# Uncertainty-Aware Label Refinement on Hypergraphs for Personalized Federated Facial Expression Recognition

Hu Ding, Yan Yan, *Senior Member, IEEE,* Yang Lu, *Member, IEEE,* Jing-Hao Xue, *Senior Member, IEEE,* Hanzi Wang, *Senior Member, IEEE*

***Abstract*—Most facial expression recognition (FER) models are trained on large-scale expression data with centralized learning. Unfortunately, collecting a large amount of centralized expression data is difficult in practice due to privacy concerns of facial images. In this paper, we investigate FER under the framework of personalized federated learning, which is a valuable and practical decentralized setting for real-world applications. To this end, we develop a novel uncertainty-Aware label refineMent on hYpergraphs (AMY) method. For local training, each local model consists of a backbone, an uncertainty estimation (UE) block, and an expression classification (EC) block. In the UE block, we leverage a hypergraph to model complex high-order relationships between expression samples and incorporate these relationships into uncertainty features. A personalized uncertainty estimator is then introduced to estimate reliable uncertainty weights of samples in the local client. In the EC block, we perform label propagation on the hypergraph, obtaining high-quality refined labels for retraining an expression classifier. Based on the above, we effectively alleviate heterogeneous sample uncertainty across clients and learn a robust personalized FER model in each client. Experimental results on two challenging real-world facial expression databases show that our proposed method consistently outperforms several state-of-the-art methods. This indicates the superiority of hypergraph modeling for uncertainty estimation and label refinement on the personalized federated FER task. The source code will be released at https://github.com/mobei1006/AMY.**

***Index Terms*—Facial expression recognition, Federated learning, Hypergraph Networks**

## I. Introduction

OVER the past few decades, facial expression recognition (FER) has received considerable attention with a variety of applications, including social robotics and human-computer interaction. The main goal of FER is to classify the input facial image into one of the expression categories, including anger (AN), disgust (DI), fear (FE), happiness (HA), sadness (SA), surprise (SU), and neutrality (NE).

A large number of FER methods [1]–[10] have been developed and achieved promising performance in unconstrained scenarios. These methods typically rely on large-scale expression images to be collected and shared for centralized training. However, collecting and sharing a large number of expression images is a great challenge due to privacy concerns and the sensitivity of facial images. In many practical applications, expression images are often distributed across local clients and are not shared. Consequently, how to exploit decentralized expression data for FER merits further investigation.

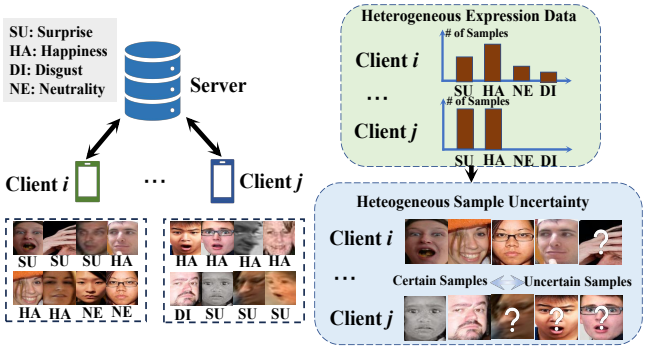



Fig. 1: Illustration of the challenges of heterogeneous expression data and heterogeneous sample uncertainty. The uncertainty arises from low-quality/ambiguous expression samples and noisy labels. The degree of sample uncertainty varies for client $i$ and client $j$. The images are taken from the RAF-DB database [17].

To address privacy concerns associated with centralized learning, federated learning (FL) [11] has recently emerged as a promising decentralized learning paradigm. FL often learns a global model by aggregating locally trained model parameters. Some recent methods [12]–[15] study FER within the framework of FL, capitalizing on its characteristics to learn a global FER model through local training. Regrettably, due to the differences in user behavior and preferences, the data on each client may be inconsistent, making the global model struggle to adapt to local clients effectively [16]. Therefore, customizing personalized models tailored to individual clients becomes essential. In this paper, we study personalized federated FER (PF-FER), which aims to learn a personalized FER model in each client rather than obtaining a global model. Such a way allows each client to better adapt to local data and achieve improved recognition performance.

For PF-FER, facial expression data scattered across different

This work was partly supported by the National Natural Science Foundation of China under Grants 62372388, 62071404, and U21A20514, and by the Fundamental Research Funds for the Central Universities under Grant 20720240076.

Hu Ding, Yan Yan, Yang Lu, and Hanzi Wang are with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen 361102, China (e-mail: dinghu@stu.xmu.edu.cn; yanyan@xmu.edu.cn; luyang@xmu.edu.cn; hanzi.wang@xmu.edu.cn).

Jing-Hao Xue is with the Department of Statistical Science, University College London, London, WC1E 6BT, UK (e-mail: jinghao.xue@ucl.ac.uk).

clients pose a problem of heterogeneity due to different user preferences. In particular, the uncertainty arises from low-quality/ambiguous expression samples and noisy labels on local data. This leads to heterogeneous sample uncertainty, which is detrimental to the learning of expression features in each client since the local model easily overfits some uncertain samples (such as noisy labeled samples). As shown in Fig. 1, the degree of sample uncertainty varies across different clients. Therefore, it is critical to suppress heterogeneous sample uncertainty in PF-FER.

Recent FER methods [18]–[21] focus on sample uncertainty suppression. One representative work is self-cure network (SCN) [18], which leverages a self-attention mechanism to learn the importance weights of images. Based on these weights, SCN further relabels uncertain samples according to the classifier outputs. Lei *et al.* [22] address sample uncertainty based on graph embedding. Note that the above FER methods work on centralized learning and are not designed for PF-FER, in which heterogeneous sample uncertainty is a main challenge. Moreover, they either ignore sample relationships or only consider the pairwise (i.e., first-order) connections between samples. In other words, high-order relationships that involve complex interactions between multiple expression samples are not well exploited. As a result, these methods usually *give unreliable uncertainty estimation and incorrect relabeling results.*

To address the aforementioned challenges, we develop a novel uncertainty-**A**ware label refine**M**ent on h**Y**pergraphs (**AMY**) for PF-FER. In AMY, we propose to leverage hypergraph networks to model the intricate high-order relationships between multiple samples on local data, where we introduce a personalized module in each client. This enables us to effectively estimate sample uncertainty and perform label correction, achieving a robust and accurate local model.

Specifically, for local training, each client is composed of a backbone, an uncertainty estimation (UE) block, and an expression classification (EC) block. In the UE block, we capture the uncertainty relationships between samples based on a hypergraph network. Then, these relationships are encoded into the uncertainty features. Based on these features, a personalized uncertainty estimator is leveraged to reliably estimate the uncertainty weights of samples. We also employ a weight regularization loss to explicitly enlarge the differences between weights from certain samples and uncertain samples. In the EC block, we perform label propagation on the hypergraph, obtaining refined labels. These refined labels are combined with the model predictions to obtain final high-quality labels for retraining an expression classifier.

After local training on each client, the local models and local class prototypes are uploaded to the server. The server aggregates these models and prototypes and then sends the aggregated results back to each client for regularization, reducing the influence of heterogeneous expression data.

Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first attempt to address heterogeneous sample uncertainty in the personalized FL framework for FER (i.e., PF-FER).
- We jointly perform uncertainty learning and label propagation based on hypergraph modeling that captures the complex relationships between expression samples. As a result, reliable uncertainty weights and high-quality label refinement results are obtained for retraining a robust FER model in the local client.
- We conduct experiments on two challenging real-world facial expression databases to validate the effectiveness of our method against several state-of-the-art uncertainty learning methods and personalized FL methods.

