

## FedMHO: Heterogeneous One-Shot Federated Learning Towards Resource-Constrained Clients

Authors	Yao, Dezhong;Liu, Tongtong;Shi, Yuexin;Xu, Zhiqiang
Citation	D. Yao, T. Liu, Y. Shi, Z. Xu, "FedMHO: Heterogeneous One-Shot Federated Learning Towards Resource-Constrained Clients," 2026, pp. 5686-5697.
DOI	<a href="https://doi.org/10.1145/3774904.3792743">10.1145/3774904.3792743</a>
Publisher	Association for Computing Machinery
Rights	Licence for published version: Creative Commons Attribution 4.0 International
Download date	2026-06-15 05:30:51
Item License	<a href="http://creativecommons.org/licenses/by/4.0/">http://creativecommons.org/licenses/by/4.0/</a>
Link to Item	<a href="https://hdl.handle.net/20.500.14634/2331">https://hdl.handle.net/20.500.14634/2331</a>

# FedMHO: Heterogeneous One-Shot Federated Learning Towards Resource-Constrained Clients

Dezhong Yao\*  
dyao@hust.edu.cn

Huazhong University of Science and Technology  
Wuhan, China

Yuexin Shi  
shiyuexin@hust.edu.cn

Huazhong University of Science and Technology  
Wuhan, China

Tongtong Liu\*  
tliu@hust.edu.cn

Huazhong University of Science and Technology  
Wuhan, China

Zhiqiang Xu  
zhiqiang.xu@mbzuai.ac.ae

Mohamed bin Zayed University of Artificial Intelligence  
Abu Dhabi, UAE

## Abstract

Federated Learning (FL) is increasingly adopted in edge computing scenarios, where a large number of heterogeneous clients operate under constrained or sufficient resources. The iterative training process of FL incurs considerable computation and communication overhead, which is unfriendly for resource-constrained devices. One-shot FL is a promising approach to addressing communication issues inherent in conventional FL, and model-heterogeneous FL solves the problem of diverse computing resources across clients. However, existing methods face challenges in effectively managing model-heterogeneous one-shot FL, often leading to unsatisfactory global model performance or reliance on auxiliary datasets. To address these challenges, we propose a novel FL framework named FedMHO, which leverages deep classification models on resource-sufficient clients and lightweight generative models on resource-constrained devices. On the server side, FedMHO involves a two-stage process that includes data generation and knowledge fusion. Furthermore, we introduce FedMHO-MD and FedMHO-SD to mitigate the knowledge-forgetting problem during the knowledge fusion stage, and an unsupervised data optimization solution to improve the quality of synthetic samples. Comprehensive experiments demonstrate the effectiveness of our methods, as they outperform state-of-the-art baselines in various experimental setups.

## CCS Concepts

• **Computing methodologies** → **Distributed computing methodologies**; **Artificial intelligence**; **Distributed artificial intelligence**.

## Keywords

federated learning, model heterogeneity, one-shot learning, resource constraint, communication efficiency

## ACM Reference Format:

Dezhong Yao, Tongtong Liu, Yuexin Shi, and Zhiqiang Xu. 2026. FedMHO: Heterogeneous One-Shot Federated Learning Towards Resource-Constrained Clients. In *Proceedings of the ACM Web Conference 2026 (WWW '26)*, April 13–17, 2026, Dubai, United Arab Emirates. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/XXXXXX.XXXXXX>

## 1 Introduction

Federated Learning (FL) has emerged as a promising paradigm for training machine learning models across distributed devices without sharing raw data [32]. Despite impressive theoretical and experimental advancements, practical implementation of FL is confronted with challenges [9, 21, 51]. A considerable challenge is the necessity for multiple communication rounds among several clients and a central server, which can be both costly and intolerable due to associated time and energy constraints [44, 47]. Moreover, frequent communication poses a high risk of privacy attacks [31], such as a man-in-the-middle attack [40] or the potential for reconstructing training data from gradients [49]. To address issues associated with communication and security in conventional FL, the concept of one-shot FL has been introduced [11], which aims to obtain an acceptable global model within a single communication round.

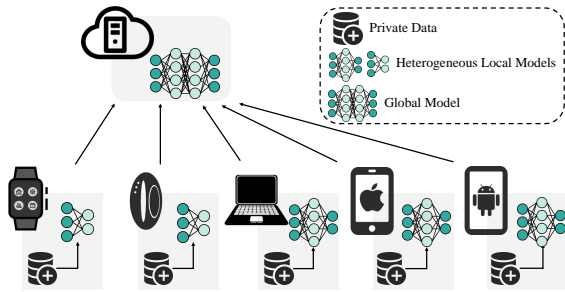
Another challenge in real-world FL scenarios, such as those in healthcare [1], recommendation systems [28], and financial services [54], is the heterogeneity of clients' computing resources [46]. For example, a naive solution to develop a health monitoring model via FL involves weighted averaging of the parameters of local models trained on each user's smartphone, smartwatch, or smart wristband. Smartphones generally possess 4GB-16GB of RAM, whereas the Xiaomi Smart Wristband 8 has 1.4MB of RAM, and the Amazfit Band 7 has 8MB of RAM. This heterogeneity in client computing resources is common in FL applications [42]. Existing FL frameworks typically require clients to deploy homogeneous local models. However, employing uniformly small models prevents clients with abundant computing resources from utilizing their computational potential [4]. Conversely, deploying uniformly large models excludes clients with limited computing resources from participating in the FL process, thereby preventing the global model from acquiring knowledge from these resource-constrained clients. Therefore, deploying homogeneous models can harm the performance of the final global model. A prevalent method for effectively addressing computing heterogeneity is deploying heterogeneous models on

\*Corresponding authors. D. Yao, T. Liu, and Y. Shi were with the School of Computer Science and Technology at Huazhong University of Science and Technology.



This work is licensed under a Creative Commons Attribution 4.0 International License. *WWW '26, Dubai, United Arab Emirates.*

© 2026 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-2307-0/2026/04  
<https://doi.org/10.1145/XXXXXX.XXXXXX>



**Figure 1: The heterogeneous one-shot Federated Learning (FL) framework. The clients with varying computing capabilities deploy heterogeneous local models. Each client communicates with the central server only once. After fully training their local models on private data, clients send these models to the server, where they are aggregated into a global model.**

different clients that align with their respective computing capabilities [48]. Therefore, for FL scenarios that include a mixture of resource-sufficient clients and resource-constrained clients, as illustrated in Figure 1, studying model-heterogeneous one-shot FL becomes critical.

The existing one-shot FL methods [15, 52] still face various challenges in model heterogeneity scenarios. DOSFL [56] adopts dataset distillation [43] on each client. However, sending distilled data to the central server introduces additional communication costs, which goes against the original intention of addressing the communication bottleneck in one-shot FL. Some other methods [11, 26] leverage knowledge distillation [16] to aggregate local models with auxiliary public data. However, the performance of these methods depends on the auxiliary data size and the domain similarity between auxiliary data and local data [38]. Several data-free methods [15, 52] have been proposed to address these issues. DENSE [52] and Co-Boosting [8] train an additional generator on the server side to generate auxiliary data that resembles the local data distribution. However, the efficacy of this generator heavily depends on the local model used for its training. If clients with limited computing resources deploy lightweight local models, and these local models exhibit restricted performance, the generator’s performance will suffer, thereby hindering the final global model from achieving high performance. FEDCVAE [15] deploys generative models on all participating clients, and the global model on the server side is trained from scratch using the synthetic samples generated by these generators. Nevertheless, using synthetic samples to convey local data information is not as effective as directly using local model parameters. A more detailed explanation is provided in Section 3.

In this paper, we propose a novel method, FedMHO<sup>1</sup>, to address the challenge of **Model-Heterogeneous One-shot Federated Learning**. Our method involves deploying deep classification models on resource-sufficient clients while utilizing lightweight generative models on resource-constrained clients. To validate the effectiveness of our method, we adopt Conditional Variational Autoencoders (CVAE) [36] as the generative models. More complex generators can be employed for more complex tasks. During global model training, FedMHO introduces a two-stage process encompassing data

generation and knowledge fusion. In the data generation stage, the decoders received from clients generate synthetic samples based on local label distribution. To improve the fidelity of the synthetic samples, we employ an unsupervised data optimization solution. In the subsequent knowledge fusion stage, the global model is initialized by the average parameters of the local classification models and then updated based on the synthetic samples generated in the data generation stage. Furthermore, during the global model training, we propose two solutions named FedMHO-MD and FedMHO-SD to prevent the forgetting of knowledge from classification models. In FedMHO-MD, the local classification models act as multiple teacher models to distill the global model. FedMHO-SD involves self-distillation, wherein the initialized global model acts as a teacher model to distill the global model.

Our main contributions are summarized as follows. (1) We propose FedMHO, a one-shot FL framework designed for heterogeneous clients. This framework deploys classification models on computing resource-sufficient clients and lightweight generative models on computing resource-constrained clients. (2) We propose an unsupervised data processing solution to optimize the quality of synthetic samples, consequently improving the performance of the final global model. (3) To address the knowledge-forgetting problem during the training of the global model, we propose two effective strategies: FedMHO-MD and FedMHO-SD. (4) Compared to the optimal baseline, the average accuracies of FedMHO, FedMHO-MD, and FedMHO-SD are improved by 5.17%, 8.35%, and 8.25%, respectively, demonstrating the effectiveness of the proposed methods.

