This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Joint Versus Independent Multiview Hashing for Cross-View Retrieval

Peng Hu, Xi Peng, *Member, IEEE*, Hongyuan Zhu, *Member, IEEE*, Jie Lin, *Member, IEEE*, Liangli Zhen, and Dezhong Peng

***Abstract*—Thanks to the low storage cost and high query speed, cross-view hashing (CVH) has been successfully used for similarity search in multimedia retrieval. However, most existing CVH methods use all views to learn a common Hamming space, thus making it difficult to handle the data with increasing views or a large number of views. To overcome these difficulties, we propose a decoupled CVH network (DCHN) approach which consists of a semantic hashing autoencoder module (SHAM) and multiple multiview hashing networks (MHNs). To be specific, SHAM adopts a hashing encoder and decoder to learn a discriminative Hamming space using either a few labels or the number of classes, that is, the so-called flexible inputs. After that, MHN independently projects all samples into the discriminative Hamming space that is treated as an alternative ground truth. In brief, the Hamming space is learned from the semantic space induced from the flexible inputs, which is further used to guide view-specific hashing in an independent fashion. Thanks to such an independent/decoupled paradigm, our method could enjoy high computational efficiency and the capacity of handling the increasing number of views by only using a few labels or the number of classes. For a newly coming view, we only need to add a view-specific network into our model and avoid retraining the entire model using the new and previous views. Extensive experiments are carried out on five widely used multiview databases compared with 15 state-of-the-art approaches. The results show that the proposed independent hashing paradigm is superior to the common joint ones while enjoying high efficiency and the capacity of handling newly coming views.**

***Index Terms*—Common hamming space, cross-view retrieval, decoupled cross-view hashing network (DCHN), multiview hashing, multiview representation learning.**

Manuscript received March 3, 2020; revised July 14, 2020; accepted September 17, 2020. This work was supported in part by the National Key Research and Development Project of China under Contract 2017YFB1002201; in part by the National Natural Science Foundation of China under Grant 61625204, Grant 61836006, Grant 61971296, Grant 61806135, Grant U1831121, Grant U19A2078, and Grant U19A2081; in part by the Sichuan Science and Technology Planning Projects under Grant 2019YFG0495, Grant 2020YFG0319, and Grant 2020YFH0186; in part by the Fundamental Research Funds for the Central Universities under Grant YJ201949; and in part by the Agency for Science, Technology and Research under its AME Programmatic Funds under Project A1892b0026. This article was recommended by Associate Editor P. De Meo. *(Corresponding author: Dezhong Peng.)*

Peng Hu is with the College of Computer Science, Sichuan University, Chengdu 610065, China, and also with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore.

Xi Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China.

Hongyuan Zhu and Jie Lin are with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore.

Liangli Zhen is with the Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore.

Dezhong Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China, also with the Frontier Academic Center, Shenzhen Peng Cheng Laboratory, Shenzhen 518052, China, and also with the College of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: pengdz@scu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.3027614

I. Introduction

With the rapid growth of multiview data, such as image, text, and video on the Internet, there are increasing demands on developing cross-view methods for a variety of applications [1]–[6]. Among them, cross-view retrieval has arisen great interest from the community, which aims to retrieve the interested content across different views/modalities, for example, retrieving the corresponding text counterpart for a given image query. Due to the low storage cost and high query speed of hash codes [7], [8], cross-view hashing (CVH) has achieved promising performance and is becoming increasingly popular for the large-scale multimedia retrieval. Although CVH has been paid more attention to by both academia and industry [9], [10], there still remain many challenges. Especially, different views may lie in completely disparate spaces with large semantic gaps, thus resulting in inferior retrieval performance.

To eliminate the semantic gap, numerous CVH methods have been proposed to project multiview data into a common Hamming space by narrowing the heterogeneous gap. In general, most existing multiview hashing approaches could be roughly classified into two categories, that is: 1) shallow [11]–[13] and 2) deep methods [9], [10], [14]. The shallow approaches usually learn some single-layer linear or nonlinear transformations to project multiview data into a shared Hamming space [15], [16]. One major limitation of linear methods is that they may be incapable of capturing the high-level nonlinear semantics of real-world data. To address this limitation, some kernel methods [12], [17] have been proposed. However, it is still an open issue and a daunting task to choose a suitable kernel function [18]. To adaptively capture the nonlinearity in data, several recent works attempted to use the deep neural network (DNN) to learn a common hash space across different views in an unsupervised [19] or supervised [9], [10], [20] way.

To be specific, although the aforementioned CVH methods have achieved promising performance, they need all views to jointly learn the common Hamming space as shown in Fig. 1(a), thus facing the following two disadvantages.

2168-2267 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

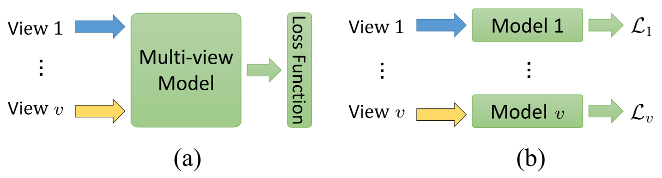


Fig. 1. Difference between (a) existing joint CVH and (b) our decoupled hashing paradigm. In brief, most existing methods including one-step and two-step approaches use all views to learn a common representation or graph. Therefore, it is difficult to deal with a large number of views and increasing views since their loss is optimized involving all views. In contrast, we independently train each view-specific network with its own independent objective function $\mathcal{L}_k$, thus overcoming the above limitations suffered from the common paradigms.

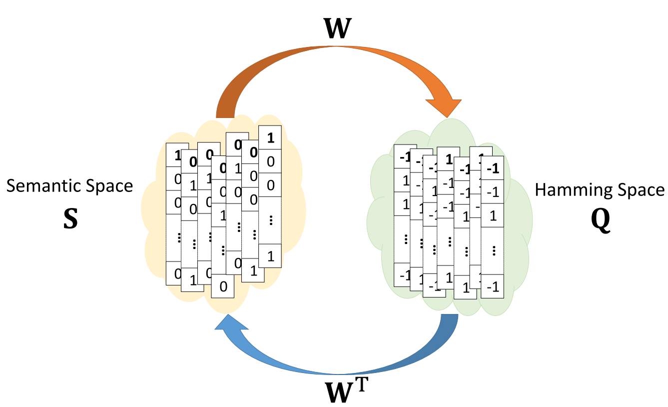


Fig. 2. Proposed SHAM utilizes the semantic information (e.g., labels or classes) to learn an encoder $\mathbf{W}$ and a decoder $\mathbf{W}^T$ by mutually converting the semantic and Hamming spaces. SHAM is one key component of our independent hashing paradigm.

1) A large number of computational resources are required to handle a large number of views. In other words, when the view number is large, the time cost and memory will rapidly increase due to the joint learning paradigm. Note that the joint learning here means that all views rather than the hashing processes are jointed. In other words, almost all existing CVH approaches, including one-step [9], [14], [20] and two-step methods [21]–[23], are joint learning methods that require using all views to learn a common representation or similarity graph.
2) The entire model has to be retrained for the newly coming views, namely, almost all existing works ignore how to handle increasing views in an efficient and flexible manner.

To tackle the above two disadvantages, we propose a novel multiview hashing method, called decoupled CVH networks (DCHN), which consists of a SHAM and multiple multiview hashing networks (MHNs). In brief, SHAM encodes the flexible input into the discriminative Hamming space in which the corresponding hash code is further decoded to reconstruct the input as shown in Fig. 2. After that, the learned Hamming space is treated as an alternative ground truth, which guides the optimization of MHN (see Fig. 1). The major difference from the existing hashing paradigm is that we do not jointly learn a common Hamming space from all views, dubbed *hashing from all views*. Instead, we obtain a hash space from the flexible input (i.e., the labels or the number of classes) and use it as the ground truth, dubbed *hashing from labels*. In other words, almost all existing works take a feedforward pipeline to learn a common hash code from all views, whereas our method takes a reversed pipeline that obtains the hash code by learning our SHAM from the flexible inputs and then further using it to train the view-specific MHNs.

The advantages of our DCHN are two-fold. First, thanks to our "hashing from labels" paradigm, all view-specific MHNs and the corresponding loss functions are independent of each other. Therefore, these networks can be independently trained regardless of when and where they should be trained, that is, our method breaks the temporal and spatial connections of view-specific networks. For a new coming view, existing CVH methods have to retrain their models using all views (old and new views) due to their joint hashing paradigms, whereas our method will only establish and optimize a new MHN, thanks to our independent hashing paradigm. Second, our DCHN even allows being trained in a sequential manner, which is helpful to the scenario with limited resources, that is, we would perform training and inference view-by-view. These advantages are brought by the designed independent training strategy regardless of any specific form of the deep networks. As a result, our method could run on the low resource devices (e.g., mobile devices) even though a large number of views and newly coming views are available.

The main contributions of our work could be summarized as follows.

1) To the best of our knowledge, the proposed SHAM could be the first multiview hashing method that learns from a few labels or only the number of categories, that is, the so-called flexible inputs. Such an advantage significantly makes our method highly attractive in practice. For example, limited by privacy and copyright, multiple owners of different views cannot share their data with each other. In such a case, it is highly expected to develop a method like our DCHN, which could first use some insensitive data (e.g., the class number) to learn the Hamming space and then use the Hamming space to separately learn the hash code and perform retrieval.
2) Thanks to the proposed paradigm of "hashing from labels," a novel deep multiview hashing method (DCHN) is proposed, which could independently learn the hash code for different views, thus embracing the capacity of handling increasing views and large-scale views. To the best of our knowledge, such a hashing method with the aforementioned capacities has been less touched in the previous studies.

