Semi-Paired Discrete Hashing: Learning Latent Hash Codes for Semi-Paired Cross-View Retrieval

Xiaobo Shen, Fumin Shen, Quan-Sen Sun, Yang Yang, Yun-Hao Yuan, and Heng Tao Shen, *Senior Member, IEEE*

***Abstract*—Due to the significant reduction in computational cost and storage, hashing techniques have gained increasing interests in facilitating large-scale cross-view retrieval tasks. Most cross-view hashing methods are developed by assuming that data from different views are well paired, e.g., text-image pairs. In real-world applications, however, this fully-paired multiview setting may not be practical. The more practical yet challenging semi-paired cross-view retrieval problem, where pairwise correspondences are only partially provided, has less been studied. In this paper, we propose an unsupervised hashing method for semi-paired cross-view retrieval, dubbed semi-paired discrete hashing (SPDH). In specific, SPDH explores the underlying structure of the constructed common latent subspace, where both paired and unpaired samples are well aligned. To effectively preserve the similarities of semi-paired data in the latent subspace, we construct the cross-view similarity graph with the help of anchor data pairs. SPDH jointly learns the latent features and hash codes with a factorization-based coding scheme. For the formulated objective function, we devise an efficient alternating optimization algorithm, where the key binary code learning problem is solved in a bit-by-bit manner with each bit generated with a closed-form solution. The proposed method is extensively evaluated on four benchmark datasets with both fully-paired and semi-paired settings and the results demonstrate the superiority of SPDH over several other state-of-the-art methods in term of both accuracy and scalability.**

***Index Terms*—Cross-view retrieval, discrete hashing, semi-paired data.**

I. Introduction

Recent years have witnessed the explosive growth of the multimedia data, which brings great challenges to information search and retrieval. Hashing [1] has attracted considerable interests for its great gains of both storage and computation in massive multimedia data. It has been widely used in approximate nearest neighbor search [1], image retrieval [2], image processing [3], [4], and so on. The basic idea of hashing is to learn a set of short binary codes for high-dimensional data while preserving similarity structure in the original space. Until now many hashing methods [1], [2], [5]–[12] have been proposed. Nevertheless, these hashing methods are single-view approaches, which focus on learning binary codes from data with only single view.

In many real-world applications, we often meet such a case that one object can be represented by multiple kinds of features [13], [14]. For example, each webpage can be jointly represented with both text, image, and hyper-links. This kind of data is referred as *multiview data*[1] [13]. Existing hashing methods learning from multiview data can be mainly divided into two categories: 1) multiview hashing (MVH) and 2) cross-view hashing (CVH). By leveraging multiple views, MVH [17]–[19] aims to learn better codes than single-view hashing, but requires that all views should be available in advance. Different from the purpose of MVH, CVH is proposed to support cross-view retrieval [20], where a query of one view can search for the relevant results of another view. For instance, one might need to find images on the Web that best illustrate given texts, or find texts that best match given images. Consequently, CVH is of great practical demand and interest to many applications.

Recently, some useful attempts [15], [16], [21]–[31] have been made toward effective CVH, which exploits correlations and similarity structures across multiple views. Cross-modality similarity sensitive hashing (CMSSH) [15] is proposed to learn hash functions among different views. The work in [21] extends spectral hashing (SH) to multiview fields.

Manuscript received May 26, 2016; revised July 31, 2016; accepted August 11, 2016. This work was supported in part by the National Science Foundation of China under Grant 61273251, Grant 61402203, Grant 61673220, Grant 61502081, Grant 61300161, and Grant 61572108. This paper was recommended by Associate Editor L. Rutkowski. *(Corresponding author: Quan-Sen Sun.)*

X. Shen and Q.-S. Sun are with the School of Computer and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: njust.shenxiaobo@gmail.com; sunquansen@njust.edu.cn).

F. Shen and Y. Yang are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: fumin.shen@gmail.com; dlyyang@gmail.com).

Y.-H. Yuan is with the Department of Computer Science and Technology, Yangzhou University, Yangzhou 225000, China (e-mail: yyhzbh@163.com).

H. T. Shen is with the School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, QLD 4072, Australia (e-mail: shenht@itee.uq.edu.au).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

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

Digital Object Identifier 10.1109/TCYB.2016.2606441

[1] Here, “view” can be replaced as other terms, such as “feature,” “modal,” and “modality” [13], [15], [16].

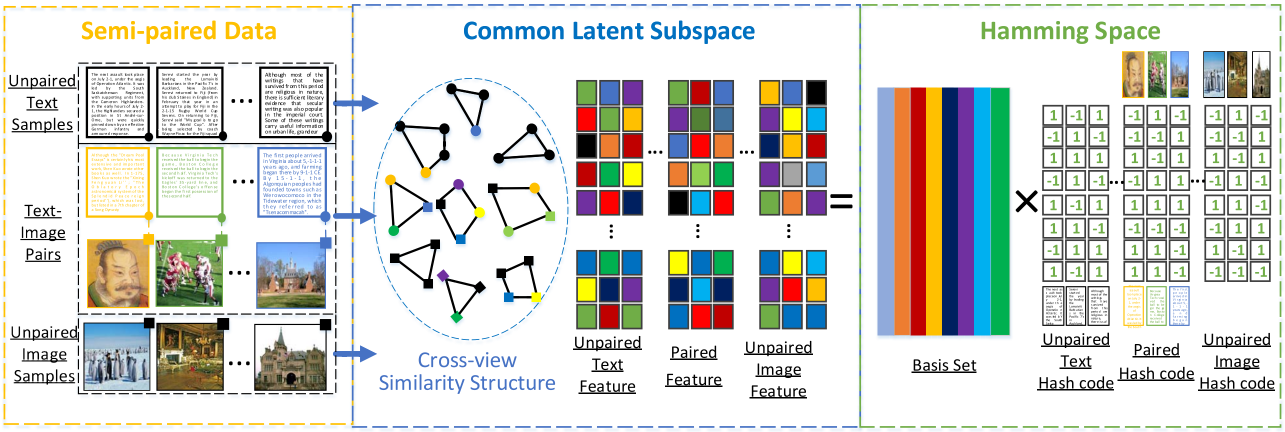


Fig. 1. Illustration of the proposed SPDH. SPDH first linearly maps semi-paired data into a common latent subspace, where both the paired and unpaired samples can be aligned well. Similarity structure of the common subspace is explored via the proposed effective cross-view similarity graph. A factorization-based discrete hash coding scheme is further presented to bridge the common subspace and Hamming space.

Both intermedia hashing (IMH) [23] and linear cross-modal hashing (LCMH) [16] introduce intraview similarity preservation and interview consistency to discover a compact Hamming space, and solve hash functions by eigenvalue decomposition. Collective matrix factorization hashing (CMFH) [24] learns unified binary codes by collective matrix factorization with latent factor model, which can support cross-view retrieval. In [25], a supervised CVH method, called semantic correlation maximization (SCM), is proposed to seamlessly integrate semantic labels into hash learning. Quantized correlation hashing (QCH) [29] is proposed to consider both quantization loss and relations between views in a unified learning process.

While showing promising performances, most existing CVH methods can only handle the fully-paired data, by requiring that all the objects are fully paired or have the one-to-one correspondence among different views. Nevertheless, such requirement may not hold anymore, when some views of objects become missing, which is common in the practical applications [32]–[35]. For example, in webpage search, many webpages may not contain any linkage information. Also, in text image retrieval, as shown in Fig. 1, some images or text descriptions do not exist, then the pairwise correspondences between them cannot be established. In practical, we are often given such kind of multiview data, where only partial objects are paired, and the others are unpaired. In this paper, such data is referred as *semi-paired data*,[2] and cross-view retrieval on such data is referred as *semi-paired cross-view retrieval*. Generally speaking, semi-paired cross-view retrieval is more challenging than the conventional one, because the cross-view prior information among the semi-paired data is limited. This limitation makes it very necessary to develop hashing methods that can work with semi-paired data. To our knowledge, only two recently proposed methods, i.e., IMH [23] and partial multimodal hashing (PM^{2}H) [27] have been formally formulated to perform the semi-paired cross-view retrieval in hashing research. Both methods consider the within-view similarity structure in each view and the cross-view consistency via the partially given correspondences. However, cross-view similarities between unpaired data cannot been fully explored in these two methods, as the graphs in their methods are defined within each view. Meanwhile, both of them learn the continuous feature and binary codes in two independent stages; the connection between two stages is lost, which may lead to the nonoptimal hash codes. In addition, their training time complexities are around quadric or cubic to the size of dataset, which are relatively high in the large-scale applications. Generally speaking, it still remains an open problem how to learn good hash codes on the large scale semi-paired data.

In this paper, we propose an unsupervised hashing method for the semi-paired cross-view retrieval problem, thus termed semi-paired discrete hashing (SPDH). SPDH aims to efficiently generate latent hash codes while preserving the intrinsic similarities of semi-paired data. The proposed SPDH is outlined in Fig. 1. We summarize the main contributions of this paper as follows.

1) SPDH focuses on the *semi-paired cross-view retrieval*, where partial pairwise correspondences are provided from training data in the unsupervised setting. This challenging problem has been less considered in the CVH research.
2) SPDH explores the underlying structure of the learned common latent subspace, where both paired and unpaired samples can be well aligned. In addition, an effective similarity graph is efficiently constructed in order to preserve the similarities of semi-paired data in the latent common space.
3) The latent features are learned with hash codes via a factorization-based binary coding scheme. Considering the discrete nature of hashing, the hash codes are optimized bit by bit with each hash bit generated by an analytical solution. The training complexity of SPDH is linear with size of the dataset. Hence, SPDH is scalable and efficient for the large-scale applications.

[2] In some literature, they are also named as weakly-paired data [32] or partially-paired data [27], [33].

4) We perform extensive evaluation of the proposed method on four benchmark datasets. The results show our approach outperforms several other state-of-the-art methods with both fully-paired and semi-paired settings.

The remainder of this paper is organized as follows. We first briefly review related works in Section II. The details of the proposed method are presented in Section III. Extensive experimental evaluation are given in Section IV. Finally, we draw conclusions in Section V.

II. Related Work

In this section, we preliminarily review several topics related to this paper.

A. Hashing on Single-View Data

Most existing hashing methods utilize single-view data to generate hash codes. Accordingly, we name this category as single-view hashing methods, which can also be broadly divided into two categories: 1) *data-independent* methods and 2) *data-dependent* methods. The data-independent hashing methods mainly include locality sensitive hashing [5] and its extensions [10], [36], [37].

