Vision-guided Text Mining for Unsupervised Cross-modal Hashing with Community Similarity Quantization

Haozhi Fan[1], **Yuan Cao**[*2]

[1] School of Engineering and Applied Science, University of Pennsylvania, USA
[2] School of Computer Science and Technology, Ocean University of China, China
h3fan@seas.upenn.edu, cy8661@ouc.edu.cn

Abstract

Cross-modal retrieval, as an emerging field within multimedia research, has gained significant attention in recent years. Unsupervised cross-modal hashing methods are attractive due to their ability to capture latent relationships within the data without label supervision and to produce compact hash codes for high search efficiency. However, the text modality exhibits worse representation ability compared with the image modality, leading to weak guidance to construct the joint similarity matrix. Moreover, most unsupervised cross-modal hashing methods are based on pairwise similarities for training, resulting in non-aggregating data distribution in the hash space. In this paper, we propose a novel Vision-guided Text Mining for Unsupervised Cross-modal Hashing via Community Similarity Quantization, termed VTM-UCH. Specifically, we first find the one-to-one correspondence between each word and each vision (image or object) based on the Contrastive Language-Image Pre-training (CLIP) model and compute the text similarities according to the clustering of their corresponding visions. Then, we define the fine-grained object-level image similarities and design the joint similarity matrix based on the text and image similarities. Accordingly, we construct an undirected graph to compute the communities as the pseudo-centers and adjust the pairwise similarities to improve the hash codes distribution. The experimental results on two common datasets verify the accuracy improvements in comparison with state-of-the-art baselines.

Code — https://github.com/louisfanhz/VTMUCH

Introduction

Cross-modal retrieval refers to the task of searching for relevant instances to the query from data of different modalities. With the ever-growing size and complexity of data, Cross-modal Hashing (CMH) emerges as a promising solution to the retrieval task due to its superior computation and storage efficiency. CMH aims to map heterogeneous data into a joint embedding space i.e. Hamming space, essentially transforming the retrieval task to a nearest-neighbor search problem. Recent CMH methods show remarkable results by leveraging the representation power of deep neural networks to learn both the hash functions and modality-specific representations of data (Song et al. 2013; Liong et al. 2015; Do, Doan, and Cheung 2016; Cao et al. 2018; Wang et al. 2021).

Figure 1: An example to illustrate the ambiguity in image-text data. The captions below the images are the paired bag-of-words, the labels above the images are manually created.

However, current CMH paradigms often deploy deep learning models without addressing the inherent deficiency embedded within cross-modal datasets, obscuring the similarity relations between the data samples. In the image-text retrieval scenario, the images generally contain richer semantics and reflect clear correspondence to other similar images, but the paired captions supposed to describe the images are usually ambiguous or even unrelated. As shown in Figure 1, the two examples are obtained from MIRFLICKR-25K (Huiskes and Lew 2008). The first caption is ambiguous because its words are not related to the first image. By observing the captions of the two samples alone, it is unconvincing that the two samples are similar. In contrast, both images clearly contain very similar contents, which is in accordance with the semantics revealed by the labels. We believe this inconsistency results from the facts that the dataset generation involves significant amount of noisy web data (Desai et al. 2021) (Jia et al. 2021), and samples across modalities are not well-matched in their semantic meanings (Han et al. 2024). Therefore, directly encoding the data can jeopardize the model's performance in unexpected ways. As a result, compensating for the defect in text modality is crucial in designing an effective CMH scheme.

Moreover, the lack of labels limit most unsupervised CMH methods to use pairwise (Yang et al. 2017) (Wang

*Corresponding author

et al. 2021), triplets (Deng et al. 2018) or ranking losses (Ding et al. 2017) (Hu et al. 2023), which only maintain point to point relations, resulting in not fully-optimized distribution in the joint-embedding space. In contrast, the availability of label information in supervised settings enables an intuitive yet effective design of center-based loss that create clusters in the joint-embedding space (Hoe et al. 2021) (Tu et al. 2023d) (Huo et al. 2024). Center-based loss generate discriminative enough hash codes even for imbalanced data and can achieve high data distribution coverage compared to that of pairwise or triplet loss (Yuan et al. 2020), but it is difficult to generate centers in unsupervised settings to adjust hash codes distribution due to the lack of labels.

In this paper, we propose an unsupervised cross-modal hashing method termed Vision-guided Text Mining for Unsupervised Cross-modal Hashing (VTM-UCH). Firstly, we deploy an object detection module to identify objects in the images. These objects are then encoded by the pre-trained Vision Transformer (ViT) of the Contrastive Language-Image Pre-Training (CLIP) (Radford et al. 2021) model to obtain feature vectors. We also encode every word for each textual sample using the text transformer of CLIP to identify the words' correspondence with objects. Then, we classify both the images and the detected objects into clusters using K-means clustering (Lloyd 1982). By knowing whether the corresponding visions (images or objects) belong to the same cluster, we are able to obtain a text similarity matrix which contains richer and more accurate semantic information. Secondly, as for the image modality, with the detected objects, we develop a fine-grained pairwise similarity metric which helps to obtain a robust image similarity matrix. Thirdly, by fusing the text and image similarity matrices, we build an undirected graph, and apply Leiden algorithm (Traag, Waltman, and van Eck 2019) to identify non-overlapping communities in the graph, which can be used as pseudo-labels. Then, we adjust the similarity matrix in consideration of the pseudo-labels to refine hash codes distribution in Hamming space. Finally, pairwise loss in view of the final similarity matrix is used for training. Our main contributions are summarized as:

- We identify the need to explicitly address the heterogeneity gap and deficiency in cross-modal datasets, and purpose a novel text mining scheme to capture semantic similarity.
- We compensate unsupervised method with center-based strategy by utilizing community-detection algorithm, which helps adjusting hash codes distribution in the Hamming space.
- Experimental results on two common multi-modal datasets show superior accuracy performance of the proposed method compared with state-of-the-arts baselines.

Related Works

In this section, we summarize previous works that are related to our approach, mainly on supervised cross-modal hashing (Sun et al. 2023, 2024b,a) and unsupervised cross-modal hashing (Hu et al. 2020; Liu et al. 2020; Zhu et al. 2023; Zhong et al. 2023).

