Flexible Online Multi-modal Hashing for Large-scale Multimedia Retrieval

Xu Lu[1], Lei Zhu[1], Zhiyong Cheng[2], Jingjing Li[3], Xiushan Nie[4], Huaxiang Zhang[1]
[1] Shandong Normal University
[2] Shandong Computer Science Center (National Supercomputer Center in Jinan),
Qilu University of Technology (Shandong Academy of Sciences)
[3] University of Electronic Science and Technology of China
[4] Shandong Jianzhu University

ABSTRACT

Multi-modal hashing fuses multi-modal features at both offline training and online query stage for compact binary hash learning. It has aroused extensive attention in research filed of efficient large-scale multimedia retrieval. However, existing methods adopt batch-based learning scheme or unsupervised learning paradigm. They cannot efficiently handle the very common online streaming multi-modal data (for batch-learning methods), or learn the hash codes suffering from limited discriminative capability and less flexibility for varied streaming data (for existing online multi-modal hashing methods). In this paper, we develop a supervised *Flexible Online Multi-modal Hashing* (FOMH) method to adaptively fuse heterogeneous modalities and flexibly learn the discriminative hash code for the newly coming data, even if part of the modalities is missing. Specifically, instead of adopting the fixed weights, the modalities weights in FOMH are automatically learned with the proposed flexible multi-modal binary projection to timely capture the variations of streaming samples. Further, we design an efficient asymmetric online supervised hashing strategy to enhance the discriminative capability of the hash codes, while avoiding the challenging symmetric semantic matrix decomposition and storage cost. Moreover, to support fast hash updating and avoid the propagation of binary quantization errors in online learning process, we propose to directly update the hash codes with an efficient discrete online optimization. Experiments on several public multimedia retrieval datasets validate the superiority of the proposed method from various aspects.

CCS CONCEPTS

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

Lei Zhu (leizhu0608@gmail.com) and Huaxiang Zhang (huaxzhang@hotmail.com) are corresponding authors.

MM '19, October 21–25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350999

KEYWORDS

Flexible, Online Multi-modal Hashing, Large-scale Multimedia Retrieval, Adaptively

ACM Reference Format:
Xu Lu, Lei Zhu, Zhiyong Cheng, Jingjing Li, Xiushan Nie, Huaxiang Zhang. 2019. Flexible Online Multi-modal Hashing for Large-scale Multimedia Retrieval. In *Proceedings of the 27th ACM Int'l Conference on Multimedia (MM'19)*, Oct. 21–25, 2019, Nice, France. ACM, NY, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350999

1 INTRODUCTION

Hashing could support large-scale similarity search with its high efficiency on both search speed and storage by encoding the high-dimensional data into compact binary hash codes. For this desirable advantage, it has aroused extensive attention in literature [28].

Specifically, to support efficient multimedia retrieval, multi-modal hashing [10, 17, 18, 22, 23, 25, 35, 40–43] is studied by combining heterogeneous multi-modal data [4, 5, 29, 30] at both offline hash learning and online retrieval stages for compact binary hash learning. It is essentially different from uni-modal and cross-modal hashing that only one of the modalities is provided at query. With this research field, several multi-modal hashing methods[1] have been proposed with impressive performance. However, most existing multi-modal hashing methods adopt batch-based model learning. When new data arrive, they have to accumulate all the database data to re-train new hash functions and recompute the hash codes of the whole database. Thus, they are inefficient when adapting to the frequently updated multimedia database. Recently, several online learning methods [1–3, 8, 12–14, 32, 36] are proposed to support the efficient search over streaming database. Nevertheless, they mainly focus on uni-modal [1–3, 8, 12–14] and cross-modal hashing [32, 36], very few of them have been introduced to deal with multi-modal hashing problem.

Two strategies can be adopted when developing the new online multi-modal hashing methods: 1) exploit the online learning tricks, such as online graph learning [34] and online matrix factorization [19], to upgrade the existing multi-modal hashing methods[2] so that they can handle the online streaming data. 2) extend the existing uni-modal online hashing methods to deal with multi-modal data by simply importing their models with the concatenated multi-modal features. These two strategies seem feasible. However, they

[1] Many researchers also term multi-modal hashing as multi-view hashing.

[2] Existing multi-modal hashing methods are generally based on graph and matrix factorization.

will cause two important problems: Firstly, these two strategies explicitly adopt fixed or equal modality combination weights to fuse heterogeneous modalities into hash codes. They cannot sufficiently capture the dynamic variation of streaming multi-modal data. Secondly, they cannot generate hash codes for newly coming multi-modal data samples when part of modalities is missing. To the best of our knowledge, the only online multi-modal hashing method is Online Dynamic Multi-view Hashing (ODMVH) [31]. Besides to held the above two problems, ODMVH will suffer from the limited performance, as it is an unsupervised method and has not exploited any discriminative semantic information.

Motivated by the above analysis, in this paper, we propose an efficient *Flexible Online Multi-modal Hashing* (FOMH) to solve all aforementioned problems in a new learning framework. Firstly, we project heterogeneous multi-modal data into common hash code with flexible binary projection scheme. The heterogeneous modality gap is alleviated with common binary projection and each modality is independently projected into hash codes so that the multi-modal fused hash code could be generated flexibly. Specifically, we propose a self-weighted modality fusion strategy where the modality combination weights can be learned adaptively according to the online streaming multi-modal data. Moreover, we design an asymmetric supervised learning strategy to efficiently preserve the discriminative pair-wise semantic similarities into the binary hash codes, while preventing challenging symmetric semantic matrix decomposition and storage cost. With the discriminative semantic supervision, the representation capability of online hash codes is enhanced. Finally, we propose a discrete online optimization method to directly solve the binary hash codes while avoiding the propagation of binary quantization errors in online learning process. The main contributions of this paper are:

- We design a flexible online multi-modal hash learning framework. It can generate discriminative multi-modal fused hash codes when the multimedia instances arrive in streaming fashion, even if part of the modalities is missing. Specifically, the modality combination weights in our method are learned with an effective self-weighting scheme to be adaptively adjusted and thus capture the variations of streaming multi-modal data. To the best of our knowledge, there is still no similar work.
- We develop an efficient asymmetric supervised online hashing module to enhance the representation capability of the hash codes by exploiting the discriminative pair-wise semantic labels. This module can successfully avoid the challenging and time-consuming symmetric semantic matrix decomposition and storage cost, and thus can support fast online hash updating.
- A discrete online hash optimization method is proposed to directly update the hash code for streaming data, where the propagation of binary quantization errors can be avoided in online learning process. Experimental results on the public multimedia datasets demonstrate the superior performance of the proposed method from various aspects.

