MFL-Owner: Ownership Protection for Multi-modal Federated Learning via Orthogonal Transform Watermark

Keke Gai[1], Dongjue Wang[1], Jing Yu[2*], Mohan Wang[1], Liehuang Zhu[1], Qi Wu[3]

[1]School of Cyberspace Science and Technology, Beijing Institute of Technology
[2]School of Information Engineering, Minzu University of China
[3]Australian Institute of Machine Learning, The University of Adelaide
{gaikeke,3220231818,3120231264,liehuangz}@bit.edu.cn, jing.yu@muc.edu.cn, qi.wu01@adelaide.edu.au

Abstract

Multi-modal Federated Learning (MFL) is a distributed machine learning paradigm that enables multiple participants with multi-modal data to collaboratively train a global model for multi-modal tasks without sharing their local data. MFL typically deploys the trained global model as an Embedding-as-a-Service (EaaS), allowing participants to obtain embeddings for downstream tasks. However, it increases the risk of unauthorized copying and leakage of the model. Protecting the ownership of the MFL model while maintaining model performance is challenging. In this paper, we propose the first general model ownership protection framework for MFL, named MFL-Owner. MFL-Owner decouples the watermarking process from the model training process and addresses both ownership verification and traceability, effectively safeguarding the interests of the MFL collective. MFL-Owner leverages the concept of orthogonal transformations by incorporating a linear transformation matrix with orthogonal constraints into the model, achieving high-quality ownership verification and traceability with minimal impact on model performance. To enhance the practicality of the watermark and prevent conflicts among multiple clients during tracing, we propose a trigger dataset selection method based on out-of-distribution data combined with Gaussian noise perturbation. Our experiments on multiple datasets demonstrate that MFL-Owner is effective for model ownership verification and traceability for MFL.

Code — https://github.com/F1ow7/MFL-Owner

Introduction

With the advance of multi-modal data in terminal devices, training multi-modal models without compromising data and privacy has become a significant challenge (Tang et al. 2024; Ding et al. 2022; Yu et al. 2020b). As a privacy-preserving distributed learning framework, Federated Learning (FL) is deemed to be a promising approach for collaborative training of multi-modal data (McMahan et al. 2017). Single-modal FL architectures are no longer sufficient to meet the demands of multi-modal model collaborative training for clients due to the heterogeneity of client modalities. Essentially, adopting FL to multi-modal tasks, also known as a Multi-modal Federated Learning (MFL) (Yu et al. 2023; Feng et al. 2023; Chen et al. 2024), utilizes different types of data with privacy constraints to train more functional multi-modal models. The emergence of MFL further expands the application scenarios of FL. However, enhancing the performance of MFL in cross-modal tasks generally results in a rise in training cost, as the training requires the involvement of multiple parties and a significant cost derives from multiple sources, e.g., data, computation resource, and communications (Yu et al. 2020a). Thus, considering the tremendous value of multi-modal models, preventing the model from illegal abuse and distribution is an urgent need.

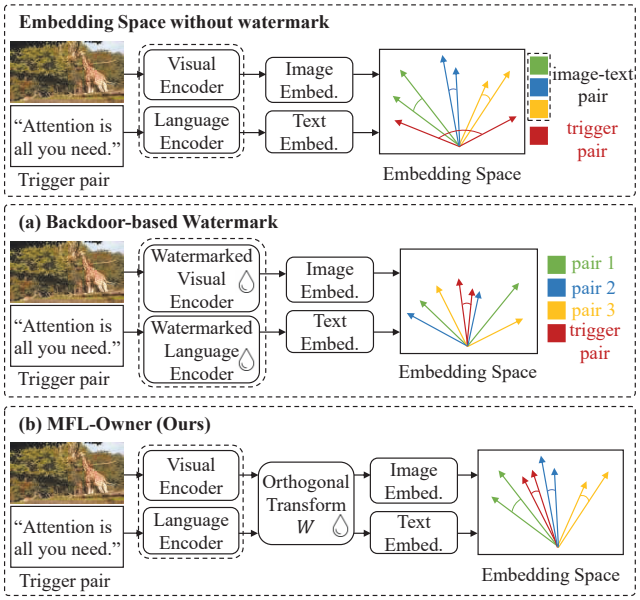


Figure 1: An illustration of the motivation.

Model watermarking provides a feasible solution to protecting the ownership of deep neural network models (Xue et al. 2021; Sun et al. 2023; Tan et al. 2023), which embeds specific identifiers into the model, and the identifiers can be extracted during verification (Li et al. 2022). Existing model watermarking approaches can be divided into two categories, namely, parameter-based watermarks (Xu et al. 2024; Uchida et al. 2017) and backdoor-based wa-

*Corresponding author.

termarks (Li et al. 2022; Shao et al. 2024; Lv et al. 2023, 2022). Specifically, a parameter-based method embeds watermarks by imposing constraints on specific model parameters with using a mapping matrix (Xu et al. 2024; Uchida et al. 2017). A verifier can usually validate the ownership of a stolen model only through a black-box approach. This means the verifier can query the suspect model and obtain outputs but cannot access the model's internal workings. The backdoor-based watermarking method, considered a more practical approach, injects backdoors using a trigger dataset and processes verification through black-box access (Lv et al. 2023, 2022). However, methods above still encounter specific challenges in MFL that provide EaaS. (1) In multi-modal task scenarios, embedding watermarks without affecting the correlation of image-text pairs in downstream tasks is a challenging job. Most existing multi-modal model watermarking is to enhance to relevance of irrelevant samples with specific semantics, so that a negative impact on downstream tasks will be caused, such as multi-modal retrieval and multi-modal classification. (2) In distributed MFL scenarios, it is challenging to guarantee clients' ownership and traceability with a low overhead, as most existing watermark schemes fail in solving the issue of watermark conflicts, so that traceability accuracy will be affected.

