_{i=1}^n \|\widetilde{y}_i^k - y^{k,*}\|^2 \geq n\|y^k - y^{k,*}\|^2$), or then

$$\beta_2\sum_{i=1}^n \|\widetilde{y}_i^k - y^{k,*}\|^2 > \sigma^2 \cdot (\alpha + \rho) \cdot d,$$

which by the Lemma H.11 implies that either $\widetilde{V}_k$ converges from which case the claim follows, or then there is an $\ell > k$ such that (H.18) holds from which case the claim follows. $\square$

I Tables of Hyperparameter Grids Used in the Experiments

Table 2: Dataset and model description.

datasets	CIFAR10	FashionMNIST	EMNIST	synthetic	Federated EMNIST
classes	10	10	10	10	47
N	50k	60k	240k	50k	90240
test dataset size	10k	10k	10k	10k	22560
distribution	IID	IID	IID	non-IID	non-IID

Table 3: Method for extracting IID features.

Linear layer size	64×10	2048×10
Pretraining architecture	ResNet44	ResNet50
Pretraining weights	CIFAR100	Imagenet

Table 4: Hyperparameter grids.

method	Hyperparameter	alternatives	privacy level	grid size
DP-FedGD	η	$\{0.001, 0.01, 0.1, 1, 10\}$	user/record	10
	C_g	{0.1,1}	user/record	
DP-FedNew / DP-FedNew-FC	α	$\{0.01, 0.1, 1\}$	user/record	90
	ρ	$\{0.01, 0.1, 1\}$	user/record	
	η	$\{0.001, 0.01, 0.1, 1, 10\}$	user/record	
	C, Δ_H	{0.1,1}	user/record	
	C_1, C_2	1	record	
DP-FedFC	η	$\{0.001, 0.01, 0.1, 1, 10\}$	user/record	20
	C_c	{0.1,1}	user/record	
	C_g	{0.1,1}	user/record	
	γ	0.001	user/record	

I.1 Full Experimental Setting

Baselines. Noble et al. (2022) proposed a DP variant of the seminal Scaffold Karimireddy et al. (2020) method designed specifically to tackle data heterogeneity. Similar to popular FedAVG (McMahan et al., 2017), each client running Scaffold performs multiple local updates before releasing their parameter to the server. Each user and server maintains a control variable for drift correction. During each local step, subtraction of the local and global control variates is added to the perturbed gradients to counter local models drifting away from the global due to heterogeneity in their data distributions. Algorithm 2 outlines record-level DP-Scaffold. Experiments in Noble et al. (2022) show that DP-Scaffold performs no worse than DP-FedAVG even on IID datasets. Moreover, we can recover DP-FedAVG by removing the control variables. Therefore, we use it as our main baseline (proxy for DP-FedAVG) in our evaluations on both IID and non-IID datasets. We use the *warm start* variant of DP-Scaffold in which the local control variates $\{c_i^0\}_{i=1}^n$ are initialized to the perturbed gradients in the first step. In each global iteration, DP-Scaffold requires clients and server to exchange the parameter and control variable information, making the communication cost $2 \times d$. Additionally, it only uses the first-order information for model update.

In the centralized case, Mehta et al. (2023) came up with the DP-FC method which involves premultiplying the noisy full batch mean gradients with the inverse of a noisy feature covariance matrix. We present federated versions of DP-FedFC in Algorithms 3 and 4. DP-FedFC is a really strong baseline because it requires server aggregating the *global* noisy feature covariance matrix from clients [1]. On the other hand, clients running DP-FedNew compute primal variables only with their local Hessians. Finally, we treat a federated version of DP-GD (DP-FedGD) as another first-order baseline method (depicted in Algorithms 5 and 6) because it has the same privacy and communication cost as DP-FedNew.

Differences from DP-Scaffold setting. Since DP-Scaffold is also applicable to non-convex problems, (Noble et al., 2022) train a 2-layer neural network from scratch for relevant classification experiments. On the other hand, we train a single linear layer due to our focus on convex problems. We extract IID datasets from public pretrained models. For a fair comparison with DP-FedNew, we consider full batch variants for all baselines with all clients participating. This means no method benefits from privacy amplification due to client and record sampling. Moreover, our learning rate grids could be different. Similar to DP-Scaffold, both DP-FedNew and DP-FedFC can be modified to perform multiple local client-side updates with client/record sampling. However, full exploration of these variations justifies a separate work.

