Generating Synthetic Data for Unsupervised Federated Learning of Cross-Modal Retrieval

Tianlong Zhang[1], Zhe Xue[1*], Adnan Mahmood[2], Junping Du[1], Yuchen Dong[1], Shilong Ou[1], Lang Feng[1], Ming-Hsuan Yang[3], Yuankai Qi[2*]

[1]Beijing University of Posts and Telecommunications
[2]Macquarie University
[3]University of California at Merced
{tlzhang, xuezhe, junpingd, dongyuchen, osl, xiangba0n1ng}@bupt.edu.cn, {adnan.mahmood, yuankai.qi}@mq.edu.au, mhyang@ucmerced.edu

Abstract

Unsupervised federated learning for cross-modal retrieval has received increasing attention in recent years as it can free the requirement for annotations and avoid uploading original clients' data to servers. Most existing methods focus on how to learn better local models and their aggregation to overcome data distribution drift across clients. Unlike prior works, we propose to address the data distribution problem by generating synthetic data, which can benefit existing federated learning methods. Specifically, we train a WGAN generator with three newly designed loss constraints on each client to improve the quality of the generated data. We first compute cluster prototypes to address the problem of lack of labels. Then, a direct contrastive loss between generated image and text features, an indirect contrastive loss with reference to cluster prototypes, and a Jensen-Shannon Divergence (JSD) loss also with reference to cluster prototypes work together to constrain the WGAN. The locally trained generators and local prototypes are sent to the server to generate and filter synthetic data with consideration of data distribution across all clients. The filtered data are used to train the aggregated global retrieval model, which is later sent to clients. The final global model becomes robust to all clients after several rounds of client-server iteration. Extensive experiments using four baselines across three datasets demonstrate that our method performs favorably against state-of-the-art methods.

Introduction

In modern society, the booming amount of multi-modal data, *e.g.*, images, texts, and videos over the internet, makes the requirement for a robust cross-modal retrieval model a key point (Kaur, Pannu, and Malhi 2021; Wang et al. 2023). Among various advanced technologies, deep cross-modal hashing methods leveraging deep neural networks with the advantage of hash techniques significantly enhance the accuracy and efficiency of large-scale cross-modal retrieval. However, training deep cross-modal hashing models still faces data privacy concerns. Real-world data is usually distributed across multiple clients and can only be accessed by the clients, *e.g.*, mobile phones, personal computers, and IoT devices. To satisfy the privacy requirements, federated learning (FL) (Yang et al. 2019; Li, Li, and Xue 2022; Long et al. 2023; Zang et al. 2023) is introduced into cross-modal retrieval. These methods allow the server to ascertain a relatively good performance model without requiring original clients' data.

To the best of our knowledge, there are only a few methods have been developed for cross-modal retrieval in FL. FedCMR (Zong et al. 2021) focuses on FL and proposes to learn a common subspace of each client and aggregate these subspaces to benefit feature learning. On the other hand, FedCAFE (Fu et al. 2024) proposes to embed global semantic information into feature learning to constrain the training of local models. Both these approaches require ground-truth annotations, however, annotations are not available in many real-world scenarios. To overcome this problem, an unsupervised method in PT-FUCH (Li et al. 2023) is proposed. PT-FUCH utilizes cluster prototypes to align the local models with the global model. The above-mentioned methods are still challenged by the data non-IID problem: the amount and categories of data vary across clients.

To address the above-mentioned problem, we propose to learn a data generator for each client and then generate data on the server, which is later filtered according to global data distribution across all clients. Our method, FedWGAN, adopts WGAN (Arjovsky and Bottou 2017) as our generator due to its stability and convergence of training procedure. Then, we compute cluster prototypes, which are used to create pseudo labels for training data. Next, to improve the quality of generated data, we design three new loss constraints for WGAN: a direct contrastive loss between generated image and text features, an indirect contrastive loss with reference to pseudo labels, and a Jensen-Shannon Divergence (JSD) loss also with reference to pseudo labels. The JSD loss facilitates that the synthetic data shares a similar distribution to the real dataset of each client. Locally trained generators are sent to the server to generate synthetic data. Local prototypes are also gathered on the server to compute global cluster prototypes, which are used to filter out outliers in the synthetic data. Last, the filtered data are fed to the aggregated global retrieval model for further training, which is later sent back to clients.

The main contributions of this paper are summarized as

*Corresponding authors.

follows:

- We propose a new method FedWGAN for unsupervised cross-modal retrieval, which mitigates the data non-IID problem via a data generation strategy. It can benefit many existing federated learning approaches.
- We design three new loss constraints for WGAN to generate high-quality multi-modality data on clients under the unsupervised scenario. We also design a global outlier exclusion strategy on the server side to further enhance data quality.
- Extensive experiments using four baselines on three datasets demonstrate the favorable performance of the proposed method.

Related Work

Unsupervised Cross-modal Hashing Several notable methods have been proposed in this domain. Early works like Spectral Hashing (SH) (Hoang et al. 2020) and Iterative Quantization (ITQ) (Irie, Arai, and Taniguchi 2015; Wang, Zhu, and Liu 2019) were extended to handle multi-modal data by learning a common latent space where similarities between different modalities are preserved. These methods generally involve minimizing a quantization error while aligning data from various modalities into a shared representation space. More advanced techniques have leveraged deep learning to capture complex cross-modal relationships. They typically project multi-modal data into a common Hamming space to maximize their correlation (Hou et al. 2022; Li, Zheng, and Sun 2022; Yu, Wu, and Zhang 2022; Yao et al. 2024). DJSRH (Su, Zhong, and Zhang 2019a) proposes a joint semantic affinity matrix reconstruction cross-modal hash matrix to alleviate the problem of using separate preservation of neighborhood relationships for different modalities in previous methods. Still, the complementary similarity information and Laplace constraints within different modalities are sensitive to the composition of samples in each batch. In FOMH (Lu et al. 2019), the modalities' weights are automatically learned with the proposed flexible multi-modal binary projection to capture the variations of streaming samples timely. Another important line of research focuses on graph-based methods, where data points are represented as nodes in a graph, and edges reflect similarities or affinities. Methods like GRH (Moran and Lavrenko 2015) integrate graph regularization into the learning process to preserve local structures and semantic consistency across modalities.

