# PROTOTYPE-BASED LAYERED FEDERATED CROSS-MODAL HASHING

Jiale Liu<sup>1</sup>, Yu-Wei Zhan<sup>1</sup>, Xin Luo<sup>1</sup>, Zhen-Duo Chen<sup>1</sup>, Yongxin Wang<sup>2</sup>, Xin-Shun Xu<sup>1</sup>

<sup>1</sup>School of Software, Shandong University
 <sup>2</sup> School of Computer Science, Shandong Jianzhu University

# ABSTRACT

Recently, deep cross-modal hashing has gained increasing attention. However, in many practical cases, data are distributed and cannot be collected due to privacy concerns, which greatly reduces the cross-modal hashing performance on each client. And due to the problems of statistical heterogeneity, model heterogeneity, and forcing each client to accept the same parameters, applying federated learning to cross-modal hash learning becomes very tricky. In this paper, we propose a novel method called prototype-based layered federated cross-modal hashing. Specifically, the prototype is introduced to learn the similarity between instances and classes on server, reducing the impact of statistical heterogeneity (non-IID) on different clients. And we monitor the distance between local and global prototypes to further improve the performance. To realize personalized federated learning, a hypernetwork is deployed on server to dynamically update different layers' weights of local model. Experimental results on benchmark datasets show that our method outperforms state-of-the-art methods.

*Index Terms*— Federated Learning, Learning to Hash, Cross-Modal Retrieval, Prototype Learning

# 1. INTRODUCTION

With a large number of texts, images, videos, and other media data being generated, it is particularly important to conduct similarity search for multimedia data reasonably and effectively [1, 2, 3]. In real applications, users often need to use data from one modality (e.g., text modality) to retrieve relevant data from another modality (e.g., image modality). Benefiting from high retrieval speed and low storage cost, cross-modal hashing attracts increasing attention. Crossmodal hashing (CMH) maps high-dimensional raw data to short binary hash codes by learning hash functions, while maintaining the similarity of the original samples in Hamming space. Although cross-modal hashing [4, 5, 6] has achieved satisfactory performance so far, they are encountering some practical problems due to the growing concern about privacy protection. In many real-world situations, multimedia data is scattered across multiple silos and those distributed data may not be directly shared or collected due to privacy concerns and regulations. This makes each client use only its own small amount of data for independent local training, which significantly degrades cross-modal hashing performance compared to traditional centralized training. To address the above issues, researchers try to combine federated learning [7, 8, 9, 10] with cross-modal hashing.

At present, federated cross-modal hashing is intractable due to the following challenges. 1) Statistical heterogeneity. The local data distribution of each client varies with its location and preferences, resulting in the data of each client being independent but not obeying the same distribution (Non-IID) [11, 12]. 2) Model heterogeneity. Traditional federated learning requires consistency of models across all clients, which is unrealistic in practical and complex applications [13, 14]. Different clients desire different models because their own application scenarios may differ. However, different models may cause huge difficulty for the communication of parameters between clients and central server. 3) Personalized federated learning. Most of prior efforts let central server acquire and process parameters from all clients first and then return the same parameters [15, 16] to all clients. The policy that each client is forced to accept the same parameters prevents each client from better adapting to its own local data, resulting in sub-optimal performance.

To address above-mentioned challenges, we propose a new federated cross-modal hashing method called Prototypebased Layered Federated Cross-Modal Hashing (PLFedCMH for short). Specifically, on the basis of class-wise hash codes, PLFedCMH introduces class prototypes generated by modality networks to assist the learning of supervised hash functions, reducing the impact of statistical heterogeneity (non-IID) on different clients. Besides, the server only needs to aggregate local class prototypes and does not need to aggregate model parameters. There is no need to consider the parameter aggregation problem caused by model heterogeneity. Last but not least, PLFedCMH is designed with personalized federated strategy. Through the hypernetwork introduced by the server, the weights of different layers on the client are dynamically updated, which can realize personalized parameter customization for different clients. To summarize, the main contributions of this paper are as follows. 1) To consider privacy concerns, a new federated method PLFedCMH is elaborately designed for training cross-modal hashing with

distributed data. 2) The proposed method could simultaneously take statistical heterogeneity, model heterogeneity, and personalized federated learning into consideration. 3) Experimental results on benchmark datasets show that our method can achieve significantly improved accurate in IID, nonIID-equal, and nonIID-unequal scenarios.

