Asymmetric Supervised Fusion-Oriented Hashing for Cross-Modal Retrieval

Zhan Yang, Xiyin Deng, Lin Guo, and Jun Long, *Member, IEEE*

***Abstract*—Hashing technologies have been widely applied for large-scale multimodal retrieval tasks owing to their excellent performance in search and storage tasks. Although some effective hashing methods have been proposed, it is still difficult to handle the intrinsic linkages that exist among different heterogeneous modalities. Moreover, optimizing the discrete constraint problem through a relaxation-based strategy results in a large quantization error and leads to a suboptimal solution. In this article, we present a novel asymmetric supervised fusion-oriented hashing method, named (ASFOH), which investigates three novel schemes to remedy the above issues. Specifically, we first explicitly formulate the problem as matrix decomposition into a common latent representation and a transformation matrix, combined with an adaptive weight scheme and nuclear norm minimization to ensure the information completeness of multimodal data. Then, we associate the common latent representation with the semantic label matrix, thereby increasing the discriminative capability of the model by constructing an asymmetric hash learning framework, thus, making the generated hash codes more compact. Finally, an efficient discrete optimization iterative algorithm based on nuclear norm minimization is proposed to decompose the nonconvex multivariate optimization problem into several subproblems with analytical solutions. Comprehensive experiments on the MIRFlirck, NUS-WIDE, and IARP-TC12 datasets testify that ASFOH outperforms the compared state-of-the-art approaches.**

***Index Terms*—Cross-modal retrieval, discrete optimization, hash.**

I. Introduction

MANY real-world information resources consist of data from different modalities. For example, social networking sites commonly consist of text and image data; movies usually contain texts, images, and demo video information. Therefore, effective data retrieval methods that can efficiently handle multimodal data acquisition are required. For such situations, cross-modal retrieval technologies, which aim to find the semantically relevant data in a modality given a query in a different modality, have in recent years become a hot research topic. However, due to the explosive growth of multimedia data, it is still an open research problem to search for the resources quickly and accurately from the huge amount of heterogeneous data. To address this issue, hashing technology shows great value, which can encode the real-valued high-dimensional feature spaces to Hamming space embeddings to greatly reduce the time cost and storage consumption, while preserving the important semantic properties. As a consequence, hashing technologies have been applied in many practical large-scale similarity retrieval task.

Earlier hashing methods usually focused on unimodal data, the key problem with these methods is the need to bridge the intramodal semantic gap, that is, finding a common Hamming space in which to represent the semantic labels of different data. However, in addition to considering the intramodal semantic gap, the cross-modal hashing methods also need to bridge the intermodal heterogeneous gap. Therefore, during the last decade, a large number of hashing methods have devoted their efforts to multimodal retrieval. Cross-modal hashing methods are mainly divided into two categories in terms of label use, that is, supervised and unsupervised hashing methods. For the unsupervised cross-modal method, it learns hash functions without the supervision information, and aims to maintain the latent distance relations by analyzing the feature distribution and topological structures. However, for the supervised cross-modal method, it enhances the binary code generation with the supervision information and exploits the linkages between different modalities to boost the quality of the learned hash codes. Representative works include CMFH [1], SMFH [2], DCMH [3], DBRC [4], and AAH [5]. In general, supervised cross-mode hashing methods exhibit higher performance under the supervision of label information compared to unsupervised methods.

Moreover, thanks to the success of deep neural networks (DNNs) in representation learning, deep cross-modal hashing methods have attracted more and more attention from researchers [6], [7], [8], [9], [10]. Although deep cross-modal hashing methods can achieve high performance for multimedia similarity search, they all require a large training dataset and time to train even with high-performance equipment (e.g., GPUs or TPUs). Therefore, the goal of this article is to propose a fast cross-modal hashing method without relying on expensive hardware devices, which is still an open problem when faced with large-scale heterogeneous data scenarios.

Manuscript received 8 April 2022; revised 6 May 2022 and 29 June 2022; accepted 27 January 2023. Date of publication 14 February 2023; date of current version 17 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62202501 and Grant 62102456; in part by the Science and Technology Plan of Hunan Province under Grant 2022JJ40638; and in part by the National Key Research and Development Program of China under Grant 2021YFB3900902. This article was recommended by Associate Editor F. Wu. *(Corresponding author: Lin Guo.)*

Zhan Yang, Lin Guo, and Jun Long are with the Big Data Institute, School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: zyang22@csu.edu.cn; guolincsu@csu.edu.cn; junlong@csu.edu.cn).

Xiyin Deng is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: xydeng98@csu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2023.3241018.

Digital Object Identifier 10.1109/TCYB.2023.3241018

Despite the promising performance of the existing supervised cross-modal hashing methods, there are still several challenges that need to be addressed.

1) The existing cross-modal hashing methods lack a dynamic information fusion and processing mechanism, which leads to insufficient information interaction and merging.
2) It is difficult to find the most suitable geometric structure for most cross-modal hashing methods. The issue may lead to an overlap between the different modalities, which affects the information completeness of the common latent representation.
3) Existing discrete cross-modal hashing methods employ only binary constraint, which will lead to the problem of weak robustness of the generated binary codes.
4) Due to optimization difficulties caused by the discrete constraints, some hashing methods approximate discrete constraints through a relaxation-based scheme, resulting in a large quantization loss.

To address the above challenges, in this article, a novel joint learning framework for both heterogeneous data fusion and hash learning is proposed. The former's task aims to decompose different modalities into a matrix of a common representation by matrix factorization and respective representative coefficient matrices to guarantee the information completeness of unified characterization for multimodal data, which is important for hash code generation. The aim of the latter is to generate high-quality hash codes discretely by considering the bit balance and bit uncorrelated constraints by uniting the common latent representation, which is important in boosting the discriminative power of the learned hash codes. The major contributions of this article can be listed as follows.

1) A joint learning framework is proposed that incorporates different modalities into a comprehensive common latent representation to simultaneously learn adaptive fusion weights and perform subsequent hash learning.
2) A semantic asymmetric discrete hash learning framework is proposed, in which the binary similarity information and category labels are learned jointly to make full use of the supervision information, and both the bit balance and bit uncorrelated constraints for discrete hash codes are taken into consideration, thereby boosting the discriminative power of the generated hash codes.
3) An efficient discrete optimization strategy is presented to directly solve the binary constraint by deriving the respective closed-form solutions.
4) Experiments demonstrate that asymmetric supervised fusion-oriented hashing (ASFOH) makes considerable improvements over the state-of-the-art baselines.

The remainder of this article is organized as follows. We start by reviewing the related works on some representative hashing methods, then move on to introducing our **ASFOH** method in Section III. In Section IV, we compare **ASFOH** with selected state-of-the-art approaches and conduct ablation experiments to investigate **ASFOH**. Finally, the conclusions are reached in Section V.

II. Related Work

A. Unsupervised Cross-Modal Hashing

Unsupervised hashing methods [11], [12], [13], [14], [15], [16] learn hash functions without semantic label information and handle intramodality and intermodality linkages pertaining to unlabeled heterogeneous data. For example, IMH [17] learns hash codes to consistently connect and represent heterogeneous data by considering both intermodality and intramodality consistency. However, it has to construct the pairwise similarity for all instances, which results in a quadratic time complexity to the size of the database. LSSH [18] is a two-step unsupervised hashing method. In the first stage, the representations of texts and images are jointly learned with sparse coding, and in the second stage, hash codes are generated by using matrix factorization. UGACH [19] uses a GAN [20] to learn the latent manifold structure of different modalities, and then proposes an unsupervised graph to process the learned manifold structure to generate the hash codes. FSH [21] handles the heterogeneous correlation among heterogeneous modalities and encodes the similarity information based on a graph fusion scheme into Hamming space to generate the hash codes. CMFH [1] utilizes a collective matrix factorization algorithm to generate binary codes on each modality. UCH [22] constructs two mutually optimized loop structures. The outer-loop framework is utilized to learn a common latent representation, and the inner-loop framework is utilized to learn binary codes. AUCH [23] replaces the hidden layer of an autoencoder with a hash layer and uses it to generate the hash codes by minimizing unsupervised classification and quantization losses. SCADH [24] is an unsupervised deep hash learning method that improves the retrieval performance of generated hash codes by jointly learning image representations and hash functions through deep convolutional networks and simultaneously extracting semantic information from noisy labels.

B. Supervised Cross-Modal Hashing

Although unsupervised cross-modal hashing techniques typically achieve acceptable performance, there is still a relatively large performance gap when compared to supervised ones. Supervised cross-modal hashing methods [25], [26], [27], [28], [29], [30], [31], [32], [33] can make full use of semantic supervision information to enhance the correlations from different heterogeneous modalities, that is, reduce the semantic and modality gap. For example, SMFH [2] preserves the local geometric consistency of different modalities through graph regularization and generates hash codes through collective matrix factorization operations while considering semantic consistency. SCM [34] generates hash codes by calculating pairwise similarity based on the reconstruction of semantic tag information. SePH [35] converts the semantic affinity information of the training dataset into the probability of binary code generation by minimizing Kullback–Leibler divergence in the Hamming space. NSDH [36] proposes a deep nonlinear descriptor to gradually exploit the hierarchical heterogeneous information of different modalities to generate

hash codes. DLFH [30] proposes an efficient discrete latent factor model to handle the semantics into the learned binary codes. ADCH [37] learns the common latent representation to generate hash codes by preserving the semantic similarity and cross-relationship from different modalities. OMH-DQ [38] is an online hash learning method that supports dynamic query adaptation. It improves the search performance of hash codes by proposing a self-weighted fusion strategy that exploits intermodal feature complementarity to improve the information completeness of a unified representation of multimodal data. NRDH [39] leverages nonlinear descriptors instead of kernelization methods to boost the discriminative capability of the generated binary codes. FGCMH [40] preserves the modality-fusion and modality-individual structures through graph convolutional neural networks for efficient retrieval on both complete and incomplete multimedia datasets. SRLCH [41] exploits relations of semantic supervision in semantic space to learn compact hash codes. SCRATCH [42] exploits the original data features and labels to learn hash codes. BATCH [43] fully mines semantic information by combining semantic mapping learning and distance–distance difference minimization strategies to generate more compact hash codes. SIDMH [44] develops a deep model-based multimodal hash learning architecture to generate semantic-driven interpretable binary codes. ASCSH [45] decomposes the original data features into a modality-specific matrix and a consistent one to handle the intrinsic relations among different modalities. Compared to the above methods, in this article, a joint framework is designed by considering multimodality data fusion and hash code learning. We restrict the common latent representation by imposing orthogonality and constrain the transformer matrices by nuclear norm minimization in this framework.

Recently, some deep hashing methods (e.g., DCMH [3], SSAH [46], SDAH [47], RDCMH [48], and DMFH [49]) have been proposed. Although DNNs have demonstrated significant retrieval performance due to their strong representative capabilities, most of them still have to deal with a time-consuming training process and difficulties in optimizing complex objective functions. Therefore, in this article, we focus on the design of an interpretable objective function, an effective learning framework, and an efficient discrete optimization strategy, which are crucial for learning discriminative compact binary codes.

III. Proposed Method

A. Notations

In this article, we use bold lowercase characters and bold uppercase characters to denote vectors and matrices, respectively. $\mathbf{A}^\top$ indicates the transpose of matrix $\mathbf{A}$, $\mathrm{tr}(\mathbf{A})$ denotes the trace of $\mathbf{A}$ if $\mathbf{A}$ is square, and $\|\mathbf{A}\|$ represents the ℓ_F norm for a matrix or ℓ_2 norm for a vector.

For ease of presentation, two modalities are discussed: 1) image and 2) text. Suppose that $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_n\}$ is the dataset, which is composed of n training objects from V different modalities. The vth modality denotes $\mathbf{X}^v = [\mathbf{x}_1^v, \mathbf{x}_2^v, \ldots, \mathbf{x}_n^v]^\top \in \mathbb{R}^{n\times d_v}$, where d_v is the dimensionality of the vth modality, and $v = \{1, 2\}$ denote the image and text modalities, respectively. $\mathbf{Y} \in \mathbb{R}^{n\times c}$ represents the ground-truth label matrix, where c is the number of categories. The final purpose of **ASFOH** is to generate the binary code $\mathbf{B} \in \mathbb{R}^{n\times k}$ to represent multimodal data, where k is the code length.