II. Related Work

Multiview learning has been paid more and more attention from academic and industry communities for multimedia retrieval [24]–[26]; multiview clustering [27], [28]; disease analysis [29]; etc. Furthermore, it is very interesting that multisource learning could benefit from multiview learning [30], which will extend the application scope of multiview learning. Hashing has been widely used in many applications due to its great advantages, for example, low storage cost and high

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

query speed, including multiview hashing [22], [25]; image retrieval [31]–[33]; etc. In this work, we only focus on the multiview hashing learning that could be roughly classified into shallow and deep models.

The key to these methods is to learn a common space shared by different views. To be specific, traditional CVH methods [12], [21], [22] learn two transformations to project the cross-view data into a common Hamming space shared across different views. For example, Lin *et al.* [21] proposed a supervised CVH method, called semantics-preserving hashing (SePH), using the semantic affinity of different views. Moreover, Lin *et al.* extended SePH by learning the predictive models (e.g., linear ridge regression, logistic regression, or kernel logistic regression) as the hashing functions in each view to project the corresponding view into the common Hamming space [12], such as SePH with logistic regression (SePH_{lr}). Li *et al.* [22] presented a supervised linear subspace ranking hashing framework (LSRH) to project two views into a shared Hamming space, which employs the Hamming distance to measure the similarity between different views. Ding *et al.* [34] developed a so-called rank-order preserving hashing method (RoPH) for a cross-view similarity search by introducing a novel regression-based rank-order preserving loss.

Recently, DNN has been successfully applied to numerous multiview problems for learning a common space [9], [10], [35]–[37]. These methods adopt different technologies to learn a common Hamming space and achieve promising results, for example, pairwise constraints [35]; deep quantization [36]; adversarial learning [20], [37]; etc. Specifically, Jiang and Li [9] proposed deep cross-modal hashing (DCMH) by integrating feature learning and hash-code learning into a unified deep framework. Li *et al.* [20] proposed a self-supervised adversarial hashing method (SSAH), which aims to utilize the ability of the adversarial learning to model the multiview data distribution. Deng *et al.* [10] proposed a triplet-based deep hashing method (TDH) for the large-scale cross-view retrieval by using a deep CNN to perform feature learning and hashing in an end-to-end manner. Hu *et al.* [38] utilized the variational inference and the similarity relationship of samples to project the samples from different views into a single shared Hamming space.

Although some works [21], [39] and our DCHN are two-step approaches, they are remarkably different in given aspects. First, the previous two-step methods need the labels of all views to learn hash codes in the first step, whereas our method could only use a few labels or the class number to learn the semantic hash encoder and decoder. Second, all views, even for new views, could be separately trained for our DCHN instead of joint training of the existing methods as shown in Fig. 1. Thanks to the above differences, our method could enjoy more efficient and scalable training, as well as handling increasing view numbers, whereas [21] and [39] might be incapable to these cases. Furthermore, although some one-step methods (e.g., SSAH [20]) could learn from labels like our DCHN, they should jointly use the label information and views to learn the hash codes as shown in Fig. 1(a). In contrast, our method could only use the class number or a few available labels to learn the semantic hash encoder and decoder. Therefore, different from the pioneer works, including SSAH [20], the hashing learning process of our method is decoupled rather than joint learning, thus being capable of handling large-scale and increasing views. In conclusion, different from the aforementioned traditional and deep cross-view methods, our method does not utilize all views to jointly learn a common Hamming representation. Instead, we utilize very little input information (a few labels or only the number of classes) to learn neural networks for our SHAM and then use the label-induced Hamming space to optimize the view-specific MHN. As a result, our method is more efficient and effective because it could handle an increasing number of views while using a few computational resources.

III. Proposed Method

As shown in Fig. 3, our DCHN employs a SHAM and v MHNs to learn the unified hash codes for all views.

A. Problem Formulation

For ease of presentation, some definitions are given below which will be used in the remaining sections. Let the kth view be denoted by $\mathcal{X}^k = \{\mathbf{x}_i^k\}_{i=1}^{N_k}$, where $\mathbf{x}_i^k$ is the ith sample of the kth view, and N_k is the number of the samples from the kth view. Besides, let $\mathcal{Y}^k = \{\mathbf{y}_i^k\}_{i=1}^{N_k}$ be the label set, where $\mathbf{y}_i^k \in \mathbb{R}^{c\times 1}$ is a binary-value label vector for the sample $\mathbf{x}_i^k$ and c is the number of categories. If the ith sample of the kth view belongs to the jth class, $y_{ij}^k = 1$; otherwise, $y_{ij}^k = 0$. For the single-label data, their semantic labels only contain one nonzero value. In contrast, the labels will be with multiple nonzero elements for multilabel data. In the experiments, we will show the effectiveness of our method to these two cases. However, in the following, we will not specifically discuss this issue for clarity.

Multiview hashing aims to learn a common hash space $\mathcal{B} = \{\mathbf{B}^k\}_{k=1}^v$ shared by multiple views, where v is the number of views, $\mathbf{B}^k = [\mathbf{b}_1^k, \ldots, \mathbf{b}_i^k, \ldots, \mathbf{b}_{N_k}^k]$ is the discrete code matrix of the kth view, $\mathbf{b}_i^k \in \{-1, 1\}^L$ is the binary code of $\mathbf{x}_i^k$, and L is the length of the hash code. In the Hamming space, the similarity between different points is measured by the Hamming distance. As the Hamming distance $H(\mathbf{b}_i^k, \mathbf{b}_j^l)$ and the inner product $\langle \mathbf{b}_i^k, \mathbf{b}_j^l \rangle$ are related by $H(\mathbf{b}_i^k, \mathbf{b}_j^l) = (1/2)(L - \langle \mathbf{b}_i^k, \mathbf{b}_j^l \rangle)$, we therefore use the inner product $(1/2)\langle \mathbf{b}_i^k, \mathbf{b}_j^l \rangle$ to measure the similarity Γ_{ij}^k of two binary codes ($\mathbf{b}_i^k$ and $\mathbf{b}_j^l$), that is, $\Gamma_{ij}^k = (1/2)\langle \mathbf{b}_i^k, \mathbf{b}_j^l \rangle$. With the above notations, our MHN aims to learn v view-specific hashing functions $f_k(\cdot)|_{k=1}^v$ to project the corresponding view into the Hamming space obtained by our SHAM.

B. Framework

As shown in Fig. 3, the proposed DCHN adopts our SHAM (see Fig. 2) consisting of the encoder $\mathbf{W}$ and the decoder $\mathbf{W}^T$ and v MHNs to learn the unified hash codes for all views. In this section, we elaborate on the implementation details of SHAM and the network architectures of MHN.

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

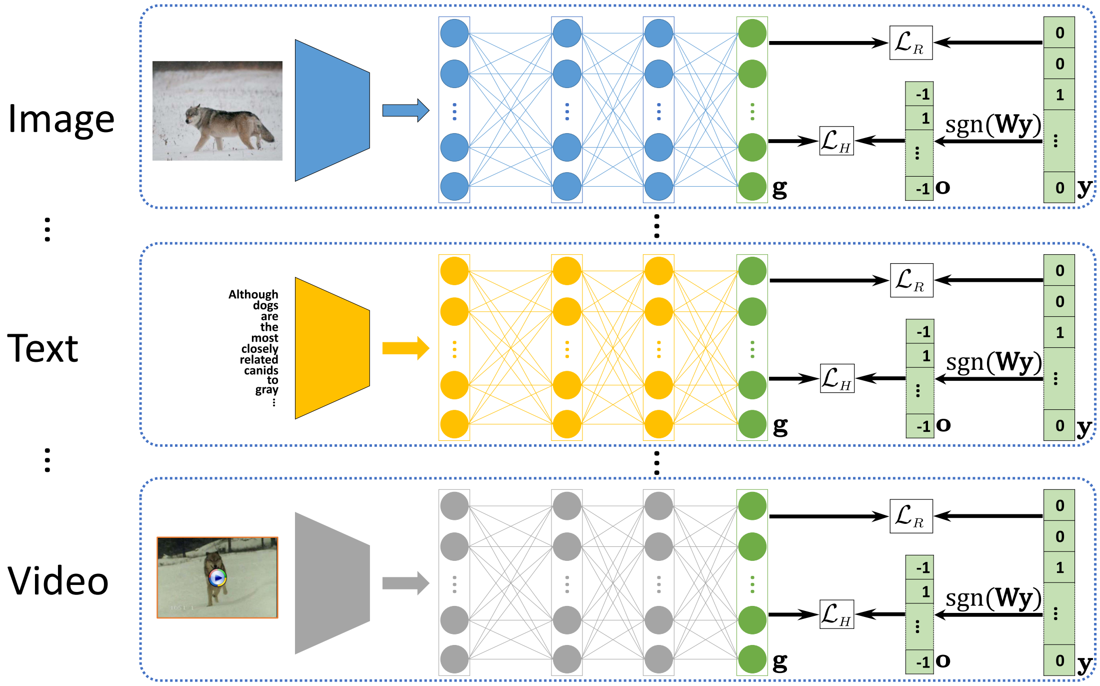


Fig. 3. Framework of the proposed DCHN method. **g** is the output of the corresponding view (i.e., image, text, video, etc.). **o** is the semantic hash code that is computed by the corresponding label **y** and semantic hashing transformation **W**. **W** is computed by the proposed semantic hashing autoencoder module (SHAM). sgn is an elementwise sign function. $\mathcal{L}_R$ and $\mathcal{L}_H$ are hash reconstruction [see (12)] and semantic hashing [see (11)] functions, respectively. In the training stage, first, **W** is used to recast the label **y** as a ground-truth hash code **o**. Then, the obtained hash code is used to guide view-specific networks with a semantic hashing reconstruction regularizer. Such a learning scheme makes the v view-specific neural networks (one network for each view) can be trained separately since they are decoupled and do not share any trainable parameters. Therefore, our DCHN can be easy to scale to a large number of views. In the inference stage, each trained view-specific network $f_k(\mathbf{x}^k, \Theta_k)$ is used to compute the hash code of the sample $\mathbf{x}^k$.

