Differential Privacy in HyperNetworks for Personalized Federated Learning

Vaisnavi Nemala
New Jersey Institute of Technology
Newark, New Jersey, USA
van2@njit.edu

Phung Lai*
University at Albany - State University of New York
Albany, New York, USA
lai@albany.edu

NhatHai Phan*
New Jersey Institute of Technology
Newark, New Jersey, USA
phan@njit.edu

ABSTRACT

Federated learning (FL) is a framework for collaborative learning among users through a coordinating server. A recent HyperNetwork-based personalized FL framework, called HyperNetFL, is used to generate local models using personalized descriptors optimized for each user independently. However, HyperNetFL introduces unknown privacy risks. This paper introduces a novel approach to preserve user-level differential privacy, dubbed User-level DP, by providing formal privacy protection for data owners in training a HyperNetFL model. To achieve that, our proposed algorithm, called UDP-Alg, optimizes the trade-off between privacy loss and model utility by tightening sensitivity bounds. An intensive evaluation using benchmark datasets shows that our proposed UDP-Alg significantly improves privacy protection at a modest cost in utility.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Security and privacy** → **Privacy-preserving protocols**.

KEYWORDS

Federated Learning; Differential Privacy; Hypernetworks

ACM Reference Format:
Vaisnavi Nemala, Phung Lai, and NhatHai Phan. 2023. Differential Privacy in HyperNetworks for Personalized Federated Learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615203

1 INTRODUCTION

Federated learning (FL) allows a server to jointly train a model through multiple local users, without the need to share the users' data. This is vital when privacy concerns are raised and the sharing of sensitive local data must be prevented [21]. Examples of FL addressing privacy risks are when data can reveal potentially sensitive information about the users, such as sensitive medical reports and data, financial transactions, or personal data disclosing racial/ethnic, political, or religious affiliations. However, data distribution often varies amongst users, such as varying geographic backgrounds or use case scenarios. Thus, Personalized Federated Learning [27] addresses the heterogeneity of users through the introduction of a personalized model for each user (versus a shared global model).

Among existing approaches, Hypernetwork-based personalized FL Framework (HyperNetFL) allows us to benefit from joint training of a HyperNetwork, which is used to generate the users' personalized models [26]. Although effective, this unique personalized federated training of HyperNetFL can lead to previously unknown concerns for maintaining privacy for the users.

We seek to address these challenges by focusing on preserving differential privacy in HyperNetFL. We specifically explore user-level differential privacy (User-level DP), which investigates the effects of the presence or absence of a user's full records on a dataset. However, unlike the User-level DP applied on recurrent language models [20], in HyperNetFL, without a global model aggregation at the server, it is non-trivial on how to carefully calibrate the noise added into the training process, so that the server will generate User-level DP model parameters without an undue cost in model utility.

Key Contributions. Motivated by this, we structure our paper around the following significant contributions: **(1)** A novel algorithm, called UDP-Alg, to provide a formal User-level DP guarantee for HyperNetFL; **(2)** An optimization of the trade-off between privacy protection with model utility, conducted on a series of experiments on image classification using benchmark datasets; and **(3)** An exploration of various effects of DP hyperparameters (such as clipping bound, noise scale, etc.) on the trade-off and from that, making a suggestion on which hyperparameters practitioners could use to better balance the trade-off.

Outline. The paper is organized as follows. We briefly review background in Section 2. Section 3 discusses the algorithm for guaranteeing User-level DP in the HyperNetFL framework in depth. Section 4 explores experimental results to empirically demonstrate the interplay between User-level DP and model utility. We conclude the paper in Section 5.

2 BACKGROUND

2.1 Federated Learning (FL)

FL is a multi-round communication protocol between a server and N users. At each round t, the server sends the latest model θ_t to a random subset of users U_t. These selected users use their local data D_u to train the model, and compute their local gradients

*Corresponding authors

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615203

$\triangle\theta_t^u = \theta_t^u - \theta_t$, and send them back to the server. Then, the server aggregates all the received gradients from the users in U_t using an aggregation function $\mathcal{G} : R^{|U_t| \times n} \rightarrow R^n$ where n is the size of $\triangle\theta_t$. The aggregated gradient is added to θ_t, which is $\theta_{t+1} = \theta_t - \lambda\mathcal{G}(\{\triangle\theta_t^u\}_{u \in U_t})$, where λ is the server's learning rate. FedAvg is a well-applied aggregation in FL algorithms [11], as $\theta_{t+1} = \theta_t - \lambda(\sum_{u \in U_t} n_u \times \triangle\theta_t^u)/\sum_{u \in U_t} n_u$.

