MetaVers: Meta-Learned Versatile Representations for Personalized Federated Learning

Jin Hyuk Lim[1*†] SeungBum Ha[2*] Sung Whan Yoon[2‡]
[1]POSCO N.EX.T Hub, Applied AI Research Cell, Republic of Korea
[2]Graduate School of Artificial Intelligence, Ulsan National Institute of Science and Technology (UNIST),
Republic of Korea
jinout@posco-inc.com {ethereal0507, shyoon8}@unist.ac.kr

Abstract

One of the daunting challenges in federated learning (FL) is the heterogeneity across clients that hinders the successful federation of a global model. When the heterogeneity becomes worse, personalized federated learning (PFL) pursues to detour the hardship of capturing the commonality across clients by allowing the personalization of models built upon the federation. In the scope of PFL for visual models, on the contrary, the recent effort for aggregating an effective global representation rather than chasing further personalization draws great attention. Along the same lines, we aim to train a large-margin global representation with a strong generalization across clients by adopting the meta-learning framework and margin-based loss, which are widely accepted to be effective in handling multiple visual tasks. Our method called `MetaVers` achieves state-of-the-art accuracies for the PFL benchmarks with the CIFAR-10, CIFAR-100, and CINIC-10 datasets while showing robustness against data reconstruction attacks. Noteworthy, the versatile representation of `MetaVers` exhibits a strong generalization when tested on new clients with novel classes. Code is available at https://github.com/eepLearning/MetaVers.

1. Introduction

Recently, a massive number of data samples are being created at distributed devices such as wireless smart devices, connected self-driving cars, and other edge nodes. Collecting private data samples from these decentralized nodes to a centralized server for training models raises serious concerns about data privacy. As one of the ways to resolve the privacy concern, federated learning (FL) has been proposed to train a single yet effective global model by aggregating local models from clients [26] while keeping data on the client side.

However, such advances in decentralized learning framework may fail to acquire a generalized model when the data distribution across clients is widely diverse, so-called heterogeneity across clients. To relieve the challenge, personalized federated learning (PFL) aims to train a personalized model for each client by leveraging the benefits from the federation across clients rather than pursuing a single global model.

A group of PFL methods is based on regularizing the gap between personalized models and the global model. For instance, pFedMe [9] regularizes the gap in parameters between personalized and the global model, and FedProto [37] focuses on the representation space by regularizing the locally computed per-class averaged features not to diverge far from the globally computed per-class averaged features.

Another branch of methods aims to devise model-based approaches. FedPer [2] and LG-FedAvg [23], explicitly divide learnable parameters into local and global, i.e., the local parameters are optimized at each client, but the global parameters are trained via federation across clients. Other methods called pFedHN of [34] and pFedGP of [1] introduce auxiliary networks that are trained over clients which facilitate the construction of client-specific local models. Most of the prior PFL algorithms focus on acquiring sufficient personalization of local models based on the assistance of the federation that leverages benefits across clients. It has been believed that the training of a global model is sub-optimal for handling heterogeneity across clients.

A few recent works [6, 25, 28] argue the importance of the successful aggregation of a global model that overcomes heterogeneity across clients. FedRep [6] and FedBABU [28] try to learn a shared representation across clients by decomposing the training of the feature extractor and classifiers, i.e., federation takes place only for the extractor, and the classifier is personalized for each client. In the work of FedRep, authors first claim that the global representation fully leverages the commonality across clients and can be easily generalized

[*] Equal contribution
[†] He contributed to this work when he was a graduate student at UNIST.
[‡] Sung Whan Yoon is the corresponding author.

to new clients with novel classes. FedBABU [28] reveals that the federation of the extractor so-called ‘body’ is the key to federated learning of deep models across clients. Once the body is trained across clients, further steps of personalization on each client with a personalized classifier achieve a noticeable performance than a naive case where the federation takes place for a whole model. Along the same lines, kNN-Per of [25] adopts k-nearest neighbors as the personalized classifiers while a global extractor is learned across clients. The stream of approaches claims the importance of a global representation that learns the way of the common feature extraction of the heterogeneous regime. However, these works only attempt to decouple the training of representation and classifiers for federated learning but do not employ particular learning strategies that are known to be effective for capturing the commonality across heterogeneity.

In this paper, a method called MetaVers exploits two powerful methodologies to achieve a versatile representation via federation: meta-learning and margin-based learning, which are confirmed to be effective for acquiring strong generalization for multi-task settings and visual representation learning, respectively. To tackle the heterogeneity across clients, MetaVers borrows the concept of distance-based meta-learning such as ProtoNet of [36], which aims to train a common feature extractor for different classification tasks. Each local few-shot episode for each client plays a distinctive classification task so that the aggregation of local gradients across clients leads to the meta-learning of the representation. Along with the meta-training framework, MetaVers encourages the large margin representation via adopting centroid triplet loss of [15] in computing local gradients. To prevent dense representation from local episodic training of scarce data, a server encourages all clients to learn sufficiently large-margin embeddings by sending the increasing margin values for the centroid triplet loss term that enables dynamic margins of the representation space.

In the extensive simulations, MetaVers achieves state-of-the-art performance on the standard PFL benchmarks based on the CIFAR-10, CIFAR-100, and CINIC-10 datasets. Also, we demonstrate the versatility of MetaVers to classify out-of-distribution data samples by measuring the performance of a newcomer client with novel classes that have not been trained at all. In the privacy perspective, MetaVers shows the robustness under the popular model inversion attack such as Deep Leakage from Gradients (DLG) [43].