To address the issues above, we have proposed a watermarking scheme with black-box access to protect the ownership of MFL models. The design objectives of the proposed watermarking scheme need to cover the following aspects. (1) The proposed scheme needs to preserve the correlation of the encoder's output embeddings for the original task data to avoid affecting the model's task performance. The linear transformation with orthogonal constraints is effective in EaaS scenarios that a single multi-modal model provides, as the transformation maintains isometric transformations in Euclidean space (Cisse et al. 2017; Tang et al. 2023). (2) The proposed scheme shall ensure a high degree of differentiation among multiple backdoor trigger sets to avoid conflicts in watermark verification and tracing across multiple clients in distributed scenarios.

In this paper, we propose MFL-Owner, a watermarking scheme for MFL models designed to verify global model ownership on the server and enable traceability in the event of model leakage. As shown in Fig. 1, we achieve watermark injection by utilizing linear transformations with orthogonal constraints, i.e., injecting a backdoor into the model through orthogonal transformation matrices added after the visual and text encoders (Cisse et al. 2017). We also introduce a client trigger set selection strategy using out-of-distribution data to ensure high differentiation among multiple backdoor trigger sets. MFL-Owner increases the discrepancy between the trigger dataset and the task dataset and enhances the difficulty of trigger dataset theft by applying Gaussian noise perturbations to the images in the trigger dataset. Our scheme also considers safeguarding the interests of the FL collective, so that a mechanism of tracing leakers is proposed. During verification and tracing, we use cosine similarity and Euclidean distance to assess differences in image-text pairs within the trigger set's embedding space, facilitating model ownership validation and leaker identification.

The main contributions are summarized as follows. (1) We propose the first general model ownership protection framework for MFL. This framework decouples the watermarking process from the FL training process and addresses ownership verification and traceability, effectively safeguarding the interests of the MFL collective. (2) We propose an orthogonal transformation-based backdoor watermarking method for distributed scenarios to improve the effectiveness and practicality of watermarking. The proposed scheme enables efficient watermark injections, ownership verification, and tracing with minimal impact on model performance. (3) Experiment evaluations have demonstrated that the proposed watermarking scheme achieves significant detection accuracy by using a small amount of out-of-distribution data as the trigger dataset, confirming its effectiveness and practicality.

Related Work

Multi-modal Federated Learning. With the increasing amount of multi-modal data, the challenge of training multi-modal models while protecting user privacy also becomes greater (Liu et al. 2020; Yu et al. 2023; Li et al. 2024). Xiong et al. (Xiong et al. 2022) extended the FedAvg (McMahan et al. 2017) aggregation from Horizontal Federated Learning (HFL) to multi-modal tasks. Succeeding work explored scenarios with modality heterogeneity among clients (Yu et al. 2023; Xiong et al. 2023; Poudel et al. 2024). Yu et al. (Yu et al. 2023) incorporated contrastive learning to develop intra-modality and inter-modality contrastive losses, improving multi-modal representation fusion. Even though prior studies demonstated that MFL was a crucial approach for harnessing the value of multi-modal data in distributed scenarios, the issue of model ownership protection within MFL had yet to be explored.

Watermarking Schemes. Current methods for watermarking deep neural networks (Li et al. 2022; Lv et al. 2023; Kuribayashi, Tanaka, and Funabiki 2020) can be classified into two categories based on the embedding technique: parameter-based watermarks and backdoor-based watermarks. *Parameter-based Watermark* (Xu et al. 2024; Uchida et al. 2017; Kuribayashi, Tanaka, and Funabiki 2020): The watermarking entity embeds information-carrying watermarks into the model parameters, where the watermark can be represented as an n-bit binary string. The watermark is integrated by adding a regularization term (Xu et al. 2024) related to the watermark in the loss function during training. Unlike black-box verification for backdoor-based watermarks, parameter-based watermarks necessitate white-box verification, which involves direct inspection of the model's parameters. Furthermore, embedding watermarks from multiple participants into the same model may result in watermark conflicts (Li et al. 2022). *Backdoor-based Watermark* (Li et al. 2022; Shao et al. 2024; Lv et al. 2023, 2022): The watermarking entity embeds specific trigger patterns into the deep neural network model using backdoor attacks. During verification, the model produces specific representation vectors for samples from the trigger dataset, enabling ownership verification. This method allows for verification

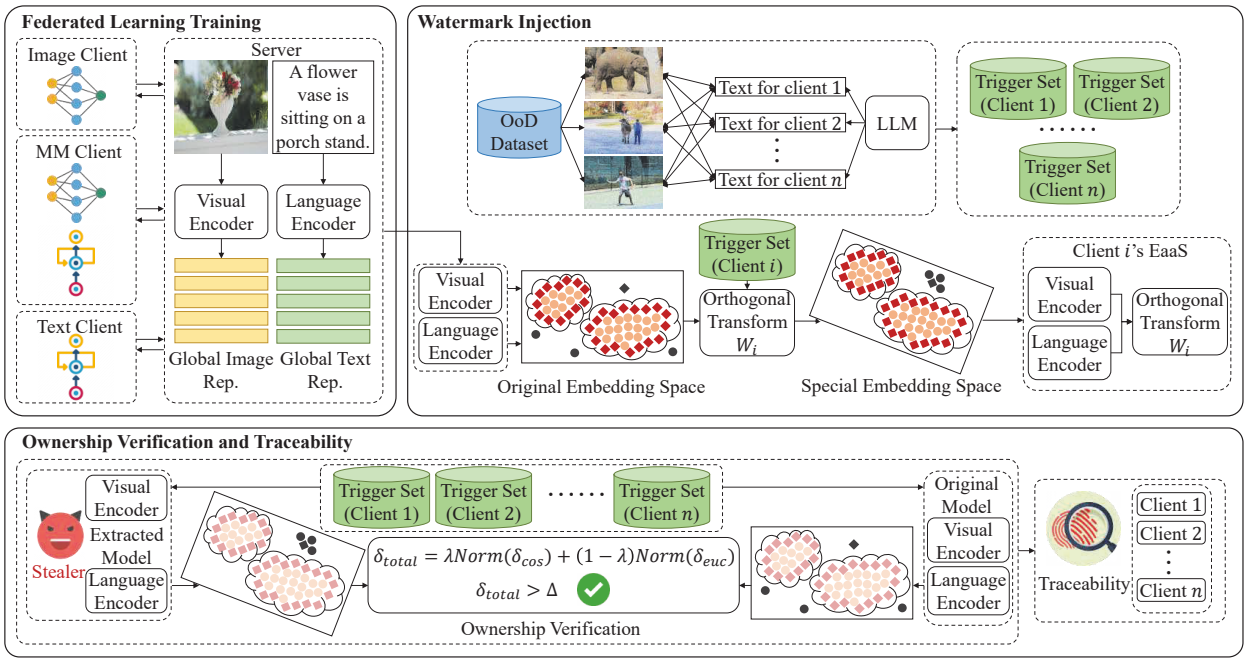


Figure 2: The framework of MFL-Owner.

via black-box access, thereby preserving the model's privacy (Li et al. 2022). Compared to parameter-based watermarking methods, backdoor-based watermarking methods are more practical due to black-box verification approach. However, their performance in MFL scenarios still requires validation.

