From Sparse to Dense: Semantic Graph Evolutionary Hashing for Unsupervised Cross-Modal Retrieval

Yang Zhao[1], Jiaguo Yu[1], Shengbin Liao[2], Zheng Zhang[3], and Haofeng Zhang[1](✉)

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.
{zhao_yang,yujiaguo,zhanghf}@njust.edu.cn
[2] National Engineering Research Center for E-learning, Huazhong Normal University, Wuhan 430079, China.
[3] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, 518055, China.

Abstract. In recent years, cross-modal hashing has attracted an increasing attention due to its fast retrieval speed and low storage requirements. However, labeled datasets are limited in real application, and existing unsupervised cross-modal hashing algorithms usually employ heuristic geometric prior as semantics, which introduces serious deviations as the similarity score from original features cannot reasonably represent the relationships among instances. In this paper, we study the unsupervised deep cross-modal hash retrieval method and propose a novel Semantic Graph Evolutionary Hashing (SGEH) to solve the above problem. The key novelty of SGEH is its evolutionary affinity graph construction method. To be concrete, we explore the sparse similarity graph with clustering results, which evolve from fusing the affinity information from code-driven graph on intrinsic data and subsequently extends to dense hybrid semantic graph which restricts the process of hash code learning to learn more discriminative results. Moreover, the batch-inputs are chosen from edge set rather than vertexes for better exploring the original spatial information in the sparse graph. Experiments on four benchmark datasets demonstrate the superiority of our framework over the state-of-the-art unsupervised cross-modal retrieval methods. Code is available at: https://github.com/theusernamealreadyexists/SGEH.

Keywords: Cross-modal Hashing · Visual-text Retrieval · Sparse Affinity Graph · Semantic Graph Evolution.

1 Introduction

Cross-modal retrieval aims to search the related results of other different modalities from a query term of one modal, *e.g.*, using a caption to retrieve the related pictures in database. With the explosive growth of multimedia data, hashing technology, which encodes continuous features into common hash space where relative samples have similar binary codes, is widely used in cross-modal retrieval technology due to its few storage, low Hamming distance computational complexity and fast retrieval speed [8,6,30,7,10,27,31,18,15].

Recently, [25] proposes an unsupervised deep cross-modal hashing method that learns hash codes via Laplacian constraint in objective function to preserve the neighborhood information from code-driven dense semantic graph. However, a significant shortcoming is that it only preserves the original relationship from different modalities, while integrating the affinity information from instances into a unified structure in advance can improve affinity relation construction, which brings a better performance. Then [32] adopts Generative Adversarial Network (GAN) to train cross-modal hash training. With the intention to preserve the correlation from inter-modal and intra-modal in the latent hash space, this method maintains a manifold structure across the attributes of different modalities. In their algorithm, hash code plays a significant role for both the generator and the discriminator. Based on CycleGAN, [34,26] proposed a new method to learn hash codes via unpaired instances. [21] proposes Deep Joint-Semantics Reconstructing Hashing (DJSRH) which fuses the semantic similarities into a unified matrix to explore the latent relevance for the input multi-modal instances. Though impressive progress the above methods have made, several challenges still exist in this task. Therefore, our study is motivated according to the following issues.

(1) The difficulty of excavating non-label relationships from intra-modal and inter-modal. Unsupervised hashing cross-modal methods usually have no access to accurate relationship among instances. Based on co-occurrence information, recent unsupervised techniques [33,21,12,32,11,28,23] usually adopt Attention Mechanism or GAN to generate affinity graph structure which aims to aggregate neighborhood information.

(2) The semantic gap of different modality. Given that each modality has its own geometric prior, each modal have its own affinity code-driven graph. the different between multi-graph may confuse the training process. Thus, coming up with a semantic-unified graph is necessary for conducting the task.

(3) The impossibility to utilize the adjacency matrix as the feature space of large graph. Graph embedding methods can be utilized to fix out the huge storage consumption of large adjacency matrix. However, graph embedding loses a lot of original information during dimension reduction process, which may lead to a sub-optimal performance and fail to preserve the similarity information.

In this paper, to tackle the aforementioned issues, we propose a novel unsupervised method called Semantic Graph Evolutionary Hashing (SGEH). We define a graph evolutionary module which extends the sparse affinity graph to a dense semantic graph, the core idea of which is illustrated in Fig. 1. Specifically, we first take the code-driven similarity graph of both visual modality and textual modality built upon geometric prior and fuse them in a automatically updated weight. Then, after keeping updating the fused graph, we generate a sparse semantic graph which can be shared by both image modal and text modal and relief the problem of lacking label. Our fuse method is inspired by [22] and

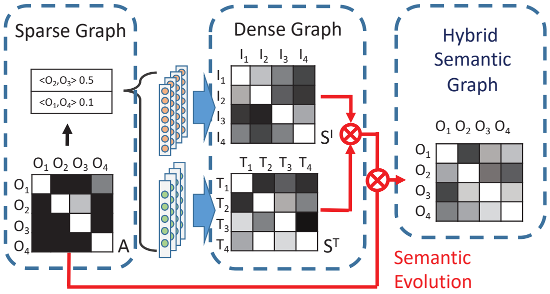

Fig. 1. The process of semantic graph evolution. Sparse graph evolves with the similarity information on geometrical priority. O means object which composes of visual information and caption, and O_i means i-th object in train set.

preserves similarity information hidden in original data from both two modalities. Subsequently, to maximize the use of the spatial information in the sparse graph, we randomly select samples from the connected node pairs in the sparse graph to construct the input of the convolutional neural network. In addition, on one hand, it is impossible to employ the huge adjacency matrix as the feature space for the input of deep network; on the other hand, focusing on neighboring sample pairs can avoid the interference of non-neighboring sample pairs. Hence, we evolve the spares affinity graph to a dense manner by taking the local geometric characters. To map data from different modalties into one common latent hash space, we try to reduce the distance between binary hash codes of the same instance from two modalities, which is reflected in our objective function. In summary, this method has the following main contributions:

- We propose a novel SGEH method by evolving the joint sparse graph obtained by cross-modal clustering into a dense semantic graph that can be used for mini-batch deep learning, which enables hash learning to obtain rich semantic associations among examples.
- We propose a graph evolutionary mechanism to learn a hybrid semantic graph structure from sparse to dense. Instead of directly using dense graphs constructed according to the geometric characteristics of the samples, the graph evolutionary mechanism first learn a sparse affinity graph, and subsequently extend it to a dense form by fusing the dense relation built upon the geometric graph.
- Comprehensive experiments are conducted on four popular datasets, and the results show the priority of the proposed model.

