FedCOM: Efficient Personalized Federated Learning by Finding Your Best Peers

Xintong Lu
BUPT
Beijing, China
xintonglu@bupt.edu.cn

Yuchao Zhang*
BUPT
Beijing, China
yczhang@bupt.edu.cn

Huan Zou
BUPT
Beijing, China
zouhuan@bupt.edu.cn

Yilei Liang
University of Cambridge
Cambridge, UK
yl841@cst.cam.ac.uk

Wendong Wang
BUPT
Beijing, China
wdwang@bupt.edu.cn

Jon Crowcroft
University of Cambridge
Cambridge, UK
Jon.Crowcroft@cl.cam.ac.uk

ABSTRACT

Personalized federated learning aims to address two key challenges in federated systems: performance degradation of the global model, and the lack of specificity to individual clients. Both of the two challenges are caused by client heterogeneity. Previous solutions often assume all clients are dealing with similar learning tasks, in which nearly the same kinds of labels should be learned and global learning performs well. However, another common scenario is ignored by most work, where label distributions among different clients involve disparate label kinds, and there are also substantial biases among different labels even within the same client. In response to this challenge, we propose FedCOM, an innovative workflow that can help the local personalized federated model to find its best peer(s) to learn from. Our method consists of two key components: gradient-based Complementary Client Matching and Personalized Federated Learning that combines these complementary aspects. Inspired by Class Activation Map (CAM) in neural network interpretability, we take partial derivatives on the loss of target minority classes to identify feature channels that make significant contributions to the classification task. Subsequently, we introduce the complementary models selected as regularization terms into local personalized optimization objectives. Experiments demonstrate that FedCOM can achieve faster convergence, while maintaining higher accuracy on MNIST, Fashion-MNIST, and CIFAR-10 with an average improvement of 1.35% compared to the local-only training strategy, as well as 44.19% and 1.28% to FedAVG and the personalized federated method Ditto.

KEYWORDS

personalized federated learning, optimization, non-IID

1 INTRODUCTION

Federated Learning (FL)[15] is a machine learning framework that enables multiple participants to collaboratively construct a federated model while upholding privacy preservation. As the saying goes, every coin has two sides. While federated learning provides privacy protection for local data, non-independent and non-identically distributed data(non-IID) problems caused by data silos come along and bring new challenges.

According to [6], non-IID settings can be categorized into five distinct settings: feature distribution skew, label distribution skew, same label with different features, same features with different labels, and quantity skew. Previous studies [6, 11] have highlighted the impact of non-IID data among participating clients on the effectiveness and convergence speed of the federated model. As the heterogeneity among clients increases, the optimization of the global model deviates even further from the local objectives[12, 27]. Consequently, relying solely on a single global model proves inadequate in meeting the performance requirements of diverse clients. To cater to the specific data distribution of individual clients, researchers have introduced the concept of training multiple federated models in parallel, addressing the limitations of a single model [3, 5, 23]. Notably, personalization and clustering-based algorithms leveraging client similarity have emerged as promising approaches in this regard, which often assume all clients deal with similar learning tasks and with clustered structures [25]. However, faced with the scenario described in Figure 1, these conventional approaches may not be effective.

In that scenario, each client encompasses both majority and minority classes, with highly skewed label distributions in both amounts and kinds across different clients. Simultaneously, the number of samples labeled the same on different clients exhibits a complementary relationship. Such an assumption is common in the real world. For example, the zoos located in China and Australia may both keep pandas and kangaroos with different numbers, where pet dogs seldom appear even though they are very common. Under such settings, similarity-based personalization methods [10, 14] that incorporate the global federated model into local training may inadvertently introduce noise for clients that have a limited number of specific labels. Similarly, due to the distribution variations among multiple clients, clustering-based optimization algorithms [17, 20, 21, 29] also struggle to accurately identify similar members, resulting in poor performance of the generated clustering models.