2 RELATED WORK

Multi-modal hashing. Most existing multi-modal hashing methods are developed within the unsupervised learning paradigms. They usually construct graphs or adopt matrix factorization to preserve the sample relations in multiple modalities into binary hash codes. Multiple Feature Hashing (MFH) [25] computes multiple affinity matrices to exploit the structural information of multi-modal features. Multi-view Alignment Hashing (MAH) [17] is based on regularized kernel nonnegative matrix factorization, and it learns hash codes by uncovering the hidden semantics and the joint probability distribution of data. Multi-view Latent Hashing (MVLH) [23] projects multiple modalities into a unified kernel feature space, where the weights of different modalities are adaptively learned according to the reconstruction error of each modality. Multi-view Discrete Hashing (MvDH) [22] performs matrix factorization to generate the hash codes as the latent representations shared by multiple modalities. The joint learning of hash codes and performing clustering analysis enables that MvDH to generate more discriminative hash codes. Due to the isolation with the semantic labels, all these unsupervised multi-modal hashing suffer from the limited discriminative capability. Moreover, Deep Multi-View Hashing (DMVH-4layer) [10] and Multi-modal Hashing with Orthogonal Regularization (DMHOR) [27] are both unsupervised deep multi-modal hashing methods which are based on deep neural networks. DMVH-4layer learns individual and shared hidden spaces from multiple views of data with view-specific and shared hidden nodes. DMHOR imposes an orthogonal regularizer on the weighting matrices of the model to reduce information redundancy.

Few supervised multi-modal hashing methods leverage semantic labels as supervision to guide the hash learning. Compact Kernel Hashing with Multiple Features (MFKH) [18] formulates the multiple feature mapping as a similarity preserving problem with optimal linearly-combined multiple kernels. Discrete Multi-view Hashing (DMVH) [35] exploits the discriminative semantic labels to directly learn discrete hash codes, which preserve both the local similarity structure and the semantic similarity of data points.

Online Hashing. It aims to efficiently update hash functions based on the online streaming data, which is common in the real-word applications. Online Kernel Hashing (OKH) [8] is the first attempt to learn hash function via an online Passive-Aggressive strategy [7], which updates hash functions to fit the newly available data pairs and meanwhile coincide with the old ones. Similarly, Adaptive Hashing (AdaptHash) [3] is a supervised online hashing method. It adopts a hinge loss to decide which hash function to be updated. Differently, the unsupervised online Sketching Hashing (SketchHash) [12] adopts an efficient variant of SVD decomposition to learn hash functions. Online Supervised Hashing (OSH) [1] is built on error correcting output codes. It makes no assumption about the number of possible class labels and accommodates new classes as they are presented in the incoming data stream. Online Hashing with Mutual Information (MIHash) [2] is developed to optimize the mutual information between neighbors and non-neighbors. Hadamard Codebook based Online Hashing (HCOH) [14] is a supervised online hashing method. It samples a codeword from Hadamard matrix to learn discriminative binary codes in online fashion. Balanced Similarity for Online Discrete Hashing (BSODH) [13] is a supervised online hashing method to preserve the similarity between the streaming data and the existing dataset via an asymmetric graph regularization.

Although impressive performance has been achieved, the aforementioned online hashing methods only deal with the uni-modal data. Only a few online hashing works are proposed for heterogeneous multi-modal data. Online Cross-modal Hashing (OCMH) [32] is designed for cross-modal retrieval by simultaneously updating hash codes and preserving cross-modal correlation. To the best of our knowledge, the most similar work to our approach is Online Dynamic Multi-view Hashing (ODMVH) [31]. However, it is an unsupervised method, and the learned hash codes will suffer limited semantics and discriminative capability. Further, it adopts simple fixed modality weights and binary projection mechanisms, which cannot adapt the variations of streaming multimedia contents, and handle the modality-missing problems.

3 THE PROPOSED METHOD

Notations. Throughout this paper, we adopt uppercase letters to represent matrices and lowercase letters to represent vectors. Suppose the training dataset is comprised of n streaming training samples represented with M different modality features. The m-th modality feature is $\mathbf{X}^{(m)} = [\widetilde{\mathbf{X}}_{t-1}^{(m)}, \mathbf{X}_t^{(m)}] \in \mathbb{R}^{d_m \times n}$, where $\mathbf{X}_t^{(m)} \in \mathbb{R}^{d_m \times n_t}$ is the new chunk of the m-th modality arriving at round t, $\widetilde{\mathbf{X}}_{t-1}^{(m)} \in \mathbb{R}^{d_m \times (n-n_t)}$ is the existing chunk of old data samples before round t (d_m is the dimension of the m-th modality, n_t is the size of the new data chunk at round t). The hash code of total data chunk is $\mathbf{B} = [\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] \in \{-1,1\}^{r\times n}$, where $\mathbf{B}_t \in \{-1,1\}^{r \times n_t}$ is the hash code of the new chunk arriving at round t, $\widetilde{\mathbf{B}}_{t-1} \in \{-1,1\}^{r\times(n-n_t)}$ is the existing hash code of old data samples before round t (r is the hash code length). $\Delta_n \overset{def}{=} \{x \in \mathbb{R}^n | x_i \geq 0, \mathbf{1}^\mathsf{T} x = 1\}$ is the probabilistic simplex.

3.1 Flexible Supervised Online Multi-modal Hash Learning

3.1.1 Flexible Multi-modal Binary Projection. Multi-modal hash codes should comprehensively preserve the multi-modal information. To this end, the multi-modal data information should be first modelled so that the hash projection learning can be performed. In literature, most existing multi-modal hashing methods [11, 17, 25, 37] 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, inspired by anchor graph [23], we design a flexible multi-modal binary projection to reduce the complexity to $O(n)$. Specifically, given the m-th modality data feature $\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, ..., \mathbf{x}_n^{(m)}] \in \mathbb{R}^{d_m \times n}$ of the whole training data, we first obtain the nonlinearly transformed representation $\phi(\mathbf{x}_i^{(m)})$ as $[\exp(\frac{\|\mathbf{x}_i^{(m)} - \mathbf{o}_1^{(m)}\|_\mathsf{F}^2}{2\sigma_m^2}), ..., \exp(\frac{\|\mathbf{x}_i^{(m)} - \mathbf{o}_p^{(m)}\|_\mathsf{F}^2}{2\sigma_m^2})]^\mathsf{T}$ where $\{\mathbf{o}_1^{(m)}, ..., \mathbf{o}_p^{(m)}\}$ are p anchors that are randomly selected from the training samples in the m-th modality, σ_m is the Gaussian kernel parameter. $\phi(\mathbf{X}^{(m)}) = [\phi(\mathbf{x}_1^{(m)}), ..., \phi(\mathbf{x}_n^{(m)})] \in \mathbb{R}^{p \times n}$ preserves the modality-specific sample correlations by simply characterizing the correlations between the samples and certain anchors and it costs $O(mnp)$.

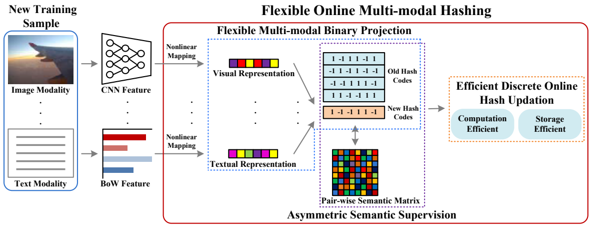

Figure 1: The basic online hash learning process of FOMH.

Different from uni-modal and cross-modal retrieval, exploiting the complementarity of heterogeneous multi-modal features, and simultaneously, alleviating the heterogeneous modality gap [16] are important to learn the multi-modal fused hash codes. However, existing multi-modal hashing methods adopt fixed or equal modality combination weights to fuse heterogeneous modalities into hash codes. They cannot timely capture the dynamic variation of streaming multi-modal data. Moreover, they cannot generate hash codes for newly coming multi-modal data samples when the part of modalities is missing.

