Model-Heterogeneous Prototypical Federated Learning Over the Air

Chuhan Sun*†, Zihan Chen‡, Liyinglan Liu§, Tony Q. S. Quek‡, Howard H. Yang*†
*Zhejiang University/University of Illinois Institute, Zhejiang University, Haining, China
†National Mobile Communications Research Laboratory, Southeast University, Nanjing, China
‡ISTD Pillar, Singapore University of Technology and Design, Singapore
§National Key Laboratory of Wireless Communications,
University of Electronic Science and Technology of China, Chengdu, China

***Abstract*—Over-the-air federated learning (OTA FL) provides a joint computation and communication approach to design FL systems with improved efficiency. By leveraging the superposition property of wireless channels, OTA FL enables the automatic aggregation of intermediate parameters–such as gradients–across a large number of clients, significantly reducing communication overhead while concurrently enhancing transmission privacy. However, gradient aggregation requires all clients to use identical model architectures, a condition often impractical in real-world scenarios due to variations in client hardware and computational capabilities. This mismatch can lead to scalability issues and system incompatibilities. To address this challenge, we propose a model-agnostic method based on model prototypes that enables collaborative training across clients with heterogeneous models. The proposed method bypasses the requirements of the conventional model weight/gradient updates with prototype vector aggregation, without requiring the model structures of all clients to be identical. To the best of our knowledge, the proposed method is the first to explore the prototypical model-heterogeneous OTA FL with desirable training performance and extremely low communication cost. We conducted extensive experiments to verify the efficacy of the proposed method. The results show that our approach not only significantly reduces communication overhead but also exploits the superior capabilities of large models to enhance the performance of smaller models.**

***Index Terms*—Federated Learning, over-the-air computation, model heterogeneous, prototypical learning**

I. Introduction

Federated learning (FL) [1]–[3] allows multiple edge devices to collaboratively train a global model without exposing their local private data. It stands as a critical component toward network intelligence by fostering cooperation across edge clients, enabling superior machine learning performance while safeguarding the privacy of their datasets. However, a typical FL training process requires frequent exchange of model gradient information between the clients and the server, which generates a lot of communication overhead. Meanwhile, aggregating a large number of parameters demands considerable computing resources. In addition, although FL does not directly share user local data, the exchange of model parameters still poses a potential risk to user privacy.

To address these issues, a line of recent studies [4]–[6] proposed incorporating over-the-air (OTA) computation into the FL system. Through this approach, all clients modulate their local gradients onto a set of common waveforms and simultaneously transmit the analog signals to the edge server. The superposition property of wireless channels enables automatic aggregation of the gradients at the server, significantly improving spectrum utilization, reducing computational overhead, and enhancing client privacy [7]. Despite these advantages, deploying OTA FL in practical systems encounters new challenges, one of which is heterogeneity.

Heterogeneity can be broadly divided into two categories: data heterogeneity and model heterogeneity. Data heterogeneity refers to the difference in data distribution between different clients, which gives rise to client label imbalance issues. Numerous methods [8]–[10] have been developed to address such challenges. In contrast, model heterogeneity has received far less attention, especially in the context of OTA FL. In particular, model heterogeneity refers to the fact that different clients could maintain different model architectures. Specifically, in real-world scenarios, the client devices usually have various system resources, such as computing power and memory [11]–[13]. As such, the resource-limited devices are only capable of maintaining lightweight models, while others with abundant resources can handle more advanced models. Under this circumstance, lower-end devices may not be able to support the global model (if that is too large), leaving them unable to participate in FL. Meanwhile, traditional OTA FL relies on the superposition properties of wireless channels to achieve automatic aggregation, requiring all clients to have identical model architectures. If we group clients for separate FL processes based on their device's resource capabilities, this may lead to unfairness, underutilization of data resources, and potentially degraded overall performance. If we aim to include all clients in the same FL process, we may be forced to use a model that the lowest-performing device can support. This would not only throttle model performance but also waste the

