FedHAP: Federated Hashing With Global Prototypes for Cross-Silo Retrieval

Meilin Yang*, Student Member, IEEE, Jian Xu*, Student Member, IEEE, Wenbo Ding†, Member, IEEE, and Yang Liu‡, Senior Member, IEEE

Abstract—Deep hashing has been widely applied in large-scale data retrieval due to its superior retrieval efficiency and low storage cost. However, data are often scattered in data silos with privacy concerns, so performing centralized data storage and retrieval is not always possible. Leveraging the approach of federated learning (FL) to perform deep hashing is a recent research trend. However, existing frameworks mostly rely on the aggregation of the local deep hashing models, which are trained by performing similarity learning with local skewed data only. Therefore, they cannot work well for non-IID clients in a real federated environment. To overcome these challenges, we propose a novel federated hashing framework that enables participating clients to jointly train a shared deep hashing model by leveraging the class-wise prototypical hash codes. Globally, sharing global prototypes with only one prototypical hash code per class assists in learning consistent code distributions across clients while minimizing the cost of communication and privacy. Locally, the use of the global prototypical hash codes are maximized by jointly training a discriminator network and the local hashing network. Extensive experiments on benchmark datasets are conducted to demonstrate that our method can significantly improve the performance of the deep hashing model in the federated environments with non-IID data distributions.

Index Terms—Federated learning, information retrieval, deep hashing.

I. INTRODUCTION

With the explosive increase of data generated from different institutions, achieving fast and storage-saving information retrieval across multiple institutions has attracted much attention in recent years. Deep hashing is a widely used method that aims to reduce storage cost and improve retrieval efficiency by encoding data points into non-invertible and compact binary hash codes with deep neural networks (DNNs) [1], [2]. Most existing deep hashing methods assume that data storage is centralized. For example, TDHPIIR [3] is an efficient privacy-preserving image retrieval method based on deep hashing, in which data owners upload the encrypted data set to the central cloud server, and provide data retrieval services via the indexes established from the database. However, centralized storage of private client data is not always feasible due to space limitations as well as increasing privacy concerns and regulations such as GDPR. Therefore, it is increasingly desirable to learn to hash over distributed data and enable cross-silo retrieval with privacy protection.

An intuitive way to achieve these goals is to keep the private data of each client local and jointly train a deep hashing model without exchanging raw data between clients. Federated learning (FL) [4] has emerged as a promising paradigm for such scenarios in recent years. In the original FL framework [5], the selected clients first locally perform multiple training epochs by stochastic gradient descent (SGD) and then transmit their model updates to a central server, where the model updates are aggregated to obtain a new global model. The global model is then sent back to clients for the next training iteration. This algorithm is called FedAvg. Once the training process is finished, the original query data will be converted into hash codes through the deep hashing model obtained from the federated learning process. Then, by calculating the distance of the hash pairs between the data to be queried and the data owned by each client, we can find the target client with data having the smallest similarity distance so as to realize the joint retrieval among distributed clients. Federated hashing can play an important role in many realistic applications. For example, in the medical field, doctors need to utilize the private data of patients such as X-ray images to retrieve the most similar or valuable cases for diagnosis assistance and disease assessment. As shown in Fig. 1, the inquirer converts the original image into the hash code using the shared deep hashing model obtained by joint training, and then broadcasts the hash code to find the target medical institution holding the image with the smallest hashing distance. Finally, the target medical institution encrypts the requested data and transmits it to the inquirer.

*Corresponding authors: Wenbo Ding; Yang Liu.

Meilin Yang and Jian Xu are with the Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University, Beijing 100190, China (e-mail: yml21@mails.tsinghua.edu.cn; xujian20@mails.tsinghua.edu.cn).

Wenbo Ding is with the Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University, Beijing 100190, China, also with the RISC-V International Open Source Laboratory, Shenzhen 518055, China, and also with Shanghai Artificial Intelligence Laboratory, Shanghai 201210, China (e-mail: ding.wenbo@sz.tsinghua.edu.cn).

Yang Liu is with the Institute for AI Industry Research (AIR), Tsinghua University, Beijing 100190, China, and also with Shanghai Artificial Intelligence Laboratory, Shanghai 201210, China (e-mail: liuyi03@air.tsinghua.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPDS.2023.3324426, provided by the authors.

Digital Object Identifier 10.1109/TPDS.2023.3324426

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Previous works [6, 7] have explored the combination of deep hashing with FL and demonstrated their effectiveness. However, these works do not address adequately the non-IID nature of data in federated hashing environments. Non-IID denotes the client-level non-identical client distributions which includes label distribution skew and feature distribution skew, etc. Ignoring the non-IID nature of data and simply relying on the model aggregation to achieve global hashing learning will lead to a catastrophic drop in model performance. For example, in a patient hashing problem across multiple hospitals, the distributions and quantities of patients from an oncology hospital and a psychiatric hospital can be quite different. Since the core of deep hashing models is similarity learning, skewed and highly imbalanced local data distributions can result in biased local models that cannot be sufficiently corrected by the FedAvg algorithm.

To tackle the above issues, in this paper, we introduce a federated deep hashing method with global contrastive prototypes (FedHAP) for cross-silo retrieval. In the FedHAP framework, each client in the federation can jointly train the hashing model using its own local data and global prototypical hash codes of each class to guide the local training. Specifically, the global prototypical hash codes are generated in the server by aggregating the local class-wise feature prototypes from clients. Note that such a aggregation can also be conducted in a secure manner by using homomorphic encryption [8] to further prevent server to exploit the plain-text hash codes and other auxiliary information to infer sensitive properties from clients’ data. Then the prototypical hash codes, together with the global model, are broadcast to clients for local training. To better utilize the global prototypical hash codes, we not only design similarity learning algorithms with supervision from the global hash codes, but also creatively leverage a discriminator network to enhance the distribution consistency between the locally generated binary hash codes and the global ones. By this way, we maximize the usage of global prototypical hash codes to learn consistent hash codes across clients without exchanging sample-level information, thereby significantly improving the performance of the federated hashing model while preserving the privacy of local data.