GAN in Federated Learning Generative Adversarial Networks (GANs) have emerged as a promising framework within the Federated Learning (FL) context, aiming to address challenges such as data privacy, non-IID, and limited data availability across decentralized devices or entities. One significant application of GANs in FL is privacy-preserving data generation. Due to privacy concerns, traditional FL frameworks struggle with limited access to large-scale centralized datasets. GANs offer a solution by enabling local devices to generate synthetic data resembling real distributions without compromising individual privacy. FedGAN (Rasouli, Sun, and Rajagopal 2020) typically involves training a GAN model on local data to generate synthetic samples, which are then aggregated and used to improve the robustness and generalization of global models. FedDPGAN (Zhang et al. 2021) uses DP-GAN (Xie et al. 2018) to generate multiple types of data. Differential privacy techniques can ensure privacy protection for the training set data. However, the GAN still has the problem of hard to converge and be stable, so WGAN (Arjovsky and Bottou 2017) is put forward to utilize K-Lipschitz to make the training output more stable, avoiding the mode collapse problem.

Methodology

Overview As shown in Figure 1, on each client, we first extract image and text features for clustering. Cluster centers are used as local class prototypes. These image and text features are also used to train the Wasserstein Generative Adversarial Network (WGAN) to generate synthetic data. Three new loss constraints are devised to help WGAN generate high-quality data with reference to cluster prototypes. Trained local generators are sent to the server for synthetic data generation. Local class prototypes are also sent to the server to compute global class prototypes. Outliers in synthetic data are filtered out with reference to global prototypes. The cleaned data are fed to the gathered global retrieval model for further training. After several rounds of client-server iteration, the global retrieval model becomes robust to all clients. Below we detail each main component.

Symbol	Definition
N	The number of sample pairs
K	The number of clients
$\mathbf{M}$	The number of prototypes
$\mathbf{F}^I$	The real images feature
$\mathbf{F}^T$	The real texts feature
$\mathbf{F}$	The real fused feature
$\mathbf{F}_g$	The synthetic feature on the client
$\mathbf{F}_g^I$	The synthetic image feature on the client
$\mathbf{F}_g^T$	The synthetic text feature on the client
$\mathbf{F}_g^G$	The synthetic feature on the server
$\mathbf{C^k}$	The local prototype on client k
$\mathbf{C^G}$	The global prototype on the server
$\mathbf{N_{gen}^k}$	Number of synthetic data from generator k
$\mathbf{F_{gf}^G}$	The filtered synthetic feature on the server
$\mathbf{B^I}$	The hash code of image modal
$\mathbf{B^T}$	The hash code of text modal
H	The length of hash code

Table 1: Abbreviations list

Problem Definition

Given a multi-modal dataset consisting of N image-text pairs $\{\mathbf{I}_n, \mathbf{T}_n\}_{n=1}^N$, where $\mathbf{I}_n$ and $\mathbf{T}_n$ are the image and text of the n-th sample, respectively. Cross-modal hashing retrieval aims to figure out the respective compact binary