Personalized Federated Learning (pFed). FL methods often encounter a significant variation in data distributions across users, which results in a substantial difference in the model's effectiveness [4, 28]. Therefore, pFed techniques have been proposed to overcome this problem by achieving personalized performance that can adapt to the varying data [8, 10, 26]. pFed approaches can be broadly categorized into four research lines: *(1) Regularization-based Approaches*, which modify local training through regularization or penalization to address data distribution drifting, resulting in a divergence between the weights of local and global models [12, 17, 23]; *(2) Clustering-based Approaches*, where the server assigns users to clusters and aggregate local models within each cluster [9, 25]; *(3) Knowledge Distillation*, where the server ensembles users' knowledge by a generator or a consensus distributed across the network [16, 26, 29]; and *(4) Meta Learning*, which leverages the concept of meta-training and meta-testing. In meta-training, a sensitive initial model is learned, which can quickly adapt to various tasks, typically using techniques like Model Agnostic Meta-Learning (MAML). This initial model is then mapped to the global model, and in the meta-testing step, it is further adapted to specific tasks on the users' side.

HyperNet-based Personalized FL. One of the state-of-the-art pFed approaches is using a single large network at the server $h(\varphi, v_u)$, called HyperNetFL [26], to generate local models θ_u, given the user's descriptors v_u. In fact, HyperNetFL learns a family of personalized models $\{\theta_u = h(v_u, \varphi)\}_{u \in [N]}$, such that the users and the server minimize their loss functions:

$$\arg \min_{\varphi, \{v_u\}_{u \in [N]}} \frac{1}{N} \sum_{u \in [N]} L_u(h(v_u, \varphi)) \tag{1}$$

2.2 Differential Privacy

Differential Privacy (DP) [5–7] provides the guarantee that adversaries are limited in learning about private data by ensuring similar model outcomes, regardless if any single training sample is in the database or not. The definition of DP is as follows:

Definition 2.1. (ϵ, δ)-DP: A randomized mechanism $\mathcal{M}$: $\mathcal{D} \rightarrow \mathcal{R}$ with a domain $\mathcal{D}$ (e.g., possible training datasets) and range $\mathcal{R}$ (e.g., all possible trained outcomes) satisfies (ϵ,δ)-DP, if for any two adjacent datasets $D, D' \in \mathcal{D}$ and for any subset of outputs $S \subseteq \mathcal{R}$, it holds that:

$$Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} Pr[\mathcal{M}(D') \in S] + \delta \tag{2}$$

The privacy budget ϵ controls how similarity between the two outcomes when D and D' may differ. A smaller ϵ enforces a stronger privacy protection. The broken probability δ is the upper bound probability for the worst-case scenarios when an adversary can infer the presence of a data sample in the training set [14].

In Definition 2.1, the explanation of adjacent databases leaves open. It depends on the application to determine the level of DP protection needed. Based on defining the adjacent databases, there are different levels of DP protection, which can be categorized into four research lines, as discussed next.

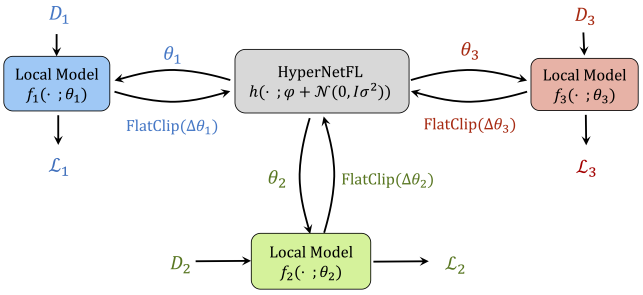


Figure 1: HyperNetFL with User-level DP guarantee.

Sample-level DP. Traditional DP mechanisms [7, 22, 24] ensure DP at the sample-level, in which D and D' are different from at most a single training sample. This DP level only protects the privacy of individual training samples, whereas we are seeking to provide privacy for the whole user histories in the training dataset.

Element-level DP. Element-level DP [2] ensures that an adversary cannot infer whether users have a sensitive element in their data. Similar to sample-level DP, element-level DP is different from our goal, since it does not provide DP protection for users.