IID Datasets. We use CIFAR10 (Krizhevsky & Hinton, 2009), EMNIST (Cohen et al., 2017b), and FashionMnist (Xiao et al., 2017). We extract features of sizes 64 and 2048 from the last layer of pretrained resnets mentioned in Table 3.

Non-IID Datasets. We pick two classification datasets used in (Noble et al., 2022) and train linear layers from scratch. The first synthetic dataset with $d_x = 60$ is drawn from a generative model proposed in (Li et al., 2020) which allows us to adjust heterogeneity between local distributions with hyperparameters $\alpha \geq 0$ and $\beta \geq 0$. Higher values of α, β indicate more heterogeneity and vice versa. Same as DP-Scaffold, we fix $\alpha = 5$ and $\beta = 5$ which specifies their most heterogeneous setting. The second dataset is *balanced* version of EMNIST (Cohen et al., 2017a), which 47 classes (digits and letters). We consider the extreme scenario in which the training dataset is divided among 47 clients, with each client holding all records of exactly one class. Following (Noble et al., 2022), we call this dataset *Federated* EMNIST. With principal component analysis, we reduce the original dimensionality to 100.

For experiments in Section 6.1 and 6.2, training data is divided among 500 clients such that each client has roughly an equal sized dataset. The dataset sizes are mentioned in Table 2.

[1] Global noisy feature covariance matrix aggregation costs one or more communication rounds. In the user-level DP variant, server needs the second round to share the aggregated noisy covariance matrix back with the clients, making the cost $2d_x^2 + d \cdot T$. However, in record-level DP version, this second round can be saved by pushing the preconditioning to server.

Tasks. In experiments on IID data, we train linear layers of size 64×10 and 2048×10. For reasons of space, all figures for layers of size 2048×10 are moved to the Appendix. We minimize the cross-entropy loss. Extensive hyperparameter tuning is necessary to avoid drawing misleading conclusions. Therefore, in each plot, we show the average test performance of the model selected after performing the hyperparameter search for each method. Test accuracies are averaged across 5 independent runs. The results for other suboptimal hyperparameter candidates are excluded. The best hyperparameters are selected based on the average test accuracy of the last 20 epochs. The hyperparameter grids are specified in Table 4.

Implementation. We generate the non-IID datasets using the code released with the DP-Scaffold paper. The extreme distribution for the Federated EMNIST dataset can be obtained by setting the *similarity* hyperparameter to 0 in their script. We implement our training simulations with PyTorch 2.0 and rely on *vmap* calls for speeding up our per-example gradient and Hessian computations. For scalability, we tune the hyperparameter with Ray Tune (Liaw et al., 2018) on a dedicated multi-GPU cluster. Ray tune maintains a job queue, and the number of models trained parallelly depends on the gpu count. Each model in our set up is trained on 1 gpu, and each gpu has enough memory to accommodate one model. Models finish their entire training on the same gpu that was allocated to them at the start of their training.

Privacy Accounting. For each ε, we obtain the lowest σ through a binary search on the expression given by Lemma B.6 in the Appendix. Due to M number of local updates, we account for $T \cdot M$ steps for DP-Scaffold instead of T steps. Unlike (Koskela & Kulkarni, 2023), we do not consider the privacy cost of tuning towards the final DP guarantee for simplicity. We fix δ to $\frac{1}{N}$ in all experiments.

J Additional Experimental Results

J.1 Impact of Varying the Local Dataset Size.

We have fixed $|D_i| = 500$ in previous experiments. We would like to check the performance consistency of all methods across the client dataset sizes.

Table 5 compares the mean accuracy at epoch 70 for the best model obtained after hyperparameter tuning for several ε-values and different number of clients for IID FashionMNIST dataset and non-IID Federated EMNIST. For Federated EMNIST, each client holds data of atmost 2 classes. We tune the learning rate η and two clipping constants C_c, C_g for DP-FedFC.

For DP-FedNew, DP-FedNew-FC, DP-FedFC, and DP-Scaffold, we observe the expected accuracy reduction as we increase n for $\varepsilon < 1$. DP-FedGD remains relatively unaffected by the variations in the dataset sizes, possibly because client's job is to only share the perturbed gradients with server, and the number of gradient evaluations stay the same. However, accuracies for DP-SCAFFOLD are still much worse than both DP-FedNew and DP-FedNew-FC even at $\varepsilon = 10$. The main conclusion for FashionMNIST is that DP-FedNew or DP-FedNew-FC generally are the most accurate methods for $\varepsilon < 1$, but get outperformed by DP-FedFC for $\varepsilon \geq 1$. For non-IID Federated EMNIST on the other hand, DP-FedNew-FC excels even for larger ε's. DP-FedFC's inferior run on non-IID data can be explained by the fact that heterogeneity induced in the data division or data generation process also changes the shape of the local covariance matrices. The sum of (noisy) local feature covariance matrices aggregated at server could differ a lot from the actual global covariance matrix. We compare user-level DP-FedNew and DP-FedFC in Figure 7. The summary is that DP-FedNew is overall the most accurate method for user-level DP.