In this paper, we propose a flexible multi-modal fusion with self-weighting to adaptively project heterogeneous modalities into hash codes. Specifically, considering that data is imported into hashing model in streaming fashion, we formulate this part as

$$\min_{\mu^{(m)}, \mathbf{P}^{(m)}, \mathbf{B}_t} \sum_{m=1}^{M} \mu^{(m)} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)} \phi(\mathbf{X}^{(m)})\|_\mathsf{F}^2 + \varepsilon \|\boldsymbol{\mu}\|_\mathsf{F}^2, \quad (1)$$
$$s.t.\ \mathbf{B}_t \in \{-1,1\}^{r \times n_t},\ \boldsymbol{\mu} = [\mu^{(1)}, \mu^{(2)}, ..., \mu^{(M)}]^\mathsf{T},\ \boldsymbol{\mu} \in \Delta_M,$$

where $\{\mathbf{P}^{(m)} \in \mathbb{R}^{r \times p}\}_{m=1}^M$ are the projection matrices for M modalities. They collaboratively project multi-modal features into the shared binary codes to alleviate the heterogeneous modality gap. Note that, in this paper, we propose to learn multiple modality-specific projection matrices. It is different from the existing multi-modal hashing methods [11, 17, 18, 23, 25, 35] where a unified projection matrix is learned for the concatenated multi-modal feature, and all multi-modal features should be provided in the online learning process. The advantage of this strategy is obvious that the hash codes can be generated according to the specific provided modality types, even if part of modalities is missing. $\{\mu^{(m)}\}_{m=1}^M$ is the weight of the m-th modality and it measures the importance of modality feature. $\varepsilon > 0$ is a hyper-parameter that aims to avoid trivial solutions. With weight learning, the complementarity of multi-modal features can be exploited properly. $\|\cdot\|_\mathsf{F}$ is the Frobenius norm of the matrix.

However, Eq.(1) introduces an additional parameter ε, whose best value is confirmed to be data related and should be manually adjusted. In implementation, it indicates that more time will be consumed on parameter adjustment when hashing chunks of online streaming data. More practically, we even cannot manually set a proper parameter for large amounts of chunks of online streaming data. In this paper, we introduce a virtual weight and propose a self-weighted multi-modal fusion strategy which can achieve the same goal as Eq.(1) without additional hyper-parameters. Specifically, we formulate this part as

$$\min_{\mathbf{P}^{(m)}, \mathbf{B}_t \in \{-1,1\}^{r \times n_t}} \sum_{m=1}^{M} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)} \phi(\mathbf{X}^{(m)})\|_\mathsf{F}. \quad (2)$$

We can derive the following theorem.

THEOREM 3.1. *The above equation is equivalent to*

$$\min_{\boldsymbol{\mu}\in\Delta_M,\boldsymbol{P}^{(m)},\boldsymbol{B}_t\in\{-1,1\}^{r\times n_t}} \sum_{m=1}^{M} \frac{1}{\mu^{(m)}} \|[\widetilde{\boldsymbol{B}}_{t-1}, \boldsymbol{B}_t] - \boldsymbol{P}^{(m)}\phi(\boldsymbol{X}^{(m)})\|_{\mathsf{F}}^2. \quad (3)$$

PROOF. Note that,

$$\begin{aligned}
&\sum_{m=1}^{M} \frac{1}{\mu^{(m)}} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2 \\
&\overset{(a)}{=}(\sum_{m=1}^{M} \frac{1}{\mu^{(m)}} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2)(\sum_{m=1}^{M} \mu^{(m)}) \\
&\overset{(b)}{\geq}(\sum_{m=1}^{M} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}})^2,
\end{aligned}$$

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

$$\begin{aligned}
&(\sum_{m=1}^{M} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}})^2 \\
&= \min_{\boldsymbol{\mu}\in\Delta_M} \sum_{m=1}^{M} \frac{1}{\mu^{(m)}} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2.
\end{aligned}$$

It is easy to derive

$$\begin{aligned}
&\min_{\mathbf{P}^{(m)},\mathbf{B}_t} \sum_{m=1}^{M} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}} \\
\Leftrightarrow &\min_{\mathbf{P}^{(m)},\mathbf{B}_t} (\sum_{m=1}^{M} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}})^2 \\
\Leftrightarrow &\min_{\boldsymbol{\mu}\in\Delta_M,\mathbf{P}^{(m)},\mathbf{B}_t} \sum_{m=1}^{M} \frac{1}{\mu^{(m)}} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2,
\end{aligned}$$

which completes the proof. □

We introduce a virtual weight $\frac{1}{\mu^{(m)}}$, which acts the function of real weight. For a more discriminative modality, the value of $\|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}}^2$ is smaller, thus the corresponding virtual weight $\frac{1}{\mu^{(m)}}$ is larger, and vice versa.

In the online hash learning process, each chunk of streaming multi-modal data will possess its own feature distributions. Determining the modality weights as Eq.(2) can automatically determine the proper modality weights and thus capture the variations of online streaming data.

3.1.2 Asymmetric Semantic Supervision. At round t, the hash code $\mathbf{B}_t$ learned from Eq.(2) only preserves the low-level multi-modal information, however, has not exploited any semantic labels. It will suffer limited semantic representation capability. In this paper, we consider the semantic supervision on $\mathbf{B}_t$ to enhance its discriminative capability. Motivated by the impressive performance of pair-wise semantics on uni-modal [21, 39] and cross-modal hashing [33, 38], we formulate this part as

$$\min_{\mathbf{B}_t\in\{-1,1\}^{r\times n_t}} \|\mathbf{B}_t^{\mathsf{T}}\mathbf{B}_t - r\mathbf{S}_t\|_{\mathsf{F}}^2, \quad (4)$$

where $\mathbf{S}_t \in \mathbb{R}^{n_t\times n_t}$ is the pair-wise similarity matrix of newly coming data samples.

Nevertheless, directly solving binary hash code $\mathbf{B}_t$ in Eq.(4) is a very challenging problem due to the discrete symmetric factorization. In the online updating, we prefer fast hash learning. In this paper, we develop an asymmetric semantic supervision, which transfers the semantics from pair-wise semantic matrix $\mathbf{S}_t$ to hash codes, and simultaneously avoids challenging computation. Specifically, to solve the problem of computational consumption, we transform Eq.(4) into an equivalent form by substituting one of $\mathbf{B}_t$ with a continuous substitution $\mathbf{D}_t \in \mathbb{R}^{r\times n_t}$ and keep their consistency during the whole optimization process. The formula is

$$\min_{\mathbf{B}_t\in\{-1,1\}^{r\times n_t},\mathbf{D}_t} \|\mathbf{B}_t^{\mathsf{T}}\mathbf{D}_t - r\mathbf{S}_t\|_{\mathsf{F}}^2 + \beta\|\mathbf{B}_t - \mathbf{D}_t\|_{\mathsf{F}}^2.$$

In the above equation, only one of the decomposed variable $\mathbf{B}_t$ is imposed with discrete constraint. Accordingly, the challenging and time-consuming symmetric matrix factorization can be avoided. As shown below, with the support of asymmetric semantic supervision, 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. Besides, it costs $O(n_t^2)$ to store the elements of $\mathbf{S}_t$ and it is unacceptable in large-scale multimedia retrieval. We will address this problem by representing $\mathbf{S}_t$ with the label matrix in the subsection for online optimization.