The work of C. Sun and H. H. Yang was supported in part by the National Key R&D Program of China under Grant 2024YFE0200700, in part by the open research fund of National Mobile Communications Research Laboratory, and in part by the National Natural Science Foundation of China under Grant 62201504. The work of Z. Chen and T. Q. S. Quek has been supported in part by the SNS JU project 6G-GOALS under the EU's Horizon program Grant Agreement No. 101139232. (Corresponding Author: Howard H. Yang)

ISBN 978-3-903176-73-7 © 2025 IFIP

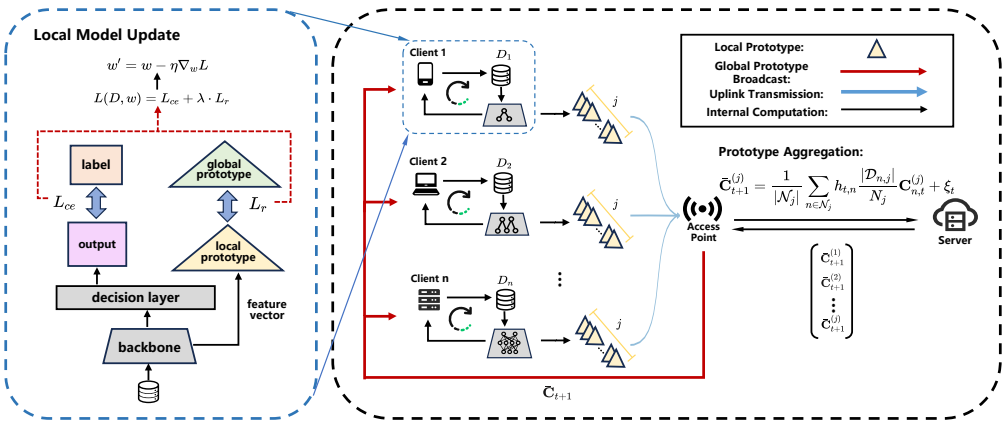


Fig. 1: An overview of the prototypical OTA-FL framework for model-heterogeneous FL system.

resources of higher-end clients. Additionally, the number of orthogonal waveforms for OTA model aggregation is indeed limited and far fewer than the number of weight parameters in deep learning models, necessitating further compression of the global model. To that end, assuming we already have a group of high-end clients training a large model, how can we leverage their superior capabilities to improve the performance of smaller models trained on low-end devices? More generally, *how can we enable learning across clients with heterogeneous model architectures in an OTA FL system?*

In this work, we leverage the model-agnostic prototypes to achieve efficient federated edge learning in model-heterogeneous OTA FL systems. Specifically, prototype vectors [14], [15] referred to the embedding vectors that capture the characteristics of data classes. The prototype can serve as feature information irrespective of the model architecture. Therefore, as long as clients are engaged in the same training task, information can still be exchanged between them through transmitting and aggregating prototype vectors. This approach allows us to implement OTA FL efficiently, with the benefit that clients with smaller models achieve improved performance. Additionally, unlike model gradients, which often involve millions of parameters, the prototype parameters are minimal, significantly reducing communication overhead.

The principal contributions of this paper are summarized as follows:

- We propose a prototypical OTA FL framework to enable collaborative training in model-heterogeneous FL systems, drastically improving OTA FL systems' scalability.
- Our proposed method only relies on the transmission and aggregation of a lightweight prototype vector, significantly reducing communication overhead compared to conventional OTA FL, which is based on identical model architectures and full model updates.
- We conducted extensive experiments with respect to the generalization and communication overhead, and the results demonstrate the efficacy of our proposed method and its effectiveness in reducing communication costs.

