Joint Semantic Preserving Sparse Hashing for Cross-Modal Retrieval

Zhikai Hu, Yiu-Ming Cheung, *Fellow, IEEE*, Mengke Li, Weichao Lan, *Graduate Student Member, IEEE*, Donglin Zhang, and Qiang Liu, *Senior Member, IEEE*

***Abstract*— Supervised cross-modal hashing has received wide attention in recent years. However, existing methods primarily rely on sample-wise semantic relationships to evaluate the semantic similarity between samples, overlooking the impact of label distribution on enhancing retrieval performance. Moreover, the limited representation capability of traditional dense hash codes hinders the preservation of semantic relationship. To overcome these challenges, we propose a new method, Joint Semantic Preserving Sparse Hashing (JSPSH). Specifically, we introduce a new concept of cluster-wise semantic relationship, which leverages label distribution to indicate which samples are more suitable for clustering. Then, we jointly utilize sample-wise and cluster-wise semantic relationships to supervise the learning of hash codes. In this way, JSPSH preserves both kinds of semantic relationships to ensure that more samples with similar semantics are clustered together, thereby achieving better retrieval results. Furthermore, we utilize high-dimensional sparse hash codes that offer stronger representation capability to preserve such more complex semantics. Finally, an interaction term is introduced in hash functions learning stage to further narrow the gap between modalities. Experimental results on three large-scale datasets demonstrate the effectiveness of JSPSH in achieving superior retrieval performance.**

***Index Terms*— Cross-modal retrieval, hashing, sample-wise semantics, cluster-wise semantics, clustering, discrete optimization.**

I. Introduction

IN THE past decade, the growing availability of multimedia data on the Internet has made cross-modal retrieval become a research hotspot. Cross-modal retrieval [1], [2], [3], [4], [5], [6], [7] refers to the task of retrieving data across different modalities, such as using a piece of text to retrieve the corresponding image, video, or audio, etc. To cope with the large amount of multimedia data and improve retrieval efficiency, hashing technology [8], [9] has been widely used in the field of cross-modal retrieval, resulting in the development of cross-modal hashing methods [10], [11], [12], [13], [14], [15]. These methods map data of different modalities into a shared Hamming subspace, enabling fast retrieval of multi-modal data through the simple XOR operation.

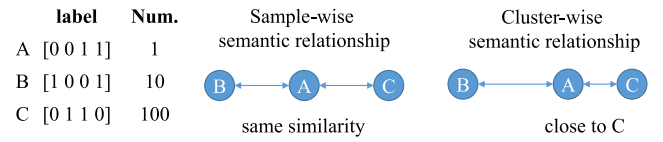


Fig. 1. When disregarding the distribution of labels, the sample-wise semantic similarity between A and B and that between A and C are identical. However, given that there are more samples affiliated with label C, it is desirable for A to be more akin to C than B, to produce more correct retrieval outcomes. This relationship is referred to as cluster-wise semantic relationship in this paper.

In general, cross-modal hashing methods can be broadly classified into unsupervised [11], [13], [14], [16], [17] and supervised methods [18], [19], [20], [21]. Supervised cross-modal hashing methods, which make use of label information, can more effectively mine the semantic relationships between multi-modal data and often achieve better retrieval results. Nevertheless, since the widely used logical labels are relatively rough supervision information, how to use them more efficiently to mine the relationships between multi-modal data and supervise the learning of corresponding hash codes is still an open problem. To the best of our knowledge, existing methods [21], [22], [23], [24], [25] typically estimate the similarity between samples based on the cosine distance or inner product of their corresponding labels, capturing the **sample-wise semantic relationship**. However, these approaches ignore the fact that the distribution of labels can be highly diverse across different datasets, and such information is crucial to further improving retrieval quality. For example, let us consider a scenario where there are three labels A [0,0,1,1], B [1,0,0,1], and C [0,1,1,0], and their corresponding sample sizes are 1, 10, and 100, respectively, as shown in Fig. 1. The similarity between A and B and that between A and C, calculated by the cosine distance of their labels, are both 1/2. However, since there are more samples corresponding to label C, we may expect that more correct samples can be retrieved during the retrieval phase if A is closer to C. Therefore, in addition to

Manuscript received 19 March 2023; revised 24 July 2023; accepted 20 August 2023. Date of publication 22 August 2023; date of current version 5 April 2024. This work was supported in part by the NSFC/Research Grants Council (RGC) Joint Research Scheme under Grant N_HKBU214/21, in part by the General Research Fund of RGC under Grant 12201321 and Grant 12202622, in part by the National Natural Science Foundation of China under Grant 61991401, Grant U20A20189, and Grant 62161160338, in part by NSFC under Grant 62202204, and in part by the Fundamental Research Funds for the Central Universities under Grant JUSRP123032. This article was recommended by Associate Editor H. Zhang. *(Corresponding author: Yiu-Ming Cheung.)*

Zhikai Hu, Yiu-Ming Cheung, and Weichao Lan are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China (e-mail: cszkhu@comp.hkbu.edu.hk; ymc@comp.hkbu.edu.hk; cswclan@comp.hkbu.edu.hk).

Mengke Li is with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, Guangdong 518000, China (e-mail: limengke@gml.ac.cn).

Donglin Zhang is with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China (e-mail: dlinzzhang@gmail.com).

Qiang Liu is with the State Key Laboratory of Synthetical Automation for Process Industries (Northeastern University), Shenyang, Liaoning 110819, China (e-mail: liuq@mail.neu.edu.cn).

Data is available on-line at https://github.com/hutt94/JSPSH.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3307608.

Digital Object Identifier 10.1109/TCSVT.2023.3307608

© 2023 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

the sample-wise semantic relationship, we can also consider a **cluster-wise semantic relationship**. In this context, the cluster-wise similarity between A and C is a measure of how well they belong to the same cluster of samples, compared to A and B. Obviously, considering the cluster-wise semantic relationships of labels in supervised cross-modal hashing can potentially lead to more accurate retrieval results.

Furthermore, the representation capability of traditional dense hash codes commonly used in cross-modal hashing is limited. Traditional hash encoding scheme map multi-modal data into dense -1 and 1 codes, requiring long hash codes to achieve better retrieval performance [26], [27], [28]. This results in additional storage space burden and lower retrieval efficiency. Meanwhile, there is also a similarity mismatch between dense hash codes and labels. Specifically, as labels consist of binary values 0 and 1, their similarity range is $S^L \in [0, 1]$, where $S^L = 0$ represents semantic irrelevance (negative relationship), and $S^L > 0$ represents semantic relevance (positive relationship). However, the similarity range of traditional dense hash codes is $S^B \in [-1, 1]$, where $S^B \leq 0$ represents negative relationships, and $S^B > 0$ represents positive relationships. To bridge the mismatch in value range, some methods [24], [29] use $2S^L - 1$ to estimate S^B. However, in this case, positive relations in S^L $(0 < S^L < 0.5)$ will be incorrectly estimated as negative relations. In addition, most of the current two-stage cross-modal hashing methods [12], [24], [25], [28], [29] learn the hash function separately for each modality, which leads to a lack of interaction between modalities, ultimately hindering the capability to bridge the heterogeneous gap.

In this paper, we propose a framework based on sparse hashing to address the aforementioned problems, which is referred to as Joint Semantic Preserving Sparse Hashing (JSPSH). Specifically, we propose a joint learning scheme that incorporates both of the commonly used sample-wise semantic relationship and a newly introduced cluster-wise semantic relationship obtained through label clustering. We utilize these relationships simultaneously to supervise the learning of hash codes. Furthermore, we leverage the representation capability of high-dimensional sparse hash codes, which have been shown to be effective in encoding multi-modal data [25], [30]. With sparse hash codes, there is no issue of mismatching similarity value domains, as the values of sparse hash codes are 0 or 1, which is the same as labels. Finally, to further narrow the heterogeneous gap between modalities during the hash function learning stage, we introduce a new interaction term to increase the interaction between them. The main contributions of this paper are summarized as follows:

- We propose a novel approach called Joint Semantic Preserving Sparse Hashing, which leverages both sample-wise and cluster-wise semantic similarity to guide the learning of hash codes. By introducing cluster-wise semantic relationships, JSPSH ensures that samples with similar semantics can be clustered together more appropriately to achieve better retrieval performance.
- To enable effective learning of these joint semantic correlations, we adopt more expressive high-dimensional sparse hash codes for encoding multi-modal data. Compared with traditional dense hash codes, it can better preserve complex semantic relationships.
- We introduce a new interaction term in the hash function learning stage, which ensures better alignment between modalities. This further improves the retrieval performance of JSPSH by strengthening the relationship between the different modalities.
- The proposed method was evaluated on three commonly used public datasets, and the experimental results demonstrate that our method outperforms existing methods, both dense and sparse hashing ones.

The remainder of this paper is organized as follows. Section II makes an overview of some related works. Section III presents the details of the proposed JSPSH. Then, Section IV provides the experiment results and analyses. Finally, a conclusion is drawn in Section V.

II. Related Work

In this section, we briefly classify existing cross-modal hashing methods based on their encoding method into two categories: traditional dense hashing and high-dimension sparse hashing methods.

A. Dense Cross-Modal Hashing

By default, cross-modal hashing usually refers to dense cross-modal hashing, which encodes multi-modal data into dense hash codes where each bit in the k-bit hash code must be 1 or -1. Depending on whether supervised information is utilized or not, these methods can be further classified into unsupervised and supervised methods. Unsupervised cross-modal hashing methods learn hash codes for multi-modal data without the use of any explicit supervision. They typically exploit the pairwise information between different modalities or the underlying manifold structure of data within each modality to learn the hash codes. A variety of unsupervised cross-modal hashing methods have been proposed in the literature. For example, Inter-Media Hashing (IMH) [10] learns linear hash functions to map multi-modal data into a common Hamming space by exploring the inter-modal and intra-modal correlation of different modalities. Collective Matrix Factorization Hashing (CMFH) [11] utilizes the pairwise information between different modalities and introduces collective matrix factorization to learn unified hash codes. Same as CMFH, Latent Semantic Sparse Hashing (LSSH) [31] learns unified hash codes for all modalities by utilizing the sparse coding and matrix factorization techniques. Besides, Composite Correlation Quantization (CCQ) [32] jointly map both multi-modal data into an isomorphic latent space and learn corresponding hash codes by composite quantization. Fusion Similarity Hashing (FSH) [33] employs a fusion strategy to learn hash codes by constructing an un-directed graph among different modalities. Collective Reconstructive Embedding (CRE) [34] also learn unified binary codes by reconstructing embedding of multi-modal data collectively. More recently, Robust Unsupervised Cross-Modal Hashing (RUCMH) [35] further improves the robustness of cross-modal hashing by exploring the relation

between modalities with only partial or even no pairwise information.