DP-FedFC's higher accuracy for record-level DP comes at increased communication cost (under secure aggregation), which is $d_x^2 + d \cdot T$ for T training steps. At this communication cost, DP-FedNew (and DP-FedNew-FC) can perform $\frac{d_x \times d_x}{d} = \frac{d_x \times d_x}{d_x \times c} = \lceil \frac{d_x}{c} \rceil$ additional steps if required. The factor $\lceil \frac{d_x}{c} \rceil$ can be large when $d_x >> c$. We remind that DP-FedNewton-FC only uses the local covariance matrices for primal variable computations.

Table 5: Effect of varying the local dataset size. The numbers in the ε columns report the mean test accuracy after the final 70th epoch for the best model trained on IID FashionMNIST (top) and non-IID Federated EMNIST (bottom) dataset obtained after tuning the hyperparameters mentioned in Table 4. The layer sizes are 64×10 and 100×47. Note that DP-FedFC achieves higher accuracy at the cost of higher communication.

n	$\|D_i\|$	method	$\epsilon = 0.1$	$\epsilon = 0.3$	$\epsilon = 0.5$	$\epsilon = 0.7$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 3$	$\epsilon = 8$	$\epsilon = 10$
20	3000	DP-FedNew	0.638	0.676	0.683	0.684	0.686	0.690	0.692	0.710	0.713
		DP-FedNew-FC	0.643	0.678	0.676	0.683	0.683	0.687	0.683	0.685	0.688
		DP-Scaffold	0.497	0.519	0.471	0.541	0.592	0.531	0.527	0.515	0.510
		DP-FedFC	0.645	0.672	0.675	0.704	0.715	0.724	0.724	0.736	0.739
		DP-FedGD	0.555	0.565	0.569	0.575	0.568	0.563	0.567	0.566	0.568
50	1200	DP-FedNew	0.643	0.679	0.683	0.687	0.689	0.689	0.690	0.708	0.714
		DP-FedNew-FC	0.642	0.670	0.684	0.684	0.682	0.686	0.685	0.685	0.685
		DP-Scaffold	0.427	0.543	0.570	0.586	0.601	0.603	0.611	0.611	0.611
		DP-FedFC	0.635	0.668	0.671	0.704	0.717	0.724	0.725	0.739	0.739
		DP-FedGD	0.561	0.571	0.572	0.571	0.572	0.575	0.571	0.569	0.568
100	600	DP-FedNew	0.608	0.675	0.686	0.686	0.687	0.691	0.691	0.710	0.713
		DP-FedNew-FC	0.604	0.673	0.678	0.684	0.687	0.681	0.684	0.683	0.687
		DP-Scaffold	0.346	0.494	0.549	0.575	0.595	0.605	0.607	0.615	0.615
		DP-FedFC	0.638	0.667	0.672	0.704	0.716	0.723	0.725	0.739	0.738
		DP-FedGD	0.561	0.569	0.574	0.578	0.578	0.569	0.577	0.572	0.569
250	240	DP-FedNew	0.601	0.679	0.683	0.687	0.687	0.690	0.691	0.708	0.716
		DP-FedNew-FC	0.583	0.678	0.681	0.681	0.686	0.682	0.684	0.684	0.685
		DP-Scaffold	0.340	0.452	0.451	0.511	0.483	0.493	0.482	0.513	0.492
		DP-FedFC	0.641	0.667	0.675	0.705	0.717	0.723	0.725	0.735	0.739
		DP-FedGD	0.574	0.568	0.577	0.575	0.576	0.572	0.568	0.569	0.569
500	120	DP-FedNew	0.575	0.654	0.682	0.688	0.688	0.690	0.691	0.708	0.713
		DP-FedNew-FC	0.574	0.651	0.681	0.687	0.687	0.686	0.687	0.683	0.688
		DP-Scaffold	0.331	0.418	0.464	0.470	0.531	0.585	0.597	0.607	0.606
		DP-FedFC	0.634	0.667	0.672	0.705	0.716	0.723	0.725	0.739	0.740
		DP-FedGD	0.560	0.571	0.572	0.569	0.568	0.578	0.579	0.574	0.575