The remainder of this paper is organized as follows. First, we give the related work in Section II. Then, we describe our proposed method in detail in Section III. Next, we perform extensive experiments on the two challenging facial expression databases in Section IV. Finally, we draw the conclusion in Section V.

## II. Related Work

In this section, we briefly review the methods closely related to our method. We first introduce facial expression recognition methods in Section II-A. Then, we introduce several state-of-the-art federated learning methods in Section II-B. Finally, we reivew some methods related to graph neural networks in Section II-C.

### A. Facial Expression Recognition (FER)

With the advance of deep learning, deep neural network-based FER methods [6]–[8], [23]–[25] have gained prominence. These methods learn discriminative expression features by either designing loss functions or performing disturbance decoupling. Xie *et al.* [6] develop a novel triplet loss based on class-wise boundaries and multi-stage outlier suppression for FER. Gu *et al.* [7] design a simple yet effective facial expression noise-tolerant network (FENN), which explores inter-class correlations to reduce the ambiguity between similar expression categories. Chen *et al.* [8] introduce a multi-relations aware network (MRAN) that focuses on both global and local attention features, and learns multi-level relationships to obtain effective expression features.

In recent years, some methods focus on addressing sample uncertainty in FER. Wang *et al.* [18] employ a self-attention mechanism to estimate sample uncertainty. She *et al.* [19] adopt the similarity between samples and labels for uncertainty estimation. Zhang *et al.* [20] propose a relative uncertainty learning (RUL) method for FER. Lei *et al.* [22] introduce a graph embedded uncertainty suppressing (GUS) method.

The above methods mainly study centralized learning on large-scale expression data. Different from these methods, we investigate the PF-FER task for privacy protection. Such a task allows multiple decentralized clients to learn personalized local models collaboratively without sharing their private expression data.

### B. Federated Learning (FL)

Recently, FL [26]–[28] has emerged as an effective decentralized learning paradigm that enables collaborative training

of multiple clients in a privacy-preserving manner. The predominant FL method is FedAvg [11], which obtains a global model by averaging model parameters trained on local clients. However, the performance of FedAvg is greatly affected when learning on non-IID data (i.e., heterogeneous data). Numerous efforts [29], [30] have been made to alleviate this problem. FedProx [29] rectifies model biases by incorporating a proximal term. CCVR [30] retrains classifiers by sampling virtual features from an approximate Gaussian mixture model. Zhang *et al.* [12] develop a federated spatiotemporal incremental learning method that leverages lifelong learning and federated learning to continuously optimize models on distributed edge clients. You *et al.* [13] introduce auxiliary clients involving auxiliary datasets related to federated learning tasks and generate Mixup templates for clients, addressing the privacy issues faced by Mixup-based methods.

Some recent works study FER under the FL framework. FedNet [15] applies the federated averaging mechanism to learn a global expression classification model. FedAffect [31] explores FER under the few-shot FL setting. The above methods learn a global model by aggregating information from clients. However, the global model may not work well for local clients. Moreover, these methods do not account for the ubiquitous sample uncertainty on the PF-FER task, leading to sub-optimal performance.

Instead of training a global model, personalized FL [16] acknowledges the heterogeneity of data among clients by constructing a personalized model for each client. FedProto [32] aggregates the local prototypes collected from clients, and then sends the global prototypes back to all clients to regularize local training. Huang *et al.* [33] introduce FedAMP, which employs federated attention message passing to enhance collaboration between similar clients. Niu and Deng [34] introduce gradient correction for federated face recognition. Liu *et al.* [35] learn personalized models via a decoupled feature customization module.

Salman and Busso [14] are the first to study PF-FER. They aggregate local models to obtain a global model and design an unsupervised penalization strategy for video-based FER. In this paper, we also work on PF-FER, where we innovatively introduce hypergraph networks and take advantage of a personalized uncertainty estimator to mitigate the adverse effect of heterogeneous sample uncertainty in local training, aligning well with practical scenarios.

### C. Graph Neural Networks (GNNs)

GNNs have shown superiority in modeling data relationships. Some FER methods [25], [36] adopt GNNs to model pairwise (i.e., first-order) relationships between samples for classification. Nevertheless, such pairwise relationships are inferior in capturing complex interactions across vertices. Recently, hypergraph networks have been employed to model high-order correlations among data, where each hyperedge can involve multiple vertices. Feng *et al.* [37] propose a hypergraph neural network (HGNN) to encode high-order data correlations for representation learning. Zhang *et al.* [38] introduce a hypergraph label propagation network (HLPN) to optimize feature embeddings.

In our PF-FER task, each client involves the uncertainty arising from low-quality/ambiguous expression samples and noisy labels. Since the data distribution is heterogeneous across different clients, PF-FER suffers from heterogeneous sample uncertainty. Conventional methods only consider the pairwise connections between expression images and ignore high-order relationships (which can indicate different levels of relation). As a result, these methods usually give unreliable uncertainty estimation and incorrect relabeling results. To address this problem, we introduce the hypergraph neural network to model the intricate high-order relationships between multiple samples on local data. This enables us to effectively estimate sample uncertainty and perform label correction. This in turn greatly addresses heterogeneous sample uncertainty across local clients.

## III. Proposed Method

In this section, we elaborately introduce our proposed method for PF-FER. First, we give the preliminaries and notations in Section III-A. Then, we provide an overview of our method in Section III-B. Subsequently, we present technical details of the uncertainty estimation block and the expression classification block in Section III-C and Section III-D, respectively. Finally, we summarize the global training of our method in Section III-E.

### A. Preliminaries and Notations

The objective of PF-FER is to collaboratively train a personalized FER model in each client by communicating between clients and the server without disclosing raw expression data.

Suppose that we have $K$ clients and each client has $C$ expression categories, where the model parameters and the local data in the $k$-th client are denoted as $\mathbf{w}_k$ and $\mathcal{D}^k$, respectively. The expression data over clients are assumed to be non-IID. During local training, a batch of expression samples $\{\mathbf{x}_i^k, y_i^k\}_{i=1}^N$, where $N$ represents the number of images in a batch, and $\mathbf{x}_i^k$ and $y_i^k \in \{1, \ldots, C\}$ respectively represent the $i$-th facial expression image and its corresponding label in the $k$-th client, is randomly sampled from $\mathcal{D}^k$.

### B. Overview

Our AMY method follows two representative FL methods (a traditional FL method FedAvg [11] and a personalized FL method FedProto [32]). It consists of a server and multiple clients. During each round of training, the server distributes its current model and global class prototypes to some selected clients. Then, each selected client trains the local model on its own expression data, and only sends the model update and its local class prototypes to the server. Note that a personalized uncertainty estimator is trained exclusively in each client and does not share its model parameters. Next, the server aggregates model updates and local class prototypes from those selected clients. The above steps iterate several training rounds and personalized local models are finally learned on multiple clients.