Supervised cross-modal hashing methods utilize the additional information provided by labels or annotations to learn hash codes. For example, Semantics Preserving Hashing (SePH) [18] uses labels to learn a similarity distribution, with the objective of maximizing the similarity between the learned hash codes and the given distribution. Generalized Semantics Preserving Hashing (GSPH) [23] propose a cross-modal hashing algorithm that can seamlessly handle multi-label and single-label, paired data, and unpaired data scenarios, making it applicable to a wide range of real-world scenarios. Besides, Discriminative Cross-modal Hashing (DCH) [36] uses labels to learn a classifier, with the aim of generating more discriminative hash codes. To further reduce quantization error, DCH employs the Discrete Cyclic Coordinate (DCC) [37] descent method to discretely update the learned hash code. Label Consistent Matrix Factorization Hashing (LCMFH) [38] and Scalable disCRete mATrix faCtorization Hashing (SCRATCH) [12] simultaneously leverage heterogeneous multi-modal data and labels to learn consistent hash codes that preserve semantic similarity as much as possible. Matrix Tri-Factorization Hashing (MTFH) [22] is the first cross-modal hashing method that attempts to represent different modal data with hash codes of different lengths, which can help capture more information from each modality. Fast Cross-Modal Hashing (FCMH) [24], on the other hand, emphasizes both global and local similarity preservation in the process of learning hash codes, and proposes a discrete update framework to optimize the objective function. To make better use of label information, Adaptive Label correlation based asymmEtric Cross-modal Hashing (ALECH) [29] uses more adaptive labels to supervise the learning of hash codes.

Thanks to the impressive performance of deep learning on various tasks [39], [40], [41], [42], deep cross-modal hashing methods [27], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53] have recently gained significant attention and have shown promising results. These methods utilize the latest deep learning techniques, such as knowledge distillation and contrastive learning, to learn feature representations from multiple modalities and use these representations to generate compact hash codes. However, they are typically more computationally expensive and hard to be optimized under the discrete constraint.

B. High-Dimension Sparse Hashing

High-dimensional sparse hashing is a technique in which data is mapped into a higher-dimensional Hamming space, with only a small subset of bits containing information. This approach contrasts with dense hashing, where all bits in the hash code must be either 1 or -1. In high-dimensional sparse hashing, the number of bits carrying information is significantly smaller than the total number of bits, resulting in a sparse representation that is more efficient in terms of storage and computation. The first high-dimensional sparse hashing work, Fly-Hash [54], was inspired by the biological fruit fly olfactory circuit and modified Locally Sensitive Hashing (LSH) [55], originally dense hashing, into a high-dimensional sparse version. The key characteristic of this approach is that it uses a hash function to project the data into a high-dimension Hamming space, where only a small number of bits contain information. Specifically, a winner-take-all strategy is employed, that is, the largest r elements of the output of hash function are set to 1 and the rest are set to 0. In Fly-Hash, the hash mapping function is randomly generated, so it cannot make use of the inherent information of data. In order to address this issue, some data-driven methods have been proposed, such as Bio-Inspired Hashing (Bio-Hash) [56] and Optimal Sparse Lifting Hashing (OSLHash) [57]. Although the performance has been significantly improved, these methods are still limited to single modality retrieval tasks.

More recently, high-dimensional sparse hashing has been firstly introduced in cross-modal hashing by High-dimensional Sparse Cross-modal Hashing (HSCH) [25]. HSCH maps multi-modal data into a high-dimensional sparse Hamming space, where only a small number of bits contain information. Compared with dense hashing, high-dimensional sparse hashing has been shown to have more efficient expression ability and better retrieval performance. Later, an online version of HSCH has also been proposed [30]. However, to date, there are still only a small number of cross-modal hashing methods based on high-dimensional sparse hashing.

III. Proposed Method

A. Notations

Assume that there are n pieces of multi-modal data $\mathbf{X}_I \in \mathbb{R}^{d_1 \times n}$ and $\mathbf{X}_T \in \mathbb{R}^{d_2 \times n}$ that represent image and text data, respectively, where d_1 and d_2 indicate the dimensions of image and text data, respectively. Their corresponding label matrix is denoted as $\mathbf{L} \in \{0, 1\}^{c \times n}$, where c represents the number of data categories. $\mathbf{L}_{ij} = 1$ if the j-th sample, either image or text, belongs to the i-th category; otherwise, it is 0. The aim is to simultaneously map $\mathbf{X}_I$ and $\mathbf{X}_T$ to a high-dimensional Hamming space and obtain a unified hash code $\mathbf{B} \in \{0, 1\}^{k \times n}$, where k denotes the dimension of the Hamming space. Unlike traditional dense hash codes, only r elements in each hash code of $\mathbf{B}$ are assigned a value of 1, and the rest are all 0. Thus, in this paper, r is utilized to indicate the length of the sparse hash code, while the sparse rate of the hash code is represented as $\tau = r/k$.

The other symbols used in this paper are defined as follows: $||\cdot||_F$ represents the Frobenius norm of a matrix. $||\cdot||_2$ represents the 2-norm of a vector. tr$(\cdot)$ represents the trace of a matrix. $\mathbf{1}_m$ represents an m-dimensional all-ones column vector. $\mathbf{I}_m$ represents an $m \times m$ identity matrix.

The proposed JSPSH is a two-stage model that consists of three main parts: semantic relationship exploring, hash codes learning, and hash functions learning. The overall framework of JSPSH is depicted in Fig. 2.

B. Semantic Relationship Exploring

1) Sample-Wise Semantics Relationship: We first leverage the label information to capture the sample-wise semantic relationship $\mathbf{S}_c$. In this semantic relationship, each sample is treated as an independent entity, and the similarity between

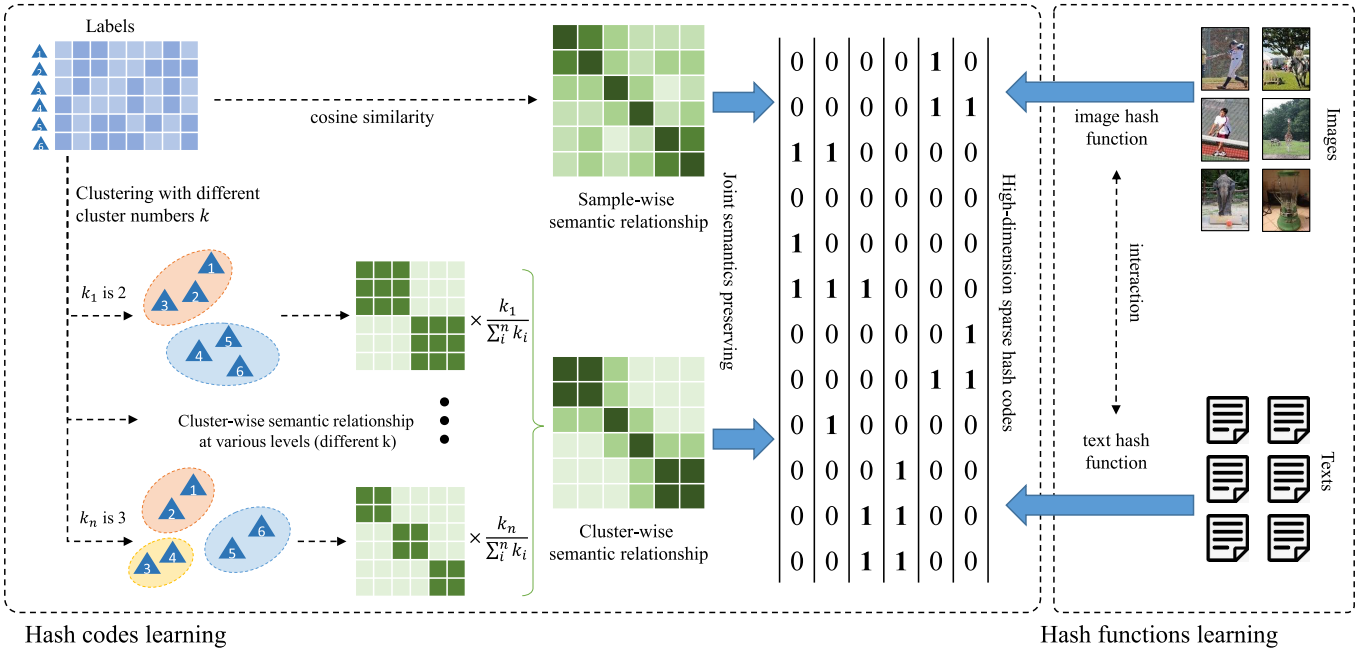


Fig. 2. The proposed JSPSH framework is a two-stage approach for learning hash codes. In the first stage, both sample-wise and cluster-wise semantic relationships are simultaneously extracted using label information. The sample-wise semantic relationship is obtained by computing the cosine distance between labels. To obtain different levels of cluster-wise semantic relationships, various cluster numbers are selected for clustering. Then, we compute the final cluster-wise semantic relationship as a weighted average of the cluster-wise semantic relationships at various levels. Finally, the sample-wise and cluster-wise semantic relationships are jointly used to train high-dimensional sparse hash codes. In the second stage, hash functions are learned for the different modalities using the learned hash codes. To reduce the heterogeneous gap between the modalities, a constraint is added between the different hash functions to enhance their interaction.

each pair of entities is calculated based on their corresponding labels. One of the most commonly used metrics is to compute the cosine similarity between the samples, resulting in an n-by-n similarity matrix $\mathbf{S}_s = \cos(\mathbf{L}, \mathbf{L})$. However, if we directly use $\mathbf{S}_s$ in the subsequent optimization process, the time complexity of the solution will be at least $O(n^2)$, making it challenging for the algorithm to be applied to large-scale datasets. To address this issue, we are inspired by [58] to decompose the cosine similarity calculation into a more efficient operation

$$\mathbf{S}_s = \bar{\mathbf{L}}^\top \bar{\mathbf{L}}, \tag{1}$$

where each column of $\bar{\mathbf{L}}$ is a normalized vector, i.e., $\bar{\mathbf{L}}_{*j} = \mathbf{L}_{*j}/||\mathbf{L}_{*j}||$. Since the dimension of $\bar{\mathbf{L}}$ is $c \times n$, we can prioritize left-side matrix multiplication in the subsequent optimization process to avoid generating an $n \times n$ matrix. This will help reduce both the time and space complexity.

It is evident that the value range of $\mathbf{S}_s$ in Eq. (1) falls within the interval [0, 1]. However, in traditional dense cross-modal methods, since the dense hash code values are either -1 or 1, their similarity values are limited to the range of $[-1, 1]$. To rectify this incompatibility, some methods [24], [29], [58] incorporate an offset term as follow

$$\mathbf{S}'_s = 2\bar{\mathbf{L}}^\top \bar{\mathbf{L}} - \mathbf{1}_n \mathbf{1}_n^\top, \mathbf{S}'_s \in [-1, 1]^{n \times n}. \tag{2}$$

Although the value ranges are aligned in Eq. (2), offset correction will lead to misclassification of positive samples in $\mathbf{S}_s$ ($0 < \mathbf{S}_s < 0.5$) as negative samples ($-1 < \mathbf{S}'_s < 0$). This problem arises because the traditional dense hash code has the ability to finely describe the relationship between negative sample pairs, i.e., it can calculate the specific value in the range [-1, 0] for the relationship between negative sample pairs. However, the similarity $\mathbf{S}_s$ obtained from labels usually marks the relationship between all negative sample pairs as 0. Therefore, simple offset correction does not fully resolve the inherent contradiction between the dense hash code and the similarity based on label construction.

In this paper, the use of high-dimensional sparse hashing allows for a perfect circumvention of this problem. The similarity calculated based on the sparse hash code $\mathbf{B} \in \{0, 1\}^{k \times n}$ also indicates the relationship between all negative sample pairs as 0, just like $\mathbf{S}_s$, resulting in a natural alignment with $\mathbf{S}_s$. Moreover, the powerful representation ability of sparse hash codes enables better mining of the relationship between all positive sample pairs.

