Data Science and Engineering
https://doi.org/10.1007/s41019-024-00248-9

RESEARCH PAPER

Category-Level Contrastive Learning for Unsupervised Hashing in Cross-Modal Retrieval

Mengying Xu[1,2] · Linyin Luo[1,2] · Hanjiang Lai[1,2] · Jian Yin[1,2]

Received: 28 November 2023 / Revised: 4 February 2024 / Accepted: 21 February 2024

Abstract
Unsupervised hashing for cross-modal retrieval has received much attention in the data mining area. Recent methods rely on image-text paired data to conduct unsupervised cross-modal hashing in batch samples. There are two main limitations for existing models: (1) learning of cross-modal representations is restricted to batches; (2) semantically similar samples may be wrongly treated as negative. In this paper, we propose a novel category-level contrastive learning for unsupervised cross-modal hashing, which alleviates the above problems and improves cross-modal query accuracy. To break the limitation of learning in small batches, a selected memory module is first proposed to take global relations into account. Then, we obtain pseudo labels through clustering and combine the labels with the Hadamard Matrix for category-centered learning. To reduce wrong negatives, we further propose a memory bank to store clusters of samples and construct negatives by selecting samples from different categories for contrastive learning. Extensive experiments show the significant superiority of our approach over the state-of-the-art models on MIRFLICKR-25K and NUS-WIDE datasets.

Keywords Cross-modal hashing · Unsupervised · Category-level

1 Introduction

With the demand for multi-modal data increasing, cross-modal retrieval researches [13, 36, 38, 39] have attracted great attention in the data mining area. In cross-modal retrievals, representation in one modality (e.g. image) is used to retrieve similar representations in another modality (e.g. text). The costly storage and computation of these representations remain a challenge in this task. Thus, hashing retrieval methods [1, 28, 34, 37] are introduced to map high-dimension cross-modal datapoints to low-dimension hash codes, which reduce storage demand and speed up retrieval. The mainstream cross-modal hashing methods can be divided into supervised [2, 12, 27] and unsupervised [6, 33]. Supervised methods learn the hash codes using label information, which have gained promising performance but require a large amount of manually annotated labels. Unsupervised methods leverage contrastive learning objective to minimize the distance between similar samples, which avoid the costly annotating process and attracts more attention in researches.

Unsupervised cross-modal hashing methods only use the image-text pairs for training, and the goal is to pull closer the similar samples. The key problem is how to judge the similarity between image-text pairs without using labels. Thus, the similarity matrix plays a crucial role in training. Many models [17, 22, 32] work on optimizing the similarity matrix to improve performance. For example, DJSRH [22] added a joint semantics affinity matrix that is built on one input batch to enhance semantic information in the similarity matrix. DSAH [32] designed a semantic-alignment loss to exploit the connections between images and texts within one batch. JDSH [17] fused the similarity matrix of image-to-image, text-to-text and image-to-text in one batch, and it used the

✉ Jian Yin
issjyin@mail.sysu.edu.cn

Mengying Xu
xumy55@mail2.sysu.edu.cn

Linyin Luo
luoly36@mail2.sysu.edu.cn

Hanjiang Lai
laihanj3@mail.sysu.edu.cn

[1] Sun Yat-sen University, No. 135, Xingang Xi Road, Guangzhou 510275, Guangdong, China

[2] Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China

Published online: 02 April 2024

distribution-based similarity decision method to enhance the discrimination ability of hash codes.

The mentioned models improve performance by using different ways of adding information to the similarity matrix, but they mainly use image-text pairs from one batch. Thus, only batch information is leveraged and the global relations in the entire data distribution are neglected, which results in information incompleteness. The recent study UCCH [9] alleviates this problem by using an additional module to save external data to bring in some relations out of batch. However, the added global relations which are randomly selected features from part of the data set is limited. Moreover, the model employs a random selection process to obtain negative samples for contrastive learning, which may wrongly treat some semantic similar samples as negative. From the above analysis, we can see that existing models face two primary challenges: (1) the learning of cross-modal representation is limited to batches, and (2) semantically similar samples may be incorrectly classified as negative.

To face the above challenges, we propose a novel category-level approach to effectively capture global relations and to improve the selection process of negative samples. As shown in Fig. 1, we first cluster cross-modal features from the encoder modules using K-means to obtain a pseudo-class label for each instance. Next, the cross-modal features are stored in a memory bank, and we acquire the hash codes of features using the hash function. We then conduct category-level contrastive learning by selecting samples from different categories as negatives. Meanwhile, we combined pseudo-class labels with the class centers obtained from a Hadamard matrix to optimize the hash code. In our approach, we learn the categorical relations of representations, not just relations in batch. In addition, negatives are chosen according to pseudo-classes labels rather than randomly. This helps alleviate the above-mentioned problems. Our method uses the MIRFLICKR-25K and NUS-WIDE datasets for image-text retrievals. The experiment results demonstrate that our proposed category-level hash methods effectively capture the similarity relationship compared to existing hash methods, resulting in superior retrieval performance.

The primary contributions of this paper are as follows:

- We propose a novel category-level cross-modal information retrieval model that utilizes the cluster center approach to completely capture the global relation, which effectively addresses the issue of the incompleteness of representation learning.
- We employ a memory bank to store the samples of different categories, and select negative samples according to classes rather than randomly, thereby preventing semantically similar samples from being treated incorrectly.

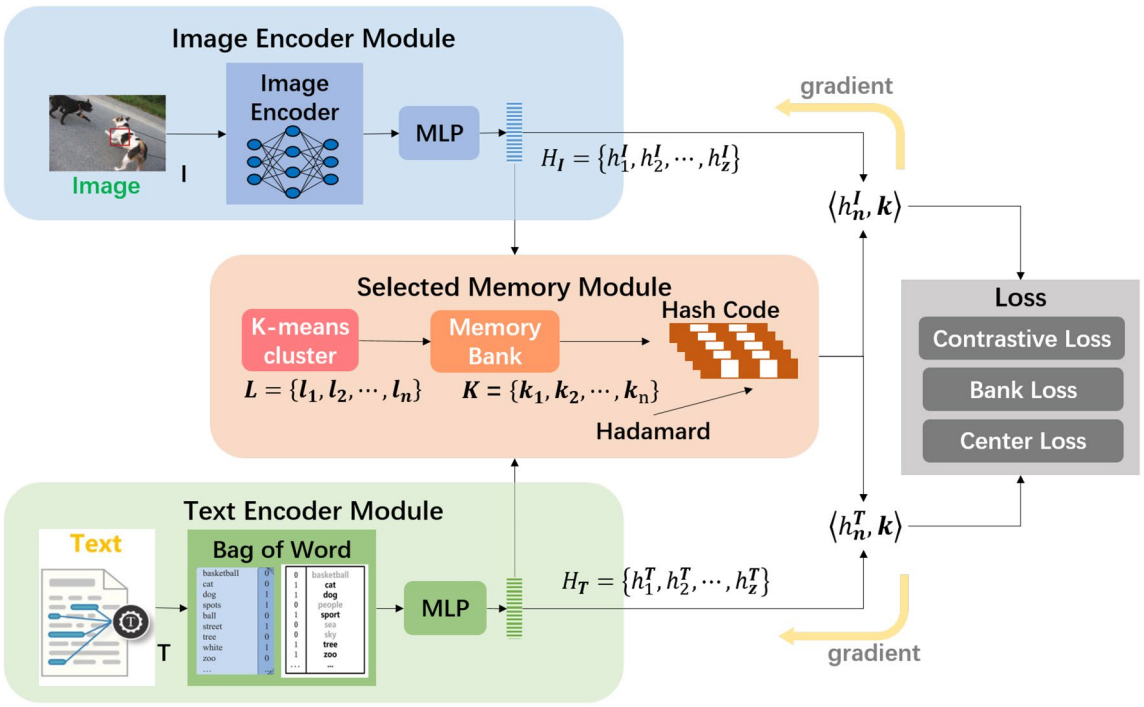

Fig. 1 Overview of the proposed architecture. The Image Encoder Module (up) encodes images. The Text Encoder Module (down) encodes texts. The Selected Memory Module (middle) assigns pseudo labels and conducts category-level contrastive learning. The three loss functions (right) are the Contrastive Loss, the Bank Loss and the Center Loss

- The experiment demonstrates that our method exhibits significant enhancements when compared with the current state-of-the-art (SOTA) methods.

