Federated Model Heterogeneous Matryoshka Representation Learning

Liping Yi[1,2], **Han Yu**[2], **Chao Ren**[2], **Gang Wang**[1,*], **Xiaoguang Liu**[1,*], **Xiaoxiao Li**[3,4]
[1]College of Computer Science, TMCC, SysNet, DISSec, GTIISC, Nankai University, China
[2]College of Computing and Data Science, Nanyang Technological University, Singapore
[3]Department of Electrical and Computer Engineering, The University of British Columbia, Canada
[4]Vector Institute, Canada
{yiliping, wgzwp, liuxg}@nbjl.nankai.edu.cn
{han.yu, chao.ren}@ntu.edu.sg, xiaoxiao.li@ece.ubc.ca

Abstract

Model heterogeneous federated learning (MHeteroFL) enables FL clients to collaboratively train models with heterogeneous structures in a distributed fashion. However, existing MHeteroFL methods rely on training loss to transfer knowledge between the client model and the server model, resulting in limited knowledge exchange. To address this limitation, we propose the Federated model heterogeneous Matryoshka Representation Learning (FedMRL) approach for supervised learning tasks. It adds an auxiliary small homogeneous model shared by clients with heterogeneous local models. (1) The generalized and personalized representations extracted by the two models' feature extractors are fused by a personalized lightweight representation projector. This step enables representation fusion to adapt to local data distribution. (2) The fused representation is then used to construct Matryoshka representations with multi-dimensional and multi-granular embedded representations learned by the global homogeneous model header and the local heterogeneous model header. This step facilitates multi-perspective representation learning and improves model learning capability. Theoretical analysis shows that FedMRL achieves a $\mathcal{O}(1/T)$ non-convex convergence rate. Extensive experiments on benchmark datasets demonstrate its superior model accuracy with low communication and computational costs compared to seven state-of-the-art baselines. It achieves up to 8.48% and 24.94% accuracy improvement compared with the state-of-the-art and the best same-category baseline, respectively.

1 Introduction

Traditional federated learning (FL) [32, 47, 46, 12] often relies on a central FL server to coordinate multiple data owners (a.k.a., FL clients) to train a global shared model without exposing local data. In each communication round, the server broadcasts the global model to the clients. A client trains it on its local data and sends the updated local model to the FL server. The server aggregates local models to produce a new global model. These steps are repeated until the global model converges. During the runtime of FL, only model parameters are transmitted between the server and clients, preserving data privacy[14, 56, 51].

However, the above design cannot handle the following heterogeneity challenges [53] commonly found in practical FL applications: (1) Data heterogeneity [42]: FL clients' local data often follow non-independent and identically distributions (non-IID). A single global model produced by aggregating

[*] Corresponding authors

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

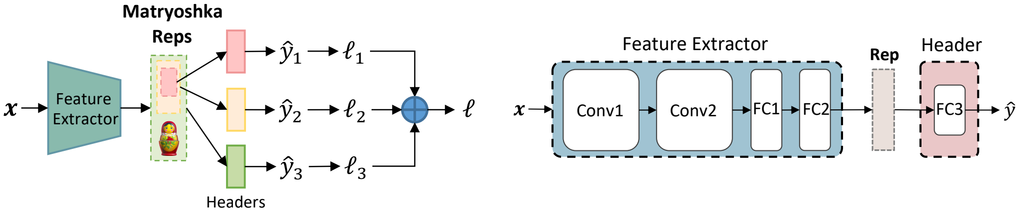


Figure 1: Left: Matryoshka Representation Learning. Right: Feature extractor and prediction header.

local models trained on non-IID data might not perform well on all clients [49, 48]. (2) System heterogeneity [11]: FL clients can have diverse system configurations in terms of computing power and network bandwidth. Training the same model structure among such clients means that the global model size must accommodate the weakest device, leading to sub-optimal performance on other more powerful clients [52, 54, 50]. (3) Model heterogeneity [43]: When FL clients are enterprises, they might have heterogeneous proprietary models which cannot be directly shared with others during FL training due to intellectual property (IP) protection concerns.

To address these challenges, the field of model heterogeneous federated learning (MHeteroFL) [55] has emerged. It enables FL clients to train local models with tailored structures suitable for local system resources and local data distributions. Existing MHeteroFL methods [41, 45] are limited in terms of knowledge transfer capabilities as they commonly leverage the training loss between server and client models for this purpose. This design leads to model performance bottlenecks, incurs high communication and computation costs, and risks exposing private local model structures and data.

Recently, Matryoshka Representation Learning (MRL) [24] has emerged to tailor representation dimensions based on the computational and storage costs required by downstream tasks to achieve a near-optimal trade-off between model performance and inference costs. As shown in Figure 1(left), the representation extracted by the feature extractor is constructed to form Matryoshka Representations involving a series of embedded representations ranging from low-to-high dimensions and coarse-to-fine granularities. Each of them is processed by a single output layer for calculating loss, and the sum of losses from all branches is used to update model parameters. This design is inspired by the insight that people often first perceive the coarse aspect of a target before observing the details, with multi-perspective observations enhancing understanding.

Inspired by MRL, we address the aforementioned limitations of MHeteroFL by proposing the Federated model heterogeneous Matryoshka Representation Learning (`FedMRL`) approach for supervised learning tasks. For each client, a shared global auxiliary homogeneous small model is added to interact with its heterogeneous local model. Both two models consist of a feature extractor and a prediction header, as depicted in Figure 1(right). `FedMRL` has two key design innovations. **(1) Adaptive Representation Fusion**: for each local data sample, the feature extractors of the two local models extract generalized and personalized representations, respectively. The two representations are spliced and then mapped to a fused representation by a lightweight personalized representation projector adapting to local non-IID data. **(2) Multi-Granularity Representation Learning**: the fused representation is used to construct Matryoshka Representations involving multi-dimension and multi-granularity embedded representations, which are processed by the prediction headers of the two models, respectively. The sum of their losses is used to update all models, which enhances the model learning capability owing to multi-perspective representation learning.

The personalized multi-granularity MRL enhances representation knowledge interaction between the homogeneous global model and the heterogeneous client local model. Each client's local model and data are not exposed during training for privacy-preservation. The server and clients only transmit the small homogeneous models, thereby incurring low communication costs. Each client only trains a small homogeneous model and a lightweight representation projector in addition, incurring low extra computational costs. We theoretically derive the $\mathcal{O}(1/T)$ non-convex convergence rate of `FedMRL` and verify that it can converge over time. Experiments on benchmark datasets comparing `FedMRL` against seven state-of-the-art baselines demonstrate its superiority. It improves model accuracy by up to 8.48% and 24.94% over the best baseline and the best same-category baseline, while incurring lower communication and computation costs.

2 Related Work

Existing MHeteroFL works can be divided into the following four categories.

MHeteroFL with Adaptive Subnets. These methods [3, 4, 5, 11, 16, 57, 65] construct heterogeneous local subnets of the global model by parameter pruning or special designs to match with each client's local system resources. The server aggregates heterogeneous local subnets wise parameters to generate a new global model. In cases where clients hold black-box local models with heterogeneous structures not derived from a common global model, the server is unable to aggregate them.

MHeteroFL with Knowledge Distillation. These methods [6, 8, 9, 17, 18, 19, 25, 26, 28, 30, 33, 35, 38, 39, 44, 58, 60] often perform knowledge distillation on heterogeneous client models by leveraging a public dataset with the same data distribution as the learning task. In practice, such a suitable public dataset can be hard to find. Others [13, 61, 62, 64] train a generator to synthesize a shared dataset to deal with this issue. However, this incurs high training costs. The rest (`FD` [21], `FedProto` [43] and others [1, 2, 15, 53, 59]) share the intermediate information of client local data for knowledge fusion.

MHeteroFL with Model Split. These methods split models into feature extractors and predictors. Some [7, 10, 34, 36] share homogeneous feature extractors across clients and personalize predictors, while others (`LG-FedAvg` [27] and [20, 29]) do the opposite. Such methods expose part of the local model structures, which might not be acceptable if the models are proprietary IPs of the clients.

MHeteroFL with Mutual Learning. These methods (`FedAPEN` [37], `FML` [41], `FedKD` [45] and others [31, 22]) add a shared global homogeneous small model on top of each client's heterogeneous local model. For each local data sample, the distance of the outputs from these two models is used as the mutual loss to update model parameters. Nevertheless, the mutual loss only transfers limited knowledge between the two models, resulting in model performance bottlenecks.

The proposed `FedMRL` approach further optimizes mutual learning-based MHeteroFL by enhancing the knowledge transfer between the server and client models. It achieves personalized adaptive representation fusion and multi-perspective representation learning, thereby facilitating more knowledge interaction across the two models and improving model performance.

3 The Proposed `FedMRL` Approach

`FedMRL` aims to tackle data, system, and model heterogeneity in supervised learning tasks, where a central FL server coordinates N FL clients to train heterogeneous local models. The server maintains a global homogeneous small model $\mathcal{G}(\theta)$ shared by all clients. Figure 2 depicts its workflow [2]:

① In each communication round, K clients participate in FL (*i.e.*, the client participant rate $C = K/N$). The global homogeneous small model $\mathcal{G}(\theta)$ is broadcast to them.

② Each client k holds a heterogeneous local model $\mathcal{F}_k(\omega_k)$ ($\mathcal{F}_k(\cdot)$ is the heterogeneous model structure, and ω_k are personalized model parameters). Client k simultaneously trains the heterogeneous local model and the global homogeneous small model on local non-IID data D_k (D_k follows the non-IID distribution P_k) via personalized Matryoshka Representations Learning with a personalized representation projector $\mathcal{P}_k(\varphi_k)$ in an end-to-end manner.

