Set and Rebase: Determining the Semantic Graph Connectivity for Unsupervised Cross-Modal Hashing

Weiwei Wang[1*], **Yuming Shen**[2*], **Haofeng Zhang**[1†], **Yazhou Yao**[1] and **Li Liu**[2]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates
{wangweiwei, zhanghf, yazhou.yao}@njust.edu.cn, ym_zmxncbv@hotmail.com, liuli1213@gmail.com

Abstract

The label-free nature of unsupervised cross-modal hashing hinders models from exploiting the exact semantic data similarity. Existing research typically simulates the semantics by a heuristic geometric prior in the original feature space. However, this introduces heavy bias into the model as the original features are not fully representing the underlying multi-view data relations. To address the problem above, in this paper, we propose a novel unsupervised hashing method called Semantic-Rebased Cross-modal Hashing (SRCH). A novel '*Set-and-Rebase*' process is defined to initialize and update the cross-modal similarity graph of training data. In particular, we *set* the graph according to the intra-modal feature geometric basis and then alternately *rebase* it to update the edges within according to the hashing results. We develop an alternating optimization routine to *rebase* the graph and train the hashing auto-encoders with closed-form solutions so that the overall framework is efficiently trained. Our experimental results on benchmarked datasets demonstrate the superiority of our model against state-of-the-art algorithms.

1 Introduction

The era of big data has witnessed continuous research attention in cross-modal hashing because of its low computational complexity and storage requirement for large-scale multimedia retrieval [Zhang *et al.*, 2018a]. The key challenges of this realm are preserving as much similarity information as possible and simultaneously mitigating the modality heterogeneity.

Among the existing methods, supervised cross-modal hashing [Jiang and Li, 2017; Shen *et al.*, 2017; Erin Liong *et al.*, 2017; Tang *et al.*, 2016; Bronstein *et al.*, 2010; Zhang and Li, 2014] obtains better retrieval performance. These techniques utilize label information which can be shared by both image and text data to exploit similarity between samples of different modalities. However, labeled dataset is limited and it may cause huge cost to label large scale multi-modal datasets by hand, which constraints the practicality of supervised algorithms despite their outstanding performance.

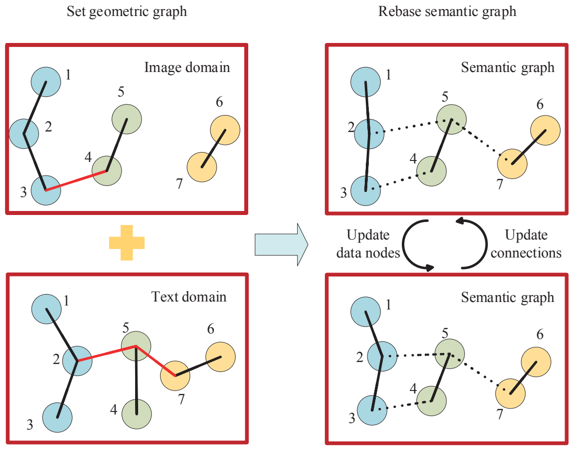


Figure 1: The schematic diagram of our setting (left) and rebasing (right) process. Dotted lines of semantic graph mean probability of these edges are smaller and solid lines represent high possibility edges. And red lines denote connections existing only in one modal.

Exempted from manual data labeling, unsupervised cross-modal hashing is regarded as a more practical alternative compared with its supervised counterpart. There are two main kinds of unsupervised cross-modal retrieval algorithms nowadays, whose main focus lie on quantization and similarity-search respectively. Cross-modal quantization minimizes the gap between binary codes and low dimension projection of origin data [Zhang and Wang, 2016; Irie *et al.*, 2015; Long *et al.*, 2016]. The second kind cross-modal similarity-search methods includes Cross View Hashing (CVH) [Kumar and Udupa, 2011], Collective Matrix Factorization Hashing (CMFH) [Ding *et al.*, 2014], Predictable Dual-view Hashing (PDH) [Rastegari *et al.*, 2013] and Inter-Media Hashing (IMH) [Song *et al.*, 2013]. Though impressive progress have these models made, there still exist several challenges in this field. Hence, we motivate our work according to the following three issues.

First, the hardness in determining data semantic relations without label. Unsupervised hashing techniques basically have no access to the actual data semantics. To obtain reasonable retrieval results, several existing single-modal [Liu *et al.*, 2011; 2014; Su *et al.*, 2018; Zhang *et al.*, 2018b;

[*]Equal contribution.
[†]Contact Author.

Liu *et al.*, 2017] and cross-modal approaches [Jian *et al.*, 2018] resort to the heuristic geometric prior, *i.e.*, determining the degree of data relevance according to the original feature distances and leaving it fixed during training. This solution is obviously sub-optimal as the original features are usually not designed for nearest-neighbor search and the distances between them can be heavily biased. One solution is to gradually update these semantic relations during training, which has been proven to be effective in [Shen *et al.*, 2018]. However, [Shen *et al.*, 2018] is designed for single-modal hashing only, fail to handle multi-modal data.

Second, the cross-modal consistency of empirical similarity. With the heuristic semantic simulation discussed above, each modality would have its own similarity connections according to its geometric prior. It is possible that two samples undergo similarity disagreement under different modalities/views, which confuses the training process. This problem also necessities similarity updating together with optimization to come up with a modal-unified graph.

