Federated Learning with Data-Agnostic Distribution Fusion

Jian-hui Duan, Wenzhong Li[*], Derun Zou, Ruichen Li, Sanglu Lu
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China
djhbarca@163.com, lwz@nju.edu.cn

Abstract

Federated learning has emerged as a promising distributed machine learning paradigm to preserve data privacy. One of the fundamental challenges of federated learning is that data samples across clients are usually not independent and identically distributed (non-IID), leading to slow convergence and severe performance drop of the aggregated global model. To facilitate model aggregation on non-IID data, it is desirable to infer the unknown global distributions without violating privacy protection policy. In this paper, we propose a novel data-agnostic distribution fusion based model aggregation method called `FedFusion` *to optimize federated learning with non-IID local datasets, based on which the heterogeneous clients' data distributions can be represented by a global distribution of several virtual fusion components with different parameters and weights. We develop a Variational AutoEncoder (VAE) method to learn the optimal parameters of the distribution fusion components based on limited statistical information extracted from the local models, and apply the derived distribution fusion model to optimize federated model aggregation with non-IID data. Extensive experiments based on various federated learning scenarios with real-world datasets show that* `FedFusion` *achieves significant performance improvement compared to the state-of-the-art.*

1. Introduction

Federated learning (FL) has emerged as a novel distributed machine learning paradigm that allows a global deep neural network (DNN) model to be trained by multiple participanting clients collaboratively. In such a paradigm, multiple clients train their local models based on datasets generated by edge devices such as sensors and smartphones, and the server is responsible to aggregate the parameters from the local models to form a global model without transferring local data to the central server. Nowadays federated learning has been drawn much attention in mobile-edge computing [21, 39] with its advantages in preserving data privacy [17, 49] and enhancing communication efficiency [30, 38, 43].

The de facto standard algorithm for federated learning is FedAvg [30], where parameters of local models are averaged element-wise with weights proportional to sizes of the client datasets. Based on FedAvg, a lot of algorithms have been proposed to improve the resource allocation fairness, communication efficiency, and convergence rate for federated learning [16, 29], which include LAG [3], Zeno [45], AFL [31], FedMA [43], etc.

One of the fundamental challenges of federated learning is the non-IID data sampling from heterogeneous clients. In real-world federated learning scenarios, local datasets are typically non-IID, and the local models trained on them are significantly different from each other. It was reported in [48] that the accuracy of a convolutional neural network (CNN) model trained by FedAvg reduces by up to 55% for a highly skewed non-IID dataset. The work in [43] showed that the accuracy of VGG model trained with FedAvg and its variants dropped from 61% to under 50% when the client number increases from 5 to 20 on heterogeneous data.

Several efforts have been made to address the non-IID challenges. FedProx [26] modified FedAvg by adding a dissimilarity bound on local datasets and a proximal term on the local model parameter to tackle heterogeneity. However, it poses restrictions on the local updates to be closer to the initial global model, which may lead to model bias. Zhao et al. [48] proposed a data sharing strategy to improve training on non-IID data by creating a small subset of data to share between all clients. However, data sharing could weaken the privacy requirement of federated learning. Several works [5, 28, 32] adopted data augmentation and model bias correction to deal with non-IID data. The clustered federated learning [2, 6, 7, 46] tackled non-IID settings by partitioning client models into clusters and performed model aggregation in cluster level.

[*] The corresponding author is Wenzhong Li (lwz@nju.edu.cn).

The personalized federated learning [18, 34, 35, 37] aimed to train personalized local models on non-IID data with the help of federated model aggregation. However, none of the existing works have considered to optimize federated model aggregation from the perspective of inferring the unknown global distribution based on the observed local model parameters, and yet the feasibility of global distribution inference subject to the data privacy policy of federated learning remains unexplored.

In this paper, we propose a novel data-agnostic distribution fusion method called `FedFusion` for federated learning on non-IID data. We introduce a distribution fusion model to describe the global data distribution as a fusion of several virtual distribution components, which is ideal for representing non-IID data generated from heterogeneous clients. However, applying a distribution fusion for federated learning is not a trivial work. Due to the data privacy policy of federated learning, the local datasets are inaccessible and their distributions are unknown to the server, so it is challenging for the server to derive the distribution parameters of a fusion model without observing to the real local data samples.

To tackle these issues, we propose an efficient method to optimize the distribution fusion federated learning with variational inference. Since the local data is inaccessible to the server, our method is based on the limited statistical information embedded in the normalization layers of the DNN models, i.e., the means and standard deviations of the feature maps (the outputs of intermediate layers). As shown in the proposed method, those information can be extracted from the local model parameters, which can be further used to infer a global distribution. Specifically, we develop a Variational AutoEncoder (VAE) method to learn the optimal parameters of distribution fusion components based on the observed information, and apply the derived parameters to optimize federated model aggregation with non-IID data. Extensive experiments based on a variety of federated learning scenarios with non-IID data show that `FedFusion` significantly outperforms the state-of-the-arts.

The contributions of our work are as follows.

- We propose a novel data-agnostic distribution fusion based model aggregation method called `FedFusion` to address the data heterogeneity problem in federated learning. It represents the global data by a fusion model of several virtual distribution components with different fusion weights, which is ideal to describe non-IID data generated from heterogeneous clients.

- We develop a VAE method to learn the optimal parameters for the data-agnostic distribution fusion model. Without violating the privacy principle of federated learning, the proposed method uses limited statistical information embedded in DNN models to infer a target global distribution with a maximum probability. Based on the inferred parameters, an optimal model aggregation strategy can be developed for federated learning under non-IID data.

- We conduct extensive experiments using five mainstream DNN models based on four real-world datasets under non-IID conditions. Compared to FedAvg and the state-of-the-art for non-IID data (FedProx, FedMA, IFCA, FedGroup, etc), the proposed `FedFusion` has better convergence and training efficiency, improving the global model's accuracy up to 12%.