So how to find the best peer(s) that a personalized model can learn from? Especially in the given scenario where the heterogeneity arises from distribution-based label skew [16] and variations in majority-minority class distribution within individual clients [26]? In this paper, to tackle this issue, we present a customized federation workflow *FedCOM*, which consists of two crucial components: *Complementary Client Matching* strategy based on neural network interpretability and *Personalized Federated Learning* with regularization terms. We summarize our contributions below:

1. We point out the limitations of current personalized federation algorithms when selecting "peers" in the presence of extreme label offset scenarios. We argue that simply incorporating the global model into local personalized training will introduce unwanted noise, meanwhile selecting similar models is ineffective under such circumstances.

2. We propose an innovative workflow called *FedCOM*, which can autonomously identify the best peers thus to maximize the accuracy of the personalized local model.

3. Experiments on MNIST, Fashion-MNIST, and CIFAR-10 datasets show that FedCOM can achieve faster convergence, with an average accuracy improvement of 1.35% compared to the local-only training strategy, as well as 44.19% and 1.28% to FedAVG and Ditto, which demonstrate the effectiveness and efficiency of our FedCOM.

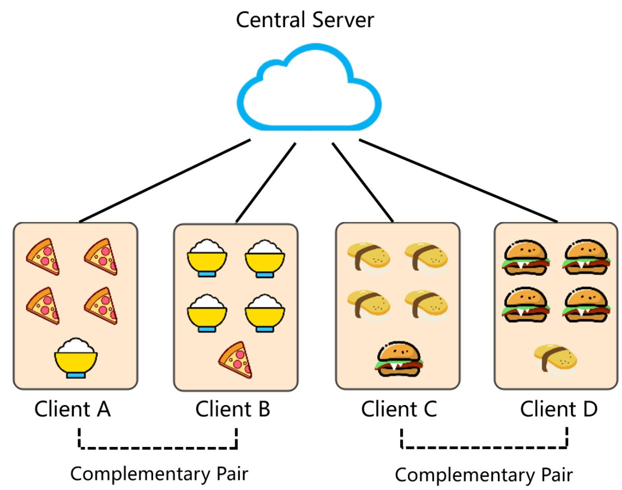


Figure 1: Real-world scenarios where FedCOM applies.

2 RELATED WORK

In this section, we provide an overview of previous research on personalization federation and neural network interpretability that are closely aligned with our work.

Personalized Federation. Personalized federation is a popular strategy aimed at producing highly customized models for different clients based on their local data distribution and requirements. MTL[13] employs a penalization optimization method to learn personalized models. The penalization term can capture the complex relationships between personalized models, and provide personalized models for completely unfamiliar participants. FedMSplit[3] is a personalized federated learning algorithm that dynamically captures the graph structure to adapt to different client types. This approach effectively addresses the challenge of inconsistent modalities encountered in multi-modal federated learning scenarios. DITTO [10] tackles the issues of fairness and robustness by introducing regularization terms into the local objective function. By adjusting the similarity between the local model and the global federated model, DITTO achieves personalized federated models while effectively mitigating fairness and robustness concerns. Marfoq et al.[14] exploited the ability of deep neural networks to extract high-quality vector representations (embeddings) from non-tabular data. They introduced a personalization mechanism based on local memorization, leveraging these embeddings to enhance model customization at the individual client level. This approach involves interpolating a pre-trained global model with a KNN model, utilizing the shared representation derived from the global model. Compared to the aforementioned personalized federated learning algorithms, Oort[8] optimizes the federated model by selecting participants with higher utility values to participate in the next round of federated training before each training round begins. By adjusting the utility value calculation formula, an optimal balance can be achieved between fairness and model effectiveness.