## 2 Related Work

### 2.1 One-shot Federated Learning

One-shot FL aims to aggregate client information into a global model within a single communication round. Guha et al. [11] first proposed One-shot FL with two methods: bagging local models or ensemble distillation using public data. FedOV [15] applies open-set voting [33, 55] to mitigate label skew in bagging, though such strategies add inference overhead on clients. FedKT [26] uses knowledge distillation with auxiliary public data, while DOSFL [56] achieves data-free One-shot FL by having clients distill local datasets for server training. DENSE trains a server-side generator from local models to create synthetic data for distillation, and Co-Boosting [8] further enhances performance by optimizing both data and ensemble integration. However, these two-stage approaches cause information loss in both classifier and generator training. To reduce this, FedSD2C [53] directly synthesizes and shares samples from local data, though it is only feasible on large high-resolution datasets. Heinbaugh et al. [15] propose FEDCVAE-ENS and FEDCVAE-KD, which deploy client-side generative models to generate data for global training. Still, the above data-free methods overlook that one-shot FL typically involves model heterogeneity.

### 2.2 Model-Heterogeneous Federated Learning

To achieve model heterogeneity, some methods extract heterogeneous sub-models from the global model to use as local models. For example, Federated Dropout [3] randomly extracts sub-models using Dropout [37]. HeteroFL [10], FjORD [18], and FedDSE [41] extract static sub-models from the global model, while

<sup>1</sup>The source code is available at <https://github.com/YXShi2000/FedMHO>.

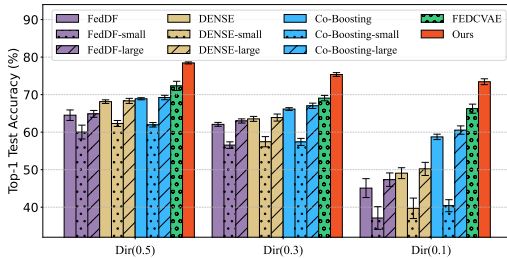


Figure 2: Top-1 test accuracy (%) on the EMNIST dataset.

FedRolex [2] and Split-Mix [17] extract dynamic sub-models. These partial training-based methods require each local model to be a sub-model of the global server model, preventing their deployment in FL scenarios involving completely different model structures. Knowledge distillation-based methods do not have the limitations of partial training-based methods and are more suitable for FL scenarios with diverse models. FedMD [25] uses transfer learning to pretrain local models on large-scale public datasets, then fine-tunes them on local datasets. FedGKT [13] uses Split Learning [12] with the global model as a downstream model of local models. FedDF [29], DS-FL [20], and Fed-ET [6] utilize unlabeled auxiliary data to transfer knowledge from local models to the global model.

### 3 Motivation

Current FL algorithms do not adequately address the challenge of one-shot FL with heterogeneous models. Model-heterogeneous FL algorithms can be broadly categorized into partial training-based methods and knowledge distillation-based methods. However, **partial training-based methods are essentially not applicable to one-shot scenarios, and knowledge distillation-based methods introduce erroneous information that misleads the global model.**

Partial training-based methods assign heterogeneous sub-models to clients, either by randomly dropping out parameters from the global model or extracting them based on specific rules. Since one-shot FL involves only a single round of communication, and the local model parameters are typically fractions (e.g.,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$ ) of the global model, most of the parameters of the global model are updated by the local data from only a subset of clients. This constraint leads to poor performance of the aggregated global model, and no existing work successfully combines partial training with one-shot FL.

Knowledge distillation-based methods employ the local models as multiple teachers and utilize an auxiliary dataset to train the global model via knowledge distillation, which proves effective in one-shot scenarios [34]. This auxiliary dataset can be sourced from a public dataset, such as FedDF, or by training an additional generator like DENSE or Co-Boosting. However, the lightweight local models of FedDF, DENSE or Co-Boosting often underperform and provide inaccurate logits (i.e., soft labels) during training of the global model or generator [5], which subsequently degrade the performance of the final global model. Figure 2 presents the experimental results on the EMNIST dataset. The experimental setup is the same as in Section 5. Here, ‘small’/‘large’ refers to the aggregation of either lightweight small local models or deep large local models, respectively. Notably, when only the deep large models are used, the global model performance slightly improves

compared to using all local models, even though data from the deployed lightweight small models is not aggregated. This indicates lightweight small models provide negative gain.

FEDCVAE deploys generative models on all clients to generate synthetic samples for training the global model. Figure 2 demonstrates that FEDCVAE has a significantly better performance compared to FedDF, DENSE and Co-Boosting. This improvement arises because, while lightweight generative models may produce low-quality samples, such as blurred contours (refer to Figure 9 in the Appendix), they do not introduce erroneous logits like poorly performing classification models do.

Nonetheless, synthetic samples cannot fully match the quality of the raw data. To maximize the use of real data and minimize erroneous information from low-performance models, we propose deploying deep classification models on clients with sufficient computing resources and lightweight generative models on clients with limited computing resources. During aggregation, the classification models directly average the parameters to initialize the global model, and then the global model is further trained by the synthesized samples. The effectiveness of our proposed method can also be demonstrated in Figure 9.

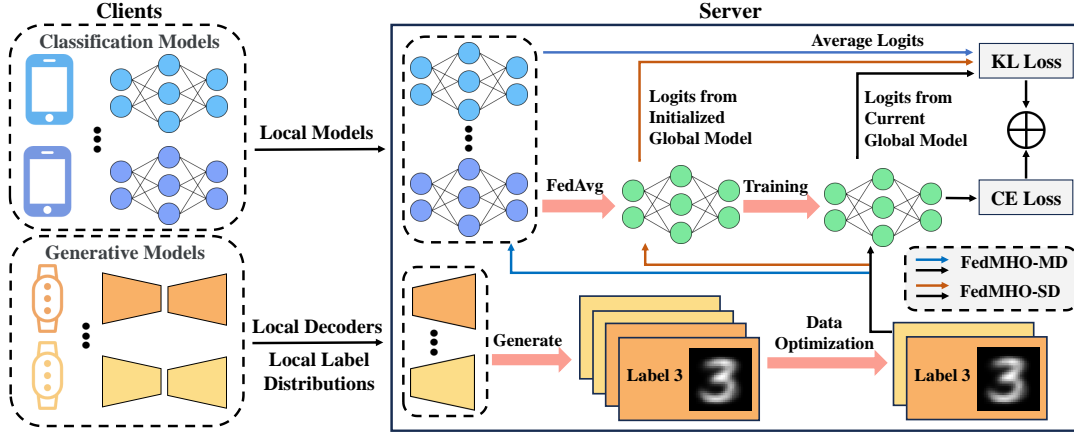
## 4 Methodology

### 4.1 Preliminaries

**4.1.1 Conditional Variational Autoencoder.** CVAE is a variation of Variational Autoencoder (VAE) [22] that incorporates conditional information into VAE. Specifically, a VAE learns a latent space representation of input data through an encoder-decoder architecture. In CVAE, this latent space is conditioned on additional information, such as class labels, enabling controlled generation based on specific conditions. Formally, let  $x$  denote the input data,  $c$  denote the conditional information,  $\phi$  denotes the encoder network, and  $\theta$  denotes the decoder network, respectively. A CVAE uses  $\phi$  to model the approximate posterior distribution  $q_\phi(z|x, c)$  over latent variables  $z$ , and uses  $\theta$  to model the generative distribution  $p_\theta(x|z, c)$  for reconstructing the input data.

**4.1.2 Knowledge Distillation.** Knowledge distillation aims to minimize the discrepancy between logits output from the teacher model and the student model on the same data. The discrepancy can be measured by the Kullback–Leibler divergence [23]  $\text{KL}[\cdot]$  as  $\min_{W_S} \text{KL}_{x \in D} [h(W_T; x) || h(W_S; x)]$ , where  $W_T$  denotes teacher model,  $W_S$  denotes student model,  $D$  denotes the data used for knowledge distillation, and  $h(*; x)$  denotes the output logits of input data  $x$ .

**4.1.3 Problem Definition.** We consider a one-shot FL setup with an  $N_c$ -class classification task, where  $K$  clients are connected to a central server. We define the set of clients as  $\{K\} = \{\{K_C\}, \{K_G\}\}$ , where  $\{K_C\}$  denotes the clients that develop classification models, and  $\{K_G\}$  denotes clients that develop generative models. Each client  $k \in \{K\}$  holds a local model  $w_k$  and local training data  $\mathcal{D}_k$ . Specifically, a computing resource-sufficient client  $k$  holds a local classification model  $\{w_k | k \in \{K_C\}\}$ , while a computing resource-constrained client  $k$  holds a lightweight generative model  $\{w_k | k \in \{K_G\}\}$  with an encoder  $w_k^\phi$  and a decoder  $w_k^\theta$ . We aim to join resource-constrained clients in FL training and train a well-performing global model  $w$  with resource-sufficient clients.



**Figure 3: An overview of FedMHO.** Resource-sufficient clients train deep classification models, and resource-constrained clients train lightweight generative models. The global model is initialized by averaging local classification models. Synthetic samples are generated to train the global model. To solve the knowledge-forgetting problem, FedMHO-MD employs classification models as teachers to distill the global model, while FedMHO-SD utilizes the initialized global model as a teacher to distill the global model.