An overview of our proposed AMY method is shown in Fig. 2. Each local model is comprised of three components

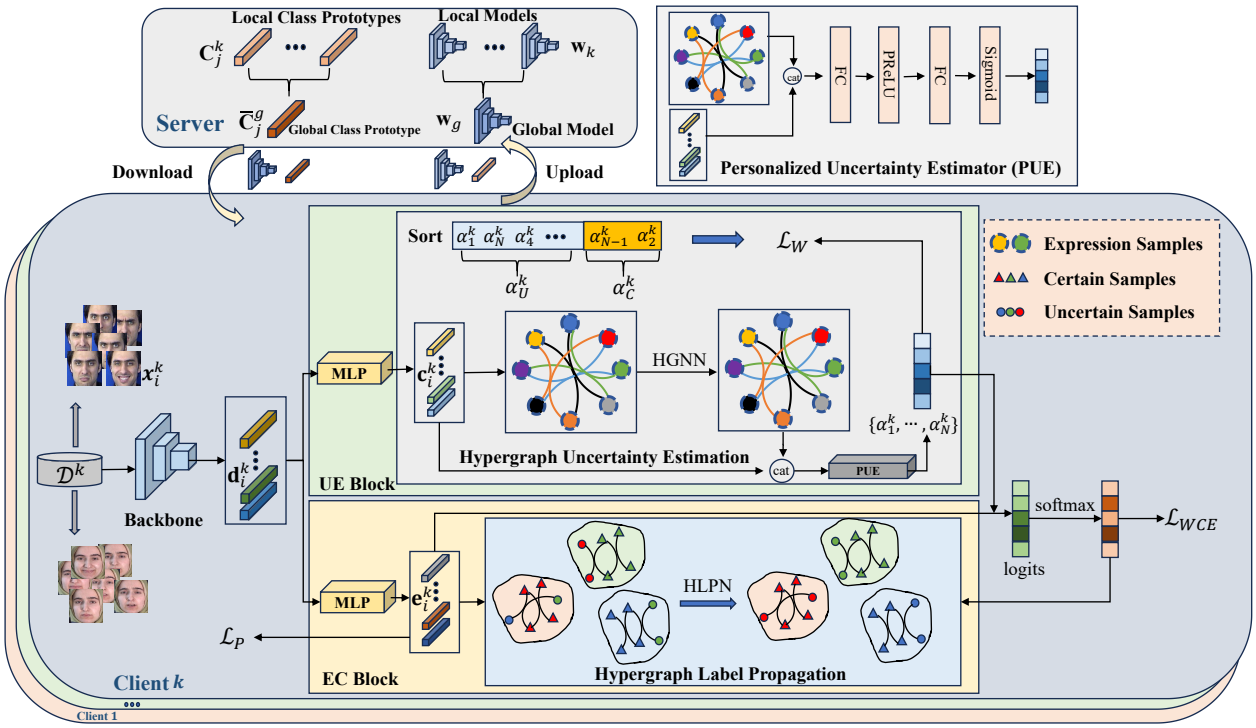



Fig. 2: Overview of our proposed AMY method. Each local model is comprised of a backbone, an uncertainty estimation (UE) block, and an expression classification (EC) block. A personalized uncertainty estimator (PUE) is private to each client and not uploaded to the server, enabling personalized training.

in each client: a backbone (we adopt the simple ResNet-18 in this paper), an uncertainty estimation (UE) block, and an expression classification (EC) block. For the local data in each client, the deep feature is initially extracted by the backbone.

On the one hand, the UE block first employs a multi-layer perception (MLP) to extract the compact feature from the deep feature. Then, the compact feature is fed into a hypergraph neural network (HGNN) to model complex relationships between samples, obtaining the relational feature. Next, the relational feature and the compact feature are concatenated to generate the uncertainty feature, which is then fed into a personalized uncertainty estimator to give an uncertainty weight for each sample. The personalized uncertainty estimator is private to each client and not uploaded to the server, enabling personalized training. We also adopt a weight regularization loss [18] to explicitly enlarge the differences between weights from certain samples and uncertain samples.

On the other hand, the EC block classifies the input into one expression category. It first employs an MLP to extract the expression feature from the deep feature and then performs label propagation on a hypergraph (HLPN), obtaining the refined labels. Finally, the refined labels are combined with model predictions to obtain the final label refinement results for retraining the expression classifier.

### C. Uncertainty Estimation (UE) Block

The UE block is designed to estimate the sample uncertainty on the client, where higher weights will be assigned to uncertain samples (i.e., the samples with a high degree of uncertainty). Existing methods either employ a fully connected layer or a simple graph structure to estimate the sample uncertainty. However, such ways may not be sufficient for modeling complex high-order relationships in facial expression images, generating unreliable uncertainty weights. To address this problem, we take advantage of hypergraph modeling, which offers great flexibility in representing data connections and exhibits superior capability in capturing complex relationships, to enable more reliable uncertainty estimation.

Given $\mathbf{x}_i^k$ in the $k$-th client, we denote the deep feature extracted from the backbone as $\mathbf{d}_i^k$. Then, the compact feature is extracted by an MLP and denoted as $\mathbf{c}_i^k$. The compact feature is further fed into an $L$-layer HGNN [37] for learning high-order relationships.

The hypergraph is denoted as $G = (V, E, W)$, where $V = \{v_1, \ldots, v_N\}$ represents the vertex set (each vertex represents a compact feature corresponding to a facial expression image), $E = \{e_1, \ldots, e_N\}$ represents the hyperedge set (each hyperedge is formed by connecting a vertex to its $K$ nearest neighbors, resulting in $N$ hyperedges connecting $K+1$ vertices each). The $K$ nearest neighbors are determined based on the Euclidean distance between vertices. $\mathbf{W}_u \in \mathbb{R}^{N\times N}$ is a weight matrix denoting the weights of the hyperedges. The distance between two vertices is measured by a Gaussian kernel function. The hypergraph structure can be represented by an incidence matrix $\mathbf{H}_u \in \mathbb{R}^{N\times N}$. For a given vertex $v \in V$ and a hyperedge $e \in E$, the element $\mathbf{H}_u(v, e)$ in the incidence

matrix is defined as

$$\mathbf{H}_u(v,e) = \begin{cases} 1 & v \in e \\ 0 & v \notin e. \end{cases} \tag{1}$$

For a vertex $v \in V$, its degree is defined as $d(v) = \sum_{e\in E} \mathbf{W}_u(e,e)\mathbf{H}_u(v,e)$. For an edge $e \in E$, its degree is defined as $\delta(e) = \sum_{v\in V} \mathbf{H}_u(v,e)$. $\mathbf{D}_{ue}$ and $\mathbf{D}_{uv}$ denote diagonal matrices for edge and vertex degrees, respectively.

We consider the relational features as a hypergraph signal denoted by $\mathbf{X}^l \in \mathbb{R}^{N\times C_l}$, where $C_l$ represents the dimension of the feature at the $l$-th layer. The convolution operation in the hypergraph convolutional network is formulated as

$$\mathbf{X}^{l+1} = \sigma(\mathbf{D}_{uv}^{-1/2}\mathbf{H}_u\mathbf{W}_u\mathbf{D}_{ue}^{-1}\mathbf{H}_u^{\mathrm{T}}\mathbf{D}_{uv}^{-1/2}\mathbf{X}^l\Theta), \tag{2}$$

where $\Theta \in \mathbb{R}^{C_l\times C_{l+1}}$ denotes the learnable parameters and $\sigma$ is a non-linear activation function. $\mathbf{X}^0 = [\mathbf{c}_1^k, \cdots, \mathbf{c}_N^k]$ is the input signal for the hypergraph neural network.

Based on the above, we transform the compact feature $\mathbf{c}_i^k$ into a relational feature $\mathbf{r}_i^k$ after $L$ layers. Then, the uncertainty feature $\mathbf{u}_i^k$ is obtained by concatenating the compact feature and the relational feature, i.e.,

$$\mathbf{u}_i^k = \mathrm{concat}(\mathbf{c}_i^k, \mathbf{r}_i^k), \tag{3}$$

where 'concat$(\cdot)$' denotes the concatenation operation.

Due to heterogeneous sample uncertainty across clients, using a common model to predict sample uncertainty in each client cannot guarantee the optimal performance. Instead, we make use of a personalized uncertainty estimator trained exclusively in the local client. The network structure of the personalized uncertainty estimator consists of two fully connected layers, a PReLU function, and a Sigmoid function. The output of the estimator is given as

$$\beta_i^k = \mathrm{Uncertain}(\mathbf{u}_i^k), \tag{4}$$

where Uncertain$(\cdot)$ denotes the uncertainty estimator and $\beta_i^k \in [0,1]$ denotes the uncertainty weight for the image $\mathbf{x}_i^k$.