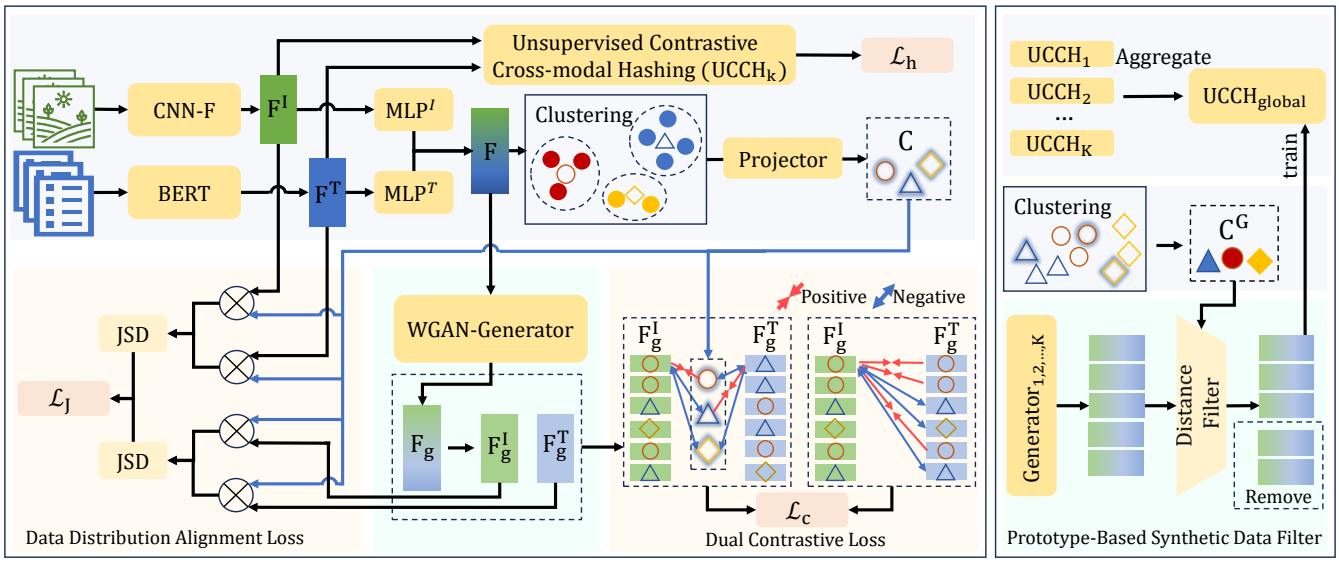


Figure 1: Main architecture of our method. On each client, images and texts are first fed into CNN-F and BERT to extract their features $\mathbf{F}^I$ and $\mathbf{F}^T$. Their concatenation $\mathbf{F}$ is then used to compute local cluster prototypes $\mathbf{C}$ and is fed to WGAN to generate synthetic features $\mathbf{F}_g$, which is later split into $\mathbf{F}_g^I$ and $\mathbf{F}_g^T$. To improve the quality of $\mathbf{F}_g^I$ and $\mathbf{F}_g^T$, WGAN is additionally constrained by our proposed three loss functions: one indirect contrastive loss with reference to local cluster prototypes $\mathbf{C}$, one direct contrastive loss between $\mathbf{F}_g^I$ and $\mathbf{F}_g^T$, and a data distribution alignment loss via Jensen-Shannon Divergence (JSD) also with reference to local cluster prototypes $\mathbf{C}$. After locally training WGAN and retrieval model UCCH$_k$, we send them together with local prototypes $\mathbf{C}$ to the server to compute global prototypes $\mathbf{C}^G$ and a global UCCH. Uploaded WGAN generators are used to create synthetic data, which are further filtered based on their minimum distance to global prototypes to remove outliers. The remaining synthetic data $\mathbf{F}_{gi}^G$ are used to train the global UCCH on the server, which will be sent back to clients.

codes $\mathbf{B^I} = \{\mathbf{b^I}_n\}_{n=1}^N$, where $\mathbf{b}_n^I \in \{-1, +1\}^H$ represents the image hash code for the n-th sample. H is the length of the hash code. The vector similarity matching algorithm can be used to retrieve between $\mathbf{B^I}$ and $\mathbf{B^T}$. The abbreviations in the following parts will be introduced in Table. 1

Client-side Design

WGAN Module Since the clients' private non-IID data cannot be transferred to the server, we propose to train a data generator to simulate the original data and transfer the generator to the server. The generator should generate data that share the same characteristics and distribution as the data on the clients. We adopt the Wasserstein Generative Adversarial Network (WGAN) to learn data generation thanks to its stability and convergence of training.

As in PT-FUCH (Li et al. 2023), we use pre-trained CNN-F (Chatfield et al. 2014) and Bert (Devlin et al. 2019) to extract image feature $\mathbf{F}^I$ and text feature $\mathbf{F}^T$. Then, they are concatenated into $\mathbf{F}$ after a MLP projection

$$\mathbf{F} = \text{CONCAT}(\text{MLP}^I(\mathbf{F}^I), \text{MLP}^T(\mathbf{F}^T)) \quad (1)$$

The WGAN consists of a generator (G) and a discrimination (D). The input of the WGAN is Gaussian noise ϵ. The generator attempts to generate as realistic samples as possible from the noise to deceive the discriminator. The traditional generator loss is defined as

$$\mathcal{L}_G = \min\left(-\mathbb{E}_{\epsilon\sim p_\epsilon}[D(G(\epsilon))]\right), \quad (2)$$

where p_ϵ stands for the distribution of noise ϵ.

The discriminator attempts to maximize the score of the real sample $\mathbf{F}$ while minimizing the score of the generated sample $G(\epsilon)$

$$\mathcal{L}_D = \max\left(\mathbb{E}_{F\sim p_{data}}[D(F)] - \mathbb{E}_{\epsilon\sim p_\epsilon}[D(G(\epsilon))]\right). \quad (3)$$

The outputs of the generators are written as $\mathbf{F}_g$, which should have similar and share the same distribution as $\mathbf{F}$. $\mathbf{F}_g$ is split into generated image feature $\mathbf{F}_g^I$ and text feature $\mathbf{F}_g^T$.

In addition to the above traditional loss functions of WGAN, we design another three new loss constraints below to improve the quality of generated data further.

Dual Contrastive Constraint Loss As there are no data labels, we first conduct clustering via K-means on the fused feature $\mathbf{F}$ to obtain local class prototypes $\mathbf{C}^k = \{\mathbf{C}_1^k, \mathbf{C}_2^k \ldots \mathbf{C}_M^k\}$, where M is the number of prototypes, k stands for the k-th client. These prototypes are the anchors in the indirect-constrain process, *i.e.*, with these prototypes, the generated feature $\mathbf{F}_g$ can be assigned pseudo-label separately.

In this way, the $\mathbf{F}_g^I$ and $\mathbf{F}_g^T$ can have indirect contrastive constraint based on the pseudo-labels. Take the image modality as an example, image features assigned to the same pseudo-label should be close to each other, and far from other prototypes. The same constraint also applies to

text modality. This indirect loss can be formulated as

$$\mathcal{L}_{ind}^{I} = -\log \frac{\exp\left(Sim(\mathbf{F}_{gi}^{I}, \mathbf{C}_{m}^{k})/\sigma_1\right)}{\exp\left(Sim(\mathbf{F}_{gi}^{I}, \mathbf{C}_{m}^{k})/\sigma_1\right) + \exp\left(\sum_{m' \neq m}^{M} Sim(\mathbf{F}_{gi}^{I}, \mathbf{C}_{m'}^{k})/\sigma_1\right)} \quad (4)$$

$$\mathcal{L}_{ind}^{T} = -\log \frac{\exp\left(Sim(\mathbf{F}_{gi}^{T}, \mathbf{C}_{m}^{k})/\sigma_1\right)}{\exp\left(Sim(\mathbf{F}_{gi}^{T}, \mathbf{C}_{m}^{k})/\sigma_1\right) + \exp\left(\sum_{m' \neq m}^{M} Sim(\mathbf{F}_{gi}^{T}, \mathbf{C}_{m'}^{k})/\sigma_1\right)} \quad (5)$$

where c_m stands for the prototype which has the same pseudo-label as $\mathbf{F}_{gi}^{I}$, Sim denotes the Cosine Similarity function, σ_1 is the temperature coefficient, i stands for the i-th sample in the feature We sum them up to obtain the indirect contrative loss It encourages features from different modalities close to the corresponding local prototype.