II. Conventional Over-the-Air Federated Learning

A. System Configuration

Consider an FL system consisting of an edge server and N clients. The clients communicate with the server through wireless transmissions over a shared spectrum. Every client $n \in \{1, \ldots, N\}$ has its local dataset $\mathcal{D}_n = \{(x_i \in \mathbb{R}^u, y_i \in \mathbb{R})\}_{i=1}^{m_n}$ with size $|\mathcal{D}_n| = m_n$, in which x_i denotes the data sample and y_i denote its respective label. The goal is to train a machine learning model $\boldsymbol{w}$ from the dataset $\mathcal{D}_n$ without sharing the clients' raw data. Formally, the task is accomplished by collaboratively solving the following optimization problem:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{w}) = \frac{1}{N} \sum_{n=1}^{N} F_n(\boldsymbol{w}). \tag{1}$$

The term $F_n : \mathbb{R}^d \to \mathbb{R}$ represents the local empirical risk of agent n, constructed from its own dataset $\mathcal{D}_n$, and $\boldsymbol{w} \in \mathbb{R}^d$ is the global model parameter with dimension d. The optimal solution to (1) is known as the empirical risk minimizer, denoted by

$$\boldsymbol{w}^* = \arg \min_{\boldsymbol{w} \in \mathbb{R}^d} \mathcal{L}(\boldsymbol{w}). \tag{2}$$

B. Federated Model Training Over the Air

In communication round t, the edge server broadcast the current global model parameters $\boldsymbol{w}_t$ to all clients. Upon receiving the latest global model, each client n uses it to calculate the local gradient $\nabla F_n(\boldsymbol{w}_t)$. The client modulates the gradient vector entry by entry to the amplitude of a set of mutually orthogonal common waveforms, composing an analog signal and sending it to the edge server through the wireless channel simultaneously. The analog signal will be automatically aggregated, which shall be affected by the channel fading $h_{t,n}$ and noise $\boldsymbol{\xi}(t)$ during the transmission. We assume that channel fading and noise are modeled as random variables that are independent and identically distributed. The server processes the received signal through a set of matched filters obtain the global gradient $\boldsymbol{g}_t$, which can be given by:

$$\boldsymbol{g}_t = \frac{1}{N}\sum_{n=1}^{N} h_{t,n}\nabla F_n\left(\boldsymbol{w}_t\right) + \boldsymbol{\xi}_t, \tag{3}$$

in which $h_{t,n}$ denotes the channel fading of client n in the t th communication round and $\boldsymbol{\xi}_t$ represents additive noise following a Gaussian distribution with variance σ_ξ^2 [7].

Then, the server updates the global parameter as

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t, \tag{4}$$

where η is the learning rate. After that, the server broadcasts the global parameter to all clients for the next round of local computation. This process iterates continuously until the global model converges.

C. Limitations of conventional OTA FL methods

In conventional OTA FL approaches, the architectures (and hence dimensions) of the local models must be precisely the same to achieve the automatic aggregation of signals. When considering system heterogeneity in FL, different edge devices may have diverse hardware resources and are capable of maintaining models with diverse sizes for training and inference. This inconsistency in model architectures across clients hinders the application of OTA computation to achieve fast aggregation of local model weights (by leveraging the superposition properties of channels). For example, consider a model-heterogeneous scenario with the coexistence of ResNet-18 and LeNet. ResNet-18 [16] has 18 layers with approximately tens of millions of parameters, represented by $\boldsymbol{w}_1 \in \mathbb{R}^{d_1}$. In contrast, the LeNet model has hundreds of thousands of parameters, represented by $\boldsymbol{w}_2 \in \mathbb{R}^{d_2}$. The summation steps in (3) cannot be completed because d_1 is not equal to d_2. In addition, for a limited number of clients equipped with small models such as LeNet for training, due to the simple model structure, the training results may fail to achieve satisfactory performance.

III. Proposed Method

In this section, we detail our proposed prototypical OTA-FL framework, designed for model heterogeneity scenarios. In particular, we introduce the concept of the prototype vector, highlighting its model-agnostic property for bridging diverse models and the advantages of reducing communication costs. An overview of the framework is given in Fig. 1.

A. Prototype

Neural network models, as the most widely-used machine learning models, can be split into two parts:

$$\mathcal{F}_n(\boldsymbol{w}_n) = g_n(\boldsymbol{\nu}_n) \circ f_n(\boldsymbol{\phi}_n), \tag{5}$$

where $\boldsymbol{w}_n = (\boldsymbol{\phi}_n, \boldsymbol{\nu}_n)$ represents the parameters of the model, $f_n(\boldsymbol{\phi}_n)$ and $g_n(\boldsymbol{\nu}_n)$ denote the representation layers and decision layers, respectively.