③ The updated homogeneous small models are uploaded to the server for aggregation to produce a new global model for knowledge fusion across heterogeneous clients.

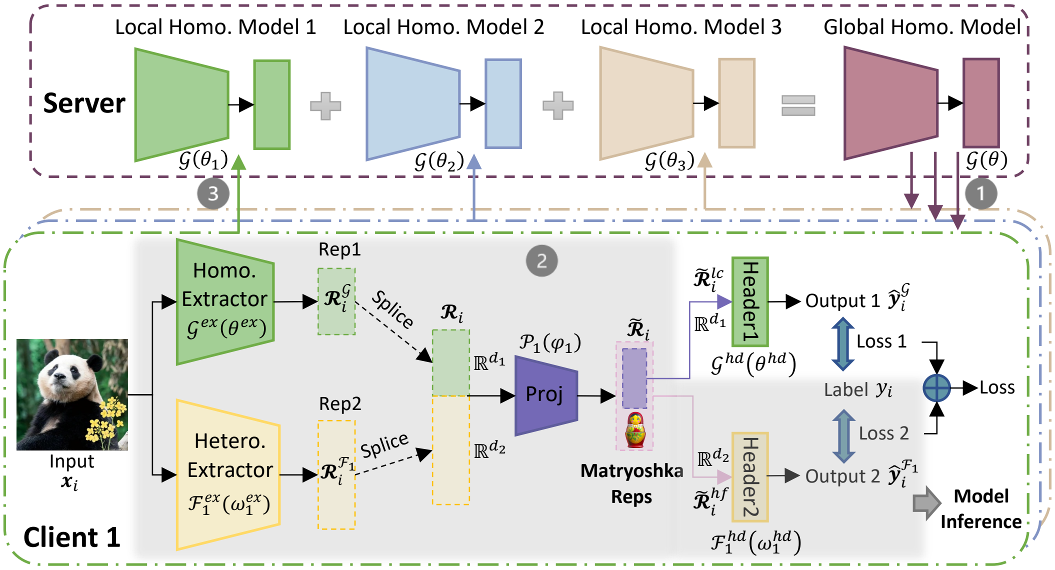
The objective of `FedMRL` is to minimize the sum of the loss from the combined models ($\mathcal{W}_k(w_k) = (\mathcal{G}(\theta) \circ \mathcal{F}_k(\omega_k)|\mathcal{P}_k(\varphi_k))$) on all clients, *i.e.*,

$$\min_{\theta,\omega_{0,\ldots,N-1}} \sum_{k=0}^{N-1} \ell\left(\mathcal{W}_k\left(D_k;\left(\theta \circ \omega_k \mid \varphi_k\right)\right)\right). \tag{1}$$

These steps repeat until each client's model converges. After FL training, a client uses its local combined model without the global header for inference. [3]

[2] Algorithm 1 in Appendix A describes the `FedMRL` algorithm.

[3] Appendix C.3 provides experimental evidence for inference model selection.



Server
Local Homo. Model 1
Local Homo. Model 2
Local Homo. Model 3
Global Homo. Model
$\mathcal{G}(\theta_1)$
$\mathcal{G}(\theta_2)$
$\mathcal{G}(\theta_3)$
$\mathcal{G}(\theta)$
3
1
2
Homo. Extractor $\mathcal{G}^{ex}(\theta^{ex})$
Rep1
$\boldsymbol{\mathcal{R}}_i^{\mathcal{G}}$
Splice
$\boldsymbol{\mathcal{R}}_i$
$\mathbb{R}^{d_1}$
$\mathcal{P}_1(\varphi_1)$
Proj
$\widetilde{\boldsymbol{\mathcal{R}}}_i$
$\widetilde{\boldsymbol{\mathcal{R}}}_i^{lc}$
$\mathbb{R}^{d_1}$
Header1
$\mathcal{G}^{hd}(\theta^{hd})$
Output 1 $\hat{\boldsymbol{y}}_i^{\mathcal{G}}$
Loss 1
Label y_i
Loss
Loss 2
Input $\boldsymbol{x}_i$
Hetero. Extractor $\mathcal{F}_1^{ex}(\omega_1^{ex})$
Rep2
$\boldsymbol{\mathcal{R}}_i^{\mathcal{F}_1}$
Splice
$\mathbb{R}^{d_2}$
Matryoshka Reps
$\mathbb{R}^{d_2}$
$\widetilde{\boldsymbol{\mathcal{R}}}_i^{hf}$
Header2
$\mathcal{F}_1^{hd}(\omega_1^{hd})$
Output 2 $\hat{\boldsymbol{y}}_i^{\mathcal{F}_1}$
Model Inference
Client 1

Figure 2: The workflow of FedMRL.

3.1 Adaptive Representation Fusion

We denote client k's heterogeneous local model feature extractor as $\mathcal{F}_k^{ex}(\omega_k^{ex})$, and prediction header as $\mathcal{F}_k^{hd}(\omega_k^{hd})$. We denote the homogeneous global model feature extractor as $\mathcal{G}^{ex}(\theta^{ex})$ and prediction header as $\mathcal{G}^{hd}(\theta^{hd})$. Client k's local personalized representation projector is denoted as $\mathcal{P}_k(\varphi_k)$. In the t-th communication round, client k inputs its local data sample $(\boldsymbol{x}_i, y_i) \in D_k$ into the two feature extractors to extract generalized and personalized representations as:

$$\boldsymbol{\mathcal{R}}_i^{\mathcal{G}} = \mathcal{G}^{ex}(\boldsymbol{x}_i;\theta^{ex,t-1}), \boldsymbol{\mathcal{R}}_i^{\mathcal{F}_k} = \mathcal{F}_k^{ex}(\boldsymbol{x}_i;\omega_k^{ex,t-1}). \tag{2}$$

The two extracted representations $\boldsymbol{\mathcal{R}}_i^{\mathcal{G}} \in \mathbb{R}^{d_1}$ and $\boldsymbol{\mathcal{R}}_i^{\mathcal{F}_k} \in \mathbb{R}^{d_2}$ are spliced as:

$$\boldsymbol{\mathcal{R}}_i = \boldsymbol{\mathcal{R}}_i^{\mathcal{G}} \circ \boldsymbol{\mathcal{R}}_i^{\mathcal{F}_k}. \tag{3}$$

Then, the spliced representation is mapped into a fused representation by the lightweight representation projector $\mathcal{P}_k(\varphi_k^{t-1})$ as:

$$\widetilde{\boldsymbol{\mathcal{R}}}_i = \mathcal{P}_k(\boldsymbol{\mathcal{R}}_i;\varphi_k^{t-1}), \tag{4}$$

where the projector can be a one-layer linear model or multi-layer perceptron. The fused representation $\widetilde{\boldsymbol{\mathcal{R}}}_i$ contains both generalized and personalized feature information. It has the same dimension as the client's local heterogeneous model representation $\mathbb{R}^{d_2}$, which ensures the representation dimension $\mathbb{R}^{d_2}$ and the client local heterogeneous model header parameter dimension $\mathbb{R}^{d_2 \times L}$ (L is the label dimension) match.

The representation projector can be updated as the two models are being trained on local non-IID data. Hence, it achieves personalized representation fusion adaptive to local data distributions. Splicing the representations extracted by two feature extractors can keep the relative semantic space positions of the generalized and personalized representations, benefiting the construction of multi-granularity Matryoshka Representations. Owing to representation splicing, the representation dimensions of the two feature extractors can be different (*i.e.*, $d_1 \leq d_2$). Therefore, we can vary the representation dimension of the small homogeneous global model to improve the trade-off among model performance, storage requirement and communication costs.

In addition, each client's local model is treated as a black box by the FL server. When the server broadcasts the global homogeneous small model to the clients, each client can adjust the linear layer dimension of the representation projector to align it with the dimension of the spliced representation. In this way, different clients may hold different representation projectors. When a new model-agnostic client joins in FedMRL, it can adjust its representation projector structure for local model training. Therefore, FedMRL can accommodate FL clients owning local models with diverse structures.