In personalized federated learning, the personalized uncertainty estimator does not upload parameters to the server for aggregation and thus it is not affected by other clients. Instead, it only leverages local data for training, indicating that each client only relies on its unique local data to estimate uncertainty. Such a way ensures that each client obtains personalized uncertainty weights based on the uniqueness of its data.

To explicitly distinguish certain samples and uncertain samples, we adopt the weight regularization loss [18] to ensure meaningful uncertainty weights. Technically, we sort all the samples according to uncertainty weights. Based on this, a threshold $\zeta$ is used to divide these samples into certain and uncertain samples. The weight regularization loss is expressed as

$$\mathcal{L}_W = \max\{0, \eta - (\beta_U^k - \beta_C^k)\}, \tag{5}$$

where $\eta$ is the margin and $\beta_U^k$ and $\beta_C^k$ represent the weight means of uncertain and certain samples, respectively.

Note that each client involves the uncertainty arising from low-quality/ambiguous expression samples and noisy labels. Uncertain samples are detrimental to the learning of expression features in each client since the local model easily overfits these samples. Some existing FER methods use the GNN to model relationships between samples for uncertainty estimation. However, the GNN can only model pairwise relationships among samples. In fact, the relationships between expression images usually exhibit more complex high-order dependencies that reflect interactions among multiple images. Hence, the HGNN is introduced to model these complex relationships, facilitating the network to identify intrinsic similarities and differences between samples, thereby more accurately estimating sample uncertainty.

### D. Expression Classification (EC) Block

In the UE block, we estimate the uncertainty weight for each sample, where the weights for uncertain samples are higher than those for certain samples. Some uncertain samples are potentially contaminated with noisy labels. Hence, it is desirable to refine the labels of these uncertain samples, facilitating obtaining a more accurate local model. Conventional relabeling methods [18] perform relabeling according to model predictions. Such a strategy relies heavily on the model's inference ability. If the predicted labels by the model are not accurate enough, it can affect the model's accuracy. Ideally, we should also consider the relationships between samples to produce more accurate relabeling results. Motivated by this, we perform label propagation on the hypergraph, which involves transductive learning to update the labels according to sample relationships.

In the EC block, we construct another hypergraph using the $K$-nearest neighbors algorithm for label propagation. The expression feature $\mathbf{e}_i^k$ extracted from another MLP serves as the basis for hypergraph construction. Each expression feature is represented as a vertex, and the relationships between samples constitute hyperedges. Hence, we generate a weight matrix $\mathbf{W}_e$ and an incidence matrix $\mathbf{H}_e$. Accordingly, $\mathbf{D}_{ee}$ and $\mathbf{D}_{ev}$ denote diagonal matrices for edge and vertex degrees, respectively.

After constructing the hypergraph, we perform label propagation on the hypergraph. Similar to HLPN [38], the closed-form solution for calculating the predicted scores $\hat{\mathbf{F}}_u$ can be obtained by solving the following equation

$$\hat{\mathbf{F}}_u = (\mathbf{I} + \frac{1}{\lambda}(\mathbf{I} - \mathbf{D}_{ev}^{-1/2}\mathbf{H}_e\mathbf{W}_e\mathbf{D}_e^{-1}\mathbf{H}_e^{\mathrm{T}}\mathbf{D}_{ev}^{-1/2}))^{-1}\mathbf{Y}, \tag{6}$$

where $\mathbf{Y}$ is the original label matrix of samples, $\mathbf{I}$ denotes an identity matrix, and $\lambda$ is a trade-off parameter to balance the influence of the hypergraph structure regularizer.

Then, we transform the score matrix $\hat{\mathbf{F}}_u$ into a probability score matrix $\mathbf{P}^{L,k} = [\mathbf{p}_1^{L,k}, \ldots, \mathbf{p}_N^{L,k}]$ by the softmax function, where $\mathbf{p}_i^{L,k}$ represents a vector of propagated label probabilities for the $i$-th image. Meanwhile, we obtain the probability score matrix $\mathbf{P}^{S,k} = [\mathbf{p}_1^{S,k}, \ldots, \mathbf{p}_N^{S,k}]$ according to the output of the expression classifier and the softmax function, where $\mathbf{p}_i^{S,k}$ represents a vector of predicted class probabilities for the $i$-th image. We estimate the labels based on the maximum predicted class probability for each sample, expressed as

$$l_i^{L,k} = \arg\max(\mathbf{p}_i^{L,k}), \tag{7}$$

$$l_i^{S,k} = \arg\max(\mathbf{p}_i^{S,k}). \tag{8}$$

Meanwhile, we use uncertainty weights to select samples with unreliable labels for label refinement. The label refinement process can be defined as

$$y' = \begin{cases} l_{joint} & \text{if } \beta_i^k \geq \delta \text{ and } l_i^{L,k} = l_i^{S,k} \\ l_{origin} & \text{otherwise,} \end{cases} \tag{9}$$

where $y'$ denotes the refined label; $\delta$ is a threshold; $l_{origin}$ denotes the originally given label; $l_i^{L,k}$ and $l_i^{S,k}$ denote the estimated labels for the $i$-th sample calculated in hypergraph label propagation and predicted by the classifier, respectively. $l_{joint}$ indicates that $l_i^{L,k}$ and $l_i^{S,k}$ are the same label.

Both the conventional graph-based and hypergraph-based label propagation methods aim to propagate labels from labeled samples to unlabeled ones based on the underlying structure. However, the key difference between them lies in the different graph structures. Conventional graph neural networks can only model pairwise relationships. In contrast, hypergraph neural networks can capture high-order relationships, where each hyperedge can connect multiple vertices. In this way, the label information can be propagated more comprehensively across the vertices, resulting in more accurate re-labeling results (as validated in our ablation study in Section IV-B).

Note that we also leverage the uncertainty weights obtained from the personalized uncertainty estimator to adjust the expression classification loss $\mathcal{L}_{WCE}$ (in the form of logit-weighted cross-entropy loss), which is expressed as

$$\mathcal{L}_{WCE} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{(1-\beta_i^k) f_{y_i^k}^k(\mathbf{e}_i^k)}}{\sum_{j=1}^{C} e^{(1-\beta_i^k) f_j^k(\mathbf{e}_i^k)}}, \tag{10}$$

where $f_j^k$ represents the $j$-th expression classifier. $\mathcal{L}_{WCE}$ has a positive correlation with $(1-\beta_i^k)$ [18].

Finally, the overall training loss in the $k$-th client is

$$\mathcal{L}_k = \mathcal{L}_{WCE} + \lambda_1 \mathcal{L}_W + \lambda_2 \mathcal{L}_P, \tag{11}$$

where $\lambda_1$ and $\lambda_2$ are two balancing parameters; $\mathcal{L}_P = \frac{1}{C}\sum_{j=1}^{C} d(\mathbf{C}_j^k, \bar{\mathbf{C}}_j^g)$ regularizes the local training using class prototypes, in which '$d(\cdot,\cdot)$' denotes the distance measure ($L_1$ distance is used) between the local class prototype $\mathbf{C}_j^k$ and the global class prototype $\bar{\mathbf{C}}_j^g$ for the $j$-th expression category (see Section III-E for more details).