Third, the sparsity of data semantic connection during training. A sparse similarity graph saves the time of training, while a densely-connected one may introduce undesired noise. In this sense, one needs to simplify the similarity graph during training, only keeping the manifest connections.

In this paper, to tackle the aforementioned issues, we propose a novel unsupervised method called Semantic-Rebased Cross-modal Hashing (SRCH). We define a special '*Set-and-Rebase*' routine to learn semantic-aware graphs for better encoding performance. The '*set*' operation constructs a geometric sparse graph which contains unimodal neighborhood relationship in each modal, and then the '*rebase*' operation is alternately coupled with binary code learning to tune and fit the geometric graph structures according to the code learning results. Our method uses sparse graph structure inspired by RCC [Shah and Koltun, 2017] and COMIC [Peng *et al.*, 2019] to preserve similarity information hidden in original data from different modals. To map data from different modalities into one common space, we suppose generated binary codes of the same sample from different modals are the same on training set for simplicity. This is reasonable because different modal data of a sample describes the same object and can use only one code to represent it. As a powerful unsupervised convention, the auto-encoding fashion is employed to improve the robustness of our model. The overview structure of our method can be found in Fig. 2. The main contributions of this method are summarized as below.

1) We propose a '*Set-and-Rebase*' mechanism to learn a sparse graph structure over the training set including geometric and semantic graphs to preserve similarity information for binary code learning.
2) Different from those existing unsupervised cross-modal hashing methods, our method focus on both similarity preserving and quantization to gain satisfied retrieval performance. Besides, the auto-encoding structure is included to improve our model, which is seldom used in cross-modal hashing.
3) Comprehensive experimental evaluations are conducted on four popular datasets, including Wiki [Rasiwasia *et al.*, 2010], MIRFlickr-25K [Huiskes and Lew, 2008], MSCOCO [Lin *et al.*, 2014] and NUS-WIDE [Chua *et al.*, 2009], showing the proposed model significantly outperforms the state-of-the-art unsupervised methods.

2 Methodology

Although our method can be used in multi-modal datasets, we conduct our experiment on two modal datasets for simplicity. Let $\boldsymbol{X}_V \in \mathbb{R}^{d_V \times n}$ and $\boldsymbol{X}_T \in \mathbb{R}^{d_T \times n}$ represents normalized image features and text vectors respectively of n training samples, where d_V and d_T are the dimensions of image feature and text vector respectively. Our task is to map these image features and text vectors into l bit binary hashing codes $\boldsymbol{B}_V$ or $\boldsymbol{B}_T$ where $\boldsymbol{B}_g \in \{+1, -1\}^{l \times n}, g \in \{V, T\}$. Because we want to map image features and corresponding text vectors into the same Hamming space in training process, we set $\boldsymbol{B}_V = \boldsymbol{B}_T = \boldsymbol{B}$ on training set for simplicity.

2.1 Model Overview

The overall pipeline of SRCH is shown in Fig. 2. Our model follows an auto-encoding schema where image and text samples feed their own projectors. The latent codes on the bottleneck are therefore quantized as the final hash codes. We *set*, *i.e.*, initialize, the cross-modal semantic graph according to the geometric prior of the original feature space and *rebase* the graph together with other model parameters in an alternating manner.

2.2 Sparse Graph *Setting* and *Rebasing*

Set the Semantic Geometric Prior

In our setting stage, geometric sparse graph structures in different modalities are built, and these geometric graphs are fixed during training. The ways to construct sparse graph structure are various due to different algorithms. One typical method is to search the nearest several nodes for any node, which can avoid isolated points and is more flexible. We use this strategy to construct our sparse graph which can be formulated like Eq. (1). $e_{(i,j)}$ in equation represents undirected edge connecting sample x_i and x_j, and $NN(x_j, k)$ means the set of k nearest neighbors of sample x_j. The threshold of the neighbor number k decides the amount of containing similarity information,

$$e_{(i,j)} = \begin{cases} 1, \; if \; x_i \in NN(x_j, k) \\ 0, \; if \; x_i \notin NN(x_j, k) \end{cases}. \tag{1}$$

The left half of Fig. 1 is a schematic diagram of our geometric sparse graph constructed on both image (up) and text (down) domain. The circles with the same color are samples from the same class and there are some wrong connections with red color in this figure due to the lack of label information as these sample pairs are close to each other in specific domain, like a cat and a dog. Those close samples are connected together through their similarity information, constituting several clusters whose number depends on the threshold.