Local DP (LDP). Different from our purpose of protecting user membership information, the key idea of local DP is to protect users' data. By observing the outcomes, it ensures adversaries cannot distinguish whether the outcomes are from input values x or x'. However, LDP approaches typically add significant amount of noise to the data/model parameters to preserve DP, resulting remarkable model utility drop.

User-level DP. [20] proposed a User-level DP that confirms the presence of an arbitrary user in the training dataset. To provide such protection, the adjacent datasets D and D' differ on all the samples belonging to an arbitrary user.

User-level DP is similar to our purpose of protecting user membership information; however, without an aggregation at the server as in traditional FL frameworks [11, 19], it is challenging to bound the sensitivity of users' queries and to quantify the amount of noise added to the model parameters so that the network at the server $h(\varphi, \cdot)$ will generate User-level DP model parameters $\{\theta_u\}_{u=1}^N$. Therefore, protecting user membership information in HyperNetFL is not trivial.

3 USER-LEVEL DP IN HYPERNETFL

In this section, we focus on answering the question: *Could we protect user membership information in HyperNetFL and how?* Based upon that, we propose our approach to preserve User-level DP in HyperNetFL.

To protect the generated model parameters $\{\theta_u\}_{u=1}^N$, a naive solution is to simply add noise, e.g., Gaussian noise or Laplacian noise [1] into the output of the HyperNetFL before sending them to users. However, this can severely alter the value of the parameters

Algorithm 1 UDP-Alg in HyperNetFL

1: **Input**: Number of users N, number of rounds T, number of local rounds K, server's learning rates λ and ζ, users' learning rate η, clipping bound S, clipping function **ClipFn**(Δ, S), a hyper-parameter z, and $L_u(B)$ is the loss function $L_u(\theta)$ on a mini-batch B
2: Initialize $h(\varphi, \cdot)$ and moments accountant $\mathcal{M}$
3: $w_u \leftarrow \min(\frac{|D_u|}{\hat{w}_u}, 1)$ for all users u # where $\hat{w}_u$ is per-user data sample cap
4: $W \leftarrow \sum_{u=1}^{N} w_u$
5: **for** $t = 1, \ldots, T$ **do**
6: $\quad U_t \leftarrow$ Sample a set of users with probability q
7: $\quad$ **for** each user $u \in U_t$ **do**
8: $\quad\quad$ set $\theta_t^u = h(\varphi, v_u)$ and $\tilde{\theta}^u = \theta^u$
9: $\quad\quad$ **for** $k = 1, \ldots, K$ **do**
10: $\quad\quad\quad$ sample mini-batch $B_k \subset D_u$
11: $\quad\quad\quad \triangle\theta_{k+1}^u = \tilde{\theta}_k^u - \eta \nabla_{\tilde{\theta}_k^u} L_u(B_k)$
12: $\quad\quad \triangle\theta_t^u = \tilde{\theta}_K^u - \theta_t^u$
13: $\quad \triangle_t^u \leftarrow \text{ClipFn}((\nabla_\varphi \theta_t^u)^\top \triangle \theta_t^u, S)$
14: $\quad \nabla\varphi = \frac{\sum_{u \in U_t} w_u \triangle_t^u}{qW}$
15: $\quad \sigma \leftarrow \frac{z \max(w_u) S}{qW}$
16: $\quad \varphi = \varphi - \lambda[\nabla\varphi + \mathcal{N}(0, I\sigma^2)]$
17: $\quad \forall u \in U_t : v_u = v_u - \zeta \nabla_{v_u} \varphi^\top \triangle \theta_t^u$
18: $\quad \mathcal{M}$.accum_priv_spending(z)
19: **ClipFn**(Δ, S): return $\pi(\Delta, S) \leftarrow \Delta \cdot \min\left(1, \frac{S}{\|\Delta\|}\right)$

and adversely affect the model utility. Therefore, it is needed to carefully calibrate the DP noise added to optimize the trade-off between privacy protection and model utility.