Supervised Cross-modal Hashing

Supervised CMH methods utilize label information to correlate encoded data samples in the Hamming space. The superior effectiveness of supervised methods is guaranteed by the dataset itself, which typically involves strict data alignment in the hash codes generation process. Recent supervised methods build on this advantage to develop effective paradigms. DCMHT (Tu et al. 2022) draws inspiration from Neural-Architecture-Search and uses a selective approach in learning hash bits. MITH (Liu et al. 2023) incorporates transformer models and distillation mechanism to consider similarities in different granularities. As noticed by the works above, only considering pairwise similarities does not ensure well-distributed embedding space. An intuitive yet effective idea in recent literatures is based on center or proxy losses, which encourages the model to learn optimized global data distribution. DAPH (Tu et al. 2023a) generates proxies using class labels and infuses the data modalities with the label information for hashing. DSPH (Huo et al. 2024) adopts similar strategy and generates proxies to cluster samples with the same labels, and simultaneously to pull away the irrelevant ones. However, the supervised nature of the above methods makes them rely on the labels which are costly. Furthermore, the powerful center-based mechanism cannot be adapted to unsupervised settings.

Unsupervised Cross-modal Hashing

Different from supervised methods, unsupervised CMH methods do not rely on label information, but often need to devise mechanisms to extract the similarity relations from data. Unsupervised methods have received considerable attention for their practicability in real-world applications. Many recent works rely on constructing similarity matrix to encode pairwise relations for model learning. UDCMH (Wu et al. 2018) purposes matrix-factorization with binary latent factor models to generated hash codes. DJSRH (Su, Zhong, and Zhang 2019) constructs a joint similarity matrix from the neighborhood structure of data samples for hash codes learning. DGCPN (Yu et al. 2021) utilizes graph-based methods to extract the structural and hierarchical similarity between samples. DAEH (Shi et al. 2022) constructs similarity matrix based upon a distance-based metric to discriminatively learn hash functions for different modalities. CIRH (Zhu et al. 2023) uses graph-based network to extract similarity relations across modalities and builds an encode-decoder structure to preserve similarity within modalities. However, these methods do not address the deficiency in datasets, especially for the text modality. UCHSTM (Tu et al. 2023b) mines the semantic similarity information from text modality to support constructing a robust similarity matrix to learn hash functions. Nevertheless, UCHSTM only focuses on extracting semantics from texts, and does not explicitly correlate text and image modalities.

Methodology

Architecture

Figure 2 shows the framework of the proposed VTM-UCH method which contains three modules to pre-process train-

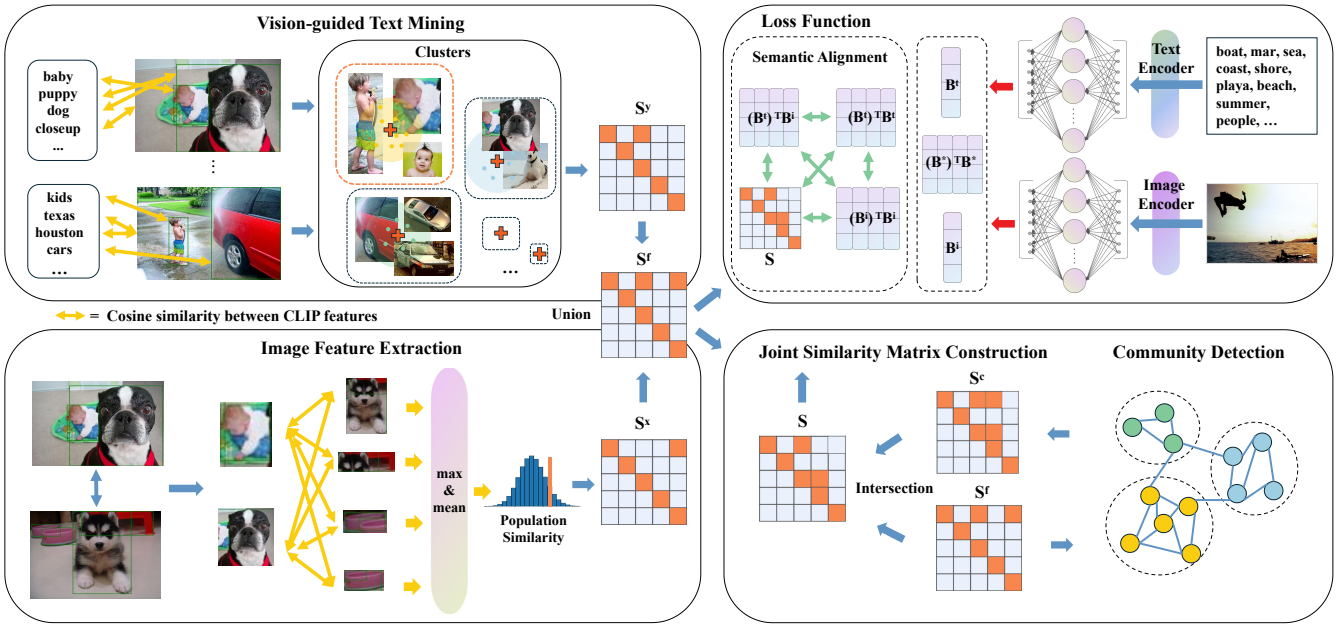

Figure 2: The framework of the proposed VTM-UCH model. The proposed model is composed of three pre-processing modules. The vision-guided text mining module finds one-to-one correspondence between word and vision (object or image), and computes text similarities by vision clustering. The image feature extraction module computes fine-grained mean image similarities between the max matching objects. The community detection module assigns each instance to a non-overlapping community, producing several pseudo-labels. The final similarity matrix is constructed according to the above three models, and used in pairwise loss for training.

ing data for text mining, feature extraction, and community detection. In the text mining module, we first conduct object detection with pre-trained FasterR-CNN module (Ren et al. 2017) on the images. Then, we match each word with its most similar vision (object or image) by their CLIP features (Radford et al. 2021). Next, all the visions are clustered by k-means clustering (Lloyd 1982). Accordingly, text similarity matrix is constructed by whether their corresponding visions appear in one common cluster. In the image feature extraction model, we utilize the objects' CLIP features to calculate fine-grained image similarity matrix by max-min population similarity. By fusing the text and image similarity matrices, we construct an undirected graph, and generate communities in the community detection module. In the end, we adjust the final similarity matrix to design a pairwise loss for training the whole framework.