2 Related Work

In this section, we give an introduction to the unsupervised cross-modal hashing retrieval methods, including shallow methods, deep network methods and methods that use contrastive learning.

Shallow hashing methods Kumar et al. first proposed the CVH [14] model to map similar samples to similar hash codes. Later, some models [6, 21, 40] introduced hashing retrieval methods to cross-modal cases and improved hash functions to reduce storage space and speed up retrieval. However, these shallow models cannot capture nonlinear semantics. Thus, deep network methods were proposed.

Deep hashing methods Unsupervised deep cross-modal hashing (UDCMH) [30] first used the deep neural network to bridge the gap between modalities, which enabled feature joint learning to optimize with binarization. After that, an unsupervised generative adversarial cross-modal hashing approach (UGACH) [35] was proposed to exploit the underlying manifold structure of multi-modal data. However, these models did not make full use of the pair information. While contrastive learning [8, 16, 31] aims to leverage pair information, the success of contrastive hashing in tasks like image retrieval [11, 15, 19] inspired numerous researchers to investigate its application in cross-modal hashing.

Contrastive hashing methods Recent approaches [4, 24] had leveraged contrastive learning to align the embeddings of different modalities and focused on how to design a more high-quality instance similarity matrix to guide the training of hashing networks. Deep joint semantics reconstructing hashing (DJSRH) [22] first used the comparison method to integrate the original neighbourhood relations from different modalities. However, DJSRH failed to capture semantic correlations among instances sufficiently and effectively. Later, JDSH [17] constructed a joint modal similarity matrix that fused similar information based on batches to generate more semantic representations. Only using features between data was insufficient to describe intricate data relationships, so Yu *et al.* devised a deep graph-neighbour coherence-preserving network (DGCPN) [33] to regulate three types of similarities, still based on batches. The problem of mistreatment of semantic similar samples still exists.

In summary, the existing methods that leverage contrastive learning are based on batches, which primarily concentrate on improving the batchsize similarity matrix and neglect the overall data semantic information. Although unsupervised contrastive cross-modal hashing (UCCH) [9] that introduced memory banks to take the place of binary values alleviated the problem of training only on batch to some extent, its added sample relation was still limited, and raised the problem of wrongly selected negatives. We can regard the previous methods as instance-level methods, ignoring the overall information relationship. In this paper, we propose a model to consider the relations in the entire data distribution information and utilize semantic relationships between categories to improve the negative samples selection for contrastive learning.

3 Methods

In this section, we first give a task description in Sect. 3.1. Then we provide an overview of the proposed model architecture in Sect. 3.2. Finally, we present our three loss functions: the Contrastive Loss, the Bank Loss and the Center Loss in Sect. 3.3.

3.1 Task Description

In cross-modal hashing retrieval, we use one modality(e.g. image) to retrieve the most similar data in another modality(e.g. text). Unsupervised cross-modal hashing retrieval is to find the sample relationship by mining the intrinsic characteristics of the data without relying on any label value. In unsupervised hashing, there is no label information, so we have to rely on pairs of images and text. The following is the formal definition. The traditional image-text dataset is composed of image-text pairs, where each pair contains an image and the corresponding text information, denoted as $D = \{I_n, T_n\}_{n=1}^{Z}$. Let $I = \{I_n\}_{n=1}^{Z}$ denote a set of images and $T = \{T_n\}_{n=1}^{Z}$ denote a set of texts. Given an image I_n, the goal is to retrieve T_m in T where T_m is the most similar to I_n. Similarly, given a text T_n, the goal is to retrieve the most similar I_m in I.

3.2 Network Architecture

The overall architecture is shown in Fig. 1. Our network can be divided into three components: the image encoder module, the text encoder module and the selected memory module. The encoder modules are used to obtain cross-modal data representations and are described in Sect. 3.2.1. The selected memory module is our core component and will be explained in detail in Sect. 3.2.2.

3.2.1 Image Encoder Module and Text Encoder Module

The input to our network is Z pairs of images and texts $\{I_n, T_n\}_{n=1}^{Z}$. We use the image encoder module and text encoder module to obtain image and text representations respectively. For images, we feed them into the VGG-19

[20] image encoder and a Multi-Layer Perceptron (MLP) to obtain the image representation $H_I = \{h_1^I, h_2^I, ..., h_z^I\}$. For texts, we employ the Bag-of-Words (BoW) method that uses 0 and 1 to denote the existence and non-existence of a word in a sentence to get tags. The tags are then passed through the MLP layer to obtain text representations $H_T = \{h_1^T, h_2^T, ..., h_z^T\}$.

3.2.2 Selected Memory Module

With pairs of multi-modal representations $\{(h_1^I, h_1^T) \dots (h_z^I, h_z^T)\}$ as input, this module aims to generate high-quality hash codes and select negative samples for contrastive learning. The specific process is as follows, we first cluster the multi-modal pairs to get pseudo-category labels. We then store all the representations in a memory bank and generate hash codes using the hash function. Next, we conduct contrastive learning by selecting negative samples from different categories. Meanwhile, we use the pseudo labels and centers obtained from a Hadamard Matrix to optimize the hash codes. Now we explain each step in detail.

The first step is to cluster the representations to get pseudo labels for pairs. We first concatenate the output of the two encoder modules to get the multi-modal representation H, which is shown in Eq. 1.

$$H = concat(H_I, H_T) \tag{1}$$

We then use k-means clustering algorithm [3] to cluster H into m groups. Specifically, the representations are predivided into m groups. The initial cluster centers are determined by randomly selecting m objects. Then, we iteratively calculate the distance between each representation and the cluster centers, assigning each representation to the cluster center that is closest to it. The cluster centers are updated for the next round's calculation. In this way, each multi-modal representation is given a pseudo-class label, denoted as $L = \{l_1, l_2, ..., l_n\}$. The value of parameter m is set to 24 in our research paper.

The second step is to store all the multi-modal representations in the memory bank. Unlike the traditional methods that usually learn representations in the batch data, we store global multi-modal information of different categories by building a memory bank, which alleviates batch learning restriction. The image-text representations stored in the bank will later be used to provide negative samples for contrastive learning. The input of memory bank is the multi-modal representations obtained by the encoders. Given query $h_i^*, *\in i, t$, We aim to directly retrieve the correlated/positive keys from $K = \{k_1, k_2, ..., k_n\}$. The i-th key k_i corresponds to the i-th image-text pair. We update the memory bank by the momentum update method as follows:

$$\mathbf{v}_{i'} = \delta \mathbf{v}_{i'} + (1 - \delta) \frac{\mathbf{h}_i^x + \mathbf{h}_i^y}{2} \tag{2}$$

$$k_i = sign(v_i) \tag{3}$$

This training objective of the memory bank accomplishes searching by keys. That is, given a multi-modal representation h, we obtain the key k_i and use the key to retrieve relevant queries in the memory bank K_n. The retrieved result is denoted as $\{k_i^+\}$. Next, we optimize the negative samples selection process for contrastive learning by selecting negative samples in different categories of the memory bank. The negative samples(denoted as $\{k_j^-\}_{j=-1}^K$) consist of distinguished class samples, which enhance the discriminative power of contrastive learning and avoid similar samples being mis-selected. Using a hash function f composed of a sign function and MLP, we obtain image hash codes $B_I = b_n^i$ and text hash codes $B_T = b_n^t$ for each pair. The hash code $b_n^*, *\in i, t$ has length q for the convenience of retrieval.