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

$$\begin{aligned}
\min_{\mathbf{P}^{(m)},\mathbf{B}_t,\mathbf{D}_t} &\sum_{m=1}^{M} \|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}} + \alpha\|\mathbf{B}_t^{\mathsf{T}}\mathbf{D}_t - r\mathbf{S}_t\|_{\mathsf{F}}^2 \\
&+ \beta\|\mathbf{B}_t - \mathbf{D}_t\|_{\mathsf{F}}^2 + \gamma\sum_{m=1}^{M}\|\mathbf{P}^{(m)}\|_{\mathsf{F}}^2, \quad s.t.\ \mathbf{B}_t \in \{-1,1\}^{r\times n_t},
\end{aligned} \quad (5)$$

where α, β, and γ are balance parameters. The first term performs flexible multi-modal binary projection to learn fused hash codes, the second and the third terms perform asymmetric semantic supervision for discriminative capability enhancement of hash codes. The last term is a regularization term to avoid over-fitting.

3.1.4 Efficient Discrete Online Optimization. In this paper, we propose a discrete online optimization for Eq.(5) to handle each chunk of streaming data efficiently at a time. It can directly solve the hash codes in a fast mode, and avoid the accumulation and propagation of binary quantization errors in the online hash learning process. At round t, the key optimization steps are as follows.

Step 1: Update $\mu^{(m)}$. For convenience, we denote $\|[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t] - \mathbf{P}^{(m)}\phi(\mathbf{X}^{(m)})\|_{\mathsf{F}}$ by $\mathrm{g}^{(m)}$. The optimization problem can be written as

$$\min_{\mu^{(m)}\geq 0, \mathbf{1}^{\mathsf{T}}\boldsymbol{\mu}=1} \sum_{m=1}^{M} (\mathrm{g}^{(m)})^2/\mu^{(m)}, \quad (6)$$

which combining with Cauchy-Schwarz inequality gives

$$\sum_{m=1}^{M} \frac{(\mathrm{g}^{(m)})^2}{\mu^{(m)}} \overset{(a)}{=} (\sum_{m=1}^{M} \frac{(\mathrm{g}^{(m)})^2}{\mu^{(m)}})(\sum_{m=1}^{M} \mu^{(m)}) \overset{(b)}{\geq} (\sum_{m=1}^{M} \mathrm{g}^{(m)})^2, \quad (7)$$

where (a) holds since $\mathbf{1}^{\mathsf{T}}\boldsymbol{\mu} = 1$ and the equality in (b) holds when $\sqrt{\mu^{(m)}} \propto \frac{\mathrm{g}^{(m)}}{\sqrt{\mu^{(m)}}}$. Since the right-hand side of Eq.(7) is constant, the optimal $\mu^{(m)}$ in Eq.(6) can be obtained by

$$\mu^{(m)} = \mathfrak{g}^{(m)} / \sum_{m=1}^{M} \mathfrak{g}^{(m)}. \tag{8}$$

Step 2: Update $\mathbf{P}^{(m)}$. By setting the derivative of Eq.(5) w.r.t. $\mathbf{P}^{(m)}$ to zero, we can easily update $\mathbf{P}^{(m)}$ as

$$\mathbf{P}^{(m)} = (\frac{1}{\mu^{(m)}} [\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t](\phi(\mathbf{X}^{(m)}))^{\top})(\frac{1}{\mu^{(m)}} \phi(\mathbf{X}^{(m)})(\phi(\mathbf{X}^{(m)}))^{\top} + \gamma \mathbf{I}_p)^{-1}. \tag{9}$$

According to Theorem 3.2, we can easily update $\mathbf{P}^{(m)}$ in an online process.

Theorem 3.2. *The time complexity $[\widetilde{\boldsymbol{B}}_{t-1}, \boldsymbol{B}_t](\phi(\boldsymbol{X}^{(m)}))^{\top}$ and $\phi(\boldsymbol{X}^{(m)})(\phi(\boldsymbol{X}^{(m)}))^{T}$ is $O(n_t)$, which is linear to the size of newly coming data.*

Proof. We can easily obtain the following equation

$$\begin{aligned} [\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t](\phi(\mathbf{X}^{(m)}))^{\top} &= \left[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t\right] \begin{bmatrix} (\phi(\widetilde{\mathbf{X}}_{t-1}^{(m)}))^{\top} \\ (\phi(\mathbf{X}_t^{(m)}))^{\top} \end{bmatrix} \\ &= \widetilde{\mathbf{B}}_{t-1}(\phi(\widetilde{\mathbf{X}}_{t-1}^{(m)}))^{\top} + \mathbf{B}_t(\phi(\mathbf{X}_t^{(m)}))^{\top}. \end{aligned} \tag{10}$$

Since $\widetilde{\mathbf{B}}_{t-1}$ and $\phi(\widetilde{\mathbf{X}}_{t-1}^{(m)})$ are irrelevant to the new data, $\widetilde{\mathbf{B}}_{t-1}(\phi(\widetilde{\mathbf{X}}_{t-1}^{(m)}))^{\top}$ can be computed at previous round. Only $\mathbf{B}_t(\phi(\mathbf{X}_t^{(m)}))^{\top}$ needs to be calculated at current round, then the computational complexity of $[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t](\phi(\mathbf{X}^{(m)}))^{\top}$ is linear to the size of newly coming data.

Similarly, the computation complexity of $\phi(\mathbf{X}^{(m)})(\phi(\mathbf{X}^{(m)}))^{\top}$ is also linear to the newly coming data size n_t. □

From Theorem 3.2, we can easily obtain that the updating complexity of $\mathbf{P}^{(m)}$ is $O(n_t)$.

Step 3: Update $\mathbf{B}_t$. Since $\widetilde{\mathbf{B}}_{t-1}$ is permanent, we only need to update the hash code of newly coming data $\mathbf{B}_t$. At round t, the objective function with respect to $\mathbf{B}_t$ is

$$\min_{\mathbf{B}_t \in \{-1,1\}} \sum_{m=1}^{M} \frac{1}{\mu^{(m)}} \|\mathbf{B}_t - \mathbf{P}^{(m)} \phi(\mathbf{X}_t^{(m)})\|_{\mathsf{F}}^2 + \alpha \|\mathbf{B}_t^{\top} \mathbf{D}_t - r\mathbf{S}_t\|_{\mathsf{F}}^2 + \beta \|\mathbf{B}_t - \mathbf{D}_t\|_{\mathsf{F}}^2, \tag{11}$$