2. Related Work

Federated learning [20] is an emerging distributed machine learning paradigm that aims to build a global model based on datasets distributing across multiple clients. One of the standard parameter aggregation methods is FedAvg [30], which combined local stochastic gradient descent (SGD) on each client with a server that performs parameter averaging. Later, the lazily aggregated gradient (LAG) method [3] allowed clients to run multiple epochs before model aggregation to reduce communication costs. The q-FedSGD [27] method improved FedAvg with a dynamic SGD update step using a scale factor to achieve fair resource allocation among heterogeneous clients. The FedMA [43] demonstrated that permutations of layers could affect the parameter aggregation results, and proposed a layer-wise parameter-permutation aggregation method to solve the problem. The FedDyn [1] method proposed a dynamic regularizer for each client at each round of aggregation, so that different models are aligned to alleviate the inconsistency between local loss and global loss.

Several works focused on optimizing federated learning under non-IID data. Zhao et al. used the earth mover's distance (EMD) to quantify data heterogeneity and proposed to use globally shared data for training to deal with non-IID [48]. The RNN-based method [14] adopted a meta-learning method to learn a new gradient from the received gradients and then applied it to update the global model. FedProx [26] modified FedAvg by adding a heterogeneity bound on local datasets and a proximal term on the local model parameter to tackle heterogeneity. FedBN [28] suggested keeping the local Batch Normalization parameters not synchronized with the global model to mitigate feature shifts in non-IID data. FedGN [10] replaced Batch Normalization with Group Normalization to avoids the accuracy loss induced by the skewed distribution of data labels. Yang et al. provided theoretical evidence on linear speedup for convergence of FedAvg under non-IID datasets with partial worker participation [47]. Duan et al. proposed a framework called

Astraea [5] to tackle local data distribution unbalance with data augmentation based on Z-score. The BVR-L-SGD [32] method used an additional local correction process to reduce the bias and variance of local gradient updates, and directly choose one local model as the global model rather than averaging them. The VHL method [40] allowed clients share an IID noisy dataset without any exact private data and used this virtual dataset to calibrate local training.

Personalized federated learning aims to train personalized local models on non-IID data with the help of federated model aggregation. The federated cluster learning [2] [46] [7] partitioned clients into clusters to address data heterogeneity, and aggregated different models for different clusters. For example, IFCA [7] alternately estimated the cluster identities of the clients and optimized the model parameters for the clusters via gradient descent. FedGroup [6] grouped clients based on similarities between their optimization directions to improve training efficiency. The works of [18] [35] [37] [34] further adopted multi-task learning and meta-learning to train personalized model for individual client. Different from clustered FL and personalized FL that form multiple personalized models, our work focuses on training a single global model from heterogeneous clients.

Despite the great efforts, there is lack of consideration of inferring the unknown global distribution based on limited observations. This paper proposes a novel data agnostic distribution fusion model with variational inference to optimize model aggregation in federated learning under non-IID conditions, which has not been explored in the past.

3. Problem Formulation

Federated learning methods involve multiple remote clients training local models based on their device-generated data and transferring local model's parameters to a central server periodically to form a global model. Typically the objective of conventional federated learning such as FedAvg [30] is to solve:

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \sum_{k=1}^{K} p_k \mathcal{L}_k(\mathbf{w}), \tag{1}$$

where $\mathcal{L}(\mathbf{w})$ is the global objective; K is the number of clients; $\mathcal{L}_k(\mathbf{w})$ is the local objective learned with local data; $p_k \geq 0$ and $\sum_k p_k = 1$ is the aggregation weight; and $\mathbf{w}$ is the model parameters to be learned. Generally, the aggregation weight is set to $p_k = \frac{n_k}{n}$, where n_k is the number of local samples in client k and $n = \sum_k n_k$ is the total number of samples.

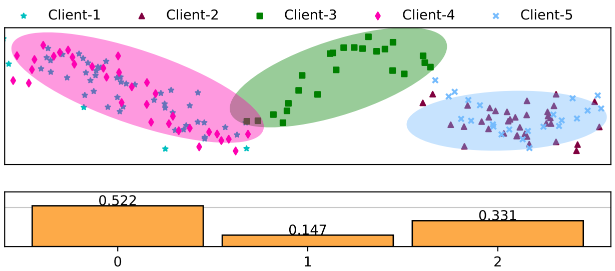


Figure 1. Illustration of Non-IID data from five clients being represented by a distribution fusion model with three virtual components.

In the above equation, the global model is aggregated by the weighted average of the local models proportional to the fixed sample size of the local dataset. If the training samples are IID distributed among the clients, the above aggradation provides an unbias estimation of the global model. However, if the training samples are Non-IID (which are more common in reality due to the heterogeneity of devices and users), the above fixed weighted averaging results in slow convergence and accuracy drop [10, 48].

To address the above issue, we introduce a distribution fusion federated learning model to optimize model aggregation with dynamic weights. We assume the local data distribution is unknown, and the global data can be described by a distribution fusion model with a mixture of several virtual components belonging to the same parametric family of distributions. As an example, Fig. 1 illustrates that the Non-IID data from five clients can be represented by a distribution fusion model with three virtual components with different mixture weights.

Note that the proposed data fusion model looks similar to a Gaussian Mixture Model (GMM). The major difference lies in that GMM assumes both global and local data are Gaussian distribution, while the proposed data fusion model allows any distribution on local data, which is more general and practical in federated learning scenarios.

We use $\tilde{\mathcal{D}}$ to denote the target global data distribution, which have M $(1 \leq M \leq K)$ virtual components: $\tilde{\mathcal{D}} = \sum_{m=1}^{M} \pi_m \bar{\mathcal{D}}_m$, where $\bar{\mathcal{D}}_m$ $(m = 1, \cdots, M)$ is the mth virtual distribution component and $\pi_m \geq 0$, $\sum_{m=1}^{M} \pi_m = 1$ are the fusion weights. Note that each client's local data can be allocated into some of the M components. We further introduce the following notations to describe the model.

