Online Enhanced Semantic Hashing: Towards Effective and Efficient Retrieval for Streaming Multi-Modal Data

Xiao-Ming Wu, Xin Luo*, Yu-Wei Zhan, Chen-Lu Ding, Zhen-Duo Chen, Xin-Shun Xu
School of Software, Shandong University
{wuxiaoming.alg, luoxin.lxin, zhanyuweilif, dingchenlu200103, chenzd.sdu}@gmail.com, xuxinshun@sdu.edu.cn

Abstract

With the vigorous development of multimedia equipment and applications, efficient retrieval of large-scale multi-modal data has become a trendy research topic. Thereinto, hashing has become a prevalent choice due to its retrieval efficiency and low storage cost. Although multi-modal hashing has drawn lots of attention in recent years, there still remain some problems. The first point is that existing methods are mainly designed in batch mode and not able to efficiently handle streaming multi-modal data. The second point is that all existing online multi-modal hashing methods fail to effectively handle unseen new classes which come continuously with streaming data chunks. In this paper, we propose a new model, termed Online enhAnced SemantIc haShing (OASIS). We design novel semantic-enhanced representation for data, which could help handle the new coming classes, and thereby construct the enhanced semantic objective function. An efficient and effective discrete online optimization algorithm is further proposed for OASIS. Extensive experiments show that our method can exceed the state-of-the-art models. For good reproducibility and benefiting the community, our code and data are already publicly available.

Introduction

With the rapid development of multimedia devices and applications, efficient retrieval of multimedia data has become a hot research topic. Learning to hash has arisen to be a promising choice because of its fast retrieval speed and low storage consumption (Tian, Ng, and Wang 2019; Chen et al. 2021a; Xu et al. 2020; Chen et al. 2021b; Weng and Zhu 2021). Roughly speaking, we could divide existing methods into uni-modal hashing (Kang, Li, and Zhou 2016; Wang et al. 2018b, 2019; Xie et al. 2020b), cross-modal hashing (Liu et al. 2019; Xie et al. 2020a; Jin, Li, and Tang 2020; Nie et al. 2020), and multi-modal hashing (Liu et al. 2012; Shen et al. 2015; Zhu et al. 2020a). Thereinto, multi-modal hashing requires that both database and query samples provide heterogeneous multi-modal features. More details among these three kinds of hashing can be found in (Zhu et al. 2020b). In this paper, we focus on multi-modal hashing which draws a significant need in real-world multimedia retrieval tasks while is relatively unexplored.

Despite the promising performance of hashing methods, failing to efficiently learn from streaming data may be an obstacle to applying them to real-world applications. Those batch-based hashing methods assume all training data are available in advance for training and such batch mode needs large space cost. To overcome the limitation, many efforts have been devoted to online hashing. Similarly, we could roughly classify the literature into online uni-modal hashing (Huang, Yang, and Zheng 2013; Cakir and Sclaroff 2015; Cakir et al. 2017; Chen, King, and Lyu 2017; Weng and Zhu 2020), online cross-modal hashing (Xie, Shen, and Zhu 2016; Qi, Wang, and Li 2017; Wang, Luo, and Xu 2020; Yi et al. 2021), and online multi-modal hashing (Xie et al. 2017; Lu et al. 2019a).

In online hashing literature, one of the most important problems is the class incremental problem, which means new (unknown) categories may appear along with new data chunks. However, most existing works fail to solve it and only a few solutions have been proposed. Some works rise to the challenge by means of offline coding strategies, such as Error Correcting Output Codes (Cakir, Bargal, and Sclaroff 2017) and Hadamard matrix (Lin et al. 2018, 2020). Some methods try to increase the number of hash bits for better representation (Mandal, Annadani, and Biswas 2018). Some efforts are made through training multiple complementary hash tables incrementally (Tian et al. 2020). Some methods design an end-to-end model to figure out this problem (Wu et al. 2019; Chen et al. 2019). However, these solutions are designed for uni-modal and cross-modal hashing while the class incremental problem in online multi-modal hashing is still an open problem without investigations. To the best of our knowledge, the only two online multi-modal hashing methods are Online Dynamic Multi-View Hashing (ODMVH) (Xie et al. 2017) and Flexible Online Multi-modal Hashing (FOMH) (Lu et al. 2019a). However, ODMVH is an unsupervised method and FOMH implicitly assumes that no new categories come with streaming data. In other words, no existing online multi-modal hashing can handle the scenario where new (unknown) categories continually appear along with new data chunks. Besides, we argue that the problem settings of some existing online hashing, which could tackle the class incremental problem, may be impractical: 1) some may relearn the hash codes of old data; 2) some may reuse the original features of old

[*] Corresponding author.

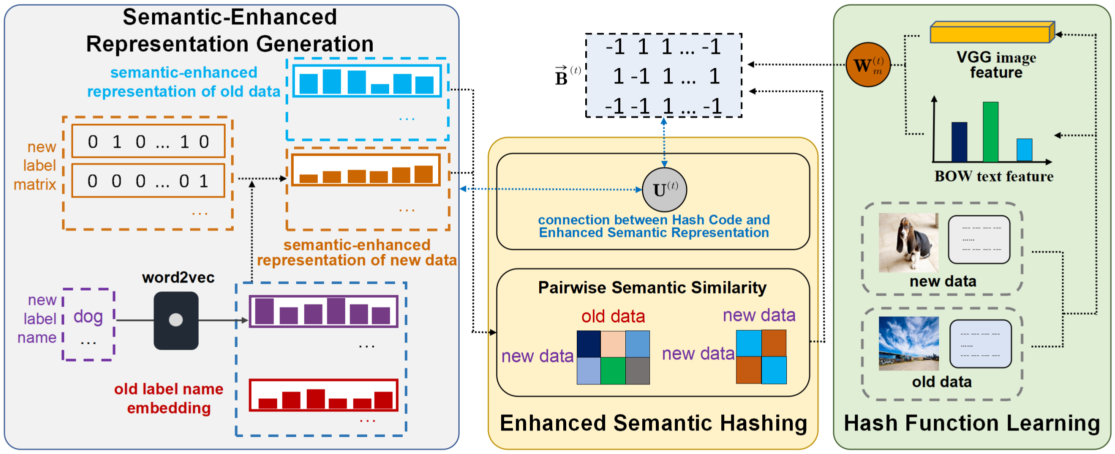

Figure 1: The framework of OASIS at the t-round.

data. When facing large-scale applications, those settings will become inefficient. Hence, in this paper, we formalize the problem by giving some constraints. 1) Multimedia data come in a streaming fashion. 2) The hash codes keep unchanged once learnt and the hash code length is fixed. 3) When updating the hash function, features of new data can be utilized but features of old data chunks are unusable. 4) New (unknown) categories may continually appear along with new data chunks.

To overcome the aforementioned limitations and satisfy the requirements, in this paper, we propose a novel method termed Online enhAnced SemantIc haShing (OASIS), which could learn hash codes and functions when the multimedia instances arrive in streaming fashion. As far as we know, this is the first attempt to investigate the class incremental problem in the context of multi-modal hashing. The main contributions are as follows.