B. Kernelization

Due to the fact that semantic linkages among different modalities are complex, it is difficult to handle them by using linear projections. The kernel trick, which can be used to preserve the latent nonlinear relations of heterogeneous data, has been widely used in the cross-modal retrieval field [50]. In this article, we utilize the RBF kernel to project heterogeneous data to a nonlinear space. Specifically, for each instance of the vth modality $\mathbf{x}_i^v$, the kernelized feature can be expressed as

$$\phi\left(\mathbf{x}_i^v\right) = \left[\exp\left(-\frac{\left\|\mathbf{x}_i^v - \mathbf{a}_1^v\right\|^2}{2\sigma_v^2}\right), \ldots, \exp\left(-\frac{\left\|\mathbf{x}_i^v - \mathbf{a}_q^v\right\|^2}{2\sigma_v^2}\right)\right] \tag{1}$$

where $\sigma_v = (1/qn)\sum_{i=1}^{n}\sum_{j=1}^{q}\|\mathbf{x}_i^v - \mathbf{a}_j^v\|$ is the kernel width and $\{\mathbf{a}_1^v, \mathbf{a}_2^v, \ldots, \mathbf{a}_q^v\}$ are q anchor instances that are randomly chosen from training data in the vth modality.

C. Problem Formulation

The main idea of the proposed **ASFOH** is to answer the following questions.

1) How can the intrinsic linkages among the different modalities be fully explored and balanced?
2) How can an efficient asymmetric hash learning structure be constructed by using semantic information?
3) How can the quality of the generated hash codes be boosted?

1) RQ1 (Common Latent Representation Completeness Learning): According to the assumption [39], common information from different modalities originates from a common latent representation. For an input of multimodal data $\mathcal{M} = \{\phi(\mathbf{X}^v)\}_{v=1}^V$ with the modality v and $\phi(\mathbf{X}^v) \in \mathbb{R}^{n\times q}$, the aim of *common latent representation completeness learning* of **ASFOH** is to fuse multimodal features into a unified representation to learn a robust embedding for the hash learning task. Robustly encoding information from different modalities is the key to affecting the quality of the generated hash codes. Consequently, the latent representation can be learned by optimizing the following objective function:

$$\min_{\{\mathbf{P}^v\}_{v=1}^V, \mathbf{H}} \sum_{v=1}^{V} \ell\left(\phi\left(\mathbf{X}^v\right), \mathbf{H}\mathbf{P}^v\right) + \mathcal{R}\left(\mathbf{H}, \mathbf{P}^v\right) \tag{2}$$

where $\mathbf{P}^v \in \mathbb{R}^{k\times q}$ is the transformer matrix for the vth modality and $\mathbf{H} \in \mathbb{R}^{n\times k}$ is a common latent representation, $\ell(\mathbf{A}, \mathbf{B})$ denotes the reconstruction loss function, $\mathcal{R}(\mathbf{H}, \mathbf{P}^v)$ are respective constraints for the variables $\mathbf{H}, \mathbf{P}^v$. Note that, to encode the information from different modalities into a common latent representation, there is an opposite learning manner [45], that is, $\ell(\phi(\mathbf{X}^v)\mathbf{P}^v, \mathbf{H})$. However, it means that the common latent representation could be encoded from each individual modality, resulting in a weak information completeness of

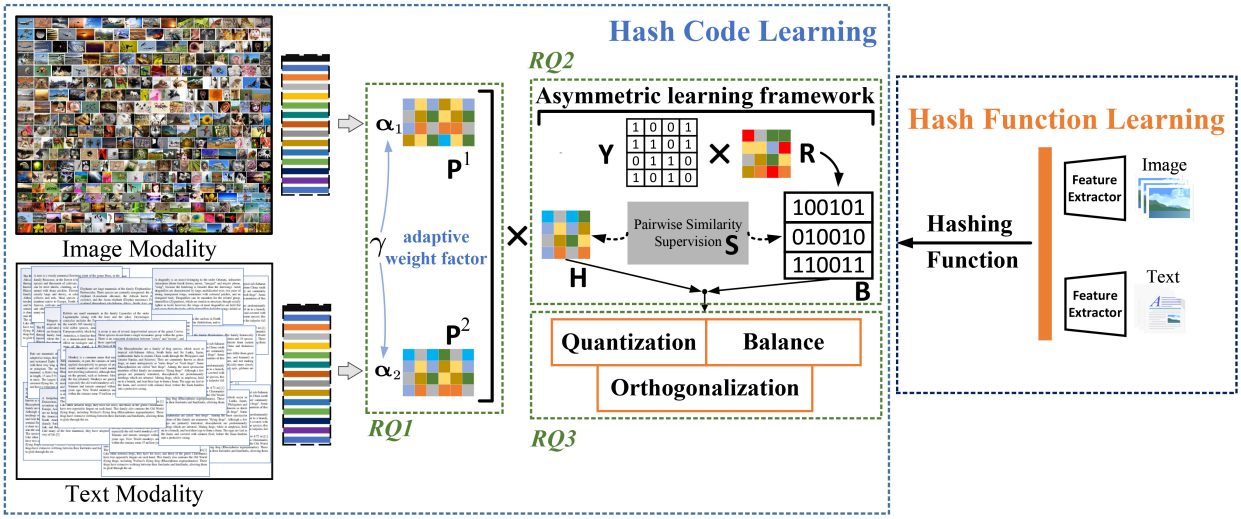


Fig. 1. Framework for the proposed **ASFOH**. In the first step, multimodal data are projected by a common latent representation and respective representative coefficient matrices and the compact codes are generated through the semantic supervisions, in which the common latent representation and asymmetric learning framework can be jointly learned to improve the interpretability of the model. In the second step, the hash functions can be flexibly generated through the learned hash codes in the first step.

the learned common latent representation. Moreover, different modalities describe different perspectives of the common latent representation, thus, it is inappropriate to treat all modalities with the same contribution. RFDH [51] first introduces the adaptive weight factor used in latent representation learning. However, RFDH ignores semantic label information, leading to a large semantic gap. Inspired by [51], we introduce an adaptive weight factor α_v for the vth modality to address the limitation, where $\sum_{v=1}^{V}\alpha_v = 1$. In some cases, $\alpha_v = 0$ means that the contribution of the vth modality is ignored, and $\alpha_v = 1$ means that the term $\ell(\phi(\mathbf{X}^v), \mathbf{H}\mathbf{P}^v)$ needs to be minimized. The above-mentioned adaptive-weighted scheme will lead to the modality imbalance problem, that is, some modalities may be ignored in favor of the modality with minimum reconstruction loss. Thus, we introduce a predefined constant $\gamma \geq 2$ to smoothen the contributions of different modalities. Therefore, the problem (2) is further represented as

$$\min_{\{\alpha_v,\mathbf{P}^v\}_{v=1}^{V},\mathbf{H}} \sum_{v=1}^{V} \alpha_v^{\gamma}\ell\big(\phi\big(\mathbf{X}^v\big), \mathbf{H}\mathbf{P}^v\big) + \mathcal{R}\big(\mathbf{H}, \mathbf{P}^v\big). \tag{3}$$

Next, we deal with the term constraints for the variables $\mathbf{P}^v, \mathbf{H}$, that is, $\mathcal{R}(\mathbf{P}^v, \mathbf{H})$. We add some regularizations to penalize the transformer matrix $\mathbf{P}^v$ and common latent representation $\mathbf{H}$. Inspired by [52], the nuclear norm is utilized by the transformer matrix $\mathbf{P}^v$ to capture the principal components of different modalities, which can make the model more accurate to find a suitable geometric structure of the common latent representation. Therefore, the problem (3) can be rewritten as

$$\begin{aligned}\min_{\{\alpha_v,\mathbf{P}^v\}_{v=1}^{V},\mathbf{H}} & \sum_{v=1}^{V}\Big(\alpha_v^{\gamma}\big\|\phi\big(\mathbf{X}^v\big) - \mathbf{H}\mathbf{P}^v\big\|^2 + \beta\big\|\mathbf{P}^v\big\|_*\Big)\\ \text{s.t.} & \sum_{v=1}^{V}\alpha_v = 1, \alpha_v \geq 0.\end{aligned} \tag{4}$$

2) RQ2 (Asymmetric Semantic Hash Learning): KSH [53] is a popular hashing method based on the symmetric learning structure, which preserves the linkages of pairs with two identical binary representations to approximate the binary similarity matrix. The definition is as follows:

$$\min_{\mathbf{B}}\big\|\mathbf{B}\mathbf{B}^{\top} - k\mathbf{S}\big\|^2 \qquad \text{s.t. } \mathbf{B} \in \{-1, 1\}^{n\times k} \tag{5}$$

where $\mathbf{S}$ is a pairwise similarity. However, (5) is a quartic optimization problem that is difficult to solve. Fortunately, some works [47], [54] have demonstrated that using an asymmetric learning structure has obvious advantages over symmetric learning structures in terms of speed and accuracy. Moreover, using real-valued matrices instead of binary matrices will result in obtaining more accurate approximations [55], [56]. Motivated by the above analysis, we project the semantic label matrix into a binary space, and some works [57], [58] have shown that embedding the semantic space into the binary space can significantly boost the quality of the generated hash codes.

Inspired by the above discussion, the semantic matrix $\mathbf{Y}$ should be associated with the binary codes $\mathbf{B}$ via a linear projection. Specifically, we introduce a projection matrix $\mathbf{R} \in \mathbb{R}^{c\times k}$, where $\mathbf{r}_i \in \mathbb{R}^{c\times 1}$ in $\mathbf{R}$ is the projection vector for the ith hash bit. In order to solve the quantization gap between the real-valued projected semantic embedding and the learned hash codes, we need to minimize the following formulation:

$$\min_{\mathbf{R}}\|\mathbf{Y}\mathbf{R} - \mathbf{B}\|^2 \qquad \text{s.t. } \mathbf{B} \in \{-1, 1\}^{n\times k}. \tag{6}$$

Then, we rewrite (5) as the following formulation:

$$\begin{aligned}\min_{\mathbf{R},\mathbf{B}} \ & \big\|\mathbf{Y}\mathbf{R}\mathbf{B}^{\top} - k\mathbf{S}\big\|^2 + \eta\|\mathbf{Y}\mathbf{R} - \mathbf{B}\|^2 + \rho\mathcal{R}(\mathbf{Y}\mathbf{R})\\ \text{s.t. } & \mathbf{B} \in \{-1, 1\}^{n\times k}\end{aligned} \tag{7}$$

where η and ρ are balance parameters and $\mathcal{R}(\cdot)$ is a regularization term, the pairwise similarity matrix $\mathbf{S}$ is constructed by the following two steps: 1) we first normalize each row vector of $\mathbf{Y}$ with its ℓ_2-norm and obtain $\tilde{\mathbf{Y}}$ and 2) then the similarity matrix $\mathbf{S}$ can be generated by $\mathbf{S} = 2\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^{\top} - \mathbf{1}\mathbf{1}^{\top}$. It is obvious that both label information $\mathbf{Y}$ and pairwise similarity matrix $\mathbf{S}$

are jointly learned into a unified framework, which enhances the semantics of the hash code generation.

3) RQ3 (Balance and Decorrelation): As mentioned above, the learned common latent representation $\mathbf{H}$ contains the complementarity among different modalities. Therefore, we need to bridge the learned hash codes and the real-valued representation. Specifically, we bridge the quantization gap between $\mathbf{H}$ and $\mathbf{B}$

$$\min_{\mathbf{H},\mathbf{B}} \|\mathbf{H}-\mathbf{B}\|^2 \qquad \text{s.t. } \mathbf{B}\in\{-1,1\}^{n\times k}. \tag{8}$$

Furthermore, some researchers [59] have shown that high-quality hash codes need to satisfy two constraints: 1) bit balance and 2) bit uncorrelated constraints. Motivated by [59], we add the two constraints to the learned binary codes. Therefore, (8) can be reformulated as follows:

$$\min_{\mathbf{H},\mathbf{B}} \|\mathbf{H}-\mathbf{B}\|^2 \text{ s.t. } \mathbf{B}\in\{-1,1\}^{n\times k}\mathbf{B}^\top\mathbf{B}=n\mathbf{I}_k, \mathbf{B}^\top\mathbf{1}_n=\mathbf{0}_k \tag{9}$$

where $\mathbf{1}_n$ and $\mathbf{0}_k$ are two vectors of size n with all-ones elements and size k with all-zeros elements. However, (9) is an NP-hard problem due to the binary constraint, that is, $\mathbf{B}\in\{-1,1\}^{n\times k}$ and the orthogonal constraints, that is, $\mathbf{B}^\top\mathbf{B}=n\mathbf{I}_k, \mathbf{B}^\top\mathbf{1}_n=\mathbf{0}_k$. To remedy the problem, we relax the orthogonal constraints by transferring them to $\mathbf{H}$, and obtain

$$\min_{\mathbf{H},\mathbf{B}} \|\mathbf{H}-\mathbf{B}\|^2 \text{ s.t. } \mathbf{B}\in\{-1,1\}^{n\times k}\mathbf{H}^\top\mathbf{H}=n\mathbf{I}_k, \mathbf{H}^\top\mathbf{1}_n=\mathbf{0}_k. \tag{10}$$

Combining (4), (7), and (10), we obtain the overall objective function of **ASFOH**

$$\begin{aligned}\min_{\mathbf{H},\mathbf{R},\mathbf{B},\{\alpha_v,\mathbf{P}^v\}_{v=1}^V} & \sum_{v=1}^V\left(\alpha_v^\gamma\left\|\phi(\mathbf{X}^v)-\mathbf{H}\mathbf{P}^v\right\|^2+\beta\left\|\mathbf{P}^v\right\|_*\right)\\ &+\left\|\mathbf{Y}\mathbf{R}\mathbf{H}^\top-k\mathbf{S}\right\|^2+\eta\|\mathbf{Y}\mathbf{R}-\mathbf{B}\|^2+\omega\|\mathbf{H}-\mathbf{B}\|^2\\ &+\rho\|\mathbf{Y}\mathbf{R}\|^2\\ \text{s.t. } & \mathbf{B}\in\{-1,1\}^{n\times k}, \sum_{v=1}^V\alpha_v=1,\alpha_v\geq 0\\ & \mathbf{H}^\top\mathbf{H}=n\mathbf{I}_k, \mathbf{H}^\top\mathbf{1}_n=\mathbf{0}_k,\end{aligned} \tag{11}$$

where ω is the tradeoff parameter.

