Bridging Model Heterogeneity in Federated Learning via Uncertainty-based Asymmetrical Reciprocity Learning

Jiaqi Wang [1] **Chenxu Zhao** [2] **Lingjuan Lyu** [3] **Quanzeng You** [4] **Mengdi Huai** [2] **Fenglong Ma** [1]

Abstract

This paper presents `FedType`, a simple yet pioneering framework designed to fill research gaps in heterogeneous model aggregation within federated learning (FL). `FedType` introduces small identical proxy models for clients, serving as agents for information exchange, ensuring model security, and achieving efficient communication simultaneously. To transfer knowledge between large private and small proxy models on clients, we propose a novel uncertainty-based asymmetrical reciprocity learning method, eliminating the need for any public data. Comprehensive experiments conducted on benchmark datasets demonstrate the efficacy and generalization ability of `FedType` across diverse settings. Our approach redefines federated learning paradigms by bridging model heterogeneity, eliminating reliance on public data, prioritizing client privacy, and reducing communication costs.

1. Introduction

Federated Learning (FL) is designed to enable the collaborative training of a machine learning model without the need to share clients' data. Many prevalent FL models, including FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020), mandate that clients employ an identical model structure and target for training a shared global model. However, clients may possess diverse model structures, introducing **model heterogeneity** within the FL framework. The goal of this challenging task is to learn personalized client models instead of a powerful global model.

Recently, many studies have emerged to tackle the challenge of model heterogeneity (Huang et al., 2022; Li & Wang, 2019; Yi et al., 2023; Lin et al., 2020; Yu et al., 2022; Wang et al., 2023). These efforts can be categorized into two groups based on the approach employed for information exchange between clients and the server. The first category focuses on the transmission of additional side information, such as logits (Huang et al., 2022), class scores (Li & Wang, 2019), and label-wise representations (Yi et al., 2023; Tan et al., 2022), which are derived from utilizing public data on individual clients. Conversely, the second category involves the direct upload of client models to the server for processes like distillation (Lin et al., 2020; Yu et al., 2022) or model reassembly (Wang et al., 2023) with the help of public data. While they successfully achieve the goal of heterogeneous model aggregation, they encounter significant drawbacks:

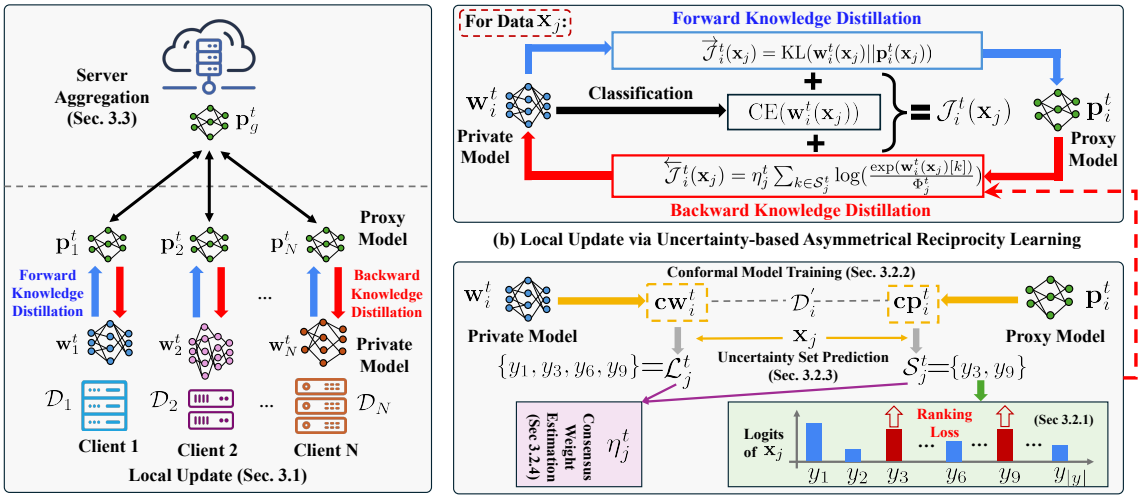
Diminishing returns in public data incorporation – As previously mentioned, prevailing approaches rely on extra public data, either at the client or server side, for conducting heterogeneous model aggregation. The latest study (Wang et al., 2023) underscores that the choice of public data plays a pivotal role in influencing model performance. Furthermore, using extra data amplifies the learning cost for models. To mitigate these challenges, a pertinent research question arises: *Can heterogeneous models be successfully aggregated without depending on any external public data?*

Disclosure risks raised by exchanging sensitive information – While exchanging side information between clients and the server can alleviate communication costs, this straightforward approach raises concerns about the potential disclosure of sensitive client information (Lyu et al., 2022). Furthermore, the complete upload of model structures and parameters poses substantial security risks (Tolpegin et al., 2020) and privacy concerns (Bouacida & Mohapatra, 2021), particularly for business corporations or entities operating in sensitive domains. In light of these challenges, another pivotal research question surfaces: *Can heterogeneous model aggregation be achieved while exchanging only non-sensitive information, mitigating the risks associated with data privacy and security?*

Necessity of efficient communication – Efficient communication stands out as a foundational challenge in federated learning. While numerous effective solutions have been devised for homogeneous federated learning (Diao et al.,

[1]Pennsylvania State University [2]Iowa State University [3]Sony AI [4]ByteDance. Correspondence to: Fenglong Ma <fenglong@psu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



(a) Overview of FedType

(b) Local Update via Uncertainty-based Asymmetrical Reciprocity Learning

(c) Uncertainty-based Behavior Imitation Learning (Sec. 3.2)

Figure 1. Overview of the proposed FedType framework. (**a**) demonstrates the workflow of the proposed FedType to address the model-heterogeneous issue in FL, (**b**) is the local update demonstration for a data sample $\mathbf{x}_j$ using the proposed uncertainty-based asymmetrical reciprocity learning, and (**c**) is the illustration of backward knowledge distillation with the proposed uncertainty-based behavior imitation learning.

2020; Yao et al., 2021; Cho et al., 2023), a notable research gap persists in heterogeneous federated learning. While the exchange of side information effectively reduces communication costs, as discussed previously, these approaches introduce vulnerabilities to FL systems. Hence, an urgent requirement arises for an efficient, secure solution that operates without relying on public data in heterogeneous FL.

To resolve the issues mentioned previously, this paper introduces a novel, powerful, yet straightforward framework, FedType, to bridge model heterogeneity gaps in **fed**erated learning without relying on any public data and ensuring efficient communication through uncertain**t**y-based as**y**mmetrical reci**p**rocity l**e**arning, as depicted in Figure 1. FedType achieves heterogeneous model aggregation, communication efficiency, and system security simultaneously.

FedType involves introducing small identical proxy models for clients, serving as agents for information exchange between clients and the server to facilitate heterogeneous model aggregation. In the local training phase, each client employs a novel bidirectional knowledge distillation strategy, referred to as **asymmetrical reciprocity learning**, to simultaneously update its large private model and small proxy model. This strategy leverages the distinctive characteristics of these two models, elaborated in Section 3.1. Additionally, an *uncertainty-based behavior imitation learning* method is developed to enhance the guidance provided by the small proxy model in facilitating the learning of the large client model, as detailed in Section 3.2. Ultimately, each local client is mandated to upload only its proxy model to the server. This facilitates model aggregation using existing approaches employed in homogeneous FL, as outlined in Section 3.3. We perform comprehensive experiments on multiple benchmark datasets, evaluating the FedType framework in both heterogeneous and homogeneous scenarios against state-of-the-art baselines. The experimental results consistently affirm the effectiveness of the FedType framework, underscoring its practicality and robust performance in real-world applications.

Remarkably, the FedType framework[1] presents several notable advantages in comparison to existing approaches. Firstly, it eliminates the necessity for relying on any public data across clients and the server, effectively mitigating the adverse effects associated with the initialization of public data in model-heterogeneous federated learning. Secondly, FedType adopts an identical yet compact proxy model to facilitate information exchange between clients and the server, resulting in a substantial reduction in communication costs. Thirdly, by utilizing proxy models, the framework ensures the privacy and security of client models, offering protection against potential harmful attacks. Fourthly, the proposed framework exhibits versatility and can be applied to both heterogeneous and homogeneous federated learning settings. Lastly, this flexible framework is compatible with any existing federated learning model.

2. Related Work

2.1. Heterogeneous Federated Learning

Most existing federated learning studies (McMahan et al., 2017; T Dinh et al., 2020; Zhang et al., 2022; Marfoq et al., 2022; Bao et al., 2023; Dennis et al., 2021; Zhou et al., 2022) have concentrated on the homogeneous setting, requiring

[1]The code is available at https://github.com/JackqqWang/FedType.

all clients to adopt an identical model structure. Recent research has begun exploring the heterogeneous setting, allowing for varying model structures across clients. Within this domain, the approaches include sub-model training (Alam et al., 2022), sparse model-adaption (Chen et al., 2023), and hypernetworks (Shamsian et al., 2021), which impose constraints on the relationship between the clients' models and the global model, thus still restricting the freedom of clients in using their preferred models.

To facilitate more flexible cooperation among heterogeneous models, several studies have explored alternatives to averaging model parameters, such as aggregating extra information like logits (Huang et al., 2022), class scores (Li & Wang, 2019), and label-wise representations (Yi et al., 2023; Tan et al., 2022). However, these methods potentially raise privacy concerns. In response, recent research has focused on exchanging model parameters through strategies like ensemble learning (Lin et al., 2020), mutual learning (Yu et al., 2022; Shen et al., 2023), or model reassembly (Wang et al., 2023).

These approaches exhibit two prevalent limitations: (1) Their dependence on public data. In real-world applications, obtaining access to public data may not always be feasible, and selecting suitable public data without preliminary knowledge of the client's data poses a complex challenge. (2) All the mentioned studies require sharing sensitive information, such as model structure, model parameters, or data-related insights from the private local models, with the server. This raises significant privacy concerns.

2.2. Conformal Prediction

Conformal prediction has seen significant popularity in recent years, particularly due to its capacity to generate prediction sets with guaranteed error rates under minimal assumptions. Initially introduced in (Vovk et al., 1999) and further elaborated in (Shafer & Vovk, 2008) and (Balasubramanian et al., 2014), conformal prediction offers a distribution-free uncertainty quantification technique that has been effectively applied in various applications (Angelopoulos & Bates, 2021; Bhatt et al., 2021; Fisch et al., 2021; Sankaranarayanan et al., 2022). Several recent studies focus on adapting conformal prediction within FL, specifically targeting the challenges associated with label shift (Plassier et al., 2023) and the quantification of uncertainty in distributed environments (Lu et al., 2023). To the best of our knowledge, there is limited work utilizing conformal prediction to enhance the performance of heterogeneous FL frameworks.