2) Cluster-Wise Semantic Relationship: While the sample-wise semantic relationship has been widely used and shown satisfactory performance [22], [23], [29], [30], [58], it overlooks the overall distribution of labels that may play a critical role in further improving retrieval results. For instance, in the example illustrated in Fig. 1, if the sample-wise semantic similarity between label A and other labels is the same, we desire it to be closer to the label that contains more samples, which could ensure that more correct results can be retrieved. To this end, we introduce cluster-wise semantic relationship to capture this similarity tendency. Specifically, we hope to further enhance the retrieval results by exploring which labels should be closer or clustered together based on the distribution of labels.

To obtain the cluster-wise semantic relationship, we treat each label in $\mathbf{L}$ as a feature and use k-means algorithm to cluster $\mathbf{L}$. Based on the clustering results, we define the cluster-wise semantic similarity between two samples as follow:

$$\mathbf{S}_{ij} = \begin{cases} 1, & \text{if } \mathcal{C}(i,j) = 1 \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $\mathcal{C}(i,j) = 1$ indicates that the i-th label and j-th label belong to the same cluster.

As Eq. (3) shows, $\mathbf{S}$ is an $n \times n$ matrix, which would also result in an $O(n^2)$ time complexity as analyzed previously. To avoid this problem, we propose assigning new labels to samples based on the clustering results. Specifically, we treat all samples within the same cluster as the same class and assign the same one-hot label to them. Then, we obtain a new label matrix $\tilde{\mathbf{L}} \in \{0,1\}^{p \times n}$, where p is the number of clusters specified in the clustering algorithm. With these new labels, we can calculate the cluster-wise semantic similarity of the data using the following formula:

$$\tilde{\mathbf{S}}_c = \tilde{\mathbf{L}}^\top \tilde{\mathbf{L}}. \tag{4}$$

Same as Eq. (1), the time complexity $O(n^2)$ can be avoided by prioritizing left-side matrix multiplication.

During the clustering of labels, a thorny issue is determining the optimal number of clusters p. Given that label distribution varies across datasets, it is challenging to set the most appropriate p for each dataset. Fortunately, as clustering is not the ultimate objective of our proposed approach, we could focus less on the selection of p. Our goal is just to extract cluster-wise semantic information between samples through clustering. Consequently, we can instead extract different levels of cluster-wise semantic information by varying the value of p. Specifically, we can choose m different numbers of clusters, denoted as $\{p_i\}_{i=1}^m$. With different p_i, we can obtain different clustering results and corresponding new labels $\tilde{\mathbf{L}}^{(i)}$. Furthermore, this enables us to obtain a series of cluster-wise semantic similarity matrices

$$\tilde{\mathbf{S}}_c^{(i)} = \tilde{\mathbf{L}}^{(i)\top} \tilde{\mathbf{L}}^{(i)}, \ i = 1, 2, \ldots, m. \tag{5}$$

To leverage cluster-wise semantic relationships across different levels simultaneously, we compute the final cluster-wise semantic similarity $\mathbf{S}_c$ as a weighted average of the cluster-wise semantic similarities $\tilde{\mathbf{S}}_c^{(i)}$ obtained at different numbers of clusters p_i. Specifically, we use different weights w_i to adjust the contribution of each level of clustering to the final cluster-wise semantic similarity, that is,

$$\mathbf{S}_c = \sum_{i=1}^{m} w_i \tilde{\mathbf{S}}_c^{(i)} = \sum_{i=1}^{m} w_i \tilde{\mathbf{L}}^{(i)\top} \tilde{\mathbf{L}}^{(i)}, \ s.t. \ \sum_{i}^{m} w_i = 1. \tag{6}$$

Considering that a larger number of clusters p_i will result in stronger correlations between samples belonging to the same cluster, we believe the corresponding relationship $\tilde{\mathbf{S}}_c^{(i)}$ to be more informative. Therefore, we set the weights w_i in Eq. (6) proportional to p_i. Then, the weights are computed as follows:

$$w_i = \frac{p_i}{\sum_{i=1}^{m} p_i}, \ i = 1, 2, \ldots, m. \tag{7}$$

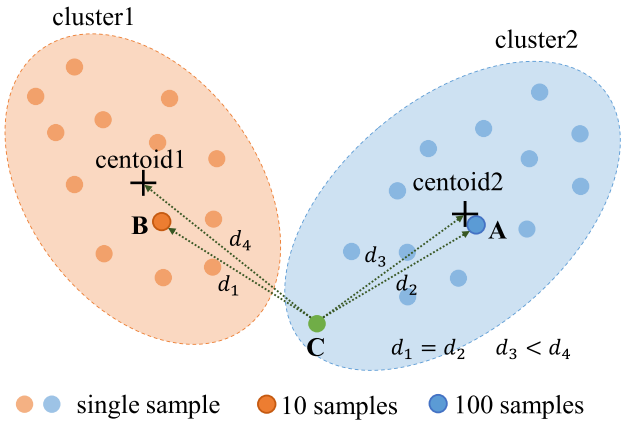


Fig. 3. When the sample-wise semantic relationship between C and B and that between C and A are the same, i.e., $d_1 = d_2$, k-means algorithm will cluster C with A because there are more samples corresponding to label A. By preserving this cluster-wise semantic relationship, it can be guaranteed that more semantically similar samples are clustered around C in the retrieval set.

Remark. Why can clustered results provide effective cluster-wise semantic relationship which benefits the retrieval results? On one hand, clustering labels that are semantically similar enhances the sample-wise semantic relationship. In other words, it helps identify which sample-wise semantic relationships need to be highlighted. On the other hand, when the sample-wise semantic relationship between labels is the same, clustering results can provide better ranking. For instance, in Fig. 3, assume that the sample-wise semantic relationship between C and B, and C and A is the same, i.e., $d_1 = d_2$. Since there are more samples corresponding to label A, the center point of cluster 2 will be closer to A. Therefore, in the clustering process, C will be closer to the center point of cluster 2, i.e., $d_3 < d_4$, and C will be clustered with A. This cluster-wise semantic relationship tends to make C and A closer to ensure that more semantically similar samples are gathered around. This decision is more advantageous when A and B are negative samples of each other. For instance, suppose that the labels A,B, and C correspond to 001, 100, and 101, respectively. In this case, it is better to make C closer to A because it can ensure more correct retrieval results.

C. Hash Codes Learning

After obtaining the sample-wise and cluster-wise semantic relationships, we will use them to jointly learn unified hash codes $\mathbf{B}$. The learned hash codes $\mathbf{B}$ should ideally preserve the semantic information at both the sample and cluster levels. To this end, we define following object function:

$$\begin{aligned} &\min_{\mathbf{B}} ||\mathbf{B}^\top \mathbf{B} - r\mathbf{S}_s||_F^2 + \alpha ||\mathbf{B}^\top \mathbf{B} - r\mathbf{S}_c||_F^2, \\ &s.t. \ \mathbf{B} \in \{0,1\}^{k \times n}, \mathbf{B}^\top \mathbf{1}_k = r\mathbf{1}_n, \end{aligned} \tag{8}$$

where hyper-parameter α is used to balance the ratio between the two types of semantic relationships. We have also introduced two constraints to the function. Specifically, $\mathbf{B} \in \{0,1\}^{k \times n}$ and $\mathbf{B}^\top \mathbf{1}_k = r\mathbf{1}_n$ ensure binary values and the sparsity of the learned hash codes $\mathbf{B}$, respectively. However, they have also made the optimization of the object function

Eq. (8) into an NP-Hard problem. To address this challenge, we adopt an asymmetric hashing strategy [59] and introduce an intermediate variable $\mathbf{H} \in \mathbb{R}^{k\times n}$. Specifically, we remove the discrete constraints of one $\mathbf{B}$ in the matrix multiplication and transform it into a continuous variable $\mathbf{H}$. We then add a constraint item between $\mathbf{B}$ and $\mathbf{H}$ to reduce quantitative losses. Additionally, to minimize redundancy among different bits of the hash codes, we further apply an orthogonal constraint on $\mathbf{H}$. As a result, Eq. (8) is transformed into the following problem:

$$\begin{aligned}&\min_{\mathbf{H},\mathbf{B}} ||\mathbf{H}^\top\mathbf{B} - r\mathbf{S}_s||_F^2 + \alpha||\mathbf{H}^\top\mathbf{B} - r\mathbf{S}_c||_F^2 + \beta||\mathbf{B}-\mathbf{H}||_F^2,\\ &s.t.\ \mathbf{B}\in\{0,1\}^{k\times n}, \mathbf{B}^\top\mathbf{1}_k = r\mathbf{1}_n, \mathbf{H}\mathbf{H}^\top = nr/k\mathbf{I}_k. \end{aligned} \tag{9}$$

Then, we can disassemble the solution of Eq. (9) into two steps of **H-Step** and **B-Step** to optimize them alternately.

H-Step: Fix $\mathbf{B}$, Eq. (9) can be reformulated into the following sub-problem:

$$\begin{aligned}&\max_{\mathbf{H}} \ \mathrm{tr}((r\mathbf{B}\mathbf{S}_s + \alpha r\mathbf{B}\mathbf{S}_c + \beta\mathbf{B})\mathbf{H}^\top),\\ &s.t.\ \mathbf{H}\mathbf{H}^\top = nr/k\mathbf{I}_k. \end{aligned} \tag{10}$$

We use $\mathbf{V} = r\mathbf{B}\mathbf{S}_s + \alpha r\mathbf{B}\mathbf{S}_c + \beta\mathbf{B}$. According to [30] and [60], the optimal solution of Eq. (10) is given by

$$\mathbf{H} = \sqrt{nr/k}[\mathbf{Q}, \bar{\mathbf{Q}}][\mathbf{T}, \bar{\mathbf{T}}]^\top, \tag{11}$$

where the matrix $\mathbf{Q}$ is obtained from the eigen-decomposition of matrix $\mathbf{V}\mathbf{V}^\top$. Define

$$\mathbf{V}\mathbf{V}^\top = [\mathbf{Q}, \tilde{\mathbf{Q}}]\begin{bmatrix}\Sigma & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}[\mathbf{Q}, \tilde{\mathbf{Q}}]^\top, \tag{12}$$

where $\Sigma \in \mathbb{R}^{k'\times k'}$ is the diagonal positive eigenvalue matrix, and k' is the rank of $\mathbf{V}\mathbf{V}^\top$. Matrix $\mathbf{Q} \in \mathbb{R}^{k\times k'}$ consists of corresponding eigenvectors of positive eigenvalues and $\tilde{\mathbf{Q}} \in \mathbb{R}^{k\times(k-k')}$ consists of $k-k'$ eigenvectors of eigenvalue 0. Then, $\bar{\mathbf{Q}} \in \mathbb{R}^{k\times(k-k')}$ can be obtained by performing the Gram-Schmidt process on $\tilde{\mathbf{Q}}$. Matrix $\mathbf{T} = \mathbf{V}^\top\mathbf{Q}\Sigma^{-1/2} \in \mathbb{R}^{n\times k'}$ and $\bar{\mathbf{T}} \in \mathbb{R}^{n\times(k-k')}$ is a random orthogonal matrix.

Considering that the calculation of $\mathbf{V}$ involves the matrix multiplication of $\mathbf{S}$, which can result in a time complexity of $O(n^2)$, we propose to calculate $\mathbf{V}$ using the following formula

$$\mathbf{V} = r(\mathbf{B}\bar{\mathbf{L}}^\top)\bar{\mathbf{L}} + \alpha r\sum_{i=1}^{m} w_i(\mathbf{B}\tilde{\mathbf{L}}^{(i)\top})\tilde{\mathbf{L}}^{(i)} + \beta\mathbf{B}. \tag{13}$$

As a result, by prioritizing left-side matrix multiplication, the time complexity of $\mathbf{V}$ decreases from $O(kn^2)$ to $O(ckn)$, where $c, k \ll n$. Section III-F gives a detailed analysis.