Note that the representation layers extract features from the samples fed into the model, transforming the input from the original feature space into the embedding space. The decision layers then use the embedding vectors to classify the samples according to the given learning task [17]. For the n-th client, the embedding vector of an input x is denoted by $\boldsymbol{z}_n = f_n(\boldsymbol{\phi}_n; x)$, parameterized by $\boldsymbol{\phi}_n$.

We denote by $\boldsymbol{C}^{(j)}$ the prototype that corresponds to the j-th class in the dataset label. For the n-th client, its prototype is defined as the mean value of the embedding vectors of all the instances in class j [18], i.e.,

$$\boldsymbol{C}_n^{(j)} = \frac{1}{|\mathcal{D}_{n,j}|}\sum_{(x,y)\in\mathcal{D}_{n,j}} f_n(\boldsymbol{\phi}_n; x), \tag{6}$$

where $\mathcal{D}_{n,j}$ represents the set of all instances in dataset $\mathcal{D}_n$ with labels belonging to class j.

In this paper, we employ the prototypes as a proxy for each class and aggregate the prototypes (instead of the entire gradients) within the OTA FL setting. The dimension of prototype $\boldsymbol{C}_n^{(j)}$ is unified to k.

Algorithm 1: Model-heterogeneous prototypical OTA-FL framework

Input: Initial global prototype $\bar{\boldsymbol{C}}_0$, initial local model $\boldsymbol{w}_{0,n}, n \in \{1,\ldots,N\}$, weight of local prototype q_n
for *each round* $t = 0, 1, 2, \ldots, T-1$ **do**
  for *each client* $n \in N$ ***in parallel*** **do**
    # Train model locally and upload local prototypes
    $\boldsymbol{C}_{n,t} \leftarrow \text{CLIENTUPDATE}(t, \bar{\boldsymbol{C}}_t)$
  # Global aggregation
  $\bar{\boldsymbol{C}}_{t+1} = \frac{1}{N}\sum_{n=1}^{N} h_{t,n} q_n \boldsymbol{C}_{n,t} + \boldsymbol{\xi}_t$
  Broadcasting $\bar{\boldsymbol{C}}_{t+1}$ to clients

Function `ClientUpdate`$(t, \bar{\boldsymbol{C}}_t)$**:**
  for *each local epoch* **do**
    for *each class* j **do**
      # Compute every class prototype
      $\boldsymbol{C}_n^{(j)} = \frac{1}{|\mathcal{D}_{n,j}|}\sum_{(x,y)\in\mathcal{D}_{n,j}} f_n(\boldsymbol{\phi}_n; x)$
    Combine $\boldsymbol{C}_n^{(j)}$ to obtain local prototype $\boldsymbol{C}_{n,t}$
    $L_{ce} \leftarrow CrossEntropyLoss$
    $L_r \leftarrow d(\boldsymbol{C}_{n,t}, \bar{\boldsymbol{C}}_t)$
    $L = L_{ce} + \lambda \cdot L_r$
    # Train local model according to the loss
    $\boldsymbol{w}_n^{t+1} = \boldsymbol{w}_n^t - \eta \nabla L$
  return $\boldsymbol{C}_{n,t}$

