IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 December 2023, accepted 22 December 2023, date of publication 1 January 2024,
date of current version 5 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3348508

RESEARCH ARTICLE

Supervised Consensus Anchor Graph Hashing for Cross Modal Retrieval

RUI CHEN AND HONGBIN WANG
Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China
Key Laboratory of Artificial Intelligence in Yunnan Province, Kunming University of Science and Technology, Kunming 650500, China

Corresponding author: Hongbin Wang (whbin2007@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61966020, and in part by the Yunnan Natural Science Funds under Grant 202201AT070157.

ABSTRACT Cross–modal hashing has gained significant attention due to its efficient computational capabilities and impressive retrieval performance. Most supervised methods rely on the auxiliary learning of a similarity matrix, which incurs computational and storage expenses with a complexity of $O(n^2)$. By capturing the adjacency relationships between anchor points and original data, the anchor graph learning strategy effectively reduces the time complexity. However, existing anchor graph hashing methods adopt heuristic sampling strategies like k–means or random sampling to determine anchor points. Unfortunately, this approach separates from the anchor graph construction and fails to accurately capture the fine–grained similarity relationships. To overcome this limitation, we introduce a novel method called supervised consensus anchor graph hashing (SCAGH) for cross–modal retrieval with linear complexity. In SCAGH, the anchor points are automatically selected and consensus anchor graph learning is integrated in an unified framework. Through mutual collaboration, a more fine–grained and discriminative consensus anchor graph can be obtained without extra hyper–parameters. Additionally, we utilize anchor graph matrix to approximate the pairwise similarity matrix so that the high complexity can be avoided and enhance the quality of hash codes. Extensive experiments on four benchmark datasets are conducted to verify the superiority of the proposed SCAGH compared to several state–of–the–art methods.

INDEX TERMS Cross–modal hashing, consensus anchor graph learning, supervised similarity preservation.

I. INTRODUCTION

The emergence of social media platforms, video–sharing websites, and e–commerce platforms has promoted the rapid growth of multimodal data (e.g., image, text, audio and video). In response, the retrieval technology has shifted its focus from traditional single–modal approaches to cross–modal retrieval [1], [2], [3], [4], [5]. This refers to the task of retrieving relevant information from one modality (such as text) based on a query from a different modality (such as images or videos). Cross–modal hashing (CMH) enables efficient and effective similarity retrieval of multimodal data by encoding multimedia data into compact binary codes and computing hamming distances between binary codes via simple XOR operations [6], [7], [8]. As an essential component of cross–modal retrieval, CMH has attracted significant attention due to its ability to enhance the search capabilities and retrieval efficiency in many practical applications, including recommendation systems [9], [10], [11], person reidentification [12], [13], and multimedia data mining [14], [15]. CMH can be classified into supervised and unsupervised methods depending on whether semantic labels are employed. Generally, supervised methods [16], [17], [18] outperform unsupervised counterparts [19], [20], [21] since the former construct similarity matrix from category labels which preserves discriminative semantic information and incorporates it into hash codes learning. In most supervised methods, the similarity matrix is predefined using category labels and remains constant throughout the training process [22], [23], [24].

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

However, the complexity of computing and storing the pairwise similarity matrix is $O(n^2)$, which requires large memory cost and cannot be applied for large–scale datasets. Anchor graph hashing [25], [26] utilize a small set of anchor points to approximate the adjacency relations among all training data. By utilizing the strategy of anchor graph learning, the similarity relationship and neighborhood structure can be maintained and the computational cost can be reduced. In previous anchor graph hashing [27], [28], [29], [30], [31], the anchor points are determined through random sampling from the original training data or performing k–means clustering to obtain cluster centers that act as anchor points. The anchors generated by heuristic sampling strategy are not representative and affect the quality of anchor graph construction. Besides, anchor points selection is isolated from anchor graph construction and the anchor graph needs to be predefined to guide hash codes learning. The fixed anchor graph fails to accurately capture the fine–grained similarity relations and cannot integrate the complementary information of multimodal data which may adversely affect retrieval performance.

To address the aforementioned challenges, we propose a novel cross–modal hashing method called supervised consensus anchor graph hashing (SCAGH). This method integrates the selection of anchor points, anchor graph construction, and hash code learning within a unified framework. Unlike previous approaches that rely on heuristic sampling strategy, SCAGH automatically selects anchor points using a more effective approach. The anchor points are determined and consensus anchor graph are constructed using an integrated and flexible optimization formulation. The collaborative process ensures that the anchor points and anchor graph work together to enhance the quality of graph similarity preservation and hash code learning. Additionally, the proposed method has linear time complexity and does not involve any hyper–parameter during the construction of the anchor graph. To the best of our knowledge, our proposed method is the first to integrate anchor point selection, consensus anchor graph construction, and hash code learning into a unified framework for cross–modal hashing retrieval. The main contributions of SCAGH can be summarized as follows:

- Instead of relying on a predefined anchor sampling strategy, the proposed SCAGH integrates the learning of anchor points, consensus anchor graph construction, and hash code learning into a unified and flexible framework. This approach allows the anchor points and consensus anchor graph to interact with each other without involving any additional hyper–parameters, ultimately improving the quality of the hash codes and enhancing retrieval performance.
- To address the computation and storage costs associated with a pairwise similarity matrix, we utilize an anchor graph matrix as an approximation. This effectively reduces the complexity from $O(n^2)$ to linear time, making SCAGH suitable for large–scale cross–modal hashing tasks.
- An alternating optimization algorithm is developed to deal with the objective function. Experimental results from four benchmark datasets show the superiority and efficiency of the proposed algorithm.

II. RELATED WORK

In this section, we will briefly introduce related cross–modal hashing methods from three areas: graph–based hashing methods, anchor graph–based hashing methods and deep hashing methods.

A. GRAPH–BASED HASHING METHODS

Graph–based hashing techniques utilize a graph structure to depict and retain the neighborhood relationship among the training samples. Spectral hashing [32] constructs a completely connected Laplacian graph to preserve global similarities, and hash codes are obtained through the eigenvector computation of the Laplacian graph. Semantic neighbor graph hashing [33] constructs the semantic graph jointly pursuing semantic supervision and local neighborhood structure to preserve the fine–grained similarity matrix. Discrete multi–graph hashing [29] uses multi–graph learning to fuse the information from multiple views and designs a quantization regularization term to minimize the distortion errors. Hypergraph–based discrete hashing [34] simultaneously performs hypergraph learning and hash codes learning to enhance the semantic correlations among instances. [35] establishes multiple instance relation graphs to exploit fine–grained similarity relations between instances. Multi–view graph cross–modal hashing [36] constructs multi–view affinity and asymmetric graphs over anchor data which serve as a unified semantic hub in a semi–supervised manner. Sparse graph based self–supervised hashing [37] exploits a sparse graph structure and self–supervised reconstruction constraint to further preserve the semantic information. For the graph–based hashing methods, $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be agraph with a set of vertices $\mathbf{V}$ and edges $\mathbf{E}$. However, the value of edges invloves calculating the similarity between vertices which require $O(n^2)$ time and store complexity during the construction of the neighborhood graph. Based on this, anchor graph–based hashing methods have been developed to reduce time and memory overhead. The proposed method SCAGH employs anchor graph matrix to approximate the pairwise similarity matrix so that the high complexity can be reduced as linear.

B. ANCHOR GRAPH–BASED HASHING METHODS

Anchor graph–based methods use a small set of samples called anchors to approximate the data neighborhood structure instead of preserving the similarity of training samples. Anchor graph hashing [25] utilizes anchor graphs to autonomously uncover the innate neighborhood structure present in the data through an unsupervised approach. Based on this, [38] applies a graph clustering algorithm to compute the eigenvectors from the anchor–simple similarities matrix. Multi–view anchor graph hashing [27] employs a multi–view anchor graph to maintain the average similarity in low–rank format, which non-linearly combines binary codes. Discrete graph hashing [28] opts for anchor graphs to approximate the neighborhood structure embedded in the input data within a discrete optimization framework. Asymmetric discrete graph hashing [39] utilizes selected anchors to generate the asymmetric affinity matrix and preserve the asymmetric discrete constraint. Semi–supervised metric learning [30] learns an optimal distance metric to preserve the semantic similarity in the anchor graph. Cross–modal transfer hashing [40] utilizes the pseudo class centers as the anchors and construct the modality–specific local anchor graph to keep the intrinsic structural neighborhood relationship. [36] constructs multi–view affinity and asymmetric graphs over anchor data which serve as a unified semantic hub in a semi–supervised manner. Anchor graph structure fusion hashing [41] incorporates intrinsic anchor fusion affinity preservation and clustering structure optimization into a unified framework. Asymmetric transfer hashing [42] characterizes the domain distribution gap by minimizing two asymmetric hash functions and learns an adaptive bipartite graph to characterize the similarity between cross–domain samples. For the aforementioned anchor graph–based hashing methods, anchor points selection and anchor graph construction are separated from each other. Formally, anchor points can be determined by randomly sampling or k-means clustering to acquire cluster center as anchor points. However, the anchor points generated by random selection or k-means method are not representative and involve hyper–parameter, thereby affecting the quality of anchor graph construction and the performance of cross modal retrieval. One–pass multi–view subspace clustering [43] combines anchor learning and graph construction into a uniform framework to boost clustering performance, which inspires us to apply consensus anchor graph learning to cross–modal retrieval. SCAGH integrates anchor points selection, consensus anchor graph construction and hash codes learning into a unified framework. Anchor points are automatically chosen and the consensus anchor graph can be constructed without extra hyper–parameters via a flexible optimization formulation. The selection of anchor points is mutually collaborated with anchor graph to enhance hash code learning quality and improve cross-modal retrieval performance.

C. DEEP HASHING METHODS

Deep hashing methods leverage powerful feature representation capabilities of neural networks and end–to–end architecture to learn hash codes, achieving satisfactory performance. Deep cross–modal hashing [44] integrates feature extraction, adaptation of heterogeneous data distributions and hash codes learning in an adversarial way. Pairwise relationship guided deep hashing [45] adds inter–modal and intra–modal pairwise embedding loss to enhance the correlation between semantic similar instances. Unsupervised deep cross modal hashing [46] incorporates the unsupervised matrix factorization hashing into an end–to–end deep learning framework and the weight parameters are learned dynamically. Deep graph–neighbor coherence preserving network [47] consolidates relationships between original data and their neighbors using graph–neighbor coherence. Discrete fusion adversarial hashing [48] integrates feature extraction, adaptation of heterogeneous data distributions and hash codes learning in an adversarial way. Deep discrete cross–modal hashing [49] enhances the semantic correlation among multi–modalities with the joint supervision of instance–pairwise, instance–labeled and class wise similarities. Semantic–rebased cross–modal hashing [50] sets and rebases a sparse graph according to the feature geometric basis and hash codes to preserve similarity information for binary codes learning. Although deep hashing methods surpass shallow methods in terms of performance, they are often constrained by time–consuming training processes and complex parameter selection requirements.

Compared to graph–based hashing methods and anchor graph–based hashing methods, SCAGH integrates the learning of anchor points, consensus anchor graph construction, and hash code learning into a unified framework to enhance retrieval performance. Despite not being a deep learning–based hashing method, SCAGH exhibits low time complexity, a simple model, and performs better than deep hashing methods on specific dataset.

III. METHODOLOGY

The proposed method will be explained from three aspects including algorithm structuring, procedure optimization and complexity analysis. The formulation has three folds: latent consistency representation, similarity structure learning and hash function learning. The framework of SCAGH is illustrated in Figure 1. The optimization process introduces the derivation process of each variable and the complexity analysis shows the time complexity of the model.

A. NATIONS AND DEFINITIONS

In the following parts, we use uppercase letter such as $\mathbf{A}$ to denote matrix, the lowercase letter $\mathbf{a}_i$ denotes the i column of matrix $\mathbf{A}$. The vector with all elements 1 is represented as $\mathbf{1}$. $\mathbf{I}_n$ indicates identity matrix and the size is $n \times n$. $Tr(\cdot)$ is the trace of matrix, $diag(\cdot)$ is the diagonal matrix operator. $Sign(\cdot)$ is the sign function which outpute $+1$ if the value is positive numbers, otherwise -1. $\|\cdot\|_2$ denotes the ℓ_2–norm of vector and $\|\cdot\|_F$ represents the Frobenius norm of the matrix $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{a}_{ij}^2}$.

Suppose there are n multimodal samples $\mathbf{O} = \{\mathbf{o}_i\}_{i=1}^{n}$, $\mathbf{o}_i = \left(\mathbf{x}_i^1, \mathbf{x}_i^2\right)$ where $\mathbf{x}_i^1 \in \mathbf{R}^{d_1}$, $\mathbf{x}_i^2 \in \mathbf{R}^{d_2}$ represents the vector of i–th sample from image and text modalities, respectively. d_1 and d_2 are the dimensionality of samples for each modality. The image and text modality original data matrix is $\mathbf{X}_1 = [\mathbf{x}_1^1, \mathbf{x}_2^1, \cdots, \mathbf{x}_n^1] \in \mathbf{R}^{d_1 \times n}$, $\mathbf{X}_2 = [\mathbf{x}_1^2, \mathbf{x}_2^2, \cdots, \mathbf{x}_n^2] \in \mathbf{R}^{d_2 \times n}$, respectively. $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n] \in \mathbf{R}^{n \times d}$ is the category label matrix, d means the total number of category and $\mathbf{y}_i$ means the corresponding label of sample $\mathbf{o}_i$. Hash codes matrix is represented by $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n] \in \{-1, +1\}^{r \times n}$, where r is the hash codes length.