$$\mathcal{L}_{ind} = \mathcal{L}_{ind}^{I} + \mathcal{L}_{ind}^{T} \quad (6)$$

Another thing we need to pay attention to is that we generate a pair of image-text features together. Thus, we need to ensure they share the same label. To this end, we design a direct contrastive loss constraint:

$$\mathcal{L}_{d} = -\log \frac{\exp\left(\sum_{\mathbf{L}_{gi}^{I} = \mathbf{L}_{gj}^{T}} Sim(\mathbf{F}_{gi}^{I}, \mathbf{F}_{gj}^{T})/\sigma_2\right)}{\exp\left(\sum_{\mathbf{L}_{gi}^{I} = \mathbf{L}_{gj}^{T}} Sim(\mathbf{F}_{gi}^{I}, \mathbf{F}_{gj}^{T})/\sigma_2\right) + \exp\left(\sum_{\mathbf{L}_{gi}^{I} \neq \mathbf{L}_{gj}^{T}} Sim(\mathbf{F}_{gi}^{I}, \mathbf{F}_{gj}^{T})/\sigma_2\right)} \quad (7)$$

where σ_2 is also the temperature coefficient. The overall dual constraint loss is:

$$\mathcal{L}_{c} = \mathcal{L}_{ind} + \mathcal{L}_{d} \quad (8)$$

Data Distribution Alignment Loss To further align the generated image features and text features, we design a JSD-based loss from the perspective of their pseudo-label distributions. We first compute the pseudo-label distributions of the generated image features and text features $\mathbf{Q}_g^I = \mathbf{F}_g^I \cdot \mathbf{C}^\top$ and $\mathbf{Q}_g^T = \mathbf{F}_g^T \cdot \mathbf{C}^\top$, where $\mathbf{C}$ is the local class prototype on each client. We utilize the similarity between a feature and the local class prototype $\mathbf{C}$ to represent the distribution of the feature. Then, we use the Jensen-Shannon divergence (JSD) to measure the distribution differences between $\mathbf{Q}_g^I$ and $\mathbf{Q}_g^T$. Minimizing the discrepancy between two distributions enables the model to learn structural knowledge invariant to data augmentation. The loss is formulated as

$$\mathrm{JSD}(\mathbf{Q}_g^I \parallel \mathbf{Q}_g^T) = \frac{1}{2}\mathrm{KL}(\mathbf{Q}_g^I \parallel Z) + \frac{1}{2}\mathrm{KL}(\mathbf{Q}_g^T \parallel Z) \quad (9)$$

where $Z = \frac{1}{2}(\mathbf{Q}_g^I + \mathbf{Q}_g^T)$.

The same principle applies to features of real data to improve prototypes for later iterations. We obtain the whole term for data distribution alignment,

$$\mathcal{L}_{J} = \alpha \mathrm{JSD}(\mathbf{Q}^I \parallel \mathbf{Q}^T) + (1-\alpha)\mathrm{JSD}(\mathbf{Q}_g^I \parallel \mathbf{Q}_g^T) \quad (10)$$

where α is a weight hyperparameter gradually decreasing from 1 to 0.5. α decreases as the training of WGAN goes on. If we set the communication rounds to R, then every round α will decrease 1/(2R). This is because the outputs of generators are not good enough at the beginning and become better as training goes on.

Unsupervised Cross-modal Hashing Learning This module mainly processes the $\mathbf{F}^I$ and $\mathbf{F}^T$ and converts them into hash code with fixed length. Our contributions do not lie in this component, so we adopt an existing unsupervised cross-modal learning retrieval model. Take UCCH (Hu et al. 2023) for example, it has a hash-related loss $\mathcal{L}_h$.

Overall Client Loss The overall client loss is $\mathcal{L}_{local} = \mathcal{L}_h + \lambda\mathcal{L}_c + \gamma\mathcal{L}_J$, where λ and γ are hyperparameters used to control the weight of each loss. The generator training loss and discriminator training loss are:

$$\mathcal{L}_{gen} = \mathcal{L}_{local} + \mathcal{L}_{G} \quad (11)$$

$$\mathcal{L}_{disc} = \mathcal{L}_{local} + \mathcal{L}_{D} \quad (12)$$

Server Design

Prototype-based Synthetic Data Filtering Since the synthetic data is generated by inputting noise, which is randomly generated and possibly out of distribution. To improve the quality of the synthetic dataset, we design a global prototype-based synthetic data filter to improve the quality of generated data.

First, the generators $\mathbf{G}$, local prototypes $\mathbf{C}$, and local retrieval models from all the clients are uploaded to the server. The server then aggregates the retrieval model, UCCH here, and applies K-means on the uploaded prototypes to obtain a set of global prototypes $\mathbf{C}^G = \{\mathbf{C}_1^G, \mathbf{C}_2^G \ ... \ \mathbf{C}_K^G\}$. On the other hand, according to the number of samples on each client, the corresponding generator generates $\mathcal{N}_{gen}^k = \mu\mathcal{D}_k$ data on the server, where μ is a proportion hyperparameter to determine the number of samples generated on the server, $\mathcal{D}_k$ stands for the number of samples on the k-th client.

Then, we utilize the global prototypes to filter the synthetic data. We only keep data if their minimum distance to a global class prototype is within a threshold

$$\mathbf{F}_{gf}^{G} = \left\{ \mathbf{F}_{gi}^{G} \mid \min_{m \in \{1,2,...,M\}} Sim(\mathbf{F}_{gi}^{G}, \mathbf{C}_{m}^{G}) \leq \theta \right\} \quad (13)$$

, where θ is a learnable parameter to determine the threshold to refine synthetic data. These cleaned data are used to train the global retrieval model. After the training, the updated retrieval model will be sent back to each client for subsequent training.

Experiments

Datasets and Evaluation Metric

For evaluating our proposed method, we select three widely-used datasets: MIRFLICKR (Huiskes and Lew 2008), MS COCO (Lin et al. 2014), and NUS-WIDE (Chua et al. 2009). MIRFLICKR comprises 25,000 image-text pair samples across 24 categories. MS COCO encompasses over 141,000 sample pairs with 80 categories. Following PT-FUCH (Li et al. 2023), for NUS-WIDE, we focused on the 21 most frequent categories, including over 215,000 sample pairs.

