Class Concentration with Twin Variational Autoencoders for Unsupervised Cross-Modal Hashing

Yang Zhao[1], Yazhou Zhu[1], Shengbin Liao[2], Qiaolin Ye[3], and Haofeng Zhang[1(✉)]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
{zhao_yang,zyz_nj,zhanghf}@njust.edu.cn
[2] National Engineering Research Center for E-learning, Huazhong Normal University, Wuhan 430079, China
[3] School of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China

Abstract. Multi-modal deep hash learning is arguably one of the most commonly used unsupervised methods in cross-modal retrieval tasks. Most existing deep hashing methods focus on maintaining similarity information in the hash code learning step. Although accurate and compact binary representations are learned, these methods fail to encourage discriminative learning of features. In this paper, we propose a new method called Class Concentrated Variational auto-encoder (CCTV) to learn discriminative hash codes. The novelty of CCTV lies in two aspects. First, the proposed method focuses on the concentration of the mean vector of latent features. Based on the assumption that the features in the shared latent space produce multivariate Gaussian, CCTV updates the mean vectors and the cluster centroids of the latent features at the same time by minimizing the class concentration loss, so as to narrow the distance between the cluster centroids and the mean vectors, and further make the concentration more compact. Secondly, under the constraint of raw similarity information, CCTV is different from previous works, it uses the mean vector of latent features as the representation of the images to reduce the influence of variance, and then embeds them in the Hamming space. Our experimental evaluation on four multimedia benchmarks shows a significant improvement over the state-of-the-art methods. Code is available at: https://github.com/theusernamealreadyexists/CCTV.

Keywords: Cross-modal hashing · Visual-text retrieval · Class concentration · Twin variational autoencoder

1 Introduction

The past decades have witnessed the rapid growth of different types of contents on the Internet. The same events or objects can be described as diverse kinds of

L. Wang et al. (Eds.): ACCV 2022, LNCS 13846, pp. 235–251, 2023.
https://doi.org/10.1007/978-3-031-26351-4_15

data which can be referred as multi-modal data with heterogeneous properties. Huge volumes of these multi-modal data affects people's need for information and the ways they search on the Internet. One of the most popular tasks is cross-modal information retrieval, which aims to search relevant data of other different modalities with query data. For instance, using a caption to retrieve the related pictures in database.

Nowadays, cross-modal retrieval has attracted growing attention from researchers. The most difficult problem of cross-modal retrieval is how to measure the similarity between different modal features of data, which is known as heterogeneity gap. In order to support similarity relationship search, it is necessary to map the incomparable data into comparable features. Hence, learning representations for multi-modal data is considered as the fundamental step to extract features of various modalities. As proposed in [24], the main research effort is to design compact and accurate representations. During the learning process of representations modelling, the features of various modalities are mapped to so-called common latent embedding space, where the features of same object or event are pulled together and those of different objects or events are pushed away on the Euclidean distance basis. The challenge of learning accurate representation lays in deciding the correlation between two modalities. Intuitively, the learned feature is explicitly encouraged to maximize intra-class compactness and inter-class separability. What's more, the key problem for compactness, which makes the stage of representation succinct, is dependent on the dimension and discreteness of multi-modal features.

Hashing technology, which encodes continuous real-valued features into latent hash space, where relative samples have similar binary codes, is widely used in cross-modal retrieval due to its few storage, low Hamming distance computational complexity and fast retrieval speed. Motivated by hashing technology, [7,17,46,47] incorporate deep learning with hashing method and learn accurate and compact representations for multi-modal information. These methods have a common module called two-stream network which designs two networks for visual and textual data respectively. Supervised approaches [1,2,18,23,26,28,29,36,39,42] intuitively can perform better than unsupervised methods due to the constraint of labelled information in training step. However, labelled information is expansive and further limited in real world large scale retrieval application right now. Thus, it is realistic to pay attention to unsupervised hashing algorithms.

To date, pairwise similarity based unsupervised cross-modal hashing (UCMH) methods can achieve better performance than those methods directly embeds high-dimension feature into Hamming space, they preserve pairwise information to construct similarity constraint. Some of these approaches preserve the similarity information through graph structure [10,38,43,51]. Although these related works achieve breakthrough, there still exists two main problems in this task. Firstly, dense graph that basically contains pre-defined local neighbourhood information in a mini-batch get much redundant information, which means most of the graph neighbourhoods are useless and mislead the neighbour-

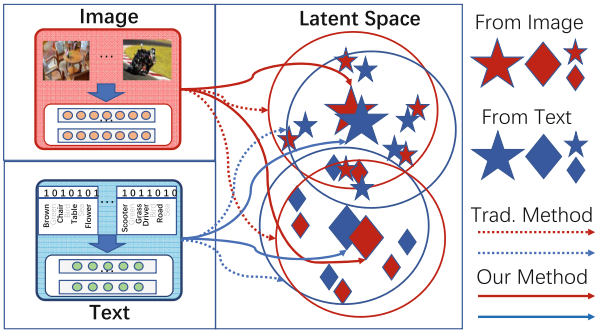

Fig. 1. Illustration of the difference between our method and traditional methods. Traditional methods try to project images and text directly into latent space, while our method projects them into their corresponding class centroids in latent space.

hood relationship in common Hamming space and consequently learn redundant hash codes, for example in Fig. 1, previous methods tries to directly map the features into latent or hamming space, where the variance of the features make the embeddings hard to separate. Secondly, previous methods fail to model the posterior distribution of the observed data and only adopt similarity information during training.

In light of these issues, we propose a novel unsupervised cross-modal hashing method called Class Concentration with Twin Variational autoencoders (CCTV). We train twin Variational Auto-Encoders (VAEs) models to encode and decode visual and textual modal features respectively. The given multi-modal data produces a distribution over the possible values of the latent features, and we directly concentrate the mean of the latent features with Deep Embedded Clustering (DEC) [44] method for updating the clustering centers and mapping at the same time. We align the distribution of two modalities by enforcing the mean of multi-modal data from the same cluster to produce the same posterior distribution. Consequently, by explicitly enforcing both the distribution of arithmetic mean of latent features and the distance between data point and each clustering center, the objects of same construction share the matched inter-modal distribution in common latent space, which generates much more accurate latent representation. Then we train a deep network to learn binary codes of latent features and minimize the reconstruction loss to learn compact representation. In general, our main contributions are as follows:

- We propose a novel deep learning framework that learns compact and accurate hash representation of multi-modal information via twin VAE models, which creatively align the mean vectors of each modality in latent space. This operation can circumvent the interference by the variances from the different features although in a same class.
- We approximate the intractable true distributions of inter-class and intra-class for class construction and jointly optimize the deep feature embedding and mean vector clustering.