As an example for demonstrating the efficacy of our proposed FedHAP, we compare the hash codes generated by our approach and the naïve FedAvg approach as in Fig. 2, where each data point is visualized using t-SNE [9]. It can be observed that the hash codes learned by FedHAP exhibit favorable intra-class compactness and inter-class separability compared with FedAvg [10] over the deep supervised hashing model DPSH [11]. It is worth noting that our approach works especially well for under-represented classes with fewer samples, as demonstrated by the clear discrimination of the two categories in blue and purple. To the best of our knowledge, this is the first work that introduces the global prototypical hash codes to assist the hashing model learning for cross-silo retrieval. We provide both theoretical justification and empirical verification for the effectiveness of using prototypes. The main contributions of this paper are summarized as follows:

- We present a novel federated supervised hashing learning framework named FedHAP for efficient and effective cross-silo retrieval. This framework integrates hashing learning with federated learning and leverages the global prototypical hash codes to enhance the performance of the global hashing model with minimal impact on communication and privacy.
- We introduce local adversarial learning to further address the non-IID challenge and improve the overall performance, which can align the distributions of local learned hash codes by enforcing the semantic consistency between the local hash codes and the shared global prototypical hash codes.
- Experimental results on multiple benchmark datasets demonstrate that our approach outperforms existing methods in both IID and non-IID FL scenarios. Furthermore, we verify the efficacy of each component in our method by ablation studies.

The remainder of this article is organized as follows. We first present an overview of deep hashing, federated learning and federated hashing in Section II. In Section III, we present the details of our proposed FedHAP framework and provide analyses on each component. We describe the experimental settings and evaluation results in Section IV. Lastly, Section V concludes the main idea and contributions of this paper.

II. RELATED WORK

A. Deep Hashing

Deep hashing methods can map sensitive data into compact binary codes irreversibly, leading to less memory consumption...
and short query time [12]. Existing deep hashing methods [13], [14], [15], [16], [17] are proposed to learn data representations in binary codes that preserve the locality and similarity of data samples. By coupling data feature extraction and binary code learning, these methods have been shown to greatly improve retrieval accuracy. These methods can be generally categorized into deep unsupervised hashing and deep supervised hashing. Unsupervised hashing methods [18], [19], [20], [21] learn hashing functions that map input data points into binary codes by exploiting the similarity distances of samples. Supervised hashing methods [11], [13], [22], [23], [24], [25], [26] aim to further exploit available supervised information (such as labels or the semantic affinities of training data) to improve performance. In recent years, deep supervised hashing methods have attracted more attention as it can achieve better accuracy than deep unsupervised hashing.

B. Federated Learning

FL was first proposed by Google in 2016 [4], and has emerged as a paradigm for distributed training of machine learning models without centralizing data. In FL, multiple clients collaboratively learn a model under the coordination of a central server while preserving their data privacy. In the most popular method for FL, i.e., FedAvg [27], clients perform multiple local updates before sending their updates for aggregation in the server, which only works well for IID data distributions. FedProx [10] is presented as a generalization and re-parametrization of FedAvg, and it achieves stable and accurate convergence behavior compared to FedAvg in highly heterogeneous settings. To address the non-IID nature of FL, various algorithms have been proposed since then [28], [29], [30], [31], [32], [33]. FedProc [32] designs a local network architecture and a contrastive loss to regulate the training of local models with the class-wise logits transmission. However, unprocessed raw intermediate logits may cause leakage of the original data and local data distribution. Knowledge Distillation based federated frameworks [28], [34] have recently emerged to tackle the non-IID issue by refining the server model using aggregated knowledge from heterogeneous users, which may need a proxy dataset or require the client to provide the server with the label distribution.

C. Federated Hashing

Recently, the framework of federated learning has been applied to hashing for various tasks [6], [7], [35], [36]. FRecLSH [36] present a locality sensitive hashing (LSH) based method which uses LSH as an index technique for fast and accurate recommendation so as to make use of data from different sources. FedMLH [35] proposes a method which utilize the label hashing technique to reduce the model size and the communication cost on the federated classification tasks. For the retrieval tasks, Federated Patient Hashing (FPH) [6] has been proposed to collaboratively train a patient information retrieval model stored in a shared memory while keeping all the patient-level information in local clients. Federated Cross-Modal Retrieval (FedCMR) [7] is the first attempt to combine federated learning with cross-modal retrieval. While the abovementioned methods have certainly proved the feasibility of federated hashing to some extent, they did not exploit the relationship between the global hashing model and the local hashing model in the FL framework, nor did they sufficiently address the prominent non-IID problem in the federated environments, such as label distribution skewness. Different from these methods, our approach extracts only a few class-wise global prototypical hash codes, which greatly avoids the leakage of data and label distribution. The class-wise global prototypical hash codes will participate in the local training process and alleviate the model drift issue. Furthermore, illuminated by adversarial learning, we design a discriminator network to further bridge the global and local generation of hash codes.

III. THE PROPOSED METHOD

In this section, we first present the problem definitions of deep hashing learning. Then, we introduce the details of our FedHAP framework and provide analysis on each of the design components. Related notations are listed in Table I.

A. Problem Formulation

Without losing generality, we assume there are $m$ clients and the $i$th client’s private data are denoted as $D_i = \{x_j\}^n_{j=1}$, where $x_j \in \mathbb{R}^{1 \times d}$ is the input data point, $n_i$ is the number of data in client $i$, and $d$ is the input dimension (e.g., the number of pixels in an image). All clients collaboratively train a hashing model $G$ parameterized by $\theta$, which maps the input data $x_j$ into output feature embedding $h_j \in \mathbb{R}^{1 \times K}$, where $K$ is the number of bits in the Hamming space. The hash code of $x_j$ is denoted as $h_j$ and can be gained by $h_j = \text{sign}(b_j)$, where $\text{sign}$ denotes the element-wise sign function. Let $H_i = \{h_j\}^n_{j=1}$, the problem to tackle here is for $m$ clients to collaboratively train a deep hashing model $G_\theta$ without exposing their data $D_i$, so that the similarities

| Variables | Meaning |
|----------|---------|
| $D_i$    | Local data set of client $i$ |
| $b_j$    | Feature embedding of sample $j$ |
| $h_j$    | Hash code of sample $j$ and $h_j = \text{sign}(b_j)$ |
| $B_i$    | The collection of sample features of client $i$ |
| $H_i$    | The collection of hash codes of client $i$ |
| $\mathcal{B}$ | Global feature prototypes |
| $\mathcal{H}$ | Global prototypical hash codes |
| $K$      | Length of Hash code |
| $C$      | Number of classes |
| $L_s$    | Similarity preserving loss |
| $L_p$    | Prototypical-contrastive loss |
| $L_d$    | Adversarial loss |
| $G$      | Hashing model |
| $D$      | Discriminator network |
| $\theta$ | Parameters of $G$ |
| $\phi$   | Parameters of $D$ |
among all data sample pairs are preserved as follows:

$$\min_{\theta} \sum_{i=1}^{m} F_i(D_i, \mathcal{H}_i; \theta).$$

(1)

### B. Local Adversarial Prototypical-Contrastive Training

In our framework, the loss function for local hashing learning consists of three parts, including the similarity preserving loss, prototypical contrastive loss and adversarial loss. In the following, we will provide detailed explanations and discussions for each loss component.