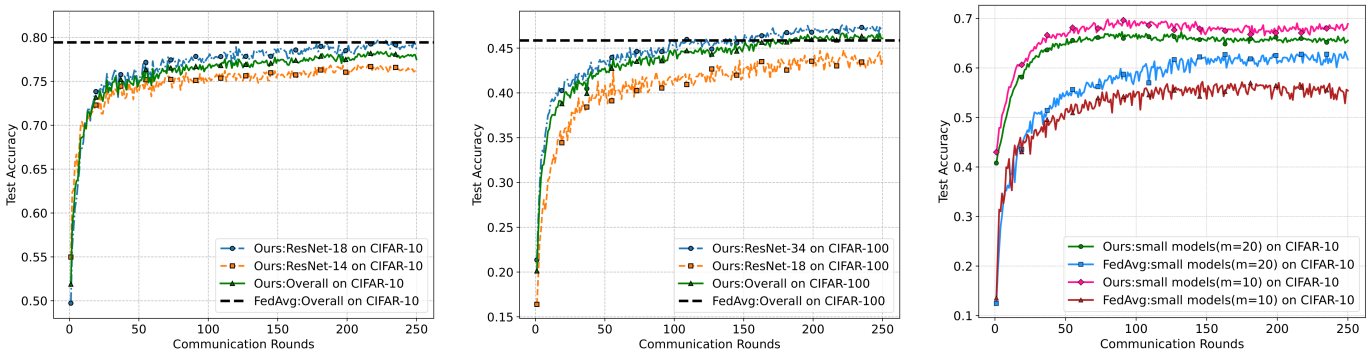
B. Local Model Update

Since the globally aggregated parameters become the prototype, each client n needs to adopt a different local loss function [18], as follows:

$$L(\mathcal{D}_n, \boldsymbol{w}_n) = L_{ce}(\mathcal{F}_n(\boldsymbol{w}_n; x), y) + \lambda \cdot L_r, \tag{7}$$

where L_{ce} denotes the typical cross-entropy loss for supervised learning, λ is the regularization coefficient, and L_r is the regularization term, given by

$$L_r = d(\boldsymbol{C}_n, \bar{\boldsymbol{C}}) = \sum_j \|\boldsymbol{C}_n^{(j)} - \bar{\boldsymbol{C}}^{(j)}\|_2^2, \tag{8}$$



(a) (b) (c)

Fig. 2: Performance comparison between our method and FedAvg, (a) training ResNet-18 and ResNet-14 on non-IID CIFAR-10 with 10 clients for each group, (b) training ResNet-34 and ResNet-18 on non-IID CIFAR-100 with 15 and 5 clients respectively, (c) training ResNet-18 and LeNet on non-IID CIFAR-10, with a total of 70 clients. Here, m refers to the number of clients in the LeNet group, and the test accuracy specifically refers to the accuracy of the small models group.

Here, L_r represents the distance d between local prototypes $\boldsymbol{C}_n$ and the global prototypes $\bar{\boldsymbol{C}}$, calculated using the squared Euclidean distance. Other divergence calculation metrics are also applicable. The added regularization term enhances the consistency between the global prototype and local prototypes, helping local models fuse information from other clients, thereby improving the model's generalization ability. We adopt stochastic gradient descent (SGD) for local training, while other optimizers can also be used.

C. Analog Global Prototype Aggregation

Similar to (3), the analog OTA global prototype aggregation can be expressed as:

$$\bar{\boldsymbol{C}}_{t+1}^{(j)} = \frac{1}{|\mathcal{N}_j|} \sum_{n \in \mathcal{N}_j} h_{t,n} \frac{|\mathcal{D}_{n,j}|}{N_j} \boldsymbol{C}_{n,t}^{(j)} + \boldsymbol{\xi}_t, \tag{9}$$

where $\boldsymbol{C}_{n,t}^{(j)}$ is the prototype of client n corresponding to class j at the t th communication round, $\mathcal{N}_j$ denotes the set of clients whose local dataset contains class j, and N_j stands for the total number of instances of class j across all clients.

In the aggregation, the proportion of class j samples in client n's local training dataset relative to the total number of class j samples across all clients will be used as the weight for the local prototype $\boldsymbol{C}_n^{(j)}$. To simplify the notation, we denote this weight by q_n, where each element of q_n is defined as $|\mathcal{D}_{n,j}|/N_j$.

Remark 1. *In traditional OTA FL, different models cannot be automatically aggregated if they do not adhere to the same architecture. Prototypes circumvent this issue by serving as feature vectors with unified dimensions, enabling clients with different models to train together. Additionally, the dimensionality of prototypes is significantly smaller than that of gradients, greatly reducing communication overhead.*[1]

[1]Taking the task of training CIFAR-10 with 10 clients each for ResNet-18 and ResNet-14 as an example, our proposed algorithm only requires 400 thousand parameter updates per communication round, while conventional method demands 217.4 million parameters per round.