– we have conducted extensive experiments on four popular datasets, and the results show that our method can achieve state-of-the-art performance.

2 Related Works

Due to its low computational complexity and fast retrieval speed, cross-modal hashing has attracted an increasing attention. It aims to mine the relationship between visual and textual modalities and embed data into common Hamming space. Similar to real-valued alternatives [5,16,30,52], cross-modal hashing methods can be also simply categorized into supervised methods [13,18,23,28,29,36,39]. and unsupervised methods. Since our method focuses on the unsupervised one, we only briefly introduce some related unsupervised methods in the following.

A large amount of unsupervised cross-modal hashing [12,13,21,34,43,48,51] have been proposed in the past few years. The earlier shallow schemes, *e.g.*, both Cross-view hashing (CVH) [20] and Inter-Media Hashing (IMH) [37], can be regarded as the extension of Spectral Hashing [41] from single-modal hashing to cross-modal hashing scenario. These methods learn hash functions by solving the eigenvalue decomposition with constructed similarity graph. Zhai *et al.* [49] presented the parametric local multi-modal hashing (PLMH), which designs a set of hashing function to generate several hashing space and accesses to non-linear global transformation. Ding *et al.* [8] employed matrix factorization methods and proposed Collective Matrix Factorization Hashing (CMFH), which bridges the modality gap by embedding different modal information into a latent common space. Zhou *et al.* designed Latent Semantic Sparse Hashing (LSSH) that extends CMFH in the manner of utilizing sparse coding in extracting latent feature process at the same time and restricts hash code learning subsequently. However, above shallow methods are week to extract the non-linear relevant information from different modalities for using hand-crafted features. As the progress of deep neural networks have made in exploring non-linear relationships, many methods [7,17,46,47] capture more semantic relevant features during binary code learning process. Most of them utilize similarity graphs generated from intrinsic data directly and obtain superior performances. Wang *et al.* [40] added an orthogonal regularizer to make the representation compact and accurate. [11] utilizes the adaptive tanh function which has concise derivation and can be used in objective function directly. [43] makes use of the matrix factorization with Laplacian constraint in training process to constraint the hash code generation, which consequently preserves the neighbour affinity information of original features in their own space.

Though impressive progress has been made by these models, there are still a few challenges to be solved that are mentioned in Sect. 1. In this paper, we focus on improving the retrieval performance of unsupervised deep cross-modal hashing. With the intention to model the posterior distribution of the observed data from both visual and textual modalities, we concentrate the mean of the latent

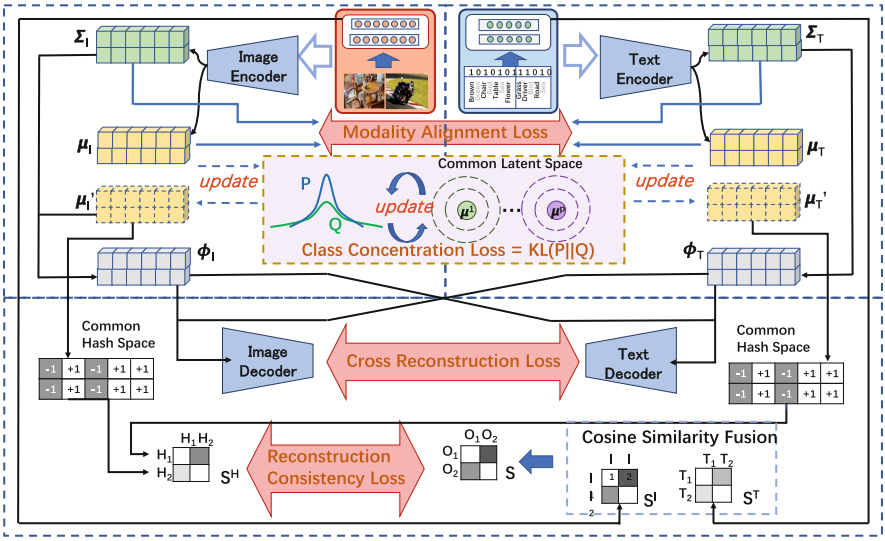

Fig. 2. The architecture of the proposed CCTV. The twin variational auto-encoder project both image and text into a common latent space, where the embeddings are aligned by their mean vectors and class discriminative information is raised by class concentration loss. Cross reconstruction and cross consistency are employed to constrain images and texts to have both their own semantic information and the semantic information of each other.

features with DEC clustering method which optimize the clustering centers iteratively, which tackles the problem of lacking label information and subsequently generate accurate representations.

3 Methodology

3.1 Preliminaries

We first introduce several definitions in our methods. With n equals to the amount of instances in each batch, the visual and textual features in each batch are denoted as $\boldsymbol{X}_I \in \mathbb{R}^{n \times d_I}$ and $\boldsymbol{X}_T \in \mathbb{R}^{n \times d_T}$ respectively. Here d_I and d_T represent the dimensions of image and caption features respectively. Furthermore, we aim to generate binary hash codes $\boldsymbol{B_I}$ and $\boldsymbol{B_T}$ by embedding continuous features into common latent hash space, where $\boldsymbol{B}_H \in \mathbb{R}^{n \times b}, (H \in \{I, T\})$ and b means hash code length. If two objects o_1 and o_2 are semantic similar, the hash codes generated by their features should be within a small Hamming distance.

Modelling the posterior distribution of the observed data can improves the performance of retrieval task. However, the lacking of labelled information before generating binary codes is not conducive to the construction of a prior constraints. Previous methods can be grouped into two categories in terms of how

to conduct feature embedding. The first category methods, such as [49], preserve the affinity information of original features and use them to learn hash codes directly. They share the following common quantification loss function:

$$\begin{aligned} \mathcal{L}_q &= \|f_I(\boldsymbol{I}) - \boldsymbol{B}_I\|_F^2 + \|f_T(\boldsymbol{T}) - \boldsymbol{B}_T\|_F^2, \\ s.t. \boldsymbol{B}_H &\in \{+1, -1\}^{m \times l}, \boldsymbol{B}_H^T \boldsymbol{B}_H = m\boldsymbol{I}, \end{aligned} \tag{1}$$

where $f_I(\cdot)$ and $f_T(\cdot)$ are the embedding functions for visual and textual data respectively and $H \in \{I, T\}$. Equation (1) aims to reduce the gap between features and hash codes. The auxiliary constraint $\boldsymbol{B}_g^T \boldsymbol{B}_g = m\boldsymbol{I}$ aims to generate mutually independent hash codes.