UDP-Alg. To achieve User-level DP in HyperNetFL (Algorithm 1 and Figure 1) without an undue cost in model utility, at each iteration t, we randomly sample U_t users from N users with the sample rate q (Line 5). Then, each of the selected users u in U_t update their model θ_u using the local data D_u (Lines 8-10). We compute the gradients of model parameters for a particular user, denoted as Δ_t^u (Line 11). Here, we clip the per-user gradients so that its L_2-norm is bounded by a predefined gradient clipping bound S (Lines 12, 18). Next, a weighted-average estimator f is employed to compute the average gradient Δ^t using the clipped gradients Δ_t^u gathered from all the selected users (Line 13). Finally, we add random Gaussian noise $\mathcal{N}(0, I\sigma^2)$ to the model update (Line 14). During the training, the moments accountant $\mathcal{M}$ is used to compute the T training steps' privacy budget consumption, which is incremented at every step of the training process (Line 17).

To tighten the sensitivity bound, our weighted-average estimator f for per-user vectors Δ^u (Line 13) is as follows:

$$f(S^t) = \frac{\sum_{u \in U_t} w_u \triangle \theta_t^u}{qW} \tag{3}$$

where Δ_t^u is the clipped gradients of local gradients $\triangle\theta_t^u$ over the network parameters at the server φ. The weight w_u is a weight associated with a user u, capturing the influence of a user to the model outcome and $W_u = \sum_{u=1}^{N} w_u$.

Since $\mathbb{E}[\sum_{u \in S^t} w_u] = qW$, the estimator f is unbiased. The sensitivity of the estimator $\mathbb{S}(f)$ is computed as: $\mathbb{S}(f) = \max_{u', e'} \|f(\{S^t \cup u'\}) - f(\{S^t\})\|_2$. $\mathbb{S}(f)$ is bounded in the following lemma.

Lemma 3.1. *If for all users u we have $\|\Delta_u^t\|_2 \leq S$, then $\mathbb{S}(f) \leq \frac{\max(w_u) S}{qW}$.*

Proof. If for all users $\|\Delta_u^t\|_2 \leq S$, then we have:

$$\mathbb{S}(f) = \frac{\sum_{u \in S^t \cup u'} w_u \Delta_u^t - \sum_{u \in S^t} w_u \Delta_u^t}{qW} \leq \frac{w'_u \Delta_u^t}{qW} \leq \frac{\max(w_u) S}{qW}$$

Consequently, Lemma 3.1 holds. □

Once the sensitivity of the estimator f is bounded, we can add Gaussian noise scaled to the sensitivity $\mathbb{S}(f)$ to obtain a privacy guarantee. By applying Lemma 3.1, the noise scale σ becomes:

$$\sigma = z\mathbb{S}(f) = \frac{z \max(w_u) S}{qW} \tag{4}$$

User-level DP Guarantee. Given the bounded sensitivity of the estimator, the moments accountant $\mathcal{M}$ [1] is used to bound the total User-level DP privacy consumption of T steps of the Gaussian mechanism with the noise $\mathcal{N}(0, I\sigma^2)$ (Line 14). Since φ is (ϵ, δ)-User-level DP, the generated model parameters $\{\theta_u\}_{u=1}^N$ and the user descriptor $\{v_u\}_{u=1}^N$ are also User-level DP thanks to the post-processing property [7]. As a result, Algorithm 1 preserves User-level DP with the noise scale $z = \sigma/\mathbb{S}(f)$ as in the following Theorem.

Theorem 3.2. *For the estimator f, the moments accountant of the sampled Gaussian mechanism correctly computes User-level DP privacy loss with the scale $z = \sigma/\mathbb{S}(f)$ for T training steps.*

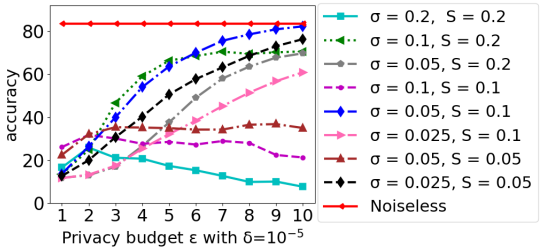
Proof. At each step, users are selected randomly with probabilities q For the estimator f, if the l_2-norm of each user's gradient update is bounded by $\mathbb{S}(f)$, then the moments accountant can be bounded by that of the sampled Gaussian mechanism with sensitivity 1, the scale $z = \sigma/\mathbb{S}(f)$, and sampling probability q. Thus, we can apply the composability as in Theorem 2.1 [1] to correctly compute the User-level DP privacy loss with the scale $z = \sigma/\mathbb{S}(f)$ for T training steps. □