- **Task contribution.** We thoroughly investigate the online multi-modal hashing and define a problem setting to better organize this research topic. The proposed task is more challenging and more practical.
- **Technical contribution.** This paper conceives a new online multi-modal hashing method and proffers an efficient and effective discrete online optimization algorithm. We generate a novel semantic-enhanced representation for data and thereby construct a semantically enhanced objective function. Besides, with the designed representation, we could handle new categories well. Extensive experiments on two benchmark datasets show that our method can exceed the state-of-the-art models.
- **Community contribution.** Our model, training and evaluation code are already publicly available[1]. We hope our work may become one of the enablers for the valuable but relatively unexplored topic and the learning to hash community.

Method

As shown in Figure 1, our proposed OASIS contains several modules, i.e., semantic-enhanced representation generation, enhanced semantic hashing, and hash function learning. Details of our OASIS are introduced in this section.

Notations and Problem Definition

In this paper, we assume that multi-modal data comes at a streaming manner. At the t-th round, suppose that we have $N^{(t)}$ old multi-modal samples $\widetilde{\mathbf{X}}^{(t)}$, which are observed before current round, and $n^{(t)}$ new ones $\vec{\mathbf{X}}^{(t)}$, where $N^{(t)} = n^{(1)} + \cdots + n^{(t-1)}$. More specifically, as all samples contain features from M modalities, we denote the feature matrix of old data from the m-th modality as $\widetilde{\mathbf{X}}_m^{(t)} \in R^{d_m \times N^{(t)}}$ and the new samples' feature matrix of the m-th modality as $\vec{\mathbf{X}}_m^{(t)} \in R^{d_m \times n^{(t)}}$, where $m \in \{1, ..., M\}$ and d_m is the dimensionality. Meanwhile, we can acquire the labels $\vec{\mathbf{L}}^{(t)} \in \{0,1\}^{(c_n^{(t)}+c_o^{(t)})\times n^{(t)}}$ of $\vec{\mathbf{X}}^{(t)}$, where $c_o^{(t)}$ is the number of old classes observed in former rounds, $c_n^{(t)}$ is the number of new classes which first appear in the t-th round and $c_o^{(t)} = c_n^{(1)} + \cdots + c_n^{(t-1)}$. Then, we learn the hash codes $\vec{\mathbf{B}}^{(t)} \in \{-1,1\}^{r\times n^{(t)}}$ for $\vec{\mathbf{X}}^{(t)}$, where r is the code length. Similarly, $\widetilde{\mathbf{L}}^{(t)}$ and $\widetilde{\mathbf{B}}^{(t)} \in \{-1,1\}^{r\times N^{(t)}} = [\vec{\mathbf{B}}^{(1)}, \cdots, \vec{\mathbf{B}}^{(t-1)}]$ are the label matrix and hash codes of $\widetilde{\mathbf{X}}_m^{(t)}$, respectively.

We let $\mathbf{I}$, $\mathbf{1}$, and $\mathbf{0}$ denote the identify matrix, an all-one matrix, and an all-zero matrix, respectively. Regarding the definition of the operator, we use $\mathrm{tr}(\cdot)$ to represent the trace operation of the matrix, $\|\cdot\|_F$ to represent the Frobenius form of the matrix, and $\mathrm{sign}(\cdot)$ to represent the sign function.

In this paper, we focus on the crucial but understudied task, i.e., online multi-modal hashing. We hope our model satisfies the following requirements. 1) Multi-modal data come in a streaming fashion ($\vec{\mathbf{X}}^{(1)}$, $\vec{\mathbf{X}}^{(2)}$, $\cdots$ and so on). 2) The hash codes keep unchanged once learnt and the hash

[1] https://github.com/DravenALG/OASIS

code length r is fixed. 3) When updating the hash function, features of new data $\vec{\mathbf{X}}^{(t)}$ can be utilized but $\widetilde{\mathbf{X}}^{(t)}$ is unusable. 4) New (unknown) categories may continually appear along with new data chunks ($c_n^{(t)} \geq 0$).

Formulation

Basic Hashing Formulation. We start with one of the commonest formulations in hashing (Kang, Li, and Zhou 2016), i.e., $\|r\mathbf{S} - \mathbf{B}^T\mathbf{B}\|_F^2$. One of the greatest weaknesses of this formulation comes from the size of the pairwise similarity matrix $\mathbf{S}$ which is square of the number of training samples. To provide a remedy to this dilemma, many efforts have been made and the frequently-used strategy in online hashing literature is redefining $\mathbf{S}$ as,

$$\mathbf{S} = 2\mathbf{P}^T\mathbf{P} - \mathbf{1}\mathbf{1}^T, \tag{1}$$

where $\mathbf{P}$ is the 2-norm normalized label matrix defined as $\mathbf{P}_j = \mathbf{L}_j/\|\mathbf{L}_j\|$, $\|\cdot\|$ represents the length of the vector, $\mathbf{L}_j$ and $\mathbf{P}_j$ represents the j-th column of $\mathbf{L}$ and $\mathbf{P}$, respectively. Then, the computational complexity can be reduced from square to linear. For the sake of accommodating to online streaming data, pairwise $\mathbf{S}^{(t)}$, whose size is $(N^{(t)} + n^{(t)}) \times (N^{(t)} + n^{(t)})$, at the t-th round could be split into four parts,

$$\begin{aligned}\mathbf{S}_{oo}^{(t)} &= 2\widetilde{\mathbf{P}}^{(t)T}\widetilde{\mathbf{P}}^{(t)} - \mathbf{1}\mathbf{1}^T, \mathbf{S}_{no}^{(t)} = 2\vec{\mathbf{P}}^{(t)T}\widetilde{\mathbf{P}}^{(t)} - \mathbf{1}\mathbf{1}^T \\ \mathbf{S}_{on}^{(t)} &= 2\widetilde{\mathbf{P}}^{(t)T}\vec{\mathbf{P}}^{(t)} - \mathbf{1}\mathbf{1}^T, \mathbf{S}_{nn}^{(t)} = 2\vec{\mathbf{P}}^{(t)T}\vec{\mathbf{P}}^{(t)} - \mathbf{1}\mathbf{1}^T,\end{aligned} \tag{2}$$

where $\mathbf{S}_{oo}^{(t)}$ is the semantic similarity between old data, $\mathbf{S}_{on}^{(t)}$ is the semantic similarity between old data and new data, $\mathbf{S}_{no}^{(t)}$ is the semantic similarity between new data and old data, $\mathbf{S}_{nn}^{(t)}$ is the semantic similarity between new data, $\vec{\mathbf{P}}^{(t)}$ is the 2-norm normalized label matrix of new data and $\widetilde{\mathbf{P}}^{(t)}$ is the 2-norm normalized label matrix of old data. The only supervised online multi-modal hashing (Lu et al. 2019a) keeps $\mathbf{S}_{nn}^{(t)}$ only and omits other three. Such strategy may lose the information of old data. The state-of-the-art online cross-modal hashing (Wang, Luo, and Xu 2020) learns from the above four parts and is endowed with impressive results. However, it is impossible to handle class incremental problem because the incorrect dimensions for matrix multiplication error will happen when performing $\vec{\mathbf{P}}^{(t)T}\widetilde{\mathbf{P}}^{(t)}$ operation if new classes come. In other words, $\mathbf{S}_{on}^{(t)}$ and $\mathbf{S}_{no}^{(t)}$ cannot be calculated due to the different number of the classes of the new and old data.