TABLE I: The model architectures used in our experiments and the corresponding model size.

Model	Model structure	Number of parameters
LeNet	5 layers	0.32 million
ResNet-14	14 layers with 3 residual blocks	9.5 million
ResNet-18	18 layers with 4 residual blocks	12 million
ResNet-34	34 layers with 4 residual blocks	22 million

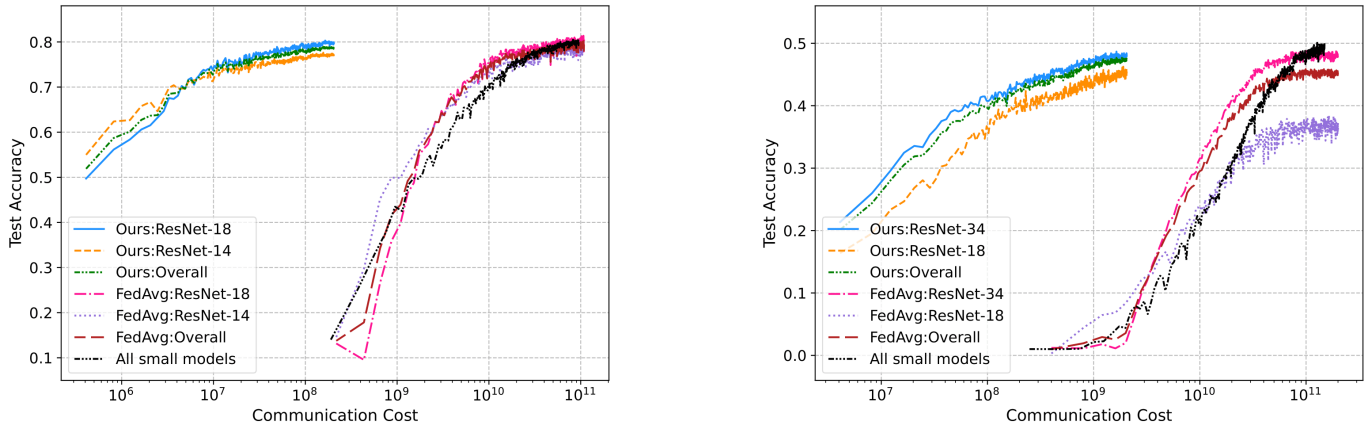
To this end, we achieve OTA FL training among heterogeneous models through prototype aggregation. Algorithm 1 provides a summary of the details of this method.

IV. Experiments

A. Experiment Setup

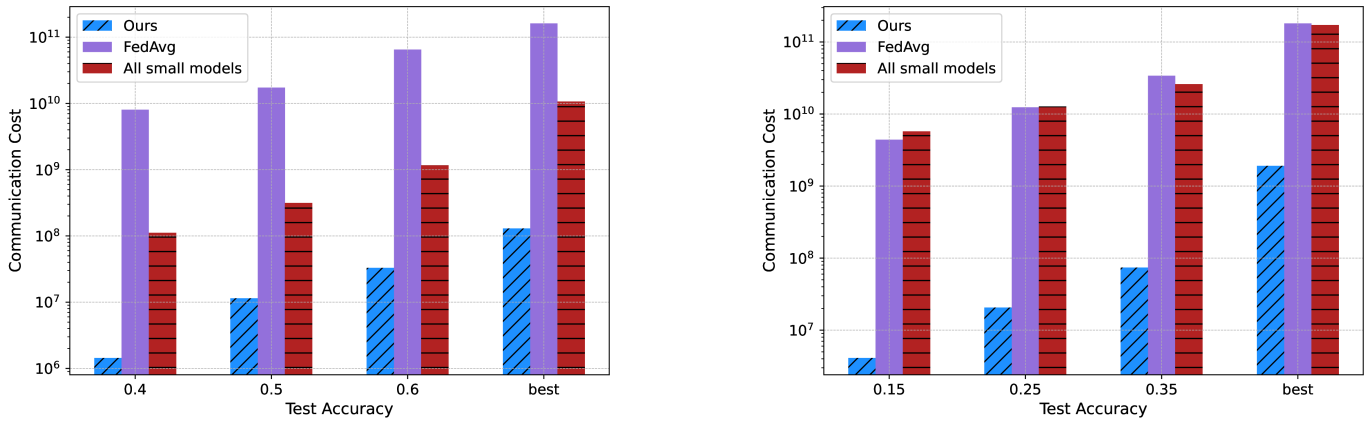
In our experiments, we consider a model-heterogeneous FL system training two different models simultaneously. The clients maintaining the same model are partitioned into the same group. The experiments are conducted on the benchmark datasets CIFAR-10 and CIFAR-100 [19] with different model combinations. The models used in the experiment and their model sizes are shown in Table I. The diverse local model sizes indicates that devices with various computational and resource capabilities in the context of model heterogeneity. The system scale and the specific number of clients in each group will be given individually in each experiment. We consider non-IID data partition across all experiments obtained by Dirichlet distribution-based method, in which the non-IIDness can be controlled by the concentration parameter $\alpha = 0.5$ in CIFAR-10 (0.3 in CIFAR-100) [20]. With regard to the channel fading in communication, we use the Rayleigh distribution with an expectation of 1 as the multiplicative noise of the signal. The interference when the edge server receives the signal is simulated by noise that conforms to the Gaussian distribution with a mean of 0 and a standard deviation of 0.02.