#### Algorithm 1 The FedMHOs Algorithm

- 1: **Input:** Number of categories  $N_c$ , local model  $w_k$ , local training epoch  $E_k$ , global training epoch  $E_g$ , set of local classification model  $\{K_C\}$ , set of generative model decoder  $\{K_G\}$ .
- 2: **Output:** The final global model  $w$ .
- 3: /\* Client Side \*/
- 4: **for** each  $\mathcal{E} = 1 \cdots E_k$  **do**
- 5:   Train  $w_k$  on  $\mathcal{D}_k$  with (2) ( $k \in \{K_C\}$ ) or (3) ( $k \in \{K_G\}$ ).
- 6: **end for**
- 7: **return** updated  $w_k$  ( $k \in \{K_C\}$ ) or updated  $w_k^\theta$  ( $k \in \{K_G\}$ ).
- 8: /\* Server Side \*/
- 9: Initialize global model  $w_0$  with (4).
- 10: Initialize an empty set for generate samples  $\mathcal{D}_s$ .
- 11: **for**  $n_c = 1 \cdots N_c$  **do**
- 12:   Generate samples belonging to category  $n_c$  using decoder  $\{w_k^\theta \mid k \in \{K_G\}\}$ .
- 13:   Optimize synthetic samples via K-means and add to  $\mathcal{D}_s$ .
- 14: **end for**
- 15: **for**  $\mathcal{E} = 1 \cdots E_g$  **do**
- 16:   Train global model  $w$  on  $\mathcal{D}_s$  with (7).
- 17: **end for**
- 18: **return** final global model  $w$ .

## 4.2 Overall Algorithm

Figure 3 illustrates the overall framework of FedMHO. Specifically, resource-sufficient clients deploy deep classification models, and resource-constrained clients deploy lightweight CVAEs. Each client fully trains its respective local model during the local training phase. Afterwards, clients with classification models send their complete models to the central server, and clients utilizing CVAEs send the CVAE decoders and the local label distributions to the central server. The server-side training process consists of data generation and knowledge fusion. In the data generation stage, each local CVAE

decoder generates synthetic samples based on its local label distribution. These samples are then refined through an unsupervised quality enhancement process. In the knowledge fusion stage, the local classification models initialize the global model by aggregating their parameters. The global model is then updated with the synthetic samples. To mitigate knowledge-forgetting during the global model training, we employ FedMHO-MD and FedMHO-SD. The specific training details are provided below. The complete algorithmic representation of our methods is provided in Algorithm 1.

**4.2.1 Local Training.** The clients train their local models on their respective local data for  $E_k$  epochs. During each training epoch, each local model weight  $w_k$  is updated as

$$w_k := w_k - \eta \cdot \nabla \mathcal{L}_k(w_k; b_l), \quad (1)$$

where  $\eta$  denotes the learning rate,  $b_l$  denotes the mini-batch from local data  $\mathcal{D}_k$ ,  $\mathcal{L}_k$  denotes the training loss function of local model  $k$ , and  $\nabla \mathcal{L}_k(\cdot)$  denotes the partial derivative of  $\mathcal{L}_k(\cdot)$  with respect to its parameter  $w_k$ . For the classification models,  $\mathcal{L}_k(w_k; b_l)$  is defined as the cross-entropy loss

$$\mathcal{L}_k(w_k; b_l) = - \sum_{i=1}^{|b_l|} y_i \log(\hat{y}_i), \quad (2)$$

where  $|b_l|$  denotes the size of the mini-batch  $b_l$ ,  $y_i$  and  $\hat{y}_i$  represent the ground truth and predicted probability of sample  $i$ , respectively. For the CVAE models,  $\mathcal{L}_k(w_k; b_l)$  is the sum of the reconstruction loss  $\mathcal{L}_{\text{recon}}$  and the KL divergence loss  $\mathcal{L}_{\text{KL}}(w_k; b_l)$ , defined as

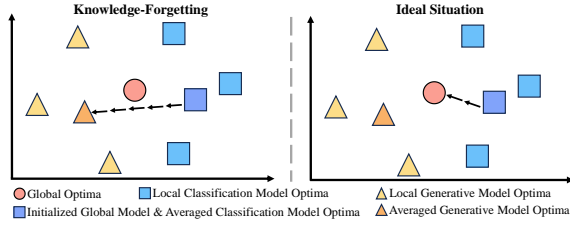
$$\mathcal{L}_k(w_k; b_l) = \mathcal{L}_{\text{recon}}(w_k; b_l) + \mathcal{L}_{\text{KL}}(w_k; b_l), \quad (3)$$

with

$$\mathcal{L}_{\text{recon}}(w_k; b_l) = \frac{1}{|b_l|} \sum_{i=1}^{|b_l|} \mathbb{E}_{q_{w_k}^\phi(z|x_i, c_i)} \left[ \log p_{w_k}^\theta(x_i|z, c_i) \right],$$

$$\mathcal{L}_{\text{KL}}(w_k; b_l) = - \frac{1}{|b_l|} \sum_{i=1}^{|b_l|} \text{KL} \left( q_{w_k}^\phi(z|x_i, c_i) \parallel p(z|c_i) \right),$$

where  $x_i$  denotes sample  $i$  in mini-batch  $b_l$ ,  $c_i$  denotes the conditional information of sample  $i$ , that is, the ground truth of  $x_i$ . The term  $q_{w_k}^\phi(z|x_i, c_i)$  refers to the approximate posterior distribution



**Figure 4: A toy example of the knowledge-forgetting problem.**

of local model  $k$ , and the term  $p_{w_k^\theta}(x_i|z, c_i)$  refers to the generative distribution of local model  $k$ . The term  $p(z|c_i)$  refers to the prior distribution, which is typically assumed to follow a standard Gaussian distribution  $z \sim \mathcal{N}(0, 1)$ . The reconstruction loss measures the difference between the generated samples and the input samples, and the KL divergence loss ensures that the distribution of the latent space of the synthetic samples approaches the prior distribution  $z$ .

**4.2.2 Data Generation.** During the data generation stage, the server utilizes the received local decoders and local label distributions to generate synthetic samples  $\mathcal{D}_s$ . Specifically, for each decoder  $\theta_k \in \{K_G\}$ , we sample  $x \in \mathcal{D}_s$  according to client  $k$ 's local label distribution, formally expressed as  $x_i \sim p_{\theta_k}(x_i|z, c_i)$ , where  $z \sim \mathcal{N}(0, 1)$ ,  $c_i$  is the label of  $x_i$ , and  $p_{\theta_k}(x|z, c_i)$  represents the conditional probability distribution defined by the client  $k$ 's decoder  $\theta_k$ . The synthetic samples are subsequently used in knowledge fusion sessions. As the generators are lightweight and the local data partitions often exhibit varying degrees of Non-IID, the generated samples may incorporate a certain level of noise. To enhance the quality of these samples, we introduce an unsupervised solution, which is detailed in Section 4.4.

**4.2.3 Knowledge Fusion.** During the knowledge fusion stage, the global model is initialized by the local classification models as

$$w_0 = \frac{1}{|\{K_C\}|} \sum_{k \in \{K_C\}} w_k, \quad (4)$$

where  $\{K_C\}$  represents the set of local classification models, and  $|\{K_C\}|$  denotes the number of elements in  $\{K_C\}$ . This initialization enables the global model to incorporate knowledge from these local models. Compared with random parameter initialization, this informed initialization allows the global model to achieve superior performance in fewer training rounds. We specifically use  $w_0$  to denote the initial state of the global model, while  $w$  refers to its later states. To additionally obtain knowledge from the clients deploying generative models, we train the global model using the synthetic samples generated during the data generation stage. The training loss function  $\mathcal{L}_{CE}(w; b_s)$  of the global model is defined as  $\mathcal{L}_{CE}(w; b_s) = -\sum_{i=1}^{|b_s|} y_i \log(\hat{y}_i)$ , where  $b_s$  denotes the mini-batch of synthetic samples from  $\mathcal{D}_s$ .