B-Step: Fix $\mathbf{H}$, Eq. (9) can be reformulated into the following sub-problem:

$$\begin{aligned}&\max_{\mathbf{B}} \ \mathrm{tr}((r\mathbf{H}\mathbf{S}_s + \alpha r\mathbf{H}\mathbf{S}_c + \beta\mathbf{H})\mathbf{B}^\top),\\ &s.t.\ \mathbf{B}\in\{0,1\}^{k\times n},\ \mathbf{B}^\top\mathbf{1}_k = r\mathbf{1}_n. \end{aligned} \tag{14}$$

The optimal solution is given by

$$\begin{aligned}\mathbf{B} &= \mathrm{sign}_r(r\mathbf{H}\mathbf{S}_s + \alpha r\mathbf{H}\mathbf{S}_c + \beta\mathbf{H})\\ &= \mathrm{sign}_r(r(\mathbf{H}\bar{\mathbf{L}}^\top)\bar{\mathbf{L}} + \alpha r\sum_{i=1}^{m} w_i(\mathbf{H}\tilde{\mathbf{L}}^{(i)\top})\tilde{\mathbf{L}}^{(i)} + \beta\mathbf{H}),\end{aligned} \tag{15}$$

where sign_r is a function that transforms a real-number vector x into a string of sparse hash code and is defined as follow:

$$\mathrm{sign}_r(x) = \begin{cases}1, & \text{if } x \text{ is the top-}r \text{ largest elements}\\ 0, & \text{otherwise}\end{cases}. \tag{16}$$

The winner-take-all strategy is adopted by the $\mathrm{sign}_r(x)$. This strategy activates only the largest r-bit elements in x and leaves the rest to 0.

D. Hash Functions Learning

After obtaining the hash codes, it is necessary to learn the hash functions that map the data of different modalities to the hash codes. One conventional approach is to use a linear classification model, that is,

$$\min_{\mathbf{P}_*} ||\mathbf{B} - \mathbf{P}_*\mathbf{X}_*||_F^2,\ *=\{I,T\}, \tag{17}$$

where $\mathbf{P}_*$ denotes the hash functions to be learned. This approach considers each bit of data mapping to a hash code as a distinct binary classification problem. Nevertheless, since $\mathbf{B}$ is strictly binary and $\mathbf{P}_*\mathbf{X}_*$ is continuous, there will inevitably be a residual distance between them, and its direction will be uncontrollable. These errors affect the validity of the generated hash codes, especially due to the winner-take-all strategy used to generate high-dimension sparse hash codes during the retrieval phase. To address this issue, [30] has proposed introducing an error correction term and using sample-wise semantic information to enhance the constraints on the mapping function as follow

$$\min_{\mathbf{P}_*} ||\mathbf{B} - \mathbf{P}_*\mathbf{X}_*||_F^2 + \gamma||r\mathbf{S}_c - \mathbf{B}^\top(\mathbf{P}_*\mathbf{X}_*)||_F^2,\ *=\{I,T\}, \tag{18}$$

where γ is generally a hyper-parameter with a small value to control the degree of error correction.

However, the aforementioned two methods have a limitation: there is a lack of interaction between modalities during the hash function learning process, which can result in misalignment of the hash codes of different modalities. In the hash function learning stage, it is assumed that data of different modalities share the same hash code $\mathbf{B} = \mathbf{B}_I = \mathbf{B}_T$, where $\mathbf{B}_I$ and $\mathbf{B}_T$ represent the hash codes of image data $\mathbf{X}_I$ and text data $\mathbf{X}_T$, respectively. However, Eq. (17) and Eq. (18) essentially use $\mathbf{B}_I$ and $\mathbf{B}_T$ independently to learn hash functions for different modalities, which weakens the assumption $\mathbf{B}_I = \mathbf{B}_T$. This can cause misalignment of the hash codes of different modalities, as shown in Fig. 4. Although both distances from $\mathbf{P}_I\mathbf{X}_I$ and $\mathbf{P}_T\mathbf{X}_T$ to $\mathbf{B}$ are small, the directions are different. Ideally, we would like to achieve the effect in Fig. 4(b). To address this, we introduce an ***interaction term*** $\mathbf{P}_I\mathbf{X}_I - \mathbf{P}_T\mathbf{X}_T$ in the hash function learning stage, which re-emphasizes the assumption $\mathbf{B}_I = \mathbf{B}_T$. Consequently, the overall optimization function becomes

$$\begin{aligned}\min_{\mathbf{P}_I,\mathbf{P}_T} &||\mathbf{B} - \mathbf{P}_I\mathbf{X}_I||_F^2 + \gamma||r\mathbf{S}_c - \mathbf{B}^\top(\mathbf{P}_I\mathbf{X}_I)||_F^2\\ &+ ||\mathbf{B} - \mathbf{P}_T\mathbf{X}_T||_F^2 + \gamma||r\mathbf{S}_c - \mathbf{B}^\top(\mathbf{P}_T\mathbf{X}_T)||_F^2\\ &+ \mu||\mathbf{P}_I\mathbf{X}_I - \mathbf{P}_T\mathbf{X}_T||_F^2 + \lambda R(\mathbf{P}_I, \mathbf{P}_T),\end{aligned} \tag{19}$$

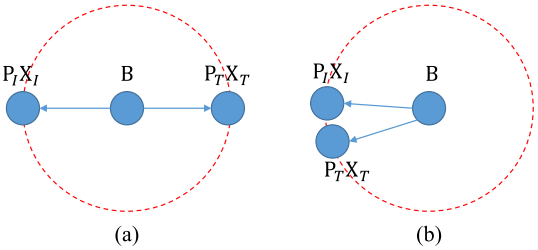


Fig. 4. When hash codes of different modalities are not aligned, two different situations can arise: (a) both $\mathbf{P}_I\mathbf{X}_I$ and $\mathbf{P}_T\mathbf{X}_T$ have small distances to $\mathbf{B}$ but in different directions, and (b) both $\mathbf{P}_I\mathbf{X}_I$ and $\mathbf{P}_T\mathbf{X}_T$ have small distances to $\mathbf{B}$ and in the same direction.

where μ and λ are two hyper-parameters and $R(\mathbf{P}_I, \mathbf{P}_T) = ||\mathbf{P}_I||_F^2 + ||\mathbf{P}_T||_F^2$ represents the regularization term imposed on $\mathbf{P}_I$ and $\mathbf{P}_T$. In Eq. (19), we use only the sample-wise semantic relationship for error correction. There are two reasons for this decision. Firstly, we believe that hash codes $\mathbf{B}$ have effectively integrated both sample-wise and cluster-wise semantic information in the hash codes learning stage. Secondly, using multiple standards for error correction, i.e., using both sample-wise and cluster-wise semantic relationships simultaneously, may introduce contradictions and be counterproductive for learning hash functions.

Finally, we can alternately solve $\mathbf{P}_I$ and $\mathbf{P}_T$ to optimize Eq. (19) as follows

$$\begin{aligned}\mathbf{P}_I = ((1+\lambda)\mathbf{I}_k + \gamma\mathbf{B}\mathbf{B}^\top)^{-1}(\mathbf{B}\mathbf{X}_I^\top + \mu\mathbf{P}_T\mathbf{X}_T\mathbf{X}_I^\top \\ + \gamma r(\mathbf{B}\bar{\mathbf{L}}^\top)(\tilde{\mathbf{L}}\mathbf{X}_I^\top))(\mathbf{X}_I\mathbf{X}_I^\top + \omega\mathbf{I}_{d_1})^{-1}, \quad (20)\\ \mathbf{P}_T = ((1+\lambda)\mathbf{I}_k + \gamma\mathbf{B}\mathbf{B}^\top)^{-1}(\mathbf{B}\mathbf{X}_T^\top + \mu\mathbf{P}_I\mathbf{X}_I\mathbf{X}_T^\top \\ + \gamma r(\mathbf{B}\bar{\mathbf{L}}^\top)(\tilde{\mathbf{L}}\mathbf{X}_T^\top))(\mathbf{X}_T\mathbf{X}_T^\top + \omega\mathbf{I}_{d_2})^{-1}, \quad (21)\end{aligned}$$

where $\omega\mathbf{I}_{d_1}$ and $\omega\mathbf{I}_{d_2}$ are two small items ($\omega = 0.01$) to avoid the singularity of matrix $\mathbf{X}_*\mathbf{X}_*^\top$.

Compared to previous methods [28], [29], [30], [61] that only involve data from the corresponding modalities in training hash functions, our proposed optimization process simultaneously involves data from all modalities in the training process. For example, when solving $\mathbf{P}_I$, both $\mathbf{X}_I$ and $\mathbf{X}_T$ are involved, which enhances the interaction between different modalities. This interaction not only narrows the heterogeneous gap but also allows for the use of information from multiple modalities to learn a better hash function $\mathbf{P}_*$.

The whole training process of JSPSH including semantic relationship exploring, hash codes learning, and hash functions learning is summarized in Algorithm 1.

Algorithm 1 JSPSH

Input: Cluster number $\{p_i\}_{i=1}^m$, Image data $\mathbf{X}_I$, text data $\mathbf{X}_T$, and corresponding labels $\mathbf{L}$;
Output: Unified hash codes $\mathbf{B}$, image hash function $\mathbf{P}_I$, and text hash function $\mathbf{P}_T$;

Semantic relationship exploring:
for $i = 1$ *to* m **do**
  Use k-means algorithm to cluster $\mathbf{L}$ into p_i clusters;
  Assign new label $\tilde{\mathbf{L}}^{(i)}$ to data based on the clustering results;
end
Hash codes learning:
Randomly initialize $\mathbf{B}$ and $\mathbf{H}$ with a standard normal distribution;
for $iter = 1$ *to max iteration* **do**
  Update $\mathbf{H}$ by Eq. (11);
  Update $\mathbf{B}$ by Eq. (15);
end
Hash functions learning:
for $iter = 1$ *to max iteration* **do**
  Update $\mathbf{P}_I$ by Eq. (20);
  Update $\mathbf{P}_T$ by Eq. (21);
end

E. Proof of Convergence

In this section, we analyze the convergence of JSPSH. During the hash code learning stage, all variables $\mathbf{B}$ and $\mathbf{H}$ have closed-form solutions to their corresponding sub-problems. Let $\mathcal{L}(\mathbf{B}, \mathbf{H})$ denote the value of the object function Eq. (9), and we have $\mathcal{L}(\mathbf{B}^{t+1}, \mathbf{H}^{t+1}) \leq \mathcal{L}(\mathbf{B}^{t+1}, \mathbf{H}^t) \leq \mathcal{L}(\mathbf{B}^t, \mathbf{H}^t)$, where t is the number of iterations. According to the bounded monotone convergence theory [62], the algorithm will converge to a stable solution. Similarly, during the hash functions learning stage, all variables $\mathbf{P}_I$ and $\mathbf{P}_T$ have closed-form solutions to their corresponding sub-problems. Using $\mathcal{L}(\mathbf{P}_I, \mathbf{P}_T)$ to denote the value of the object function Eq. (19), we have $\mathcal{L}(\mathbf{P}_I^{t+1}, \mathbf{P}_T^{t+1}) \leq \mathcal{L}(\mathbf{P}_I^{t+1}, \mathbf{P}_T^t) \leq \mathcal{L}(\mathbf{P}_I^t, \mathbf{P}_T^t)$. In summary, the convergence of the JSPSH algorithm can be guaranteed.