### 2. PROPOSED METHOD

#### 2.1. Problem Definition and Notations

Following existing cross-modal hashing literature and without loss of generality, we formulate our model in the context of image-text retrieval task. In this paper, we propose a new federated learning method for cross-modal hashing, which could support hashing model training based on different silos' data without privacy and security concerns.

Assuming there are N clients, *i*-th client possesses its own dataset  $D_i = \{(\mathbf{x}_j^{(i)}, \mathbf{t}_j^{(i)}, \mathbf{l}_j^{(i)})\}_{j=1}^{m_i} (1 \le i \le N)$ , where  $m_i$  denotes the number of samples,  $\mathbf{x}_j^{(i)}$  ( $\mathbf{t}_j^{(i)}$ ) represents image modality (text modality) of the *j*-th sample on client *i*,  $\mathbf{l}_j^{(i)}$  is the label vector. The size of all clients' datasets can be obtained by  $M = \sum_{i=1}^{N} m_i$ .

As our method introduces and leverages prototypes of classes, we let **P** be the class prototype and then have both local prototypes  $\mathbf{P}_{local}$ : { $\mathbf{P}_{local_1}$ ,  $\mathbf{P}_{local_2}$ ,  $\cdots$ ,  $\mathbf{P}_{local_N}$ } and global prototype  $\mathbf{P}_{global}$ . Considering both image and text modalities, we have  $\mathbf{P}_{local_{x_i}}$  and  $\mathbf{P}_{local_{t_i}}$  on client *i*, while  $\mathbf{P}_{global_x}$  and  $\mathbf{P}_{global_t}$  on server. Specifically, local prototypes are output values of modality networks' last layer without using activation function. Global prototype is the average of all local prototypes, which is computed on the server.

### 2.2. Similarity Preserving based on Local Data

As the core of learning to hash is to preserve the similarity, hash codes of those instances, which belong to the same class, should be relatively similar. In other words, hash codes of instances have direct relation with their labels  $\mathbf{L} \in \{0, 1\}^{M \times c}$ , where *c* is the number of classes for all data samples *M*.

To construct and preserve such relation, we first introduce hash codes of classes which is denoted as  $\mathbf{Y} \in \{-1, 1\}^{r \times c}$ where r is the hash code length. Then, we define the following optimization problem:  $\min_{\{\mathbf{B}^{(i)}, \mathbf{Y}^{(i)}\}} ||r\mathbf{L}^{(i)} - \mathbf{B}^{(i)^T}\mathbf{Y}^{(i)}||_F^2$ , where superscript i denotes i-th client,  $\mathbf{L}^{(i)}$ is its label matrix,  $\mathbf{B}^{(i)} \in \{-1, 1\}^{r \times m_i}$  is the hash codes, and  $\mathbf{Y}^{(i)} \in \{-1, 1\}^{r \times c}$  is the hash codes of classes on the i-th client. The above equation could force the hash codes of samples which share same labels to be more similar and thus achieve the similarity-preserving goal.

## 2.3. Global Information Embedding

Sec.2.2 works only with clients' own local data. Under the influence of statistical heterogeneity (non-IID), a single client cannot well consider the overall class characteristics of the

entire dataset *D*. If no remedy is taken, the local client may get stuck in its seen classes and cannot handle samples of unseen classes which are distributed on other clients.

To overcome such limitation caused by non-IID, we try to embed global information into local training. Thus, we define the following optimization function which let the local hash codes interact with global class prototypes:  $min_{\mathbf{B}^{(i)}} ||r\mathbf{L}_i^{(i)} - \mathbf{B}^{(i)T}\mathbf{P}_{global}||_F^2$ , where  $\mathbf{B}^{(i)} \in \{-1, 1\}^{r \times m_i}$  is the hash code matrix of samples on the *i*-th client,  $\mathbf{P}_{global}$  is class prototypes aggregated on the server in last federated round.