2 Methodology

2.1 Preliminaries

The overall pipeline of SGEH is shown in Fig. 2. We first introduce several definitions in our methods. Let $\boldsymbol{F}^I \in \mathbb{R}^{m\times d_I}$ and $\boldsymbol{F}^T \in \mathbb{R}^{m\times d_V}$ denote m training

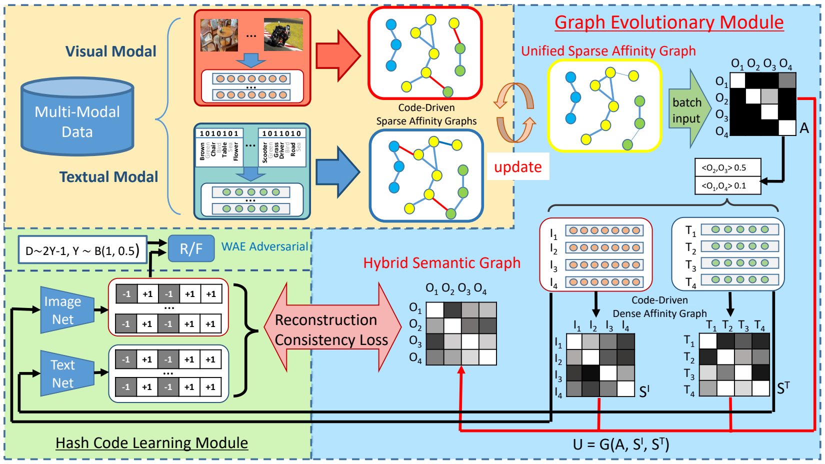

Fig. 2. The framework of our proposed Semantic Graph Evolutionary Hashing (SGEH).

visual and textual instance features respectively. m means the amount of instance in the whole training set. 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 $\boldsymbol{X}^T \in \mathbb{R}^{n \times d_V}$ 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 is hash code length.

Utilizing hash code that preserve the neighborhood information can greatly improve the performance of retrieval task. specifically, previous methods can be grouped into two categories in terms of how to guide hashing learning based on original features. The first category methods preserve the information of original features and use them to learning hash codes directly, they share the following common loss function:

$$\begin{aligned} \mathcal{L}_H &= \left\| \boldsymbol{X}^I \boldsymbol{W}^I - \boldsymbol{B}^I \right\|_F^2 + \left\| \boldsymbol{X}^T \boldsymbol{W}^T - \boldsymbol{B}^T \right\|_F^2, \\ & s.t. \boldsymbol{B}^g \in \{+1, -1\}^{n \times l}, (\boldsymbol{B}^g)^T \boldsymbol{B}^g = n\boldsymbol{I}, \end{aligned} \tag{1}$$

where $\boldsymbol{W}^g$ is learning parameter matrix and $g \in \{I, T\}$. Eq. 1 aims to reduce the gap between features and hash codes. The second constraint $(\boldsymbol{B}^g)^T \boldsymbol{B}^g = n\boldsymbol{I}$ aims to generate mutually independent hash codes.

Evolved from the first category, the second category typically generates similarity structure from the features in two modalities, and further combines with the method of graph optimization or graph fusion methods to obtain joint-semantic affinity matrices. 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 common formulation is as following:

$$\begin{aligned} &\mathcal{L}_G = \|\boldsymbol{S} - \boldsymbol{Q}\|_F^2 \,, \\ s.t.\boldsymbol{S} = g_1(&\boldsymbol{X}^I, \boldsymbol{X}^T) \in \{+1, -1\}^{n \times n}, \boldsymbol{Q} = g_2(\boldsymbol{B}^I, \boldsymbol{B}^T) \in \{+1, -1\}^{n \times n}, \end{aligned} \tag{2}$$

where m is the batch size, S preserves the affinity information of samples in each batch and is used for hash learning. Q represent the affinity matrix generated by hash codes. g_1 and g_2 means two similarity calculating functions. However, Graph structure is a typical non-euclidean structure data. Generally, building adjacency matrix based on euclidean distance not only causes similarity information loss but also calculates unrealistic similarity which confuses the training process. For instance, image feature $\boldsymbol{X}_i^I$ and $\boldsymbol{X}_j^I$ are unrelated but may got a few similarity score in a mini-batch, which misleads the process of hash code training. As demonstrated in [23], the sparse graph distilling knowledge from geometric and semantic information of the whole training set can save the most useful similarity information hidden in the data samples to avoid useless or interfering information. Then the unified semantic graph is kept updated based on these features from two modalities, which is defined as $Z = f(\boldsymbol{F}^I, \boldsymbol{F}^T)$, where $f(\cdot)$ is neighborhood information fusion function, the solution of which will be demonstrated in Subsection 2.2, from Eq. 18 to Eq. 23.

2.2 Unified Sparse Affinity Graph

Clustering results provided by pre-computed global graph can solve the problem of lacking label information to a certain degree. Therefore, in this stage, we need to generate two similarity-induced graphs from both visual and textual modalities which are used to learn a fusion semantic graph $\boldsymbol{Z} \in [0, 1]^{m \times m}$. And $\boldsymbol{Z}$ should keep samples with smaller distance responding to a larger similarity score, and ones with larger distance responding to a smaller similarity score. Finally, $\boldsymbol{Z}$ need to produce clustering results for better hash learning. What's more, it is desirable to update the similarity-induced graphs and fusion semantic graph at the same time for strengthening the accuracy of clustering result. As demonstrated in [22,23], sparse structure has strong anti-noise ability and friendly to storage. Thus, we first use Gaussian Kernel $S(\boldsymbol{F}_i^H, \boldsymbol{F}_j^H) = exp(\frac{-\left\|F_i^H - F_j^H\right\|_2^2}{2\sigma^2})$ to define the weight of two instances from same modal, where σ is the width parameter of the function and controls the radial range of the function, and lately keep k nearest neighborhoods of each vertex to keep graph sparse. Thus, we get $\boldsymbol{S}^I \in [0, 1]^{m \times m}$ and $\boldsymbol{S}^T \in [0, 1]^{m \times m}$, where $\boldsymbol{S}^H$ is the similarity-induced graph and $H \in \{I, T\}$. SGEH mimics GMC [22] loss function and translated it into double-modal expression.

