Online Multi-modal Hashing with Dynamic Query-adaption

Xu Lu[1], Lei Zhu[1], Zhiyong Cheng[2], Liqiang Nie[3], Huaxiang Zhang[1]
[1] School of Information Science and Engineering, Shandong Normal University
[2] Shandong Computer Science Center (National Supercomputer Center in Jinan),
Qilu University of Technology (Shandong Academy of Sciences)
[3] School of Computer Science and Technology, Shandong University
leizhu0608@gmail.com,huaxzhang@hotmail.com

ABSTRACT

Multi-modal hashing is an effective technique to support large-scale multimedia retrieval, due to its capability of encoding heterogeneous multi-modal features into compact and similarity-preserving binary codes. Although great progress has been achieved so far, existing methods still suffer from several problems, including: 1) All existing methods simply adopt fixed modality combination weights in online hashing process to generate the query hash codes. This strategy cannot adaptively capture the variations of different queries. 2) They either suffer from insufficient semantics (for unsupervised methods) or require high computation and storage cost (for the supervised methods, which rely on pair-wise semantic matrix). 3) They solve the hash codes with relaxed optimization strategy or bit-by-bit discrete optimization, which results in significant quantization loss or consumes considerable computation time. To address the above limitations, in this paper, we propose an *Online Multi-modal Hashing with Dynamic Query-adaption* (OMH-DQ) method in a novel fashion. Specifically, a self-weighted fusion strategy is designed to adaptively preserve the multi-modal feature information into hash codes by exploiting their complementarity. The hash codes are learned with the supervision of pair-wise semantic labels to enhance their discriminative capability, while avoiding the challenging symmetric similarity matrix factorization. Under such learning framework, the binary hash codes can be directly obtained with efficient operations and without quantization errors. Accordingly, our method can benefit from the semantic labels, and simultaneously, avoid the high computation complexity. Moreover, to accurately capture the query variations, at the online retrieval stage, we design a parameter-free online hashing module which can adaptively learn the query hash codes according to the dynamic query contents. Extensive experiments demonstrate the state-of-the-art performance of the proposed approach from various aspects.

CCS CONCEPTS

• **Information systems → Information systems applications**; *Multimedia information systems*; *Nearest-neighbor search.*

KEYWORDS

Online multi-modal hashing; Efficient discrete optimization; Dynamic query-adaption; Self-weighted

ACM Reference Format:
Xu Lu, Lei Zhu, Zhiyong Cheng, Liqiang Nie, Huaxiang Zhang. 2019. Online Multi-modal Hashing with Dynamic Query-adaption. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’19), July 21–25, 2019, Paris, France.* ACM, NY, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331217

1 INTRODUCTION

Hashing [11, 24, 31] encodes the high-dimensional data into binary hashing codes and preserves data similarity in the low-dimensional Hamming space, which can greatly accelerate the large-scale multimedia retrieval process and also significantly save the storage cost. Currently, most existing hashing work make efforts on uni-modal hashing [2, 5, 14, 17, 37, 38] and cross-modal hashing [1, 4, 10, 22, 26, 29], and less attention has been paid to the multi-modal hashing [7, 12, 21, 25, 30].

In multimedia retrieval, it is common that both the targeted query and database data are described by heterogeneous multi-modal features. Each modality characterizes the instance from a different view and possess its own characteristics [32, 33, 39]. Therefore, it is necessary to combine features from different modalities to comprehensively represent query and database data for accurate retrieval. However, multi-modal feature representation usually resulted in high dimensional features and high-cost matching problems. To support efficient large-scale multimedia retrieval, the multi-modal hashing method , which considers combining the multi-modal features at both the offline hash code learning and the online query hashing stages, has been studied [7, 12, 21, 30]. This technique is different from the uni-modal and cross-modal hashing where only one of the modalities is provided at query. One may argue to extend uni-modal hashing to multi-modal hashing by simply substituting the input with the concatenated unified vector comprising of multi-modal features. However, this simple extension may fail to fully exploit the complementary of different modality features. Based on the motivation, several multi-modal hashing methods have been proposed with various unsupervised and supervised learning approaches.

Although great progress has been achieved so far [13, 18, 19], they still suffer from several fatal problems:

Lei Zhu and Huaxiang Zhang are corresponding authors.

SIGIR ’19, July 21–25, 2019, Paris, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331217

1) *Fixed weights.* The adopted weighting scheme is essential to multi-modal feature combining for multi-modal hash code learning. Nevertheless, all existing methods [7, 12, 13, 18, 19, 21, 28] generate hash codes for various queries with a fixed weight vector, which is learned in the offline hash code learning stage. The fixed weights obviously cannot capture the variations of dynamic queries.

2) *Limited semantics or scalability.* Most multi-modal hashing methods [7, 12, 19, 21] are unsupervised. They have not exploited any semantic labels, which have demonstrated to be very useful on enhancing the discriminative capability and thus significantly improve the performance in uni-modal and cross-modal hashing. As a result, the hash codes learned by unsupervised methods only contain limited semantics. On the other hand, although existing supervised multi-modal hashing methods can preserve the semantics, they often consume considerable computation and storage. For example, Compact Kernel Hashing with Multiple Features (MFKH) [13] and Discrete Multi-view Hashing (DMVH) [28] need to construct semantic graph, which is an $n \times n$ matrix reflecting the similarities between pairs of instances, to model the semantic correlation for hashing learning, resulting in high time complexity and storage cost.

3) *Optimization challenge.* The binary constraint on hash codes leads to an NP-hard combinatorial optimization problem. Existing multi-modal hashing methods adopt two kinds of optimization strategies. The first is a two-step relaxing+rounding relaxed optimization strategy [7, 12, 21, 30], which relaxes the discrete constraints and then obtains binary codes by simply thresholding. As indicated by recent literature [18, 19], this simplified hash optimization strategy could bring significant quantization errors and thus lead to sub-optimal solutions. The other one is the bit-by-bit hash optimization based on Discrete Cyclic Coordinate Descend (DCC) [19, 28]. This strategy is still time-consuming since only one bit is optimized in each step and thus learning all hashing bits requires lots of iterations.

In light of the above analysis, in this paper, we propose an *Online Multi-modal Hashing with Dynamic Query-adaption* (OMH-DQ) method to address all these problems. Our method learns discriminative hash codes with pair-wise semantic supervision and efficient (both computation and storage) discrete optimization. Moreover, OMH-DQ adaptively generates query hash codes with online query hashing module to capture the query variations. The main contributions of this paper are:

- Instead of adopting the fixed modality combination weights to generate online query hash codes as existing multi-modal hashing methods, we propose a query-adaptive and self-weighted online hashing module to accurately capture the variations of different queries. Moreover, the online module is parameter-free. It could avoid time-consuming and inaccurate parameter adjustment in the unsupervised query hashing process.
- We develop an efficient hash code learning module to simultaneously correlate the learned hash codes with low-level data distribution and high-level semantics. In particular, this design not only enhances the discriminative capability of hash codes with pair-wise semantics, but also avoids the challenging symmetric semantic matrix factorization and storage cost of semantic graph.
- An efficient discrete optimization method is proposed to directly solve the binary hash codes without relaxing quantization errors. The hash codes are learned in a fast mode with simple operation, achieving high learning efficiency and retrieval accuracy. Experimental results on public multimedia retrieval datasets demonstrate the state-of-the-art performance of the proposed method from various aspects.

Table 1: Key characteristics of representative multi-modal hashing methods and the proposed OMH-DQ (n is the number of training samples). The complexity includes both computation and space complexity.

Methods	Weight	Learning	Optimization	Complexity
CHMIS [30]	fixed	unsupervised	relaxation	$O(n^2)$
MFH [21]	fixed	unsupervised	relaxation	$O(n^2)$
MVAGH [7]	fixed	unsupervised	relaxation	$O(n^2)$
MAH [12]	fixed	unsupervised	relaxation	$O(n^2)$
MVLH [19]	fixed	unsupervised	discrete	$O(n)$
MVDH [18]	fixed	unsupervised	discrete	$O(n^2)$
MFKH [13]	fixed	supervised	relaxation	$O(n^2)$
DMVH [28]	fixed	supervised	discrete	$O(n^2)$
Ours	**Query adaptive**	**supervised**	**discrete**	$\mathbf{O(n)}$

The rest of this paper is arranged as follows. Section 2 reviews the related work of existing multi-modal hashing methods. The details of the proposed method are introduced in Section 3. Section 4 presents the experiments. Finally, Section 5 concludes the paper.

