Feature Distillation is the Better Choice for Model-Heterogeneous Federated Learning

Yichen Li[1,2], **Xiuying Wang**[3], **Wenchao Xu**[4] **Haozhao Wang**[1], **Yining Qi**[1], **Jiahua Dong**[2], **Ruixuan Li**[1*]
[1]School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, China
[2]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates
[3]International School, Beijing University of Posts and Telecommunications, Beijing, China
[4]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
{ycli0204,hz_wang}@hust.edu.cn, leowang980@bupt.edu.cn

Abstract

Model-Heterogeneous Federated Learning (Hetero-FL) has attracted growing attention for its ability to aggregate knowledge from heterogeneous models while keeping private data locally. To better aggregate knowledge from clients, ensemble distillation, as a widely used and effective technique, is often employed after global aggregation to enhance the performance of the global model. However, simply combining Hetero-FL and ensemble distillation does not always yield promising results and can make the training process unstable. The reason is that existing methods primarily focus on logit distillation, which, while being model-agnostic with softmax predictions, fails to compensate for the knowledge bias arising from heterogeneous models. To tackle this challenge, we propose a stable and efficient Feature Distillation for model-heterogeneous Federated learning, dubbed **FedFD**, that can incorporate aligned feature information via orthogonal projection to integrate knowledge from heterogeneous models better. Specifically, a new feature-based ensemble federated knowledge distillation paradigm is proposed. The global model on the server needs to maintain a projection layer for each client-side model architecture to align the features separately. Orthogonal techniques are employed to re-parameterize the projection layer to mitigate knowledge bias from heterogeneous models and thus maximize the distilled knowledge. Extensive experiments show that FedFD achieves superior performance compared to state-of-the-art methods.

1 Introduction

Federated Learning (FL) has become a core method for collaboratively training neural networks across distributed clients while ensuring data privacy is protected [34, 50, 29]. Recently, this framework has attracted considerable attention and is being applied in various domains, such as autonomous driving systems [32, 37], recommender systems [27, 26], and intelligent healthcare solutions [7, 40].

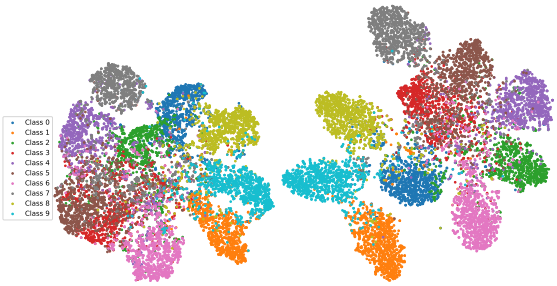
Typically, FL has been actively studied with a homogeneous model setting. With the development of IoT products, different devices often possess varying computing resources, namely different model training capabilities [39, 31]. In such a scenario, it is essential to explore model-heterogeneous federated learning (Hetero-FL) to maximize the utilization of distributed computing resources. The key challenge here lies in how to aggregate the shared knowledge from different models.

*Ruixuan Li is the corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

To address this issue, knowledge distillation [15] has emerged as a promising approach, focusing on aggregating the output soft predictions of multiple local models into the global model. The authors in [52] propose aggregating the logit output instead of model parameters and [21] utilize the aggregated class score to regulate the local training process. Building on the partial parameter aggregation proposed in [6], many works use the ensemble distillation to improve the global model like [33, 49]. [48] and [55] have focused on knowledge selection with logit distillation to optimize the aggregated model. [43] identifies the accurate and precise knowledge from local and ensemble predictions.

Although these methods achieve performance gains by integrating the ensemble distillation technique with Hetero-FL, how this distillation technique works in Hetero-FL remains uncertain, especially when it comes to logit-based distillation. While logit distillation can theoretically address the shifts in the class probability distribution caused by data heterogeneity in FL with homogeneous models perfectly by using a publicly available dataset with a similar distribution [46], *it is a sub-optimal strategy for model heterogeneity where the key challenge lies in different hidden layer representations.* During the distillation, each heterogeneous model will map the sample into a distinct feature space, resulting in significant variations in their softmax predictions (logits) [10]. It is widely acknowledged that the logit representation only focuses on the output layer but *can not align the representation in different feature spaces of heterogeneous models well, decreasing the distillation effectiveness*.



(a) Logit Representation. (b) Feature Representation.

Figure 1: The t-SNE visualization of ensemble knowledge representation by aggregated heterogeneous models on CIFAR-10.

We use t-SNE [45] to visualize the ensemble knowledge representation from client-side models on the distillation dataset in Figure.1. Compared with feature representation in our method, aggregated logit representation has **fuzzy** classification boundaries, indicating that the performance of the teacher model is not promising enough. *Moreover, we empirically find that the logit distillation will cause an unstable federated training process, then directly aggregating logits as the teacher's prediction is not always effective.* The specific experiments will be provided in Section 3.2.

To break the limitations of logit distillation for FD with heterogeneous models, we in this paper explore feature distillation to ensemble the distilled knowledge where the feature representation is closely related to the model structure. Using feature representation for ensemble distillation in Hetero-FL poses a novel challenge, as in a centralized environment, feature knowledge is often distilled from a large model (teacher model) to a smaller model (student model) with an external projection layer [41, 35]. This differs from the Hetero-FL, where small models on various client sides usually ensemble distill knowledge to a large model on the server, and the structures of these client-side models vary. Thus, designing the utilization of the projection layer and aligning feature representations of different client-side models becomes a primary challenge in ensemble distillation for Hetero-FL.

To explore this idea, we propose a stable and effective feature distillation method for Hetero-FL named **FedFD** that can align feature representations with orthogonal projection to mitigate the knowledge bias aggregated from heterogeneous models. More specifically, for each distillation sample, clients will extract the feature representation, and the server aggregates the representations of the same client-side model architecture to obtain a feature cluster formed by aggregated features from different model architectures. For each representation within the feature cluster, the server needs to train a projection layer to align the extracted feature representation of the server model with it and optimize the parameters of both the projection layer and the feature extractor of the server model. Furthermore, to prevent knowledge bias among aggregated feature representations, we utilize orthogonal projection techniques to maximize the transferred knowledge.