### 4.3 The Knowledge-Forgetting Problem

During the fusion of the clients' knowledge from the generative models, the global model may forget the knowledge learned from the classification models, as depicted in Figure 4. We define this phenomenon as the knowledge-forgetting problem. We propose two methods to alleviate this problem: FedMHO-MD and FedMHO-SD. In FedMHO-MD, we use local classification models as multiple

**Table 1: Diverse local models used in different methods.**

Model	FedAvg, FedDF, DENSE, Co-Boosting	FEDCVAE	FedMHOs
Large	VGG-9	CVAE-large	VGG-9
Small	CNN	CVAE-small	CVAE-small

teacher models to distill their knowledge into the global model. The corresponding loss function  $\mathcal{L}_{KL}(w; b_s)$  is defined as

$$\mathcal{L}_{KL}(w; b_s) = \text{KL} \left[ \left( \frac{1}{|\{K_C\}|} \sum_{k \in \{K_C\}} h_k(w_k, b_s) \right) || h(w, b_s) \right], \quad (5)$$

where  $h_k(w_k, b_s)$  and  $h(w, b_s)$  denote the output logits of samples in the mini-batch  $b_s$  through local model  $w_k$  and global model  $w$ , respectively. In FedMHO-SD, we use the initialized global model  $w_0$  as the teacher model to self-distill the global model. The corresponding loss function  $\mathcal{L}_{KL}(w; b_s)$  is defined as

$$\mathcal{L}_{KL}(w; b_s) = \text{KL} [h_k(w_0, b_s) || h(w, b_s)]. \quad (6)$$

where  $h_k(w_0, b_s)$  denotes the output logits of samples in the mini-batch  $b_s$  through the initialized global model  $w_0$ . Overall, the complete loss function  $\mathcal{L}_g$  for training the global model is defined as:

$$\mathcal{L}_g(w; b_s) = \lambda \mathcal{L}_{CE}(w; b_s) + (1 - \lambda) \mathcal{L}_{KL}(w; b_s), \quad (7)$$

where  $\lambda \in [0, 1]$  denotes the trade-off parameter between  $\mathcal{L}_{CE}$  and  $\mathcal{L}_{KL}$ . In our experiments, we observe that the values of the two loss functions are of similar magnitude, thus, we set  $\lambda = 0.5$  by default.

### 4.4 Unsupervised Data Optimization

We observe that a lightweight CVAE occasionally generates samples that exhibit label confusion. For example, when the decoder receives an input conditioned on label 2 for generating synthetic samples, the model might produce samples with labels 1 or 3 instead. While such instances of mislabeling are infrequent, they introduce noise into the training data of the global model. To enhance the quality of synthetic samples and improve the global model, we propose an unsupervised data optimization strategy based on the K-means algorithm [30].

Specifically, for synthetic samples corresponding to a specific class  $n_c \in N_c$ , we use the flattened pixel values as features and apply K-means clustering to group these samples into one cluster. The feature dimension  $F$  is determined by the number of pixels in the image. For example, for an image of size  $10 \times 10$  pixels, the feature dimension  $F$  is 100. After clustering, we obtain the cluster center  $C_c$  corresponding to each category  $n_c$ . Subsequently, we selectively retain the samples closest to each category's cluster center. By default, we set the ratio of the number of remaining samples to the number of original samples, denoted as  $\mathcal{R}_{th}$ , to 80%. The results of the ablation experiments on  $\mathcal{R}_{th}$  are shown in Section 5.3.6. A detailed algorithmic description is provided in Appendix A.2.

**Table 2: FLOPs of various models with batch size set to 1.**

Model	MNIST & FMNIST	SVHN	EMNIST
VGG-9	126.47M	145.48M	126.47M
CNN	467.23K	4.00M	3.96M
CVAE-large	152.67M	201.76M	152.67M
CVAE-small	408.06K	2.08M	1.06M

## 5 Experiments

### 5.1 Experimental Setup

**5.1.1 Datasets.** Following FEDCVAE [15], we evaluate our methods on four widely used datasets: MNIST [24], Fashion-MNIST [45] (abbreviated as FMNIST), SVHN [50] and EMNIST [7]. To simulate statistical heterogeneity, we employ the Dirichlet distribution as in [19], which is denoted as  $Dir(\alpha)$ . Smaller values of  $\alpha$  correspond to greater heterogeneity among clients' local data partitions. We set  $\alpha = \{0.5, 0.3, 0.1\}$  by default.

**5.1.2 Methods.** Under the constraint of a single communication round (one-shot FL), regularization-based methods such as FedProx [27] and FedGen [57] are not directly applicable. Given the data-free nature of our proposed methods, we compare our proposed FedMHO and its two variants, FedMHO-MD and FedMHO-SD, against three state-of-the-art data-free one-shot FL baselines: DENSE [52], Co-Boosting [8], and FEDCVAE [15], as well as two widely-used baselines: FedAvg [32] and FedDF [29]. Detailed descriptions, variant choices, and naming conventions are provided in Appendix A.3.1. To ensure a fair comparison under the heterogeneous one-shot setting, we restrict FedAvg and FedDF to a single communication round and report the voting outcome of model prototypes as the experimental result.

**5.1.3 Models.** To simulate model-heterogeneous scenarios, we employ two distinct local model prototypes. For the models with more parameters, we utilize VGG-9 and CVAE-large, while for the models with less parameters, we employ CNN and CVAE-small. This selection is designed to reflect the computing power disparities typical in real-world applications, such as those between smartphones and wristbands. Table 1 provides a summary of the specific local models used by each method, and Table 2 details the FLOPs for each local model with a batch size of 1. To demonstrate the efficiency and feasibility of our proposed FedMHOs in resource-constrained environments, our methods consistently train the local models that have the lowest FLOPs. In our evaluation, we report the global model's Top-1 test accuracy in the form of mean  $\pm$  standard deviation, ensuring the reliability of the results.

**5.1.4 Configurations.** We follow the FL setup in FEDCVAE [15] with 10 clients. By default, five clients deploy classification models and the remaining five deploy generative models. For generator-based methods, we produce 6,000 synthetic samples each for MNIST, FMNIST, and SVHN, and 12,000 synthetic samples for EMNIST. The detailed hyperparameter settings are provided in Appendix A.3.3 and Table 7.

### 5.2 Performance Comparison

We present a comprehensive comparison of our proposed FedMHOs with the baselines in Table 3. The results demonstrate that our methods outperform the baselines across various datasets and data partitions, with FedMHO-MD and FedMHO-SD consistently achieving

the top two rankings in Top-1 test accuracy. Specifically, FedMHO-MD and FedMHO-SD achieve the highest Top-1 test accuracy in 6 out of 12 experiments, respectively. Under the  $Dir(0.5)$  local data partition, FedMHO-MD or FedMHO-SD achieves superior results compared to the best-performing baseline. The improvements are 3.06%, 5.44%, 7.23%, and 6.12% on the MNIST, Fashion, SVHN, and EMNIST datasets, respectively.

Even in the absence of the loss function  $\mathcal{L}_{KD}$ , which mitigates the knowledge-forgetting problem of the global model, FedMHO generally outperforms other baselines, except when training on the MNIST and Fashion datasets with the  $Dir(0.1)$  local data partition. We attribute this to the pronounced knowledge deviation between generative and classification local models trained on highly heterogeneous local data. This deviation intensifies the effect of knowledge-forgetting when the global model is trained on synthetic samples. While the best-performing baseline, FEDCVAE, outperforms FedMHO under highly non-IID local data partitions, it fails to surpass FedMHO-MD or FedMHO-SD. Additionally, since FedMHOs' global models are well initialized, FedMHOs converge faster than the baselines during global model training, as shown in Figure 5. Overall, our experimental results demonstrate the effectiveness and robustness of our proposed FedMHOs, highlighting their potential for real-world applications.

### 5.3 Ablation Study

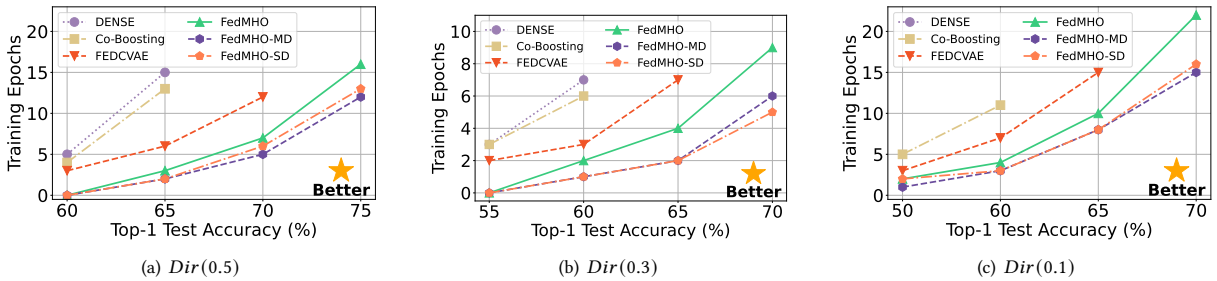
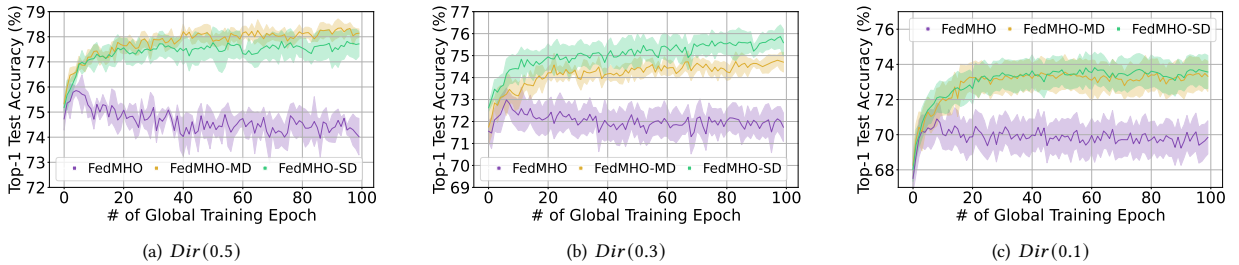
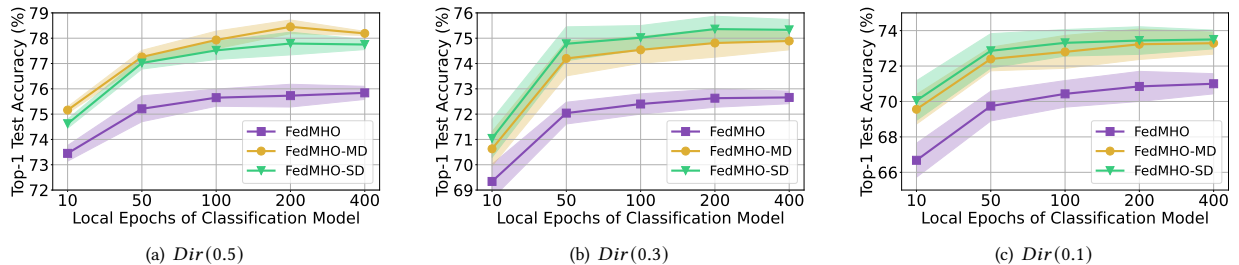
**5.3.1 Contribution of  $\mathcal{L}_{KD}$ .** As mentioned in Section 4.2.3, we employ the loss function  $\mathcal{L}_{KD}$  to mitigate knowledge-forgetting during global model training. Table 3 illustrates that both FedMHO-MD and FedMHO-SD can benefit from  $\mathcal{L}_{KD}$ , thereby improving the global model's Top-1 test accuracy. To further substantiate the necessity of  $\mathcal{L}_{KD}$ , we extend the training epochs of the global model to 100 rounds. The variation of Top-1 test accuracy throughout the training process is depicted in Figure 6. As shown in the figure, in the absence of  $\mathcal{L}_{KD}$ , the global model learns new knowledge from the synthetic data but forgets the original knowledge obtained from aggregating the classification models. This finding underscores the importance of addressing knowledge-forgetting.