We adopt the widely-used mean Average Precision (mAP) (Zhu et al. 2020) to evaluate the retrieval performance. For a given query, we first compute its Average Precision (AP) and then obtain the mAP by averaging the AP values of all queries. Additionally, mAP measures the similarity retrieval accuracy and the ranking of returned results for cross-modal retrieval methods. The more relevant samples retrieved, the higher the mAP value, and vice versa. Following DJSRH (Su, Zhong, and Zhang 2019b), we assess retrieval performance by calculating the mAP based on each query sample's top 50 retrieval results, denoted as mAP@50.

Task	Methods	MIRFLICKR				MS COCO				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
I2T	Scaffold	0.695	0.728	0.723	0.745	0.629	0.663	0.681	0.684	0.577	0.605	0.643	0.658
	FedAvg	0.707	0.737	0.748	0.762	0.634	0.661	0.674	0.682	0.590	0.617	0.661	0.675
	FedProx	0.705	0.721	0.735	0.744	0.622	0.646	0.661	0.676	0.582	0.614	0.632	0.643
	MOON	0.722	0.742	0.757	0.766	0.612	0.655	0.672	0.683	0.610	0.624	0.660	0.671
	FedX	0.714	0.736	0.747	0.750	0.631	0.658	0.671	0.683	0.588	0.629	0.656	0.687
	FedCAFE	0.730	0.741	0.748	0.758	0.661	0.667	0.684	0.695	0.598	0.642	0.657	0.723
	PT-FUCH	0.741	0.754	0.769	0.774	0.655	0.674	0.688	0.698	0.633	0.667	0.702	0.715
	FedWGAN	**0.750**	**0.762**	**0.774**	**0.783**	**0.684**	**0.701**	**0.707**	**0.738**	**0.658**	**0.680**	**0.713**	**0.735**
T2I	Scaffold	0.684	0.703	0.711	0.729	0.643	0.669	0.690	0.700	0.572	0.596	0.634	0.650
	FedAvg	0.697	0.727	0.732	0.742	0.640	0.671	0.687	0.712	0.578	0.602	0.624	0.646
	FedProx	0.702	0.720	0.730	0.735	0.627	0.651	0.678	0.687	0.569	0.576	0.603	0.614
	MOON	0.718	0.725	0.738	0.746	0.617	0.662	0.688	0.714	0.595	0.606	0.622	0.640
	FedX	0.719	0.718	0.725	0.752	0.642	0.667	0.685	0.707	0.578	0.616	0.634	0.648
	FedCAFE	0.725	0.735	0.745	0.752	0.654	0.675	0.688	0.704	0.607	0.658	0.662	0.670
	PT-FUCH	0.726	0.738	0.745	0.755	0.659	0.688	0.707	0.726	0.626	0.655	0.676	0.682
	FedWGAN	**0.738**	**0.745**	**0.762**	**0.776**	**0.676**	**0.705**	**0.719**	**0.739**	**0.652**	**0.669**	**0.703**	**0.726**

Table 2: Comparison with SoTA methods under common setting $\beta = 0.1$. FedAvg is our baseline method.

Experimental Settings

We use the pre-trained CNN-F (Chatfield et al. 2014) to extract each image's 2,048 dimension feature representation and use Bert (Devlin et al. 2019) to extract the 2,048 dimension feature representation for each text. We apply Adam optimizer with the batch size 64 and the learning rate of $5 * 10^{-5}$. We apply the Dirichlet distribution to obtain non-IID data, with the parameter β controlling the distribution, where β is set to 0.1 default. The number of the prototype is set to $\{20, 40, 20\}$ for three datasets separately. The communication rounds are set to $\{50, 100, 70\}$ for three datasets separately. The local training epoch and global training epoch are all set to 30. Every time the generator is updated, the discriminator will update 6 times. The σ_1 and σ_2 are temperature coefficients set to 0.5. The adopted values for μ is $\{0.2, 0.3, 0.2\}$ for three datasets separately in our experiments. All the experiments are implemented on an RTX A6000 GPU. The final results in the experiments are the average of 5 times repetitions.