We propose a novel solution to tackle the class incremental problem which is detailedly shown below.

Semantic-Enhanced Representation. Label names are often naturally well separated from each other and contain good expressions of category-specific semantics (Wang et al. 2018c). Thus, we introduce the embedding of label names into our framework to gain more powerful semantic information to guide the hash learning.

In online setting, as new labels may continually appear along with new data chunk, we use $\mathbf{K}_o^{(t)} \in \mathbb{R}^{c_o^{(t)} \times f}$ to represent the old labels' name embeddings and $\mathbf{K}_n^{(t)} \in \mathbb{R}^{c_n^{(t)} \times f}$ to denote new labels' name embeddings, where f is the dimensionality of word2vec vector. By concatenating them, the overall label name embeddings can be obtained $\mathbf{K}^{(t)} \in \mathbb{R}^{(c_n^{(t)} + c_o^{(t)}) \times f}$. In our work, Google's word2vec model (Mikolov et al. 2013) is leveraged and $f = 300$.

Then, we propose a semantic-enhanced matrix $\vec{\mathbf{G}}^{(t)} \in R^{f \times n^{(t)}}$ that comprises both label matrix and label name embeddings. The semantic-enhanced matrix is defined as,

$$\vec{\mathbf{G}}_j^{(t)} = \frac{\mathbf{K}^{(t)T}\vec{\mathbf{L}}_j^{(t)}}{\|\mathbf{K}^{(t)T}\vec{\mathbf{L}}_j^{(t)}\|}, j = 1, 2 \cdots n^{(t)}, \tag{3}$$

where $\vec{\mathbf{L}}_j^{(t)}$ represents the j-th column of $\vec{\mathbf{L}}^{(t)}$, $\vec{\mathbf{G}}_j^{(t)}$ represents the j-th column of $\vec{\mathbf{G}}^{(t)}$, and $\|\cdot\|$ represents the length of vectors. Furthermore, we turn each column of the matrix into a unit vector, which can easily express the cosine similarity when calculating $\mathbf{S}^{(t)}$. The semantic-enhanced matrix offers instances semantic-rich representations and plays an important role in guiding the learning of OASIS.

Enhanced Semantic Hashing. Now, let's revisit Eq.(2). With the semantic-enhanced representation of instances, we could redefine the pairwise semantic similarity as follows,

$$\begin{aligned}\mathbf{S}_{oo}^{(t)} &= \widetilde{\mathbf{G}}^{(t)T}\widetilde{\mathbf{G}}^{(t)}, \mathbf{S}_{no}^{(t)} = \vec{\mathbf{G}}^{(t)T}\widetilde{\mathbf{G}}^{(t)}, \\ \mathbf{S}_{on}^{(t)} &= \widetilde{\mathbf{G}}^{(t)T}\vec{\mathbf{G}}^{(t)}, \mathbf{S}_{nn}^{(t)} = \vec{\mathbf{G}}^{(t)T}\vec{\mathbf{G}}^{(t)},\end{aligned} \tag{4}$$

where $\widetilde{\mathbf{G}}^{(t)} \in \mathbb{R}^{f \times N^{(t)}}$ denotes the semantic-enhanced representation of old samples and can be obtained before round t. It is worth noting that Eq.(4) could naturally handle the class incremental problem. As the size of $\vec{\mathbf{G}}^{(t)}$ is $f \times n^{(t)}$ and the the size of $\widetilde{\mathbf{G}}^{(t)}$ is $f \times N^{(t)}$, the dimensions for matrix multiplication is correct. We elaborately bypass the obstacle mentioned above and could construct the connections between new and old classes. Then, we propose the loss guided by enhanced semantic pairwise similarity as,

$$\begin{aligned}&\min_{\vec{\mathbf{B}}^{(t)} \in \{-1,1\}^{r \times n^{(t)}}} \|r\mathbf{S}_{nn}^{(t)} - \vec{\mathbf{B}}^{(t)T}\vec{\mathbf{B}}^{(t)}\|_F^2 \\ &+ \|r\mathbf{S}_{on}^{(t)} - \widetilde{\mathbf{B}}^{(t)T}\vec{\mathbf{B}}^{(t)}\|_F^2 + \|r\mathbf{S}_{no}^{(t)} - \vec{\mathbf{B}}^{(t)T}\widetilde{\mathbf{B}}^{(t)}\|_F^2,\end{aligned} \tag{5}$$

where $\mathbf{S}_{oo}^{(t)}$ is omitted because to-be-learnt $\vec{\mathbf{B}}^{(t)}$ is irrelevant to it. In this equation, knowledge of both old and new classes can be embedded and may constructively guide the hash learning.

To further enhance the semantic information that guides the hash learning, we also directly construct the connection between hash codes and semantic-enhanced representation,

$$\begin{aligned}&\min_{\vec{\mathbf{B}}^{(t)}, \mathbf{U}^{(t)}} \|\vec{\mathbf{G}}^{(t)} - \mathbf{U}^{(t)}\vec{\mathbf{B}}^{(t)}\|_F^2 + \|\widetilde{\mathbf{G}}^{(t)} - \mathbf{U}^{(t)}\widetilde{\mathbf{B}}^{(t)}\|_F^2 \\ &+ \theta\|\mathbf{U}^{(t)}\|_F^2, \ \ s.t. \vec{\mathbf{B}}^{(t)} \in \{-1, 1\}^{r \times n^{(t)}},\end{aligned} \tag{6}$$

where $\mathbf{U}^{(t)}$ is a mapping matrix and θ is a hyperparameter controlling the regularization term. Here, we use the same mapping matrix for both old and new data in order that knowledge previously learnt could be compatible with new knowledge. Note that, although this loss looks like the widely-used term which uses hash codes for classification, the idea behind Eq.(6) is totally different.

Hash Function Learning. Currently, linear hash functions occupy the mainstream position in online hashing domain while neural network based methods are extremely scarce. The possible reasons may be that: 1) Online hashing focuses more on efficiency as almost all publications report the training time comparison results; 2) Training of neural network based hash functions is much more time-consuming than the training of linear functions. Following the vast majority, we design our hash function learning module using the efficient and straightforward linear mapping as follows,

$$\begin{aligned}&\min_{\vec{\mathbf{B}}^{(t)}\in\{-1,1\}^{r\times n^{(t)}},\mathbf{W}_m^{(t)}}\sum_{m=1}^{M}(\|\vec{\mathbf{B}}^{(t)}-\mathbf{W}_m^{(t)}\vec{\mathbf{X}}_m^{(t)}\|_F^2\\&+\|\widetilde{\mathbf{B}}^{(t)}-\mathbf{W}_m^{(t)}\widetilde{\mathbf{X}}_m^{(t)}\|_F^2+\delta\|\mathbf{W}_m^{(t)}\|_F^2),\end{aligned}\tag{7}$$

where M represents the total number of modalities, $\mathbf{W}_m^{(t)}$ is the projection of the m-th modality, and δ is a hyperparameter balancing the regularization term. Although Eq.(7) is simple, it has several advantages beyond simplicity and efficiency: 1)This equation reduces the quantization loss between the learnt hash code and the real-valued mapping results; 2) The catastrophic forgetting problem could be alleviated through simultaneously considering old and new data. Although above loss contains $\widetilde{\mathbf{X}}_m^{(t)}$, our method still meets the third requirement as can be seen in **Algorithm 1** whose inputs do not need features of old data chunks. This is because of the proposed novel online optimization which is presented below.