We adopt the following two schemes as the baselines for fair performance comparison. Our method and baselines are



(a) (b)

Fig. 3: Communication cost (the cumulative number of transmitted parameters) comparison between our method and the baselines: (a) training ResNet-18 and ResNet-14 on non-IID CIFAR-10 with 10 clients for each group, (b) training ResNet-34 and ResNet-18 on non-IID CIFAR-100 with 15 and 5 clients respectively.



(a) (b)

Fig. 4: Comparison of communication cost (the cumulative number of transmitted parameters) required to achieve different accuracy levels and best accuracy for small models, (a) training ResNet-18 and LeNet on non-IID CIFAR-10 with 50 and 20 clients respectively, (b) training ResNet-34 and ResNet-18 on non-IID CIFAR-100 with 15 and 5 clients respectively.

evaluated for performance under the same configuration of OTA FL systems.

- **FedAvg** is adopted as the baseline where clients with identical models perform training through gradient aggregation, essentially conducting conventional OTA FL within the group.
- Another baseline is to choose the model with fewer parameters for OTA FL across all clients, referred to as **All small models**, to accommodate the capabilities of the most resource-constrained devices in the system.

For performance comparison, the following two metrics are used for evaluation with respect to model generalization and communication efficiency performance.

- **Test accuracy:** We measure the accuracy of each client's local model and compute the average accuracy over each group and the overall system, weighted by the proportion of samples in each client's local training dataset relative to the total number of samples across all training datasets.
- **Communication cost:** We calculate the communication cost by recording the total number of parameters for either uplink or downlink transmission. The number of parameters to be transmitted is equal to the number of required orthogonal waveforms.

B. Performance Evaluation

Evaluation on generalization and communication efficiency. Fig. 2a and Fig. 2b demonstrate that our algorithm achieves satisfactory performance across multiple datasets and model combinations, indicating strong generalization performance in diverse model heterogeneity scenarios. Our method achieves a high level of accuracy, close to or even surpassing FedAvg, suggesting desirable training effectiveness. Notably,

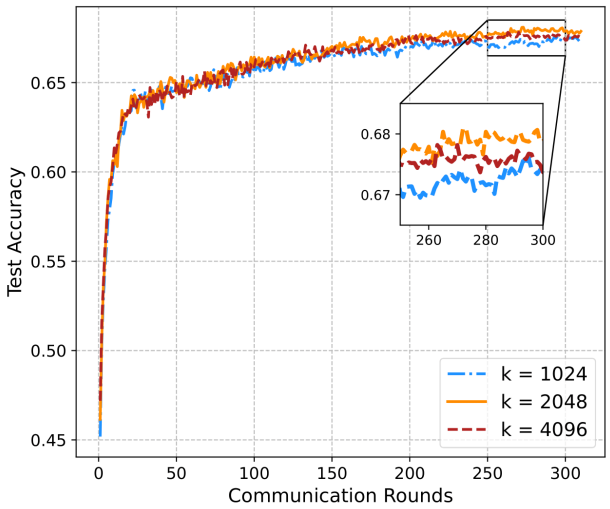


Fig. 5: Performance comparison under different dimensions of prototype vectors.