D. Efficient Discrete Optimization

However, the objective function, that is, (11), contains five variables and, therefore, is nonconvex and hard to optimize. To solve this problem, we adopt the alternating direction minimization (ADM) [60] algorithm to make the objective function separable. Specifically, we replace $\mathbf{P}^v$ with an auxiliary variables $\mathbf{J}^v$, and then obtain the following formulation:

$$\begin{aligned}\min_{\mathbf{H},\mathbf{R},\mathbf{B},\{\alpha_v,\mathbf{P}^v,\mathbf{J}^v\}_{v=1}^V} & \sum_{v=1}^V\left(\alpha_v^\gamma\left\|\phi(\mathbf{X}^v)-\mathbf{H}\mathbf{P}^v\right\|^2+\beta\left\|\mathbf{J}^v\right\|_*\right)\\ &+\left\|\mathbf{Y}\mathbf{R}\mathbf{H}^\top-k\mathbf{S}\right\|^2+\eta\|\mathbf{Y}\mathbf{R}-\mathbf{B}\|^2+\omega\|\mathbf{H}-\mathbf{B}\|^2\\ &+\rho\|\mathbf{Y}\mathbf{R}\|^2+\delta\sum_{v=1}^V\left\|\mathbf{P}^v-\mathbf{J}^v\right\|^2+\sum_{v=1}^V\left\langle\Theta^v,\mathbf{P}^v-\mathbf{J}^v\right\rangle\\ \text{s.t. } & \mathbf{B}\in\{-1,1\}^{n\times k}, \sum_{v=1}^V\alpha_v=1,\alpha_v\geq 0\\ & \mathbf{H}^\top\mathbf{H}=n\mathbf{I}_k, \mathbf{H}^\top\mathbf{1}_n=\mathbf{0}_k\end{aligned} \tag{12}$$

where δ is the penalty parameter, $\Theta^v\in\mathbb{R}^{q\times k}$ is the Lagrange multipliers, and $<\cdot,\cdot>$ is the Frobenius inner product defined as $<\Theta^v,\mathbf{P}^v-\mathbf{J}^v>=tr(\Theta^{v\top}(\mathbf{P}^v-\mathbf{J}^v))$.

The problem (12) can be solved by the following iterative optimization algorithms.

1) α_v-Subproblem: For updating α_v, we solve the following problem by fixing the other variables:

$$\min_{\{\alpha_v\}_{v=1}^V}\sum_{v=1}^V\alpha_v^\gamma\left\|\phi(\mathbf{X}^v)-\mathbf{H}\mathbf{P}^v\right\|^2 \text{ s.t. } \sum_{v=1}^V\alpha_v=1,\alpha_v\geq 0. \tag{13}$$

We construct a Lagrangian formulation by using the Lagrange multiplier method, then (13) can be rewritten as

$$\min_{\{\alpha_v\}_{v=1}^V,\nu}\sum_{v=1}^V\alpha_v^\gamma\left\|\phi(\mathbf{X}^v)-\mathbf{H}\mathbf{P}^v\right\|^2-\nu\left(\mathbf{1}^\top\alpha-1\right) \tag{14}$$

where $\alpha=[\alpha_1,\ldots,\alpha_V]^\top\in\mathbb{R}^V$ is the vector of weights for the corresponding modalities.

Setting the derivative with respect to α_v and ν to 0, we obtain

$$\alpha_v=\frac{{\Delta_v}^{1/1-\gamma}}{\sum_{v=1}^V{\Delta_v}^{1/1-\gamma}} \tag{15}$$

where $\Delta_v=\|\phi(\mathbf{X}^v)-\mathbf{H}\mathbf{P}^v\|^2$.

2) $\mathbf{R}$-Subproblem: For updating $\mathbf{R}$, we solve the following problem by fixing the other variables:

$$\min_{\mathbf{R}}\left\|\mathbf{Y}\mathbf{R}\mathbf{H}^\top-k\mathbf{S}\right\|^2+\eta\|\mathbf{Y}\mathbf{R}-\mathbf{B}\|^2+\rho\|\mathbf{Y}\mathbf{R}\|^2. \tag{16}$$

Taking the derivative with respect to $\mathbf{R}$ to 0, we have

$$\mathbf{R}=\left(\mathbf{Y}^\top\mathbf{Y}\right)^{-1}\left(k\mathbf{Y}^\top\mathbf{S}\mathbf{H}+\eta\mathbf{Y}^\top\mathbf{B}\right)\left(\mathbf{H}^\top\mathbf{H}+(\eta+\rho)\mathbf{I}\right)^{-1}. \tag{17}$$

Note that the terms $\mathbf{Y}^\top\mathbf{Y}$ and $\mathbf{Y}^\top\mathbf{S}$ can be considered as fixed variables computed only once outside of the iterations. Therefore, we set $\mathbf{C}=\mathbf{Y}^\top\mathbf{Y}\in\mathbb{R}^{c\times c}$ and $\mathbf{D}=\mathbf{Y}^\top\mathbf{S}\in\mathbb{R}^{c\times n}$. Then, (17) can be rewritten as

$$\mathbf{R}=\mathbf{C}^{-1}\left(k\mathbf{D}\mathbf{H}+\eta\mathbf{Y}^\top\mathbf{B}\right)\left(\mathbf{H}^\top\mathbf{H}+(\eta+\rho)\mathbf{I}\right)^{-1}. \tag{18}$$

3) $\mathbf{H}$-Subproblem: For updating $\mathbf{H}$, we solve the following problem by fixing the other variables:

$$\begin{aligned}\min_{\mathbf{H}} & \sum_{v=1}^V\alpha_v^\gamma\left\|\phi(\mathbf{X}^v)-\mathbf{H}\mathbf{P}^v\right\|^2+\left\|\mathbf{Y}\mathbf{R}\mathbf{H}^\top-k\mathbf{S}\right\|^2+\omega\|\mathbf{H}-\mathbf{B}\|^2\\ \text{s.t. } & \mathbf{H}^\top\mathbf{H}=n\mathbf{I}_k, \mathbf{H}^\top\mathbf{1}_n=\mathbf{0}_k.\end{aligned} \tag{19}$$

We rewrite (19) as

$$\max_{\mathbf{H}}\operatorname{tr}\left(\mathbf{Q}\mathbf{H}^\top\right) \text{ s.t. } \mathbf{H}^\top\mathbf{H}=n\mathbf{I}_k, \mathbf{H}^\top\mathbf{1}_n=\mathbf{0}_k \tag{20}$$

where $\mathbf{Q}=\sum_{v=1}^V\alpha_v^{\gamma\phi(\mathbf{X}^v)\mathbf{P}^{v\top}}+k\mathbf{D}^\top\mathbf{R}+\omega\mathbf{B}$. We first denote $\Delta=\mathbf{I}_n-(1/n)\mathbf{1}_n\mathbf{1}_n{}^\top$. Then, the singular value decomposition (SVD) of $\mathbf{Q}\Delta\mathbf{Q}^\top$ can be formulated as

$$\mathbf{Q}\Delta\mathbf{Q}^\top=\left[\mathbf{V}\middle|\bar{\mathbf{V}}\right]\left[\begin{array}{c|c}\Lambda & \mathbf{0}\\ \hline \mathbf{0} & \mathbf{0}\end{array}\right]\left[\mathbf{V}\middle|\bar{\mathbf{V}}\right]^\top$$

where $\Lambda\in\mathbb{R}^{k^*\times k^*}$, $\mathbf{V}\in\mathbb{R}^{n\times k^*}$, k^* is the rank of $\mathbf{Q}\Delta\mathbf{Q}^\top$, and $\bar{\mathbf{V}}$ is the matrix of the remaining $k-k^*$ eigenvectors, corresponding to zero eigenvalue. Then, we obtain an orthogonal matrix $\tilde{\mathbf{V}}\in\mathbb{R}^{n\times(k-k^*)}$ by computing $\bar{\mathbf{V}}$ with the Gram–Schmidt process. Further denote $\mathbf{O}=\Delta\mathbf{Q}^\top\mathbf{V}\Lambda^{-1/2}\in\mathbb{R}^{k\times k^*}$,

and a random matrix $\tilde{\mathbf{O}} \in \mathbb{R}^{k\times(k-k^*)}$. If $k = k^*$, $\tilde{\mathbf{V}}$, $\tilde{\mathbf{V}}$, and $\tilde{\mathbf{O}}$ are empty. According to the proof in the work [61], we can get the optimal solution of $\mathbf{H}$ as

$$\mathbf{H} = \sqrt{n}\left[\mathbf{V}\middle|\tilde{\mathbf{V}}\right]\left[\mathbf{O}\middle|\tilde{\mathbf{O}}\right]^\top. \tag{21}$$

4) $\mathbf{B}$-Subproblem: For updating $\mathbf{B}$, we solve the following problem by fixing the other variables:

$$\min_{\mathbf{B}} \eta\|\mathbf{YR} - \mathbf{B}\|^2 + \omega\|\mathbf{H} - \mathbf{B}\|^2 \text{ s.t. } \mathbf{B} \in \{-1, 1\}^{n\times k}. \tag{22}$$

Equation (22) can be equivalently transformed into the following formula:

$$\max_{\mathbf{B}} tr(\eta\mathbf{YR} + \omega\mathbf{H})\mathbf{B}^\top). \tag{23}$$

As $\text{tr}(\mathbf{B}^\top\mathbf{B}) = nk$. By setting the derivative with respect to $\mathbf{B}$ to 0, we have

$$\mathbf{B} = \text{sgn}(\eta\mathbf{YR} + \omega\mathbf{H}) \tag{24}$$

where sgn() is the element-wise indicator operator.

5) $\mathbf{P}^v$-Subproblem: For updating $\mathbf{P}$, we solve the following problem by fixing the other variables:

$$\min_{\{\mathbf{P}^v\}_{v=1}^V} \sum_{v=1}^{V} \alpha_v^\gamma \left\|\phi\left(\mathbf{X}^v\right) - \mathbf{HP}^v\right\|^2 + \delta\sum_{v=1}^{V}\left\|\mathbf{P}^v - \mathbf{J}^v\right\|^2 + \sum_{v=1}^{V}\left\langle\Theta^v, \mathbf{P}^v - \mathbf{J}^v\right\rangle. \tag{25}$$

Unfolding the objective function (25), we obtain

$$\min_{\mathbf{P}^v} \alpha_v^\gamma\ tr\left(-2\phi\left(\mathbf{X}^v\right)\mathbf{P}^{v\top}\mathbf{H}^\top + \mathbf{HP}^v\mathbf{P}^{v\top}\mathbf{H}^\top\right) + \delta\ \text{tr}\left(\mathbf{P}^v\mathbf{P}^{v\top} - 2\mathbf{J}^v\mathbf{P}^{v\top}\right) + tr\left(\Theta^{v\top}\mathbf{P}^v\right) \tag{26}$$

By setting the derivative with respect to $\mathbf{P}^v$ to 0, we have

$$\mathbf{P}^v = \left(2\alpha_v^\gamma\mathbf{H}^\top\mathbf{H} + 2\delta\mathbf{I}\right)^{-1}\left(2\alpha_v^\gamma\mathbf{H}^\top\phi\left(\mathbf{X}^v\right) + 2\delta\mathbf{J}^v - \Theta^v\right). \tag{27}$$

6) $\mathbf{J}^v$ and Θ Subproblems: Update auxiliary variable $\mathbf{J}^v$ and Lagrange multiplier Θ^v

$$\mathbf{J}^v = \arg\min_{\mathbf{J}^v} \frac{\beta}{\delta}\left\|\mathbf{J}^v\right\|_* + \frac{1}{2}\left\|\mathbf{J}^v - \left(\mathbf{P}^v + \frac{\Theta^v}{\delta}\right)\right\|^2$$
$$\Theta^v = \Theta^v + \delta\left(\mathbf{P}^v - \mathbf{J}^v\right). \tag{28}$$

The variable $\mathbf{J}^v$ can be solved with the aid of the singular value thresholding (SVT) [62] algorithm.