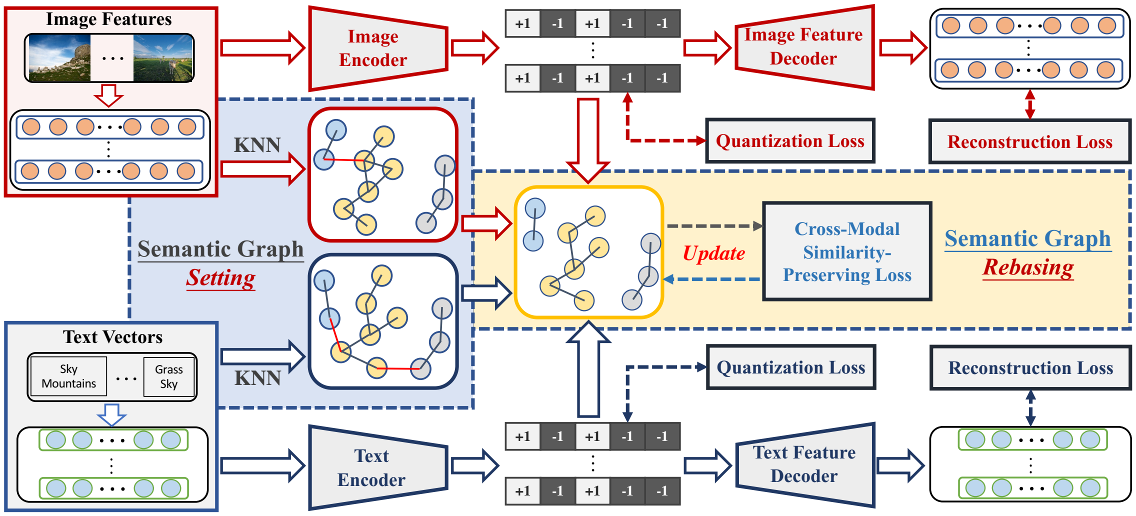


Figure 2: The framework of our proposed Semantic-Rebased Cross-modal Hashing (SRCH). The *Set* and *Rebase* steps plotted in dashed boxes which gradually learns data relevance throughout the training process.

Rebase the Graph by Code Similarity

In rebasing stage, we use continuous real values ranging from 0 to 1 to represent the probability of semantic connection between samples pairs instead of binary values to perform graph fusion. These probability values are learned according to the similarity of two samples in various modalities written as Eq. (2), where the detailed representation of $\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j)$ will be given in following part. In general, the probability of those connections which exist in both graphs are larger than those only exist in one graph. For those connections which neither exist in image domain nor in text domain, the probability values are set to zero. The iterative learning process is illustrated as the right half of Fig. 1.

$$S_{(i,j)} = p(e_{(i,j)} = 1) = \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{2}$$

2.3 Objective Function

Auto-Encoding Objective

In order to generate efficient binary codes which contain sufficient similarity information in both image domain and text domain, the structure of auto-encoder is utilized with its loss function as formulated below.

$$\mathcal{L}_g^A = \|\boldsymbol{W}_g \boldsymbol{X}_g - \boldsymbol{B}\|_F^2 + \left\|\boldsymbol{X}_g - \boldsymbol{W}_g^T \boldsymbol{B}\right\|_F^2, \tag{3}$$
$$s.t.\ \boldsymbol{B} \in \{+1, -1\}^{l \times n}, \boldsymbol{B}\boldsymbol{B}^T = n\boldsymbol{I}, \boldsymbol{W}_g^T \boldsymbol{W}_g = \boldsymbol{I},$$

where $g \in \{V, T\}$ and the second constraint $\boldsymbol{B}\boldsymbol{B}^T = n\boldsymbol{I}$ is aimed to generate mutually independent binary codes.

In Eq. (3), the first item aims to reduce the gap between mapped low dimension feature and discrete binary codes. For simplicity, the transpose matrix of $\boldsymbol{W}$ is used to replace the inverse mapping matrix and thus a constraint $\boldsymbol{W}_g^T \boldsymbol{W}_g = \boldsymbol{I}$ is added on the projection matrix. In this case the regularization term $\|\boldsymbol{W}_g\|_F^2$ is unnecessary due to the fact that $\|\boldsymbol{W}_g\|_F^2 = tr(\boldsymbol{W}_g^T \boldsymbol{W}_g) = tr(\boldsymbol{I}) = const.$

Semantic Loss

Aiming to allow generated binary codes to retrieve cross-modal samples, a sparse semantic graph S is learned from both image and text set and is used to improve the quality of hashing codes B and $S_{(i,j)}$ represents the probability of connection between sample i and j. The updating process of S will be presented detailedly in the next part.

$$\mathcal{L}_g^S = \sum_{(i,j)\in\varepsilon_g} C_{g(i,j)} \left\|S_{(i,j)}(\boldsymbol{z}_i - \boldsymbol{z}_j)\right\|_2^2. \tag{4}$$

Here $\boldsymbol{Z}$ is continuous low-dimensional embedding of input features which is shared by both modalities similar to binary codes of training set B. For simplicity, we suppose the dimension of Z is equal to B. In this equation, binary codes B is replaced by Z due to that the distance of continuous Euclidean space contain more information than that of discrete Hamming space. Besides, we hope they are close to each other, as formulated below.

$$\mathcal{L}_Z = \frac{1}{2}\|\boldsymbol{Z} - \boldsymbol{B}\|_F^2. \tag{5}$$

Besides, the symbol ε_g in Eq. (4) is our constructed sparse graph in image domain ($g = V$) or text domain ($g = T$), and $C_g (g \in \{V, T\})$ are the weights of edges in this sparse graph to balance the contribution of each data point in objective function. And the computation of $C_{g(i,j)}$ is presented in Eq. (6).

$$C_{g(i,j)} = \frac{\frac{1}{N}\sum_{m=1}^{N} a_{g(m)}}{\sqrt{a_{g(i)} a_{g(j)}}}. \tag{6}$$

The variable $a_{g(m)}$ in Eq. (6) is the degree of m-th data point in the graph and the numerator is the average degree of data points.