Later considerable attentions have been paid to data-dependent hashing methods, which apply some machine learning techniques to generate more compact data-related hash functions. One of the earliest work in this category is SH [2], which utilizes the distribution of data and turns to be eigen-decomposition problem of a graph Laplacian matrix [38]. Later anchor graph hashing [6] is proposed to adopt anchor graph for hashing learning, and can be more efficient than SH. To ease the quantization loss in the binarization process, Gong *et al.* [7] proposed iterative quantization (ITQ), which aims to find an optimal rotation matrix such that the difference between binary codes and original data is minimized. Besides, some other variants in this category can be found in [1], [8], [9], [11], [12], and [39]–[43].

B. Hashing on Multiview Data

Recent years more and more multiview data [13], [14] have been available in real applications, consequently hashing on the multiview data has received lots of attentions.

One category is MVH [17]–[19], [44]–[46], which fuses multiple sources from the same objects to get better binary codes than the single-view methods. Multiple feature hashing [17] preserves the local structure of each view and globally considers the alignments of all views to learn a group of hash functions. Multiple feature kernel hashing [45] learns hash functions by preserving certain similarities with linearly combined multiple kernels corresponding to different features. Kim *et al.* [18] proposed multiview SH, which computes the α-averaged similarity matrix from all views, and adopts the sequential learning approach to obtain the hash function. Recently, based on regularized kernel non-negative matrix factorization, multiview alignment hashing [19] is proposed to seek a matrix factorization to effectively fuse the multiple views.

Besides, another category is CVH, which supports the cross-view retrieval. The proposed method belongs to this category. Until now many CVH methods [15], [16], [21]–[31] have been proposed. Bronstein *et al.* [15] proposed the cross-modality search hashing (CMSSH), which learns hash functions among different views. Kumar and Udupa [21] extended SH to multiview fields. LCMH [16] introduces intraview similarity preservation and interview consistency to discover a compact Hamming space, and solves hash functions by eigenvalue decomposition. CMFH [24] learns unified binary codes by collective matrix factorization with latent factor model, which can support cross-view retrieval. Zhang and Li [25] proposed a supervised CVH method, called SCM, to seamlessly integrate semantic labels into hash learning. QCH [29] is proposed to consider both quantization loss and relations between views in a unified learning process.

C. Learning on Semi-Paired Data

In real-world applications, semi-paired data is prevalent due to the fact that some data in certain views are often missing. One the other hand, manually pairing the unpaired data is difficult, because it often requires the expertise of some specific domain. Hence, it is significant to directly learn on semi-paired data. Note that most multiview learning methods [13], e.g., canonical correlation analysis (CCA) [47] and partial least squares [48] assume all data are fully paired, and they fail to directly deal with semi-paired data.

Until now a few multiview learning methods [32]–[35] have been proposed to perform various tasks on semi-paired data. For example, Li *et al.* [33] first proposed a method in multiview clustering, named partial multiview clustering to perform clustering tasks on semi-paired data. The clustering experiments on two-view data demonstrate its effectiveness. To break the full pairwise requirement of CCA, Rasiwasia *et al.* [34] proposed cluster-CCA to perform joint dimensionality reduction on the semi-paired data. The correspondences between the sets are defined by class labels, but label information may not be available in the real-world applications. Recently, local group-based consistent feature learning method (LGCFL) [35] is proposed to do cross-view retrieval on semi-paired data. In essence, LGCFL is a supervised learning method.

In hashing area, two pioneering works have focused on cross-view retrieval on semi-paired data. One is IMH [23]. IMH constructs the intraview similarity preservation term in each view, and utilizes the partial available correspondence to align two views. Then it is formulated as an eigenvalue decomposition problem via spectral relaxation. The sign function is finally adopted to binarize the continuous features into binary codes. The other is the recently proposed PM^{2}H [27]. Similarly, PM^{2}H ensures the data consistency among different modalities and preserves data similarity within the same modality through graph Laplacian. Hash codes are finally obtained via the additional orthogonal rotation using ITQ.

TABLE I
IMPORTANT NOTATIONS USED IN THIS PAPER

Notation	description
$\mathbf{X}^{(i)}$	data matrix of the i-th view
$\mathbf{W}^{(i)}$	projection matrix of the i-th view
$\mathbf{Y}^{(i)}$	feature matrix of the i-th view
$\mathbf{Y}$	feature matrix of all the samples
$\mathbf{U}$	basis matrix
$\mathbf{B}$	latent hash code matrix
$\mathbf{P}$	projection matrix
$\mathbf{L}$	cross-view graph Laplacian matrix
θ_i	learning weight of the i-th view
d_i	dimensionality of the i-th view
d	dimensionality of the common subspace
n	the number of total objects
n_i	the number of objects in the i-th view
c	the length of hash code

III. SEMI-PAIRED DISCRETE HASHING

A. Problem Statement

We present the problem statement of the semi-paired cross-view retrieval. The database consists of samples from one view while the query consists of samples from a different view. For simplicity, we hereafter consider only two views (e.g., text view and image view). Specifically, suppose we have two different views $\mathbf{X}^{(1)} = [\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}, \ldots, \mathbf{x}_{n_1-n_0}^{(1)}, \mathbf{x}_{n_1-n_0+1}^{(1)}, \ldots, \mathbf{x}_{n_1}^{(1)}]$, $\mathbf{X}^{(2)} = [\mathbf{x}_1^{(2)}, \mathbf{x}_2^{(2)}, \ldots, \mathbf{x}_{n_0}^{(2)}, \mathbf{x}_{n_0+1}^{(2)}, \ldots, \mathbf{x}_{n_2}^{(2)}]$, where $\mathbf{x}^{(1)} \in \mathbb{R}^{d_1}$, $\mathbf{x}^{(2)} \in \mathbb{R}^{d_2}$ (usually $d_1 \neq d_2$), d_i and n_i ($i = 1, 2$) are the dimensionality and the number of samples in the ith view, respectively. Without loss of generality, we assume the last n_0 samples in the first view and the first n_0 samples in the second view come from the same n_0 objects, that is, $\{\mathbf{x}_{n_1-n_0+i}^{(1)}, \mathbf{x}_i^{(2)}\}_{i=1}^{n_0}$ are pairs, where n_0 is the number of pairs, while rest samples lack such one-to-one correspondence. We denote n as the number of total objects, and $n = n_1 + n_2 - n_0$. Besides, we further assume samples in each view are zero-centered, i.e., $\sum_{i=1}^{n_1} \mathbf{x}_i^{(1)} = 0$ and $\sum_{i=1}^{n_2} \mathbf{x}_i^{(2)} = 0$.

The goal of SPDH is to learn the view-specific hash functions to map samples of different views into a common Hamming space, where similarities of semi-paired data should be preserved. The important notations in this paper are summarized in Table I.

B. Proposed Formulation

1) Common Latent Subspace Learning: In multiview learning [13], it is critical to analyze the relationships between views. It is commonly known that if data described in different views are related to similar topics, they are expected to share a certain common subspace [20]. Specifically, in our problem, we assume that there exists the view-specific projection matrix $\mathbf{W}^{(i)}$ for the ith view ($i = 1, 2$), by which $\mathbf{X}^{(i)}$ can be mapped into such common subspace. To achieve this goal, we optimize the following problem:

$$\min_{\mathbf{W}^{(1)}, \mathbf{Y}^{(1)}} \left\| \mathbf{W}^{(1)T} \mathbf{X}^{(1)} - \mathbf{Y}^{(1)} \right\|_F^2$$
$$\min_{\mathbf{W}^{(2)}, \mathbf{Y}^{(2)}} \left\| \mathbf{W}^{(2)T} \mathbf{X}^{(2)} - \mathbf{Y}^{(2)} \right\|_F^2 \quad (1)$$

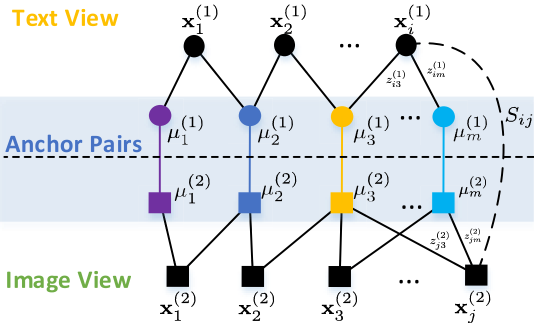


Fig. 2. Illustration of the proposed cross-view similarity graph construction. Anchor pairs, i.e., $\{\boldsymbol{\mu}_k^{(1)}, \boldsymbol{\mu}_k^{(2)}\}_{k=1}^m$ are randomly selected from paired samples from different views. The cross-view similarity S_{ij} between $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_j^{(2)}$ can be inferred according to their within-view similarities with the anchor pairs. If two samples share more co-occurring neighborhood anchor pairs, the similarity between them should be larger.

where $\mathbf{W}^{(1)} \in \mathbb{R}^{d_1 \times d}$ and $\mathbf{W}^{(2)} \in \mathbb{R}^{d_2 \times d}$ are two view-specific projection matrices, $\mathbf{Y}^{(1)} = [\bar{\mathbf{Y}}^{(1)}, \tilde{\mathbf{Y}}] \in \mathbb{R}^{d \times n_1}$ and $\mathbf{Y}^{(2)} = [\tilde{\mathbf{Y}}, \bar{\mathbf{Y}}^{(2)}] \in \mathbb{R}^{d \times n_2}$ are the generated feature matrices of two views, $\tilde{\mathbf{Y}} \in \mathbb{R}^{d \times n_0}$ is the shared feature matrix of the paired samples, and $\bar{\mathbf{Y}}^{(i)} \in \mathbb{R}^{d \times (n_i - n_0)}$ is the feature matrix of the unpaired samples from the ith view, d is the dimensionality of the common subspace.

2) Similarity Preservation: Similarity preservation [1], [2], [6], [15] is crucial for hashing to achieve good performance. Therefore, similarity (local) structure of the features in the common subspace should be preserved as much as possible.

It is very challenging to directly analyze the similarities among semi-paired data, as they belong to different views and partial pairwise information is also not available. The conventional graph construction approaches, e.g., k-NN approach [2], [49] cannot be directly employed in our problem. In this paper, inspired with the idea of anchor graph [6], we design a simple yet effective cross-view graph construction approach to uncover similarities among the semi-paired data.