1) **Similarity Preserving Loss ($L_s$):** First, in order to find a local hashing model that preserves the local similarity structure between the original and mapped spaces, a local similarity preserving loss is required. Here, we propose a general federated hashing framework that can adopt different kinds of similarity preserving losses. That is, $L_s$ can vary based on the selected deep hashing methods. Here, we introduce triplet ranking loss in detail as an example and list the rest of the hashing methods adopted in Appendix C, available online. We suppose there is a set of hash-encoded triplets $(b_j, b_{j+}, b_{j-})$ where $b_{j+}$ is a positive pair of $b_j$, meaning that $b_j$ and $b_{j+}$ have the same class, and $b_{j-}$ is a negative pair of $b_j$. Denote $y$ as a binary flag, which is 0 for a negative pair and 1 for a positive pair. The similarity between the codes can be evaluated using a general distance metric $d(\cdot, \cdot)$, e.g., Euclidean distance and cosine distance. Based on these, the triplet ranking loss for a triplet $(b_j, b_{j+}, b_{j-})$ are formulated as

$$L_{\text{triplet}}^{i,j} = \max \left( d(b_j, b_{j+}) - d(b_j, b_{j-}) + a, 0 \right),$$

where $m$ and $a$ are margin parameters. Therefore

$$L_s^i = \sum_j L_{\text{triplet}}^{i,j}. \hspace{1cm} (2)$$

Note that $L_s^i$ is only calculated on local private data. It is also worth noting that the $d(b_j, b_{j+})$ can approximate the distance between the binary hash codes $d(h_j, h_{j+})$ when a penalty term is added to the continuous outputs of the hashing network during training to relax the binary constraints, which is often formulated as $\|b_j - 1\|_1$ or $\|b_j\|$ with tanh activation. This term can be seen as a kind of quantization technique which is widely used in hashing learning [37].

2) **Prototypical-Contrastive Loss ($L_p$):** Since the distribution of local data could be highly skewed from the global distribution, the locally trained models have high overfitting risks and will also result in a sub-optimal global model in the FedAvg algorithm. Therefore, to enhance the local hashing learning by leveraging data from other clients, we further introduce a global prototypical triplet loss $L_p^i$ as shown in (4). Here, the global prototypes collectively are denoted as $\hat{B} = \{\hat{b}_c\}_{c=1}^C$, where $C$ is the number of classes and $\hat{b}_c$ is the prototype for class $c$. In (4), $\hat{b}_c$ denotes the global prototype that is of the same class as $b_j$ while $\hat{b}_{j+}$ denotes the global prototype that is of a different class from $b_j$.

$$L_p^i = \sum_{j,j^+,j^-} \max \left( d(b_j, \hat{b}_{j+}) - d(b_j, \hat{b}_{j-}) + a, 0 \right). \hspace{1cm} (4)$$

We provide the following theorem to demonstrate that under mild assumptions, the summation of $L_p^i$ over clients is a surrogate loss for the triplet loss over the whole dataset. Therefore, the adverse impact of non-IID local data is mitigated by the prototypical-contrastive loss.

**Theorem 1:** Suppose that the union of local datasets is overall class balanced and the squared $L_2$-norm is used as the distance metric, then the summation of objective (4) over all clients is equivalent to the centralized empirical risk

$$\sum_{i=1}^{m} L_p^i = L_{\text{global}}, \hspace{1cm} (5)$$

where the centralized empirical risk is defined as follows

$$L_{\text{global}} \triangleq \sum_{j,j^+,j^-} \max \left( d(b_j, b_{j+}) - d(b_j, b_{j-}) + a, 0 \right),$$

and the summation is taken over the samples from all clients. 

**Proof:** The detailed proof is provided in Appendix A, available online.

However, obtaining the precise global prototypes in each model training step could be infeasible since multiple local training epochs will be conducted before global aggregation and sharing the precise data size of each class in each client might also be considered to violate the privacy in FL. Fortunately, as our task is learning binary codes, we can get a good approximation of the global prototypes by using the prototypical hash codes, which are exactly the binarized forms of the feature prototypes. Such quantization step is commonly adopted in deep hashing models [19], [38], which helps adjust the parameters of the deep hashing model to minimize discrepancies between the original data and the hash codes [38]. Additionally, the binary approximation further protects the original prototypes from being reversed.

**Obtain Global Prototypical Hash Codes $\hat{H}$:** Now we will explain how to generate the prototypical hash codes. As shown in Fig. 4, each client first aggregates their feature embeddings in a class-wise manner by

$$\hat{b}_{i,c} = \sum_{j=1}^{n_{i,c}} b_{j,c},$$

(7)

where $n_{i,c}$ indicates the number of data samples of class $c$ in client $i$ and $b_{j,c}$ represents the output feature vector of the $j$th data of class $c$. Next, clients send their class-level feature vectors $\hat{B}_i = \{\hat{b}_{i,c}\}_{c=1}^C$ to the server which then performs the aggregation and binarization

$$\hat{h}_c = \text{sign} \left( \hat{b}_c \right) = \text{sign} \left( \sum_{i=1}^{m} \hat{b}_{i,c} \right). \hspace{1cm} (8)$$

The prototypical hash codes are collected as $\hat{H} = \{\hat{h}_c\}_{c=1}^C$ and used to substitute the $B$ to calculate the loss by

$$L_p^i = \sum_{j,j^+,j^-} \max \left( d(b_j, \hat{h}_{j+}) - d(b_j, \hat{h}_{j-}) + a, 0 \right). \hspace{1cm} (9)$$