The third step is to optimize the class hash centers and pull the class hash centers as far away as possible, this provides optimized class hash centers for the next hash codes training process. In brief, we train the image and text hash codes based on the optimized class hash centers and the pseudo-class labels from the first step. We initialize the hash class centers from the Hadamard matrix. In this way, we can ensure that the class centers vector is composed of 1 and -1. In addition, any two rows (or columns) of the Hadamard matrix are orthogonal according to the features of the orthogonal square. Thus, the constructed class vectors are orthogonal to each other.

Now we optimize the hash centers to maximize the distances between class hash centers. We refer to [25], a method which optimizes the class hash center with the constraint that the Hamming distance between any two centers is not less than the minimum distance d. This method uses Gilbert-Varshamov bound [23] to obtain a large d. Each of these hash centers corresponds to one class respectively.

Given samples in m classes, we optimize m hash centers $C = \{c_1, c_2, ..., c_m\}$ of our hash codes by maximizing the optimization target and with the restricted condition shown in Eq. 4.

$$\begin{aligned} &\max_{c_1,...,c_m \in \{-1,1\}^q} \frac{1}{m(m-1)} \sum_i \sum_{j:j\neq i} \left\| c_i - c_j \right\|_H \\ &\text{s.t. } \left\| c_i - c_j \right\|_H \geq d (1 \leq i, j \leq m, j \neq i), \end{aligned} \tag{4}$$

where q is the hash code length, d is the minimal distance and $||.||_H$ represents the Hamming distance. We use the Gilbert-Varshamov bound [23] to determine a large minimal distance d while ensuring the feasibility of our optimization procedure. Specifically, there are m q-bit codes that the

minimal Hamming distance of any two codes is d, as long as m, q and d satisfy $\frac{2^q}{c} \leq \sum_{i=0}^{d-1} \binom{q}{i}$. Hence, to obtain a large d, we only need the maximum of d to satisfy the equation. Due to the monotone increasing function $f(d) = \sum_{i=0}^{d} \binom{q}{i}$, we have

$$\begin{cases} \frac{2^q}{c} \leq \sum_{i=0}^{d^\star-1} \binom{q}{i} \\ \frac{2^q}{c} > \sum_{i=0}^{d^\star-2} \binom{q}{i} \end{cases} \tag{5}$$

Since $d^\star$ is an integer 1, 2, 3, ..., q, we can find its value by exhaustively searching. This objective makes hash centers of different categories as far as possible.

We further leverage the following facts to improve the implementation of Eq. 4.

1. The hamming distance $\left\|c_i - c_j\right\|_H$ is equivalent to $-c_i^T c_j$. Maximizing the hash distance is equivalent to minimizing the inner product [26].
2. $4\left\|c_i - c_j\right\|_H = \left\|c_i - c_j\right\|_2^2 = c_i^T c_i + c_j^T c_j - 2c_i^T c_j = 2q - 2c_i^T c_j$ [25], where, $\|.\|_2$ is ℓ_2 norm.
3. The inner product of two hash centers is bounded by an upper limit to ensure accuracy, which limits the minimal hamming distance d with $\left\|c_i - c_j\right\|_H \geq d$.

Thus, Eq. 4 can be simplified as follows:

$$\begin{aligned} \min_{c_1,c_2,\ldots,c_m \in \{-1,1\}^q} & \sum_{j:j\neq i} c_i^T c_j \\ \text{s.t. } & c_i^T c_j \leq (q-2d) \\ & (1 \leq i,j \leq m, i \neq j). \end{aligned} \tag{6}$$

To simplify the implementation, we adopt an optimization procedure that alternately updates one of the hash centers c_i while keeping other centers $c_j (1 \leq j \leq m; j \neq i)$ fixed. Specifically, when all $c_j (j \neq i)$ are fixed, the subproblem w.r.t. c_i can be formulated as:

$$\begin{aligned} \min_{c_i \in \{-1,1\}^q} & \sum_{j:j\neq i} c_i^T c_j \\ \text{s.t. } & c_i^T c_j \leq q - 2d (1 \leq j \leq m, j \neq i), \end{aligned} \tag{7}$$

To solve the subproblem in Eq. 7, we utilize the $\ell_p - box$ algorithm proposed in [29]. The $\ell_p - box$ algorithm showed that the binary constraint $z \in \{-1,1\}^q$ is equivalent to $z \in [-1,1]^q \bigcap \left\{z : \|z\|_p^p = q\right\}$. The proof of the $\ell_p - box$ algorithm can refer to [25]. We set $p = 2$ for simplicity. By dropping the binary constraint, we can reformulate Eq. 7 into the following equivalent form.

$$\begin{aligned} \min_{c_i,z_1,z_2} & \sum_{j:j\neq i} c_i^T c_j \\ \text{s.t. } & c_i^T c_j \leq (q-2d) \\ & (1 \leq j \leq m, i \neq j) \\ & c_i = z_1, c_i = z_2, z_1 \in \mathcal{S}_b, z_2 \in \mathcal{S}_p, \end{aligned} \tag{8}$$

where $S_b = \left\{z : -1_q < z < 1_q\right\}$ and $S_p = \left\{z : \|z\|_2^2 = q\right\}, 1_q$ represents a q-dimensional vector with all ones. We obtain S_b and S_p by using the $\ell_p - box$ algorithm above.

An equality constraint can replace the inequality constraint by adding an auxiliary variable z_3. The inequality constraints $c_i^T c_j \leq (q-2d)$ is equal to the equality constraint $c_i^T C_{\sim i} + z_3 = (q-2d)1_{m-1}$ where $C_{\sim i} = [c_1, c_2, ..., c_i - 1, c_i + 1, ..., c_m]$ and $z_3 \in R_+^{m-1}$, $R_+^{c-1} = \left\{z : z \in [0, +\infty)^{m-1}\right\}$, further reformulating the problem in Eq. 8 as:

$$\begin{aligned} \min_{c_i,z_1,z_2,z_3} & \sum_{j:j\neq i} c_i^T c_j \\ \text{s.t. } & c_i^T C_{\sim i} + z_3 = (q-2d)1_{m-1} \\ & (1 \leq j \leq m, i \neq j) \\ & c_i = z_1, c_i = z_2, z_1 \in \mathcal{S}_b, z_2 \in \mathcal{S}_p, z_3 \in R_+^{m-1}. \end{aligned} \tag{9}$$

The augmented Lagrange function w.r.t. Equation 9 is:

$$\begin{aligned} L(c_i, z_1, z_2, z_3, y_1, y_2, y_3) = & \sum_{j\neq i} c_i^T c_j \\ & + y_1^T (c_i - z_1) + \frac{\mu}{2}\|c_i - z_1\|_2^2 \\ & + y_2^T (c_i - z_2) + \frac{\mu}{2}\|c_i - z_2\|_2^2 \\ & + y_3^T (c_i^T C_{\sim i} + z_3 - e) \\ & + \frac{\mu}{2}\left\|c_i^T C_{\sim i} + z_3 - e\right\|_2^2 \\ & \text{s.t. } z_1 \in S_b, z_2 \in S_p, z_3 \in R_+^{m-1}, \\ e = (q-2d)1_{m-1} & \end{aligned} \tag{10}$$

where y_1, y_2, y_3 are Lagrange multipliers.