1) Semantic Hashing Autoencoder Module: In this section, we first introduce the formulation of our SHAM and then present the details of the optimization procedure. As shown in Fig. 2, SHAM is a traditional autoencoder that is with only one hidden layer shared by the encoder and decoder. The encoder aims to project the flexible input into a discriminative Hamming space in which the obtained hash codes are further used to reconstruct the input. Formally

$$\arg\min_{\mathbf{W},\mathbf{V},\mathbf{Q}} \|\mathbf{S}-\mathbf{VWS}\|_F^2 \quad \text{s.t. } \mathbf{WS}=\mathbf{Q},\quad \mathbf{Q}\in\{-1,1\}^{L\times m} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm, **W** and **V** are the hidden layer transformations of the encoder and decoder, **Q** is the semantic binary matrix, m is the number of objects in the semantic space, and $\mathbf{S}\in\mathbb{R}^{c\times m}$ is the semantic input matrix induced by the flexible inputs. In brief, when m labels are available, denoted as DCHN$_m$, **S** is a label assignment matrix constructed by the corresponding m label vectors, that is, $\mathbf{S}\in\mathbb{R}^{c\times m}$. However, there are not any available labels to construct **S** when only the class number c is available, denoted as DCHN$_0$. Motivated by the widely used one-hot label encoding [40], which could maximize the difference between distinct classes, **S** could be constructed according to, for example, a label-indicator matrix of which each column is a different one-hot class vector when only the class number c is available, that is, $\mathbf{S}\in\mathbb{R}^{c\times c}$.

To further simplify our SHAM, we adopt the tied weights via $\mathbf{V}=\mathbf{W}^T$ [41], [42]. Then, (1) could be rewritten as follows:

$$\arg\min_{\mathbf{W},\mathbf{Q}} \left\|\mathbf{S}-\mathbf{W}^T\mathbf{Q}\right\|_F^2 \quad \text{s.t. } \mathbf{WS}=\mathbf{Q},\quad \mathbf{Q}\in\{-1,1\}^{L\times m}. \tag{2}$$

It is difficult to solve an objective function with the hard constraint such as $\mathbf{WS}=\mathbf{Q}$ [42]. Therefore, to optimize the objective in (2), we relax the constraint into a soft constraint and rewrite the objective as

$$\arg\min_{\mathbf{W},\mathbf{Q}} \left\|\mathbf{S}-\mathbf{W}^T\mathbf{Q}\right\|_F^2+\lambda\|\mathbf{WS}-\mathbf{Q}\|_F^2 \quad \text{s.t. } \mathbf{Q}\in\{-1,1\}^{L\times m} \tag{3}$$

where $\lambda>0$ is a balance parameter. It is well known that (3) is intractable as **Q** are with binary values. Different from most of the previous CVH methods which relaxes the above discrete problem as a continuous one, our SHAM enforces the discrete constraint $\mathbf{Q}\in\{-1,1\}^{L\times m}$ to directly learn the hash codes **Q**. Equation (3) could be solved using a tractable alternating minimization algorithm as follows.

First, fixing **W**, it gives the derivative of (3) as follows:

$$-\mathbf{W}\left(\mathbf{S}-\mathbf{W}^T\mathbf{Q}\right)+\lambda(\mathbf{Q}-\mathbf{WS})=\mathbf{0} \tag{4}$$

then

$$\mathbf{Q}=\operatorname{sgn}\left((1+\lambda)\left(\mathbf{WW}^T+\lambda\mathbf{I}\right)^{-1}\mathbf{WS}\right) \tag{5}$$

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 1 Learning Algorithm for the Proposed SHAM

Require: The class number c or some available labels, the balance parameter λ, and the length of the hash code L.
Construct the semantic matrix $\mathbf{S}$ according to c or the available labels.
Randomly initialize the matrix $\mathbf{W}$.
repeat
  Compute $\mathbf{Q}$ according to (5).
  Compute $\mathbf{W}$ by solving the Sylvester equation (8).
until convergence
return $\mathbf{W}$.

where $\mathrm{sgn}(\cdot)$ is an elementwise sign function that is defined as

$$\mathrm{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{6}$$

Second, fixing $\mathbf{Q}$, it gives the derivative of (3) as follows:

$$\begin{aligned} \mathbf{Q}\left(\mathbf{S}^T - \mathbf{Q}^T\mathbf{W}\right) + \lambda(\mathbf{Q} - \mathbf{W}\mathbf{S})\mathbf{S}^T = \mathbf{0} \\ \left(\mathbf{Q}\mathbf{Q}^T\right)\mathbf{W} + \mathbf{W}\left(\lambda\mathbf{S}\mathbf{S}^T\right) = (1+\lambda)\mathbf{Q}\mathbf{S}^T. \end{aligned} \tag{7}$$

Let $\mathbf{A} = \mathbf{Q}\mathbf{Q}^T$, $\mathbf{C} = \lambda\mathbf{S}\mathbf{S}^T$, and $\mathbf{D} = (1+\lambda)\mathbf{Q}\mathbf{S}^T$, then

$$\mathbf{A}\mathbf{W} + \mathbf{W}\mathbf{C} = \mathbf{D}. \tag{8}$$

Equation (8) is the well-known Sylvester equation that could be efficiently solved using the Bartels–Stewart algorithm [43]. Therefore, our SHAM can be solved in an alternating manner and derive a closed-form optimal solution to problem (8) with respect to (w.r.t.) $\mathbf{W}$. The detailed alternation solution process has been summarized in Algorithm 1. With the learned $\mathbf{W}$, the semantic hash code corresponding to $\mathbf{y}_i^k$ can be obtained as follows:

$$\mathbf{o}_i^k = \mathrm{sgn}\left(\mathbf{W}\mathbf{y}_i^k\right). \tag{9}$$

2) Multiview Hashing Network: For the kth view, an MHN $f_k(\cdot, \Theta_k)$ is established to obtain the corresponding hash code with the help of SHAM, where Θ_k denotes the parameters of the network. To be exact, for the ith sample $\mathbf{x}_i^k$ from the kth view, we have $\mathbf{g}_i^k = f_k(\mathbf{x}_i^k; \Theta_k) \in \mathbb{R}^L$. $\mathbf{g}_i^k$ is desired to approach the corresponding semantic binary code $\mathbf{o}_i^k$ to hash the representation and discrimination while reconstructing its label $\mathbf{y}_i^k$ to preserve as much discrimination as possible.

First, we define a likelihood function between semantic similarity and hash codes as follows:

$$p\left(S_{ij}|\mathbf{g}_i^k, \mathbf{o}_j^k\right) = \begin{cases} \frac{1}{1+e^{-\Gamma_{ij}^k}} & S_{ij} = 1 \\ \frac{e^{-\Gamma_{ij}^k}}{1+e^{-\Gamma_{ij}^k}} & S_{ij} = 0 \end{cases} \tag{10}$$

where $\Gamma_{ij}^k = (1/2)\langle \mathbf{g}_i^k, \mathbf{o}_j^k \rangle$ is used to measure the similarity of $\mathbf{g}_i^k$ and $\mathbf{o}_j^k$ in the Hamming space, $\mathbf{o}_i^k$ is obtained via (9), and $\mathbf{S}$ is a similarity matrix. $S_{ij} = 1$ if $\mathbf{y}_i^k$ and $\mathbf{y}_j^k$ share at least one class, and $S_{ij} = 0$ otherwise. The probability p between any two instances is proportional to the discriminant criterion

Algorithm 2 Learning Algorithm for the Proposed MHN

Require: The training data set of all views $\{\mathcal{X}^k\}_{k=1}^v$, the corresponding label sets $\{\mathcal{Y}^k\}_{k=1}^v$, the learned semantic hashing transformation $\mathbf{W}$ through SHAM, the length of the hash code L, the balance parameter β, the learning rate α, and the batch size N_b.
parfor $k = 1, 2, \ldots, v$ **do** ▹ parallel for loop
  Randomly initialize the parameters Θ_k of the kth view-specific network.
  while not converge **do**
    Randomly sample N_b points from the kth view $\{\mathcal{X}_k, \mathcal{Y}_k\}$ to construct a view-specific mini-batch.
    For each sampled point $\mathbf{x}_i^k$, calculate $\mathbf{g}_i^k = f_k(\mathbf{x}_i^k; \Theta_k)$ by forward-propagation, and the semantic hash code of the label $\mathbf{y}_i^k$ according to (9).
    Compute the MHN loss for the kth view-specific neural network with (13) in the mini-batch.
    Update Θ_k by minimizing the obtained loss by using back propagation as follows:
    $\Theta_k \leftarrow \Theta_k - \alpha \nabla_{\Theta_k} \mathcal{L}_k$.
  end while
end parfor
return Optimized view-specific hashing models.

in which the similarity of the same class should be large and the similarities of the different classes should be small. Thus, maximizing the likelihood aims to maximize the within-class similarity while minimizing the between-class similarity in the Hamming space. Then, we can use the negative log-likelihood to formulate this hashing objective as follows:

$$\mathcal{L}_H\left(\mathcal{X}^k, \mathcal{Y}^k\right) = \frac{1}{N_k}\sum_{i,j=1}^{N_k}\left(\ln\left(1+e^{\Gamma_{ij}}\right) - S_{ij}\Gamma_{ij}\right) \tag{11}$$

where $\ln(\cdot)$ is the natural logarithm operator. The objective function is used to maximize the similarity between the same class and minimize the similarity between the distinct classes. Obviously, minimizing (11) can make the learned representation approach the binary codes without relaxation while preserving the discrimination into the learned hash representations. To further push as much discrimination as possible into the Hamming space, we restrict the learned representation to reconstruct its label with the learned hash decoder $\mathbf{W}^T$ as follows:

$$\mathcal{L}_R\left(\mathcal{X}^k, \mathcal{Y}^k\right) = \frac{1}{N_k}\sum_{i=1}^{N_k}\left\|\mathbf{W}^T\mathbf{g}_i^k - \mathbf{y}_i^k\right\|_2^2 \tag{12}$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm. Hence, the loss function of MHN can be formulated as follows:

$$\mathcal{L}_k = \beta\mathcal{L}_H\left(\mathcal{X}^k, \mathcal{Y}^k\right) + (1-\beta)\mathcal{L}_R\left(\mathcal{X}^k, \mathcal{Y}^k\right) \tag{13}$$

where β is a positive balance parameter.