3) **Adversarial Loss ($L_d$):** Simply applying triplet-based loss might not be enough to align the local-global hash codes as
the loss will disappear when distance between the positive pair is smaller than that of the negative pair by a margin $a$. Inspired by the effectiveness of adversarial training in achieving consistent feature distributions in domain adaptation and other fields, we further introduce a local discriminator network $D$ with trainable parameters $\phi$ to enhance the consistency of local and global distributions of hash codes. The discriminator network is initially used in adversarial learning to identify whether the data come from a real dataset or a neural network [39] and its output is the probability that the input data come from the real dataset. In this paper, we treat the global hash codes $H = \{h_i\}_{i=1}^C$ as the real dataset and the hash codes $\hat{H}_i = \{\hat{h}_i\}_{i=1}^C$ generated by the local hashing model as the latter. We utilize the local labels $Y_i = \{y_j\}_{j=1}^n$ and global labels $\hat{Y} = \{\hat{y}_c\}_{c=1}^C$ as constraints on $H_i$ and $\hat{H}$ to realize the discrimination of hash codes of a specific class. Specifically, we use the one-hot vector of the class label as extra information and concatenate it with $H_i$ and $\hat{H}$ respectively as the input vectors of $D$, and the output is the probability score (between 0 and 1). The output closer to 1 denotes that the input data is more likely to come from the global prototypical hash codes. This adversarial training process is shown in Fig. 5. The adversarial loss $\mathcal{L}_d^i$ of client $i$ can be written as follows:

$$\mathcal{L}_d^i = \left(-\frac{1}{n_i} \sum_{j=1}^{n_i} (1 - \log(D_{\phi}(h_j|y_j))) + \frac{1}{C} \sum_{c=1}^{C} \left( \log(D_{\phi}(\hat{h}_c|\hat{y}_c)) \right) \right),$$

(10)

where the first term in the above equation is the cross-entropy loss for the local dataset, followed by the cross-entropy loss for the global prototypical hash codes.

4) Overall Local Objective: To maximize the performance of hash learning, we take into account all three loss components and the overall local loss function $F_i$ can be obtained by combining (3), (9) and (10), formulated as follows:

$$F_i = \rho \cdot \mathcal{L}_s^i + (1 - \rho) \cdot \mathcal{L}_p^i + \lambda \cdot \mathcal{L}_d^i,$$

(11)

where $\rho$ and $\lambda$ are penalty parameters to balance different loss components. Specially, when we take (11) as the global objective, $\mathcal{L}_s^i$ and $\mathcal{L}_d^i$ can actually be regarded as the assistant terms that help to minimize the global loss. It is worth noting that the convergence analysis for the overall loss is provided in the Appendix B, available online, which offers a rigorous and detailed explanation of the convergence behavior of the proposed method.

**Algorithm 1:** FedHAP.

**Input:** Image set $X$, Number of clients $m$, Hashing model $H$, Discriminator network $D$, communication rounds $T$, Local training epochs $E$.

**Initialize:** $\theta_0$, $\phi_0$, $\mathcal{H}_0$.

1: for $t = 0$ to $T - 1$ do

2: Server broadcasts global model parameters of $\theta_t$, $\phi_t$ and prototypical hash codes $\mathcal{H}_t$ to each client $i$.

3: for each client $i$ in parallel do

4: for $e = 0$ to $E - 1$ do

5: Calculate adversarial loss $\mathcal{L}_d^i$ with (10)

6: Update $\phi_t^i$ using back propagation

7: Update $\theta_t^i$ using back propagation

9: end for

10: Send local model parameters $\theta_t^i$, $\phi_t^i$ and class-wise feature vectors $B_t^i$ to the central server

11: end for

12: Server executes:

13: Update global hashing model $\theta_{t+1} = \frac{1}{m} \sum_{i=1}^{m} \theta_t^i$

14: Update global discriminator $\phi_{t+1} = \frac{1}{m} \sum_{i=1}^{m} \phi_t^i$

15: Update global prototypical hash codes $\mathcal{H}_{t+1}$

16: end for

Fig. 3. Framework of our proposed FedHAP.
C. FedHAP Framework

For all clients to learn the above hashing model collaboratively, we propose FedHAP, which is shown in Fig. 3 and Algorithm 1. First, each client \(i\) performs local training of the deep hashing model and the discriminator network with its local data and global prototypical hash codes. Especially, during each local training step, the original input of data will be converted into low-dimensional features by the convolutional neural network and the hashing learning module, which are then used to compute the similarity preserving loss \(L^s\) and prototypical triplet loss \(L^p\) with the guidance of global prototypical hash codes. Next, the feature embedding will be converted into binary hash codes using the sign function. Furthermore, the local and global hash codes with their semantic labels will be simultaneously fed into the discriminator network to generate the corresponding adversarial loss \(L^d\). Then, the parameters of the hashing network and discriminator network will be updated by backpropagation. After multiple local rounds of training, clients upload the locally updated model parameters \(\theta^i\), \(\phi^i\) and the locally generated prototypical features \(\mathbf{B}_i = \{\mathbf{B}_i^{c}\}_{c=1}^C\) to the central server. The central server is responsible for aggregating \(\{\theta^i\}_{i=1}^m\), \(\{\phi^i\}_{i=1}^m\) and \(\{\mathbf{B}_i\}_{i=1}^m\) received from clients and then delivering the aggregated models \(\theta, \phi\) and \(\mathcal{H}\) to them for the next training round until convergence. Note that our proposed FedHAP only requires the transmission of class-wise prototypes between clients and servers, so that the sample-level privacy of each client is still preserved, which is the primary goal of federated learning. The transmission of class-level data is also considered reasonable in previous works [32], [40]. Moreover, the class-level hash codes are made of low-dimensional vectors, e.g., a 12-bit vector containing only 1 or 0, and the binary codes cannot be reversed. Therefore, the raw data can not be recovered from the global prototypical hash codes. Finally, we want to point out that our framework can be easily combined with secure computation techniques such as homomorphic encryption and secure aggregation to further improve privacy protection.

IV. EXPERIMENTS

In this section, we conduct extensive experiments to verify the effectiveness of our proposed approach in the non-IID scenarios and compare it with other robust federated learning frameworks. We evaluate all methods on three benchmark datasets, including NUS-WIDE [41], MIRFlickr25K [42], MS-COCO [43]. A set of ablation experiments are also designed to further verify the individual efficacy of different components. In addition, we applied our method to a medical dataset for further demonstration of the effectiveness of FedHAP in medical information retrieval tasks.

A. Experimental Setup

1) Datasets: NUS-WIDE contains 269,648 web images with textual tags and we use the images associated with the 21 most frequent concepts, where each of these concepts associates with at least 5,000 images, resulting in a total of 195,834 images. A total of 2,100 data pairs in the dataset are selected randomly as the query set and the remainder of the dataset is used as the retrieval database. MIRFlickr25K is a commonly used dataset consisting of 25000 images and 25,000 image-text pairs that were downloaded from the social photography site Flickr.com. In our experiment, we select 20,015 data points in total, among which 10,000 pictures are randomly selected for training. For the remaining data, 2000 data pairs are selected randomly as the query set and the rest is used as the retrieval database. MS-COCO 2014 originates from the Microsoft COCO dataset, and the 2014 release of MS-COCO contains 82,783 training, 40,504 validation, and 40,775 testing images (approximately 1/2 train, 1/4 val, and 1/4 test). We randomly select 4992 pairs for the query set and leave the remaining pairs as the retrieval database. In addition, 10,000 pairs are randomly selected from the retrieval database for training. VinDr-CXR [44] is currently the largest public CXR dataset with radiologist-generated annotations. VinDr-CXR contains 67914 images which are used for classification tasks of thoracic lesions and diseases on CXR scans. We randomly select 2000 pairs for the query set and 10,000 pairs are randomly selected for training.