3. Methodology

As illustrated in Figure 1(a), `FedType` comprises two key components: local update and server update. During the t-th communication round, `FedType` initiates the training process for a proxy model $\mathbf{p}_i^t$ and a client private model $\mathbf{w}_i^t$ using the data $\mathcal{D}_i$ from the i-th client. It is essential to note that in model-heterogeneous federated learning, the structures of private models $\{\mathbf{w}_1^t, \cdots, \mathbf{w}_N^t\}$ differ from one another, where N represents the number of clients. Conversely, the structures of proxy models $\{\mathbf{p}_1^t, \cdots, \mathbf{p}_N^t\}$ are uniform. Furthermore, it is worth mentioning that the parameter size of proxy model $\mathbf{p}_i^t$ is typically much smaller than that of private model $\mathbf{w}_i^t$.

Only the learned proxy models $\{\mathbf{p}_1^t, \cdots, \mathbf{p}_N^t\}$[2] are transmitted to the server for model aggregation during the server update, leading to a significant reduction in communication costs. The resulting aggregated model $\mathbf{p}_g^t$ is then disseminated to each client in the subsequent communication round as the initialized proxy model, denoted as $\mathbf{p}_i^{t+1} = \mathbf{p}_g^t$ ($\forall i \in [1, \cdots, N]$). These two updates are executed iteratively until `FedType` converges. The whole algorithm flow can be found in Appendix A. Next, we provide the details of each update.

3.1. Local Update via Uncertainty-based Asymmetrical Reciprocity Learning

The inherent challenge posed by model heterogeneity renders the direct aggregation of uploaded client models unfeasible. Despite various proposed approaches for heterogeneous model aggregation (Lin et al., 2020; Yu et al., 2022; Huang et al., 2022; Li & Wang, 2019; Yi et al., 2023), as discussed in Section 1, they still exhibit several limitations. In contrast to existing work, we present a simple yet novel uncertainty-based asymmetrical reciprocity learning (`UARL`) approach to tackle the challenges posed by model heterogeneity. As shown in Figure 1 (c), `UARL` is a bidirectional knowledge distillation (KD)-based model.

The forward KD (FKD) follows the traditional "teacher-student" knowledge distillation approach. In FKD, a small proxy/student model $\mathbf{p}_i^t$ is distilled from the large, private client/teacher model $\mathbf{w}_i^t$ using the client data $\mathcal{D}_i$ for the i-th client during the t-th communication round. The forward loss can be formulated as follows:

$$\overrightarrow{\mathcal{J}}_i^t = \sum_{j=1}^{|\mathcal{D}_i|} \mathrm{KL}(\mathbf{w}_i^t(\mathbf{x}_j) || \mathbf{p}_i^t(\mathbf{x}_j)), \tag{1}$$

where $|\mathcal{D}_i|$ denotes the number of data stored in the i-th client, $\mathrm{KL}(\cdot, \cdot)$ is the Kullback–Leibler divergence, and $\mathbf{x}_j \in \mathcal{D}_i$ represents the input data.

The backward KD (BKD) poses a challenge, as the capability of the small proxy model $\mathbf{p}_i^t$ is typically weaker than that of the large client model $\mathbf{w}_i^t$. Directly applying traditional knowledge distillation may lead to a degradation in

[2] In the cross-device setting, we will randomly select $B \ll N$ clients at each communication round.

Algorithm 1: Epoch-Level Algorithm Flow of UARL.

Input: Client training data $\mathcal{D}_i$, private model $\mathbf{w}_i^{t-1}$, proxy model initialized by $\mathbf{p}_g^{t-1}$, validation data $\mathcal{D}_i'$, local training epoch R, hyperparameters

Divide the shuffled $\mathcal{D}_i$ into R parts $\{\mathcal{D}_i^1, \cdots, \mathcal{D}_i^R\}$;
Initialize epoch-level models: $\mathbf{w}_i^0 = \mathbf{w}_i^{t-1}$ and $\mathbf{p}_i^0 = \mathbf{p}_g^{t-1}$;
for each epoch $r = 1, \cdots, R$ **do**
  Train the conformal prediction models $\mathbf{cw}_i^r$ and $\mathbf{cp}_i^r$ using the learned epoch-level models $\mathbf{w}_i^{r-1}$ and $\mathbf{p}_i^{r-1}$ with the validation data $\mathcal{D}_i'$ according to Eq. (5);
  Initialize the loss $\mathcal{J}_i^r = 0$;
  for each sample $\mathbf{x}_j \in \mathcal{D}_i^r$ **do**
    Calculate the classification loss via $\mathrm{CE}(\mathbf{w}_i^{r-1}(\mathbf{x}_j), \mathbf{y}_j)$;
    Calculate the FKD loss $\overrightarrow{\mathcal{J}}_i^r(\mathbf{x}_j)$ via Eq. (1);
    Calcuate the performance change Δ^r on the proxy model on $\mathcal{D}_i'$;
    Obtain the prediction set $\mathcal{S}_j^r$ via Eq. (6) using $\mathbf{cp}_i^r$;
    Obtain the prediction set $\mathcal{L}_j^r$ via Eq. (6) using $\mathbf{cw}_i^r$;
    Calculate η_j^r according to Eq. (8);
    Calculate the BKD loss $\overleftarrow{\mathcal{J}}_i^r(\mathbf{x}_j)$;
    $\mathcal{J}_i^r$ += $\mathrm{CE}(\mathbf{w}_i^{r-1}(\mathbf{x}_j), \mathbf{y}_j) + \overrightarrow{\mathcal{J}}_i^r(\mathbf{x}_j) + \overleftarrow{\mathcal{J}}_i^r(\mathbf{x}_j)$;
  end
  Update the models $\mathbf{w}_i^r$ and $\mathbf{p}_i^r$ by optimizing $\mathcal{J}_i^r$;
end

Return: Trained models $\mathbf{w}_i^t$ and $\mathbf{p}_i^t$.

the power of client models. To overcome this asymmetrical reciprocity issue, we introduce a novel uncertainty-based behavior imitation learning method to transfer diverse knowledge from the proxy model $\mathbf{p}_i^t$ to the client model $\mathbf{w}_i^t$. Further details can be found in Section 3.2.

Let $\overleftarrow{\mathcal{J}}_i^t$ denote the backward KD loss. We then use the following loss function to train the i-th client at the t-th communication round:

$$\mathcal{J}_i^t = \sum_{j=1}^{|\mathcal{D}_i|} \mathrm{CE}(\mathbf{w}_i^t(\mathbf{x}_j), \mathbf{y}_j) + \overrightarrow{\mathcal{J}}_i^t + \overleftarrow{\mathcal{J}}_i^t, \tag{2}$$

where $\mathbf{y}_j$ denotes the vectorized ground truth obtained from the real label set $\mathcal{Y}_j$ of $\mathbf{x}_j$. The algorithm flow can be found in Algorithm 1.

3.2. Uncertainty-based Behavior Imitation Learning

The initialization of the proxy model $\mathbf{p}_i^t$ is carried out using the aggregated global model $\mathbf{p}_g^{t-1}$ (refer to Section 3.3). This global model encapsulates diverse knowledge from other clients, making it crucial to transfer this knowledge to the private client model. However, due to the inherently weaker capability of the proxy model $\mathbf{p}_i^t$, direct application of the traditional "teacher-student" learning paradigm for knowledge transfer to the large model $\mathbf{w}_i^t$ is impractical. Such a direct approach may introduce additional noise to the large model, potentially impeding the overall training efficiency of the entire framework.

3.2.1. Backward Knowledge Distillation Loss

To tackle this challenge, we introduce an uncertainty-based behavior imitation learning approach. This method exclusively relies on the use of *partial logits* generated by the proxy model $\mathbf{p}_i^t$ with a high level of confidence or certainty. The intention is to employ these confident predictions to guide the learning process of the large model $\mathbf{w}_i^t$. In essence, when dealing with a specific data sample $\mathbf{x}_j \in \mathcal{D}_i$, if the proxy model $\mathbf{p}_i^t$ exhibits high confidence in predicting certain classes, transferring this behavioral information—rather than the complete logits—can still be beneficial for the training of the large model.

Let $\mathcal{S}_j^t$ represent the set of class labels predicted with high confidence by $\mathbf{p}_i^t$. In our proposed behavior imitation learning, the objective is to strengthen the probability/logit associated with the labels in $\mathcal{S}_j^t$ as predicted by the large model $\mathbf{w}_i^t$. Importantly, we refrain from imposing a strict requirement for the large model $\mathbf{w}_i^t$ to prioritize ranking the labels in $\mathcal{S}_j^t$ at the top positions. To address this, we introduce a novel ranking-based behavior imitation learning loss, outlined as follows:

$$\begin{aligned} \overleftarrow{\mathcal{J}}_i^t &= \sum_{j=1}^{|\mathcal{D}_i|} \eta_j^t \sum_{k \in \mathcal{S}_j^t} \log\left(\frac{\exp(\mathbf{w}_i^t(\mathbf{x}_j)[k])}{\Phi_j^t}\right), \\ \Phi_j^t &= \sum_{s \in \mathcal{S}_j^t} \exp(\mathbf{w}_i^t(\mathbf{x}_j)[s]) + \sum_{v \in \mathcal{V}_j^t} \exp(\mathbf{w}_i^t(\mathbf{x}_j)[v]), \end{aligned} \tag{3}$$

where η_j^t is the estimated consensus weight to determine the amount of the transferred knowledge from the proxy model $\mathbf{p}_i^t$ to the large model $\mathbf{w}_i^t$. $\mathcal{V}_j^t = \mathcal{Y} - \mathcal{S}_j^t$ denotes the class labels associated with low confidence, and $\mathcal{Y}$ represents the complete set of class labels in $\mathcal{D}_i$. In Eq. (3), η_j^t and $\mathcal{S}_j^t$ are unknown variables. To proceed, we introduce a novel approach to quantify an uncertainty set $\mathcal{S}_j^t$ for each data $\mathbf{x}_j$ through dynamic conformal prediction, and the uncertainty sets will be further used to estimate η_j^t.