2 RELATED WORK

Multi-modal hashing takes advantage of multi-modal features to learn hash codes[1]. Most multi-modal hashing methods learn the hash codes with the unsupervised learning paradigm. They basically model the sample relations in each modality with graph, based on which the hashing learning is performed. Composite Hashing with Multiple Information Sources (CHMIS) [30] is one of the pioneering methods of this kind. It integrates multi-modal feature distribution in multiple pre-constructed graphs into the binary hash codes by adjusting the weight of each modality-specific graph to maximize the hashing performance. Multiple Feature Hashing (MFH) [21] considers to learn the hashing codes by preserving the local structures with graphs of all multi-modal features. Multi-view Anchor Graph Hashing (MVAGH) [7] solves nonlinear integrated binary codes by a subset of eigenvectors of an averaged multi-view graph. It can improve learning efficiency with a low-rank form of the averaged similarity matrix induced by multi-view anchor graph. Multi-view Alignment Hashing (MAH) [12] combines multi-modal features to learn hash codes and meanwhile removes feature redundancies. The authors formulate a multi-graph regularized nonnegative matrix factorization framework, where hash codes are learned by uncovering the hidden semantics and capturing the joint probability distribution of data. The above four multi-modal hashing methods adopt the relaxed optimization strategy to compute hash codes. This strategy first relaxes the discrete constraints and then calculates binary codes by a simple thresholding method. As

[1] Many approaches are termed as multi-view hashing. They can be considered as multi-modal hashing by substituting the multi-view features with heterogeneous modality features.

indicated by the recent literature [17], this relaxed hash optimization strategy could bring significant quantization errors and lead to sub-optimal solutions.

To address the problems, Multi-view Latent Hashing (MVLH) [19] and Multi-view Discrete Hashing (MVDH) [18] propose to directly learn discrete multi-modal hash codes. MVLH performs discrete hash learning in a latent kernel feature space shared by multiple views. MVDH adopts matrix factorization to directly generate discrete hash codes as the latent representations of multiple views. Due to the independence on the semantic labels, all these unsupervised multi-modal hashing still suffer from the limited discriminative capability problem.

A few supervised multi-modal hashing methods have been proposed to exploit the semantic labels as supervision. Compact Kernel Hashing with Multiple Features (MFKH) [13] formulates the multiple feature hashing as a similarity preserving problem with optimal linearly-combined multiple kernels. Hash codes in MFKH are learned with relaxed optimization strategy. Discrete Multi-view Hashing (DMVH) [28] is a discrete supervised multi-modal hashing method. It exploits the discriminative semantic labels to directly learn discrete hash codes supervised with the pre-constructed semantic matrix. It preserves both the local similarity structure and the semantic similarity of data points into the hash codes. With the supervision of semantic labels, these supervised multi-modal hashing methods are reported to achieve superior performance than unsupervised ones.

All existing multi-modal hashing methods simply adopt fixed modality combination weights learned from offline hash code learning to generate all query hash codes. The fixed weights cannot adaptively capture the variations of dynamic out-of-sample queries. Furthermore, most existing methods construct graph to model similarity or semantic correlation for hashing learning. The graphs require considerable computation and storage cost. This disadvantage limits their scalability on large-scale multimedia retrieval. The key characteristics of representative hashing methods and the proposed method are summarized in Table 1.

3 THE PROPOSED METHODOLOGY

Throughout this paper, we utilize uppercase letters to represent matrices and lowercase letters to represent vectors. Suppose that $\mathbf{O} = \{\mathbf{o}_i\}_{i=1}^{n}$ is the training dataset, which contains n training samples represented with M different modality features. The m-th modality feature is $\{\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, ..., \mathbf{x}_n^{(m)}] \in \mathbb{R}^{d_m \times n}\}_{m=1}^{M}$, where d_m is the dimensionality of the m-th modality, M is the number of modalities. Different modalities of one sample $\mathbf{o}_i$ share the same semantic, that is, they belong to the same category. $\mathbf{S} \in \{-1, 1\}^{n \times n}$ is a pair-wise semantic matrix. Our method is to learn hash code $\mathbf{B} \in \{-1, 1\}^{r \times n}$ to represent multimedia instances, where r is the length of hash code. $\Delta_n \overset{def}{=} \{x \in \mathbb{R}^n | x_i \geq 0, \mathbf{1}^{\mathsf{T}} x = 1\}$ is the probabilistic simplex. The basic framework of the proposed OMH-DQ is illustrated in Figure 1.

3.1 Discriminative Multi-modal Projection Learning

3.1.1 Consensus Multi-modal Feature Mapping. To learn effective binary projection for multi-modal data, the projected features should comprehensively preserve multi-modal information. The first task is to model the multi-modal data information so that the binary projection learning can be performed. Most existing methods [7, 12, 21, 30] construct graph to accomplish this task. The graph construction process costs $O(n^2)$ computation and storage complexity, which is practically unacceptable for large-scale multimedia retrieval. In this paper, we propose an efficient consensus multi-modal feature mapping to reduce the complexity to $O(n)$.

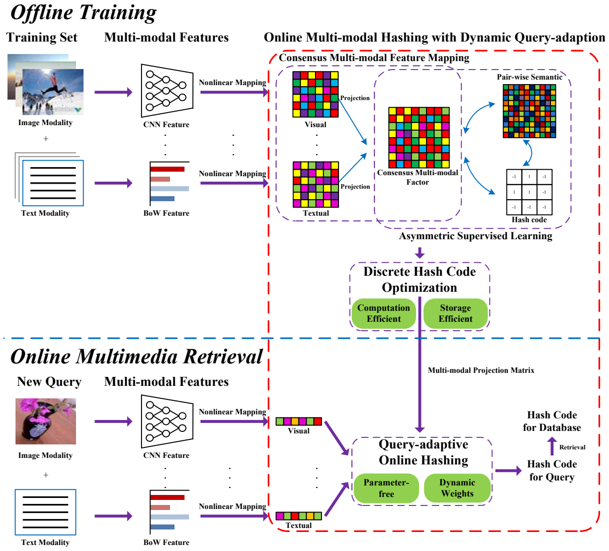

Figure 1: The basic framework of the proposed OMH-DQ.

Given the m-th modality feature $\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, ..., \mathbf{x}_n^{(m)}] \in \mathbb{R}^{d_m \times n}$, we first obtain the nonlinearly transformed representation $\varphi(\mathbf{x}_i^{(m)})$ as $[\exp(\frac{\|\mathbf{x}_i^{(m)} - \mathbf{a}_1^{(m)}\|_F^2}{2\sigma_m^2}), ..., \exp(\frac{\|\mathbf{x}_i^{(m)} - \mathbf{a}_p^{(m)}\|_F^2}{2\sigma_m^2})]^{\mathsf{T}}$ where $\{\mathbf{a}_1^{(m)}, ..., \mathbf{a}_p^{(m)}\}$ are p anchors that are randomly selected from the training samples in the m-th modality, σ_m is the Gaussian kernel parameter. $\varphi(\mathbf{X}^{(m)}) = [\varphi(\mathbf{x}_1^{(m)}), ..., \varphi(\mathbf{x}_n^{(m)})] \in \mathbb{R}^{p \times n}$ preserves the modality-specific sample correlations by simply characterizing the correlations between the sample and certain anchors. The computation complexity of this part is $O(mnp)$.

In multimedia retrieval, the heterogeneous modality gap and inter-modality redundancy in multi-modal data are detrimental to hashing learning. In this paper, we propose to collaboratively project the nonlinearly transformed representation $\varphi(\mathbf{X}^{(m)})|_{m=1}^{M}$ into a consensus multi-modal factor $\mathbf{H} \in \mathbb{R}^{r \times n}$ (r is the hash code length) as the hash code learning basis. Motivated by these considerations, we can formulate this part as

$$\begin{aligned} &\min_{\mu^{(m)}, \mathbf{W}^{(m)}, \mathbf{H}} \sum_{m=1}^{M} \mu^{(m)} \|\mathbf{H} - \mathbf{W}^{(m)} \varphi(\mathbf{X}^{(m)})\|_F^2 + \zeta \|\boldsymbol{\mu}\|_F^2 \\ &s.t. \ \boldsymbol{\mu} = [\mu^{(1)}, \mu^{(2)}, ..., \mu^{(M)}]^{\mathsf{T}}, \boldsymbol{\mu} \in \Delta_M \end{aligned} \tag{1}$$

where $\mathbf{W}^{(m)} \in \mathbb{R}^{r \times p}, m = 1, .., M$ is the projection matrix of the m-th modality, $\mathbf{H} \in \mathbb{R}^{r \times n}$ is the consensus multi-modal factor, $\mu^{(m)}$ is the weight of the m-th modality and it measures the importance of modality feature. With weight setting, complementarity of multi-modal features can be exploited properly. Similar to MVDH [18], we formulate a second term in Eq.(1) to smoothen the weight distribution. $\zeta > 0$ is a hyper-parameter that plays the balance between the regularization terms. Specifically, without this regularization term (or $\zeta \rightarrow 0$), the weight of the best modality with the minimum reconstruction loss will be assigned to 1 and that of other modalities will be 0. On the other hand, when $\zeta \rightarrow \infty$, an equal weight will be assigned to every modality. Under such circumstance, the effects of weights on exploring the complementarity of multi-modal features are missing. Therefore, it is advisable for this parameter-weighted hash learning to involve an additional parameter ζ, whose best value is confirmed to be data related. In practice, it means that more time will be consumed on parameter adjustment in the offline hash learning process. Furthermore, the parameter adjustment requirement is also contradictory to the fact that we cannot manually set a proper parameter for each query in the online retrieval process.