2) Baselines and Settings: We compare FedHAP with the following benchmark federated learning frameworks: FedAvg [10], FedProx [27], FedCMR [7], MOON [29]. FedCMR is proposed as a federated hashing framework and MOON is proposed to handle non-IID local data distributions across clients. Moreover, We evaluate the performance of FedHAP across multiple deep hashing algorithms: DCH [24], DPSH [11], Greedy Hash [25], OrthoCos [45]. These deep hashing algorithms are deployed under the abovementioned benchmark federated learning frameworks respectively and the parameter settings are based on the original papers. To ensure a fair comparison, the feature extraction networks of all the deep hashing algorithms are derived from CNN-F [46], which has been pre-trained on the ImageNet dataset [47] in order to extract a 4,096-dimensional
representation vector for each data point. The discriminator network of FedHAP is a two-layer feed-forward neural network.

The experiments are conducted in both IID and non-IID scenarios, where the training settings of each scenario are identical for all baselines and our method. We consider a federated learning setup with $m = 20$ participating clients. For the IID scenarios, we simulate the IID data distributions by randomly and evenly partitioning the shuffled training sets into 20 clients, and thus each client is assigned with data from a uniform distribution. For the non-IID scenarios, as previous works [10], [27], our work has focused on label distribution skew, where a non-IID dataset is formed by partitioning a class-balanced dataset based on the labels. Specifically, the data are sorted by class and each client receives a data shard that contains samples belonging to a randomly selected subset of classes. It is worth noting that this partitioning method can result in a deeper heterogeneity of data samples across clients than Dirichlet distribution-based partitioning [29].

In our experimental setup, the number of training iterations was determined dynamically based on the convergence of the model. We monitored the training process and stopped the iterations when the model demonstrated convergence. The number of local epochs is set to 5 for all clients, regardless of the data distribution being non-IID or IID. In the non-IID scenarios, the data category owned by each client is set to 3. Adam [48] is employed as the local optimizer, and the initial learning rate is set to 0.005 with batch size = 128. The detailed settings of each dataset are summarized in Table II. To find a better combination of hyper-parameters in our method, we conduct sensitivity analysis of these hyper-parameters and achieve high results with $\rho = 0.5$ and $\lambda = 0.05$.

3) Evaluation Metric: Hamming ranking [49] is a classical retrieval method that is used to evaluate the performance of the image retrieval task. In our experiments, we evaluate the retrieval quality based on mean Average Precision (mAP) [50] which is the arithmetic mean of the average precision values for the retrieval tasks. Standard evaluation metrics such as precision-recall curves (PR) are also provided. For a fair comparison, all methods use identical training and testing sets.

### B. Performance Evaluation

1) Results in the Non-IID and IID Scenarios: As we focus on the framework assessment in the non-IID scenarios, we begin with presentation and discussion of the results in such settings. Tables III–VI list the mAP results of DCH, DPH, Greedy-Hash and OrthoCos algorithms over the different federated learning frameworks, respectively. Compared with the existing frameworks, our federated hashing learning framework consistently improves the mAP results on NUS-WIDE, MIRFlickr and MS-COCO. Specifically, our method raises the averaged mAP over the abovementioned algorithms by approximately 4.66-14.38%, 0.75-9.80% and 1.02-8.16% on NUS-WIDE, MIRFlickr and MS-COCO @ 12 bits. Moreover, extensive retrieval performance results on MIRFlickr25K and NUS-WIDE with regard to precision-recall curves (PR), precision curves and recall curves with respect to different numbers of top returned samples are plotted in Figs. 6 and 7. Those curves demonstrate that FedHAP outperforms baseline methods impressively, which is desirable for practical precision-first retrieval. Specifically, FedHAP achieves higher precision when the recall levels are low or the number of retrieved samples is small. Besides, we also verify the performance of IID scenarios.

Tables VII and VIII show the mAP results in the IID scenarios based on the Ortho and GreedyHash algorithms on MIRFlickr and MS-COCO datasets at different hashing bits. The results demonstrate that FedHAP can also outperform existing federated hashing frameworks in the IID scenarios by a significant margin. This may be attributed to the fact that, in the IID scenario, although the data distribution among clients is IID, the limited amount of data from individual users introduces performance bias compared to the centralized training because it fails to capture the global data distribution, leading to poor generalization ability and instability of local model. Our proposed method tackles this challenge by introducing global prototypical hash codes, which serve as statistical centroids. These centroids lead to smaller intra-cluster distances and tighter clustering within the same category during local training, ensuring consistency between local optimum and the global optimum and enhancing the model’s performance, thereby addressing the limitations caused by the limited data of each client.

2) Ablation Studies: In the above section, we have verified the effectiveness of our proposed method. However, there are a number of components in (11), including $\mathcal{L}_s$, $\mathcal{L}_p$ and $\mathcal{L}_d$. Therefore, we further conduct ablation experiments for a better understanding of each component of our method. The results are shown in Table IX. Several conclusions can be drawn from the results. First of all, comparing the results of using $\mathcal{L}_s$ only with other results, it can be seen that each component in the framework plays a significant role in improving the model performance. Second, we found that the method can still work well when only $\mathcal{L}_p$ loss is applied. This means that the local clients only train their models with $\mathcal{L}_p$ after receiving the global prototypical hash codes. Finally, the optimal results are achieved by combining all components together in our proposed FedHAP.

3) Effects of the Number of Clients: To analyze the performance of our proposed method when the client number varies, we further test the abovementioned baselines and our method with different numbers of clients from 20 to 100 over NUS-WIDE in the non-IID scenarios, where the data samples are randomly distributed and the length of the hash code is 48 bits. The mAP results are reported in Fig. 8, from which we can see that our method still consistently outperforms all baselines under the different numbers of clients. We also notice that as the
number of clients increases, the performance of the model will decrease slightly. This is not surprising since the amount of data per client will decrease when the number of clients increases, which will result in severer distribution discrepancy of local data and a higher probability of overfitting for local clients.