E. Hash Function Learning

In the query phase, a new query $\mathbf{x}_b^v$ needs to be computed to obtain the binary code $\mathbf{b}_b^v$. Since **ASFOH** is a two-step hashing, the binary codes $\mathbf{B}$ learned in the first step need to be used as supervision information to learn the hash functions. Although the use of complex classification algorithms (such as SVM-kernel [63] and DNN [64]) as hash functions can improve retrieval performance, it results in additional time overhead. Therefore, to balance the retrieval speed and performance, in this article, we leverage linear mapping as the base function of the hash functions in order to verify the effectiveness of **ASFOH**. Specifically, given the kernelized training set $\phi(\mathbf{X}^v)$ and learned hash codes $\mathbf{B}$, the hash function can be learned as

$$\min_{\mathbf{W}^v}\left\|\mathbf{B} - \phi\left(\mathbf{X}^v\right)\mathbf{W}^v\right\|^2 + \xi\left\|\mathbf{W}^v\right\|^2 \tag{29}$$

Algorithm 1 ASFOH

Input: Training instances $\mathbf{X}^v$, label matrix $\mathbf{Y}$, parameter $k, \gamma, \beta, \eta, \rho, \omega, \xi$, maximum iteration number ϱ.
Output: Binary codes $\mathbf{B}$.
Procedure:
1. Construct $\phi(\mathbf{X}^v)$ with randomly selected q anchors;
2. Initialize $\alpha = [\frac{1}{V}, ..., \frac{1}{V}]^\top$;
3. Initialize $\mathbf{R}, \mathbf{P}^v, \mathbf{J}^v, \Theta^v$ randomly;
4. Initialize $\mathbf{B}, \mathbf{H}$ randomly with a standard normal distribution;
5. $\mathbf{S} = 2\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top - \mathbf{1}\mathbf{1}^\top$;
6. $\mathbf{C} = \mathbf{Y}^\top\mathbf{Y}$, $\mathbf{D} = \mathbf{Y}^\top\mathbf{S}$, $\delta = 10^{-3}$, $\max_\delta = 10^4$.
% Step 1: Hash code learning
7. **Repeat**
   α_v-Step: Update α_v via Eq. (15);
   $\mathbf{R}$-Step: Update $\mathbf{R}$ via Eq. (18);
   $\mathbf{H}$-Step: Update $\mathbf{H}$ via Eq. (21);
   $\mathbf{B}$-Step: Update $\mathbf{B}$ via Eq. (24);
   $\mathbf{P}^v$-Step: Update $\mathbf{P}^v$ via Eq. (27);
   Update Lagrange multipliers $\mathbf{J}^v$ and Θ^v using Eq. (28);
   Update the parameter δ by $\delta = \min(\max_\delta, \tau\delta)$;
   Until up to ϱ.
8. **End**.
% Step 2: Hash function learning
9. Learn the hash mapping matrix $\mathbf{W}^v$ via Eq. (30);
Return Hash function $sgn(\phi(\mathbf{X}^v)\mathbf{W}^v)$.

where $\mathbf{W}^v \in \mathbb{R}^{q\times k}$ is the parameter of the hash function, and ξ is the tradeoff parameter. Then, the optimal $\mathbf{W}^v$ is

$$\mathbf{W}^v = \left(\phi\left(\mathbf{X}^v\right)^\top\phi\left(\mathbf{X}^v\right) + \xi\mathbf{I}_q\right)^{-1}\left(\phi\left(\mathbf{X}^v\right)^\top\mathbf{B}\right). \tag{30}$$

Out-of-Sample Extension: Based on the linear projection parameter $\mathbf{W}^v$ obtained above, the proposed **ASFOH** can easily map the query $\mathbf{x}_b^v$ into the hash codes $\mathbf{b}_q^v$

$$\mathbf{b}_q^v = \text{sgn}\left(\phi\left(\mathbf{x}_b^v\right)\mathbf{W}^v\right) \tag{31}$$

where $\phi(\mathbf{x}_b^v)$ is the kernelization of $\mathbf{x}_b^v$. The overall training procedures of **ASFOH** are described in Algorithm 1.

F. Complexity Analysis

The time cost of optimizing **ASFOH** includes the following parts. In the hash code generation stage, for each iteration, the time cost of updating $\mathbf{H}$, $\mathbf{R}$, and $\mathbf{B}$ is $\mathcal{O}(kcn + k^3 + k^2 + k^2n + Vkn + \sum_{v=1}^V kqn)$, $\mathcal{O}(c^3 + kcn + k^2n + ck^2 + k^3)$, and $\mathcal{O}(kcn)$, respectively. For variables $\mathbf{J}_v$ and $\mathbf{P}^v$ are $\mathcal{O}(kq^2)$ and $\mathcal{O}(kqn + k^2n + k^3 + qk^2)$, respectively. In the hash function learning stage, the linear projection parameter $\mathbf{W}^v$ is roughly $\mathcal{O}(q^3 + q^2n + kq^2 + kqn)$. Since $\varrho, c, k, q, V \ll n$, the total complexity of **ASFOH** is $\mathcal{O}((kc + k^2 + Vk + kq + q^2)n\varrho)$, where ϱ denotes the number of iterations. Taken together, the complexity of **ASFOH** is linear in n.

IV. Experiments

To evaluate the retrieval performance of **ASFOH**, we conducted extensive experiments in which we compared it to some state-of-the-art baselines on multiple datasets. The server used in the experiments is equipped with an Intel Xeon Silver 4210 CPU@2.2 GHz and 128-GB memory. A number of ablation studies were also performed.

A. Datasets

The *MIRFlickr* dataset [65] is composed of 25 000 images annotated with 24 categories. The instances with no less than

TABLE I
PERFORMANCE COMPARISON (MAP) OF **ASFOH** AND BASELINES ON THE THREE DATASETS WITH VARIOUS CODE LENGTHS

Task	Method	MIRFlickr					NUS-WIDE					IAPR-TC12				
		8 bits	16 bits	32 bits	64 bits	128 bits	8 bits	16 bits	32 bits	64 bits	128 bits	8 bits	16 bits	32 bits	64 bits	128 bits
I→T	CMFH	0.5599	0.5687	0.5680	0.5685	0.5687	0.3406	0.3437	0.3399	0.3409	0.3440	0.3579	0.3698	0.3704	0.3783	0.3804
	FSH	0.5911	0.6016	0.6149	0.6194	0.6242	0.3620	0.3732	0.3894	0.4014	0.4084	0.3677	0.3741	0.3830	0.3902	0.3993
	DCH	0.6659	0.6738	0.6859	0.6897	0.7030	0.5840	0.5808	0.5907	0.5932	0.5843	0.4118	0.4448	0.4597	0.4639	0.4727
	SMFH	0.5898	0.6071	0.6112	0.6257	0.6290	0.4619	0.4756	0.4811	0.4897	0.4962	0.3691	0.3689	0.3897	0.3905	0.3907
	SCM-seq	0.6235	0.6373	0.6478	0.6537	0.6611	0.5013	0.5120	0.5422	0.5488	0.5483	0.3590	0.3732	0.3709	0.3932	0.4104
	SePH-km	0.6641	0.6685	0.6818	0.6830	0.6873	0.5256	0.5537	0.5627	0.5622	0.5698	0.4094	0.4235	0.4246	0.4417	0.4523
	SCRATCH	0.7092	0.7131	0.7222	0.7265	0.7346	0.6038	0.6207	0.6338	0.6459	0.6496	0.4376	0.4518	0.4631	0.4890	0.4917
	BATCH	0.7440	0.7419	0.7589	0.7601	0.7619	0.6221	0.6337	0.6479	0.6668	0.6662	0.4613	0.4811	0.5091	0.5271	0.5322
	ASCSH	0.7542	0.7705	0.7898	0.7998	0.8072	0.6451	0.6861	0.6968	0.7107	0.7179	0.4665	0.5112	0.5369	0.5641	0.5698
	ASFOH	**0.7578**	**0.7761**	**0.7921**	**0.8089**	**0.8095**	**0.6754**	**0.6996**	**0.7059**	**0.7154**	**0.7182**	**0.4914**	**0.5211**	**0.5505**	**0.5774**	**0.5948**
T→I	CMFH	0.5615	0.5615	0.5606	0.5606	0.5608	0.3456	0.3498	0.3435	0.3486	0.3529	0.3613	0.3718	0.3807	0.3887	0.3914
	FSH	0.5869	0.5979	0.6114	0.6186	0.6251	0.3623	0.3717	0.3835	0.3973	0.4007	0.3608	0.3696	0.3912	0.3913	0.3911
	DCH	0.7256	0.7511	0.7585	0.7681	0.7909	0.7106	0.7103	0.7098	0.7260	0.7223	0.4667	0.5098	0.5516	0.5726	0.5777
	SMFH	0.6197	0.6338	0.6414	0.6499	0.6501	0.4978	0.5117	0.5106	0.5222	0.5231	0.3798	0.3836	0.4072	0.4182	0.4185
	SCM-seq	0.6103	0.6206	0.6298	0.6372	0.6427	0.4709	0.4836	0.5067	0.5141	0.5161	0.3601	0.3739	0.3715	0.3916	0.3926
	SePH-km	0.7033	0.7076	0.7212	0.7293	0.7348	0.6102	0.6407	0.6515	0.6608	0.6651	0.4461	0.4627	0.4814	0.4971	0.4973
	SCRATCH	0.7591	0.7762	0.7822	0.7978	0.8063	0.7210	0.7392	0.7549	0.7680	0.7755	0.4959	0.5153	0.5679	0.6109	0.6116
	BATCH	0.8078	0.8192	0.8237	0.8352	0.8388	0.7455	0.7514	0.7712	0.7797	0.7807	0.5195	0.5516	0.6069	0.6411	0.6463
	ASCSH	0.8109	0.8371	**0.8577**	0.8669	0.8692	0.7422	0.7792	0.8108	0.8249	0.8310	0.4771	0.5348	0.5878	0.6481	0.6512
	ASFOH	**0.8152**	**0.8395**	0.8522	**0.8675**	**0.8721**	**0.7887**	**0.8193**	**0.8302**	**0.8381**	**0.8422**	**0.5455**	**0.5994**	**0.6445**	**0.6832**	**0.7094**

20 textual tags will be chosen in our experiment. We randomly split the dataset into 18 015/2000 training(retrieval)/query sets. Each textual data is described by a 1386-dim bag-of-words (BoW) feature vector and each visual data is described by a 512-dim GIST feature vector.

The *NUS-WIDE* dataset [66] is composed of 270 000 images, and each image contain single or multiple semantic labels for 81 ground-truth tags. In our experiment, we chose the top ten most frequent tags and the corresponding 186 577 instances as the final data, and we randomly split the dataset into 184 577/2000 training(retrieval)/query sets. Each textual data is described by a 1000-dim BoW feature vector and each visual data is described by a 500-dim bag-of-visual words (BoVW) feature vector.

The *IAPR-TC12* dataset [67] has 20 000 image–text pair instances crawled from flickr. It is annotated with 255 labels. We randomly divided the dataset into 18 000/2000 training(retrieval)/query sets. Each textual data is described by a 2912-dim BoW vector and each visual data is described by a 512-dim GIST feature vector.

B. Compared Baselines and Evaluation Metric

The most typical task using **ASFOH** is multimedia similarity search. In this article, we validate the effectiveness of the proposed method via two retrieval tasks: 1) image2text, that is, retrieving texts using an image query and 2) text2image, that is, retrieving images using a text query. To evaluate the effectiveness of **ASFOH**, several state-of-the-art cross-modal hashing methods are considered as baselines. Specifically, they can be classified into: 1) two unsupervised cross-modal hashing methods, that is, CMFH [1] and FSH [21]; 2) seven supervised ones, that is, DCH [68], SMFH [2], SCM-seq [34], SePH-km [35], SCRATCH [42], BATCH [43], and ASCSH [45]; and 3) four deep ones, that is, DCMH [3], SSAH [46], RDCMH [48], and DMFH [49]. All comparison approaches are implemented through the codes or parameters supplied by the papers. The widely used mean average precision (mAP), precision–recall curve (PR), and **Top-k precision (P@k)** metrics are utilized to evaluate the retrieval performance.