**5.3.2 Contribution of Unsupervised Data Optimization.** To demonstrate the impact of the unsupervised data optimization, we present the experimental results obtained without employing this technique in Table 4. As our methods preserve 80% of the generated data by default when utilizing unsupervised data optimization, the global model training on the MNIST/FMNIST/SVHN/EMNIST datasets utilizes 4,800/4,800/4,800/96,000 synthetic samples, respectively. To ensure a fair comparison, we set an equivalent number of synthetic samples when excluding unsupervised data optimization. A comparison between Table 4 and Table 3 indicates that the integration of unsupervised data optimization consistently enhances the global model performance. For example, when the local data partition follows a  $Dir(0.1)$  distribution, FedMHO-SD's Top-1 test accuracy exhibits improvements of 1.22%, 4.23%, 0.71%, and 0.60% on MNIST, FMNIST, SVHN, and EMNIST, respectively.

**5.3.3 Number of Local Epochs.** We vary the number of local epochs of VGG-9 to  $\{10, 50, 200, 400\}$  and plot the resulting Top-1 test accuracy of the global model in Figure 7. When the number of local training epochs is less than 50, increasing the number of

**Table 3: Top-1 test accuracy (%) comparison under various datasets and data partitions. Bold denotes the best result, and underline denotes the second best result.**

Dataset	Partition	FedAvg	FedDF	DENSE	Co-Boosting	FEDCVAE	FedMHO	FedMHO-MD	FedMHO-SD
MNIST	Dir(0.5)	85.91±0.57	88.73±1.29	90.51±2.39	91.42±1.44	92.61±0.25	93.45±0.65	<b>95.71±0.68</b>	<u>95.67±0.47</u>
	Dir(0.3)	74.57±0.73	78.46±0.91	86.12±1.01	89.27±0.48	91.37±0.65	92.43±0.42	<u>93.98±0.69</u>	<b>94.25±0.18</b>
	Dir(0.1)	49.12±2.99	62.58±4.50	73.76±1.72	81.30±0.54	89.73±0.40	88.48±0.99	<u>91.22±0.56</u>	<b>91.55±0.27</b>
Fashion	Dir(0.5)	60.24±1.47	65.66±1.36	72.29±2.63	72.98±1.61	69.11±1.35	75.30±0.62	<u>77.27±0.48</u>	<b>77.73±0.59</b>
	Dir(0.3)	47.36±0.21	59.21±2.20	68.14±0.97	69.50±0.64	68.39±0.49	72.36±0.92	<u>75.69±0.93</u>	<b>76.45±0.53</b>
	Dir(0.1)	30.37±0.79	42.21±2.80	56.12±3.94	61.73±1.36	64.01±2.47	62.14±1.45	<b>71.04±0.50</b>	<u>69.65±0.99</u>
SVHN	Dir(0.5)	60.73±1.91	72.23±1.32	77.61±1.44	77.84±0.42	67.66±0.46	82.27±0.89	<b>85.16±0.73</b>	84.60±0.71
	Dir(0.3)	50.97±3.21	70.77±1.05	71.98±1.87	74.35±1.08	66.23±0.42	80.13±0.72	<b>84.02±0.46</b>	83.11±0.06
	Dir(0.1)	37.91±3.94	55.39±2.68	57.31±1.37	63.22±0.71	64.41±0.19	77.46±0.54	<b>81.09±0.56</b>	80.53±0.30
EMNIST	Dir(0.5)	61.65±1.50	64.52±1.40	68.13±0.48	68.90±0.28	72.33±1.23	75.73±0.45	<b>78.45±0.27</b>	<u>77.79±0.45</u>
	Dir(0.3)	57.37±1.49	62.08±0.52	63.54±0.66	66.16±0.37	69.07±0.72	72.63±0.36	<u>74.81±0.56</u>	<b>75.36±0.51</b>
	Dir(0.1)	39.69±3.18	45.09±2.52	49.07±1.45	58.74±0.72	66.26±1.21	70.85±0.85	<u>73.23±0.86</u>	<b>73.44±0.78</b>

**Figure 5: Global training epochs to reach target Top-1 test accuracy on the EMNIST dataset.****Figure 6: Top-1 test accuracy (%) curve of FedMHOs on the EMNIST dataset.****Figure 7: Top-1 test accuracy (%) of FedMHOs on the EMNIST dataset under various numbers of local epoch of classification models.**

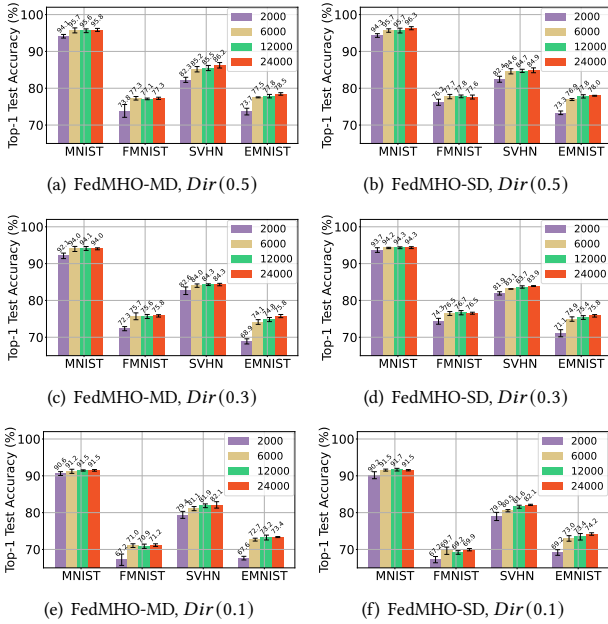
epochs improves the performance of the local classification model, providing a strong initialization for the global model and enhancing its final accuracy. However, when the local epochs exceed 50 rounds, the performance of the global model stabilizes, indicating that the

local VGG-9 models are fully trained. Our method is insensitive to the number of local training epochs.

**5.3.4 Number of Synthetic Samples.** We vary the number of synthetic samples to {2,000, 6,000, 12,000, 24,000} and present the experimental results in Figure 8. We observe that as the number of

**Table 4: Top-1 test accuracy (%) of FedMHO-MD, FedMHO-SD without utilizing unsupervised data optimization.**

Dataset	Partition	FedMHO-MD	FedMHO-SD
MNIST	<i>Dir</i> (0.5)	94.45±0.42	94.70±0.38
	<i>Dir</i> (0.3)	92.33±0.23	93.17±0.32
	<i>Dir</i> (0.1)	90.66±0.31	90.33±0.86
FMNIST	<i>Dir</i> (0.5)	74.56±1.02	75.11±1.93
	<i>Dir</i> (0.3)	73.15±0.52	74.12±2.40
	<i>Dir</i> (0.1)	65.13±0.91	65.42±3.68
SVHN	<i>Dir</i> (0.5)	84.63±0.72	84.11±0.93
	<i>Dir</i> (0.3)	83.23±1.09	82.54±0.32
	<i>Dir</i> (0.1)	80.27±0.86	79.82±1.15
EMNIST	<i>Dir</i> (0.5)	76.79±0.18	76.86±0.33
	<i>Dir</i> (0.3)	74.33±0.68	74.89±0.47
	<i>Dir</i> (0.1)	72.81±0.40	72.84±0.51

**Figure 8: Top-1 test accuracy (%) of FedMHO-MD and FedMHO-SD under various number of synthetic samples.**

synthetic samples increases, the Top-1 test accuracy of the global model improves accordingly. However, the degree of improvement becomes less pronounced once the number of synthetic samples surpasses 6,000.