2. Related Work

After the work of [26] called FedAvg, which is widely accepted as a baseline of federated learning (FL), immense efforts have been dedicated to tackling the heterogeneity across clients. We categorize related works into three parts: **i)** FL with heterogeneous clients whose goal is to achieve higher accuracy on an overall test split with a single global model, **ii)** Personalized federated learning (PFL) that pursues to improve the local performance of each client and **iii)** Distance-based meta-learning.

2.1. Federated Learning with Heterogeneity

The heterogeneity across clients, also known as the non-independent and identically distributed (non-IID) setting, is shown to hinder the aggregated model from converging to the optimal global model [18,22,32]. When the heterogeneity becomes severe, locally trained model parameters of different clients largely diverge from each other so that the accuracy of the FL dramatically deterioates [42]. To solve this problem, various types of approaches have been proposed by extending FedAvg baseline [16, 21, 27, 30, 32, 39, 42]. The work of [42] claims that a small set of shared data is sufficient to prevent the divergence of local models. FedAvgM [16] considers the momentum of gradients to regularize the dramatic change in the global model. FedMA [39] aggregates a selected part of parameters, i.e., layer-wise federation inspired by Probabilistic Federated Neural Matching [41]. These early works for handling non-IID settings rather focus on regularizing the local training to guarantee the convergence of the global model. In contrast, MetaVers explicitly adopts meta-learning that enables training a common model rather than regularizing the diversity of local training across clients. FedAwS [40] aggregates both the model parameters and the class embeddings. This method utilizes the positive term of the contrastive loss to update the local model. However, sharing the class embeddings, which can contain a private data, contravenes a privacy perspective of FL scenario.

2.2. Personalized Federated Learning (PFL)

PFL allows each client to prepare its own personalized model by taking the benefits from federation across clients.

Employing auxiliary models: A branch of methods employs the auxiliary networks, which facilitate the personalization of each client’s model. pFedHN [34] learns a globally federated hypernetwork that aims to generate a personalized model for each client. Another method called pFedGP [1] combines Gaussian processes with the PFL framework to achieve an effective deep kernel function across all clients.

Splitting model architecture: Another group of approaches tries to split models into parts and handle them separately in federation. Some prior works [2, 8, 13, 23] combine a certain part of the aggregated model with local models. Other works divide the model architecture into local and global training parts: FedPer [2] locally trains a personal classifier layer with a globally trained feature extractor. In contrast, LG-FedAvg [23] locally trains feature extractors with a global classifier for reduced communication burden.

Using prototypes: A method called FedProto [37] aggregates per-class averaged features, i.e., prototype, instead of gradients. In this setting, each client trains its personal model

	Method							
	FedAvg [26]	FedRep [6]	kNN-Per [25]	FedBABU [28]	pFedGP [1]	Per-FedAvg [17]	FedProto [37]	**MetaVers**
Global body†	✗	✓	✓	✓	✗	✗	✗	✓
Head type*	Weights	Weights	Weights	Weights	Weights	Weights	Prototypes	Prototypes
Fine-tuning#	✗	✗	✗	✓	✗	✓	✗	✗

† means whether a method trains a global representation via decoupling body and head.

* 'Weights' indicates normal classifier weights, and 'Prototypes' indicates per-class averaged features.

# means whether a method requires fine-tuning steps through gradient computation in testing.

Table 1. Comparison between MetaVers and the existing PFL methods

by utilizing the aggregated global prototype to regularize local prototypes closer to the global one.

Adopting multi-task training: When interpreting the training of each local client as 'task', then the learning strategies to handle multiple tasks can be borrowed to improve PFL methods. For instance, MOCHA [35] and FedU [10] adopt multi-task learning while training distinct but similar personal models. Another group of methods interprets PFL as transfer learning, so they aim to transfer knowledge from the global model to local models [5, 20]. By expanding the multi-task learning viewpoint to meta-learning which pursues to extract the generalized knowledge from the given task distribution, the optimization-based meta-learning is utilized to tackle PFL [4, 11, 17]. Among them, Per-FedAvg [11] understands the objective of PFL as closely related to the bi-level optimization setting of Model-Agnostic Meta-Learning (MAML) of [12]. Our MetaVers and Per-FedAvg are relevant in view of meta-learning, but they show significant differences in both technical and philosophical viewpoints.

Pursuing global representation: In very recent works of [6, 25, 28], researchers emphasize the importance of the shared representation in the following perspectives: i) The successful federation of a global model over heterogeneity fully captures the commonality across clients [6]. ii) The global representation enables rapid personalization via a small number of updates [28]. iii) Moreover, the representation can work well even for a newcomer client who has never participated in the federation [6, 25]. Their strategies are based on decoupling the training of the representation part and the classifier part, which are called 'body' and 'head', respectively. FedRep of [6] freezes locally-trained classifiers while training the feature extractor, then the parameters of the extractor are aggregated across clients. FedBABU of [28] never learns the classifiers but only trains the feature extractor during the federation. kNN-Per of [25] adopts k-nearest neighbors as the classifier and trains the global feature extractor across clients.

2.3. Novelty of MetaVers over Relevant Works

Comparison to prototype-based method: In the view of utilizing prototypes, FedProto [37] seems to be closely related to our work, but there is a significant difference, i.e., FedProto does not aggregate model parameters but collects the global prototypes as the tool for regularizing local training so that FedProto cannot obtain a meta-trained model.