Overall Objective Function. In summary, by combining all metioned modules together and creating reasonable modifications, the total loss function of OASIS is shown below,

$$\begin{aligned}&\min_{\vec{\mathbf{B}}^{(t)}\in\{-1,1\}^{r\times n^{(t)}},\vec{\mathbf{V}}^{(t)},\mathbf{U}^{(t)},\mathbf{W}_m^{(t)}}\|\vec{\mathbf{B}}^{(t)}-\vec{\mathbf{V}}^{(t)}\|_F^2+\\&\alpha(\|r\mathbf{S}_{nn}^{(t)}-\vec{\mathbf{B}}^{(t)T}\vec{\mathbf{V}}^{(t)}\|_F^2+\|r\mathbf{S}_{on}^{(t)}-\widetilde{\mathbf{B}}^{(t)T}\vec{\mathbf{V}}^{(t)}\|_F^2\\&+\|r\mathbf{S}_{no}^{(t)}-\vec{\mathbf{B}}^{(t)T}\widetilde{\mathbf{V}}^{(t)}\|_F^2)+\beta(\|\vec{\mathbf{G}}^{(t)}-\mathbf{U}^{(t)}\vec{\mathbf{V}}^{(t)}\|_F^2\\&+\|\widetilde{\mathbf{G}}^{(t)}-\mathbf{U}^{(t)}\widetilde{\mathbf{V}}^{(t)}\|_F^2+\theta\|\mathbf{U}^{(t)}\|_F^2)+\gamma\sum_{m=1}^{M}(\|\vec{\mathbf{B}}^{(t)}\\&-\mathbf{W}_m\vec{\mathbf{X}}_m^{(t)}\|_F^2+\|\widetilde{\mathbf{B}}^{(t)}-\mathbf{W}_m^{(t)}\widetilde{\mathbf{X}}_m^{(t)}\|_F^2+\delta\|\mathbf{W}_m\|_F^2)\\&s.t.,\vec{\mathbf{V}}^{(t)}\vec{\mathbf{V}}^{(t)T}=n^{(t)}\mathbf{I}_r,\vec{\mathbf{V}}^{(t)}\mathbf{1}^T=\mathbf{0},\end{aligned}\tag{8}$$

where $\vec{\mathbf{V}}^{(t)}$ is the real-valued approximation of hash code with several constraints, α, β, and γ are tradeoff parameters. By using $\vec{\mathbf{V}}^{(t)}$ to approximate $\vec{\mathbf{B}}^{(t)}$, the hard optimization problem of $\vec{\mathbf{B}}^{(t)}$ can be simplified. Besides, the constraint of $\vec{\mathbf{V}}^{(t)}\vec{\mathbf{V}}^{(t)T}=n^{(t)}\mathbf{I}_r$ can make each bit represent as much information as possible, and the $\vec{\mathbf{V}}^{(t)}\mathbf{1}^T=\mathbf{0}$ constraint can make the hash code more discriminative.

Online Optimization

We propose a novel discrete online optimization for Eq.(8) to learn hash codes and function at the t-th round. Specifically, we optimize the variables one by one and repeat the procedure several times until convergence. At round t, the optimization steps are presented in the following.

Step 1: Update $\vec{\mathbf{V}}^{(t)}$. When other variables remain unchanged, the optimization problem of $\vec{\mathbf{V}}^{(t)}$ can be simplified to the following form.

$$\begin{aligned}&\max_{\vec{\mathbf{V}}^{(t)}}\ \mathrm{tr}(\mathbf{Z}\vec{\mathbf{V}}^{(t)T}),\\&s.t.\vec{\mathbf{V}}^{(t)}\vec{\mathbf{V}}^{(t)T}=n^{(t)}\mathbf{I}_r,\vec{\mathbf{V}}^{(t)}\mathbf{1}^T=\mathbf{0},\end{aligned}\tag{9}$$

where $\mathbf{Z}=2\alpha r\mathbf{D}_1^{(t)}\vec{\mathbf{G}}^{(t)}+\vec{\mathbf{B}}^{(t)}+\gamma\mathbf{U}^{(t)T}\vec{\mathbf{G}}^{(t)}$ and $\mathbf{D}_1^{(t)}=\vec{\mathbf{B}}^{(t)}\vec{\mathbf{G}}^{(t)T}+\mathbf{D}_1^{(t-1)}$. Notably, $\mathbf{D}_1$ is an **intermediate variable**. By temporarily storing $\mathbf{D}_1^{(t-1)}$, which is calculated at the previous $(t-1)$-th round, and directly using it at the current t-th round, the calculation of $\mathbf{D}_1^{(t)}$ can be very efficient.

Let us return to the optimization of $\vec{\mathbf{V}}^{(t)}$. The simplified optimization problem in Eq.(9) is similar to the form in (Liu et al. 2014) and can be optimized as follows. First, we perform the eigenvalue decomposition of $\mathbf{ZJZ}^T$ and the formula is as follows,

$$\mathbf{ZJZ}^T=[\mathbf{O}\ \overline{\mathbf{O}}][\begin{matrix}\mathbf{\Sigma}^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{matrix}][\mathbf{O}\ \overline{\mathbf{O}}],\tag{10}$$

where $\mathbf{J}=\mathbf{I}-\frac{1}{n^{(t)}}\mathbf{1}\mathbf{1}^T$. Then we calculate $\mathbf{N}=\mathbf{JZ}^T\mathbf{O}\mathbf{\Sigma}^{-1}\in R^{n^{(t)}\times r'}$, where r' is the number of positive eigenvalues. The $\overline{\mathbf{N}}\in R^{n^{(t)}\times(r-r')}$ is initially set to a random matrix followed by the Gram-Schmidt orthogonalization. Finally, we can get the solution of $\vec{\mathbf{V}}^{(t)}$,

$$\vec{\mathbf{V}}^{(t)}=\sqrt{n^{(t)}}[\mathbf{O}\ \overline{\mathbf{O}}][\mathbf{N}\ \overline{\mathbf{N}}]^T.\tag{11}$$

Step 2: Update $\mathbf{U}^{(t)}$. Fixing other variables and setting the derivative of Eq.(8) w.r.t. $\mathbf{U}^{(t)}$ to zero, the update formula can be obtained as follows,

$$\mathbf{U}^{(t)}=\mathbf{D}_2^{(t)}(\mathbf{D}_3^{(t)}+\theta\mathbf{I})^{-1},\tag{12}$$

where $\mathbf{D}_2^{(t)}=\vec{\mathbf{G}}^{(t)}\vec{\mathbf{V}}^{(t)T}+\mathbf{D}_2^{(t-1)}$ and $\mathbf{D}_3^{(t)}=\vec{\mathbf{V}}^{(t)}\vec{\mathbf{V}}^{(t)T}+\mathbf{D}_3^{(t-1)}$. Notably, $\mathbf{D}_2$ and $\mathbf{D}_3$ are **intermediate variables**. By temporarily storing them at last round and directly using them at current round, the optimization can be extremely efficient.