3.2.2. Conformal Model Training

Conformal prediction (Angelopoulos & Bates, 2021) stands out as a reliable and interpretable approach for quantifying uncertainty, providing prediction sets accompanied by a designated level of confidence or probability. In mathematical terms, conformal prediction involves an uncertainty set function $f(\mathbf{p}_j^t, \mathbf{x}_j)$ that maps $\mathbf{x}_j$ to a subset of $\mathcal{Y}$ (i.e.,

$f(\mathbf{p}_j^t, \mathbf{x}_j) = \mathcal{S}_j^t \subseteq \mathcal{Y})$, satisfying the condition:

$$P(\mathcal{Y}_j \in \mathcal{S}_j^t)) \geq 1 - \theta, \tag{4}$$

where θ represents a predefined confidence level.

To estimate the prediction set $\mathcal{S}_j^t$, we need to train a conformal model $\mathbf{cp}_j^t$ first using the validation dataset $\mathcal{D}_i'$ where $\mathcal{D}_i \cap \mathcal{D}_i' = \emptyset$ as follows:

$$\mathbf{cp}_i^t = \text{Cmodel}(\mathbf{p}_i^t, \mathcal{D}_i'), \tag{5}$$

where Cmodel() is constructed using the split conformal prediction framework, detailed in Appendix B.

Existing conformal prediction approaches are principally developed to quantify uncertainty in static, well-trained models and are not tailored to address dynamic scenarios. However, the `FedType` framework undergoes iterative training, resulting in dynamic changes to the models $\mathbf{p}_i^t$ and $\mathbf{w}_i^t$ at each communication round and even each epoch r.[3] Consequently, the uncertainty set $\mathcal{S}_j^t$ is predicted dynamically. Next, we will use the trained conformal model $\mathbf{cp}_i^t$ to generate the prediction set $\mathcal{S}_j^t$ for each training data $\mathbf{x}_j \in \mathcal{D}_i$.

3.2.3. Uncertainty Set Prediction

A straightforward approach would involve directly applying existing conformal prediction methods, such as regularized adaptive prediction sets (RAPS) (Angelopoulos et al., 2020), to generate the uncertainty set $\mathcal{S}_j^t$ for $\mathbf{x}_j$ at each communication round t. However, as mentioned earlier, the capability of the proxy model $\mathbf{p}_i^t$ is weak, especially at the initial stages of training. Utilizing prediction sets from such unreliable models may introduce adverse effects when training Eq. (3).

To address this challenge, we introduce a dynamic adjustment mechanism for the size of the prediction set $\mathcal{S}_j^t$ based on the observed changes in model performance on the training data $\mathcal{D}_i$. In essence, a decrease in a model's performance may be indicative of low-quality prediction sets. Therefore, to enhance the informativeness of the prediction sets, it is necessary to reduce their size by refining the conformal prediction algorithm. Based on this intuition, we propose a dynamic conformal prediction for federated learning training based on RAPS using the following uncertainty prediction set generation:

$$\begin{aligned}\mathcal{S}_j^t := \{y : \pi_j(y) \cdot u + \rho_j(y) \\ + g(\Delta^t, \lambda) \cdot (o_j(y) - \kappa_{reg})^+ \leq \tau\}.\end{aligned} \tag{6}$$

$\pi_j(y)$ represents the probability assigned to the label y as predicted by the conformal model $\mathbf{cp}_i^t$ for the j-th data point $\mathbf{x}_j$. The parameter u is a predefined randomized factor determining the value jump for each new label y. Furthermore, $\rho_j(y) = \sum_{y'=1}^{|\mathcal{Y}|} \pi_j(y') \mathbb{1}_{\{\pi_j(y') > \pi_j(y)\}}$ denotes the total probability mass associated with the set of labels that are more likely than the label y.

Besides, $g(\Delta^t, \lambda)$ is a piecewise calibration function[4], which is defined as follows:

$$g(\Delta^t, \lambda) = \begin{cases} \lambda \cdot \Delta^t - \Delta^t + \lambda, & \text{if } \Delta^t < 0, \\ \lambda, & \text{otherwise.} \end{cases} \tag{7}$$

$\Delta^t = \mathcal{A}(\mathbf{p}_i^t, \mathcal{D}_i') - \mathcal{A}(\mathbf{p}_i^{t-1}, \mathcal{D}_i')$, where $\mathcal{A}(\mathbf{p}_i^t, \mathcal{D}_i')$ denotes the validation accuracy on $\mathcal{D}_i'$ using the trained model $\mathbf{p}_i^t$. In the event of a performance drop, i.e., $\Delta^t < 0$, we actively reduce the size of the prediction set by adjusting the value of $g(\Delta^t, \lambda)$. On the other hand, if $\Delta^t \geq 0$, we maintain $g(\Delta^t, \lambda) = \lambda$, where $\lambda \geq 0$ serves as a predefined regularization hyperparameter.

In addition, $(z)^+$ denotes the positive part of z, and κ_{reg} is another regularization hyperparameter. The variable $o_j(y) = |y' \in \mathcal{Y} : \pi_j(y') \geq \pi_j(y)|$ represents the ranking of y among the labels in $\mathcal{Y}$ based on the prediction probability π_j. Finally, τ denotes the cumulative sum of the sorted, penalized classifier scores. The details of the designed dynamic conformal prediction can be found in Appendix B.

3.2.4. Consensus Weight Estimation

In Eq. (3), η_j^t is a key factor to control the behavior of knowledge transfer from the proxy model $\mathbf{p}_i^t$ to the large model $\mathbf{w}_i^t$. Intuitively, if the proxy model $\mathbf{p}_i^t$ is significantly confident on certain predicted labels on data $\mathbf{x}_j$ i.e., the small size of $\mathcal{S}_j^t$, then it is necessary to transfer such high-quality knowledge to the large model $\mathbf{w}_i^t$ as much as possible, as the guidance of model training. On the other hand, we need to reduce the amount of transferred knowledge with low quality, i.e., the uncertainty set $\mathcal{S}_j^t$ with large size.

The quality of knowledge is a relative variable, which can be determined by the size of uncertainty sets predicted by both models $\mathbf{p}_i^t$ and $\mathbf{w}_i^t$ on $\mathbf{x}_j$ using Eq. (6) with the conformal models $\mathbf{cp}_i^t$ and $\mathbf{cw}_i^t$ learned by Eq. (5), denoted as $\mathcal{S}_j^t$ and $\mathcal{L}_j^t$, respectively. Mathematically, we define the consensus weight η_j^t as follows:

$$\eta_j^t = \begin{cases} |\mathcal{S}_j^t \cap \mathcal{L}_j^t| / |\mathcal{S}_j^t \cup \mathcal{L}_j^t|, & \text{if } |\mathcal{S}_j^t| \geq |\mathcal{L}_j^t|, \\ |\mathcal{S}_j^t \cap \mathcal{L}_j^t| / |\mathcal{S}_j^t|, & \text{if } |\mathcal{S}_j^t| < |\mathcal{L}_j^t|, \end{cases} \tag{8}$$

where $|\cdot|$ denotes the size of the set. We can observe

[3] In our implementation, the parameters of two models change at each epoch r during each communication round t in the training stage. Here, we omit the notation of epoch r in the rest of this section for simplicity and readability, which can be treated as $r = 1$ in each communication round t.

[4] Here, we define a simple linear function to characterize the dynamic adjustment of prediction sets. We can also use other monotonically decreasing functions as alternative solutions.

that such a design encourages the proxy model to transfer confident knowledge to the large model.

3.3. Server Update

After training each client using Eq. (2), we will obtain both proxy and private models. The proxy models $\{\mathbf{p}_1^t, \cdots, \mathbf{p}_N^t\}$ will be uploaded to the server to conduct aggregation, resulting in a share proxy global model $\mathbf{p}_g^t$ using any existing data-free aggregation approaches, such as FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020). The global model $\mathbf{p}_g^t$ will be distributed to each client at the next communication round as the initialization of the proxy client model, i.e., $\mathbf{p}_i^{t+1} = \mathbf{p}_g^t$. The proposed `FedType` framework will be iteratively executed for the client update and server update until convergence.

4. Experiments

4.1. Experimental Setups

Data preparation. We assess the effectiveness of the proposed `FedType` approach through image classification tasks conducted in the cross-device scenario on FMNIST, CIFAR-10, and CIFAR-100 datasets, and cross-silo scenario on Fed-ISIC19 dataset (Ogier du Terrail et al., 2022). For the cross-device experiments, we follow existing work (Yurochkin et al., 2019; Hsu et al., 2019) to set heterogeneity degrees by adjusting the Dirichlet distribution's concentration parameter α. We set $\alpha = 1, 0.5, 0.1$, respectively in our experiments. The details of the data partition can be found in Appendix C.

Baselines. In our scenario, clients employ distinct network structures, and we abstain from using public data. FML (Shen et al., 2023) is the only baseline with the same setting as ours, which uses bidirectional knowledge distillation to learn the model. Besides, we employ FedProto (Tan et al., 2022) as another baseline, which aggregates class prototypes instead of model parameters.

The outcomes of `FedType` consist of three models: a shared global model ($\texttt{FedType}_{\text{global}} = \mathbf{p}_g^T$), a set of proxy client models ($\texttt{FedType}_{\text{proxy}} = \{\mathbf{p}_1^T, \cdots, \mathbf{p}_N^T\}$), and a set of private client models ($\texttt{FedType}_{\text{private}} = \{\mathbf{w}_1^T, \cdots, \mathbf{w}_N^T\}$), where T represents the total number of communication rounds. Subsequently, we compare the average client accuracy across these three types of models. Moreover, the inherent flexibility of model aggregation in `FedType` enables us to assess various representative aggregation approaches, including FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), pFedMe (T Dinh et al., 2020), and pFedBayes (Zhang et al., 2022). More details of the baselines can be found in Appendix D.

Impletation details. To replicate the model-heterogeneous scenario, we assemble a model pool comprising private client models, encompassing ResNet-18, ResNet-34, ResNet-50, RestNet-101, ResNet-152, VGG-11, VGG-13, VGG-16, and VGG-19, which will be randomly assigned to a client. In our primary experiments, we designate ResNet-18 as the small proxy model, as presented in Table 1. Additionally, we delve into alternative options for proxy model selection, detailed in Table 3. Our reported metric is the **average accuracy** on *100 clients for the cross-device setting and 6 clients for the cross-silo setting*. More details about the model and hyperparameters can be found in Appendix E and F, respectively.

4.2. Results of the Heterogeneous Model Setting

4.2.1. Cross-device Evaluation

Table 1 presents the average client accuracy across three image datasets, employing different aggregation methods under varying label heterogeneity distributions (α's). Observing the results, it is evident that the proposed $\texttt{FedType}_{\text{private}}$ consistently outperforms both the shared global model $\texttt{FedType}_{\text{global}}$ and the learned proxy model $\texttt{FedType}_{\text{proxy}}$ across diverse aggregation methods.