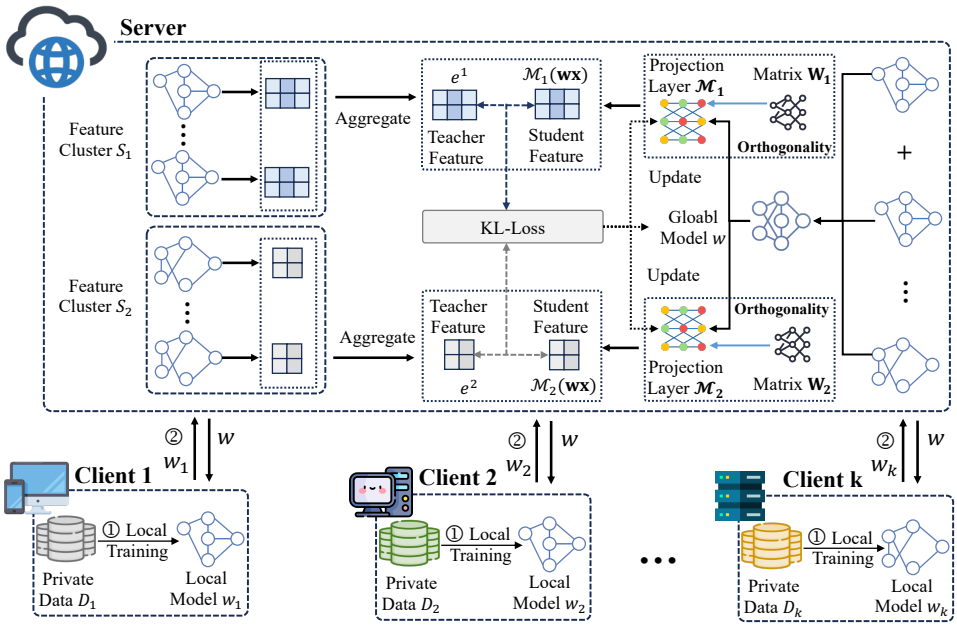


Figure 2: The framework of FedFD. Before knowledge distillation, each client first trains on its local dataset and then uploads its local model to the server. The server aggregates these models to obtain a global model. During the distillation process, clients perform hierarchical feature alignment. This involves firstly aggregating feature representations from client-side models with consistent architectures. Then, for each aggregated feature representation, a projection layer is maintained. This projection layer is obtained by transforming a square matrix to ensure its orthogonality. Finally, feature distillation is achieved by aligning the feature representations of different model architectures using KL divergence separately.

Through extensive experiments over three datasets and different settings (various model architectures and data distributions), we show that the proposed framework significantly improves the model accuracy as compared to state-of-the-art algorithms. The contributions of this paper are:

- We provide an in-depth analysis of model-agnostic federated knowledge distillation, identifying that existing methods primarily rely on logit distillation, which poses significant challenges for misleading distilled knowledge representation with heterogeneous models.
- We propose a novel framework named FedFD which can be seen as an off-the-shelf personalization add-on for Hetero-FL and it inherits privacy protection and efficiency properties as traditional distillation methods.
- We conduct extensive experiments on various datasets and settings. Experimental results illustrate that our proposed model outperforms the state-of-the-art methods by up to **16.09%** in terms of test accuracy on different tasks.

2 Related Work

Federated Learning is a framework that trains a unified global model by aggregating individual models from multiple clients, each trained on their locally stored datasets [25, 47, 28, 30]. One of the notable FL architectures is FedAvg [34], which strengthens the global model by aggregating parameters from locally trained models on private data. [23] involves incorporating a proximal term to mitigate the impact of differing data distributions across devices. While these methods are devoted to the homogeneous model across clients, [6] proposes to enable the training of heterogeneous local models with varying computation complexities on different clients. Similar to [6], [51] integrates flexibility in both width and depth, utilizing skip connections to bypass certain layers and structured pruning to manage width. [17] is a strategy that adjusts to varying depths by aggregating common layers from clients' networks, such as the VGG network, to create global models from different layer

groups. This paper focuses on federated learning with heterogeneous models across clients with improved knowledge distillation techniques.

Knowledge Distillation leverages the knowledge of a pre-trained model to supervise a smaller model, facilitating its application and deployment in environments with limited resources [14]. The domain primarily encompasses two areas: logits distillation [38, 18, 1, 16] and feature distillation [44, 36, 13, 5]. Logits distillation, centered on classification tasks, introduces an additional goal to minimize the prediction discrepancy between the student and teacher models, initially using KL divergence [14] and later extended through spherical normalization [8], label decoupling [57], and probability reweighing [38]. Our research focuses on feature distillation due to its versatility across tasks [5] and modalities [42]. The FSP matrix is manually designed to capture feature relationships across residual layers [54]. Similarly, other studies proposed transferring knowledge via Gram matrices [24]. Activation boundaries and gradients capturing the loss landscape have also proven effective as supervisory signals [44]. We in this paper particularly focus on the feature distillation in FD by orthogonal projection and ensemble distillation.

Federated Distillation involves extracting knowledge from multiple teacher models, each trained by different clients, and transferring it to a student model [53, 9, 2]. [33] introduces the concept of applying knowledge distillation in a server setting, utilizing an unlabeled proxy dataset to transfer knowledge from local models to a global model. [4] further develops this by linearly aggregating multiple local models, using weights derived from the Bayesian posterior, to create a series of combined models. These combined models are then distilled into a single global model. To break out the limitation of relying on an unlabeled auxiliary dataset, [58, 56, 49] propose suggest replacing the proxy dataset with data generated by generative models, enabling the distillation without the need for actual data. We analyze the challenge of employing existing FD methods in the FL with heterogeneous models and propose to develop feature distillation instead of logit distillation.