$$\begin{aligned} \min_{\{\boldsymbol{S}^I, \boldsymbol{S}^T\}} &\sum_{H \in \{I,T\}} \sum_{i,j=1}^{m} \left\|\boldsymbol{F}_i^H - \boldsymbol{F}_j^H\right\|_2^2 \boldsymbol{S}_{ij}^H + \lambda_1 \sum_{H \in \{I,T\}} \sum_{i}^{m} \left\|\boldsymbol{S}_i^H\right\|_2^2 \,, \\ &s.t.\boldsymbol{S}_{ii}^H = 0, \boldsymbol{S}_{ij}^H \geq 0, \mathbf{1}^T \boldsymbol{S}_i^H = 1, \end{aligned} \tag{3}$$

and the optimal solution of $\boldsymbol{S}^I$ and $\boldsymbol{S}^T$ can be written in closed-form as Eq. (4) with Lagrange Multiplier Method, which is proved in [22].

$$\boldsymbol{S}_{ij}^H = \begin{cases} \dfrac{\boldsymbol{b}_{i,p+1} - \boldsymbol{b}_{ij}}{p\boldsymbol{b}_{i,p+1}, - \sum_{h=1}^{p} \boldsymbol{b}_{ih}} & j \le p, \\ 0 & j > p, \end{cases} \tag{4}$$

where p is a hyper parameter which adjusts the number of neighbors kept in graph and $\left\| \boldsymbol{F}_i^H - \boldsymbol{F}_j^H \right\|_2^2$ is simplified as $\boldsymbol{b}_{ij}$.

Then we need to learn a sparse fusion semantic graph $\boldsymbol{Z}$ to represent the similarity connection in both visual and textual modalities. We design the loss function describing the distance between $\boldsymbol{Z}$ and $\boldsymbol{S^H}$ as:

$$\min_{\boldsymbol{Z}} \sum_{H \in \{I,T\}} w^H \left\| \boldsymbol{Z} - \boldsymbol{S}^H \right\|_F^2, s.t. \boldsymbol{Z}_{ij} \ge 0, \boldsymbol{1}^T \boldsymbol{Z}_i = 1, \tag{5}$$

where w^H is the weight of similarity-induced graph. As deliberated in Section 1, it is desirable to weight both visual modal and textual modal automatically.

The last problem we need to figure out is how to produce clustering result directly on $\boldsymbol{Z}$. This can be tackled be adding an rank constraint on the graph Laplacian matrix of $\boldsymbol{Z}$ as

$$\min_{\boldsymbol{D}} Tr\left(\boldsymbol{D}^T \boldsymbol{L} \boldsymbol{D}\right), s.t. \boldsymbol{D}^T \boldsymbol{D} = \boldsymbol{I}, \tag{6}$$

where $\boldsymbol{L}$ is the Laplacian matrix of $\boldsymbol{Z}$ and $\boldsymbol{D} \in \mathbb{R}^{m \times c}$ is the embedding matrix composed by cluster center vectors. It comes from a theorem that if a matrix is non-negative, then the dimension of the nullspace of Laplacian matrix of the graph of this matrix is the number of connected components of the graph. Thus, let c denotes the number of connected components of $\boldsymbol{Z}$, if rank(L)$=m-c$, the vertexes in $\boldsymbol{Z}$ can be divided into C clusters. However, rank($\boldsymbol{L}$)$=m-c$ is difficult to achieve. Given that $\boldsymbol{L}$ is positive semi-definite, the constraint rank($\boldsymbol{L}$)$=m-c$ can be achieved if the summation of top-c eigenvalue of $\boldsymbol{L}$ equals to zero. Ky Fan's Theorem [4] told us $\sum_{i=1}^{c} v_i = \min Tr\left(\boldsymbol{D}^T \boldsymbol{L} \boldsymbol{D}\right), s.t. \boldsymbol{D}^T \boldsymbol{D} = \boldsymbol{I}$, where v_i is the ith smallest eigenvalue of $\boldsymbol{L}$. Then based on the above fact, this problem can be further tackled by restricting $Tr\left(\boldsymbol{D}^T \boldsymbol{L} \boldsymbol{D}\right) = 0$. Mathematically, we got the loss function of producing sparse fusion semantic graph U as:

$$\begin{aligned} \mathcal{L}^A = & \sum_{H \in \{I,T\}} \sum_{i,j=1}^{m} \left\| \boldsymbol{F}_i^H - \boldsymbol{F}_j^H \right\|_2^2 \boldsymbol{S}_{ij}^H + \lambda_1 \sum_{H \in \{I,T\}} \sum_{i}^{m} \left\| \boldsymbol{S}_i^H \right\|_2^2 \\ & + \sum_{H \in \{I,T\}} w^H \left\| \boldsymbol{Z} - \boldsymbol{S}^H \right\|_F^2 + 2\lambda_2 Tr\left(\boldsymbol{D}^T \boldsymbol{L} \boldsymbol{D}\right), \\ & s.t. \boldsymbol{S}_{ii}^H = 1, \boldsymbol{S}_{ij}^H \ge 0, \boldsymbol{1}^T S_i^H = 1, \boldsymbol{Z}_{ij} \ge 0, \boldsymbol{1}^T Z_i = 1, \boldsymbol{D}^T \boldsymbol{D} = \boldsymbol{I}, \end{aligned} \tag{7}$$

where $H \in \{I, T\}$, $\boldsymbol{S}_i \in \mathbb{R}^{n \times 1}$, $\boldsymbol{U}_i \in \mathbb{R}^{n \times 1}$, w^H is the weight of similarity-induced graph, $\boldsymbol{L}$ is the Laplacian matrix of Z and $\boldsymbol{D} \in \mathbb{R}^{m \times c}$ is the clustering center matrix. The optimization of above problem will be solved in section 2.5.

2.3 Semantic Graph Evolution

Although the matrix we obtained at this time can more accurately reflect the relationship between instances, we need to use mini-batch method in the neural network model, which means that in each batch, similarity matrix composed of randomly selected samples tends to be sparse for each instance only have p neighbors. In the hash learning stage, the semantic similarity matrix is utilized to constrain the hash similarity matrix generated by the hash code. However, sparse matrices are incompetent to guide the hash learning process. For example, two pairs of data whose similarity is 0 in the sparse semantic matrix are not also 0 in the hash similarity matrix. At the same time, the sparse matrix has the problem of information loss. Thus, in order to solve the problem of sparse, we need to evolve the similarity matrix into a denser form.