Step 3: Update $\vec{\mathbf{B}}^{(t)}$. When other variables are fixed, the generation of hash codes can be simplified as,

$$\max_{\vec{\mathbf{B}}^{(t)}}\ \mathrm{tr}(\mathbf{Q}\vec{\mathbf{B}}^{(t)T}),\quad s.t.\ \vec{\mathbf{B}}^{(t)}\in\{-1,1\}^{r\times n^{(t)}},\tag{13}$$

where $\mathbf{Q}=2\alpha r\mathbf{D}_4^{(t)}\vec{\mathbf{G}}^{(t)}+\vec{\mathbf{V}}^{(t)}+\gamma\sum_{m=1}^{M}\mathbf{W}_m^{(t)}\vec{\mathbf{X}}_m^{(t)}$ and $\mathbf{D}_4^{(t)}=\vec{\mathbf{V}}^{(t)}\vec{\mathbf{G}}^{(t)T}+\mathbf{D}_4^{(t-1)}$. Notably, similar with $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$, $\mathbf{D}_4$ is called **intermediate variable** and is helpful for effcient opitmization.

Because of $\mathrm{tr}(\mathbf{Q}\vec{\mathbf{B}}^{(t)T})=\sum_{i,j}\mathbf{Q}_{ij}\vec{\mathbf{B}}_{ij}^{(t)}$ and $\vec{\mathbf{B}}^{(t)}\in\{-1,1\}^{r\times n^{(t)}}$, we can let $\vec{\mathbf{B}}_{ij}=1$ when $\mathbf{Q}_{ij}$ is positive and $\vec{\mathbf{B}}_{ij}=-1$ when $\mathbf{Q}_{ij}$ is negative to maximize the simplified optimization function, where i and j represent the i-th row and j-th column of the matrix. Therefore, we can get the following formula,

$$\vec{\mathbf{B}}^{(t)} = \mathrm{sign}(\mathbf{Q}). \tag{14}$$

Step 4: Update $\mathbf{W}_m^{(t)}$. Similar to the update of $\mathbf{U}^{(t)}$, the update formula of $\mathbf{W}_m^{(t)}$ can be given as follows,

$$\mathbf{W}_m^{(t)} = \mathbf{D}_5^{(t)}(\mathbf{D}_6^{(t)} + \delta\mathbf{I})^{-1}, \tag{15}$$

where $\mathbf{D}_5^{(t)} = \vec{\mathbf{B}}^{(t)}\vec{\mathbf{X}}_m^{(t)T} + \mathbf{D}_5^{(t-1)}$ and $\mathbf{D}_6^{(t)} = \vec{\mathbf{X}}_m^{(t)}\vec{\mathbf{X}}_m^{(t)T} + \mathbf{D}_6^{(t-1)}$. Here, we could temporarily store $\mathbf{D}_5^{(t-1)}$ and $\mathbf{D}_6^{(t-1)}$ at the last round and use them at t-th round to get $\mathbf{D}_5^{(t)}$ and $\mathbf{D}_6^{(t)}$ directly. In light of these intermediate variables, efficiency of the optimization can be ensured.

Overall Algorithm. For better understanding, we summarize the proposed online optimization in **Algorithm 1**. It's worth noting that although we can find $\widetilde{\mathbf{X}}_m^{(t)}$ in the objective function, the old feature is not needed in the real optimization. **Our model satisfies all four requirements raised in the Problem Definition Section**.

Besides, the sizes of $\mathbf{D}_i^{(t-1)}$ ($i = \{1, \cdots, 6\}$) are $r \times f$, $f \times r$, $r \times r$, $r \times f$, $r \times d_m$, and $d_m \times d_m$. Although these matrices contain rich information from old data, they are irrelevant with $N^{(t)}$.

Complexity Analysis. The complexity of updating $\vec{\mathbf{V}}^{(t)}$ is $O((10r + 3r^2 + 3rf + f)n^{(t)} + 2rf + r^3)$, the complexity of getting $\mathbf{U}^{(t)}$ is $O((2r + fr + f + r^2)n^{(t)} + 2r^2 + r^3 + fr^2)$, the complexity of learning $\vec{\mathbf{B}}^{(t)}$ is $O((5r + 2fr + f + \sum_{m=1}^{M} rd_m)n^{(t)} + rf)$, and the complexity of generating $\mathbf{W}_m^{(t)}$ is $O(M(d_m^2 + rd_m + d_m)n^{(t)} + M(2d_m^2 + d_m^3 + rd_m^2 + rd_m))$. It is worth noting that variables $\mathbf{D}_i^{(t-1)}$ ($i = \{1, \cdots, 6\}$) can be obtained at the previous round and used directly at this round. Thus, there is no need to consider the time complexity of calculating them.

From the above analysis, we can find that the online optimization is **linearly related** to the size of new data chunk $n^{(t)}$ and is **irrelevant** to the old database $N^{(t)}$. To conclude, our proposed optimization is efficient and scalable to large-scale online retrieval applications.

Retrieval Precedure

When query sample comes, we first learn its hash code $\mathbf{b}_{query} = \mathrm{sign}(\sum_{m=1}^{M} \mathbf{W}_m^{(t)}\mathbf{x}_{query-m})$, where $\mathbf{W}_m^{(t)}$ is the up to date hashing projection and $\mathbf{x}_{query-m}$ represents query's feature of the m-th modality.

Then, we can calculate the Hamming distance of hash codes between query and database to measure the similarity. Those instances, which are considered to be similar, can be returned as retrieval results.

Experiment

Experimental Settings

Datasets. In this paper, we chose two widely-used datasets, i.e., MIRFlickr (Huiskes and Lew 2008) and NUS-WIDE (Chua et al. 2009). **MIRFlickr** has $25,000$ instances and 24 categories in total. Following (Jiang and Li 2019), we removed instances with rare tags that appear less than 20 times and finally had $20,015$ instances left. Then, we passed images through pre-trained VGG network to obtain 4096-d deep features and expressed texts as 1386-d BOW features. **NUS-WIDE** has $269,648$ instances and 81 categories. Following the setting of (Lu et al. 2019a), only the top 21 most common categories are used and $195,834$ instances are finally left. Similar with MIRFlickr, we fed images into the VGG network to obtain 4096-d deep features and represented the text modality as 5018-d BOW features.

Algorithm 1: The optimization of OASIS at the t-th round.

Input: $\vec{\mathbf{X}}_m^{(t)}$ ($m = 1, 2...M$), $\vec{\mathbf{L}}^{(t)}$, $\vec{\mathbf{K}}^{(t)}$, and $\mathbf{D}_i^{(t-1)}$ ($i = \{1, \cdots, 6\}$).
Output: $\mathbf{D}_i^{(t)}$ ($i = \{1, \cdots, 6\}$), $\mathbf{B}^{(t)}$, and hash function.
Main Algorithm:

```
Randomly initialize $\mathbf{U}^{(t)}$, $\vec{\mathbf{B}}^{(t)}$, and $\mathbf{W}_m^{(t)}$.
while not converged or not reach the max iterations do
    Update $\vec{\mathbf{V}}^{(t)}$ with Eq.(11); save $\mathbf{D}_1^{(t)}$.
    Update $\mathbf{U}^{(t)}$ with Eq.(12); save $\mathbf{D}_2^{(t)}$ and $\mathbf{D}_3^{(t)}$.
    Update $\vec{\mathbf{B}}^{(t)}$ with Eq.(14); save $\mathbf{D}_4^{(t)}$.
    Update $\mathbf{W}_m^{(t)}$ with Eq.(15); save $\mathbf{D}_5^{(t)}$ and $\mathbf{D}_6^{(t)}$.
end while
```

Evaluation Protocols. Mean Average Precision (MAP) is adopted as the evaluation criterion and larger value indicates better performance. As stated in above sections, during retrieval phase, we calculated the Hamming distance of hash codes between queries and database to measure the similarity. As both MIRFlickr and NUS-WIDE are multi-label datasets, we considered two samples to be similar if they share at least one common label.