To address the problem, in this paper, we introduce a virtual weight and propose a new consensus multi-modal feature mapping which can achieve the same goal as Eq.(1) without additional hyper-parameters. Specifically, we formulate this part as

$$\min_{\mathbf{W}^{(m)},\mathbf{H}} \sum_{m=1}^{M} \|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}} \tag{2}$$

where $\|\cdot\|_{\mathsf{F}}$ is the Frobenius norm of the matrix. We can derive the following theorem.

Theorem 3.1. *Eq.(2) is equivalent to*

$$\min_{\mu\in\Delta_M,\boldsymbol{W}^{(m)},\boldsymbol{H}} \sum_{m=1}^{M} \frac{1}{\mu^{(m)}}\|\boldsymbol{H}-\boldsymbol{W}^{(m)}\varphi(\boldsymbol{X}^{(m)})\|_{\mathsf{F}}^2 \tag{3}$$

Proof. Note that,

$$\sum_{m=1}^{M} \frac{1}{\mu^{(m)}}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2 \overset{(a)}{=} (\sum_{m=1}^{M} \frac{1}{\mu^{(m)}}\|\mathbf{H}-$$
$$\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2)(\sum_{m=1}^{M}\mu^{(m)}) \overset{(b)}{\geq} (\sum_{m=1}^{M}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}})^2$$

where (a) holds since $\sum_{m=1}^{M}\mu^{(m)} = 1$ and (b) holds according to the Cauchy-Schwarz inequality. This equation indicates

$$(\sum_{m=1}^{M}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}})^2 = \min_{\mu\in\Delta_M}\sum_{m=1}^{M}\frac{1}{\mu^{(m)}}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2$$

It is easy to derive

$$\min_{\mathbf{W}^{(m)},\mathbf{H}}\sum_{m=1}^{M}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}} \Leftrightarrow \min_{\mathbf{W}^{(m)},\mathbf{H}}(\sum_{m=1}^{M}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}})^2$$
$$\Leftrightarrow \min_{\mu\in\Delta_M,\mathbf{W}^{(m)},\mathbf{H}}\sum_{m=1}^{M}\frac{1}{\mu^{(m)}}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2$$

which completes the proof. □

As shown in Eq.(3), $\frac{1}{\mu^{(m)}}$ can be considered as a virtual weight which acts as the function of real weight. If the m-th modality is discriminative, then the value of $\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}$ should be small, and thus the corresponding virtual weight $\frac{1}{\mu^{(m)}}$ of the m-th modality is large. Similarly, a less discriminative modality will be assigned a small weight.

3.1.2 Asymmetric Supervised Learning. The projection matrices learned from Eq.(2) only exploit low-level multi-modal information while have not exploited any semantic labels which have already demonstrated to be able to impressively improve the performance on uni-modal hashing [17, 34] and cross-modal hashing [10, 23]. In this paper, we further leverage explicit semantic labels to guide the projection learning process and thus to enhance the discriminative capability of hash codes. Specifically, in this paper, we consider the pair-wise semantic supervision, motivated by its recent success on uni-modal [16, 36] and cross-modal hashing [27, 35].

Let $\mathbf{B} \in \{-1,1\}^{r\times n}$ denote the hash code to be learned. Intuitively, it can be learned by preserving the high-level semantic correlations (described by similarity matrix $\mathbf{S}$) into the hash codes

$$\min_{\mathbf{B}}\|r\mathbf{S}-\mathbf{B}^{\mathsf{T}}\mathbf{B}\|_{\mathsf{F}}^2,\ s.t.\ \mathbf{B}\in\{-1,1\}^{r\times n} \tag{4}$$

where $\mathbf{S}\in\mathbb{R}^{n\times n}$ is the pair-wise similarity matrix, its element in the i-th row and j-th column is defined as

$$s_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same category,} \\ -1, & \text{vice versa} \end{cases}$$

However, directly solving discrete $\mathbf{B}$ in Eq.(4) is very challenging due to the discrete symmetric factorization, not to mention the fast hashing optimization. Furthermore, storing the elements of $\mathbf{S}$ will consume $O(n^2)$, which is unacceptable in large-scale multimedia retrieval. In this paper, to avoid these problems, we develop an asymmetric hashing learning module that transfers semantics from pair-wise semantic matrix $\mathbf{S}$ to hash codes. Specifically, we substitute one of $\mathbf{B}$ with the rotated consensus multi-modal factor $\mathbf{RH}$ ($\mathbf{R}\in\mathbb{R}^{r\times r}$ is rotation matrix) and keep their consistency during the optimization process. The formula is

$$\min_{\mathbf{B},\mathbf{R},\mathbf{H}}\|r\mathbf{S}-\mathbf{B}^{\mathsf{T}}\mathbf{RH}\|_{\mathsf{F}}^2+\beta\|\mathbf{B}-\mathbf{RH}\|_{\mathsf{F}}^2,\ s.t.\ \mathbf{B}\in\{-1,1\}^{r\times n},\mathbf{R}^{\mathsf{T}}\mathbf{R}=\mathbf{I}_r$$

This formulation has two advantages: 1) The symmetric matrix factorization can be obviously avoided. Only one of the decomposed variable is imposed with discrete constraint. The second regularization term can guarantee the acceptable information loss. 2) The learned hash codes can not only reflect the low-level multi-modal data distribution via $\mathbf{H}$, but also involve the high-level semantics in $\mathbf{S}$. As shown below, with the support of asymmetric hashing learning, the hash codes can be learned with a simple $\mathsf{sgn}(\cdot)$ operation instead of bit-by-bit discrete optimization as existing discrete multi-modal hashing methods. In addition, as shown below, the $O(n^2)$ storage cost brought by $\mathbf{S}$ can be reduced to $O(n)$ when representing $\mathbf{S}$ with the label matrix in our approach.

3.1.3 Overall Objective Formulation. By integrating the above two parts into a unified learning framework, we derive the overall objective function of hash code learning in OMH-DQ as

$$\begin{aligned}\min_{\mathbf{W}^{(m)},\mathbf{H},\mathbf{B},\mathbf{R}} &\sum_{m=1}^{M}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}+\alpha\|r\mathbf{S}-\mathbf{B}^{\mathsf{T}}\mathbf{RH}\|_{\mathsf{F}}^2 \\ &+\beta\|\mathbf{B}-\mathbf{RH}\|_{\mathsf{F}}^2+\delta\sum_{m=1}^{M}\|\mathbf{W}^{(m)}\|_{\mathsf{F}}^2 \\ &s.t.\ \mathbf{B}\in\{-1,1\}^{r\times n},\ \mathbf{R}^{\mathsf{T}}\mathbf{R}=\mathbf{I}_r\end{aligned} \tag{5}$$

where α, β, and δ are balance parameters. The first term performs consensus multi-modal feature mapping to combine multiple modalities, bridge the heterogeneous modality gap, and avoid inter-modality redundancy. The second and the third terms perform asymmetric supervised hashing learning. The last term is a regularization term to avoid over-fitting.

3.1.4 Efficient Discrete Hash Code Optimization. Solving hash codes is actually an NP-hard problem due to the discrete constraint. It is always a challenging task from the birth of this technique. Most existing multi-modal hashing methods [7, 12, 13, 21, 30] adopt two-step relaxing+rounding optimization strategy. They basically solve the relaxed continuous solutions first and then calculate the binary hash codes by thresholding. However, this simplified strategy will lead to significant quantization loss. Although there exist discrete multi-modal hash methods [18], they learn the hash codes bit-by-bit with DCC, which is still time-consuming.

In this paper, with the support of objective formulation, we propose to directly learn the discrete multi-modal hash code. Moreover, different from existing multi-modal hashing methods, we avoid explicitly computing the similarity matrix $\mathbf{S}$, which can achieve linear computation and storage efficiency.