### E. Global Training

After local training, we send the local models to the server. Meanwhile, to mitigate the influence of heterogeneous data, we calculate the local class prototypes $\{\mathbf{C}_j^k\}_{j=1}^C$ of the deep expression features for each client and upload them to the server. The above process is given as

$$\mathbf{C}_j^k = \frac{1}{D_j^k} \sum_{\mathbf{x}_i^k \in \mathcal{D}_j^k} \mathbf{e}_i^k, \qquad j = 1, \cdots, C, \tag{12}$$

where $\mathcal{D}_j^k$ is a subset of the local dataset $\mathcal{D}^k$ consisting of $D_j^k$ training samples belonging to the $j$-th expression category.

**Algorithm 1:** The overall training of our method

**Input**: Local datasets; the number of clients $K$; local epochs $E$; global rounds $T$.
**Output**: Personalized trained models.
**Server**
1: **for** each round $t = 1, \cdots, T$ **do**
2: Randomly select a subset of clients $S_t$ and send the global model and global class prototypes learned at round $t-1$ to clients.
3: **for** each client $k \in S_t$ **do**
4: Update the local model and local class prototypes.
5: **end for**
6: Obtain a new global model and global class prototypes.
7: **end for**
**Client**
1: **for** each local epoch $i = 1, \cdots, E$ **do**
2: // *UE Block*
3: Use the UE block to obtain the uncertainty weights of the samples.
4: Update the local model by the stochastic gradient descent (SGD) algorithm.
5: // *UC Block*
6: Use the EC block to relabel uncertain samples.
7: **end for**
8: Calculate local class prototypes.
9: **return** The local model and local class prototypes.

The server receives the local models and the local class prototypes, and aggregates them to obtain the global model $\mathbf{w}_g$ and the global class prototypes $\{\bar{\mathbf{C}}_j^g\}_{j=1}^C$, calculated as

$$\mathbf{w}_g = \sum_{k \in \mathcal{S}} p_k \mathbf{w}_k, \tag{13}$$

$$\bar{\mathbf{C}}_j^g = \sum_{k \in \mathcal{S}} p_k \mathbf{C}_j^k, \qquad j = 1, \cdots, C, \tag{14}$$

where $p_k$ represents the weight of the $k$-th client and $\mathcal{S}$ is a randomly selected subset (i.e., active clients) from $K$ clients.

The server sends the aggregated model and the global class prototypes back to the clients. The above process is iterated several times, enabling each client to learn an effective personalized FER model. we summarize the overall training of our method in Algorithm 1.

## IV. Experimental Results

In this section, we first introduce the experimental settings in Section IV-A. Then, we conduct ablation studies in Section IV-B. Finally, we compare our method with several state-of-the-art methods in Section IV-C.

### A. Experimental Settings

We conduct experiments on two challenging real-world facial expression databases: RAF-DB [17] and FERPlus [39]. The RAF-DB database contains 30,000 facial images. We use

TABLE I: The average recognition accuracy (%) obtained by different variants of our method with the different values of $\alpha$ on the RAF-DB and FERPlus databases. The best results are boldfaced.

| Method | RAF-DB | | | | | FERPlus | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\alpha$=0.1 | $\alpha$=0.5 | $\alpha$=1 | $\alpha$=5 | $\alpha$=10 | $\alpha$=0.1 | $\alpha$=0.5 | $\alpha$=1 | $\alpha$=5 | $\alpha$=10 |
| Baseline | 87.07 | 76.73 | 61.89 | 54.57 | 65.50 | 86.36 | 69.64 | 76.86 | 70.38 | 73.73 |
| Baseline+UE w.o.W | 90.21 | 80.65 | 71.83 | 68.60 | 69.81 | 86.68 | 83.37 | 76.90 | 75.61 | 75.91 |
| Baseline+UE | 89.22 | 80.52 | 73.01 | 65.68 | 70.90 | 87.39 | 83.36 | 75.80 | 76.37 | 76.42 |
| Baseline+UE+EC | **90.97** | **80.70** | **73.40** | **71.97** | **71.52** | **88.23** | **83.93** | **77.79** | **76.83** | **77.30** |

seven basic expressions, including 12,271 training images and 3,068 test images. The FERPlus database, an extension of FER2013, contains 28,709 training images, 3,589 validation images, and 3,589 test images with eight expression categories.

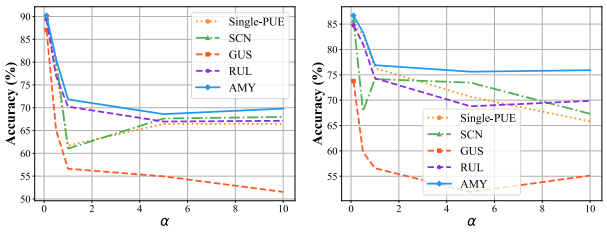
In our experiments, all facial images are first resized to $256 \times 256$ and subsequently randomly cropped to $224 \times 224$. We adopt ResNet-18 [40] as the backbone for all the competing methods. We use a two-layer HGNN in the UE block. We conduct experiments using PyTorch on a single NVIDIA GeForce RTX 3090 GPU. We perform 100 communication rounds, each involving 50% of active clients. The database is partitioned into $K$=10 clients using a Dirichlet distribution controlled by the $\alpha$ parameter (i.e., Dir($\alpha$)), where the value of $\alpha$ is set to 0.1, 0.5, 1, 5, or 10. A lower value of $\alpha$ indicates a more heterogeneous distribution over clients. Due to limitations of local device resources, local training is performed for only one round, with a batch size of 32. The optimization of local models uses the SGD algorithm with a learning rate of 0.10. The margin $\eta$ in Eq. (5) is set to 0.2. The threshold $\zeta$ for separating certain samples and uncertain samples is empirically set to 0.7. The threshold $\delta$ in Eq. (9) for updating labels is set to 0.6. The values of $\lambda_1$ and $\lambda_2$ in Eq. (11) are set to 0.8 and 1.0, respectively.

### B. Ablation Studies

Some ablation results are given in Table I, Fig. 3, Fig. 4, and Fig. 5. The baseline method is a combination of FedAvg and FedProto, where ResNet-18 and a simple fully connected layer are used for expression classification in each client. Unless specified, the value of $\alpha$ is set to 5 in ablation studies.

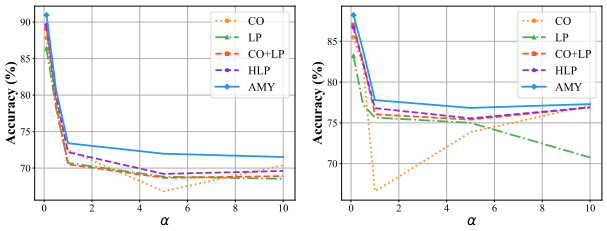
**Effectiveness of the Uncertainty Estimation (UE) Block.** From Table I, we can see that Baseline+UE achieves better performance than Baseline under the different values of $\alpha$, indicating the effectiveness of our UE block, which leverages a hypergraph neural network and a personalized uncertainty estimator to estimate reliable uncertainty weights in the local client.

To further show the advantages of high-order relationships in the hypergraph, we also evaluate some competitors (including SCN [18], GUS [22], and RUL [20]) of the UE block, where these competitors are used to estimate the sample uncertainty. We also evaluate a variant of the UE block (denoted as Single-PUE) by only using the personalized uncertainty estimator without the hypergraph network. The comparison results are shown in Fig. 3. Among these competitors, our method achieves the best results under the different values of $\alpha$. Our method also outperforms Single-PUE. These results show the superiority of applying hypergraph modeling and the personalized uncertainty estimator to estimate sample uncertainty.




(a) RAF-DB (b) FERPlus

Fig. 3: Comparison of different competitors of the UE blo ck at the different values of $\alpha$ on the RAF-DB and FERPlus databases.




(a) RAF-DB (b) FERPlus

Fig. 4: Comparison of different competitors of the EC block at the different values of $\alpha$ on the RAF-DB and FERPlus databases.