F. Complexity Analysis

The JSPSH algorithm involves three main components: label clustering, hash code learning, and hash function learning. The time complexity of the label clustering stage is $O(\sum_i^m tcp_in)$, where t is the maximum iteration. It is important to note that this stage is performed only once, and the results are saved and utilized for subsequent calculations. Therefore, the time complexity of this stage is not counted. In the hash codes learning stage, the time complexity of solving $\mathbf{H}$ and $\mathbf{B}$ in each round are $O(ckn + \sum_i^m kp_in + kn + k^2n + k^3)$ and $O(ckn + \sum_i^m kp_in + kn + nk\log_2 r)$, respectively. In the hash functions learning stage, the time complexities of solving $\mathbf{P}_1$ and $\mathbf{P}_2$ in each round are $O(k^2(k+n+1) + kn(d_1 + d_2 + c) + cd_1(n+k) + kd_1 + d_1^2(n + d_1 + 1) + kd_1(k + d_1))$ and $O(k^2(k+n+1) + kn(d_1 + d_2 + c) + cd_2(n+k) + kd_2 + d_2^2(n + d_2 + 1) + kd_2(k + d_2))$, respectively. As k, c, r, d_1, d_2, and p_i are all constants and much smaller than n, the time complexity of the JSPSH algorithm can be considered linear to the size of the training set n, i.e., $O(n)$. Therefore, it can efficiently process large-scale datasets.

IV. Experiment

A. Experimental Settings

1) Datasets: To measure the retrieval ability of JSPSH, we conducted experiments on three commonly used large-scale datasets, including MIRFlickr [63], IAPR TC-12 [64] and NUS-WIDE [65].

TABLE I
THE MAP RESULTS (MAP@50) OF THE PROPOSED JSPSH AND OTHER COMPARED BASELINES ON THREE DATASETS. THE BEST RESULTS ARE IN BOLDFACE

Task	Methods	MIRFlickr					NUS-WIDE					IAPR TC-12				
		2bits	4bits	8bits	16bits	32bits	2bits	4bits	8bits	16bits	32bits	2bits	4bits	8bits	16bits	32bits
I2T	DCH	0.6725	0.7131	0.7186	0.7267	0.7517	0.5717	0.6151	0.6865	0.7113	0.6817	0.4058	0.5514	0.6146	0.6359	0.6416
	SCRATCH	0.5404	0.7065	0.7667	0.7709	0.7910	0.5618	0.5956	0.6777	0.6663	0.6772	0.2639	0.3489	0.4317	0.4940	0.5313
	DLFH	0.5817	0.7624	0.8003	0.8548	0.8539	0.3823	0.6014	**0.8024**	0.8087	**0.8621**	0.3645	0.4845	0.5542	0.6555	0.7016
	LFMH	0.6474	0.6553	0.7570	0.8051	0.8197	0.4901	0.5642	0.5645	0.6954	0.7786	0.4327	0.4845	0.5443	0.5694	0.6310
	BATCH	0.6585	0.7127	0.8030	0.8263	0.8470	0.5615	0.6900	0.7498	0.7983	0.7984	0.4215	0.4261	0.5193	0.6270	0.6818
	WATCH	0.6968	0.6852	0.7436	0.7670	0.7846	0.5573	0.6241	0.6790	0.7481	0.7724	0.4069	0.4086	0.4586	0.5736	0.6837
	ALECH	0.6727	0.7257	0.7792	0.8202	0.8487	0.6015	0.7084	0.7486	0.7728	0.7890	0.4190	0.4505	0.5331	0.6222	0.6793
	HSCH	0.7736	**0.8765**	0.8668	0.8654	0.8796	0.6908	0.6809	0.7221	0.7474	0.7839	0.6327	0.6878	0.7110	0.7341	0.7482
	JSPSH	**0.7901**	0.8421	**0.8858**	**0.8973**	**0.8989**	**0.7284**	**0.7472**	0.7857	**0.8089**	0.8268	**0.6384**	**0.7175**	**0.7393**	**0.7576**	**0.7733**
T2I	DCH	0.6986	0.7719	0.8517	0.8701	0.8787	0.6454	0.7420	0.8058	0.8511	0.8331	0.4347	0.5913	0.7310	0.7864	0.8142
	SCRATCH	0.5400	0.7391	0.8020	0.8096	0.7804	0.5601	0.6079	0.7792	0.7594	0.7647	0.2562	0.3711	0.5385	0.6778	0.7396
	DLFH	0.5817	0.7894	0.8504	0.8898	0.9085	0.3918	0.7309	0.8340	0.8625	**0.9161**	0.3655	0.5762	0.5810	0.6906	0.7598
	LFMH	0.6936	0.7678	0.8561	0.9004	0.9101	0.5601	0.6634	0.7516	0.8331	0.8770	0.4403	0.5066	0.6524	0.7471	0.8291
	BATCH	0.7601	0.8225	0.8761	0.8923	0.8969	0.7315	0.7880	0.8440	0.8705	0.8773	0.4427	0.5364	0.6501	0.7918	0.8412
	WATCH	0.7535	0.7962	0.8454	0.8744	0.8829	0.6664	0.7239	0.8191	0.8459	0.8655	0.4351	0.4507	0.5539	0.7083	0.8371
	ALECH	0.7516	0.8271	0.8740	0.8972	0.8938	0.6738	0.8089	0.8572	0.8537	0.8617	0.4654	0.5729	0.6683	0.7809	0.8466
	HSCH	0.8673	0.9174	0.9218	0.9233	0.9230	0.8345	0.8550	0.8723	0.8759	0.8879	0.7766	0.8503	0.8737	0.8918	0.8956
	JSPSH	**0.8861**	**0.9213**	**0.9348**	**0.9415**	**0.9423**	**0.8617**	**0.8683**	**0.8803**	**0.8905**	0.8950	**0.8137**	**0.8627**	**0.8881**	**0.8945**	**0.9064**

MIRFlickr is a dataset that comprises 25,000 image-text pairs, divided into 24 categories. Each image is represented by a 512-dimensional GIST feature, and each text is represented by a 1,386-dimensional bag-of-words vector. To ensure effective training, we eliminated data with textual tags less than 20 and selected 20,015 pairs of valid data. From the remaining data, we randomly selected 2,000 data points as the query set and used the rest for retrieval and training sets.

IAPR TC-12 dataset consists of 20,000 image-text pairs with a total of 255 different classes. Each piece of data is labeled with at least one of these categories. Each image is represented by a 512-dimensional GIST feature, and each text is represented by a 2,912-dimensional bag-of-words vector. Following the setting in [30], we randomly selected 2000 data points as the query set, and used the remaining data points for retrieval and training.

NUS-WIDE is a larger dataset compared to the previous two datasets, consisting of 269,648 image-text pairs and 81 different categories. Following the settings in [43], for the experiments conducted in this paper, only the 10 most frequently occurring categories of samples, totaling 186,577 pairs, were used. Each image is represented by a 500-dimensional SIFT feature, while the corresponding text is represented by a 1,000-dimensional binary tagging vector representation. We randomly selected 2,000 pieces of data as the query set, while the remaining samples were used as the retrieval and training sets.

2) Evaluation Metrics: In this paper, we conducted two cross-modal retrieval tasks: I2T, which retrieves images based on text queries, and T2I, which retrieves text based on image queries. We employed three commonly used metrics to evaluate the performance of JSPSH and all compared methods, namely mean average precision (mAP), precision-recall (PR) curve, and top-K precision curve. A higher mAP and top-K precision value as well as a larger area under the PR curve indicate better retrieval performance. When calculating precision, we considered a search result to be correct if it shares at least one label with the query.

3) Baselines and Implementation Details: To verify the effectiveness of the proposed JSPSH, we compared it with nine state-of-the-art cross-modal hashing methods, including DCH [36], SCRATCH [12], DLFH [66], LFMH [67], BATCH [68], WATCH [28], ALECH [29] and HSCH [30]. Among these methods, HSCH is the only high-dimensional sparse cross-modal hashing method, while the remaining methods are traditional dense hashing methods. The codes for all comparison methods are kindly provided by their authors, and all parameters follow the settings in the corresponding papers. All experiments are conducted on the server equipped with Intel i7-12700KF CPU@ 3.7 GHZ and 64 GB RAM.

4) Parameters Setting: The parameters used in JSPSH are set as follows: $\alpha = 1$, $\beta = 10$, $\mu = 3$, $\gamma = 0.01$, $\lambda = 0.01$, and $\tau = 0.05$ for all datasets. However, since the label distribution varies across datasets, we set different clustering parameters $\{p_i\}_{i=1}^{m}$ for each dataset. Specifically, we set clustering parameters to {100, 200, 500} for MIRFlickr and IAPR TC-12 datasets, whereas for the larger scale NUS-WIDE dataset, we set clustering parameters to {50, 100, 200, 500, 1000}.

B. Retrieval Performance

In this section, we analyze the retrieval performance of the proposed JSPSH and compare it with other methods from three aspects. Table I presents the mAP results of all methods on the three datasets. Moreover, Fig. 5 and Fig. 6 illustrate the PR curve and top-K precision curve of all methods on the MIRFlickr dataset, respectively, with hash code lengths varying from 2 to 32 bits. Based on these results, we draw the following conclusions:

- The superiority of high-dimensional sparse hashing methods, JSPSH and HSCH, over traditional dense hashing algorithm is evident from the mAP results presented in Table I. In particular, JSPSH and HSCH exhibit robustness in low-dimensional scenarios, such as $r = 2$ or 4, thereby demonstrating their potential in encoding abundant information using a fewer number of hash bits. This highlights the representation capability of

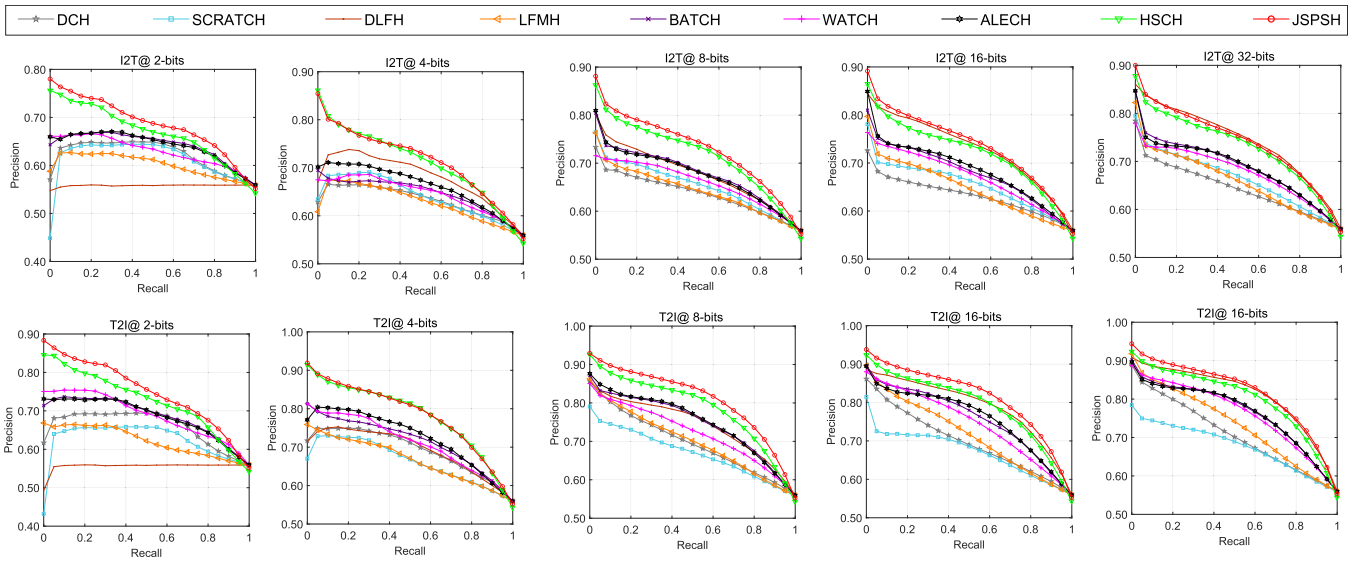


Fig. 5. The PR curves of JSPSH and compared baselines on MIRFlickr.