n	$\|D_i\|$	method	$\epsilon = 0.1$	$\epsilon = 0.3$	$\epsilon = 0.5$	$\epsilon = 0.7$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 3$	$\epsilon = 8$	$\epsilon = 10$
47	1920	DP-FedNew-FC	0.390	0.489	0.527	0.556	0.582	0.613	0.622	0.628	0.629
		DP-Scaffold	0.135	0.270	0.329	0.375	0.416	0.483	0.512	0.552	0.558
		DP-FedFC	0.390	0.489	0.527	0.550	0.569	0.545	0.567	0.574	0.595
		DP-FedGD	0.389	0.488	0.524	0.545	0.533	0.565	0.564	0.576	0.569
100	893	DP-FedNew-FC	0.383	0.481	0.548	0.552	0.584	0.610	0.619	0.626	0.627
		DP-Scaffold	0.085	0.217	0.281	0.322	0.361	0.439	0.469	0.493	0.492
		DP-FedFC	0.382	0.491	0.521	0.548	0.565	0.560	0.559	0.585	0.580
		DP-FedGD	0.383	0.482	0.519	0.541	0.558	0.562	0.562	0.574	0.564
250	359	DP-FedNew-FC	0.376	0.491	0.533	0.557	0.582	0.614	0.621	0.628	0.631
		DP-Scaffold	0.067	0.148	0.207	0.256	0.293	0.347	0.365	0.372	0.370
		DP-FedFC	0.384	0.481	0.526	0.555	0.566	0.577	0.560	0.550	0.587
		DP-FedGD	0.385	0.484	0.525	0.545	0.553	0.557	0.550	0.548	0.544
500	179	DP-FedNew-FC	0.384	0.462	0.539	0.568	0.593	0.620	0.631	0.639	0.649
		DP-Scaffold	0.053	0.122	0.157	0.192	0.205	0.234	0.243	0.253	0.249
		DP-FedFC	0.401	0.495	0.539	0.561	0.575	0.584	0.587	0.550	0.669
		DP-FedGD	0.391	0.498	0.534	0.547	0.555	0.573	0.558	0.609	0.558

J.2 RECORD-LEVEL DP RESULTS FOR LAYER SIZE 2048×10.