The proposed cross-view graph construction approach is illustrated in Fig. 2. The main idea is utilizing the within-view similarity to measure the cross-view similarity with the help of anchor data pairs. Specifically, we first randomly select m pairs of paired samples as anchor data pairs, denoted as $\{\boldsymbol{\mu}_k^{(1)}, \boldsymbol{\mu}_k^{(2)}\}_{k=1}^m$. Then cross-view similarity between $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_j^{(2)}$ can be calculated as

$$S_{ij} = \sum_{k=1}^{m} Z_{ik}^{(1)} \times Z_{jk}^{(2)} \quad (2)$$

where $Z_{ik}^{(1)}$ denotes the similarity between $\mathbf{x}_i^{(1)}$ and $\boldsymbol{\mu}_k^{(1)}$, and $Z_{jk}^{(2)}$ denotes the similarity between $\mathbf{x}_j^{(2)}$ and $\boldsymbol{\mu}_k^{(2)}$, both of which can be computed similarly with that in anchor graph [6]. The physical meaning of (2) is obvious: if $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_j^{(2)}$ share more co-occurring neighborhood anchor data pairs, S_{ij} should be larger. For example, $\mathbf{x}_1^{(1)}$ and $\mathbf{x}_1^{(2)}$ in Fig. 2 share two common neighborhood anchor data pairs, i.e., $\{\boldsymbol{\mu}_k^{(1)}, \boldsymbol{\mu}_k^{(2)}\}_{k=1}^2$, while $\mathbf{x}_1^{(1)}$ and $\mathbf{x}_3^{(2)}$ share no common neighborhood anchor data pairs, then we infer that $\mathbf{x}_1^{(1)}$ will be more similar to $\mathbf{x}_1^{(2)}$ than $\mathbf{x}_3^{(2)}$.

To this end, the similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ among the semi-paired data can be constructed as

$$\mathbf{S} = \mathbf{Z}\boldsymbol{\Lambda}^{-1}\mathbf{Z}^T \tag{3}$$

where $\mathbf{Z} = [\bar{\mathbf{Z}}^{(1)}; \tilde{\mathbf{Z}}; \bar{\mathbf{Z}}^{(2)}]$, $\bar{\mathbf{Z}}^{(i)} \in \mathbb{R}^{(n_i-n_0)\times m}$ is a similarity matrix defined in the ith view ($i = 1, 2$), measuring the similarities between unpaired samples and anchors, $\tilde{\mathbf{Z}} \in \mathbb{R}^{n_0\times m}$ denotes the similarities between paired samples and anchors, which can be computed in either view. $\Lambda = \operatorname{diag}(\mathbf{Z}^T\mathbf{1}) \in \mathbb{R}^{m\times m}$ is used for normalizing each row.

After obtaining $\mathbf{S}$, we achieve similarity preservation by minimizing the following problem:

$$\min_{\mathbf{Y}} \sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij}\left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2 = \operatorname{Tr}\left(\mathbf{Y}\mathbf{L}\mathbf{Y}^T\right) \tag{4}$$

where $\mathbf{L} = \mathbf{I} - \mathbf{S} \in \mathbb{R}^{n\times n}$ is the graph Laplcian [2], and $\mathbf{Y} = [\bar{\mathbf{Y}}^{(1)}, \tilde{\mathbf{Y}}, \bar{\mathbf{Y}}^{(2)}] \in \mathbb{R}^{d\times n}$ is the latent features of all the samples in the common subspace.

3) Hash Coding Scheme: We next consider learning the target hash codes. Some existing hashing methods directly use the simple sign function for binarization, and others employ some learning-based binarization schemes, e.g., ITQ. In this paper, we propose to utilize matrix factorization technique [50] to establish the hash coding scheme, which learns binary latent factors to well reconstruct $\mathbf{Y}$ under a certain basis set. Specifically, we have the following optimization problem:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{B}} \quad & \|\mathbf{Y} - \mathbf{U}\mathbf{B}\|_F^2 \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{c\times n} \end{aligned} \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{d\times c}$ is the basis matrix, and $\mathbf{B}$ is the binary code matrix, c is the code length. $\mathbf{U}$ can be regarded as a set of certain semantic concepts, and $\mathbf{Y}$ can be regarded as the linear combinations of these concepts under the binary discrete constraint.

4) Overall Objective Function: Due to the difficulty of directly learning discrete binary codes, one conventional way is to bypass the discrete optimization problem by some certain relaxation strategy, which, however, separates the binary code learning into two mutually-independent stages, i.e., learning continuous representations and transforming into binary codes via some binarization methods. Typically, such optimization scheme ignores the correlation of the above two stages, which may severely limit the representative power of the generated binary codes.

To cope with the above problem, we propose to jointly learn latent feature representation and consistent hash codes within one framework. By summarizing the above three parts, i.e., (1), (4), and (5), we finally formulate our joint optimization problem as follows:

$$\begin{aligned} \min_{\substack{\mathbf{W}^{(i)},\theta^{(i)},\\ \mathbf{Y},\mathbf{U},\mathbf{B}}} \quad & \sum_{i=1}^{2}\theta^{(i)}\left(\left\|\mathbf{W}^{(i)T}\mathbf{X}^{(i)} - \mathbf{Y}^{(i)}\right\|_F^2\right) \\ & + \alpha\operatorname{Tr}\left(\mathbf{Y}\mathbf{L}\mathbf{Y}^T\right) + \gamma\|\mathbf{Y} - \mathbf{U}\mathbf{B}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Y}\mathbf{Y}^T = \mathbf{I},\ \text{and}\ \mathbf{B} \in \{-1, 1\}^{c\times n} \\ & \text{and} \sum_{i=1}^{2}\theta^{(i)} = 1, \theta^{(i)} > 0, i = 1, 2 \end{aligned} \tag{6}$$

where $\theta^{(i)}$ is a variable for weighting the relative importance of the ith view in the learning process, α and γ are two non-negative tradeoff parameters, weighting the relative importances of the similarity preservation term and the hash code reconstruction error term. We further impose the orthogonality constraint on $\mathbf{Y}$ to make the latent features uncorrelated. Note that without this orthogonality constraint, the proposed model always has a trivial solution, that is, $\mathbf{W}^{(i)}$, $\mathbf{Y}$, $\mathbf{U}$ all equal to $\mathbf{0}$, and $\mathbf{B}$ is arbitrary, which is useless in our application.

C. Optimization Algorithm

Directly minimizing the objective function in (6) is intractable as it is a nonconvex optimization problem. Meanwhile, the discrete and orthogonal constraints makes the problem more difficult to solve. However, we will show that it is tractable to solve the problem with respect to one variable while keeping other variables fixed. In the following, we describe an iterative algorithm to update these variables until convergence.

1) Optimization on $\mathbf{W}^{(i)}$: By dropping some terms irrelevant to $\mathbf{W}^{(i)}$ ($i = 1, 2$), we have

$$\min_{\mathbf{W}^{(i)}} \left\|\mathbf{W}^{(i)T}\mathbf{X}^{(i)} - \mathbf{Y}^{(i)}\right\|_F^2. \tag{7}$$

Let the derivative of (7) with respect to $\mathbf{W}^{(i)}$ equal to $\mathbf{0}$, then, we obtain

$$\mathbf{W}^{(i)} = \left(\mathbf{X}^{(i)}\mathbf{X}^{(i)T} + \epsilon\mathbf{I}\right)^{-1}\mathbf{X}^{(i)}\mathbf{Y}^{(i)T} \tag{8}$$

where $\mathbf{I} \in \mathbb{R}^{d_i\times d_i}$ is an identity matrix, which is used to avoid the overfitting of $\mathbf{W}^{(i)}$, ϵ is a small non-negative parameter, we simply set $\epsilon = 0.001$.

2) Optimization on $\mathbf{Y}$: For purpose of clear presentation, we first define two view-specific element selection matrix $\mathbf{Q}^{(1)} = [\mathbf{1}_{n_1\times n_1}, \mathbf{0}_{n_1\times(n-n_1)}]^T$, and $\mathbf{Q}^{(2)} = [\mathbf{0}_{n_2\times(n-n_2)}, \mathbf{1}_{n_2\times n_2}]^T$. With the help of $\mathbf{Q}^{(i)}$ ($i = 1, 2$), we can easily use $\mathbf{Y}$ to represent $\mathbf{Y}^{(i)}$, i.e., $\mathbf{Y}^{(i)} = \mathbf{Y}\mathbf{Q}^{(i)}$.

Putting $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ into (6), we have

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \sum_{i=1}^{2}\theta^{(i)}\left(\left\|\mathbf{W}^{(i)T}\mathbf{X}^{(i)} - \mathbf{Y}\mathbf{Q}^{(i)}\right\|_F^2\right) \\ & + \alpha\operatorname{Tr}\left(\mathbf{Y}\mathbf{L}\mathbf{Y}^T\right) + \gamma\|\mathbf{Y} - \mathbf{U}\mathbf{B}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Y}\mathbf{Y}^T = \mathbf{I}. \end{aligned} \tag{9}$$

We can further rewrite (9) in a more compact form

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \|\mathbf{F} - \mathbf{Y}\mathbf{Q}\|_F^2 + \alpha\operatorname{Tr}\left(\mathbf{Y}\mathbf{L}\mathbf{Y}^T\right) \\ \text{s.t.} \quad & \mathbf{Y}\mathbf{Y}^T = \mathbf{I} \end{aligned} \tag{10}$$

Algorithm 1 Curvilinear Search Algorithm Based on Cayley Transformation

```
Input: initial point $\mathbf{Y}^{(0)} \in \mathcal{M}_n^{d}$[3], matrix $\mathbf{F}$, $\mathbf{Q}$, graph Laplacian $\mathbf{L}$.
Output: $\mathbf{Y}^{(k)}$.
Initialize $k = 0$, $\epsilon > 0$, and $0 < \rho_1 < \rho_2 < 1$.
repeat
    Compute the gradient $\mathbf{G}$ according to (11);
    Generate the skew-symmetric matrix $\mathbf{A} = \mathbf{G}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{G}$;
    Compute the step size $\tau_k$, that satisfies the Armijo-Wolfe conditions [51] via the line search along the path $\mathbf{H}_k(\tau)$ defined by (12);
    Set $\mathbf{Y}^{(k+1)} = \mathbf{H}(\tau_k)$;
    Set $k = k + 1$;
until convergence
```

where $\mathbf{F} = [\sqrt{\theta^{(1)}}\mathbf{W}^{(1)T}\mathbf{X}^{(1)}, \sqrt{\theta^{(2)}}\mathbf{W}^{(2)T}\mathbf{X}^{(2)}, \sqrt{\gamma}\mathbf{U}\mathbf{B}]$, $\mathbf{Q} = [\sqrt{\theta^{(1)}}\mathbf{Q}^{(1)}, \sqrt{\theta^{(2)}}\mathbf{Q}^{(2)}, \sqrt{\gamma}\mathbf{I}]$, and $\mathbf{I} \in \mathbb{R}^{n\times n}$ is an identity matrix. Basically, it is difficult to find a global solution in (10) as it is a nonconvex minimization problem with the orthogonal constraint. In this paper, we use a gradient descent optimization procedure [52] with curvilinear search for a local optimal solution.