3 Methodology

In this section, we first formulate the standard FL process with both homogeneous and heterogeneous models. Then, we analyze the failure of logit distillation in Hetero-FL with experiments. Last, we introduce our feature distillation-based method, FedFD. The workflow of FedFD is shown in Algorithm 1 and Figure.2 illustrates the FedFD framework.

3.1 Problem Formulation

A typical FL problem can be formalized by collaboratively training a global model for K total clients in FL. We consider each client k can only access to his local private dataset $D_k = \{x_k^{(i)}, y_k^{(i)}\}$, where $x_k^{(i)}$ is the i-th input data sample and $y_k^{(i)} \in \{1, 2, \cdots, C\}$ is the corresponding label of $x_k^{(i)}$ with C classes. We denote the number of data samples in dataset D_k by $|D_k|$. The objective of the FL system is to learn a global model w that minimizes the total empirical loss over the dataset D:

$$\begin{aligned} &\min_w \mathcal{L}(w) = \sum_{k=1}^{K} \frac{|D_k|}{|D|} \mathcal{L}_k(w), \\ &\text{where } \mathcal{L}_k(w) = \frac{1}{|D_k|} \sum_{i=1}^{|D_k|} \mathcal{L}_{CE}(w; x_i^k, y_i^k), \end{aligned} \tag{1}$$

where $\mathcal{L}_k(w)$ is the local loss in the k-th client and $\mathcal{L}_{CE}$ is the cross-entropy loss function that measures the difference between the prediction and the ground truth labels. ***For homogeneous models***, the server only needs to average local parameters to obtain the global follow:

$$w := \sum_{k=1}^{K} \frac{|D_k|}{|D|} \cdot w_k. \tag{2}$$

While ***for heterogeneous models***, we follow the protocol proposed in [6] and assume that all client models can be divided into p different architecture sets $\{S_1, S_2, \ldots, S_p\}$. Then, we perform the

global aggregation as follows:

$$w_p^s = \sum_{w_i \in S_p} \frac{w_i}{|S_p|}, \quad w_{p-1}^s \setminus w_p^s = \frac{1}{K - |S_p|} \sum_{w_i \in S_p} w_{p-1}^s \setminus w_p^s,$$

$$w = w_p^s \cup (w_{p-1}^s \setminus w_p^s), \ldots, \cup (w_1^s \setminus w_2^s). \tag{3}$$

where w_p^s denotes the aggregated model of the p-th architecture of client-side models. For notational convenience, we have dropped the training iteration index and simplified the weight for each aggregated client-side model in Eq. (3), which is often weighed by the ratio of sample numbers.

3.2 Logit Distillation Fails in Hetero-FL

In existing FL methods, the key technique is to transfer knowledge from the teacher model to the student model by utilizing the model's soft predictions (logits) on the distillation dataset. Here, the teacher model prediction is defined with aggregated logits from multiple clients, which can also be referred to as ensemble distillation in FL. This technique works because in Homo-FL (for homogeneous models), where the client models share the same structure, the logits output by these models do not suffer from model-based biases and can mitigate further data heterogeneity. However, it is inadvisable to directly apply this logit distillation technique to Hetero-FL (for heterogeneous models), and existing related work has not addressed whether the logits from different client model architectures can be directly aggregated as the teacher model prediction. Based on Figure.1, we conduct further experiments. We selected several robust ensemble distillation methods and conducted experiments on CIFAR-10 under both Homo-FL and Hetero-FL settings. The details of these methods are described in Section 4.1, and the learning curves are shown in Figure.3.

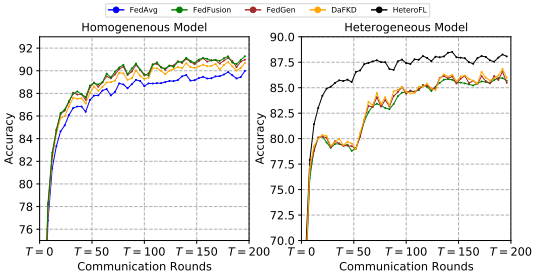


Figure 3: Learning curves of logit distillation-based methods on CIFAR-10 with different client-side model architectures.

Under the Homo-FL setting, all methods converge quickly and stably, demonstrating the effectiveness of logit distillation. However, in the Hetero-FL setting, the FL algorithm experiences training instability, where peaks repeatedly appear in the curve, and the methods ultimately fail to converge to the optimal value. Although logit distillation can accelerate model convergence in the early stages of training, as the local model gradually converges on local data, the distillation process becomes highly unstable. This is contrary to the results observed in Homo-FL. The reason is that logit distillation solely relies on soft predictions without considering differences in model architectures. Therefore, we will next introduce our feature-based distillation method to address this issue.

3.3 FedFD: Feature Distillation for Hetero-FL

In this paper, we seek to unlock the potential of feature distillation in Hetero-FL. Despite the generality of feature distillation, it is frequently accompanied by complex design choices and heuristics. These decisions stem from loss functions between intermediate feature representations, leading to extra training costs. To overcome these limitations, we just employ the feature distillation that solely utilizes feature representation before the classifier.

In Hetero-FL, although the teacher model is typically the ensemble output of client-side small models, it is still necessary to attach a projection layer after the global model (student model) to ensure that knowledge can be back-propagated through the projection layer into the model parameters during the distillation process. However, this approach poses **two** challenges: (1) Due to the diversity of client-side model architectures, if the server maintains a personalized projection layer for each client, it may lead to difficult training because each client contributes only a tiny amount of knowledge. Additionally, in FL scenarios, the number of clients participating in pre-training is usually large, and maintaining an excessive number of projection layers can accumulate and result in significant storage costs for the server. (2) The feature knowledge derived from different client-side models may conflict

Algorithm 1: FedFD

Input : T: communication round; K: client number; D_k: local dataset for the client k; w: global model; w_k: local model; $\mathcal{M}_k$: projection layer; S_k: the k-th model architecture; $\mathbf{W}_k$: matrix for orthogonality.