Evolved from the first category, the second type of methods, such as [35], typically generate clustering centers with the method of deep clustering in common latent space, and further update the latent embedding. Both the design of construing matrices and the strategy of employing the matrices in training stage have an impact on the final performance. To be specific, the loss functions of these algorithms (optimizing objectives) are typically composed of two parts: $\mathcal{L}_q$ and clustering loss $\mathcal{L}_c$, the loss function can be formulated as follows:

$$\mathcal{L} = \lambda \mathcal{L}_q + (1 - \lambda)\mathcal{L}_c, \quad s.t. \lambda \in [0, 1], \tag{2}$$

where λ is a hype-parameter to balance $\mathcal{L}_q$ and $\mathcal{L}_c$.

The goal of our model is to learn accurate and compact binary representations in a shared latent Hamming space for a combination of two modalities data. The basic module of CCTV is the VAE [19], which introduces a recognition model $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ that is an approximation to the true posterior $P_\theta(\boldsymbol{z}|\boldsymbol{x})$, where $\boldsymbol{x}$ means original data point and $\boldsymbol{z}$ is the unobserved latent variable and produces the prior distribution. VAE approximates the prior over the latent variables to be the multivariate Gaussian $P_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$ and let the variational approximate posterior also be a multivariate Gaussian:

$$\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) = log\mathcal{N}\left(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \boldsymbol{I}\right), \tag{3}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is the mean and standard deviation of the approximate posterior, respectively.

From the perspective of coding theory, they are generated by non-linear encoder. Furthermore, the latent variable $\boldsymbol{z}$ is sampled using $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is an auxiliary variable with independent margin and $\odot$ means element-wise product. To learn the recognition model parameters $\boldsymbol{\phi}$ and generative the model parameters $\boldsymbol{\theta}$ simultaneously, the estimator can be written as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) = D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}(\boldsymbol{z})}) - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})], \tag{4}$$

where the first item is the Kullback-Leibler (KL) divergence between intractable true posterior and its approximation, and the second item is the lower bound on the marginal likelihood of data point.

3.2 Proposed Architecture

The overall pipeline of CCTV is shown in Fig. 2. We design a twin-VAE model to learn the conditional probability distributions over the latent variables, $q_{\phi_I}(\boldsymbol{z}|\boldsymbol{x}_I)$ for visual features and $q_{\phi_T}(\boldsymbol{z}|\boldsymbol{x}_T)$ for textual features using approximate posteriors, $q_{\theta_I}(\boldsymbol{z}|\boldsymbol{x}_I)$ and $q_{\theta_T}(\boldsymbol{z}|\boldsymbol{x}_T)$. The embeddings of data points from different modalities are supposed to share a common latent space.

Cross Reconstruction (CR) Loss. To minimize the information gap between the original data and latent feature in each modal, the twin-VAE reconstruction loss should include two single VAE (SV) loss for each modality:

$$\mathcal{L}_{SV} = \sum_{H\in\{I,T\}} D_{KL}(q_{\phi_H}(\boldsymbol{z}|\boldsymbol{x}_H)||p_{\boldsymbol{\theta}_H(\boldsymbol{z})}) - \mathbb{E}_{q_{\phi_H}(\boldsymbol{z}|\boldsymbol{x}_H)}[logp_{\boldsymbol{\theta}_H}(\boldsymbol{x}_H|\boldsymbol{z})], \tag{5}$$

where $\boldsymbol{x}_H$ means a specific original feature in $\boldsymbol{X}_H$.

In addition to using single modal information reconstruction loss to constrain the latent space embedding process of data, it is also necessary to consider the alignment between different-modal data. That means using the modality-specific recognition model to approximate the true posterior in another modality. To be specific, the cross VAE (CV) loss is formulated as following:

$$\begin{aligned}\mathcal{L}_{CV} =& D_{KL}(q_{\phi_I}(\boldsymbol{z}|\boldsymbol{x}_I)||p_{\boldsymbol{\theta}_T(\boldsymbol{z})}) - \mathbb{E}_{q_{\phi_I}(\boldsymbol{z}|\boldsymbol{x}_I)}[logp_{\boldsymbol{\theta}_T}(\boldsymbol{x}_T|\boldsymbol{z})] \\ &+ D_{KL}(q_{\phi_T}(\boldsymbol{z}|\boldsymbol{x}_T)||p_{\boldsymbol{\theta}_I(\boldsymbol{z})}) - \mathbb{E}_{q_{\phi_T}(\boldsymbol{z}|\boldsymbol{x}_T)}[logp_{\boldsymbol{\theta}_I}(\boldsymbol{x}_I|\boldsymbol{z})].\end{aligned} \tag{6}$$

In the view of coding theory, the unobserved variables $\boldsymbol{z}$ is represented as a code with specific length. Given several data samples, they produce a possible distribution over the possible latent variables. Thus, same as VAE [19], we refer to the true conditional probability distributions over the latent variables, $q_{\phi_I}(\boldsymbol{z}|\boldsymbol{x}_I)$ and $q_{\phi_T}(\boldsymbol{z}|\boldsymbol{x}_T)$, as encoders and realize $q_{\theta_I}(\boldsymbol{z}|\boldsymbol{x}_I)$ and $q_{\theta_T}(\boldsymbol{z}|\boldsymbol{x}_T)$ with decoders. Since given an unobserved variable, it generates a corresponding distribution over the value of original data point. Here we combines the aforementioned object loss function and term them as cross reconstruction (CR) loss:

$$\mathcal{L}_{CR} = \mathcal{L}_{SV} + \mathcal{L}_{CV}. \tag{7}$$

Class Concentration (CC) Loss. Only reducing information loss can not generate discriminative latent representation. Thus, we propose a class concentration loss function. From the perspective of metric learning, after specifying the distance metric method, high-quality clustering results should narrow the gaps within the classes, and widen the gaps between the classes. This phenomenon shows that the data in each cluster has its own unique distribution. However, earlier deep unsupervised methods directly cluster data and did not consider the prior and posterior distribution of the data. Suppose that latent variables produce centered isotropic multivariate Gaussian distributions [19], we consider that each cluster refer to a unique Gaussian. Accordingly, the latent features in a cluster should share the same mean $\boldsymbol{\mu}$ which has an interpretation as a vector.