We update each variable c_i, z_1, z_2, z_3 in the following way.

1. Update h_i: We update c_i by fixing other variables except c_i, the subproblem of L in Eq. 10 w.r.t.c_i is an unconstrained objective. The gradient of L w.r.t. c_i is

$$\begin{aligned} \frac{\partial L(c_i)}{\partial c_i} = & 2\mu c_i + \mu C_{\sim i} C_{\sim i}^T c_i + \sum_{j\neq i} c_j \\ & + y_1 + y_2 + C_{\sim i} y_3 \\ & - \mu(z_1 + z_2 + C_{\sim i} e - C_{\sim i} z_3) \end{aligned} \tag{11}$$

By setting this gradient to zero in Eq. 11, we can update c_i by

$$
\begin{aligned}
c_i \leftarrow & (2\mu I_q + \mu C_{\sim i} C_{\sim i}^T)^{-1} \\
& (\mu(z_1 + z_2 + C_{\sim i} e - C_{\sim i} z_3) \\
& - \sum_{j \neq i} c_j - y_1 - y_2 - C_{\sim i} y_3
\end{aligned} \tag{12}
$$

2. Update z_1, z_2, z_3: The subproblem of L in Eq. 10 w.r.t. z_1, z_2, z_3 is:

$$
\begin{cases}
L(z_1) = y_1^T (c_i - z_1) + \frac{\mu}{2} \|c_i - z_1\|_2^2 \\
L(z_2) = y_2^T (c_i - z_2) + \frac{\mu}{2} \|c_i - z_2\|_2^2 \\
L(z_3) = y_3^T (c_i^T C_{\sim i} + z_3 - e) \\
\qquad + \frac{\mu}{2} \left\| c_i^T C_{\sim i} + z_3 - e \right\|_2^2 \\
\qquad \text{s.t. } z_1 \in S_b, z_2 \in S_p, z_3 \in R_+^{m-1}
\end{cases} \tag{13}
$$

Then, we set the gradients to zero in Eq. 13, we can obtain update z_1, z_2, z_3 by

$$
\begin{cases}
z_1 \leftarrow P_{S_b}\left(c_i + \frac{1}{\mu} y_1\right) \\
z_2 \leftarrow P_{S_p}\left(c_i + \frac{1}{\mu} y_2\right) \\
z_3 \leftarrow P_{R_+^{m-1}}\left(e - c_i^T C_{\sim i} - \frac{1}{\mu} y_3\right)
\end{cases} \tag{14}
$$

Following [29], we project z_1, z_2, z_3 into S_b, S_p, R_+^{m-1} respectively. All of these projections have closed-form solutions:

$$
\begin{cases}
z_1 \leftarrow min(1, max(-1, c_i + \frac{1}{\mu} y_1)) \\
z_2 \leftarrow \sqrt{q} \frac{c_i + \frac{1}{\mu} y_2}{\left\| c_i + \frac{1}{\mu} y_2 \right\|_2} \\
z_3 \leftarrow max(0, e - c_i^T C_{\sim} i - \frac{1}{\mu} y_3)
\end{cases} \tag{15}
$$

3. Update y_1, y_2, y_3: The Lagrange multipliers y_1, y_2 and y_3 can be updated by

$$
\begin{cases}
y_1 \leftarrow y_1 + \mu(c_i - z_1) \\
y_2 \leftarrow y_2 + \mu(c_i - z_2) \\
y_3 \leftarrow y_3 + \mu(c_i^T C_{\sim i} - z_3 - e)
\end{cases} \tag{16}
$$

Overall, we can learn class hash centers by maximizing the optimization target. Next, we will describe the loss functions based on class hash centers and the pseudo-class label that improve image hash codes B_I and text hash codes B_T.

3.3 Loss Functions

For training, we use the contrastive loss to align images and texts, the bank loss to maintain the storage accuracy and efficiency of the memory bank, and the center loss to minimize the distance between features and their corresponding class. Now we explain them respectively.

3.3.1 The Contrastive Loss

The contrastive loss is a traditional loss function in image-text matching that has been used by many previous works for optimization. It aims to bring the positive sample pairs closer together and pull the negative sample pairs away.

We apply the contrastive learning loss to maintain apparent similarity among image-text pairs. The function is as follows:

$$
\boldsymbol{L_{co}} = \min_{B_I, B_T} \left\| \mu S - \cos(B_I, B_T) \right\|^2, \tag{17}
$$

where S is a diagonal similarity matrix to evaluate the similarity of the image and text. $S = 1$ means that they are similar and conversely $S = 0$ means that they are not similar. We calculate the cosine similarity matrix $\cos(B_I, B_T)$ to denote the similarity between B_I, B_T and describe the neighbourhood structure in the Hamming space. We minimize the error between the diagonal similarity matrix S and cosine matrix $\cos(B_I, B_T)$.

3.3.2 The Bank Loss

The memory bank is a dictionary-like structure designed for storing image-text pairs. Each key in the bank corresponds to an image-text pair. The Bank Loss function aims to retrieve the most relevant key to the query $h_i^* (* \in \{x, y\})$ directly from all keys K_n. In $\{k_1, k_2, ..., k_n\}$, there is a positive sample key k_i^+ that matches the query $h_i^* (* \in \{x, y\})$. As mentioned previously, when querying, only one sample is positive while the rest are negative keys $\{k_j^-\}$.

We utilize an efficient loss function called InfoNCE [18], as referenced in [9], which maximizes instance-level discrimination and minimizes cross-modal variation:

$$
\boldsymbol{L_b} = -\sum_{n=1}^{z} \log P\left(n \mid \mathbf{h}_n^*\right) \tag{18}
$$

$P\left(n \mid \mathbf{h}_n^*\right)$is the probability of $\mathbf{h}_n^*$ being recognized as the n-th point. The probability calculation method is defined as follows:

$$
P = \frac{exp(\langle \mathbf{h}_n^*, k_n^+ / \tau \rangle)}{exp(\langle \mathbf{h}_n^*, k_n^+ / \tau \rangle) + \sum_{m=1}^{z} exp(\langle \mathbf{h}_n^*, k_n^- / \tau \rangle)} \tag{19}
$$

where $* \in I, T$ and τ represents a temperature hyper-parameter.

The Bank Loss is an unsupervised loss used for learning features given a batch of data without labels. It can be considered as the negative log-likelihood of the non-parameter

softmax classifier. For each i-th image-text pair (i.e., h_i^x, h_i^y), the loss force the pair to correspond with its respective positive sample (i.e., k_i^+) in key.

3.3.3 The Center Loss

To make representaions close to their corresponding hash centers, we incorporate the Center Loss into traditional loss functions. The objective of the Center Loss is to enhance similarity within similar instances and increase dissimilarity between dissimilar ones by leveraging category information.

Using the m hash centers mentioned in the method, we assign a center to one of the m image classes. We use the center loss function, which ensures output hash codes are close to their corresponding class centers while being far away from other hash centers. Specifically, given m hash centers $c_1, c_2, ..., c_m$, N sample with respective output hash codes $b_1, b_2, ..., b_N$. The function is defined as:

$$\begin{aligned} \boldsymbol{L_{ce}} = -\frac{1}{z}\sum_{n=1}^{z}\sum_{i=1}^{m} l_{n,i} log P_{n,i} \\ + (1 - l_{n,i}) log(1 - P_{n,i}) \end{aligned} \tag{20}$$

with

$$P_{n,i} = \frac{e^{-S(b_n, c_i)}}{\sum_{k=1}^{m} e^{-S(b_n, c_k)}}. \tag{21}$$

where $S(.)$ represents the scaled cosine similarity metric, $S(.) = cos()$.