4) Effects of the Degree of Non-IID: Fig. 9 shows the performance of different algorithms with respect to different degree of non-IID distribution, which is manifested by the number of classes assigned to each client. We observe that both the baseline frameworks and FedHAP are sensitive to the degree of non-IID of the local client data. Meanwhile, FedHAP outperforms all the other federated frameworks at all levels of non-IID. Especially, FedHAP outperforms other federated frameworks by a wider margin as the degree of Non-IID increases (the number of classes

## TABLE III
MAP RESULTS WITH THE DCH ALGORITHM IN THE NON-IID SCENARIOS

| Model    | NUS-WIDE |          |          |          |          |
|----------|----------|----------|----------|----------|----------|
|          | 12bit(%) | 24bit(%) | 48bit(%) | 12bit(%) | 24bit(%) | 48bit(%) | 12bit(%) | 24bit(%) | 48bit(%) |
| FedAvg   | 38.67    | 39.45    | 40.23    | 70.11    | 70.74    | 77.65    | 51.94    | 56.35    | 58.64    |
| FedProx  | 38.42    | 39.25    | 40.98    | 70.81    | 70.81    | 77.82    | 53.18    | 53.28    | 59.25    |
| FedCMR   | 40.69    | 43.47    | 44.09    | 69.65    | 69.92    | 73.86    | 53.42    | 57.61    | 58.94    |
| MOON     | 39.50    | 43.21    | 44.11    | 74.64    | 75.08    | 77.85    | 53.89    | 57.80    | 58.92    |
| FedHAP (ours) | 53.24 | 61.54    | 62.76    | 74.95    | 75.28    | 77.97    | 56.40    | 58.98    | 61.74    |

The bold entities indicate the best result.

## TABLE IV
MAP RESULTS WITH THE DPSH ALGORITHM IN THE NON-IID SCENARIOS

| Model    | NUS-WIDE |          |          |          |          |
|----------|----------|----------|----------|----------|----------|
|          | 12bit(%) | 24bit(%) | 48bit(%) | 12bit(%) | 24bit(%) | 48bit(%) | 12bit(%) | 24bit(%) | 48bit(%) |
| FedAvg   | 42.57    | 43.72    | 44.16    | 74.52    | 75.01    | 78.30    | 54.76    | 57.39    | 59.65    |
| FedProx  | 44.86    | 45.81    | 45.86    | 74.85    | 75.07    | 78.64    | 56.77    | 59.63    | 60.37    |
| FedCMR   | 48.41    | 49.52    | 50.21    | 75.01    | 76.21    | 77.74    | 55.73    | 60.57    | 62.31    |
| MOON     | 42.67    | 47.74    | 49.75    | 74.87    | 76.13    | 78.69    | 54.65    | 57.53    | 62.11    |
| FedHAP (ours) | 56.11 | 59.12    | 62.79    | 75.76    | 76.22    | 78.97    | 57.79    | 60.99    | 62.47    |

The bold entities indicate the best result.

## TABLE V
MAP RESULTS WITH THE GREEDYHASH ALGORITHM IN THE NON-IID SCENARIOS

| Model    | NUS-WIDE |          |          |          |          |
|----------|----------|----------|----------|----------|----------|
|          | 12bit(%) | 24bit(%) | 48bit(%) | 12bit(%) | 24bit(%) | 48bit(%) | 12bit(%) | 24bit(%) | 48bit(%) |
| FedAvg   | 53.68    | 55.82    | 55.94    | 69.93    | 71.21    | 73.60    | 48.57    | 52.16    | 57.25    |
| FedProx  | 49.29    | 54.41    | 59.96    | 67.59    | 69.62    | 73.56    | 46.57    | 56.80    | 60.41    |
| FedCMR   | 51.06    | 63.42    | 63.59    | 65.57    | 69.76    | 70.23    | 48.15    | 55.80    | 55.86    |
| MOON     | 50.92    | 56.79    | 61.28    | 72.15    | 73.46    | 76.23    | 48.74    | 56.51    | 57.98    |
| FedHAP (ours) | 58.32 | 63.60    | 64.24    | 72.63    | 73.56    | 76.82    | 52.91    | 60.13    | 62.66    |

The bold entities indicate the best result.

## TABLE VI
MAP RESULTS WITH THE ORTHOCOS ALGORITHM IN THE NON-IID SCENARIOS

| Model    | NUS-WIDE |          |          |          |          |
|----------|----------|----------|----------|----------|----------|
|          | 12bit(%) | 24bit(%) | 48bit(%) | 12bit(%) | 24bit(%) | 48bit(%) | 12bit(%) | 24bit(%) | 48bit(%) |
| FedAvg   | 40.90    | 50.13    | 53.84    | 64.83    | 65.02    | 73.32    | 46.59    | 49.49    | 52.63    |
| FedProx  | 40.49    | 47.91    | 52.20    | 66.62    | 67.63    | 69.73    | 43.52    | 45.49    | 47.56    |
| FedCMR   | 40.73    | 49.45    | 49.92    | 62.95    | 65.78    | 67.91    | 48.33    | 51.43    | 55.20    |
| MOON     | 39.95    | 50.12    | 50.94    | 64.94    | 65.15    | 67.52    | 46.57    | 49.43    | 52.70    |
| FedHAP (ours) | 55.28 | 61.39    | 65.14    | 72.75    | 76.34    | 77.72    | 56.49    | 58.81    | 61.25    |

The bold entities indicate the best result.
Fig. 6. Precision and recall results on MIRFlickr25K in the non-IID scenarios: (a) Precision-recall curves (PR) @ 48 bits. (b) Recall curves with respect to different numbers of top retrieved samples. (c) Precision curves with respect to different numbers of top retrieved samples.

Fig. 7. Precision and recall results on NUS-WIDE in the non-IID scenarios: (a) Precision-recall curves (PR) @ 48 bits. (b) Recall curves with respect to different numbers of top retrieved samples. (c) Precision curves with respect to different numbers of top retrieved samples.

Fig. 8. mAP results under the different numbers of clients on NUS-WIDE with non-IID data: (a) GreedyHash algorithm @ 48 bits; (b) DPSH algorithm @ 48 bits.