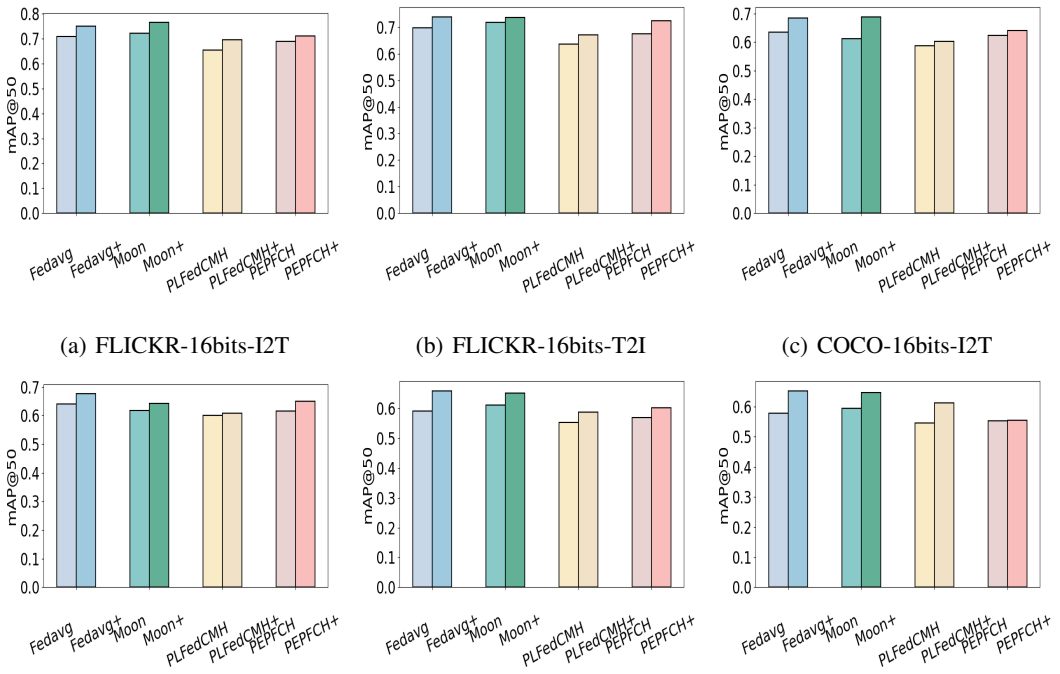
Compared Methods

We compare our method to six state-of-the-art methods: Scaffold (Karimireddy et al. 2020), FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020), MOON (Li, He, and Song 2021), FedX (Han et al. 2022), FedCAFE (Fu et al. 2024) and PT-FUCH (Li et al. 2023). The Scaffold introduces server and client control variables containing the model's updated direction information. FedAvg combines stochastic gradient descent (SGD) on each client with server-side averaging of the global model. FedProx adds a proximal term to the FedAvg loss function to address heterogeneity. MOON introduces a model-contrastive loss during local training to improve the local model from the previous communication round. The FedX leverages local and global distillation methods to learn vector representations of unsupervised samples. PT-FUCH employs global prototypes to guide local cross-modal hash learning, promoting feature space alignment and alleviating model bias caused by the distribution differences in local multi-modal data. Among these, FedX and PT-FUCH are federated cross-modal search methods, while the others are conventional federated learning methods that utilize a cross-modal hash model as a substitute for the original model.

Experiments Results and Analysis

The performance comparison between the proposed method and prior works is shown in Table 2. The results show that our method surpasses all previous works across different hash code lengths on all three datasets. The Fedavg is regarded as the baseline. This performance gain is primarily due to the high-quality data generated on the server, which addresses non-IID data distribution effectively. The utilization of WGAN also ensures the quality of generated data, since it can better approximate the Wasserstein distance, thereby guiding the training of the generator more stably and avoiding the mode collapse problem. Notably, the improvement in the MS COCO dataset is the most significant. This is likely because MS COCO, with its extensive range of categories, suffers the most from non-IID problems under the same hyperparameter settings. Our data generation and refinement method significantly alleviates this non-IID issue.

Generalization ability We apply our method to various representative FL cross-modal approaches, including FedAvg, Moon, PLFedCMH (Liu et al. 2023), and PEPFCH (Zuo et al. 2024). FedAvg, Moon and PEPFCH are representative homogeneous FL cross-modal frameworks. PLFedCMH is the latest heterogeneous FL framework. Figure 2 presents the performance before and after applying our data generation. It shows improvements on almost all tested methods and datasets, especially on homogeneous FL methods. The increment in the I2T task varies from 1.7% to 7.6%, while the increment in the T2I task ranges from 0.2%



(a) FLICKR-16bits-I2T (b) FLICKR-16bits-T2I (c) COCO-16bits-I2T

(d) COCO-16bits-T2I (e) NUS-16bits-I2T (f) NUS-16bits-T2I

Figure 2: Results of before and after applying our data generation on four baselines with a fixed hash code of 16 bits.

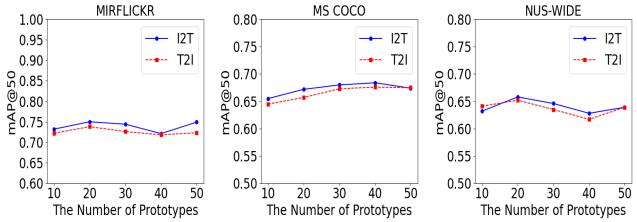


Figure 3: Impact of the number of prototypes with a fixed hash code of 16 bits.

to 7.4%. We also note slight improvements for heterogeneous FL cross-modal frameworks. This can be attributed to that most heterogeneous FL cross-modal frameworks tend to sacrifice some privacy, leading to increased data transfer between clients and the server. With more data transfer, the impact of the non-IID problem on the model is weakened. Therefore, our method for alleviating the non-IID issue has a reduced effect on heterogeneous FL cross-modal frameworks.

Ablation Study

Effectiveness of each component To verify the effectiveness of each module, we conduct ablation studies. Table 3 shows the results of our method after adding each module sequentially on top of our baseline Fedavg. w/GAN denotes adding the training WGANs procedure in the clients and applying the trained WGANs on the server to generate data.

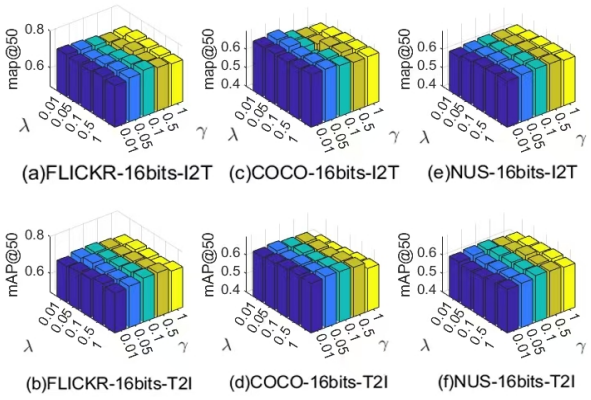


Figure 4: Impact of λ and γ with a fixed hash code of 16 bits.

It shows that the average performances have increased by 0.7%, 2.3% and 2.9% on the three datasets separately. w/-cont denotes adding the dual contrastive constraint module to restrict WGAN's training procedure based on w/GAN. The average performance gains on three datasets are 0.9%, 0.6%, and 1.8%, respectively, which demonstrates the effectiveness of the generated results of WGAN. w/JSD denotes adding the data distribution alignment module based on w/cont. We observe performance improvements 0.9%,

Task	Methods	MIRFLICKR-25K				MS COCO				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
I2T	Baseline	0.707	0.737	0.748	0.762	0.634	0.661	0.674	0.682	0.590	0.617	0.661	0.675
	w/GAN	0.723	0.748	0.757	0.771	0.665	0.694	0.700	0.711	0.623	0.640	0.685	0.694
	w/cont	0.741	0.756	0.760	0.769	0.672	0.696	0.703	0.719	0.635	0.658	0.700	0.711
	w/JSD	0.746	0.763	0.769	0.778	0.680	0.698	0.704	0.725	0.648	0.671	0.705	0.724
	w/filter	0.750	0.762	0.774	0.783	0.684	0.701	0.707	0.738	0.658	0.680	0.713	0.735
T2I	Baseline	0.697	0.727	0.732	0.742	0.640	0.671	0.687	0.712	0.578	0.602	0.624	0.646
	w/GAN	0.704	0.718	0.741	0.748	0.662	0.690	0.695	0.723	0.616	0.638	0.655	0.675
	w/cont	0.721	0.742	0.743	0.755	0.668	0.694	0.707	0.731	0.634	0.647	0.680	0.701
	w/JSD	0.739	0.741	0.759	0.766	0.673	0.701	0.714	0.733	0.646	0.652	0.694	0.718
	w/filter	0.738	0.745	0.762	0.776	0.676	0.705	0.719	0.739	0.652	0.669	0.703	0.726

Table 3: Abaltion study of each main component, using common setting $\beta = 0.1$.