Notations and Problem Formulation

Given a training dataset $\mathcal{D}$ containing N instances, where each instance $(\mathbf{x}_i, \mathbf{y}_i)$ denotes a pair of image $\mathbf{x}_i$ and text $\mathbf{y}_i$. We utilize the pre-trained CLIP image and text encoders to extract modality-specific feature vectors $\mathbf{x}_i \in \mathcal{R}^d$ and $\mathbf{y}_i \in \mathcal{R}^d$, where d denotes the dimensionality of the CLIP features. Additionally, we define $\mathbf{x}_{i,a} \in \mathcal{R}^d$ as the feature vector of the a-th detected object in image $\mathbf{x}_i$, where the objects are encoded by pre-trained CLIP image encoder. We also define $\mathbf{y}_{i,a} \in \mathcal{R}^d$ as the feature vector of the a-th word in text $\mathbf{y}_i$, where the words are encoded by pre-trained CLIP text encoder. In most cases, the numbers of objects and words in an instance $(\mathbf{x}_i, \mathbf{y}_i)$ are not the same.

Our goal is to learn hash functions to project the high-dimensional image and text data to a common discrete Hamming space, where the Hamming distances between similar data points are small. We denote the hash functions as $f(\cdot;\theta^x)$ and $f(\cdot;\theta^y)$ for image and text modalities, respectively. They take the image and text feature vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ as input to obtain their continuous hash codes $\mathbf{h}_i^x = f(\mathbf{x}_i;\theta^x) \in (-1,1)^K$ and $\mathbf{h}_i^y = f(\mathbf{y}_i;\theta^y) \in (-1,1)^K$, where θ^x and θ^y denote trainable parameters in two modalities, and K denotes the number of hash bits. We use a tanh activation over the logits to ensure that the hashing bits are mapped between -1 and 1. Then, the discrete binary codes for $\mathbf{x}_i, \mathbf{y}_i$ can be easily computed as $\mathbf{b}_i^x = \mathbf{sgn}(\mathbf{h}_i^x)$ and $\mathbf{b}_i^y = \mathbf{sgn}(\mathbf{h}_i^y)$. All the hash codes generated from the training dataset in two modalities are denoted as $\mathbf{B}^x = [\mathbf{b}_1^x, ..., \mathbf{b}_N^x] \in \{-1,1\}^{K\times N}$, and $\mathbf{B}^y = [\mathbf{b}_1^y, ..., \mathbf{b}_N^y] \in \{-1,1\}^{K\times N}$, respectively. In addition, the image and text similarity matrices are denoted as $\mathbf{S^x} \in \{-1,1\}^{N\times N}$ and $\mathbf{S^y} \in \{-1,1\}^{N\times N}$, respectively.

We use boldface lowercase letters (e.g., $\mathbf{x}$) to denote vectors and boldface uppercase letters (e.g., $\mathbf{X}$) to denote matrices.

Vision-guided Text Mining

As shown in Figure 2, we first compute the similarities between words and visions (objects and images) in every input text-image pair, where objects are obtained from a pre-trained object detection module and words are simply the split of text input. We adopt FasterR-CNN (Ren et al. 2017) trained on the Open Images Dataset (Kuznetsova et al. 2020) for object detection, with a hyperparameter τ as the detection-score threshold. As the number of words varies greatly across data instances and across datasets, we limit the number of words present in each text by a hyperparameter N_w. Since the vision and word feature vectors are of the same dimensionality produced by pre-trained CLIP (Radford et al. 2021), we can compute their cosine similarities for one-to-one corresponding matching. As for the word $\mathbf{y}_{i,a}$, we identify its most similar object as

$$\tilde{\mathbf{x}}_{i,a} = \underset{\mathbf{x}_{i,*}}{\operatorname{argmax}} \cos(\mathbf{y}_{i,a}, \mathbf{x}_{i,*}) \tag{1}$$

$$= \underset{\mathbf{x}_{i,*}}{\operatorname{argmax}} \frac{\mathbf{y}_{i,a} \cdot \mathbf{x}_{i,*}}{\|\mathbf{y}_{i,a}\| \|\mathbf{x}_{i,*}\|} \in \mathcal{R}^d \tag{2}$$

where $*$ is an arbitrary detected object in $\mathbf{x}_i$. As a result, suppose there are $N_{w,i}$ words in text $\mathbf{y}_i$, the vision-guided features for text $\mathbf{y}_i$ is denoted as:

$$\mathbf{Y}_i = \{\tilde{\mathbf{x}}_{i,1}; ...; \tilde{\mathbf{x}}_{i,n_{w,i}}\} \in \mathcal{R}^{d \times n_{w,i}} \tag{3}$$

In order to quantify similarity relations between instances, we cluster detected objects from all the training images along with all the training images based on their distances in the CLIP feature space by using K-Means clustering (Lloyd 1982). Given the number of clusters N_c, K-Means's objective is to find a set of centroids $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_{N_c}\}$ that satisfy

$$\begin{aligned} \mathbf{C} &= \underset{\mathbf{C}}{\operatorname{argmin}} \sum_{k=1}^{N_c} \sum_{\mathbf{x}_{i,a} \in C_k} \|\mathbf{x}_{i,a} - \mu_k\|^2 \\ \mu_k &= \frac{1}{|C_k|} \sum_{\mathbf{x}_{i,a} \in C_k} \mathbf{x}_{i,a} \end{aligned} \tag{4}$$

We define a function l_S that assigns a unique label to each detected object (we just describe objects for convenience) based on its closest centroid:

$$l_S(x_{i,a}) = \underset{k}{\operatorname{argmin}} \|\mathbf{x}_{i,a} - \mu_k\| \tag{5}$$