Fig. 9. mAP results under the different levels of non-IID data on MIRFlickr: (a) GreedyHash algorithm @ 48 bits; (b) OrthoCos algorithm @ 48 bits.

| Method     | FedAvg | FedProx | FedCMR | MOON | FedHAP |
|------------|--------|---------|--------|------|--------|
| OrthoCos   | 12 bits| 67.17   | 64.74  | 67.22 | 67.16  | 76.65  |
|            | 24 bits| 71.85   | 73.63  | 71.86 | 71.82  | 78.95  |
|            | 48 bits| 73.35   | 74.31  | 73.35 | 73.32  | 80.94  |
| GreedyHash | 12 bits| 72.56   | 68.72  | 69.38 | 70.74  | 74.31  |
|            | 24 bits| 72.70   | 72.44  | 70.67 | 73.42  | 75.54  |
|            | 48 bits| 75.21   | 74.30  | 74.61 | 76.78  | 77.64  |

The bold entities indicate the best result.

| Method     | FedAvg | FedProx | FedCMR | MOON | FedHAP |
|------------|--------|---------|--------|------|--------|
| OrthoCos   | 12 bits| 50.52   | 48.74  | 50.25 | 50.55  | 56.82  |
|            | 24 bits| 54.73   | 53.14  | 54.66 | 54.66  | 58.91  |
|            | 48 bits| 60.04   | 58.15  | 59.10 | 60.04  | 61.64  |
| GreedyHash | 12 bits| 57.46   | 51.92  | 53.14 | 51.47  | 57.83  |
|            | 24 bits| 58.12   | 56.42  | 58.07 | 57.31  | 58.64  |
|            | 48 bits| 63.25   | 62.41  | 63.19 | 62.34  | 62.85  |

The bold entities indicate the best result.

at each clients reduces). In the most extreme case where every client only holds one class of the data, FedHAP still achieves 73.79% and 72.33% accuracy for the GreedyHash algorithm and OrthoCos algorithm, respectively.
5) Comparison of Training Time: To investigate the training time of the baselines and our proposed method, we conducted additional experiments to calculate the average consuming time required for each user’s local training per round and the average time required to achieve convergence, which are shown in Tables X and XI. The model is considered to have converged when no significant improvement is observed over a certain number of iterations. The experiments investigate the non-IID scenario using the DCH hashing algorithm @ 48 bits on the NUS-WIDE dataset. The experimental results show that our framework exhibits a slightly higher training latency compared to FedAvg. We attribute this increase in training latency to the introduction of local adversarial learning in our method. However, our training latency is comparable to FedProx and significantly lower than FedCMR and MOON. Importantly, it should be noted that the inference process of our method only involves the feedforward network and does not incorporate the adversarial network, thereby avoiding any additional inference latency compared to these baselines.

6) Performance on Cross-Modal Tasks: Cross-modal retrieval [51] can achieve the query between different modalities and the goal of cross-modal hashing is to learn hashing models for different modalities. For example, we can use text queries to retrieve images and use image queries to retrieve texts in the database. To demonstrate the generality of FedHAP on deep hashing retrieval tasks, we perform additional experiments to demonstrate the extensiveness and effectiveness of FedHAP in cross-modal retrieval tasks. Different from unimodal retrieval, we have to learn hashing models and get prototypical hash codes for different modalities (image and text). And the generated prototypical hash codes for each modality will be engaged in its training process of corresponding hashing model and discriminator network to further ensure the distribution consistency of hash codes across modalities. The experiments are studied on NUS-WIDE and MIRFlickr over the representative cross-modal hashing algorithm DADH [52] @ 48 bits in the non-IID scenarios. Furthermore, more comprehensive experimental settings and additional details can be found in Appendix D, available online. According to the results presented in Table XII, it is evident that FedHAP outperforms other methods and achieves the highest performance in the cross-modal retrieval task. It further demonstrates its potential for real-world applications in cross-modal or multi-modal data analysis and retrieval tasks.

7) Performance on Medical Dataset: In this section, we are interested to know whether our proposed method can also be effective in some real-world domains such as the medical domain. Since FL has critical applications in the medical domain due to the strict privacy constraints of patient data, we further apply our method to the medical dataset, VinDr-CXR, which is a large CXR dataset with high-quality labels. The same feature extraction net is applied to VinDr-CXR which is consistent with the other datasets. And the experimental results are shown in Table XIII demonstrate that our framework consistently outperforms other methods for the medical information retrieval task.

| Components | GreedyHash | OrthoCos |
|------------|------------|-----------|
| $\mathcal{L}_s$ | $\mathcal{L}_p$ | $\mathcal{L}_d$ | NUS-WIDE | MIRFlickr | MS-COCO | NUS-WIDE | MIRFlickr | MS-COCO |
| ✓ | ✓ | ✓ | 55.94 | 73.60 | 57.25 | 53.84 | 73.32 | 52.63 |
| ✓ | ✓ | ✓ | 62.36 | 73.78 | 60.21 | 60.35 | 75.38 | 58.91 |
| ✓ | ✓ | ✓ | 62.72 | 73.82 | 60.68 | 60.92 | 75.64 | 60.22 |
| ✓ | ✓ | ✓ | 64.24 | 74.42 | 57.86 | 56.34 | 74.64 | 55.65 |

The bold entities indicate the best result.
V. CONCLUSION

In this paper, we propose a novel federated hashing approach FedHAP for efficient cross-silo retrieval, which enables multiple parties to collaboratively train a hashing model without centralizing their data. We innovatively introduce the adversarial and prototypical-contrastive learning with global prototypical hash codes to maintain the distribution alignment of the locally generated and globally generated hash codes, achieving a significant improvement in model effectiveness over a variety of hashing models and Non-IID distributions. Comprehensive experimental results on multiple datasets and various retrieval tasks also demonstrated the superiority and generality of our proposed FedHAP framework.