In this paper, we propose to directly update $\mathbf{B}_t$ in an efficient discrete way based on augmented Lagrange multiplier (ALM) [20]. We introduce an auxiliary variable $\mathbf{Z} \in \{-1, 1\}^{r \times n}$ to replace $\mathbf{B}_t$ and keep their consistency during the optimization process. Eq.(11) can be transformed to the following form

$$\begin{aligned} \min_{\mathbf{B}_t \in \{-1,1\}} Tr(-2 \sum_{m=1}^{M} \frac{1}{\mu^{(m)}} \mathbf{B}_t^{\top} \mathbf{P}^{(m)} \phi(\mathbf{X}_t) - 2\alpha r \mathbf{B}_t^{\top} \mathbf{D}_t \mathbf{S}_t^{\top}, \\ + \alpha \mathbf{B}_t^{\top} \mathbf{D}_t \mathbf{D}_t^{\top} \mathbf{Z} - 2\beta \mathbf{B}_t^{\top} \mathbf{D}_t) + \frac{\rho}{2} \|\mathbf{B}_t - \mathbf{Z} + \frac{\mathbf{G}}{\rho}\|_{\mathsf{F}}^2, \end{aligned} \tag{12}$$

where ρ adjusts the balance between terms, $\mathbf{G} \in \mathbb{R}^{r \times n_t}$ measures the gap between the target variable $\mathbf{B}_t$ and the auxiliary variable $\mathbf{Z}$. We can obtain the closed-solution of $\mathbf{B}_t$ as

$$\begin{aligned} \mathbf{B}_t = \mathtt{sign}(2 \sum_{m=1}^{M} \frac{1}{\mu^{(m)}} \mathbf{P}^{(m)} \phi(\mathbf{X}_t^{(m)}) + 2\alpha r \mathbf{D}_t \mathbf{S}_t^{\top} - \alpha \mathbf{D}_t \mathbf{D}_t^{\top} \mathbf{Z}. \\ + 2\beta \mathbf{D}_t + \rho \mathbf{Z} - \mathbf{G}). \end{aligned} \tag{13}$$

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

$$\mathbf{S}_t = 2\mathbf{S}'_t - \mathbf{E} = 2\mathbf{L}_t^{\top} \mathbf{L}_t - \mathbf{1}_{n_t} \mathbf{1}_{n_t}^{\top}, \tag{14}$$

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

$$\mathbf{D}_t \mathbf{S}_t^{\top} = 2\mathbf{D}_t \mathbf{L}_t^{\top} \mathbf{L}_t - \mathbf{D}_t \mathbf{1}_{n_t} \mathbf{1}_{n_t}^{\top},$$

which consumes $O(n_t)$. Since the code length, the number of anchor points and semantic categories are irrelevant to the size of newly coming data, the computational complexity of updating $[\widetilde{\mathbf{B}}_{t-1}, \mathbf{B}_t]$ is linear to the size of newly coming data size n_t.

Step 4: Update $\mathbf{D}_t$. We can easily obtain

$$\mathbf{D}_t = (\alpha \mathbf{B}_t \mathbf{B}_t^{\top} + \beta \mathbf{I}_r)^{-1} (\alpha r \mathbf{B}_t \mathbf{S}_t + \beta \mathbf{B}_t), \tag{15}$$

where $\mathbf{B}_t \mathbf{S}_t$ can be transformed by Eq.(14) as $\mathbf{B}_t \mathbf{S}_t = 2\mathbf{B}_t \mathbf{L}_t^{\top} \mathbf{L}_t - \mathbf{B}_t \mathbf{1}_{n_t} \mathbf{1}_{n_t}^{T}$.

Step 5: Update Z. We can obtain the closed solution of **Z** as

$$\mathbf{Z} = \mathtt{sign}(-\alpha \mathbf{D}_t \mathbf{D}_t^{\top} \mathbf{B}_t + \rho \mathbf{B}_t + \mathbf{G}). \tag{16}$$

Step 6: Update G. The update rule is

$$\mathbf{G} = \mathbf{G} + \rho(\mathbf{B}_t - \mathbf{Z}). \tag{17}$$

3.2 Retrieval Process

Given a new chunk of query samples $\{\mathbf{X}_q^{(m)}\}_{m=1}^{M}$ and n_q is the number of newly coming query samples. We iteratively solve query hash codes and modality weights as

Step 1: Update $\mu_q^{(m)}$. As Section 3.1.4, we denote $\|\mathbf{B}_q - \mathbf{P}^{(m)} \phi(\mathbf{X}_q^{(m)})\|_{\mathsf{F}}$ by $\mathfrak{g}_q^{(m)}$. The optimal $\mu_q^{(m)}$ is given by $\mu_q^{(m)} = \mathfrak{g}_q^{(m)} / \sum_{m=1}^{M} \mathfrak{g}_q^{(m)}$.

Step 2: Update $\mathbf{B}_q$. We can obtain the closed solution of $\mathbf{B}_q$ as $\mathbf{B}_q = \mathtt{sign}(\sum_{m=1}^{M} \frac{1}{\mu_q^{(m)}} \mathbf{P}^{(m)} \phi(\mathbf{X}_q^{(m)}))$.

As shown above, the query hash codes can be flexibly generated according to the specific query modality contents with the automatically determined modality weights. The hash codes can also capture the variations of queries.

3.3 Complexity Analysis

We provide the complexity analysis of FOMH at round t. When updating $\mathbf{P}^{(m)}$, we only need to update the terms related to the newly coming samples, while other terms are constants which were calculated at last round. Besides, updating $\mathbf{B}_t$, $\mathbf{D}_t$ and $\mu^{(m)}$ are also related to newly coming samples. Hence, the computational complexity of online optimization procedure is linear to the size of newly coming samples, which is efficient in computation complexity. Moreover, since all variables are only related to the number of anchor points p, hash code length r, the size of newly coming samples n_t, and the number of classes c, which are typically small, the optimization process is efficiency in memory cost.

In addition, we can avoid explicitly computing the pair-wise similarity matrix $\mathbf{S}_t$, but substituting it with $\mathbf{L}_t$, and thus we successfully reduce the space complexity to $O(n_t)$. In a sum, both the computational and the space complexity of FOMH are linear with

Table 1: MAP comparison results when only image modality is provided.

Methods	Image-only											
	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
OKH [8]	0.6320	0.6588	0.6834	0.7128	0.4415	0.4693	0.4820	0.4983	0.3891	0.3904	0.3909	0.3914
OSH [1]	0.6332	0.6358	0.6498	0.6582	0.4711	0.5019	0.5190	0.5379	0.3897	0.3900	0.3903	0.3905
MIHash [2]	0.7086	0.7204	0.7239	0.7387	0.5283	0.5571	0.5636	0.5722	0.3926	0.3930	0.3935	0.3940
HCOH [14]	0.6650	0.6975	0.7207	0.7387	0.4907	0.5170	0.5471	0.5642	0.3930	0.3933	0.3941	0.3957
FOMH-*map*	0.6869	0.6968	0.7167	0.7326	0.4475	0.4822	0.5057	0.5088	0.3929	0.3946	0.3955	0.3965
FOMH-*supe*	0.5932	0.5958	0.6003	0.6102	0.3739	0.3857	0.3950	0.4112	0.3773	0.3850	0.3855	0.3856
FOMH-*relax*	0.6317	0.6639	0.6652	0.6710	0.3805	0.3845	0.3870	0.3955	0.3888	0.3887	0.3885	0.3885
Ours	**0.7260**	**0.7318**	**0.7458**	**0.7567**	**0.6193**	**0.6334**	**0.6432**	**0.6532**	**0.3971**	**0.3987**	**0.4028**	**0.4046**

Table 2: MAP comparison results when only text modality is provided.