Problem Formulation

Problem Definition

We define the global model obtained by the server after completing multi-modal FL training as Θ, which includes both an image encoder and a text encoder. We define the server's watermarking process as F, where the server individually embeds watermarks for each client, resulting in the model $\Theta_w = F(\Theta)$. The server uses the model Θ_w with the client's specific watermark embedded to deploy EaaS for the client. To protect the global model's ownership and safeguard the FL collective's interests, we focus on ownership verification and tracing in the event of a model leak.

Definition of Ownership Verification. The server verifies whether the objective model is stolen by querying it with image-text pairs. This verification process avoid revealing parameters of both the model under verification and the original model. Specifically, the model ownership verification process in MFL-Owner is detailed in Eq. (1).

$$\text{Verify}\left(\tilde{\Theta}, D_T\right) = \begin{cases} \text{True}, \tilde{\Theta} \in \mathcal{A}(\Theta) \\ \text{False, if otherwise} \end{cases}, \quad (1)$$

where $\tilde{\Theta}$ is the model to be validated, D_T is the trigger set, and $\mathcal{A}(\Theta)$ denotes the model stolen by the adversary.

Definition of Traceability. Traceability involves tracking the stolen model back to the malicious client within the FL. After confirming the ownership of a suspicious model, the FL group should implement a tracing mechanism to identify the model leaker. Specifically, the model tracing process in MFL-Owner is detailed in Eq. (2).

$$\text{Trace}\left(\tilde{\Theta}, D_T\right) = i, \tilde{\Theta} \in \mathcal{A}(\Theta), \quad (2)$$

where $\tilde{\Theta}$ is the model to be validated, D_T is the trigger set, $\mathcal{A}(\Theta)$ denotes the model stolen by the adversary, and i is the traceable model thief.

Threat Model

In MFL-Owner, we assume that the server is responsible for watermarking and is a trusted entity. The server acts as a verifier that conducts both model ownership verification and tracing. Clients are key participants and stakeholders in FL. We assume that some of clients are conquered by malicious adversaries, while others are honest entities.

Adversary's Goal. The malicious adversaries aim to acquire the jointly trained global model, then illegally distribute, replicate, and sell it, thereby undermining the interests of the FL collective.

Adversary's Capabilities. The adversary possesses a dataset D_c to query the client's EaaS and replicate the model through a theft attack. The adversary has sufficient EaaS query attempts to steal the model but cannot access the model's structure, training data. Specifically, the adversary can input image-text pairs from D_c and use the embedding values provided by EaaS as outputs to train a new model. The adversary then replicates and sells this trained model, evading ownership protection mechanisms.

Methodology

Overview of MFL-Owner

Our proposed MFL-Owner is a framework for global model ownership verification and tracing in FL for multi-modal data. The scheme constructs independent trigger datasets for each client, which are used to train a linear transformation layer incorporated into the original model (Cisse et al. 2017) . The linear transformation coverts the model's original embedding space into a special embedding space that contains trigger information. During the training of the linear transformation, we maintain the approximate orthogonality of the matrix to minimize its impact on the original embedding space. Models stolen by adversaries inherits the backdoor from the linear transformation layer, so that outputs including the trigger information in the special embedding space are created.

As illustrated in the Fig. 2, MFL-Owner comprises three stages, namely, FL training, watermark injection, ownership verification and traceability. (1) *FL Training.* In MFL training, multiple clients with different data modalities use the local data to optimize local models (Xiong et al. 2022). Model ownership protection and tracing offered by MFL-Owner are applicable to various MFL methods with desirable scalability, as watermarking is decoupled from model training (Yu et al. 2023; Liu et al. 2020; Xiong et al. 2023). (2) *Watermark Injection.* MFL-Owner embeds watermarks on the server side to ensure that the MFL training process remains unaffected, thus avoiding any additional computational and communication overhead for the clients. During this phase, the server embeds unique watermarks for each client to enable ownership verification and tracing. Upon completing model training and watermark injection, the server uses the watermark-embedded global model to deploy EaaS for the respective clients. (3) *Ownership Verification and Traceability.* In this phase, the verifier confirms model ownership by comparing the similarity differences in the embedding vectors of image-text pairs from the trigger dataset between the model under verification and the model without the orthogonal transformation layer. Next, the verifier traces the source of the model leak to hold the responsible party accountable and protect the interests of the MFL collective.

Watermark Injection via Orthogonal Transform

To minimize the impact of backdoor embedding on model performance, we use orthogonal transformation matrices for watermarking (Cisse et al. 2017; Tang et al. 2023). For traceability in the event of a model leakage, we select independent trigger datasets for each client. The orthogonal transformation matrices that are trained with each client's unique trigger dataset are then injected into the global model

Trigger Selection. To ensure the effectiveness of ownership verification and tracing, the trigger datasets shall meet following two conditions. (1) The initial similarity between image-text pairs in the trigger dataset shall be sufficiently low. (2) Each client's trigger dataset must have unique image-text pairs without duplication. We first analyze the frequency of object classes for the image-text pairs. The class set for the trigger dataset is constructed from randomly sampled high-frequency object classes. We then select m images from a small number of high-frequency object classes available online as the common images for each client's trigger dataset. We add Gaussian noise to the selected trigger image to blur the semantic information of the original image and reduce the probability of the attacker to speculate on the trigger image. For client k, we select a specific text s_k to pair with these m images, creating m unique image-text pairs $D_k = ((i_1, s_k), \ldots, (i_m, s_k))$. To avoid conflicts during tracing caused by similar watermarks, we ensure that the selected texts for each client have significant semantic differences. Specifically, we use generative Large Language Models (LLM) to generate trigger text with large semantic differences for clients. Finally, the server obtains the trigger sets for each client, where the image-text pairs contain the same images but distinct texts.