We first denote $\mathbf{G}$ as the gradient of (9) with respect to $\mathbf{Y}$, which can be computed as follows:

$$\mathbf{G} = 2(\mathbf{Y}\mathbf{Q} - \mathbf{F})\mathbf{Q}^T + 2\mathbf{Y}\mathbf{L}. \tag{11}$$

We then further define the skew-symmetric matrix $\mathbf{A} = \mathbf{G}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{G}$. The new trial point is determined by Crank–Nicolson-like scheme

$$\mathbf{H}(\tau) = \mathbf{Y} - \frac{\tau}{2}\mathbf{A}^T(\mathbf{Y} + \mathbf{H}(\tau)) \tag{12}$$

where τ is the step size. From (12), $\mathbf{H}(\tau)$ is given in the following closed form:

$$\mathbf{H}(\tau) = \mathbf{Y}\mathbf{M} \text{ and } \mathbf{M} = \left(\mathbf{I} - \frac{\tau}{2}\mathbf{A}^T\right)\left(\mathbf{I} + \frac{\tau}{2}\mathbf{A}^T\right)^{-1}. \tag{13}$$

Equation (13) is referred as the Cayley transformation. The iterations will end when τ satisfies the Armijo–Wolfe conditions [51]. In practical, it can further accelerated by Barzukau–Borwein step size as in [52]. The details of the curvilinear search algorithm for this subproblem are shown in Algorithm 1.

3) Optimization on $\mathbf{U}$*:* Typically the subproblem for the basis matrix $\mathbf{U}$, i.e., (5) is a least square minimization problem. By setting the derivative with respect to $\mathbf{U}$ to zero, we have a closed-form solution

$$\mathbf{U} = \mathbf{Y}\mathbf{B}^T\left(\mathbf{B}\mathbf{B}^T + \epsilon\mathbf{I}\right)^{-1} \tag{14}$$

where a small diagonal matrix $\epsilon\mathbf{I} \in \mathbb{R}^{c\times c}$ is added as the regularization to avoid overfitting.

4) Optimization on $\mathbf{B}$*:* As we see in (5), it is an NP-hard problem due to the discrete constraint on $\mathbf{B}$. Most aforementioned methods chose to first solve a relaxed problem through discarding the discrete constraints, and then threshold (or quantize) the solved continuous solution to achieve the approximate binary solution. Unfortunately, such an approximate solution is typically of low quality and often makes the resulting hash functions less effective possibly due to the accumulated quantization error. Fortunately, we next show the formulated binary code learning problem here, can be solved in a discrete optimization manner without continuous relaxation.

By dropping some terms irrelevant to $\mathbf{B}$, we can first transform (5) into the following form:

$$\begin{aligned} \min_{\mathbf{B}} \quad & -2\mathrm{Tr}\left(\mathbf{Y}^T\mathbf{U}\mathbf{B}\right) + \|\mathbf{U}\mathbf{B}\|_F^2 \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{c\times n}. \end{aligned} \tag{15}$$

Similar to the recent advance in binary optimization [12], we propose to learn the hash codes $\mathbf{B}$ by the *discrete cyclic coordinate descent* method. In other words, we learn $\mathbf{B}$ bit by bit, and each bit corresponds to one row of $\mathbf{B}$.

Let $\mathbf{b}^T \in \{-1, 1\}^{1\times n}$ denote the ith row of $\mathbf{B}$, and $\bar{\mathbf{B}} \in \{-1, 1\}^{(c-1)\times n}$ denote all other rows in $\mathbf{B}$ excluding $\mathbf{b}^T$. Similarly, let $\mathbf{u} \in \mathbb{R}^{d\times 1}$ denote the ith column of $\mathbf{U}$, and $\bar{\mathbf{U}} \in \mathbb{R}^{d\times(c-1)}$ denote all other columns in $\mathbf{U}$ excluding $\mathbf{u}$. Then, we can obtain

$$\begin{aligned} \mathrm{Tr}\left(\mathbf{Y}^T\mathbf{U}\mathbf{B}\right) &= \mathrm{Tr}\left(\mathbf{Y}^T\left(\mathbf{u}\mathbf{b}^T + \bar{\mathbf{U}}\bar{\mathbf{B}}\right)\right) \\ &= \mathrm{Tr}\left(\mathbf{Y}^T\mathbf{u}\mathbf{b}^T\right) + \text{const.} \end{aligned} \tag{16}$$

Similarly

$$\begin{aligned} \|\mathbf{U}\mathbf{B}\|_F^2 &= \left\|\mathbf{u}\mathbf{b}^T + \bar{\mathbf{U}}\bar{\mathbf{B}}\right\|_F^2 \\ &= \|\mathbf{b}\mathbf{u}^T\|_F^2 + 2\mathrm{Tr}\left(\mathbf{b}\mathbf{u}^T\bar{\mathbf{U}}\bar{\mathbf{B}}\right) + \left\|\bar{\mathbf{U}}\bar{\mathbf{B}}\right\|_F^2 \\ &= 2\mathrm{Tr}\left(\mathbf{b}\mathbf{u}^T\bar{\mathbf{U}}\bar{\mathbf{B}}\right) + \text{const.} \end{aligned} \tag{17}$$

Here, $\|\mathbf{b}\mathbf{u}^T\|_F^2 = \mathrm{Tr}(\mathbf{b}\mathbf{u}^T\mathbf{u}\mathbf{b}^T) = n\mathbf{u}^T\mathbf{u} = \text{const}$.

Substitute (16) and (17) into (15), and we obtain the following optimization problem:

$$\begin{aligned} \min_{\mathbf{b}} \quad & \left(\bar{\mathbf{B}}^T\bar{\mathbf{U}}^T\mathbf{u} - Y^T\mathbf{u}\right)^T\mathbf{b} \\ \text{s.t.} \quad & \mathbf{b} \in \{-1, 1\}^{n\times 1}. \end{aligned} \tag{18}$$

This problem has a closed-form solution

$$\mathbf{b} = \mathrm{sign}\left(\left(\mathbf{Y} - \bar{\mathbf{U}}\bar{\mathbf{B}}\right)^T\mathbf{u}\right) \tag{19}$$

where $\mathrm{sign}(\cdot)$ is the sign function.

5) Optimization on $\theta^{(i)}$*:* Although our model considers two views, here, we directly focus on solving the optimization problem for arbitrary M ($M \geq 2$) views. By discarding some terms irrelevant to $\theta^{(i)}$, we have

$$\begin{aligned} \min_{\theta^{(i)}} \quad & \sum_{i=1}^{M} \theta^{(i)}\pi(i) + \lambda\|\Theta\|_F^2 \\ \text{s.t.} \quad & \sum_{i=1}^{M} \theta^{(i)} = 1, \theta^{(i)} > 0, i = 1, \ldots, M \end{aligned} \tag{20}$$

[3] $\mathcal{M}_n^d$ represents a feasible set, which is defined as $\mathcal{M}_n^d = \{\mathbf{Y} \in \mathbb{R}^{n\times d} | \mathbf{Y}^T\mathbf{Y} = \mathbf{I}\}$.

where $\pi(i) = \|\mathbf{W}^{(i)T}\mathbf{X}^{(i)} - \mathbf{Y}^{(i)}\|_F^2$, $\Theta = [\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(M)}]^T$. In (20), we add a regularization term $\|\Theta\|_F^2$ to fully exploit the complementary information of all views. $\lambda > 0$ is a regularization parameter for controlling the smoothness of Θ. The larger λ leads to the smoother weights of views. This subproblem is a quadratic programming problem, which is efficiently solved by many optimization solvers. Note that without the regularization term, a trivial solution exists, that is, $\theta^{(i)} = 1$ corresponding to the minimum $\pi^{(i)}$ over all the M views, and $\theta^{(i)} = 0$ otherwise. In this situation, the model finally selects only one view, but ignores other views.

D. Hash Function Learning

Until now we have learned the view-specific linear mappings from original spaces to the common subspace. To obtain the hash codes, it still remains seeking a mapping from the common subspace to the Hamming space.

Here for simplicity, we also assume there exists a linear mapping between these two subspaces, whose corresponding transformation matrix $\mathbf{P} \in \mathbb{R}^{d \times c}$ can be obtained by solving the following optimization problem:

$$\min_{\mathbf{P}} \left\|\mathbf{P}^T\mathbf{Y} - \mathbf{B}\right\|_F^2 + \delta\|\mathbf{P}\|_F^2 \tag{21}$$

where δ is the regularization parameter. Clearly, this problem has the following closed form:

$$\mathbf{P} = \left(\mathbf{Y}\mathbf{Y}^T + \delta\mathbf{I}\right)^{-1}\mathbf{Y}\mathbf{B}^T. \tag{22}$$

Finally, the final hash function $\mathbf{H}^{(i)}$ of the ith view ($i = 1, 2$) is defined as

$$\mathbf{H}^{(i)}\left(\mathbf{x}^{(i)}\right) = \operatorname{sign}\left(\mathbf{P}^T\mathbf{z}\right) = \operatorname{sign}\left(\left(\mathbf{W}^{(i)}\mathbf{P}\right)^T\mathbf{x}^{(i)}\right) \tag{23}$$

where $\mathbf{z} = \mathbf{W}^{(i)T}\mathbf{x}^{(i)} \in \mathbb{R}^{d \times 1}$, $\mathbf{x}^{(i)} \in \mathbb{R}^{d_i \times 1}$ is an arbitrary sample in the i-th view.

From (23), we see the hash code can be consequently generated using a *two-stage* mechanism: given a new sample $\mathbf{x}^{(i)}$, it is first mapped as the latent feature $\mathbf{z}$ in the common subspace using the view-specific mapping $\mathbf{W}^{(i)}$, then further transformed into a c-dimensional binary code via the learned common mapping $\mathbf{P}$ defined in (22). The training procedure of SPDH is shown in Algorithm 2.

E. Convergence and Computational Complexity Analysis

We first discuss the convergence of SPDH. We have the following convergence theorem of SPDH.

Theorem 1: The alternate updating rules in Algorithm 2 monotonically decrease the objective function value of (6) in each iteration, and Algorithm 2 will converge to a local minimum of (6).

Proof: The subproblems of $\mathbf{W^{(i)}}$, $\mathbf{U}$, and $\theta^{(i)}$ are convex, thus these subproblems are obviously guaranteed to have the global minimums; although the subproblems of $\mathbf{Y}$ and $\mathbf{B}$ are not convex, the optimization for these two subproblems can decrease the objective function value. Thus the proposed optimization of each subproblem can decrease the objective function value in each iteration. In addition, according to definition of the formulation, the objective function value is lower-bounded by 0. Summarizing the above parts, we can conclude that the proposed algorithm theoretically converges to a local minimum. To this end, Theorem 1 is proved. ■

Algorithm 2 Semi-Paired Discrete Hashing (SPDH)

Input: $\mathbf{X}^{(i)} \in \mathbb{R}^{d_i \times n_i}$ ($i = 1, 2$), code length c, number of anchor data pairs m, dimensionality of the common subspace d, parameters α, γ.