Particularly noteworthy is the global model's lower performance, especially in challenging tasks. This aligns with our design, where the shared global model primarily serves as an agent for information exchange in personalized federated learning. Despite sharing the same network structure as the global model, the proxy models, through uncertainty-based asymmetrical reciprocity learning, acquire valuable knowledge, contributing to their enhanced performance. Furthermore, comparing different aggregation methods reveals that personalized approaches (pFedMe and pFedBayes) exhibit superior performance compared to general methods (FedAvg and FedProx), aligning with our expectations.

Notably, an increase in the value of α corresponds to an overall performance improvement for $\texttt{FedType}_{\text{proxy}}$ and $\texttt{FedType}_{\text{private}}$. This observation aligns with our data partition method, where training and testing data follow the same distribution. A larger α increases label categories for each client, rendering the classification task more challenging. However, the performance of $\texttt{FedType}_{\text{global}}$ contradicts this trend. This is attributed to the shared global model serving as an average representation of all proxy client models, performing better in scenarios where data follows an independent and identical distribution, such as a large α.

4.2.2. Cross-silo Evaluation

An additional experiment is conducted to assess the effectiveness of the proposed `FedType` under the cross-silo setting, utilizing the Fed-ISIC19 dataset. The results are illustrated in Figure 2, where x-axis denotes the aggregation method, and y-axis is the average client accuracy. Similar observations to those in Table 1 emerge, where the private

Table 1. Performance (%) comparison under the heterogeneous cross-device settings.

Agg. Method	Model	FMNIST			CIFAR-10			CIFAR-100		
		$\alpha=1$	$\alpha=0.5$	$\alpha=0.1$	$\alpha=1$	$\alpha=0.5$	$\alpha=0.1$	$\alpha=1$	$\alpha=0.5$	$\alpha=0.1$
✗	FedProto	85.05	87.66	89.04	76.59	78.17	82.96	58.03	66.31	68.60
✗	FML	86.54	90.71	92.63	80.80	85.24	88.58	58.77	66.90	68.74
FedAvg	$\texttt{FedType}_{\text{global}}$	84.11	83.93	81.32	66.40	63.39	58.17	38.36	38.17	35.45
	$\texttt{FedType}_{\text{proxy}}$	86.09	89.45	93.16	80.65	82.57	85.04	56.24	61.06	62.31
	$\texttt{FedType}_{\text{private}}$	**87.26**	**91.22**	**94.77**	**82.56**	**86.83**	**91.90**	**57.33**	**65.69**	**68.14**
FedProx	$\texttt{FedType}_{\text{global}}$	86.96	86.44	84.29	68.26	65.86	63.75	41.88	39.31	36.53
	$\texttt{FedType}_{\text{proxy}}$	87.03	91.50	92.64	82.19	82.48	87.80	58.56	61.22	62.64
	$\texttt{FedType}_{\text{private}}$	**87.65**	**93.84**	**94.98**	**83.69**	**86.92**	**92.03**	**59.18**	**65.45**	**68.37**
pFedMe	$\texttt{FedType}_{\text{global}}$	87.82	87.13	85.86	68.71	65.22	64.95	41.55	40.92	38.60
	$\texttt{FedType}_{\text{proxy}}$	88.63	92.05	93.38	82.64	83.00	88.14	59.04	62.68	64.89
	$\texttt{FedType}_{\text{private}}$	**88.96**	**92.36**	**94.86**	**83.47**	**87.24**	**92.16**	**59.78**	**67.07**	**69.51**
pFedBayes	$\texttt{FedType}_{\text{global}}$	88.20	87.85	86.04	68.41	66.87	63.32	43.73	41.24	38.72
	$\texttt{FedType}_{\text{proxy}}$	89.69	92.11	93.29	83.33	84.49	89.10	59.47	62.96	63.51
	$\texttt{FedType}_{\text{private}}$	**90.26**	**93.17**	**95.88**	**84.09**	**88.67**	**92.38**	**59.62**	**67.35**	**69.60**

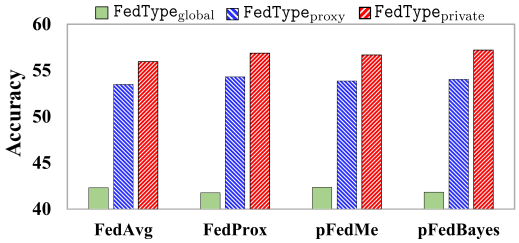


Figure 2. Results (%) of the cross-silo evaluation.

models outperform the proxy models, which, in turn, surpass the global models. This consistent trend validates the efficacy of the proposed uncertainty-based asymmetrical reciprocity learning in mutually enhancing the capabilities of proxy and private models.

4.2.3. Ablation Study

We compare our model with the following variants of FedType (i.e., $\texttt{FedType}_{\text{private}}$) in this ablation study: (1) $\texttt{FedType}_{\text{sym}}$: We replace the backward loss $\overleftarrow{\mathcal{J}}_i^t$ with traditional knowledge distillation loss in Eq. (2), which can be treated as symmetrical reciprocity learning. (2) $\texttt{FedType}_{\text{Top}K}$: We choose top K labels based on the ranking of the logits to construct the label set rather than using the proposed conformal prediction set estimation in the Section 3.2.3. Here, we set $K = 3$. (3) $\texttt{FedType}_{\eta=1}$: We simply set the consensus weight $\eta_j^t = 1$ in Eq. (8). (4) $\texttt{FedType}_{g=0.5}$: We simply set the function $g(\Delta^t, \lambda) = \lambda = 0.5$ in Eq. (6). The ablation study results are shown in Table 2. The cross-device scenario is validated on the CIFAR-10 dataset by setting $\alpha = 0.5$ and the number of clients as 100. The aggregation method on the server is FegAvg for both cross-device and cross-silo scenarios.

Observing Table 2 reveals that each component effectively enhances performance but to varying degrees. Firstly, $\texttt{FedType}_{\text{sym}}$ exhibits the least favorable performance among all baselines, underscoring the imperative need for considering asymmetrical knowledge distillation. Secondly, the outcomes of $\texttt{FedType}_{\text{Top}K}$ suggest the effectiveness of asymmetrical reciprocity learning. However, opting for an arbitrarily fixed number of top ranks proves suboptimal. Therefore, it becomes essential to dynamically determine the ranks for each sample. Thirdly, the results of $\texttt{FedType}_{\eta=1}$ and $\texttt{FedType}_{g=0.5}$ demonstrate that employing uncertainty set prediction contributes to a performance increase compared with $\texttt{FedType}_{\text{Top}K}$. In conclusion, considering asymmetrical reciprocity learning, dynamically adjusting the prediction set based on the model training performance, and estimating the consensus weights of samples prove to be valuable strategies for enhancing overall performance.

Table 2. Ablation study performance (%) comparison.

Dataset	CIFAR-10	Fed-ISIC19
$\texttt{FedType}_{\text{sym}}$	82.24	49.89
$\texttt{FedType}_{\text{Top}K}$	84.69	51.26
$\texttt{FedType}_{\eta=1}$	85.12	52.40
$\texttt{FedType}_{g=0.5}$	86.24	54.17
FedType	**86.83**	**55.95**

4.2.4. Dynamic Conformal Prediction

In Eq. (6), we introduce a calibration function $g(\Delta^t, \lambda)$ to regulate the size of prediction sets based on the model's performance change. In this experiment, our objective is to examine the impact of selecting the function $g(\Delta^t, \lambda)$ when $\Delta^t \in [-1, 0]$. Let $g_1 = g(\Delta^t, \lambda) = \lambda \cdot \Delta^t - \Delta^t + \lambda$ and $g_2 = g(\Delta^t, \lambda) = \lambda$ ($\forall \Delta^t \in [-1, 1]$). We also consider two alternative quadratic functions: $g_3 = g(\Delta^t, \lambda) = \lambda \cdot {\Delta^t}^2 + \lambda$ and $g_4 = g(\Delta^t, \lambda) = -\lambda \cdot {\Delta^t}^2 - \Delta^t + \lambda$. When $\Delta^t \in (0, 1]$, $g(\Delta^t, \lambda) = \lambda$ for all functions with $\lambda = 0.5$.

Figure 3 (a) and (b) illustrate the g function's representation and its impact on private models respectively. Analyzing the experimental outcomes, we note the following observations: (1) The performance with functions g_1, g_3, or g_4 consis-

Table 3. Proxy model study. The approximate model sizes are shown using the model parameters (in millions).

Proxy Model	Parameter Size	FMNIST			CIFAR-10			CIFAR-100			Fed-ISIC19
		$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.1$	✗
ShuffleNet-V2	2.27M	87.11	88.86	91.58	81.25	81.79	84.48	50.44	57.17	60.35	47.48
MobileNet-V1	3.21M	87.59	88.93	91.13	81.04	82.95	85.06	51.31	59.30	62.76	50.32
EfficientNet-B0	5.29M	**88.43**	89.55	92.14	82.13	83.76	87.02	54.12	61.89	63.61	53.13
ResNet-18	11.17M	87.26	**91.22**	**94.77**	**82.56**	**86.83**	**91.90**	**57.33**	**65.69**	**68.14**	**55.95**

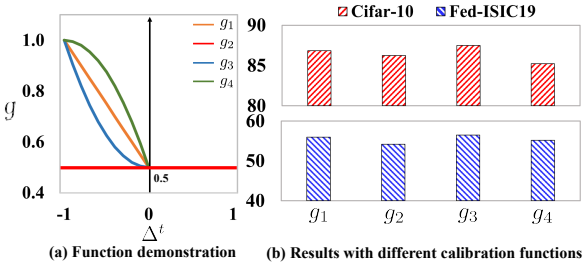


Figure 3. Calibration function $g(\Delta^t, \lambda)$ study.

tently surpasses that of g_2. This suggests that adjusting the value of λ in response to accuracy fluctuations positively influences performance on both the Cifar-10 and Fed-ISIC19 datasets. (2) Among g_1, g_3, and g_4, g_3 exhibits superior performance on these datasets. A potential explanation is its steeper slope near a Δ^t value of -1, aiding $g(\Delta^t, \lambda)$ in rapidly declining to identify a more optimal prediction set for the subsequent training epoch. (3) While outcomes with different g function configurations vary, they remain within a stable range, indicating the robustness and adaptability of our proposed `FedType` to the selection of the g function.