- $\mathbf{c}_k \in \{0,1\}^M$ is a zero-one vector representing distribution allocation, where $c_{km} = 1$ represents that the distribution of the kth client is allocated to the mth virtual component.
- $\mathbf{b}_k = \{b_{km} \in [0,1] | m = 1, \cdots, M\}$, s.t., $\sum_{m=1}^{M} b_{km} c_{km} = 1$, represents the sampling ratio of the kth client which are hyperparameters in the proposed distribution fusion model. Noted that the sampling ratio $b_{km} \neq 0$ only when the corresponding allocation component $c_{km} = 1$. Specifically, $\mathbf{b}_k$ is used to sample a proportion of data from the allocated components to reconstruct the original data distribution for model optimization.

In brief, the above $\mathbf{c}_k$ represents an allocation policy and $\mathbf{b}_k$ represents a sampling policy, and they are parameters to be optimized for the distribution fusion model.

With the above notations, federated learning with distribution fusion can be described as the following objective:

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \sum_{m=1}^{M} \pi_m \sum_{k=1}^{K} b_{km} \mathcal{L}_k(\mathbf{w}). \tag{2}$$

Note that in an extreme condition where the data are IID among all clients, the number of virtual components $M = 1$ and the objective in Eq. (2) equals to simple averaging, which makes the conventional FedAvg [30] (i.e., Eq. 1) a special case of the proposed model.

However, due to the privacy policy of federated learning, the local datasets are inaccessible and their distributions are unknown to the server, so it is impossible for the server to derive the distribution parameters π_m, $\mathbf{c}_k$, $\mathbf{b}_k$ by observing the real data. Next, we propose a variational inference method to approximate the optimal parameters for the distribution fusion model.

4. Variational Inference for Data-Agnostic Distribution Fusion

Due to privacy protection, the local data distributions are unknown to the server, making derivation of target distribution $\tilde{\mathcal{D}}$ difficult. We argue tha although the private data is unknown, there are some statistical information embedded in the received model parameters which can be used by the server to infer the local distributions. For example, in a DNN model, the statistical information can be extracted from the *normalization layers* such as batch normalization [12], layer normalization, instance normalization, and group normalization, which typically contain the following statistical variables:

- $\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k$: the means and standard deviations of the feature maps (the outputs of intermediate layers) of the kth client's DNN model.

- $\hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\gamma}}_k$: the *shifted means* and *scaled standard deviations* [12] of the feature maps of the kth client's DNN model.

Note that the above parameters are either "pooling" from the batch channel or learned from the data samples [12], which contain statistical information such as means and deviations of feature maps that implicitly correlate to the dataset's distribution. We use $\mathbf{d}_k = \{\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\gamma}}_k\}$ to denote the observed statistical variables of the kth client. Since the real distribution is unknown, we can approximate the objective in Eq. (2) by optimizing the allocation policy $\mathbf{c}_k$ and sampling policy $\mathbf{b}_k$ for the distribution fusion model given the received models' statistical information. Hence we convert a dataset-dependent optimization problem into a data-agnostic problem based on the observable statistical information on the server.

4.1. Variational AutoEncoder

We propose a Variational AutoEncoder (VAE) method to derive the optimal parameters π_m, $\mathbf{c}_k$ and $\mathbf{b}_k$ of the fusion model. The plate notions of the VAE are shown in Fig. 2.

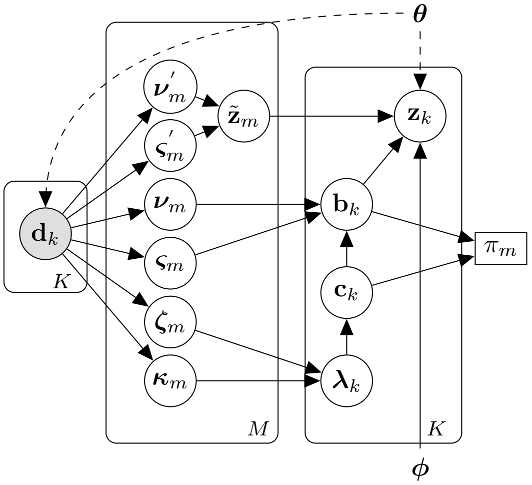


Figure 2. The variational Bayesian autoencoder using plate notations, where $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are global variables representing the encoder's parameters and the decoder's parameters respectively.

Using the stick-breaking construction of the Indian Buffet Process (IBP) [41], we infer that $\mathbf{c}_k$ is sampled from a Bernoulli distribution which is parameterized by $\boldsymbol{\lambda}_k = \{\lambda_{km} | m = 1, \cdots, M\}$, where $\boldsymbol{\lambda}_k$ is sampled in i.i.d. from a Beta distribution $Beta(\boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m)$ parameterized by $\boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m$. Similarly, we infer that $\mathbf{b}_k$ is sampled from a Gaussian prior distribution $\mathcal{N}(\boldsymbol{\nu}_m, \boldsymbol{\varsigma}_m)$ which is parameterized by $\boldsymbol{\nu}_m$ and $\boldsymbol{\varsigma}_m$.

We denote $\mathbf{z}_k = \sum_{m=1}^{M} b_{km} \cdot \tilde{\mathbf{z}}_m$, which is a latent variable used by a variational decoder $\boldsymbol{\theta}$ to reconstruct the observed $\mathbf{d}_k$. In the expression, $\tilde{\mathbf{z}}_k$ means the latent vector sampled from every allocated distribution of kth client from the Gaussian prior distribution $\mathcal{N}(\boldsymbol{\nu}'_m, \boldsymbol{\varsigma}'_m)$, which can adaptively adjust the latent posterior to a suitable probabilistic distribution as discussed in [19].

As illustrated in Fig. 2, the parameters of $Beta(\boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m)$, $\mathcal{N}(\boldsymbol{\nu}_m, \boldsymbol{\varsigma}_m)$ and $\mathcal{N}(\boldsymbol{\nu}'_m, \boldsymbol{\varsigma}'_m)$ can be inferred with an variational encoder $\boldsymbol{\phi}$ based on the observable information $\mathbf{d}_k$, i.e., $\{\boldsymbol{\nu}_m, \boldsymbol{\varsigma}_m, \boldsymbol{\nu}'_m, \boldsymbol{\varsigma}'_m, \boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m\} = \boldsymbol{\phi}(\mathbf{d}_k)$. In the meanwhile, the variables of $\mathbf{b}_k$ and $\tilde{\mathbf{z}}_m$ are used to compute a latent variable $\mathbf{z}_k$, which is further fed to a decoder $\boldsymbol{\theta}$ to reconstruct the observed data $\mathbf{d}_k$ with nonlinear transformation. By optimizing the parameters of the encoder-decoder, the optimal allocation policy $\mathbf{c}_k$ and the sampling policy $\mathbf{b}_k$ can be derived, which can be further used to derived the fusion weights π_m.