Comparison to Per-FedAvg: Per-FedAvg [11] is built upon the concept of optimization-based meta-learning which requires fine-tuning steps. Due to the nature of optimization-based approaches, Per-FedAvg requires further fine-tuning of the global model to work successfully on the client side. When comparing with our MetaVers, the global model of Per-FedAvg cannot work as a generalized representation because it relies on local fine-tuning. In contrast, MetaVers is based on distance-based meta-learning such as Prototypical Networks of [36], which focuses on the training of the global feature extractor so that the shared embedding of MetaVers is capable of acquiring the sufficient personalization at each local client without an additional update.

Comparison to global representation method: As the prior works including FedRep [6], kNN-Per [25] and FedBABU [28], MetaVers never trains local classifiers but computes prototypes for each local episode as the classifiers so that the federation is focused on finding a global representation. The main difference is that MetaVers employs two explicit methodologies to fully aggregate the commonality across clients, i.e., the meta-learning framework and large-margin loss. As a result, MetaVers outperforms the existing methods in the PFL benchmarks and shows noticeable performance for a newcomer client who contains novel categories that have not been trained before. We emphasize that MetaVers offers the advanced method to train a strong global representation across clients.

Table 1 shows the comparison between MetaVers and the existing PFL methods from three different viewpoints: the federation of a global body, head classifier type, and fine-tuning steps for personalization in testing.

2.4. Distance-based Meta-Learning

When saying the relevant method, Prototypical Networks [36] train a representation across widely-varying classification, where the features from the same class are concentrated to the class-specific averaged features so-called prototype. MetaVers extends the concept of Prototypical Networks into the PFL settings to train a representation across varying episodes from different clients. Each client of MetaVers processes its episodes that resemble Prototypical Networks, and the computed gradients are aggregated at the server.

3. Proposed Method

3.1. Problem Setting of PFL

In the PFL setting, the number of distributed clients is n. Each client contains the local dataset $\mathcal{D}_i$, which follows the data distribution $\mathcal{P}_i$. The embedding model for feature extraction is $f(\,\cdot\,;\theta)$ with model parameter θ, and the classifier weight is ϕ_i for client i. The local loss computed at i-th client is $\mathcal{L}(f(x;\theta_i),\phi_i,y)$. Then the objective function of PFL is to find the client-specific model parameters $\{\Theta^*,\Phi^*\} = \{\theta_i^*,\phi_i^*\}_{i=1}^n$ that minimizes averaged local loss values across clients. For the purpose of finding a shared representation across clients, the feature extractor weight should be a single global model, i.e., $\theta^* = \theta_i^*$ for all $i \in [n]$. In addition, MetaVers does not require training classifier weights, so we can further drop the classifier weights in the objective function: $\theta^* = \operatorname{argmin}_\theta \frac{1}{n}\sum_{i=1}^n \frac{1}{|\mathcal{D}_i|}\sum_{(x,y)\sim\mathcal{P}_i}\mathcal{L}\big(f(x;\theta),y\big)$.

3.2. Proposed Algorithm: MetaVers

Our proposed algorithm, MetaVers, handles repetitive communication rounds in the same way as FedAvg of [26].

Intialization: At the beginning of round $\tau = 1,\cdots,T$, a client receives a global embedding network $f(\,\cdot\,;\theta^{(\tau)})$ parameterized by $\theta^{(\tau)}$. Also, the central server transmits the round-specific distance margin value $\mathfrak{m}_{\text{global}}^{(\tau)}$ to every client.

Local update at each client: In each round, each client constructs its own training episode $\mathcal{E}_i^{(\tau)}$. The episode is generated by sampling N local classes and their labeled samples. First of all, the support set $\mathcal{S}$ with K samples per class and the query set $\mathcal{Q}$ with Q samples per class are sampled. The prototype $\mathbf{c}_k$ for local class k is computed by taking the average of feature vectors from support samples:

$$\mathbf{c}_k = \frac{1}{|\mathcal{S}_k|}\sum_{x\in\mathcal{S}_k} f(x;\theta^{(\tau)}), \tag{1}$$

where $\mathcal{S}_k$ is the subset of $\mathcal{S}$ whose label is $k \in \{1,...,N\}$.

Each client calculates the Euclidean distance-based cross-entropy loss $\mathcal{L}_S$ by measuring the distance between the feature vectors of query samples and the class prototypes, i.e.,

$$\mathcal{L}_S = \frac{1}{|\mathcal{Q}|}\sum_{k=1}^{N}\sum_{x\in\mathcal{Q}_k}\Big[d(f(x;\theta^{(\tau)}),\mathbf{c}_k) + log\sum_{l=1}^{N}\exp\left(-d(f(x;\theta^{(\tau)}),\mathbf{c}_l)\right)\Big]. \tag{2}$$

To promote a well-clustered representation, MetaVers employs a particular aggregation of margins values. First, each client computes the class centroid $\mathbf{a}_k$ which is the per-class averaged features of the samples in its local episode, including supports and queries. The average distance between different centroids, so-called local distance margin $\mathfrak{m}_i^{(\tau)}$ is then computed:

$$\mathfrak{m}_i^{(\tau)} = \frac{1}{(N-1)^2}\sum_k\sum_{l\neq k} d(\mathbf{a}_k,\mathbf{a}_l) \tag{3}$$