C. Implementation Details

Regarding the hyperparameters of **ASFOH**, the parameters $\gamma, \beta, \eta, \rho$, and ω of hash code learning are selected by using grid search (in cases from 10^{-5} to 10^{4}, 10 times per step), and the parameter τ of the augmented Lagrange multiplier is set to 1.1. The value of anchor q is set to different values, in MIRFlickr and IAPR-TC12 datasets, q is set to 1200, in the NUS-WIDE dataset, q is set to 5000. The best performance is achieved when $\{\gamma = 3, \beta = 10^{-1}, \eta = 10^{-1}, \rho = 10^{-3}, \omega = 10^{-5}, \xi = 10^{-3}\}$, $\{\gamma = 2, \beta = 10^{0}, \eta = 10^{3}, \rho = 10^{-3}, \omega = 10^{-4}, \xi = 10^{-3}\}$ and $\{\gamma = 4, \beta = 10^{-1}, \eta = 10^{3}, \rho = 10^{0}, \omega = 10^{0}, \xi = 10^{-3}\}$ on MIRFlickr, NUS-WIDE, and IAPR-TC12 datasets, respectively.

D. Comparisons With State-of-the-Art Baselines

Table I shows the mAP results of each method on the three datasets, where it can be noted as follows.

1) On the three benchmark datasets, the proposed **ASFOH** outperforms all compared baselines in all cases on image2text and text2image tasks. Specifically, for the image2text task, compared to the best baseline, that is, ASCSH, **ASFOH** achieves a relative improvement of 0.5%, 1.7%, and 3.3% corresponding to MIRFlickr, NUS-WIDE, and IAPR-TC12 datasets, respectively, and at least 0.2%, 3.4%, and 10.1% performance gains for the text2image task on MIRFlickr, NUS-WIDE, and IAPR-TC12, respectively. The improvement in mAP scores indicates that the use of an adaptive weight scheme and asymmetric hash learning framework can enhance the hash code learning process, and in addition, the discriminative power of the learned hash codes can be improved by considering the balance and decorrelation constraints.
2) The mAP scores of most hashing methods increase with the increase in the code lengths. This is specifically shown by our proposed **ASFOH**, which yielded a significant increase in retrieval performance in cases where there were long code lengths. The reason for this phenomenon is that longer code lengths can encode more semantic information.

3) Supervised cross-modal hashing methods, that is, SMFH, SCM-Seq, SePH-km, SCRATCH, BATCH, and ASCSH are generally better than the unsupervised ones, that is, CMFH, FSH, and DCH. The reason is that supervised information can better shorten the gap between similar pairs and widen the gap between dissimilar ones, making the learned hash codes more discriminative.
4) By comparing discrete optimization-based methods, that is, **ASFOH**, ASCSH, BATCH, SCRATCH, and DCH, to relaxation-based hashing methods, that is, FSH, SePH-km, SCM-seq, SMFH, and CMFH, we find that the discrete optimization-based methods outperform relaxation-based ones, which means that relaxation optimization algorithms suffer from a large quantization error and this results in a suboptimal solution.
5) Although ASCSH is an efficient discrete cross-modal hashing method, **ASFOH** achieves a much higher mAP performance than ASCSH. The reasons for this may be as follows.
    a) ASCSH is a one-step hashing method, and it learns the common latent representation by projecting the data features. This learning paradigm cannot guarantee the information completeness of the learned common latent representation.
    b) ASCSH only considers the binary constraint and ignores the desirable decorrelation and balance constraints of hash bits, which could lower the quality of the hash code generated. **ASFOH** flexibly solves the limitations of ASCSH through an ingenious learning architecture, thus, achieving better retrieval performance.

The major causes of superior performance for **ASFOH** lie in the following fact.

1) To improve the hash performance and guarantee the complementarity of multimodal features, **ASFOH** learns a compact common latent representation by mimicking the degradation process of data transmission in the proposed joint learning framework.
2) The nuclear norm minimization operation efficiently optimizes the low-dimensional embedding to improve the robustness of the common latent representation.
3) To enhance discriminative capability, an adaptive supervised asymmetric hash learning framework is developed, which can link the learned hash codes with the common latent representation and the high-level semantics.

Besides, we also report the P@k and PR curves of all the compared baselines on the three datasets and report them in Figs. 2 and 3, respectively. From Fig. 2, by measuring the area under the PR curve, it is apparent that **ASFOH** is also superior to all the compared baselines in most cases on image2text and text2image tasks, which further shows the effectiveness of the proposed **ASFOH**. From Fig. 3, it is apparent that the proposed **ASFOH** outperforms all the baselines based on the P@k metric, which attests to its effectiveness for image2text and text2image tasks. Specifically, even compared with the most competitive baseline, that is, ASCSH, **ASFOH** still demonstrates obvious advantages.

TABLE II
MAP VALUES OF **ASFOH** AND BASELINES WITH EXTREMELY LOW BITS PERFORMING ON MIRFLICKR AND NUS-WIDE DATASETS

Task	Method	MIRFlickr			NUS-WIDE		
		5 bits	6 bits	7 bits	4 bits	5 bits	6 bits
I→T	SCRATCH	0.6886	0.6929	0.6699	0.5451	0.5801	0.5832
	ASCSH	0.6911	0.7035	0.7276	0.5539	0.6128	0.6359
	ASFOH	**0.7242**	**0.7337**	**0.7360**	**0.6270**	**0.6380**	**0.6589**
T→I	SCRATCH	0.7440	0.7357	0.7171	0.6383	0.6657	0.6886
	ASCSH	0.7410	0.7671	0.7768	0.6633	0.7019	0.7434
	ASFOH	**0.7720**	**0.7830**	**0.7949**	**0.7250**	**0.7469**	**0.7724**

E. Results on Extremely Short Hash Codes

As described above, the hash codes of each object belonging to the same category should be the same, thus, the total number of hash codes theoretically required is $\log_2(c)$, where c is the number of categories. However, most hashing methods cannot guarantee a satisfactory retrieval performance with the extremely short hash codes, that is, $k = \lceil \log_2(c) \rceil$. In the experiments, the code lengths in the MIRFlickr and NUS-WIDE datasets are set to $\{5, 6, 7\}$ and $\{4, 5, 6\}$, respectively. As shown in Table II, the proposed **ASFOH** method consistently outperforms the best two compared baselines, that is, SCRATCH and ASCSH, on MIRFlickr and NUS-WIDE datasets in all cases, which demonstrates its effectiveness. Specifically, when embedding extremely short hash code (e.g., 5 bit for the MIRFlickr dataset and 4 bit for the NUS-WIDE dataset), for the image2text task, compared to the best compared baseline, that is, ASCSH, **ASFOH** achieves at least 6.9% and 7.0% performance gain corresponding to MIRFlickr and NUS-WIDE datasets, respectively, and at least 2.9% and 6.5% performance gain for text2image task on MIRFlickr and NUS-WIDE datasets, respectively. Short bit length hash codes save storage space and improve retrieval efficiency, therefore, our proposed **ASFOH** method is more suitable for practical applications.

F. Training Efficiency

To evaluate the efficiency of **ASFOH**, we reported its training time and compared it against other approaches on the NUS-WIDE dataset as shown in Table III. Except for SePH-km, the other approaches shown in Table III claim a computational complexity of $\mathcal{O}(n)$, but the differences between them are considerably wide. One such difference is in the training time for methods, such as CMFH, DCH, SCM-seq, and ASCSH. This increases significantly with increase in the code length. As for SMFH, SCRATCH, BATCH, and **ASFOH**, increase in code length only results in a slight increase in training time. The reason for this is that SCRATCH, BATCH, and **ASFOH** generate hash codes directly with a discrete optimization strategy in one step. Besides that, the above approaches use a simple linear hash function to learn the out-of-sample instances. In addition, the SMFH method learns the linear hash function directly without generating hash codes. Although this method has a low computational complexity, it results in lower retrieval accuracy due to the presence of a large quantization error. Specifically, the training costs shown in Table III demonstrate the following.

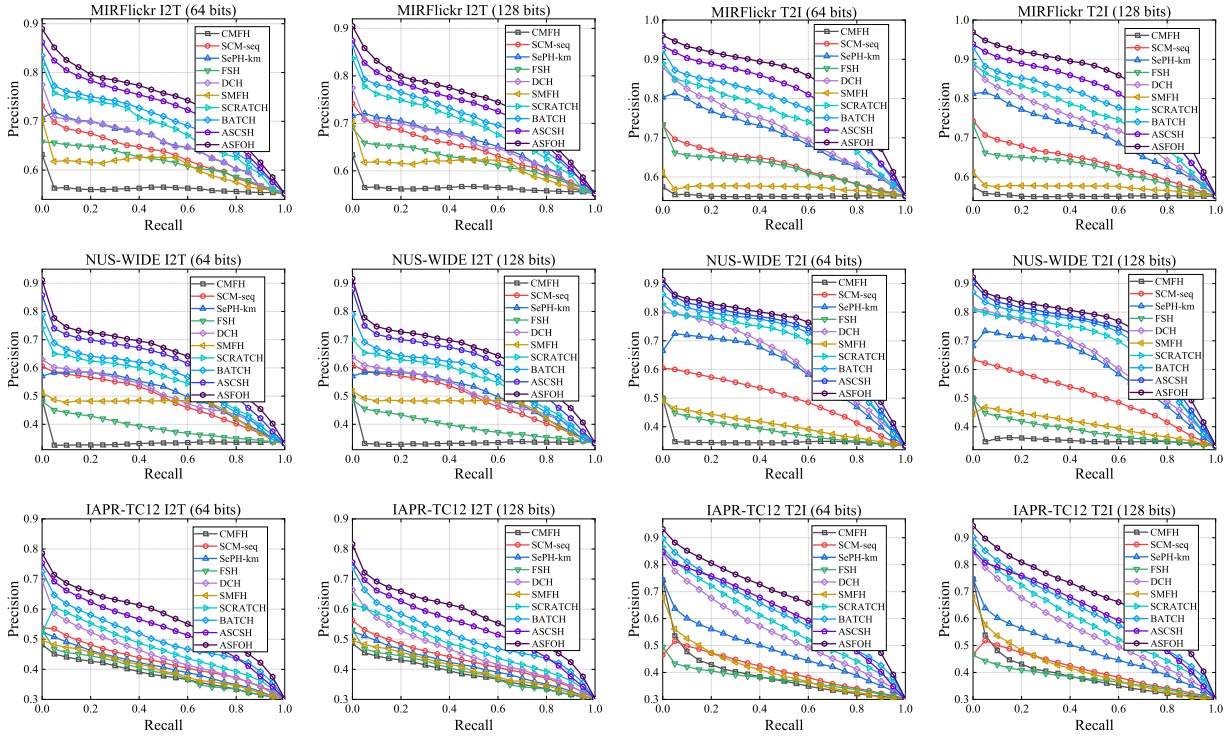


Fig. 2. PR curves of the compared baselines on MIRFlickr, NUS-WIDE, and IAPR-TC12 datasets.

