Flexible Cross-Modal Hashing

Item Type	Article
Authors	Yu, Guoxian;Liu, Xuanwu;Wang, Jun;Domeniconi, Carlotta;Zhang, Xiangliang
Citation	Yu, G., Liu, X., Wang, J., Domeniconi, C., & Zhang, X. (2020). Flexible Cross-Modal Hashing. IEEE Transactions on Neural Networks and Learning Systems, 1–11. doi:10.1109/tnnls.2020.3027729
Eprint version	Post-print
DOI	10.1109/tnnls.2020.3027729
Publisher	Institute of Electrical and Electronics Engineers (IEEE)
Journal	IEEE Transactions on Neural Networks and Learning Systems
Rights	(c) 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.
Download date	2024-03-29 12:32:14
Link to Item	http://hdl.handle.net/10754/665599

Flexible Cross-Modal Hashing

Guoxian Yu, Xuanwu Liu, Jun Wang, Carlotta Domeniconi, Xiangliang Zhang

***Abstract*—Hashing has been widely adopted for large-scale data retrieval in many domains, due to its low storage cost and high retrieval speed. Existing cross-modal hashing methods optimistically assume that the *correspondence* between training samples across modalities is readily available. This assumption is unrealistic in practical applications. In addition, existing methods generally require the *same* number of samples across different modalities, which restricts their flexibility.**

We propose a flexible cross-modal hashing approach (FlexCMH) to learn effective hashing codes from weakly-paired data, whose correspondence across modalities is partially (or even totally) unknown. FlexCMH first introduces a clustering-based matching strategy to explore the structure of each cluster, and thus to find the potential correspondence between clusters (and samples therein) across modalities. To reduce the impact of an incomplete correspondence, it jointly optimizes the potential correspondence, the cross-modal hashing functions derived from the correspondence, and a hashing quantitative loss in a unified objective function. An alternative optimization technique is also proposed to coordinate the correspondence and hash functions, and to reinforce the reciprocal effects of the two objectives. Experiments on public multi-modal datasets show that FlexCMH achieves significantly better results than state-of-the-art methods, and it indeed offers a high degree of flexibility for practical cross-modal hashing tasks.

Keywords: Cross modal hashing, weakly-paired data, flexibility, clustering-based match, optimization

I. INTRODUCTION

Hashing has sparked increasing interest from both research and industry, due to its low storage cost and high retrieval speed with big data [41], [43], [31], [28], [9], [11], [8], [4]. Hashing aims at compressing high-dimensional vectorial data into a short binary code, while preserving the structure, to facilitate efficient retrieval with significantly reduced storage needs. By using the index constructed from the hash code, big data retrieval can be achieved in constant or sub-linear time [19], [32], [14], [23], [4].

With the wide range of applications of Internet of Things, rapid influxes of multi-modal data ask for efficient cross-modal hashing solutions. For example, given an image/video about a historic event, one may want to cross-modally retrieve texts describing the details of the event. As such, how to perform cross-modal hashing on widely-witnessed multi-modal data becomes a topic of interest in hashing [17], [46], [41], [43], [8] [45], [14]. Existing cross-modal hashing solutions can be classified into unsupervised or supervised. Unsupervised methods seek hash coding functions by taking into account the underlying data structure, distributions, or topological information [38], [2]. Supervised (semi-supervised) approaches leverage supervised information (i.e., semantic labels) to improve the performance [46], [20], [10], [40], [3], [11], [14].

Existing cross-modal hashing methods optimistically assume that the *correspondence* between samples of different modalities is known [16]. However, in real applications, some objects are only available in one modality, or their corresponding (or paired) objects in another modality are only partially (or even completely) unknown. This can happen, for example, when one wants to search images from texts, and there are 100 images and 200 documents, but only the correspondence between 50 images and 80 documents is known. In other words, the image-text collection is *weakly-paired*, and only the semantic labels are shared across modalities. To the best of our knowledge, how to flexibly learn a hash code from weakly-paired data is still a *challenging and open* problem in cross-modal hashing.