Then, the set of labels for the vision-guided text $\mathbf{y}_i$ is denoted as

$$L_S(\mathbf{y}_i) = \{l_S(\tilde{\mathbf{x}}_{i,1}), ..., l_S(\tilde{\mathbf{x}}_{i,n_{w,i}})\} \tag{6}$$

Finally, the text similarity matrix $\mathbf{S}^y \in \{-1, 1\}^{N \times N}$ can be constructed using the above-defined labels. In particular, if two text instances share at least one common label, we consider them similar, then the corresponding entry of $\mathbf{S}^y$ is defined as 1:

$$\mathbf{S}^y_{ij} = \begin{cases} 1, & \text{if } L_S(\mathbf{Y}_i) \cap L_S(\mathbf{Y}_j) \neq \emptyset \\ -1, & \text{otherwise} \end{cases} \tag{7}$$

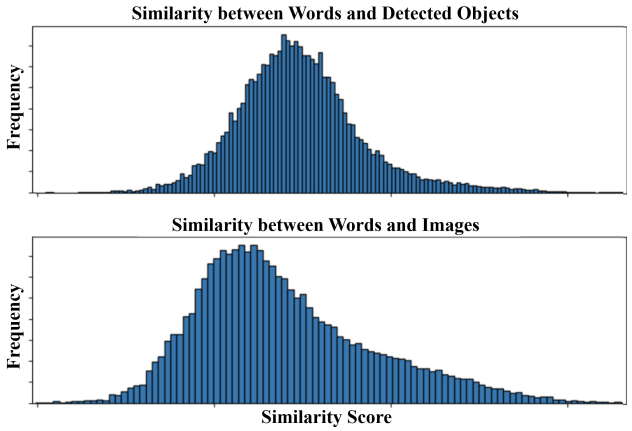

Figure 3: The bottom diagram shows the frequencies of cosine similarities between words and original images, the upper diagram shows the same statistics between words and detected objects.

Note that the labels identified by object detection model are irrelevant to our model, because injecting the information of labels from object detection model may pollute the underlying feature space. we introduce clustering to utilize the semantic-rich information contained in the pre-trained CLIP encoder. By using visual information to supplement text, the model pays more attention to the words corresponding visions essentially, and thus introduces less noise inherent in text modality. We show experimentally that incorporating object detection is effective in extracting the semantic information from CLIP embedding space.

Here, we explain why we utilize both images and detected objects to match the word feature vectors. As shown in Fig. 3, we report the statistics of the cosine similarity between words and detected objects, and between words and the original images. The former has a mean of 0.221 and standard deviation of 0.0108, while the latter has a mean of 0.210 and standard deviation of 0.0113. Obviously, the similarities between words and detected objects present a better similarity distribution, which can enhance the text semantic mining. In the experimental part, we also implement ablation studies to verify this conclusion.

Image Feature Extraction

The image feature extraction module aims to extract fine-grained semantic information from each image using its detected objects. For every pair of images $\mathbf{x}_i$ and $\mathbf{x}_j$ in the dataset, we compute the pairwise cosine similarity between their detected objects as

$$\cos(\mathbf{x}_{i,a}, \mathbf{x}_{j,b}) = \frac{\mathbf{x}_{i,a} \cdot \mathbf{x}_{j,b}}{\|\mathbf{x}_{i,a}\| \|\mathbf{x}_{j,b}\|} \in [-1, 1] \tag{8}$$

For an object $\mathbf{x}_{i,a}$ detected in $\mathbf{x}_i$ and an object $\mathbf{x}_{j,b}$ detected in x_j. We then obtain the similarity between image x_i and x_j by averaging the sum of the max cosine similarities between

Task	Method	MIRFLICKR-25K				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
I → T	CVH	0.600	0.599	0.596	0.598	0.372	0.362	0.406	0.390
	IMH	0.612	0.601	0.592	0.579	0.470	0.473	0.476	0.459
	CMFH	0.621	0.624	0.625	0.627	0.455	0.459	0.465	0.467
	FSH	0.613	0.630	0.658	0.670	0.506	0.508	0.517	0.524
	RFDH	0.639	0.632	0.642	0.643	0.451	0.477	0.489	0.496
	DJSRH	0.641	0.655	0.677	0.682	0.513	0.514	0.534	0.554
	JDSH	0.671	0.681	0.687	0.697	0.552	0.561	0.590	0.588
	DSAH	0.686	0.693	0.701	0.712	0.561	0.587	0.598	0.599
	UKD-SS	0.693	0.692	0.706	0.714	0.542	0.577	0.603	0.625
	DAEH	0.701	0.698	0.705	0.715	0.581	0.580	0.609	0.623
	CIRH	0.699	0.703	0.715	0.728	0.573	0.601	0.610	0.621
	UCHSTM	0.731	0.737	0.742	0.751	**0.623**	0.632	0.637	**0.649**
	VTM-UCH	**0.743**	**0.754**	**0.751**	**0.754**	0.616	**0.633**	**0.639**	0.631
T → I	CVH	0.591	0.583	0.576	0.576	0.401	0.384	0.442	0.432
	IMH	0.603	0.595	0.589	0.580	0.478	0.483	0.472	0.462
	CMFH	0.642	0.662	0.676	0.685	0.529	0.577	0.614	0.645
	FSH	0.615	0.635	0.663	0.672	0.479	0.481	0.497	0.513
	RFDH	0.622	0.627	0.630	0.626	0.436	0.410	0.442	0.458
	DJSRH	0.646	0.656	0.668	0.679	0.528	0.530	0.529	0.554
	JDSH	0.674	0.675	0.682	0.695	0.549	0.550	0.573	0.581
	DSAH	0.683	0.696	0.708	0.715	0.576	0.589	0.614	0.633
	UKD-SS	0.702	0.697	0.705	0.713	0.567	0.594	0.613	0.621
	DAEH	0.690	0.686	0.696	0.709	0.582	0.578	0.605	0.613
	CIRH	0.699	0.709	0.721	0.734	0.577	0.605	0.615	0.626
	UCHSTM	0.733	0.745	0.745	0.735	0.611	0.624	0.635	**0.658**
	VTM-UCH	**0.751**	**0.758**	**0.760**	**0.761**	**0.635**	**0.649**	**0.651**	0.645