The details of the encoder-decoder process are explained as follows.

Encoder $\boldsymbol{\phi}$: As shown in Fig. 2, in order to infer the latent vector $\mathbf{z}_k$, we should derive the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\lambda}_k, \mathbf{c}_k, \mathbf{b}_k)$. We employ a multi-head nonlinear model to infer the approximation of true posterior $p(\boldsymbol{\lambda}_k, \mathbf{c}_k, \mathbf{b}_k | \mathbf{d}_k)$ with variational posteriors, and apply the stochastic gradient variational Bayes (SGVB) [19] algorithm to learn the model.

From Fig. 2 we know that variables in variational posterior are conditionally independent with the priori $p(\mathbf{d}_k)$. So we can decouple the variables as: $q_{\boldsymbol{\phi}}(\boldsymbol{\lambda}, \mathbf{c}, \mathbf{b}) = \prod_{k=1}^{K} \prod_{m=1}^{M} q_{\boldsymbol{\phi}}(b_{km}) \cdot q_{\boldsymbol{\phi}}(c_{km}|\lambda_{km}) \cdot q_{\boldsymbol{\phi}}(\lambda_{km})$, where the variational posterior distributions can be derived as [33]:

$$\begin{aligned} \mathbf{b}_k &\sim \mathcal{N}(\boldsymbol{\nu}_m, \varsigma_m), \\ \boldsymbol{\lambda}_k &\overset{i.i.d.}{\sim} Beta(\boldsymbol{\zeta}_m, \boldsymbol{\kappa}_m), \\ \mathbf{c}_k &\sim Bernoulli(\prod_{m=1}^{M} \lambda_{km}). \end{aligned} \tag{3}$$

Decoder $\boldsymbol{\theta}$: The decoder $\boldsymbol{\theta}$ takes the latent variable $\mathbf{z}_k$ as input to reconstruct the original observed data. According to Fig. 2, the derivation of $\mathbf{z}_k$ relies on three variables $\mathbf{b}_k$, $\boldsymbol{\lambda}_k$, and $\mathbf{c}_k$, whose variational posteriors are Gaussian, Beta, and Bernoulli distribution respectively, as shown in Eq. (3). We infer the three latent variables as follows.

Since the posterior of $\mathbf{b}_k$ is a Gaussian distribution with differentiable Monte Carlo expectations, it can be easily inferred with the Stochastic Gradient Variational Bayes (SGVB) estimator as [19].

The posterior of $\boldsymbol{\lambda}_k$ is a Beta distribution, which is hard to be inferred with conventional variational inference methods. Following the works of [23, 33], we approximate the posterior Beta with the Kumaraswamy distribution, a two-parameter continuous distribution also on the unit interval with a density function defined as:

$$Kumaraswamy(x; \boldsymbol{\zeta}_k, \boldsymbol{\kappa}_k) = \boldsymbol{\zeta}_k \boldsymbol{\kappa}_k x^{\boldsymbol{\zeta}_k - 1}(1 - x^{\boldsymbol{\zeta}_k})^{\boldsymbol{\kappa}_k - 1}, \tag{4}$$

where $\boldsymbol{\zeta}_k$ and $\boldsymbol{\kappa}_k$ are parameters of the distribution. It was proved that the Kumaraswamy approaches to the Beta albeit with high entropy, and it satisfies the differentiable and non-centered parameterization (DNCP) property with its closed-form inverse CDF [33]. Therefore the samples of $\boldsymbol{\lambda}_k$ can be drawn via the inverse transform of Kumaraswamy, which is expressed by

$$\boldsymbol{\lambda}_k \sim (1 - \xi^{\frac{1}{\boldsymbol{\kappa}_k}})^{\frac{1}{\zeta_k}}, \text{ where } \xi \sim Uniform(0, 1). \tag{5}$$

For the zero-one allocation vector $\mathbf{c}_k$, we reparameterize it with the Beta distribution as in Eq. (3). Using the Gumbel-Max trick [13] to draw samples from a Bernoulli distribution with binary probabilities, we have:

$$\mathbf{c}_{km} = arg \max_i (g_i + \log \prod_{i=1}^{2} \boldsymbol{\lambda}_{ki}), \tag{6}$$

where g_i is an IID sample drawn from $Gumbel(0, 1)$. After deriving $\mathbf{b}_k$ and $\mathbf{c}_k$ and sampling latent vector $\tilde{\mathbf{z}}_k$ from every component where client k is allocated, we can compute the latent variable $\mathbf{z}_k$ by $\mathbf{z}_k = \sum_{m=1}^{M} b_{km} \cdot \tilde{\mathbf{z}}_m$. Then we use $\mathbf{z}_k$ to reconstruct the original observed data $\mathbf{d}_k$ with $p_{\boldsymbol{\theta}}(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\gamma}_k | \mathbf{z}_k)$. In our implementation. the decoder $\boldsymbol{\theta}$ is parameterized by using a deep neural network to learn the model.

To derive the component weight π_m, we use a variant of the EM algorithm [4] with a softmax function:

$$\pi_m = \frac{\exp(\frac{1}{K} \sum_{k=1}^{K} q_{\boldsymbol{\phi}}(c_{km}) \cdot b_{km})}{\sum_{m=1}^{M} \exp(\frac{1}{K} \sum_{k=1}^{K} q_{\boldsymbol{\phi}}(c_{km}) \cdot b_{km})}. \tag{7}$$

4.2. Optimizing the Variational AutoEncoder

In this section, we introduce the algorithm to optimize the variational autoencoder based on the derivation in the above section. For convenient, we omit the latent variables $\{\mathbf{b}_k, \mathbf{c}_k, \boldsymbol{\lambda}_k\}$ and their priors in representing the encoder model $\boldsymbol{\phi}$.