Watermark Injection. The impact on model performance is a crucial factor that limits the development of backdoor watermarks. To ensure the practicality and security of the watermark, MFL-Owner's watermarking needs to meet following two conditions, which are (1) minimal impact on the models's task performance and (2) high concealment. To achieve this, MFL-Owner employs the concept of orthogonal constraints to minimize the impact on the model's original performance (Tang et al. 2023). Specifically, the server first trains a linear transformation matrix with orthogonal constraints using each client's trigger dataset, aiming to minimize the effect on the original embedding space. Then, the server applies the linear transformation matrix to convert the original embedding space into a specialized space containing the watermark information.

Linear Transformation Matrix with Orthogonal Constraints. The server begins by initializing a random linear transformation $\mathcal{W}$. Let the m image embeddings from the image-text pairs in client i's trigger set be denoted as the image trigger embeddings $\mathcal{X} = \{\mathbf{x}_k\}_{k=1}^{m} \subseteq \mathbb{R}^{d \times m}$, and the m identical text embeddings as the text trigger embeddings $\mathcal{T} = \{\mathbf{t}_k\}_{k=1}^{m} \subseteq \mathbb{R}^{d \times m}$. The server then trains the linear transformation $\mathcal{W}$ using the image and text trigger sets, aligning the embeddings of these two sets into a shared space. The optimization process is detailed in Eq. (3).

$$\mathcal{W} = \underset{\mathcal{W} \in \mathbb{R}^{d \times d}}{\operatorname{argmin}} \|\mathcal{W}\mathcal{X} - \mathcal{W}\mathcal{T}\|_{\mathrm{F}}, \tag{3}$$

where d is the dimension of the embedding. To ensure the correlation of the embedding values for normal image-text pairs, aside from the trigger set, and to achieve an isometric transformation of the embedding values for normal image-text pairs in Euclidean space, we add orthogonal constraints during the training of the linear transformation matrix (Cisse et al. 2017). Specifically, we employ the alternating training method described in Eq. (4) to maintain the linear transformation matrix close to an orthogonal matrix.

$$\mathcal{W} \leftarrow (1 + \beta)\mathcal{W} - \beta\mathcal{W}\mathcal{W}^T\mathcal{W}, \tag{4}$$

where β is used to control the degree to which the matrix adheres to the orthogonal constraint. After multiple iterations,

we obtain the linear transformation matrix $\mathcal{W}$ with orthogonal constraints, which satisfies needs to preserve the original task performance during the watermarking process.

Embedding Space Transformation. The server computes the special embedding space E_s that contains the watermark information, as detailed in Eq. (5).

$$E_s = \mathcal{W}E_o = \{\mathcal{W}E_{o_i}, \mathcal{W}E_{o_t}\}, \tag{5}$$

where E_o denotes the original representation space, E_{o_i} denotes the image representation space, and E_{o_t} denotes the text representation space. Since the attacker uses the special embedding space E_s to perform model extraction attacks, the stolen model will inherit the backdoor of the linear transformation matrix, which can then be used for subsequent model ownership verification and tracing.

Ownership Verification and Traceability

When an adversary steals the model and offers EaaS to the public, the server can use the trigger dataset to verify model ownership, thereby ensuring copyright protection. Moreover, Model tracing is a crucial component of MFL-Owner. Once model ownership verification is complete, the server must trace the source of the model leak to safeguard the interests of the FL collective. We propose a method that improves the efficiency and practicality of verification by using similarity differences in the embedding vectors of image-text pairs to verify model ownership and trace leakers.

The server sequentially inputs the image-text pairs from each client's trigger dataset into both the original model without the linear transformation layer and the stolen model. Because the stolen model has learned the backdoor information from the linear transformation matrix, it produces similar embedding values for the image-text pairs in the trigger dataset. In contrast, the original model without the linear transformation matrix produces different embedding values for the image-text pairs in the trigger dataset. As a result, the similarity of embedding values for the image-text pairs in the trigger dataset differs significantly between the stolen model and the original model without the linear transformation layer. We use the p-value from hypothesis testing to measure the statistical difference in the distribution of image-text pairs in the trigger dataset between the different embedding spaces. Additionally, if the image-text pairs from a specific client's trigger set have higher similarity in the stolen model's embedding space compared to those from other clients, we conclude that the model is stolen from that client. Specifically, we calculate the cosine and Euclidean distances for each image-text pair in the trigger set within the embedding spaces of both the original model without the linear transformation layer and the stolen model.

$$d^i_{cos}(E) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{t}_i}{\|\mathbf{x}_i\| \, \|\mathbf{t}_i\|}, d^i_{euc}(E) = \left\| \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} - \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} \right\|, \tag{6}$$

where $E \in (E_o, E_s)$, E_o denotes the embedding spaces of the original model, and E_s denotes the embedding spaces of the stolen model. Then, the server calculates the mean distance of the image-text pairs in each embedding space for the trigger sets, as shown in Eq. (7).

$$\begin{aligned} \overline{d_{cos}}(E) &= \frac{1}{|C(E)|} \sum_{i \in D_k} d^i_{cos}(E), \\ \overline{d_{euc}}(E) &= \frac{1}{|L(E)|} \sum_{i \in D_k} d^i_{euc}(E), \end{aligned} \tag{7}$$

where $C(E) = \{d^i_{cos}(E) \| i \in D_k\}$, and $L(E) = \{d^i_{euc}(E) \| i \in D_k\}$. Finally, the server calculates the difference in the average distance of the trigger set image-text pairs between the embedding spaces of the stolen model and the original model without the linear transformation layer to determine the verification score, as shown in Eq. (8).