Output: projection matrices $\mathbf{W}^{(i)} \in \mathbb{R}^{d_i \times d}$ ($i = 1, 2$), $\mathbf{P} \in \mathbb{R}^{d \times c}$, binary codes $\mathbf{B} \in \mathbb{R}^{c \times n}$.

```
Initialize $\mathbf{W}^{(i)}$, $\mathbf{Y}$, $\mathbf{U}$, $\mathbf{B}$;
Generate the graph $\mathbf{S}$ using (4);
repeat
    Update $\mathbf{W}^{(i)}$ ($i = 1, 2$) using (8);
    Update $\mathbf{Y}$ by calling Algorithm 1;
    Update $\mathbf{U}$ using (14);
    Update $\mathbf{B}$ bit by bit using (19);
    Update $\theta^{(i)}$ by solving (20);
until convergence
Obtain $\mathbf{P}$ according to (22);
Formulate hash function $\mathbf{H}^{(i)}$ ($i = 1, 2$) using (23).
```

Next, we analyze the computational complexity of SPDH. The computational complexity of training SPDH mainly includes the following several parts. In graph construction step, the complexity for generating $\mathbf{Z}$ is roughly $\mathcal{O}(dmns)$, where s is the number of neighbor anchors. For implementation, we do not need to explicitly compute $\mathbf{S}$, instead directly use $\mathbf{Z}$ to speed up the complexity of gradient computation. The computation complexity for updating $\mathbf{W}^{(i)}$ is $\mathcal{O}(d_i^2 n_i + d_i^3)$. In the step of updating $\mathbf{Y}$, it is very time-consuming to directly compute the gradient of $\mathbf{Y}$ via (11) for large-scale applications. In fact, $\mathbf{Q}$ is the element selection matrix, thus the calculation of the first term of (11) requires $\mathcal{O}(dn)$ via element selection; the second term of (11) is equivalent to $\mathbf{Y} - \mathbf{Y}\mathbf{Z}\mathbf{\Lambda}^{-1}\mathbf{Z}^T$, which requires $\mathcal{O}(dmn + dm^2)$. Besides, updating $\mathbf{Y}$ for each iteration is $\mathcal{O}(4d^2n + d^3)$ [52]. Thus, the complexity of optimizing $\mathbf{Y}$ is $T_1(dmn + dm^2 + 4d^2n + d^3)$, where T_1 is the number of iterations for updating $\mathbf{Y}$. Updating $\mathbf{U}$ needs the complexity of $\mathcal{O}(cdn + c^2n + c^2d + c^3)$. In the step of updating $\mathbf{B}$, the complexity for updating one bit in $\mathbf{B}$ is $\mathcal{O}(cdn)$; accordingly updating $\mathbf{B}$ needs $\mathcal{O}(T_2cd^2n)$, where T_2 (around 2∼4) is the number of iterations for updating $\mathbf{B}$. Updating $\theta^{(i)}$ requires $\mathcal{O}(M^3)$. Empirically, the outer iterations will be repeated within ten times to reach the convergence in our experiments. Finally, the computation of $\mathbf{P}$ requires the time complexity of $\mathcal{O}(d^2n + cdn)$. For the query part, the computational cost for encoding any query $\mathbf{x}^{(i)}$ is $\mathcal{O}(cd_i)$.

From the above complexity analysis, we clearly see that the training time complexity of SPDH scales linearly with size of the dataset. Consequently, it is suitable for the large-scale applications.

IV. Experiments

In this section, we evaluate the performance of the proposed SPDH for both fully-paired and semi-paired cross-view

TABLE II
STATISTICS OF FOUR BENCHMARK DATASETS

Datasets	Wiki	Pascal VOC	MIRFlickr	NUS-WIDE
Dataset Size	2,866	5,649	20,015	186,577
Training Set	2,173	2,808	18,013	184,711
DataBase Set	2,173	2,808	18,013	184,711
Query Set	693	2,841	2,002	1,866
Image Feature	128	512	150	500
Text Feature	10	399	500	1,000

retrieval tasks, and compare it with several state-of-the-art hashing methods.

A. Datasets

In this experiment, four widely used benchmark datasets, i.e., Wiki [20], Pascal VOC [53], MIRFlickr [54], and NUS-WIDE [55] are adopted for evaluation. All datasets are with views of image and text. Some statistics of them are given in Table II.

1) Wiki: It has 2866 multimedia documents, where each image is represented by a 128-D scale-invariant feature transform histogram vector and each text is represented by a 10-D latent dirichlet allocation topics vector. We use 75% of the pairs as the training set and database, the remaining 25% as the query set.

2) Pascal VOC: It consists of 9963 image tag pairs, which can be categorized into 20 different classes. Each image is represented by a 512-D Gist vector, and each text is represented by a 399-D word frequency vector. Here, images with only one object are selected in the experiment, which results in 2808 samples as the training set and database, resting 2841 as the query set [56].

3) MIRFlickr: It originally consists of 25 000 images collected from Flickr website. Each image is associated with some of 24 provided unique labels. We only keep those textual tags which have at least 20 textual tags for our experiment, and subsequently we get 20 015 points for our experiment. For each instance, the image view is represented with a 150-D edge histogram and the text view as a 500-D feature vector derived from PCA on the bag-of-words vector. We take 10% the dataset as the query set, and the rest as the training set and database.

4) NUS-WIDE: It consists of 269 648 images from 81 ground-truth concepts with a total number of 5018 unique tags. Only the top ten most frequent labels and the corresponding 186 577 annotated samples are kept. The images are represented by 500-D bag-of-visual words and tags is represented by 1000-D tag occurrence vectors. Following a literature convention [25], [29], we randomly select 1% of the dataset to form the query set, and the remaining 99% as the training set and database.

B. Experimental Setting

Two cross-view retrieval tasks are used for evaluation: use an image query in the visual view to search the relevant texts from the text view (shorted as $\mathbb{T}_{I\to T}$); use a text query in the text view to search the relevant images from the visual view (shorted as $\mathbb{T}_{T\to I}$).

We compare SPDH with various state-of-the-art CVH methods under two different experimental settings. In fully-paired experimental setting, six fully-paired CVH methods, including CVH [21], CMSSH [15], LCMH [16], CCA-ITQ[4] [7], CMFH [24], and QCH [29] are selected for comparisons. In the semi-paired experimental setting, we select two unsupervised CVH methods, i.e., IMH [23] and PM^{2}H [27], and one multiview learning method, i.e., cluster-CCA [34], all of which can deal with the semi-paired data. Source codes of CMSSH, CCA-ITQ, CMFH, QCH, and IMH are kindly provided by the authors, while other methods are implemented by ourselves as their codes are not publicly available. Note that SPDH is unsupervised, thus for a fair comparison, we do not incorporate semantic information for all comparison methods. In SPDH, there are several parameters, i.e., d, m, α, and γ. We empirically set $d = c$, and $m = 100$. α and γ are ranged from $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$, and finally chosen by cross-validation on the training dataset. For a fair comparison, parameters of all the other methods are carefully tuned according to the corresponding literatures, and their best performances are reported here.

The mean average precision (mAP), precision of top 50 samples, and precision-recall curve are adopted for evaluating the retrieval performance. mAP is the mean of all the queries' average precision (AP) in the database. For a query q, AP is defined as

$$\mathrm{AP}(q) = \frac{1}{L_q}\sum_{r=1}^{R} P_q(r)\delta_q(r) \tag{24}$$

where L_q is the number of the ground truth neighbors in the retrieved list, $P_q(r)$ is the precision of the top r retrieved results and $\delta_q(r) = 1$ if the rth result is the true neighbor and 0 otherwise. In our experiments, we set $R = 50$.

C. Performance Evaluation

1) Evaluation on Fully-Paired Data: We first evaluate the proposed method by performing cross-view retrieval tasks on fully-paired data. Six fully-paired hashing methods, i.e., CVH, CMSSH, LCMH, CCA-ITQ, CMFH, and QCH are selected for comparisons.

The mAP results with different code lengths on four benchmark datasets are reported in Table III, where we can clearly see that SPDH generally obtains the best results in the most cases. QCH and CMFH achieve comparable performances, where CMFH nearly outperforms QCH on Wiki and NUS-WIDE, and QCH outperforms CMFH on Pascal VOC and MIRFlickr. CCA-ITQ also has relatively good results, especially on MIRFlickr. Among the remaining three methods, CVH generally outperforms CMSSH and LCMH. Generally speaking, the above results clearly reveal that SPDH can achieve the promising cross-view retrieval performance on fully-paired data.

2) Evaluation on Semi-Paired Data: One advantage of SPDH is to deal with the semi-paired scenario; accordingly we further evaluate the retrieval performance of SPDH on

[4] Different from the supervised method the original paper [7], we learn binary codes of both views in the unsupervised setting.

TABLE III
COMPARISONS OF mAP WITH DIFFERENT HASH CODE LENGTHS IN THE FULLY-PAIRED SETTING ON FOUR BENCHMARK DATASETS

Task	Method	Wiki			Pascal VOC			MIRFlickr			NUS-WIDE		
		16	32	64	16	32	64	16	32	64	16	32	64
$\mathbb{T}_{I\rightarrow T}$	CVH	0.1985	0.2119	0.2174	0.1419	0.1394	0.1369	0.6462	0.6419	0.6397	0.5012	0.5108	0.4955
	CMSSH	0.1558	0.1636	0.1588	0.1380	0.1249	0.1542	0.5971	0.5690	0.5948	0.3699	0.4353	0.4346
	LCMH	0.1668	0.1898	0.1984	0.1859	0.1954	0.2148	0.5971	0.6045	0.6117	0.3953	0.4045	0.4304
	CCA-ITQ	0.2285	0.2241	0.2029	0.2705	0.2856	0.3007	0.6751	**0.6903**	0.6875	0.5385	0.5410	0.5412
	CMFH	0.2398	0.2520	0.2664	0.2722	0.3007	0.3042	0.6635	0.6666	0.6714	0.5015	0.5327	0.5407
	QCH	0.2386	0.2469	0.2411	0.3013	0.3104	0.3244	0.6602	0.6783	0.6894	0.4605	0.4582	0.4920
	SPDH	**0.2592**	**0.2612**	**0.2765**	**0.3225**	**0.3296**	**0.3428**	**0.6869**	0.6867	**0.6948**	**0.5834**	**0.6017**	**0.5930**
$\mathbb{T}_{T\rightarrow I}$	CVH	0.2446	0.2911	0.3023	0.1668	0.1559	0.1673	0.6523	0.6527	0.6498	0.5680	0.5653	0.5487
	CMSSH	0.2110	0.1852	0.1602	0.1662	0.1192	0.1615	0.5607	0.6027	0.6217	0.3485	0.4305	0.4331
	LCMH	0.2071	0.2458	0.2984	0.2360	0.2807	0.3539	0.5816	0.5959	0.6025	0.3893	0.3912	0.3971
	CCA-ITQ	0.3204	0.3207	0.3234	0.2740	0.2922	0.3389	0.6797	0.6791	0.6807	0.4365	0.4831	0.5175
	CMFH	0.3677	0.4317	0.4319	0.5054	0.5399	0.5410	0.6589	0.6576	0.6557	0.5513	0.5434	0.5568
	QCH	0.3445	0.3532	0.3640	0.3148	0.4038	0.4275	0.6695	0.6735	0.6736	0.4633	0.4810	0.4932
	SPDH	**0.4209**	**0.4404**	**0.4584**	**0.6517**	**0.6643**	**0.6791**	**0.6819**	**0.6896**	**0.6865**	**0.6121**	**0.5916**	**0.5981**