The dashed lines in Fig. 2 denote the generative model $p_{\boldsymbol{\theta}}(\mathbf{z}_k) p_{\boldsymbol{\theta}}(\mathbf{d}_k|\mathbf{z}_k)$, and the solid lines denote the variational approximation $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)$ to the intractable posterior $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{d}_k)$. We approximate $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{d}_k)$ with $q_{\phi}(\mathbf{z}_k|\mathbf{d}_k)$ by minimizing their KL-divergence [15]:

$$\boldsymbol{\phi}^*, \boldsymbol{\theta}^* = arg \min_{\boldsymbol{\theta}, \boldsymbol{\phi}} \mathbb{D}_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k) \,||\, p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{d}_k)). \tag{8}$$

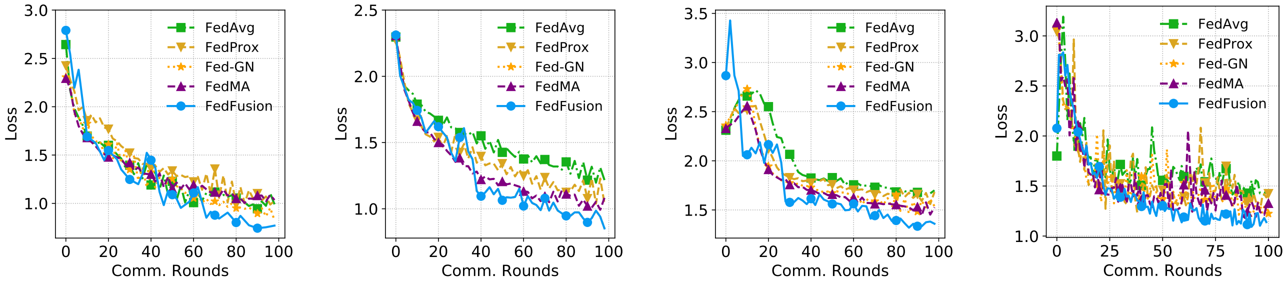
To derive the optimal value of the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, we compute the marginal likelihood of $\mathbf{d}_k$:

$$\begin{aligned} \log p(\mathbf{d}_k) = {} & \mathbb{D}_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k) \,||\, p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{d}_k)) \\ & + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{z}_k, \mathbf{d}_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)} \right], \end{aligned} \tag{9}$$

where the first term is the KL-divergence of the approximate distribution and the posterior distribution, and the second term is called the ELBO (Evidence Lower BOund) on the marginal likelihood of the k-th client's dataset.

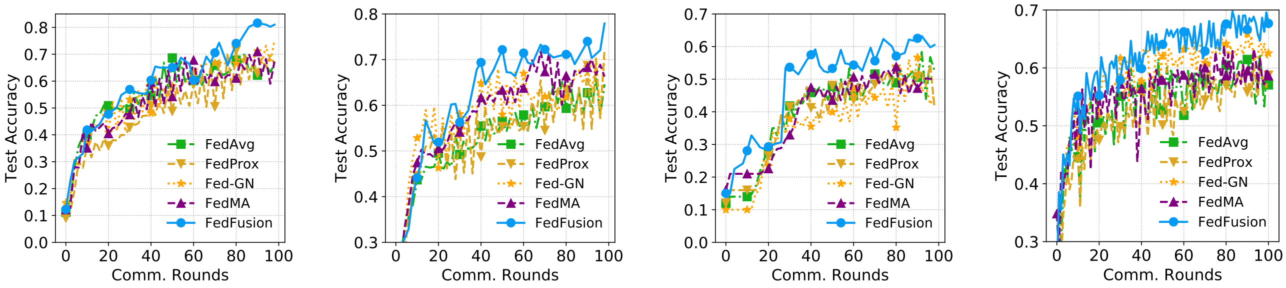
Since $\log p(\mathbf{d}_k)$ is non-negative, the minimization problem of Eq. (8) can be converted to maximizing the corresponding ELBO. To solve the problem, we change its form as:

$$\begin{aligned} & \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{z}_k, \mathbf{d}_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)} \right] = \\ & \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)} \left[log \frac{p(\mathbf{z}_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)} \right]}_{\text{Encoder}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{d}_k)} [\log p_{\boldsymbol{\theta}}(\mathbf{d}_k|\mathbf{z}_k)]}_{\text{Decoder}}. \end{aligned} \tag{10}$$



(a) ResNet18 on CIFAR10 (b) DenseNet121 on CIFAR10 (c) MobileNetV2 on CIFAR10 (d) BiLSTM on Sent140

Figure 3. Training loss of different algorithms.



(a) ResNet18 on CIFAR-10 (b) DenseNet121 on CIFAR-10 (c) MobileNetV2 on CIFAR-10 (d) BiLSTM on Sent140

Figure 4. Training efficiency of different algorithms.

The above form is a variational encoder-decoder structure: the model $q_{\phi}(\mathbf{z}_k|\mathbf{d}_k)$ can be viewed as a probabilistic encoder that given an observed statistics $\mathbf{d}_k$ it produces a distribution over the possible values of the latent variable $\mathbf{z}_k$. The model $p_{\boldsymbol{\theta}}(\mathbf{s}_k|\mathbf{z}_k)$ can be refered to as a probabilistic decoder that reconstructs the value of $\mathbf{d}_k$ based on the latent variable $\mathbf{z}_k$. According to the theory of variational inference [19], the problem in Eq. (10) can be solved with the SGD method using a nonlinear deep neural network (DNN) to optimize the mean squared error loss function. The overall FedFusion algorithm is illustrated in Algorithm 1, and its convergence is provided in the following theorem (proof in the supplementary).