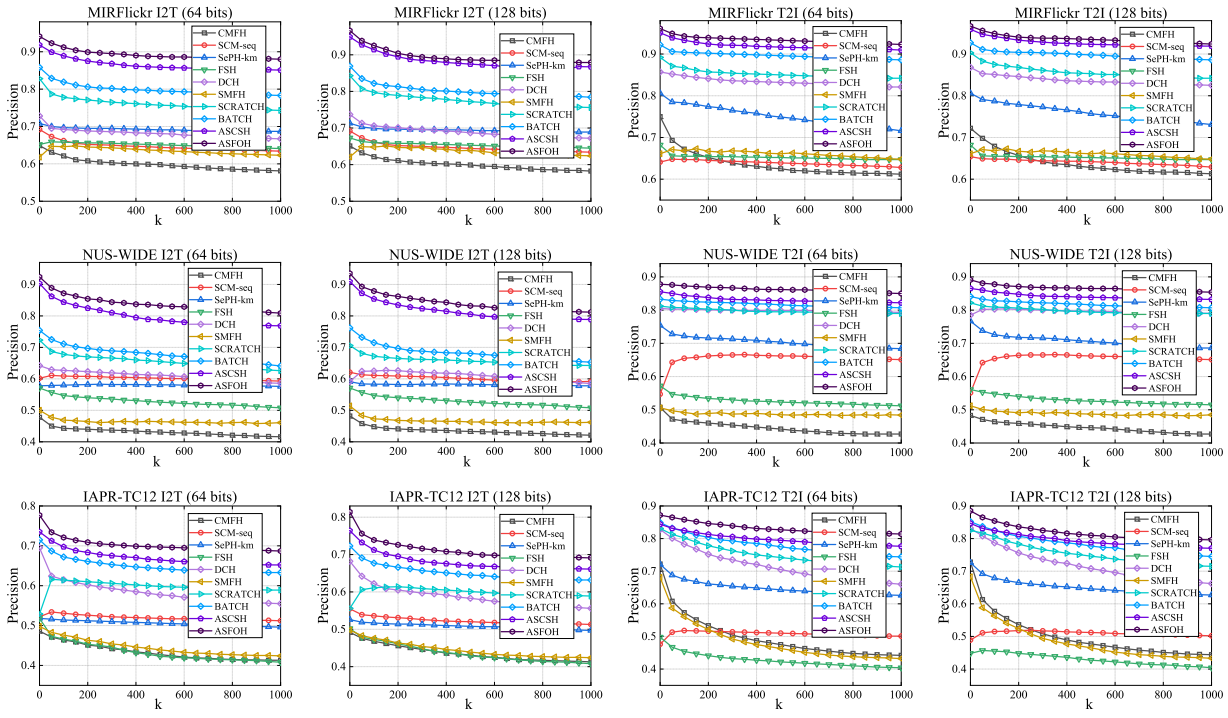


Fig. 3. Top-k precision curves of the compared baselines on MIRFlickr, NUS-WIDE, and IAPR-TC12 datasets.

1) The computational cost of hashing methods that use pairwise similarity information, that is, SMFH, SePH-km, ASCSH, and **ASFOH**, is more expensive than that of directly using semantic label information, that is, DCH, SCRATCH, and BATCH. The reason is that the pairwise similarity information contains the semantic relationships between each pair of instances, thus, the time complexity of matrix computation will be higher.

2) Compared with the state-of-the-art ASCSH whose retrieval performance is closest to **ASFOH**, the training cost of **ASFOH** is much lower than that of ASCSH and does not significantly increase with increase in the code length. This is because the discrete optimization strategy we proposed can generate all hash codes simultaneously, avoiding the high training cost caused by bit-by-bit optimization schemes.

TABLE III
TRAINING COSTS (SECONDS) OF **ASFOH** AND BASELINES ON THE NUS-WIDE DATASETS WITH VARIOUS CODE LENGTHS

Method	8 bits	16 bits	32 bits	64 bits	128 bits
CMFH	15.57	26.63	38.24	42.58	48.40
FSH	33.35	37.75	38.12	44.10	46.94
DCH	3.27	3.54	4.78	8.32	28.15
SMFH	3.94	3.46	4.29	5.78	6.86
SCM-seq	4.78	8.45	14.42	24.77	63.62
SePH-km	1578.77	1854.54	2240.87	2498.54	2877.28
SCRATCH	2.43	2.47	3.04	3.52	7.45
BATCH	0.13	0.15	0.22	0.28	0.47
ASCSH	13.26	29.91	41.21	73.53	156.90
ASFOH	0.32	0.36	0.91	3.03	13.86

TABLE IV
AVERAGE RUNNING TIME (MILLISECONDS) FOR EACH INSTANCE OF **ASFOH** ON THE NUS-WIDE DATASET

Task	Metrics	16 bits	64 bits
I→T	T_{R100}	4.23e-2	4.26e-2
	T_{R1000}	4.56e-2	4.58e-2
	$T_{H\leq 2}$	9.69e-1	7.36e-1
T→I	T_{R100}	4.21e-2	4.22e-2
	T_{R1000}	4.61e-2	4.57e-2
	$T_{H\leq 2}$	9.59e-1	7.23e-1

3) Although the time cost of **ASFOH** is higher than some baselines, the retrieval performance of **ASFOH** has improved significantly. In addition, we randomly sample different numbers of instances from NUS-WIDE datasets to prove that the time cost of our proposed **ASFOH** is linear to the number of the training datasets.

The corresponding training costs of **ASFOH** on these datasets are reported in Fig. 4. From the table, it can be seen that the training costs of **ASFOH** are generally linear to the size of training datasets, making it suitable for large-scale multimedia datasets. This makes **ASFOH** more suitable for practical applications where it showcases a good tradeoff between time cost and search performance.

Furthermore, Table IV demonstrates the average running time (milliseconds) of our model on the NUS-WIDE dataset for each instance on the I2T and T2I tasks, respectively. The running performance of **ASFOH** is evaluated by three metrics, that is, T_{R100}, T_{R1000}, and $T_{H\leq 2}$, where T_{R100} and T_{R1000} denote the running time needed to return the top-100 instances and top-1000 instances, respectively, and $T_{H\leq 2}$ refers to the running time required to retrieve instances within a Hamming distance of 2. In Table IV, it is apparent that: 1) the running time for returning 1000 instances is higher than that for returning 100 instances, since it takes longer to return more instances and 2) the 16-bit code length requires more running time than the 64-bit code length under the $T_{H\leq 2}$ metric, since the Hamming space gets sparser as the code length increases, resulting in few instances falling into a Hamming ball with radius 2.

G. Comparison With Deep Hashing

Currently, DNNs have made significant progress in image representation. In order to verify the generalization ability and validity of **ASFOH**, we compare **ASFOH** with several state-of-the-art deep-learning-based hashing methods, that is, DMFH, RDCMH, SSAH, and DCMH. We extract the last layer of the VGG-16 network, that is, a 4096-D vector, as each visual data for fairness. The deep version of **ASFOH**, will be referred to as **ASFOH-Deep**. Table V shows the mAP values of different methods on the datasets of MIRFlickr and NUS-WIDE, where the code length are set to 16, 32, and 64 bits. Specifically, it can be observed as follows.

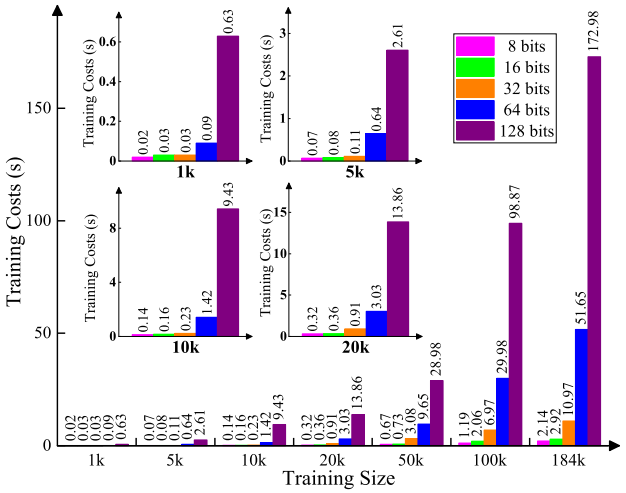


Fig. 4. Training costs (seconds) of **ASFOH** on NUS-WIDE with different training size.

1) **ASFOH-Deep** gets the highest mAP value in each case. Specifically, **ASFOH-Deep** achieves a boost of 29.1% in average mAP scores on the NUS-WIDE dataset for different lengths of hash codes in the two retrieval tasks.
2) For the cross-modal retrieval, deep-learning hashing methods can get better accuracy for the case of retrieving text from images, but have the opposite effect in the case of retrieving images from text, which may be due to the fact that the increased dimensionality of the deep image representation makes retrieval from text to images difficult.
3) *ASFOH-Deep* has a competitive retrieval performance though it is not an end-to-end deep hashing method. The reason for this is that **ASFOH-Deep** can ensure the information completeness of multimodal data and make full use of the semantic supervisions. In addition, the proposed discrete optimization strategy is important for learning better hash codes.

H. Analytical Experiments

To gain deep insight into **ASFOH**, we further design some variants of **ASFOH** to perform ablation studies as follows.

1) Effects of Kernelization: To verify the effectiveness of kernelization, we design a variant of **ASFOH**, termed **ASFOH-K**, which uses the original high-dimensional features rather than the kernel-based ones, that is, the first term of (11) is replaced by $\|\mathbf{X}^v - \mathbf{H}\mathbf{P}^v\|^2$.

2) Effects of Nuclear Norm: To verify the effects of the nuclear norm for the matrix $\mathbf{P}^v$, that is, ***RQ1: Common Latent***

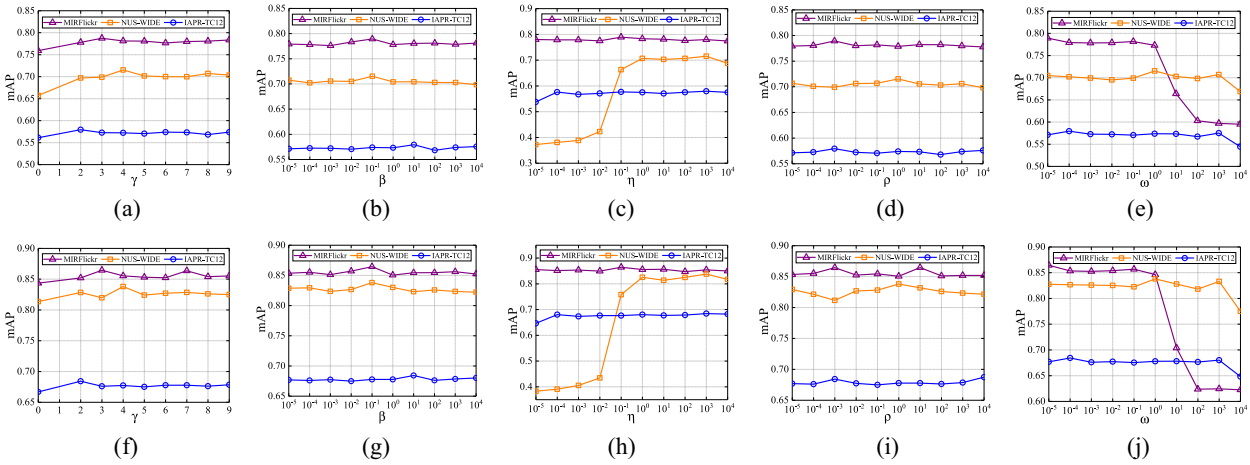


Fig. 5. Parameter sensitivity analysis of γ, β, η, ρ, and ω on the three datasets. (a) I2T @ 64 bits. (b) I2T @ 64 bits. (c) I2T @ 64 bits. (d) I2T @ 64 bits. (e) I2T @ 64 bits. (f) T2I @ 64 bits. (g) T2I @ 64 bits. (h) T2I @ 64 bits. (i) T2I @ 64 bits. (j) T2I @ 64 bits.

TABLE V
MAP VALUES OF **ASFOH** AND DEEP HASHING ON MIRFLICKR AND NUS-WIDE DATASETS

Task	Method	MIRFlickr			NUS-WIDE		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
I→T	DCMH [3]	0.7410	0.7465	0.7485	0.5903	0.6031	0.6093
	SSAH [46]	0.7970	0.8090	0.8100	0.6421	0.6424	0.6333
	RDCMH [48]	0.7723	0.7735	0.7789	0.6231	0.6236	0.6273
	DMFH [49]	0.7997	0.8105	0.8121	0.6472	0.6490	0.6506
	ASFOH-Deep	**0.8574**	**0.8778**	**0.8816**	**0.8429**	**0.8619**	**0.8720**
T→I	DCMH [3]	0.7827	0.7900	0.7932	0.6389	0.6511	0.6571
	SSAH [46]	0.7820	0.7970	0.7990	0.6212	0.6361	0.6457
	RDCMH [48]	0.7931	0.7924	0.8001	0.6641	0.6685	0.6694
	DMFH [49]	0.7968	0.8040	0.8082	0.6318	0.6369	0.6681
	ASFOH-Deep	**0.8252**	**0.8435**	**0.8535**	**0.7961**	**0.8142**	**0.8231**

Representation Completeness Learning, we design a variant of **ASFOH**, named **ASFOH-NN**, which uses the Frobenius norm rather than the nuclear norm based ones, that is, the second term of (11) is replaced by $\|\mathbf{P}^v\|^2$.

3) Effects of Semantic Projection: To verify the effects of semantic projection, that is, ***RQ2: Asymmetric Semantic Hash Learning***, we design a variant of **ASFOH**, named **ASFOH-SR**, which ignores the term $\min_{\mathbf{R}} \|\mathbf{YR} - \mathbf{B}\|^2$.