Thus, the proposed method (CCTV) clusters the mean vectors $\{\boldsymbol{\mu}_i\}_{i=1}^n$ of latent features in order to generate latent features which are compact within a class and scattered between classes. For n latent points $\{\boldsymbol{z}_i\}_{i=1}^n$ with k clusters $\{\boldsymbol{c}_j\}_{j=1}^k$ in latent space, we are supposed to learn k clustering centers and update θ_I and θ_T, which are encoder learning parameters in an end-to-end fashion. To achieve this goal, we utilize the auxiliary target distribution mentioned in DEC [44]. The construction can be described in two steps. First, we assign a distribution for measuring the distance between mean vector of latent feature $\boldsymbol{\mu}_i$ and cluster centroid $\boldsymbol{c}_j$. Second, we calculate the KL divergence loss to update encoder parameters and cluster centers.

To be specific, we adopt t-distribution to measure the distance between mean vector of embedded feature $\boldsymbol{\mu}_i$ and cluster centroid $\boldsymbol{c}_j$, this step is formulated as:

$$\boldsymbol{A}_{ij} = \frac{\left(1 + \left\|\boldsymbol{\mu}_i - \boldsymbol{c}_j\right\|^2 / \alpha\right)^{-(\alpha+1)/2}}{\sum_h \left(1 + \|\boldsymbol{\mu}_i - \boldsymbol{c}_h\|^2 / \alpha\right)^{-(\alpha+1)/2}}, \tag{8}$$

where α is the degree of freedom of t-distribution. Since $\boldsymbol{A}_{ij}$ give a distance measuring method, A_i can be regraded as a soft assignment. For instance, if $\boldsymbol{A}_{ij}$ has the largest value among other scalars over $\boldsymbol{A}_i$, it means the possibility of $\boldsymbol{\mu}_i$ being assigned to cluster center $\boldsymbol{c}_j$ is the biggest.

Then, we construct auxiliary target distribution which can be written as following according to DEC:

$$\boldsymbol{B}_{ij} = \frac{\boldsymbol{A}_{ij}^2 / \boldsymbol{d}_j}{\sum_h \boldsymbol{A}_{ih}^2 / \boldsymbol{d}_h}, \tag{9}$$

where $\boldsymbol{d}_j = \sum_i \boldsymbol{A}_{ij}$. This auxiliary target distribution can strengthen predictions and emphasize features with high confidence. What's more, the loss contribution of each centroid is standardized to prevent a large number of categories from distorting the hidden space. we try to refine the centroids by keeping cluster assignment distribution close to auxiliary target distribution. Thus, we adopt KL divergence loss as class concentration loss (CC) to reduce the distance between two distribution:

$$\mathcal{L}_{CC} = KL\left(\boldsymbol{A} \| \boldsymbol{B}\right) = \sum_i \sum_j \boldsymbol{A}_{ij} \left(\log \boldsymbol{A}_{ij} - \log \boldsymbol{B}_{ij}\right), \tag{10}$$

so that we can cluster the mean vectors of latent variables to k points. As a result, latent features of same cluster share concentrated mean vector and still keep fine distribution characters with variance.

Modality Alignment (MA) Loss. The projected text and image also need to be matched in the latent space. Here, inspired by the concept of alignment for attributes and visual features in zero shot learning [33], we simultaneously align the mean vector and variance of VAE, and define the following modality alignment loss:

$$\mathcal{L}_{MA} = \frac{1}{n} \sum_{i=1}^{n} (|\boldsymbol{\mu}_{Ii} - \boldsymbol{\mu}_{Ti}\|_2^2 + \|\Sigma_{Ii}^{\frac{1}{2}} - \Sigma_{Ti}^{\frac{1}{2}}\|_F^2)^{\frac{1}{2}}, \tag{11}$$

where, $\boldsymbol{\mu}_{Ii}$ and $\boldsymbol{\mu}_{Ti}$ represent the mean vectors projected from i-th pair of image and text respectively in a mini batch. Similarly, Σ_{Ii} and Σ_{Ti} stand for the corresponding variance matrices.

Reconstruction Consistency (RC) Loss. In this subsection, we utilize the original semantic matrices $\boldsymbol{S}_I$ and $\boldsymbol{S}_T$, which represents the original affinity relations of the input instances, to restrict the generation stage of hash code. Since it is hard to measure the distance between continuous feature and hash codes referring to Eq. (1), we consider to preserve the information of latent continues features indirectly by reducing the loss of information between the original information and the hash code in the manner of structuring affinity matrices. We calculate the similarity matrices in a mini batch from raw visual and textual modalities as:

$$\boldsymbol{S}_{H(ij)} = \frac{\boldsymbol{X}_{H(i)}(\boldsymbol{X}_{H(j)})^T}{\left\|\boldsymbol{X}_{H(i)}\right\| \left\|\boldsymbol{X}_{H(j)}\right\|}, \tag{12}$$

where $H \in \{I, T\}$. Furthermore, we adopt manner of DJSRH [38] to get hybrid semantic affinity matrix $\boldsymbol{U} = f(\boldsymbol{S}_I, \boldsymbol{S}_T)$. To be concrete, $\boldsymbol{S}_I$ and $\boldsymbol{S}_T$ are merged in a trade-off method:

$$\boldsymbol{S} = \omega \boldsymbol{S}_I + (1 - \omega)\boldsymbol{S}_T, \tag{13}$$

where $\omega \in [0, 1]$ is the weight of two affinity matrices. This algorithm coincides with the diffusion method in [9] which provides powerful evidence of effectiveness. Then, second order neighbourhood information is structured by $\boldsymbol{S}\boldsymbol{S}^T$. Finally, similarity information across original affinity structure in two modalities is combined by the following manner:

$$\boldsymbol{U} = \gamma \boldsymbol{S} + (1 - \gamma)\frac{\boldsymbol{S}\boldsymbol{S}^T}{n}. \tag{14}$$

In latent Hamming space, relevant vertices have small Hamming distance. Thus, hash codes can be understood as discrete features. earlier unsupervised cross-modal algorithms directly generate hash codes using sign function with latent features. However, it is impossible to derive the result of sign function with respect to the input. Thus, we follow [3,11,38] and take a scaled tanh activation function into consideration:

$$\boldsymbol{b}_i = tanh(\kappa \boldsymbol{\mu}_i) \in [-1, +1]^{m \times d}, \kappa \in \mathbb{R}^+, \tag{15}$$

where κ is an auto-increasing parameter during training. With the increasing of κ, the result of Eq. (15) is close to the sign function and approximates the binary value of input feature.