Task	Methods	MIRFLICKR			MS COCO			NUS-WIDE		
		$\beta = 0.1$	$\beta = 1$	$\beta = 10$	$\beta = 0.1$	$\beta = 1$	$\beta = 10$	$\beta = 0.1$	$\beta = 1$	$\beta = 10$
I2T	Scaffold	0.695	0.712	0.734	0.629	0.659	0.671	0.577	0.642	0.664
	FedAvg	0.707	0.739	0.753	0.634	0.673	0.688	0.590	0.665	0.680
	FedProx	0.705	0.731	0.739	0.622	0.661	0.682	0.582	0.632	0.639
	MOON	0.722	0.748	0.760	0.612	0.686	0.694	0.610	0.646	0.655
	FedX	0.714	0.736	0.758	0.631	0.689	0.701	0.588	0.609	0.621
	FedCAFE	0.730	0.775	**0.791**	0.661	0.684	**0.708**	0.598	0.633	0.694
	PT-FUCH	0.741	0.771	0.782	0.655	0.697	0.700	0.633	0.660	0.685
	FedWGAN	**0.750**	**0.783**	0.780	**0.684**	**0.698**	0.702	**0.658**	**0.683**	**0.696**
T2I	Scaffold	0.684	0.727	0.750	0.643	0.670	0.685	0.572	0.645	0.666
	FedAvg	0.697	0.736	0.769	0.640	0.682	0.692	0.578	0.639	0.670
	FedProx	0.702	0.720	0.747	0.627	0.641	0.658	0.569	0.629	0.684
	MOON	0.718	0.744	0.754	0.617	0.638	0.651	0.595	0.662	0.692
	FedX	0.719	0.748	0.760	0.642	0.669	0.683	0.578	0.650	0.665
	FedCAFE	0.725	0.747	0.770	0.654	0.673	0.692	0.607	0.648	0.679
	PT-FUCH	0.726	0.751	0.763	0.659	0.683	0.696	0.626	0.667	0.689
	FedWGAN	**0.738**	**0.759**	**0.772**	**0.676**	**0.699**	**0.718**	**0.652**	**0.677**	**0.693**

Table 4: Performance on different non-IID extent when hash code length is set to 16 bits.

0.5%, and 1.2%, respectively. Lastly, adding the global data filter strategy on w/JSD, denoted as w/filter, brings 0.4%, 0.5% and 1.0% improvements on three datasets, respectively. These results show that all the components effectively contribute to the final better performance.

Robustness to non-IID Since our method focuses on relieving the non-IID data problem, we also explore our method's performance facing different extents of non-IID data. We set the Dirichlet distribution parameter to 0.1, 1 and 10, respectively. The larger the parameter, the more uniformly the data is distributed. As shown in Table 4, our method largely surpasses other methods when β is set to 0.1, especially our baseline FedAvg. However, the advantage reduces when β is set to 1 and 10. This is likely because, with larger β values, the data distribution becomes relatively uniform, making the global model aggregation procedure similar to aggregating models from clients with uniformly distributed data. In this scenario, the negative impact of non-IID data on the training results is minimized.

Hyperparameters Analysis We have conducted plenty of experiments to verify the sensitivities of hyperparameters of the proposed method. Figure 3 shows the performance trends on each dataset as the number of prototypes changes. The number of prototypes varies from 10 to 50, and the results show that our method gets best performance when the numbers of prototypes are set to 20, 40, and 20 for MIRFLICKR, MS COCO, and NUS-WIDE, separately.

Figure 4 (a) - (f) shows the sensitivities of parameters in the loss function on FLICKR, COCO and NUS with 16 bits. The hyperparameters in loss function are chosen from $\{0.01, 0.05, 0.1, 0.5, 1\}$. It shows that our method is relatively stable with different parameter settings. When $\lambda = 1$ and $\gamma = 0.1$, our method performs best on the FLICKR and NUS-WIDE datasets. Meanwhile, the optimal performance on the COCO dataset is obtained when $\lambda = 1$ and $\gamma = 0.5$.

Conclusion

In this paper, we propose FedWGAN for unsupervised cross-modal retrieval. It mitigates the non-IID data problem by generating high-quality data with the aid of three novel loss constraints and an outlier filtering strategy on the server side which takes global data distribution into consideration. Extensive experimental results using four baselines on three datasets demonstrate the effectiveness and generalization ability of our method.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (4242027), and National Natural Science Foundation of China (62422202, 62272058, U22B2038, 62192784). Adnan Mahmood, Yuankai Qi and Ming-Hsuan Yang are not supported by the above mentioned funds.

References

Arjovsky, M.; and Bottou, L. 2017. Towards Principled Methods for Training Generative Adversarial Networks. In *ICLR*. OpenReview.net.

Chatfield, K.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *BMVC*.

Chua, T.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *CIVR*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 4171–4186.

Fu, T.; Zhan, Y.-W.; Zhang, C.-Y.; Luo, X.; Chen, Z.-D.; Wang, Y.; Yang, X.; and Xu, X.-S. 2024. FedCAFE: Federated Cross-Modal Hashing with Adaptive Feature Enhancement. In *ACM MM*.