The client-specific margin value $\mathfrak{m}_i^*$ for the triplet loss is obtained by taking the larger value among the local and global margins, i.e., $\mathfrak{m}_i^* = \max\{\mathfrak{m}_{\text{global}}^{(\tau)},\mathfrak{m}_i^{(\tau)}\}$. For centroid triplet loss, the centroid $\mathbf{a}_k$ is used as the anchor point. The positive points are sampled from the features of the support and query samples from class k. The negative points are samples from the feature vectors of supports and queries from other classes. Then for a triplet $(\mathbf{a}_k, x_p\in\mathcal{S}_k\cup\mathcal{Q}_k, x_n\in\mathcal{S}_l\cup\mathcal{Q}_l)$ where $l\neq k$, the loss can be calculated:

$$\mathcal{L}_T(\mathbf{a}_k,x_p,x_n) = \max\Big\{d\big(\mathbf{a}_k,f(x_p;\theta^{(\tau)})\big) - d\big(f(x_p;\theta^{(\tau)}),f(x_n;\theta^{(\tau)})\big) + \mathfrak{m}_i^*, 0\Big\}. \tag{4}$$

By considering all cases, the triplet loss term $\mathcal{L}_T$ is obtained: $\mathcal{L}_T = \sum_k\sum_{(x_p,x_n)}\mathcal{L}_T(\mathbf{a}_k,x_p,x_n)$. The computational overhead of centroid triplet loss for MetaVers is: $O(cN^2)$, where c is the number of classes and N is the number of samples. It seems to be burdensome at a glance, because it is square of the number of the sample. The reason why we adopt triplet-based loss is that triplet loss is directly designed to enlarge the margin between classes in a sample-by-sample way. Also, using a single anchor point, which is the per-class centroid, reduces the number of pairs to be considered when compared with conventional triplet loss of [33]. Our triplet loss computation is similar to the triplet-center loss proposed in the work of [15]. The difference is that we consider entire feature vectors from different classes in the negative pair terms but the triplet-center loss of [15] utilizes different class centroids as negative points. That makes MetaVers enlarge the margin more strongly.

MetaVers makes each client adopt a larger margin than the local distance. The round and client-specific margin value guides each client to acquire better representation, i.e., when the local embedding is already well-separated and clustered than the global margin, then the client adopts its own local distance $\mathfrak{m}_i$ as the margin, otherwise, the client takes the global margin value $\mathfrak{m}_{\text{global}}$ from the server to promote the local embeddings to show the larger margin separation.

By combining cross-entropy loss $\mathcal{L}_S$ and triplet loss $\mathcal{L}_T$, the embedding network $f(\,\cdot\,;\theta^{(\tau)})$ is then updated:

$$\theta_i^{(\tau)} \leftarrow \theta^{(\tau)} - \eta\nabla(\gamma\mathcal{L}_S + (1-\gamma)\mathcal{L}_T), \tag{5}$$

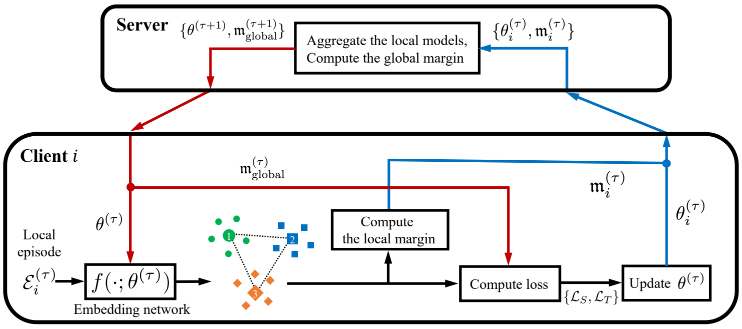

Server
$\{\theta^{(\tau+1)}, \mathfrak{m}^{(\tau+1)}_{\text{global}}\}$
Aggregate the local models,
Compute the global margin
$\{\theta_i^{(\tau)}, \mathfrak{m}_i^{(\tau)}\}$
Client i
$\mathfrak{m}^{(\tau)}_{\text{global}}$
$\mathfrak{m}_i^{(\tau)}$
$\theta^{(\tau)}$
$\theta_i^{(\tau)}$
Local episode
$\mathcal{E}_i^{(\tau)}$
$f(\cdot;\theta^{(\tau)})$
Embedding network
Compute the local margin
Compute loss
$\{\mathcal{L}_S, \mathcal{L}_T\}$
Update $\theta^{(\tau)}$

Figure 1. Federated learning process of `MetaVers`

where γ is a hyperparameter for balancing the cross-entropy loss and the triplet loss. The locally updated parameter $\theta_i^{(\tau)}$ and the average distance between centroids $\mathfrak{m}_i^{(\tau)}$ are then uploaded to the server when client i is active in this round.

Aggregation process at the server: The average of the aggregated local models from active clients is set to be the global model for the next round:

$$\theta^{(\tau+1)} \leftarrow \frac{1}{|C_\tau|}\sum_{i\in C_\tau}\theta_i^{(\tau)}, \tag{6}$$

where C_τ is a set of active clients during round τ. The server aggregates the local average distance values $\mathfrak{m}_i^{(\tau)}$ from the active clients to newly calculate the average distance values across clients. For a stable update, the server considers the past global margin values with a fixed interval of W rounds:

$$\mathfrak{m}^{(\tau+1)}_{\text{global}} \leftarrow \frac{1}{W}\Big\{\sum_{t=\tau-W+1}^{\tau-1}\mathfrak{m}^{(t)}_{\text{global}} + \frac{1}{|C_\tau|}\sum_{i\in C_\tau}\mathfrak{m}_i\Big\}. \tag{7}$$

Figure 1 shows the learning process. Also, the pseudocode is in Supplementary material.

Testing: After T rounds, each client utilizes the shared representation $f(\cdot;\theta^T)$ as a feature extractor without further optimization. Each client then computes the per-class averaged features, i.e., prototypes of its local classes, and uses them as classifiers. Queries are classified into the nearest prototypes by computing the Euclidean distance metric.