Toward Semantic Graph Sparsity

To strengthen the semantic connection for edges which exist in ε_g, Least Square Error constraint is added on semantic graph S with all ones matrix. And the objective function in each domain $g \in \{V, T\}$ is presented below.

$$\mathcal{L}_g^R = \sum_{(i,j)\in\varepsilon_g} C_{g(i,j)}(S_{(i,j)} - 1)^2, \tag{7}$$

where $g \in \{V, T\}$.

Our final loss of each modality is presented as following.

$$\mathcal{L}_g = \mathcal{L}_g^A + \lambda\mathcal{L}_g^S + \alpha\mathcal{L}_g^R + \beta\mathcal{L}_Z. \tag{8}$$

And our whole loss can be written as:

$$\mathcal{L} = \mathcal{L}_V + \mathcal{L}_T. \tag{9}$$

2.4 Optimization

In this part, we try to solve the optimal value of $\boldsymbol{B}$ and $\boldsymbol{W}_g$ from Eq. (9). As there are four variables in total and they are coupled with each other, the problem is split into four steps as stated below.

W-step. In this stage, $\boldsymbol{B}$ is fixed. Considering the constraint $\boldsymbol{W}_g^T\boldsymbol{W}_g = \boldsymbol{I}$, the whole loss function related to $\boldsymbol{W}_g$ can be simplified as following expression,

$$\max_{\boldsymbol{W}_g} tr(\boldsymbol{W}_g\boldsymbol{X}_g\boldsymbol{B}^T), s.t.\ \boldsymbol{W}_g^T\boldsymbol{W}_g = \boldsymbol{I}, \tag{10}$$

where the condition $\boldsymbol{B}\boldsymbol{B}^T = n\boldsymbol{I}$ is also used during simplifying process. The optimal solution of $\boldsymbol{W}_g$ can be written in closed-form as Eq. (11) with SVD algorithm, which is proved in [Hu *et al.*, 2018],

$$\boldsymbol{W}_g^* = \boldsymbol{Q}\boldsymbol{U}^T, \tag{11}$$

where $\boldsymbol{U}$ and $\boldsymbol{Q}$ are the left and right singular vectors of the compact Singular Value Decomposition (SVD) of $\boldsymbol{X}_g\boldsymbol{B}^T$.

Z-step. Fix all the other variables except Z, and then take the derivative of (9) with regard to $\boldsymbol{Z}$, we can get

$$\frac{\partial\mathcal{L}_{\boldsymbol{Z}}}{\partial\boldsymbol{Z}} = 2\beta(\boldsymbol{Z} - \boldsymbol{B}) + 2\lambda\boldsymbol{Z}\boldsymbol{H}, \tag{12}$$

where

$$\boldsymbol{H}_g = \sum_{(i,j)\in\varepsilon_g} C_{g(i,j)}S_{(i,j)}^2(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T, \tag{13}$$

and $\boldsymbol{H} = \boldsymbol{H}_I + \boldsymbol{H}_T$. From Eq. (12), we can directly get the optimal solution of $\boldsymbol{Z}^*$, presented as below,

$$\boldsymbol{Z}^* = \beta\boldsymbol{B}(\beta\boldsymbol{I}_n + \lambda\boldsymbol{H})^{-1}. \tag{14}$$

S-step (graph *rebase*). Due to the complexity of optimizing whole semantic graph $\boldsymbol{S}$, we update the graph values by element for simplicity and thus the sub-problem turns into a quadratic optimization of scalars. And we can get the solution of $S_{(i,j)}^*$ directly through its gradient,

$$S_{(i,j)}^* = \frac{\alpha}{\alpha + \lambda\left\|\boldsymbol{z}_i - \boldsymbol{z}_j\right\|_2^2} = \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{15}$$

from which we can find that $S_{(i,j)} = 1$ only when $\boldsymbol{z}_i = \boldsymbol{z}_j$. And in other cases, $S_{(i,j)}$ is ranging from 0 to 1.

B-step. Due to the discreteness of hash codes $\boldsymbol{B}$, the gradient method is not suitable for solving it from such a loss function. We rewrite Eq. (9) related to $\boldsymbol{B}$ as following expression,

$$\max_{\boldsymbol{B}} tr(\boldsymbol{B}^T(\beta\boldsymbol{Z} + \sum_{g\in\{V,T\}} 2\boldsymbol{W}_g\boldsymbol{X}_g)), \tag{16}$$

$$s.t.\ \boldsymbol{B} \in \{+1, -1\}^{l\times n}, \boldsymbol{B}\boldsymbol{B}^T = n\boldsymbol{I},$$

where the condition $\boldsymbol{B}\boldsymbol{B}^T = n\boldsymbol{I}$ is used in (16). In fact, here $\boldsymbol{B}$ can also be solved by SVD like $\boldsymbol{W}_g$, just dividing $\boldsymbol{B}$ by $\sqrt{n}$, and then binarize intermediate solution. However, two-step optimization can also increase potential quantization loss and we omit the constraint when solving $\boldsymbol{B}$ like ITQ [Gong *et al.*, 2013]. Thus, the optimal binary code $\boldsymbol{B}^*$ can be obtained as follows,

$$\boldsymbol{B}^* = sign(\beta\boldsymbol{Z} + \sum_{g\in\{V,T\}} 2\boldsymbol{W}_g\boldsymbol{X}_g). \tag{17}$$