4.2.5. Alternative Proxy Model Selection

The proxy model used in all the previous experiments is ResNet-18. To examine the impact of proxy model selection on performance, we select four widely-used compact models as proxies, including MobileNet-V1 (Howard et al., 2017), ShuffleNet-V2 (Ma et al., 2018), and EfficientNet-B0 (Tan & Le, 2019). Model details can be found in Appendix E. The results are shown in Table 3, where the aggregation method is still FedAvg. We can observe that the performance variation with these different proxy models generally aligns with their respective sizes and capabilities, which aligns with our expectations. This study of proxy models underscores the resilience of our approach to variations in proxy model choice and further affirms the generalizability of our proposed framework.

4.2.6. Client Model Architecture Analysis

In our experiments, we use a mixed combination of client models. To explore the relationship between the model architecture difference and the framework performance, we conduct the following experiment with $\alpha = 0.5$, and the other settings are the same as the experiments shown in Table 1. In this experiment, we fixed the proxy model, which is ResNet-18, but used different private models. `FedType`-ResNet denotes all the private client models selected from the ResNet family pool, including ResNet-18, ResNet-34, ResNet-50, RestNet-101, and ResNet 152. `FedType`-VGG denotes all the private models belonging to the VGG family, including VGG-11, VGG-13, VGG-16, and VGG-19. `FedType`-Mix is used in the main experiments, using the mixed ResNet and VGG client models.

We report the results in Table 4. We have several exciting observations. First, the private model performs better than the proxy model, which is better than the global model. This observation is the same as we discussed in the main results shown in Table 1. Second, `FedType`-ResNet outperforms `FedType`-Mix, which further outperforms `FedType`-VGG. The observation demonstrates that if the proxy and private models have similar model architectures, the proposed bidirectional knowledge distillation performs the best.

4.2.7. Upper-bound Performance Exploration

In this experiment, we aim to investigate the performance upper bound, which will be obtained by training the federated learning framework in the homogeneous setting, where each client uses the largest model (i.e., VGG-19 in our experiments). Since the homogeneous setting aims to train a shared global model, we report the average performance tested on each client with the learned global model from **Homo-VGG-19** in Table 5, compared with the performance of private models obtained by our proposed `FedType` with $\alpha = 0.5$. To maximize the performance of the private model with our setting, we also test another model, named **VGG-19+ResNet-18**, in which all clients used VGG-19 as the private model and ResNet-18 as the proxy model.

We can observe that the upper bound performance obtained by the Homo-VGG-19 model performs the best, and VGG-19+ResNet-18 is better than `FedType` but with marginal improvements, which aligns with our expectations. However, our model and VGG-19+ResNet-18 use ResNet-18 (with 11.17 million parameters) as the proxy model to exchange information between clients and the server. However, the Homo-VGG-19 model has 144 million parameters, which is 12.9X of our model. Considering the communication costs, `FedType` is an effective solution to address the

Table 4. Performamce (%) of different client model combinations, where all clients' personal models are from ResNet or VGG family.

Dataset	Model	FedAvg			FedProx			pFedMe			pFedBayes		
		global	proxy	private	global	proxy	private	global	proxy	private	global	proxy	private
FMNIST	FedType-ResNet	84.55	89.63	91.12	85.51	91.86	93.61	87.41	93.55	93.64	87.62	92.33	94.86
	FedType-Mix	83.93	89.45	91.22	86.44	91.50	93.84	87.13	92.05	92.36	87.85	92.11	93.17
	FedType-VGG	83.22	88.87	90.86	84.31	91.07	92.46	86.65	91.77	92.20	87.04	91.98	92.46
CIFAR-10	FedType-ResNet	64.59	83.62	86.88	66.11	83.79	87.25	66.89	84.08	87.10	67.03	84.55	88.96
	FedType-Mix	63.39	82.57	86.83	65.86	82.48	86.92	65.22	83.00	87.24	66.87	84.49	88.67
	FedType-VGG	61.22	80.61	83.07	64.95	80.87	84.14	63.40	81.36	84.52	65.48	81.27	86.11
CIFAR-100	FedType-ResNet	38.95	62.48	65.53	40.66	63.07	67.89	41.56	62.97	68.16	41.82	63.57	67.94
	FedType-Mix	38.17	61.06	65.69	39.31	61.22	65.45	40.92	62.68	67.07	41.24	62.96	67.35
	FedType-VGG	35.96	58.74	62.41	36.77	58.91	62.88	39.52	59.61	63.05	40.08	61.63	63.85
FedISIC-19	FedType-ResNet	42.86	53.55	55.84	42.84	55.51	56.90	42.21	53.88	57.03	42.37	55.22	57.10
	FedType-Mix	42.30	53.48	55.95	41.77	54.30	56.86	42.35	53.85	56.68	41.82	54.04	57.21
	FedType-VGG	40.76	51.06	52.90	42.04	51.98	53.69	42.13	52.61	53.12	41.01	52.48	54.55

Table 5. Performance upper-bound analysis.

Dataset	Model	FedAvg	FedProx	pFedMe	pFedBayes
FMNIST	Homo-VGG-19	92.86	94.05	94.26	94.87
	VGG-19+ResNet-18	91.35	94.12	93.45	94.33
	FedType	91.22	93.84	92.36	93.17
CIFAR-10	Homo-VGG-19	87.89	88.62	89.10	89.92
	VGG-19+ResNet-18	87.04	87.80	87.98	89.55
	FedType	86.83	86.92	87.24	88.67
CIFAR-100	Homo-VGG-19	65.91	67.52	69.81	69.98
	VGG-19+ResNet-18	65.77	66.71	67.89	68.89
	FedType	65.69	65.45	67.07	67.35
FedISIC-19	Homo-VGG-19	56.89	58.94	59.06	59.22
	VGG-19+ResNet-18	56.62	57.12	57.59	57.96
	FedType	55.95	56.86	56.68	57.21

model heterogeneity challenge in federated learning.

4.2.8. Scalability Analysis

In this experiment, we validate the scalability of the proposed FedType with a different number of clients on the CIRFA-10 dataset with a sample rate 10% and $\alpha = 0.5$. The client model pool contains ResNet-18, ResNet-34, and ResNet-50. The proxy model used in this experiment is MobileNet-V1, and the aggregation method is FedAvg. Table 6 presents our experimental findings with client counts of 50, 100, and 300. A noticeable trend is the marginal decline in performance as the number of clients rises. This outcome is attributable to the distribution of the same total volume of training data among a greater number of clients, resulting in reduced data availability per client. Such a scenario leads to a performance dip within a specific range. These results effectively illustrate the scalability of our proposed FedType, highlighting its adaptability to varying client numbers in federated learning environments.

4.3. Results of the Homogeneous Model Setting

While the primary focus of the proposed FedType is addressing the challenge of model heterogeneity, it showcases versatility by extending its application to the model homogeneous setting. To illustrate its generalization ability, we homogenize all clients to utilize the same model structure, VGG-11, with FedType employing ResNet-18 as the proxy model in this experiment. A comparison is made with two representative federated learning approaches, FedAvg and pFedMe, and the results are outlined in Figure 4.3. Notably, FedType, with fewer parameters (only around 8% of baselines' parameters), achieves comparable performance with existing homogeneous federated learning models. This underscores the effectiveness and advantages of FedType, even in a homogeneous model setting. ***More experimental results can be found in Appendix Sections G to M.***

Table 6. scalability study in the cross-device setting.

Setting	50	100	300
$\texttt{FedType}_{\text{global}}$	62.88	59.40	56.01
$\texttt{FedType}_{\text{proxy}}$	84.33	81.46	77.90
$\texttt{FedType}_{\text{private}}$	86.74	84.26	80.15

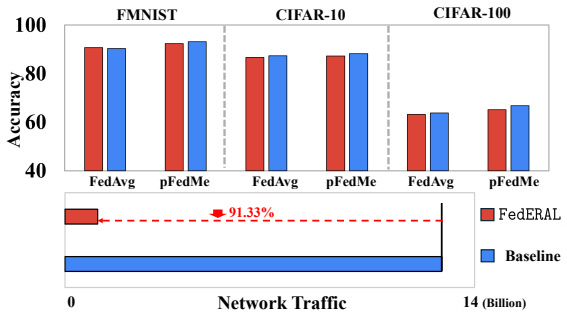


Figure 4. Homogenenous evaluation.

5. Conclusion

We have designed FedType to effectively tackle the challenges of model heterogeneity without relying on public data. By implementing a novel uncertainty-based asymmetrical reciprocity learning method, we have not only demonstrated the feasibility but also showed the superiority of FedType in handling diverse model structures while safeguarding client privacy and minimizing communication costs. Our comprehensive experiments across multiple benchmark datasets illustrate the effectiveness of FedType in different scenarios. We believe that FedType significantly contributes to the advancement of general federated learning and holds immense potential for practical applications in real-world scenarios.

Acknowledgements

This work is partially supported by the National Science Foundation under Grant No. 2333790, 2212323, and 2238275 and the National Institutes of Health under Grant No. R01AG077016.

Impact Statement

The introduction of `FedType` represents a significant advancement in the field of federated learning (FL), addressing crucial challenges associated with heterogeneous model aggregation. By leveraging small identical proxy models, `FedType` ensures secure and efficient information exchange among clients. The implications of this research are far-reaching, promising to enhance the application of federated learning in real-world scenarios where data privacy and efficient communication are paramount. By eliminating the reliance on public data and ensuring robust performance across diverse conditions, `FedType` has the potential to revolutionize the deployment of FL in sensitive and resource-constrained environments, including healthcare, finance, and beyond.

The proposed framework has two major limitations. First, selecting the proxy model is non-trivial and further influences the framework performance. In future work, we plan to design an automatic strategy to adaptively select the proxy model according to the types of private models. Second, although the proposed framework is effective and efficient, the designed uncertainty-based asymmetrical reciprocity learning between the proxy and private model runs slightly slower than the naive bidirectional knowledge distillation. Thus, we need to develop a more efficient approach to accelerate this step. From our perspective, there is no negative social impact on the designed framework.

References

Alam, S., Liu, L., Yan, M., and Zhang, M. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in Neural Information Processing Systems*, 35:29677–29690, 2022.

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022.

Balasubramanian, V., Ho, S.-S., and Vovk, V. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

Bao, W., Wang, H., Wu, J., and He, J. Optimizing the collaboration structure in cross-silo federated learning. *arXiv preprint arXiv:2306.06508*, 2023.

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. Predictive inference with the jackknife+. 2021.

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Bhatt, U., Antorán, J., Zhang, Y., Liao, Q. V., Sattigeri, P., Fogliato, R., Melançon, G., Krishnan, R., Stanley, J., Tickoo, O., et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 401–413, 2021.