B. CONSENSUS ANCHOR GRAPH LEARNING

In the exciting anchor graph based hashing methods, the anchor points are fixed by k–means clustering or random sampling. After selecting the anchor points separately in each modality, the anchor graphs $\mathbf{Z}^{(m)}$ are then defined the similarities between all data samples $x_i^{(m)}$ and anchors points $u_j^{(m)}$ as follows [29]. where $[i]$ denotes the indices of nearest anchors of $x_i^{(m)}$, $D^2(x_i^{(m)}, u_j^{(m)}) = \left\| x_i^{(m)} - u_i^{(m)} \right\|^2$,

$$\mathbf{Z}_{ij}^{(m)} = \begin{cases} \frac{exp(D^2(x_i^{(m)}, u_j^{(m)})/\delta)}{\sum_{j \in [i]} exp(D^2(x_i^{(m)}, u_j^{(m)})/\delta)}, & \forall j \in [i] \\ 0, & otherwise \end{cases} \tag{1}$$

δ is the predefined bandwith parameter. The above anchor graph learning strategy has several problems. (1) The selection of anchor points and construction of anchor graphs are separated from each other. (2) The anchor graph of each modal requires to perform fusion algorithm to get a consensus graph. To overcome these problems, based on the assumption that heterogeneous data are derived from a shared consistent latent space, it is reasonable that multimodal data has common anchor points and consensus anchor graph [43], [51], [52]. In view of this, we define the respective projection matrix $\mathbf{W}_i$ of different modalities to integrate anchor points selection and consensus anchor graph construction in a unified formulation:

$$\begin{aligned} \min_{\alpha_i} \quad & \sum_{i=1}^{M} (\alpha_i)^2 \left\| \mathbf{X}_i - \mathbf{W}_i \mathbf{A} \mathbf{Z} \right\|_F^2 \\ \text{s.t.} \quad & \sum_{i=1}^{M} \alpha_i = 1, \mathbf{W_i}^T \mathbf{W_i} = \mathbf{I}, \mathbf{A}^T \mathbf{A} = \mathbf{I}, \\ & \mathbf{Z} > 0, \mathbf{Z}^T \mathbf{1} = 1, \end{aligned} \tag{2}$$

where M is the total number of modalities and we chose image and text modality for training in this paper. $\mathbf{X_i} \in \mathbf{R}^{d_m \times n}$ is the original data of i^{th} modality with d_m and n being the corresponding dimension and the size of samples. $\mathbf{W}_i \in \mathbf{R}^{d_m \times d}$ and α_i are the projection matrix and the weight of i^{th} modality respectively. $\mathbf{A} \in \mathbf{R}^{d \times l}$ represents the unified anchor points matrix, in which d and l are the common dimension and the numbers of anchors. $\mathbf{Z} \in \mathbf{R}^{l \times n}$ is the consensus anchor graph representing the similarity relation between n samples and l anchor points. Using Eq.(2), the anchor point $\mathbf{A}$ can be directly learned without the need to adopt heuristic sampling strategies such as k–means or random sampling for anchor point determination. Compared with Eq.(1), Eq.(2) automatically constructs a common consensus anchor graph $\mathbf{Z}$ for different modalities instead of calculating similarity between anchor points and samples or performing fusion algorithms in Eq.(1). In summary, Eq.(2) automatically selects anchor points and integrates consensus anchor graph learning, allowing the anchor points and consensus anchor graph to interact with each other without requiring any additional hyper–parameters.

C. SIMILARITY PERSERVE AND OBJECTIVE FUNCTION

Hash codes should preserve the neighbor similarity in the original space as much as possible. More specifically, when the instances $\mathbf{o}_i$ and $\mathbf{o}_j$ have high similarity, the Hamming distance between the hash codes $\mathbf{b}_i$ and $\mathbf{b}_j$ should be smaller, and vice versa. The approximation of the similarity between the hash codes and the original data samples can be expressed using the following formula:

$$\begin{aligned} \min_{\mathbf{B}} \quad & \left\| \mathbf{B}\mathbf{B}^T - \mathbf{S} \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, +1\}^{r \times n}, \mathbf{B}\mathbf{B}^T = \mathbf{I} \end{aligned} \tag{3}$$

$\mathbf{S}$ represents the pairwise similarity matrix constructed by $\mathbf{S} = \mathbf{Z} \Lambda \mathbf{Z^T}$, $\Lambda = diag(\mathbf{Z}^T \mathbf{1}) = \mathbf{I}$. According to [53] and [54], $\beta Tr(\mathbf{S}) \geq Tr(\mathbf{S}^T \mathbf{S})$ when $\beta \geq \Lambda_{max}$ and Λ_{max} is the largest eigen–value of matrix $\mathbf{S}$. Eq.3 can be approximated rewritten as:

$$\begin{aligned} \min_{\mathbf{B}, \mathbf{B_s}} \quad & \beta tr(\mathbf{S}) - 2tr(\mathbf{B} \mathbf{Z^T} \mathbf{Z} \mathbf{B}^T) \\ & = \beta tr(\mathbf{Z^T}\mathbf{Z}) - 2tr(\mathbf{B}\mathbf{Z^T}\mathbf{B_s}^T) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, +1\}^{r \times n}, \mathbf{B}\mathbf{B}^T = \mathbf{I} \end{aligned} \tag{4}$$

where $\mathbf{B_s} = sign(\mathbf{B}\mathbf{Z}^T) \in \{-1, +1\}^{r \times l}$ are the binary anchor points. To further improve the quality of anchor graph and fully utilize the supervised information of the category label, the label matrix is embedded into the hash codes by projection matrix P and used to associate anchor points and consensus anchor graphs:

$$\min_{\mathbf{P}, \mathbf{B}} \lambda \left\| \mathbf{Z}^T - \mathbf{Y}\mathbf{A} \right\|_F^2 + \gamma \left\| \mathbf{B} - \mathbf{P}\mathbf{Y}^T \right\|_F^2 \tag{5}$$

As images and texts are typical unstructured data and there are complex nonlinear relations in the original data which is difficult for the linear model to accurately approximate. Therefore, we adopt RBF kernel mapping to futher preseve the nonlinear underlying structure among original instances. Specifically, the q samples are randomly selected from training set as $\{a_j\}_{j=1}^{q}$ and the original data $\mathbf{X}_i$ can be transformed as kernelized feature by $\phi(\mathbf{X}_i) = \left[exp(\frac{\|x_i - a_1\|}{2\sigma^2}, \ldots, exp(\frac{\|x_i - a_q\|}{2\sigma^2}) \right]$, where σ denotes the kernel width and $\sigma = 1/nq \sum_{i=1}^{n} \sum_{j=1}^{q} \left\| x_i - a_j \right\|$. By combining Eq.(2), Eq.(4) and Eq.(5), the overall objective function is:

$$\begin{aligned} \min_{\substack{\alpha_i, \mathbf{W}_i, \mathbf{A}, \mathbf{Z}, \\ \mathbf{B}, \mathbf{B_s} \mathbf{P}}} \quad & \sum_{i=1}^{M} (\alpha_i)^2 \left\| \phi(\mathbf{X}_i) - \mathbf{W}_i \mathbf{A} \mathbf{Z} \right\|_F^2 \\ & + \beta tr(\mathbf{Z^T}\mathbf{Z}) - 2tr(\mathbf{B}\mathbf{Z^T}\mathbf{B_s}^T) \\ & + \lambda \left\| \mathbf{Z}^T - \mathbf{Y}\mathbf{A} \right\|_F^2 + \gamma \left\| \mathbf{B} - \mathbf{P}\mathbf{Y}^T \right\|_F^2 \\ \text{s.t.} \quad & \sum_{i=1}^{M} \alpha_i = 1, \mathbf{W_i}^T \mathbf{W_i} = \mathbf{I}, \mathbf{A}^T \mathbf{A} = \mathbf{I}, \end{aligned}$$

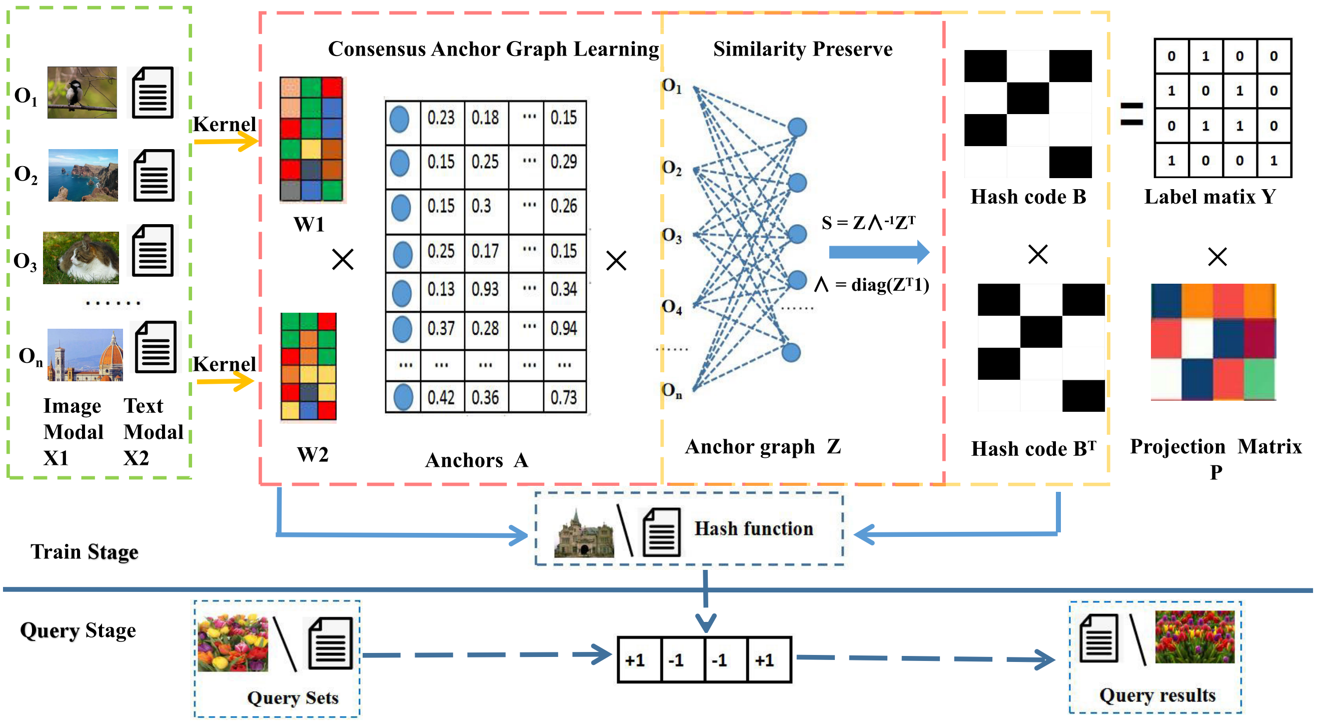


FIGURE 1. Framwork of the proposed SCAGH. SCAGH is composed of training and query two stages. The training stage includes two modules: consensus anchor graph learning and similarity preserve. During the training stage, the original data features X1, X2 are converted to kernel mapping features through kernel mapping. The kernel mapping features of different modalities are decomposed into anchor points A and consensus anchor graph Z through projection matrices W1 and W2. In this, the selection of anchor points and construction of anchor graph are seamlessly integrated into a unified process with consensus anchor graph learning. Subsequently, the consensus anchor graph Z is utilized to create the pairwise similarity matrix S, and the learned hash codes B effectively maintain the similarity between neighboring points in the original data through similarity preservation. Additionally, modality–specific linear hash functions are learned based on the training data X1, X2 and hash codes B by the linear regression. In the query stage, the query instances are transformed into binary codes using the hash functions and cross–modal retrieval can be achieved by calculating the similarity between binary codes of query instances and hash codes B of training instance.

$$\begin{aligned}&\mathbf{Z} > 0, \mathbf{Z}^T\mathbf{1} = 1\\&\mathbf{B} \in \{-1,+1\}^{r\times n}, \mathbf{B_s} \in \{-1,+1\}^{r\times l},\\&\mathbf{BB}^T = \mathbf{I}, \mathbf{PP}^T = \mathbf{I}\end{aligned}\tag{6}$$

D. OPTIMIZATION

The optimization problem in Eq.(6) is nonconvex and cannot be solved directly. It is convex and solvable when updating one of the variables and fixing others. To this end, an iterative optimization algorithm is used.

1) W_i-STEP

Fixed α_i, $\mathbf{Z}$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B_s}$ and $\mathbf{P}$, the objective function of $\mathbf{W}_i$ can be rewritten as:

$$\min_{\mathbf{W}_i} \sum_{i=1}^{M} {\alpha_i}^2 \left\|\phi\left(\mathbf{X}_i\right) - \mathbf{W}_i\mathbf{AZ}\right\|_F^2 \tag{7}$$

Since $\mathbf{W}_i$ is independent in each modal, Eq.(7) can be transformed into the following representation:

$$\begin{aligned}\max_{\mathbf{W}_i}\quad & tr(\mathbf{W}_i^T\mathbf{Q}_i)\\ \text{s.t.}\quad & \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}\end{aligned}\tag{8}$$

where $\mathbf{Q}_i = \alpha_i\phi\left(\mathbf{X}_i\right)\mathbf{Z}^T\mathbf{A}^T$, the singular value decomposition (SVD) result is $\mathbf{Q}_i = \mathbf{U}_Q\mathbf{L}_Q\mathbf{V}_Q^T$, $\mathbf{L}_Q$ is diagonal matrix, $\mathbf{U}_Q$ and $\mathbf{V}_C$ are orthogonal matrixs. The optimal solution of $\mathbf{W}_i$ can be calculated by $\mathbf{W}_i = \mathbf{U}_Q\mathbf{V}_Q^T$.