Then, the parameters Θ_k^* of the kth MHN are optimized by minimizing (13) as follows:

$$\Theta_k^* = \arg\min_{\Theta_\mathbf{k}} \mathcal{L}_k. \tag{14}$$

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
GENERAL STATISTICS OF PKU XMEDIA, NUS-WIDE, AND MS-COCO, WHERE "CLASS" STANDS FOR THE NUMBER OF THE CLASSES, AND "DATABASE," "TRAIN," AND "QUERY" ARE THE NUMBERS OF THE RETRIEVAL DATABASE, TRAINING, AND QUERY SETS, RESPECTIVELY

Dataset	Class	View	Database	Train	Query	Feature
PKU XMedia	20	Image	4,000	4,000	1,000	4,096D VGG
		Text	4,000	4,000	1,000	3,000D BoW
		Audio clip	800	800	200	29D MFCC
		3D model	400	400	100	4,700D LightField
		Video	969	969	174	4,096D C3D
MIRFLICKR-25K	24	Image	20,015	10,000	2,000	4,096D VGG
		Text	20,015	10,000	2,000	1,386D BoW
IAPR TC-12	255	Image	20,000	10,000	2,000	4,096D VGG
		Text	20,000	10,000	2,000	2,912D BoW
NUS-WIDE	21	Image	188,321	10,500	2,100	4,096D VGG
		Text	188,321	10,500	2,100	1,000D BoW
MS-COCO	80	Image	117,218	10,000	5,000	4,096D VGG
		Text	117,218	10,000	5,000	300D Doc2Vec

This objective function allows training DCHN using SGD and its variants in an end-to-end manner. The parallel optimization process is summarized in Algorithm 2. With the learned Θ_k^*, the hash code of $\mathbf{x}_i^k$ can be obtained as follows:

$$\mathbf{b}_i^k = \operatorname{sgn}\Big(f_k\Big(\mathbf{x}_i^k; \Theta_k^*\Big)\Big) \in \{-1, 1\}^L. \tag{15}$$

IV. EXPERIMENTAL STUDY

To evaluate the proposed method, we compare our DCHN with 15 state-of-the-art cross-view methods on five datasets in terms of effectiveness and efficiency. The used datasets contain PKU XMedia [44], MIRFLICKR-25K [45], IAPR TC-12 [46], NUS-WIDE [47], and MS-COCO [48]. Moreover, we also conduct an ablation study and parameter sensitivity to investigate the effectiveness of our method.

A. Datasets and Compared Methods

In this section, we briefly introduce the five aforementioned datasets. For a fair comparison, we partition the database into retrieval, training, and query sets by following [20] and [44]. In Table I, we show the general statistics of the five datasets. In our experiments, all tested methods adopt the same features for fair comparisons, that is to say, the pretrained extractors (e.g., VGG, Doc2Vec, etc.) are not fine tuned in the training process. To the best of our knowledge, NUS-WIDE and MS-COCO are the two largest datasets in the community. To obtain valid samples, the instances without any category information are pruned from these two datasets. The sizes of the pruned datasets are shown in Table I. For PKU XMedia, we use the training set as the retrieval database and the test set as the query set since this dataset is small. Note that the training sets are the subsets of the corresponding retrieval databases for all the datasets by the following [9] and [20].

To evaluate the performance of our method on multiple views (the view number is larger than two), we conduct comparisons with five real-value multiview methods and ten binary-value (i.e., hashing) approaches. Specifically, we also conduct comparisons on the PKU XMedia database with five real-value multiview methods (i.e., four shallow methods MCCA [49], GMLDA [50], MvDA [51], and MvDA-VC [51], and one deep method MAN [26]) and seven shallow binary-value hashing methods (i.e., SePH [21], SePH$_{lr}$ [12], RoPH [34], LSRH [22], KDLFH [23], DLFH [23], and MTFH [13]). On the binary-view datasets (*NUS-WIDE and MS-COCO*), we conduct comparisons with ten hashing approaches, that is, the above seven shallow methods and three deep methods (i.e., DJSRH [14], DCMH [9], and SSAH [20]). Note that we fail to conduct these three deep hashing methods on the PKU XMedia database due to two reasons. On one hand, these methods are highly computationally inefficient. On the other hand, they could only handle binary-view data. To handle multiview data, it is necessary to transform a multiview data to multiple binary-view datasets, thus leading to higher computational complexity. Moreover, we also investigate the performance of a variant of our method, called the "Baseline." The only one difference between the Baseline and our DCHN is that the former directly adds a softmax classifier layer [52] on the top of MHN with the cross-entropy loss function to investigate the effectiveness of our SHAM.

B. Implementation Details

The proposed DCHN approach is trained on two NVIDIA GTX 1080Ti in PyTorch. We use the ADAM optimizer [53] to train our approach. The batch size and maximal epoch are set as 64 and 100, respectively. The learning rate α is empirically set as 0.0001 for each view. For all the used datasets, the image features are extracted by a trained 19-layer VGGNet [40] model that has been pretrained on ImageNet. Specifically, the used features are output from the fc7 layer of the VGGNet. For the MS-COCO database, we use a trained Doc2Vec model[1] [54], which has been pretrained on Wikipedia, to extract 300-D features from the sentences. For the other datasets, the text feature of each document is a bag-of-words vector (BoW). For the other views of PKU XMedia (i.e., audio, 3-D, and video), the features are given by the authors. In our view-specific MHNs, three fully connected layers are utilized to learn the common hash codes for all views. Each FC layer follows a ReLU layer except the last layer. The numbers of hidden units of these FC layers are, respectively, 4096, 4096, and L, where L is the length of hash codes. For our DCHN$_m$, m labels are randomly sampled to construct the semantic matrix $\mathbf{S}$ from the image view of the corresponding training set in our experiments. Accordingly, c random orthogonal one-hot vectors are used to construct the semantic matrix $\mathbf{S}$ for the no-label case, that is, DCHN$_0$.

C. Experimental Comparisons

To evaluate the performance of our DCHN, we adopt the mean average precision (MAP) as the evaluation metric. Table II demonstrates the results of our method on the PKU XMedia dataset [44]. Moreover, Table III shows the comparisons on the MIRFLICKR-25K and IAPR TC-12 datasets. Similarly, Table IV shows the comparisons on the NUS-WIDE and MS-COCO datasets. In the experiments, the length of hash codes L is set as 16, 32, 64, and 128 bits for a comprehensive evaluation. To investigate the ability of our method to

[1] https://github.com/jhlau/doc2vec

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
PERFORMANCE COMPARISON IN TERMS OF mAP SCORES ON THE PKU XMEDIA DATASET