3.3. Effect of Increasing Margin on Embedding

Let us recall the triplet loss term of eq. (4). Consider a triplet $(\mathbf{a}_k, x_p, x_n)$ that produces non-zero triplet loss value. Without losing generality, let us assume that there are two different classes k and l in the given episode. Then the triplet loss term for the given triplet $(\mathbf{a}_k, x_p, x_n)$ becomes

$$\begin{aligned}\mathcal{L}_T &= d\big(\mathbf{a}_k, f(x_p;\theta)\big) - d\big(f(x_p;\theta), f(x_n;\theta)\big) + \mathfrak{m}_i^* \\ &= \|f(x_p;\theta)-\mathbf{a}_k\| - \|f(x_p;\theta)-f(x_n;\theta)\| + \mathfrak{m}_i^* \\ &\overset{(a)}{\le} \|f(x_n;\theta)-\mathbf{a}_k\| + \mathfrak{m}_i^*. \end{aligned}\tag{8}$$

The inequality (a) follows the fact that $\|x-y\| - \|x-z\| \le \|y-z\|$. When the global margin $\mathfrak{m}_{\text{global}}$ from the server is larger than the local average of the distance between centroids, i.e., $\mathfrak{m}_{\text{global}} > \mathfrak{m}_i = \|\mathbf{a}_k - \mathbf{a}_l\|$, then the loss is upper bounded as follows:

$$\begin{aligned}\mathcal{L}_T &\le \|f(x_n;\theta)-\mathbf{a}_k\| + \|\mathbf{a}_k-\mathbf{a}_l\| + \Delta \\ &\le \|f(x_n;\theta)-\mathbf{a}_l\| + \Delta\end{aligned}\tag{9}$$

where $\Delta = \mathfrak{m}_{\text{global}} - \mathfrak{m}_i > 0$. To suppress the bound, the local margin $\mathfrak{m}_i$ should be enlarged to the global margin $\mathfrak{m}_{\text{global}}$. Moreover, the distance between queries and the corresponding prototype should be reduced. It implies that the clients with less-separated class prototypes are encouraged to learn a better representation with a sufficient margin.

3.4. Convergence Analysis

Herein, we provide a brief result of the convergence analysis. The full description including mathematical definitions, assumptions and proofs are in the Supplementary material.

Basic notations: $\mathcal{E}_i$ represents a training episode at client i. $\theta^{(\tau)}$ indicates the global model parameter at the beginning of the round τ. $\theta_i^{(\tau)}$ means the locally updated model parameter after an episodic-training at client i in the round τ. Finally, $\mathcal{L}_i(\theta;\mathcal{E}_i)$ is the local loss value based on the model parameter θ and the given episode $\mathcal{E}_i$ from client i. Also, L is the L-smoothness of local loss function. $\alpha \in (0,1]$ and σ^2 are used for bounding the local gradients. Based on the notation, following Lemma and Theorem are guaranteed.

Lemma 1. *For every client $i \in [1,n]$, the difference of local losses at round $\tau+1$ and τ is bounded:*

$$\begin{aligned}&\mathcal{L}_i(\theta^{(\tau+1)}) - \mathcal{L}_i(\theta^{(\tau)}) \\ &\le (-\eta\alpha + \frac{1}{2}L\eta^2)\Big(\|\nabla\mathcal{L}(\theta^{(\tau)})\|^2 + \sigma^2\Big),\end{aligned}\tag{10}$$

where η is the learning rate of local update.

Theorem 1. (Convergence) *For any client $i \in [1,n]$ with a learning rate $\eta^* < \frac{2\alpha}{L}$, the local loss is a decreasing function in the number of rounds:*

$$\mathcal{L}_i(\theta^{(\tau+t)}) < \mathcal{L}_i(\theta^{(\tau)}). \tag{11}$$

	CIFAR-10			CIFAR-100			CINIC-10		
# clients	50	100	500	50	100	500	50	100	500
Local	86.0	82.9	75.9	51.4	45.6	31.2	60.7	58.3	50.8
FedAvg [26]	57.8	58.1	56.7	25.6	24.1	22.7	49.3	51.2	50.2
LG-FedAvg [23]	87.9	83.6	64.7	43.6	37.5	20.3	59.5	59.9	52.5
pFedMe [9]	86.4	85.0	80.3	49.8	47.7	32.5	69.9	68.9	58.8
FedProto [37]	85.9	79.0	51.0	47.8	17.8	10.9	58.2	40.3	26.0
Per-FedAvg [11]	71.1	79.1	67.7	38.2	34.1	32.8	53.8	53.5	59.6
pFedHN [34]	90.2	87.4	83.2	60.0	52.3	34.1	70.4	69.4	64.2
pFedGP [1]	89.2	88.8	87.6	63.3	61.3	50.6	71.8	71.3	68.1
FedPer [2]	83.8	81.5	76.8	48.3	43.6	25.6	70.6	68.4	62.2
FedRep [6]	82.4	80.7	77.3	45.1	38.8	30.2	67.1	64.7	61.5
kNN-Per [25]	89.6	89.5	84.8	61.8	56.0	38.7	71.8	72.0	69.2
FedBABU [28]	87.2	86.2	85.5	53.4	52.3	49.0	68.7	66.5	67.8
MetaVers (Ours)	**90.8**	**90.2**	**89.9**	**66.7**	**64.8**	**55.8**	**73.2**	**73.2**	**72.5**

The results with SEM (Standard Error of the Mean) are in Supplementary material.