In addition, we also try to keep the consistency of class prototypes between one local client and the global server. This could reduce the influence of class distribution differences on local training and improve the accuracy of local cross-modal hashing retrieval. The loss function is as follows,

$$O_1 = MSE(\mathbf{P}_{local_i}, \mathbf{P}_{global}),\tag{1}$$

where  $MSE(\cdot)$  is the mean square error and  $\mathbf{P}_{local_i}$  is the local class prototypes generated in current federated round.

#### 2.4. Hash Learning for Local Clients

As most recent cross-modal hashing methods are deep ones, we could freely assume that there exist image and text modality networks. On the *i*-th client, let  $\mathbf{F}^{(i)} = f(\mathbf{x}^{(i)}; \boldsymbol{\theta}_{x_i})$  denote the extracted image features, where  $\boldsymbol{\theta}_{x_i}$  represents the parameter of image modality network. For text modality, let  $\mathbf{G}^{(i)} = g(\mathbf{t}^{(i)}; \boldsymbol{\theta}_{t_i})$  and  $\boldsymbol{\theta}_{t_i}$  represent the output and the parameter of text modality network of client *i*.

Then, as deep hashing could synchronously conduct hash code learning and feature extraction, based on Sec.2.2 and Sec.2.3, we could give the following equation,

$$O_{2} = \alpha (\|r\mathbf{L}^{(i)} - \mathbf{F}^{(i)}^{T}\mathbf{Y}^{(i)}\|_{F}^{2} + \|r\mathbf{L}^{(i)} - \mathbf{G}^{(i)}^{T}\mathbf{Y}^{(i)}\|_{F}^{2}) + \beta (\|r\mathbf{L}^{(i)} - \mathbf{F}^{(i)}^{T}\mathbf{P}_{global_{x}}\|_{F}^{2} + \|r\mathbf{L}^{(i)} - \mathbf{G}^{(i)}^{T}\mathbf{P}_{global_{t}}\|_{F}^{2}) + \mu (\|\mathbf{B}^{(i)} - \mathbf{F}^{(i)}\|_{F}^{2} + \|\mathbf{B}^{(i)} - \mathbf{G}^{(i)}\|_{F}^{2}), s.t. \ \mathbf{B}^{(i)} \in \{-1, 1\}^{r \times m_{i}}, \mathbf{Y}^{(i)} \in \{-1, 1\}^{r \times c},$$

$$(2)$$

where  $\alpha$ ,  $\beta$ , and  $\mu$  are the trade-off parameters.

### 2.5. Overall Loss and Optimization for Local Clients

To make the to-be-learnt hash codes preserve both intra-client similarity and inter-client similarity, we combine Eq.(1) and Eq.(2). Besides, as PLFedCMH is a federated method which tries to help existing CMH accommodate to distributed scenario, we should include the original loss of deep CMH  $O_{hash}$ . Thus, the overall objective function is,

$$min \quad O_1 + \eta O_2 + \xi O_{hash},\tag{3}$$

where  $\eta$  and  $\xi$  are trade-off parameters.

The optimization of  $O_1 + \eta O_2$  could easily follow the strategy of most existing deep CMH methods, that is iteratively optimizing one variable with the others fixed. When updating network parameters  $\boldsymbol{\theta}_{x_i}$  and  $\boldsymbol{\theta}_{t_i}$ , the back-propagation algorithm could be leveraged.  $\mathbf{B}^{(i)}$  could be computed by  $\mathbf{B}^{(i)} = sign(\mathbf{F}^{(i)} + \mathbf{G}^{(i)})$ . We could use the optimization of [5] to discretely generate  $\mathbf{Y}^{(i)}$  bit by bit.

# Algorithm 1 PLFedCMH Algorithm

**Input:**  $\{D_1, \dots, D_N\}$ . Total communication rounds R. Hypernetwork learning rate  $\gamma$ .

**Output:** Trained personalized models  $\{\bar{\theta}_{x_1}, \cdots, \bar{\theta}_{x_N}\}$  and  $\{\bar{\theta}_{t_1}, \cdots, \bar{\theta}_{t_N}\}$ .