By Theorem 3.1, Eq.(5) is equivalent to the following problem

$$\begin{aligned}\min_{\mu^{(m)},\mathbf{W}^{(m)},\mathbf{R},\mathbf{H},\mathbf{B}} & \sum_{m=1}^{M}\frac{1}{\mu^{(m)}}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2 \\ & +\alpha\|r\mathbf{S}-\mathbf{B}^{\top}\mathbf{R}\mathbf{H}\|_{\mathsf{F}}^2+\beta\|\mathbf{B}-\mathbf{R}\mathbf{H}\|_{\mathsf{F}}^2+\delta\sum_{m=1}^{M}\|\mathbf{W}^{(m)}\|_{\mathsf{F}}^2 \\ & s.t.\ \mathbf{B}\in\{-1,1\}^{r\times n},\ \mathbf{R}^{\top}\mathbf{R}=\mathbf{I}_r,\ \boldsymbol{\mu}\in\Delta_M\end{aligned} \tag{6}$$

Besides, we propose a new and effective optimization algorithm based on augmented Lagrangian multiplier (ALM) [9, 15] to solve the problem in Eq.(6). Our idea is to introduce auxiliary variables to separate constraints, and transform the objective function to an equivalent one that can be tackled more easily. Formally, we introduce two auxiliary variables $\mathbf{Z}_R$ and $\mathbf{Z}_B$, and set $\mathbf{Z}_R=\mathbf{R}$ and $\mathbf{Z}_B=\mathbf{B}$. Eq.(6) is transformed as

$$\begin{aligned}&\min\sum_{m=1}^{M}\frac{1}{\mu^{(m)}}\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2+\alpha\|r\mathbf{S}-\mathbf{B}^{\top}\mathbf{R}\mathbf{H}\|_{\mathsf{F}}^2+\beta\|\mathbf{B}-\mathbf{R}\mathbf{H}\|_{\mathsf{F}}^2+\\ &\delta\sum_{m=1}^{M}\|\mathbf{W}^{(m)}\|_{\mathsf{F}}^2+\frac{\lambda}{2}(\|\mathbf{R}-\mathbf{Z}_R+\frac{\mathbf{G}_R}{\lambda}\|_{\mathsf{F}}^2+\|\mathbf{B}-\mathbf{Z}_B+\frac{\mathbf{G}_B}{\lambda}\|_{\mathsf{F}}^2)\\ &s.t.\ \mathbf{B},\mathbf{Z}_B\in\{-1,1\}^{r\times n},\ \mathbf{R}^{\top}\mathbf{R}=\mathbf{I}_r,\ \mathbf{Z}_R^{\top}\mathbf{Z}_R=\mathbf{I}_r,\ \boldsymbol{\mu}\in\Delta_M\end{aligned} \tag{7}$$

where $\mathbf{G}_R\in\mathbb{R}^{r\times r}$ and $\mathbf{G}_B\in\mathbb{R}^{r\times n}$ measure the difference between the target and auxiliary variables, $\lambda>0$ adjusts the balance between terms. Specifically, we drive the following iterative optimization steps to solve Eq.(7).

Step 1: By fixing other variables, update $\mu^{(m)}$. For convenience, we denote $\|\mathbf{H}-\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})\|_{\mathsf{F}}$ by $h^{(m)}$. The original problem can be written as:

$$\min_{\mu^{(m)}\geq 0,\mathbf{1}^{\top}\boldsymbol{\mu}=1}\sum_{m=1}^{M}\frac{h^{(m)2}}{\mu^{(m)}} \tag{8}$$

which combining with Cauchy-Schwarz inequality gives

$$\sum_{m=1}^{M}\frac{h^{(m)2}}{\mu^{(m)}}\overset{(a)}{=}(\sum_{m=1}^{M}\frac{h^{(m)2}}{\mu^{(m)}})(\sum_{m=1}^{M}\mu^{(m)})\overset{(b)}{\geq}(\sum_{m=1}^{M}h^{(m)})^2$$

where (a) holds since $\mathbf{1}^{\top}\boldsymbol{\mu}=1$ and the equality in (b) holds when $\sqrt{\mu^{(m)}}\propto\frac{h^{(m)}}{\sqrt{\mu^{(m)}}}$. Since the right-hand side of Eq.(3.1.4) is constant, the optimal $\mu^{(m)}$ in Eq.(8) can be obtained by

$$\mu^{(m)}=\frac{h^{(m)}}{\sum_{m=1}^{M}h^{(m)}} \tag{9}$$

Step 2: By fixing other variables, update $\mathbf{W}^{(m)}$. We set the derivative of objective function with respect to $\mathbf{W}^{(m)}$ to zero, and then we have

$$\mathbf{W}^{(m)}=(\frac{1}{\mu^{(m)}}\mathbf{H}\varphi^{\top}(\mathbf{X}^{(m)}))(\frac{1}{\mu^{(m)}}\varphi(\mathbf{X}^{(m)})\varphi^{\top}(\mathbf{X}^{(m)})+\delta\mathbf{I}_p)^{-1} \tag{10}$$

Step 3: By fixing other variables, update R. The objective function with respect to $\mathbf{R}$ can be represented as

$$\min_{\mathbf{R}^{\top}\mathbf{R}=\mathbf{I}_r}\mathtt{tr}(-2\beta\mathbf{R}^{\top}\mathbf{B}\mathbf{H}^{\top}-2\alpha r\mathbf{R}^{\top}\mathbf{B}\mathbf{S}\mathbf{H}^{\top}+\alpha\mathbf{R}^{\top}\mathbf{B}\mathbf{B}^{\top}\mathbf{R}\mathbf{H}\mathbf{H}^{\top}-\lambda\mathbf{R}^{\top}(\mathbf{Z}_R-\frac{\mathbf{G}_R}{\lambda}))$$

We substitute $\mathbf{R}^{\top}\mathbf{B}\mathbf{B}^{\top}\mathbf{R}\mathbf{H}\mathbf{H}^{\top}$ with $\mathbf{R}^{\top}\mathbf{B}\mathbf{B}^{\top}\mathbf{Z}_{\mathsf{R}}\mathbf{H}\mathbf{H}^{\top}$ and the above equation can be transformed with the following equivalent form

$$\max_{\mathbf{R}^{\top}\mathbf{R}=\mathbf{I}_r}\mathtt{tr}(\mathbf{R}^{\top}\mathbf{C}) \tag{11}$$

where $\mathbf{C}=\beta\mathbf{B}\mathbf{H}^{\top}+\alpha r\mathbf{B}\mathbf{S}\mathbf{H}^{\top}-\alpha\mathbf{B}\mathbf{B}^{\top}\mathbf{Z}_R\mathbf{H}\mathbf{H}^{\top}+\lambda\mathbf{Z}_R-\mathbf{G}_R$. The optimal $\mathbf{R}$ is defined as $\mathbf{R}=\mathbf{P}\mathbf{Q}^{\top}$, where $\mathbf{P}$ and $\mathbf{Q}$ are comprised of left-singular and right-singular vectors of $\mathbf{C}$ respectively [40].

Note that, the $\mathbf{S}\in\mathbb{R}^{n\times n}$ is included in the term $\mathbf{B}\mathbf{S}\mathbf{H}^{\top}$ when updating $\mathbf{R}$. If we compute $\mathbf{S}$ explicitly, the computational complexity is $O(n^2)$. In this paper, we utilize $c\times n$ matrix $\tilde{\mathbf{L}}$ (c is the number of semantic categories) to store the label information instead of directly calculating $\mathbf{S}$, and can reduce the computational complexity to $O(n)$. Let $\tilde{\mathbf{L}}_{ki}=\frac{l_{ki}}{\|l_i\|_2}$, as the element at the k-th row and the i-th column in the matrix $\tilde{\mathbf{L}}$. Then we can get the similarity matrix $\tilde{\mathbf{S}}=\tilde{\mathbf{L}}^{\top}\tilde{\mathbf{L}}$. The semantic similarity matrix $\mathbf{S}$ can be calculated as

$$\mathbf{S}=2\tilde{\mathbf{S}}-\mathbf{E}=2\tilde{\mathbf{L}}^{\top}\tilde{\mathbf{L}}-\mathbf{1}_n\mathbf{1}_n^{\top} \tag{12}$$

where $\mathbf{1}_n$ is an all-one column vector with length n, and $\mathbf{E}$ is a matrix with all elements as 1. Then we can get

$$\mathbf{B}\mathbf{S}\mathbf{H}^{\top}=2(\mathbf{B}\mathbf{L}^{\top})(\mathbf{H}\mathbf{L}^{\top})^{\top}-(\mathbf{B}\mathbf{1}_n)(\mathbf{H}\mathbf{1}_n)^{\top}$$

Thus the calculation of $\mathbf{C}$ can be transformed as

$$\mathbf{C}=\beta\mathbf{B}\mathbf{H}^{\top}+\alpha r\mathbf{B}\mathbf{S}\mathbf{H}^{\top}=\beta\mathbf{B}\mathbf{H}^{\top}+2\alpha r(\mathbf{B}\mathbf{L}^{\top})(\mathbf{H}\mathbf{L}^{\top})^{\top}+2\alpha r(\mathbf{B}\mathbf{1}_n)(\mathbf{H}\mathbf{1}_n)^{\top}$$

which consumes $O(n)$.