Bouacida, N. and Mohapatra, P. Vulnerabilities in federated learning. *IEEE Access*, 9:63229–63249, 2021.

Chen, D., Yao, L., Gao, D., Ding, B., and Li, Y. Efficient personalized federated learning via sparse model-adaptation. *arXiv preprint arXiv:2305.02776*, 2023.

Cho, Y. J., Wang, J., Chirvolu, T., and Joshi, G. Communication-efficient and model-heterogeneous personalized federated learning via clustered knowledge transfer. *IEEE Journal of Selected Topics in Signal Processing*, 17(1):234–247, 2023.

Dennis, D. K., Li, T., and Smith, V. Heterogeneity for the win: One-shot federated clustering. In *International Conference on Machine Learning*, pp. 2611–2620. PMLR, 2021.

Diao, E., Ding, J., and Tarokh, V. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations*, 2020.

Fisch, A., Schuster, T., Jaakkola, T., and Barzilay, R. Few-shot conformal prediction with auxiliary tasks. In *International Conference on Machine Learning*, pp. 3329–3339. PMLR, 2021.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Huang, W., Ye, M., and Du, B. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10143–10153, 2022.

Kuleshov, V., Fenner, N., and Ermon, S. Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning*, pp. 2796–2804. PMLR, 2018.

Li, D. and Wang, J. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33: 2351–2363, 2020.

Lu, C., Yu, Y., Karimireddy, S. P., Jordan, M., and Raskar, R. Federated conformal predictors for distributed uncertainty quantification. In *International Conference on Machine Learning*, pp. 22942–22964. PMLR, 2023.

Lyu, L., Yu, H., Ma, X., Chen, C., Sun, L., Zhao, J., Yang, Q., and Philip, S. Y. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, 2022.

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32, 2019.

Marfoq, O., Neglia, G., Vidal, R., and Kameni, L. Personalized federated learning through local memorization. In *International Conference on Machine Learning*, pp. 15070–15092. PMLR, 2022.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Neal, R. M. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Ogier du Terrail, J., Ayed, S.-S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Advances in Neural Information Processing Systems*, 35:5315–5334, 2022.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. Inductive confidence machines for regression. In *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*, pp. 345–356. Springer, 2002.

Plassier, V., Makni, M., Rubashevskii, A., Moulines, E., and Panov, M. Conformal prediction for federated uncertainty quantification under label shift. *arXiv preprint arXiv:2306.05131*, 2023.

Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Sankaranarayanan, S., Angelopoulos, A., Bates, S., Romano, Y., and Isola, P. Semantic uncertainty intervals for disentangled latent spaces. *Advances in Neural Information Processing Systems*, 35:6250–6263, 2022.

Shafer, G. and Vovk, V. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Shamsian, A., Navon, A., Fetaya, E., and Chechik, G. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pp. 9489–9502. PMLR, 2021.

Sharma, A., Veer, S., Hancock, A., Yang, H., Pavone, M., and Majumdar, A. Pac-bayes generalization certificates for learned inductive conformal prediction. *arXiv preprint arXiv:2312.04658*, 2023.

Shen, T., Zhang, J., Jia, X., Zhang, F., Lv, Z., Kuang, K., Wu, C., and Wu, F. Federated mutual learning: a collaborative machine learning method for heterogeneous data, models, and objectives. *Frontiers of Information Technology & Electronic Engineering*, 24(10):1390–1402, 2023.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

T Dinh, C., Tran, N., and Nguyen, J. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., and Zhang, C. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8432–8440, 2022.

Tolpegin, V., Truex, S., Gursoy, M. E., and Liu, L. Data poisoning attacks against federated learning systems. In *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*, pp. 480–501. Springer, 2020.

Vovk, V. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74:9–28, 2015.

Vovk, V., Gammerman, A., and Saunders, C. Machine-learning applications of algorithmic randomness. 1999.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Wang, J., Yang, X., Cui, S., Che, L., Lyu, L., Xu, D., and Ma, F. Towards personalized federated learning via heterogeneous model reassembly. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yao, D., Pan, W., O'Neill, M. J., Dai, Y., Wan, Y., Jin, H., and Sun, L. Fedhm: Efficient federated learning for heterogeneous models via low-rank factorization. *arXiv preprint arXiv:2111.14655*, 2021.

Yi, L., Wang, G., Liu, X., Shi, Z., and Yu, H. Fedgh: Heterogeneous federated learning with generalized global header. *arXiv preprint arXiv:2303.13137*, 2023.

Yu, S., Qian, W., and Jannesari, A. Resource-aware federated learning using knowledge extraction and multi-model fusion. *arXiv preprint arXiv:2208.07978*, 2022.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pp. 7252–7261. PMLR, 2019.

Zhang, X., Zhou, X., Lin, M., and Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

Zhang, X., Li, Y., Li, W., Guo, K., and Shao, Y. Personalized federated learning via variational bayesian inference. In *International Conference on Machine Learning*, pp. 26293–26310. PMLR, 2022.

Zhou, Y., Wu, J., Wang, H., and He, J. Adversarial robustness through bias variance decomposition: A new perspective for federated learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pp. 2753–2762. ACM, 2022.

A. Algorithm Flow

In this subsection, we show the whole algorithm flow of our proposed `FedType` in Algorithm 2.

Algorithm 2: Algorithm Flow of `FedType`.

Input: Client training data $\mathcal{D}_1, \cdots, \mathcal{D}_N$, private models $\{\mathbf{w}_1, \cdots, \mathbf{w}_N\}$, proxy models $\{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$, validation data $\{\mathcal{D}'_1, \cdots, \mathcal{D}'_N\}$, communication round T, local training epoch R, and hyperparameters.

for each communication round $t = 1, 2, \cdots, T$ **do**
  Client Update
    for each epoch $r = 1, \cdots, R$ **do**
      for active client $i \in [1, \cdots, B]$ **do**
        Conduct uncertainty-based asymmetrical reciprocity learning following the Algorithm 1.
      end
    end
    Obtain private models $\{\mathbf{w}_1^t, \cdots, \mathbf{w}_N^t\}$ and proxy models $\{\mathbf{p}_1^t, \cdots, \mathbf{p}_N^t\}$;
    Upload proxy models $\{\mathbf{p}_1^t, \cdots, \mathbf{p}_N^t\}$ to the server;
  Server Update
    Obtain the aggregated model $\mathbf{p}_g^t$ via FedAvg or other existing methods;
    Distribute the aggregated model $\mathbf{p}_t^g$ back to clients.
end

B. Details of Conformal Prediction

B.1. Conformal Prediction

Conformal prediction is a promising statistical framework for describing the prediction uncertainty (Vovk et al., 2005; Shafer & Vovk, 2008; Angelopoulos & Bates, 2021; Angelopoulos et al., 2022; Barber et al., 2023; Sharma et al., 2023). It guarantees finite-sample coverage under the mild assumption of exchangeability. The main idea of conformal prediction involves establishing a non-conformity score S to assess the conformity between new test data and existing data distribution. Compared with existing traditional methods for estimating prediction uncertainty, such as Bayesian neural networks (Neal, 2012; Kuleshov et al., 2018; Maddox et al., 2019) and Platt scaling (Platt et al., 1999; Guo et al., 2017), conformal prediction offers numerous advantages, including its post-hoc nature, and being distribution-free and model-agnostic.

The original version of conformal prediction, also known as full conformal prediction or transductive conformal prediction, is regarded for its exceptional uncertainty estimation capabilities. However, its considerable computational cost restricts its practical application. To address this, several alternative methods have been developed to reduce the computational demands of full conformal prediction while retaining its core utility, for example, Split Conformal Prediction (Inductive CP) (Papadopoulos et al., 2002), Cross-CP (Vovk, 2015), and jackknife+ (Barber et al., 2021). In particular, Split Conformal Prediction has garnered significant attention due to its ease of implementation. Our paper is based on split conformal prediction, which we will introduce below.

B.2. Split Conformal Prediction Model Training

Next, we describe the process for deriving $\mathbf{cp}_i^t$ as outlined in Eq. (5) following the methodology presented in the work (Angelopoulos et al., 2020). We describe the process of obtaining $\mathbf{cp}_i^t$ in Algorithm 3 below.

B.3. Regularized Adaptive Prediction Sets Converge Guarantee

Note that our proposed dynamic adjustment mechanism retains all the characteristics of traditional conformal prediction requirements. Additionally, it ensures the isolation of $\mathcal{D}_i$ and $\mathcal{D}_i^{'}$ throughout the entire federated learning process, including during local training phases. Thus, regularized adaptive prediction sets converge can be guaranteed. More details about the **upper bound proof** and **lower bound proof** can be found in Section D in (Angelopoulos & Bates, 2021).

C. Datasets

We assess the effectiveness of the proposed `FedType` approach through image classification tasks conducted in both cross-device and cross-silo scenarios. For the **cross-device** evaluation, we employ three image classification datasets –

Algorithm 3: Algorithm Flow of Cmodel Training with Eq. (5).

Input: A trained proxy model $\mathbf{p}_i^t$ using the training dataset $\mathcal{D}_i$, validation data $\mathcal{D}_i'$.

- **Step 1:** Define the nonconformity score function S based on the nonconformity measure and given hyperparameters;
- **Step 2:** Calculate the nonconformity score s_j for each $\mathbf{x}_j \in \mathcal{D}_i'$, i.e., $s_j = S(\mathbf{x}_j)$, forming the nonconformity score set $\{s_j\}$;
- **Step 3:** Obtain the score quantile Q_θ related to θ on the nonconformity score set, i.e., $Q_\theta := \frac{1}{n}\lceil(1-\theta)(n+1)\rceil$-th quantile of $\{s_j\}_{j=1}^n$, where $n = |\mathcal{D}_i^{'}|$;
- **Step 4:** Integrate $\mathbf{p}_i^t$ with Q_θ. $\mathbf{cp}_i^t$ is first initialized by trained $\mathbf{p}_i^t$ and then predicts and calculates the non-conformity score on the test data coupled with the candidate label, which is then compared with the Q_θ.

Return: The trained conformal model $\mathbf{cp}_i^t$.