3.2 Multi-Granular Representation Learning

To construct multi-dimensional and multi-granular Matryoshka Representations, we further extract a low-dimension coarse-granularity representation $\widetilde{\boldsymbol{\mathcal{R}}}_i^{lc}$ and a high-dimension fine-granularity representation $\widetilde{\boldsymbol{\mathcal{R}}}_i^{hf}$ from the fused representation $\widetilde{\boldsymbol{\mathcal{R}}}_i$. They align with the representation dimensions $\{\mathbb{R}^{d_1}, \mathbb{R}^{d_2}\}$ of two feature extractors for matching the parameter dimensions $\{\mathbb{R}^{d_1 \times L}, \mathbb{R}^{d_2 \times L}\}$ of the two prediction headers,

$$\widetilde{\boldsymbol{\mathcal{R}}}_i^{lc} = \widetilde{\boldsymbol{\mathcal{R}}}_i^{1:d_1}, \widetilde{\boldsymbol{\mathcal{R}}}_i^{hf} = \widetilde{\boldsymbol{\mathcal{R}}}_i^{1:d_2}. \tag{5}$$

The embedded low-dimension coarse-granularity representation $\widetilde{\boldsymbol{\mathcal{R}}}_i^{lc} \in \mathbb{R}^{d_1}$ incorporates coarse generalized and personalized feature information. It is learned by the global homogeneous model header $\mathcal{G}^{hd}(\theta^{hd,t-1})$ (parameter space: $\mathbb{R}^{d_1 \times L}$) with generalized prediction information to produce:

$$\hat{y}_i^{\mathcal{G}} = \mathcal{G}^{hd}(\widetilde{\boldsymbol{\mathcal{R}}}_i^{lc}; \theta^{hd,t-1}). \tag{6}$$

The embedded high-dimension fine-granularity representation $\widetilde{\boldsymbol{\mathcal{R}}}_i^{hf} \in \mathbb{R}^{d_2}$ carries finer generalized and personalized feature information, which is further processed by the heterogeneous local model header $\mathcal{F}_k^{hd}(\omega_k^{hd,t-1})$ (parameter space: $\mathbb{R}^{d_2 \times L}$) with personalized prediction information to generate:

$$\hat{y}_i^{\mathcal{F}_k} = \mathcal{F}_k^{hd}(\widetilde{\boldsymbol{\mathcal{R}}}_i^{hf}; \omega_k^{hd,t-1}). \tag{7}$$

We compute the losses ℓ (*e.g.*, cross-entropy loss [63]) between the two outputs and the label y_i as:

$$\ell_i^{\mathcal{G}} = \ell(\hat{y}_i^{\mathcal{G}}, y_i),\ \ell_i^{\mathcal{F}_k} = \ell(\hat{y}_i^{\mathcal{F}_k}, y_i). \tag{8}$$

Then, the losses of the two branches are weighted by their importance $m_i^{\mathcal{G}}$ and $m_i^{\mathcal{F}_k}$ and summed as:

$$\ell_i = m_i^{\mathcal{G}} \cdot \ell_i^{\mathcal{G}} + m_i^{\mathcal{F}_k} \cdot \ell_i^{\mathcal{F}_k}. \tag{9}$$

We set $m_i^{\mathcal{G}} = m_i^{\mathcal{F}_k} = 1$ by default to make the two models contribute equally to model performance. The complete loss ℓ_i is used to simultaneously update the homogeneous global small model, the heterogeneous client local model, and the representation projector via gradient descent:

$$\begin{aligned} \theta_k^t &\leftarrow \theta^{t-1} - \eta_\theta \nabla \ell_i, \\ \omega_k^t &\leftarrow \omega_k^{t-1} - \eta_\omega \nabla \ell_i, \\ \varphi_k^t &\leftarrow \varphi_k^{t-1} - \eta_\varphi \nabla \ell_i, \end{aligned} \tag{10}$$

where $\eta_\theta, \eta_\omega,\ \eta_\varphi$ are the learning rates of the homogeneous global small model, the heterogeneous local model and the representation projector. We set $\eta_\theta = \eta_\omega = \ \eta_\varphi$ by default to ensure stable model convergence. In this way, the generalized and personalized fused representation is learned from multiple perspectives, thereby improving model learning capability.

4 Convergence Analysis

Based on notations, assumptions and proofs in Appendix B, we analyse the convergence of `FedMRL`.

Lemma 1 ***Local Training.*** *Given Assumptions 1 and 2, the loss of an arbitrary client's local model w in local training round $(t+1)$ is bounded by:*

$$\mathbb{E}[\mathcal{L}_{(t+1)E}] \leq \mathcal{L}_{tE+0} + (\frac{L_1 \eta^2}{2} - \eta) \sum_{e=0}^{E} \|\nabla \mathcal{L}_{tE+e}\|_2^2 + \frac{L_1 E \eta^2 \sigma^2}{2}. \tag{11}$$

Lemma 2 ***Model Aggregation.*** *Given Assumptions 2 and 3, after local training round $(t+1)$, a client's loss before and after receiving the updated global homogeneous small models is bounded by:*

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \leq \mathbb{E}[\mathcal{L}_{(t+1)E}] + \eta \delta^2. \tag{12}$$

Theorem 1 ***One Complete Round of FL.*** *Given the above lemmas, for any client, after receiving the updated global homogeneous small model, we have:*

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \leq \mathcal{L}_{tE+0} + (\frac{L_1\eta^2}{2} - \eta)\sum_{e=0}^{E}\|\nabla\mathcal{L}_{tE+e}\|_2^2 + \frac{L_1E\eta^2\sigma^2}{2} + \eta\delta^2. \tag{13}$$

Theorem 2 ***Non-convex Convergence Rate of FedMRL.*** *Given Theorem 1, for any client and an arbitrary constant $\epsilon > 0$, the following holds:*

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_{tE+e}\|_2^2 \leq \frac{\frac{1}{T}\sum_{t=0}^{T-1}[\mathcal{L}_{tE+0} - \mathbb{E}[\mathcal{L}_{(t+1)E+0}]] + \frac{L_1E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}} < \epsilon, \\ s.t.\ \eta < \frac{2(\epsilon - \delta^2)}{L_1(\epsilon + E\sigma^2)}. \end{aligned} \tag{14}$$

Therefore, we conclude that any client's local model can converge at a non-convex rate of $\epsilon \sim \mathcal{O}(1/T)$ in FedMRL if the learning rates of the homogeneous small model, the client local heterogeneous model and the personalized representation projector satisfy the above conditions.

5 Experimental Evaluation

We implement FedMRL on Pytorch, and compare it with seven state-of-the-art MHeteroFL methods. The experiments are carried out over two benchmark supervised image classification datasets on 4 NVIDIA GeForce 3090 GPUs (24GB Memory).[4]

5.1 Experiment Setup

Datasets. The benchmark datasets adopted are CIFAR-10 and CIFAR-100 [5] [23], which are commonly used in FL image classification tasks for the evaluating existing MHeteroFL algorithms. CIFAR-10 has $60,000$ 32×32 colour images across 10 classes, with $50,000$ for training and $10,000$ for testing. CIFAR-100 has $60,000$ 32×32 colour images across 100 classes, with $50,000$ for training and $10,000$ for testing. We follow [40] and [37] to construct two types of non-IID datasets. Each client's non-IID data are further divided into a training set and a testing set with a ratio of $8:2$.

- **Non-IID (Class):** For CIFAR-10 with 10 classes, we randomly assign 2 classes to each FL client. For CIFAR-100 with 100 classes, we randomly assign 10 classes to each FL client. The fewer classes each client possesses, the higher the non-IIDness.
- **Non-IID (Dirichlet):** To produce more sophisticated non-IID data settings, for each class of CIFAR-10/CIFAR-100, we use a Dirichlet(α) function to adjust the ratio between the number of FL clients and the assigned data. A smaller α indicates more pronounced non-IIDness.

Models. We evaluate MHeteroFL algorithms under model-homogeneous and heterogeneous FL scenarios. FedMRL's representation projector is a one-layer linear model (parameter space: $\mathbb{R}^{d2\times(d_1+d_2)}$).

- **Model-Homogeneous FL:** All clients train CNN-1 in Table 2 (Appendix C.1). The homogeneous global small models in FML and FedKD are also CNN-1. The extra homogeneous global small model in FedMRL is CNN-1 with a smaller representation dimension d_1 (*i.e.*, the penultimate linear layer dimension) than the CNN-1 model's representation dimension d_2, $d_1 \leq d_2$.
- **Model-Heterogeneous FL:** The 5 heterogeneous models {CNN-1, ..., CNN-5} in Table 2 (Appendix C.1) are evenly distributed among FL clients. The homogeneous global small models in FML and FedKD are the smallest CNN-5 models. The homogeneous global small model in FedMRL is the smallest CNN-5 with a reduced representation dimension d_1 compared with the CNN-5 model representation dimension d_2, *i.e.*, $d_1 \leq d_2$.

[4] https://github.com/LipingYi/FedMRL

[5] https://www.cs.toronto.edu/%7Ekriz/cifar.html

Table 1: Average test accuracy (%) in model-heterogeneous FL.

FL Setting	N=10, C=100%		N=50, C=20%		N=100, C=10%	
Method	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Standalone	96.53	72.53	95.14	62.71	91.97	53.04
LG-FedAvg [27]	96.30	72.20	94.83	60.95	91.27	45.83
FD [21]	96.21	-	-	-	-	-
FedProto [43]	96.51	72.59	95.48	62.69	92.49	53.67
FML [41]	30.48	16.84	-	21.96	-	15.21
FedKD [45]	80.20	53.23	77.37	44.27	73.21	37.21
FedAPEN [37]	-	-	-	-	-	-
`FedMRL`	**96.63**	**74.37**	**95.70**	**66.04**	**95.85**	**62.15**
`FedMRL`-*Best B.*	*0.10*	*1.78*	*0.22*	*3.33*	*3.36*	*8.48*
`FedMRL`-*Best S.C.B.*	*16.43*	*21.14*	*18.33*	*21.77*	*22.64*	*24.94*

"-": failing to converge. "■": the best MHeteroFL method. "▨ Best B.": the best baseline. "□ Best S.C.B.": the best same-category (mutual learning-based MHeteroFL) baseline. The underscored values denote the largest accuracy improvement of `FedMRL` across 6 settings.

Comparison Baselines. We compare `FedMRL` with state-of-the-art algorithms belonging to the following three categories of MHeteroFL methods:

- `Standalone`. Each client trains its heterogeneous local model only with its local data.
- **Knowledge Distillation Without Public Data:** `FD` [21] and `FedProto` [43].
- **Model Split:** `LG-FedAvg` [27].
- **Mutual Learning:** `FML` [41], `FedKD` [45] and `FedAPEN` [37].