To be specific, suppose there are t edges in ε_Z, which is the edge set of $\boldsymbol{Z}$, and n $(t+1 < n \leq 2t)$ related vertexes for each batch. Then we use V and E to represent its vertex set and edge set respectively. And subsequently we constructs a local sparse graph $\boldsymbol{A} = (V, E)$ on batch-inputs. The features of vertexes in V can be denoted as $\boldsymbol{X^I}$ and $\boldsymbol{X}^T$, which are defined above respectively. We define the similarity matrices in mini batch from visual and textual modalities as:

$$\boldsymbol{S}_{ij}^H = \boldsymbol{X}_i^H (\boldsymbol{X}_j^H)^T / (\left\|\boldsymbol{X}_i^H\right\| \left\|\boldsymbol{X}_j^H\right\|), \tag{8}$$

where $\boldsymbol{S}_{ij}^H \in [-1, +1]$, $H \in \{I, T\}$. $\boldsymbol{X}_i^H$ means the i-th row in $\boldsymbol{X}^H$ and $\boldsymbol{X}_j^H$ stands for the j-th row in $\boldsymbol{X}^H$. We employ $\boldsymbol{S^I}$ and $\boldsymbol{S^T}$ to integrate the original similarity information in image and text modal. Then unified sparse affinity graph A is evolved from spare to dense by fusing the information from $\boldsymbol{S^I}$ and $\boldsymbol{S^I}$. Then we get hybrid semantic affinity matrix $\boldsymbol{U} = \mathcal{G}(\boldsymbol{A}, \boldsymbol{S}^I, \boldsymbol{S}^T) \in [-1, +1]^{n \times n}$ to describe the affinity structure in both two modalities, with $\boldsymbol{U}_{ij}$ describing the captured fusion semantic affinity information between the input samples $\boldsymbol{e_i}$ and $\boldsymbol{e}_j$. The hybrid semantic affinity matrix is calculated as:

$$\boldsymbol{U} = \mathcal{G}_1(\boldsymbol{A}, \boldsymbol{S}^I, \boldsymbol{S}^T) = (1 - \lambda_3)[\lambda_4 \boldsymbol{S}_I + (1 - \lambda_4)\boldsymbol{S}_T] + \lambda_3 \boldsymbol{A}, \tag{9}$$

where λ_3 adjust the importance of clustering result and λ_4 balances the weights between affinity structure information of visual modality and textual modality. The manner of constructing $\boldsymbol{U}$ combines the similarity information across both clustering result and original affinity structure in two modalities. Given that samples selected in batch are highly related, $\boldsymbol{U}$ refines the affinity more accurate than randomly training samples in mini-batch, which makes it capturing more effective latent common similarity relationship over multi-modal perspective. In another word, $\boldsymbol{U}$ reflects the original affinity connection among input samples, after which we can subsequently learn binary hash code that are employed to achieve cross-modal retrieval task. Alternatively, following the form of combination in [21], we can make $\boldsymbol{A}$ evolved in the form as:

$$\overline{\boldsymbol{S}} = \lambda_4 \boldsymbol{S}_I + (1 - \lambda_4)\boldsymbol{S}_T, \boldsymbol{U} = \mathcal{G}_2(\boldsymbol{A}, \boldsymbol{S}^I, \boldsymbol{S}^T) = \lambda_3 \boldsymbol{A} + (1 - \lambda_3)(\frac{\overline{\boldsymbol{S}}\overline{\boldsymbol{S}}^T}{n}). \tag{10}$$

2.4 Hash-Code Learning

In this subsection, we utilize the accurate semantic matrix $\boldsymbol{U}$, which represents the original affinity relations of the input instances, to restrict the generation stage of hash code. In latent hash hypercube, adjacent vertices share small Hamming distance and more similar hash codes. Thus, hash codes can be understood as discrete features. To calculate the similarity with neighborhoods in Hamming space, the similarity function can be defined as:

$$\mathcal{Z}(\boldsymbol{B}_i^H, \boldsymbol{B}_j^H) = \boldsymbol{B}_i^H (\boldsymbol{B}_j^H)^T / (\|\boldsymbol{B}_i^H\| \, \|\boldsymbol{B}_j^H\|), \tag{11}$$

where $H \in \{I, T\}$, $\boldsymbol{B}_i^H$ means the i-th row in $\boldsymbol{B}^H$ and $\boldsymbol{B}_j^H$ means the j-th row in $\boldsymbol{B}^H$. The result of Eq.(11) is the cosine affinity score which representing the angular connection among discrete features. We adopt manner of Eq.(2) that minimize the reconstruction error between the similarity matrix of hash code and the affinity matrix $\boldsymbol{U}$ of continuous features to keep their similarity consistency. Therefore, we define pairwise cosine similarity matrices as $\boldsymbol{Q}$ and $\boldsymbol{Q}_{ij}^{HH} = \mathcal{Z}(\boldsymbol{B}_i^H, \boldsymbol{B}_j^H)$. Then, we employ

$$\mathcal{L}^{\boldsymbol{B}_{HH}} = \min_{\boldsymbol{B}_H} \left\|\alpha U - \mathcal{Z}(\boldsymbol{B}_i^H, \boldsymbol{B}_j^H)\right\|_F^2, \tag{12}$$

as the formulation to compute the difference between U and $\boldsymbol{Q}_{ij}^{HH}$. In Eq. (12), α is a hyper-parameter which makes reconstruction more flexible, as discussed in [21]. Given that $\boldsymbol{U} \in [-1, +1]^{n \times n}$, $\alpha \boldsymbol{U} \in [-\alpha, +\alpha]^{n \times n}$. For example, supposed that $\boldsymbol{U}_{ij} = 0.8$, which means that ith instance and jth instance got 0.8 similarity score, then the similarity score of corresponding hash codes calculated from Hamming space need to be close to 0.8. $\alpha > 1$ means the similarity score of hash codes pair need to lager than 0.8 and accordingly make the nodes in Hamming space dense, while $\alpha < 1$ means the similarity score of hash codes pair need to smaller than 0.8 and accordingly make the nodes in Hamming space sparse. We empirically find that it is beneficial to threshold $\alpha > 1$. 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. The result is not related to the length of the vector, but only related to the direction of the vector. Setting $\alpha > 1$ means we force the binary hash code close to the latent clustering center in Hamming space in direction than it should be, which bring a better performance as we are trying to Increase the distance between categories and reduce the distance within category.