With the Center Loss, our hash codes can capture the global information, never make up for the partial information of local data, and have better discriminability.

In all, the objective function of our hashing network is a combination of the three loss functions:

$$\boldsymbol{L} = \boldsymbol{L_{co}} + \boldsymbol{L_b} + \boldsymbol{L_{ce}} \tag{22}$$

The conventional loss function for image-text research solely comprises contrastive loss, which is the first component of our proposed approach. Our novel loss function, considered from the global perspective, captures the category-level relationships and incorporates bank loss to facilitate effective matching. The utilization of Bank Loss and Center Loss acquires comprehensive global relationships. Our loss incorporates supplementary representations of distinct classes, addressing the limitation associated with batchsize learning. Additionally, Combining these losses alleviates misclassification errors arising from similarities among samples.

4 Experiments

In this section, we present our experiment results. Datasets are described in Sect. 4.1. Evaluation metrics and implementation details are presented in Sect. 4.2. The analysis of results is in Sect. 4.3.

4.1 Datasets

We use two wide-used cross-modal hashing datasets, the MIRFLICKR-25K and NUS-WIDE for experiments.

- **MIRFLICKR-25K** [10]: This dataset is obtained from the Flickr website. It gathers a total of 25,000 photos. Additionally, each image is paired with a corresponding tag, creating image-text pairs. We apply similar pre-processing operations to our data as in UCCH [9]. Input images are the original images, while texts are processed with the Bag-of-Words (BoW) method, and text tags as 1386-dimension vectors are derived.
- **NUS-WIDE** [5]: The dataset comprises 269,648 web images that are categorized into 81 classes. Each image is associated with a corresponding text tag. Preprocessing for NUS-WIDE follows the same steps as in MIRFLICKR-25K. Note that in NUS-WIDE, the final dimension of the text tag vector is 1000.

4.2 Experimental Setting

4.2.1 Evaluation Metrics

The Mean Average Precision (mAP) [7] is a comprehensive metric that captures the average precision across all classes. It is the aggregated weighted average of all the accuracies (AP). In terms of learning to rank, it pertains to the precision averaged over multiple queries.

The updated calculation approach for mAP was officially submitted at the 2012 PASCAL VOC Challenge. This updated calculation presupposes the existence of M-positive examples within N samples, and for the M positive examples, the recall values are ($1/M$, $2/M$, ..., M/M), denoted as $P(k)$. For each value $P(k)$, we multiply the correlation coefficients to compute the corresponding integrated accuracy and then average them to obtain the final AP value. The formula is shown as follows:

$$AP@K = \frac{\sum_1^k P(k) * rel(k)}{\sum rel(k)}. \tag{23}$$

We use the above calculation method of mAP to measure the overall quality of our learned model.

4.2.2 Implementation Details

We use PyTorch to implement our model. For the dataset, we randomly select 10000 and 10500 instances as the training set for MIRFLICKR-25K and NUS-WIDE. Meanwhile, we randomly select non-overlapping 2000 and 2100 instances as the test sets of the two datasets. The batchsize is 128 in training, and we use SGD for the optimizer. The learning rate is set to 10^{-4} and is gradually reduced during training. We used the pre-trained weights provided by VGG-19.

4.3 Experimental Results

This section presents the results of our experiments. Two cross-modal query tasks (Image2Text and Text2Image) are conducted on the two datasets to compare the performance of our model and other baselines. First, we give a comprehensive analysis of the overall model performance compared with the current SOTA model. Then, since our model is the category-level approach, we specifically analyze the effects of the category-level methods and the traditional instance-level methods. Next, we show the P-R curve of traditional retrieval performance. Finally, we analyze the effect of each of our specific parts.

4.3.1 Comparison to the SOTA Methods

To evaluate the effectiveness of our methods, we conduct a comparative analysis with current state-of-the-art (SOTA) unsupervised cross-modal hashing approaches. The comparative SOTA methods we choose include both shallow models and deep networks. CVH and CMFH represent two shallow hashing models in unsupervised cross-modal hashing. DJSRH, JDSH, DGCPN and UCCH are methods based on deep networks, which are also contrastive learning methods. In detail, deep network methods are divided into the traditional methods (such as DJSRH and JDSH) and the introduction of additional relations methods (such as DGCPN which introduces graph relations and UCCH which introduces additional information randomly). For a fair comparison, all methods use VGG-19 as the image encoder. The evaluation metric is the Mean Average Precision @ all (mAP@all).

The results shown in Table 1 is the mAP@all metric on MIRFLICKR-25k in two cross-modal information retrieval tasks, for hash codes with lengths of 16 bit, 32 bit, and 64 bit. I2T means using the image to retrieve text, while T2I refers to using the text to query image. Compared to the shallow cross-modal hashing method, it shows the proposed method has better performance. For example, the average increase in the hash length of the various hash codes is 36.72% and 37.95% for I2T and T2I in CMFH, respectively. This shows that the deep network framework brings better results. Compared to the deep network cross-modal hashing method, it also shows our method achieves superior retrieval accuracy. Compared with the methods DJSRH, and JDSH, which are only in leveraging batchsize data, the performance in the I2T tasks has shown an improvement of 13.09% and 11.18%, while in the T2I tasks, it has improved by 13.03% and 8.04%. It illustrates that our approach can improve performance by capturing not only the relationships within a batch but also the whole relationship. The proposed method demonstrates a significant improvement of 5.017% and 7.07% on the I2T and T2I tasks respectively, when compared to DGCPN with additional relations. In contrast with the highest performance UCCH, the average accuracy has been enhanced by 1.64% and 1.83% in the I2T and T2I tasks. This shows that introducing relationships from the perspective of categories is more logical than introducing graph relationships or random information. These results demonstrate the superior performance of our method over existing baseline methods on the MIRFLICKR-25k dataset.

Table 2 presents the results of the I2T and T2I tasks conducted on the NUS-WIDE dataset, including 16 bit, 32 bit, and 64 bit hash codes. Compared to the SOTA unsupervised shallow cross-modal hash method CMFH, our model achieves approximately double the average performance across different hash lengths for I2T and T2I tasks respectively. Similarly, when compared to the highest performance

Table 1 Results compared with SOTA models in MIRFLICKR-25 dataset

Model	I2T			T2I		
	16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
CVH	0.620	0.608	0.594	0.629	0.615	0.599
CMFH	0.557	0.557	0.556	0.553	0.553	0.553
DJSRH	0.665	0.673	0.681	0.662	0.671	0.692
JDSH	0.669	0.678	0.691	0.673	0.677	0.678
DGCPN	0.709	0.717	0.731	0.713	0.712	0.732
UCCH	0.739	0.751	0.756	0.737	0.755	0.756
Our	0.750	0.762	0.772	0.752	0.763	0.774

Table 2 Results compared with SOTA models in NUS-WIDE dataset

	I2T			T2I		
	16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
CVH	0.487	0.495	0.456	0.470	0.475	0.444
CMFH	0.339	0.338	0.343	0.306	0.306	0.306
DJSRH	0.568	0.580	0.604	0.585	0.578	0.617
JDSH	0.594	0.609	0.612	0.592	0.616	0.627
DGCPN	0.610	0.614	0.635	0.617	0.621	0.642
UCCH	0.658	0.669	0.679	0.666	0.674	0.688
Our	0.683	0.689	0.690	0.690	0.696	0.701

deep cross-modal hash method UCCH, our method achieves an average enhancement in hash code length of 2.79% and 2.73% in the I2T and T2I tasks. The results in Table 2 also demonstrate the superior performance of our method. Overall, it reflects the same conclusion as presented in Table 1. It is noteworthy that our method is more effective on the NUS-WIDE dataset compared to the MIRFLICKR-25k dataset. This observation further demonstrates the significance of capturing the global correlation when dealing with larger volumes of data.