2) A-STEP

Fixed α_i, $\mathbf{W}_i$, $\mathbf{Z}$, $\mathbf{B}$, $\mathbf{P}$ and $\mathbf{B_s}$, the objective of $\mathbf{A}$ can be simplified as:

$$\begin{aligned}\min_{\mathbf{W}_i}\quad & \sum_{i=1}^{M} {\alpha_i}^2 \left\|\phi\left(\mathbf{X}_i\right) - \mathbf{W}_i\mathbf{AZ}\right\|_F^2\\ & + \lambda\left\|\mathbf{Z}^T - \mathbf{YA}\right\|_F^2\\ \text{s.t.}\quad & \mathbf{A}^T\mathbf{A} = \mathbf{I}\end{aligned}\tag{9}$$

Eq.(9) can be transformed as the trace form:

$$\begin{aligned}\max_{\mathbf{A}}\quad & tr(\mathbf{A}^T\mathbf{C})\\ \text{s.t.}\quad & \mathbf{A}^T\mathbf{A} = \mathbf{I}\end{aligned}\tag{10}$$

where $\mathbf{C} = \sum_{i=1}^{M}{\alpha_i}^2\mathbf{W}_i^T\phi\left(\mathbf{X}_i\right) + \lambda\mathbf{Y}^T\mathbf{Z}^T$, the optimal solution method of Eq.(10) is similar to Eq.(8). Taking the singular value decomposition (SVD) of $\mathbf{C}$, we can get $\mathbf{A} = \mathbf{U}_C\mathbf{V}_C^T$.

3) Z- STEP

Fixed α_i, $\mathbf{W}_i$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B_s}$ and $\mathbf{P}$, the objective function of $\mathbf{Z}$ can be written as:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \sum_{i=1}^{M} {\alpha_i}^2 \left\| \phi\left(\mathbf{X}_i\right) - \mathbf{W}_i \mathbf{A} \mathbf{Z}^T \right\|_F^2 \\ & + \lambda \left\| \mathbf{Z}^T - \mathbf{Y}\mathbf{A} \right\|_F^2 \\ & + \beta tr(\mathbf{Z}^T \mathbf{Z}) - 2tr(\mathbf{B}\mathbf{Z}^T {\mathbf{B_s}}^T) \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \quad \mathbf{Z}^T \mathbf{1} = 1 \end{aligned} \tag{11}$$

Eq.(11) can be rewritten as:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \mathbf{Z}^T (\sum_{i=1}^{M} {\alpha_i}^2 + \gamma + \lambda) \mathbf{Z} \\ & - 2(\sum_{i=1}^{M} {\alpha_i}^2 \phi\left(\mathbf{X}_i\right) \mathbf{W}_i \mathbf{A} \\ & + \lambda \mathbf{Y}\mathbf{A} + \mathbf{B}^T \mathbf{B_s}) \mathbf{Z} \end{aligned} \tag{12}$$

The Eq.(12) of $\mathbf{Z}$ can be easily transformed as the following Quadratic Programming (QP) problem:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \frac{1}{2} \mathbf{Z}_{;,\mathbf{j}}^{\mathbf{T}} \mathbf{G} \mathbf{Z} + f^T \mathbf{Z}_{;,\mathbf{j}} \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \quad \mathbf{Z}_{;,\mathbf{j}}^{\mathbf{T}} \mathbf{1} = 1 \end{aligned} \tag{13}$$

where

$$\begin{aligned} \mathbf{G} &= 2(\sum_{i=1}^{M} {\alpha_i}^2 + \gamma + \lambda) \mathbf{I} \\ f^T &= -2(\sum_{i=1}^{M} {\alpha_i}^2 \phi\left(\mathbf{X}\right)_{i[:,j]}^T \mathbf{W}_i \mathbf{A} \\ & \quad + \lambda \mathbf{Y}_{[:,j]} \mathbf{A} + \mathbf{B}_{[:,j]}^T \mathbf{B}_s) \end{aligned} \tag{14}$$

The optimization of Eq.(13) can be solved by performing the QP problem of each $\mathbf{Z}$. Specifically, because the each column of $\mathbf{Z}$ is l dimensional vector, the time complexity of solving $\mathbf{Z}$ is $O(nl^3)$.

4) P -STEP

By updating $\mathbf{P}$ while fixing other varibales, we have:

$$\begin{aligned} \min_{\mathbf{P}} \quad & \gamma \left\| \mathbf{B} - \mathbf{P}\mathbf{Y}^T \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{P}\mathbf{P}^T = \mathbf{I} \end{aligned} \tag{15}$$

Eq.(15) can be transformed as the following form:

$$\max_{\mathbf{P}} \quad tr(\mathbf{P}^T \mathbf{B} \mathbf{Y}) \tag{16}$$

Eq.(16) could be sovled by singular value decomposition (SVD) method $\mathbf{B}\mathbf{Y} = \mathbf{U}_P \mathbf{L}_P \mathbf{V}_P^T$ and the optimal solution can be got by $\mathbf{P} = \mathbf{U}_p \mathbf{V}_P^T$.

5) B -STEP

Fixed α_i, $\mathbf{W}_i$, $\mathbf{Z}$, $\mathbf{A}$, $\mathbf{B_s}$ and $\mathbf{P}$ the expression about $\mathbf{B}$ is obtained as:

$$\begin{aligned} \min_{\mathbf{B}} \quad & \gamma \left\| \mathbf{B} - \mathbf{P}\mathbf{Y}^T \right\|_F^2 - 2tr(\mathbf{B}\mathbf{Z}^T \mathbf{B_s}) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{r \times n} \end{aligned} \tag{17}$$

Since $tr(\mathbf{B}\mathbf{B}^T)$ is constants, Eq.(17) is transformed as:

$$\begin{aligned} \max_{\mathbf{B}} \quad & 2tr(\mathbf{B}(\mathbf{Z}^T {\mathbf{B_s}}^T + \gamma \mathbf{Y}\mathbf{P}^T)) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{r \times n} \end{aligned} \tag{18}$$

Finally, the solution of $\mathbf{B}$ is given by:

$$\mathbf{B} = sign(\mathbf{B_s}\mathbf{Z} + \gamma \mathbf{Y}\mathbf{P}^T) \tag{19}$$

6) B_S- STEP

By fixing other variables, the corresponding representation about $\mathbf{B_s}$ is:

$$\begin{aligned} \min_{\mathbf{B_s}} \quad & -2tr(\mathbf{B}\mathbf{Z}^T {\mathbf{B_s}}^T) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{r \times l} \end{aligned} \tag{20}$$

With the same scheme to deal with Eq.(18), this problem can be solved as follow:

$$\mathbf{B_s} = sign(\mathbf{B}\mathbf{Z}^T) \tag{21}$$

7) α_I - STEP

Fixed $\mathbf{W}_i$, $\mathbf{Z}$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B_s}$ and $\mathbf{P}$ the expression about $\mathbf{B}$ is obtained as:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \sum_{i=1}^{M} {\alpha_i}^2 \parallel \mathbf{R_i} \parallel^2 \\ \text{s.t.} \quad & \alpha^T \mathbf{1} = 1 \end{aligned} \tag{22}$$

where $\mathbf{R_i} = \sum_{i=1}^{M} \left\| \phi\left(\mathbf{X}_i\right) - \mathbf{W}_i \mathbf{A} \mathbf{Z}^T \right\|_F$. According to Cauchy-Schwarz inequality, when $\alpha_1 \mathbf{R_1} = \alpha_2 \mathbf{R_2} = \ldots = \alpha_M \mathbf{R_M}$ we can obtain the best optimal solution of α_i as follows:

$$\alpha_i = \frac{\frac{1}{\mathbf{R_i}}}{\sum_{j=1}^{M} \frac{1}{\mathbf{R_j}}} \tag{23}$$

E. HASH FUNCTION LEARNING

SCAGH is a two–step hash method which includes hash codes learning and hash function learning separately. The hash codes can be generated by above method and we adopt the linear regression to learn the modality–specific hash functions. The hash mapping matrix $\mathbf{F}_k$ can be obtained by:

$$\min_{\mathbf{F}_k} \sum_{k=1}^{M} \|\mathbf{B} - \mathbf{X}_k \mathbf{F}_k\| + \theta \|\mathbf{F}_k\|_F^2 \tag{24}$$

Algorithm 1 Optimization for SCAGH

```
Training Stage:
 Input: Data frature matrix $\mathbf{X}_1, \mathbf{X}_2$; label matrix$\mathbf{Y}$;paramters $\gamma, \lambda, \beta, \theta$ and hash codes length $r$.
 Output: Hash codes matrix $\mathbf{B}$ and hash functions $\mathbf{F}_1, \mathbf{F}_2$.
  Procedure:
1.Normalize training image and text feature $\phi$ $(\mathbf{X}_1)$,$\phi$ $(\mathbf{X}_2)$ by kernel mapping.
2.Randomly initialize $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{Z}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{B}$,$\mathbf{B_s}$ and hash functions $\mathbf{F}_1, \mathbf{F}_2$.

repeat
    updata $\alpha_1, \alpha_2$ by Eq. (23),
    updata $\mathbf{W}_1, \mathbf{W}_2$ by Eq. (8),
    updata $\mathbf{A}$ by Eq. (10),
    updata $\mathbf{Z}$ by Eq. (13),
    updata $\mathbf{B}$ by Eq. (19),
    updata $\mathbf{B_s}$ by Eq. (21),
    learning hash function$\mathbf{F}_1, \mathbf{F}_2$ by Eq. (26).
until Convergency or reach the maximum iterations
Query Stage:
 Input:Query data $\mathbf{X}_i, \mathbf{X}_t$; query label matrix$\mathbf{Y}_q$; paramters $\theta$ and hash codes length $r$.
 Output:Query image and query hash code matrix $\mathbf{B}_1, \mathbf{B}_2$.
  Procedure:
1.Normalize query image and text feature $\mathbf{X}_i$, $\mathbf{X}_t$ by means.
2.Generate image hash codes by $\mathbf{B}_1 = sign(\mathbf{F}_1\mathbf{X}_i)$.
  Generate image hash codes by $\mathbf{B}_2 = sign(\mathbf{F}_2\mathbf{X}_t)$.
```

where θ is the regularization coefficient and $\|\mathbf{F}_k\|_F^2$ is the regularization term. Eq.(24) is optimized by:

$$\min_{\mathbf{F}_k} \sum_{k=1}^{M} -2tr(\mathbf{B}^T\mathbf{X}_k\mathbf{F}_k) + tr(\mathbf{F}_k^T\mathbf{X}_k^T\mathbf{X}_k\mathbf{F}_k) + \theta tr(\mathbf{F}_k^T\mathbf{F}_k) \tag{25}$$

By taking the derivative of $\mathbf{F}_k$:

$$\mathbf{F}_k = (\mathbf{X}_k^T\mathbf{X}_k + \theta\mathbf{I})^{-1}\mathbf{X}_k^T\mathbf{B} \tag{26}$$

For the query data matrix of the k–th modality $\mathbf{X}_k^q$, the hash codes $\mathbf{B}_q$ can be generated by:

$$\mathbf{B}_q = sign(\mathbf{X}_k^q\mathbf{F}_k) \tag{27}$$

The optimization algorithm of SCAGH is summarized in Algorithm 1.

F. COMPLEXITY ANALYSIS

The time complexity $O(\cdot)$ of variables $\mathbf{B}$, $\mathbf{B_s}$, $\mathbf{P}$, $\mathbf{W}_i$ and $\mathbf{A}$ will be analyzed in this section. To be specific, the time complexity for hash codes matrix $\mathbf{B}$ and $\mathbf{B_s}$ are $O(rn(l+d))$ and O(rnl), for mapping matrix $\mathbf{P}$ and $\mathbf{W}_i$ are $O(rd(n+d))$ and $O(d_m(ln+ld+d^2))$, for adaptive anchor matrix $\mathbf{A}$ is $O(d(ln+ndm+l^2))$ for weight of modality α_i is $O(dml(d+n))$, for anchor–simple similarity matrix $\mathbf{Z}$ is $O(nl^3)$. The total computational complexity is $O(d_m(2ln+2ld+d^2)+rd(n+d)+d(ln+ndm+l^2)+rn(2l+d)+nl^3)$ where dm is the dimension of data, r is the hash codes length, l and n are the number of anchor and data. Since $dm, r, l \ll n$, the computational complexity of the proposed method is linear complexity to the number of samples n.

IV. EXPERIMENTS

A series of comparative experiments with baseline methods and deep hashing methods are conducted. The convergence property, parameter sensitivity, and time complexity of the proposed method will be discussed.