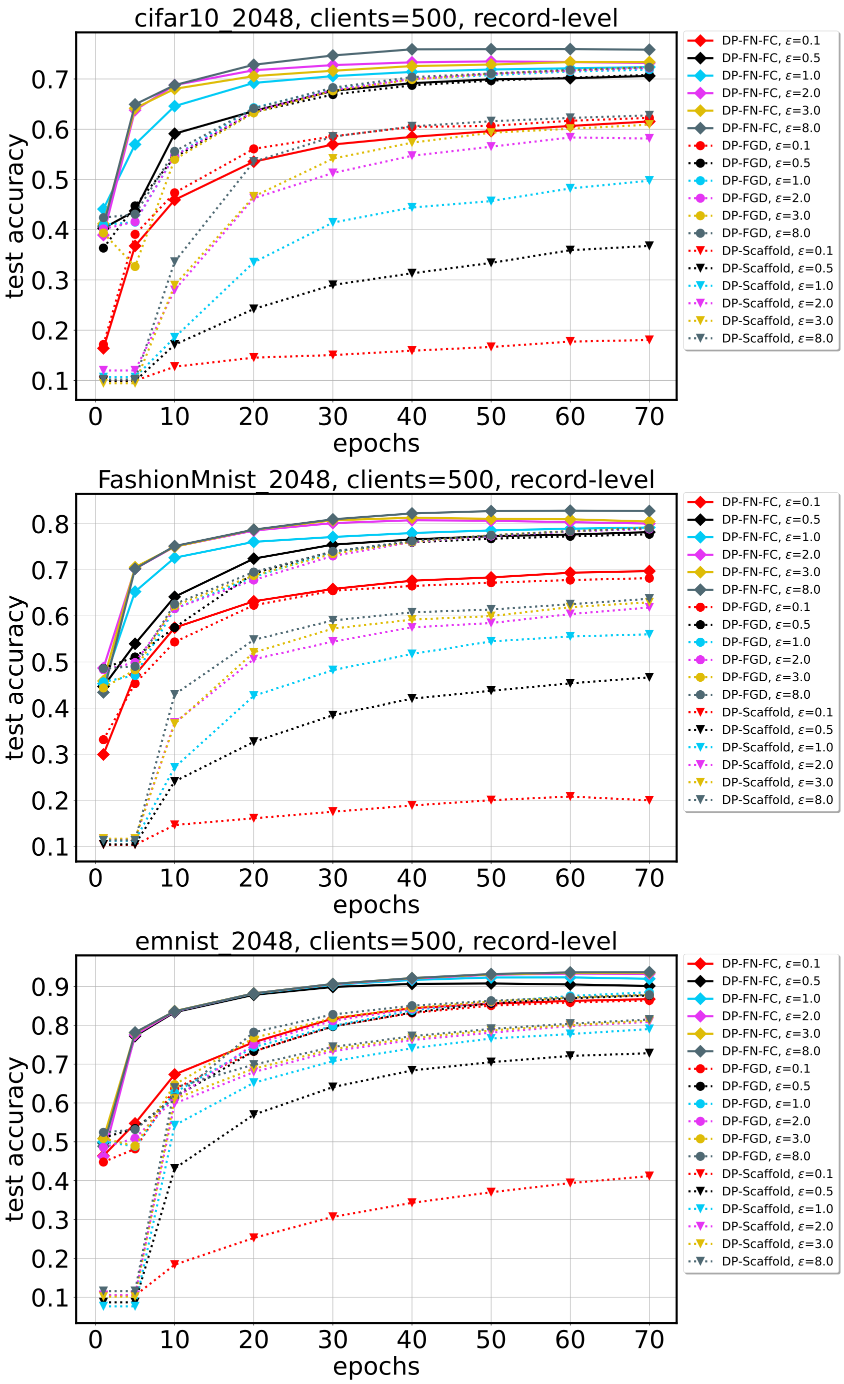

Figure 3: IID data: Record-level results for DP-FedNew-FC (DP-FN-FC in plots), DP-FedGD (DP-FGD in plots), and DP-Scaffold. For each ϵ, we plot the test accuracies of the best model obtained after hyperparameter tuning. The model size is 2048×10. Check Figure 6 for comparison with DP-FedFC.