Neural Network Interpretability. How neural networks work is a vital topic in the field of artificial intelligence. Over the years, massive notable works have emerged, delving into the inner workings and the reasons why deep neural networks have such excellent feature representation ability. Examples of these works include Activation Maximization [18], Layer-wise Relevance Propagation [1], Class Activation Map[28], etc. In this part, we only focus on research surrounding Class Activation Mapping (CAM). The phenomenon of class activation mapping (CAM) in Convolutional Neural Networks (CNNs) was first proposed by [28]. They

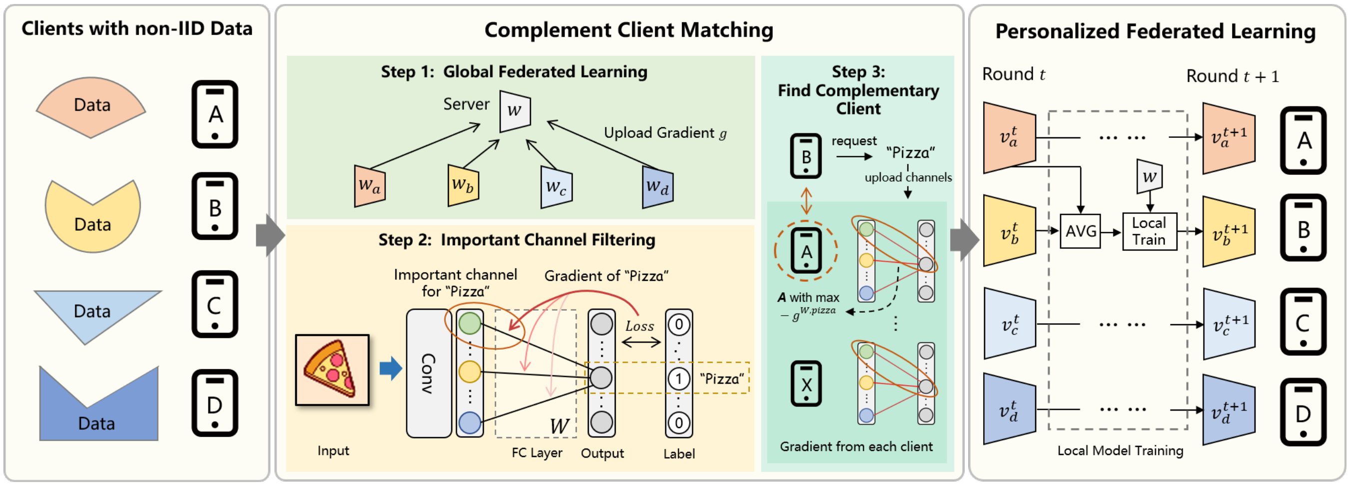


Figure 2: The workflow of FedCOM.

demonstrated this by employing a weighted summation of feature maps derived from the last convolutional layer. The generated activation values (non-zero values after ReLU activation) can highlight the regions in the image where objects are located. To overcome the constraints imposed by CAM on the number of fully connected layers and model architectures, [22] proposed an alternative method called Grad-CAM with greater flexibility and adaptability. This approach performs backward propagation through the logits of the target class to obtain gradients related to output feature maps from the last layer. These gradients are then used as weights for the corresponding feature channels. Grad-CAM has provided valuable insights and inspiration for our client-matching strategies. The rest of the work also includes HiResCAM[4], GradCAM++[2], AblationCAM[19], etc.

3 FEDCOM DESIGN

As shown in Fig 2, FedCOM workflow consists of two key steps: *Complementary Client Matching* and *Personalized Local Training* with regularization terms. In our method, each client $k \in [K]$ maintains a global model w for global federated learning, and a local model v_k with the local objective. First, we execute a global federated learning step for w and exploit the loss gradients for the target class to find complementary client pairs for client k. Then, we average the weight of v_k with that of the complementary pairs and update it via the local dataset with the regularization of w.

3.1 Complementary Client Matching

For the last MLP layers, the channels with larger weights are more significant for the classification task, for higher logits value would be gained when target patterns are recognized. In this perspective, we propose to match the complementary clients via channels.

For client k, we denote the collection of minority classes on it as C_{minor}, and the federated loss at t round as L_t. During the federated optimization, the clients who maintain large amounts of data of $c_i \in C_{minor}$ would contribute much in the significant channels of c_i, so these most important channels can be recognized and the complementary clients as well during training. As we optimize w by $w_{t+1} = w_t - \eta \times g$, the large contribution means minor gradient value. Then, our objective is to find clients that minimize the gradient sum over important feature channels.