Server executes:

1: Initialization

2: for each federated round  $r \in \{1, \cdots, R\}$  do

3: **for** each client i **in parallel do**  $\bar{\boldsymbol{\theta}}_{i}^{(r+1)} = \{\boldsymbol{\theta}_{i}^{1}, \boldsymbol{\theta}_{i}^{2}, \cdots, \boldsymbol{\theta}_{i}^{K}\} * HN_{i}(\mathbf{s}_{i}, \boldsymbol{\zeta}_{i})$ 

4: 
$$\boldsymbol{\sigma}_i = \{\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_i, \cdots, \boldsymbol{\sigma}_i\} * HN_i(\mathbf{S}_i)$$

5: 
$$\Delta \boldsymbol{\theta}_{x_i}, \Delta \boldsymbol{\theta}_{t_i} \leftarrow LocalUpdate(\boldsymbol{\theta}_i^{(r+1)})$$

6: 
$$\{\boldsymbol{\theta}_{x_i}^1, \boldsymbol{\theta}_{x_i}^2, \cdots, \boldsymbol{\theta}_{x_i}^K\}^{(r+1)} = \boldsymbol{\theta}_{x_i}^{(r)} + \Delta \boldsymbol{\theta}_x$$

7: 
$$\{\boldsymbol{\theta}_{t_i}^1, \boldsymbol{\theta}_{t_i}^2, \cdots, \boldsymbol{\theta}_{t_i}^K\}^{(r+1)} = \boldsymbol{\theta}_{t_i}^{(r)} + \Delta \boldsymbol{\theta}_{t_i}$$

8: 
$$\mathbf{s}_{i}^{(r+1)} = \mathbf{s}_{i}^{(r)} - \gamma \nabla_{(r)} \text{Eq.}(3)$$

9: 
$$\zeta_{i}^{(r+1)} = \zeta_{i}^{(r)} - \gamma \nabla_{\zeta_{i}^{(r)}} \text{Eq.(3)}$$

10: end for 
$$\zeta_i^{(i)}$$

11: Update global prototypes

12: end for

LocalUpdate  $(\bar{\boldsymbol{\theta}}_i^{(r+1)})$ :

13: Receive 
$$(\bar{\boldsymbol{\theta}}_i^{(r+1)})$$
 from server.

14: Set 
$$\boldsymbol{\theta}_{r_{t}} = (\bar{\boldsymbol{\theta}}_{r_{t}}^{(r+1)}), \boldsymbol{\theta}_{t_{t}} = (\bar{\boldsymbol{\theta}}_{t_{t}}^{(r+1)})$$

16: **for** batch
$$(\mathbf{x}^{(i)}, \mathbf{t}^{(i)}, \mathbf{l}^{(i)}) \in D_i$$
 **do**

17: Update local prototypes. Update  $\theta_{x_i}, \theta_{t_i}, \mathbf{Y}$ , and B 18: end for

19: end for

20: return  $\Delta \boldsymbol{\theta}_{x_i} = \boldsymbol{\theta}_{x_i} - (\bar{\boldsymbol{\theta}}_{x_i}^{(r+1)}), \Delta \boldsymbol{\theta}_{t_i} = \boldsymbol{\theta}_{t_i} - (\bar{\boldsymbol{\theta}}_{t_i}^{(r+1)})$ 

### 2.6. Generating layered weights through the server

For each client's local image model and text model, we set up the corresponding hypernetworks on the server side, which are composed of some fully connected layers.

Taking the image modality of *i*-th client as an example, we have the hypernetwork  $HN_{x_i}(\mathbf{s}_{x_i}, \boldsymbol{\zeta}_{x_i})$  [17, 18]. The input of the hypernetwork is the embedding vector  $\mathbf{s}_{x_i}$ , and  $\boldsymbol{\zeta}_{x_i}$ is the parameter of the hypernetwork. The model parameters of the image modality before updating on the server are  $\boldsymbol{\theta}_{x_i} = \{\boldsymbol{\theta}_{x_i}^1, \cdots, \boldsymbol{\theta}_{x_i}^k, \cdots, \boldsymbol{\theta}_{x_i}^K\}$ , where  $\boldsymbol{\theta}_{x_i}^k$  are the parameters of *k*-th layer  $(1 \le k \le K)$ . When the parameters of the client image modality are uploaded to the server, the server updates the layered parameters of the client through the image modality hypernetwork  $HN_{x_i}(\mathbf{s}_{x_i}, \boldsymbol{\zeta}_{x_i})$ :  $\bar{\boldsymbol{\theta}}_{x_i} = \{\bar{\boldsymbol{\theta}}_{x_i}^1, \cdots, \bar{\boldsymbol{\theta}}_{x_i}^K\}$   $\{\boldsymbol{\theta}_{x_i}^1, \cdots, \boldsymbol{\theta}_{x_i}^K\} * HN_{x_i}(\mathbf{s}_{x_i}, \boldsymbol{\zeta}_{x_i})$ . So the parameters  $\boldsymbol{\theta}_{x_i}$  of the *i*-th client are updated to  $\overline{\boldsymbol{\theta}}_{x_i}$  before the next round of local training. According to the chain rule, we can have the gradient of  $\mathbf{s}_i$  and  $\boldsymbol{\zeta}_i$  from Eq. (3).