**Effectiveness of the Weight Regularization Loss.** As shown in Table I, without using the weight regularization loss, Baseline+UE w.o. W gives much worse results than Baseline+UE, indicating the importance of the weight regularization loss in the UE block. By using the weight regularization loss, the model can give meaningful uncertainty weights for expression samples. These results are consistent with the experimental results on SCN [18].

**Effectiveness of the Expression Classification (EC) Block.** As shown in Table I, by employing the EC block, Baseline+UE+EC achieves better results than Baseline+UE. This indicates the importance of the EC block.

To further validate the superiority of label propagation on the hypergraph, we evaluate several competitors of the EC block. These competitors include the traditional label propagation (LP) [41], the simple relabeling method (CO) that performs relabeling by only using the classifier outputs as done in SCN, and the combination of the above two competitors

TABLE II: The average recognition accuracy (%) comparison on the RAF-DB and FERPlus databases with the different values of $\alpha$. The best results are boldfaced.

| Method | Venue | RAF-DB | | | | | FERPlus | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\alpha$=0.1 | $\alpha$=0.5 | $\alpha$=1 | $\alpha$=5 | $\alpha$=10 | $\alpha$=0.1 | $\alpha$=0.5 | $\alpha$=1 | $\alpha$=5 | $\alpha$=10 |
| Local | – | 88.76 | 65.53 | 51.18 | 42.57 | 42.75 | 83.90 | 72.19 | 66.38 | 59.63 | 54.96 |
| FedAvg | AISTATS 2017 | 86.10 | 72.76 | 58.05 | 54.44 | 60.83 | 86.76 | 66.70 | 75.42 | 70.93 | 68.81 |
| FedProx | MLSys 2020 | 87.16 | 73.82 | 55.35 | 52.92 | 53.19 | 87.27 | 68.62 | 76.85 | 66.77 | 67.47 |
| FedPer | ArXiv 2019 | 86.62 | 73.70 | 54.79 | 51.47 | 60.53 | 87.56 | 65.44 | 74.45 | 65.00 | 71.44 |
| pFedMe | NeurIPS 2020 | 88.12 | 61.43 | 51.70 | 41.46 | 38.17 | 69.24 | 68.48 | 51.56 | 41.43 | 36.90 |
| Ditto | ICML 2021 | 89.73 | 64.91 | 52.62 | 42.81 | 44.26 | 84.43 | 67.88 | 67.16 | 56.33 | 57.61 |
| FedAMP | AAAI 2021 | 84.59 | 56.19 | 42.10 | 48.71 | 45.36 | 74.82 | 59.62 | 59.46 | 56.37 | 56.55 |
| FedProto | AAAI 2022 | 82.76 | 61.09 | 50.58 | 42.79 | 44.26 | 80.54 | 67.35 | 64.90 | 58.91 | 56.41 |
| FedRep | CVPR 2023 | 85.57 | 78.64 | 57.08 | 55.32 | 62.61 | 86.47 | 71.28 | 76.90 | 75.11 | 72.16 |
| Baseline | – | 87.07 | 76.73 | 61.89 | 54.57 | 65.50 | 86.36 | 69.64 | 76.86 | 70.38 | 73.73 |
| Baseline+SCN | CVPR 2020 | 87.99 | 80.06 | 69.62 | 68.59 | 68.49 | 88.04 | 82.93 | 76.30 | 75.93 | 74.68 |
| Baseline+RUL | NeurIPS 2021 | 89.40 | 76.96 | 70.24 | 66.97 | 67.14 | 84.78 | 81.09 | 74.31 | 68.79 | 69.87 |
| Baseline+GUS | ArXiv 2022 | 87.04 | 65.15 | 56.62 | 54.95 | 51.54 | 73.77 | 59.77 | 56.58 | 51.91 | 55.13 |
| AMY (Ours) | – | **90.97** | **80.70** | **73.40** | **71.97** | **71.52** | **88.23** | **83.93** | **77.79** | **76.83** | **77.30** |

TABLE III: The classification accuracy (%) obtained by different values of $\lambda_1$ on the RAF-DB database.

| $\lambda_1$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| acc | 70.07 | 70.98 | 71.44 | 70.89 | 71.02 | 71.44 | 71.25 | **71.97** | 71.88 | 71.96 |

TABLE IV: The classification accuracy (%) obtained by different values of $\lambda_2$ on the RAF-DB database.

| $\lambda_2$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| acc | 69.97 | 70.76 | 70.20 | 68.92 | 70.81 | 69.56 | 70.76 | 71.21 | 71.34 | **71.97** |

(CO+LP). We also evaluate a variant (HLP) of the EC block, where label refinement is only based on label propagation on the hypergraph. The results are shown in Fig. 4. Traditional label propagation (LP) only employs pairwise relationships between expression samples, failing to comprehensively capture complex sample relationships. The relabeling method using classifier outputs (CO) neglects the relationships among expression data and depends solely on the model outputs for label refinement. Compared with these competitors, our method AMY consistently attains the best performance, highlighting the significance of leveraging high-order relationships between expression samples for label refinement. Note that our method achieves higher performance than HLP, showing the effectiveness of combining the label refinement results from both label propagation on the hypergraph and model predictions.

**Visualization of Uncertainty Weights.** We visualize the uncertainty weights obtained by SCN and our method AMY during the training stage on the RAF-DB database. The results are given in Fig. 5. Our method can give more reliable uncertainty weights than SCN. This further validates the effectiveness of the hypergraph network and the personalized uncertainty estimator for learning uncertainty weights.

**Influence of $\lambda_1$.** In Table III, we evaluate the influence of different values of $\lambda_1$ in Eq. (11) on the final performance. We can see that the model gives the best performance when the value of $\lambda_1$ is 0.8. This indicates the importance of weight regularization loss.

TABLE V: The classification accuracy (%) obtained by different values of $\eta$ on the RAF-DB database.

| $\eta$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| acc | 70.96 | **71.97** | 69.55 | 60.72 | 55.67 | 61.45 | 51.30 | 52.40 | 53.97 |

TABLE VI: The classification accuracy (%) obtained by different values of $K$ on the RAF-DB database.

| $K$ | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
|---|---|---|---|---|---|---|---|---|
| acc | 70.62 | 70.32 | **71.97** | 70.23 | 69.80 | 70.67 | 71.14 | 69.95 |

TABLE VII: The classification accuracy (%) obtained by different values of $\delta$ on the RAF-DB database.

| $\delta$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| acc | 63.14 | 69.37 | 62.16 | 70.89 | 70.90 | **71.97** | 70.56 | 71.64 | 71.74 |

**Influence of $\lambda_2$.** In Table IV, we conduct an ablation experiment on the influence of $\lambda_2$ in Eq. (11). From Table IV, we can observe that when the value of $\lambda_2$ becomes larger, the performance of our method is also improved. Hence, the class prototype regularization contributes significantly to enhancing the final performance.

**Influence of $\eta$.** In Table V, we conduct an ablation experiment on the influence of $\eta$ in Eq. (5) (which is used to distinguish certain samples and uncertain samples). From Table V, we can observe that the best performance is achieved when the value of $\eta$ is set to 0.2. When the value of $\eta$ is too small, it is difficult to distinguish between the two types of samples. If the value of $\eta$ is too large, the gap between the two types of samples becomes too significant, leading to incorrect uncertainty weight estimation.

**Influence of the Number of Neighbors ($K$) in the Hypergraph.** In Table VI, we validate the influence of different neighbor numbers ($K$) in the hypergraph on the performance. We can see that a larger value of $K$ does not necessarily result in better performance, since the relationships between samples may not be well captured with too many neighbors. The optimal performance for the hypergraph is achieved when the value of $K$ is set to 10.