Fig. 3 shows that our communication efficiency achieves orders of improvement compared to the baselines. Overall, these results have shown that the prototype-based local update could serve as an effective and communication-efficient knowledge sharing approach, which bridges the training barrier of clients with diverse models.

Evident improvements in small models. Fig. 2c reveals that, aside from model combinations with relatively mild heterogeneity, even models with substantial differences in architectures and parameter sizes, such as ResNet and LeNet, can be effectively trained based on our proposed method. Meanwhile, our method enhances the performance of the small model group when it trains alongside the large model group. Fig. 4 also shows that our algorithm achieves substantial communication cost savings over the two baselines as the small model group reaches various accuracy levels.

Sensitivity analysis on prototype dimension. Fig. 5 investigates the effects of the prototype dimensions, indicating that a well-searched prototype dimension could achieve desirable performance on both accuracy and communication efficiency. In particular, lower prototype dimensions, such as $k = 1024$, lead to relatively poor performance due to insufficient representational capacity. When the dimension increased to $k = 2048$, the performance improved correspondingly. However, with an excessively large prototype vector dimension, such as $k = 4096$, the test accuracy degraded, possibly due to the increased difficulty of training and prototype alignment with such a high prototype dimension. Moreover, increasing the prototype dimension also leads to higher communication overhead.

V. Conclusion

In this paper, we developed a novel prototypical OTA FL framework that enables communication-efficient collaborative learning across edge devices with heterogeneous model architectures. Our method allows clients to transmit and aggregate model-agnostic prototype vectors instead of full model weight/gradient; as the sizes of the prototype vectors are much smaller compared to model weights, communication overheads are significantly reduced. We performed extensive experiments to validate the effectiveness of this approach. The results demonstrate that our method outperforms across diverse datasets and model combinations, demonstrating strong generalization capability.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, Fort Lauderdale, USA, Apr. 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[3] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[4] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.

[5] A. Şahin and R. Yang, "A survey on over-the-air computation," *IEEE Commun. Surv. Tutor.*, vol. 25, no. 3, pp. 1877–1908, Apr. 2023.

[6] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[7] Z. Chen, H. H. Yang, and T. Q. Quek, "Edge intelligence over the air: Two faces of interference in federated learning," *IEEE Commun. Mag.*, vol. 61, no. 12, pp. 62–68, Dec. 2023.

[8] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.* PMLR, Jul. 2020, pp. 5132–5143.

[9] Z. Chen, H. H. Yang, T. Q. S. Quek, and K. F. E. Chong, "Spectral co-distillation for personalized federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, Mar. 2020.

[11] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 2351–2363, Dec. 2020.

[12] Z. Shi, L. Zhang, Z. Yao, L. Lyu, C. Chen, L. Wang, J. Wang, and X.-Y. Li, "Fedfaim: A model performance-based fair incentive mechanism for federated learning," *IEEE Trans. Big Data*, pp. 1–13, Jun. 2022.

[13] Z. Chen, H. H. Yang, Y. Tay, K. F. E. Chong, and T. Q. Quek, "The role of federated learning in a wireless world with foundation models," *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 42–49, 2024.

[14] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017.

[15] N. Hoang, T. Lam, B. K. H. Low, and P. Jaillet, "Learning task-agnostic embedding of multiple black-box experts for multi-task model fusion," in *Proc. Int. Conf. Mach. Learn.* PMLR, Jul. 2020, pp. 4282–4292.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[17] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[18] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 8, Jun. 2022, pp. 8432–8440.

[19] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[20] A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh, "Federated learning based on dynamic regularization," in *Int. Conf. on Learn. Represent.*, Apr. 2021.