Evaluation Metrics. We evaluate MHeteroFL algorithms from the following three aspects:

- **Model Accuracy.** We record the test accuracy of each client's model in each round, and compute the average test accuracy.
- **Communication Cost.** We compute the number of parameters sent between the server and one client in one communication round, and record the required rounds for reaching the target average accuracy. The overall communication cost of one client for target average accuracy is the product between the cost per round and the number of rounds.
- **Computation Overhead.** We compute the computation FLOPs of one client in one communication round, and record the required communication rounds for reaching the target average accuracy. The overall computation overall for one client achieving the target average accuracy is the product between the FLOPs per round and the number of rounds.

Training Strategy. We search optimal FL hyperparameters and unique hyperparameters for all MHeteroFL algorithms. For FL hyperparameters, we test MHeteroFL algorithms with a $\{64, 128, 256, 512\}$ batch size, $\{1, 10\}$ epochs, $T = \{100, 500\}$ communication rounds and an SGD optimizer with a 0.01 learning rate. The unique hyperparameter of `FedMRL` is the representation dimension d_1 of the homogeneous global small model, we vary $d_1 = \{100,\ 150, ..., 500\}$ to obtain the best-performing `FedMRL`.

5.2 Results and Discussion

We design three FL settings with different numbers of clients (N) and client participation rates (C): ($N = 10, C = 100\%$), ($N = 50, C = 20\%$), ($N = 100, C = 10\%$) for both model-homogeneous and model-heterogeneous FL scenarios.

5.2.1 Average Test Accuracy

Table 1 and Table 3 (Appendix C.2) show that `FedMRL` consistently outperforms all baselines under both model-heterogeneous or homogeneous settings. It achieves up to a 8.48% improvement in average test accuracy compared with the best baseline under each setting. Furthermore, it achieves up to a 24.94% average test accuracy improvement than the best same-category (*i.e.*, mutual learning-based MHeteroFL) baseline under each setting. These results demonstrate the superiority of `FedMRL`

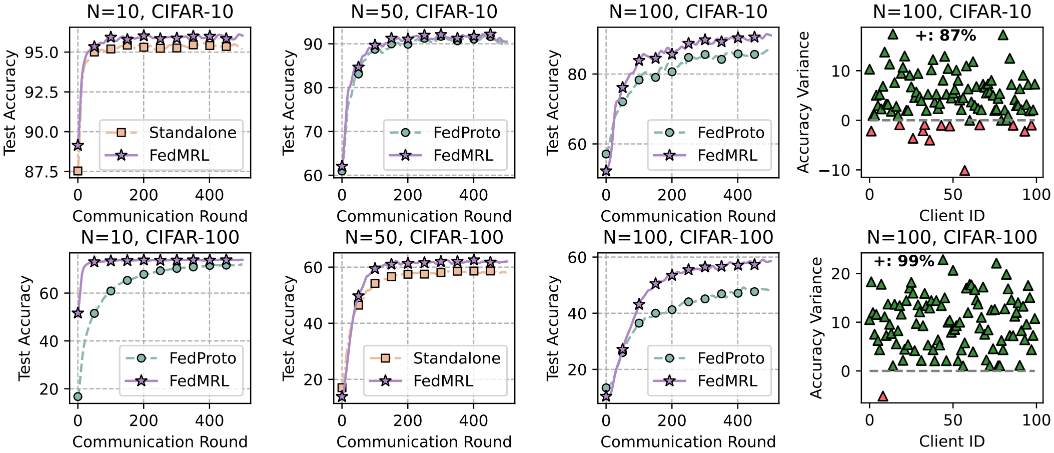


Figure 3: Left six: average test accuracy vs. communication rounds. Right two: individual clients' test accuracy (%) differences (FedMRL - FedProto).

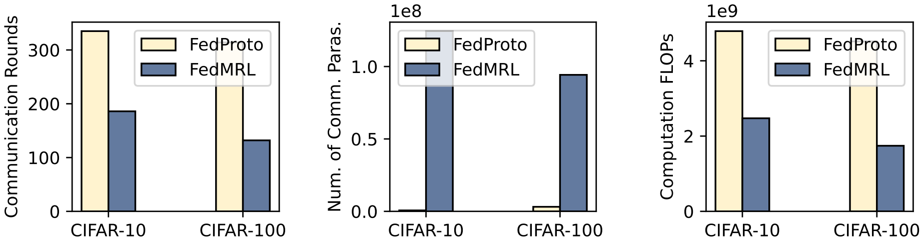


Figure 4: Communication rounds, number of communicated parameters, and computation FLOPs required to reach 90% and 50% average test accuracy targets on CIFAR-10 and CIFAR-100.

in model performance owing to its adaptive personalized representation fusion and multi-granularity representation learning capabilities. Figure 3(left six) shows that FedMRL consistently achieves faster convergence speed and higher average test accuracy than the best baseline under each setting.

5.2.2 Individual Client Test Accuracy

Figure 3(right two) shows the difference between the test accuracy achieved by FedMRL vs. the best-performing baseline FedProto (*i.e.*, FedMRL - FedProto) under ($N = 100, C = 10\%$) for each individual client. It can be observed that 87% and 99% of all clients achieve better performance under FedMRL than under FedProto on CIFAR-10 and CIFAR-100, respectively. This demonstrates that FedMRL possesses stronger personalization capability than FedProto owing to its adaptive personalized multi-granularity representation learning design.

5.2.3 Communication Cost

We record the communication rounds and the number of parameters sent per client to achieve 90% and 50% target test average accuracy on CIFAR-10 and CIFAR-100, respectively. Figure 4 (left) shows that FedMRL requires fewer rounds and achieves faster convergence than FedProto. Figure 4 (middle) shows that FedMRL incurs higher communication costs than FedProto as it transmits the full homogeneous small model, while FedProto only transmits each local seen-class average representation between the server and the client. Nevertheless, FedMRL with an optional smaller representation dimension (d_1) of the homogeneous small model still achieves higher communication efficiency than same-category mutual learning-based MHeteroFL baselines (FML, FedKD, FedAPEN) with a larger representation dimension.

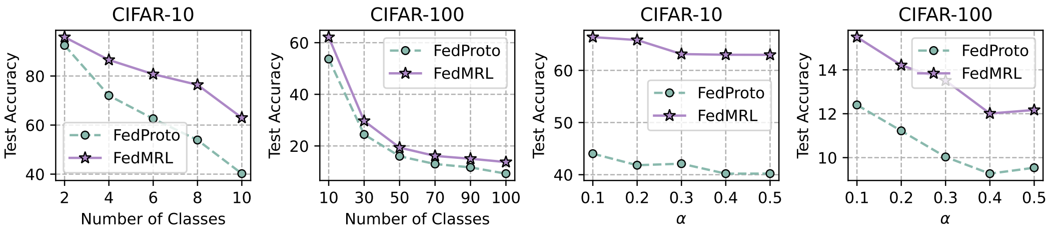


Figure 5: Robustness to non-IIDness (Class & Dirichlet).

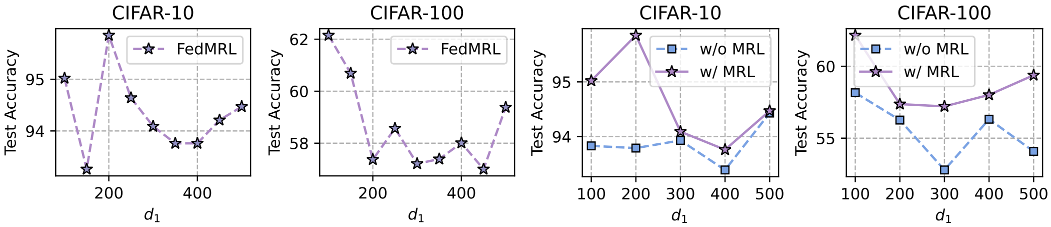


Figure 6: Left two: sensitivity analysis results. Right two: ablation study results.

5.2.4 Computation Overhead

We also calculate the computation FLOPs consumed per client to reach 90% and 50% target average test accuracy on CIFAR-10 and CIFAR-100, respectively. Figure 4(right) shows that FedMRL incurs lower computation costs than FedProto, owing to its faster convergence (*i.e.*, fewer rounds) even with higher computation overhead per round due to the need to train an additional homogeneous small model and a linear representation projector.

5.3 Case Studies

5.3.1 Robustness to Non-IIDness (Class)

We evaluate the robustness of FedMRL to different non-IIDnesses as a result of the number of classes assigned to each client under the $(N = 100, C = 10\%)$ setting. The fewer classes assigned to each client, the higher the non-IIDness. For CIFAR-10, we assign $\{2, 4, \ldots, 10\}$ classes out of total 10 classes to each client. For CIFAR-100, we assign $\{10, 30, \ldots, 100\}$ classes out of total 100 classes to each client. Figure 5(left two) shows that FedMRL consistently achieves higher average test accuracy than the best-performing baseline - FedProto on both datasets, demonstrating its robustness to non-IIDness by class.

5.3.2 Robustness to Non-IIDness (Dirichlet)

We also test the robustness of FedMRL to various non-IIDnesses controlled by α in the Dirichlet function under the $(N = 100, C = 10\%)$ setting. A smaller α indicates a higher non-IIDness. For both datasets, we vary α in the range of $\{0.1, \ldots, 0.5\}$. Figure 5(right two) shows that FedMRL significantly outperforms FedProto under all non-IIDness settings, validating its robustness to Dirichlet non-IIDness.