$$\delta_{total} = \lambda \cdot Norm(\delta_{cos}) + (1 - \lambda) \cdot Norm(\delta_{euc}), \tag{8}$$

where λ denotes the regulatory factor, $\delta_{cos} = \overline{d_{cos}}(E_o) - \overline{d_{cos}}(E_s)$, and $\delta_{euc} = \overline{d_{euc}}(E_o) - \overline{d_{euc}}(E_s)$. When the verification score δ_{total} exceeds the preset threshold, we conclude that the ownership of the original model has been compromised. By estimating the highest verification score as the source of the model leakage, we identify the client that is associated with the trigger set as the model leaker and complete the tracing process. Additionally, we assess the watermark detection capability by testing the detection rate of trigger samples. Specifically, the detection rate is the proportion of image-text pairs in the trigger set whose verification score surpasses the preset threshold.

Experiments

Experiment Configuration

Datasets. We conducted experiments on five datasets to explore the impact of our method on CLIP-based Vision-Language Pre-trained models in FL, including Flickr30k (Plummer et al. 2015), CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009), ImageNet-1k (Deng et al. 2009), and VOC2007 (Everingham et al. 2010). We tested the image-text retrieval performance of the watermarked model using the Flickr30k dataset and evaluated its multi-modal classification performance using the CIFAR-10, CIFAR-100, ImageNet-1K, and VOC2007 datasets. The images in the trigger set were sourced from the Visual Genome dataset (Krishna et al. 2017). The text in the trigger set was generated by LLM.

Implementation Details. We considered a MFL scenario with five or more clients. Each client collaboratively trained a global CLIP model stored on the server. The server deployed an EaaS for each client using models embedded with the respective client's watermark. Each client was associated with a trigger set. The trigger set consisted of 512 image-text pairs, where a specific client was associated with the same 512 unique texts. We analyzed the object class in the Visual Genome dataset and randomly selected three classes from the high-frequency classes as the target classes related to the images. Subsequently, 512 images were randomly selected from the Visual Genome dataset depending on the target classes. We added Gaussian noise to the images. We used Adam to train five orthogonal matrices $\mathcal{W}$ with a learning

Method	Dataset	Metric	Results (%)					$\overline{\Delta}$ (%)
			Client 1	Client 2	Client 3	Client 4	Client 5	
Original	Flickr30k	R@5	98.59/99.12	98.59/99.12	98.59/99.12	98.59/99.12	98.59/99.12	0.00/0.00
	CIFAR-10	ACC	95.59	95.59	95.59	95.59	95.59	0.00
	CIFAR-100	ACC	75.82	75.82	75.82	75.82	75.82	0.00
	ImageNet-1k	ACC	75.53	75.53	75.53	75.53	75.53	0.00
	VOC2007	ACC	78.29	78.29	78.29	78.29	78.29	0.00
EmbM	Flickr30k	R@5	85.47/66.39	86.48/67.55	86.37/67.38	87.08/68.09	86.41/67.75	-12.23/-31.69
	CIFAR-10	ACC	76.63	75.58	76.65	77.53	75.67	-19.18
	CIFAR-100	ACC	64.74	67.44	65.76	68.89	67.23	-9.01
	ImageNet-1k	ACC	68.05	67.21	65.02	67.48	68.10	-8.36
	VOC2007	ACC	42.90	41.09	43.53	44.18	43.44	-35.26
Fed-VLPM/o	Flickr30k	R@5	41.90/9.65	43.31/12.28	39.26/9.65	44.37/10.53	38.38/8.78	-57.15/-88.94
	CIFAR-10	ACC	86.76	88.36	83.13	86.18	88.29	-9.05
	CIFAR-100	ACC	38.87	37.56	34.83	45.46	45.50	-35.38
	ImageNet-1k	ACC	47.79	40.92	42.60	51.01	46.39	-29.79
	VOC2007	ACC	58.82	51.18	66.55	59.19	57.72	-19.60
Ours	Flickr30k	R@5	97.71/95.61	97.71/96.49	97.36/96.49	97.89/96.49	97.71/93.86	-0.91/-3.33
	CIFAR-10	ACC	95.68	95.70	95.73	95.64	95.72	-0.10
	CIFAR-100	ACC	73.17	72.83	73.05	72.79	73.23	-2.81
	ImageNet-1k	ACC	74.98	74.88	74.91	74.90	75.09	-0.58
	VOC2007	ACC	75.38	74.88	75.14	75.59	76.73	-2.75

Table 1: Performance of different methods across various downstream tasks. $\overline{\Delta}$ denotes the mean change in performance of the five client models relative to the Original. The value before "/" indicates the recall for image retrieval, while the value after "/" indicates the recall for text retrieval.

Method	No.	p-value	Dection Performance (%)		
			$\delta_{cos}/\delta_{euc}$	δ_{total}	DR
Fed-VLPM/o	1	$< 10^{-306}$	88.78/125.76	76.82	96.62
	2		87.70/125.49	76.65	96.61
	3		86.60/124.87	76.43	96.78
	4		90.45/127.20	77.21	96.77
	5		86.63/124.25	76.36	96.06
Ours	1	$< 10^{-306}$	63.39/62.25	65.71	96.84
	2		62.55/61.81	65.55	96.86
	3		62.99/63.17	65.77	97.33
	4		65.07/63.52	66.07	96.93
	5		61.95/61.64	65.45	97.60

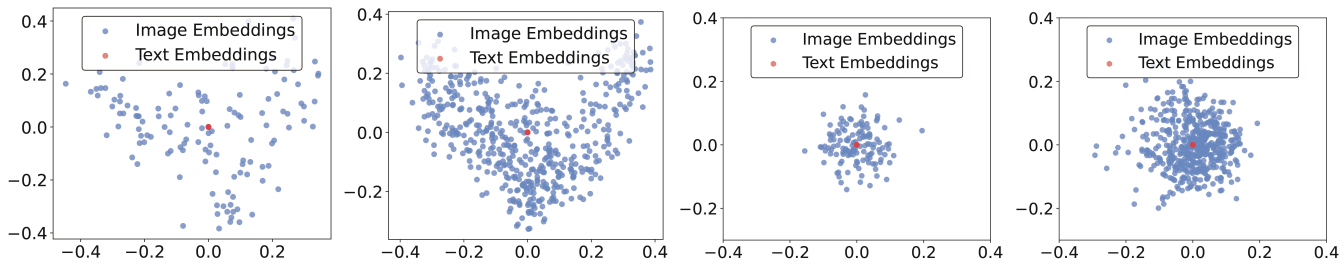
Table 2: Watermark detection performance of different methods across the five clients.

rate of 0.001 for 1000 training epochs on one RTX 4090 GPU. We set the regulatory factor λ to 0.5. All experiments used Python 3.9, PyTorch 2.2.2, and CUDA 11.8.