Table 1: The mAP results of VTM-UCH and baselines with 16 to 128 hash bits on MIRFLICKR-25K and NUS-WIDE.

each object pairs:

$$\begin{aligned}\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{n_{o,i} + n_{o,j}} (\sum_a \max_b \cos(\mathbf{x}_{i,a}, \mathbf{x}_{j,b}) \\ + \sum_b \max_a \cos(\mathbf{x}_{i,a}, \mathbf{x}_{j,b}))\end{aligned} \tag{9}$$

where $n_{o,i}$ and $n_{o,j}$ denote the numbers of objects detected in images x_i and x_j, respectively. Since the computing complexity of computing similarities between all the detected objects is $O(N^2)$, which becomes expensive when the training size is large. In practice, we limit the number of detected objects in an image to at most $0.5 \max(n_{o,*})$, where $\max(n_{o,*})$ denotes the maximum number of objects detected in all the images.

The image similarity matrix $\mathbf{S}^x \in \{-1, 1\}^{N \times N}$ is then calculated as

$$\mathbf{S}^x_{ij} = \begin{cases} 1, & \text{if } \mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j) > \mu_x + \eta\sigma_x \\ -1, & \text{otherwise} \end{cases} \tag{10}$$

where μ_x and σ_x denote the population mean and standard deviation of the cosine similarities of all the image pairs in the training set, η denotes a hyperparameter to allow for tuning the sparsity of the image similarity matrix.

Community Detection

By fusing the image and text similarity matrices $\mathbf{S}^x$ and $\mathbf{S}^y$, we construct a fused similarity matrix $\mathbf{S}^f \in \{-1, 1\}^{N \times N}$ as

$$\mathbf{S}^f_{ij} = \begin{cases} 1, & if\ \mathbf{S}^x_{ij} = 1 \ \mathbf{or}\ \mathbf{S}^y_{ij} = 1 \\ -1, & \text{otherwise.} \end{cases} \tag{11}$$

The fused similarity matrix S^f captures the pairwise similarity relations for all data instances. However, since the fused similarity matrix is discrete and binary, it implicitly assumes that two instances should either be very close or very far in Hamming space, which is inappropriate in an unsupervised setting because not all similar pairs will be correctly identified.

Therefore, we aim to declare two samples similar/dissimilar only when there is strong evidence. For this purpose, we incorporate community-detection algorithm which identify communities from similarity matrix such that samples within their corresponding communities are well-connected. In other words, if $\mathbf{S}^f$ suggests two samples are similar, we want to confirm they also exist in the same community. By injecting structural correlations, we ensure a better exploitation of the embedding space of the encoder model.

We adopt Leiden algorithm (Traag, Waltman, and van Eck 2019), a community detection algorithm aiming to optimize the modularity $\mathcal{H}$:

$$\mathcal{H} = \frac{1}{2m} \sum_c (e_c - \gamma \frac{K_c^2}{2m}) \tag{12}$$

where K_c denotes the sum of nodes degrees in community c, m denotes the total number of edges in a graph, e_c denotes the number of edges in community c, and γ denotes a resolution parameter. Intuitively, Leiden algorithm aims to maximize the difference between the expected number of edges $K_c^2/(2m)$ and the actual number of edges e_c in a community. Mention that we densify the communities for better centroid distribution.

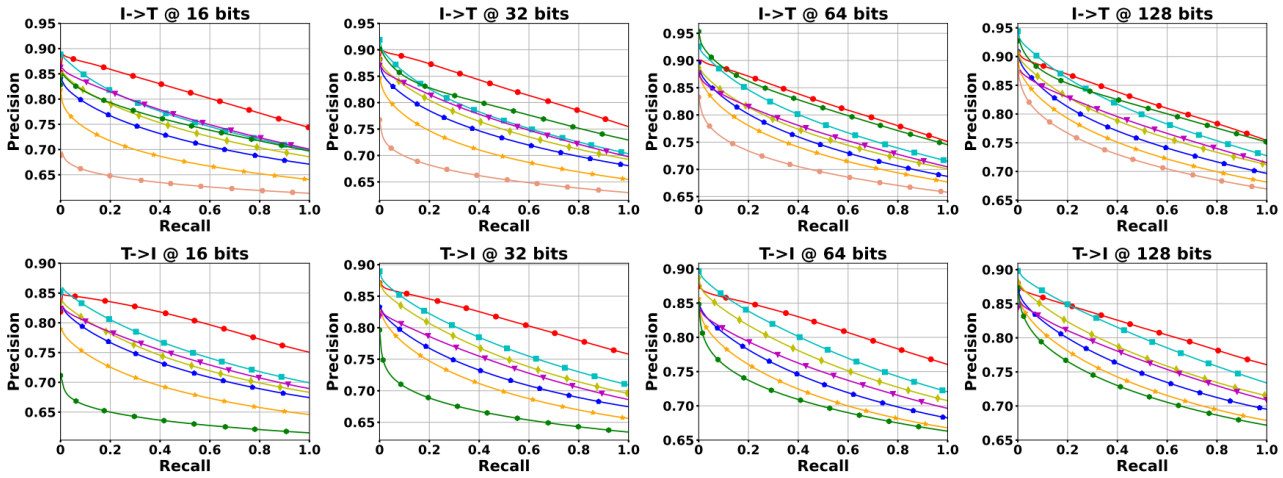

Figure 4: The precision-recall curves with 16 to 128 hash bits on MIRFLICKR-25k.