TABLE IV
COMPARISONS OF mAP WITH DIFFERENT CODE LENGTHS IN THE SEMI-PAIRED SETTING OF 10% AVAILABLE PAIRWISE INFORMATION ON FOUR BENCHMARK DATASETS

Task	Method	Wiki			Pascal VOC			MIRFlickr			NUS-WIDE		
		16	32	64	16	32	64	16	32	64	16	32	64
$\mathbb{T}_{I\rightarrow T}$	Cluster-CCA	0.1870	0.1912	0.1738	0.1651	0.1424	0.1293	0.6194	0.5947	0.6105	0.4838	0.4235	0.4570
	IMH	0.1900	0.1940	0.2020	0.1616	0.1518	0.1544	0.6271	0.6199	0.6349	0.4663	0.4794	0.4904
	PM^{2}H	0.1888	0.1926	0.1967	0.1699	0.1589	0.1510	0.6203	0.6252	0.5925	0.4973	0.4503	0.4649
	SPDH	**0.2057**	**0.2073**	**0.2061**	**0.1759**	**0.1655**	**0.1645**	**0.6432**	**0.6409**	**0.6402**	**0.5382**	**0.5245**	**0.5268**
$\mathbb{T}_{T\rightarrow I}$	Cluster-CCA	0.2489	0.2601	0.2136	0.1780	0.1517	0.1681	0.6029	0.6003	0.6110	0.4133	0.4091	0.4271
	IMH	0.2912	0.3400	0.3761	0.2059	0.2002	0.2085	0.6309	0.6282	0.6291	0.5117	0.5313	0.5274
	PM^{2}H	0.2617	0.3255	0.3429	0.2021	0.2201	0.2184	0.6031	0.5990	0.5925	0.4396	0.4364	0.4032
	SPDH	**0.3689**	**0.4022**	**0.4107**	**0.2127**	**0.2259**	**0.2247**	**0.6797**	**0.6430**	**0.6342**	**0.5533**	**0.5446**	**0.5479**

semi-paired data. We compare SPDH with three semi-paired cross-view methods, i.e., cluster-CCA, IMH, and PM^{2}H. For cluster-CCA, the clusters are obtained according to the similarities between all the samples and the available anchor pairs. Then, we apply cluster-CCA to learn the linear projections, and further get the orthogonal rotation matrix by ITQ to minimize the quantization loss. It is infeasible to train IMH on the whole NUS-WIDE dataset, thus we select a subset of 20 000 samples for training IMH.

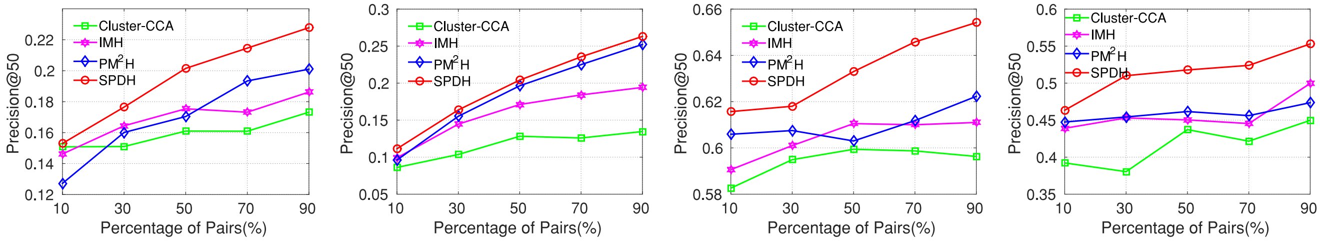
We report the mAP results with fixed 10% pairwise information in Table IV. The results show that SPDH generally outperform other methods by different degrees in different cases. IMH also obtains good performances, which is followed by PM^{2}H. Cluster-CCA is worst among these methods. Besides, we vary the percentage of pairwise information from 10% to 90%, and report the precision of the top 50 samples with 32 bit code length in two cross-view retrieval tasks, as shown in Figs. 3 and 4. From these results, we can see that all the methods improve the retrieval performance as the percentage of pairwise information increases. SPDH consistently outperforms other methods in the most cases. IMH outperforms PM^{2}H on MIRFlickr and NUS-WIDE; PM^{2}H outperforms IMH in the case of large percentage of pairwise information of the other two datasets. In addition, the precision-recall curves with 90% pairwise information on 32 bit code length are also reported in Figs. 5 and 6, where we can see that the precision-recall curves of SPDH are above those of the comparison methods. Furthermore, we show some retrieval examples of SPDH and the three comparisons on Wiki dataset in the supplementary. Please refer to the supplementary materials for details. Generally speaking, the above results clearly demonstrate that SPDH can handle the semi-paired cross-view retrieval task very well.

D. Evaluation on the Proposed Coding Scheme

To demonstrate the effectiveness of the proposed discrete coding scheme in SPDH, we report the performances of two other coding schemes, i.e., sign and ITQ as comparisons. For these two comparisons, we first learn $\mathbf{W}^{(i)}$ $(i = 1, 2)$, then further use sign function, or ITQ to obtain the hash codes. We simply name two comparison methods: 1) SPH-Sign and 2) SPH-ITQ. Fig. 7 shows the mAP comparisons with 10% pairwise information on four benchmark datasets. From Fig. 7, we see that SPH-ITQ obtains better performances than the SPH-Sign, since SPH-ITQ utilizes the additional rotation matrix to reduce the quantization errors. SPDH obviously has the best performances in most cases, which reveals the superiority of the proposed coding scheme. Another superiority of SPDH is to jointly learn latent continuous feature and binary codes within one framework, while two comparison methods ignore the correlation between these two parts.

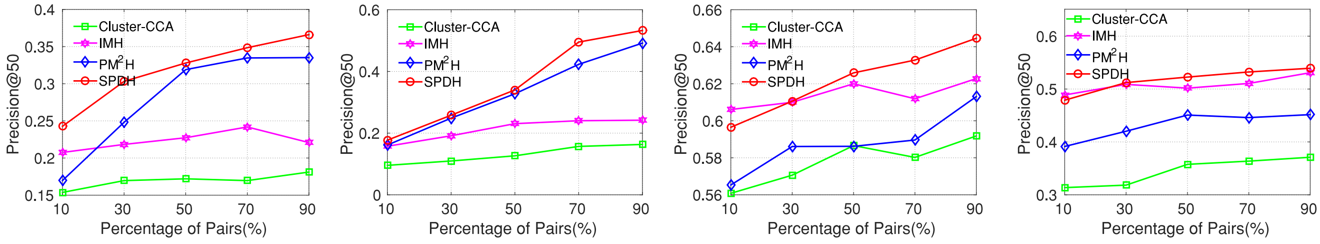
E. Evaluation on the Proposed Discrete Optimization

To further demonstrate the effectiveness of the discrete optimization manner in SPDH, we compare it with a relaxed manner, which optimizes the hash codes in (5) by directly



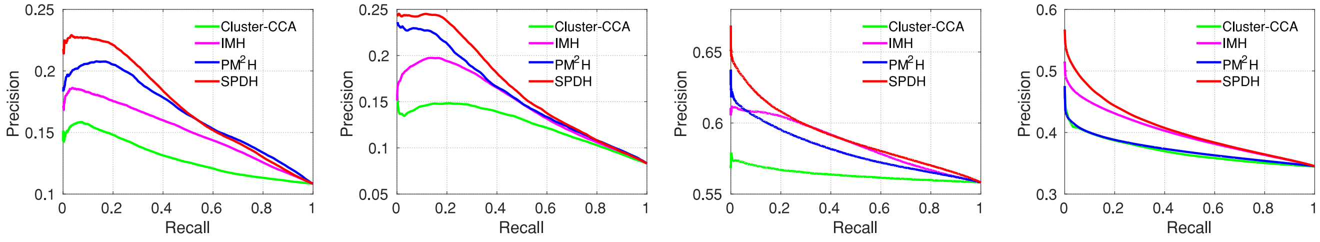
(a) (b) (c) (d)

Fig. 3. Comparisons of $\mathbb{T}_{I\rightarrow T}$ with different percentages of pairwise information on (a) Wiki, (b) Pascal VOC, (c) MIRFlickr, and (d) NUS-WIDE.



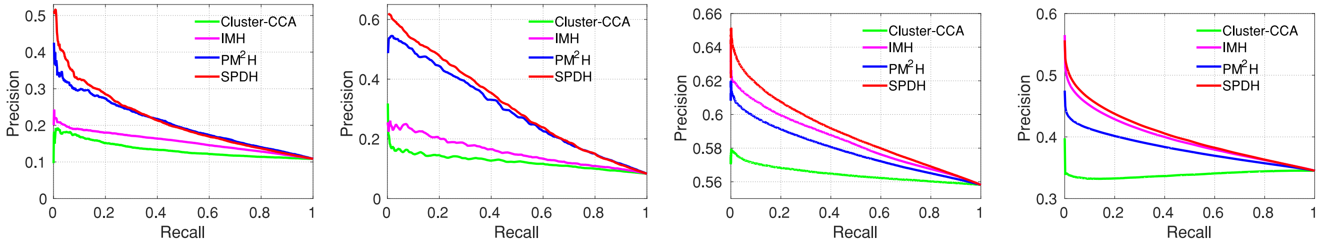
(a) (b) (c) (d)

Fig. 4. Comparisons of $\mathbb{T}_{T\rightarrow I}$ with different percentages of pairwise information on (a) Wiki, (b) Pascal VOC, (c) MIRFlickr, and (d) NUS-WIDE.



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

Fig. 5. Precision-recall curves of $\mathbb{T}_{I\rightarrow T}$ on (a) Wiki, (b) Pascal VOC, (c) MIRFlickr, and (d) NUS-WIDE.