Baselines. Given the absence of watermarking schemes specifically tailored for MFL, we adapted existing multimodal watermarking methods to the FL to serve as our baselines. Specifically, (1) Original is a MFL method utilizing the CLIP model. In Original, multiple clients collaboratively train the CLIP model without incorporating any watermarks, making it suitable as a baseline for evaluating model performance. (2) EmbM (Peng et al. 2023) is a approach to backdoor in word embeddings. EmbM selects a set of moderate-frequency words from a general text corpus to form the trigger set, then selects a target embedding as the watermark and inserts it into the text embeddings containing the trigger words as a backdoor. (3) Fed-VLPM/o is derived by extending VLPMarker (Tang et al. 2023) to the FL and removing the orthogonal constraint. Fed-VLPM/o decouples the training process from the watermarking process.

Experiment Evaluation

Comparison with Baselines. MFL-Owner aimed to protect the model's copyright without compromising downstream task performance. We compared MFL-Owner with the baselines to investigate the impact of our proposed watermarking scheme on downstream tasks. Table 1 showed the performance of different methods across various downstream tasks. To ensure a fair evaluation, we assumed that the data across different clients was identical. Since the Original model lacked watermarks, all five clients achieved identical recall or accuracy on the same dataset. Table 1 showed that MFL-Owner had a more negligible impact on downstream tasks than EmbM and Fed-VLPM/o. EmbM showed a decrease in accuracy ranging from 8.36% to 35.26%, and Fed-VLPM/o showed a decrease in accuracy ranging from 9.05% to 35.38%, significantly higher than that of MFL-Owner. In multimodal classification, the average accuracy of the five clients using MFL-Owner decreased by a range of 0.10% to 2.81%, which was nearly identical to the performance of the Original. For multimodal retrieval, EmbM and Fed-VLPM/o experienced a substantial drop in recall, severely impacting the model's usability, whereas the



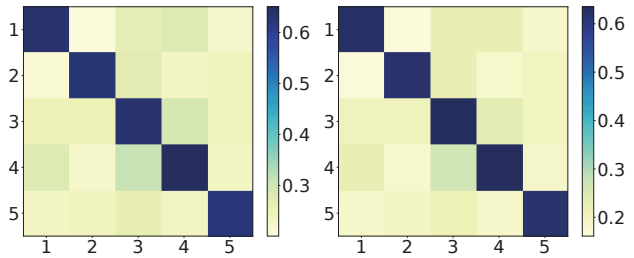
(a) Original (Trigger = 128) (b) Original (Trigger = 512) (c) MFL-Owner (Trigger = 128) (d) MFL-Owner (Trigger = 512)

Figure 3: Distributions of image-text pairs in trigger sets for Original and MFL-Owner under different numbers of triggers.

decline in recall for MFL-Owner remains within an acceptable range. The above experimental results indicated that incorporating orthogonal constraints during the training of $\mathcal{W}$ mitigated the negative impact of watermarking on the model's original performance.

Detection Performance. Table 2 showed the watermark detection performance of the five clients under different methods. Fig. 3 more intuitively illustrated the distribution differences between the original image-text embeddings and the watermarked image-text embeddings. After watermarking, images that were originally unrelated to the text clustered around it. Furthermore, the p-values for Fed-VLPM/o and MFL-Owner were both below 10^{-306}, indicating a statistically significant difference in distribution before and after watermarking. Comparing Fed-VLPM/o with MFL-Owner, we found that the cosine and Euclidean distances between the image-text embeddings of the trigger set obtained through the watermarked model in Fed-VLPM/o were closer, and the verification score δ_{total} was higher. This improvement occurred because Fed-VLPM/o only had loss (Eq. (3)) during the training of matrix $\mathcal{W}$, whereas MFL-Owner had additional constraints (Eq. (4)) beyond loss (Eq. (3)). Since the objectives of the two constraints differed, Fed-VLPM/o achieved better results in reducing the distances between trigger set image-text pairs after the same number of training epochs. However, Table 1 revealed that Fed-VLPM/o had a significant impact on downstream tasks. Regarding watermark detection rates (DR), Fed-VLPM/o and MFL-Owner achieved average DR of 96.57% and 97.11%, respectively. Considering both model usability and copyright protection, MFL-Owner demonstrated better overall performance.

Traceability. To validate the traceability of MFL-Owner, we input trigger data from different clients into the different watermarked models and calculated the variation difference of cosine and Euclidean distance before and after passing through the orthogonal transformation layer, as shown in Fig. 4. We defined $\delta_{cos}(c_i, c_j)$ as the change in cosine distance when the trigger set data from client c_j was input into the watermarked model of client c_i. $\delta_{cos}(c_i, c_j)$ corresponded to the cell in row i and column j of the heatmap in Fig. 4. $\delta_{euc}(c_i, c_j)$ had a similar meaning. We found that the image-text embedding distances were closer, that

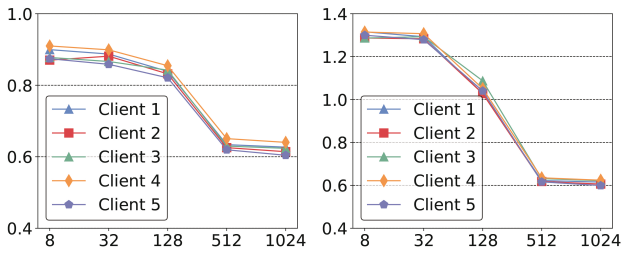


(a) δ_{cos} (b) δ_{euc}

Figure 4: Distance variation difference between the embeddings of image-text pairs in trigger sets output by watermarked models from different clients.

is, the image-text change difference before and after passing through the orthogonal transformation layer is greater, only when a client's watermarked model was fed with its own trigger set data, and farther when other clients' data was used. This showed that MFL-Owner can trace back to specific clients by distinguishing differences in cosine and Euclidean distances. The traceability of MFL-Owner was due to the significant semantic differences in the texts chosen by each client and the lack of correlation between these texts and the selected images. As a result, only the specific watermarked model of each client brought its own text and image embeddings closer together.