Algorithm 1 The detailed process of SRCH

```
Input:
    Normalized image Features $X_V$ and text Vectors $X_T$;
    Parameters k, $\alpha$, $\beta$, and $\lambda$;
Output:
    Projection matrices in image domain $W_V$ and text domain $W_T$; Hash codes $B$;
Construct a sparse graph $\varepsilon_g$ of $X_g$ using m-kNN algorithm on image ($g = V$) and text ($g = T$) domain respectively;
Pre-compute weights of $\varepsilon_g$ using Eq. (6);
Initialize hash codes of training set $B$ and sparse semantic graph $S$ (Graph Set);
while stopping criterion not satisfied do
    Fix $B$, optimize $W_g, g \in \{V, T\}$ using Eq. (11);
    Fix $W_g (g \in \{V, T\})$, $B$ and $S$, update $Z$ with Eq. (14);
    Fix $Z$, update $S$ using Eq. (15) (Graph Rebase);
    Fix $W_g (g \in \{V, T\})$ and $Z$, update $B$ using Eq. (17).
end while
```

Since the iterative processes of $\boldsymbol{W}_g$, $\boldsymbol{Z}$, $\boldsymbol{S}$ and $\boldsymbol{B}$ are dependent on each other, at the beginning of iteration, initializing the value of $\boldsymbol{B}$ and $\boldsymbol{S}$ is required. For simplicity, we set the initial value of $\boldsymbol{B}$ as uniformly distributed random integer from the set of $\{+1, -1\}$. As for $\boldsymbol{S}$, we set $S_{(i,j)}$ with 1 if the edge (i, j) exists in at least one geometric sparse graph and 0 in other cases. When the optimization begins, the objective function Eq. (9) is computed in iterative manner. The stopping criterion works when the difference of the objective function value between two nearest iterations is less than a preset threshold, and the latest value of $\boldsymbol{W}_g$, $\boldsymbol{Z}$, $\boldsymbol{S}$ and $\boldsymbol{B}$ is our optimal result when the stopping condition is met. It should be noted that the objective function value is set to infinity at the beginning of the iteration. The whole iterative optimization process is summarized in Alg. 1.

2.5 Out-of-Sample Code Computation

For those samples which are outside the training set, their binary codes in image domain ($\boldsymbol{B}_I$) or text domain ($\boldsymbol{B}_T$) can be computed using following formulation.

$$\boldsymbol{B}_g = sign(\boldsymbol{W}_g^*\boldsymbol{X}_g), \tag{18}$$

where $g \in \{V, T\}$. Therefore, hashing codes of those data from test set or the whole retrieval set can be calculated with optimal projection matrix $\boldsymbol{W}_g^*$.

Methods(Task)	Wiki			MIRFlickr-25k			MSCOCO			NUS-WIDE		
	16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit
CVH(I2T) [Kumar and Udupa, 2011]	0.1596	0.1440	0.1317	0.5792	0.5652	0.5652	0.4987	0.4712	0.4434	0.4006	0.3818	0.3702
CMFH(I2T) [Ding *et al.*, 2014]	0.1734	0.1688	0.1844	0.5804	0.5729	0.5545	0.4424	0.4230	0.4922	0.3817	0.4298	0.4168
PDH(I2T) [Rastegari *et al.*, 2013]	0.1964	0.1679	0.1504	0.5441	0.5443	0.5458	0.4373	0.4389	0.4392	0.3686	0.3686	0.3685
ACQ(I2T) [Irie *et al.*, 2015]	0.1259	0.1200	0.1148	0.6174	0.5942	0.5756	0.5589	0.5529	0.5147	0.4400	0.4160	0.3958
IMH(I2T) [Song *et al.*, 2013]	0.1512	0.1447	0.1331	0.5578	0.5659	0.5595	0.4163	0.4353	0.4428	0.3496	0.3567	0.3707
QCH(I2T) [Wu *et al.*, 2015]	0.1595	0.1439	0.1318	0.5797	0.5659	0.5549	0.4959	0.4694	0.4419	0.4015	0.3824	0.3707
UGACH(I2T) [Jian *et al.*, 2018]	0.3593	0.3759	0.3852	0.6430	0.6787	0.6801	0.5498	0.5837	0.5991	0.5408	0.5345	0.5537
UCH*(I2T) [Chao *et al.*, 2019]	—	—	—	0.6540	0.6690	0.6790	0.4470	0.4710	0.4850	—	—	—
SCM(I2T) [Zhang and Li, 2014]	0.1689	0.1483	0.1364	0.6757	0.6900	0.6903	0.5931	0.6025	0.6209	0.5314	0.5551	0.5563
Our SRCH(I2T)	**0.3739**	**0.3800**	**0.3914**	**0.6808**	**0.6916**	**0.6997**	**0.5978**	**0.6052**	**0.6226**	**0.5441**	**0.5565**	**0.5671**
CVH(T2I) [Kumar and Udupa, 2011]	0.3416	0.2891	0.2454	0.5840	0.5667	0.5667	0.5072	0.4788	0.4457	0.4051	0.3846	0.3721
CMFH(T2I) [Ding *et al.*, 2014]	0.1758	0.1698	0.1793	0.5834	0.5669	0.5561	0.4532	0.4351	0.4993	0.3940	0.4515	0.4477
PDH(T2I) [Rastegari *et al.*, 2013]	0.3448	0.2926	0.2512	0.5443	0.5441	0.5461	0.4370	0.4399	0.4402	0.3664	0.3667	0.3670
ACQ(T2I) [Irie *et al.*, 2015]	0.3435	0.2912	0.2471	0.6281	0.6015	0.5806	0.5650	0.5606	0.5197	0.4452	0.4198	0.3988
IMH(T2I) [Song *et al.*, 2013]	0.2363	0.2366	0.2183	0.5608	0.5693	0.5632	0.4130	0.4349	0.4426	0.3503	0.3562	0.3717
QCH(T2I) [Wu *et al.*, 2015]	0.3414	0.2894	0.2455	0.5850	0.5672	0.5567	0.5054	0.4778	0.4450	0.4057	0.3851	0.3725
UGACH(T2I) [Jian *et al.*, 2018]	0.3374	0.3673	0.3805	0.6564	0.6918	0.6990	0.5661	0.5948	0.6069	0.5449	0.5540	0.5654
UCH*(T2I) [Chao *et al.*, 2019]	—	—	—	0.6610	0.6670	0.6680	0.4460	0.4690	0.4880	—	—	—
SCM(T2I) [Zhang and Li, 2014]	0.2949	0.2545	0.2197	0.6275	0.6446	0.6787	0.5236	0.5161	0.5337	0.5233	0.5669	0.5569
Our SRCH(T2I)	**0.3766**	**0.4006**	**0.4061**	**0.6971**	**0.7081**	**0.7146**	**0.6003**	**0.6060**	**0.6228**	**0.5533**	**0.5670**	**0.5754**