**5.3.5 Number of Clients.** We vary the number of clients  $K$  from the default 10 to  $\{6, 12, 20\}$  while maintaining an equal number of clients deploying the large model and the small model, with each being  $\frac{K}{2}$ . We conduct experiments on the EMNIST dataset and present the results in Table 5. Compared to the two state-of-the-art baseline methods DENSE, Co-Boosting and FEDCVAE, our proposed FedMHO-MD and FedMHO-SD maintain their effectiveness with varying numbers of participating clients.

**5.3.6 Ratio of Data Optimization.** In Section 5.3.2, we discuss the contribution of unsupervised data optimization. By default, the

**Table 5: Top-1 test accuracy (%) when training EMNIST on *Dir*(0.5) data partition with various numbers of clients  $K$ .**

Method	$K = 6$	$K = 12$	$K = 20$
DENSE	72.96±0.25	67.88±0.45	65.73±0.69
Co-Boosting	73.50±0.38	68.05±0.24	65.46±0.48
FEDCVAE	75.29±0.54	72.65±0.52	71.24±0.83
FedMHO	76.03±0.28	75.23±0.52	74.01±0.65
FedMHO-MD	<b>78.69±0.15</b>	<b>78.16±0.30</b>	<u>76.55±0.56</u>
FedMHO-SD	<u>78.12±0.21</u>	<u>77.45±0.14</u>	<b>75.94±0.38</b>

**Table 6: Top-1 test accuracy (%) of FedMHO, FedMHO-MD, FedMHO-SD on EMNIST with various  $\mathcal{R}_{th}$ .**

Method	Partition	$\mathcal{R}_{th} = 40\%$	$\mathcal{R}_{th} = 60\%$	$\mathcal{R}_{th} = 90\%$
FedMHO-MD	<i>Dir</i> (0.5)	77.58±0.55	78.04±0.44	78.12±0.34
	<i>Dir</i> (0.3)	74.37±0.83	74.60±0.64	74.78±0.47
	<i>Dir</i> (0.1)	72.99±0.95	73.20±0.67	73.15±0.31
FedMHO-SD	<i>Dir</i> (0.5)	77.29±0.62	77.41±0.30	77.75±0.26
	<i>Dir</i> (0.3)	75.06±0.52	75.14±0.56	75.12±0.38
	<i>Dir</i> (0.1)	73.06±0.61	73.11±0.54	73.29±0.62

ratio of remaining samples to original samples  $\mathcal{R}_{th}$  is set to 80%. To investigate the impact of  $\mathcal{R}_{th}$  on the experimental results, we conduct experiments on the EMNIST dataset. We fix the number of synthetic data to 12,000 and vary the value of  $\mathcal{R}_{th}$  between 40%, 60%, and 80%. The experimental results are presented in Table 6. Comparing the results with Table 3 (where  $\mathcal{R}_{th}$  is set to 80%) and Table 4 (which lacks unsupervised data optimization), we find that our default setting of 80% is optimal. Setting  $\mathcal{R}_{th}$  within the range of 60%-90% has little impact on our proposed FedMHO-MD and FedMHO-SD, demonstrating the robustness of our methods.

## 6 Conclusion

In this paper, we propose a novel one-shot FL method called FedMHO, which aims to enhance the performance of the global model when training with clients deploying local models of varying sizes. FedMHO involves a data generation stage and a knowledge fusion stage to aggregate deep classification models and lightweight generative models. Moreover, we provide an unsupervised data solution to improve the quality of synthetic samples and propose two strategies, FedMHO-MD and FedMHO-SD, to mitigate knowledge forgetting. Extensive experiments conducted across various settings validate the efficacy of our method. Overall, FedMHO is the most practical framework currently available for conducting data-free one-shot FL for computing resource-constrained clients. A promising future direction is to explore potential privacy attacks in one-shot FL.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No.92582111). The computation is completed in the HPC Platform of Huazhong University of Science and Technology.

## References

- [1] Syed Thouheed Ahmed, AC Kaladevi, Achyut Shankar, Faye Alqahtani, et al. 2025. Privacy Enhanced Edge-AI Healthcare Devices Authentication: A Federated Learning Approach. *IEEE Transactions on Consumer Electronics* 71, 2 (2025), 5676–5682.
- [2] Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. 2022. FedRolex: Model-Heterogeneous Federated Learning with Rolling Sub-Model Extraction. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35. 29677–29690.
- [3] Sebastian Caldas, Jakub Konečný, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. *arXiv preprint arXiv:1812.07210* (2018).
- [4] Haokun Chen, Hang Li, Yao Zhang, Jinhe Bi, Gengyuan Zhang, Yueqi Zhang, Philip Torr, Jindong Gu, Denis Krompass, and Volker Tresp. 2025. FedBiP: Heterogeneous One-Shot Federated Learning with Personalized Latent Diffusion Models. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 30440–30450.
- [5] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2021. Distilling Knowledge via Knowledge Review. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5008–5017.
- [6] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. 2022. Heterogeneous Ensemble Knowledge Transfer for Training Large Models in Federated Learning. In *Proc. of the 31st International Joint Conferences on Artificial Intelligence Organization (IJCAI)*. 2881–2887.
- [7] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *Proc. of the 2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2921–2926.
- [8] Rong Dai, Yonggang Zhang, Ang Li, Tongliang Liu, Xun Yang, and Bo Han. 2024. Enhancing One-Shot Federated Learning Through Data and Ensemble Co-Boosting. In *Proc. of the 12th International Conference on Learning Representations (ICLR)*.
- [9] Rahool Dembani, Ioannis Karvelas, Nur Arifin Akbar, Stamatia Rizou, Domenico Tegolo, and Spyros Fountas. 2025. Agricultural Data Privacy and Federated Learning: A Review of Challenges and Opportunities. *Computers and Electronics in Agriculture* 232 (2025), 110048.
- [10] Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In *Proc. of the 9th International Conference on Learning Representations (ICLR)*.
- [11] Neel Guha, Ameet Talwalkar, and Virginia Smith. 2019. One-shot Federated Learning. *arXiv preprint arXiv:1902.11175* (2019).
- [12] Otkrist Gupta and Ramesh Kaskar. 2018. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications* 116 (2018), 1–8.
- [13] Chaoyang He, Murali Annamaram, and Salman Avestimehr. 2020. Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 14068–14080.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.
- [15] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. 2023. Data-Free One-Shot Federated Learning Under Very High Statistical Heterogeneity. In *Proc. of the 11th International Conference on Learning Representations (ICLR)*.
- [16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* (2015).
- [17] Junyuan Hong, Haotao Wang, Zhangyang Wang, and Jiayu Zhou. 2022. Efficient Split-Mix Federated Learning for On-Demand and In-Situ Customization. In *Proc. of the 10th International Conference on Learning Representations (ICLR)*.
- [18] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. 2021. FjORD: Fair and Accurate Federated Learning under heterogeneous targets with Ordered Dropout. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 12876–12889.
- [19] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. *arXiv preprint arXiv:1909.06335* (2019).
- [20] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. 2023. Distillation-Based Semi-Supervised Federated Learning for Communication-Efficient Collaborative Training with Non-IID Private Data. *IEEE Transactions on Mobile Computing* 22, 1 (2023), 191–205.
- [21] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.
- [22] Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [23] Solomon Kullback and Richard A Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86.
- [24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-Based Learning Applied to Document Recognition. *Proc. of the IEEE* 86, 11 (1998), 2278–2324.
- [25] Daliang Li and Junpu Wang. 2019. FedMD: Heterogeneous Federated Learning via Model Distillation. *arXiv preprint arXiv:1910.03581* (2019).
- [26] Qinbin Li, Bingsheng He, and Dawn Xiaodong Song. 2020. Practical One-Shot Federated Learning for Cross-Silo Setting. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*. 1484–1490.
- [27] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. *Proc. of Machine Learning and Systems (MLSys) 2* (2020), 429–450.
- [28] Yichen Li, Yijing Shan, Yi Liu, Haozhao Wang, Wei Wang, Yi Wang, and Ruixuan Li. 2025. Personalized Federated Recommendation for Cold-Start Users via Adaptive Knowledge Fusion. In *Proc. of the ACM on Web Conference (WWW)*. ACM, 2700–2709.
- [29] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble Distillation for Robust Model Fusion in Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 2351–2363.
- [30] Stuart Lloyd. 1982. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137.
- [31] L Lyu, H Yu, X Ma, C Chen, L Sun, J Zhao, Q Yang, and PS Yu. 2024. Privacy and Robustness in Federated Learning: Attacks and Defenses. *IEEE Transactions on Neural Networks and Learning Systems* 35, 7 (2024), 8726–8746.
- [32] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proc. of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 1273–1282.
- [33] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. 2018. Open Set Learning with Counterfactual Images. In *Proc. of the European Conference on Computer Vision (ECCV)*. 613–628.
- [34] Saber Salehkaleybar, Arsalan Sharifnassab, and S Jamaloddin Golestani. 2021. One-Shot Federated Learning: Theoretical Limits and Algorithms to Achieve Them. *The Journal of Machine Learning Research* 22, 1 (2021), 8485–8531.
- [35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4510–4520.
- [36] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 28.
- [37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
- [38] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. 2021. Does Knowledge Distillation Really Work?. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 6906–6919.
- [39] Mingxing Tan and Quoc Le. 2021. EfficientNetV2: Smaller Models and Faster Training. In *Proc. of the 38th International Conference on Machine Learning (ICML)*. PMLR, 10096–10106.
- [40] Derui Wang, Chaoran Li, Sheng Wen, Surya Nepal, and Yang Xiang. 2020. Man-in-the-Middle Attacks Against Machine Learning Classifiers Via Malicious Generative Models. *IEEE Transactions on Dependable and Secure Computing* 18, 5 (2020), 2074–2087.
- [41] Haozhao Wang, Yabo Jia, Meng Zhang, Qinghao Hu, Hao Ren, Peng Sun, Yonggang Wen, and Tianwei Zhang. 2024. FedDSE: Distribution-aware Sub-model Extraction for Federated Learning over Resource-constrained Devices. In *Proc. of the ACM on Web Conference (WWW)*. 2902–2913.
- [42] Kaibin Wang, Qiang He, Feifei Chen, Chunyang Chen, Faliang Huang, Hai Jin, and Yun Yang. 2023. FlexiFed: Personalized Federated Learning for Edge Clients with Heterogeneous Model Architectures. In *Proc. of the ACM Web Conference (WWW)*. 2979–2990.
- [43] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. 2018. Dataset distillation. *arXiv preprint arXiv:1811.10959* (2018).
- [44] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. Communication Efficient Federated Learning via Knowledge Distillation. *Nature Communications* 13, 1 (2022), 2032.
- [45] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747* (2017).
- [46] Dezhong Yao, Wanning Pan, Yuexin Shi, Michael J O’Neill, Yutong Dai, Yao Wan, Peilin Zhao, Hai Jin, and Lichao Sun. 2025. FedHM: Efficient Federated Learning for Heterogeneous Models via Low-rank Factorization. *Artificial Intelligence* (2025), 104333.
- [47] Dezhong Yao, Ziquan Zhu, Tongtong Liu, Zhiqiang Xu, and Hai Jin. 2024. Re-thinking Personalized Federated Learning from Knowledge Perspective. In *Proc. of the 53rd International Conference on Parallel Processing*. 991–1000.
- [48] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. 2023. Heterogeneous Federated Learning: State-of-the-art and Research Challenges. *Comput. Surveys* 56, 3 (2023), 1–44.