TABLE VIII: The classification accuracy (%) obtained by the different numbers of layers $\upsilon$ in the HGNN on the RAF-DB database.

| $\upsilon$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| acc | 70.33 | **71.97** | 71.23 | 70.54 | 70.21 |

TABLE IX: The classification accuracy (%) obtained by different values of $\zeta$ on the RAF-DB database.

| $\zeta$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| acc | 68.19 | 70.21 | 70.40 | 70.53 | 71.21 | 70.87 | **71.97** | 70.46 | 68.43 |

**Influence of the Threshold $\delta$ for Updating Labels.** In Table VII, we evaluate the influence of the threshold $\delta$ in Eq. (9) for updating labels. We can see that when the value of $\delta$ is set to 0.6, our method achieves the best results. Higher thresholds may cause the problem that many noisy-labeled samples are not updated. Meanwhile, lower thresholds can result in many incorrect label refinements.

**Influence of the Number of Layers in the HGNN.** In the UE block, we leverage an HGNN to model high-order relationships between facial expression images. The number of layers in the HGNN can greatly influence the performance. We evaluate the number of layers in the HGNN on the final performance. The results are given in Table VIII.

We can see that our method gives the best performance when the number of layers in the HGNN is set to 2. Each layer of the HGNN aggregates feature information from each vertex and its neighborhoods. With the increasing number of layers, information can be aggregated from more distant vertices, allowing the model to capture broader contexts. On the one hand, when the number of layers is too small, the model fails to capture sufficient higher-order dependencies, leading to unreliable uncertainty estimation, particularly in scenarios where the clients suffer from heterogeneous sample uncertainty under personalized federated learning. On the other hand, when the number of layers is too large, the model easily suffers from overfitting since it learns information from irrelevant facial areas. Thus, the estimation of uncertainty weights is also unreliable.

**Influence of the Threshold for Separating the Certain Sample Set and Uncertain Sample Set.** In the UE block, we use a threshold to divide the whole samples into a certain sample set and an uncertain sample set. The certain sample set (whose uncertainty weights are low) contains high-quality expression samples that are beneficial for model training. On the contrary, the uncertain sample set (whose uncertainty weights are high) contains blurred or occluded samples, which can degrade the model performance. We evaluate the influence of different thresholds $\zeta$ on the performance. The results are shown in Table IX. Our method can obtain the best performance when the threshold is set to 0.7.

**Comparison Results on Centralized Learning.** In this section, we train our method on centralized learning, where only the UE block and EC block are used to train a model with all the training data. In Table X, we compare our method with SCN [18] and FRDL [25]. Note that, different from federated learning, a pretrained face model [18] is usually used in centralized learning. Therefore, we also report the results with the pretrained model. From Table X, we can see that our method can achieve better performance than SCN, indicating the feasibility of our method in centralized learning. Note that our method performs worse than the state-of-the-art FER method FDRL. This is because our method is a very lightweight model (mainly based on ResNet-18), which is desirable in federated learning (since the memory capacity of local devices may be limited). On the contrary, the state-of-the-art method FDRL relies on sophisticated network design, which may not be applicable in PF-FER. The above experiments further validate the effectiveness of our model in the personalized federated FER task.

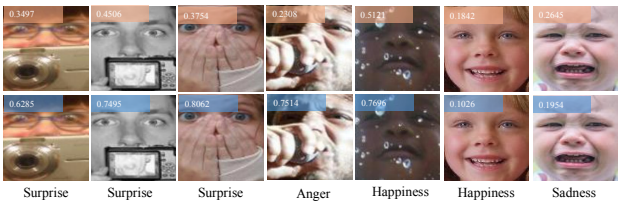



Fig. 5: Visualization of the uncertainty weights estimated by SCN (the first row) and our method AMY (the second row) on the RAF-DB database, where a larger weight indicates a higher degree of uncertainty for a sample.

TABLE X: Performance comparisons between several methods under centralized learning on the RAF-DB database. The classification accuracy (%) is reported.

| Method | RAF-DB | RAF-DB (pretrain) |
|---|---|---|
| SCN w.o. Relabel | 76.57 | 86.63 |
| AMY w.o. Relabel | 77.87 | 86.95 |
| SCN | 78.31 | 87.03 |
| AMY | 78.98 | 87.54 |
| FDRL | 80.12 | 89.47 |

### C. Comparison with State-of-the-Art Methods

We report the performance comparison between our method and several state-of-the-art methods in Table II. The state-of-the-art methods include traditional FL methods (FedAvg [11] and FedProx [29]), and representative personalized FL methods (FedProto [32], FedPer [42], Ditto [43], pFedMe [44], FedAMP [33], and FedRep [45]). In addition, we compare with personalized FL methods that incorporate uncertainty learning, including SCN [18], GUS [22], and RUL [20]. These methods also employ the same baseline (FedAvg+FedProto) as our method to ensure the fairness of our comparative experiments. All the competing methods are trained using publicly available codes under the same settings. The 'Local' method represents individual training for each client.

Our method AMY consistently outperforms the other competing methods. The 'Local' method performs poorly as it only trains the models locally, lacking knowledge from other clients. Some personalized FL methods (such as FedAMP) achieve worse results than FedAvg and FedProx. This can be ascribed to the simplicity of traditional FL methods. Both traditional FL and personalized FL methods do not fully consider

the challenge of heterogeneous sample uncertainty specific to the PF-FER task. Baseline+SCN estimates uncertainty using a single fully connected layer without considering relationships between samples. Baseline+GUS and Baseline+RUL only explore pairwise relationships between samples for uncertainty estimation, neglecting high-order relationships. On the one hand, our method addresses heterogeneous data across clients using class prototype regularization. On the other hand, our method captures complex relationships between samples using hypergraphs, which can be used for both uncertainty prediction and label refinement. The above results demonstrate that AMY is highly effective for FER in the context of personalized federated learning.

Note that in the traditional federated learning, a higher degree of data heterogeneity increases the difficulty of training the global model. The inconsistency in data distribution across clients can greatly influence the learning of a global model that aggregates information from local clients. However, in PF-FER, we focus on the local models. When data heterogeneity on the client is higher, the distribution of categories on that client may become more extreme (e.g., the client may only contain the images from the 'happy' category). Compared with a more balanced category distribution, the local model can be more easily fine-tuned in such an extreme case, leading to improved performance in the client. Therefore, our method at $\alpha = 1$ achieves better performance than our method at $\alpha = 5$ on FERPlus.

## V. Conclusion and Future Work

In this paper, we propose a novel method AMY, which learns personalized FER models over clients in a privacy-preserving manner, for PF-FER. AMY takes advantage of hypergraphs to model complex relationships between expression samples. Based on hypergraph modeling, the local model can give reliable uncertainty weights by a personalized uncertainty estimator in the UB block and generate high-quality label refinement results by label propagation in the EC block. As a result, our proposed method effectively addresses heterogeneous sample uncertainty across clients in PF-FER. Experiments on two challenging real-world facial expression databases validate the superiority of our method.

In our current method, we use a fixed threshold to divide the whole samples into a certain sample set and an uncertain sample set. However, relying on a single threshold may not effectively adapt to all clients. In addition, during the hypergraph label propagation process, we also use a fixed threshold to determine which samples need to be relabeled. In future work, we can use learnable parameters that can be adjusted based on the local data distribution and uncertainty, enabling more reasonable adaptation.

## References

[1] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Island loss for learning discriminative features in facial expression recognition," in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, 2018, pp. 302–309.