Different from previous methods, CCTV is the first to embed the mean vectors μ to hash codes as far as we know. The purpose of hash learning is to map the continuous features into Hamming space where the relevant object share small Hamming distance. However, the noise of features in the original continuous space is harmful to generate concentrated distribution of data points.

Thus, we can narrow the binary features within same cluster in Hamming space by removing the noise of data points between the original continuous space. Since we utilize multivariate Gaussian as the prior distribution in latent space, the distance between data point and cluster centroid can be regarded as noise. Accordingly, it is beneficial to choose the mean of feature as the input of tanh function. Then, to calculate the similarity with neighbourhoods in Hamming space, the similarity function is defined as:

$$\mathcal{Z}(\boldsymbol{B}_{I(i)}, \boldsymbol{B}_{T(j)}) = \frac{\boldsymbol{B}_{I(i)}(\boldsymbol{B}_{T(j)})^T}{\left\|\boldsymbol{B}_{I(i)}\right\| \left\|\boldsymbol{B}_{T(j)}\right\|}, \tag{16}$$

where $\boldsymbol{B}_{I(i)}$ means the i-th row in $\boldsymbol{B}_I$ and $\boldsymbol{B}_{T(j)}$ means the j-th row in $\boldsymbol{B}_T$. The result of Eq. (16) is the cosine affinity score which represents the angular connection among discrete features. Minimizing the reconstruction error between the similarity matrix of hash code and the affinity matrix $\boldsymbol{U}$ of continuous features keeps their similarity consistency. Therefore, we define the formulation of reconstruction consistency (RC) Loss as the following manner:

$$\mathcal{L}_{RC} = \left\|\beta \boldsymbol{U} - \mathcal{Z}(\boldsymbol{B}_i^H, \boldsymbol{B}_j^H)\right\|_F^2, \tag{17}$$

where β is a trade-off parameter which makes reconstruction more flexible, referring to [38]. For instance, supposed that $U_{ij} = 0.7$, which means that ith instance and jth instance got 0.7 similarity score, then the similarity score of corresponding hash codes calculated from Hamming space need to be close to 0.7. $\beta > 1$ means the similarity score of hash codes pair need to lager than 0.7 and thus make the nodes in Hamming space compact, while $\beta < 1$ means the similarity score of hash codes pair need to smaller than 0.7 and accordingly make the nodes in Hamming space sparse. We empirically find that it is beneficial to set $\beta > 1$. And this phenomenon can be attributed to the fact that cosine similarity measures the similarity between two vectors by measuring the cosine of the angle between them.

Consequently, we provide our loss function of CCTV:

$$\mathcal{L} = \mathcal{L}_{SV} + \epsilon_1 \mathcal{L}_{CV} + \epsilon_2 \mathcal{L}_{CC} + \epsilon_3 \mathcal{L}_{MA} + \epsilon_4 \mathcal{L}_{RC}, \tag{18}$$

where ϵ_1, ϵ_2, ϵ_3 and ϵ_4 are hyper parameters to balance the total loss.

4 Experiments

4.1 Datasets

WIKI [31]: This dataset consists of 2,866 samples in total with 10 classes. Each image is described by a paragraph which represents related image, from 1 to 10. In our experiment, we randomly select 500 samples from the total dataset as the query set, and the remaining samples form the training set are composed as the retrieval database.

NUS-WIDE [6]: It consists of 269,648 multi-modal instances, each of which contains an image and the related captions with 81 class labels. Following previous methods, the top 10 largest categories is selected and totally contain over

Table 1. The mAP@all results on image query text ($I \rightarrow T$) and text query image ($T \rightarrow I$) retrieval tasks at various encoding lengths and datasets. The best performances are shown as bold. In this table, '*' on the right of methods' names means the scores are according to results in their own paper, and '–' means the score is not reported.