Output : $w, \{w_1, \ldots, w_k\}$: global and local models.

for $c = 1$ ***to*** T **do** `// communication round`
  Server randomly selects a subset of devices S_t;
  Server send the global model w to devices.
  for *each selected client* $k \in S_t$ **in parallel do**
    Train the local model w_k with (1);
    Send the local model w_k back to the server.
  end
  $w \leftarrow \text{ServerAggregation}(\{w_k\}_{k \in S_t})$ with (3);
  Get the aggregated feature representation e^d with (5);
  Orthogonalize $\mathbf{W}_D$ to obtain projection layer $\mathcal{M}_d$ with (7);
  Distill the feature knowledge to the global model with (9).
end

within different projection layers, and combining the complex non-linear knowledge together may not optimally train the feature extractor module of the global model.

To address these issues, we propose the feature-based distillation method, FedFD, with two main components: hierarchical feature alignment and parameter orthogonality. To obtain the classification model, in each round, the participating client k firstly locally trains the model w_k with (1) and sends it to the server. After receiving uploaded models, the server aggregates multiple local models to get the global model w with (3). Like [33][49][58], the server treats the knowledge of the client model on the distillation dataset as a teacher model to distill the global model and enhance its performance. Based on previous experiments, we explore using feature representation as the distilled knowledge. Denote that the local model w_k of client k consists of a feature extractor $\mathbf{w}_k^d$ and a classifier head θ_k. d is the dimension of the output feature. For each distillation sample $\mathbf{x}$, its feature representation e_k^d over the extractor $\mathbf{w}_k^d$ can be obtained as follows:

$$e_k^d = f(\mathbf{w}_k^d; \mathbf{x}), \ \ \forall k \in [1, K]. \tag{4}$$

Then, we divide the features into m groups $\{S_{d_1}, \ldots, S_{d_m}\}$, where each group $S_d = \{\mathbf{w}_1^d, \ldots, \mathbf{w}_k^d\}$ contains all extractors with the same structure outputting the d-dimensional feature. We use $|S_d|$ to represent the number of features in the group S_d. Next, we aggregate all feature representations in the group S_d as:

$$e^d = \frac{1}{|S_d|} \sum_{i=1}^{|S_d|} e_i^d, \ \ \text{where} \ \ S_d = \{\mathbf{w}_1^d, \ldots, \mathbf{w}_{|S_d|}^d\}. \tag{5}$$

Instead of maintaining the projection layer separately for each client, the server now only needs to train $(m-1)$ projection layers $\{\mathcal{M}_2, \ldots, \mathcal{M}_m\}$. *This not only reduces training parameters but also ensures that each projection layer has sufficient knowledge for distillation.*

However, this does not resolve the knowledge conflict among different model architectures, as the server still needs to integrate knowledge from various projection layers to optimize the global model parameters $\mathbf{w}$. We indicate the knowledge $\mathbf{S}_d$ after the projection layer $\mathcal{M}_d$ as:

$$\mathbf{S}_d = \mathcal{M}_d(\mathbf{w}\mathbf{x}) = \mathcal{M}_d\mathbf{Z}, \ \ \text{where} \ \ \mathbf{Z} = \mathbf{w}\mathbf{x}. \tag{6}$$

While in knowledge distillation, a larger number of distillation samples often leads to better distillation effects, as they can better cover the knowledge within the training data. This results in $\mathbf{Z}$ being a full-rank matrix; consequently, the $\mathbf{S}_d$ will also be a full-rank matrix that can be regarded as a non-linear mapping process, where the global model fails to discern the knowledge, leading to knowledge conflicts.

To address this issue, since we cannot alter the knowledge distribution of $\mathbf{Z}$, we choose to process the projection layer $\mathcal{M}_d$ so that they can map the knowledge $\mathbf{Z}$ into separate feature spaces. Here, we

will investigate orthogonal projection transformations, which can resolve knowledge conflicts and maintain feature shapes due to their rotational invariance. It is noteworthy that, due to the differing feature dimensions between $\mathbf{Z}$ and $\mathbf{S}_d$, with $\mathbf{Z}$ having a higher dimension than $\mathbf{S}_d$, $\mathcal{M}_d$ is no longer an orthogonal square matrix but rather a column matrix, specifically a Stiefel matrix manifold with orthogonal column vectors [12]. The favorable topological properties of such a matrix can ensure support for gradient descent updates.

To maintain the orthogonality of the column vectors in $\mathcal{M}_d$, we have observed many existing methods such as Cayley transformation [3] and Gram-Schmidt method. However, these methods all come with relatively high time performance costs. In this paper, we propose to generate the $\mathcal{M}_d$ with a skew-symmetric matrix $\mathbf{W}_d$ as follows:

$$\exp(\mathbf{W}_d) \cdot \exp(\mathbf{W}_d)^T = \exp(\mathbf{W}_d + \mathbf{W}_d^T) = \exp(-\mathbf{W}_d^T + \mathbf{W}_d^T) = \mathbf{I}. \tag{7}$$

$$\exp(\mathbf{W}_d) = \mathbf{I} + \mathbf{W}_d + \frac{\mathbf{W}_d^2}{2!} + \frac{\mathbf{W}_d^3}{3!} + \cdots + \frac{\mathbf{W}_d^n}{n!}. \tag{8}$$

where $\mathbf{W}_d = -\mathbf{W}_d^T$. The parameters of W_d are randomly initialized. Although this is an infinite series, a reasonably accurate value can usually be obtained by taking the first few terms at a very low computational cost. Ultimately, $\mathcal{M}_d$ can be obtained by truncating the column vectors of $\exp(\mathbf{W}_d)$ to match the feature dimension of $\mathbf{S}_d$. The client updates $\mathbf{W}_d$ through back-propagation with Eq.(9). *By employing orthogonal projection, we ensure linear transformations of features, preventing knowledge conflicts and preserving feature shapes, thereby achieving maximized knowledge distillation.*

Through orthogonal projection, $\mathcal{M}_d(\mathbf{wx})$ and e^d share the same dimensionality. Next, we update the parameters w and $\mathcal{M}$ to align the feature representations using Kullback-Leibler divergence:

$$\min_{\mathbf{w},\{\mathcal{M}_2,\ldots,\mathcal{M}_m\}} \frac{1}{m-1} \sum_{i=2}^{m} KL(\mathcal{M}_i(\mathbf{wx}), e^i). \tag{9}$$

After knowledge distillation, the server will broadcast the global model w to the clients participating in the following communication round.