Furthermore, we have designed **three types of online experimental settings**. The **first** online setting assumes that all categories are known before training. In other words, for t from 2 to the maximum, $c_n^{(t)}$ (the number of new classes which first appear at the t-th round) is always equal to 0. On the contrary, the second and third settings simulate the scenario where new (unknown) categories continually appear along with new data chunks ($c_n^{(t)} > 0$). For the **second** online setting, new data carries some old categories which first appear before current round and some new categories. For the **third** setting, all categories of new data are new (unknown).

The first online setting: For both datasets, we randomly selected $2,000$ samples to form the test set and left the remaining samples as training set. That is, the size of training set of MIRFlickr is $18,015$ and NUS-WIDE has $193,834$ training points. The first 9 chunks of MIRFlickr have the same size of $2,000$ while the 10-th chunk includes the remaining 15 samples. For NUS-WIDE, size of the first 19 chunks is $10,000$ while the remaining $3,834$ samples constitute the 20-th chunk. All data chunks are constructed by randomly picking up samples. The second online setting: MIRFlickr is divided into 10 rounds to adapt to the online

Table 1: MAP results at the last round.

Experimental Settings	Method	MIRFlickr				NUS-WIDE			
		16bit	32bit	64bit	128bit	16bit	32bit	64bit	128bit
the first setting	FOMH (Lu et al. 2019a)	0.5828	0.5759	0.5836	0.5864	0.5011	0.5371	0.5625	0.5237
	OMH-DQ (Lu et al. 2019b)	0.6945	0.6842	0.7158	0.7025	0.6242	0.6508	0.6873	0.7094
	SAPMH (Zheng et al. 2020)	0.5995	0.7139	0.7134	0.7108	0.4380	0.4828	0.4396	0.4952
	OASIS	**0.8314**	**0.8457**	**0.8534**	**0.8553**	**0.7727**	**0.7933**	**0.8026**	**0.8041**
the second setting	FOMH (Lu et al. 2019a)	0.6040	0.6293	0.6553	0.6679	0.4811	0.4931	0.4931	0.4931
	OMH-DQ (Lu et al. 2019b)	0.6763	0.6783	0.7005	0.6918	0.6230	0.7758	0.7975	0.8031
	SAPMH (Zheng et al. 2020)	0.6559	0.6550	0.6898	0.6930	0.5512	0.4381	0.6504	0.5443
	OASIS	**0.8199**	**0.8119**	**0.8247**	**0.8220**	**0.7950**	**0.8069**	**0.8132**	**0.8335**
the third setting	FOMH (Lu et al. 2019a)	0.0547	0.0485	0.0514	0.0501	0.0075	0.0075	0.0075	0.0075
	OMH-DQ (Lu et al. 2019b)	0.0519	0.0834	0.1199	0.1686	0.0197	0.0282	0.0330	0.0328
	SAPMH (Zheng et al. 2020)	0.0400	0.0521	0.0643	0.0703	0.0178	0.0200	0.0203	0.0229
	OASIS	**0.1551**	**0.1248**	**0.3870**	**0.3996**	**0.3281**	**0.3357**	**0.3042**	**0.3124**

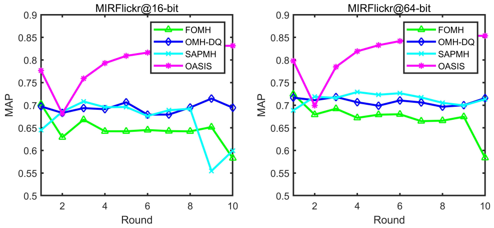

Figure 2: MAP results at all rounds on MIRFlickr.

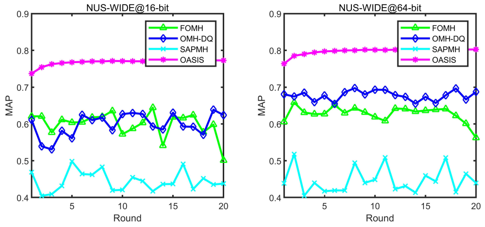

Figure 3: MAP results at all rounds on NUS-WIDE.

scene, and NUS-WIDE is divided into 20 rounds. Specifically, we assigned some labels for each round, ensuring that labels at the new round include the labels appear at previous rounds. Then, we constructed eligible data according to the set of labels at each round. It is worth noting that the data at the new round would exclude the data at previous rounds to avoid duplication. Subsequently, the training data and test data of each round are randomly selected from the data of each round at a ratio of $9 : 1$. The third online setting: First, we stipulated that MIRFlickr has 7 rounds and NUS-WIDE has 10 rounds. We allocated some labels for each round. Different from the second setting, we set labels at the new round be totally different from the labels at the last rounds. Then, we selected the data based on the labels at each round. The training data and test data of each round are randomly selected from the data of each round at a ratio of $9 : 1$. More details of data splits can be found in our code.

Baselines. Three state-of-the-art methods, including FOMH (Lu et al. 2019a), OMH-DQ (Lu et al. 2019b), and SAPMH (Zheng et al. 2020), are adopted as baselines. To the best of our knowledge, FOMH is the latest and best online multi-modal hashing model. Both OMH-DQ and SAPMH are the most advanced batch-based multi-modal hashing and their hash functions and hash codes are retrained on all accumulated data (i.e., using $\vec{\mathbf{X}}^{(t)}$ and $\widetilde{\mathbf{X}}^{(t)}$) at each round. For all baselines, codes are publicly available and we set their parameters following provided instructions.

As stated in previous section, no existing multi-modal hashing methods try to handle the class incremental problem with streaming data, including our baselines. As the comparisons on the second and third settings still need to be conducted, we had to release the fourth requirement of the problem setting for baselines. That is, we assumed that baselines had already known all categories before training. It is worth noting that our OASIS always meets the four requirements formalized in the Problem Definition Section.

Implementation Details Zero-mean strategy is widely used in hashing domain and has been proven effective (Wang et al. 2018a; Yao et al. 2019; Wang et al. 2020). Therefore, we also introduced such technique to pre-process the data. However, as OASIS is an online hashing, the used zero-mean operation should adapt for streaming data chunks: $\vec{\mathbf{X}}_m^{(t)} - \frac{\widetilde{\mathbf{u}}_m^{(t)} N^{(t)} + \vec{\mathbf{u}}_m^{(t)} n^{(t)}}{N^{(t)} + n^{(t)}} \mathbf{1}^T$, where $\widetilde{\mathbf{u}}_m^{(t)}$ and $\vec{\mathbf{u}}_m^{(t)}$ represent the feature mean of old data and new data of the m-th modality, respectively. It is worth noting that the calculation result of $(\widetilde{\mathbf{u}}_m^{(t)} N^{(t)} + \vec{\mathbf{u}}_m^{(t)} n^{(t)})/(N^{(t)} + n^{(t)})$ at current round can be used as the $\widetilde{\mathbf{u}}_m^{(t+1)}$ at the next round without recalculation.