Table 2. Test accuracy over 50, 100, 500 clients on CIFAR-10, CIFAR-100, and CINIC-10.

The mathematical claim guarantees the decreasing behavior of the local loss function, which directly implies the convergence of the PFL performance of MetaVers.

4. Experiments

MetaVers is evaluated on the various personalized federated learning (PFL) setups. Also, we run additional experiments to verify the strong generalization of the learned representation.

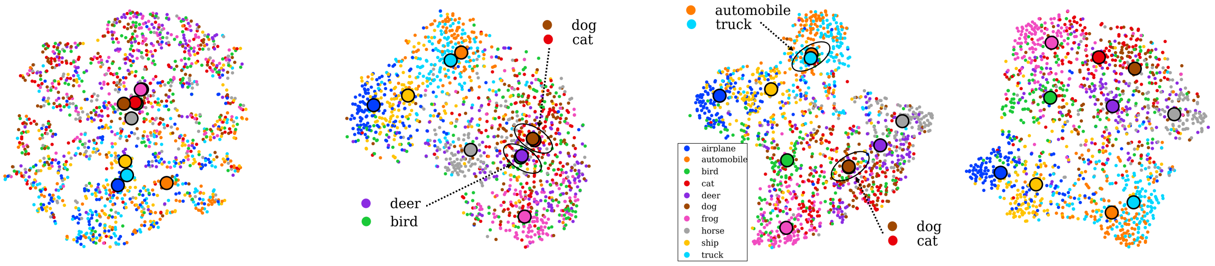
4.1. Evaluation on Standard PFL Benchmarks

We compare MetaVers with other methods on the recent PFL settings that are considered in [1, 34]. Among the existing PFL benchmarks, we select the setting in [1, 34] due to the following two reasons: i) a wide range of the number of clients and ii) a strongly limited number of active clients. We believe that the settings can reflect realistic decentralized training setups. The benchmarks are based on the following datasets: CIFAR-10, CIFAR-100, and CINIC-10 [7]. CINIC-10 is a larger dataset that collects samples from two datasets: CIFAR-10 and ImageNet [31]. In our experiment, we set the total number of clients to be 50, 100, and 500. Also, the number of active clients for each round is 5 for all cases. We set the total number of classes in each client to be 2, 4, and 10 classes for CIFAR-10, CINIC-10, and CIFAR-100, respectively. Details for dataset splitting and the non-IID settings are described in Supplementary material. In a recent work of [3], a PFL benchmark with CIFAR-10 is suggested by imposing heterogeneity through Dirichlet allocation. When compared to the Standard PFL benchmarks of [1, 34], less number of clients are given, i.e., 100 clients, and more active clients are allowed, i.e., 20 active clients. However, the degree of heterogeneity can be controlled by Dirichlet allocation. The evaluation on the benchmark is in Supplementary material.

Experiment Setups: By following the settings of [1, 34], we allow 1,000 server-client communication rounds. Five active clients are selected in each round besides pFedHN. For 'Local' method, each local model is allowed to be trained with 100 local episodes without communications. 'LG-FedAvg' requires extra 200 rounds after pretraining the FedAvg model via 1,000 rounds. 10 fine-tuning steps are used for FedBABU in testing.

Implementation: As done in [1, 34], we use a LeNet-based model [19] with two 2 x 2 convolutional layers where a 2 x 2 max-pooling layer follows each one. After the second max-pooling layer, two fully-connected layers and one classifier layer follow. For MetaVers, the last classifier layer is not used. Details on the optimizer and learning rates are in Supplementary material. For demonstrating how the algorithm scales in the size of model architecture, we additionally demonstrate MetaVers with the ResNet architecture [14] in Supplementary material.

Results: Table 2 shows the test accuracies averaged over three random seeds. We claim that **i)** MetaVers achieves state-of-the-art performance with considerable margins for all cases. **ii)** MetaVers are more solid in the case with more clients when compared to prior methods. We emphasize that the margins of accuracies between MetaVers and the runner-up algorithms are considerable when the number of clients increases, i.e., the gaps for 500-client cases are +2.3%, +5.2%, and +4.4% for CIFAR-10/100 and CINIC-100, respectively. In these cases, each client contains a very small number of samples where the local training suffers from overfitting. MetaVers is robust for this scenario because each client newly constructs a few-shot episode by



(a) Local (b) Triplet loss (c) Cross-Entropy (d) MetaVers

Figure 2. t-SNE for (a) Local (no federation) and MetaVers with (b) Only Triplet (w/ fixed margin), (c) Only Cross-Entropy and (d) MetaVers. Data is sampled from the CINIC-10 test set.

	CIFAR-100		
Method	Client 50	Client 100	Client 500
FedAvg-Proto [26]	62.4 ± 0.3	60.7 ± 0.2	50.1 ± 0.4
FedBABU-Proto [28]	58.9 ± 0.6	57.4 ± 0.6	43.5 ± 1.0
FedRep-Proto [6]	45.3 ± 2.0	42.4 ± 0.8	34.6 ± 0.7
pFedGP-Proto [1]	61.3 ± 0.2	60.3 ± 0.2	52.6 ± 0.1
MetaVers (Ours)	**66.7** ± **0.4**	**64.8** ± **0.3**	**55.8** ± **0.1**

Table 3. Testing with prototype-based classifiers ('-Proto')

	CIFAR-100 with Client 100
Method	nVAR
FedAvg [26]	1022.8
FedBABU [28]	1097.2
FedRep [6]	2031.4
MetaVers (Ours)	**769.1**