5.3.3 Sensitivity Analysis - d_1

FedMRL relies on a hyperparameter d_1 - the representation dimension of the homogeneous small model. To evaluate its sensitivity to d_1, we test FedMRL with $d_1 = \{100, 150, \ldots, 500\}$ under the $(N = 100, C = 10\%)$ setting. Figure 6(left two) shows that smaller d_1 values result in higher average test accuracy on both datasets. It is clear that a smaller d_1 also reduces communication and computation overheads, thereby helping FedMRL achieve the best trade-off among model performance, communication efficiency, and computational efficiency.

5.4 Ablation Study

We conduct ablation experiments to validate the usefulness of MRL. For FedMRL with MRL, the global header and the local header learn multi-granularity representations. For FedMRL without MRL, we directly input the representation fused by the representation projector into the client's local header for loss computation (*i.e.*, we do not extract Matryoshka Representations and remove the global header). Figure 6(right two) shows that FedMRL with MRL consistently outperforms FedMRL without MRL, demonstrating the effectiveness of the design to incorporate MRL into MHeteroFL. Besides, the accuracy gap between them decreases as d_1 rises. This shows that as the global and local headers learn increasingly overlapping representation information, the benefits of MRL are reduced.

6 Discussion

We discuss how FedMRL tackles heterogeneity and its privacy, communication and computation.

Tackling Heterogeneity. FedMRL allows each client to tailor its heterogeneous local model according to its system resources, which addresses system and model heterogeneity. Each client achieves multi-granularity representation learning adapting to local non-IID data distribution through a personalized heterogeneous representation projector, alleviating data heterogeneity.

Privacy. The server and clients only communicate the homogeneous small models. Since we do not limit the representation dimensions d_1, d_2 of the proxy homogeneous global model and the heterogeneous client model are the same, sharing the proxy homogeneous model does not disclose the representation dimension and structure of the heterogeneous client model. Meanwhile, local data are always stored by clients for local training, so local data privacy is also protected.

Communication Cost. The server and clients transmit homogeneous small models with fewer parameters than the client's heterogeneous local model, consuming significantly lower communication costs in one communication round compared with transmitting complete local models like FedAvg.

Computational Overhead. Besides training the heterogeneous local model, each client also trains the homogeneous global small model and a lightweight representation projector with far fewer parameters than the heterogeneous local model. The computational overhead in one round is slightly increased. Since we design personalized Matryoshka Representations learning adapting to local data distribution from multiple perspectives, the model learning capability is improved, accelerating model convergence and consuming fewer rounds. Therefore, the total computational cost is reduced.

7 Conclusion

This paper proposes a novel MHeteroFL approach - FedMRL - to jointly address data, system and model heterogeneity challenges in FL. The key design insight is the addition of a global homogeneous small model shared by FL clients for enhanced knowledge interaction among heterogeneous local models. Adaptive personalized representation fusion and multi-granularity Matryoshka Representations learning further boosts model learning capability. The client and the server only need to exchange the homogeneous small model, while the clients' heterogeneous local models and data remain unexposed, thereby enhancing the preservation of both model and data privacy. Theoretical analysis shows that FedMRL is guaranteed to converge over time. Extensive experiments demonstrate that FedMRL significantly outperforms state-of-the-art models regarding test accuracy, while incurring low communication and computation costs. [6]

Acknowledgments and Disclosure of Funding

This research is supported in part by the National Science Foundation of China under Grant 62272252 & 62272253, and the Fundamental Research Funds for the Central Universities; the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (No. AISG2-RP-2020-019); the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore; and the China Scholarship Council (CSC).

[6] Appendix D elaborates FedMRL's border impact and limitations.

References

[1] Jin-Hyun Ahn et al. Wireless federated distillation for distributed edge learning with heterogeneous data. In *Proc. PIMRC*, pages 1–6, Istanbul, Turkey, 2019. IEEE.

[2] Jin-Hyun Ahn et al. Cooperative learning VIA federated distillation OVER fading channels. In *Proc. ICASSP*, pages 8856–8860, Barcelona, Spain, 2020. IEEE.

[3] Samiul Alam et al. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. In *Proc. NeurIPS*, virtual, 2022. .

[4] Sara Babakniya et al. Revisiting sparsity hunting in federated learning: Why does sparsity consensus matter? *Transactions on Machine Learning Research*, 1(1):1, 2023.

[5] Yun-Hin Chan, Rui Zhou, Running Zhao, Zhihan JIANG, and Edith C. H. Ngai. Internal cross-layer gradients for extending homogeneity to heterogeneity in federated learning. In *Proc. ICLR*, page 1, Vienna, Austria, 2024. OpenReview.net.

[6] Hongyan Chang et al. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. In *Proc. NeurIPS Workshop*, virtual, 2021. .

[7] Jiangui Chen et al. Fedmatch: Federated learning over heterogeneous question answering data. In *Proc. CIKM*, pages 181–190, virtual, 2021. ACM.

[8] Sijie Cheng et al. Fedgems: Federated learning of larger server models via selective knowledge fusion. *CoRR*, abs/2110.11027, 2021.

[9] Yae Jee Cho et al. Heterogeneous ensemble knowledge transfer for training large models in federated learning. In *Proc. IJCAI*, pages 2881–2887, virtual, 2022. ijcai.org.

[10] Liam Collins et al. Exploiting shared representations for personalized federated learning. In *Proc. ICML*, volume 139, pages 2089–2099, virtual, 2021. PMLR.

[11] Enmao Diao. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *Proc. ICLR*, page 1, Virtual Event, Austria, 2021. OpenReview.net.

[12] Randy Goebel, Han Yu, Boi Faltings, Lixin Fan, and Zehui Xiong. *Trustworthy Federated Learning*. Springer, Cham, 2023.

[13] Xuan Gong et al. Federated learning via input-output collaborative distillation. In *Proc. AAAI*, pages 22058–22066, Vancouver, Canada, 2024. AAAI Press.

[14] Shangwei Guo, Tianwei Zhang, Guowen Xu, Han Yu, Tao Xiang, and Yang Liu. Byzantine-resilient decentralized stochastic gradient descent. *IEEE Transactions on Circuits and Systems for Video Technology (TCVT)*, 32(6):4096–4106, 2021.

[15] Chaoyang He et al. Group knowledge transfer: Federated learning of large cnns at the edge. In *Proc. NeurIPS*, virtual, 2020. .

[16] S. Horváth. FjORD: Fair and accurate federated learning under heterogeneous targets with ordered dropout. In *Proc. NIPS*, pages 12876–12889, Virtual, 2021. OpenReview.net.

[17] Wenke Huang et al. Few-shot model agnostic federated learning. In *Proc. MM*, pages 7309–7316, Lisboa, Portugal, 2022. ACM.

[18] Wenke Huang et al. Learn from others and be yourself in heterogeneous federated learning. In *Proc. CVPR*, pages 10133–10143, virtual, 2022. IEEE.

[19] Sohei Itahara et al. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Trans. Mob. Comput.*, 22(1):191–205, 2023.

[20] Jaehee Jang et al. Fedclassavg: Local representation learning for personalized federated learning on heterogeneous neural networks. In *Proc. ICPP*, pages 76:1–76:10, virtual, 2022. ACM.

[21] Eunjeong Jeong et al. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. In *Proc. NeurIPS Workshop on Machine Learning on the Phone and other Consumer Devices*, virtual, 2018. .

[22] Shivam Kalra, Junfeng Wen, Jesse C. Cresswell, Maksims Volkovs, and Hamid R. Tizhoosh. Proxyfl: Decentralized federated learning through proxy model sharing. *Nature Communications*, 14, 2023.

[23] Alex Krizhevsky et al. *Learning multiple layers of features from tiny images*. Toronto, ON, Canada, , 2009.

[24] Aditya Kusupati et al. Matryoshka representation learning. In *Proc. NeurIPS*, New Orleans, LA, USA, 2022.

[25] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. In *Proc. NeurIPS Workshop*, virtual, 2019. .

[26] Qinbin Li et al. Practical one-shot federated learning for cross-silo setting. In *Proc. IJCAI*, pages 1484–1490, virtual, 2021. ijcai.org.

[27] Paul Pu Liang et al. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 1(1), 2020.

[28] Tao Lin et al. Ensemble distillation for robust model fusion in federated learning. In *Proc. NeurIPS*, virtual, 2020. .

[29] Chang Liu et al. Completely heterogeneous federated learning. *CoRR*, abs/2210.15865, 2022.

[30] Disha Makhija et al. Architecture agnostic federated learning for neural networks. In *Proc. ICML*, volume 162, pages 14860–14870, virtual, 2022. PMLR.

[31] Koji Matsuda et al. Fedme: Federated learning via model exchange. In *Proc. SDM*, pages 459–467, Alexandria, VA, USA, 2022. SIAM.

[32] Brendan McMahan et al. Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS*, volume 54, pages 1273–1282, Fort Lauderdale, FL, USA, 2017. PMLR.

[33] Duy Phuong Nguyen et al. Enhancing heterogeneous federated learning with knowledge extraction and multi-model fusion. In *Proc. SC Workshop*, pages 36–43, Denver, CO, USA, 2023. ACM.

[34] Jaehoon Oh et al. Fedbabu: Toward enhanced representation for federated image classification. In *Proc. ICLR*, virtual, 2022. OpenReview.net.

[35] Sejun Park et al. Towards understanding ensemble distillation in federated learning. In *Proc. ICML*, volume 202, pages 27132–27187, Honolulu, Hawaii, USA, 2023. PMLR.