A. DATASETS INTRODUCTION

The performance of the proposed method is evaluated on three widely used datasets: WIKIData,[1] Labelme,[2] MIRFLICKR[3] and NUSWIDE10.[4]

WIKIData: this dataset originally contains 2,866 image-text pairs collected from Wikipedia, and it is splited into 2,173 training pairs and 693 testing pairs. Each sample belongs to one of 10 category labels and is described with visual modality and text modality. Visual modality is composed of 128-dimensional SIFT feature vectors and text modality is represented by 10-dimensional topic vectors.

Labelme: the dataset is made up of 2,688 samples along with 8 unique outdoor scenes categories such as "coast", "streets" and "highway" and each sample belongs to one scene. Each image is represented by a 512–dimensional GIST vector and each text is represented by 245–dimensional phrase frequency. In the experiments, 2,016 image–text pairs are randomly selected for training and the rest 672 image–text pairs are used for testing.

MIRFLICKR: contains 25,000 samples crawled from the Flickr website with 24 semantic concepts. Each image is represented by 512–dimensional GIST [55] feature vector and its corresponding text is described as a 1,386–dimensional bag–of–words vector. We remove samples without labels or textual tags less than 20 and 20,015 samples are kept for training, 2,000 samples are randomly selected as the test set.

NUSWIDE10: is a large–scale multi–label dataset, which contains 269,648 instances from 81 semantic categories. The image is described with 500–dimensional SIFT feature vectors, and the text is represented with 1,000–dimensional bag–of–words vectors. Since some labels are sparse, we keep the most common 10 concepts as the category label and the corresponding 186,577 samples are used for experiment. Among them, 184,577 samples are used for training and 2,000 samples are used for testing.

[1] http://www.svcl.ucsd.edu/projects/crossmodal/

[2] http://people.csail.mit.edu/torralba/code/spatialenvelope/

[3] https://press.liacs.nl/mirflickr/

[4] https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html

TABLE 1. The parameter values corresponding to datasets.

Datasets	γ	λ	β	θ
WIKIData	1e4	10	100	1e-2
Labelme	100	1e4	10	1e-3
MIRFLICK	100	1e-2	1e3	1e-3
NUSWIDE10	1e4	10	1e3	1e-3

B. EVALUATION METRICS

The experiments are conducted on SCAGH and other baseline methods with two retrieval tasks: (1) Img2Txt i.e., retrieving relevant text instances with image, and (2) Txt2Img i.e., retrieving relevant image instances with text. Some commonly adopted evaluation protocols include the Mean Average Precision (MAP), Precision–Recall (PR) curves, and top–N Precision (Precision@N) curves.

The definition of MAP is given by:

$$MAP = \frac{1}{N}\sum_{q=1}^{N} AP(q) \tag{28}$$

where N means the number of query samples and AP is the Average Precision:

$$AP(q) = \frac{1}{n_q}\sum_{r=1}^{R} P_q(r)\delta_q(r) \tag{29}$$

where n_q is the number of instances related to the query samples of the retrieval set. R is the size of the retrieval dataset. $P_q(r)$ means the precision of top–r retrieval samples. $\delta_q(r)$ is an indicator function and return 1 if the r–th item is related, otherwise return 0. The Precision–Recall (PR) curve is made with precision as the ordinate and recall as the abscissa. It reflects the varied precision values with different recall. The top–N Precision (Precision@N) curve measures the precision of the top N search results.

C. BASELINES AND IMPLEMENTATION DETAILS

We compare SCAGH with state–of–the–art methods and the descriptions are listed as follows.

- AGH [25] utilizes anchor graphs to discover the neighborhood structure of the original data to learn appropriate hash codes.
- GRH [56] employs graph regularisation to smooths the distribution of hash codes so that similar data points receive similar binary codes.
- SGH [26] uses feature transformation to approximate the whole similarity pairwise graph, which reduces the computation and storage cost.
- GSPF [57] maintains semantic similarity between data points in various settings, including single–label, multi-label, as well as paired and unpaired scenarios.
- FSH [54] embeds the graph–based fusion similarity from multiple modalities to a Hamming space.
- SRLCH [58] directly exploits relation information in class labels to make similar data from different modalities closer in the hamming subspace.
- OCMFH [59] uses collective matrix factorization in an online optimization scheme to learn hash codes for streaming data.
- RDMH [60] incorporates semantic labels into the hash codes and utilizes an auto encoder strategy to learn the hash function.
- WASH [61] obtains enhanced semantic information from the ground truth labels, and then perform matrix decomposition of the semantic information to obtain the shared representation.
- ATH [42] jointly optimizes the asymmetric hash functions and the bipartite graph on cross–domain data to alleviate the domain distribution gap.

GSPF [57], RDMH [60], WASH [61], SRLCH [58], GRH [56] and SCAGH are supervised methods, unsupervised methods include FSH [54], OCMFH [59] and three graph hashing methods AGH [25], SGH [26]and ATH [42]. All experiments are performed with MATLAB R2018a on a work station of 20–core 3.5 GHz CPU and 64GB RAM. In order to reduce the computational complexity, we randomly select 5,000 samples from NUSWIDE10 dataset as the training set for GSPF [57] and GRH [56]. 20,000 samples are selected from NUSWIDE10 for RDMH [60] and ATH [42]. SCAGH has different parameters including γ, λ, β, and θ. For different datasets, SCAGH has different parameter settings. The details are list as Table 1 and the specific parameters analysis process is shown in IV-D0.e.

D. EXPERIMENTAL RESULTS AND DISCUSSION

In the experiment, the best values are highlighted in blod and the runner–up values are underlined. To eliminate the impact of random errors, the MAP results on different datasets are generated by running 5 times and taking the average values. The retrieval tasks includes Image→ Text and Text → Image: (1)Image→ Text: retrieve relevant text using images; (2)Text → Image: retrieve relevant images using texts.

a: RESULTS ON WIKIDATA

MAP results of SCAGH and baselines methods on WIKIData are reported in Table 2 and the lengths of hash codes are varied from 8 to 128 bits. The Precision–Recall curves and top–Precision curves are shown in Figure 2 and Figure 3, respectively. The observations can be summarized as follows.

- SCAGH obtained the significantly best results compared with all baseline methods in the cases of different hash codes lengths, which verifies the superiority on small dataset.
- The proposed approach has an obvious improvement of 8% over the second best method GSPF on both retrieval tasks, which indicates the effectiveness of label mapping and associating anchors and anchor graph with semantic label.

TABLE 2. MAP values of different bits on WIKIData.

Methods	Image→ Text					Text→ Image				
	8bits	16bits	32bits	64bits	128bits	8bits	16bits	32bits	64bits	128bits
AGH [25]	0.1347	0.1232	0.1179	0.1150	0.1137	0.1159	0.1122	0.1140	0.1123	0.1122
GRH [56]	0.1177	0.1231	0.1231	0.1251	0.1262	0.1117	0.1109	0.1061	0.1151	0.1085
SGH [26]	0.1320	0.1776	0.1408	0.1666	0.1448	0.1112	0.1197	0.1130	0.1148	0.1187
GSPF [57]	0.2417	0.2806	0.2948	0.2992	0.3041	<u>0.5985</u>	<u>0.6476</u>	0.6612	0.6708	0.6725
FSH [54]	0.2194	0.2444	0.2515	0.2530	0.2527	0.2695	0.3690	0.3974	0.4343	0.4368
SRLCH [58]	<u>0.2688</u>	<u>0.2845</u>	<u>0.3126</u>	<u>0.3176</u>	<u>0.3412</u>	0.5894	0.6419	<u>0.7019</u>	<u>0.7028</u>	<u>0.7166</u>
OCMFH [59]	0.1835	0.1872	0.1878	0.1951	0.2046	0.5650	5842	0.6001	0.6162	0.4833
RDMH [60]	0.1607	0.1681	0.1660	0.1648	0.1808	0.3121	0.4126	0.4339	0.4569	0.5015
WASH [61]	0.1873	0.1648	0.1658	0.1750	0.1673	0.3623	0.2964	0.2788	0.2520	0.2441
ATH [42]	0.1475	0.1397	0.1461	0.1830	0.1337	0.1104	0.1121	0.1090	0.1095	0.1114
SCAGH	**0.3173**	**0.3593**	**0.3749**	**0.3836**	**0.3866**	**0.7021**	**0.7415**	**0.7485**	**0.7556**	**0.7603**

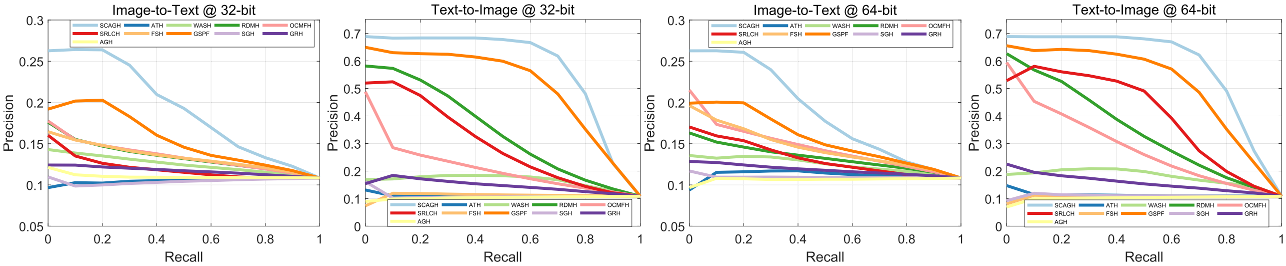


FIGURE 2. Precision–Recall curves on WIKIData by varying code length.

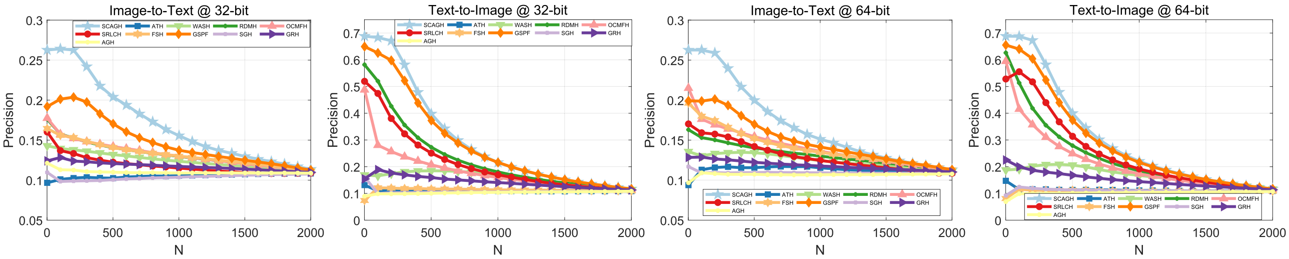


FIGURE 3. topN–Precision curves on WIKIData by varying code length.

- The prformance of the Image → Text task is significantly lower than the Text → Image task. One reason is that the text modality maintains more semantic information than the image when being mapped to the same hash codes. Another one is that the image modality includes more noise and outliers than text modality.
- The Precision–Recall and topN–precision curves of all methods using 32 and 64 bits are shown in Figure 2 and Figure 3, respectively. The proposed method has been further demonstrated to be effective through the results, which indicate that SCAGH outperforms other baselines.

b: RESULTS ON LABELME

The MAP values of SCAGH and nine baseline methods on MIRFLICKR are reported in Table 3. The Precision–Recall curves and topN–Precision of all methods are shown in Figure 4 and Figure 5, respectively.

- SCAGH represents the best performance than all baseline methods on text → image task but its performance on image → text is slightly lower than RDMH and WASH. The underlying reasons for this could be attributed to the following factors: (1) LableMe has the smallest number of training samples and class labels among the four datasets used. This limited dataset size may potentially impact the experimental performance. (2) For LableMe, the image modal has a higher feature dimension than text modal. The distribution characteristics of the dimension of image and text modalities may result in relatively lower performance in image retrieval with respect to text. (3) The differences between the hash codes of different images may be larger than the differences in text hash codes. This may result in larger distances between images in the Hamming space, thereby affecting the accuracy of image retrieval for text.
- The conclusion drawn from the Table 3 is that as the length of the hash code increases, the retrieval

TABLE 3. MAP values of different bits on LabelMe.