FMNIST, CIFAR-10, and CIFAR-100. To introduce heterogeneity in federated learning, a methodology inspired by existing work (Yurochkin et al., 2019; Hsu et al., 2019) is followed. This involves manipulating the concentration parameter α of the Dirichlet distribution to partition the datasets. Here, α signifies the label heterogeneity across clients, with smaller values concentrating labels on a few categories for a client (where $\alpha \to 0$ implies a client stores data with a single category). Conversely, larger α values lead to clients holding a more diverse set of label categories (where $\alpha \to +\infty$ indicates each client possesses data spanning all categories). Our data distribution process involves initially allocating data to clients, followed by a subsequent split into local model training, testing, and conformal learning sets in a 7:2:1 ratio. *While the splitting method guarantees an identical distribution for both sets, aligning effectively with the personalized federated learning setting, it is noteworthy that the increased value of α introduces heightened difficulty in the classification tasks.*

For the **cross-silo** setting, the Fed-ISIC19 dataset (Ogier du Terrail et al., 2022) is employed, featuring 23,247 dermoscopy images encompassing eight distinct types of melanoma. Following the data partition strategy of FLamby (Ogier du Terrail et al., 2022), the number of training/testing data on six clients is distributed as 9,930/2,483, 3,163/791, 2,690/673, 655/164, 351/88, and 180/45, respectively. The cross-silo setting necessitates the involvement of all clients in the training process at each communication round.

D. Baseline Introduction

In our paper, though there is no previous heterogeneous FL work sharing the exact setting with our work, we introduce the selected baselines used for the homogeneous setting comparison.

- **FedAvg** (McMahan et al., 2017): It is the vanilla baseline. In this approach, active local clients train their models and send the parameters to a central server. The server then computes the average of these local model parameters and redistributes the aggregated global model to the active clients for subsequent rounds of local training.
- **FedProx** (Li et al., 2020): this work introduces a proximal term in the local training of each client, quantifying the divergence between the local and global models. This term acts as a constraint, ensuring that the local models' personalized optimization does not deviate significantly from the global model.
- **pFedMe** (T Dinh et al., 2020): It uses regularized loss and decouples the personalization problem into a bi-level optimization. pFedMe aims to develop personalized models for each client. It does so by integrating Moreau envelopes into the learning process to help regularize the local updates during training;
- **pFedBayes** (Zhang et al., 2022): It proposes an algorithm to take consideration of the global distribution while conducting local model training. By leveraging Bayesian inference, pFedBayes not only customizes models to fit individual client needs better but also provides a robust framework for managing the inherent uncertainties and variabilities in distributed datasets.

Notably, these approaches are all used as model aggregation solutions in the experiments due to the flexibility of our proposed framework.

E. Model Details and Implementation

We introduce the models that we use in our experiment one by one. For all the following **client private models**, we adjust the dimension of the linear layer to fit the number of classes of the datasets accordingly.

- **ResNet family**: ResNet (He et al., 2016) is a type of convolutional neural network (CNN) architecture proposed in 2015. The key innovation of ResNet is its use of residual blocks. These blocks allow the network to learn residual functions with reference to the layer inputs, instead of learning unreferenced functions. In our work, we use ResNet-18, 34, 50, 101, and 152. The implementation is based on the Pytorch official library[5].
- **VGG family**: VGG (Simonyan & Zisserman, 2014), short for Visual Geometry Group, refers to a deep CNN architecture. VGG was one of the first to demonstrate that depth of the network (i.e., the number of layers) is a critical component for achieving high performance in visual recognition tasks. In our work, we use VGG-11, 13, 16, and 19. The implementation is based on the Pytorch official library[6].

Except for ResNet-18, we also test the following models as the **proxy modes** in our experiments:

- **ShuffleNet**: ShuffleNet(Zhang et al., 2018) is an efficient CNN designed primarily for mobile and computing devices with limited computational capacity. The key innovation in ShuffleNet is the use of pointwise group convolutions and channel shuffle operations to significantly reduce the computational cost and the number of parameters. In our work, we use ShuffleNet V2 (Ma et al., 2018) and implement it using the Pytorch library[7].
- **MobileNet**: MobileNet (Howard et al., 2017) is a class of efficient CNN architectures designed specifically for mobile and embedded vision applications. The key innovation in MobileNet is the use of depthwise separable convolutions, which significantly reduce the number of parameters and the computational burden compared to standard convolutions used in more traditional CNN architectures. In our experiment, we use MobileNet V1 (Howard et al., 2017) and implement it based on the Github resource [8].
- **EffcientNet**: EfficientNet is an efficient CNN structure introduced in (Tan & Le, 2019). The primary innovation of EfficientNet lies in its novel compound scaling method, which uniformly scales the depth, width, and resolution of the network with a set of fixed scaling coefficients. In our paper, we use EfficientNet-B0 and implement via Pytorch library[9].

F. Hyperparameter Details

Our experimental setup involves 100 communication rounds, 100 clients, a 20% sample ratio for the cross-device experiments, and five local training epochs. All experiments are conducted on an NVIDIA A100 with CUDA version 12.0, running on a Ubuntu 20.04.6 LTS server. All baselines and the proposed `FedType` are implemented using PyTorch 2.0.1. For the local update, we set the learning rate as 0.0001, the batch size is 16, and the optimizer used in the optimization is Adam. Following the provided value in the work (Angelopoulos et al., 2020), we set $\lambda = 0.5$, $\kappa_{reg} = 5$, and $\theta = 0.1$ for the local conformal model and proxy conformal model in Eq. (6), the conformal learning batch size is 32. The learning rate in the Platt scaling process is 0.01, and the maximum iteration is 10 following the default setting.

G. Empirical Study of Convergence

In this section, we present the convergence results of our proposed `FedType` framework as illustrated in Figure 5 across different datasets: FMNIST, CIFAR-10, CIFAR-100, and Fed-ISIC19, shown in panels (a), (b), (c), and (d), respectively. These results are obtained under the setting of $\alpha = 0.5$ with FedAvg employed as the aggregation method. The x-axis represents the number of communication rounds, while the y-axis tracks the average accuracy of clients. Observations from these results indicate that the private model, the proxy model, and the global model within the `FedType` framework all

[5] https://pytorch.org/vision/main/models/resnet.html
[6] https://pytorch.org/hub/pytorch_vision_vgg/
[7] https://pytorch.org/hub/pytorch_vision_shufflenet_v2/
[8] https://github.com/jmjeon94/MobileNet-Pytorch/blob/master/MobileNetV1.py
[9] https://pytorch.org/vision/main/models/efficientnet.html

achieve convergence. This effectively demonstrates the robust convergence properties of our proposed approach empirically.

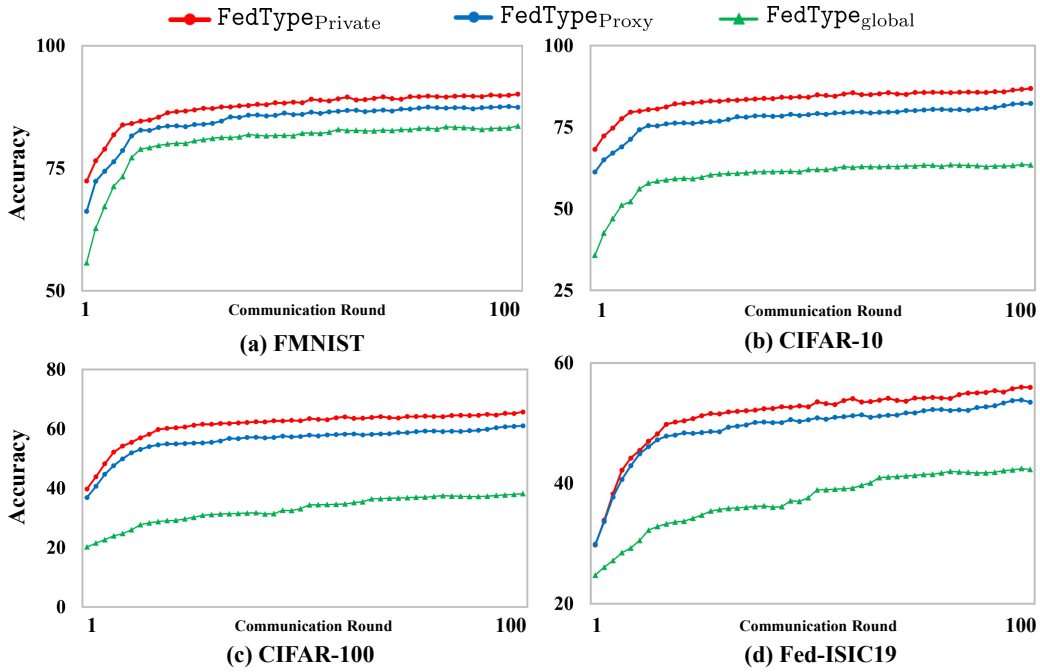


Figure 5. The empirical convergence of our proposed `FedType`.

H. Study of Consensus Weight η

As shown in Section 3.2.4 and Eq. (3), η serves as a crucial metric in our framework, quantifying the degree of consensus to regulate knowledge transfer from the proxy model $\mathbf{p}_i^t$ to the larger model $\mathbf{w}_i^t$. In this section, we analyze the behavior of η across various datasets: FMNIST, CIFAR-10, CIFAR-100, and Fed-ISIC19, as depicted in Figure 6 in panels (a), (b), (c), and (d), respectively, where x-axis denotes the number of communication rounds, and y-axis represents the average η values of all training data for all active/selected clients at round t. These observations are made under the setting of $\alpha = 0.5$ for FMNIST, CIFAR-10, and CIFAR-100, utilizing FedAvg as the aggregation method.

The results indicate a consistent increase in the value of η across communication rounds, suggesting that the large model and the proxy model are achieving greater consensus as training progresses. Notably, an η value of 1 implies complete uniformity in the uncertainty set for input data between the large and proxy models. Interestingly, for simpler datasets like FMNIST and Fed-ISIC19, η attains more instances of 1 and converges more rapidly compared to the more complex CIFAR-10 and CIFAR-100 datasets. These experimental findings highlight the dynamic nature of η throughout the iterative learning process and validate our strategy of assigning increased weight to the knowledge transferred from the proxy model to the large model as consensus grows.

I. Calibration Function Study

In Section 4.2.4, we detailed a dynamic calibration function designed to adjust the prediction set size in response to changes in model performance. This section delves deeper into the training process, specifically examining changes quantified with $\alpha = 0.5$ and FedAvg as the aggregation method. We focus on the FMNIST, CIFAR-10, CIFAR-100 datasets under a cross-device setting, and Fed-ISIC19 under a cross-silo setting. At the 50th communication round, we highlight changes in active clients' **average proxy set size** and **average client accuracy** (last batch of the training epoch), as presented in Table 7.