## 2.7. Overall Algorithm and Framework of PLFedCMH

Algorithm 1 shows a federated round of the proposed PLFed-CMH. 1) On the server side, two hypernetworks corresponding to different modality networks are used to generate the layered weights of all clients. 2) The server transmits the updated client layered weights and the aggregated abstract global class prototypes to clients. 3) The client updates the personalized model parameter values for modality networks after receiving the layered weights. 4) On the local client, features of the private samples are extracted through the image and text modality networks to obtain the rich semantic information of samples and the class prototypes of the local client. 5) After local training, the local model parameter updates and the local class prototypes for both modalities are uploaded to the server. 6) The server aggregates the local prototypes to obtain global prototypes. And the hypernetworks calculate the layered weights through the gradient change of the model.

### **3. EXPERIMENT**

## **3.1.** Experiment settings

**Datasets.** Following existing literature [20, 21], two benchmark datasets are chosen for evaluation, i.e., FashionVC [22] and Ssense [20]. FashionVC is from online fashion community Polyvore. After removing categories with less than 25 samples, FashionVC contains 19,862 image-text pairs with hierarchical labels. Ssense is also from the fashion field, which contains 15,696 hierarchically labeled image-text pairs after removing categories with less than 70 samples. In this paper, only the most fine-grained part of the hierarchical labels is employed for evaluation.

Following setting in [19], three ways to partition the data set are used, i.e., nonIID-equal, nonIID-unequal, and IID. NonIID-equal and nonIID-unequal are cases of statistical heterogeneity. In the nonIID-equal case, each client has a different data distribution, and the classes may overlap or not overlap at all between different clients. However, the number of categories is the same for each client, and the number of samples in each category is also the same. Given the different number of samples in each class in the dataset, we needed to accommodate the smaller classes in order to achieve nonIIDequal, so the total number of samples from all clients we used is 18% to 20% of the entire dataset. In the nonIID-unequal case, the dataset is 100% used by the clients, and the number of samples per class is completely different. In the IID case, data is shuffled and then evenly divided among the clients, which have the same data distribution.

	nonIID-equal					nonIID-unequal					IID							
FashionVC	Image-to-Text			Text-to-Image		Image-to-Text		Text-to-Image			Image-to-Text		Text-to-Image					
	16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit									
centralized FedAvg [11] FedCMR [13] FedProx [14] FedProto [19] PLFedCMH	0.766 0.398 0.180 0.261 0.297 <b>0.614</b>	0.762 0.583 0.404 0.584 0.644 <b>0.652</b>	0.756 0.637 0.372 0.643 0.656 <b>0.657</b>	0.937 0.378 0.159 0.228 0.284 <b>0.631</b>	0.949 0.583 0.302 0.586 0.659 <b>0.671</b>	0.946 0.632 0.268 0.633 0.677 <b>0.702</b>	0.766 0.584 0.577 0.548 0.603 <b>0.710</b>	0.762 0.712 0.620 0.710 0.744 <b>0.766</b>	0.756 0.741 0.661 0.747 0.760 <b>0.763</b>	0.937 0.753 0.576 0.706 0.742 <b>0.876</b>	0.949 0.912 0.613 0.900 0.936 <b>0.941</b>	0.946 0.932 0.707 0.930 <b>0.947</b> <b>0.947</b>	0.766 0.544 0.255 0.603 0.678 <b>0.731</b>	0.762 0.724 0.679 0.720 0.743 <b>0.761</b>	0.756 0.743 0.677 0.741 0.761 <b>0.769</b>	0.937 0.666 0.229 0.770 0.858 <b>0.900</b>	0.949 0.903 0.767 0.901 0.934 <b>0.947</b>	0.946 0.926 0.712 0.915 0.942 <b>0.951</b>
	nonIID-equal					nonIID-unequal					IID							
0			nonIII	D-equal					nonIID	unequal					Π	D		
Ssense	In	nage-to-Te	nonIII ext	D-equal	ext-to-Ima	ge	In	nage-to-Te	nonIID- ext	unequal	ext-to-Ima	ge	In	nage-to-Te	II ext	D Te	ext-to-Ima	ge
Ssense	In 16bit	nage-to-Te 32bit	nonIII ext 64bit	D-equal Te 16bit	ext-to-Ima 32bit	ge 64bit	In 16bit	nage-to-Te 32bit	nonIID- ext 64bit	unequal Te 16bit	ext-to-Ima 32bit	ge 64bit	In 16bit	nage-to-Te 32bit	Il ext 64bit	D Te 16bit	ext-to-Ima 32bit	ge 64bit