Given that each instance is still represented by hash codes from two modalities, we need to restrict the reconstruction in the manner of intro-modal and inter-modal. Specifically, we employ $\boldsymbol{Q}^{II}$ and $\boldsymbol{Q}^{TT}$ as the intro-modal reconstruction for image modal and text modal respectively, and $\boldsymbol{Q}^{IT}$ is engaged as inter-modal reconstruction. Finally, the consistency loss between binary hash code and continues original features can be summarized as:

$$\begin{aligned}\mathcal{L}^{\boldsymbol{B}} =&\mathcal{L}^{\boldsymbol{B}_{II}} + \eta_1 \mathcal{L}^{\boldsymbol{B}_{TT}} + \eta_2 \mathcal{L}^{\boldsymbol{B}_{IT}} = \min_{\boldsymbol{B}_I} \left\| \alpha U - \mathcal{Z}(\boldsymbol{B}_i^I, \boldsymbol{B}_j^I) \right\|_F^2 \\ &+ \eta_1 \min_{\boldsymbol{B}_T} \left\| \alpha U - \mathcal{Z}(\boldsymbol{B}_i^T, \boldsymbol{B}_j^T) \right\|_F^2 + \eta_2 \min_{\boldsymbol{B}_I, \boldsymbol{B}_T} \left\| \alpha U - \mathcal{Z}(\boldsymbol{B}_i^I, \boldsymbol{B}_j^I) \right\|_F^2,\end{aligned} \tag{13}$$

where η_1 and η_2 are the trade-off parameters to balance the reconstruction of inter-modal and intro-modal in latent hash space.

To avoid wasting bits and align representation distributions, it is worth generating mutually independent hash codes as the constraint $\boldsymbol{B}_g^T \boldsymbol{B}_g = m\boldsymbol{I}$ in Eq. 1. However, the above loss cannot tackle this problem. Following [19], we regularize the latent discrete features with axuiliary discriminator $d(\cdot, \zeta)$. To be concrete, we assume the each row in $\boldsymbol{B}$ comes from a distribution $\mathcal{D} = 2\mathcal{Y} - 1, \mathcal{Y} \sim \mathcal{B}(1, 0.5)$, which maximizes the code entropy. We suppose that each binary code is proiored by a binomial distribution which is $\mathcal{D}$. Then, to adversarially regularize the latent variables, we utilize auxiliary discriminator d which involves two fully-connected layers successively with ReLu and sigmoid non-linearities. In a word, it is to balance the amount of zeros and ones in each binary code and further maximize the code entropy. In this end, we can employ the following discriminator d to balance -1 and +1 in a binary hash code:

$$d(\boldsymbol{B}_i, \zeta) \in (-1, +1); d(\boldsymbol{y}^{\boldsymbol{B}_i}, \zeta) \in (-1, +1), \tag{14}$$

where $\boldsymbol{y}^{\boldsymbol{B}_i}$ obeys the same distribution as $\boldsymbol{B}_i$ for implicit regularizing $\boldsymbol{B}_i$. Therefore, our final loss can be written as:

$$\mathcal{L} = \mathcal{L}^A + \mathcal{L}^B - \frac{\eta_4}{b} \sum_{H \in I,T} \sum_{i=1}^{b} (log(1 - d(\boldsymbol{B}_i, \zeta)) + log\ d(\boldsymbol{y}^{\boldsymbol{B}_i}, \zeta))). \tag{15}$$

2.5 Optimization

In this method, the global sparse graph is employed to solve the problem of missing label information and we first need to optimize Eq. 7.

Sparse Affinity Graph. In this stage, we basically refer to the alternating rules used in [22] to optimize Eq. 7. As there are four variables in total and are coupled with each other, the problem is split into four step. It is beneficial to get detailed information from the above method. Here, we directly give the close-form solution of the variables $\boldsymbol{Z}, \boldsymbol{S}, \boldsymbol{D}, \boldsymbol{w^H}$:

*step*1 : Fix $\boldsymbol{Z}, \boldsymbol{D}$ and w^H, update $\boldsymbol{S}^H$. When $\boldsymbol{Z}, \boldsymbol{D}$ and w^H fixed, the last item of Eq. 7 is constant and accordingly original problem is translated into following pattern:

$$\begin{aligned}\min_{\boldsymbol{S}^H} &\sum_{H \in \{I,T\}} \sum_{i,j=1}^{m} \left\| \boldsymbol{F}_i^H - \boldsymbol{F}_j^H \right\|_2^2 \boldsymbol{S}_{ij}^H + \lambda_1 \sum_{H \in \{I,T\}} \sum_{i}^{m} \left\| \boldsymbol{S}_i^H \right\|_2^2 \\ &+ \sum_{H \in \{I,T\}} w^H \left\| \boldsymbol{Z} - \boldsymbol{S}^H \right\|_F^2, \quad s.t. \boldsymbol{S}_{ii}^H = 1, \boldsymbol{S}_{ij}^H \geq 0, \boldsymbol{1}^T S_i^H = 1,\end{aligned} \tag{16}$$

Updating both two view is independent, we update each $\boldsymbol{S}^H$ in the following way:

$$\begin{aligned}&\min_{\boldsymbol{S}^H} \sum_{i,j=1}^{m} \left\|\boldsymbol{F}_i^H - \boldsymbol{F}_j^H\right\|_2^2 \boldsymbol{S}_{ij}^H + \lambda_1 \sum_{i}^{m} \left\|\boldsymbol{S}_i^H\right\|_2^2 + w^H \left\|\boldsymbol{Z} - \boldsymbol{S}^H\right\|_F^2, \\ &s.t. \boldsymbol{S}_{ii}^H = 1, \boldsymbol{S}_{ij}^H \geq 0, \mathbf{1}^T S_i^H = 1,\end{aligned} \tag{17}$$

For simplicity, we suppose that a feature of sample is similar to its neighbours and accordingly can update the representation using its p neighbor data points, where p is the number of neighbors. We employ the solution from [22] and give the final solution as follows:

$$\boldsymbol{S}_{ij}^{H*} = \begin{cases} \dfrac{\boldsymbol{b}_{i,p+1} - \boldsymbol{b}_{ij} + 2w^H \boldsymbol{Z}_{ij} - 2w^H \boldsymbol{Z}_{i,p+1}}{p\boldsymbol{b}_{i,p+1}, - \sum_{h=1}^{p} \boldsymbol{b}_{ih} - 2pw^H \boldsymbol{Z}_{i,p+1} + 2\sum_{h=1}^{p} w^H \boldsymbol{Z}_{ih}} & j \leq p, \\ 0 & j > p, \end{cases} \tag{18}$$

*step*2 : Fix $\boldsymbol{Z}, \boldsymbol{D}$ and $\boldsymbol{S}^H$, update w^H. In this step, fixing problem 7 is the same way to solve the problem (5).