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

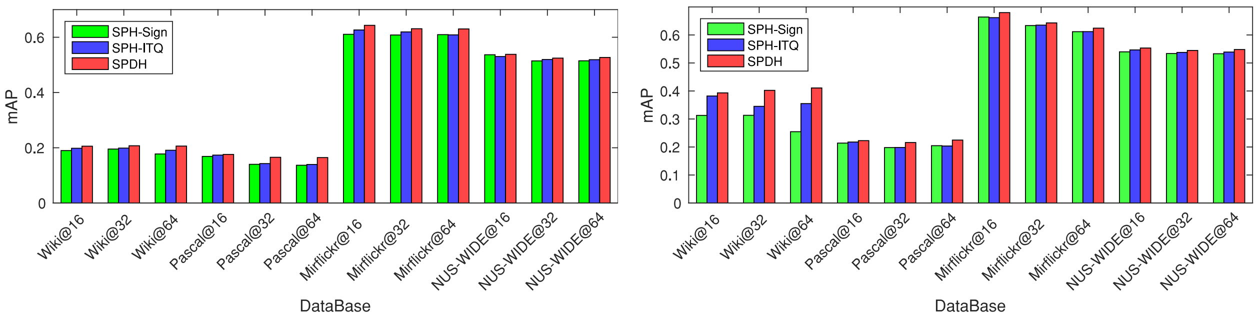
Fig. 6. Precision-recall curves of $\mathbb{T}_{T\rightarrow I}$ on (a) Wiki, (b) Pascal VOC, (c) MIRFlickr, and (d) NUS-WIDE.

discarding the discrete constraints. It leads to a least square minimization problem with a close-form solution of **B**. We report the mAP comparisons between these two optimization manners in Fig. 8. From Fig. 8, we can see that our proposed discrete optimization generally obtains better performances than the relaxed one. It demonstrates that the proposed discrete optimization can yield better-quality hash codes, while the conventional relaxed optimization will inevitably incur the quantization errors without consideration of discrete nature.

F. Efficiency and Convergence Analysis

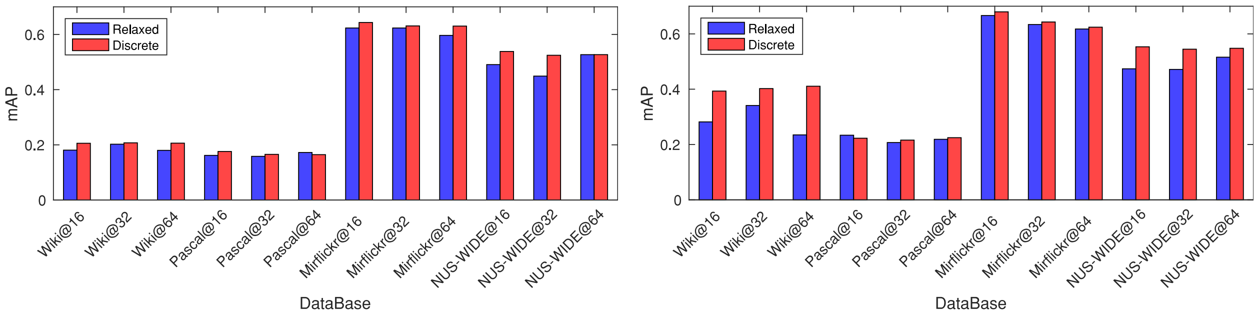
The algorithms are developed in MATLAB version R2015a. All the computations reported in this paper are performed on a Red Hat Enterprise 64-Bit Linux workstation with 12-core Intel Xeon CPU X5690 3.47 GHz and 96 GB memory. We conduct the comparisons of computation efficiency. Table V lists the training time of all the hashing methods with 16 bit code length. Here, we do not report the search time, since they are very similar among all the methods. For NUS-WIDE dataset, the training size is further varied in a wide range from 5000 to 184 711.

From Table V, we see that CMSSH and CCA-ITQ are the most efficient, followed by CVH. LCMH and CMFH are similar in computation efficiency. QCH is the slowest among the fully-paired hashing methods. In the semi-paired category, IMH is the most time consuming on the large-scale



(a) (b)

Fig. 7. Comparisons between the proposed coding scheme and two conventional ones on two different retrieval tasks. (a) $\mathbb{T}_{I\to T}$. (b) $\mathbb{T}_{T\to I}$.



(a) (b)

Fig. 8. Comparisons between the proposed discrete optimization and a relaxed one on two different retrieval tasks. (a) $\mathbb{T}_{I\to T}$. (b) $\mathbb{T}_{T\to I}$.

TABLE V

Comparison of Training Time (in Seconds) on Four Benchmark Datasets. ("-" Denotes the Unknown Computation Time, Because the Training Cannot be Performed in This Case)

Dataset	Wiki	Pascal VOC	MIRFlickr	NUS-WIDE						
Dim of Image/Text	128/10	512/399	150/500	500/1000						
Training Size	2,173	2,808	18,013	5,000	10,000	20,000	30,000	50,000	100,000	184,711
CVH	0.50	4.96	1.09	18.29	19.27	24.00	25.13	28.92	18.83	30.99
CMSSH	0.04	0.20	0.41	0.49	0.63	1.22	1.59	2.44	4.43	8.43
LCMH	0.77	1.12	6.07	1.80	4.83	12.93	24.36	36.14	85.24	1.58×10^2
CCA-ITQ	0.41	0.42	0.83	1.33	1.57	2.20	2.88	4.17	7.15	11.65
CMFH	0.22	1.89	6.66	5.57	9.91	18.53	27.31	44.40	84.98	1.53×10^2
QCH	1.40	7.47	25.42	22.48	36.81	61.53	95.72	1.73×10^2	2.81×10^2	7.86×10^2
IMH	4.31	5.20	9.33×10^2	46.39	3.15×10^2	2.30×10^3	7.62×10^3	-	-	-
PM^{2}H	9.76	41.77	1.82×10^2	48.87	1.02×10^2	2.50×10^2	3.58×10^2	5.20×10^2	1.04×10^3	1.93×10^3
SPDH	2.42	3.69	17.58	7.95	14.08	26.60	37.21	65.36	1.17×10^2	2.13×10^2

dataset. Due to the large memory cost, IMH fails to learn on the NUS-WIDE dataset with more than 50 000 samples on our workstation. Compared with IMH and PM^{2}H is relatively fast on the large-scale dataset. SPDH is obviously more efficient than IMH and PM^{2}H. For example, the training time of SPDH is nearly ten times faster than that of PM^{2}H, and 200 times faster than that of IMH on NUS-WIDE with 30 000 training samples. The above results demonstrate that the proposed SPDH is more efficient and scalable than two other semi-paired hashing methods on the large-scale retrieval task.

Furthermore, we analyze the convergence of the proposed SPDH. Fig. 9 shows the convergence curves of SPDH on four benchmark datasets. As we can see clearly in Fig. 9, SPDH quickly converges within around ten iterations.

G. Parameter Analysis

We empirically analyze the sensitivity of three parameters in the proposed SPDH, i.e., the number of anchor pairs m and trading-off parameters α and γ. To reveal their effects on the performance, we report the mAP of the varying parameters with the fixed 16 code length and 10% pairwise information. We evaluate one parameter while fixing the other parameters. In the experiment, m is ranged from [10, 50, 100, 200, 500, 1000, 2000], α and γ are varied from the range of $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$.

Fig. 10 shows mAP results with varying parameters on two tasks, i.e., $\mathbb{T}_{I\to T}$, $\mathbb{T}_{T\to I}$. From Fig. 10, we see that with the increase of α, mAP first maintains or slightly improves, and then drops with different degrees on different datasets. Similar phenomenon can also be observed from the change of γ. The above results indicate that the similarity term

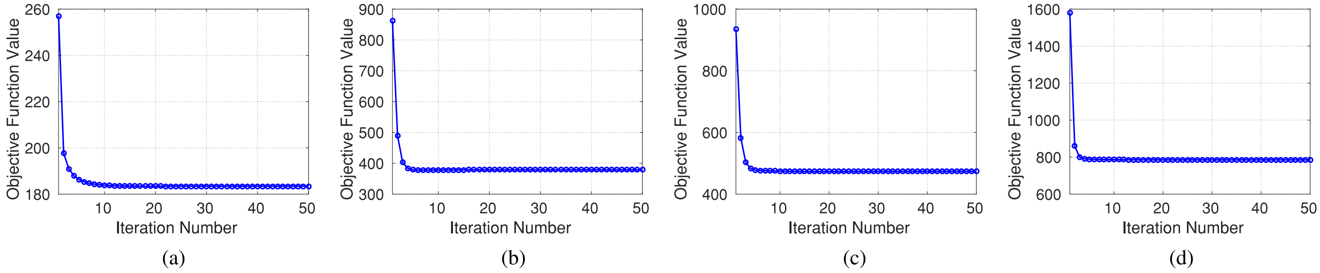


Fig. 9. Convergence analysis of the proposed SPDH on (a) Wiki, (b) Pascal VOC, (c) MIRFlickr, and (d) NUS-WIDE.

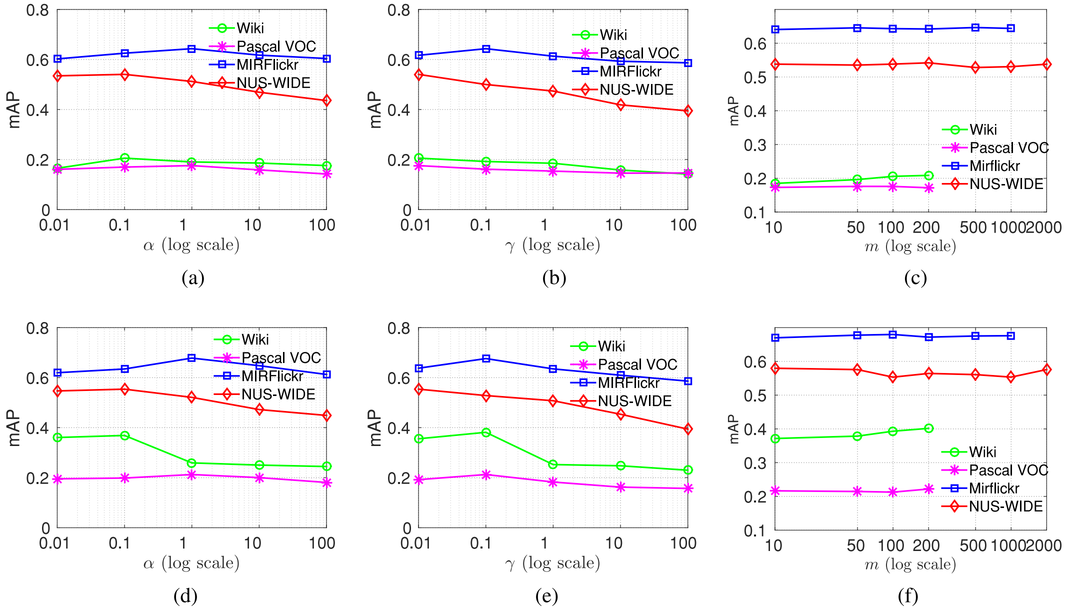


Fig. 10. Parameters analysis of the proposed SPDH on (a)–(c) $\mathbb{T}_{I\rightarrow T}$ and (d)–(f) $\mathbb{T}_{T\rightarrow I}$. (a) and (d) mAP on varying α. (b) and (e) mAP on varying γ. (c) and (f) mAP on varying m.

and hash code reconstruction term can help to improve the performance. However, if α or γ are set to be large, the balance of each term will be affected, leading to relatively poor results. Empirically, the superior performance can be obtained when $\alpha, \gamma \in [0.01, 1]$. Besides, it can be found that mAP is generally not sensitive to the change of m, which implies the robustness of the proposed cross-view graph.