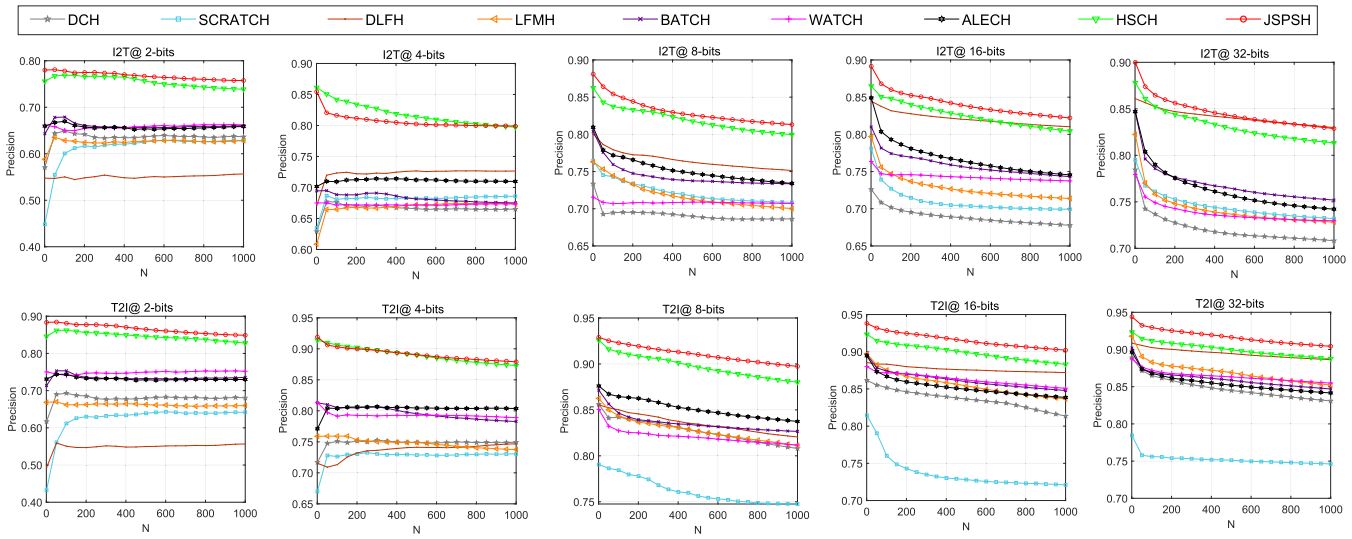


Fig. 6. The top-K precision curves of JSPSH and compared baselines on MIRFlickr.

high-dimensional sparse hash codes, thereby proving their efficacy in the field of retrieval tasks.

- JSPSH consistently outperforms HSCH in terms of retrieval performance, which highlights the efficacy of cluster-wise semantic relationships. Both JSPSH and HSCH leverage high-dimensional sparse hash codes to encode information, with the main difference being that HSCH only uses sample-wise semantic relationships in hash code learning while JSPSH utilizes both sample-wise and cluster-wise semantic relationships. The supervised learning of hash codes with the help of cluster-wise semantic relationships provides more precise information based on label distribution to determine which hash codes should be closer in Hamming space, resulting in better grouping of semantically similar samples and superior retrieval performance.

The PR curve and top-K precision curve depicted in Fig. 5 and Fig. 6 further support these analyses. It is evident that the gap between traditional dense hashing methods and high-dimensional sparse hashing methods is substantial, particularly when the dimension of hash codes is low, such as $r = 2$ and 4. When comparing JSPSH and HSCH, it is observed that JSPSH consistently outperforms HSCH in terms of retrieval precision, under the same recall rate. Furthermore, JSPSH always ensures that a higher number of relevant samples appear within the top-K retrieved results, except when $r = 4$. These observations suggest that JSPSH is better suited to ensure that semantically similar samples are distributed around the query. In other words, with the help of cluster-wise semantic information, JSPSH can ensure that samples are more appropriately clustered in the retrieval set.

C. Efficiency Analyses

In Section III-F, we presented a theoretical analysis showing that the time complexity of JSPSH is linearly related to the size of the training set. To validate this analysis, we provide experimental data on the training time complexity, training

TABLE II
THE TRAIN TIME COMPLEXITY, TRAINING TIME (SECONDS), AND RETRIEVAL TIME (SECONDS) OF THE PROPOSED JSPSH AND OTHER COMPARED BASELINES ON MIRFLICKR DATASET

Methods	Train Time Complexity	Training Time					Retrieval Time				
		2bits	4bits	8bits	16bits	32bits	2bits	4bits	8bits	16bits	32bits
DCH	$O(n^2)$	91.3000	91.7000	95.7600	103.6200	116.7200	0.0213	0.0227	0.0241	0.0256	0.0306
SCRATCH	$O(n)$	0.1450	0.1404	0.1699	0.2123	0.2769	0.0212	0.0217	0.0232	0.0242	0.0296
DLFH	$O(n)$	0.4440	0.5070	1.2300	3.1100	8.4690	0.0220	0.0222	0.0241	0.0242	0.0301
LFMH	$O(n)$	11.5430	11.5262	12.2425	12.3298	12.7813	0.0210	0.0218	0.0233	0.0237	0.0292
BATCH	$O(n)$	0.1463	0.1630	0.1762	0.2043	0.2450	0.0203	0.0214	0.0231	0.0230	0.0283
WATCH	$O(n)$	1.0581	1.0670	1.1542	1.1940	1.4222	0.0205	0.0212	0.0228	0.0236	0.0286
ALECH	$O(n)$	0.3759	0.3668	0.4246	0.4980	0.5246	0.0212	0.0226	0.0235	0.0249	0.0298
HSCH	$O(n)$	0.9629	1.2864	1.8052	3.0035	5.7168	0.0220	0.0220	0.0228	0.0252	0.0325
JSPSH	$O(n)$	1.3120	1.5303	1.8770	2.8482	4.4943	0.0245	0.0248	0.0244	0.0277	0.0320

TABLE III
THE MAP RESULTS (MAP@50) OF JSPSH AND ITS FOUR VARIANTS ON MIRFLICKR AND IAPR TC-12 DATASETS. THE BEST RESULTS ARE IN BOLDFACE

Task	Methods	MIRFlickr					IAPR TC-12				
		2bits	4bits	8bits	16bits	32bits	2bits	4bits	8bits	16bits	32bits
I2T	JSPSH-1	0.7787	0.8614	0.8667	0.8760	0.8877	0.6074	0.6943	0.7152	0.7442	0.7487
	JSPSH-2	0.7887	**0.8704**	0.8538	0.8870	0.8958	0.6315	0.6816	0.7029	0.7366	0.7665
	JSPSH-3	0.7739	0.8408	0.8735	0.8871	0.8936	**0.6441**	0.7073	0.7265	**0.7643**	0.7659
	JSPSH-4	0.7842	0.8362	0.8762	0.8914	0.8983	0.6297	0.7116	0.7313	0.7439	0.7627
	JSPSH-5	0.7227	0.7600	0.7829	0.7917	0.8403	0.4359	0.5406	0.6219	0.6498	0.6729
	JSPSH	**0.7901**	0.8421	**0.8858**	**0.8973**	**0.8989**	0.6384	**0.7175**	**0.7393**	0.7576	**0.773**3
T2I	JSPSH-1	0.8754	0.9157	0.9275	0.9359	0.9387	0.7732	0.8565	0.8753	0.8863	0.9027
	JSPSH-2	0.8807	0.9184	0.9187	0.9373	0.9410	0.7880	0.8265	0.8559	0.8793	0.8954
	JSPSH-3	0.8773	0.9140	0.9290	0.9378	0.9418	0.8101	0.8615	0.8794	**0.8951**	0.9048
	JSPSH-4	0.8844	0.9156	0.9288	0.9385	0.9397	0.8110	0.8617	0.8833	0.8894	0.9038
	JSPSH-5	0.7887	0.8509	0.8758	0.8812	0.9154	0.4633	0.5962	0.7385	0.8011	0.8410
	JSPSH	**0.8861**	**0.9213**	**0.9348**	**0.9415**	**0.9423**	**0.8137**	**0.8627**	**0.8881**	0.8945	**0.9064**

time, and retrieval time of all methods on the MIRFlickr dataset. The results are presented in Table II. Regarding the training time, while the time complexity of most methods is $O(n)$, there are variations in the actual time required due to different coefficients such as c^2k and k^3 in time complexity. Since they are significantly smaller than n, they are disregarded when calculating the time complexity. Generally, the training time of JSPSH is comparable to other methods. We believe that a slight increase in training time is a reasonable trade-off considering the significant improvement in retrieval performance offered by JSPSH. As for the retrieval time, all methods achieve similar performance with the same hash code length. This indicates that sparse hash codes do not impose an additional computational burden during the retrieval phase.

D. Ablation Study

In JSPSH, we made three key contributions. First, we introduced the concept of cluster-wise semantic relationships and used it in conjunction with sample-wise semantic relationships to jointly supervise the learning of hash codes. Second, we replaced traditional dense hash codes with high-dimensional sparse hash codes, whose effectiveness has already been validated in Section IV-B. Third, we introduced an interaction term during the hash function learning process to narrow the heterogeneous gap. To validate the effectiveness of the first and third contributions, we conducted ablation experiments on five variants of JSPSH. Specifically, JSPSH-1 used only sample-wise semantic relations to train hash codes.

TABLE IV
THE DIFFERENCES BETWEEN VARIANTS OF JSPSH IN ABLATION STUDY

Methods	Semantic Relationships			Hash	Interaction term
	$\mathbf{S}_s$	$\mathbf{S}_c$	#cluster		
JSPSH-1	✓			sparse	✓
JSPSH-2	✓	✓	100	sparse	✓
JSPSH-3	✓	✓	500	sparse	✓
JSPSH-4	✓	✓	{100,200,500}	sparse	
JSPSH-5	✓	✓	{100,200,500}	dense	✓
JSPSH	✓	✓	{100,200,500}	sparse	✓

JSPSH-2 and JSPSH-3 used both semantic relations to jointly train hash codes, but only used $p = 100$ and $p = 500$ for the cluster-wise semantic relationship obtained from clustering results, respectively. JSPSH-4 used both semantic relations to jointly train hash codes and $\{p_i\} = \{100, 200, 500\}$, but removed the interaction term during hash functions learning stage. Finally, JSPSH-5 replaces the high-dimension sparse hash codes in JSPSH with dense hash codes, keeping other settings unchanged. The specific differences between all variants are summarized in Table IV. The results are reported in Table III.