For OASIS, the parameter settings are: $\alpha = 100$, $\beta = 0.01$, $\gamma = 10$, $\theta = 0.1$, and $\delta = 1$. We set iteration number as 3. Our experiments are conducted on a Linux workstation with Intel XEON E5-2650 2.20GHz CPU, 128GB RAM.

Results and Comprehensive Analysis

MAP Results. The MAP values under all experimental settings on two datasets are shown in Table 1. We trained all methods with all data chunks and the test set mentioned

Table 2: Comparison of training time (seconds) with 16-bit hash code on MIRFlickr.

Method	chunk 1	chunk 2	chunk 3	chunk 4	chunk 5	chunk 6	chunk 7	chunk 8	chunk 9
FOMH (Lu et al. 2019a)	1.41	1.36	1.59	1.71	1.04	1.65	1.65	1.49	1.38
OMH-DQ (Lu et al. 2019b)	5.84	5.96	6.64	7.30	8.93	8.93	10.11	11.55	13.14
SAPMH (Zheng et al. 2020)	13.32	18.01	23.88	28.66	33.29	42.05	45.09	49.27	63.99
OASIS	2.93	3.25	3.10	2.88	2.68	2.68	2.87	2.46	2.87

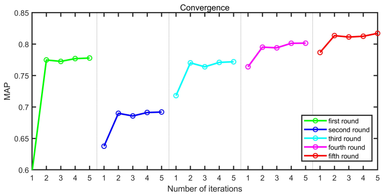

Figure 4: Convergence analysis on MIRFlickr.

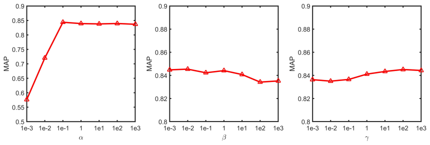

Figure 5: Parameter sensitivity of OASIS on MIRFlickr.

above is used to retrieve similar samples form database which is consists of all training data that has already arrived. It can be observed that our method has an extraordinary performance in all cases, demonstrating the effectiveness of OASIS. Especially, under the settings where new labels continually come, OASIS still works well and the performance gains of our methods are significant, validating the motivations and corroborating the effectiveness of our model. Furthermore, MAP results versus every round under the first setting are displayed in Figure 2 and Figure 3. From these figures, the similar observation can be found that our method exceeds the state-of-the-art baselines including batch-based multi-modal hashing and online multi-modal hashing methods.

Training Time Analysis. The training time consumptions of all methods with 16-bit hash code on MIRFlickr are shown in Table 2. Here, the first setting is adopted. Since there are only 15 training samples at the 10-th round, we only listed the training time of the first 9 rounds. As OMH-DQ and SAPMH are batch-based models, their hash functions and hash codes are retrained on all accumulated data at each round. Thus, we can find that their training time continuously increases. This phenomenon reflects that batch-based methods fail to efficiently handle streaming data and may be impractical when used in large-scale applications. As FOMH and OASIS are online models, we can find that their training time does not increase with data coming and is only linearly related to the size of new data chunk. Although the time consumption of FOMH is slightly lower, considering the excellent performance of our model, OASIS is more practical and it is worthwhile to get much better accuracy at the cost of a little more time.

Convergence Analysis. The results of the convergence experiments under the first setting are shown in Figure 4. In this figure, we reported the MAP results versus number of iterations at the first five rounds on MIRFlickr. It can be seen that our model can quickly achieve satisfactory performance after two iterations.

Parameters Sensitivity Analysis. The experimental results versus different parameter values are shown in Figure 5. We only conducted experiments on α, β, and γ because θ and δ balance the regularization terms and can be empirically set. In addition, when performing sensitivity experiments on one parameter, we fixed other parameters. It can be seen from the experimental results that when α is small, it has a massive impact on the experimental results so that the semantic pairwise similarity plays a vital role in our model. In addition, we can also see that the model can get better results when β is smaller and γ is larger. Although OASIS has five parameters in all, two of them can be empirically set and two of them are not sensitive. In other words, our method could be easily applied in practical scenarios because its parameters can be readily tuned and selected. We finally set $\alpha = 100$, $\beta = 0.01$, $\gamma = 10$, $\theta = 0.1$, and $\delta = 1$.

Conclusion

In this paper, we propose a novel hashing method for multi-modal online retrieval, termed Online enhAnced SemantIc haShing (OASIS). OASIS invents a semantic-enhanced representation to describe instances and designs a new objective function to fully learn from the rich semantic information. Besides, with the help of the semantic-enhanced representation, OASIS can handle new classes coming with streaming data well. Sufficient experiments have demonstrated that the performance of our model can surpass the start-of-the-art models. In addition to the technical contribution of OASIS, this paper tries to give explicit task definition and hopes to benefit the hashing community with our efforts.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61872428, 62172256, in part by Shandong Provincial Key Research and Development Program under Grant 2019JZZY010127, in part by Natural Science Foundation of Shandong Province under Grant ZR2019ZD06, ZR2020QF036, in part by the Fundamental Research Funds of Shandong University under Grant 2019GN075, and in part by the Major Program of the National Natural Science Foundation of China under Grant 61991411.

References

Cakir, F.; Bargal, S. A.; and Sclaroff, S. 2017. Online Supervised Hashing. *Computer Vision and Image Understanding*, 156: 162–173.

Cakir, F.; He, K.; Adel Bargal, S.; and Sclaroff, S. 2017. Mihash: Online Hashing with Mutual Information. In *Proceedings of the IEEE International Conference on Computer Vision*, 437–445.

Cakir, F.; and Sclaroff, S. 2015. Adaptive Hashing for Fast Similarity Search. In *Proceedings of the IEEE International Conference on Computer Vision*, 1044–1052.

Chen, T.-Y.; Zhang, L.; Zhang, S.-c.; Li, Z.-l.; and Huang, B.-c. 2019. Extensible Cross-Modal Hashing. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2109–2115.

Chen, X.; King, I.; and Lyu, M. R. 2017. FROSH: FasteR Online Sketching Hashing. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Chen, X.; Yang, H.; Zhao, S.; Lyu, M. R.; and King, I. 2021a. Making Online Sketching Hashing Even Faster. *IEEE Transactions on Knowledge and Data Engineering*, 33(3): 1089–1101.

Chen, Y.; Hou, Y.; Leng, S.; Zhang, Q.; Lin, Z.; and Zhang, D. 2021b. Long-Tail Hashing. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1328–1338.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. NUS-WIDE: A Real-World Web Image Database from National University of Singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, 1–9.

Huang, L.-K.; Yang, Q.; and Zheng, W.-S. 2013. Online Hashing. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1422–1428.

Huiskes, M. J.; and Lew, M. S. 2008. The MIR Flickr Retrieval Evaluation. In *Proceedings of the ACM International Conference on Multimedia Information Retrieval*, 39–43.