Method (Query / Database)	Image				Text				Audio				3D				Video				Avg.
	Text	Audio	3D	Video	Image	Audio	3D	Video	Image	Text	3D	Video	Image	Text	Audio	Video	Image	Text	Audio	3D	
MCCA [49]	0.115	0.145	0.172	0.125	0.120	0.124	0.147	0.115	0.133	0.114	0.176	0.137	0.126	0.095	0.122	0.104	0.090	0.077	0.094	0.104	0.122
GMLDA [50]	0.614	0.159	0.855	0.622	0.625	0.140	0.747	0.504	0.264	0.187	0.276	0.187	0.489	0.422	0.138	0.433	0.372	0.305	0.110	0.443	0.395
MvDA [51]	0.623	0.295	0.879	0.667	0.618	0.242	0.767	0.564	0.287	0.237	0.416	0.270	0.458	0.391	0.228	0.427	0.324	0.266	0.154	0.437	0.427
MvDA-VC [51]	0.655	0.221	0.877	0.707	0.642	0.185	0.762	0.598	0.243	0.207	0.371	0.229	0.530	0.456	0.194	0.496	0.425	0.355	0.135	0.520	0.440
MAN [26]	**0.985**	0.640	0.580	0.634	0.948	0.641	0.582	0.632	**0.923**	**0.966**	0.540	0.611	**0.889**	**0.942**	0.578	0.566	**0.944**	**0.981**	**0.649**	0.477	0.735
Baseline (16 bits)	0.382	0.137	0.346	0.384	0.328	0.111	0.213	0.228	0.128	0.123	0.126	0.126	0.292	0.188	0.113	0.184	0.301	0.226	0.098	0.171	0.210
SePH [21] (16 bits)*	0.858	0.591	0.825	0.743	0.892	0.638	0.895	0.822	0.483	0.472	0.413	0.424	0.576	0.547	0.475	0.533	0.424	0.465	0.357	0.422	0.593
SePH$_{lr}$ [12] (16 bits)*	0.867	0.209	0.834	0.811	0.938	0.240	0.874	0.875	0.234	0.240	0.297	0.247	0.481	0.507	0.219	0.475	0.365	0.402	0.164	0.443	0.486
RoPH [34] (16 bits)*	0.786	0.716	0.839	0.744	0.725	0.504	0.327	0.496	0.438	0.414	0.461	0.445	0.504	0.481	0.440	0.444	0.366	0.324	0.407	0.307	0.509
LSRH [22] (16 bits)*	0.736	0.291	0.646	0.522	0.828	0.319	0.696	0.497	0.242	0.309	0.224	0.240	0.459	0.490	0.238	0.350	0.294	0.316	0.204	0.253	0.408
KDLFH [23] (16 bits)*	0.767	0.321	0.595	0.368	0.754	0.433	0.701	0.511	0.237	0.331	0.342	0.263	0.387	0.407	0.328	0.392	0.199	0.287	0.232	0.286	0.407
DLFH [23] (16 bits)*	0.143	0.108	0.511	0.391	0.367	0.110	0.454	0.351	0.184	0.123	0.162	0.162	0.247	0.233	0.101	0.153	0.055	0.056	0.100	0.165	0.209
MTFH [13] (16 bits)*	0.089	0.066	0.086	0.077	0.085	0.059	0.117	0.080	0.066	0.070	0.073	0.081	0.081	0.156	0.073	0.077	0.102	0.099	0.078	0.091	0.085
DCHN$_0$ (16 bits)	0.900	**0.900**	**0.901**	**0.898**	**0.969**	**0.968**	**0.970**	**0.968**	0.608	0.608	**0.610**	**0.602**	0.647	0.647	**0.649**	**0.637**	0.580	0.580	0.577	**0.580**	**0.740**
DCHN$_{100}$ (16 bits)	0.904	**0.886**	**0.904**	**0.904**	**0.966**	**0.945**	**0.966**	**0.966**	0.565	0.565	**0.572**	**0.570**	0.687	0.686	**0.677**	**0.685**	0.610	0.609	0.591	**0.611**	**0.743**
Baseline (32 bits)	0.540	0.275	0.133	0.603	0.498	0.236	0.127	0.436	0.214	0.227	0.109	0.200	0.083	0.123	0.079	0.071	0.338	0.307	0.156	0.129	0.244
SePH [21] (32 bits)*	0.890	0.626	0.879	0.790	0.915	0.698	0.906	0.850	0.469	0.489	0.528	0.461	0.626	0.620	0.519	0.619	0.516	0.474	0.402	0.492	0.638
SePH$_{lr}$ [12] (32 bits)*	0.889	0.291	0.864	0.856	0.953	0.302	0.891	0.899	0.283	0.303	0.342	0.288	0.517	0.487	0.266	0.546	0.392	0.420	0.192	0.420	0.520
RoPH [34] (32 bits)*	0.830	0.788	0.875	0.803	0.768	0.573	0.356	0.580	0.473	0.489	0.497	0.467	0.530	0.511	0.511	0.517	0.500	0.456	0.470	0.296	0.564
LSRH [22] (32 bits)*	0.862	0.340	0.773	0.689	0.900	0.348	0.843	0.713	0.281	0.291	0.261	0.293	0.577	0.518	0.286	0.451	0.411	0.401	0.226	0.373	0.492
KDLFH [23] (32 bits)*	0.854	0.557	0.832	0.654	0.890	0.610	0.893	0.771	0.447	0.459	0.455	0.405	0.564	0.605	0.475	0.549	0.396	0.414	0.335	0.421	0.579
DLFH [23] (32 bits)*	0.186	0.152	0.785	0.637	0.648	0.147	0.864	0.689	0.181	0.156	0.250	0.235	0.422	0.356	0.145	0.420	0.060	0.067	0.110	0.236	0.337
MTFH [13] (32 bits)*	0.087	0.065	0.110	0.107	0.081	0.074	0.123	0.056	0.088	0.065	0.091	0.083	0.074	0.114	0.102	0.062	0.090	0.047	0.061	0.073	0.083
DCHN$_0$ (32 bits)	0.907	**0.907**	**0.907**	**0.906**	**0.967**	**0.966**	**0.967**	**0.967**	0.634	0.634	**0.638**	**0.630**	0.706	0.706	**0.708**	**0.703**	0.574	0.574	0.573	**0.574**	**0.757**
DCHN$_{100}$ (32 bits)	0.905	**0.905**	**0.905**	**0.905**	**0.968**	**0.968**	**0.968**	**0.969**	0.617	0.617	**0.623**	**0.612**	0.699	0.699	**0.699**	**0.699**	0.599	0.599	0.598	**0.598**	**0.758**

* indicates the binary-view methods. In our experiments, these methods treat multi-view dataset as multiple binary-view datasets.

TABLE III
PERFORMANCE COMPARISON IN TERMS OF MAP SCORES ON THE MIRFLICKR-25K AND IAPR TC-12 DATASETS

Method	MIRFLICKR-25K								IAPR TC-12							
	Image $\rightarrow$ Text				Text $\rightarrow$ Image				Image $\rightarrow$ Text				Text $\rightarrow$ Image			
	16	32	64	128	16	32	64	128	16	32	64	128	16	32	64	128
Baseline	0.581	0.520	0.553	0.573	0.578	0.544	0.556	0.579	0.329	0.292	0.309	0.298	0.332	0.295	0.311	0.304
SePH [21]	0.729	0.738	0.744	0.750	0.753	0.762	0.764	0.769	0.467	0.476	0.486	0.493	0.463	0.475	0.485	0.492
SePH$_{lr}$ [12]	0.729	0.746	0.754	0.763	0.760	0.780	0.785	0.793	0.410	0.434	0.448	0.463	0.461	0.495	0.515	0.525
RoPH [34]	0.733	0.744	0.749	0.756	0.757	0.759	0.768	0.771	0.457	0.481	0.493	0.500	0.451	0.478	0.488	0.495
LSRH [22]	0.756	0.780	0.788	0.800	0.772	0.786	0.791	0.802	0.474	0.490	0.512	0.522	0.474	0.492	0.511	0.526
KDLFH [23]	0.734	0.755	0.770	0.771	0.764	0.780	0.794	0.797	0.306	0.314	0.351	0.357	0.307	0.315	0.350	0.356
DLFH [23]	0.721	0.743	0.760	0.767	0.761	0.788	0.805	0.810	0.306	0.314	0.326	0.340	0.305	0.315	0.333	0.353
MTFH [13]	0.581	0.571	0.645	0.543	0.584	0.556	0.633	0.531	0.303	0.303	0.307	0.300	0.303	0.303	0.308	0.302
DJSRH [14]	0.620	0.630	0.645	0.660	0.620	0.626	0.645	0.649	0.368	0.396	0.419	0.439	0.370	0.400	0.423	0.437
DCMH [9]	0.737	0.754	0.763	0.771	0.753	0.760	0.763	0.770	0.423	0.439	0.456	0.463	0.449	0.464	0.476	0.481
SSAH [20]	0.797	0.809	0.810	0.802	0.782	0.797	0.799	0.790	0.501	0.503	0.496	0.479	0.504	0.530	0.554	0.565
DCHN$_0$	**0.806**	**0.823**	**0.836**	**0.842**	**0.797**	**0.808**	**0.823**	**0.827**	0.487	0.492	**0.550**	**0.573**	0.481	0.488	0.543	**0.567**
DCHN$_{100}$	**0.813**	**0.816**	**0.823**	**0.840**	**0.808**	**0.803**	**0.814**	**0.830**	**0.533**	**0.558**	**0.582**	**0.596**	**0.527**	**0.557**	**0.582**	**0.595**

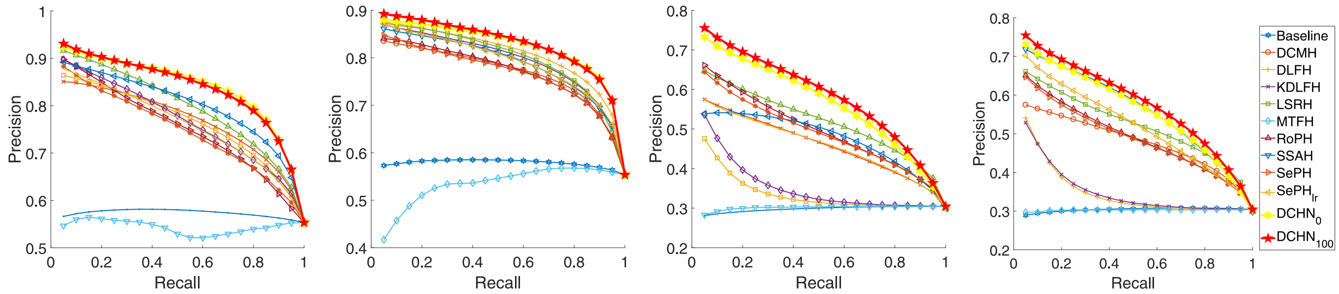
TABLE IV
PERFORMANCE COMPARISON IN TERMS OF MAP SCORES ON THE NUS-WIDE AND MS-COCO DATASETS

Method	NUS-WIDE								MS-COCO							
	Image $\rightarrow$ Text				Text $\rightarrow$ Image				Image $\rightarrow$ Text				Text $\rightarrow$ Image			
	16	32	64	128	16	32	64	128	16	32	64	128	16	32	64	128
Baseline	0.281	0.337	0.263	0.341	0.299	0.339	0.276	0.346	0.362	0.336	0.332	0.373	0.348	0.341	0.347	0.359
SePH [21]	0.644	0.652	0.661	0.664	0.654	0.662	0.670	0.673	0.586	0.598	0.620	0.628	0.587	0.594	0.618	0.625
SePH$_{lr}$ [12]	0.607	0.624	0.644	0.651	0.630	0.649	0.665	0.672	0.527	0.571	0.592	0.600	0.555	0.596	0.618	0.621
RoPH [34]	0.638	0.656	0.662	0.669	0.645	0.665	0.671	0.677	0.592	0.634	0.649	0.657	0.587	0.628	0.643	0.652
LSRH [22]	0.622	0.650	0.659	0.690	0.600	0.662	0.685	0.692	0.580	0.563	0.561	0.567	0.580	0.611	0.615	0.632
KDLFH [23]	0.323	0.367	0.364	0.403	0.325	0.365	0.368	0.408	0.373	0.403	0.451	0.542	0.370	0.400	0.449	0.542
DLFH [23]	0.316	0.367	0.381	0.404	0.319	0.379	0.386	0.415	0.352	0.398	0.455	0.443	0.359	0.393	0.456	0.442
MTFH [13]	0.265	0.473	0.434	0.445	0.243	0.418	0.414	0.485	0.288	0.264	0.311	0.413	0.301	0.284	0.310	0.406
DJSRH [14]	0.433	0.453	0.467	0.442	0.457	0.468	0.468	0.501	0.478	0.520	0.544	0.566	0.462	0.525	0.550	0.567
DCMH [9]	0.569	0.595	0.612	0.621	0.548	0.573	0.585	0.592	0.548	0.575	0.607	0.625	0.568	0.595	0.643	0.664
SSAH [20]	0.636	0.636	0.637	0.510	0.653	0.676	0.683	0.682	0.550	0.577	0.576	0.581	0.552	0.578	0.578	0.669
DCHN$_0$	**0.648**	**0.660**	**0.669**	0.683	**0.662**	**0.677**	**0.685**	**0.697**	**0.602**	**0.658**	**0.682**	**0.706**	**0.591**	**0.652**	**0.669**	**0.696**
DCHN$_{100}$	**0.654**	**0.671**	**0.681**	**0.691**	**0.668**	**0.683**	**0.697**	**0.707**	**0.662**	**0.701**	**0.703**	**0.720**	**0.650**	**0.689**	**0.693**	**0.714**

learn the common Hamming space with the flexible inputs, two variants of our method (i.e., DCHN$_0$ and DCHN$_{100}$) are also investigated. For DCHN$_0$, only the number of categories is available in SHAM. Regarding DCHN$_{100}$, 100 labels are available. From the experimental results, we can draw the following conclusions.