By comparing JSPSH-1, JSPSH-2, JSPSH-3, and JSPSH, we can verify the role of the cluster-wise semantic relationship. The results lead to the following conclusions:

- The introduction of cluster-wise semantic information, irrespective of its level, proves beneficial to the final retrieval performance. In most cases, JSPSH-2, JSPSH-3, and JSPSH perform better than JSPSH-1, which only uses sample-wise semantic information to learn hash codes.

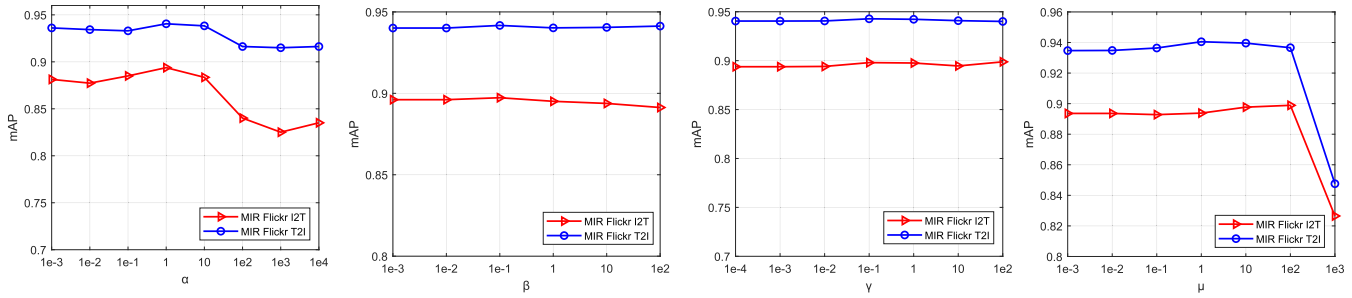


Fig. 7. Parameters analyses of JSPSH on MIRFlickr dataset.

- The cluster-wise semantic relationship required by different data sets varies. Specifically, for the MIRFlickr data set, the results obtained by JSPSH-2 ($p = 100$) and JSPSH-3 ($p = 500$) are comparable. However, for the IAPR TC-12 data and above, the results of JSPSH-3 are significantly better than those of JSPSH-2. Theoretically, the larger the value of p, the more accurate the cluster-wise semantic information is, which is more conducive to the learning of hash codes. However, the validity of this information also depends on the distribution of the label itself, which requires further investigation.
- The cluster-wise semantic relationship that is adapted to hash codes with different representation capabilities varies. When the representation ability of the hash code is limited, that is when r is small, too complex semantic information may not be beneficial to the learning of the hash code. For instance, when $r = 4$, the I2T results of JSPSH-2 on the MIRFlickr dataset are significantly higher than those of other variants. Conversely, when the hash code representation ability is adequate, that is when r is larger, more appropriate semantic information can stimulate its representation potential. For instance, when $r = 16$, the results of JSPSH-3 outperform all other variants on the IAPR TC-12 dataset.

Through the above analysis, it can be concluded that finding suitable cluster-wise semantic relations as supervisory information for different datasets is a challenging task. To address this issue, we adopt the strategy of weighted average, which helps to mitigate the different requirements to a certain extent. The results demonstrate that JSPSH performs better than JSPSH-2 and JSPSH-3 in most cases.

Furthermore, the effectiveness of the interaction term in the hash function learning phase can be demonstrated by comparing JSPSH-4 and JSPSH. It can be seen that the retrieval performance of JSPSH has always been better than that of JSPSH-4. This proves that the interaction term we proposed can effectively strengthen the interaction between modalities, further narrow the heterogeneous gap, and achieve better retrieval results. Besides, the performance of JSPSH significantly outperforms that of JSPSH-5, indicating that high-dimensional sparse hash codes possess a stronger representation capability compared to traditional dense hash codes, given the same number of hash bits.

E. Further Analyses

1) Parameter Sensitive: We conducted experiments on the MIRFlickr dataset to analyze the sensitivity of parameters α, β, γ, and μ. Parameter α adjusts the proportion of sample-wise and cluster-wise semantic relationships, while parameters β, γ, and μ are weights of three different auxiliary terms, namely quantization error term, error correction term, and interaction term. Figure 1 shows the corresponding mAP performance. Our observations are as follows:

- Parameter α: When a is small ($\alpha < 10$), its impact on the retrieval performance is relatively slight. However, when a is large ($\alpha > 10$), the retrieval performance drops significantly. This is because the cluster-wise semantic relationship should be an auxiliary to the sample-wise semantic relationship in JSPSH. When α is excessively large, the cluster-wise semantic relationship dominates, subverting the primary and secondary relationship, and leading to a decline in retrieval performance.
- Parameters β, γ, and μ: These parameters correspond to auxiliary terms and the performance of JSPSH is not so sensitive to them. Only when their values are too large, such as $\mu = 1000$, does the retrieval performance drop significantly.

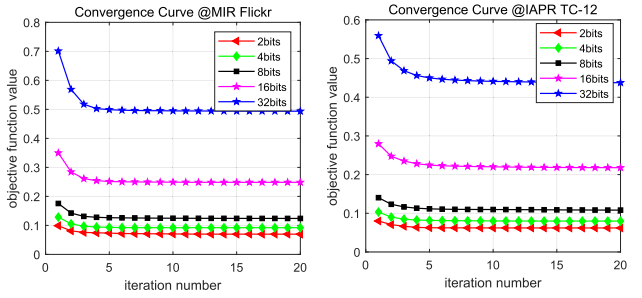


Fig. 8. The convergence curves on MIRFlickr and IAPR TC-12 datasets.

2) Convergence Analysis: In Section III-E, we provide a theoretical analysis of the convergence of JSPSH. To gain a deeper understanding, we conduct additional experiments on MIRFlickr and IAPR TC-12 datasets to further analyze the convergence empirically. Fig. 8 presents the convergence results, where we normalize the objective function value for ease of observation. It is worth noting that after a single iteration, we observe a sharp drop in the objective value and the model consistently converges after five iterations. These findings provide additional evidence of the efficient and effective convergence of our proposed model.

3) Comparison With Deep Hashing Methods: To further validate the efficacy of JSPSH, we conducted a comparison study with some state-of-the-art deep cross-modal hashing methods, including DCMH [43], SSAH [44], EDGH [45],

TABLE V
THE MAP RESULTS OF THE PROPOSED JSPSH AND OTHER DEEP BASELINES ON MIRFLICKR DATASET

Methods	I2T			T2I		
	4bits	16bits	64bits	4bits	16bits	64bits
DCMH	0.7021	0.7410	0.7485	0.7324	0.7827	0.7932
SSAH	0.7410	0.7820	0.8000	0.7424	0.7910	0.7950
EDGH	0.7104	0.7569	0.7959	0.7225	0.7787	0.7985
MLCAH	-	0.7960	0.8150	-	0.7940	0.8050
DADH	0.7521	0.8020	0.8179	0.7442	0.7920	0.8064
CPAH	-	0.7950	0.7960	-	0.7780	0.7850
MLSPH	0.7445	0.8076	0.8337	0.7335	0.7852	0.8146
DMFH	-	0.7802	0.7946	-	0.7978	0.8101
MDCH	-	0.8063	0.8232	-	0.8048	0.8337
JSPSH	**0.8498**	**0.8821**	**0.8909**	**0.8265**	**0.8567**	**0.8633**

MLCAH [69], DADH [70], CPAH [71], MLSPH [72], DMFH [51], and MDCH [73], on the MIRFlickr dataset. To ensure a fair comparison, same as [12], [29], and [30], we replaced the shallow features used in the prior experiment with 4096-dimensional CNN features that were extracted using the pre-trained CNN-F network [74] on ImageNet [75]. Table V represents the mAP results, and for all baselines, we directly report the results from the original papers. As demonstrated, JSPSH consistently outperforms all the baselines. A plausible reason may be that deep hashing methods tend to relax the discrete constraints of hash codes and optimize the objective function in batches. In contrast, JSPSH can effectively guarantee the quality of the hash codes by designing a discrete update algorithm and updating it in a global manner. Besides, when the dense hash code length is reduced, there is a notable decline in the performance of these deep hashing methods. Conversely, JSPSH still achieves stable performance under the same circumstances. Furthermore, even with a hash code length of 4 bits, JSPSH surpasses the majority of deep methods, highlighting the expressive capability of sparse hash codes.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel approach Joint Semantic Preserving Sparse Hashing (JSPSH) for cross-modal retrieval. It overcomes the limitations of existing methods that only consider sample-wise semantic relationships. We have proposed a new concept of cluster-wise semantic relationships that takes into account the distribution of labels to identify which samples should be closer to each other. By preserving both sample-wise and cluster-wise semantic relationships, JSPSH is able to learn more efficient hash codes. Additionally, to capture more precise semantic information, we have utilized high-dimensional sparse hash codes that are more expressive for multi-modal data representation than traditional dense hash codes. To further bridge the gap between heterogeneous modalities, we have proposed an interaction term during hash functions learning to align the hash codes of different modalities. The experimental results demonstrate that the proposed JSPSH outperforms existing state-of-the-art methods.

Although the effectiveness of the proposed cluster-wise semantic relationship has been demonstrated in improving retrieval performance, the k-means clustering algorithm used in this paper still has some limitations in capturing this relationship. Specifically, since the number of clusters for the labels is unknown, we adopt a compromise strategy that involves selecting different numbers of clusters and performing a weighted average on the results. However, as shown in Section IV-D, this strategy is not always the optimal solution. In future work, we plan to explore new methods to obtain more effective cluster-wise semantic information, thereby further improving retrieval performance.

REFERENCES

[1] J. C. Pereira et al., "On the role of correlation and abstraction in cross-modal multimedia retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 521–535, Mar. 2014.

[2] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 154–162.

[3] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 7–16.

[4] Y. Peng and J. Qi, "Reinforced cross-media correlation learning by context-aware bidirectional translation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1718–1731, Jun. 2020.

[5] X. Liu, J. Yi, Y.-M. Cheung, X. Xu, and Z. Cui, "OMGH: Online manifold-guided hashing for flexible cross-modal retrieval," *IEEE Trans. Multimedia*, early access, Apr. 12, 2022, doi: 10.1109/TMM.2022.3166668.

[6] X. Liu, X. Wang, and Y.-M. Cheung, "FDDH: Fast discriminative discrete hashing for large-scale cross-modal retrieval," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6306–6320, Nov. 2022.

[7] J. Yi, X. Liu, Y.-M. Cheung, X. Xu, W. Fan, and Y. He, "Efficient online label consistent hashing for large-scale cross-modal retrieval," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[8] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Commun. ACM*, vol. 51, no. 1, pp. 117–122, Jan. 2008.

[9] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approx. Reasoning*, vol. 50, no. 7, pp. 969–978, Jul. 2009.

[10] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2013, pp. 785–796.

[11] 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.

[12] Z.-D. Chen, C.-X. Li, X. Luo, L. Nie, W. Zhang, and X.-S. Xu, "SCRATCH: A scalable discrete matrix factorization hashing framework for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2262–2275, Jul. 2020.

[13] S. Su, Z. Zhong, and C. Zhang, "Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3027–3035.

[14] D. Yang, D. Wu, W. Zhang, H. Zhang, B. Li, and W. Wang, "Deep semantic-alignment hashing for unsupervised cross-modal retrieval," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2020, pp. 44–52.

[15] C. Sun, H. Latapie, G. Liu, and Y. Yan, "Deep normalized cross-modal hashing with bi-direction relation reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4937–4945.

[16] S. Liu, S. Qian, Y. Guan, J. Zhan, and L. Ying, "Joint-modal distribution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1379–1388.