Considering $\mathbf{S}^f$ as an adjacency matrix such that $\mathbf{S}^f_{ij} = 1$ indicates an edge between nodes s^f_i and s^f_j, we use Leiden algorithm to identify partition $\mathcal{P} = \{Comm_1, Comm_2, ...\}$ that contains non-overlapping communities. We define a function l_C that assign each node a unique community label according to which community it belongs to:

$$l_C(s^f_i) = k \quad s.t. \quad s^f_i \in Comm_k \tag{13}$$

We then construct the community similarity matrix S^c as

$$\mathbf{S}^c_{ij} = \begin{cases} 1, & if\ l_C(s^f_i) = l_C(s^f_j) \\ -1, & \text{otherwise.} \end{cases} \tag{14}$$

Joint Similarity Matrix Construction

In the end, the community similarity matrix can then be used to refine the fused similarity matrix $\mathbf{S}^f$ such that the resulting matrix identify confident similar/dissimilar pairs of samples:

$$\mathbf{S}_{ij} = \begin{cases} 1, & \text{if } \mathbf{S}^f_{ij} = 1 \text{ and } \mathbf{S}^c_{ij} = 1 \\ -1, & \text{if } \mathbf{S}^f_{ij} = -1 \text{ and } \mathbf{S}^c_{ij} = -1 \\ \mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j), & \text{otherwise} \end{cases} \tag{15}$$

where $\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j) \in [-1, 1]$ denotes continuous image similarities computed by image CLIP features. For pairs of samples that are not confidently enough to assign similarity relations, we choose to directly use $\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)$ to measure similarities between the instances.

Loss Function

The similarity matrix $\mathbf{S}$ is used as a target measure to learn the networks that parameterize hashing functions. In particular, we minimize the reconstruction error between $S^{f'}$ and the inner product between hash codes:

$$\begin{aligned} \mathcal{L}_r = \rho\|(\mathbf{B}^x)^{\mathrm{T}}\mathbf{B}^x - K\mathbf{S}\|^2_F \\ +(1-\rho)\|(\mathbf{B}^y)^{\mathrm{T}}\mathbf{B}^y - K\mathbf{S}\|^2_F \\ +\|(\mathbf{B}^x)^{\mathrm{T}}\mathbf{B}^y - K\mathbf{S}\|^2_F \end{aligned} \tag{16}$$

where ρ is the trade-off parameter to balance the intra-modal reconstruction error.

we also add the semantic alignment loss to preserve the intra-modal and inter-modal similarities of the same data pair instances:

$$\begin{aligned} \mathcal{L}_s = \|(\mathbf{B}^x)^{\mathrm{T}}\mathbf{B}^x - (\mathbf{B}^y)^{\mathrm{T}}\mathbf{B}^y\|^2_F \\ +\|(\mathbf{B}^x)^{\mathrm{T}}\mathbf{B}^y - (\mathbf{B}^x)^{\mathrm{T}}\mathbf{B}^x\|^2_F \\ +\|(\mathbf{B}^x)^{\mathrm{T}}\mathbf{B}^y - (\mathbf{B}^y)^{\mathrm{T}}\mathbf{B}^y\|^2_F \end{aligned} \tag{17}$$

The overall loss function is

$$\mathcal{L} = \mathcal{L}_r + \alpha\mathcal{L}_s \tag{18}$$

where α denotes a hyperparameter to tune the contribution of the semantic alignment loss.

Experiments

In this section, we summarize the experiments performed on two commonly used datasets for CMH: MIRFLICKR-25K (Huiskes and Lew 2008) and NUS-WIDE (Chua et al. 2009).

Datasets and Evaluation Metrics

MIRFLICKR-25K is a widely used dataset that contains 25,000 pairs of images and captions, where each image is annotated by 24 tags. We follow the experimental setup of (Su, Zhong, and Zhang 2019) to process the dataset, selecting 20,015 pairs of images and captions for our experiments. Each of the selected image may be of different size and each caption is a list of words selected from a vocabulary containing 1,386 words. For MIRFLICKR-25K, we randomly sample 2,000 pairs of images and captions as the query set, and sample 5,000 pairs from the remaining data as the training set. All data not in the query set is used as retrieval set.

NUS-WIDE dataset contains 269,649 pairs of images and captions, where each image is annotated by 81 tags. We select 190,421 image-text pairs that belong to the 21 most frequently used tags as our dataset, out of which 2,000 pairs are randomly sampled as the query set, 5,000 pairs are sampled

Variant	I →T				T → I			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
VTM-UCH-Nocomm	0.740	0.752	0.750	0.744	0.745	0.756	**0.760**	0.758
VTM-UCH-Comm	0.704	0.721	0.726	0.726	0.702	0.723	0.724	0.727
VTM-UCH-Img	0.724	0.733	0.735	0.739	0.734	0.736	0.745	0.749
VTM-UCH-COCO	0.724	0.731	0.730	0.731	0.728	0.732	0.733	0.738
VTM-UCH	**0.743**	**0.754**	**0.751**	**0.754**	**0.751**	**0.758**	**0.760**	**0.761**

Table 2: The mAP scores of different variants of VTM-UCH on MIRFLICKR-25K.

from the remaining data as the training set. All data not in the query set is used as retrieval set.

To evaluate the performance of our model, we calculate Mean Average Precision (mAP) scores, as well as Precision and Recall for all the queries based on ranking Hamming distances between query hash code and retrieved hash codes.

Baselines

We compare our VTM-UCH model with 13 state-of-the-art baselines, including 4 shallow methods: Cross-view Hashing (CVH) (Kumar and Udupa 2011), Inter-media Hashing (IMH) (Song et al. 2013), Collective Matrix Factorization Hashing (CMFH) (Ding, Guo, and Zhou 2014), Fusion Similarity Hashing (FSH) (Liu et al. 2017), and 9 deep methods: Robust and Flexible Discrete Hashing (RFDH) (Wang, Wang, and Gao 2018), Deep Joint-Semantics Reconstructing Hashing (DJSRH) (Su, Zhong, and Zhang 2019), Joint-modal Distribution-based Similarity Hashing (JDSH) (Liu et al. 2020), Deep Semantic-alignment Hashing (DSAH) (Yang et al. 2020), Unsupervised Knowledge Distillation for Cross-modal Hashing (UKD-SS) (Hu et al. 2020), Deep Adaptively-enhanced Hashing (DAEH) (Shi et al. 2022), Correlation-identity Reconstruction Hashing (CIRH) (Zhu et al. 2023), UCHSTM (Tu et al. 2023c), and HEH (Zhong et al. 2023).