In summary, on the one hand, the results presented in Tables 1 and 2 show that our method is significantly better than the shallow methods. On the other hand, our method comparison with the SOTA deep network methods(such as traditional methods and the introduction of additional relations methods) has significantly improved. This demonstrates the efficacy of our approach in capturing global relationships at the category level, thereby yielding significant outcomes. The results consistently demonstrate the superior performance of our method compared to existing baseline approaches, highlighting substantial and noteworthy improvements.

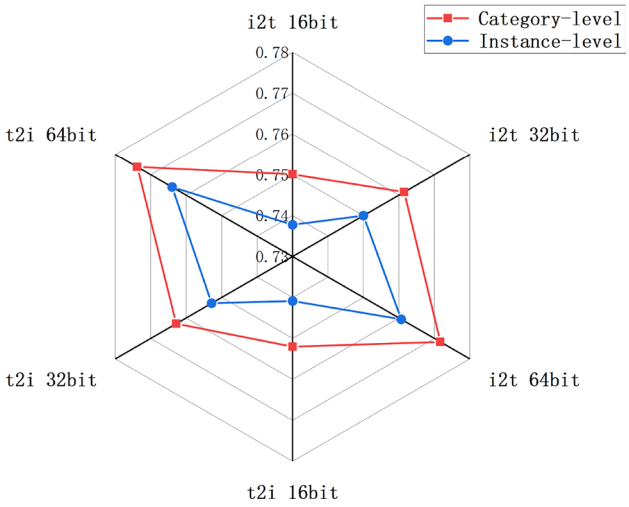

Fig. 2 Comparison of category-level and instance-level approach in MIRFLICKR-25k dataset

4.3.2 Analysis of the Effect of the Category-Level Strategy

Our approach employs the category hierarchy strategy to enhance data information, thereby alleviating the limitation of batch size and preventing samples wrongly treated. Therefore, the application of a category-level strategy is an important component of our approach. To validate the effectiveness of our method employing a category-level strategy, we conduct a comprehensive comparison at both instance-level and category level within the same network framework and experimental conditions.

This part involves the analysis of results on the MIRFLICKR-25k and NUS-WIDE datasets. The results of MIRFLICKR-25k are visually depicted in Fig. 2, while the results of the NUS-WIDE dataset are in Fig. 3. For instance-level approach, we refer to the current best model UCCH [9]. UCCH also stores additional information in the form of a bank, which prevents unfair comparison brought

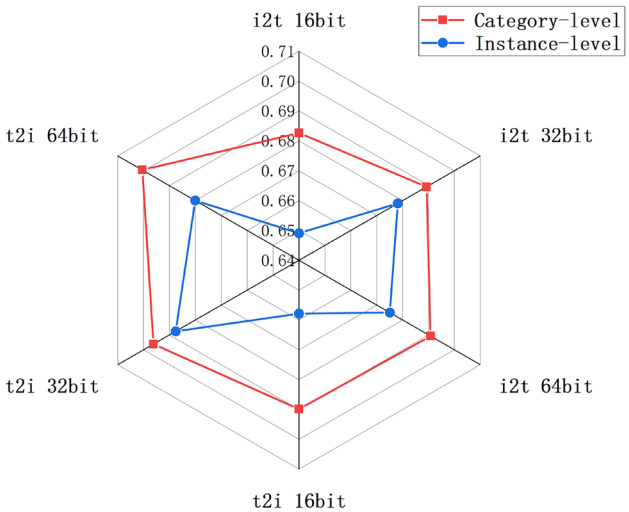

Fig. 3 Comparison of category-level and instance-level approach in NUS-WIDE dataset

increased information. Both UCCH and our model use the same VGG-19 to get the image representation, and then get the hash codes through the hash function. For category-level experiments, we follow the steps described in Sect. 3. The experiments show that the category-level approach achieved notable performance improvements.

We use the radar charts to show the effects of category-level strategy on the MIRFLICKR25 and NUS-WIDE datasets, which is shown in Figs. 2 and 3. The radar charts compare the two strategies(instance-level and category-level) using six axes. Each axis indicates different bit sizes for I2T and T2I information retrieval tasks. Specifically, the bit sizes consideration encompass 16 bit, 32 bit, and 64 bit in I2T and T2I information retrieval. The results show the distinctions between the two methods across multiple dimensions. Note that the central point remains constant, closer to the center indicates a decrease in performance, while closer to the outer edges signifies better performance for the model.

The blue lines depicted in Figs. 2 and 3 illustrate the traditional instance-level cross-modal hashing method, while the red lines in the radar chart represent the category-based hashing method proposed by our research. These charts show notable visual contrast between the red lines and blue lines. The red lines in both figures are outside the blue lines, which means the performance improvement is particularly significant, especially in the 16 bit. The observed outcome can be attributed to the red line using the category-level approach, in contrast to the blue line employing the conventional instance-level approach. In comparison to the superior instance-level UCCH method, the clear distinction between the lines shows a significant performance improvement, indicating that our category-level approach exhibits a significantly higher performance than the best instance-level method. Thereby it highlights the efficacy of category-level information. The category-level method achieves higher query accuracy in cross-modal information retrieval and adaptly captures the correlation between all instances.

4.3.3 Retrieval Performance

In addition to the mAP indicators, we also calculate precision and recall to measure retrieval performances. The precision-recall (PR) curves are utilized as the metrics for hash lookup to evaluate the effectiveness of cross-modal information retrieval. The results are reported on the MIRFLICKR-25k and NUS-WIDE datasets. The results of