Methods	Image→ Text					Text→ Image				
	8bits	16bits	32bits	64bits	128bits	8bits	16bits	32bits	64bits	128bits
AGH [25]	0.1875	0.1631	0.1498	0.1497	0.1417	0.1440	0.1377	0.1365	0.1286	0.1303
GRH [56]	0.2699	0.2842	0.3685	0.3840	0.4131	0.1414	0.1423	0.1481	0.1369	0.1349
SGH [26]	0.2107	0.2030	0.1983	0.2492	0.2840	0.2943	0.2861	0.1923	0.1403	0.1209
GSPF [57]	0.3553	0.3539	0.3562	0.3550	0.3541	0.3633	0.3619	0.3602	0.3619	0.3626
FSH [54]	0.1663	0.1705	0.1857	0.1865	0.1725	0.2057	0.2051	0.2052	0.2066	0.1891
SRLCH [58]	0.6012	0.7908	0.8337	0.8674	0.8891	0.6166	0.8282	0.8800	0.9043	0.9181
OCMFH [59]	0.3025	0.2923	0.2917	0.2802	0.2790	0.3394	0.3079	0.3054	0.3013	0.2913
RDMH [60]	0.8303	0.8563	**0.8849**	0.8753	**0.8932**	0.9169	0.9046	0.9292	0.9256	0.9284
WASH [61]	0.8439	0.8687	0.8705	**0.8848**	0.8857	0.9154	0.9210	0.9236	0.9288	0.9308
ATH [42]	0.1445	0.1620	0.1288	0.1742	0.2088	0.1991	0.2426	0.1861	0.1925	0.2754
SCAGH	**0.8471**	**0.8766**	0.8787	0.8844	0.8895	**0.9219**	**0.9275**	**0.9300**	**0.9309**	**0.9360**

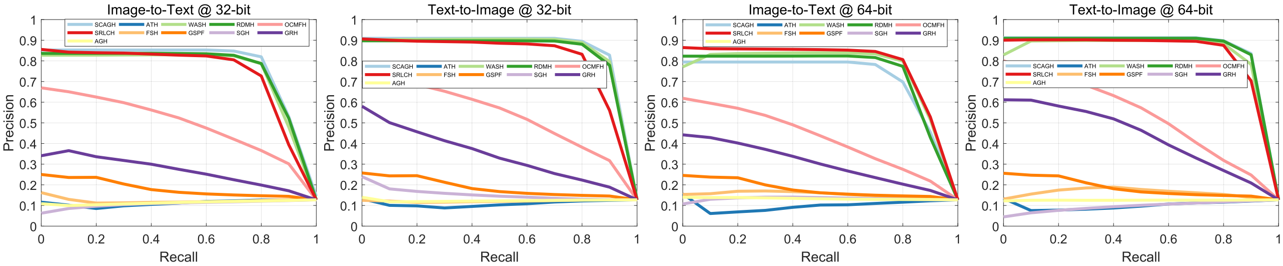


FIGURE 4. Precision–Recall curves on Labelme by varying code length.

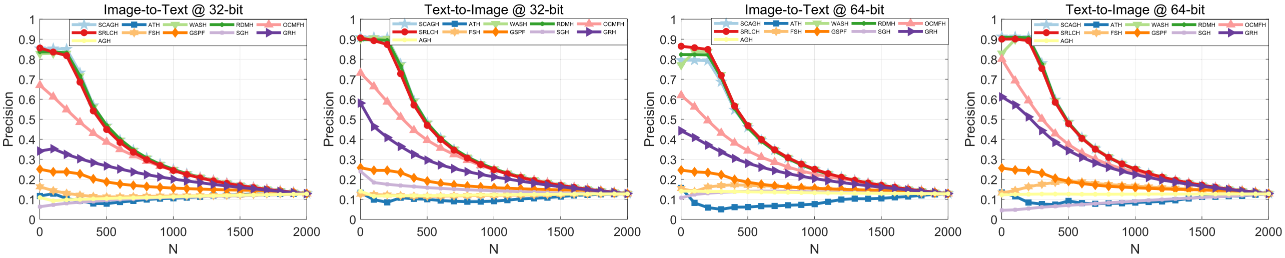


FIGURE 5. topN–Precision curves on Labelme by varying code length.

performance also gradually improves. One possible reason is that the longer hash codes contain more comprehensive and effective semantic information.

- Both WIKIData and Labelme are single label and small sample datasets, but the MAP value on Labelme dataset is much higher than that on WIKIData. This is because the feature dimensions of the samples in Labelme are higher and the semantic gap between image and text modality is smaller than WIKIData.
- As demonstrated in Figure 4 and Figure 5, the retrieval performance of methods SCAGH and WASH, RDMH is comparable, which aligns with the MAP results.

c: RESULTS ON MIRFLICKR

The MAP values of SCAGH and nine baseline methods on MIRFLICKR are reported in Table 4. The Precision–Recall curves and topN–Precision of all methods are shown in Figure 6 and Figure 7, respectively.

- According to Table 4, SCAGH shows slightly lower performance than WASH and RDMH for the Image → Text task at 8 bits and 16 bits. However, SCAGH surpasses all methods for the Text → Image task, indicating its effectiveness on MIRFLICKR dataset.
- SCAGH achieves the best performance compared to other graph hashing methods AGH, SGH, GRH and ATH. This is benefited by SCAGH adaptively select the anchor points and construct consensus anchor graph in a unified formula, which improves the quality of anchor graph and hash codes and enhances research performance.
- It should be noted that GSPF, RDMH, WASH, and SCAGH utilize category label to enhance performance, yet SCAGH surpasses all of them. The reason lies in the fact that the proposed method incorporates semantic label into the hash codes and associates anchor points and anchor graph through labels, hence extensively leveraging the supervised information of semantic label.

TABLE 4. MAP values of different bits on MIRFLICKR.

Methods	Image→ Text					Text→ Image				
	8bits	16bits	32bits	64bits	128bits	8bits	16bits	32bits	64bits	128bits
AGH [25]	0.5651	0.5728	0.5770	0.5660	0.5661	0.5860	0.5744	0.5741	0.5688	0.5671
GRH [56]	0.5658	0.5725	0.5694	0.5774	0.5745	0.5535	0.5632	0.5611	0.5666	0.5658
SGH [26]	0.5614	0.5632	0.5566	0.5575	0.5595	0.5662	0.5564	0.5571	0.5648	0.5499
GSPF [57]	0.6432	0.6557	0.6696	0.6703	0.6763	0.6774	0.6972	0.7143	0.7210	0.7331
FSH [54]	0.6112	0.6445	0.6580	0.6737	0.6855	0.6129	0.6435	0.6559	0.6710	0.6870
SRLCH [58]	0.5782	0.6172	0.5985	0.6437	0.6140	0.6008	0.6323	0.6288	0.6797	0.6457
OCMFH [59]	0.6235	0.5883	0.5837	0.5868	0.6039	0.6123	0.5997	0.5780	0.5761	0.6054
RDMH [60]	0.6160	0.6588	0.6581	0.6673	0.6980	0.6461	0.6958	0.6991	0.7109	0.7490
WASH [61]	**0.6982**	**0.7048**	0.7086	0.7077	0.7101	0.7418	0.7573	0.7634	0.7628	0.7671
ATH [42]	0.5589	0.5586	0.5679	0.5633	0.5625	0.5578	0.5618	0.5655	0.5656	0.5633
SCAGH	0.6922	7007	**0.7147**	**0.7245**	**0.7269**	**0.7579**	**0.7714**	**0.7963**	**0.8053**	**0.8129**

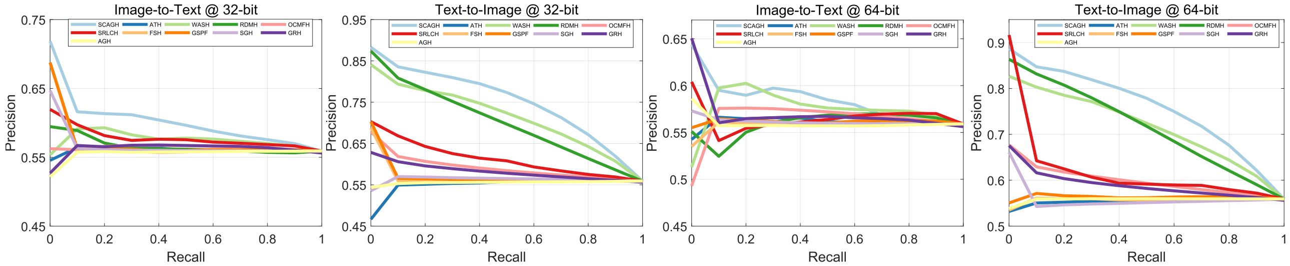


FIGURE 6. Precision–Recall curves on MIRFLICKR by varying code length.

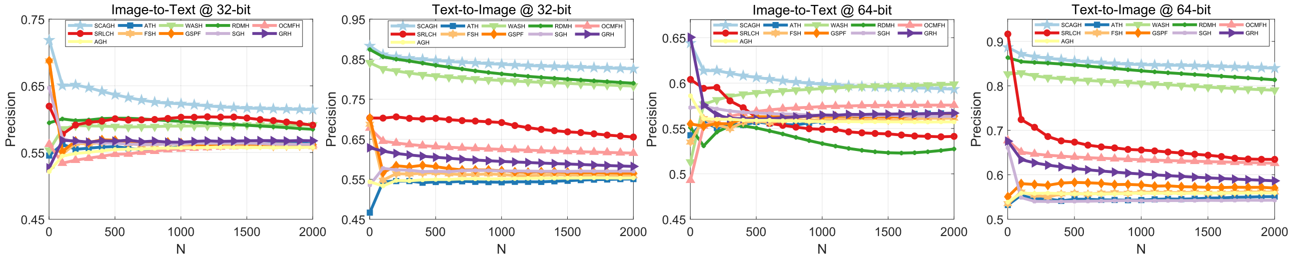


FIGURE 7. topN–Precision curves on MIRFLICKR by varying code length.

- As illustrated in Figure 6 and Figure 7, the Precision–Recall and topN–Precision curves of SCAGH are situated at the top among all baselines. Therefore, it suggests that the retrieval performance of SCAGH is higher than other methods.

d: RESULTS ON NUSWIDE10

The MAP values of SCAGH and nine baseline methods on NUSWIDE10 are reported in Table 5. The Precision–Recall curves and topN–Precision curves are shown in Figure 8 and Figure 9 respectively.

- The results indicate that SCAGH achieves the highest MAP values in both of the retrieval tasks. Notably, SCAGH improves the second–best results by approximately 5% and 6% on the image → text and text → image tasks, respectively. Moreover, as the hash code length increases, the performance gap between SCAGH and the second–best method widens significantly.
- In general, supervised methods like RDMH, GSPF, and WASH exhibit superior retrieval performance compared to unsupervised methods such as FSH or OCMFH. However, the graph hashing methods including AGH, SGH, and ATH, do not achieve comparable results to the unsupervised methods. This is because these methods were originally developed for single–mode retrieval and were later adapted by us for cross–mode retrieval.
- Figure 8 and Figure 9 display the Precision–Recall and topN–Precision curves for all methods. Based on the figures, it is apparent that SCAGH exhibits greater precision at fixed recall values and retrieved sample sizes.

e: PARAMETER SENSITIVITY ANALYSIS

This subsection analyzes the effect of parameters including γ, λ, β, and θ. SCAGH possesses different parameters including γ, λ, β, and θ. The parameters γ and λ are

TABLE 5. MAP values of different bits on NUSWIDE10.

Methods	Image→ Text					Text→ Image				
	8bits	16bits	32bits	64bits	128bits	8bits	16bits	32bits	64bits	128bits
AGH [25]	0.3602	0.3514	0.3456	0.3420	0.3402	0.3325	0.3367	0.3376	0.3393	0.3401
GRH [56]	0.3490	0.3563	0.3646	0.3669	0.3679	0.3687	0.3858	0.3966	0.4134	0.4214
SGH [26]	0.3282	0.3553	0.3597	0.3630	0.3412	0.3351	0.3368	0.3353	0.3357	0.3359
GSPF [57]	0.5692	0.5783	0.5998	0.6031	0.6043	0.7064	0.7372	0.7412	0.7535	0.7470
FSH [54]	0.5372	0.5603	0.5693	0.5771	0.5904	0.5347	0.5618	0.5681	0.5837	0.5895
SRLCH [58]	0.5567	0.6096	0.5916	0.6104	0.6324	0.6594	0.7144	0.7036	0.7222	0.7459
OCMFH [59]	0.3988	0.4135	0.4089	0.4189	0.3972	0.4039	0.4203	0.4311	0.4450	0.4538
RDMH [60]	0.5576	0.5498	0.6068	0.6334	0.6260	0.6472	0.6469	0.7166	0.7424	0.7504
WASH [61]	0.5869	0.5931	0.5977	0.5962	0.5967	0.6760	0.6896	0.6981	0.6972	0.7011
ATH [42]	0.3809	0.4901	0.5330	0.5789	0.5867	0.5986	0.6483	0.6805	0.6983	0.7017
SCAGH	**0.6140**	**0.6424**	**0.6650**	**0.6731**	**0.6810**	**0.7408**	**0.7767**	**0.8025**	**0.8157**	**0.8209**

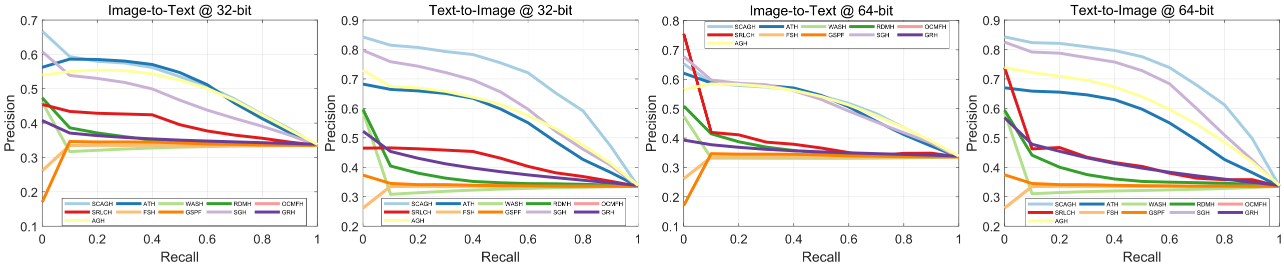


FIGURE 8. Precision–Recall curves on NUSWIDE10 by varying code length.