Modularity. FedFD demonstrates a key characteristic: modularity. Existing FL techniques can be seamlessly integrated with FedFD as a ready-to-use enhancement with the following several benefits:

- **Optimization:** The proposed framework can accommodate aggregation methods beyond HeteroFL [6] for global model updates, retaining convergence advantages.
- **Privacy:** FedFD maintains the same level of network communication as standard FL algorithms, avoiding privacy issues associated with uploading generative models or extra prototype-like information.
- **Flexibility:** Although we search for the additional distillation datasets here, our framework can be combined with existing data-free distillation techniques, balancing resource constraints and computational overheads in selecting distillation datasets.

4 Experiments

In this section, we evaluate our proposed method using three datasets and various baselines. We investigate the relationship between data heterogeneity and training efficiency. Additionally, we conduct ablation studies to examine each module in FedFD. Finally, we conduct a sensitivity analysis to verify the effectiveness of our method.

4.1 Experiment Setup

Dataset: We conduct our experiments with heterogeneously partitioned datasets over three datasets: CIFAR-10, CIFAR-100 [19], and Tiny-ImageNet [20]. Like [58, 49], we use the Dirichlet distribution Dir(α) on labels to simulate the data heterogeneity. We apply all the training samples and distribute them to user models, and we use all the testing samples for the performance evaluation.

Baselines: For a fair comparison with other key works, we follow the same protocols proposed by [6] to set up FD tasks with heterogeneous models, which is recorded as "-hetero". We evaluate our

Table 1: Performance comparison of various methods with the test accuracy.

Categories	Methods	Metrics	CIFAR-10			CIFAR-100			Tiny-ImageNet		
			α=10.0	α=1.0	α=0.1	α=10.0	α=1.0	α=0.1	α=10.0	α=1.0	α=0.1
Classic FL	HeteroFL	Local	80.11±0.95	75.83±1.35	63.53±4.66	53.37±1.11	49.33±1.76	38.07±4.52	29.71±2.85	24.12±5.03	18.54±3.40
		Global	88.45±0.03	87.53±0.15	78.02±0.65	58.51±0.19	57.42±0.12	53.98±0.43	32.38±0.51	29.88±2.72	23.25±2.96
	MOON-hetero	Local	80.53±0.77	75.87±0.99	63.98±3.20	52.85±2.03	50.21±2.42	38.21±3.61	28.36±1.70	24.47±2.42	17.94±2.51
		Global	88.05±0.14	87.92±0.17	79.05±0.89	58.39±0.20	57.55±0.17	55.01±0.34	31.28±1.61	30.68±1.92	24.91±1.78
Homo-FL	FedFusion-hetero	Local	82.35±0.51	75.27±1.00	62.20±5.37	51.23±1.73	48.21±2.08	37.53±5.15	33.98±2.07	25.62±2.29	20.31±5.25
		Global	86.69±0.30	85.70±0.15	78.47±1.11	59.53±0.11	58.86±0.12	56.12±0.35	35.05±1.87	31.56±1.23	26.09±0.73
	FedGen-hetero	Local	82.29±0.62	76.01±0.97	62.74±3.17	51.99±3.67	47.80±1.12	39.79±6.02	34.04±2.93	25.37±2.56	21.77±6.89
		Global	87.34±0.07	86.81±0.08	79.22±2.03	58.49±0.06	57.63±0.86	56.05±0.52	34.71±1.90	32.00±1.56	26.84±1.52
	DaFKD-hetero	Local	83.83±1.01	77.69±2.06	64.71±2.29	52.76±3.33	48.99±1.73	39.98±5.09	33.85±0.96	26.02±3.19	22.26±4.49
		Global	89.26±0.31	87.84±0.64	80.07±1.23	58.97±0.43	58.52±0.19	57.33±1.00	35.69±1.48	32.57±1.34	26.54±1.73
Hetero-FL	FedMD	Local	70.33±3.98	63.47±4.13	60.67±5.09	42.28±6.11	39.64±5.45	26.90±7.70	23.84±3.76	19.37±3.20	13.89±6.70
		Global	78.91±2.32	75.48±3.31	66.67±2.50	46.37±3.11	49.40±4.65	39.44±9.89	26.37±2.80	22.60±2.43	19.42±3.17
	MSFKD	Local	82.26±0.63	77.31±3.04	62.52±9.36	52.06±2.20	49.94±1.27	39.72±3.70	34.30±3.11	27.93±2.71	22.59±5.84
		Global	87.79±0.18	86.98±0.31	80.00±1.75	58.88±0.38	59.09±0.24	56.29±0.83	36.29±0.60	31.62±0.87	26.70±2.33
	FedGD	Local	82.40±1.01	76.02±1.39	63.71±8.06	51.02±2.07	50.17±1.86	39.17±2.78	35.08±2.55	26.50±1.64	23.47±4.83
		Global	87.52±0.24	87.22±0.13	79.31±0.75	59.26±0.41	58.03±0.26	56.34±0.65	36.86±1.06	30.66±1.59	27.53±2.20
	FedFD (ours)	Local	**84.91**±0.42	**78.03**±1.49	**65.33**±9.22	**54.98**±1.76	**52.24**±1.90	**41.68**±4.12	**36.78**±5.13	**30.90**±5.25	**23.41**±4.99
		Global	**90.06**±0.03	**89.64**±0.23	**82.74**±0.58	**61.07**±0.22	**60.86**±0.10	**59.24**±0.48	**40.27**±1.33	**34.24**±1.13	**29.09**±1.69

method with the following baselines. **(1) Representative FL models:** HeteroFL [6], MOON-hetero [22]; **(2) FL for homogeneous model:** FedFusion-hetero [33], FedGen-hetero [58], DaFKD-hetero [49]; **(3) FL for heterogeneous model:** FedMD [21], MSFKD [48], FedGD [55].