Theorem 1 *(Convergence Bound): With learning epoch T, local epoch E, diameter of domain Γ, and learning rate η, the following convergence bound holds for* FedFusion*:*

$$\mathbb{E}[f(\mathbf{w}^T)] - f(\mathbf{w}^*) \leq \frac{L}{E+T}\left(\frac{A}{\tau} + \frac{E+1}{2}\Gamma^2\right). \quad (11)$$

Algorithm 1: The FedFusion algorithm.

```
Initialize $\mathbf{w}^0$.
for each communication round $t = 0, 1, \ldots, T-1$ do
    $\mathbf{w}_k^{t+1}$ := the model received from client k
    $\mathbf{d_k} := (\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\gamma}_k)$  // extracted from $\mathbf{w}_k^{t+1}$
    repeat
        Inference $\boldsymbol{\kappa}_m, \boldsymbol{\zeta}_,, \boldsymbol{\nu}_m, \varsigma_m, \boldsymbol{\nu}_m^{'}$ and $\varsigma_m^{'}$ based on encoder $\boldsymbol{\phi}$
        $\mathbf{b}_k, \boldsymbol{\lambda}_k, \mathbf{c}_k$ := sampling from distributions with Eq. 3, 5, 6
        $\tilde{\mathbf{z}}_m$ := sampling from $\mathcal{N}(\boldsymbol{\nu}_m^{'}, \varsigma_m^{'})$
        $\mathbf{z}_k := \sum_{m=1}^{M} b_{km} \cdot \tilde{\mathbf{z}}_m$
        Recover $\mathbf{z_k}$ to $\mathbf{d_k}$ based on decoder $\boldsymbol{\theta}$ with Eq. 10
    until VAE converge;
    $\mathbf{w}^{t+1} := \sum_{m=1}^{M} \pi_m \sum_{k=1}^{K} b_{km} \cdot c_{km} \cdot \mathbf{w}_k^{t+1}$  // model aggregation
    broadcast $\mathbf{w}^{t+1}$ to all clients
```

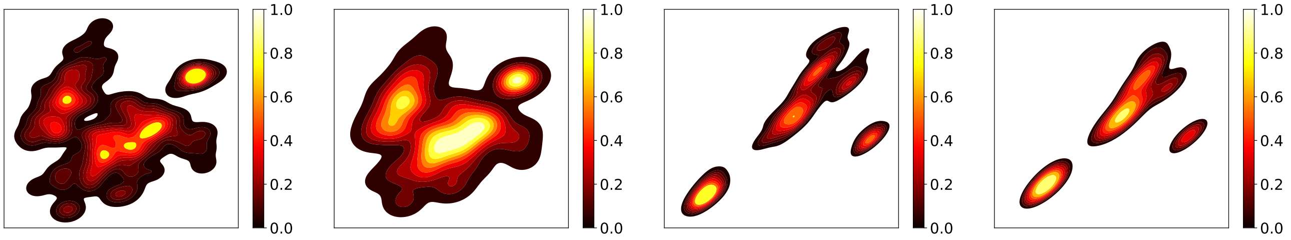
5. Experiments

5.1. Experimental Setup

Implementation: We implement the proposed FedFusion[1] algorithm and the considered baselines in PyTorch. We train the models in a simulated federated learning environment consisting of one server and multiple participating clients. Unless explicitly specified, the default number of clients is 50, and the learning rate $\beta = 0.01$. We conduct experiments on a GPU-equipped personal computer (CPU: Intel Core i7-8700 3.2GHz, GPU: Nvidia GeForce RTX 2070, Memory: 32GB DDR4 2666MHz, and OS: 64-bit Ubuntu 16.04).

Models and datasets: Our experiments are based on 5 mainstream deep learning models: ResNet18 [9], LeNet [24], DenseNet121 [11], MobileNetV2 [36], and BiLSTM.

[1] https://github.com/LiruichenSpace/FedFusion



(a) (b) (c) (d)

Figure 5. Visualization of data distribution. (a) the original distribution of MNIST, (b) the inferred distribution of MNIST with `FedFusion`, (c) the original distribution of CIFAR-10, (d) the inferred distribution of CIFAR-10 with `FedFusion`.

	Dataset	CIFAR-10			FMNIST	MNIST	Sent140
	Model	ResNet18	DenseNet121	MobileNetV2	LeNet	LeNet	BiLSTM
Single-model	FedAvg	68.78 (±0.89)	63.33 (±0.67)	54.69 (±3.92)	79.20 (±1.15)	97.32 (±0.04)	58.33 (±2.03)
	FedProx	70.18 (±0.45)	66.85 (±0.93)	55.03 (±2.77)	80.03 (±0.98)	97.55 (±0.02)	59.73 (±1.38)
	Fed-GN	72.57 (±0.78)	70.02 (±1.36)	56.43 (±1.92)	81.11 (±0.74)	97.88 (±0.02)	63.41 (±1.94)
	FedMA	73.43 (±1.03)	70.13 (±1.71)	59.61 (±2.01)	81.02 (±1.35)	98.06 (±0.03)	60.86 (±2.42)
Multi-model	FeSEM	67.78 (±2.58)	62.65 (±0.82)	53.82 (±3.69)	78.18 (±1.45)	96.24 (±0.17)	59.57 (±3.41)
	IFCA	73.04 (±1.45)	70.85 (±2.03)	58.93 (±2.45)	80.82 (±1.29)	97.09 (±0.11)	60.82 (±2.74)
	FedCluster	72.57 (±0.78)	68.77 (±1.38)	58.18 (±1.22)	79.11 (±0.74)	97.88 (±0.02)	63.41 (±1.94)
	FedGroup	74.38 (±1.92)	71.63 (±0.74)	59.86 (±2.09)	81.32 (±2.07)	97.37 (±0.61)	63.61 (±3.26)
	`FedFusion`	**81.26** (±0.82)	**75.92** (±1.25)	**62.88** (±1.21)	**83.16** (±0.74)	**98.49** (±0.04)	**67.51** (±1.71)

Table 1. Comparison of average test accuracy on non-IID datasets.

We use 4 real world datasets: MNIST [25], Fashion-MNIST [44], CIFAR-10 [22], and Sentiment140 [8], which are widely used for evaluating FL algorithms in the literature. MNIST is a dataset for hand written digits classification with 60000 samples of 28×28 greyscale image. Fashion-MNIST is an extended version of MNIST for benchmarking machine learning algorithms. CIFAR-10 is a large image dataset with 10 categories, each of which has 6000 samples of size 32×32. Sentiment140 is a natural language process dataset containing 1,600,000 extracted tweets annotated in scale 0 to 4 for sentiment detection.