Specifically, the whole process of client matching includes the two following steps as shown in :

1. At the beginning of t round federation, client k optimize the global model w with its own training data locally. During backward, the gradients of each parameter can be calculated. Given the target class c_i, we can get the partial derivatives on the classifier layer W, as well as the gradients of channels related to c_i. Then, we filter out the minimum top-x channels in gradients, whose index set constitutes $\{idx\}_k$. At the end of the round, client k uploads the channel set and model updates to the central server.

2. Once the server receives the channel set $\{idx\}_k$ uploaded by the requester client k, it begins to search who has the minimum sum of gradients on these channels among all clients. Then the target complementary pairs of client k can be found.

3.2 Personalized Federated Learning

After the global federated training, model w should fit:

$$w \in \arg\min_{w} G(F_1(w), \dots F_K(w)) \tag{1}$$

Algorithm 1 Complementary Client Matching

Input: $W, \{W_k\}_{k\in[K]}, c_i$

Each client calculates its gradient of classifier layer W :

for $k \in [K]$ **do**

  $g_k := W - W_k$

end for

Client k sends g_k and the target class c_i to the server

Server fetches the important channel set of c_i

  $\{idx\}_k = index(\text{top-}x_{small}(g_k(i)))$

find a client k_{com} for client k:

  $k_{com} = \arg\min_{p\in[K]} \sum_{\{idx\}_k} g_p(i, idx)$

return k_{com}

where $F_k(w)$ is the local objective for client k, and $G(\cdot)$ is a function that aggregates the local objectives $F_k(w)_{k\in[K]}$ from each clients.

Following previous work [10], we keep a local model for personalized tasks for data heterogeneity. To merge the contribution from different clients, we conduct a partial federated learning between the complementary pairs on their local models. For client k and one of its complementary client k_{com}, we average their local model v_k and $v_{k_{com}}$:

$$v_k := \text{MEAN}(v_k, v_{k_{com}}), \tag{2}$$

then the new v_k gains an initial weight of the client k_{com}.

Then, since DITTO has achieved great success in personalized learning by global regularization, we follow it by introducing a regular term with w into local training. The optimized object for client k is as follows:

$$\min_{v_k} \; h_k(v_k; w) := F_k(v_k) + \frac{\lambda}{2}||v_k - w||^2 \tag{3}$$

The degree to which the local model approaches the complementary model is controlled by the hyperparameter λ.

The interpolation between the local model and the complementary model becomes smaller. In this way, personalized training not only preserves model updates contributed by majority classes locally, but also incorporates information related to the minority class from other clients.

Algorithm 2 Personalized Federated Learning

Input: $T, F_k(w), G(\cdot), \eta, \lambda, w^0, \{v_k^0\}_{k\in[K]}, s$

for $t = 1, \cdots, T-1$ **do**

  for client $k \in [K]$ **do**

    Solve the local sub-problem of $G(\cdot)$ starting from w^t to obtain w_k^t:

      $w_k^t \Leftarrow \text{UPDATE\_GLOBAL}(w^t, \nabla F_k(w^{t-1}))$

    Send $\Delta_k^t := w_k^t - w^t$ back

    Search complementary client k_{com} for k

    Combine local model with k_{com}

      $v_k \Leftarrow \text{MEAN}(v_k, v_{k_{com}})$

    Update v_k for s local iterations:

      $v_k = v_k - \eta(\nabla F_k(v_k) + \lambda||v_k - w^t||)$

    Send v_k back to the server

  end for

  Server aggregate $\{\Delta_k^t\}$:

    $w_k^{t+1} \Leftarrow \text{AGGREGATE}(w^t, \{\Delta_k^t\}_{k\in[K]})$

end for

return $\{v_k\}_{k\in[K]}, w^T$

To simulate realistic scenarios, we assigned one majority and one minority class to each client and deliberately created highly imbalanced data distributions among them. During the test, the data of these two classes were with the same amount. We set the number of clients to 10 and grouped them into 5 complementary pairs with all data of two classes. For each pair, the training data is complementary to another. Then, the whole dataset can be shared without overlap.