[2] D. Ruan, Y. Yan, S. Chen, J.-H. Xue, and H. Wang, "Deep disturbance-disentangled learning for facial expression recognition," in *Proceedings of the ACM International Conference on Multimedia*, 2020, pp. 2833–2841.

[3] H. Zhang, W. Su, J. Yu, and Z. Wang, "Weakly supervised local-global relation network for facial expression recognition," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2021, pp. 1040–1046.

[4] R. Mo, Y. Yan, J.-H. Xue, S. Chen, and H. Wang, "D$^3$Net: Dual-branch disturbance disentangling network for facial expression recognition," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 779–787.

[5] T. Liu, J. Li, J. Wu, L. Zhang, S. Zhao, J. Chang, and J. Wan, "Cross-domain facial expression recognition via disentangling identity representation," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2023, pp. 1213–1221.

[6] W. Xie, H. Wu, Y. Tian, M. Bai, and L. Shen, "Triplet loss with multistage outlier suppression and class-pair margins for facial expression recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 690–703, 2022.

[7] Y. Gu, H. Yan, X. Zhang, Y. Wang, Y. Ji, and F. Ren, "Toward facial expression recognition in the wild via noise-tolerant network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 5, pp. 2033–2047, 2023.

[8] D. Chen, G. Wen, H. Li, R. Chen, and C. Li, "Multi-relations aware network for in-the-wild facial expression recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3848–3859, 2023.

[9] H. Liu, H. Cai, Q. Lin, X. Li, and H. Xiao, "Adaptive multilayer perceptual attention network for facial expression recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6253–6266, 2022.

[10] X. Zhang, F. Zhang, and C. Xu, "Joint expression synthesis and representation learning for facial expression recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1681–1695, 2022.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[12] L. Zhang, G. Gao, and H. Zhang, "Spatial-temporal federated learning for lifelong person re-identification on distributed edges," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023.

[13] X. You, C. Liu, J. Li, Y. Sun, and X. Liu, "Fedmdo: Privacy-preserving federated learning via mixup differential objective," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2024.

[14] A. Salman and C. Busso, "Privacy preserving personalization for video facial expression recognition using federated learning," in *Proceedings of the International Conference on Multimodal Interaction*, 2022, pp. 495–503.

[15] M. S. B. Siddiqui, S. A. Shusmita, S. Sabreen, and M. G. R. Alam, "FedNet: Federated implementation of neural networks for facial expression recognition," in *Proceedings of the International Conference on Decision Aid Sciences and Applications*, 2022, pp. 82–87.

[16] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *Proceedings of the World Conference on Smart Trends in Systems, Security and Sustainability*, 2020, pp. 794–797.

[17] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2852–2861.

[18] K. Wang, X. Peng, J. Yang, S. Lu, and Y. Qiao, "Suppressing uncertainties for large-scale facial expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6897–6906.

[19] J. She, Y. Hu, H. Shi, J. Wang, Q. Shen, and T. Mei, "Dive into ambiguity: Latent distribution mining and pairwise uncertainty estimation for facial expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6248–6257.

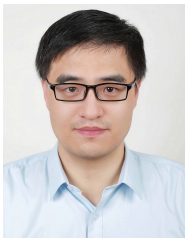
[20] Y. Zhang, C. Wang, and W. Deng, "Relative uncertainty learning for facial expression recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 616–17 627, 2021.

[21] Y. Yang, L. Hu, C. Zu, Q. Zhou, X. Wu, J. Zhou, and Y. Wang, "Facial expression recognition with contrastive learning and uncertainty-guided relabeling," *International Journal of Neural Systems*, vol. 33, p. 2350032, 2023.

[22] J. Lei, Z. Liu, Z. Zou, T. Li, X. Juan, S. Wang, G. Yang, and Z. Feng, "Mid-level representation enhancement and graph embedded uncertainty suppressing for facial expression recognition," *arXiv preprint arXiv:2207.13235*, 2022.

[23] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2016, pp. 1–10.
[24] Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong, "Identity-aware convolutional neural network for facial expression recognition," in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, 2017, pp. 558–565.
[25] D. Ruan, Y. Yan, S. Lai, Z. Chai, C. Shen, and H. Wang, "Feature decomposition and reconstruction learning for effective facial expression recognition," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7660–7669.
[26] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
[27] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
[28] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data *et al.*, "A field guide to federated optimization," *arXiv preprint arXiv:2107.06917*, 2021.
[29] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.
[30] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-iid data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5972–5984, 2021.
[31] D. Shome and T. Kar, "FedAffect: Few-shot federated learning for facial expression recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4168–4175.
[32] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "FedProto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8432–8440.
[33] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 7865–7873.
[34] Y. Niu and W. Deng, "Federated learning for face recognition with gradient correction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 1999–2007.
[35] C.-T. Liu, C.-Y. Wang, S.-Y. Chien, and S.-H. Lai, "FedFR: Joint optimization federated framework for generic and personalized face recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 1656–1664.
[36] F. Chen, J. Shao, S. Zhu, and H. T. Shen, "Multivariate, multi-frequency and multimodal: Rethinking graph neural networks for emotion recognition in conversation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 761–10 770.
[37] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 3558–3565.
[38] Y. Zhang, N. Wang, Y. Chen, C. Zou, H. Wan, X. Zhao, and Y. Gao, "Hypergraph label propagation network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 6885–6892.
[39] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training deep networks for facial expression recognition with crowd-sourced label distribution," in *Proceedings of the ACM International Conference on Multimodal Interaction*, 2016, pp. 279–283.
[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[41] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5070–5079.
[42] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.
[43] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning*, 2021, pp. 6357–6368.
[44] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 394–21 405, 2020.
[45] Y.-R. Yang, K. Wang, and W.-J. Li, "FedRep: A byzantine-robust, communication-efficient and privacy-preserving framework for federated learning," *arXiv preprint arXiv:2303.05206*, 2023.

**Hu Ding** is currently pursuing the master's degree with the School of informatics, Xiamen University, China. His research interests include facial expression recognition and federated learning.



**Yan Yan** (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from Tsinghua University, China, in 2009. He worked as a Research Engineer with the Nokia Japan Research and Development Center from 2009 to 2010. He worked as a Project Leader with the Panasonic Singapore Laboratory in 2011. He is currently a Full Professor with the School of Informatics, Xiamen University, China. He has published around 100 papers in the international journals and conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *IJCV*, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, CVPR, ICCV, ECCV, AAAI, and ACM MM. His research interests include computer vision and pattern recognition.

**Yang Lu** (Member, IEEE) received the B.Sc. and M.Sc. degrees in software engineering from University of Macau, Macau, China, in 2012 and 2014, respectively, and the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong, China, in 2019. He is currently an Assistant Professor with the Department of Computer Science and Technology, School of Informatics, Xiamen University, Xiamen, China. His current research interests include machine learning, deep learning, federated learning and long-tail learning.

**Jing-Hao Xue** (Senior Member, IEEE) received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998, and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is currently a Professor with the Department of Statistical Science, University College London. His research interests include statistical pattern recognition, machine learning, and computer vision. He received the Best Associate Editor Award of 2021 from the IEEE Transactions on Circuits and Systems for Video Technology, and the Outstanding Associate Editor Award of 2022 from the IEEE Transactions on Neural Networks and Learning Systems.

**Hanzi Wang** (Senior Member, IEEE) is currently a Distinguished Professor of "Minjiang Scholars" in Fujian province and a Founding Director of the Center for Pattern Analysis and Machine Intelligence at Xiamen University, China. He received his Ph.D. degree in Computer Vision from Monash University, where he was awarded the Douglas Lampard Electrical Engineering Research Prize and Medal for the best Ph.D. thesis. His research interests include computer vision and pattern recognition.