Task	Method	WIKI			MIRFlicker-25K			MSCOCO			NUS-WIDE		
		16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit
$I \rightarrow T$	CVH [20]	0.157	0.144	0.131	0.579	0.565	0.565	0.499	0.471	0.370	0.400	0.381	0.370
	CMFH [8]	0.173	0.169	0.184	0.580	0.572	0.554	0.442	0.423	0.492	0.381	0.429	0.416
	PDH [32]	0.196	0.168	0.184	0.544	0.544	0.545	0.442	0.423	0.492	0.368	0.368	0.368
	ACQ [15]	0.126	0.120	0.115	0.617	0.594	0.578	0.559	0.552	0.514	0.440	0.416	0.395
	IMH [37]	0.151	0.145	0.133	0.557	0.565	0.559	0.416	0.435	0.442	0.349	0.356	0.370
	QCH [42]	0.159	0.143	0.131	0.579	0.565	0.554	0.496	0.470	0.441	0.401	0.382	0.370
	UCH* [22]	–	–	–	0.654	0.669	0.679	0.447	0.471	0.485	–	–	–
	DJSRH [38]	0.274	0.304	0.350	0.649	0.662	0.669	0.561	0.585	0.585	0.496	0.529	0.528
	DGCPN [48]	0.226	0.326	0.410	0.651	0.670	0.702	0.469	0.586	0.630	0.517	0.553	0.567
	DSAH [45]	0.249	0.333	0.381	0.654	0.693	0.700	0.518	0.595	0.632	0.539	0.566	0.576
	JDSH [27]	0.253	0.289	0.325	0.665	0.681	0.697	0.571	0.613	0.624	0.545	0.553	0.572
	CCTV	**0.405**	**0.409**	**0.413**	**0.690**	**0.701**	**0.716**	**0.604**	**0.640**	**0.645**	**0.548**	**0.569**	**0.580**
$T \rightarrow I$	CVH [20]	0.342	0.299	0.245	0.584	0.566	0.566	0.507	0.479	0.446	0.405	0.384	0.372
	CMFH [8]	0.176	0.170	0.179	0.583	0.566	0.556	0.453	0.435	0.499	0.394	0.451	0.447
	PDH [32]	0.344	0.293	0.251	0.544	0.544	0.546	0.437	0.440	0.440	0.366	0.366	0.367
	ACQ [15]	0.344	0.291	0.247	0.628	0.601	0.580	0.565	0.561	0.520	0.445	0.419	0.398
	IMH [37]	0.236	0.237	0.218	0.560	0.569	0.563	0.560	0.561	0.520	0.350	0.356	0.371
	QCH [42]	0.341	0.289	0.246	0.585	0.567	0.556	0.505	0.478	0.445	0.405	0.385	0.372
	UCH* [22]	–	–	–	0.661	0.667	0.668	0.446	0.469	0.488	–	–	–
	DJSRH [38]	0.246	0.287	0.333	0.658	0.660	0.665	0.563	0.577	0.572	0.499	0.530	0.536
	DGCPN [48]	0.186	0.297	0.522	0.648	0.676	0.703	0.474	0.594	0.634	0.509	0.556	0.574
	DSAH [45]	0.249	0.315	0.393	0.678	0.700	0.708	0.533	0.590	0.630	0.546	0.572	0.578
	JDSH [27]	0.256	0.303	0.320	0.660	0.692	0.710	0.565	0.619	0.632	0.545	0.566	0.576
	CCTV	**0.535**	**0.557**	**0.564**	**0.679**	**0.703**	**0.714**	**0.615**	**0.654**	**0.662**	**0.549**	**0.574**	**0.584**

186 thousand instances and randomly choose 2,000 from them as query set, and employ the others as retrieval database.

MIRFlickr-25K [14]: The original training set and validation set contains more than 25 thousand samples from 38 categories. The class labels are represented as one-hot form where 1 represents the image belongs to this class while 0 is the opposite. We randomly choose 1,000 samples as the query set and set the others as the retrieval set.

MSCOCO [25]: The dataset contains more than 123 thousand images-caption pairs from real-world with 80 class labels. We randomly choose 2,000 from them as query set and the others as retrieval database.

4.2 Evaluation Metrics

To evaluate the efficiency of our method and the baseline approaches, we employ several frequently used evaluation metrics:

Mean Average Precision (mAP): mAP is a metric for evaluating the retrieval performance and its formal definition can be found in [50]. In addition, the

performance of all baselines and the proposed method are evaluated on 16 bit, 32 bit and 64 bit hash codes.

Precision-Recall (P-R curve): This curve shows the precision and recall rates according to the retrieved images. It is worthy noting that the beginning plot of curve means the precision and recall rate of the retrieval under the condition that the binary codes of both query and returned items are the same.

4.3 Implementation Details

Our experiments follow previous methods to employ the fc7 layer of VGG-16 to extract the 4,096-dimensional deep features $\boldsymbol{X}_I \in \mathbb{R}^{n\times 4096}$ from original images, while for original textual features we utilize the universal sentence encoder [4] to represent final textual features $\boldsymbol{X}_T$ whose dimension is 512. It is worth noting that to calculate the consistency loss as the manner of Eq. (17), we need to force the items in the same ranges. However, the cosine similarity ranges from -1 to $+1$, while the affinity value elements in $\boldsymbol{U}$ are non-negative, which can be obtained by Eq. (12) and Eq. (14). Therefore, we refine the $\boldsymbol{S}_H$ with $\boldsymbol{S}_H \leftarrow 2\boldsymbol{S}_H - 1, H \in \{I, T\}$. Additionally, we fix the batch size as 8 and employ the SGD optimizer with 0.9 momentum and 0.0005 weight decay. We experimentally take $\alpha = 1$, $\omega = 0.5$, $\gamma = 0.6$ and $\beta = 1.5$ for all four datasets. Then we set $\lambda = 0.6$, $\epsilon = \epsilon_2 = \epsilon_3 = \epsilon_4 = 0.1$ for NUM-WIDE, $\lambda = 0.9$, $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon_4 = 0.1$ for MiRFlickr, $\lambda = 0.3$, $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon_4 = 0.3$ for WIKI and $\lambda = 0.6$, $\epsilon_1 = \epsilon_2 = \epsilon_3 = \epsilon_4 = 0.1$ for MSCOCO.

4.4 Retrieval Performance

To evaluate the performance of the proposed method, we compare our CCTV with several recent competing methods, and record the result in Table 1. The results of the compared methods are obtained by using the codes released by themselves or reproduced according to the settings introduced in their original papers. It can be seen from the table that the proposed method can achieve satisfactory retrieval results on the four data sets. No matter the code length is 16, 32 or 64 bits, the performance of CCTV is higher than all other methods, especially on WIKI and MSCOCO. Specifically, the performance of CCTV's image retrieval text task on WIKI can be improved by about 20% compared with those unsupervised non-depth algorithms (the first six rows in Table 1). At the same time, the retrieval performance improves with the increase of the binary hash code length, which reflects another advantage of this method, that is, the more information the model obtains, the better the retrieval effect. This phenomenon demonstrates that the effective training method of the CCTV model reduces the loss caused by the lack of label information, so that the retrieval performance of the model still has certain advantages compared with other benchmark methods.

In addition, we also employ the P-R curves to evaluate the proposed method compared with other baselines. We choose a small-scale dataset WIKI and a large-scale dataset NUS-WIDE to illustrate the performance, and draw the P-R

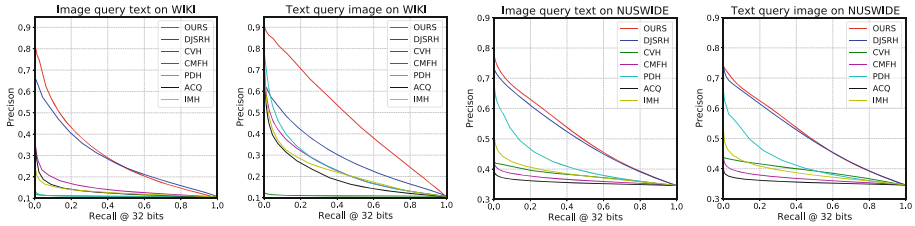

Fig. 3. P-R curves compared with other methods on NUS-WIDE and WIKI for 32 bits.