Methods	Text-only											
	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
OKH [8]	0.5800	0.5839	0.5851	0.5855	0.3639	0.3672	0.3722	0.3769	0.3847	0.3896	0.3922	0.3957
OSH [1]	0.5824	0.5841	0.5847	0.5865	0.3657	0.3674	0.3701	0.3742	0.3938	0.3950	0.3976	0.4012
MIHash [2]	0.5880	0.5912	0.5920	0.5968	0.3831	0.3829	0.3832	0.3879	0.4106	0.4133	0.4219	0.4336
HCOH [14]	0.5933	0.5941	0.5974	0.5997	0.3689	0.3746	0.3776	0.3789	0.4099	0.4150	0.4249	0.4378
FOMH-*map*	0.6343	0.6319	0.6271	0.6313	0.4623	0.4874	0.4908	0.5261	0.4792	0.5027	0.5202	0.5243
FOMH-*supe*	0.5834	0.5847	0.5851	0.5863	0.3524	0.3647	0.3699	0.3714	0.3933	0.3947	0.3977	0.4021
FOMH-*relax*	0.5989	0.6057	0.5993	0.6005	0.3749	0.3930	0.3919	0.4093	0.3955	0.3958	0.3982	0.4014
Ours	**0.6410**	**0.6602**	**0.6618**	**0.6697**	**0.4752**	**0.4960**	**0.5108**	**0.5279**	**0.4959**	**0.5041**	**0.5277**	**0.5507**

Table 3: MAP comparison results when both image and text modalities are provided.

Methods	Both Image and Text											
	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 [25]	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 [18]	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 [17]	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 [23]	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 [22]	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 [35]	0.7231	0.7326	0.7495	0.7641	0.5665	0.5856	0.6063	0.6285	0.4123	0.4288	0.4355	0.4563
ODMVH [31]	0.6960 (37 bits)				0.4437 (41 bits)				0.3917 (81 bits)			
OKH [8]	0.6361	0.6621	0.6749	0.6891	0.4366	0.4636	0.4781	0.4987	0.3830	0.3950	0.3948	0.3941
OSH [1]	0.6121	0.6303	0.6405	0.6509	0.4475	0.4789	0.5035	0.5269	0.3924	0.3950	0.3961	0.3993
MIHash [2]	0.7088	0.7154	0.7308	0.7401	0.5467	0.5660	0.5689	0.5760	0.3923	0.3926	0.3939	0.3937
HCOH [14]	0.6815	0.6999	0.7267	0.7482	0.4993	0.5500	0.5562	0.5755	0.3931	0.3939	0.3942	0.3947
FOMH-*map*	0.7434	0.7410	0.7458	0.7422	0.5548	0.5860	0.6169	0.6068	0.4022	0.3993	0.4042	0.4059
FOMH-*supe*	0.5994	0.5963	0.6341	0.6358	0.3539	0.3649	0.4057	0.4024	0.3887	0.3896	0.3976	0.3915
FOMH-*relax*	0.6467	0.7371	0.7360	0.7373	0.4249	0.4856	0.4695	0.5273	0.4043	0.4073	0.4009	0.3978
Ours	**0.7557**	**0.7632**	**0.7654**	**0.7705**	**0.6277**	**0.6317**	**0.6373**	**0.6441**	**0.5008**	**0.5148**	**0.5172**	**0.5294**

the size of newly coming samples. Our approach is scalable for large-scale multimedia retrieval.

4 EXPERIMENTS

Evaluation Datasets In this paper, we conduct experiments on three publicly available multimedia retrieval datasets, which are widely used for performance evaluation of multi-modal hashing methods [18, 22, 35]. Their detailed experimental settings are as follows: 1) **MIR Flickr** [9] is comprised of 20,015 image-text samples and each sample pair is annotated with several user assigned tags which are from 24 provided unique labels. Each image is represented as a 4,096-dimensional visual feature extracted by the Caffe implementation of VGG Net [24], while text is represented as a 1,386-dimensional bag-of-words feature. We randomly select 100 sample pairs from each category, and obtain 2,243 samples as the query set. The remaining 17,772 sample pairs are used as the retrieval set. And a random subset of 5,000 sample pairs from the retrieval set is used for offline training stage. 2) **NUS-WIDE** [6] is comprised of 195,834 image-text samples where each sample pair

Table 4: Comparison of training time (seconds) in different retrieval tasks when the code length is fixed to 128 bits.

Methods	**MIR Flickr**		**NUS-WIDE**		**MS COCO**	
	Image	Text	Image	Text	Image	Text
OKH [8]	1.31	1.15	1.04	1.08	**2.63**	**2.48**
OSH [1]	54.15	15.80	51.57	16.03	226.08	76.96
MIHash [2]	78.08	68.10	77.90	64.68	279.98	257.42
HCOH [14]	3.86	1.54	3.74	1.14	12.26	6.55
FOMH-*map*	31.92	2.79	34.45	2.43	117.26	21.04
Ours	**0.61**	**0.60**	**0.73**	**0.76**	5.82	5.75
	Both Image and Text		Both Image and Text		Both Image and Text	
MFH [25]	56.91		60.25		591.68	
MFKH [18]	37.82		39.47		112.84	
MAH [17]	107.72		114.30		102.98	
MVLH [23]	219.81		213.15		444.89	
MvDH [22]	1774.94		1821.82		7994.75	
DMVH [35]	314.01		308.49		453.62	
ODMVH [31]	172.48 (37 bits)		187.09 (41 bits)		427.92 (81 bits)	
OKH [8]	1.87		1.89		**2.39**	
OSH [1]	47.82		49.58		119.81	
MIHash [2]	84.65		83.64		146.02	
HCOH [14]	4.94		4.17		5.34	
FOMH-*map*	33.89		35.65		130.37	
Ours	**1.10**		**1.29**		11.58	

belongs to at least one of 21 categories. We randomly select 100 images for each category and the corresponding 2,085 image-text samples to form the query set. The remaining 193,749 image-text samples are as the retrieval set, and 5,000 instances from them are randomly selected for offline training. In experiments, each image is represented as a 4,096-dimensional feature extracted by the Caffe implementation of VGG Net and each text is represented as a 1,000-dimensional bag-of-words feature. 3) **MS COCO** [15] contains 82,783 training images and 40,504 validation images with at least one of 80 object categories. In experiments, each image is represented as a 4,096-dimensional feature extracted by the Caffe implementation of VGG Net and each text is represented as a 2,000-dimensional bag-of-words feature. In our experiments, we select the 82,783 images as the retrieval set and randomly select 18,000 images within them as the training set. Besides, we randomly select 80 images for each class from the validation images and we obtain 5,981 images to form the query set.

Evaluation Protocols and Baselines We adopt Mean Average Precision (MAP) [26, 44] 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 [26]. Larger value indicates the better retrieval performance.