J.3 USER-LEVEL DP EXPERIMENTS

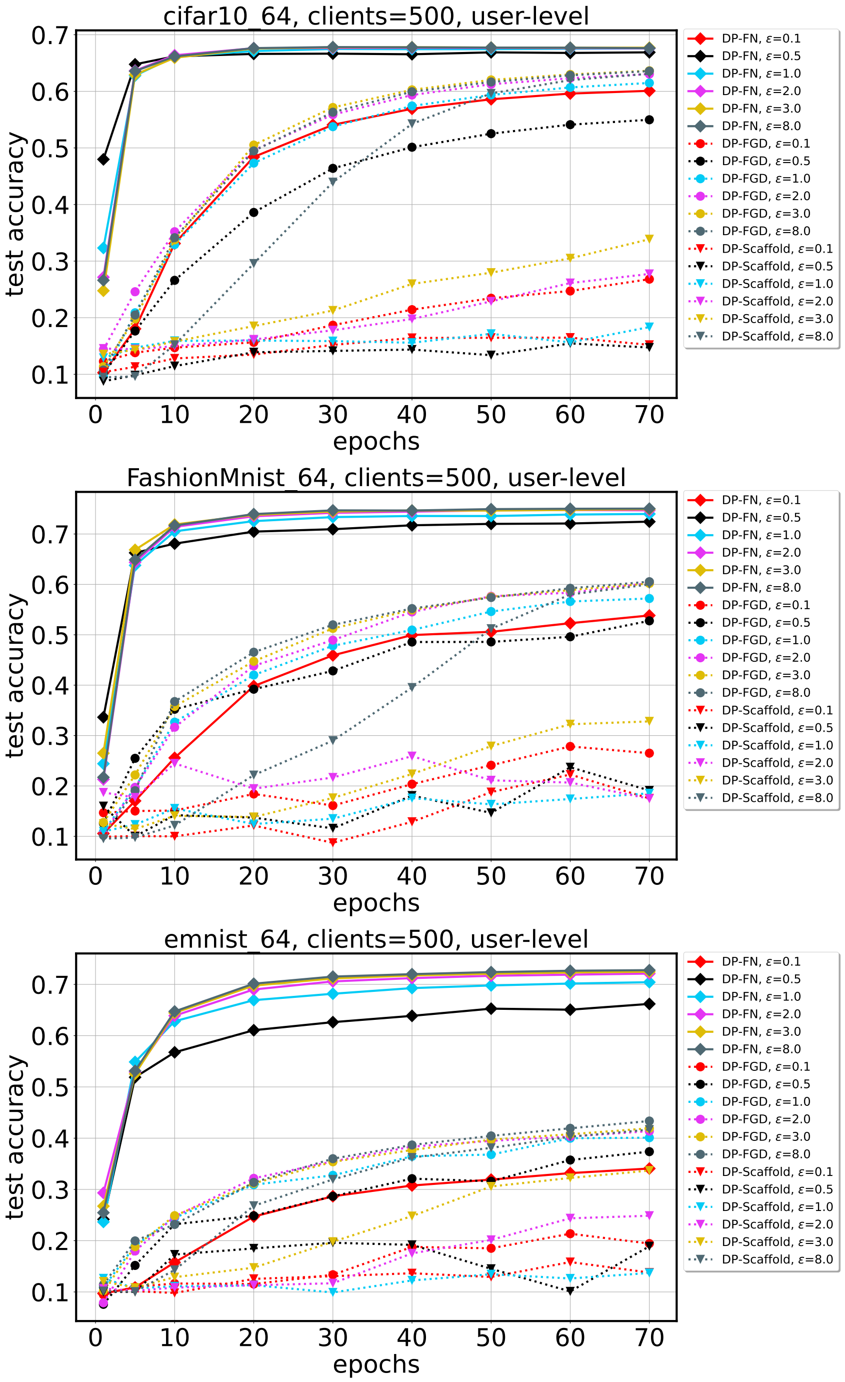

Figure 4: IID data: User-level results for DP-FedNew (DP-FN in plots), DP-Scaffold, and DP-FedGD (DP-FGD in plots). For each ϵ, we plot the test accuracies of the best model obtained after hyperparameter tuning. The model size is 64×10. Check Figure 7 for comparison with DP-FedFC on a layer size 2048×10.

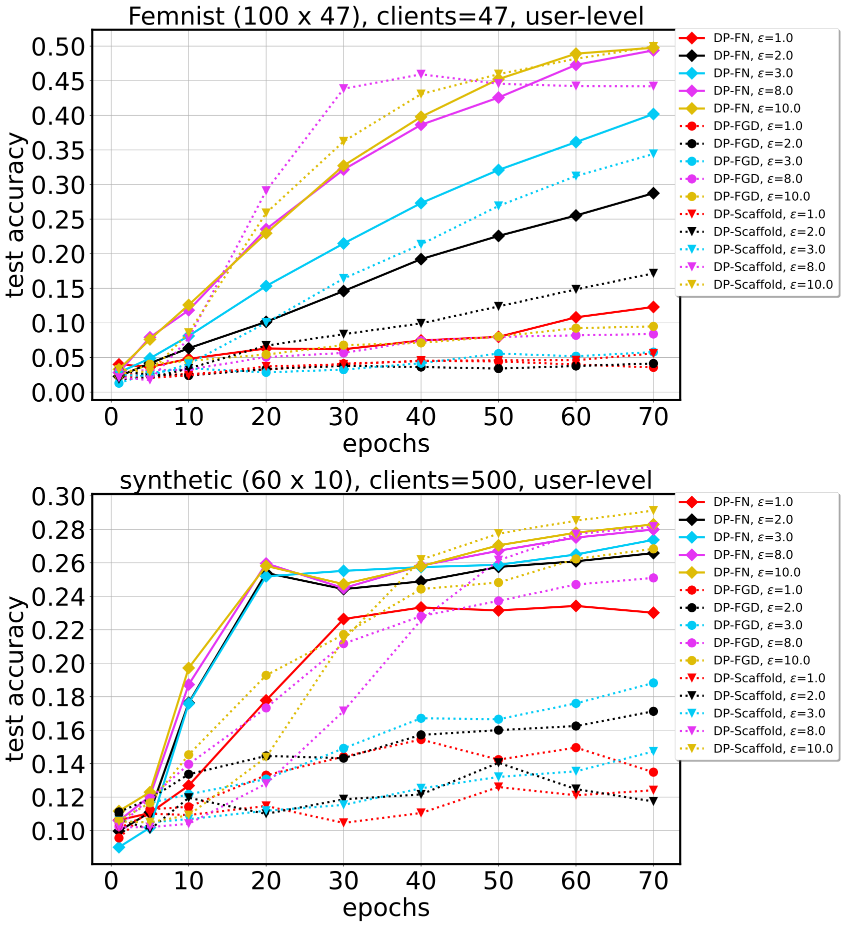

Figure 5: Non-IID data: User-level results for DP-FedNew (DP-FN in plots) and DP-FedGD (DP-FGD in plots). For each ϵ, we plot the test accuracies of the best model obtained after hyperparameter tuning. The model sizes are 100×47 and 60×10.