Table 4. Normalized VAR (nVAR) of global representations

picking a very small number of samples for each round rather than fitting to its whole dataset so that it prevents overfitting of local updates. **iii)** MetaVers achieves more dominant performance gains in more complicated datasets when compared to prior methods. MetaVers achieves outperforming performance in the CIFAR-100 cases with more diverse image categories, i.e., the gaps over the runner-ups are +3.4%, +3.5%, and +5.2% for 50, 100, and 500 client cases, respectively. This advantage is from the strength of meta-learning, which is more solid in training the shared knowledge of wide-range task distribution. **iv)** When compared with FedProto of [37], and Per-FedAvg of [11] that utilize prototype-based learning and optimization-based meta-learning, MetaVers shows outstanding performance. Also, we note that FedProto suffers from the limited number of active clients in our experiments, where the work of FedProto assumes full participation of clients at every round. **v)** For the prior methods with shared representations, including FedPer of [2], FedRep of [6], kNN-Per of [25], and FedBABU of [28], MetaVers is only the method that shows consistent gains for all benchmarks over the personal model-based PFL methods such as pFedHN of [34] and pFedGP of [1]. It confirms the superiority of the strong generalization capability of MetaVers to prior shared-representation-based approaches.

Evaluation with Prototype Classifiers: MetaVers does not have a learnable classifier, not only in inference but also in the training process. Instead, MetaVers computes local prototypes as the classifiers. We tested other methods with a shared representation, such as FedAvg, FedBABU, and FedRep, by changing their classifiers into prototype-based classifiers as MetaVers. We conjecture that the prototype-based classification reflects how much the global representation is intra-class compact and inter-class separated. Also, pFedGP is included as a cutting-edge PFL algorithm with personal models. As shown in Table 3, MetaVers shows the best performance implying that the representation capability of MetaVers is outperforming.

Representation Analysis: To quantify how much the global representation shows well-clustered feature distribution, we computed normalized Variance (nVAR), which is the mean of per-class variance of features divided by the squared distance to the nearest interfering prototype: nVAR $= \mathbb{E}_{k\in\mathcal{C}}[\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}}[||\mathbf{c}_k - f(\mathbf{x};\theta)||^2/||\mathbf{c}_k - \mathbf{n}_k||^2]]$, where $\mathcal{C}$ is the set of classes, i.e, 100 for the CIFAR-100 case, $\mathcal{D}$ is the union of test samples across clients, k is the class index, $\mathbf{c}_k$ is the prototype of class k, $f(\cdot;\theta)$ is the feature extractor of the global representation for each algorithm, and $\mathbf{n}_k$ is the nearest interfering prototype of class k. A smaller nVAR value indicates that the feature distributions are well-clustered and separated, which means a strong generalization over clients. The results in Table 4 clearly show that MetaVers acquires a compact and separated representation compared to other shared methods with a global representation.

Qualitative analysis by t-SNE visualization: We visualize the learned representation of MetaVers by using t-SNE of [38] to confirm the large-margin representation learning of MetaVers. As shown in Figure 2, we compare (d) MetaVers with (a) Local, b) MetaVers with a fixed-margin triplet loss, and (c) MetaVers with the cross-entropy loss. In Figure 2, the small dots in different colors indicate samples in 10 different classes of CINIC-10, and large points represent the class centroids. We observe that

	CINIC-10		
Method	Client 50	Client 100	Client 500
Local	60.7	58.3	50.8
Only CE loss	72.8	72.5	71.7
Only Triplet loss	72.6	72.7	71.9
MetaVers (Ours)	**73.2**	**73.2**	**72.5**

Table 5. Performance on CINIC-10 for Client 50 to 500 cases

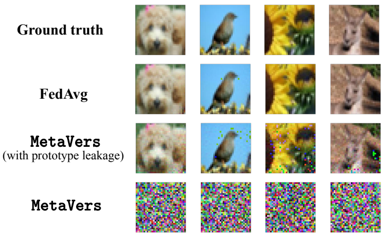


Figure 3. Single image reconstruction leakage after 300 iterations.

(d) MetaVers is the only case where the centroids are sufficiently well-separated without any overlapping. Notably, for cases (a), (b), and (c), some centroids are closely located to hinder the discrimination between classes. The result confirms that MetaVers dynamically enlarges the margins.

Ablation Studies on Loss Terms: Accuracies for the variants of loss adaptation for MetaVers are shown in Table 5. MetaVers with the dynamic margin shows the best performance over the versions of the fixed-margin ($\mathfrak{m} = 0.75$) triplet loss and cross-entropy (CE) loss. Also, we emphasize that all the versions of MetaVers show a moderate decline in accuracies as the number of clients increases.

Robustness to Gradient-based Attack A recent approach called Deep Leakage from Gradients of [43] raises a crucial threat to the FL framework that aggregates the local gradients at the central server. DLG optimizes a dummy input to mimic shared local gradients, gradually approaching the original input sample, and repeatedly rehearses loss and gradient computations for data reconstruction. For MetaVers, however, a server cannot access the local class prototypes which are essential for rehearsing the loss computation. Therefore, MetaVers can be robust to leakage from gradients. To prove the concept, we actually carry out the single image reconstruction experiments for FedAvg [26] and MetaVers. For further analysis, we tested a variant of MetaVers that shares the local prototypes with the server (denoted as MetaVers with prototype leakage). In the experiments on CIFAR-10 and CIFAR-100, we perform 300 iterations for estimating the original image via the L-BFGS optimizers [24] with a learning rate of 1. As shown in Figure 3, the attack on FedAvg and MetaVers with prototype leakage appears to be successful. Surprisingly, MetaVers is shown to prevent the attacker from reconstructing the images without any portion of data leakage. MetaVers does not share the prototypes so the estimation of local prototypes is essential for DLG to rehearse the loss computation, and it is shown to be a very challenging task for the attacker due to the large dimension of prototypes. Consequently, we confirm that MetaVers can train an effective representation without taking the risk of data leakage.