[36] Krishna Pillutla et al. Federated learning with partial model personalization. In *Proc. ICML*, volume 162, pages 17716–17758, virtual, 2022. PMLR.

[37] Zhen Qin et al. Fedapen: Personalized cross-silo federated learning with adaptability to statistical heterogeneity. In *Proc. KDD*, pages 1954–1964, Long Beach, CA, USA, 2023. ACM.

[38] Felix Sattler et al. Fedaux: Leveraging unlabeled auxiliary data in federated learning. *IEEE Trans. Neural Networks Learn. Syst.*, 1(1):1–13, 2021.

[39] Felix Sattler et al. CFD: communication-efficient federated distillation via soft-label quantization and delta coding. *IEEE Trans. Netw. Sci. Eng.*, 9(4):2025–2038, 2022.

[40] Aviv Shamsian et al. Personalized federated learning using hypernetworks. In *Proc. ICML*, volume 139, pages 9489–9502, virtual, 2021. PMLR.

[41] Tao Shen et al. Federated mutual learning. *CoRR*, abs/2006.16765, 2020.

[42] Alysa Ziying Tan et al. Towards personalized federated learning. *IEEE Trans. Neural Networks Learn. Syst.*, 1(1):1–17, 2022.

[43] Yue Tan et al. Fedproto: Federated prototype learning across heterogeneous clients. In *Proc. AAAI*, pages 8432–8440, virtual, 2022. AAAI Press.

[44] Jiaqi Wang et al. Towards personalized federated learning via heterogeneous model reassembly. In *Proc. NeurIPS*, page 13, New Orleans, Louisiana, USA, 2023. OpenReview.net.

[45] Chuhan Wu et al. Communication-efficient federated learning via knowledge distillation. *Nature Communications*, 13(1):2032, 2022.

[46] Qiang Yang, Lixin Fan, and Han Yu. *Federated Learning: Privacy and Incentive*. Springer, Cham, 2020.

[47] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Federated Learning*. Springer, Cham, 2020.

[48] Liping Yi, Xiaorong Shi, Nan Wang, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. pfedkt: Personalized federated learning with dual knowledge transfer. *Knowledge-Based Systems*, 292:111633, 2024.

[49] Liping Yi, Xiaorong Shi, Nan Wang, Ziyue Xu, Gang Wang, and Xiaoguang Liu. pfedlhns: Personalized federated learning via local hypernetworks. In *Proc. ICANN*, volume 1, page 516–528. Springer, 2023.

[50] Liping Yi, Xiaorong Shi, Nan Wang, Jinsong Zhang, Gang Wang, and Xiaoguang Liu. Fedpe: Adaptive model pruning-expanding for federated learning on mobile devices. *IEEE Transactions on Mobile Computing*, pages 1–18, 2024.

[51] Liping Yi, Xiaorong Shi, Wenrui Wang, Gang Wang, and Xiaoguang Liu. Fedrra: Reputation-aware robust federated learning against poisoning attacks. In *Proc. IJCNN*, pages 1–8. IEEE, 2023.

[52] Liping Yi, Gang Wang, and Xiaoguang Liu. QSFL: A two-level uplink communication optimization framework for federated learning. In *Proc. ICML*, volume 162, pages 25501–25513. PMLR, 2022.

[53] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM'23)*, page 11, Canada, 2023. ACM.

[54] Liping Yi, Gang Wang, Xiaofei Wang, and Xiaoguang Liu. Qsfl: Two-level communication-efficient federated learning on mobile edge devices. *IEEE Transactions on Services Computing*, pages 1–16, 2024.

[55] Liping Yi, Han Yu, Zhuan Shi, Gang Wang, Xiaoguang Liu, Lizhen Cui, and Xiaoxiao Li. FedSSA: Semantic Similarity-based Aggregation for Efficient Model-Heterogeneous Personalized Federated Learning. In *IJCAI*, 2024.

[56] Liping Yi, Jinsong Zhang, Rui Zhang, Jiaqi Shi, Gang Wang, and Xiaoguang Liu. Su-net: An efficient encoder-decoder model of federated learning for brain tumor segmentation. In *Proc. ICANN*, volume 12396, pages 761–773. Springer, 2020.

[57] Fuxun Yu et al. Fed2: Feature-aligned federated learning. In *Proc. KDD*, pages 2066–2074, virtual, 2021. ACM.

[58] Sixing Yu et al. Resource-aware federated learning using knowledge extraction and multi-model fusion. *CoRR*, abs/2208.07978, 2022.

[59] Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proc. AAAI*, pages 16768–16776, Vancouver, Canada, 2024. AAAI Press.

[60] Jie Zhang et al. Parameterized knowledge transfer for personalized federated learning. In *Proc. NeurIPS*, pages 10092–10104, virtual, 2021. OpenReview.net.

[61] Jie Zhang et al. Towards data-independent knowledge transfer in model-heterogeneous federated learning. *IEEE Trans. Computers*, 72(10):2888–2901, 2023.

[62] Lan Zhang et al. Fedzkt: Zero-shot knowledge transfer towards resource-constrained federated learning with heterogeneous on-device models. In *Proc. ICDCS*, pages 928–938, virtual, 2022. IEEE.

[63] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proc. NeurIPS*, pages 8792–8802, Montréal, Canada, 2018. Curran Associates Inc.

[64] Zhuangdi Zhu et al. Data-free knowledge distillation for heterogeneous federated learning. In *Proc. ICML*, volume 139, pages 12878–12889, virtual, 2021. PMLR.

[65] Zhuangdi Zhu et al. Resilient and communication efficient learning for heterogeneous federated systems. In *Proc. ICML*, volume 162, pages 27504–27526, virtual, 2022. PMLR.

A Pseudo codes of `FedMRL`

```
Algorithm 1: FedMRL
Input: $N$, total number of clients; $K$, number of selected clients in one round; $T$, total number of rounds; $\eta_\omega$, learning rate of client local heterogeneous models; $\eta_\theta$, learning rate of homogeneous small model; $\eta_\varphi$, learning rate of the representation projector.
Output: client whole models removing the global header
$[\mathcal{G}(\theta^{ex,T-1}) \circ \mathcal{F}_0(\omega_0^{T-1})|\mathcal{P}_0(\varphi_0^{T-1}), \ldots, \mathcal{G}(\theta^{ex,T-1}) \circ \mathcal{F}_{N-1}(\omega_{N-1}^{T-1})|\mathcal{P}_{N-1}(\varphi_{N-1}^{T-1})]$.

Randomly initialize the global homogeneous small model $\mathcal{G}(\theta^{\mathbf{0}})$, client local heterogeneous
  models $[\mathcal{F}_0(\omega_0^0), \ldots, \mathcal{F}_{N-1}(\omega_{N-1}^0)]$ and local heterogeneous representation projectors
  $[\mathcal{P}_0(\varphi_0^0), \ldots, \mathcal{P}_{N-1}(\varphi_{N-1}^0)]$.
for each round t=1,...,T-1 do
    // Server Side:
    $S^t \leftarrow$ Randomly sample $K$ clients from $N$ clients;
    Broadcast the global homogeneous small model $\theta^{t-1}$ to sampled $K$ clients;
    $\theta_k^t \leftarrow$ ClientUpdate($\theta^{t-1}$);
    /* Aggregate Local Homogeneous Small Models */
    $\theta^t = \sum_{k=0}^{K-1} \frac{n_k}{n} \theta_k^t$.

    // ClientUpdate:
    Receive the global homogeneous small model $\theta^{t-1}$ from the server;
    for $k \in S^t$ do
        /* Local Training with MRL */
        for $(\boldsymbol{x}_i, y_i) \in D_k$ do
            $\boldsymbol{\mathcal{R}}_i^{\mathcal{G}} = \mathcal{G}^{ex}(\boldsymbol{x}_i; \theta^{ex,t-1}), \boldsymbol{\mathcal{R}}_i^{\mathcal{F}_k} = \mathcal{F}_k^{ex}(\boldsymbol{x}_i; \omega_k^{ex,t-1})$;
            $\boldsymbol{\mathcal{R}}_i = \boldsymbol{\mathcal{R}}_i^{\mathcal{G}} \circ \boldsymbol{\mathcal{R}}_i^{\mathcal{F}_k}$;
            $\widetilde{\boldsymbol{\mathcal{R}}}_i = \mathcal{P}_k(\boldsymbol{\mathcal{R}}_i; \varphi_k^{t-1})$;
            $\widetilde{\boldsymbol{\mathcal{R}}}_i^{lc} = \widetilde{\boldsymbol{\mathcal{R}}}_i^{1:d_1}, \widetilde{\boldsymbol{\mathcal{R}}}_i^{hf} = \widetilde{\boldsymbol{\mathcal{R}}}_i^{1:d_2}$;
            $\hat{y}_i^{\mathcal{G}} = \mathcal{G}^{hd}(\widetilde{\boldsymbol{\mathcal{R}}}_i^{lc}; \theta^{hd,t-1}); \hat{y}_i^{\mathcal{F}_k} = \mathcal{F}_k^{hd}(\omega_k^{hd,t-1})$;
            $\ell_i^{\mathcal{G}} = \ell(\hat{y}_i^{\mathcal{G}}, y_i); \ell_i^{\mathcal{F}_k} = \ell(\hat{y}_i^{\mathcal{F}_k}, y_i)$;
            $\ell_i = m_i^{\mathcal{G}} \cdot \ell_i^{\mathcal{G}} + m_i^{\mathcal{F}_k} \cdot \ell_i^{\mathcal{F}_k}$;
            $\theta_k^t \leftarrow \theta^{t-1} - \eta_\theta \nabla \ell_i$;
            $\omega_k^t \leftarrow \omega_k^{t-1} - \eta_\omega \nabla \ell_i$;
            $\varphi_k^t \leftarrow \varphi_k^{t-1} - \eta_\varphi \nabla \ell_i$;
        end
        Upload updated local homogeneous small model $\theta_k^t$ to the server.
    end
end
```