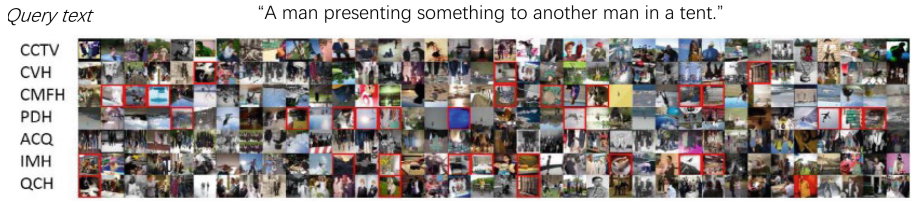

Fig. 4. Visualization of top 36 retrieved images by textual query on MSCOCO Dataset with random query text written on the top through. Returned samples with red boxes are false-positive candidates. (Color figure online)

curves of the 32-bit hash codes generated by different models. Figure 3 shows the result curves of image retrieving text (I2T) and text retrieving image (T2I). As can be seen from the figure, for the generated 32-bit hash code, the curves of our method lie high above those of the other methods, which means that our CCTV model can achieve satisfactory results on both datasets.

4.5 Visualization

In this subsection, we visually demonstrate the performance of our proposed method by using text retrieve images. We randomly select a query text from MSCOCO as an example, and display the top 36 retrieved images and visualize them in Fig. 4. Among the first 36 returned images, our method can obtain all the correct images based on the query, while at least one of the retrieved results from other methods is wrong. At the same time, it can be found that these incorrect returned results usually have a shape or color similar to the correct retrieved results, which means that they preserve too much redundant information from the training samples.

4.6 Ablation Study

To verify whether the proposed several modules are effective for improving the final performance, in this subsection we conduct ablation studies by removing them and record the experimental results. Since some modules are the core part

Table 2. The ablation studies of three proposed modules on WIKI and MIRFlickr.

Task	Modules	WIKI			MIRFlickr		
		16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
$I \rightarrow T$	w./o. $\mathcal{L}_{CV}$	0.372	0.399	0.408	0.672	0.686	0.709
	w./o. $\mathcal{L}_{CC}$	0.386	0.404	0.410	0.673	0.692	0.709
	With All	0.405	0.409	0.413	0.690	0.701	0.716
$T \rightarrow I$	w./o. $\mathcal{L}_{CV}$	0.502	0.536	0.558	0.651	0.666	0.675
	w./o. $\mathcal{L}_{CC}$	0.531	0.553	0.561	0.666	0.679	0.680
	With all	0.535	0.557	0.564	0.679	0.703	0.714

of this method, such as modality alignment and reconstruction consistency, they cannot be removed. The verified modules are cross reconstruction module and the class concentration module, and we record the results of both $I \rightarrow T$ and $T \rightarrow I$ for 16, 32, 64 bits on WIKI and MIRFlicker in Table 2. This phenomenon reveals that the twin VAE module with reconstruction is dominant for the performance, and the final learned model is relatively close to only twin VAE. In addition, we can also find that the cross reconstruction and class concentration play very important roles.

5 Conclusion

In this paper, we have proposed a class concentration twin variational autoencoder to solve the problem of insufficient separability of hash codes in unsupervised cross-modal retrieval. A Twin VAE network is designed to generate the latent mean vector and variance, which are subsequently clustered by employing the class concentration loss to improve the degree of discrimination. In addition, reconstruction consistency loss is also applied to keep the graph similarity between hash codes and original features. Extensive experiments on four popular datasets are conducted and the results demonstrate that our method can achieve state-of-the-art performance. The ablation studies also verify that each module designed in this method contributes to the final performance.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants No. 61872187, No. 62077023 and No. 62072246, in part by the Natural Science Foundation of Jiangsu Province under Grant No. BK20201306, and in part by the "111" Program under Grant No. B13022.

References

1. Bronstein, M.M., Bronstein, A.M., Michel, F., Paragios, N.: Data fusion through cross-modality metric learning using similarity-sensitive hashing. In: CVPR, pp. 3594–3601 (2010)
2. Cao, Y., Long, M., Wang, J., Yu, P.S.: Correlation hashing network for efficient cross-modal retrieval. In: BMVC (2017)
3. Cao, Z., Long, M., Wang, J., Yu, P.S.: Hashnet: deep learning to hash by continuation. In: ICCV, pp. 5608–5617 (2017)
4. Cer, D., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
5. Chen, H., Ding, G., Liu, X., Lin, Z., Liu, J., Han, J.: IMRAM: iterative matching with recurrent attention memory for cross-modal image-text retrieval. In: CVPR, pp. 12655–12663 (2020)
6. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of Singapore. In: ICIVR, pp. 1–9 (2009)
7. Deng, C., Chen, Z., Liu, X., Gao, X., Tao, D.: Triplet-based deep hashing network for cross-modal retrieval. IEEE TIP **27**(8), 3893–3903 (2018)
8. Ding, G., Guo, Y., Zhou, J.: Collective matrix factorization hashing for multimodal data. In: CVPR, pp. 2075–2082 (2014)
9. Donoser, M., Bischof, H.: Diffusion processes for retrieval revisited. In: CVPR, pp. 1320–1327 (2013)
10. Gu, Y., Wang, S., Zhang, H., Yao, Y., Yang, W., Liu, L.: Clustering-driven unsupervised deep hashing for image retrieval. Neurocomputing **368**, 114–123 (2019)
11. Hu, D., Nie, F., Li, X.: Deep binary reconstruction for cross-modal hashing. IEEE TMM **21**(4), 973–985 (2018)
12. Hu, H., Xie, L., Hong, R., Tian, Q.: Creating something from nothing: unsupervised knowledge distillation for cross-modal hashing. In: CVPR, June 2020
13. Hu, M., Yang, Y., Shen, F., Xie, N., Hong, R., Shen, H.T.: Collective reconstructive embeddings for cross-modal hashing. IEEE TIP **28**(6), 2770–2784 (2018)
14. Huiskes, M.J., Lew, M.S.: The MIR Flickr retrieval evaluation. In: ACM MM, pp. 39–43 (2008)
15. Irie, G., Arai, H., Taniguchi, Y.: Alternating co-quantization for cross-modal hashing. In: CVPR, pp. 1886–1894 (2015)
16. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. arXiv preprint arXiv:2102.05918 (2021)
17. Jiang, Q.Y., Li, W.J.: Deep cross-modal hashing. In: CVPR, pp. 3232–3240 (2017)
18. Jiang, Q.Y., Li, W.J.: Discrete latent factor model for cross-modal hashing. IEEE TIP **28**(7), 3490–3501 (2019)
19. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Kumar, S., Udupa, R.: Learning hash functions for cross-view similarity search. In: IJCAI (2011)
21. Li, C., Deng, C., Li, N., Liu, W., Gao, X., Tao, D.: Self-supervised adversarial hashing networks for cross-modal retrieval. In: CVPR, June 2018
22. Li, C., Deng, C., Wang, L., Xie, D., Liu, X.: Coupled cyclegan: unsupervised hashing network for cross-modal retrieval. In: AAAI, pp. 176–183 (2019)
23. Li, C., Chen, Z., Zhang, P., Luo, X., Nie, L., Xu, X.: Supervised robust discrete multimodal hashing for cross-media retrieval. IEEE TMM **21**(11), 2863–2877 (2019)