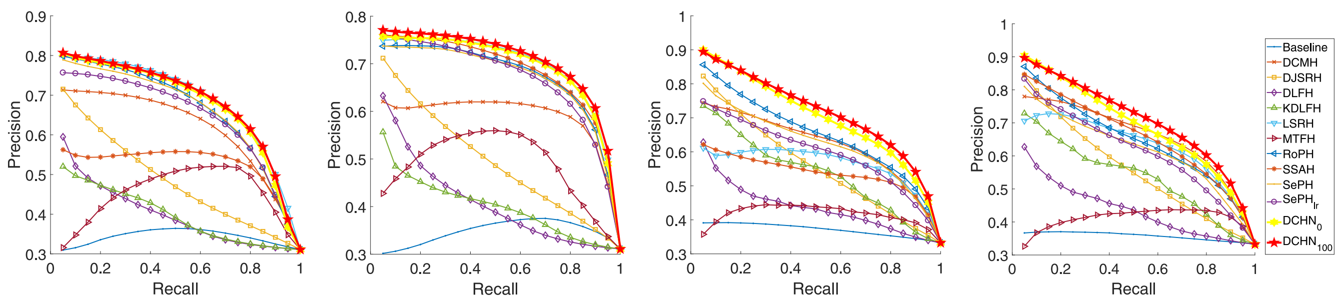
Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) (b) (c) (d)

Fig. 4. Precision–recall curves for (a) Img2Txt and (b) Txt2Img on MIRFLICKR-25K (c) Img2Txt and (d) Txt2Img on IAPR TC-12. The code length is 128.



(a) (b) (c) (d)

Fig. 5. Precision–recall curves for (a) Img2Txt and (b) Txt2Img on NUS-WIDE (c) Img2Txt and (d) Txt2Img on MS-COCO. The code length is 128.

1) Deep learning-based methods do not always outperform the traditional methods, and some shallow methods even perform better than the deep methods. The potential reason is that the used deep features contain higher level semantic information, which could boost the performance of the traditional cross-hashing methods.
2) All evaluated methods require using all views to jointly train their models. By incorporating the interview information into the training stage, some approaches outperform our DCHN$_0$ in a few tasks. However, it should be pointed out that our DCHN$_{100}$ can achieve the best performance using only 100 available labels without utilizing the interview information.
3) Directly using labels to decouple the hashing learning (i.e., Baseline) is inferior to other CVH methods, which indicates that the semantic information cannot be directly transferred into a common Hamming space with a separate training manner. Thus, we should elaborately design the decoupling methods for multiview hashing tasks like our DCHN.
4) Most existing CVH methods are specially designed for binary-view cases, which cannot be directly utilized to address the multiview problem (more than two views). In contrast, our DCHN could not only handle multiview data but also the *unfixed* views, while achieving the best performance even though comparing with real-value multiview methods. Besides the comparisons with the MAP score, we also adopt the precision–recall curves w.r.t. the code length of 128 to evaluate the performance on the MIRFLICKR-25K, IAPR TC-12, NUS-WIDE, and MC-COCO datasets (see Figs. 4 and 5). The result shows that our DCHN could also obtain encouraging results.

TABLE V
EFFICIENCY ANALYSIS IN TERMS OF THE TRAINING TIME COST (S) FOR A NEW VIEW ON THE XMEDIA, NUS-WIDE, AND MS-COCO DATASETS

<table>
<tr><th rowspan="2">Method</th><th colspan="5">XMedia</th><th colspan="2">NUS-WIDE</th><th colspan="2">MS-COCO</th></tr>
<tr><th>Image</th><th>Text</th><th>Audio</th><th>3D</th><th>Video</th><th>Image</th><th>Text</th><th>Image</th><th>Text</th></tr>
<tr><td>DJSRH* [14]</td><td colspan="5">289.93</td><td colspan="2">156.91</td><td colspan="2">90.16</td></tr>
<tr><td>DCMH* [9]</td><td colspan="5">245.43</td><td colspan="2">154.44</td><td colspan="2">93.16</td></tr>
<tr><td>SSAH* [20]</td><td colspan="5">9673.81</td><td colspan="2">5,921.92</td><td colspan="2">3671.59</td></tr>
<tr><td>MAN [26]</td><td colspan="5">90.65</td><td colspan="2">200.81</td><td colspan="2">382.30</td></tr>
<tr><td>DCHNj</td><td colspan="5">88.14</td><td colspan="2">153.98</td><td colspan="2">95.68</td></tr>
<tr><td>DCHNp</td><td colspan="5">34.61</td><td colspan="2">110.23</td><td colspan="2">60.11</td></tr>
<tr><td>DCHNs</td><td>30.67</td><td>27.52</td><td>2.78</td><td>4.90</td><td>7.68</td><td>103.58</td><td>42.93</td><td>57.93</td><td>23.28</td></tr>
</table>

* indicates the two-view methods, which should be trained by $v(v-1)/2$ times for v views.

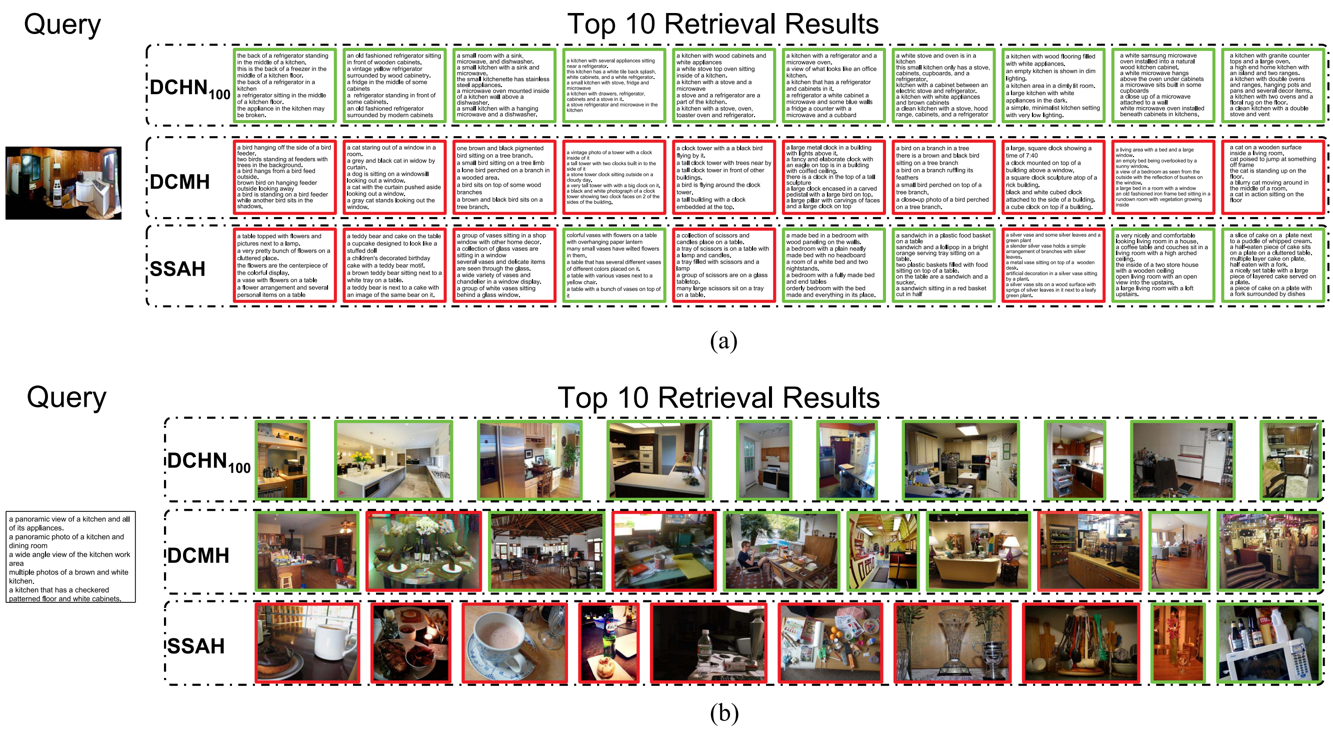
Finally, we evaluate the retrieval performance of our methods comparing with the deep CVH methods in terms of qualitative results on the MS-COCO dataset as show in Fig. 6. The correct results are the retrieved samples in the retrieval database that share at least one same class with a given query. From this figure, we can see that our DCHN achieves the best top retrieval results with a few labels (100 labels). In conclusion, our DCHN can embrace more advantages (e.g., more flexible) without losing any performance.