- [49] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. 2021. See through Gradients: Image Batch Recovery via GradInversion. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16337–16346.
- [50] Netzer Yuval. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. In *Proc. of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.
- [51] Shanbao Zhan, Lianfen Huang, Gaoyu Luo, Shaolong Zheng, Zhibin Gao, and Han-Chieh Chao. 2025. A Review on Federated Learning Architectures for Privacy-Preserving AI: Lightweight and Secure Cloud-Edge-End Collaboration. *Electronics* 14, 13 (2025), 2512.
- [52] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. 2022. DENSE: Data-Free One-Shot Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35. 21414–21428.
- [53] Junyuan Zhang, Songhua Liu, and Xinchao Wang. 2024. One-shot Federated Learning via Synthetic Distiller-Distillate Communication. *arXiv preprint arXiv:2412.05186* (2024).
- [54] Wenbo Zheng, Lan Yan, Chao Gou, and Fei-Yue Wang. 2021. Federated Meta-Learning for Fraudulent Credit Card Detection. In *Proc. of the 30th International Joint Conferences on Artificial Intelligence Organization (IJCAI)*. 4654–4660.
- [55] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. 2021. Learning Placeholders for Open-Set Recognition. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4401–4410.
- [56] Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. 2020. Distilled One-shot Federated Learning. *arXiv preprint arXiv:2009.07999* (2020).
- [57] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2023. Data-Free Knowledge Distillation for Heterogeneous Federated Learning. In *Proc. of the 40th International Conference on Machine Learning (ICML)*. PMLR, 12878–12889.

## A Appendix

### A.1 Detailed Related Work Discussion

We compare our proposed FedMHO with three state-of-the-art baselines: DENSE [52], Co-Boosting [8], and FEDCVAE [15]. DENSE and Co-Boosting, like vanilla FL, deploy classification models on clients. In contrast, FEDCVAE deploys CVAEs, sending decoders and label distributions to the server to generate synthetic samples for global training. FedMHO differs from FEDCVAE in three ways: (1) it uses classifiers on resource-rich clients instead of CVAEs; (2) it starts global training from a pre-trained model, not from scratch; and (3) it tackles knowledge forgetting with two variants, FedMHO-MD and FedMHO-SD.

### A.2 The Unsupervised Data Optimization Algorithm

We provide the algorithm description of our proposed unsupervised data optimization in Algorithm 2.

### A.3 Supplementary Experiments

*A.3.1 Detailed Baseline Methods Description.* A brief introduction to the baseline methods is provided below.

- FedAvg [32] is a foundational FL method that learns a global model by simply averaging the parameters of local models.
- FedDF [29] employs auxiliary public datasets on the server side to ensemble distill the knowledge of the local models into the global model.
- DENSE [52] leverages local models to train a generator on the server side, and after generating synthetic data, it aggregates the local models similarly to FedDF.
- Co-Boosting [8] is an extension of DENSE. In each round of server-side training, it assigns different weights to the local model based on the feedback of the global model, thereby training more effective generators and improving the global model.

- FEDCVAE [15] trains generative models on each client and uses the synthetic data to train the global model.

FEDCVAE-KD and FEDCVAE-ENS are both proposed in [15]. In our experiments, we focus on FEDCVAE-ENS since it consistently outperforms FEDCVAE-KD as reported in [15]. For brevity, we denote FEDCVAE-ENS as **FEDCVAE** throughout the paper. We use **FedMHO** to represent the scenario where the knowledge-forgetting issue is not explicitly addressed during global model training. We further define **FedMHO-MD** and **FedMHO-SD** as two variants that incorporate distinct strategies to mitigate knowledge-forgetting. For simplicity, we refer to the set **{FedMHO, FedMHO-MD, FedMHO-SD}** as **FedMHOs** in the main text and tables. To ensure a fair comparison under the heterogeneous one-shot setting, we restrict both FedAvg and FedDF to a single communication round. Since the server cannot perform iterative aggregation in this setting, we report the voting outcome of model prototypes as the experimental result.

*A.3.2 Detailed Datasets Description.* A brief introduction to these datasets is provided below.

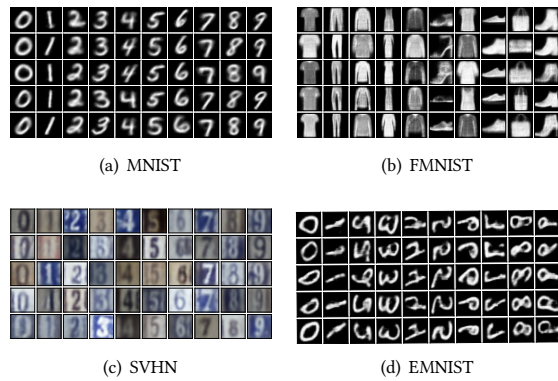
- The MNIST dataset comprises a total of 70,000 samples, divided into 60,000 training samples and 10,000 testing samples. All images are single-channel grayscale with a resolution of 28×28 pixels, spanning 10 distinct categories that represent digits from 0 to 9.
- The FMNIST dataset is designed as a replacement for MNIST, maintaining the same sample size and image dimensions as MNIST but introducing a different set of 10 categories. These categories represent various types of clothing items, providing a more challenging classification task than the original MNIST.
- The EMNIST dataset, specifically using the 'balanced' split, extends the complexity by offering 47 distinct categories. It includes a total of 131,600 samples, divided into 112,800 training samples and 18,800 testing samples. This dataset

---

#### Algorithm 2 The Unsupervised Data Optimization Algorithm

---

- 1: **Input:** Number of categories  $N_c$ , synthetic samples  $\mathcal{D}_s$ , the ratio of the number of remaining samples to the number of original samples  $\mathcal{R}_{th}$ , feature dimension  $F$ .
  - 2: **Output:** Optimized synthetic sample  $\mathcal{D}_s$ .
  - 3: **for**  $n_c = 1 \cdots N_c$  **in parallel do**
  - 4:   Select the synthetic samples with the label  $n_c$  to form the dataset  $\mathcal{D}_s^c$ .
  - 5:   Cluster  $\mathcal{D}_s^c$  into one category using the K-means algorithm and get the cluster center  $C_c$ .
  - 6:   **for each** sample  $i \in \mathcal{D}_s^c$  **in parallel do**
  - 7:     /\* Calculate the Euclidean Distance \*/
  - 8:      $Dis_i = \sqrt{\sum_{f \in F} (i^f - C_c^f)^2}$ .
  - 9:   **end for**
  - 10:   Filter out the  $(1 - \mathcal{R}_{th}) \times |\mathcal{D}_s^c|$  samples with the largest  $Dis_i$  and update  $\mathcal{D}_s$ .
  - 11: **end for**
  - 12: **return** Optimized  $\mathcal{D}_s$ .
-



**Figure 9: Visualization of synthetic data on MNIST, FMNIST, SVHN, and EMNIST, respectively. Five synthetic samples are present for each category.**

**Table 7: Detailed hyperparameter settings utilized in experiments. ‘LR’ is short for ‘Learning Rate’.**

Hyperparameter	MNIST	FMNIST	SVHN	EMNIST
Batch Size	64			
Local Epoch (VGG-9)	200			
Local Epoch (CVAE)	30	40	40	50
Local Optimizer (VGG-9)	SGD			
Local LR (VGG-9)	$5e-3$	$5e-3$	$5e-3$	$1e-3$
Momentum (VGG-9)	0.9			
Local optimizer (CVAE)	Adam			
Local LR (CVAE)	$5e-2$	$5e-2$	$1e-3$	$3e-3$
Global Epoch	10	20	20	30
Global Optimizer	Adam			
Global LR	$1e-5$	$5e-4$	$5e-5$	$5e-5$
Number of Synthetic Data	6,000	6,000	6,000	12,000

is particularly valuable for evaluating models on a broader range of handwritten characters, encompassing digits, uppercase letters, and lowercase letters.