For MNIST and Fasion-MNIST, the percentage of majority and minority classes are 95% and 5%, and for CIFAR-10 the rates are adjusted to 80% and 20% due to a higher training difficulty. Considering the task of binary classification is a bit simple, we added one noisy class for each client with the same amount of the minority class in training data to enhance model learning.

4.1.2 Details. Our FL system was implemented using PyTorch. Each client adopted a local batch size of 32, a learning rate of 0.001, and the SGD optimizer with a momentum of 0.9 for weight updates. Regarding personalized training, we set the hyperparameter λ to 0.01 both for Ditto and our FedCOM. During the complementary client matching step, we take $x = 5$ for channel filtering. All experiments are running with a single Intel i7-9750H CPU.

4 EXPERIMENTS

In this section, we provide empirical evidence to substantiate the effectiveness and performance improvements of FedCOM.

4.1 Setups

4.1.1 Fundamental Settings. We chose image classification as the central task and utilized a modified LeNet [9] as our CNN network. The experiments encompassed three popular datasets: MNIST[16], Fashion-MNIST[24], and CIFAR-10[7].

4.2 Effectiveness of FedCOM

We compare our method with pure local training, FedAVG, and Ditto. The average accuracy of each method is listed in Table 1. The accuracy of FedCOM not only exceeds the local model independently trained by a single client but also the

Table 1: Accuracy evaluation on MNIST, Fashion-MNIST, and CIFAR-10 datasets.

Methods	MNIST	Fashion-MNIST	CIFAR-10
local-only	0.9567	0.9462	0.7326
FedAVG	0.6095	0.4975	0.2432
Ditto	0.9661	0.9402	0.7321
FedCOM (Ours)	**0.9738**	**0.9519**	**0.7503**

Table 2: Matching accuracy of our *Complementary Client Matching* algorithm.

Top-x	MNIST	Fashion-MNIST	CIFAR-10
1	94.5 %	96.7 %	97.0 %
5	98.5 %	100 %	98.3 %
20	100 %	100 %	97.7 %

personalized federated method Ditto, which also proves the effectiveness of our complement-based strategy.

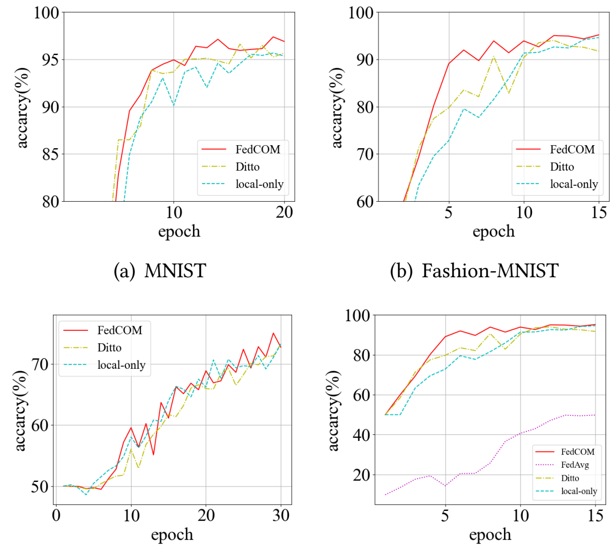
Although the pure local training strategy also reaches a high accuracy, the problems are that the convergence is slower and the model will overfit due to the lack of minority classes, especially when the minority class is with extremely little data and the model is big. However, a small model can even not be initiated well in some cases, which brings a conflict to the pure local strategy. And since the test data is equally distributed for the two classes, at least about 50% accuracy can be reached easily from the majority class. In that case, the improvement from our FedCOM for the minority class is obvious and important. The results demonstrate that when facing a high heterogeneity among clients, FedCOM can meet the accuracy requirements of each client as much as possible while providing privacy protection for each client.