Table 1: MAP results for various code lengths of text retrieval performance by image query (I2T) and image retrieval performance by text query (T2I). In this table '*' on the right of methods' names represents the values are according to results in their original paper, and '—' means not reported.

Dataset	Wiki & MSCOCO	MIRFlickr-25k & NUS-WIDE
Setting Reference	[Irie *et al.*, 2015]	[Jian *et al.*, 2018]

Table 2: Experimental settings on four datasets.

Methods (Task)	Wiki			MSCOCO		
	16bit	32bit	64bit	16bit	32bit	64bit
SRCH (I2T)	0.3739	0.3800	0.3914	0.5978	0.6052	0.6226
SRCH w/o QL (I2T)	0.3134	0.3585	0.3800	0.5916	0.6033	0.6184
SRCH w/o RL (I2T)	0.3016	0.3308	0.3592	0.5688	0.5924	0.6162
SRCH w/o SL (I2T)	0.2135	0.2576	0.2601	0.4815	0.5181	0.5201
SRCH (T2I)	0.3766	0.4006	0.4061	0.6003	0.6060	0.6228
SRCH w/o QL (T2I)	0.3304	0.3826	0.4019	0.5917	0.6029	0.6198
SRCH w/o RL (T2I)	0.3159	0.3362	0.3778	0.5659	0.5930	0.6204
SRCH w/o SL (T2I)	0.1741	0.2208	0.2427	0.4961	0.5074	0.5086

Table 3: MAP of ablation study on Wiki and MSCOCO.

3 Experiments

3.1 Experimental Settings

Datasets. We conduct our experiments on four typical datasets, including Wiki [Rasiwasia *et al.*, 2010], MIRFlickr-25K [Huiskes and Lew, 2008], MSCOCO [Lin *et al.*, 2014] and NUS-WIDE [Chua *et al.*, 2009]. Detailed experimental settings are listed as Tab. 2.

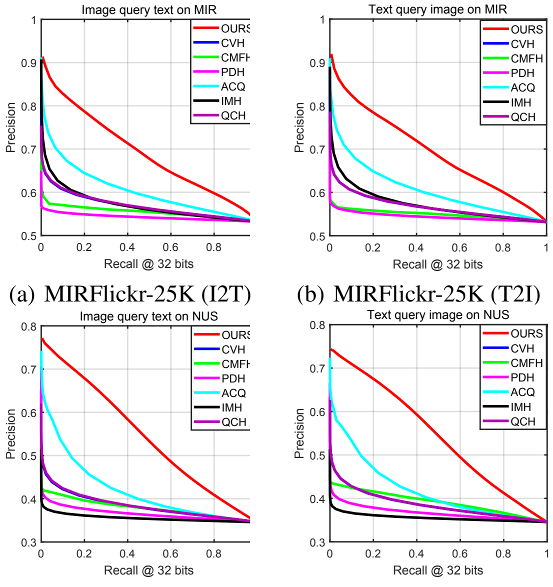
Implementation details. For all of our experiments, we follow the recent convention to use the VGG-16 `fc7` features as the image-side input with the dimension of 4096 and the universal sentence encoder feature [Cer *et al.*, 2018] for text representation whose dimension is 512. α, β, and λ are all hyper parameters of our experiment, and their values are set to 0.0001, 0.001 and 10 respectively in our model. And parameter k in the m-kNN algorithm is set to 10.

3.2 Comparison with Existing Methods

Baselines. As unsupervised cross-modal hashing algorithms are limited till now, especially in recent years, we compare our SRCH with six non-deep methods in this paper, including CVH [Kumar and Udupa, 2011], CMFH [Ding *et al.*, 2014], PDH [Rastegari *et al.*, 2013], ACQ [Irie *et al.*, 2015], IMH [Song *et al.*, 2013] and QCH [Wu *et al.*, 2015]. Besides, we also make comparisons with one supervised non-deep model SCM [Zhang and Li, 2014] and two deep unsupervised models UGACH [Jian *et al.*, 2018] and UCH [Chao *et al.*, 2019] to prove our improvement. It is noticeable that these methods except SCM are all unsupervised cross-modal hashing techniques. All of these methods use identical features to ours as inputs and we reproduce all results using the codes provided by the original authors.