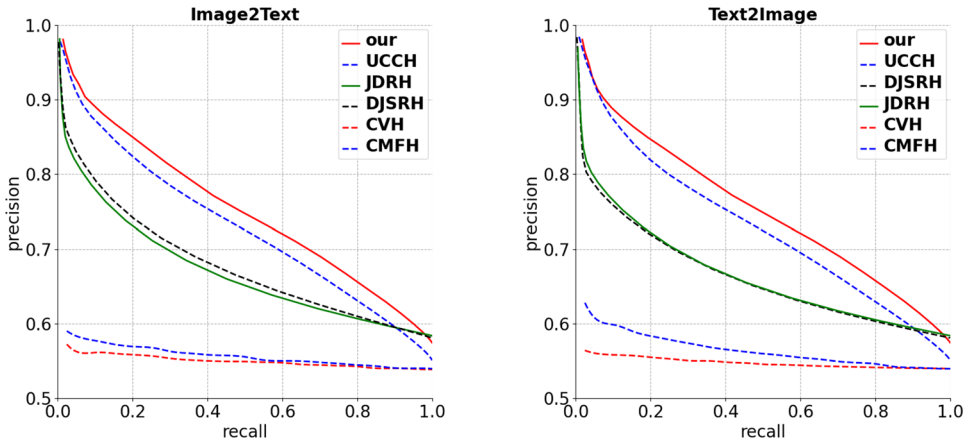

Fig. 4 Resulte@32 on MIRFLICKR-25k dataset

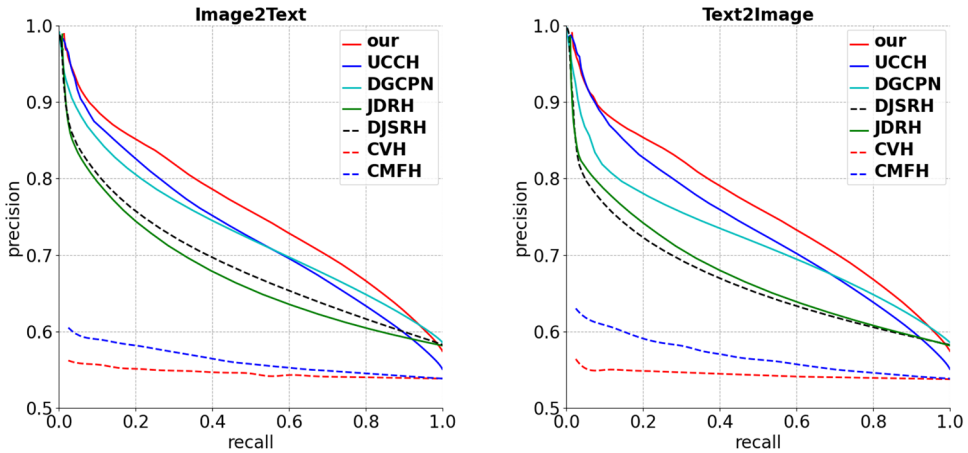

Fig. 5 Resulte@64 on MIRFLICKR-25k dataset

MIRFLICKR-25k are in Figs. 4 and 5, while the results of NUS-WIDE dataset are in Figs. 6 and 7.

The precision-recall curves with code lengths 32 bit and 64 bit are drawn to evaluate the performance of the cross-modal hashing methods on the MIRFLICKR-25K, as shown in Figs. 4 and 5. The better the performance, the positioning of the line more in the upper right corner. If the precision-recall (PR) curves of one model completely cover the PR curves of the other models, it can be inferred that this model shows superior performance. We can see that the PR curves of our method are higher than all those of baselines in the MIRFLICKR-25K dataset. Our methods provide superior performance compared to SOTA methods with various code lengths.

Figures 6 and 7 illustrate the PR curves for various modes of comparison. The results demonstrate that the proposed approach outperforms existing contrast methods. It suggests that the category-based hash approach can achieve higher retrieval accuracy in cross-modal retrieving and effectively capture the correlation between instances.

4.3.4 Ablation Study

In this section, we investigate the contributions of different loss functions in the design.

To evaluate the performance of each component comprehensively, we compare the model we proposed with its three variants in the MIR and NUS-WIDE datasets, as shown in

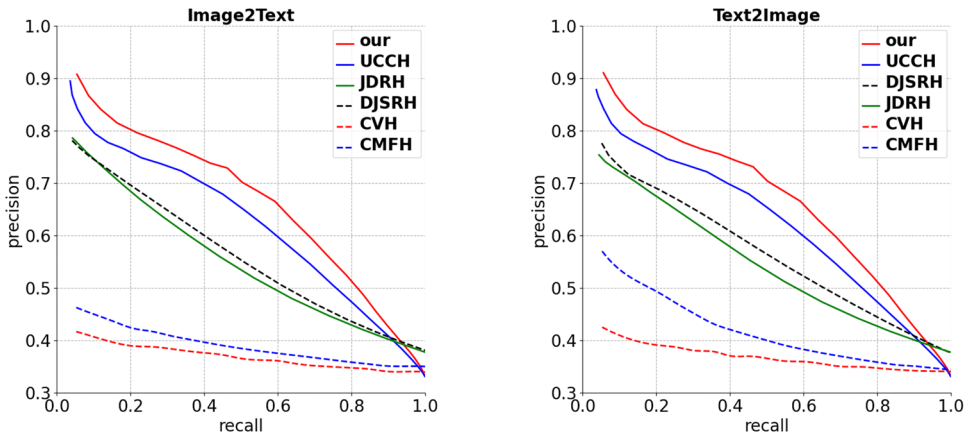

Fig. 6 Resulte@32 on NUS-WIDE dataset

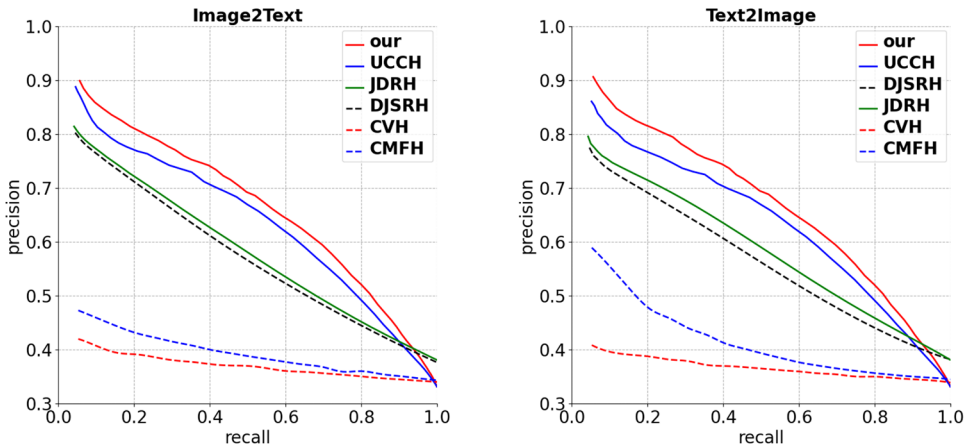

Fig. 7 Resulte@64 on NUS-WIDE dataset

Table 3 Ablation study results in MIRFLICKR-25 dataset

	i2t			t2i		
	16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
Our	0.750	0.762	0.772	0.752	0.763	0.774
w/o Center loss	0.738	0.750	0.761	0.741	0.753	0.764
w/o Bank loss	0.606	0.637	0.642	0.594	0.637	0.644
w/o Contrastive loss	0.739	0.752	0.764	0.739	0.751	0.763

Table 4 Ablation study results in NUS-WIDE dataset

	i2t			t2i		
	16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
Our	0.683	0.689	0.691	0.690	0.696	0.700
w/o Center loss	0.649	0.678	0.675	0.658	0.688	0.685
w/o Bank loss	0.647	0.668	0.666	0.647	0.668	0.665
w/o Contrastive loss	0.681	0.678	0.689	0.688	0.690	0.697

Tables 3 and 4. The first line of each table is the method this paper proposed. The second line eliminates the Center Loss. The third line removes the Bank Loss. Eliminating the bank loss necessitates the removal of the memory bank module, which contributes no additional information. However, it also impacts our capacity to capture global relationships. and the fourth line excludes the Contrastive Loss.

The results in Table 3 show that integrating three loss functions in our model improves performance on the MIR-FLICKR-25K dataset. From the mAP@all results in the first row and the second row, we have observed that the Center Loss is improved by 1.55% and 1.37% on average, in i2t and t2i tasks. It shows the effective implementation of the Center Loss significantly contributes to enhancing the performance of our model. The Center loss enables the gathering of similar information in the dataset, resulting in a convergence of similar representations and a separation of dissimilar ones. This process can facilitate the data distribution in multiple clusters, thereby capturing category-level information.

Based on the mAP@all results presented in the first row and the third row, we find that the Bank Loss is improved by 21.19% and 22.21% on average, in i2t and t2i tasks. This shows that Bank Loss plays a crucial role in changing the retrieval of one modality to another modality into the corresponding retrieval of one modality to one key, which makes the retrieval performance better. To a certain extent, it plays an important role in facilitating image-text alignment and introducing additional information. In addition, the feature stored in the memory bank, if the bank loss is removed, is equivalent to not adding additional relations, which has a great impact on obtaining the overall relationship. Therefore, the impact of removing the bank is large.