Table 1. The MAP results of various methods on FashionVC and Ssense with different splits over clients.

 Table 2. The MAP results on Ssense.

Method	In	nage-to-Te	ext	Text-to-Image				
	16bit	32bit	64bit	16bit	32bit	64bit		
PLFedCMH PFedCMH LFedCMH	<b>0.947</b> 0.938 0.938	<b>0.956</b> 0.952 0.940	<b>0.958</b> 0.957 0.951	<b>0.977</b> 0.963 0.965	<b>0.981</b> 0.978 0.966	<b>0.982</b> 0.980 0.976		

**Experiment details.** As this paper focuses on federated learning for CMH, we utilized existing SOTA SHDCH [21] for ours and all federated baselines. The hypernetworks for both modalities contain four fully-connected layers with ReLU activation function. The hyper-parameters are set as follows:  $\alpha = 0.5$ ,  $\beta = 0.5$ ,  $\mu = 10$ ,  $\eta = 10^{-5}$ , and  $\xi = 1$ . The learning rate of modality networks is 0.0001 and the learning rate of hypernetworks is 0.001.

**Evaluation metrics.** Two cross-modal retrieval tasks are conducted. "Image-to-Text" task utilizes an image as a query to retrieve right texts, and "Text-to-Image" task is to retrieve desired images with a text query. We adopted the widely used Mean Average Precision (MAP) to evaluate the performance, where higher values indicate better performance.

## 3.2. Comparison with Baselines

The results of MAP values on FashionVC and Ssense datasets under nonIID and IID settings are presented in Table 1. We compared the MAP values of PLFedCMH with several SOTA baselines, including FedAvg [11], FedCMR [13], FedProx [14] and FedProto [19]. Furthermore, results of "centralized" are also provided, which denotes the result of accumulating all data learned on a single server and is the upper bound of the federated learning algorithm.

As found from Table 1, MAP values of all baselines with 16-bit hash codes under nonIID-equal setting can only reach less than 40% of centralized's performance due to poor characterization capability of short-bit hash codes, while our method reaches a MAP value of 61.4%. One possible rea-

son is that our method introduces class prototypes to assist in the learning of hash codes, which reduces the effect of statistical heterogeneity on different clients. Besides, our method achieves the best results in most cases, which implies the effectiveness of using prototypes to learn the similarity between instances and classes on the server.

### 3.3. Ablation experiments

To fully validate the performance of PLFedCMH, two variants are designed. The first variant removes layered update weights on the server to different clients, which is named PFedCMH. The other variant is termed LFedCMH, which excludes the prototypes. Comparison results with nonIIDunequal split are listed in Table 2. From those tables, we can find our PLFedCMH could perform better than two designed variants. Such phenomena reveal that both updating layered parameters and using prototypes to learn the similarity between instances and classes on the server are effective.