D. Efficiency Analysis About Increasing Views

To investigate the efficiency of our view-independent training paradigm, we report the time costs of three variants of our method on the XMedia database in Table V. To be specific,

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a)

(b)

Fig. 6. Cross-view retrieval result examples on the MS-COCO dataset by our proposed approach as well as compared methods DCMH and SSAH. For a given query (image/text on the left part), the top ten retrieval results (text/image on the right part) are shown on the above. In these examples, the correct retrieval results are with green borders, while the wrong results are with red borders. (a) Ten retrieval result examples for an image query. (b) Ten retrieval result examples for a text query.

"DCHN*j*" means that all the views are jointly used to train the model as the most existing CVH method. "DCHN*p*" means that all MHNs are trained in parallel at the same time and the reported time is to train all the views. "DCHN*s*" means that each MHN is separately trained and only one view is trained at the same time. In other words, the time cost of DCHN*s* could be regarded as the time for handling a newly coming view. For comparisons, we adopt DCMH, SSAH, and DJSRH as the baselines due to two reasons. On one hand, these three methods are based on DNNs and adopt the joint-view hashing paradigm. On the other hand, these methods are with GPU codes, whereas the methods, such as SePH, are shallow methods which are with only CPU codes. In the experiment, the maximal epochs of all the compared methods are set as 20 for a fair comparison within an acceptable time. Note that all tested methods could not converge in the same epochs, and we only focus on evaluating the efficiency of view-independent training on the epoch level. From Section IV-F, we could see that our method could approach convergence near the 20th epoch. Furthermore, our SHAM could precompute $\mathbf{W}$ only once on the available labels or class numbers before training MHN. It could converge very fast as shown Fig. 8 and cost little time to compute $\mathbf{W}$ (e.g., only 2.14 s for 50 iterations on an Intel i9-10900X CPU@3.70 GHz). Thus, we only compare the efficiency of our MHN with the other methods. Our view-independent training paradigm also could improve the training efficiency on two-view cases, for example, NUS-WIDE, MS-COCO, etc.

From the results, one could have the following conclusions. First, multiview methods (i.e., MAN and our method) are significantly more efficient than the two-view methods (i.e., DCMH, SSAH, and DJSRH) to handle multiple views (more than two views). Second, the proposed parallel method (DCHN*p*) is remarkably more efficient than the joint learning methods (i.e., MAN, DCMH, SSAH, and DJSRH). Such dominance is more distinct when the view number increases. For example, our method (DCHN*s*) can only cost much less time and resources to train a new view-specific model for a new view instead of the entire model as other methods. From the table, we can see that our view-independent training paradigm (DCHN*p*) can remarkably speed up the cross-view training up to 60.73% comparing with DCHN*j*. Furthermore, for a new view, DCHN*s* can reduce the time up to 96.85% comparing with DCHN*j* and MAN. Note that all these three variants are with the same retrieval accuracy with the same configurations (i.e., random seed, hyperparameters, etc.). Therefore, our DCHN is more efficient to handle new views and increasing views than the existing multiview methods. Such dominance is more outstanding comparing with these two-view methods.

E. Parameter Analysis and Ablation Study

To determine the value of λ and β, we randomly select some samples (2000 for each dataset) from the retrieval database to serve as the validation set by following [20]. In this section, Fig. 7 investigates the influence of these two parameters on NUS-WIDE with the code length of 32, as well as the ablation study. From the result, one could see that the best performance is achieved when $\lambda = 1$ and $\beta = 0.01$.

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

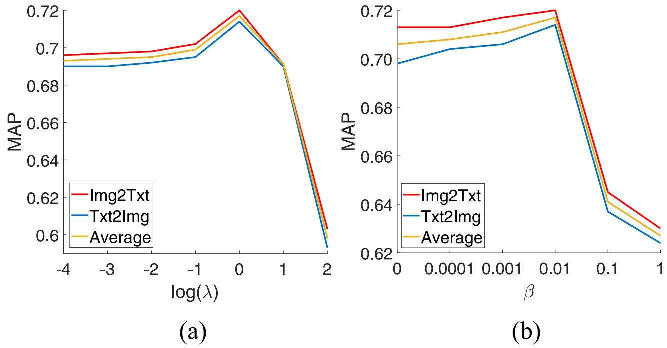


Fig. 7. Parameter analysis on MS-COCO. Note that $\beta = \{1, 0\}$ correspond to the ablation of the encoder and decoder of SHAM.

TABLE VI
COMPARISONS OF DIFFERENT NUMBER OF AVAILABLE LABELS IN TERMS OF MAP SCORES ON MS-COCO WITH 32 BITS

Available Labels	Image $\to$ Text	Text $\to$ Image	Average
0	0.658	0.652	0.655
10	0.673	0.666	0.669
50	0.693	0.679	0.686
100	0.701	**0.689**	**0.695**
150	**0.702**	0.688	**0.695**
200	**0.702**	0.683	0.693

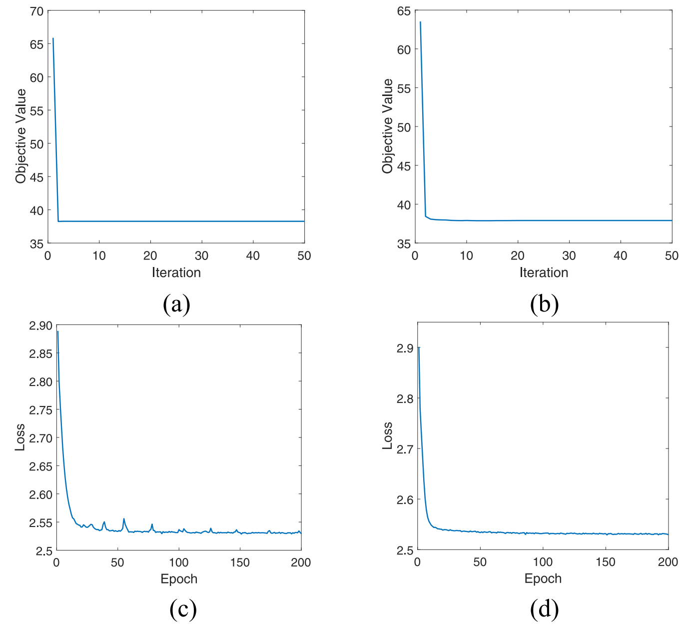


Fig. 8. Convergence analysis on MS-COCO. Objective value of (3) versus different numbers of iterations (a) only with the number of classes (DCHN_0) and (b) with 100 available labels (DCHN_{100}). Losses of MHN versus different numbers of epochs for (c) image with 100 available labels (DCHN_{100}) and (d) text with 100 available labels (DCHN_{100}).

Furthermore, the performance will decrease if the Hashing encoder (when $\beta = 1$) or the decoder (when $\beta = 0$) is removed, which indicates that both the encoder and the decoder contribute to the cross-view retrieval performance.

Furthermore, to investigate the impact of the number of available labels, we compare the performance of our DCHN on the different numbers of available labels in terms of MAP scores on MS-COCO with 32 bits. The comparison results are shown in Table VI. From the results, we could see that the real labels could improve the performance of DCHN_0 which does not use any labels. More labels could bring better retrieval performance in a certain range (i.e., 0–100 in Table VI). Furthermore, our DCHN could fast achieve a satisfied retrieval accuracy in a certain number of labels, and more labels could not bring significant improvement, thus our DCHN is insensitive to the number of available labels.

F. Convergence Analysis

We also evaluate the convergence of our method on the MS-COCO dataset. Fig. 8(a) and (b) shows the objective value of (3) versus a different number of iterations with the number of classes and 100 available labels on the MS-COCO dataset, respectively. From Fig. 8(a) and (b), we can see that our SHAM can quickly converge to a certain range. Moreover, Fig. 8(c) shows the losses of MHN versus different numbers of epochs for the image view on the MS-COCO dataset, and Fig. 8(d) shows the losses of MHN versus different numbers of epochs for the text view on the MS-COCO dataset. From the figures, one could see that the proposed MHN converges before the 100th epoch and the changing rates are much faster before the 20th epoch than later epochs. Therefore, we set the maximum epoch as 100. Note that the cross-view retrieval results of the proposed method are reported on the trained models of the last epoch, which is different from other methods that report the best performance throughout their training stages.

G. Visualization of Learned Representations

To visually investigate the discrimination of common representations learned by different cross-view methods, we adopt the *t*-SNE approach [55] to embed the samples from the PKU XMedia dataset into a 2-D space as shown in Fig. 9. Note that the most CVH methods cannot simultaneously project multiple views into a common Hamming space. Therefore, we only compare our method with two real-valued methods. From this figure, we can see that the learned representations of these cross-view methods from different views can overlap with each other indicating that they can project different views into a common space. The supervised methods can compact the samples with the same class and scatter the samples from different categories. Although the unsupervised method (MCCA) can project different views into a common space, the samples are scattered without any clustering center in the common space. Therefore, discrimination in multiview data is important for cross-view retrieval. From Fig. 9, we also can see that these methods attempt to project different views into a common space and separate the samples of different classes from each other. The degree of compactness for each class is consistent with the MAP results of cross-view retrieval. Obviously, our DCHN can make the different classes more scattered and the same ones more compact. That is to say, the proposed method can obtain more discriminative information from the cross-view data, which is consistent with the MAP scores for cross-view retrieval tasks.