Step 4: By fixing other variables, update H. We set the derivative of objective function with respect to $\mathbf{H}$ to zero, and then we have

$$\mathbf{H}=\frac{\sum_{m=1}^{M}\frac{1}{\mu^{(m)}}\mathbf{W}^{(m)}\varphi(\mathbf{X}^{(m)})+\beta\mathbf{R}^{\top}\mathbf{B}+\alpha r\mathbf{R}^{\top}\mathbf{B}\mathbf{S}}{\sum_{m=1}^{M}\frac{1}{\mu^{(m)}}+\alpha+\beta} \tag{13}$$

where $\mathbf{S}$ is also transformed using Eq.(12), then we have

$$\mathbf{R}^{\top}\mathbf{B}\mathbf{S}=\mathbf{B}\mathbf{S}\mathbf{R}^{\top}=2(\mathbf{B}\tilde{\mathbf{L}}^{\top})(\mathbf{R}\tilde{\mathbf{L}}^{\top})^{\top}+(\mathbf{B}\mathbf{1}_n)(\mathbf{R}\mathbf{1}_n)^{\top}$$

The time complexity of computing $\mathbf{R}^{\top}\mathbf{B}\mathbf{S}$ is reduced to $O(n)$.

Step 5: By fixing other variables, update B. The objective function with respect to $\mathbf{B}$ can be presented as

$$\min_{\mathbf{B}\in\{-1,1\}^{r\times n}}\mathtt{tr}(-2\alpha r\mathbf{B}^{\top}\mathbf{R}\mathbf{H}\mathbf{S}+\alpha\mathbf{B}^{\top}\mathbf{R}\mathbf{H}\mathbf{H}^{\top}\mathbf{R}^{\top}\mathbf{B}-2\beta\mathbf{B}^{\top}\mathbf{R}\mathbf{H}-\lambda\mathbf{B}^{\top}\mathbf{Z}_B+\mathbf{G}_B)$$

We substitute $\mathbf{B}^\top\mathbf{RHH}^\top\mathbf{R}^\top\mathbf{B}$ with $\mathbf{B}^\top\mathbf{RHH}^\top\mathbf{R}^\top\mathbf{Z}_B$. Thus the above equation is equivalent to

$$\min_{\mathbf{B}\in\{-1,1\}^{r\times n}} \mathrm{tr}(-\mathbf{B}^\top(2\alpha r\mathbf{RHS}^\top - \alpha\mathbf{RHH}^\top\mathbf{R}^\top\mathbf{Z}_\mathrm{B} + 2\beta\mathbf{RH} + \lambda\mathbf{Z}_B - \mathbf{G}_B))$$

We can obtain the closed solution of $\mathbf{B}$ as

$$\mathbf{B} = \mathrm{sgn}(2\alpha r\mathbf{RHS}^\top - \alpha\mathbf{RHH}^\top\mathbf{R}^\top\mathbf{Z}_B + 2\beta\mathbf{RH} + \lambda\mathbf{Z}_B - \mathbf{G}_B) \tag{14}$$

where $\mathbf{S}$ is also transformed using Eq.(12), then we have

$$\mathbf{RHS}^\top = 2\mathbf{RH}\tilde{\mathbf{L}}^\top\tilde{\mathbf{L}} - \mathbf{RH}\mathbf{1}_n\mathbf{1}_n^\top$$

The time complexity of computing $\mathbf{RHS}^\top$ is reduced to $O(n)$.

Step 6: By fixing other variables, update $\mathbf{Z}_B$. We can obtain the closed solution of $\mathbf{Z}_B$ as

$$\mathbf{Z}_B = \mathrm{sgn}(-\alpha\mathbf{RHH}^\top\mathbf{R}^\top\mathbf{B} + \lambda\mathbf{B} + \mathbf{G}_B) \tag{15}$$

Step 7: By fixing other variables, update $\mathbf{Z}_R$. The objective function respect to $\mathbf{Z}_R$ can be represented as

$$\max_{\mathbf{Z}_R^\top\mathbf{Z}_R=\mathbf{I}_r} \mathrm{tr}(\mathbf{Z}_R^\top\mathbf{C}_{zr}) \tag{16}$$

where $\mathbf{C} = -\alpha\mathbf{BB}^\top\mathbf{R}^\top\mathbf{HH}^\top + \lambda\mathbf{R} + \mathbf{G}_R$. The optimal $\mathbf{Z}_R$ is defined as $\mathbf{Z}_R = \mathbf{P}_{zr}\mathbf{Q}_{zr}^\top$.

Step 8: By fixing other variables, update $\mathbf{G}_B$ and $\mathbf{G}_R$. The update rules are

$$\begin{aligned}\mathbf{G}_B &= \mathbf{G}_B + \lambda(\mathbf{B} - \mathbf{Z}_B)\\ \mathbf{G}_R &= \mathbf{G}_R + \lambda(\mathbf{R} - \mathbf{Z}_R)\end{aligned} \tag{17}$$

3.2 Query-adaptive Online Hashing with Dynamic Weights

In the online retrieval process, we aim to map the new coming query instances into binary hash codes with the learned hash projection matrix $\{\mathbf{W}^{(m)}\}_{m=1}^M$. All existing multi-modal hashing methods simply adopt equal or fixed weights obtained from offline hash code learning to combine multi-modal features. However, the fixed weights obtained from offline hash code learning cannot capture the variations of the dynamic queries. This motivates us to develop query-adaptive online hashing. More importantly, we should avoid bringing any additional parameter in weighting scheme.

In this paper, with the support of the offline hash code learning framework, we present a parameter-free query-adaptive online hashing with a self-weighting scheme. The weights are actually virtual and the hash codes are iteratively updated online without any parameters by considering the specific query contents. With this design, we could obtain more accurate query hash codes for fast multimedia retrieval while relying less on parameter searching. Specifically, the objective function of our query-adaptive online hashing process is formulated as

$$\min_{\mathbf{B}_q\in\{-1,1\}^{r\times n_q}} \sum_{m=1}^{M} \|\mathbf{B}_q - \mathbf{W}^{(m)}\varphi(\mathbf{X}_q^{(m)})\|_\mathsf{F} \tag{18}$$

where $\mathbf{W}^{(m)} \in \mathbb{R}^{r\times p}$ is the linear projection matrix from Eq.(5), $\varphi(\mathbf{X}_q^{(m)}) \in \mathbb{R}^{p\times n_q}$ is the nonlinearly projected representation of the newly coming query instances, $\mathbf{B}_q \in \{-1,1\}^{r\times n_q}$ is the binary hash code of the newly coming query instances, and n_q is the number of samples of the query set.

As Theorem 3.1, Eq.(18) can be shown to be equivalent to

$$\min_{\mathbf{B}_q\in\{-1,1\}^{r\times n_q},\boldsymbol{\mu}\in\Delta_M} \sum_{m=1}^{M} \frac{1}{\mu_q^{(m)}} \|\mathbf{B}_q - \mathbf{W}^{(m)}\varphi(\mathbf{X}_q^{(m)})\|_\mathsf{F}^2 \tag{19}$$

To solve Eq.(19), we drive the following iterative steps

Step 1: By fixing $\mathbf{B}_q$, update $\mu_q^{(m)}$. As Section 3.1.4, we denote $\|\mathbf{B}_q - \mathbf{W}^{(m)}\varphi(\mathbf{X}_q^{(m)})\|_\mathsf{F}$ by $h_q^{(m)}$. The optimal $\mu_q^{(m)}$ is given by

$$\mu_q^{(m)} = \frac{h_q^{(m)}}{\sum_{m=1}^{M} h_q^{(m)}}$$

Step 2: By fixing $\mu_q^{(m)}$, update $\mathbf{B}_q$. We can obtain the closed solution of $\mathbf{B}_q$ as

$$\mathbf{B}_q = \mathrm{sgn}(\sum_{m=1}^{M} \frac{1}{\mu_q^{(m)}}\mathbf{W}^{(m)}\varphi(\mathbf{X}_q^{(m)}))$$

3.3 Discussions

3.3.1 Complexity Analysis. This section provides the complexity analysis of OMH-DQ. The time complexity of constructing the nonlinear feature mapping $\varphi(\mathbf{X}^{(m)})$ of each modality is $O(np)$. It takes $O(prn)$ for updating $\mu^{(m)}$. The computational complexity of updating $\mathbf{W}^{(m)}$ is $O(prn)$. The computational complexity of updating $\mathbf{R}$ is $O(r^2n)$. Updating $\mathbf{B}$ requires $O(prn)$. The computational complexity of updating $\mathbf{H}$ is $O(rn)$. Then the computational complexity of optimization process is $O(iter \times prn)$, where $iter$ is the number of iterations. This process scales linearly with n. At online multimedia retrieval process, it takes $O(iter \times prn)$ to generate binary hash code for a new query.