Quantitative results. Results of Mean Average Precision (MAP) on text retrieval by image query (I2T) and image retrieval by text query (T2I) are listed in Tab. 1. It can be observed that our method outperforms other methods on all four datasets regardless of the cross-modal retrieval tasks and code lengths, which demonstrates the effectiveness of this method. Concretely, our text retrieval performance by image query on Wiki dataset obtains at least 17% improvement on 16 bits, 32 bits and 64 bits compared with the other non-deep algorithms while our image retrieval performance by text query surpasses those compared methods more than 3.18%, 3.33% and 2.55% for different lengths of binary codes. On the other three datasets, the improvements of our method are also obvious, especially compared with those unsupervised non-deep cross-modal hashing techniques. The corresponding Precision-Recall (P-R) curves of all unsupervised non-deep cross-modal hashing techniques are also reported in Fig. 3.

Qualitative results. Some retrieval results are selectively reported to illustrate the empirical performance of our model. Fig. 4 shows a randomly picked T2I query and the corresponding retrieval results. Our method manages to retrieve correct images while other methods have some failures according to label matching results. Concretely, our top 10 retrieval images contain the majority of text information while others' correct results only match one keyword with query sentence.



(a) MIRFlickr-25K (I2T) (b) MIRFlickr-25K (T2I)

(c) NUS-WIDE (I2T) (d) NUS-WIDE (T2I)

Figure 3: Results of Precision VS Recall Curves of various non-deep unsupervised cross-modal hashing methods on datasets MIRFlickr-25K and NUS-WIDE with 32-bit codes.

3.3 Ablation Study

Component analysis. In this section, the structure of our model is modified to see the significance of each component. For simplicity, the first item of Eq. (3) is abbreviated as QL (Quantization Loss) while the second term is called as RL (Reconstruction Loss), and the last three terms of Eq. (8) are simplified as SL (Similarity-preserving Loss). The results of our ablation study are listed in Tab. 3 and the symbol 'w/o' in table means 'without'. Tab. 3 suggests that removing any component of our learning objective leads to retrieval performance degradation. Among these three loss structures, the effect of similarity preserving loss is larger than the other two parts of losses. Specifically, this part of loss can improve at least 8.7% on MSCOCO. The quantization loss and reconstruction loss have similar effects on cross-modal results while the later one appears to be more important. From the table, we can find that all these components are significant in our method, and this experiment reflects our advantage compared with other unsupervised methods.

Hyper-parameters. The hyper-parameters are also analyzed. We illustrate the influence of different loss penalties of α, β and λ in Fig. 5 (a), (b) and (c) respectively on the Wiki dataset [Rasiwasia *et al.*, 2010]. It can be clearly seen that our model is not extremely sensitive to these penalty weights, suggesting that it can be conveniently trained and extended on other datasets and the results are highly reproducible with minimal training tricks. We also evaluate different values of k for the *Set* operation of our training graph. Again, the model accepts different values of k as our *Rebase* step can always find the optimal graph connectivity for hashing during training. The idea of involving graph update in training is therefore endorsed as the graph initialization is no longer dominating the final retrieval performance.

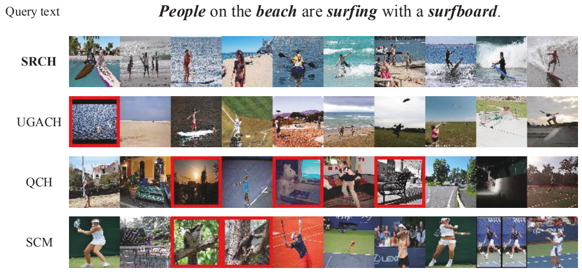


Figure 4: Qualitative results on MSCOCO Dataset with random query text written on the top through 32-bit hashing codes. Returned samples with red boxes are false-positive candidates.

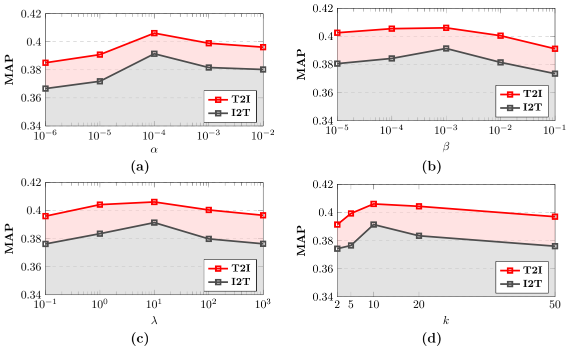


Figure 5: Hyper-parameter analysis of **(a)**: α, **(b)**: β, **(c)**: λ and **(d)**: k for the *Set* step of graph with KNN on the Wiki dataset.

4 Conclusion

This paper proposed a new kind of unsupervised cross-modal hashing method which utilized sparse graph structures to exploit similarity information to address the degradation problem in unsupervised algorithms. We made full use of similarity-preserving and quantization strategies along with reconstruction, and therefore this method can obtain more satisfied performance than other unsupervised hashing algorithms. This advantage can be found in terms of MAP values, P-R curves and qualitative retrieving results on four popular cross-modal retrieval datasets above. Furthermore, our ablation study and hyper-parameter analysis demonstrated the effectiveness of this model in many aspects.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No.61872187) and in part by the "111" Program (No.B13022).