Configurations: Unless otherwise mentioned, we set the number of local training epoch $E = 10$, communication round $T = 200$, and the client number $K = 20$ with an active ratio $r = 0.4$. For local training, the batch size is 64 and the weight decay is $1e-4$. The learning rate is 0.01 for distillation and 0.001 for training the local model. For the model on the server, we employ ResNet-18 [11] as the basic backbone. Like [6], we construct ten different computation complexity levels $\{a, b, c, \ldots j\}$ with the hidden channel decay rate 10%. For example, model "a" has all the model parameters, while models "b" and "c" have the 90% and 80% effective parameters. We use "a-d-g" three different model architectures in our experiments and conduct more experiments with other architectures in Section 4.2. Similarly to [6], we have conducted statistics based on the performance of the global model for all secondary experiments.

4.2 Performance Overview

Test Accuracy. Table 1 shows the test accuracy of various methods with heterogeneous data across three datasets. Each experiment set is run twice, and we take each run's final ten rounds' accuracy and calculate the average value and standard variance. We report both the global model accuracy and the average accuracy of all local models. Firstly, we observe that the performance of the global model significantly outpaces that of the local model. For the Classic FL method, MOON does not achieve a substantial advantage with heterogeneous models, suggesting that the contrastive loss between different model architectures provides limited improvement to the method. As for the Homo-FD method, while logit distillation introduces performance fluctuations, it offers some enhancement after an increase in training epochs. Both FedGen and DaFKD are data-free distillation methods, and compared to FedFusion, their performance improvements are not as significant as reported in the original papers. A possible reason is that the training of the generator remains unstable, especially considering we employ relatively complex datasets instead of handwriting-digit recognition datasets like MNIST. Regarding Hetero-FL methods, FedMD performs poorly due to the absence of aggregated parameters. The other two methods follow the logit distillation paradigm and exhibit superior performance. FedFD achieves the best performance in all cases by up to 16.09% in terms of global model accuracy.

Communication Efficiency. Figure.2 (Table) compares the communication efficiency of various methods by measuring the communication rounds required to achieve the test accuracy and Figure.4 focuses on the learning curves. Although knowledge distillation methods can accelerate model convergence, the training process of logit distillation is unstable, as we have detailed above. With the orthogonal projection technique in feature distillation, FedFD achieves the fastest convergence with a stable training process.

Ablation Study. As shown in Table 3, we evaluate the effects of each module in our model via ablation studies. -w/o feature alignment and -w/o orthogonal projection denote the performance of our model without using hierarchical feature alignment and orthogonal projection. Compared with FedFD, the performance of FedFD -w/o orthogonal projection and FedFD -w/o orthogonal projection degrades evidently with a range of 0.63%$\sim$2.43%. While moving both components, the performance will drop significantly. Specifically, the orthogonal projection technique plays a much more important role with the obvious improvement in test accuracy,

Table 2: Evaluation of different baselines on three datasets (α = 1.0), in terms of the number of communication rounds to reach target test accuracy (acc). The best results are **bold**.

Methods	CIFAR-10		CIFAR-100		Tiny-ImageNet	
	acc=70%	*acc*=80%	*acc*=55%	*acc*=60%	*acc*=25%	*acc*=30%
HeteroFL	12±3	25±6	20±3	> 200	124±16	> 200
MOON-hetero	11±4	26±7	22±9	> 200	100±9	188±7
FedFusion-hetero	20±7	52±7	61±11	> 200	96±25	153±28
FedGen-hetero	21±11	38±15	198±20	> 200	115±12	176±24
DaFKD-hetero	23±9	41±9	44±13	191±27	85±8	126±21
FedMD	62±20	> 200	139±21	> 200	> 200	> 200
MSFKD	19±9	45±17	33±10	171±20	87±22	115±31
FedGD	17±8	34±5	40±12	189±21	106±14	150±23
FedFD (ours)	**10**±5	**20**±9	**16**±6	**60**±18	**67**±8	**99**±19

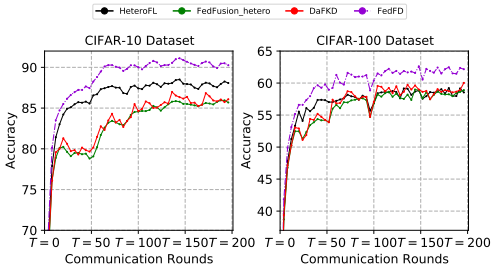


Figure 4: Convergence and efficiency comparison of various methods on two datasets.

and the hierarchical feature alignment technique has a relatively small impact on test accuracy, which may be due to the scale of the experiment, but it can save storage costs. Experiment results verify that all modules are essential to train a robust Hetero-FL model.

Table 3: Ablation study of FedFD with two main components.

Method	CIFAR-10		CIFAR-100	
	α=1.0	α=0.1	α=1.0	α=0.1
FedFD	**89.64**	**82.74**	**60.86**	**59.24**
-w/o feature alignment	89.01	81.56	59.77	58.67
-w/o orthogonal projection	87.96	80.92	59.33	57.29
-w/o both components	85.70	78.47	58.86	56.12

Table 4: Evaluation of combination of various client-side models levels for CIFAR-10 dataset ($\alpha = 1.0$).