In addition, in the discrete optimization process, we avoid explicitly computing the pair-wise similarity matrix $\mathbf{S}$, but substituting it with the expression of $\tilde{\mathbf{L}}$. We successfully reduce the space complexity to $O(n)$. In sum, both the computational and the space complexity of OMH-DQ are linear with the size of the dataset. Our approach is scalable for large-scale multimedia retrieval.

3.3.2 Convergence Analysis. The objective functions Eq.(6) is convex to one variable by fixing the others[2]. Thus, optimizing one variable in each step will lead to a lower or equal value of the objective function. Our iterative updating rules will monotonically decrease the objective function value. After several iterations, the optimization process eventually achieves a local minimum. In addition, in experiments, we will empirically verify the convergence of the proposed OMH-DQ on three benchmark datasets.

4 EXPERIMENTS

4.1 Experimental Configuration

4.1.1 Evaluation Datasets. In this paper, we conduct experiments on three publicly available multimedia retrieval datasets: MIR Flickr [6], NUS-WIDE [3] and MS COCO [8]. These three datasets are widely used for performance evaluation of multi-modal hashing methods [13, 18, 28]. The statistics of the three datasets used in experiments are summarized in Table 2.

[2] For optimizing the hash code in Step 4, we can directly obtain a closed-form solution.

Table 2: General statistics of three datasets.

Dataset	Training Size	Retrieval Size	Query Size	Categories	Visual Modality	Text Modality
MIR Flickr [6]	5,000	17,772	2,243	24	CNN (4,096-D) (by Caffe implementation of VGG Net [20])	BoW (1,386-D)
NUS-WIDE [3]	5,000	193,749	2,085	21	CNN (4,096-D) (by Caffe implementation of VGG Net [20])	BoW (1,000-D)
MS COCO [8]	18,000	82,783	5,981	80	CNN (4,096-D) (by Caffe implementation of VGG Net [20])	BoW (2,000-D)

Table 3: MAP comparison results on MIR Flickr, NUS-WIDE and MS COCO.

Methods	MIR Flickr				NUS-WIDE				MS COCO			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
MFH [21]	0.5795	0.5824	0.5831	0.5836	0.3556	0.3579	0.3629	0.3569	0.3948	0.3966	0.3960	0.3980
MFKH [13]	0.6369	0.6128	0.5985	0.5807	0.4663	0.4323	0.3917	0.3750	0.4216	0.4211	0.4230	0.4229
MAH [12]	0.6488	0.6649	0.6990	0.7114	0.4608	0.4936	0.5371	0.5477	0.3967	0.3943	0.3966	0.3988
MVLH [19]	0.6541	0.6421	0.6044	0.5982	0.4277	0.3966	0.3751	0.3772	0.3993	0.4012	0.4065	0.4099
MVDH [18]	0.6828	0.7210	0.7344	0.7527	0.5083	0.5533	0.5855	0.6022	0.3978	0.3966	0.3977	0.3998
DMVH [28]	0.7231	0.7326	0.7495	0.7641	0.5665	0.5856	0.6063	0.6285	0.4123	0.4288	0.4355	0.4563
Ours	**0.7988**	**0.8047**	**0.8154**	**0.8184**	**0.6378**	**0.6488**	**0.6608**	**0.6744**	**0.5059**	**0.5148**	**0.5263**	**0.5488**

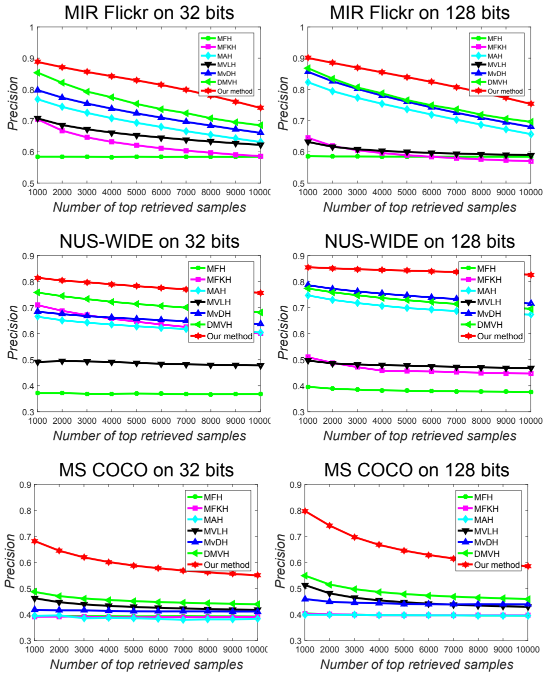

Figure 2: The topK-precision curves on three datasets.

4.1.2 Evaluation Metrics. We adopt two standard evaluation metrics, Mean Average Precision (MAP) [4, 22] and topK-precision [18, 40], to evaluate the multimedia retrieval performance. In the retrieval phase, we adopt Hamming distance to measure the similarity between the binary codes of the query instances and the ones in multimedia database. Two instances are considered to be semantically similar when they share at least one semantic tag. For all the queries, we first calculate their APs and then obtain the average value as MAP as [22]. topK-precision curves reflect the change of precision with respect to the number of top-ranked K instances. For both evaluation protocols, larger value indicates the better retrieval performance.

4.1.3 Evaluation Baselines. In experiments, we compare the proposed method with six state-of-the-art multi-modal hashing methods, including Multiple Feature Hashing (MFH) [21], Multiple Feature Kernel Hashing (MFKH) [13], Multi-modal Alignment Hashing (MAH) [12], Multi-view Latent Hashing (MVLH) [19], Multi-modal Discrete Hashing (MVDH) [18], Discrete Multi-modal Hashing (DMVH) [28]. Those methods have been discussed in Section 2.

4.1.4 Implementation Details. The proposed OMH-DQ has several parameters: α, β, and δ in Eq.(5), and the number of anchors p. In the experiments, the best performance of OMH-DQ is achieved when the number of anchors p is set as 500, 1,200 and 1,000 on MIR Flickr, NUS-WIDE and MS COCO, respectively. α and β are balance parameters to support asymmetric supervised learning, and δ is a regularization parameter to avoid over-fitting. The best performance is achieved when $\{\alpha = 10^{-5}, \beta = 10^{-1}, \gamma = 10^{-3}\}$, $\{\alpha = 10^{1}, \beta = 10^{5}, \gamma = 10^{-5}\}$, and $\{\alpha = 10^{-3}, \beta = 10^{3}, \gamma = 10^{-5}\}$ on MIR Flickr, NUS-WIDE and MS COCO, respectively. We carefully tune the parameters of all the baselines and finally report their best results for performance comparison. On three datasets, we conduct five successive experiments with different randomly partitioned datasets and report the average results. All our experiments are conducted on a workstation with a 3.40 GHz Intel(R) Core(TM) i7-6700 CPU and 64 GB RAM.

4.2 Comparison Results

4.2.1 Retrieval Accuracy Comparison. The MAP values of all compared methods varying with different hash code lengths (16 bits, 32 bits, 64 bits, and 128 bits) on three datasets (MIR Flickr, NUS-WIDE and MS COCO) are presented in Table 3. Their corresponding topK-precision curves with the increasing number of the retrieved samples on three datasets are shown in Fig. 2.

From these results, we easily find that the proposed OMH-DQ consistently outperforms all the compared baselines on three datasets. For example, when the hash code length is fixed as 64 bits, OMH-DQ obtains the MAP of 0.8154 on MIR Flickr, 0.6608 on NUS-WIDE, and 0.5263 on MS COCO, while the second best performance is 0.7495 on MIR Flickr, 0.6063 on NUS-WIDE, and 0.4355 on MS COCO. The topK-precision curves on three datasets demonstrate the similar trend as the MAP results. From these curves, we can find that, with the number of retrieved samples increasing, the precision of OMH-DQ consistently outperforms the baselines by a large margin. These results clearly validate the superiority of OMH-DQ over the baselines. In addition, we observe that the performance of OMH-DQ improves as the code length increases from 16 bits to 128 bits, while that of the baseline methods, such as MFKH and MVLH, degrade

Table 4: Comparison of training and query time (seconds) on MIR Flickr, NUS-WIDE and MS COCO.

Methods	Training time (s)			Query time (s)		
	MIR Flickr	NUS-WIDE	MS COCO	MIR Flickr	NUS-WIDE	MS COCO
MFH [21]	56.91	92.04	591.68	14.48	130.78	**102.95**
MFKH [13]	37.82	56.26	112.84	14.20	**106.14**	148.59
MAH [12]	107.72	149.42	102.98	204.64	644.12	1070.34
MVLH [19]	219.81	1313.39	444.89	15.06	286.05	158.90
MVDH [18]	1774.94	1004.21	7994.75	930.13	3687.26	2041.58
DMVH [28]	314.01	968.59	453.62	14.37	186.74	168.06
OMH-DQ-*linear*	6.46	7.04	34.00	14.72	153.11	116.83
Ours	**1.14**	**2.11**	**16.99**	**10.93**	135.02	103.66

Table 5: MAP comparison with variants of the proposed method on MIR Flickr, NUS-WIDE and MS COCO.