Theorem. *If the weights* $\boldsymbol{w}^H$ *are fixed, solving problem* 5 *is equivalent to solving the following probelm:*

$$\min_{\boldsymbol{Z}} \sum_{H \in \{I,T\}} \sqrt{\|\boldsymbol{Z} - \boldsymbol{S}^H\|_F^2}, s.t. \boldsymbol{Z}_{ij} \geq 0, \mathbf{1}^T Z_i = 1, \tag{19}$$

Proof. The Lagrange function of Eq (19) is:

$$\sum_{H \in \{I,T\}} \sqrt{\|\boldsymbol{Z} - \boldsymbol{S}^H\|_F^2} + \Theta\left(\Lambda, \boldsymbol{Z}\right), s.t. \boldsymbol{Z}_{ij} \geq 0, \mathbf{1}^T Z_i = 1, \tag{20}$$

where Λ is the Lagrange multiplier, and $\Theta\left(\Lambda, \boldsymbol{Z}\right)$ is the formalized term derived from constraints. Taking the derivative of Eq. (20) with respect to $\boldsymbol{Z}$ and setting the derivative to zero, we get the following equation:

$$w^{H*} = \frac{1}{\sqrt[2]{\|\boldsymbol{Z} - \boldsymbol{S}^H\|_F^2}}. \tag{21}$$

*step*3 : Fix all the other variables except $\boldsymbol{Z}$, and it can be proved that solving Eq. 7 is equivalent to solving the following problem:

$$\min_{\boldsymbol{Z}_i} \sum_{H \in \{H,I\}} \left\|\boldsymbol{Z}_i - \boldsymbol{S}_i^H + \frac{\lambda_1}{4w^H} \boldsymbol{d}_i\right\|_2^2, s.t. \boldsymbol{Z}_{ij} \geq 0, \mathbf{1}^T Z_i = 1, \tag{22}$$

where $\boldsymbol{Z}_i$ means the i-th row in $\boldsymbol{Z}$ and $\boldsymbol{d}_{ij}$ means the similarity score between $\boldsymbol{S}_i^{\boldsymbol{H}}$ and $\boldsymbol{Z}_i$. The problem in Eq. 22 can be solved with Lagrange Multiplier

Method as proved in [22] with several steps:

$$\begin{aligned} &\boldsymbol{q}^H = \boldsymbol{S}_i^H - \frac{\lambda_1}{4w^H}\boldsymbol{d}_i, \quad \boldsymbol{p} = \frac{\sum_{H\in\{I,T\}} q^H}{2} + \frac{1}{m} - \frac{\boldsymbol{1}^T\boldsymbol{q}^H\boldsymbol{1}}{2m}, \\ &f(t) = \frac{1}{m}\sum_{j=1}^{m}(t-\boldsymbol{p}_j)_+ - t, \quad \boldsymbol{Z}_{ij}{}^* = (\boldsymbol{p}_j - t^*)_+, \end{aligned} \tag{23}$$

where t^* makes $f(t^*) = 0$ and $(\cdot)_+ = max(\cdot, 0)$.In summary, the produce for solving the proposed problem in Eq. 7 can be found in the supplementary material.

*step*4 :Fix all the other variables except $\boldsymbol{D}$, optimizing problem (7) is equivalent to problem (6), which is formed by the c eigenvectors of $\boldsymbol{L}$ corresponding to the c smallest eigenvalues.

Deep hash learning. In traditional hash methods [5,19], the process of mapping continuous features to discrete space causes huge quantization loss stem from the fact that the sign function, which outputs $+1$ for positive number and -1 for negative number, can not be derived. To handle this problem, we follow [21] to adopt a scaled tanh function:

$$\boldsymbol{B} = tanh(\beta\boldsymbol{Y}) \in [-1,+1]^{m\times d}, \beta \in \mathbb{R}^+, \tag{24}$$

where $\boldsymbol{Y}$ represent that final output of Convolutional Neural Network. It is noticed that β is kept increasing during deep training stage. To be noted that it is motivated by a crucial fact that $\lim_{\alpha\to\infty} tanh(\beta y) = sgn(y)$.

3 Experiments

3.1 Datasets

Four datasets, including **Wiki** [16], **NUS-WIDE** [2], **MIRFlickr-25K** [9] and **MSCOCO** [13], are employed to evaluate the proposed methods, more details about the four datasets can be found in the supplementary material.

3.2 Implementation Details

For all of our experiments, we follow previous methods to employ the VGG-16 fc7 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 [1] to represent final textual features $\boldsymbol{X}^T$ whose dimension is 512. Besides, considering the computational burden in the solution process of $\boldsymbol{Z}$, we randomly pick up 20,000 instances from NUS-WIDE and MSCOCO dataset. It is worth noting that to calculate the consistency loss as the manner of Eq. 2, we need to force the items in the ranges. However, the cosine similarity ranges -1 from $+1$ while the affinity value elements in $\boldsymbol{Z}$ are non-negative, which can be obtained by Eq. 23. Therefore, as $\boldsymbol{A}$ is the batch-input of $\boldsymbol{Z}$, we preprocess the

Table 1. The mAP@all results on image query text ($I \rightarrow T$) and text query image ($T \rightarrow I$) retrieval at various encoding lengths and datasets. The best performances are shown in bold.

Task	Method	**WIKI**			**MIRFlicker-25K**			**MSCOCO**			**NUS-WIDE**		
		16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit
$I \rightarrow T$	CMFH	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	0.196	0.168	0.184	0.544	0.544	0.545	0.442	0.423	0.492	0.368	0.368	0.368
	IMH	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	0.159	0.143	0.131	0.579	0.565	0.554	0.496	0.470	0.441	0.401	0.382	0.370
	DJSRH	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	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	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	0.253	0.289	0.325	0.665	0.681	0.697	0.571	0.613	0.624	0.545	0.553	0.572
	SGEH	**0.396**	**0.422**	**0.441**	**0.665**	**0.695**	**0.703**	**0.578**	**0.617**	**0.634**	**0.565**	**0.584**	**0.579**
$T \rightarrow I$	CMFH	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	0.344	0.293	0.251	0.544	0.544	0.546	0.437	0.440	0.440	0.366	0.366	0.367
	IMH	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	0.341	0.289	0.246	0.585	0.567	0.556	0.505	0.478	0.445	0.405	0.385	0.372
	DJSRH	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	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	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	0.256	0.303	0.320	0.660	0.692	0.710	0.565	0.619	0.632	0.545	0.566	0.576
	SGEH	**0.452**	**0.510**	**0.530**	**0.687**	**0.706**	**0.711**	**0.578**	**0.626**	**0.635**	**0.570**	**0.588**	**0.595**

$\boldsymbol{A}$ with $\boldsymbol{A} \leftarrow 2\boldsymbol{A} - 1$. 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 $\alpha = 1.5$ and $\lambda_3 = 0.4$ for all four datasets. Then we set $c = 5$, $p = 10000$, $\lambda_4 = 0.6$, $\eta_1 = \eta_2 = 0.1$ for NUM-WIDE, $c = 5$, $p = 3000$, $\lambda_4 = 0.9$, $\eta_1 = \eta_2 = 0.1$ for MIRFlicker, $c = 5$, $p = 1000$, $\lambda_4 = 0.3$, $\eta_1 = \eta_2 = 0.3$ for Wiki and $c = 5$, $p = 3000$, $\lambda_4 = 0.6$, $\eta_1 = \eta_2 = 0.1$ for MSCOCO.