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

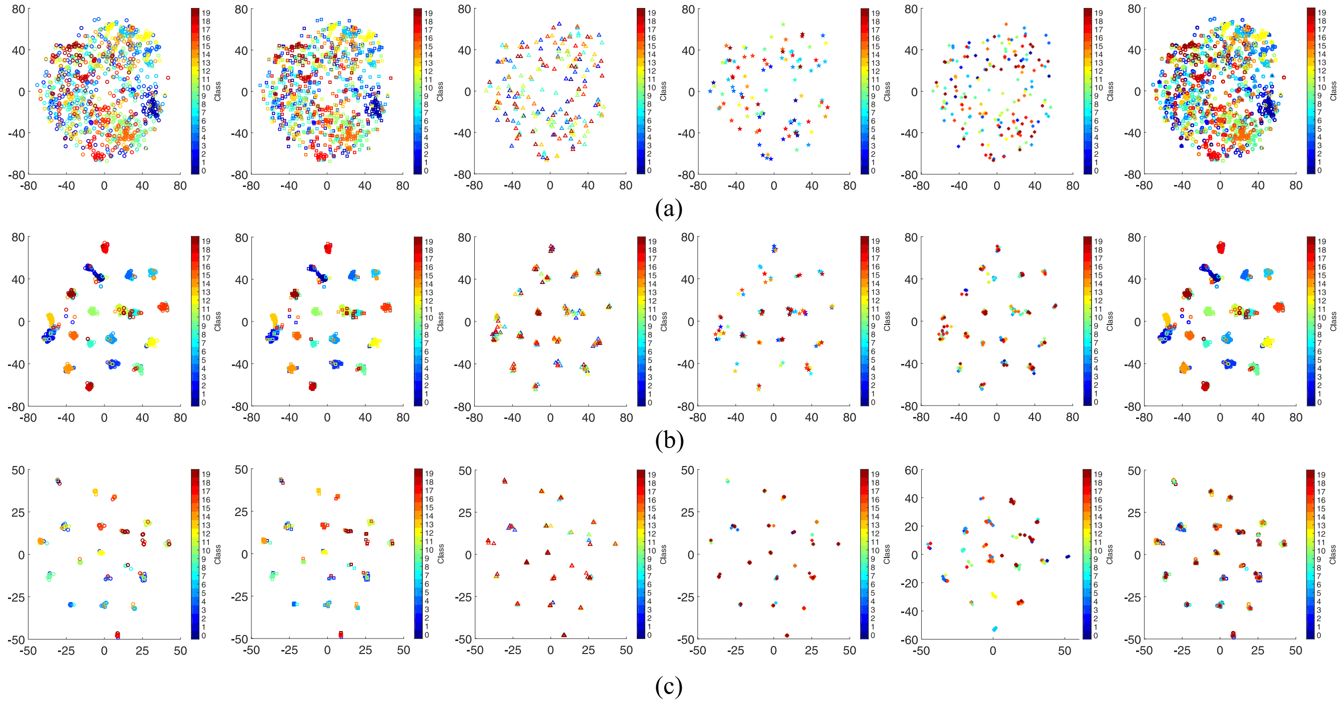


Fig. 9. Visualization for the query data on the PKU XMedia dataset by using the *t*-SNE method [55]. In this figure, the different shape of markers represents its corresponding view, and the different colors denote their corresponding classes, respectively. Each column represents a different view. From the left to the right, they are image, text, audio, 3-D model, and video, respectively. The length of the hash code is 32 bits. (a) MCCA [49]. (b) MAN [26]. (c) DCHN_{100}.

V. CONCLUSION

In this article, we proposed DCHNs to handle the data with an *unfixed* number of views. The major novelty of our idea is that DCHN does not jointly learn a common Hamming space as existing works did. Instead, we first learned a Hamming space from the flexible input via SHAM and then used it to perform view-specific hashing via the corresponding MHN. Such a hashing paradigm makes separately training view-specific networks possible, thus enjoying the advantages of handling large-scale views and increasing views. Extensive experiments showed that the proposed DCHN achieves state-of-the-art cross-view retrieval performance on three benchmark datasets while enjoying high computational efficiency. As for the future work, we attempt to extend our method to separately learn discrete representations from very few labeled data, which is more difficult in cross-view retrieval.

REFERENCES

[1] C. Xu, D. Tao, and C. Xu, "Multi-view learning with incomplete views," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5812–5825, Dec. 2015.

[2] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2833–2843, Jun. 2016.

[3] Z. Kang *et al.*, "Multi-graph fusion for multi-view spectral clustering," *Knowl. Based Syst.*, vol. 189, Sep. 2019, Art. no. 105102.

[4] L. Nie, W. Wang, R. Hong, M. Wang, and Q. Tian, "Multimodal dialog system: Generating responses via adaptive decoders," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 1098–1106.

[5] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[6] P. Hu, D. Peng, Y. Sang, and Y. Xiang, "Multi-view linear discriminant analysis network," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5352–5365, Nov. 2019.

[7] E. Yang, T. Liu, C. Deng, and D. Tao, "Adversarial examples for hamming space search," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1473–1484, Apr. 2020.

[8] C. Deng, E. Yang, T. Liu, and D. Tao, "Two-stream deep hashing with class-specific centers for supervised image search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2189–2201, Jun. 2020.

[9] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 3232–3240.

[10] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3893–3903, Aug. 2018.

[11] X. Liu, L. Huang, C. Deng, B. Lang, and D. Tao, "Query-adaptive hash code ranking for large-scale multi-view visual search," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4514–4524, Oct. 2016.

[12] Z. Lin, G. Ding, J. Han, and J. Wang, "Cross-view retrieval via probability-based semantics-preserving hashing," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4342–4355, Dec. 2017.

[13] X. Liu, Z. Hu, H. Ling, and Y.-M. Cheung, "MTFH: A matrix tri-factorization hashing framework for efficient cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. 10, 2019, doi: 10.1109/TPAMI.2019.2940446.

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

[15] M. Rastegari, J. Choi, S. Fakhraei, D. Hal, and L. Davis, "Predictable dual-view hashing," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1328–1336.

[16] X. Liu, L. Huang, C. Deng, J. Lu, and B. Lang, "Multi-view complementary hash tables for nearest neighbor search," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 1107–1115.

[17] X. Shen, F. Shen, Q.-S. Sun, and Y.-H. Yuan, "Multi-view latent hashing for efficient multimedia search," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 831–834.

[18] X. Peng, J. Feng, S. Xiao, W.-Y. Yauu, J. T. Zhou, and S. Yang, "Structured AutoEncoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

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

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

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

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

[22] K. Li, G.-J. Qi, J. Ye, and K. A. Hua, "Linear subspace ranking hashing for cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 9, pp. 1825–1838, Sep. 2017.

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

[24] L. Zhu, X. Lu, Z. Cheng, J. Li, and H. Zhang, "Flexible multi-modal hashing for scalable multimedia retrieval," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 2, pp. 1–20, 2020.

[25] L. Zhu, Z. Huang, X. Liu, X. He, J. Sun, and X. Zhou, "Discrete multimodal hashing with canonical views for robust mobile landmark search," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2066–2079, Sep. 2017.

[26] P. Hu, D. Peng, X. Wang, and Y. Xiang, "Multimodal adversarial network for cross-modal retrieval," *Knowl. Based Syst.*, vol. 180, pp. 38–50, Sep. 2019.

[27] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multiview clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[28] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: Multiview clustering without parameter selection," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5092–5101.

[29] Z. Zhang, Q. Zhu, G.-S. Xie, Y. Chen, Z. Li, and S. Wang, "Discriminative margin-sensitive autoencoder for collective multi-view disease analysis," *Neural Netw.*, vol. 123, pp. 94–107, Mar. 2020.

[30] L. Nie, X. Song, and T.-S. Chua, "Learning from multiple social networks," in *Synthesis Lectures on Information Concepts, Retrieval, and Services*, vol. 8. San Rafael, CA, USA: Morgan Claypool Publ., 2016, pp. 1–118.

[31] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 490–496, Feb. 2018.

[32] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Supervised discrete hashing with relaxation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 608–617, Mar. 2018.

[33] L. Zhu, Z. Huang, Z. Li, L. Xie, and H. T. Shen, "Exploring auxiliary context: discrete semantic transfer hashing for scalable image retrieval," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5264–5276, Nov. 2018.

[34] K. Ding, B. Fan, C. Huo, S. Xiang, and C. Pan, "Cross-modal hashing via rank-order preserving," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 571–585, Mar. 2017.

[35] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1618–1625.

[36] E. Yang, C. Deng, C. Li, W. Liu, J. Li, and D. Tao, "Shared predictive cross-modal deep quantization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5292–5303, Nov. 2018.

[37] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4032–4044, Aug. 2019.

[38] P. Hu, X. Wang, L. Zhen, and D. Peng, "Separated variational hashing networks for cross-modal retrieval," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 1721–1729.

[39] W.-C. Kang, W.-J. Li, and Z.-H. Zhou, "Column sampling based discrete supervised hashing," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1230–1236.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556.

[41] M. Ranzato, Y.-L. Boureau, and Y. L. Cun, "Sparse feature learning for deep belief networks," in *Advances in Neural Information Processing Systems*. Vancouver, BC, Canada: Curran Assoc., Inc., 2008, pp. 1185–1192.

[42] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3174–3183.

[43] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation AX + XB = C [F4]," *Commun. ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[44] X. Zhai, Y. Peng, and J. Xiao, "Learning cross-media joint representation with sparse and semisupervised regularization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 965–978, Jun. 2014.

[45] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *Proc. ACM Int. Conf. Multimedia Inf. Retrieval*, 2008, pp. 39–43.

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

[47] N. Rasiwasia *et al.*, "A new approach to cross-modal multimedia retrieval," in *Proc. Int. Conf. Multimedia*, 2010, pp. 251–260.

[48] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[49] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," in *Proc. Conf. Data Min. Data Warehouses*, 2010, pp. 1–4.

[50] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 2160–2167.

[51] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 188–194, Jan. 2016.

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

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980.

[54] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," in *Proc. Workshop Represent. Learn. NLP*, 2016, pp. 78–86.

[55] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Authorized licensed use limited to: ASTAR. Downloaded on November 03,2020 at 02:33:26 UTC from IEEE Xplore. Restrictions apply.