4 EXPERIMENTAL RESULTS

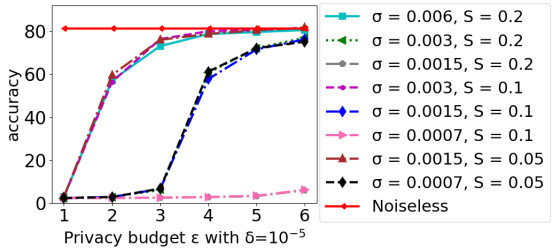
We conduct extensive experiments to shed light on understanding 1) the interplay between privacy and model utility and 2) the immediate effects of DP hyperparameters, such as the clipping bound, learning rate, noise scale, etc., on the trade-off between model utility and privacy protection.

4.1 Datasets

To achieve our goal, we conduct an extensive experiment using the CIFAR-10 [13], FEMNIST (Federated Extended MNIST) [3], and CelebA datasets [18]. For these datasets, we generate non-iid data distribution across users in terms of the number of local training data. In the CIFAR-10 dataset, there are 50,000 training and 10,000 testing samples across 100 users. In the FEMNIST dataset, we remove some users that have a very small number of data samples (i.e., less than 30 samples); therefore, we use 3,400 users with 600,000 training samples and 150,000 testing samples. In the CelebA dataset, there are 155,529 training and 19,962 testing samples with 6,348 clients. There are 10 classes, 62 classes, and 2 classes in the CIFAR-10, FEMNIST, and CelebA datasets, respectively.



(a) CIFAR-10 dataset



(b) FEMNIST dataset

Figure 2: Image classification on the CIFAR-10 and FEMNIST datasets.

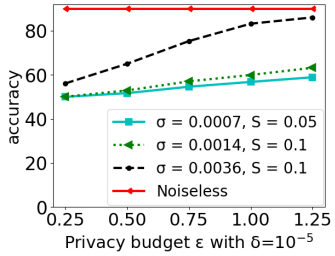


Figure 3: Image classification on the CelebA dataset.

4.2 Model Configurations and Evaluation

We adopt the model configuration in [26], in which we use a LeNet-based network [15] with two convolution and two fully connected layers for the local model and a fully-connected network with three hidden layers and multiple linear heads per target weight tensor for the HyperNetFL. SGD optimizers with a learning rate of 0.01 for the HyperNetFL and 0.001 for the local models are used.

We evaluate our method with image classification using a model accuracy and privacy budget ϵ. The higher the accuracy is, the better model is. The lower the ϵ is, the better privacy protection is. We compare our work with a **Noiseless** model, which is a HyperNetFL trained without any privacy-preserving mechanisms.

To examine the effects of DP hyperparameters on the trade-off between utility and privacy, we tested a wide range of hyperparameters, including the gradient clipping bound $S \in [0.05, 0.1, 0.2]$, the scale $z \in [5, 10]$, and the sample rate $q \in [0.05, 0.1, 0.2]$. The broken probability is $\delta = 10^{-5}$.

4.3 Experimental Results

To answer our evaluation questions, we conducted the following experiments: **(1)** investigating the interplay between privacy budget and model utility and **(2)** studying the impacts of different hyperparameters on the privacy budget and model utility.

Privacy Budget (ϵ, δ) and Model Utility. In the CIFAR-10 dataset, UDP-Alg achieves a good model performance at a tight privacy budget ϵ (Figure 2a). At $\epsilon = 4$, the model accuracy is 53.95%. It significantly improves and reaches the upper-bound Noiseless model performance when $\epsilon = 10$ with 82.23% accuracy. This result is obtained when the noise $\sigma = 0.05$ and the clipping bound $S = 0.1$. In the FEMNIST and CelebA datasets (Figures 2b and 3), we observe a similar phenomenon, but obtain a good model performance at smaller privacy budgets. In the FEMNIST dataset, at $\epsilon = 3$, the gap between the UDP-Alg model that has $\sigma = 0.0015$ and $S = 0.05$ with the Noiseless model is only 5.31%. In the CelebA dataset, the gap is even smaller, with only 3.84%, at a more rigorous privacy budget $\epsilon = 1.25$. These results are promising and consistent across the datasets, showing the effectiveness of our proposed algorithm in providing User-level DP in HyperNetFL.