Experimental setup

For the training task, the input image is resized to 224×224, and the input text is split into individual words, each as a separate input. We use the pre-trained CLIP model as the backbone during initialization. While fine-tuning CLIP backbone, we only fine-tune the text encoder, but freeze the parameters of the image encoder. The output of the last layer is a 512-dimensional vector for both the image encoder and text encoder. Fully connected layers followed by a tanh activation are used to learn hashing functions. We provide the parameters configuration used for the two datasets here. As for MIRFLICKR-25K, $N_c = 14$, as for NUS-WIDE, $N_c = 22$. All the other parameters are the same across the two datasets. $\tau = 0.3$, $\rho = 0.1$, $N_w = 64$, $\alpha = 1$.

Results and Analyses

Table 1 shows the mAP scores of the proposed VTM-UCH and several state-of-the-art baselines. We find that our method generally outperfroms the baselines, especially on MIRFLICKR-25K. Figure 4 shows the Precision-Recall curves of all the methods, which exhibits similar results with Table 1. We also configure four variants of VTM-UCH to analyze the influence of each component in the model architecture. VTM-UCH-Nocomm denotes the variant that does not incorporate community detection, using the fused similarity matrix (11) directly in loss function (16). VTM-UCH-Comm denotes the variant that uses similarity matrix constructed from community labels, i.e. the matrix (14), in loss function (16). VTM-UCH-Img denotes the variant that only using images to match words in (2), instead of using both the original images and their detected objects. VTM-UCH-COCO denotes the variant that uses FasterR-CNN (Ren et al. 2017) trained on the MS-COCO Dataset (Lin et al. 2014) as object detection module. The results verify that all the component contribute to the whole model.

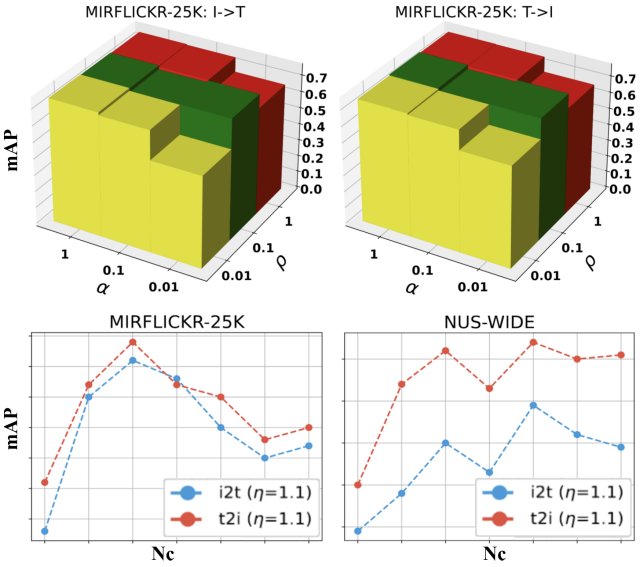

Figure 5: Hyperparameter analysis on α and ρ and N_c with 32 hash bits on two datasets.

Parameter Analysis

We perform analysis on hyper-parameter $\alpha, \rho, \mathrm{N}_c$, which denote the weight of the semantic alignment loss, the parameter to balance the intra-modal and inter-modal reconstruction error, and the number of clustering centers used by K-Means clustering, respectively. We empirically conduct the experiments on MIRFLICKR-25K with 32 hash bits. As shown in Figure 5, our method is not sensitive to the hyperparameters.

Conclusion

In this paper we propose a novel unsupervised cross-modal hashing architecture via vision-guided text mining. Experimental results show that the proposed VTM-UCH outperforms state-of-the-art approaches.

Acknowledgements

This work was supported in part by the grant of the National Science Foundation of China under Grant No. 62202438; the Natural Science Foundation of Shandong Province Grant No. ZR2024MF128.

References

Cao, Y.; Liu, B.; Long, M.; and Wang, J. 2018. Cross-Modal Hamming Hashing. In *Computer Vision – ECCV 2018*, 207–223. Springer International Publishing.

Chua, T.-S.; 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 *Proceedings of the ACM International Conference on Image and Video Retrieval*.

Deng, C.; Chen, Z.; Liu, X.; Gao, X.; and Tao, D. 2018. Triplet-Based Deep Hashing Network for Cross-Modal Retrieval. *IEEE Transactions on Image Processing*, 27(8): 3893–3903.

Desai, K.; Kaul, G.; Aysola, Z.; and Johnson, J. 2021. RedCaps: web-curated image-text data created by the people, for the people. arXiv:2111.11431.

Ding, G.; Guo, Y.; and Zhou, J. 2014. Collective matrix factorization hashing for multimodal data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2083–2090.

Ding, K.; Fan, B.; Huo, C.; Xiang, S.; and Pan, C. 2017. Cross-Modal Hashing via Rank-Order Preserving. *IEEE Transactions on Multimedia*, 19(3): 571–585.

Do, T.-T.; Doan, A.-D.; and Cheung, N.-M. 2016. Learning to Hash with Binary Deep Neural Network. In *Computer Vision – ECCV 2016*, 219–234. Springer International Publishing.

Han, H.; Zheng, Q.; Dai, G.; Luo, M.; and Wang, J. 2024. Learning to Rematch Mismatched Pairs for Robust Cross-Modal Retrieval. arXiv:2403.05105.

Hoe, J. T.; Ng, K. W.; Zhang, T.; Chan, C. S.; Song, Y.-Z.; and Xiang, T. 2021. One Loss for All: Deep Hashing with a Single Cosine Similarity based Learning Objective. arXiv:2109.14449.

Hu, H.; Xie, L.; Hong, R.; and Tian, Q. 2020. Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3123–3132.

Hu, P.; Zhu, H.; Lin, J.; Peng, D.; Zhao, Y.-P.; and Peng, X. 2023. Unsupervised Contrastive Cross-Modal Hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3877–3889.

Huiskes, M. J.; and Lew, M. S. 2008. The MIR flickr retrieval evaluation. In *Proceedings of the ACM International Conference on Multimedia Information Retrieval*, 39–43.

Huo, Y.; Qin, Q.; Dai, J.; Wang, L.; Zhang, W.; Huang, L.; and Wang, C. 2024. Deep Semantic-Aware Proxy Hashing for Multi-Label Cross-Modal Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(1): 576–589.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q. V.; Sung, Y.; Li, Z.; and Duerig, T. 2021. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. arXiv:2102.05918.