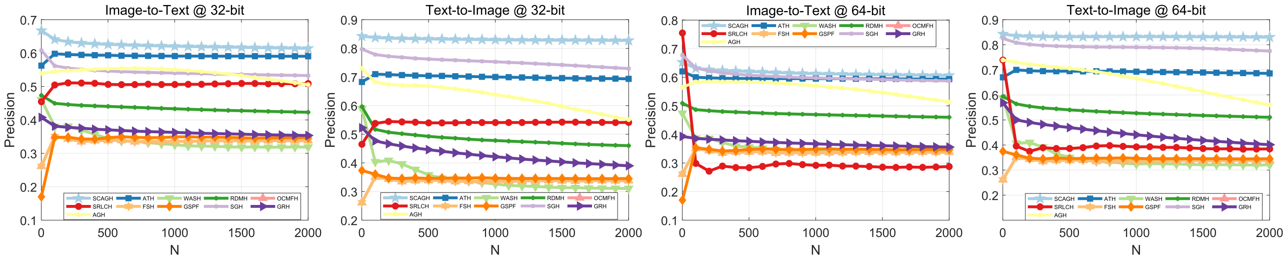


FIGURE 9. topN–Precision curves on NUSWIDE10 by varying code length.

utilized for mapping semantic labels, while θ is utilized to regulate hash function's regularization component in preventing overfitting. β must be greater than or equal to the highest eigen–value of the anchor similarity matrix. Assume the spectral radius of anchor similarity matrix $\mathbf{Z}$ is expressed as $\rho(\mathbf{Z})$, $\rho(\mathbf{Z}) \leq \| \mathbf{Z}\|_*$ and $\| \mathbf{Z}\|_*$ represents any matrix norm of $\mathbf{Z}$. Because $\| \mathbf{Z}\|_1 = 1$, $\| \mathbf{Z}\|_1$ represents the column norms of $\mathbf{Z}$, the value of β should be greater than or equal to 1. We explore the impact of individual parameters by varying γ, λ and θ in the range of $\left[10^{-3}, 10^{-2}, \ldots, 1e4\right]$, and β in the range of $[1e0, 1e1, \ldots, 1e4]$, while keeping the other parameters fixed. The code length is set at 128 bits and the experimental results are presented in Figure 10.

With significant changes observed, the impact of γ and θ on all datasets is noteworthy. When γ exceeds 1, there is a considerable increase in MAP, but it decreases sharply as θ drops below 10. On WIKIData, we set $\gamma = 1e4$ and $\theta = 1e-2$, while for NUSWIDE10, we adjust $\gamma = 1e4$ and $\theta = 1e-3$, and for MIRFLICK and Labelme datasets, we use different parameters $\gamma = 100$, $\theta = 1e-3$. With other parameters, SCAGH achieves stable MAP across all four datasets over a wide range. We set λ within the range of $[1e-2, 1e4]$ and β to either 10, $1e2$ or $1e3$. For detailed parameter settings, please refer to Table 1.

f: CONVERGENCE ANALYSIS

The efficiency of the SCAGH depends on the convergence rate which is affected by an iterative optimization algorithm. A convergence curve of SCAGH is presented in Figure 11, where the Y–axis shows the loss value of the objective function and the X–axis represents the iteration numbers. The results demonstrate that SCAGH achieves convergence within 30 iterations on all datasets with a fast convergence rate, indicating the effectiveness of the proposed method. It is observed that the convergence rate of MIRFLICK is slower compared to other datasets. This can be due to the smaller

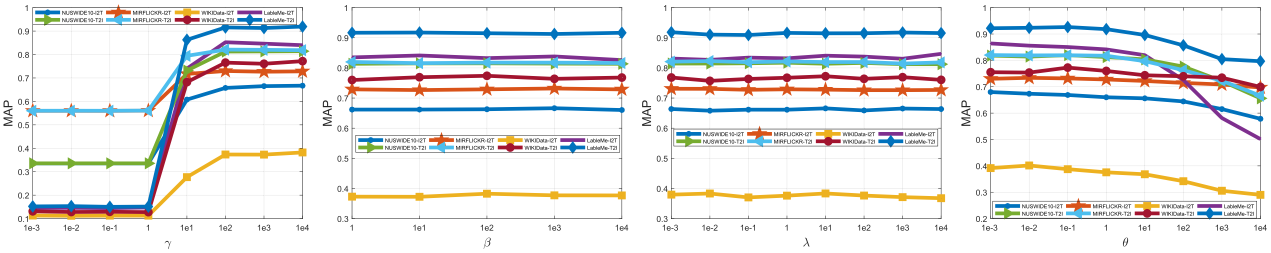


FIGURE 10. Paramters sensitivity analysize on four datasets.

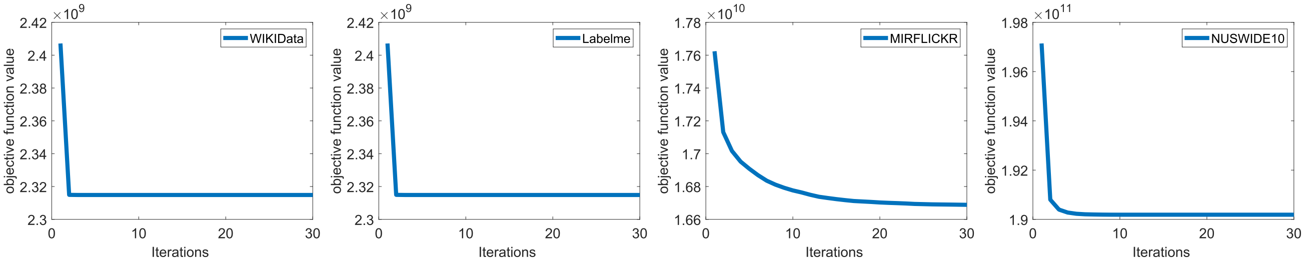


FIGURE 11. Convergence analysis on four datasets with 128 bits.

TABLE 6. Comparision of training time (in seconds) on Wikidata and MIRFLICK.

Methods	WIKIData				MIRFLICK			
	16bit	32bits	64bits	128bits	16bits	32bits	64bits	128bits
AGH [25]	0.2605	0.2697	0.2768	0.3002	5.6742	5.9456	5.9736	6.3270
GRH [56]	30.9119	67.7439	117.6361	267.4130	240.2930	470.2454	990.4199	1968.3156
SGH [26]	**0.0719**	**0.0617**	**0.0711**	**0.0769**	1.8673	<u>1.9918</u>	<u>2.1225</u>	<u>2.2256</u>
GSPF [57]	18.0642	32.4783	58.6285	117.5646	197.9540	381.0362	736.0753	1452.1726
FSH [54]	0.2981	0.2698	0.2685	<u>0.2742</u>	6.1809	6.2550	6.4405	6.3064
SRLCH [58]	<u>0.0759</u>	0.2376	0.2402	0.3184	**0.6386**	2.0760	2.9707	4.6177
OCMFH [59]	0.0995	<u>0.0897</u>	<u>0.1313</u>	0.3167	6.4674	6.7605	7.4859	9.2287
RDMH [60]	0.1517	0.2465	0.4099	0.6300	15.6614	14.0000	46.5035	73.4674
WASH [61]	0.5471	0.5642	0.5786	0.6079	<u>1.8804</u>	**1.9705**	**2.0463**	**2.1001**
ATH [42]	0.7669	0.7856	0.8232	0.9452	62.8412	63.1901	64.3411	66.5441
SCAGH	1.4463	1.2175	1.2317	1.2308	13.3908	13.3130	13.2940	13.5342

value $1e-2$ of the parameter λ used in dataset MIRFLICK compared to other datasets.

g: TRAIN AND QUERY TIME ANALYSIS

Train time and query time experiments of SCAGH and all baselines are conducted, with hash code lengths ranging from 16 to 128 bits. Due to Labelme has a similar size to WIKIData, GRH, GSPF and GSPF cannot be applied to the entire NUSWIDE10 dataset. Therefore, WIKIData and MIRFLICK datasets are chosen for evaluation. The results of all methods under the same settings is illustrated in Table 6 and Table 7. Specifically, GSPF and GRH take longer time for training and querying because they generates hash codes bit by bit. SRLCH and WASH take less time compared to other methods since neither of them requires the construction of affinity graph. Although SCAGH takes longer time than SGH since the anchor graph matrix is solved by using quadratic programming, the MAP value and retrieval performance of SGH are much worse than SCAGH. We find that the training time consumed on 16 bits is longer than that on other bits. This is because hash codes are stored in the form of array, i.e. [16, 32, 64, 128]. When 16 bits starts training, it is necessary to initialize and generate some parameter variables and load them into the MATLAB workspace. Although this may result in additional time overhead, the overall time difference is not substantial. The training time for WIKIData is mainly concentrated within the range of 1.2 to 1.4 seconds, while for MIRFLICK, it mainly centers around 13 seconds. Therefore, considering both the retrieval performance and training time, SCAGH is superior and scalable for large–scale datasets.

h: ANCHOR NUMBER ANALYSIS

During the process of adaptively selecting anchors and constructing anchor matrix, the quantity of anchors is uncertain. This section primarily investigates the effects of choosing different numbers of anchor points on model performance. We set the number of anchor points l within the range of $[1k, 2k, \ldots, 10k]$, where k denotes the number of categories and the dimension of anchor points. In Figure 12, we present the outcomes of both image query text and text query image tasks within four datasets WIKIData, Labelme, MIRFLICK and NUSWIDE with the optimal numbers of anchor points being 8k,5k,6k and 4k respectively.

TABLE 7. Comparision of querying time (in seconds) on Wikidata and MIRFLICK.

Methods	WIKIData				MIRFLICK			
	16bit	32bits	64bits	128bits	16bits	32bits	64bits	128bits
AGH [25]	0.1515	0.1775	0.2337	0.3443	6.2778	7.0948	8.4339	11.3936
GRH [56]	40.9830	85.2890	169.0169	350.0302	403.4096	791.8707	1662.1499	3293.9827
SGH [26]	0.1177	**0.1079**	**0.1064**	**0.1052**	6.1043	6.1738	6.1254	**6.0309**
GSPF [57]	16.1654	29.7816	54.1025	109.0547	103.4734	217.6428	437.4701	885.8517
FSH [54]	3.6104	4.1297	4.2945	4.4992	99.8186	96.7713	106.2700	84.1648
SRLCH [58]	0.1964	0.1902	0.2515	0.3435	**1.7824**	**2.5806**	**4.0791**	6.9214
OCMFH [59]	4.0327	2.7537	2.8513	3.0161	88.7413	91.7148	107.1927	100.7047
RDMH [60]	0.1440	0.1740	0.2066	0.3296	6.7176	7.4490	8.8593	11.9440
WASH [61]	0.3807	0.6329	1.1930	2.2279	12.3379	19.0555	32.0493	58.0653
ATH [42]	**0.1119**	0.1383	0.1985	0.3056	6.2152	6.8858	8.3221	11.0391
SCAGH	0.1648	0.1545	0.2109	0.3236	7.0253	7.7669	9.1616	12.0517

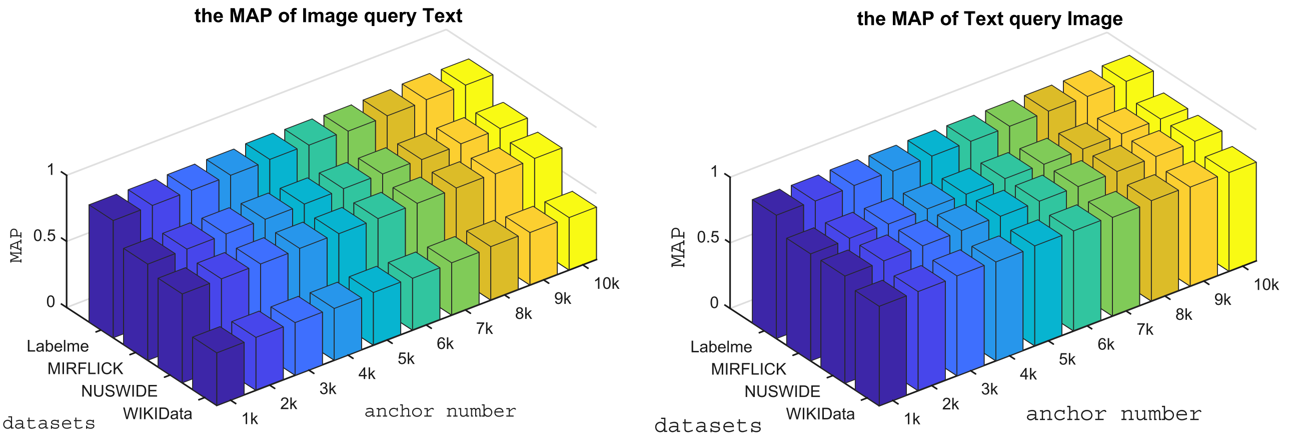
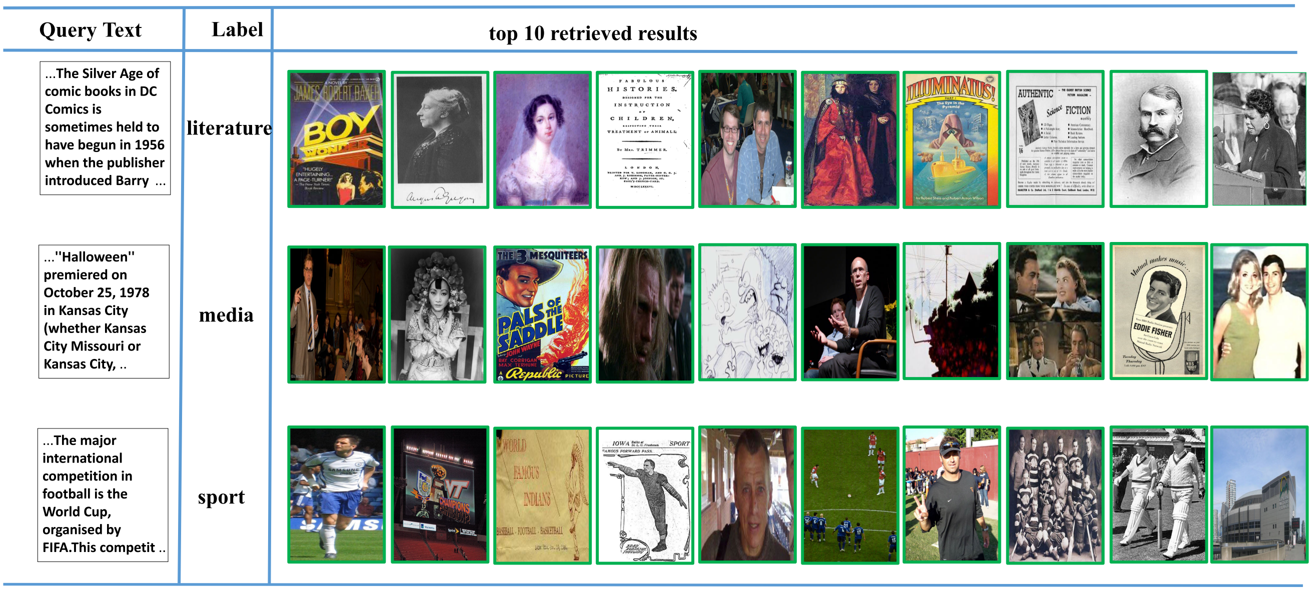


FIGURE 12. The MAP of different number of anchors.