Jiang, Q.-Y.; and Li, W.-J. 2019. Discrete Latent Factor Model for Cross-Modal Hashing. *IEEE Transactions on Image Processing*, 28(7): 3490–3501.

Jin, L.; Li, Z.; and Tang, J. 2020. Deep Semantic Multimodal Hashing Network for Scalable Image-Text and Video-Text Retrievals. *IEEE Transactions on Neural Networks and Learning Systems*.

Kang, W.-C.; Li, W.-J.; and Zhou, Z.-H. 2016. Column Sampling Based Discrete Supervised Hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1230–1236.

Lin, M.; Ji, R.; Liu, H.; Sun, X.; Chen, S.; and Tian, Q. 2020. Hadamard Matrix Guided Online Hashing. *International Journal of Computer Vision*, 128(8): 2279–2306.

Lin, M.; Ji, R.; Liu, H.; and Wu, Y. 2018. Supervised Online Hashing via Hadamard Codebook Learning. In *Proceedings of the ACM International Conference on Multimedia*, 1635–1643.

Liu, W.; Mu, C.; Kumar, S.; and Chang, S.-F. 2014. Discrete Graph Hashing. In *Proceedings of the Advances in Neural Information Processing Systems*, 3419–3427.

Liu, X.; He, J.; Liu, D.; and Lang, B. 2012. Compact Kernel Hashing with Multiple Features. In *Proceedings of the ACM International Conference on Multimedia*, 881–884.

Liu, X.; Yu, G.; Domeniconi, C.; Wang, J.; Ren, Y.; and Guo, M. 2019. Ranking-Based Deep Cross-Modal Hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4400–4407.

Lu, X.; Zhu, L.; Cheng, Z.; Li, J.; Nie, X.; and Zhang, H. 2019a. Flexible Online Multi-Modal Hashing for Large-Scale Multimedia Retrieval. In *Proceedings of the ACM International Conference on Multimedia*, 1129–1137.

Lu, X.; Zhu, L.; Cheng, Z.; Nie, L.; and Zhang, H. 2019b. Online Multi-Modal Hashing with Dynamic Query-Adaption. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 715–724.

Mandal, D.; Annadani, Y.; and Biswas, S. 2018. Growbit: Incremental Hashing for Cross-modal Retrieval. In *Proceedings of the Asian Conference on Computer Vision*, 305–321.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations Workshop Track*.

Nie, X.; Liu, X.; Xi, X.; Li, C.; and Yin, Y. 2020. Fast Unmediated Hashing for Cross-Modal Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*.

Qi, M.; Wang, Y.; and Li, A. 2017. Online Cross-Modal Scene Retrieval by Binary Representation and Semantic Graph. In *Proceedings of the ACM International Conference on Multimedia*, 744–752.

Shen, X.; Shen, F.; Sun, Q.-S.; and Yuan, Y.-H. 2015. Multi-View Latent Hashing for Efficient Multimedia Search. In *Proceedings of the ACM International Conference on Multimedia*, 831–834.

Tian, X.; Ng, W.; Wang, H.; and Kwong, S. 2020. Complementary Incremental Hashing with Query-Adaptive Re-Ranking for Image Retrieval. *IEEE Transactions on Multimedia*.

Tian, X.; Ng, W. W.; and Wang, H. 2019. Concept Preserving Hashing for Semantic Image Retrieval With Concept Drift. *IEEE Transactions on Cybernetics*.

Wang, D.; Gao, X.; Wang, X.; and He, L. 2018a. Label Consistent Matrix Factorization Hashing for Large-Scale Cross-Modal Similarity Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10): 2466–2479.

Wang, D.; Huang, H.; Lu, C.; Feng, B.-S.; Wen, G.; Nie, L.; and Mao, X.-L. 2018b. Supervised Deep Hashing for Hierarchical Labeled Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7388–7395.

Wang, D.; Wang, Q.; An, Y.; Gao, X.; and Tian, Y. 2020. Online Collective Matrix Factorization Hashing for Large-Scale Cross-Media Retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1409–1418.

Wang, G.; Li, C.; Wang, W.; Zhang, Y.; Shen, D.; Zhang, X.; Henao, R.; and Carin, L. 2018c. Joint Embedding of Words and Labels for Text Classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2321–2331.

Wang, Y.; Luo, X.; and Xu, X.-S. 2020. Label Embedding Online Hashing for Cross-Modal Retrieval. In *Proceedings of the ACM International Conference on Multimedia*, 871–879.

Wang, Z.; Zhang, Z.; Luo, Y.; and Huang, Z. 2019. Deep Collaborative Discrete Hashing with Semantic-Invariant Structure. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 905–908.

Weng, Z.; and Zhu, Y. 2020. Online Hashing with Efficient Updating of Binary Codes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12354–12361.

Weng, Z.; and Zhu, Y. 2021. Online Hashing With Bit Selection for Image Retrieval. *IEEE Transactions on Multimedia*, 23: 1868–1881.

Wu, D.; Dai, Q.; Liu, J.; Li, B.; and Wang, W. 2019. Deep Incremental Hashing Network for Efficient Image Retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9069–9077.

Xie, D.; Deng, C.; Li, C.; Liu, X.; and Tao, D. 2020a. Multi-Task Consistency-Preserving Adversarial Hashing for Cross-Modal Retrieval. *IEEE Transactions on Image Processing*, 29: 3626–3637.

Xie, L.; Shen, J.; Han, J.; Zhu, L.; and Shao, L. 2017. Dynamic Multi-View Hashing for Online Image Retrieval. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3133–3139.

Xie, L.; Shen, J.; and Zhu, L. 2016. Online Cross-Modal Hashing for Web Image Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 294–300.

Xie, Y.; Liu, Y.; Wang, Y.; Gao, L.; Wang, P.; and Zhou, K. 2020b. Label-Attended Hashing for Multi-Label Image Retrieval. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 955–962.

Xu, J.; Xu, Z.; Walker, P.; and Wang, F. 2020. Federated Patient Hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6486–6493.

Yao, T.; Wang, G.; Yan, L.; Kong, X.; Su, Q.; Zhang, C.; and Tian, Q. 2019. Online Latent Semantic Hashing for Cross-Media Retrieval. *Pattern Recognition*, 89: 1–11.

Yi, J.; Liu, X.; Cheung, Y.-m.; Xu, X.; Fan, W.; and He, Y. 2021. Efficient Online Label Consistent Hashing for Large-Scale Cross-Modal Retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 1–6.

Zheng, C.; Zhu, L.; Cheng, Z.; Li, J.; and Liu, A. 2020. Adaptive Partial Multi-view Hashing for Efficient Social Image Retrieval. *IEEE Transactions on Multimedia*.

Zhu, L.; Lu, X.; Cheng, Z.; Li, J.; and Zhang, H. 2020a. Deep Collaborative Multi-View Hashing for Large-Scale Image Search. *IEEE Transactions on Image Processing*, 29: 4643–4655.

Zhu, L.; Lu, X.; Cheng, Z.; Li, J.; and Zhang, H. 2020b. Flexible Multi-Modal Hashing for Scalable Multimedia Retrieval. *ACM Transactions on Intelligent Systems and Technology*, 11(2): 1–20.