- The SVHN dataset is designed for tasks involving real-world image data. It consists of 99,289 samples, with 73,257 designated for training and 26,032 for testing. Each sample is a  $32 \times 32$  color image, categorized into 10 classes representing digits from 0 to 9. This dataset is notable for its variability in terms of image quality and background noise, making it a robust benchmark in more challenging scenarios.

**A.3.3 Detailed Hyperparameter Settings.** For training local classification models, we utilize the SGD optimizer with a momentum of 0.9 and a learning rate of  $5e-3$ , training for 200 local epochs. For local generative models, we use the Adam optimizer with varying learning rates specific to each dataset:  $5e-2$  for MNIST,  $5e-2$  for FMNIST,  $1e-3$  for SVHN, and  $3e-3$  for EMNIST, with training conducted over 30, 40, 40, and 50 local epochs, respectively. When training the global model, we use the Adam optimizer with a learning rate of  $1e-5$ ,  $5e-4$ ,  $5e-5$ , and  $5e-5$  for MNIST, FMNIST, SVHN, and EMNIST, respectively. The detailed hyperparameter settings are provided in Table 7.

**A.3.4 Homogeneous Local Model.** To verify the effectiveness and practicality of our methods, we use the homogeneous large models listed in Table 1 to train the baseline methods, with the experimental results presented in Table 8. We observe that replacing small models with large models improves the Top-1 test accuracy of the baselines. Among these methods, FEDCVAE achieves the best performance across various datasets when  $Dir(\alpha)$  is set to 0.1. This improvement is due to each client’s local data largely contains only a subset of categories. This configuration facilitates the local training of a large generative model that generates high-quality synthetic data.

Despite half of the clients in our methods utilizing small local generative models, our proposed FedMHO-MD and FedMHO-SD achieve comparable or superior performance to the baselines that exclusively employ large local models. For example, when the client data distribution follows  $Dir(0.5)$  partition, FedMHO-MD achieves the highest Top-1 test accuracy on the MNIST dataset, while FedMHO-SD achieves the highest accuracy on the SVHN dataset. Both methods achieve the second-highest Top-1 test accuracy on the Fashion and EMNIST datasets, with differences of only 3.09% and 2.68% compared to the Co-Boosting method, which is also trained with the large model. Therefore, we conclude that our methods enable resource-constrained clients to participate in FL training and help achieve a high-performing global model. Considering that real-world FL scenarios often involve model heterogeneity, our method demonstrates significant practicality.

**A.3.5 Visualization of Synthetic Samples.** We present the visualization of synthetic samples generated by our methods in Figure 9. Due to the use of a lightweight CVAE, the contours of the generated images appear somewhat blurred compared to real data. However, the category information is generally accurate, and there is noticeable diversity among samples within the same category.

**A.3.6 Extensions for Various Classification Models.** To further validate the efficacy of our methods, we adopt a wider range of classification model prototypes for FL training. Within this context, smaller models are kept constant, and large models encompass EfficientNet-V2 [39], ResNet-50 [14], MobileNet-V2 [35], and VGG-9. Each model prototype is deployed on two clients. The local optimizer for classification models adopts Adam with a learning rate of  $5e-4$ . Other configurations remain constant. The experimental results on the EMNIST dataset are detailed in Table 9. The Top-1 test accuracies of our proposed FedMHO-MD and FedMHO-SD consistently outperform other baseline methods.

## A.4 Discussion

The experimental results presented in Section 5 provide significant insights into the performance of our proposed FedMHOs. Overall, FedMHO-MD and FedMHO-SD exhibit the highest Top-1 test accuracy and the fastest global model convergence speed across various datasets and data partitions. This demonstrates the effectiveness and efficiency of our methods in tackling the challenges associated with heterogeneous one-shot FL. In our experiments, we deploy lightweight generative or classification models for validation on relatively simple datasets. For more challenging datasets, more complex local models can be adopted to train local data and thereby achieve a satisfactory global model after aggregation.

**Table 8: Top-1 test accuracy (%) of FedAvg, FedDF, DENSE, Co-Boosting, and FEDCVAE when training with homogeneous large local models, and Top-1 test accuracy (%) of FedMHO-MD and FedMHO-SD when training with heterogeneous local models as in Table 3.**

Dataset	Partition	FedAvg	FedDF	DENSE	Co-Boosting	FEDCVAE	FedMHO-MD	FedMHO-SD
MNIST	<i>Dir</i> (0.5)	88.44±1.08	91.13±0.25	93.56±0.74	93.61±0.72	94.14±0.61	<b>95.71±0.68</b>	<u>95.67±0.47</u>
	<i>Dir</i> (0.3)	79.96±1.67	86.00±0.64	91.63±0.44	91.95±0.37	93.72±0.35	<u>93.98±0.69</u>	<b>94.25±0.18</b>
	<i>Dir</i> (0.1)	56.67±2.38	78.06±1.01	74.99±0.29	82.34±0.42	<b>93.01±0.63</b>	91.22±0.56	<u>91.55±0.27</u>
Fashion	<i>Dir</i> (0.5)	67.13±1.43	68.63±0.78	80.59±1.34	<b>80.82±0.71</b>	76.18±0.54	77.27±0.48	<u>77.73±0.59</u>
	<i>Dir</i> (0.3)	54.16±1.22	62.70±1.31	67.18±1.69	69.11±0.86	74.90±0.72	<u>75.69±0.93</u>	<b>76.45±0.53</b>
	<i>Dir</i> (0.1)	38.39±1.85	51.10±1.98	49.26±0.84	55.60±0.76	<b>74.11±1.46</b>	<u>71.04±0.50</u>	69.65±0.99
SVHN	<i>Dir</i> (0.5)	64.14±0.87	75.33±0.97	79.99±0.53	79.56±0.44	85.02±0.45	<b>85.16±0.73</b>	<u>84.60±0.71</u>
	<i>Dir</i> (0.3)	56.62±1.75	73.04±0.80	73.69±0.77	74.08±0.51	<u>83.82±0.60</u>	<b>84.02±0.46</b>	83.11±0.06
	<i>Dir</i> (0.1)	45.89±2.34	59.72±1.25	65.16±1.60	69.24±0.64	<b>83.71±0.49</b>	<u>81.09±0.56</u>	80.53±0.30
EMNIST	<i>Dir</i> (0.5)	68.55±1.46	71.07±0.90	80.84±1.16	<b>81.13±0.65</b>	77.67±0.78	<u>78.45±0.27</u>	77.79±0.45
	<i>Dir</i> (0.3)	62.03±0.71	66.81±1.11	75.45±1.93	75.96±0.85	<b>76.48±0.34</b>	74.81±0.56	<u>75.36±0.51</u>
	<i>Dir</i> (0.1)	44.16±0.56	49.13±1.07	64.98±2.68	68.39±1.06	<b>74.55±0.14</b>	73.23±0.86	<u>73.44±0.78</u>

**Table 9: Top-1 test accuracy (%) comparison on EMNIST dataset under various classification models.**

Method	<i>Dir</i> (0.5)	<i>Dir</i> (0.3)	<i>Dir</i> (0.1)
FedAvg	62.38±2.03	50.71±2.43	33.07±1.57
FedDF	68.49±0.84	61.65±0.91	48.98±1.68
DENSE	72.57±0.36	70.53±0.72	55.76±1.35
Co-Boosting	72.81±0.48	71.14±0.53	58.23±0.69
FEDCVAE	73.12±1.12	68.38±1.55	66.03±2.34
FedMHO	78.16±0.50	72.19±1.47	62.04±1.73
FedMHO-MD	<u>79.89±0.55</u>	<b>75.76±1.17</b>	<b>68.22±1.35</b>
FedMHO-SD	<b>80.38±0.24</b>	<u>75.44±0.98</u>	<u>67.19±1.08</u>

Furthermore, in evaluating the advantages of FedMHO-MD versus FedMHO-SD for end users, the comprehensive experimental results from Section 5 indicate that both methods perform comparably. If server-side training overhead is a concern, FedMHO-SD emerges as a more advantageous option. Compared to FedMHO-MD, FedMHO-SD reduces inference requirements when using knowledge distillation to mitigate the knowledge-forgetting problem. However, the server-side overhead in FL scenarios is usually controllable. Therefore, we propose both FedMHO-MD and FedMHO-SD as state-of-the-art algorithms for heterogeneous one-shot FL.