...The Silver Age of comic books in DC Comics is sometimes held to have begun in 1956 when the publisher introduced Barry ...

..."Halloween" premiered on October 25, 1978 in Kansas City (whether Kansas City Missouri or Kansas City, ..

...The major international competition in football is the World Cup, organised by FIFA.This competit ..

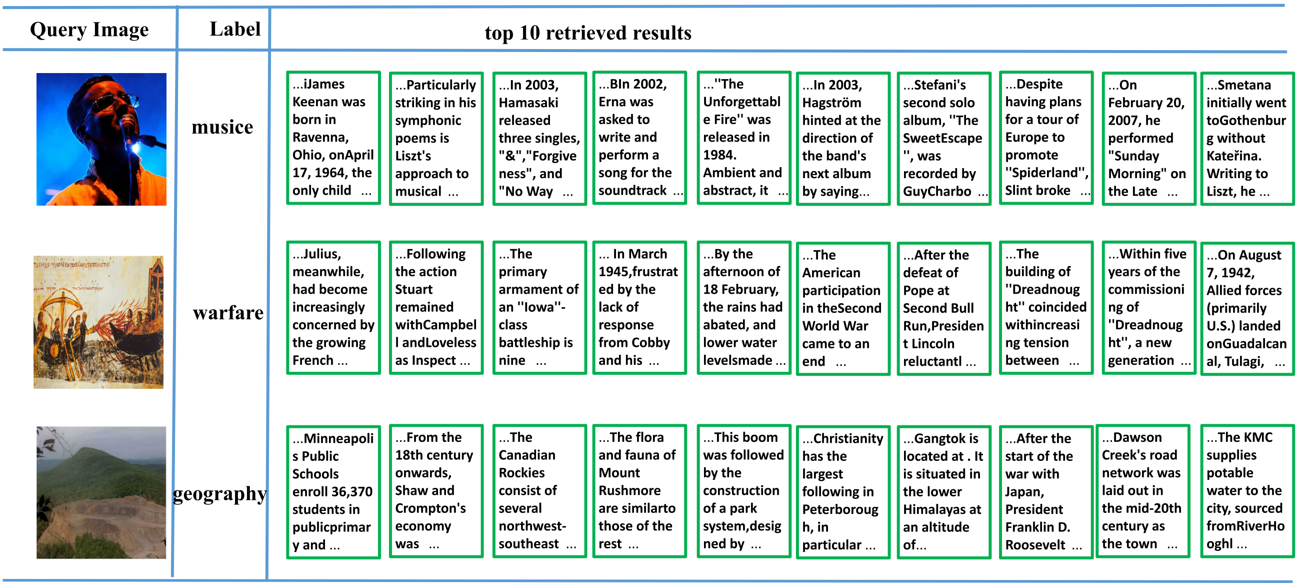
FIGURE 13. Examples of text query image on WIKIData.

i: ABLATION STUDY

To verify the contributions of different components to the overall performance, ablation experiments are conducted on the proposed method in the subsection. The variants of SCAGH include SCAGH_L, SCAGH_A and SCAGH_K. SCAGH_L removes supervised label information by setting the parameter λ and γ to zero. SCAGH_A randomly selects the same number of anchor points from training

TABLE 8. MAP results of SCAGH and its variants.

Dataset	Method	Image→ Text					Text→ Image				
		8bits	16bit	32bits	64bits	128bits	8bits	16bits	32bits	64bits	128bits
WIKIData	SCAGH_K	0.2762	0.3263	0.3434	0.3443	0.3573	0.6726	0.6991	0.7158	0.7317	0.7127
	SCAGH_L	0.1121	0.1124	0.1112	0.1118	0.1119	0.1155	0.1173	0.1188	0.1238	0.1229
	SCAGH_A	0.2310	0.1926	0.2106	0.2041	0.2255	0.1837	0.2077	0.1964	0.3587	0.2155
	SCAGH	**0.3173**	**0.3593**	**0.3749**	**0.3836**	**0.3866**	**0.7021**	**0.7415**	**0.7485**	**0.7556**	**0.7603**
Labelme	SCAGH_K	0.7452	0.7986	0.7945	0.8053	0.8084	0.8929	0.9100	0.9170	0.9137	0.9195
	SCAGH_L	0.1409	0.1411	0.1388	0.1410	0.1387	0.1414	0.1424	0.1402	0.1421	0.1421
	SCAGH_A	0.2116	0.2631	0.2709	0.2171	0.2241	0.2409	0.2317	0.2544	0.2296	0.2355
	SCAGH	**0.8471**	**0.8766**	**0.8787**	**0.8844**	**0.8895**	**0.9219**	**0.9275**	**0.9300**	**0.9309**	**0.9360**
MIRFLICK	SCAGH_K	0.6653	0.6821	0.7030	0.7108	0.7041	0.7445	0.7567	0.7670	0.7763	0.7684
	SCAGH_L	0.5573	0.5573	0.5582	0.5572	0.5575	0.5574	0.5574	0.5584	0.5574	0.5577
	SCAGH_A	0.5436	0.5599	0.5926	0.6012	0.5508	0.5147	0.5466	0.5843	0.6185	0.5346
	SCAGH	**0.6922**	**0.7007**	**0.7147**	**0.7245**	**0.7269**	**0.7579**	**0.7714**	**0.7963**	**0.8053**	**0.8129**
NUSWIDE	SCAGH_K	0.5925	0.6380	0.6515	0.6618	0.6699	0.7269	0.7756	0.7994	0.8027	0.8106
	SCAGH_L	0.3368	0.3392	0.3485	0.3492	0.3711	0.3372	0.3405	0.3539	0.3497	0.3838
	SCAGH_A	0.5060	0.3388	0.3716	0.3770	0.4661	0.4962	0.3418	0.3766	0.3844	0.5194
	SCAGH	**0.6140**	**0.6424**	**0.6650**	**0.6731**	**0.6810**	**0.7408**	**0.7767**	**0.8025**	**0.8157**	**0.8209**



Query Image	Label	top 10 retrieved results									
	musice	...iJames Keenan was born in Ravenna, Ohio, onApril 17, 1964, the only child ...	...Particularly striking in his symphonic poems is Liszt's approach to musical ...	...In 2003, Hamasaki released three singles, "&","Forgiveness", and "No Way ...	...BIn 2002, Erna was asked to write and perform a song for the soundtrack ...	..."The Unforgettable Fire" was released in 1984. Ambient and abstract, it ...	...In 2003, Hagström hinted at the direction of the band's next album by saying...	...Stefani's second solo album, "The SweetEscape", was recorded by GuyCharbo ...	...Despite having plans for a tour of Europe to promote "Spiderland", Slint broke ...	...On February 20, 2007, he performed "Sunday Morning" on the Late ...	...Smetana initially went toGothenburg without Kateřina. Writing to Liszt, he ...
	warfare	...Julius, meanwhile, had become increasingly concerned by the growing French ...	...Following the action Stuart remained withCampbell andLoveless as Inspect ...	...The primary armament of an "Iowa"-class battleship is nine ...	... In March 1945,frustrated by the lack of response from Cobby and his ...	...By the afternoon of 18 February, the rains had abated, and lower water levelsmade ...	...The American participation in theSecond World War came to an end ...	...After the defeat of Pope at Second Bull Run,President Lincoln reluctantl ...	...The building of "Dreadnought" coincided withincreasing tension between ...	...Within five years of the commissioning of "Dreadnought", a new generation ...	...On August 7, 1942, Allied forces (primarily U.S.) landed onGuadalcanal, Tulagi, ...
	geography	...Minneapolis Public Schools enroll 36,370 students in publicprimary and ...	...From the 18th century onwards, Shaw and Crompton's economy was ...	...The Canadian Rockies consist of several northwest-southeast ...	...The flora and fauna of Mount Rushmore are similarto those of the rest ...	...This boom was followed by the construction of a park system,designed by ...	...Christianity has the largest following in Peterborough, in particular ...	...Gangtok is located at . It is situated in the lower Himalayas at an altitude of...	...After the start of the war with Japan, President Franklin D. Roosevelt ...	...Dawson Creek's road network was laid out in the mid-20th century as the town ...	...The KMC supplies potable water to the city, sourced fromRiverHooghl ...

FIGURE 14. Examples of image query text on WIKIData.

set. SCAGH_K discards the kernel mapping for multimodal datas. The MAP results of SCAGH and its variants on four benchmark datasets with different hash code lengths are presented in Table 8. From the experimental results, the following observations can be made.

- According to Table 8, the MAP value of SCAGH_L has significantly decreased, which indicates that supervised semantic labels have a promoting effect on improving the retrieval performance of the proposed model.
- Compared to SCAGH_A which randomly select anchor points, the result that SCAGH outperforms SCAGH_A demonstrates the effectiveness of adaptively learns unified anchor points for different multimodal data.
- SCAGH_K shows the performance of the porposed model decreases without kernel mapping, implying the necessity of kernel mapping for capturing the nonlinear structure among original data features.

j: VISUALIZATION STUDY

To visually explore the effectiveness of the proposed method, experiments of image query text and text query image on WIKIData are performed. Figure 13 and Figure 14 present the visualization results of query samples with different category labels. The figures show the query images or texts in the first column, followed by the respective class labels in the second column and the top 10 retrieved results are sorted from left to right on the third column. The relevant retrieved samples are marked in green box and hash codes length is 128 bits. It can be observed that the retrieval results are semantic relevant to the query instances in real scenarios of cross–modal retrieval.

k: DEEP HASHING COMPARATIVE EXPERIMENT

We conducted experiments on the MIRFLICKR_deep dataset to compare with several state–of–the–art deep cross–hashing methods, including DFAH [48], DDCH [49], DCMH [44],

TABLE 9. MAP values of SCAGH and deep hahsing methods on MIRflick.

Task	Methods	16bit	32bit	64bit
Image query Text	SCAGH	**0.7756**	**0.7910**	**0.8066**
	DFAH [47]	0.7388	0.7402	0.7509
	DDCH [48]	0.7394	0.7450	0.7575
	SRCH [49]	0.6808	0.6916	0.6997
	DGCPN [46]	0.7320	0.7420	0.7510
	UDCMH [45]	0.6890	0.6980	0.7140
	PRDH [44]	0.7126	0.7128	0.7201
	DCMH [43]	0.7410	0.7465	0.7485
Text query Image	SCAGH	0.7571	0.7692	0.7827
	DFAH [47]	0.7614	0.7644	0.7764
	DDCH [48]	0.7596	0.7742	0.7847
	SRCH [49]	0.6971	0.7081	0.7146
	DGCPN [46]	0.7290	0.7410	0.7490
	UDCMH [45]	0.6920	0.7040	0.7180
	PRDH [44]	0.7467	0.7540	0.7505
	DCMH [43]	**0.7827**	**0.7900**	**0.7932**

SRCH [50], DGCPN [47], PRDH [45] and UDCMH [46]. As described in [44], the 4096–dimensional image features are extracted using the pre–trained CNN–F network [62], while the 1386–dimensional text features are represented by the bag–of–words vectors. The hash codes lengths are set to 16 bits, 32 bits, and 64 bits for simplicity. The MAP values obtained from the experiments are presented in Table 9 and consistent with the results reported in the original papers for all deep hashing methods.

From Table 9, we can see that while DCMH performs better than SCAGH on the Text query Image task with an increase of approximately 2%, its mAP value is not as good on the Image query Text task and is approximately 4 % lower than SCAGH. It is worth noting that deep hashing methods benefit from the end–to–end framework and extract deep–level semantic features through designed or fine–tuned neural networks. Although SCAGH is not a deep hashing model, it can outperform most of the state–of–the–art deep cross–modal hashing methods, which further confirms the effectiveness in the proposed method.