J.4 Comparison with DP-FedFC.

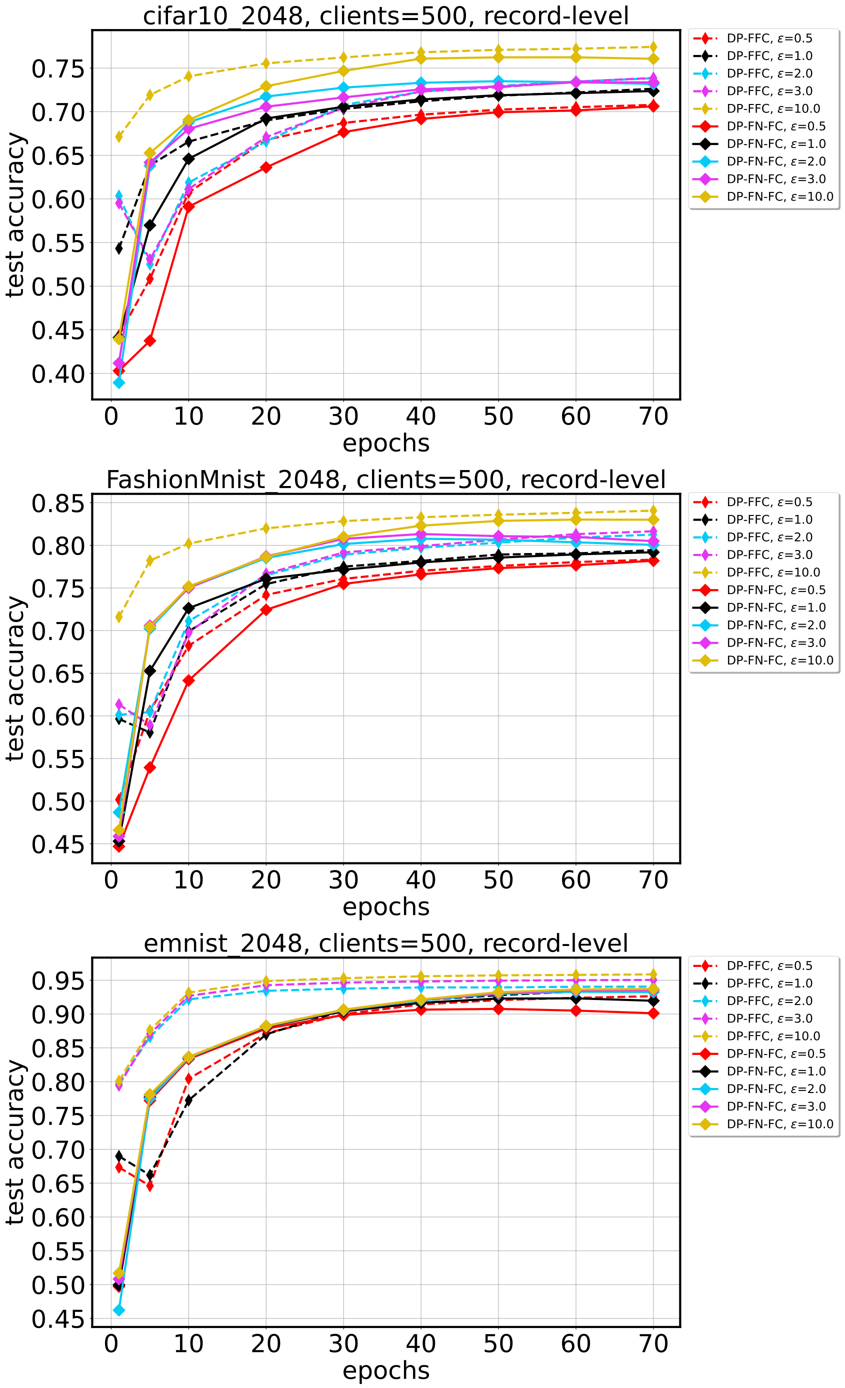

Figure 6: IID data: Record-level results for DP-FedNew-FC (DP-FN-FC in plots), DP-FedFC (DP-FFC in plots). For each ϵ, we plot the test accuracies of the best model obtained after hyperparameter tuning. The model size is 2048×10.