Several cross-modal hashing methods have been recently introduced to tackle weakly-paired multi-modal data [38], [27], [35]. They typically project multi-modality data onto a latent space under the guidance of paired samples, and then seek new pairwise mappings between samples and hash functions in this space. As such, they require enough paired data and the same number of training samples across modalities. Few multi-view learning approaches were also proposed to handle the general weakly-paired samples [18], [22], [49], [29]. These approaches also require the same number of training samples across different views, or the same number of samples for matched classes. However, these requirements are violated in many cases, where samples across different modalities are *partially-paired*, or even *completely-unpaired*, and the numbers of member samples of matched clusters (or classes) across modalities are *not the same*.

In this paper, we propose a Flexible Cross-Modal Hashing (FlexCMH) solution (as illustrated in Fig. 1) to handle partially-paired (and even completely unpaired) multi-modal data. Our main contributions are summarized as follows:

(i) We design a novel matching strategy that uses centroids of clusters, the local structure of centroids, and an incomplete correspondence between samples to seek a matching between samples in different modalities. This strategy neither requires the same number of samples within the matched clusters, nor across different modalities. Therefore, FlexCMH can be applied with flexibility in general cross-modal hashing settings.

(ii) We propose a unified objective function to simultaneous-

---

Manuscript received July 11, 2019; Revised November 30, 2019; April 15, 2020; July 22, 2020; and September 25, 2020. This work was supported by Natural Science Foundation of China (61872300, 62031003 and 62072380), and Qilu Scholar Startup Fund of Shandong University. *(Corresponding author: Jun Wang.)*

G. Yu and J. Wang are with School of Software, Joint SDU-NTU Centre for Artificial Intelligence Research, Shandong University, Jinan 250101, China. Email: guoxian85@gmail.com, kingjun@sdu.edu.cn

X. Liu is with the Alibaba Group, Hangzhou 310000, China. Email: xuanwu.lxw@alibaba-inc.com

C. Domeniconi is with the Department of Computer Science, George Mason University, VA 22030, USA. Email: carlotta@cs.gmu.edu

G. Yu and X. Zhang are with Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955, SA. Email: xiangliang.zhang@kaust.edu.sa.

ly consider the cross-modal matching loss, the intra-modal representation loss, and the quantitative loss to learn adaptive hashing codes. We also introduce an alternative optimization technique to jointly optimize the match and hash functions in a reciprocal boosting fashion.

(iii) Experiments on benchmark multi-modal datasets show that FlexCMH significantly outperforms related and representative cross-modal hashing approaches [2], [46], [20], [18], [22], [27], [35] [25], [47] in weakly-paired scenarios, and it holds a competitive performance in different open settings.

The rest of this paper is organized as follows. Section II gives a brief review of related work. Section III introduces the objective function of FlexCMH and its optimization. Section IV presents the experimental setup, results, and analysis. Finally, Section V draws conclusions and provides directions for future work.