V. CONCLUSION

Prior anchor–based methods relied on heuristic sampling strategies such as k–means or random sampling for selecting anchor points. The anchor graph is constructed based on Euclidean distance between anchor points and samples. However, these two independent processes result in poor quality of hash codes and limited retrieval performance. In this paper, we develop a novel cross modal hashing method term as SCAGH. Different from previous anchor–based methods, SCAGH jointly integrates the anchor points selection, consensus anchor graph construction and hash codes learning into a unified framework to improve the retrieval performance. More specially, SCAGH automatically learns the anchor points and constructs the consensus anchor graph without additional hyper–parameters as previous methods do. Moreover, the $O(n^2)$ complexity can be avoided by approximating the pairwise similarity matrix with anchor graph matrix. Experimental results on four benchmark datasets validate that the superiority of the proposed SCAGH, demonstrating its effectiveness for cross–modal retrieval tasks. In future research, we will focus on integrating graph convolutional neural networks to construct a more comprehensive cross–modal deep hash model, which further enhance the performance of cross–modal hashing retrieval.

REFERENCES

[1] S. Chun, S. J. Oh, R. S. de Rezende, Y. Kalantidis, and D. Larlus, "Probabilistic embeddings for cross-modal retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8411–8420.

[2] M. Cheng, Y. Sun, L. Wang, X. Zhu, K. Yao, J. Chen, G. Song, J. Han, J. Liu, E. Ding, and J. Wang, "ViSTA: Vision and scene text aggregation for cross-modal retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5174–5183.

[3] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proc. 25th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 154–162.

[4] J. Gu, J. Cai, S. Joty, L. Niu, and G. Wang, "Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7181–7189.

[5] D. Gao, L. Jin, B. Chen, M. Qiu, P. Li, Y. Wei, Y. Hu, and H. Wang, "FashionBERT: Text and image matching with adaptive loss for cross-modal retrieval," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.* New York, NY, USA: Association for Computing Machinery, Jul. 2020, pp. 2251–2260.

[6] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, Apr. 2018.

[7] P. Kaur, H. S. Pannu, and A. K. Malhi, "Comparative analysis on cross-modal information retrieval: A review," *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100336.

[8] K. Wang, Q. Yin, W. Wang, S. Wu, and L. Wang, "A comprehensive survey on cross-modal retrieval," 2016, *arXiv:1607.06215*.

[9] Y. Li, S. Wang, Q. Pan, H. Peng, T. Yang, and E. Cambria, "Learning binary codes with neural collaborative filtering for efficient recommendation systems," *Knowl.-Based Syst.*, vol. 172, pp. 64–75, May 2019.

[10] C. Peng, L. Zhu, Y. Xu, Y. Li, and L. Guo, "Binary multi-modal matrix factorization for fast item cold-start recommendation," *Neurocomputing*, vol. 507, pp. 145–156, 2022.

[11] J. Yi, Y. Zhu, J. Xie, and Z. Chen, "Cross-modal variational auto-encoder for content-based micro-video background music recommendation," *IEEE Trans. Multimedia*, vol. 25, pp. 515–528, 2023.

[12] L. Xie and X. Fang, "Online discrete anchor graph hashing for mobile person re-identification," *J. Adv. Transp.*, vol. 2021, pp. 1–8, Dec. 2021.

[13] X. Xu, S. Liu, N. Zhang, G. Xiao, and S. Wu, "Channel exchange and adversarial learning guided cross-modal person re-identification," *Knowl.-Based Syst.*, vol. 257, Dec. 2022, Art. no. 109883.

[14] Z. Ye and Y. Peng, "Sequential cross-modal hashing learning via multi-scale correlation mining," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 4, pp. 1–20, Nov. 2019.

[15] R.-C. Tu, X.-L. Mao, Q. Lin, W. Ji, W. Qin, W. Wei, and H. Huang, "Unsupervised cross-modal hashing via semantic text mining," *IEEE Trans. Multimedia*, vol. 25, pp. 8946–8957, 2023.

[16] X. Liu, A. Li, J.-X. Du, S.-J. Peng, and W. Fan, "Efficient cross-modal retrieval via flexible supervised collective matrix factorization hashing," *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 28665–28683, Nov. 2018.

[17] H. T. Shen, L. Liu, Y. Yang, X. Xu, Z. Huang, F. Shen, and R. Hong, "Exploiting subspace relation in semantic labels for cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3351–3365, Oct. 2021.

[18] Y. Wang, Z.-D. Chen, X. Luo, R. Li, and X.-S. Xu, "Fast cross-modal hashing with global and local similarity embedding," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10064–10077, Oct. 2022.

[19] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2083–2090.

[20] M. Cheng, L. Jing, and M. K. Ng, "Robust unsupervised cross-modal hashing for multimedia retrieval," *ACM Trans. Inf. Syst.*, vol. 38, no. 3, pp. 1–25, Jul. 2020.

[21] Y. Fang, H. Zhang, and Y. Ren, "Unsupervised cross-modal retrieval via multi-modal graph regularized smooth matrix factorization hashing," *Knowl.-Based Syst.*, vol. 171, pp. 69–80, May 2019.

[22] P.-F. Zhang, C.-X. Li, M.-Y. Liu, L. Nie, and X.-S. Xu, "Semi-relaxation supervised hashing for cross-modal retrieval," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 1762–1770.

[23] X. Luo, X.-Y. Yin, L. Nie, X. Song, Y. Wang, and X.-S. Xu, "SDMCH: Supervised discrete manifold-embedded cross-modal hashing," in *Proc. 27Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2518–2524.

[24] Y. Chen, H. Zhang, Z. Tian, J. Wang, D. Zhang, and X. Li, "Enhanced discrete multi-modal hashing: More constraints yet less time to learn," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1177–1190, Mar. 2022.

[25] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[26] Q.-Y. Jiang and W.-J. Li, "Scalable graph hashing with feature transformation," in *Proc. IJCAI*, vol. 15, 2015, pp. 2248–2254.

[27] S. Kim and S. Choi, "Multi-view anchor graph hashing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3123–3127.

[28] W. Liu, C. Mu, S. Kumar, and S. F. Chang, "Discrete graph hashing," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3419–3427.

[29] L. Xiang, X. Shen, J. Qin, and W. Hao, "Discrete multi-graph hashing for large-scale visual search," *Neural Process. Lett.*, vol. 49, no. 3, pp. 1055–1069, Jun. 2019.

[30] H. Hu, K. Wang, C. Lv, J. Wu, and Z. Yang, "Semi-supervised metric learning-based anchor graph hashing for large-scale image retrieval," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 739–754, Feb. 2019.

[31] J. Guo and W. Zhu, "Collective affinity learning for partial cross-modal hashing," *IEEE Trans. Image Process.*, vol. 29, pp. 1344–1355, 2020.

[32] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, vol. 282, no. 3, pp. 1753–1760.

[33] L. Jin, K. Li, H. Hu, G.-J. Qi, and J. Tang, "Semantic neighbor graph hashing for multimodal retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1405–1417, Mar. 2018.

[34] D. Tang, H. Cui, D. Shi, and H. Ji, "Hypergraph-based discrete hashing learning for cross-modal retrieval," in *Proc. Adv. Multimedia Inf. Process. (PCM), 19th Pacific-Rim Conf. Multimedia*, Hefei, China. Cham, Switzerland: Springer, Sep. 2018, pp. 776–786.

[35] C. Hou, Z. Li, Z. Tang, X. Xie, and H. Ma, "Multiple instance relation graph reasoning for cross-modal hash retrieval," *Knowl.-Based Syst.*, vol. 256, Nov. 2022, Art. no. 109891.

[36] X. Shen, H. Zhang, L. Li, W. Yang, and L. Liu, "Semi-supervised cross-modal hashing with multi-view graph representation," *Inf. Sci.*, vol. 604, pp. 45–60, Aug. 2022.

[37] W. Wang, H. Zhang, Z. Zhang, L. Liu, and L. Shao, "Sparse graph based self-supervised hashing for scalable image retrieval," *Inf. Sci.*, vol. 547, pp. 622–640, Feb. 2021.

[38] Y. Fujiwara, S. Kanai, Y. Ida, A. Kumagai, and N. Ueda, "Fast algorithm for anchor graph hashing," *Proc. VLDB Endowment*, vol. 14, no. 6, pp. 916–928, Feb. 2021.

[39] X. Shi, F. Xing, K. Xu, M. Sapkota, and L. Yang, "Asymmetric discrete graph hashing," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2541–2547.

[40] E. Yu, J. Sun, L. Wang, X. Chang, H. Zhang, and A. G. Hauptmann, "Cross-modal transfer hashing based on coherent projection," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2019, pp. 477–482.

[41] L. Wang, J. Yang, M. Zareapoor, and Z. Zheng, "Anchor graph structure fusion hashing for cross-modal similarity search," 2022, *arXiv:2202.04327*.

[42] J. Lu, J. Zhou, Y. Chen, W. Pedrycz, and K.-W. Hung, "Asymmetric transfer hashing with adaptive bipartite graph learning," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 533–545, 2024.

[43] S. Liu, S. Wang, P. Zhang, K. Xu, X. Liu, C. Zhang, and F. Gao, "Efficient one-pass multi-view subspace clustering with consensus anchors," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 7, pp. 7576–7584.

[44] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3270–3278.

[45] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1618–1625.

[46] G. Wu, Z. Lin, J. Han, L. Liu, G. Ding, B. Zhang, and J. Shen, "Unsupervised deep hashing via binary latent factor models for large-scale cross-modal retrieval," in *Proc. IJCAI*, 2018, vol. 1, no. 3, pp. 2854–2860.

[47] J. Yu, H. Zhou, Y. Zhan, and D. Tao, "Deep graph-neighbor coherence preserving network for unsupervised cross-modal hashing," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 5, pp. 4626–4634.

[48] J. Li, E. Yu, J. Ma, X. Chang, H. Zhang, and J. Sun, "Discrete fusion adversarial hashing for cross-modal retrieval," *Knowl.-Based Syst.*, vol. 253, Oct. 2022, Art. no. 109503.

[49] E. Yu, J. Ma, J. Sun, X. Chang, H. Zhang, and A. G. Hauptmann, "Deep discrete cross-modal hashing with multiple supervision," *Neurocomputing*, vol. 486, pp. 215–224, May 2022.

[50] W. Wang, Y. Shen, H. Zhang, Y. Yao, and L. Liu, "Set and rebase: Determining the semantic graph connectivity for unsupervised cross-modal hashing," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 853–859.

[51] S. Wang, X. Liu, X. Zhu, P. Zhang, Y. Zhang, F. Gao, and E. Zhu, "Fast parameter-free multi-view subspace clustering with consensus anchor guidance," *IEEE Trans. Image Process.*, vol. 31, pp. 556–568, 2022.

[52] M.-S. Chen, C.-D. Wang, D. Huang, J.-H. Lai, and P. S. Yu, "Efficient orthogonal multi-view subspace clustering," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 127–135.

[53] Y. Fang, K. A. Loparo, and X. Feng, "Inequalities for the trace of matrix product," *IEEE Trans. Autom. Control*, vol. 39, no. 12, pp. 2489–2490, Dec. 1994.

[54] H. Liu, R. Ji, Y. Wu, F. Huang, and B. Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6345–6353.

[55] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[56] S. Moran and V. Lavrenko, "Regularised cross-modal hashing," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 907–910.

[57] D. Mandal, K. N. Chaudhury, and S. Biswas, "Generalized semantic preserving hashing for n-label cross-modal retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2633–2641.

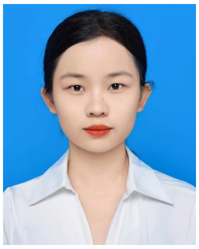
[58] L. Liu, Y. Yang, M. Hu, X. Xu, F. Shen, N. Xie, and Z. Huang, "Index and retrieve multimedia data: Cross-modal hashing by learning subspace relation," in *Proc. 23rd Int. Conf. Database Syst. Adv. Appl. (DASFAA)*, Gold Coast, QLD, Australia. Cham, Switzerland: Springer, May 2018, pp. 606–621.

[59] D. Wang, Q. Wang, Y. An, X. Gao, and Y. Tian, "Online collective matrix factorization hashing for large-scale cross-media retrieval," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1409–1418.

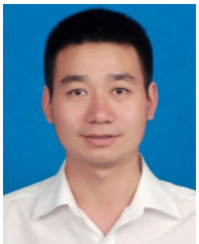
[60] D. Zhang and X.-J. Wu, "Robust and discrete matrix factorization hashing for cross-modal retrieval," *Pattern Recognit.*, vol. 122, pp. 1–12, Feb. 2022.

[61] C. Zhang, H. Li, Y. Gao, and C. Chen, "Weakly-supervised enhanced semantic-aware hashing for cross-modal retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6475–6488, Jun. 2023.

[62] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," 2014, *arXiv:1405.3531*.



RUI CHEN is currently pursuing the master's degree with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology. Her current research interest includes cross modal hashing retrieval.



HONGBIN WANG received the Ph.D. degree in computer system structure from Jilin University. He is currently a Professor with the Kunming University of Science and Technology and the Deputy Director of the Yunnan Key Laboratory of Artificial Intelligence. His research interests include natural language processing, data mining, and cross modal retrieval.