24. Li, X., Shen, C., Dick, A., Van Den Hengel, A.: Learning compact binary codes for visual tracking. In: CVPR, pp. 2419–2426 (2013)
25. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
26. Lin, Z., Ding, G., Hu, M., Wang, J.: Semantics-preserving hashing for cross-view retrieval. In: CVPR, pp. 3864–3872 (2015)
27. Liu, S., Qian, S., Guan, Y., Zhan, J., Ying, L.: Joint-modal distribution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval. In: ACM SIGIR, pp. 1379–1388 (2020)
28. Luo, X., Yin, X.Y., Nie, L., Song, X., Wang, Y., Xu, X.S.: SDMCH: supervised discrete manifold-embedded cross-modal hashing. In: IJCAI, pp. 2518–2524 (2018)
29. Mandal, D., Chaudhury, K.N., Biswas, S.: Generalized semantic preserving hashing for n-label cross-modal retrieval. In: CVPR, pp. 4076–4084 (2017)
30. Peng, Y., Qi, J.: CM-GANs: cross-modal generative adversarial networks for common representation learning. ACM TOMM **15**(1), 1–24 (2019)
31. Rasiwasia, N., et al.: A new approach to cross-modal multimedia retrieval. In: ACM MM, pp. 251–260 (2010)
32. Rastegari, M., Choi, J., Fakhraei, S., Hal, D., Davis, L.: Predictable dual-view hashing. In: ICML, pp. 1328–1336 (2013)
33. Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., Akata, Z.: Generalized zero-and few-shot learning via aligned variational autoencoders. In: CVPR, pp. 8247–8255 (2019)
34. Shen, H.T., et al.: Exploiting subspace relation in semantic labels for cross-modal hashing. IEEE TKDE (2020)
35. Shen, X., Zhang, H., Li, L., Zhang, Z., Chen, D., Liu, L.: Clustering-driven deep adversarial hashing for scalable unsupervised cross-modal retrieval. Neurocomputing **459**, 152–164 (2021)
36. Shi, Y., You, X., Zheng, F., Wang, S., Peng, Q.: Equally-guided discriminative hashing for cross-modal retrieval. In: IJCAI, pp. 4767–4773 (2019)
37. Song, J., Yang, Y., Yang, Y., Huang, Z., Shen, H.T.: Inter-media hashing for large-scale retrieval from heterogeneous data sources. In: ACM SIGKDD, pp. 785–796 (2013)
38. Su, S., Zhong, Z., Zhang, C.: Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval. In: ICCV, pp. 3027–3035 (2019)
39. Sun, C., Song, X., Feng, F., Zhao, W.X., Zhang, H., Nie, L.: Supervised hierarchical cross-modal hashing. In: ACM SIGIR, pp. 725–734 (2019)
40. Wang, D., Cui, P., Ou, M., Zhu, W.: Learning compact hash codes for multimodal representations using orthogonal deep structure. IEEE TMM **17**(9), 1404–1416 (2015)
41. Weiss, Y., Torralba, A., Fergus, R., et al.: Spectral hashing. In: NeurIPS, vol. 1, p. 4. Citeseer (2008)
42. Wu, B., Yang, Q., Zheng, W.S., Wang, Y., Wang, J.: Quantized correlation hashing for fast cross-modal search. In: IJCAI, pp. 3946–3952. Citeseer (2015)
43. Wu, G., et al.: Unsupervised deep hashing via binary latent factor models for large-scale cross-modal retrieval. In: IJCAI, pp. 2854–2860 (2018)
44. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: ICML, pp. 478–487 (2016)
45. Yang, D., Wu, D., Zhang, W., Zhang, H., Li, B., Wang, W.: Deep semantic-alignment hashing for unsupervised cross-modal retrieval. In: ICMR, pp. 44–52 (2020)

46. Yang, E., Deng, C., Li, C., Liu, W., Li, J., Tao, D.: Shared predictive cross-modal deep quantization. IEEE TNNLS **29**(11), 5292–5303 (2018)
47. Yang, E., Deng, C., Liu, W., Liu, X., Tao, D., Gao, X.: Pairwise relationship guided deep hashing for cross-modal retrieval. In: AAAI (2017)
48. Yu, J., Zhou, H., Zhan, Y., Tao, D.: Deep graph-neighbor coherence preserving network for unsupervised cross-modal hashing (2021)
49. Zhai, D., Chang, H., Zhen, Y., Liu, X., Chen, X., Gao, W.: Parametric local multimodal hashing for cross-view similarity search. In: IJCAI (2013)
50. Zhang, H., et al.: Deep unsupervised self-evolutionary hashing for image retrieval. IEEE Trans. Multimedia **23**, 3400–3413 (2021)
51. Zhang, J., Peng, Y., Yuan, M.: Unsupervised generative adversarial cross-modal hashing. In: AAAI (2018)
52. Zheng, Z., Zheng, L., Garrett, M., Yang, Y., Xu, M., Shen, Y.D.: Dual-path convolutional image-text embeddings with instance loss. ACM TOMM **16**(2), 1–23 (2020)