Method	Model Heterogeneity			
	a-d-g	a-f	b-d-f	a-c-d-f-i
HeteroFL	87.53±0.15	88.15±0.18	85.01±0.30	82.53±0.27
FedFusion-hetero	85.70±0.15	86.85±0.10	85.37±0.28	84.10±0.15
FedGD	87.96±0.08	88.47±0.21	85.23±0.40	84.01±0.46
FedFD	**89.64**±0.23	**89.92**±0.19	**87.47**±0.35	**85.92**±0.28

Data Heterogeneity. Table 1 illustrates the variation in test accuracy across different levels of data heterogeneity. A clear trend emerges, with all methods exhibiting improved accuracy as data heterogeneity decreases. Simultaneously, we observed that the data heterogeneity has a greater impact on local models than on global models. Specifically, when α = 0.1, the performance of local models across all baselines on CIFAR-100 declines significantly, this underscores the urgent need to investigate Hetero-FL, as they are more susceptible to the influence of data distribution. Notably, FedFD consistently outperforms other methods, achieving the most significant improvements across all settings.

Parameter Sensitivity Analysis. In this section, we first explore the model heterogeneity issue with different models. As shown in Table 4, we conduct experiments using four different models and define that a larger number of models represented a higher level of model heterogeneity. As the degree of heterogeneity increased, the performance of all methods declined, with FedFusion showing a particularly significant drop. This verified the shortcoming of logit distillation in the context of model heterogeneity. In contrast, FedFD consistently retains superior performance.

Table 5: The scalability of FedFD and other baselines (α=0.01).

Dataset	HeteroFL	FedFusion-hetero	MSFKD	FedFD
CIFAR-10	63.29±5.19	64.14±8.16	65.30±6.77	**67.03**±9.34
CIFAR-100	44.59±1.32	45.31±1.09	44.90±2.73	**48.05**±2.41
Tiny-ImageNet	21.86±4.33	22.97±2.89	23.26±2.54	**25.12**±1.53

Table 6: The different architectures of models (CNN and ResNet).

Method	CIFAR-10		CIFAR-100		Tiny-ImageNet	
	α=1.0	α=0.1	α=1.0	α=0.1	α=10.0	α=1.0
FedMD	67.59±3.61	53.10±9.23	30.52±3.98	24.47±1.70	21.29±1.67	19.95±2.80
FedFD	**71.28**±1.25	**59.51**±7.13	**34.93**±0.26	**28.98**±0.74	**30.79**±1.28	**27.96**±0.47

Then, we examine the scalability of our method. As shown in Table 5, although all methods exhibit significant performance degradation in large-scale experiments, forcing some clients to own very few samples, FedFD outperforms other baselines, demonstrating its robustness and effectiveness.

Model Architecture. Although the above experiments validate the effectiveness of FedFD within the HeteroFL framework, *FedFD fundamentally constitutes an advanced feature distillation method independent of model architectures and parameter aggregation strategies*. We selected HeteroFL as the framework due to its widespread recognition, which facilitates comprehensive comparisons with other baselines. In Table 6, we conducted experiments using two architectures to mitigate potential impacts arising from the HeteroFL framework. Here, we adopt the FedMD to perform knowledge distillation to fuse client knowledge directly. The experimental results demonstrate the adaptability of FedFD with diverse architectures.

5 Conclusion

We introduced FedFD, a straightforward framework, to tackle feature distillation using heterogeneous models within federated learning. FedFD serves as a minimal personalization extension for any federated learning algorithms involving global model aggregation, ensuring privacy and communication efficiency. Comprehensive experiments demonstrate that FedFD can enhance test accuracy.

Acknowledgments and Disclosure of Funding

This work is supported by the National Key Research and Development Program of China under grant 2024YFC3307900; the National Natural Science Foundation of China under grants 62376103, 62302184, 62436003 and 62206102; Major Science and Technology Project of Hubei Province under grant 2024BAA008; Hubei Science and Technology Talent Service Project under grant 2024DJC078; Ant Group through CCF-Ant Research Fund; and Fundamental Research Funds for the Central Universities under grant YCJJ20252319. The computation is completed in the HPC Platform of Huazhong University of Science and Technology.

References

[1] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10925–10934, 2022.

[2] Ilai Bistritz, Ariana Mann, and Nicholas Bambos. Distributed distillation for on-device learning. In *Proceedings of Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020.

[3] Arthur Cayley. *The collected mathematical papers*, volume 7. University, 1894.

[4] Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[5] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5008–5017, 2021.

[6] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.

[7] Jiahua Dong, Hongliu Li, Yang Cong, Gan Sun, Yulun Zhang, and Luc Van Gool. No one left behind: Real-world federated class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2054–2070, 2024.

[8] Jia Guo, Minghao Chen, Yao Hu, Chen Zhu, Xiaofei He, and Deng Cai. Reducing the teacher-student gap via spherical knowledge disitllation. *arXiv preprint arXiv:2010.07485*, 2020.

[9] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 11020–11029, 2020.

[10] Zhiwei Hao, Jianyuan Guo, Kai Han, Yehui Tang, Han Hu, Yunhe Wang, and Chang Xu. One-for-all: Bridge the gap between heterogeneous architectures in knowledge distillation. *Advances in Neural Information Processing Systems*, 36:79570–79582, 2023.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Ping He, Xiaohua Xu, Jie Ding, and Baichuan Fan. Low-rank nonnegative matrix factorization on stiefel manifold. *Information Sciences*, 514:131–148, 2020.

[13] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1921–1930, 2019.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *CoRR*, abs/1811.11479, 2018.

[16] Ying Jin, Jiaqi Wang, and Dahua Lin. Multi-level logit distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24276–24285, 2023.

[17] Honggu Kang, Seohyeon Cha, Jinwoo Shin, Jongmyeong Lee, and Joonhyuk Kang. Nefl: Nested federated learning for heterogeneous clients. *arXiv preprint arXiv:2308.07761*, 2023.

[18] Youmin Kim, Jinbae Park, YounHo Jang, Muhammad Ali, Tae-Hyun Oh, and Sung-Ho Bae. Distilling global and local logits with densely connected relations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6290–6300, 2021.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[20] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[21] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[22] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[23] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[24] Xiaojie Li, Jianlong Wu, Hongyu Fang, Yue Liao, Fei Wang, and Chen Qian. Local correlation consistency for knowledge distillation. In *European Conference on Computer Vision*, pages 18–33. Springer, 2020.