Our method can flexibly generate the hash codes according to the specific query modality types. It can be employed on both uni-modal retrieval and multi-modal retrieval. Firstly, we conduct comparison experiment when only image or text is provided in both offline training stage and retrieval stage. Under such circumstances, we compare the proposed method with four state-of-the-art uni-modal online hashing methods, including Online Kernel Hashing (OKH) [8], Online Supervised Hashing (OSH) [1], Mutual Information (MIHash) [2] and Hadamard Codebook based Online Hashing (HCOH) [14]. Second, we conduct comparison experiment when both image and text modality are provided in both offline training stage and retrieval stage. We compare the proposed method with six state-of-the-art batch-based multi-modal hashing methods, including Multiple Feature Hashing (MFH) [25], Multiple Feature Kernel Hashing (MFKH) [18], Multi-modal Alignment Hashing (MAH) [17], Multiview Latent Hashing (MVLH) [23], Multi-modal Discrete Hashing (MVDH) [22], Discrete Multi-modal Hashing (DMVH) [35], and an online multi-modal hashing method Online Dynamic Multi-view Hashing (ODMVH) [31]. We also compare the performance with uni-modal online hashing methods by importing the models with the concatenated multi-modal features. Those methods have been discussed in Section 2. Note that none of these baselines can handle the modality-missing problems, while our method can flexibly generate the hash codes according to the specific query contents.

Implementation Details The proposed FOMH has several parameters: α, β, and γ in Eq.(5), ρ in Eq.(13), and the number of anchors p. α, β and ρ are balance parameters to support asymmetric semantic supervision, γ is a regularization parameter to avoid over-fitting. The best performance is achieved when $\{\alpha = 10^{-1}, \beta = 10, \gamma = 10^{-3}, \rho = 10^{-1}\}$, $\{\alpha = 10^{-5}, \beta = 10^{-1}, \gamma = 10^{-3}, \rho = 10^{-1}\}$, and $\{\alpha = 10^{-5}, \beta = 10^{-1}, \gamma = 10^{-5}, \rho = 10^{-3}\}$ on MIR Flickr, NUS-WIDE and MS COCO, respectively. The best performance of FOMH is achieved when the number of anchors p is set as 500, 1,000 and 1,000 on MIR Flickr, NUS-WIDE and MS COCO, respectively. At each round, a new data chunk is added to the database, and the size of each chunk is set to 1000. The training data of MIR Flickr and NUS-WIDE are both split to 5 chunks, while the training data of MS COCO is split to 18 chunks. 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.

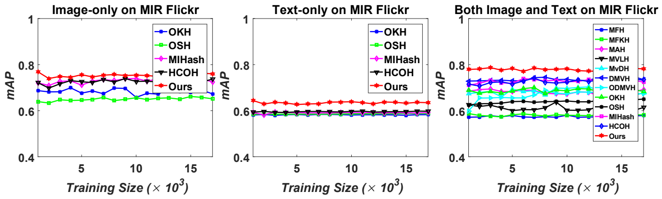

Figure 2: MAP comparison results with training size on MIR Flickr.

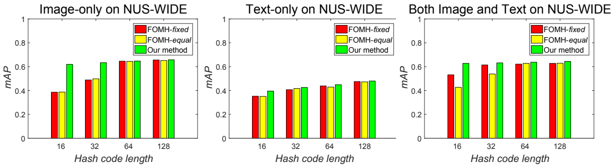

Figure 3: MAP comparison results of FOMH-*fixed*, FOMH-*equal* and our method on NUS-WIDE.

Retrieval Accuracy Comparison The MAP values of all compared methods varying with different hash code lengths on three datasets (MIR Flickr, NUS-WIDE and MS COCO) are presented in Table 1, 2, and 3. Note that, ODMVH [31] adaptively augments the hash codes without the predefining the code length, thus we dynamically augmented code length and record its mAP value. Figure 2 shows the performance comparison with respect to different numbers of training samples on MIR Flickr when the code length fixed to 64 bits. From Table 1 and 2, we can find that our method is superior to other online hashing baselines, when only one modality feature is provided. According to Table 3, we can find that when both image and text modalities are provided, our method still outperforms all the batch-based and online multi-modal hashing methods in various retrieval scenarios.

Run Time Comparison In this subsection, we conduct experiments to compare the training efficiency between FOMH and baselines. We fix the code length to 128 bits and present the comparison results of training time in Table 4. We can find that, whether only one modality is provided or multiple modalities are provided, on MIR Flickr and NUS-WIDE, the training efficiency of FOMH is higher than that of the baselines. On MS COCO, although OKH is more efficient, FOMH obtains acceptable training efficiency and more satisfactory performance. When multiple modalities are provided, on three datasets, the training process will consume more time compared with the case of only one modality is provided. The

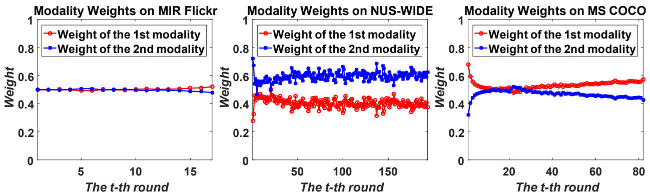

Figure 4: Modality weights variations with the streaming data.

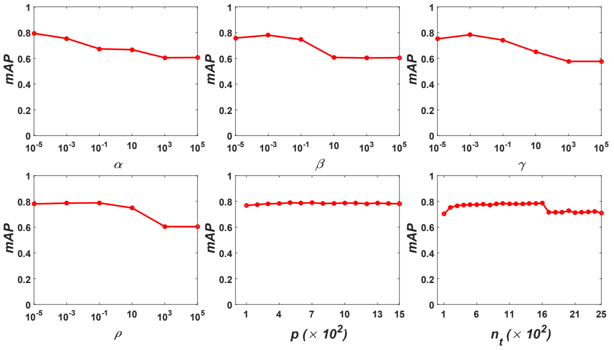

Figure 5: Parameter variations on MIR Flickr.

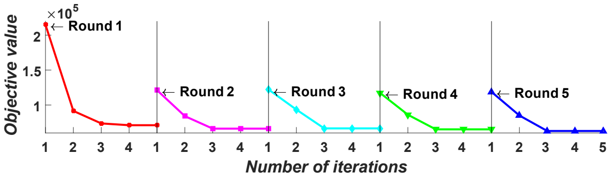

Figure 6: Convergence analysis on MIR Flickr.

reason is that when multiple modalities are provided, FOMH should take more time to learn weights and fuse different modalities.

Model Analysis To verify the effectiveness of our method, we also design several variants. 1) FOMH-*map*: it simply imports the original features of different modalities into the hashing model. This variant is to evaluate the effects of nonlinearly multi-modal feature mapping on training and retrieval accuracy. 2) FOMH-*supe*: it learns hash codes without any semantic supervision. This variant is to evaluate the effects of asymmetric semantic supervision on enhancing the discriminative capability of hash codes. 3) FOMH-*relax*: it firstly relaxes the discrete constraints and then obtains binary codes by mean-thresholding. This variant is to validate the effects of discrete online hash optimization on hashing performance. We record the comparison retrieval performance in Table 1, 2 and 3. Beside, we develop two variations to validate the effectiveness of the virtual weights. 1) FOMH-*fixed*: It applies fixed weights learned from the offline training stage for retrieval stage. 2) FOMH-*equal*: It fixes the weight of each modality to 1 at both the offline training stage and retrieval stage. Figure 3 shows the comparison results of FOMH-*fixed*, FOMH-*equal*, and our method on NUS-WIDE. From the results, we can find that the performance of our method is obviously higher than that of all other variants. In addition, the training efficiency of FOMH-*map* is also presented in Table 4. We can clearly observe that the training efficiency of our method is higher than that of this variant, which validates that the nonlinearly feature mapping can reduce the computational complexity.