We generate non-IID data partition according to the work [30]. For each dataset, we use 80% as training dada to form non-IID local datasets as follows. We sort the data by their labels and divide each class into 200 shards. Each client draw samples from the shards to form a local dataset with probability $pr(x) = \begin{cases} \eta \in [0,1], & \text{if x} \in class_j, \\ \mathcal{N}(0.5,1), & \text{otherwise.} \end{cases}$ It means that the client draws samples from a particular class j with a fixed probability η, and from other classes based on a Gaussian distribution. The larger η is, the more likely the samples concentrate on a particular class, and the more heterogeneous the datasets are. By default we set $\eta = 0.5$.

5.2. Performance Comparison

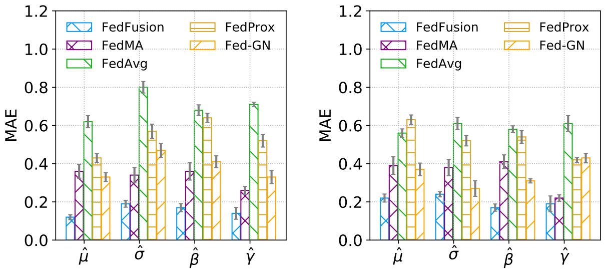
We compare the performance of `FedFusion` with 4 state-of-the-art methods: FedAvg [30], FedProx [26], Fed-GN [10], and FedMA [43]. The results are analyzed as follows.

Convergence: In this experiment, we study the convergence of the compared algorithms by showing the total communication epochs versus train loss. Fig. 3 shows the convergence of different algorithms for different models on different datasets. It is shown that the loss of all algorithms tends to be stable after a number of communication rounds. Clearly, `FedFusion` has the lowest loss, and converges the fastest among all algorithms.

Training Efficiency: In this experiment, we study the test accuracy versus time during the training of a DNN model with federated learning. Fig. 4 shown the results of training different models on different datasets. It is shown that `FedFusion` trains much faster than the baseline algorithms, and it reaches higher accuracy in a shorter period.

Visualization of Data Distribution: To intuitively illustrate how well the proposed `FedFusion` can approximate the original data distribution, we visualize the results in Fig. 5. Firstly, we plot the original distribution by projecting the data samples of the full dataset from all clients to a 2D plane with t-SNE [42] as shown in Fig. 5(a) and Fig. 5(c). Then, we apply the proposed algorithm to infer the parameters of the distribution fusion model, based on which we generate the same number of synthetic data samples as the original dataset. The synthetic data are further projected to a 2D plane with t-SNE as shown in Fig. 5(b) and Fig. 5(d) for comparison. According to the figure, the inferred distribution looks very close to the original distribution, which implies that the federated server can well approximate the global distribution parameters without accessing to local data.

Bias of Model Parameters: To show the power of the proposed VAE method for parameter optimization, we calculate the mean absolute error (MAE) of the statistical parameters $(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\sigma}}_k, \hat{\boldsymbol{\beta}}_k, \hat{\boldsymbol{\gamma}}_k)$ compared to a centrally-trained model based on global dataset, and the results are illustrated in Fig. 6(a) and Fig. 6(b). It is shown that `FedFusion` has a much lower bias in the statistical parameters than that of the other algorithms, which means that `FedFusion` provides a better approximation to the global data distribution.



(a) ResNet18 on CIFAR-10

(b) BiLSTM on Sent140

Figure 6. Comparison of parameter bias.

Global Model Accuracy: In this experiment, we compare the global model accuracy of different federated parameter aggregation algorithms after training to converge. For thorough comparison, we include 4 clustered and personalized FL algorithms FeSEM [46], IFCA [7], FedCluster [2], and FedGroup [6] as additional baselines. Since clustered and personalized FL methods output multiple models, we show the average results of all their output models. We repeat each experiment for 20 rounds and show the average performance in Table 1. Comparing the global model accuracy of different federated learning methods, `FedFusion` significantly outperforms the other algorithms for all DNN models. It outperforms FedMA by 7.83%, 5.79%, and 3.27% for accuracy in ResNet18, DenseNet121, and MobileNetV2 respectively on CIFAR-10. It achieves 2.14% improvement in LeNet on F-MNIST; 0.37% improvement in LeNet on MNIST; and 6.65% improvement in BiLSTM on Sent140 accordingly. Compared to FedAvg, the performance improvement of `FedFusion` is significant, which achieves up to 12.59% higher in DenseNet121 on CIFAR-10. In comparison to clustered/personalized FL, `FedFusion` outperforms the state-of-the-art method FedGroup by 6.88%, 1.84%, 1.12%, and 3.90% on the 4 datasets. In summary, `FedFusion` achieves the highest accuracy among all compared algorithms, which shows the superiority of federated model aggregation with the inference of the proposed global model distribution fusion.

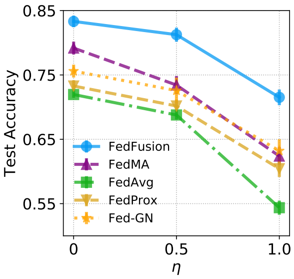


Figure 7. Test accuracy with different heterogeneity η (ResNet18 on CIFAR-10).

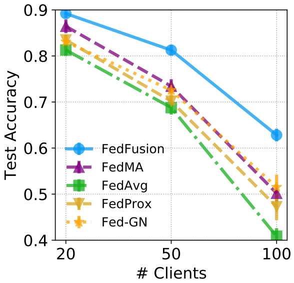


Figure 8. Test accuracy with different number of clients (ResNet18 on CIFAR-10).

Hyperparameter Analysis: We further analyze the influence of two hyperparameters: the heterogeneity of local datasets and the number of clients involved in federated learning.

The heterogeneity of local datasets is represented by η, the probability that a client tends to sample from a particular class. The more η approaches to 1, the more heterogeneous the local datasets are. Fig. 7 shows the test accuracy under different levels of heterogeneity. As η increases, the test accuracy of all models decreases. `FedFusion` yields the highest test accuracy and slowest performance drop among all compared algorithms, showing more robust against η, i.e., the degree of heterogeneity under non-IID data partition.