4.3 Convergence

As shown in Fig 3, FedCOM stands out from other baselines due to its outstanding performance in converging faster while maintaining higher accuracy, which shows the efficiency of our method. Specifically, we can notice the following two phenomena. The accuracy of FedCOM on validation sets has the most obvious upward trend at the beginning of training and ranks higher than other baselines in most epochs. These phenomena prove that FedCOM can learn the updates caused by the target class more selectively from complementary models.

4.4 Accuracy of Client Matching

We count the accuracy of our complementary client matching algorithm during optimization to demonstrate its effectiveness, as shown in Table 2. The results illustrate that our method can have high accuracy in finding out the potential



(a) MNIST (b) Fashion-MNIST

(c) CIFAR-10 (d) FMNIST with FedAVG

Figure 3: Validation accuracy during training. Our FedCOM outperforms other baselines with the fastest convergence as well as the highest accuracy.

complementary client and is robust to the hyperparameter x under federated learning. Besides, our method works well at every stage, brought by our gradient-based design.

5 CONCLUSION AND FUTURE WORK

We present a personalized federation workflow FedCOM from the perspective of complement, specially designed for the real-world scenario involving label shifts, quantity variations and complement relationships in data distributions between clients. We first propose an interesting and efficient complementary matching method based on the gradients of feature channels, and then fuse the local models with a partial federated learning step. Following previous regular-based work, we also utilize the global federated model as a regularization in local personalized learning. Experiments show that FedCOM can bring obvious progress in helping minority class learning and outperforms several famous federated methods like FedAVG and Ditto.

In future work, we expect to further optimize the client-matching strategy to minimize the impact of pre-training federation rounds on the performance of FedCOM.

REFERENCES

[1] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015), e0130140.

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 839–847.

[3] Jiayi Chen and Aidong Zhang. 2022. FedMSplit: Correlation-Adaptive Federated Multi-Task Learning across Multimodal Split Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 87–96.

[4] Rachel Lea Draelos and Lawrence Carin. 2020. Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks. *arXiv e-prints* (2020), arXiv–2011.

[5] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7865–7873.

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

[7] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[8] Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. 2021. Oort: Efficient Federated Learning via Guided Participant Selection.. In *OSDI*. 19–35.

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

[10] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*. PMLR, 6357–6368.

[11] 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.

[12] Xin-Chun Li and De-Chuan Zhan. 2021. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 995–1005.

[13] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. 2021. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems* 34 (2021), 15434–15447.

[14] Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. 2022. Personalized federated learning through local memorization. In *International Conference on Machine Learning*. PMLR, 15070–15092.

[15] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[16] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[17] Manan Mehta and Chenhui Shao. 2023. A Greedy Agglomerative Framework for Clustered Federated Learning. *IEEE Transactions on Industrial Informatics* (2023).

[18] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems* 29 (2016).

[19] Harish Guruprasad Ramaswamy et al. 2020. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 983–991.

[20] Yichen Ruan and Carlee Joe-Wong. 2022. Fedsoft: Soft clustered federated learning with proximal local updating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8124–8131.

[21] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems* 32, 8 (2020), 3710–3722.

[22] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[23] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated multi-task learning. *Advances in neural information processing systems* 30 (2017).

[24] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[25] Jian Xu, Xinyi Tong, and Shao-Lun Huang. 2023. Personalized Federated Learning with Feature Alignment and Classifier Collaboration. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=SXZr8aDKia

[26] Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. 2022. Federated learning with label distribution skew via logits calibration. In *International Conference on Machine Learning*. PMLR, 26311–26329.

[27] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).

[28] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.

[29] Huan Zou, Yuchao Zhang, Xirong Que, Yilei Liang, and Jon Crowcroft. 2022. Efficient federated learning under non-IID conditions with attackers. In *Proceedings of the 1st ACM Workshop on Data Privacy and Federated Learning Technologies for Mobile Edge Network*. 13–18.