Figure 4 shows the modality weights variations with the arrival of streaming data on three datasets when the code length is fixed to 64. We can find that the modality weights change adaptively according to the streaming data.

Figure 5 shows the performance variations with the involved parameters α, β, γ, ρ, the number of anchor points and the size of a chunk on MIR Flickr with the code length fixed to 64. Similar results can be found on other code lengths and datasets. We can find that the best performance is achieved when α is fixed to 10^{-5}, β is fixed to 10^{-3}, γ is fixed to 10^{-3}, stable performance can be achieved when ρ in a range of $\{10^{-5}, 10^{-3}, 10^{-1}\}$, the number of anchor points in a range from 500 to 1,300, and the size of a chunk in a range from 900 to 1500.

At each round, when the new streaming data arrives, $\mathbf{B}_t$ is updated based on iterative process. Figure 6 shows the convergence on MIR Flickr for the streaming input data at the t-th round. As can be seen, when $t = 1$, it merely takes four iterations to get convergence. Furthermore, when $t > 1$, it costs only three iterations for updating $\mathbf{B}_t$, which validates that the updating of variables in our method monotonically decreases the objective function value and eventually reaches a local minimum efficiently.

5 CONCLUSION

In this paper, we propose a supervised *Flexible Online Multi-modal Hashing* (FOMH), which is efficient on handling the streaming multi-modal data. We develop a self-weighted and flexible multi-modal fusion strategy, so that the heterogeneous modalities are flexibly fused even if part of the modalities is missing, and the modality weights are learned adaptively according to the streaming data. In addition, discriminative common hash code in FOMH is efficiently learned with the asymmetric semantic supervision. Furthermore, with a fast discrete online optimization, the hash code in FOMH can be directly updated in online fashion, with avoiding the propagation of binary quantization errors in online learning process. Extensive experiments demonstrate the superiority of the proposed approach.

6 ACKNOWLEDGMENTS

The work is partially supported by the National Natural Science Foundation of China (Nos. 61772322, 61802236, U1836216, 61572298), the Technology and Development Project of Shandong (Nos. 2017GGX10117 and 2017CXGC0703), and in part by the Natural Science Foundation of Shandong, China (No. ZR2019QF002)

REFERENCES

[1] Fatih Çakir, Sarah Adel Bargal, and Stan Sclaroff. 2017. Online supervised hashing. *CVIU* 156 (2017), 162–173.

[2] Fatih Çakir, Kun He, Sarah Adel Bargal, and Stan Sclaroff. 2017. MIHash: Online Hashing with Mutual Information. In *ICCV*. 437–445.

[3] Fatih Çakir and Stan Sclaroff. 2015. Adaptive Hashing for Fast Similarity Search. In *ICCV*. 1044–1052.

[4] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose Catherine Kanjirathinkal, and Mohan S. Kankanhalli. 2019. MMALFM: Explainable Recommendation by Leveraging Reviews and Images. *TOIS* 37, 2 (2019), 16:1–16:28.

[5] Zhiyong Cheng, Jialie Shen, and Steven C. H. Hoi. 2016. On Effective Personalized Music Retrieval by Exploring Online User Behaviors. In *SIGIR*. 125–134.

[6] 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.

[7] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *JMLR* 7 (2006), 551–585.

[8] Long-Kai Huang, Qiang Yang, and Wei-Shi Zheng. 2013. Online Hashing. In *IJCAI*. 1422–1428.

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

[10] Yoonseop Kang, Saehoon Kim, and Seungjin Choi. 2012. Deep Learning to Hash with Multiple Representations. In *ICDM*. 930–935.

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

[12] Cong Leng, Jiaxiang Wu, Jian Cheng, Xiao Bai, and Hanqing Lu. 2015. Online sketching hashing. In *CVPR*. 2503–2511.

[13] Mingbao Lin, Rongrong Ji, Hong Liu, Xiaoshuai Sun, Yongjian Wu, and Yunsheng Wu. 2019. Towards Optimal Discrete Online Hashing with Balanced Similarity. In *AAAI*.

[14] Mingbao Lin, Rongrong Ji, Hong Liu, and Yongjian Wu. 2018. Supervised Online Hashing via Hadamard Codebook Learning. In *ACM MM*. 1635–1643.

[15] 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.

[16] Anan Liu, Yuting Su, Weizhi Nie, and Mohan S. Kankanhalli. 2017. Hierarchical Clustering Multi-Task Learning for Joint Human Action Grouping and Recognition. *TPAMI* 39, 1 (2017), 102–114.

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

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

[19] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2010. Online Learning for Matrix Factorization and Sparse Coding. *JMLR* 11 (2010), 19–60.

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

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

[22] 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.

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

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

[25] 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.

[26] 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.

[27] Daixin Wang, Peng Cui, Mingdong Ou, and Wenwu Zhu. 2015. Deep Multimodal Hashing with Orthogonal Regularization. In *IJCAI*. 2291–2297.

[28] J. Wang, T. Zhang, j. song, N. Sebe, and H. T. Shen. 2018. A Survey on Learning to Hash. *TPAMI* 40, 4 (2018), 769–790.

[29] Meng Wang, Yue Gao, Ke Lu, and Yong Rui. 2013. View-Based Discriminative Probabilistic Modeling for 3D Object Retrieval and Recognition. *TIP* 22, 4 (2013), 1395–1407.

[30] Meng Wang, Hao Li, Dacheng Tao, Ke Lu, and Xindong Wu. 2012. Multimodal Graph-Based Reranking for Web Image Search. *TIP* 21, 11 (2012), 4649–4661.

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

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

[33] 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.

[34] Peng Yang, Peilin Zhao, and Xin Gao. 2018. Bandit Online Learning on Graphs via Adaptive Optimization. In *IJCAI*. International Joint Conferences on Artificial Intelligence Organization, 2991–2997.

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

[36] Tao Yao, Gang Wang, Lianshan Yan, Xiangwei Kong, Qingtang Su, and Caiming Zhang. 2019. Online latent semantic hashing for cross-media retrieval. *Pattern Recognition* 89 (2019), 1–11.

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

[38] 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).

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

[40] 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.

[41] L. Zhu, Z. Huang, Z. Li, L. Xie, and H. T. Shen. 2018. Exploring Auxiliary Context: Discrete Semantic Transfer Hashing for Scalable Image Retrieval. *TNNLS* 29, 11 (2018), 5264–5276.

[42] Lei Zhu, Zi Huang, Xiaobai Liu, Xiangnan He, Jiande Sun, and Xiaofang Zhou. 2017. Discrete Multimodal Hashing With Canonical Views for Robust Mobile Landmark Search. *TMM* 19, 9 (2017), 2066–2079.

[43] Lei Zhu, Jialie Shen, Liang Xie, and Zhiyong Cheng. 2017. Unsupervised Topic Hypergraph Hashing for Efficient Mobile Image Retrieval. *TCYB* 47, 11 (2017), 3941–3954.

[44] 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.