The mAP@all results presented in the first and fourth rows illustrate that the Contrastive Loss average improves 1.28% and 1.59% in i2t and t2i tasks. It demonstrates that the Contrastive Loss can help improve the matching of different modalities data to some extent. However, the Bank Loss also plays a certain role in different mode matching. Thus, the Contrastive Loss in this context exhibits limited enhancement.

The results from Table 4 in the NUS-WIDE dataset also support similar conclusions. The Center Loss, the Bank Loss and the Contrastive Loss correspond to an average increase of 2.92%,4.78% and 0.37%, respectively. However, it is worth noting in Table 4 that The effectiveness of Center Loss is significantly greater than that of Contrastive Loss. On one hand, the impact of Center Loss surpasses that of Contrastive Loss, possibly due to increased data volume and heightened significance of category relationships. Therefore, the traditional Contrastive Loss effect is reduced. On the other hand, our Bank Loss in matching may have contributed to its role in the matching process. It is crucial to emphasize the increasing significance of Center Loss functions in enhancing performance as the size of the dataset grows.

5 Conclusion

This paper proposed a cross-modal hash retrieval model based on the category-level to address the problem of incomplete information in traditional instance-level contrastive learning. We proposed a selected memory module to give pseudo labels to image-text pairs. With the help of pseudo labels and class centers obtained from a Hadamard Matrix, we conducted category-level contrastive learning. Experiment results showed that the selected memory module and the three loss functions we used contributed to alleviating the problems we observed. We conducted comprehensive experimental analyses on the MIR-FLICKR-25K and NUS-WIDE datasets results demonstrate the effectiveness of our method. In the future, we will further discuss how to enhance the optimization of positive samples.

Funding This work is supported by the Research Foundation of Science and Technology Plan Project of Guangzhou City (2023B01J0001, 2024B01W0004).

References

1. Bronstein MM, Bronstein AM, Michel F et al (2010) Data fusion through cross-modality metric learning using similarity-sensitive hashing. In: Computer vision pattern recognition

2. Cao Y, Liu B, Long M et al (2018) Cross-modal hamming hashing. In: European conference on computer vision
3. Chaudhuri D, Chaudhuri B (1997) A novel multiseed nonhierarchical data clustering technique. IEEE Trans Syst Man Cybern Part B (Cybernetics) 27(5):871–876
4. Chen T, Kornblith S, Norouzi M et al (2021) A simple framework for contrastive learning of visual representations. In: International conference on machine learning
5. Chua TS, Tang J, Hong R, et al (2009) NUS-WIDE: a real-world web image database from National University of Singapore. In: ACM international conference on image and video retrieval
6. Ding G, Guo Y, Zhou J et al (2016) Large-scale cross-modality search via collective matrix factorization hashing. IEEE Trans Image Process 25(11):5427–5440
7. Harter SP, Hert CA (1997) Evaluation of information retrieval systems: approaches, issues, and methods. Ann Rev Inf Sci Technol (ARIST) 32:3–94
8. He K, Fan H, Wu Y et al (2020) Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9729–9738
9. Hu P, Zhu H, Lin J et al (2022) Unsupervised contrastive cross-modal hashing. IEEE Trans Pattern Anal Mach Intell 45(3):3877–3889
10. Huiskes MJ, Lew MS (2008) The MIR flickr retrieval evaluation. In: Proceedings of the 1st ACM international conference on multimedia information retrieval. Association for Computing Machinery, New York, pp 39–43
11. Jang YK, Cho NI (2021) Self-supervised product quantization for deep unsupervised image retrieval. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 12085–12094
12. Jiang QY, Li WJ (2016) Deep cross-modal hashing. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 3270–3278
13. Kim W, Son B, Kim I (2021) ViLT: vision-and-language transformer without convolution or region supervision. In: International conference on machine learning, PMLR, pp 5583–5594
14. Kumar S, Udupa R (2011) Learning hash functions for cross-view similarity search. In: Proceedings of the twenty-second international joint conference on artificial intelligence, IJCAI 11, pp 1360–1365
15. Li Y, Wang Y, Miao Z et al (2020) Contrastive self-supervised hashing with dual pseudo agreement. IEEE Access 8:165034–165043
16. Lin Z, Ding G, Hu M et al (2015) Semantics-preserving hashing for cross-view retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3864–3872
17. Liu S, Qian S, Guan Y et al (2020) Joint-modal distribution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 1379–1388
18. Oord Avd, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748
19. Qiu Z, Su Q, Ou Z et al (2021) Unsupervised hashing with contrastive information bottleneck. arXiv preprint arXiv:2105.06138
20. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
21. Song J, Yang Y, Yang Y et al (2013) Inter-media hashing for large-scale retrieval from heterogeneous data sources. In: Proceedings of the 2013 ACM SIGMOD international conference on management of data, New York, SIGMOD 13, pp 785–796
22. Su S, Zhong Z, Zhang C (2019) Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3027–3035
23. Varshamov RR (1957) Estimate of the number of signals in error correcting codes. Doklady Akademia Nauk Sssr 117(5):739–741
24. Wang H, Xiao R, Li Y et al (2017) PiCO: contrastive label disambiguation for partial label learning. In: International conference on learning representations
25. Wang L, Pan Y, Lai H et al (2022) Image retrieval with well-separated semantic hash centers. In: Asian conference on computer vision
26. Wang L, Pan Y, Liu C et al (2023) Deep hashing with minimal-distance-separated hash centers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 23455–23464
27. Wang X, Zou X, Bakker EM et al (2020) Self-constraining and attention-based hashing network for bit-scalable cross-modal retrieval. Neurocomputing 400:255–271
28. Williams-Lekuona M, Cosma G, Phillips I (2022) A framework for enabling unpaired multi-modal learning for deep cross-modal hashing retrieval. J Imaging 8:328
29. Wu B, Ghanem B (2018) ℓ_p-Box ADMM: a versatile framework for integer programming. IEEE Trans Pattern Anal Mach Intell 41(7):1695–1708
30. Wu G, Lin Z, Han J et al (2018a) 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, pp 2854–2860
31. Wu Z, Xiong Y, Yu SX et al (2018b) Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3733–3742
32. Yang D, Wu D, Zhang W et al (2020) Deep semantic-alignment hashing for unsupervised cross-modal retrieval. In: Proceedings of the 2020 international conference on multimedia retrieval, pp 44–52
33. Yu J, Zhou H, Zhan Y et al (2021) Deep graph-neighbor coherence preserving network for unsupervised cross-modal hashing. In: Proceedings of the AAAI conference on artificial intelligence, pp 4626–4634
34. Zhang D, Li WJ (2014) Large-scale supervised multimodal hashing with semantic correlation maximization. In: AAAI conference on artificial intelligence
35. Zhang J, Peng Y, Yuan M (2018a) Unsupervised generative adversarial cross-modal hashing. In: Proceedings of the AAAI conference on artificial intelligence
36. Zhang Q, Lei Z, Zhang Z et al (2020) Context-aware attention network for image-text retrieval. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3536–3545
37. Zhang X, Lai H, Feng J (2017) Attention-aware deep adversarial hashing for cross-modal retrieval. In: European conference on computer vision
38. Zhang X, Lai H, Feng J (2018b) Attention-aware deep adversarial hashing for cross-modal retrieval. In: Proceedings of the European conference on computer vision (ECCV), pp 591–606
39. Zhen L, Hu P, Wang X et al (2019) Deep supervised cross-modal retrieval. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10394–10403
40. Zhou J, Ding G, Guo Y (2014) Latent semantic sparse hashing for cross-modal similarity search. In: Proceedings of the 37th international ACM SIGIR conference on research development in information retrieval, New York, SIGIR '14, pp 415–424