[25] Yichen Li, Qunwei Li, Haozhao Wang, Ruixuan Li, Wenliang Zhong, and Guannan Zhang. Towards efficient replay in federated incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12820–12829, June 2024.

[26] Yichen Li, Qiyu Qin, Gaoyang Zhu, Wenchao Xu, Haozhao Wang, Yuhua Li, Rui Zhang, and Ruixuan Li. A systematic survey on federated sequential recommendation. *arXiv preprint arXiv:2504.05313*, 2025.

[27] Yichen Li, Yijing Shan, Yi Liu, Haozhao Wang, Wei Wang, Yi Wang, and Ruixuan Li. Personalized federated recommendation for cold-start users via adaptive knowledge fusion. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2700–2709, New York, NY, USA, 2025. Association for Computing Machinery.

[28] Yichen Li, Haozhao Wang, Yining Qi, Wei Liu, and Ruixuan Li. Re-fed+: A better replay strategy for federated incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[29] Yichen Li, Haozhao Wang, Wenchao Xu, Tianzhe Xiao, Hong Liu, Minzhu Tu, Yuying Wang, Xin Yang, Rui Zhang, Shui Yu, Song Guo, and Ruixuan Li. Unleashing the power of continual learning on non-centralized devices: A survey, 2024.

[30] Yichen Li, Yuying Wang, Haozhao Wang, Yining Qi, Tianzhe Xiao, and Ruixuan Li. Fedssi: Rehearsal-free continual federated learning with synergistic synaptic intelligence. In *Forty-second International Conference on Machine Learning*.

[31] Yichen Li, Wenchao Xu, Haozhao Wang, Ruixuan Li, Yining Qi, and Jingcai Guo. Personalized federated domain-incremental learning based on adaptive knowledge matching, 2024.

[32] Yijing Li, Xiaofeng Tao, Xuefei Zhang, Junjie Liu, and Jin Xu. Privacy-preserved federated learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8423–8434, 2021.

[33] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

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

[35] Roy Miles, Ismail Elezi, and Jiankang Deng. Vkd: Improving knowledge distillation using orthogonal projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15720–15730, 2024.

[36] Roy Miles, Adrian Lopez Rodriguez, and Krystian Mikolajczyk. Information theoretic representation distillation. *arXiv preprint arXiv:2112.00459*, 2021.

[37] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(3):1–37, 2022.

[38] Yulei Niu, Long Chen, Chang Zhou, and Hanwang Zhang. Respecting transfer gap in knowledge distillation. *Advances in Neural Information Processing Systems*, 35:21933–21947, 2022.

[39] Kilian Pfeiffer, Martin Rapp, Ramin Khalili, and Jörg Henkel. Federated learning for computationally constrained heterogeneous devices: A survey. *ACM Computing Surveys*, 55(14s):1–27, 2023.

[40] Bjarne Pfitzner, Nico Steckhan, and Bert Arnrich. Federated learning in a medical context: a systematic literature review. *ACM Transactions on Internet Technology (TOIT)*, 21(2):1–31, 2021.

[41] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2015.

[42] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[43] Jiawei Shao, Fangzhao Wu, and Jun Zhang. Selective knowledge sharing for privacy-preserving federated distillation without a good teacher. *Nature Communications*, 15(1):349, 2024.

[44] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[45] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[46] Chunnan Wang, Xiang Chen, Junzhe Wang, and Hongzhi Wang. Atpfl: Automatic trajectory prediction model design under federated learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6563–6572, 2022.

[47] Chunnan Wang, Xiang Chen, Junzhe Wang, and Hongzhi Wang. ATPFL: automatic trajectory prediction model design under federated learning framework. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, New Orleans, LA, USA,June 18-24,June 18-24*, pages 6553–6562, 2022.

[48] Dong Wang, Naifu Zhang, Meixia Tao, and Xu Chen. Knowledge selection and local updating optimization for federated knowledge distillation with heterogeneous models. *IEEE Journal of Selected Topics in Signal Processing*, 17(1):82–97, 2022.

[49] Haozhao Wang, Yichen Li, Wenchao Xu, Ruixuan Li, Yufeng Zhan, and Zhigang Zeng. Dafkd: Domain-aware federated knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20412–20421, 2023.

[50] Haozhao Wang, Haoran Xu, Yichen Li, Yuan Xu, Ruixuan Li, and Tianwei Zhang. Fedcda: Federated learning with cross-rounds divergence-aware aggregation. In *The Twelfth International Conference on Learning Representations*, 2023.

[51] Kaibin Wang, Qiang He, Feifei Chen, Chunyang Chen, Faliang Huang, Hai Jin, and Yun Yang. Flexifed: Personalized federated learning for edge clients with heterogeneous model architectures. In *Proceedings of the ACM Web Conference 2023*, pages 2979–2990, 2023.

[52] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1):2032, 2022.

[53] Guile Wu and Shaogang Gong. Peer collaborative learning for online knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, 2021.

[54] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4133–4141, 2017.

[55] Jie Zhang, Chen Chen, Weiming Zhuang, and Lingjuan Lyu. Target: Federated class-continual learning via exemplar-free distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4782–4793, 2023.

[56] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10164–10173.

[57] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022.

[58] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pages 12878–12889. PMLR, 2021.

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: The summarized contributions are provided in the abstract and introduction.

   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: We indicated the limitations.

   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: [NA]

   Justification: We did not include new theorems.

   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: We could provide more details and codes to support the reproducibility.

   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: We could provide our code if required.

   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: We provided the settings.

   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: [Yes]

   Justification: The results are accompanied by error bars, confidence intervals, or statistical significance tests.

   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: We provide the amount of compute required for experiments.

   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 computing 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 reviewed the NeurIPS Code of Ethics.

   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: [NA]

    Justification: There is no societal impact of the work performed.

    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: The 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: We cited the original paper that produced the code package or dataset.

    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: [NA]

    Justification: The paper does not release new assets.

    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: The 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: The 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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We did not use LLM for the core method.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.