3.3 Retrieval Performance

Baselines. Previous methods can be categorized into two kinds according to whether takes the whole retrieved points into consideration or not. Hence, in order to prove that our method has superior performance under different evaluation indicators, we conduct experiments on two aspects. Specifically, on the one hand, we compare the mAP results with IMH [20], CMFH [3], PDH [17], QCH[24], DJSRH [21], DSAH [29], JDSH [14], DGCPN [31] conducted on Wiki, MIRFlicker, MSCOCO and NUS-WIDE datasets, with the whole retrieved points occupied (*i.e.*, mAP@all), and the results are shown in Tab. 1. All the compared method are conducted according to their released codes or description in their original papers. The retrieval performance on mAP@50 can be found in the supplementary material.

Quantitative Results. It can be observed that the proposed SGEH outperforms all of other unsupervised cross-modal hashing methods in Tab. 1 on all four datasets regardless of the cross-modal retrieval tasks and code lengths, which demonstrates the effectiveness of the proposed methods. Specifically, our image

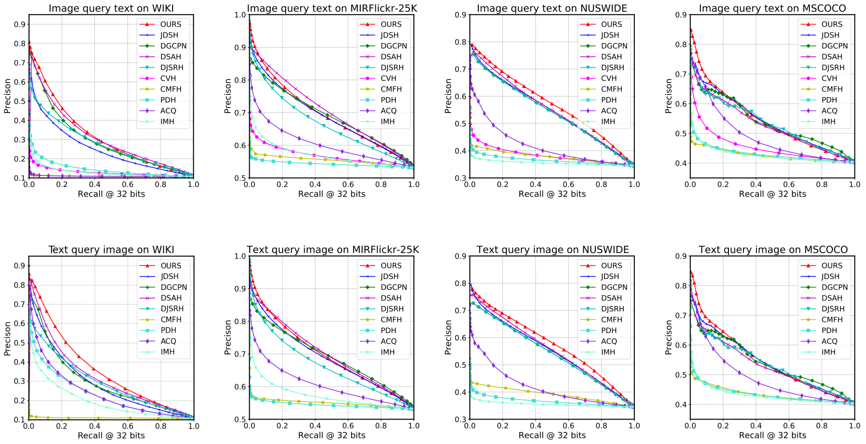

Fig. 3. Results of Precision VS Recall Curves of various unsupervised hashing methods on datasets WIKI, MIRFLickr-25K, MSCOCO and NUS-WIDE with 32-bit codes.

query for text retrieval performance on Wiki dataset improves a lot compared with other deep methods on three kinds of hash codes, especially on 16 bits and 32 bits , while the text query for image retrieval performance outperforms them more than 10.8%, 19.5%, 0.8% on 16 bits, 32 bits, and 64 bits respectively. While improvements on NUS-WIDE and MSCOCO are related lower, which is stemmed from that we only using 20,000 samples as training set. The corresponding Precision-Recall (P-R) curves of represented methods are also retorted in Fig. 3, which can further prove the effectiveness of our method. In particular, our curves for Wiki are all located above those of the other methods, which means that the precision of our approach can significantly surpass that of the other works at the same recall rates. As for the multi-label datasets, Our P-R curves on MSCOCO and NUS-WIDE are also higher than the other, but not as obviously as the curves on Wiki. On the MIRFlickr-25K, we can obtain that the results are slightly worse than DSAH for 32 bits when the recall rate is higher than 0.14 when image queries text and 0.12 when text queries image. However, taking text query image for instance, it can be seen that our curve get (recall $=$ 0.05, precision $=$ 0.81), which means that our method can obtain images with 81% accuracy among the $0.05 \times 20,000 = 1000$ return images.

3.4 Ablation Study

To further demonstrate the effectiveness of each part in SGEH, we design several variants to evaluate the performance when adding the proposed each components. Following the introduction order in Section 2, SGEH-1 and SGEH-2 are the basic variants which respectively only employ $\boldsymbol{A}$ as similarity matrix and only employ $\boldsymbol{S}^I$ with $\boldsymbol{S}^T$ as similarity matrix. SGEH is the variant that merges the

Table 2. The mAP@all on MIRFlickr-25K to evaluate the value of each component.

Model	Configuration	32bits		64bits	
		$I \rightarrow T$	$T \rightarrow I$	$I \rightarrow T$	$T \rightarrow I$
SGEH-1	$\boldsymbol{U} = \boldsymbol{A}$	0.658	0.679	0.650	0.678
SGEH-2	$\boldsymbol{U} = \lambda_4 \boldsymbol{S}^I + (1 - \lambda_4)\boldsymbol{S}^T$	0.684	0.695	0.680	0.701
SGEH-3	$\boldsymbol{U} = \mathcal{G}_1(\boldsymbol{A}, \boldsymbol{S}^I, \boldsymbol{S}^T)$	0.685	0.694	0.692	0.699
SGEH-4	$\boldsymbol{U} = \mathcal{G}_2(\boldsymbol{A}, \boldsymbol{S}^I, \boldsymbol{S}^T)$	0.688	0.692	0.693	0.702
SGEH	$\eta_4 = 0.001$	0.695	0.706	0.703	0.711

affinity metrics in the manner of $\mathcal{G}(\boldsymbol{A}, \boldsymbol{S}^I, \boldsymbol{S}^T) = (1 - \lambda_3)[\lambda_4 \boldsymbol{S}_I + (1 - \lambda_4)\boldsymbol{S}_T] + \lambda_3 \boldsymbol{A}$. SGEH is the variant based on SGEH-4 which further supplements the loss of discriminator. The results are shown in Tab. 2, and from which we can discover that each component of our proposed method has its own effect. Tab. 2 suggests that removing any component of our final framework leads to performance degradation. Specially, compared with the results of SGEH-1 and SGEH-2, the better performance of SGEH-3 and SGEH-4 shows illustrate the effectiveness of the proposed fusion strategy Eq. (9). The combination of clustering information from total dataset and neighborhood information in each mini-batch can much more accurately define the similarity relationship, impelling to learn more consistent hash codes and accordingly achieving better performance. What's more, SGEH demonstrate the important role of hashcode regularization. It facilitates the proposed method for the end-to-end batch-wise training which better refine the similarity relationship by combining the clustering results and mini-batch neighborhood information than previous mini-batch pattern.