Kumar, S.; and Udupa, R. 2011. Learning hash functions for cross-view similarity search. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1360–1365.

Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; Duerig, T.; and Ferrari, V. 2020. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *International Journal of Computer Vision*, 128(7): 1956–1981.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, 740–755. Springer International Publishing.

Liong, V. E.; Lu, J.; Wang, G.; Pierre, M.; and Zhou, J. 2015. Deep hashing for compact binary codes learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2475–2483.

Liu, H.; Ji, R.; Wu, Y.; Huang, F.; and Zhang, B. 2017. Cross-modality binary code learning via fusion similarity hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6345–6353.

Liu, S.; Qian, S.; Guan, Y.; Zhan, J.; and Ying, L. 2020. Joint-modal distribution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1379–1388.

Liu, Y.; Wu, Q.; Zhang, Z.; Zhang, J.; and Lu, G. 2023. Multi-Granularity Interactive Transformer Hashing for Cross-modal Retrieval. In *Proceedings of the ACM International Conference on Multimedia*, 893–902. Association for Computing Machinery.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2): 129–137.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6): 1137–1149.

Shi, Y.; Zhao, Y.; Liu, X.; Zheng, F.; Ou, W.; You, X.; and Peng, Q. 2022. Deep adaptively-enhanced hashing with discriminative similarity guidance for unsupervised cross-modal retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10): 7255–7268.

Song, J.; Yang, Y.; Yang, Y.; Huang, Z.; and Shen, H. T. 2013. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 785–796. Association for Computing Machinery.

Su, S.; Zhong, Z.; and Zhang, C. 2019. Deep Joint-Semantics Reconstructing Hashing for Large-Scale Unsupervised Cross-Modal Retrieval. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3027–3035.

Sun, Y.; Dai, J.; Ren, Z.; Chen, Y.; Peng, D.; and Hu, P. 2024a. Dual Self-Paced Cross-Modal Hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15184–15192.

Sun, Y.; Liu, K.; Li, Y.; Ren, Z.; Dai, J.; and Peng, D. 2024b. Distribution Consistency Guided Hashing for Cross-Modal Retrieval. In *Proceedings of the ACM International Conference on Multimedia*, 5623–5632.

Sun, Y.; Ren, Z.; Hu, P.; Peng, D.; and Wang, X. 2023. Hierarchical consensus hashing for cross-modal retrieval. *IEEE Transactions on Multimedia*, 26: 824–836.

Traag, V. A.; Waltman, L.; and van Eck, N. J. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1).

Tu, J.; Liu, X.; Lin, Z.; Hong, R.; and Wang, M. 2022. Differentiable Cross-modal Hashing via Multimodal Transformers. In *Proceedings of the ACM International Conference on Multimedia*, 453–461. Association for Computing Machinery.

Tu, R.-C.; Mao, X.-L.; Ji, W.; Wei, W.; and Huang, H. 2023a. Data-Aware Proxy Hashing for Cross-modal Retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 686–696. Association for Computing Machinery.

Tu, R.-C.; Mao, X.-L.; Lin, Q.; Ji, W.; Qin, W.; Wei, W.; and Huang, H. 2023b. Unsupervised Cross-Modal Hashing via Semantic Text Mining. *IEEE Transactions on Multimedia*, 25: 8946–8957.

Tu, R.-C.; Mao, X.-L.; Lin, Q.; Ji, W.; Qin, W.; Wei, W.; and Huang, H. 2023c. Unsupervised cross-modal hashing via semantic text mining. *IEEE Transactions on Multimedia*, 25: 8946–8957.

Tu, R.-C.; Mao, X.-L.; Tu, R.-X.; Bian, B.; Cai, C.; Wang, H.; Wei, W.; and Huang, H. 2023d. Deep Cross-Modal Proxy Hashing. *IEEE Transactions on Knowledge and Data Engineering*, 35(7): 6798–6810.

Wang, D.; Wang, Q.; and Gao, X. 2018. Robust and flexible discrete hashing for cross-modal similarity search. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10): 2703–2715.

Wang, Y.; Xue, B.; Cheng, Q.; Chen, Y.; and Zhang, L. 2021. Deep Unified Cross-Modality Hashing by Pairwise Data Alignment. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-21*, 1129–1135. International Joint Conferences on Artificial Intelligence Organization.

Wu, G.; Lin, Z.; Han, J.; Liu, L.; Ding, G.; Zhang, B.; and Shen, J. 2018. Unsupervised Deep Hashing via Binary Latent Factor Models for Large-scale Cross-modal Retrieval. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 2854–2860. International Joint Conferences on Artificial Intelligence Organization.

Yang, D.; Wu, D.; Zhang, W.; Zhang, H.; Li, B.; and Wang, W. 2020. Deep semantic-alignment hashing for unsupervised cross-modal retrieval. In *Proceedings of the International Conference on Multimedia Retrieval*, 44–52.

Yang, E.; Deng, C.; Liu, W.; Liu, X.; Tao, D.; and Gao, X. 2017. Pairwise Relationship Guided Deep Hashing for Cross-Modal Retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Yu, J.; Zhou, H.; Zhan, Y.; and Tao, D. 2021. Deep Graph-neighbor Coherence Preserving Network for Unsupervised Cross-modal Hashing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5): 4626–4634.

Yuan, L.; Wang, T.; Zhang, X.; Tay, F. E.; Jie, Z.; Liu, W.; and Feng, J. 2020. Central Similarity Quantization for Efficient Image and Video Retrieval. arXiv:1908.00347.

Zhong, F.; Chu, C.; Zhu, Z.; and Chen, Z. 2023. Hypergraph-enhanceh hashing for unsupervised cross-modal retrieval via robust similarity guidance. In *Proceedings of the ACM International Conference on Multimedia*, 3517–3527.

Zhu, L.; Wu, X.; Li, J.; Zhang, Z.; Guan, W.; and Shen, H. T. 2023. Work Together: Correlation-identity reconstruction hashing for unsupervised cross-modal retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 8838–8851.