B Theoretical Proofs

We first define the following additional notations. $t \in \{0, \ldots, T-1\}$ denotes the t-th round. $e \in \{0, 1, \ldots, E\}$ denotes the e-th iteration of local training. $tE + 0$ indicates that clients receive the global homogeneous small model $\mathcal{G}(\theta^t)$ from the server before the $(t+1)$-th round's local training. $tE + e$ denotes the e-th iteration of the $(t+1)$-th round's local training. $tE + E$ marks the ending of the $(t+1)$-th round's local training. After that, clients upload their updated local homogeneous small models to the server for aggregation. $\mathcal{W}_k(w_k)$ denotes the whole model trained on client k, including the global homogeneous small model $\mathcal{G}(\theta)$, the client k's local heterogeneous model $\mathcal{F}_k(\omega_k)$, and the personalized representation projector $\mathcal{P}_k(\varphi_k)$. η is the learning rate of the whole model trained on client k, including $\{\eta_\theta, \eta_\omega, \eta_{\boldsymbol{\varphi}}\}$.

Assumption 1 ***Lipschitz Smoothness***. *The gradients of client k's whole local model w_k are $L1$–Lipschitz smooth [43],*

$$\begin{gathered}\|\nabla \mathcal{L}_k^{t_1}(w_k^{t_1}; \boldsymbol{x}, y) - \nabla \mathcal{L}_k^{t_2}(w_k^{t_2}; \boldsymbol{x}, y)\| \le L_1 \|w_k^{t_1} - w_k^{t_2}\|, \\ \forall t_1, t_2 > 0, k \in \{0, 1, \ldots, N-1\}, (\boldsymbol{x}, y) \in D_k.\end{gathered} \tag{15}$$

The above formulation can be re-expressed as:

$$\mathcal{L}_k^{t_1} - \mathcal{L}_k^{t_2} \leq \langle \nabla \mathcal{L}_k^{t_2}, (w_k^{t_1} - w_k^{t_2}) \rangle + \frac{L_1}{2} \| w_k^{t_1} - w_k^{t_2} \|_2^2. \tag{16}$$

Assumption 2 ***Unbiased Gradient and Bounded Variance***. *Client k's random gradient $g_{w,k}^t = \nabla \mathcal{L}_k^t(w_k^t; \mathcal{B}_k^t)$ ($\mathcal{B}$ is a batch of local data) is unbiased,*

$$\mathbb{E}_{\mathcal{B}_k^t \subseteq D_k}[g_{w,k}^t] = \nabla \mathcal{L}_k^t(w_k^t), \tag{17}$$

and the variance of random gradient $g_{w,k}^t$ is bounded by:

$$\mathbb{E}_{\mathcal{B}_k^t \subseteq D_k}[\| \nabla \mathcal{L}_k^t(w_k^t; \mathcal{B}_k^t) - \nabla \mathcal{L}_k^t(w_k^t) \|_2^2] \leq \sigma^2. \tag{18}$$

Assumption 3 ***Bounded Parameter Variation***. *The parameter variations of the homogeneous small model θ_k^t and θ^t before and after aggregation at the FL server are bounded by:*

$$\| \theta^t - \theta_k^t \|_2^2 \leq \delta^2. \tag{19}$$

B.1 Proof of Lemma 1

Proof 1 *An arbitrary client k's local whole model w can be updated by $w_{t+1} = w_t - \eta g_{w,t}$ in the (t+1)-th round, and following Assumption 1, we can obtain*

$$\begin{aligned} \mathcal{L}_{tE+1} &\leq \mathcal{L}_{tE+0} + \langle \nabla \mathcal{L}_{tE+0}, (w_{tE+1} - w_{tE+0}) \rangle + \frac{L_1}{2} \| w_{tE+1} - w_{tE+0} \|_2^2 \\ &= \mathcal{L}_{tE+0} - \eta \langle \nabla \mathcal{L}_{tE+0}, g_{w,tE+0} \rangle + \frac{L_1 \eta^2}{2} \| g_{w,tE+0} \|_2^2. \end{aligned} \tag{20}$$

Taking the expectation of both sides of the inequality concerning the random variable ξ_{tE+0},

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{tE+1}] &\leq \mathcal{L}_{tE+0} - \eta \mathbb{E}[\langle \nabla \mathcal{L}_{tE+0}, g_{w,tE+0} \rangle] + \frac{L_1 \eta^2}{2} \mathbb{E}[\| g_{w,tE+0} \|_2^2] \\ &\overset{(a)}{=} \mathcal{L}_{tE+0} - \eta \| \nabla \mathcal{L}_{tE+0} \|_2^2 + \frac{L_1 \eta^2}{2} \mathbb{E}[\| g_{w,tE+0} \|_2^2] \\ &\overset{(b)}{\leq} \mathcal{L}_{tE+0} - \eta \| \nabla \mathcal{L}_{tE+0} \|_2^2 + \frac{L_1 \eta^2}{2} (\mathbb{E}[\| g_{w,tE+0} \|]_2^2 + \mathrm{Var}(g_{w,tE+0})) \\ &\overset{(c)}{=} \mathcal{L}_{tE+0} - \eta \| \nabla \mathcal{L}_{tE+0} \|_2^2 + \frac{L_1 \eta^2}{2} (\| \nabla \mathcal{L}_{tE+0} \|_2^2 + \mathrm{Var}(g_{w,tE+0})) \\ &\overset{(d)}{\leq} \mathcal{L}_{tE+0} - \eta \| \nabla \mathcal{L}_{tE+0} \|_2^2 + \frac{L_1 \eta^2}{2} (\| \nabla \mathcal{L}_{tE+0} \|_2^2 + \sigma^2) \\ &= \mathcal{L}_{tE+0} + (\frac{L_1 \eta^2}{2} - \eta) \| \nabla \mathcal{L}_{tE+0} \|_2^2 + \frac{L_1 \eta^2 \sigma^2}{2}. \end{aligned} \tag{21}$$

(a), (c), (d) follow Assumption 2 and (b) follows $Var(x) = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$.

Taking the expectation of both sides of the inequality for the model w over E iterations, we obtain

$$\mathbb{E}[\mathcal{L}_{tE+1}] \leq \mathcal{L}_{tE+0} + (\frac{L_1 \eta^2}{2} - \eta) \sum_{e=1}^{E} \| \nabla \mathcal{L}_{tE+e} \|_2^2 + \frac{L_1 E \eta^2 \sigma^2}{2}. \tag{22}$$

B.2 Proof of Lemma 2

Proof 2

$$\begin{aligned} \mathcal{L}_{(t+1)E+0} &= \mathcal{L}_{(t+1)E} + \mathcal{L}_{(t+1)E+0} - \mathcal{L}_{(t+1)E} \\ &\overset{(a)}{\approx} \mathcal{L}_{(t+1)E} + \eta \| \theta_{(t+1)E+0} - \theta_{(t+1)E} \|_2^2 \\ &\overset{(b)}{\leq} \mathcal{L}_{(t+1)E} + \eta \delta^2. \end{aligned} \tag{23}$$

(a): we can use the gradient of parameter variations to approximate the loss variations, i.e., $\Delta\mathcal{L} \approx \eta \cdot \|\Delta\theta\|_2^2$. *(b) follows Assumption 3.*

Taking the expectation of both sides of the inequality to the random variable ξ, we obtain

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \le \mathbb{E}[\mathcal{L}_{(t+1)E}] + \eta\delta^2. \tag{24}$$

B.3 Proof of Theorem 1

Proof 3 *Substituting Lemma 1 into the right side of Lemma 2's inequality, we obtain*

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \le \mathcal{L}_{tE+0} + (\frac{L_1\eta^2}{2} - \eta)\sum_{e=0}^{E}\|\nabla\mathcal{L}_{tE+e}\|_2^2 + \frac{L_1 E\eta^2\sigma^2}{2} + \eta\delta^2. \tag{25}$$

B.4 Proof of Theorem 2

Proof 4 *Interchanging the left and right sides of Eq. (25), we obtain*

$$\sum_{e=0}^{E}\|\nabla\mathcal{L}_{tE+e}\|_2^2 \le \frac{\mathcal{L}_{tE+0} - \mathbb{E}[\mathcal{L}_{(t+1)E+0}] + \frac{L_1 E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}}. \tag{26}$$

Taking the expectation of both sides of the inequality over rounds $t = [0, T-1]$ to w, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_{tE+e}\|_2^2 \le \frac{\frac{1}{T}\sum_{t=0}^{T-1}[\mathcal{L}_{tE+0} - \mathbb{E}[\mathcal{L}_{(t+1)E+0}]] + \frac{L_1 E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}}. \tag{27}$$

Let $\Delta = \mathcal{L}_{t=0} - \mathcal{L}^* > 0$, then $\sum_{t=0}^{T-1}[\mathcal{L}_{tE+0} - \mathbb{E}[\mathcal{L}_{(t+1)E+0}]] \le \Delta$, we can get

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_{tE+e}\|_2^2 \le \frac{\frac{\Delta}{T} + \frac{L_1 E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}}. \tag{28}$$