[17] P. Hu, H. Zhu, J. Lin, D. Peng, Y.-P. Zhao, and X. Peng, "Unsupervised contrastive cross-modal hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3877–3889, Mar. 2023.

[18] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3864–3872.

[19] X. Nie, X. Liu, X. Xi, C. Li, and Y. Yin, "Fast unmediated hashing for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3669–3678, Sep. 2021.

[20] X. Li, J. Yu, Y. Wang, J.-Y. Chen, P.-X. Chang, and Z. Li, "DAHP: Deep attention-guided hashing with pairwise labels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 933–946, Mar. 2022.

[21] T. Li, X. Yang, B. Wang, C. Xi, H. Zheng, and X. Zhou, "Bi-CMR: Bidirectional reinforcement guided hashing for effective cross-modal retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 9, pp. 10275–10282.

[22] X. Liu, Z. Hu, H. Ling, and Y.-M. Cheung, "MTFH: A matrix tri-factorization hashing framework for efficient cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 964–981, Mar. 2021.

[23] 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.

[24] 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.

[25] Y. Wang, Z.-D. Chen, X. Luo, and X.-S. Xu, "High-dimensional sparse cross-modal hashing with fine-grained similarity embedding," in *Proc. Web Conf.*, Apr. 2021, pp. 2900–2909.

[26] X. Wang, X. Liu, S. Peng, Y.-M. Cheung, Z. Hu, and N. Wang, "Fast semantic preserving hashing for large-scale cross-modal retrieval," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 1348–1353.

[27] P.-F. Zhang, J. Duan, Z. Huang, and H. Yin, "Joint-teaching: Learning to refine knowledge for resource-constrained unsupervised cross-modal retrieval," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1517–1525.

[28] D. Zhang, X.-J. Wu, T. Xu, and J. Kittler, "WATCH: Two-stage discrete cross-media hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6461–6474, Jun. 2023.

[29] H. Li, C. Zhang, X. Jia, Y. Gao, and C. Chen, "Adaptive label correlation based asymmetric discrete hashing for cross-modal retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1185–1199, Feb. 2023.

[30] Y. Wang, Z.-D. Chen, X. Luo, and X.-S. Xu, "A high-dimensional sparse hashing framework for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 12, pp. 8822–8836, Dec. 2022.

[31] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2014, pp. 415–424.

[32] M. Long, Y. Cao, J. Wang, and P. S. Yu, "Composite correlation quantization for efficient multimodal retrieval," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2016, pp. 579–588.

[33] 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.

[34] M. Hu, Y. Yang, F. Shen, N. Xie, R. Hong, and H. T. Shen, "Collective reconstructive embeddings for cross-modal hashing," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2770–2784, Jun. 2019.

[35] 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.

[36] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2494–2507, May 2017.

[37] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 37–45.

[38] D. Wang, X. Gao, X. Wang, and L. He, "Label consistent matrix factorization hashing for large-scale cross-modal similarity search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2466–2479, Oct. 2019.

[39] W. Lan and L. Lan, "Compressing deep convolutional neural networks by stacking low-dimensional binary convolution filters," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 9, pp. 8235–8242.

[40] Q. Zhou et al., "Training-free transformer architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10884–10893.

[41] M. Li, Y.-M. Cheung, and Z. Hu, "Key point sensitive loss for long-tailed visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4812–4825, Apr. 2023.

[42] W. Cai, H. Zhang, X. Xu, S. He, K. Zhang, and J. Qin, "Contextual-assisted scratched photo restoration," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 13, 2023, doi: 10.1109/TCSVT.2023.3256372.

[43] 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.

[44] C. Li, C. Deng, N. Li, W. Liu, X. Gao, and D. Tao, "Self-supervised adversarial hashing networks for cross-modal retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4242–4251.

[45] Y. Shi, X. You, F. Zheng, S. Wang, and Q. Peng, "Equally-guided discriminative hashing for cross-modal retrieval," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4767–4773.

[46] W. Gu, X. Gu, J. Gu, B. Li, Z. Xiong, and W. Wang, "Adversary guided asymmetric hashing for cross-modal retrieval," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2019, pp. 159–167.

[47] Z. Hu, X. Liu, X. Wang, Y.-M. Cheung, N. Wang, and Y. Chen, "Triplet fusion network hashing for unpaired cross-modal retrieval," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2019, pp. 141–149.

[48] X. Liu, Y.-M. Cheung, Z. Hu, Y. He, and B. Zhong, "Adversarial tri-fusion hashing network for imbalanced cross-modal retrieval," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 4, pp. 607–619, Aug. 2021.

[49] H. Hu, L. Xie, R. Hong, and Q. Tian, "Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3120–3129.

[50] L. Zhu, H. Cui, Z. Cheng, J. Li, and Z. Zhang, "Dual-level semantic transfer deep hashing for efficient social image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1478–1489, Apr. 2021.

[51] X. Nie, B. Wang, J. Li, F. Hao, M. Jian, and Y. Yin, "Deep multiscale fusion hashing for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 401–410, Jan. 2021.

[52] E. Yang, D. Yao, T. Liu, and C. Deng, "Mutual quantization for cross-modal search with noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7541–7550.

[53] Z. Zhang, H. Luo, L. Zhu, G. Lu, and H. T. Shen, "Modality-invariant asymmetric networks for cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 5091–5104, May 2023.

[54] S. Dasgupta, C. F. Stevens, and S. Navlakha, "A neural algorithm for a fundamental computing problem," *Science*, vol. 358, no. 6364, pp. 793–796, Nov. 2017.

[55] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proc. 34th Annu. ACM Symp. Theory Comput.*, May 2002, pp. 380–388.

[56] C. Ryali, J. Hopfield, L. Grinberg, and D. Krotov, "Bio-inspired hashing for unsupervised similarity search," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8295–8306.

[57] W. Li, J. Mao, Y. Zhang, and S. Cui, "Fast similarity search via optimal sparse lifting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–9.

[58] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. AAAI*, 2014, vol. 28, no. 1, pp. 1–7.

[59] C. Da, S. Xu, K. Ding, G. Meng, S. Xiang, and C. Pan, "AMVH: Asymmetric multi-valued hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 898–906.

[60] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, "Discrete graph hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–13.

[61] Z.-D. Chen, Y. Wang, H.-Q. Li, X. Luo, L. Nie, and X.-S. Xu, "A two-step cross-modal hashing by exploiting label correlations and preserving similarity in both steps," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1694–1702.

[62] W. Rudin et al., *Principles of Mathematical Analysis*, vol. 3. New York, NY, USA: McGraw-Hill, 1976.

[63] M. J. Huiskes and M. S. Lew, "The MIR Flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retr.*, Oct. 2008, pp. 39–43.

[64] H. J. Escalante et al., "The segmented and annotated IAPR TC-12 benchmark," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 419–428, Apr. 2010.

[65] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr.*, Jul. 2009, pp. 1–9.

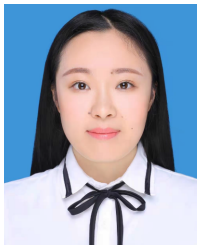
[66] Q.-Y. Jiang and W.-J. Li, "Discrete latent factor model for cross-modal hashing," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3490–3501, Jul. 2019.

[67] D. Zhang, X.-J. Wu, and J. Yu, "Label consistent flexible matrix factorization hashing for efficient cross-modal retrieval," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 3, pp. 1–18, Aug. 2021.
[68] Y. Wang, X. Luo, L. Nie, J. Song, W. Zhang, and X.-S. Xu, "BATCH: A scalable asymmetric discrete cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 11, pp. 3507–3519, Nov. 2021.
[69] X. Ma, T. Zhang, and C. Xu, "Multi-level correlation adversarial hashing for cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 22, no. 12, pp. 3101–3114, Dec. 2020.
[70] C. Bai, C. Zeng, Q. Ma, J. Zhang, and S. Chen, "Deep adversarial discrete hashing for cross-modal retrieval," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2020, pp. 525–531.
[71] D. Xie, C. Deng, C. Li, X. Liu, and D. Tao, "Multi-task consistency-preserving adversarial hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 29, pp. 3626–3637, 2020.
[72] X. Zou, X. Wang, E. M. Bakker, and S. Wu, "Multi-label semantics preserving based deep cross-modal hashing," *Signal Process., Image Commun.*, vol. 93, pp. 116–131, Apr. 2021.
[73] Q. Lin, W. Cao, Z. He, and Z. He, "Mask cross-modal hashing networks," *IEEE Trans. Multimedia*, vol. 23, pp. 550–558, 2021.
[74] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–12.
[75] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

Zhikai Hu received the B.S. degree in computer science from China Jiliang University, Hangzhou, China, in 2015, and the M.S. degree in computer science from Huaqiao University, Xiamen, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China, under the supervision of Prof. Yiu-Ming Cheung. His current research interests include multimedia information retrieval and data mining.

Yiu-Ming Cheung (Fellow, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. He is currently a Chair Professor (Artificial Intelligence) of the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China. His research interests include machine learning and visual computing, and their applications in data science, pattern recognition, multi-objective optimization, and information security. He is a fellow of AAAS, IET, BCS, and AAIA. He is the Awardee of RGC Senior Research Fellow. He is serving as the Editor-in-Chief for IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. Also, he is an Associate Editor of IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, *Pattern Recognition*, *Pattern Recognition Letters*, and *Neurocomputing*. For more information, visit the link: (https://www.comp.hkbu.edu.hk/∼ymc).

Mengke Li received the B.S. degree in communication engineering from Southwest University, Chongqing, China, in 2015, the M.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2018, and the Ph.D. degree from the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China, under the supervision of Prof. Yiu-Ming Cheung, in 2022. She is currently an Associate Researcher with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Guangdong, China. Her current research interests include image restoration and imbalanced data learning.



Weichao Lan (Graduate Student Member, IEEE) received the B.S. degree in electronics and information engineering from Sichuan University, Chengdu, China, in 2019. She is currently pursuing the Ph.D. degree with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China, under the supervision of Prof. Yiu-Ming Cheung. Her present research interests include network compression and acceleration, and lightweight models.

Donglin Zhang received the Ph.D. degree from Jiangnan University, Wuxi, China, in 2022. He is currently an Associate Professor with the School of Artificial Intelligence and Computer Science, Jiangnan University. He has published over ten scientific articles in refereed journals and conferences, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON CYBERNETICS, *ACM TOMM*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, and *PR*. His research interests include multimedia information processing and big data mining.

Qiang Liu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control theory and engineering from Northeastern University, Shenyang, China. He is currently a Full Professor with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. He was a Research Associate with the Department of Chemical Engineering, University of Southern California, Los Angeles, CA, USA from September 2014 to October 2016. His research interests include big data analytics, machine learning, statistical process monitoring, and fault diagnosis. He has published more than 70 peer-reviewed papers. He is the principal investigator of two key projects supported by the Natural Science Foundation of China, and the National Key Research and Development Program of China. He was a recipient of the Outstanding Young Scholar of Liaoning Revitalization Talents Program, China. His article titled "Perspectives on Big Data Modeling of Process Industries" was selected as one of the F5000-Top academic articles in Chinese top-quality SCI tech journals in 2019. He is an Editor/a Guest Editor of a few international journals, including an Associate Editor of *Intelligence and Robotics* and the Head Guest Editor of a Special Issue on "Advanced Intelligent Manufacturing System: Theory, Algorithms, and Industrial Applications" in the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.