Fig. 8 compares the test accuracy of the global model for a different numbers of involved clients. When the number of clients increases from 20 to 100, the accuracy of `FedFusion` decreases much slower than that of the baselines, and it achieves the highest test accuracy among all compared federated learning algorithms in all cases.

6. Conclusion

This paper proposed a novel data-agnostic distribution fusion method called `FedFusion` to optimize federated learning with data heterogeneity. In the proposed method, the server aggregated the local models by allocating the clients' data distributions into several virtual distribution components with different fusion weights. The optimal parameters of the distribution fusion model were derived by a variational autoencoder (VAE) method. Extensive experiments showed that `FedFusion` significantly outperforms the state-of-the-art on a variety of scenarios.

Acknowledgments

This work was partially supported by the Natural Science Foundation of Jiangsu Province (Grant No. BK20222003), the National Natural Science Foundation of China (Grant Nos. 61972196, 61832008, 61832005), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Sino-German Institutes of Social Computing.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *ICLR*, 2021. 2

[2] Cheng Chen, Ziyi Chen, Yi Zhou, and Bhavya Kailkhura. Fedcluster: Boosting the convergence of federated learning via cluster-cycling. In *Big Data*, pages 5017–5026, 2020. 1, 3, 8

[3] Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. Lag: Lazily aggregated gradient for communication-efficient distributed learning. *NIPS*, pages 5050–5060, 2018. 1, 2

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977. 5

[5] Moming Duan, Duo Liu, Xianzhang Chen, Renping Liu, Yujuan Tan, and Liang Liang. Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems*, pages 59–71, 2021. 1, 3

[6] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Ternary cosine similarity-based clustered federated learning framework toward high accuracy in heterogeneous data. *CoRR*, abs/2010.06870, 2020. 1, 3, 8

[7] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *NeurIPS*, 2020. 1, 3, 8

[8] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision, 2009. 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016. 6

[10] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B. Gibbons. The Non-IID data quagmire of decentralized machine learning. *ICML*, 2020. 2, 3, 7

[11] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 6

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. *ICML*, 1:448–456, 2015. 4

[13] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 5

[14] Jinlong Ji, Xuhui Chen, Qianlong Wang, Lixing Yu, and Pan Li. Learning to learn gradient aggregation by gradient descent. In *IJCAI*, pages 2614–2620, 2019. 2

[15] James M. Joyce. Kullback-leibler divergence. *International Encyclopedia of Statistical Science*, 2011. 5

[16] P. Kairouz, H. McMahan, B. Avent, Aurélien Bellet, Mehdi Bennis, A. Bhagoji, Keith Bonawitz, and et al. Advances and open problems in federated learning. *ArXiv*, abs/1912.04977, 2019. 1

[17] Marcel Keller, Valerio Pastro, and Dragos Rotaru. Overdrive: Making SPDZ great again. In *EUROCRYPT*, volume 10822, pages 158–189, 2018. 1

[18] Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. In *NeurIPS*, volume 32, 2019. 2, 3

[19] Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 4, 5, 6

[20] Jakub Konečnỳ, H. Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *NIPS Optimization for Machine Learning Workshop*, 2015. 2

[21] Jakub Konecný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *ArXiv*, abs/1610.02527, 2016. 1

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 7

[23] Ponnambalam Kumaraswamy. A generalized probability density function for double-bounded random processes. *Journal of Hydrology,*, 1980. 5

[24] Yann Lecun, LÃ©on Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. 6

[25] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs*, 2, 2010. 7

[26] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, pages 429–450. 2020. 1, 2, 7

[27] Tian Li, Maziar Sanjabi, and Virginia Smith. Fair resource allocation in federated learning. In *ICLR*, 2020. 2

[28] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *ICLR*, 2021. 1, 2

[29] Wei Yang Lim, Nguyen Cong Luong, D. Hoang, Y. Jiao, Ying-Chang Liang, Qiang Yang, D. Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22:2031–2063, 2020. 1

[30] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. *AISTATS*, 54:1273–1282, 2017. 1, 2, 3, 4, 7

[31] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *ICML*, volume 97, pages 4615–4625, 2019. 1

[32] Tomoya Murata and Taiji Suzuki. Bias-variance reduced local sgd for less heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 7872–7881, 2021. 1, 3

[33] Eric T. Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. In *ICLR*, 2017. 5

[34] Yifan Niu and Weihong Deng. Federated learning for face recognition with gradient correction. In *AAAI 2022*, pages 1999–2007, 2022. 2, 3

[35] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *ICLR*, 2020. 2, 3

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 6

[37] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9489–9502, 2021. 2, 3

[38] V. Smith, S. Forte, C. Ma, M. Takac, M. I. Jordan, and M. Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18(230):1–47, 2018. 1

[39] Shizhao Sun, Wei Chen, Jiang Bian, Xiaoguang Liu, and Tie-Yan Liu. Ensemble-compression: A new method for parallel training of deep neural networks. In *ECML-KDD*, pages 187–202, 2017. 1

[40] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *Proceedings of the 39th International Conference on Machine Learning*, pages 21111–21132, 2022. 3

[41] Yee Whye Teh, Dilan Grür, and Zoubin Ghahramani. Stick-breaking construction for the indian buffet process. In *AISTATS*, volume 2, pages 556–563, San Juan, Puerto Rico, 21–24 Mar 2007. 4

[42] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 7

[43] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *ICLR*, 2020. 1, 2, 7

[44] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. 2017. 7

[45] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *ICML*, volume 97, pages 6893–6901, 2019. 1

[46] Ming Xie, Guodong Long, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning. *CoRR*, abs/2005.01026, 2020. 1, 3, 8

[47] Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-IID federated learning. In *ICLR*, 2021. 2

[48] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and V. Chandra. Federated learning with Non-IID data. *ArXiv*, abs/1806.00582, 2018. 1, 2, 3

[49] H. Zhu and Y. Jin. Multi-objective evolutionary federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4):1310–1322, 2020. 1