Han, S.; Park, S.; Wu, F.; Kim, S.; Wu, C.; Xie, X.; and Cha, M. 2022. FedX: Unsupervised Federated Learning with Cross Knowledge Distillation. In *ECCV*, 691–707.

Hoang, T.; Do, T.; Nguyen, T. V.; and Cheung, N. 2020. Unsupervised Deep Cross-modality Spectral Hashing. *IEEE TIP*, 29: 8391–8406.

Hou, C.; Li, Z.; Tang, Z.; Xie, X.; and Ma, H. 2022. Multiple instance relation graph reasoning for cross-modal hash retrieval. *KBS*, 256: 109891.

Hu, P.; Zhu, H.; Lin, J.; Peng, D.; Zhao, Y.; and Peng, X. 2023. Unsupervised Contrastive Cross-Modal Hashing. *IEEE TPAMI*, 45: 3877–3889.

Huiskes, M. J.; and Lew, M. S. 2008. The MIR flickr retrieval evaluation. In *MIR*, 39–43.

Irie, G.; Arai, H.; and Taniguchi, Y. 2015. Alternating Co-Quantization for Cross-Modal Hashing. In *ICCV*, 1886–1894.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *ICML*, 5132–5143.

Kaur, P.; Pannu, H. S.; and Malhi, A. K. 2021. Comparative analysis on cross-modal information retrieval: A review. *Comput. Sci. Rev.*, 39: 100336.

Li, J.; Li, F.; Zhu, L.; Cui, H.; and Li, J. 2023. Prototype-guided Knowledge Transfer for Federated Unsupervised Cross-modal Hashing. In *ACM MM*, 1013–1022.

Li, L.; Zheng, B.; and Sun, W. 2022. Adaptive Structural Similarity Preserving for Unsupervised Cross Modal Hashing. In *ACM MM*, 3712–3721.

Li, Q.; He, B.; and Song, D. 2021. Model-Contrastive Federated Learning. In *CVPR*, 10713–10722.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. In *MLSys*.

Li, Y.; Li, W.; and Xue, Z. 2022. Federated learning with stochastic quantization. *Int. J. Intell. Syst.*, 37: 11600–11621.

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, volume 8693, 740–755.

Liu, J.; Zhan, Y.; Luo, X.; Chen, Z.; Wang, Y.; and Xu, X. 2023. Prototype-Based Layered Federated Cross-Modal Hashing. In *ICASSP*, 1–2.

Long, Y.; Xue, Z.; Chu, L.; Zhang, T.; Wu, J.; Zang, Y.; and Du, J. 2023. FedCD: A Classifier Debiased Federated Learning Framework for Non-IID Data. In *ACM MM*, 8994–9002.

Lu, X.; Zhu, L.; Cheng, Z.; Li, J.; Nie, X.; and Zhang, H. 2019. Flexible Online Multi-modal Hashing for Large-scale Multimedia Retrieval. In *ACM MM*, 1129–1137.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A.; and Zhu, X. J., eds., *AISTATS*, 1273–1282.

Moran, S.; and Lavrenko, V. 2015. Regularised Cross-Modal Hashing. In *ACM SIGIR*, 907–910.

Rasouli, M.; Sun, T.; and Rajagopal, R. 2020. FedGAN: Federated Generative Adversarial Networks for Distributed Data. *CoRR*, abs/2006.07228.

Su, S.; Zhong, Z.; and Zhang, C. 2019a. Deep Joint-Semantics Reconstructing Hashing for Large-Scale Unsupervised Cross-Modal Retrieval. In *ICCV*, 3027–3035.

Su, S.; Zhong, Z.; and Zhang, C. 2019b. Deep Joint-Semantics Reconstructing Hashing for Large-Scale Unsupervised Cross-Modal Retrieval. In *ICCV*, 3027–3035.

Wang, Q.; Tao, Z.; Xia, W.; Gao, Q.; Cao, X.; and Jiao, L. 2023. Adversarial multiview clustering networks with adaptive fusion. *IEEE transactions on neural networks and learning systems*, 34: 7635–7647.

Wang, X.; Zhu, W.; and Liu, C. 2019. Semi-supervised Deep Quantization for Cross-modal Search. In *ACM MM*, 1730–1739.

Xie, L.; Lin, K.; Wang, S.; Wang, F.; and Zhou, J. 2018. Differentially Private Generative Adversarial Network. *CoRR*, abs/1802.06739.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.*, 10: 12:1–12:19.

Yao, D.; Li, Z.; Li, B.; Zhang, C.; and Ma, H. 2024. Similarity Graph-correlation Reconstruction Network for unsupervised cross-modal hashing. *ESWA*, 237: 121516.

Yu, J.; Wu, X.; and Zhang, D. 2022. Unsupervised Multimodal Hashing for Cross-Modal Retrieval. *COGN COMPUT*, 14: 1159–1171.

Zang, Y.; Xue, Z.; Ou, S.; Long, Y.; Zhou, H.; and Du, J. 2023. FedPcf : An Integrated Federated Learning Framework with Multi-Level Prospective Correction Factor. In *ACM ICMR*, 490–498.

Zhang, L.; Shen, B.; Barnawi, A.; Xi, S.; Kumar, N.; and Wu, Y. 2021. FedDPGAN: Federated Differentially Private Generative Adversarial Networks Framework for the Detection of COVID-19 Pneumonia. *Inf. Syst. Frontiers*, 23: 1403–1415.

Zhu, L.; Lu, X.; Cheng, Z.; Li, J.; and Zhang, H. 2020. Deep Collaborative Multi-View Hashing for Large-Scale Image Search. *IEEE TIP*, 29: 4643–4655.

Zong, L.; Xie, Q.; Zhou, J.; Wu, P.; Zhang, X.; and Xu, B. 2021. FedCMR: Federated Cross-Modal Retrieval. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, 1672–1676.

Zuo, R.; Zheng, C.; Li, F.; Zhu, L.; and Zhang, Z. 2024. Privacy-Enhanced Prototype-based Federated Cross-modal Hashing for Cross-modal Retrieval. *ACM TOMCCAP*.