## 4. CONCLUSION

In this paper, we propose a novel federated learning method PLFedCMH for cross-modal hashing with distributed data. We introduce class prototypes generated by modal networks to assist the hash learning, reducing the impact of statistical heterogeneity (non-IID) on different clients. At the same time, distance between local and global prototypes is considered to improve the performance. The server dynamically updates the weights of different layers of the client, which can realize personalized parameter customization for different clients. On the other hand, the server only needs to aggregate local category prototypes without aggregating model parameters, reducing the impact of model heterogeneity. Experimental results show that the proposed method achieves the best performance on benchmark datasets.

### 5. REFERENCES

- [1] Yongxin Wang, Xin Luo, Liqiang Nie, Jingkuan Song, Wei Zhang, and Xin-Shun Xu, "Batch: A scalable asymmetric discrete cross-modal hashing," *TKDE*, vol. 33, no. 11, pp. 3507–3519, 2020.
- [2] Abin Jose, Daniel Filbert, Christian Rohlfing, and Jens-Rainer Ohm, "Deep hashing with hash center update for efficient image retrieval," in *ICASSP*. IEEE, 2022, pp. 4773–4777.
- [3] Xiaoqing Liu, Huanqiang Zeng, Yifan Shi, Jianqing Zhu, and Kai-Kuang Ma, "Deep rank cross-modal hashing with semantic consistent for image-text retrieval," in *ICASSP.* IEEE, 2022, pp. 4828–4832.
- [4] Xin Luo, Peng-Fei Zhang, Zi Huang, Liqiang Nie, and Xin-Shun Xu, "Discrete hashing with multiple supervision," *TIP*, vol. 28, no. 6, pp. 2962–2975, 2019.
- [5] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen, "Supervised discrete hashing," in *CVPR*, 2015, pp. 37–45.
- [6] Qing-Yuan Jiang and Wu-Jun Li, "Deep cross-modal hashing," in *CVPR*, 2017, pp. 3232–3240.
- [7] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.
- [8] Felix Yu, Ankit Singh Rawat, Aditya Menon, and Sanjiv Kumar, "Federated learning with only positive labels," in *ICML*. PMLR, 2020, pp. 10946–10956.
- [9] Yong Liu, Xinghua Zhu, Jianzong Wang, and Jing Xiao, "A quantitative metric for privacy leakage in federated learning," in *ICASSP*. IEEE, 2021, pp. 3065–3069.
- [10] Divyansh Jhunjhunwala, Advait Gadhikar, Gauri Joshi, and Yonina C Eldar, "Adaptive quantization of model updates for communication-efficient federated learning," in *ICASSP*. IEEE, 2021, pp. 3110–3114.
- [11] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*. PMLR, 2017, pp. 1273–1282.
- [12] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *TNNLS*, vol. 32, no. 8, pp. 3710–3722, 2020.

- [13] Linlin Zong, Qiujie Xie, Jiahui Zhou, Peiran Wu, Xianchao Zhang, and Bo Xu, "Fedcmr: Federated crossmodal retrieval," in *SIGIR*, 2021, pp. 1672–1676.
- [14] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith, "Federated optimization in heterogeneous networks," *MLSys*, vol. 2, pp. 429–450, 2020.
- [15] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang, "Personalized cross-silo federated learning on non-iid data.," in *AAAI*, 2021, pp. 7865–7873.
- [16] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang,"Towards personalized federated learning," *TNNLS*, 2022.
- [17] David Ha, Andrew Dai, and Quoc V Le, "Hypernetworks," arXiv preprint arXiv:1609.09106, 2016.
- [18] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu, "Layer-wised model aggregation for personalized federated learning," in *CVPR*, 2022, pp. 10092–10101.
- [19] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in AAAI, 2022, vol. 1, p. 3.
- [20] Changchang Sun, Xuemeng Song, Fuli Feng, Wayne Xin Zhao, Hao Zhang, and Liqiang Nie, "Supervised hierarchical cross-modal hashing," in *SIGIR*, 2019, pp. 725–734.
- [21] Yu-Wei Zhan, Xin Luo, Yongxin Wang, and Xin-Shun Xu, "Supervised hierarchical deep hashing for crossmodal retrieval," in ACM MM, 2020, pp. 3386–3394.
- [22] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie, "Neural compatibility modeling with attentive knowledge distillation," in *SIGIR*, 2018, pp. 5–14.