Method	CIFAR-100	*mini*ImageNet
FedAvg [26]	51.6	38.8
Fine-tuning via FedAvg [29]	63.2	61.6
Few-Round Learning [29]	72.9	69.3
MetaVers (Ours)	**73.6**	**71.2**

Table 6. Generalization for novel clients with novel classes

4.2. Generalization on Novel Classes

We evaluate MetaVers on the other protocol for unseen class inference from Few-Round Learning (FRL) of [29]. Following the exact setting of FRL, we partition CIFAR-100 and *mini*ImageNet into 64 train, 16 validation, and 20 test classes. Training is done for the train split with the non-IID setup, then the model is tested on a newcomer client with five unseen classes from the test split. We describe the exact settings in the Supplementary material. Although few-round learning (FRL) requires a few-round of additional communications for adapting the model to the novel client, MetaVers is tested without any further optimization of model parameters. In Table 6, we can observe the strong generalization capability MetaVers on even novel clients.

Additional studies for the behavior of dynamic margin values are presented in the Supplementary material.

5. Conclusions

We propose a meta-learning-based personalized federated learning (PFL) called MetaVers for versatile and large-margin representations. We adopt meta-learning for the PFL setting that aggregates the gradients from varying local episodes. In addition, a particular dynamic margin learning promotes better-clustered representations. MetaVers outperforms other competing methods in PFL benchmarks.

Acknowledgements

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01336, Artificial Intelligence Graduate School Program (UNIST)), (No. 2021-0-02201, Federated Learning for Privacy Preserving Video Caching Networks) (No. 2023-RS-2022-00156361, Innovative Human Resource Development for Local Intellectualization support program), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1C1C1012797).

References

[1] Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. Personalized federated learning with gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 8392–8406, 2021. 1, 2, 3, 6, 7

[2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. 1, 2, 6, 7

[3] Daoyuan Chen, Dawei Gao, Weirui Kuang, Yaliang Li, and Bolin Ding. pfl-bench: A comprehensive benchmark for personalized federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9344–9360, 2022. 6

[4] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018. 3

[5] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. In *IEEE Intelligent Systems*, volume 35, pages 83–93, 2020. 3

[6] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning (ICML)*, pages 2089–2099. PMLR, 2021. 1, 3, 6, 7

[7] Luke N. Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018. 6

[8] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. 2

[9] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21394–21405, 2020. 1, 6

[10] Canh T. Dinh, Tung T. Vu, Nguyen H. Tran, Minh N. Dao, and Hongyu Zhang. A new look and convergence rate of federated multi-task learning with laplacian regularization. *arXiv preprint arXiv:2102.07148*, 2021. 3

[11] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3557–3568, 2020. 3, 6, 7

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pages 1126–1135. PMLR, 2017. 3

[13] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[15] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1945–1954, 2018. 2, 4

[16] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 2

[17] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019. 3

[18] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*, volume 119, pages 5132–5143. PMLR, 2020. 2

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324. Ieee, 1998. 6

[20] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

[21] 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*, volume 2, pages 429–450, 2020. 2

[22] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[23] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B. Allen, Randy P. Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020. 1, 2, 6

[24] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. In *Mathematical Programming*, volume 45, pages 503–528. Springer, 1989. 8

[25] Othmane Marfoq, Giovanni Neglia, Laetita Kameni, and Richard Vida. Personalized federated learning through local memorization. In *International Conference on Machine Learning (ICML)*, pages 15070–15092. PMLR, 2022. 1, 2, 3, 6, 7

[26] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 4, 6, 7, 8

[27] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning (ICML)*, pages 4615–4625. PMLR, 2019. 2

[28] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for rederated image classification. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2, 3, 6, 7

[29] Younghyun Park, Dong-Jun Han, Do-Yeon Kim amd Jun Seo, and Jaekyun Moon. Few-round learning for federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 28612–28622, 2021. 8

[30] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. In *International Journal of Computer Vision*, volume 115, pages 211–252. Springer, 2015. 6

[32] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. In *IEEE Transactions on Neural Networks and Learning Systems*, volume 31, pages 3400–3413. IEEE, 2019. 2

[33] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *International Conference on Machine Learning (ICML)*, pages 1126–1135. PMLR, 2017. 4

[34] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning (ICML)*, pages 9489–9502. PMLR, 2021. 1, 2, 6, 7

[35] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, 2017. 3

[36] Jake Snell, Kevin Swerskey, and Rechard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2, 3

[37] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning over heterogeneous devices. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, pages 8432–8440, 2022. 1, 2, 3, 6, 7

[38] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. In *Journal of Machine Learning Research*, volume 9, 2008. 7

[39] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[40] Felix Yu, Ankit Singh Rawat, Aditya Menon, and Sanjiv Kumar. Federated learning with only positive labels. In *International Conference on Machine Learning (ICML)*, pages 10946–10956. PMLR, 2020. 2

[41] Mikhail Yurochkin, Zhiwei Fan, Aritra Guha, Paraschos Koutris, and XuanLong Nguyen. Scalable inference of topic evolution via models for latent geometric structures. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 2

[42] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 2

[43] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 2, 8