References

[Bronstein *et al.*, 2010] Michael M Bronstein, Alexander M Bronstein, Fabrice Michel, and Nikos Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, pages 3594–3601, 2010.

[Cer *et al.*, 2018] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[Chao *et al.*, 2019] Li Chao, Deng Cheng, Wang Lei, Xie De, and Liu Xianglong. Coupled cyclegan: Unsupervised hashing network for cross-modal retrieval. *arXiv preprint arXiv:1903.02149*, 2019.

[Chua *et al.*, 2009] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, 2009.

[Ding *et al.*, 2014] Guiguang Ding, Yuchen Guo, and Jile Zhou. Collective matrix factorization hashing for multimodal data. In *CVPR*, pages 2083–2090, 2014.

[Erin Liong *et al.*, 2017] Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. Cross-modal deep variational hashing. In *ICCV*, pages 4077–4085, 2017.

[Gong *et al.*, 2013] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE TPAMI*, 35(12):2916–2929, 2013.

[Hu *et al.*, 2018] Di Hu, Feiping Nie, and Xuelong Li. Discrete spectral hashing for efficient similarity retrieval. *IEEE TIP*, 28(3):1080–1091, 2018.

[Huiskes and Lew, 2008] Mark Huiskes and Michael Lew. The mir flickr retrieval evaluation. In *ICMIR*, pages 39–43. ACM, 2008.

[Irie *et al.*, 2015] Go Irie, Hiroyuki Arai, and Yukinobu Taniguchi. Alternating co-quantization for cross-modal hashing. In *ICCV*, pages 1886–1894, 2015.

[Jian *et al.*, 2018] Zhang Jian, Peng Yuxin, and Yuan Mingkuan. Unsupervised generative adversarial cross-modal hashing. In *AAAI*, pages 539–546, 2018.

[Jiang and Li, 2017] Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. In *CVPR*, pages 3232–3240, 2017.

[Kumar and Udupa, 2011] Shaishav Kumar and Raghavendra Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, pages 1360–1365, 2011.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[Liu *et al.*, 2011] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *ICML*, 2011.

[Liu *et al.*, 2014] Wei Liu, Cun Mu, Sanjiv Kumar, and Shih-Fu Chang. Discrete graph hashing. In *NeurIPS*, pages 3419–3427, 2014.

[Liu *et al.*, 2017] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*, pages 2862–2871, 2017.

[Long *et al.*, 2016] Mingsheng Long, Yue Cao, Jianmin Wang, and Philip S Yu. Composite correlation quantization for efficient multimodal retrieval. In *ACM SIGIR*, pages 579–588, 2016.

[Peng *et al.*, 2019] Xi Peng, Zhenyu Huang, Jiancheng Lv, Hongyuan Zhu, and Joey Tianyi Zhou. Comic: Multi-view clustering without parameter selection. In *ICML*, pages 5092–5101, 2019.

[Rasiwasia *et al.*, 2010] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM MM*, pages 251–260, 2010.

[Rastegari *et al.*, 2013] Mohammad Rastegari, Jonghyun Choi, Shobeir Fakhraei, Daume Hal, and Larry Davis. Predictable dual-view hashing. In *ICML*, pages 1328–1336, 2013.

[Shah and Koltun, 2017] Sohil Atul Shah and Vladlen Koltun. Robust continuous clustering. *PNAS*, 114(37):9814–9819, 2017.

[Shen *et al.*, 2017] Yuming Shen, Li Liu, Ling Shao, and Jingkuan Song. Deep binaries: Encoding semantic-rich cues for efficient textual-visual cross retrieval. In *ICCV*, pages 4097–4106, 2017.

[Shen *et al.*, 2018] Fumin Shen, Yan Xu, Li Liu, Yang Yang, Zi Huang, and Heng Tao Shen. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *IEEE TPAMI*, 40(12):3034–3044, 2018.

[Song *et al.*, 2013] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *ACM SIGMOD*, pages 785–796, 2013.

[Su *et al.*, 2018] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. In *NeurIPS*, pages 798–807, 2018.

[Tang *et al.*, 2016] Jun Tang, Ke Wang, and Ling Shao. Supervised matrix factorization hashing for cross-modal retrieval. *IEEE TIP*, 25(7):3157–3166, 2016.

[Wu *et al.*, 2015] Botong Wu, Qiang Yang, Wei-Shi Zheng, Yizhou Wang, and Jingdong Wang. Quantized correlation hashing for fast cross-modal search. In *IJCAI*, pages 3946–3952, 2015.

[Zhang and Li, 2014] Dongqing Zhang and Wu-Jun Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, pages 2177–2183, 2014.

[Zhang and Wang, 2016] Ting Zhang and Jingdong Wang. Collaborative quantization for cross-modal similarity search. In *CVPR*, pages 2036–2045, 2016.

[Zhang *et al.*, 2018a] Haofeng Zhang, Li Liu, Yang Long, and Ling Shao. Unsupervised deep hashing with pseudo labels for scalable image retrieval. *IEEE TIP*, 27(4):1626–1638, 2018.

[Zhang *et al.*, 2018b] Zheng Zhang, Li Liu, Fumin Shen, Heng Tao Shen, and Ling Shao. Binary multi-view clustering. *IEEE TPAMI*, 41(7):1774–1782, 2018.