V. Conclusion

In this paper, we studied a challenging but less explored problem in hashing research, i.e., *semi-paired cross-view retrieval*. A novel hashing method termed SPDH was proposed to handle this task. SPDH well aligned both paired and unpaired samples in the common latent subspace by exploring similarities of the semi-paired data via a new cross-view graph. A factorization-based hash coding scheme was further presented to embed the latent features into target hash codes as semantic discrete representations. The hash codes were discretely optimized in a bit-by-bit manner with each hash bit generated with a closed-form solution. SPDH was validated on four benchmark datasets, and it yielded the promising accuracy and scalability.

There are several interesting works that deserve further studies based on our model. First, our model currently considers the similarity (local) structure of the data, the low-rank (global) structure can also be explored to further align the semi-paired data. The more robust $\ell_{2,p}$-norm-based loss terms [57] can be incorporated to effectively control different levels of noises. Second, currently this paper focuses on the two-view case; the multiview (more than two) extension of our model can be developed to learn hash codes on the more complex semi-paired data with arbitrary views. Applying this extension to the general multiview semi-paired retrieval task is an interesting work. Third, our model only uses a linear projection to generate hash codes, deep learning can be employed in our model to discover the nonlinear data structure, obtaining more compact hash codes.

References

[1] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," *arXiv preprint arXiv:1408.2927*, 2014.

[2] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2008, pp. 1753–1760.

[3] Z. Tang, X. Zhang, and S. Zhang, "Robust perceptual image hashing based on ring partition and NMF," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 711–724, Mar. 2014.

[4] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 200–214, Jan. 2016.

[5] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Data Bases*, Edinburgh, U.K., 1999, pp. 518–529.

[6] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 1–8.

[7] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[8] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, "Discrete graph hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 3419–3427.

[9] L. Chen, D. Xu, I. W.-H. Tsang, and X. Li, "Spectral embedded hashing for scalable image retrieval," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1180–1190, Jul. 2014.

[10] Z. Jin, C. Li, Y. Lin, and D. Cai, "Density sensitive hashing," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1362–1371, Aug. 2014.

[11] X. Liu, Y. Mu, D. Zhang, B. Lang, and X. Li, "Large-scale unsupervised hashing with shared structure learning," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1811–1822, Sep. 2015.

[12] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 37–45.

[13] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *CoRR*, vol. abs/1304.5634, 2013.

[14] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[15] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 3594–3601.

[16] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao, "Linear cross-modal hashing for efficient multimedia search," in *Proc. ACM Int. Conf. Multimedia*, Barcelona, Spain, 2013, pp. 143–152.

[17] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proc. ACM Int. Conf. Multimedia*, Scottsdale, AZ, USA, 2011, pp. 423–432.

[18] S. Kim, Y. Kang, and S. Choi, "Sequential spectral learning to hash with multiple representations," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, 2012, pp. 538–551.

[19] L. Liu, M. Yu, and L. Shao, "Multiview alignment hashing for efficient image search," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 956–966, Mar. 2015.

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

[21] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, 2011, pp. 1360–1365.

[22] Y. Zhen and D.-Y. Yeung, "Co-regularized hashing for multimodal data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1385–1393.

[23] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, New York, NY, USA, 2013, pp. 785–796.

[24] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 2083–2090.

[25] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2177–2183.

[26] Y. Hu, Z. Jin, H. Ren, D. Cai, and X. He, "Iterative multi-view hashing for cross media indexing," in *Proc. ACM Int. Conf. Multimedia*, Orlando, FL, USA, 2014, pp. 527–536.

[27] Q. Wang, L. Si, and B. Shen, "Learning to hash on partial multi-modal data," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3904–3910.

[28] D. Wang, X. Gao, X. Wang, and L. He, "Semantic topic multimodal hashing for cross-media retrieval," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3890–3896.

[29] B. Wu, Q. Yang, W.-S. Zheng, Y. Wang, and J. Wang, "Quantized correlation hashing for fast cross-modal search," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3946–3952.

[30] Y. Zhen, Y. Gao, D.-Y. Yeung, H. Zha, and X. Li, "Spectral multimodal hashing and its application to multimedia retrieval," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 27–38, Jan. 2016.

[31] X. Shen, F. Shen, Q.-S. Sun, Y.-H. Yuan, and H. T. Shen, "Robust cross-view hashing for multimedia retrieval," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 893–897, Jun. 2016.

[32] C. H. Lampert and O. Krömer, "Weakly-paired maximum covariance analysis for multimodal dimensionality reduction and transfer learning," in *Proc. Eur. Conf. Comput. Vis.*, Crete, Greece, 2010, pp. 566–579.

[33] S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Partial multi-view clustering," in *Proc. AAAI Conf. Artif. Intell.*, Quebec City, QC, Canada, 2014, pp. 1968–1974.

[34] N. Rasiwasia, D. Mahajan, V. Mahadevan, and G. Aggarwal, "Cluster canonical correlation analysis," in *Proc. Int. Conf. Artif. Intell. Stat.*, Reykjavík, Iceland, 2014, pp. 823–831.

[35] C. Kang, S. Xiang, S. Liao, C. Xu, and C. Pan, "Learning consistent feature representation for cross-modal multimedia retrieval," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 370–381, Mar. 2015.

[36] J. Ji, J. Li, S. Yan, B. Zhang, and Q. Tian, "Super-bit locality-sensitive hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 108–116.

[37] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1092–1104, Jun. 2012.

[38] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14. Vancouver, BC, Canada, 2001, pp. 585–591.

[39] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[40] F. Shen *et al.*, "Hashing on nonlinear manifolds," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1839–1851, Jun. 2015.

[41] F. Shen, W. Liu, S. Zhang, Y. Yang, and H. T. Shen, "Learning binary codes for maximum inner product search," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 4148–4156.

[42] X. Liu, B. Du, C. Deng, M. Liu, and B. Lang, "Structure sensitive hashing with adaptive product quantization," *IEEE Trans. Cybern.*, to be published.

[43] R. Ye and X. Li, "Compact structure hashing via sparse and similarity preserving embedding," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 718–729, Mar. 2016.

[44] D. Zhang, F. Wang, and L. Si, "Composite hashing with multiple information sources," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Beijing, China, 2011, pp. 225–234.

[45] X. Liu, J. He, D. Liu, and B. Lang, "Compact kernel hashing with multiple features," in *Proc. ACM Int. Conf. Multimedia*, Nara, Japan, 2012, pp. 881–884.

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

[47] D. R. Hardoon, S. Szedmák, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.

[48] C. Dhanjal, S. R. Gunn, and J. Shawe-Taylor, "Efficient sparse kernel feature extraction based on partial least squares," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1347–1361, Aug. 2009.

[49] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1083–1094, May 2015.

[50] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Las Vegas, NV, USA, 2008, pp. 650–658.

[51] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.

[52] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Math. Program.*, vol. 142, nos. 1–2, pp. 397–434, 2013.

[53] S. J. Hwang and K. Grauman, "Reading between the lines: Object localization using implicit cues from image tags," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1145–1158, Jun. 2012.

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

[55] T.-S. Chua *et al.*, "NUS-WIDE: Areal-world Web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retrieval*, Santorini, Greece, 2009, Art. no. 48.

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

[57] Y. Yang, Z.-J. Zha, Y. Gao, X. Zhu, and T.-S. Chua, "Exploiting Web images for semantic video indexing via robust sample-specific loss," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1677–1689, Oct. 2014.

Xiaobo Shen received the bachelor's degree in software engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2011, where he is currently pursuing the Ph.D. degree.

In 2015, he visited the School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, QLD, Australia, for one year. His current research interests include machine learning and computer vision.

Fumin Shen received the bachelor's degree from Shandong University, Jinan, China, in 2007, and the Ph.D. degree from the Nanjing University of Science and Technology, Nanjing, China, in 2014.

He is currently a Lecturer with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include computer vision and machine learning, including face recognition, image analysis, and hashing methods.

Quan-Sen Sun received the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2006.

He is a Professor with the Department of Computer Science, NUST. He visited the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2004 and 2005, respectively. He has published over 80 scientific papers. His current research interests include pattern recognition, image processing, remote sensing information system, and medicine image analysis.

Yang Yang received the bachelor's degree from Peking University, Beijing, China, in 2006, the master's degree from Jilin University, Changchun, China, in 2009, and the Ph.D. degree from the University of Queensland, Brisbane, QLD, Australia, in 2012, under the supervision of Prof. H. T. Shen and Prof. X. Zhou.

He is currently with the University of Electronic Science and Technology of China, Chengdu, China. He was a Research Fellow with the National University of Singapore, Singapore, from 2012 to 2014, under the supervision of Prof. T.-S. Chua.

Yun-Hao Yuan received the M.Eng. degree in computer science and technology from Yangzhou University, Yangzhou, China, in 2009, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2013.

He is an Associate Professor with the Department of Computer Science and Technology, Yangzhou University. He has published over ten scientific papers. His current research interests include pattern recognition, image processing, computer vision, and information fusion.

Dr. Yuan is currently a member of the International Society of Information Fusion.

Heng Tao Shen (M'09–SM'10) received the B.Sc. (with first class Hons.) and Ph.D. degrees from the Department of Computer Science, National University of Singapore, Singapore, in 2000 and 2004, respectively.

He joined the University of Queensland, Brisbane, QLD, Australia, as a Lecturer and became a Professor in 2011. He is a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, and the School of Information Technology and Electrical Engineering, University of Queensland. His current research interests include multimedia/mobile/Web search and big data management.

Dr. Shen was a recipient of the Chris Wallace Award for Outstanding Research Contribution in 2010 from CORE Australasia. He is an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and serves as the PC Co-Chair for ACM Multimedia 2015.