The Impact of the Trigger Number. We examined how the trigger number affects detection performance, setting trigger numbers to {8, 32, 128, 512, 1024}. Fig. 5 showed changes in cosine and Euclidean distances for different clients with varying trigger set sizes. Our method performed well with fewer trigger samples but showed decreased verification performance as the number of image-text pairs increased. When the trigger count was low (e.g., 8), the matrix $\mathcal{W}$ can more easily bring the distances between images and text closer. As the number of triggers growed, the number of images increased while the number of texts stayed constant, making it harder for the $\mathcal{W}$ to reduce these distances. Fig. 3 also indicated that smaller trigger sets result in closer distances between image-text pairs output by the watermark model. However, fewer triggers reduce the cost for



(a) δ_{cos} (b) δ_{euc}

Figure 5: Detection performance under different trigger numbers.

Method	Trigger Number			
	32	128	512	1024
Fed-VLPM/o	0.7056s	0.7095s	0.7186s	0.7307s
Ours	0.8298s	0.8313s	0.8507s	0.8923s

Table 3: Computation overhead of orthogonal transformation matrix training.

an adversary to obtain the set, increasing the risk to model copyright. Thus, choosing the right trigger number involves balancing watermark detectability with model copyright security. The appropriate number of trigger sets can be selected according to the specific requirements of the application scenario. Table 3 showed the computation overhead of one round of orthogonal transformation training with different trigger number. In the experiment configuration, both MFL-Owner and Fed-VLPM/o require training a 768×768 matrix. As shown in Table 3, we observed that the training time of MFL-Owner is marginally higher than that of Fed-VLPM/o, which is consistent with our expectations and within an acceptable range. This is due to the fact that, compared to Fed-VLPM/o, MFL-Owner introduces an orthogonal constraint step in each training iteration.

The Impact of the Client Number. We investigated the impact of the client number on downstream tasks. Table 4 showed the average performance of different client numbers across various downstream tasks. As shown in Table 4, with a total of 5, 10, 20, and 40 clients, the average performance of each client in downstream tasks (including multimodal retrieval and classification) showed no significant differences. This result aligned with our expectations, as in MFL-Owner, the watermarking for each client was independent, and changes in the number of clients did not affect the watermarking process. As a result, the number of clients had no noticeable effect on the model's downstream task performance.

Discussion and Future Work

MFL-Owner operates independently of the FL model training, making it compatible with various MFL paradigms with strong adaptability and scalability. In practical applications, our low-overhead watermarking approach effectively supports ownership protection and traceability in MFL.

Client Number	Dataset	Metric	Results(%)
5	Flickr30k	R@5	97.68/95.79
	CIFAR-10	ACC	95.69
	CIFAR-100	ACC	73.01
	ImageNet-1k	ACC	74.95
	VOC2007	ACC	75.54
10	Flickr30k	R@5	97.61/95.54
	CIFAR-10	ACC	95.67
	CIFAR-100	ACC	72.97
	ImageNet-1k	ACC	74.98
	VOC2007	ACC	76.22
20	Flickr30k	R@5	97.58/95.43
	CIFAR-10	ACC	95.69
	CIFAR-100	ACC	73.03
	ImageNet-1k	ACC	74.98
	VOC2007	ACC	75.97
40	Flickr30k	R@5	97.61/95.49
	CIFAR-10	ACC	95.61
	CIFAR-100	ACC	73.04
	ImageNet-1k	ACC	74.97
	VOC2007	ACC	75.52

Table 4: Average performance of different client number across various downstream tasks.

However, MFL-Owner has limitations in scenarios where the global model requires continuous updates. When the embedding space changes, the initial orthogonal transformation matrix may not accommodate the updated global model, as it depends on the original embedding values. Changes in the embedding space will reduce the effectiveness of watermarking. Future research will focus on designing a watermarking method with an orthogonal transformation matrix that adapts to updated embeddings, enhancing practicality.

Conclusion

In this paper, we proposed a general model ownership protection framework called MFL-Owner, which was designed to provide both model ownership protection and traceability while safeguarding the collective interests of participants in MFL. By decoupling the watermarking process from model training, MFL-Owner was highly adaptable to various MFL models and offered considerable practicality and scalability. To minimize the impact on downstream tasks, MFL-Owner employed orthogonal transformations for watermarking, preserving the original structure of transformations between image-text pair embeddings. Furthermore, the trigger dataset selection strategy employed by MFL-Owner, in conjunction with Gaussian noise perturbation, effectively addressed the challenge of multi-client verification conflicts. Experimental results demonstrated that MFL-Owner effectively verified ownership and traceability in MFL.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFF0905300, National Natural Science Foundation of China (Grant No.s U24B200674, 62372044), and Beijing Municipal Science and Technology Commission Project (Z241100009124008).

References

Chen, H.; Zhang, Y.; Krompass, D.; Gu, J.; and Tresp, V. 2024. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11285–11293. Vancouver, Canada.

Cisse, M.; Bojanowski, P.; Grave, E.; Dauphin, Y.; and Usunier, N. 2017. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, volume 70, 854–863. Sydney, NSW, Australia.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. Miami, Florida, USA.

Ding, Y.; Yu, J.; Liu, B.; Hu, Y.; Cui, M.; and Wu, Q. 2022. Mukea: Multimodal knowledge extraction and accumulation for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5089–5098. New Orleans, LA, USA.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338.