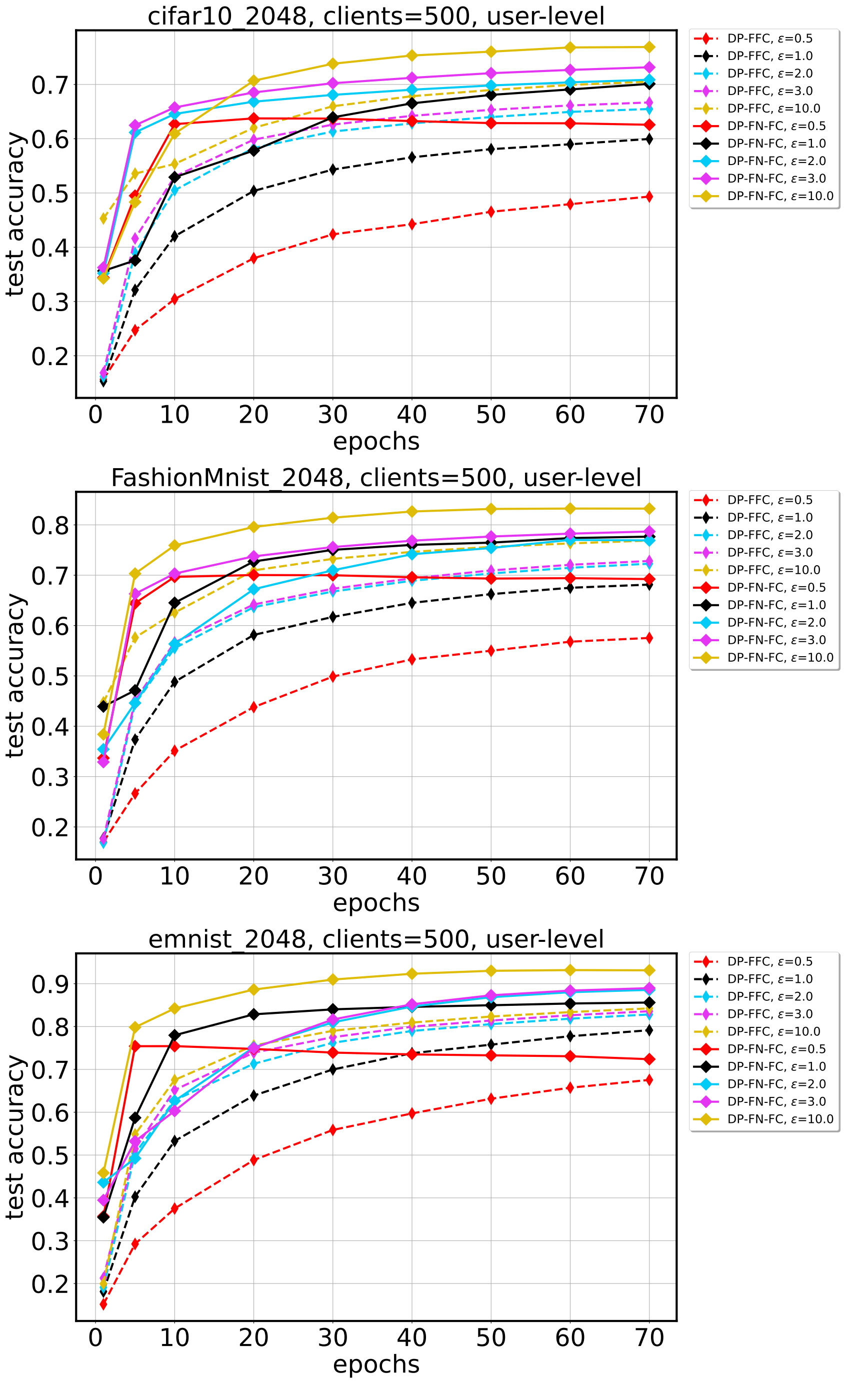

Figure 7: IID data: User-level results for DP-FedNew-FC (DP-FN-FC in plots), DP-FedFC (DP-FFC in plots). For each ϵ, we plot the test accuracies of the best model obtained after hyperparameter tuning. The model size is 2048×10.