4) Effects of Bit Balance and Bit Uncorrelated Constraints: To verify the efficiency of bit balance and bit uncorrelated constraints, that is, ***RQ3: Balance and Decorrelation***, we design three variants of **ASFOH**, that is, **ASFOH-B**, **ASFOH-BB**, and **ASFOH-BD**. Specifically, 1) **ASFOH-B** only considers the binary constraint, that is, $\mathbf{B} \in \{-1, 1\}^{n\times k}$; 2) **ASFOH-BB** considers both the binary and balance constraints, that is, $\mathbf{B} \in \{-1, 1\}^{n\times k}$, $\mathbf{B}^\top \mathbf{1}_n = \mathbf{0}_k$; and 3) **ASFOH-BD** considers the binary and decorrelation constraints, that is, $\mathbf{B} \in \{-1, 1\}^{n\times k}$ and $\mathbf{B}^\top \mathbf{B} = n\mathbf{I}_k$.

Table VI reports the ablation results of the above variants, from where it can be observed as follows.

1) It is difficult to directly use the linear model to deal with the complex nonlinear relationships contained in the multimodal data. However, the kernelization operation can implicitly deal with the nonlinear relationships in the heterogeneous data, thereby making the generated hash codes have a better retrieval performance.
2) *ASFOH* outperforms **ASFOH-NN**, which demonstrates that the nuclear norm operation effectively adjusts the low-dimensional representation to explore and balance the completeness of the learned common latent subspace to support the scheme of ***RQ1***.
3) *ASFOH* outperforms **ASFOH-SR**, which shows that the semantic projection strategy is helpful in improving the semantics of generated hash codes to support the scheme of ***RQ2***, thereby enhancing retrieval performance.
4) By comparing **ASFOH-BB** and **ASFOH-BD** to **ASFOH-B**, it can be observed that the learned binary codes can boost the discriminative power under the positive contribution of the *decorrelation and balance constraints*; by comparing **ASFOH-BB** to **ASFOH-BD**, it can be noted that the *decorrelation constraint* has a better performance than the *balance constraint* for the discriminative power of the learned hash codes; the *two constraints* are able to improve the search performance of the model in a coordinated manner. The experimental results demonstrate the effectiveness of the ***RQ3*** strategy.

I. Parameter Sensitivity Analysis

To further evaluate the effects of different parameters on the three datasets, we set the values of $\gamma, \beta, \eta, \rho$, and ω in the range of $\{10^{-5}, \ldots, 10^4\}$ and the code length is set to 64 bit. The mAP results are plotted in Fig. 5. On the MIRFlickr dataset, the mAP scores are relatively stable over a large range of these parameters, that is, $\gamma \in [3, 7]$, $\beta, \eta, \rho \in [10^{-5}, 10^4]$, and $\omega \in [10^{-5}, 10^0]$. On the NUS-WIDE dataset, **ASFOH** can obtain stable retrieval performance under a wider range of these parameters, that is, $\gamma = 4$, $\beta, \rho \in [10^{-5}, 10^4]$, $\eta \in [10^0, 10^4]$, and $\omega \in [10^{-5}, 10^3]$. In addition, **ASFOH** can obtain expected retrieval performance on the IAPR-TC12 dataset when $\gamma = 2$, $\beta, \rho \in [10^{-5}, 10^4]$, $\eta \in [10^{-4}, 10^4]$, and $\omega \in [10^{-5}, 10^3]$.

Note that, in order to validate the effectiveness of the dynamic fusion mechanism, we further tested the performance of the proposed **ASFOH** when the parameter $\gamma = 0$, which ensures that all modalities have the same contribution to the

TABLE VI
PERFORMANCE COMPARISON (MAP) OF THE VARIANTS OF **ASFOH** ON MIRFLICKR, NUS-WIDE, AND IAPR-TC12 DATASETS

Task	Method	MIRFlickr					NUS-WIDE					IAPR-TC12				
		8 bits	16 bits	32 bits	64 bits	128 bits	8 bits	16 bits	32 bits	64 bits	128 bits	8 bits	16 bits	32 bits	64 bits	128 bits
I→T	**ASFOH-K**	0.7158	0.7295	0.7556	0.7692	0.7705	0.6732	0.6903	0.7024	0.7098	0.7116	0.4611	0.4874	0.5168	0.5367	0.5563
	ASFOH-B	0.6972	0.7113	0.7330	0.7481	0.7498	0.6044	0.6086	0.6122	0.6275	0.6401	0.4448	0.4875	0.5017	0.5192	0.5224
	ASFOH-BB	0.7116	0.7148	0.7397	0.7575	0.7659	0.6096	0.6213	0.6289	0.6426	0.6541	0.4632	0.4938	0.5320	0.5503	0.5694
	ASFOH-BD	0.7158	0.7189	0.7477	0.7729	0.7762	0.6223	0.6354	0.6490	0.6642	0.6768	0.4678	0.5125	0.5397	0.5618	0.5823
	ASFOH-SR	0.6634	0.6758	0.6919	0.7146	0.7267	0.5979	0.6064	0.6098	0.6231	0.6423	0.3995	0.4158	0.4353	0.4482	0.4679
	ASFOH-NN	0.7312	0.7517	0.7608	0.7772	0.7880	0.6611	0.6697	0.6834	0.6871	0.6872	0.4439	0.4789	0.5228	0.5501	0.5749
	ASFOH	**0.7578**	**0.7761**	**0.7921**	**0.8089**	**0.8123**	**0.6754**	**0.6996**	**0.7059**	**0.7154**	**0.7182**	**0.4914**	**0.5211**	**0.5505**	**0.5774**	**0.5948**
T→I	**ASFOH-K**	0.7825	0.8012	0.8327	0.8499	0.8525	0.7788	0.8107	0.8194	0.8279	0.8365	0.5245	0.5646	0.6085	0.6433	0.6731
	ASFOH-B	0.7356	0.7676	0.7790	0.7867	0.7896	0.6379	0.6452	0.6497	0.6511	0.6572	0.4458	0.5280	0.5893	0.6313	0.6578
	ASFOH-BB	0.7589	0.7724	0.7856	0.7991	0.8080	0.6610	0.6677	0.6714	0.6776	0.6815	0.4516	0.5422	0.6099	0.6510	0.6742
	ASFOH-BD	0.7708	0.7861	0.7942	0.8107	0.8185	0.6779	0.6923	0.6970	0.7111	0.7245	0.4583	0.5552	0.6157	0.6677	0.6789
	ASFOH-SR	0.6987	0.7234	0.7330	0.7581	0.7736	0.6161	0.6298	0.6437	0.6573	0.6668	0.4643	0.4912	0.5176	0.5554	0.5869
	ASFOH-NN	0.7809	0.8241	0.8398	0.8510	0.8578	0.7604	0.8043	0.8111	0.8207	0.8245	0.5200	0.5814	0.6312	0.6680	0.6874
	ASFOH	**0.8152**	**0.8395**	**0.8522**	**0.8675**	**0.8721**	**0.7887**	**0.8193**	**0.8302**	**0.8381**	**0.8422**	**0.5455**	**0.5994**	**0.6445**	**0.6832**	**0.7094**

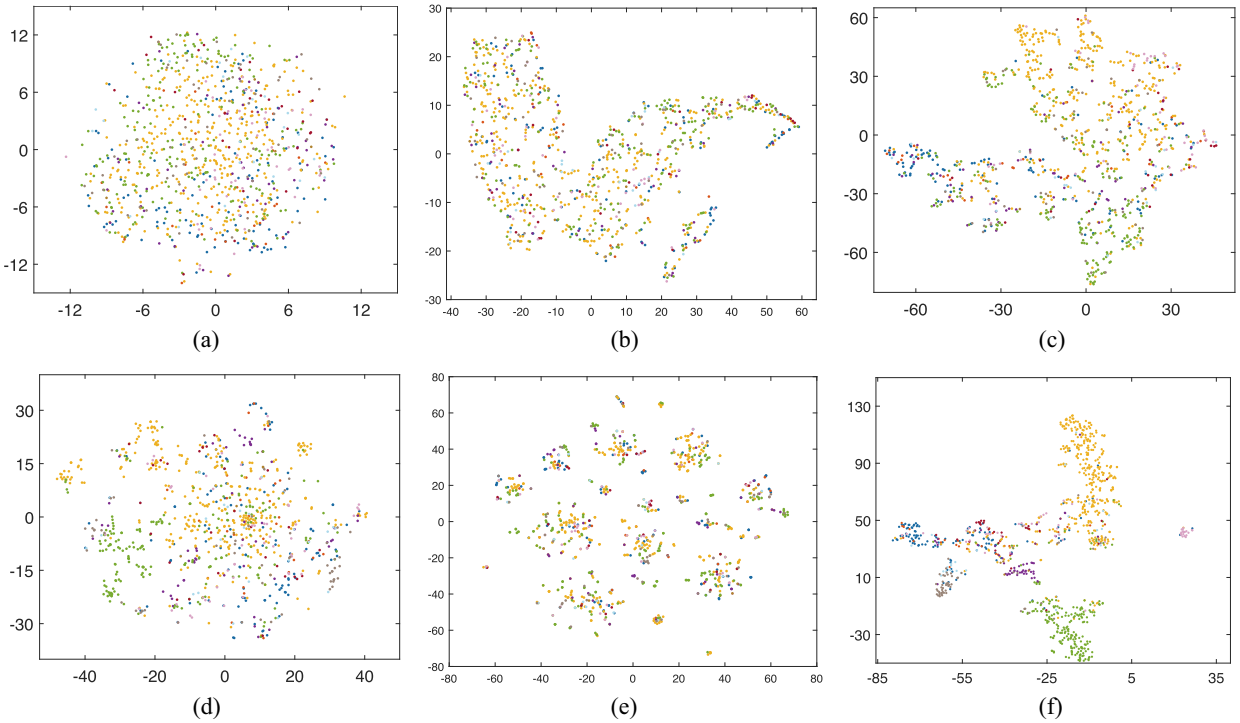


Fig. 6. t-SNE visualization on the NUS-WIDE dataset@32 bits. (a) Original image embeddings. (b) Kernelized image embeddings. (c) Final image embeddings. (d) Original text embeddings. (e) Kernelized text embeddings. (f) Final text embeddings.

common unified latent representation. From Fig. 5, it can be observed that when the value of γ is set to zero, the retrieval performance is reduced, which further attests to the effectiveness of the proposed dynamic fusion mechanism.

J. Embedding Visualization

In this section, the t-distributed stochastic neighbor embedding (t-SNE) [69] is utilized to visualize the different embeddings on the NUS-WIDE dataset as shown in Fig. 6. Note that, since the datasets used in our experiments are all multilabel datasets, it is difficult to define the specific classification of each instance in the t-SNE visualization software. Therefore, we selected ten types of samples with only one category in the NUS-WIDE dataset, that is, ten types of single-label instances, for visual display. Compared with the original embeddings and the kernelized ones, more obvious separation is achieved after performing our proposed subspace projection operations, that is, the final embeddings, which we use to demonstrate that our proposed **ASFOH** model exhibits better discriminability in a robust common latent space.

K. Convergence Analysis

To further prove that the proposed alternative optimization algorithms are able to converge, experiments related to the curves of the normalized objective values were conducted on the three datasets with the code lengths of 16, 32, and 64 bits, and the results are plotted in Fig. 7. Specifically, the objective function can converge quickly within 3–5 iterations, which proves the efficiency of the proposed alternative optimization process.

As described in Section III-D, our proposed method requires optimizing seven variables, and the remaining six variables need to be fixed when one of them needs to be updated, and then by continuously iterating this process until the overall algorithm converges. To facilitate the analysis, we denote the objective function by $\mathcal{L}(\alpha_v^{(t)}, \mathbf{R}^{(t)}, \mathbf{H}^{(t)}, \mathbf{B}^{(t)}, \mathbf{P}^{v(t)}, \mathbf{J}^{v(t)}, \Theta^{v(t)})$.

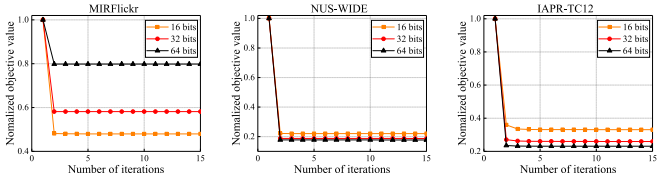


Fig. 7. Convergence curves of **ASFOH** on the three datasets. (a) MIRFlickr. (b) NUS-WIDE. (c) IAPR-TC12.

We find that the value of the objective function $\mathcal{L}$ decreases until it finally stabilizes from the optimization process in Section III-D. Due to the existence of an analytical solution for each variable, the objective function decreases until it finally converges during the execution of the algorithm. Therefore, the convergence of the proposed optimization scheme is theoretically guaranteed.