4 Conclusion

This paper proposed Semantic Graph Evolutionary Hashing (SGEH) for unsupervised cross-modal retrieval. SGEH first employs sparse affinity graph to update the unified sparse affinity graph, which is shared by both visual modal and textual modal. And subsequently the sparse graph is evolved from sparse to dense by fusing code-driven similarity information. Consequently, the sparse graph extends to the Hybrid Semantic Graph which is utilized to restrict hash code learning. The key novelty of this method is the graph evolution scheme. Then hash code can be learned via construction consistence loss with a more effective feature space. Extensive experiments demonstrate the superiority of our proposed method and detailed ablation study shows the effect of each module utilized in our method.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants No. 61872187, No. 62072246 and No. 62077023, 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. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Céspedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
2. 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: Proceedings of the ACM International Conference on Image and Video Retrieval. pp. 1–9 (2009)
3. Ding, G., Guo, Y., Zhou, J.: Collective matrix factorization hashing for multimodal data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2075–2082 (2014)
4. Fan, K.: On a theorem of weyl concerning eigenvalues of linear transformations i. Proceedings of the National Academy of Sciences of United States of America **35**(11), 652 (1949)
5. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(12), 2916–2929 (2012)
6. He, L., Xu, X., Lu, H., Yang, Y., Shen, F., Shen, H.T.: Unsupervised cross-modal retrieval through adversarial learning. In: ICME. pp. 1153–1158 (2017)
7. Hu, H., Xie, L., Hong, R., Tian, Q.: Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2020)
8. Hu, M., Yang, Y., Shen, F., Nie, N., Hong, R., Shen, H.: Collective reconstructive embeddings for cross-modal hashing. IEEE IEEE Transactions on Image Processing **28**(6), 2770–2784 (2019)
9. Huiskes, M.J., Lew, M.S.: The mir flickr retrieval evaluation. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 39–43 (2008)
10. Li, C., Deng, C., Li, N., Liu, W., Gao, X., Tao, D.: Self-supervised adversarial hashing networks for cross-modal retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2018)
11. Li, C., Deng, C., Li, N., Liu, W., Gao, X., Tao, D.: Self-supervised adversarial hashing networks for cross-modal retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4242–4251 (2018)
12. Li, C., Deng, C., Wang, L., Xie, D., Liu, X.: Coupled cyclegan: Unsupervised hashing network for cross-modal retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 176–183 (2019)
13. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of European Conference on Computer Vision. pp. 740–755. Springer (2014)
14. 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: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1379–1388 (2020)
15. Lu, X., Zhu, L., Li, J., Zhang, H., Shen, H.T.: Efficient supervised discrete multi-view hashing for large-scale multimedia search. IEEE Transactions on Multimedia **22**(8), 2048–2060 (2020). https://doi.org/10.1109/TMM.2019.2947358
16. Rasiwasia, N., Costa Pereira, J., Coviello, E., Doyle, G., Lanckriet, G.R., Levy, R., Vasconcelos, N.: A new approach to cross-modal multimedia retrieval. In: Proceedings of the ACM International Conference on Multimedia. pp. 251–260 (2010)

17. Rastegari, M., Choi, J., Fakhraei, S., Hal, D., Davis, L.: Predictable dual-view hashing. In: Proceedings of International Conference on Machine Learning. pp. 1328–1336. PMLR (2013)
18. Shen, H.T., Liu, L., Yang, Y., Xu, X., Huang, Z., Shen, F., Hong, R.: Exploiting subspace relation in semantic labels for cross-modal hashing. IEEE Transactions on Knowledge and Data Engineering (2020)
19. Shen, Y., Qin, J., Chen, J., Yu, M., Liu, L., Zhu, F., Shen, F., Shao, L.: Auto-encoding twin-bottleneck hashing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2827 (2020)
20. Song, J., Yang, Y., Yang, Y., Huang, Z., Shen, H.T.: Inter-media hashing for large-scale retrieval from heterogeneous data sources. In: Proceedings of the International Conference on Management of Data. pp. 785–796 (2013)
21. Su, S., Zhong, Z., Zhang, C.: Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval. In: Proceedings of the International Conference on Computer Vision. pp. 3027–3035 (2019)
22. Wang, H., Yang, Y., Liu, B.: Gmc: Graph-based multi-view clustering. IEEE Transactions on Knowledge and Data Engineering **32**(6), 1116–1129 (2019)
23. Wang, W., Shen, Y., Zhang, H., Yao, Y., Liu, L.: Set and rebase: determining the semantic graph connectivity for unsupervised cross modal hashing. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 853–859 (2020)
24. Wu, B., Yang, Q., Zheng, W.S., Wang, Y., Wang, J.: Quantized correlation hashing for fast cross-modal search. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 3946–3952. Citeseer (2015)
25. Wu, G., Lin, Z., Han, J., Liu, L., Ding, G., Zhang, B., Shen, J.: Unsupervised deep hashing via binary latent factor models for large-scale cross-modal retrieval. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 2854–2860 (2018)
26. Wu, L., Wang, Y., Shao, L.: Cycle-consistent deep generative hashing for cross-modal retrieval. IEEE Transactions on Image Processing **28**(4), 1602–1612 (2018)
27. Xie, L., Shen, J., Zhu, L.: Online cross-modal hashing for web image retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence (2016)
28. Xu, R., Li, C., Yan, J., Deng, C., Liu, X.: Graph convolutional network hashing for cross-modal retrieval. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 982–988 (2019)
29. Yang, D., Wu, D., Zhang, W., Zhang, H., Li, B., Wang, W.: Deep semantic-alignment hashing for unsupervised cross-modal retrieval. In: Proceedings of the 2020 International Conference on Multimedia Retrieval. pp. 44–52 (2020)
30. Yang, E., Deng, C., Liu, W., Liu, X., Tao, D., Gao, X.: Pairwise relationship guided deep hashing for cross-modal retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017)
31. Yu, J., Zhou, H., Zhan, Y., Tao, D.: Deep graph-neighbor coherence preserving network for unsupervised cross-modal hashing. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021)
32. Zhang, J., Peng, Y., Yuan, M.: Unsupervised generative adversarial cross-modal hashing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
33. Zhang, X., Lai, H., Feng, J.: Attention-aware deep adversarial hashing for cross-modal retrieval. In: Proceedings of European Conference on Computer Vision (September 2018)

34. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the International Conference on Computer Vision. pp. 2223–2232 (2017)