From these results, we draw several insights: (1) In scenarios where accuracy improves across training epochs, such as

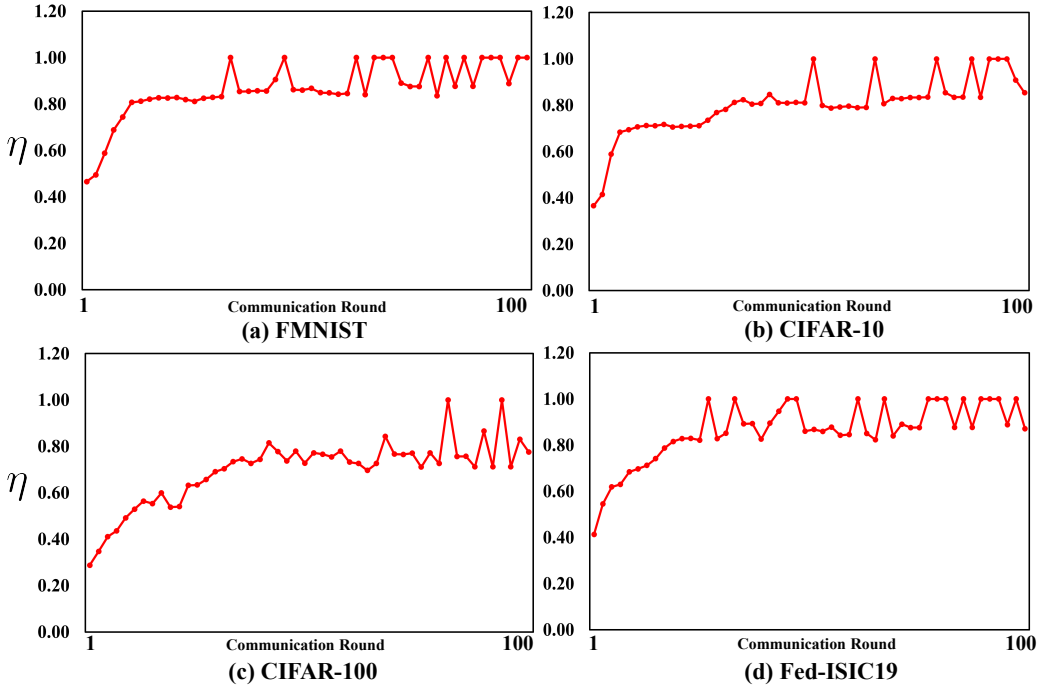


Figure 6. The value of η of our proposed `FedType`.

with CIFAR-10 and Fed-ISIC19, the prediction set size of the proxy model remains fairly consistent. This stability is attributed to our specifically designed function $g(\Delta^t, \lambda)$. In instances where Δ^t falls within the range of $(0, 1]$, the function maintains $g(\Delta^t, \lambda) = \lambda$, with λ consistently set at 0.5. (2) Conversely, where there's a decrease in accuracy, as observed in the CIFAR-100 dataset between the 1st and 3rd epochs (accuracy dropping from 62.15% to 61.42%), the prediction set size correspondingly reduces from 3.06 to 2.56. In contrast, between the 3rd and 5th epochs of the same dataset, as accuracy increases from 52.56% to 52.77%, the set size also rises from 2.63% to 2.69%. Similar patterns are noted in the FMNIST dataset. These findings underscore the efficacy of our dynamic calibration function in adapting the prediction set size in tandem with fluctuations in model accuracy during the training process.

Table 7. Calibration function quantitative analysis. $|\mathcal{S}^{50}|$ denotes the average size of the prediction sets by proxy models.

Dataset	**FMNIST**		**CIFAR-10**		**CIFAR-100**		**Fed-ISIC19**	
Epoch	Accuracy	$\lvert\mathcal{S}^{50}\rvert$	Accuracy	$\lvert\mathcal{S}^{50}\rvert$	Accuracy	$\lvert\mathcal{S}^{50}\rvert$	Accuracy	$\lvert\mathcal{S}^{50}\rvert$
1	87.85 (-)	2.15 (-)	83.26 (-)	2.13 (-)	62.15 (-)	3.06 (-)	52.10 (-)	2.69 (-)
2	87.62 (↓)	1.75 (↓)	83.57 (↑)	2.11 (↓)	61.52 (↓)	2.75 (↓)	52.47 (↑)	2.56 (↓)
3	88.21 (↑)	2.00 (↑)	83.92 (↑)	2.13 (↑)	61.42 (↓)	2.56 (↓)	52.56 (↑)	2.63 (↑)
4	88.43 (↑)	2.19 (↑)	84.11 (↑)	2.13 (-)	62.56 (↑)	2.62 (↑)	52.62 (↑)	2.56(↓)
5	88.56 (↑)	2.25 (↑)	84.21 (↑)	2.19 (↑)	62.87 (↑)	3.12 (↑)	52.77 (↑)	2.69 (↑)

J. Conformal Prediction Set Visualization

In this section, our focus is on evaluating the performance of conformal models for both private and proxy models on identical data points throughout the iterative training process. This evaluation is aimed at understanding how these models progress towards consensus on a certain data sample $\mathbf{x}_j$. Utilizing the CIFAR-10 dataset and adhering to the settings outlined in Table 1, we graphically represent the prediction set of one data sample randomly selected from CIFAR-10 for both private and proxy models in Figure 7. On this graph, the x-axis represents the communication round, while the y-axis corresponds to label IDs (ranging from 1 to 10, corresponding to the dataset's labels).

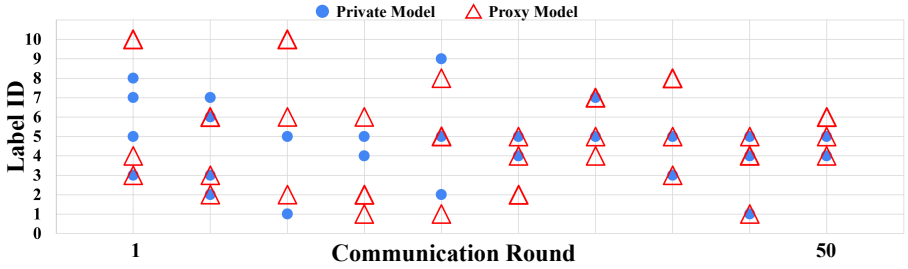


Figure 7. Prediction set consensus study.

Table 8. Performance (%) of the text classification task under the heterogeneous cross-device settings .

Dataset	$\texttt{FedType}_{\text{global}}$	$\texttt{FedType}_{\text{proxy}}$	$\texttt{FedType}_{\text{private}}$
Heterogenous	91.21	91.86	92.16
Homogenous	90.13	90.89	91.44

From our analysis, it is evident that the labels within the prediction sets generated by the conformal models of both private and proxy models increasingly overlap as the communication rounds progress. Notably, in later communication rounds, we observe a reduction in the size of both prediction sets, with each set encapsulating the ground truth label 5. This convergence of label predictions between the two models' prediction sets suggests a gradual move toward consensus within a specific range for the given input data. This observation is crucial as it demonstrates the effectiveness of the iterative training process in aligning the outputs of the private and proxy models, contributing significantly to the overall model consensus and performance.

K. Experiment Results on Text Classification Task

In addition to our previous experiments, we also conducted tests on text classification using the AG news dataset to further validate our `FedType` framework. For these experiments, we created a heterogeneous model pool by adding 3, 4, or 5 linear layers to the base of a pre-trained DistillBERT model. Specifically, we used DistillBERT with 3 linear layers as the proxy model in the heterogeneous setting and DistillBERT with 4 linear layers for the homogeneous setting. The experimental setup included using $\alpha = 0.5$ and FedAvg as the aggregation method, with a total of 50 communication rounds, keeping all other settings consistent with those in Table 1. During training, we only fine-tuned the appended linear layers, keeping the pre-trained DistillBERT layers fixed. The results, presented in Table 8, affirm the effectiveness of `FedType` in text classification tasks.

Interestingly, despite the utilization of a common pre-trained language model, which led to similar accuracy levels as observed in our image classification tasks, there were still noticeable differences in the performance of the global, proxy, and private models. This aligns with the observations from our image classification experiments, thereby reinforcing the efficacy of `FedType` in handling text classification tasks. These results not only demonstrate the versatility of our approach but also its adaptability across different types of data.

L. Homogeneous Results on Fed-ISIC19 dataset

In this section, we report the experiment results on Fed-ISIC19 under the homogeneous setting due to the page limit of the main paper. We maintain all the other settings as the heterogeneous setting except replacing the private model of clients with VGG-11, which is the same as the setting in Section 4.3. The results are shown in figure 8. Based on the experiment results, we observe that they generally align with the results under the heterogenous setting in Figure 2. It further demonstrates the effectiveness and robustness of our proposed `FedType` under both the heterogenous setting and the homogeneous setting.

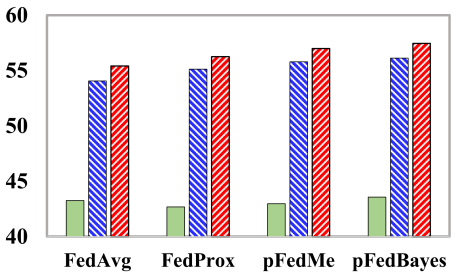


Figure 8. Homogeneous evaluation with Fed-ISIC19 dataset under the cross-silo setting.

Table 9. Comparision of the average of 3 epoch training time (second) on datasets

Setting	FMNIST	CIFAR-10	CIFAR-100	Fed-ISIC19
Private model training in `FedType`	265.67	279.33	312.67	526.00
Private model training with symmetrical KD loss	238.00	241.67	275.33	493.67
Proxy model training in `FedType`	161.33	164.67	176.33	371.00
Proxy model training with only CE loss	129.67	131.33	152.00	253.33

M. Resource Usage Discussion

In this section, we evaluate the computational efficiency of our proposed `FedType` framework. Our focus is on the duration of each training epoch, particularly how it compares to scenarios without our specially designed module. To this end, we compare the training time of the private model in `FedType` with its counterpart that only uses symmetrical KD loss. Similarly, we assess the training time of the proxy model in `FedType` against training with only the CE loss. The average training time over three epochs is reported. The results in Table 9 indicate that, while our approach does entail additional training time within a certain range, this increase is justifiable considering the performance improvements detailed in the ablation study (Table 4.2.3) and the communication cost reduction observed in Figure 4.3. This aspect is also addressed in the discussion on limitations, where we contemplate strategies to further minimize computational costs in future work.

You can have as much text here as you want. The main body must be at most 8 pages long. For the final version, one more page can be added. If you want, you can use an appendix like this one.

The `\onecolumn` command above can be kept in place if you prefer a one-column appendix, or can be removed if you prefer a two-column appendix. Apart from this possible change, the style (font size, spacing, margins, page numbering, etc.) should be kept the same as the main body.