V. Conclusion

In this article, we have introduced a novel nuclear norm minimization-based asymmetric hashing method, which fully guarantees the completeness of information among multimodal data. In the proposed **ASFOH**, the core component hash function learning module leverages both a dynamic fusion mechanism and an asymmetric hash learning framework to encode more consensus and discriminative information from different modalities to learn high-quality binary codes. Besides, **ASFOH** solves the semantic error caused by using only the pairwise similarity information by mining the rich semantic information existing in the multilabel information, and, thus, improves the search accuracy of the learned hash codes. In addition, a discrete optimization strategy without a relaxation strategy is proposed to solve the discrete optimization problem, which bridges the quantization gap while ensuring the efficiency of hash code learning. The retrieval performance of **ASFOH** is quantitatively evaluated on three datasets, which demonstrates that our model outperforms the compared state-of-the-art approaches. In the future, we will consider the interactions of fine-grained local features and coarse-grained global features of the instances and extend the proposed method to deep transformer-based models.

References

[1] G. Ding, Y. Guo, J. Zhou, and Y. Gao, "Large-scale cross-modality search via collective matrix factorization hashing," *IEEE Trans. Image Process.*, vol. 25, pp. 5427–5440, 2016.

[2] J. Tang, K. Wang, and L. Shako, "Supervised matrix factorization hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 25, pp. 3157–3166, 2016.

[3] Q. Jiang and W. Li, "Deep cross-modal hashing," in *Proc. CVPR*, 2017, pp. 3270–3278.

[4] D. Hu, F. Nie, and X. Li, "Deep binary reconstruction for cross-modal hashing," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 973–985, Apr. 2019.

[5] X. Fang, K. Jiang, N. Han, S. Teng, G. Zhou, and S. Xie, "Average approximate hashing-based double projections learning for cross-modal retrieval," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 11780–11793, Nov. 2022.

[6] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep learning to hash by continuation," in *Proc. ICCV*, 2017, pp. 5609–5618.

[7] Q. Li, Z. Sun, R. He, and T. Tan, "Deep supervised discrete hashing," in *Proc. NeurIPS*, 2017, pp. 2479–2488.

[8] J. Zhang, Y. Peng, and M. Yuan, "SCH-GAN: Semi-supervised cross-modal hashing by generative adversarial network," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 489–502, Feb. 2020.

[9] Z. Wang, Z. Zhang, Y. Luo, Z. Huang, and H. T. Shen, "Deep collaborative discrete hashing with semantic-invariant structure construction," *IEEE Trans. Multimedia*, vol. 23, no. 5, pp. 1274–1286, May 2021.

[10] G. Dong, X. Zhang, X. Shen, L. Lan, Z. Luo, and X. Ying, "Discriminative geometric-structure-based deep hashing for large-scale image retrieval," *IEEE Trans. Cybern.*, early access, May 23, 2022, doi: 10.1109/TCYB.2022.3173315.

[11] H. Zhang, L. Liu, Y. Long, and L. Shako, "Unsupervised deep hashing with pseudo labels for scalable image retrieval," *IEEE Trans. Image Process.*, vol. 27, pp. 1626–1638, 2018.

[12] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3034–3044, Dec. 2018.

[13] G. Wu et al., "Unsupervised deep hashing via binary latent factor models for large-scale cross-modal retrieval," in *Proc. IJCAI*, 2018, pp. 2854–2860.

[14] S. Su, Z. Zhong, and C. Zhang, "Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval," in *Proc. ICCV*, 2019, pp. 3027–3035.

[15] S. Li, Z. Chen, X. Li, J. Lu, and J. Zhou, "Unsupervised variational video hashing with 1D-CNN-LSTM networks," *IEEE Trans. Multimedia*, vol. 22, no. 6, pp. 1542–1554, Jun. 2020.

[16] T. Hoang, T. Do, H. Le, D. L. Tan, and N. Cheung, "Simultaneous compression and quantization: A joint approach for efficient unsupervised hashing," *Comput. Vis. Image Understand.*, vol. 191, Feb. 2020, Art. no. 102852.

[17] 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*, 2013, pp. 785–796.

[18] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 415–424.

[19] J. Zhang, Y. Peng, and M. Yuan, "Unsupervised generative adversarial cross-modal hashing," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 539–546.

[20] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[21] H. Liu, R. Ji, Y. Wu, F. Huang, and B. Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *Proc. CVPR*, 2017, pp. 6345–6353.

[22] C. Li, C. Deng, L. Wang, D. Xie, and X. Liu, "Coupled CycleGAN: Unsupervised hashing network for cross-modal retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 176–183.

[23] B. Zhang and J. Qian, "Autoencoder-based unsupervised clustering and hashing," *Appl. Intell.*, vol. 51, no. 1, pp. 493–505, 2021.

[24] H. Cui, L. Zhu, J. Li, Y. Yang, and L. Nie, "Scalable deep hashing for large-scale social image retrieval," *IEEE Trans. Image Process.*, vol. 29, pp. 1271–1284, 2020.

[25] P. Hu, X. Peng, H. Zhu, J. Lin, L. Zhen, and D. Peng, "Joint versus independent multiview hashing for cross-view retrieval," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4982–4993, Oct. 2021.

[26] F. Zheng, Y. Tang, and L. Shako, "Hetero-manifold regularisation for cross-modal hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1059–1071, May 2018.

[27] D. Tian, D. Zhou, M. Gong, and Y. Wei, "Interval type-2 fuzzy logic for semisupervised multimodal hashing," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3802–3812, Jul. 2021.

[28] 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*, 2017, pp. 1618–1625.

[29] Y. Chen and X. Lu, "Deep category-level and regularized hashing with global semantic similarity learning," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 6240–6252, Dec. 2021.

[30] Q. Jiang and W. Li, "Discrete latent factor model for cross-modal hashing," *IEEE Trans. Image Process.*, vol. 28, pp. 3490–3501, 2019.

[31] T. Yao, X. Kong, H. Fu, and Q. Tian, "Discrete semantic alignment hashing for cross-media retrieval," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4896–4907, Dec. 2020.

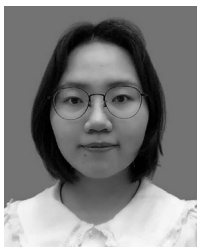
[32] L. Wu, Y. Wang, and L. Shako, "Cycle-consistent deep generative hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 28, pp. 1602–1612, 2019.

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

[34] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2177–2183.
[35] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. CVPR*, 2015, pp. 3864–3872.
[36] Z. Yang et al., "NSDH: A nonlinear supervised discrete hashing framework for large-scale cross-modal retrieval," *Knowl.-Based Syst.*, vol. 217, Apr. 2021, Art. no. 106818.
[37] X. Luo, P. Zhang, Y. Wu, Z. Chen, H. Huang, and X. Xu, "Asymmetric discrete cross-modal hashing," in *Proc. ICMR*, 2018, pp. 204–212.
[38] X. Lu, L. Zhu, Z. Cheng, L. Nie, and H. Zhang, "Online multi-modal hashing with dynamic query-adaption," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 715–724.
[39] Z. Yang, J. Long, L. Zhu, and W. Huang, "Nonlinear robust discrete hashing for cross-modal retrieval," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1349–1358.
[40] X. Lu, L. Zhu, L. Liu, L. Nie, and H. Zhang, "Graph convolutional multi-modal hashing for flexible multimedia retrieval," in *Proc. ACM MM*, 2021, pp. 1414–1422.
[41] H. T. Shen et al., "Exploiting subspace relation in semantic labels for cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3351–3365, Oct. 2021.
[42] Z. Chen, C. Li, X. Luo, L. Nie, W. Zhang, and X. 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.
[43] Y. Wang, X. Luo, L. Nie, J. Song, W. Zhang, and X. Xu, "BATCH: A scalable asymmetric discrete cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 11, pp. 3507–3519, Nov. 2021.
[44] X. Lu, L. Liu, L. Nie, X. Chang, and H. Zhang, "Semantic-driven interpretable deep multi-modal hashing for large-scale multimedia retrieval," *IEEE Trans. Multimedia*, vol. 23, no. 12, pp. 4541–4554, Dec. 2021.
[45] M. Meng, H. Wang, J. Yu, H. Chen, and J. Wu, "Asymmetric supervised consistent and specific hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 30, pp. 986–1000, 2021.
[46] C. Li, C. Deng, N. Li, W. Liu, X. Gao, and D. Tao, "Self-supervised adversarial hashing networks for cross-modal retrieval," in *Proc. CVPR*, 2018, pp. 4242–4251.
[47] Z. Yang, O. I. Raymond, W. Huang, Z. Liao, L. Zhu, and J. Long, "Scalable deep asymmetric hashing via unequal-dimensional embeddings for image similarity search," *Neurocomputing*, vol. 412, pp. 262–275, Oct. 2020.
[48] X. Liu, G. Yu, C. Domeniconi, J. Wang, Y. Ren, and M. Guo, "Ranking-based deep cross-modal hashing," in *Proc. AAAI*, 2019, pp. 4400–4407.
[49] 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.
[50] T. Yao et al., "Fast discrete cross-modal hashing with semantic consistency," *Neural Netw.*, vol. 125, pp. 142–152, May 2020.
[51] D. Wang, Q. Wang, and X. Gao, "Robust and flexible discrete hashing for cross-modal similarity search," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2703–2715, Oct. 2018.
[52] A. Huang, T. Zhao, and C. Lin, "Multi-view data fusion oriented clustering via nuclear norm minimization," *IEEE Trans. Image Process.*, vol. 29, pp. 9600–9613, 2020.
[53] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang, "Supervised hashing with kernels," in *Proc. CVPR*, 2012, pp. 2074–2081.
[54] C. Da, S. Xu, K. Ding, G. Meng, S. Xiang, and C. Pan, "AMVH: Asymmetric multi-valued hashing," in *Proc. CVPR*, 2017, pp. 898–906.
[55] A. Gordo, F. Perronnin, Y. Gong, and S. Lazebnik, "Asymmetric distances for binary Embeddings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 33–47, Jan. 2014.
[56] Z. Zhang et al., "Scalable supervised asymmetric hashing with semantic and latent factor embedding," *IEEE Trans. Image Process.*, vol. 28, pp. 4803–4818, 2019.
[57] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. CVPR*, 2015, pp. 37–45.
[58] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 490–496, Feb. 2018.
[59] X. Liu, X. Nie, Q. Zhou, L. Nie, and Y. Yin, "Model optimization boosting framework for linear model hash learning," *IEEE Trans. Image Process.*, vol. 29, pp. 4254–4268, 2020.
[60] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. NIPS*, 2011, pp. 612–620.
[61] W. Liu, C. Mu, S. Kumar, and S. Chang, "Discrete graph hashing," in *Proc. NIPS*, 2014, pp. 3419–3427.
[62] J. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[63] M. Kafai and K. Eshghi, "CROification: Accurate kernel classification with the efficiency of sparse linear SVM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 34–48, Jan. 2019.
[64] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[65] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proc. ACM Int. Conf. Multimedia Inf. Retrieval*, 2008, pp. 39–43.
[66] T. 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. CIVR*, 2009, pp. 1–9.
[67] H. J. Escalante et al., "The segmented and annotated IAPR TC-12 benchmark," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 419–428, 2010.
[68] 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, pp. 2494–2507, 2017.
[69] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

Zhan Yang received the Ph.D. degree in computer science and technology from Central South University, Changsha, China, in 2020.

He is currently a Lecturer with the Big Data Institute, School of Computer Science and Engineering, Central South University. His research interests include multimedia retrieval, computer vision, and pattern recognition.



Xiyin Deng received the B.S. degree in software engineering from Central South University, Changsha, China, in 2000, where she is currently pursuing the M.S. degree.

Her major research interests include information retrieval, cross-modal retrieval, and image processing.

Lin Guo received the B.Sc. degree in electronic information engineering, the M.Sc. degree in software engineering, and the Ph.D. degree in computer science from Central South University, Changsha, China, in 2012, 2016, and 2020, respectively.

From 2018 to 2020, he was a Visiting Scholar with the University of Victoria, Victoria, BC, Canada, with Prof. J. Pan. He is currently a Postdoctoral Researcher with the Big Data Institute, Central South University. His research interests include cross-modal retrieval, stochastic optimization, and computer vision.

Jun Long (Member, IEEE) received the B.S. degree in software engineering from Changsha Railway Campus, Changsha, China, in 1996, and the M.S. and Ph.D. degrees in computer science from Central South University, Changsha, in 2004 and 2011, respectively.

He is currently a Professor with the Big Data Institute, Central South University. His major research interests include machine learning, computer vision, and natural language processing.