ACKNOWLEDGMENT

The authors would also like to thank anonymous reviewers for their insightful comments.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
[3] C. Zhang, L. Zhu, S. Zhang, and W. Yu, “TDHPPR: An efficient deep hashing based privacy-preserving image retrieval method,” Neurocomputing, vol. 406, pp. 386–398, 2020.
[4] H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” in Proc. 29th Conf. Neural Inform. Process. Syst., Barcelona, Spain, 2016, pp. 5–10.
[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Proc. 20th Int. Conf. Artif. Intell. Statist., 2017, pp. 1273–1282.
[6] J. Xu, X. Xu, P. Walker, and F. Wang, “Federated patient hashing,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 6486–6493.
[7] L. Zong, Q. Xie, J. Zhou, P. Wu, X. Zhang, and B. Xu, “FedCMR: Federated cross-modal retrieval,” in Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2021, pp. 1672–1676.
[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, pp. 1–19, 2019.
[9] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. 11, pp. 2579–2605, 2008.
[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” in Proc. Mach. Learn. Syst., vol. 2, pp. 429–450, 2020.
[11] W. J. Li, S. Wang, and W. C. Kang, “Feature learning based deep supervised hashing with pairwise labels,” in Proc. 25th Int. Joint Conf. Artif. Intell., 2016, pp. 1711–1717.
[12] L. Chi and X. Zhu, “Hashing techniques: A survey and taxonomy,” ACM Comput. Surv., vol. 50, no. 1, pp. 1–36, 2017.
[13] V. Erin Ljong, J. Lu, G. Wang, P. Moulin, and J. Zhou, “Deep hashing for compact binary codes learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 2475–2483.
[14] H. Lai, Y. Pan, Y. Liu, and S. Yan, “Simultaneous feature learning and hash coding with deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3270–3278.
[15] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, “Supervised hashing for image retrieval via image representation learning,” in Proc. 28th AAAI Conf. Artif. Intell., 2014, pp. 2156–2162.
[16] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, “Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification,” IEEE Trans. Image Process., vol. 24, no. 12, pp. 4766–4779, Dec. 2015.
[17] F. Zhao, Y. Huang, L. Wang, and T. Tan, “Deep semantic ranking based hashing for multi-label image retrieval,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1556–1564.
[18] Y. Weiss et al., “Spectral hashing,” in Proc. 21st Int. Conf. Neural Inf. Process. Syst., 2008, pp. 1753–1760.
[19] Y. Gong, S. Lazebnik, A. Torralba, and A. P. Pentland, “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 12, pp. 2916–2929, Dec. 2013.
[20] W. Kong and W.-J. Li, “Isotropic hashing,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1753–1760.
[21] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, “Discrete graph hashing,” in Proc. 27th Int. Conf. Neural Inf. Process. Syst., 2014, pp. 3419–3427.
[22] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, “Supervised hashing with kernels,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2012, pp. 2074–2081.
[23] M. Rastegari, A. Farhadi, and D. Forsyth, “Attribute discovery via predictable discriminative binary codes,” in Proc. Eur. Conf. Comput. Vis., Springer, 2012, pp. 876–889.
[24] Y. Cao, M. Long, B. Liu, and J. Wang, “Deep cauchy hashing for hamming space retrieval,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1229–1237.
[25] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Proc. Int. Conf. Artif. Intell. Statist., PMLR, 2017, pp. 1273–1282.
[26] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, “Ensemble distillation for robust model fusion in federated learning,” in Proc. Adv. Neural Inf. Process. Syst., 2020, pp. 2351–2363.
[27] Q. Li, B. He, and D. Song, “Model-contrastive federated learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 10713–10722.
[28] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” Proc. Adv. Neural Inf. Process. Syst., vol. 30, pp. 7611–7620, 2020.
[29] T.-M. H. Hsu, H. Qi, and M. Brown, “Measuring the effects of non-identical data distribution for federated visual classification,” 2019, arXiv:1909.06355.
[30] X. Mu et al., “FedProc: Prototypical contrastive federated learning on non-IID data,” Future Gener. Comput. Syst., vol. 143, pp. 93–104, 2021.
[31] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, “Federated learning with matched averaging,” in Proc. 5th Int. Conf. Learn. Representations, Addis Ababa, Ethiopia, 2020.
[32] R. G. Lopez, S. Fernandez, and T. Stander, “Data-free knowledge distillation for deep neural networks,” in Proc. 31st Conf. Neural Inf. Process. Syst., Long Beach, CA, USA, 2017.
[33] Z. Dai, C. Dun, Y. Tang, A. Kyrillidis, and A. Shrivastava, “Federated multiple label hashing (FedMLH): Communication efficient federated learning on extreme classification tasks,” 2021, arXiv:2110.12292.
[34] H. Hu, G. Dobbie, Z. Salcic, M. Liu, J. Zhang, and X. Zhang, “A locality sensitive hashing based approach for federated recommender system,” in Proc. IEEE/ACM 20th Int. Symp. Cluster Cloud Internet Comput., 2020, pp. 836–842.
[35] X. Luo et al., “A survey on deep hashing methods,” ACM Trans. Knowl. Discov. Data, vol. 17, 2020, Art. 15.
[36] H. Zhu, M. Long, J. Wang, and Y. Cao, “Deep hashing network for efficient similarity retrieval,” in Proc. AAAI Conf. Artif. Intell., 2016, pp. 2415–2421.
[37] I. Goodfellow et al., “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.
[38] J. Wu et al., “Hierarchical personalized federated learning for user modeling,” in Proc. Web Conf., 2021, pp. 957–968.
[39] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, “NUS-WIDE: A real-world web image database from National University of Singapore,” in Proc. ACM Int. Conf. Image Video Retrieval, 2009, pp. 1–9.
[40] M. J. Huiskes and M. S. Lew, “The MIR flickr retrieval evaluation,” in Proc. 1st ACM Int. Conf. Multimedia Inf. Retrieval, 2008, pp. 39–43.
[41] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., Springer, 2014, pp. 740–755.
Meilin Yang (Student Member, IEEE) is currently working toward the master’s degree with the Tsinghua Shenzhen International Graduate School, Tsinghua University. Her research interests include federated learning, deep hash learning and image retrieval.

Jian Xu (Student Member, IEEE) received the BS and MS degrees from Zhejiang University, in 2017 and 2020, respectively. He is currently working toward the PhD degree with Tsinghua University and also a visiting scholar with the University of California, Berkeley. His research interests include federated learning, distributed optimization and their applications.

Yang Liu (Senior Member, IEEE) is an Associate Professor with Institute for AI Industry Research, Tsinghua University. Before joining Tsinghua, she was the principal researcher and research team lead with WeBank. Her research interests include federated learning, machine learning, multi-agent systems, statistical mechanics and AI industrial applications. Her research work was recognized with multiple awards, such as AAAI Innovation Award and CCF Technology Award.

Wenbo Ding (Member, IEEE) received the BS and PhD degrees (Hons.) from Tsinghua University, in 2011 and 2016, respectively. He worked as a post-doctoral research fellow with Georgia Tech under the supervision of Professor Z. L. Wang from 2016 to 2019. He is now an Associate Professor and PhD supervisor with Tsinghua-Berkeley Shenzhen Institute, Tsinghua Shenzhen International Graduate School, Tsinghua University, where he leads the Smart Sensing and Robotics (SSR) Group. He is also with RISC-V International Open Source Laboratory, Shenzhen, China, 518055. His research interests are diverse and interdisciplinary, which include self-powered sensors, human-machine interfaces, wearable devices for health and robotics with the help of signal processing, machine learning, and mobile computing. He has received many prestigious awards, including the Gold Medal of the 47th International Exhibition of Inventions Geneva and the IEEE Scott Helt Memorial Award.