Methods	MIR Flickr				NUS-WIDE				MS COCO			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
OMH-DQ-*img*	0.6538	0.6651	0.6623	0.6599	0.4598	0.4709	0.4733	0.4562	0.3915	0.3918	0.3923	0.3941
OMH-DQ-*txt*	0.6714	0.6811	0.6857	0.6855	0.4160	0.4278	0.4474	0.4528	0.4892	0.4932	0.5005	0.5037
OMH-DQ-*con*	0.6325	0.6336	0.6340	0.6347	0.3631	0.3640	0.3644	0.3648	0.4652	0.4795	0.4822	0.4964
OMH-DQ-*linear*	0.6936	0.7172	0.7183	0.7270	0.4843	0.4916	0.4976	0.5009	0.4846	0.4997	0.5020	0.5066
OMH-DQ-*relax*	0.7718	0.7516	0.6527	0.6232	0.5714	0.6129	0.6317	0.6423	0.4870	0.4946	0.5075	0.5210
Ours	**0.7988**	**0.8047**	**0.8154**	**0.8184**	**0.6378**	**0.6488**	**0.6608**	**0.6744**	**0.5059**	**0.5148**	**0.5263**	**0.5488**

significantly. This result indicates that longer hash codes can bring more discriminative information in OMH-DQ. Finally, we can find that OMH-DQ even obtains better performance with shorter code length than performance of many baselines with longer code length. For example, on MIR Flickr, the MAP 0.7988 obtained by OMH-DQ when the code length is 16 bits is even better than the MAP of 0.7641 obtained by DMVH when the code length is 128 bits.

Four advantages of OMH-DQ bring the improved retrieval precision: 1) OMH-DQ learns the consensus multi-modal factor collaboratively from multi-modal data, which can explore complementarity of multi-modal features to boost the hash performance. 2) We develop an asymmetric supervised hash learning module, where the learned hash codes are simultaneously correlated with the low-level consensus multi-modal factor and the high-level semantic labels to enhance discriminative capability. 3) The discrete hash code optimization solves hash code directly without quantization errors. 4) With a query-adaptive online hashing strategy, the hash codes are adaptively learned according to the specific query contents.

4.2.2 Run Time Comparison. In this subsection, we conduct experiments to compare the computational efficiency between OMH-DQ and baselines. Table 4 presents the comparison of training and query time on three datasets when code length is fixed to 128 bits. From it we can observe that the proposed OMH-DQ consumes the least time at the training stage and comparable time at the query stage. The training time of OMH-DQ is 16.99 seconds on MS COCO, while that of the second best competitor MAH is 102.98 seconds, the advantage of our method is very significant. At the online stage, although OMH-DQ adaptively learns hash codes of the new queries, this process has not degraded the query efficiency. It can achieve comparable or better (on MIR Flickr) performance than the second best ones.

The high efficiency of the OMH-DQ is mainly attributed to three reasons: 1) With the nonlinear feature mapping, OMH-DQ efficiently preserves the multi-modal features into hash codes with linear computational complexity. 2) Asymmetric hashing learning avoids direct decomposition of the pair-wise similarity matrix and thus reduces the complexity of hash code learning. 3) The proposed discrete hash code optimization method efficiently solves hash code within a simple operation instead of updating hash codes bit by bit.

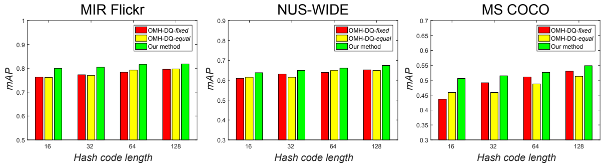

Figure 3: Effects of our query-adaptive online hashing.

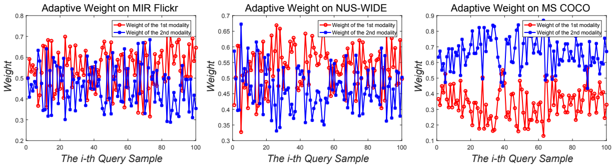

Figure 4: Modality weights adapted to dynamic queries.

4.3 Ablation Analysis

4.3.1 Is the weighting scheme effective? In our method, query hash codes are adaptively learned in an online mode. The designed virtual weights for modality combination can capture the variations of the queries, and thus boost the retrieval performance. To evaluate its effectiveness, we design two variants of our method: 1) OMH-DQ-*fixed*: It adopts fixed modality combination weights obtained from the offline learning to generate the query hash codes. 2) OMH-DQ-*equal*: It fixes the weight of each modality to 1 at both the offline learning and online hashing phases. Fig. 3 shows the comparison of the retrieval performance. From the figures, we can find that the performance of our method is obviously higher than that of the other two variants. Moreover, we show that our proposed model adaptively learns weight according to the dynamic query contents in Fig. 4.

4.3.2 Is self-weighted multi-modal fusion effective? In this paper, we propose a self-weighted multi-modal fusion strategy to preserve

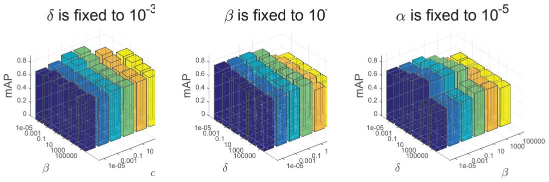

Figure 5: Parameter variations on MIR Flickr.

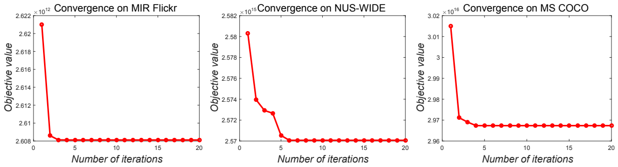

Figure 6: Convergence analysis on three datasets.

the multi-modal data into hash codes while exploring their complementarity. To evaluate its effectiveness, we design three variants of our method: 1) OMH-DQ-*img*: It only inputs the visual modality feature into the hashing model. 2) OMH-DQ-*txt*: It only inputs the textual modality feature into the hashing model. 2) OMH-DQ-*con*: It simply concatenates multi-modal features as a unified vector input and then imports it into the hashing model. The performance comparison is shown in Table 5. We can observe that, on three datasets, our method performs better than the other three variants. By exploring the complementarity of multi-modal features, the self-weighted multi-modal fusion can improve the retrieval performance.

4.3.3 How about the nonlinear feature mapping on the performance improvement? OMH-DQ nonlinearly projects features of different modalities into a new representation based on the anchors, which help to reduce the computational complexity of hash code learning to $O(n)$. In addition, it can effectively characterize the nonlinear instance relations and thus has stronger description capability than original feature. To validate its effectiveness, we design a variant of our method named OMH-DQ-*linear* for comparison. OMH-DQ-*linear* simply imports the original features of different modalities into the hashing model. The retrieval performance comparison between OMH-DQ-*linear* and our method is shown in Table 5. We can find that our method achieves superior retrieval precision than OMH-DQ-*linear*. Besides, we compare the training and query efficiency of OMH-DQ-*linear* and our method on three datasets when the code length is fixed on 128 bits and show the results in Table 4. We can find that without nonlinear feature mapping, both the training and query efficiency are reduced. On MS COCO, OMH-DQ-*linear* spends 116.83 seconds at query phase, which takes 13.17 seconds more than our method. These results demonstrate that the nonlinear feature mapping can reduce the computational complexity and improve the retrieval precision.

4.3.4 Can discrete hash code optimization avoid quantization errors? We propose an efficient discrete optimization method to directly learn binary hash codes instead of the relaxing+rounding optimization strategy commonly used in the previous works. To validate its effects, we design a variant of our method named OMH-DQ-*relax* for comparison. During hash code optimization, OMH-DQ-*relax* firstly relaxes the discrete constraints to learn hash codes and then obtains binary codes by mean-thresholding. The objective function of OMH-DQ-*relax* is similar to our method as Eq.(5) and the difference is on the updating rule of the hash code matrix $\mathbf{B}$ as $\mathbf{B} = (\beta \mathbf{I}_r + \alpha \mathbf{R}\mathbf{H}\mathbf{H}^T\mathbf{R}^T)^{-1}(\beta \mathbf{R}\mathbf{H} + \alpha r \mathbf{R}\mathbf{H}\mathbf{S}^T)$. The final binary hash codes $\mathbf{B}$ are obtained by mean-thresholding. The optimization of other variables is similar to that of our proposed OMH-DQ. The retrieval performance of OMH-DQ-*relax* and our method are shown in Table 5. We can observe that the performance of our proposed method is obviously better than that of OMH-DQ-*relax* on three datasets. With discrete optimization, the quantization loss can be effectively reduced and thus the retrieval performance can be improved.