Effects of Different Noise and Clipping Tradeoffs on Model Utility. Figures 2a, 2b, and 3 show model accuracy of our mechanism with varying levels of clipping S and noise σ, across different datasets. Given that S remains unchanged, when σ decreases, the model accuracy slightly increases. For example, in the CIFAR-10 dataset, given $S = 0.1$, with $\sigma = 0.1$, the model accuracy is 21.27% and with $\sigma = 0.05$, the model accuracy remarkably improves to 82.23%. We observe the same phenomenon in the FEMNIST dataset.

When σ remains constant, we notice that as the clipping bound S decreases, there is an increase in model accuracy. For example, in the CIFAR-10 dataset, given $\sigma = 0.05$, when $S = 0.2$, the model accuracy is 69.67%. Decreasing $S = 0.1$, the accuracy significantly improves to 82.23%. This trend is prominent in the FEMNIST dataset.

When the noise is large, it significantly modifies the parameter values, leading to a detrimental impact on the model performance. The results suggest that using a small σ and correspondingly small S (thus fixing z so the privacy consumption of each round is unchanged) provides better model utility and privacy trade-offs.

5 CONCLUSION AND FUTURE WORKS

In this work, we developed a novel approach to preserve user-level DP in HyperNetFL. By incorporating user sampling in the training process and tightening sensitivity bounds, we mitigated the trade-off between model utility and privacy loss. Rigorous evaluations show that UDP-Alg achieves good results at small privacy budgets indicating rigorous privacy protection.

Our work opens several research directions in the near future. We will examine UDP-Alg in a variety of datasets and applications. That will provide meaningful observation to guide us how to design private algorithms with adaptive hyperparameters across training rounds. This will significantly improve model utility and stability of HyperNetFL models under the same privacy protection.

ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under grants IIS-2041096 and CNS-1935928.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *ACM CCS*. 308–318.

[2] Hilal Asi, John Duchi, and Omid Javidbakht. 2019. Element level differential privacy: The right granularity of privacy. *arXiv preprint arXiv:1912.04042* (2019).

[3] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097* (2018).

[4] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting shared representations for personalized federated learning. In *ICML*. 2089–2099.

[5] Cynthia Dwork. 2011. A firm foundation for private data analysis. *Commun. ACM* 54, 1 (2011), 86–95.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference*. 265–284.

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

[8] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. 2022. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *CVPR*. 10112–10121.

[9] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *NeurIPS* 33 (2020), 19586–19597.

[10] David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106* (2016).

[11] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning* (2021).

[12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*. 5132–5143.

[13] A. Krizhevsky et al. 2009. Learning multiple layers of features from tiny images. (2009).

[14] Phung Lai, NhatHai Phan, Tong Sun, Rajiv Jain, Franck Dernoncourt, Jiuxiang Gu, and Nikolaos Barmpalios. 2022. User-Entity Differential Privacy in Learning Natural Language Models. arXiv:2211.01141 [cs.CR]

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[16] Daliang Li and Junpu Wang. 2019. Fedmd: Heterogenous federated learning via model distillation. *NeurIPS 2019 Workshop* (2019).

[17] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems* 2 (2020), 429–450.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang. 2015. Deep Learning Face Attributes in the Wild. In *ICCV*.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. 1273–1282.

[20] H.B. McMahan, D. Ramage, K. Talwar, and L. Zhang. 2017. Learning differentially private recurrent language models. *ICLR* (2017).

[21] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. 2016. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629* 2 (2016).

[22] X. Pan, M. Zhang, S. Ji, and M. Yang. 2020. Privacy risks of general-purpose language models. In *IEEE SP*. 1314–1331.

[23] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. 2021. Adaptive federated optimization. *ICLR* (2021).

[24] A. Roth. 2012. Buying private data at auction: the sensitive surveyor's problem. *ACM SIGecom Exchanges* 11, 1 (2012), 1–8.

[25] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 8 (2020), 3710–3722.

[26] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik. 2021. Personalized Federated Learning using Hypernetworks. *ICML* (2021).

[27] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated multi-task learning. *NeurIPS* 30 (2017).

[28] Canh T Dinh, Nguyen Tran, and Josh Nguyen. 2020. Personalized federated learning with moreau envelopes. *NeurIPS* 33 (2020), 21394–21405.

[29] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*. 12878–12889.