Feng, T.; Bose, D.; Zhang, T.; Hebbar, R.; Ramakrishna, A.; Gupta, R.; Zhang, M.; Avestimehr, S.; and Narayanan, S. 2023. Fedmultimodal: A benchmark for multimodal federated learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4035–4045. Long Beach, CA, USA.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1): 32–73.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Kuribayashi, M.; Tanaka, T.; and Funabiki, N. 2020. Deepwatermark: Embedding watermark into DNN model. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 1340–1346. Auckland, New Zealand.

Li, B.; Fan, L.; Gu, H.; Li, J.; and Yang, Q. 2022. FedIPR: Ownership verification for federated deep neural network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4521–4536.

Li, Z.; Hou, Z.; Liu, H.; Li, T.; Yang, C.; Wang, Y.; Shi, C.; Xie, L.; Zhang, W.; Xu, L.; et al. 2024. Federated Learning in Large Model Era: Vision-Language Model for Smart City Safety Operation Management. In *Companion Proceedings of the ACM on Web Conference 2024*, 1578–1585. Singapore, Singapore.

Liu, F.; Wu, X.; Ge, S.; Fan, W.; and Zou, Y. 2020. Federated learning for vision-and-language grounding problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11572–11579. New York, NY, USA.

Lv, P.; Li, P.; Zhu, S.; Zhang, S.; Chen, K.; Liang, R.; Yue, C.; Xiang, F.; Cai, Y.; Ma, H.; et al. 2022. Ssl-wm: A black-box watermarking approach for encoders pre-trained by self-supervised learning. *arXiv preprint arXiv:2209.03563*.

Lv, P.; Yue, C.; Liang, R.; Yang, Y.; Zhang, S.; Ma, H.; and Chen, K. 2023. A data-free backdoor injection approach in neural networks. In *32nd USENIX Security Symposium*, 2671–2688. Anaheim, CA, USA.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. Fort Lauderdale, FL, USA.

Peng, W.; Yi, J.; Wu, F.; Wu, S.; Zhu, B. B.; Lyu, L.; Jiao, B.; Xu, T.; Sun, G.; and Xie, X. 2023. Are You Copying My Model? Protecting the Copyright of Large Language Models for EaaS via Backdoor Watermark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 7653–7668. Toronto, Canada.

Plummer, B. A.; Wang, L.; Cervantes, C. M.; Caicedo, J. C.; Hockenmaier, J.; and Lazebnik, S. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, 2641–2649. Santiago, Chile.

Poudel, P.; Shrestha, P.; Amgain, S.; Shrestha, Y. R.; Gyawali, P.; and Bhattarai, B. 2024. CAR-MFL: Cross-Modal Augmentation by Retrieval for Multimodal Federated Learning with Missing Modalities. *arXiv preprint arXiv:2407.08648*.

Shao, S.; Yang, W.; Gu, H.; Qin, Z.; Fan, L.; and Yang, Q. 2024. FedTracker: Furnishing Ownership Verification and Traceability for Federated Learning Model. *IEEE Transactions on Dependable and Secure Computing*, pp(99): 1–18.

Sun, Y.; Liu, T.; Hu, P.; Liao, Q.; Fu, S.; Yu, N.; Guo, D.; Liu, Y.; and Liu, L. 2023. Deep intellectual property protection: A survey. *arXiv preprint arXiv:2304.14613*.

Tan, J.; Zhong, N.; Qian, Z.; Zhang, X.; and Li, S. 2023. Deep neural network watermarking against model extraction attack. In *Proceedings of the 31st ACM International Conference on Multimedia*, 1588–1597. Ottawa, ON, Canada.

Tang, Y.; Yu, J.; Gai, K.; Qu, X.; Hu, Y.; Xiong, G.; and Wu, Q. 2023. Watermarking Vision-Language Pre-trained Models for Multi-modal Embedding as a Service. *arXiv preprint arXiv:2311.05863*.

Tang, Y.; Yu, J.; Gai, K.; Zhuang, J.; Xiong, G.; Hu, Y.; and Wu, Q. 2024. Context-I2W: Mapping Images to Context-dependent Words for Accurate Zero-Shot Composed Image Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 5180–5188. Vancouver, Canada.

Uchida, Y.; Nagai, Y.; Sakazawa, S.; and Satoh, S. 2017. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, 269–277. Bucharest, Romania.

Xiong, B.; Yang, X.; Qi, F.; and Xu, C. 2022. A unified framework for multi-modal federated learning. *Neurocomputing*, 480: 110–118.

Xiong, B.; Yang, X.; Song, Y.; Wang, Y.; and Xu, C. 2023. Client-Adaptive Cross-Model Reconstruction Network for Modality-Incomplete Multimodal Federated Learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, 1241–1249. Ottawa, ON, Canada.

Xu, Y.; Tan, Y.; Zhang, C.; Chi, K.; Sun, P.; Yang, W.; Ren, J.; Jiang, H.; and Zhang, Y. 2024. RobWE: Robust Watermark Embedding for Personalized Federated Learning Model Ownership Protection. *arXiv preprint arXiv:2402.19054*.

Xue, M.; Zhang, Y.; Wang, J.; and Liu, W. 2021. Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations. *IEEE Transactions on Artificial Intelligence*, 3(6): 908–923.

Yu, J.; Zhang, W.; Lu, Y.; Qin, Z.; Hu, Y.; Tan, J.; and Wu, Q. 2020a. Reasoning on the relation: Enhancing visual representation for visual question answering and cross-modal retrieval. *IEEE Transactions on Multimedia*, 22(12): 3196–3209.

Yu, J.; Zhu, Z.; Wang, Y.; Zhang, W.; Hu, Y.; and Tan, J. 2020b. Cross-modal knowledge reasoning for knowledge-based visual question answering. *Pattern Recognition*, 108: 107563.

Yu, Q.; Liu, Y.; Wang, Y.; Xu, K.; and Liu, J. 2023. Multimodal Federated Learning via Contrastive Representation Ensemble. In *The Eleventh International Conference on Learning Representations*. Kigali, Rwanda.