4.3.5 Convergency and Parameter Sensitivity Analysis. We conduct empirical experiments to observe the performance variations with the involved parameters α, β, δ. We report the results on MIR Flickr when the code length is fixed to 32 bits. Similar results can be found on other code lengths and datasets. Since α, β and δ are equipped in the overall objective function, we vary the value of them from the range of $\{10^{-5}, 10^{-3}, 10^{-1}, 10, 10^3, 10^5\}$ while fixing the others. Detailed experimental results are presented in Fig.5. From Fig.5(a), (b), (c), we can find that the performance is relatively stable when α is in the range of $\{10^{-5}, 10^{-3}\}$, β is in the range of $\{10^{-1}, 10^1\}$, and δ is in the range of $\{10^{-5}, 10^{-3}, 10^{-1}, 10^1\}$.

In addition, we conduct convergency analysis on MIR Flickr, NUS-WIDE and MS COCO with the code length as 64 bits. The convergency curves are shown in Fig. 6. The performance on other code lengths is similar. We can observe from the figures that, the updating of variables monotonically decreases the objective function value and eventually reaches a local minimum at each iteration.

5 CONCLUSION

In this paper, we propose a novel *Online Multi-modal Hashing with Dynamic Query-adaption* (OMH-DQ) to support large-scale multimedia retrieval. We formulate a unified hash code learning model that can enhance the discriminative capability of hash codes, avoid the challenging symmetric similarity matrix factorization, and facilitate efficient (computation and storage) discrete hash code optimization. The discrete hash codes are learned directly without relaxing quantization information loss. More importantly, we design a query-adaptive online hashing module, so that the generated query hash codes can capture the varied query contents. Experiments on several public multimedia retrieval datasets demonstrate the superiority of the proposed approach.

6 ACKNOWLEDGMENTS

The work is partially supported by the National Natural Science Foundation of China (Nos. U1836216, 61802236) and the Key Research and Development Foundation of Shandong Province (Nos. 2017GGX10117, 2017CXGC0703), in part by the Natural Science Foundation of Shandong, China (No. ZR2019QF002).

REFERENCES

[1] Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. 2018. Cross-Modal Retrieval in the Cooking Context: Learning Semantic Text-Image Embeddings. In *SIGIR*. 135–44.

[2] Suthee Chaidaroon, Travis Ebesu, and Yi Fang. 2018. Deep Semantic Text Hashing with Weak Supervision. In *SIGIR*. 1109–1112.

[3] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *CIVR*. 48.

[4] Guiguang Ding, Yuchen Guo, and Jile Zhou. 2014. Collective matrix factorization hashing for multimodal data. In *CVPR*. 2075–2082.

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

[6] Mark J. Huiskes and Michael S. Lew. 2008. The MIR flickr retrieval evaluation. In *SIGMM*. 39–43.

[7] Saehoon Kim and Seungjin Choi. 2013. Multi-view anchor graph hashing. In *ICASSP*. 3123–3127.

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

[9] Zhouchen Lin, Minming Chen, and Yi Ma. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055* (2010).

[10] Zijia Lin, Guiguang Ding, Jungong Han, and Jianmin Wang. 2017. Cross-view retrieval via probability-based semantics-preserving hashing. *TCYB* 47, 12 (2017), 4342–4355.

[11] Han Liu, Xiangnan He, Fuli Feng, Liqiang Nie, Rui Liu, and Hanwang Zhang. 2018. Discrete Factorization Machines for Fast Feature-based Recommendation. In *IJCAI*. 3449–3455.

[12] Li Liu, Mengyang Yu, and Ling Shao. 2015. Multiview alignment hashing for efficient image search. *TIP* 24, 3 (2015), 956–966.

[13] Xianglong Liu, Junfeng He, Di Liu, and Bo Lang. 2012. Compact kernel hashing with multiple features. In *ACM MM*. 881–884.

[14] Fuchen Long, Ting Yao, Qi Dai, Xinmei Tian, Jiebo Luo, and Tao Mei. 2018. Deep Domain Adaptation Hashing with Adversarial Learning. In *SIGIR*. 725–734.

[15] Katta G. Murty. 2013. *Nonlinear Programming: Theory and Algorithms* (3rd ed.). Wiley Publishing.

[16] Fumin Shen, Xin Gao, Li Liu, Yang Yang, and Heng Tao Shen. 2017. Deep Asymmetric Pairwise Hashing. In *ACM MM*. 1522–1530.

[17] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. 2015. Supervised discrete hashing. In *CVPR*. 37–45.

[18] Xiaobo Shen, Funmin Shen, Liliu, Yunhao Yuan, Weiwei Liu, and Quansen Sun. 2018. Multiview Discrete Hashing for Scalable Multimedia Search. *ACM TIST* 9, 5 (2018), 53.

[19] Xiaobo Shen, Fumin Shen, Quan-Sen Sun, and Yunhao Yuan. 2015. Multi-view latent hashing for efficient multimedia search. In *ACM MM*. 831–834.

[20] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[21] Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Jiebo Luo. 2013. Effective multiple feature hashing for large-scale near-duplicate video retrieval. *TMM* 15, 8 (2013), 1997–2008.

[22] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. 2013. Intermedia hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*. 785–796.

[23] Di Wang, Xinbo Gao, Xiumei Wang, and Lihuo He. 2018. Label Consistent Matrix Factorization Hashing for Large-Scale Cross-Modal Similarity Search. *TPAMI* (2018).

[24] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. 2018. A Survey on Learning to Hash. *TPAMI* 40, 4 (2018), 769–790.

[25] Liang Xie, Jialie Shen, Jungong Han, Lei Zhu, and Ling Shao. 2017. Dynamic Multi-View Hashing for Online Image Retrieval. In *IJCAI*. 3133–3139.

[26] Liang Xie, Jialie Shen, and Lei Zhu. 2016. Online Cross-Modal Hashing for Web Image Retrieval. In *AAAI*. 294–300.

[27] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. 2017. Pairwise Relationship Guided Deep Hashing for Cross-Modal Retrieval. In *AAAI*. 1618–1625.

[28] Rui Yang, Yuliang Shi, and Xin-Shun Xu. 2017. Discrete Multi-view Hashing for Effective Image Retrieval. In *ICMR*. 175–783.

[29] Dongqing Zhang and Wu-Jun Li. 2014. Large-Scale Supervised Multimodal Hashing with Semantic Correlation Maximization. In *AAAI*. 2177–2183.

[30] Dan Zhang, Fei Wang, and Luo Si. 2011. Composite hashing with multiple information sources. In *SIGIR*. 225–234.

[31] Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. 2016. Discrete Collaborative Filtering. In *SIGIR*. 325–334.

[32] Hanwang Zhang, Meng Wang, Richang Hong, and Tat-Seng Chua. 2016. Play and Rewind: Optimizing Binary Representations of Videos by Self-Supervised Temporal Hashing. In *MM*. 781–790.

[33] Hanwang Zhang, Na Zhao, Xindi Shang, Huan-Bo Luan, and Tat-Seng Chua. 2016. Discrete Image Hashing Using Large Weakly Annotated Photo Collections. In *AAAI*. 3669–3675.

[34] Peichao Zhang, Wei Zhang, Wu-Jun Li, and Minyi Guo. 2014. Supervised hashing with latent factor models. In *SIGIR*. 173–182.

[35] Xi Zhang, Siyu Zhou, Jiashi Feng, Hanjiang Lai, Bo Li, Yan Pan, Jian Yin, and Shuicheng Yan. 2017. HashGAN: Attention-aware Deep Adversarial Hashing for Cross Modal Retrieval. *CoRR* abs/1711.09347 (2017).

[36] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. 2016. Deep Hashing Network for Efficient Similarity Retrieval. In *AAAI*. 2415–2421.

[37] Lei Zhu, Zi Huang, Xiaojun Chang, Jingkuan Song, and Heng Tao Shen. 2017. Exploring Consistent Preferences: Discrete Hashing with Pair-Exemplar for Scalable Landmark Search. In *MM*. 726–734.

[38] Lei Zhu, Zi Huang, Zhihui Li, Liang Xie, and Heng Tao Shen. 2018. Exploring Auxiliary Context: Discrete Semantic Transfer Hashing for Scalable Image Retrieval. *TNNLS* 29, 11 (2018), 5264–5276.

[39] Lei Zhu, Jialie Shen, Xiaobai Liu, Liang Xie, and Liqiang Nie. 2016. Learning Compact Visual Representation with Canonical Views for Robust Mobile Landmark Search. In *IJCAI*. 3959–3967.

[40] Lei Zhu, Jialie Shen, Liang Xie, and Zhiyong Cheng. 2017. Unsupervised visual hashing with semantic assistant for content-based image retrieval. *TKDE* 29, 2 (2017), 472–486.