If the above equation converges to a constant ϵ, i.e.*,*

$$\frac{\frac{\Delta}{T} + \frac{L_1 E\eta^2\sigma^2}{2} + \eta\delta^2}{\eta - \frac{L_1\eta^2}{2}} < \epsilon, \tag{29}$$

then

$$T > \frac{\Delta}{\epsilon(\eta - \frac{L_1\eta^2}{2}) - \frac{L_1 E\eta^2\sigma^2}{2} - \eta\delta^2}. \tag{30}$$

Since $T > 0, \Delta > 0$, we can get

$$\epsilon(\eta - \frac{L_1\eta^2}{2}) - \frac{L_1 E\eta^2\sigma^2}{2} - \eta\delta^2 > 0. \tag{31}$$

Solving the above inequality yields

$$\eta < \frac{2(\epsilon - \delta^2)}{L_1(\epsilon + E\sigma^2)}. \tag{32}$$

For $\epsilon - \delta^2$, Assumption 3 assumes that the parameter variations of the homogeneous small model θ_k^t and θ^t before and after aggregation are bounded by $|\theta^t - \theta_k^t|_2^2 \le \delta^2$. $\theta_k^t = \theta^{t-1} - \eta\sum_{e=0}^{E-1} g_{\theta^{t-1}}$, so $|\theta^t - \theta_k^t|_2^2 = |\theta^t - \theta^{t-1} + \eta\sum_{e=0}^{E-1} g_{\theta^{t-1}}|_2^2 \approx \eta^2\sum_{e=0}^{E-1}|g_{\theta^{t-1}}|_2^2$, considering that the global homogeneous small models during two consecutive rounds have relatively small variations compared with parameter variations between the local and global homogeneous model. Eq. (28) and (29) define ϵ as the upper bound of the average gradient of the local training whole model (including homogeneous small model, heterogeneous client model and the local representation projector) *during T rounds and E epochs per round, i.e.,* $\frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}|\mathcal{L}_{tE+e}|_2^2 < \epsilon$*, we can simplify it to* $\sum_{e=0}^{E-1}|\mathcal{L}_{tE+e}|_2^2 < \epsilon$*. Since the homogeneous model* θ *is only one part of the local training whole model, so* $\epsilon > \sum_{e=0}^{E-1}|\mathcal{L}_{tE+e}|_2^2 > \sum_{e=0}^{E-1}|g_{\theta^{t-1}}|_2^2$*. Since we use the leaning rate* $\eta \in (0,1)$*,* $\eta^2 \in (0,1)$*, so* $\epsilon > \sum_{e=0}^{E-1}|\mathcal{L}_{tE+e}|_2^2 > \sum_{e=0}^{E-1}|g_{\theta^{t-1}}|_2^2 > \eta^2\sum_{e=0}^{E-1}|g_{\theta^{t-1}}|_2^2$*. Since* δ^2 *is the upper bound of* $\eta^2\sum_{e=0}^{E-1}|g_{\theta}^{t-1}|_2^2$*, so* $\epsilon > \delta^2$ *and* $\epsilon - \delta^2 > 0$*.*

Since L_1*,* ϵ*,* σ^2*,* $\epsilon - \delta^2$ *are all constants greater than 0,* η *has solutions. Therefore, when the learning rate* $\eta = \{\eta_\theta, \eta_\omega, \eta_\varphi\}$ *satisfies the above condition, any client's local whole model can converge. Since all terms on the right side of Eq. (28) except for* $1/T$ *are constants, hence* `FedMRL`*'s non-convex convergence rate is* $\epsilon \sim \mathcal{O}(1/T)$*.*

C More Experimental Details

Here, we provide more experimental details of used model structures, more experimental results of model-homogeneous FL scenarios, and also the experimental evidence of inference model selection.

C.1 Model Structures

Table 2 shows the structures of models used in experiments.

Table 2: Structures of 5 heterogeneous CNN models.

Layer Name	CNN-1	CNN-2	CNN-3	CNN-4	CNN-5
Conv1	5×5, 16	5×5, 16	5×5, 16	5×5, 16	5×5, 16
Maxpool1	2×2	2×2	2×2	2×2	2×2
Conv2	5×5, 32	5×5, 16	5×5, 32	5×5, 32	5×5, 32
Maxpool2	2×2	2×2	2×2	2×2	2×2
FC1	2000	2000	1000	800	500
FC2	500	500	500	500	500
FC3	10/100	10/100	10/100	10/100	10/100
model size	10.00 MB	6.92 MB	5.04 MB	3.81 MB	2.55 MB

Note: 5×5 denotes kernel size. 16 or 32 are filters in convolutional layers.

C.2 Homogeneous FL Results

Table 3 presents the results of `FedMRL` and baselines in model-homogeneous FL scenarios.

Table 3: Average test accuracy (%) in model-homogeneous FL.

FL Setting	N=10, C=100%		N=50, C=20%		N=100, C=10%	
Method	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Standalone	96.35	74.32	95.25	62.38	92.58	54.93
LG-FedAvg [27]	96.47	73.43	94.20	61.77	90.25	46.64
FD [21]	96.30	-	-	-	-	-
FedProto [43]	95.83	72.79	95.10	62.55	91.19	54.01
FML [41]	94.83	70.02	93.18	57.56	87.93	46.20
FedKD [45]	94.77	70.04	92.93	57.56	90.23	50.99
FedAPEN [37]	95.38	71.48	93.31	57.62	87.97	46.85
`FedMRL`	**96.71**	**74.52**	**95.76**	**66.46**	**95.52**	**60.64**
`FedMRL`-*Best B.*	*0.24*	*0.20*	*0.51*	*3.91*	*2.94*	*5.71*
`FedMRL`-*Best S.C.B.*	*1.33*	*3.04*	*2.45*	*8.84*	*5.29*	*9.65*

"-": failing to converge. "■": the best MHeteroFL method. "▨ Best B.": the best baseline. "□ Best S.C.B.": the best same-category (mutual learning-based MHeteroFL) baseline. The underscored values denote the largest accuracy improvement of `FedMRL` across 6 settings.

C.3 Inference Model Comparison

There are 4 alternative models for model inference in `FedMRL`: (1) mix-small (the combination of the homogeneous small model, the client heterogeneous model's feature extractor, and the representation projector, *i.e.*, removing the local header), (2) mix-large (the combination of the homogeneous small model's feature extractor, the client heterogeneous model, and the representation projector, *i.e.*, removing the global header), (3) single-small (the homogeneous small model), (4) single-large (the client heterogeneous model). We compare their model performances under $(N = 100, C = 10\%)$ settings. Figure 7 presents that mix-small has a similar accuracy to mix-large which is used as the default inference model, and they significantly outperform the single homogeneous small model and the single heterogeneous client model. Therefore, users can choose mix-small or mix-large for model inference based on their inference costs in practical applications.

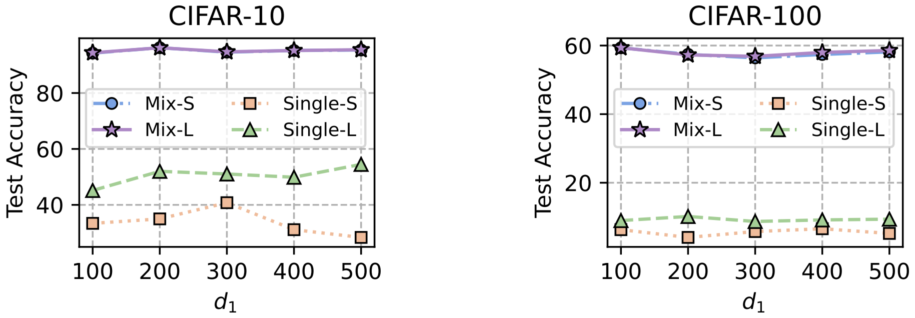


Figure 7: Accuracy of four optional inference models: mix-small (the whole model without the local header), mix-large (the whole model without the global header), single-small (the homogeneous small model), single-large (the client heterogeneous model).

D Broader Impacts and Limitations

Broader Impacts. `FedMRL` improves model performance, communication and computational efficiency for heterogeneous federated learning while effectively protecting the privacy of the client heterogeneous local model and non-IID data. It can be applied in various practical FL applications.

Limitations. The multi-granularity embedded representations within Matryoshka Representations are processed by the global small model's header and the local client model's header, respectively. This increases the storage cost, communication costs and training overhead for the global header even though it only involves one linear layer. In future work, we will follow the more effective Matryoshka Representation learning method (MRL-E) [24], removing the global header and only using the local model header to process multi-granularity Matryoshka Representations simultaneously, to enable a better trade-off among model performance and costs of storage, communication and computation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: As illustrated in the abstract and Section 1, we propose a novel model heterogeneous federated learning approach (FedMRL) with two core designs: (1) adaptive personalized representation fusion and (2) multi-granularity representation learning. Theoretical analysis and experiments demonstrate its effectiveness.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: As described in Section D.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: As shown in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The complete workflow of `FedMRL` is described in Alg. 1. The experimental setups are given in Section 5.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
    (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
    (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
    (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
    (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data are given in Section 5.1. The codes can be accessible from the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: As shown in Section 5.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We conduct 3 trails for each experimental setting and report the average results of these 3 trails.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As shown in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully read the NeurIPS Code of Ethics, and our paper satisfies it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: As shown in Section D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: As shown in Section 5.1.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The codes of `FedMRL` are given in the supplemental materials.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.