II. RELATED WORK

Like single-modal hashing methods which are based on a structure preserving criterion, existing cross-modal hashing can be categorized into three types: pairwise [46], [20], [33], [28], [19], [32], [9], [23], [8], [14], [4], multi-wise [37], [42], [26], and implicit similarity preserving [15], [34]. Semantic correlation maximization (SCM) [46] optimizes the hashing functions by maximizing the correlation between two modalities with respect to the pairwise semantic similarity. Semantics Preserving Hashing (SePH) [20] projects the corresponding features of any instance for each modality into unified binary hash codes based on the semantic consistency between views. Collective Matrix Factorization Hashing (CMFH) [7] learns unified binary codes using collective matrix factorization with a latent factor model on multi-modal data. Efficient Discrete Latent Semantic Hashing (DLSH) [28] simultaneously discovers the latent shared space of heterogeneous multi-modal data and enhances the discriminative capability of hash codes with explicit semantic labels. Fast Discrete Cross-modal Hashing (FDCH) [25] regresses semantic labels to corresponding hashing codes with a drift to improve the cross-modal retrieval performance. Deep Cross-Modal Hashing (DCMH) [13] combines hashing learning and deep feature learning by preserving the semantic similarity between modalities. Ranking-based Deep Cross-modal Hashing (RDCMH) [26] preserves the multi-level semantic similarity order between labeled and unlabeled multi-label samples for cross-modal hashing. Deep semantic-preserving ordinal hashing (DSPOH) [14] adopts deep neural networks to learn hashing functions by exploring the ranking structure. Cross-modal similarity sensitive hashing (CMSSH) [2], as a representative implicit similarity preserving-based method, models the projection of features in each view to hash codes as binary classification problems with positive and negative examples, and utilizes a boosting method to efficiently learn the hash functions. However, these methods all assume the training samples are completely-paired between modalities, and cannot be applied to weakly-paired samples.

Several cross-modal hashing approaches have been proposed to handle the weakly-paired data. For example, Inter-Media Hashing (IMH) maps view-specific features onto a common Hamming space by learning linear hash functions with intra-modal and inter-modal consistencies [38]. Semi-Paired Discrete Hashing (SPDH) aligns both paired and unpaired samples in a common latent subspace by successfully exploring the similarity between samples via a cross-view graph [35]. Generalized Semantic Preserving Hashing (GSPH) [5] proposes a simple hashing framework that can work with different settings, like single-label, multi-label, and both paired and unpaired data, while effectively capturing the semantic relationship between samples. Triplet Fusion Network Hashing (TFNH) [12] designs a triplet network to handle both paired and unpaired data, and to narrow the gap between the modalities by two modality-based classifiers. Generalized Semi-supervised and Structured Subspace Learning (GSS-SL) [47] proposes a label graph constraint to ensure the intrinsic geometric structures of different feature spaces are consistent with the structures of the label space. Composite Correlation Quantization (CCQ) jointly finds correlation-maximal mappings that transform different modalities into an isomorphic latent space, and learns composite quantizations that convert the isomorphic latent features into compact binary codes [27]. Furthermore, several multi-view learning solutions were also introduced to tackle weakly-paired samples. Weakly-paired Maximum Correlation Analysis (WMCA) extends the maximum covariance analysis to the weakly-paired case by jointly learning the latent pairs and subspace for dimensionality reduction and transfer learning [18]. Multi-Modal Projection Dictionary Learning (MMPDL) jointly learns the projective dictionary and pairing matrix for the fusion classification [22]. Mandal *et al.* [29] learnt coupled dictionaries from the respective data views and sparse representation coefficients with respect to their own dictionaries. They then maximized the correlation between sample coefficients of the same class, and simultaneously minimized the correlation of different classes to seek the matching between samples and to fuse weakly-paired multi-view data. However, these solutions lack flexibility in one or multiple ways. They either require enough paired samples across modalities (or views); they need the same number of training samples across modalities [18], [27], [35], [22], [29]; or they isolate the matching exploration from the follow-up hashing learning [49].

To address these standing issues, we propose a Flexible Cross-Modal Hashing (FlexCMH) to handle weakly-paired multi-modal data. Specifically, FlexCMH introduces a clustering-based matching strategy to explore the potential correspondence between clusters and their member samples. In addition, it jointly optimizes this correspondence, cross-modal hashing functions derived from it, and the hashing quantitative loss in a unified objective function to simultaneously learn a compact hashing code. The workflow of FlexCMH is shown in Fig. 1.

III. PROPOSED METHOD

A. Problem Definition

Let M be the number of modalities, and the number of training samples for the m-th modality is N_m. $\mathbf{X}^m \in \mathbb{R}^{d_m \times N_m}$ represents the data matrix for the m-th modality, where both N_m and d_m are modal-dependent. For example, in a two-modality Wiki-image search application, $\mathbf{x}_i^1$ is the image feature

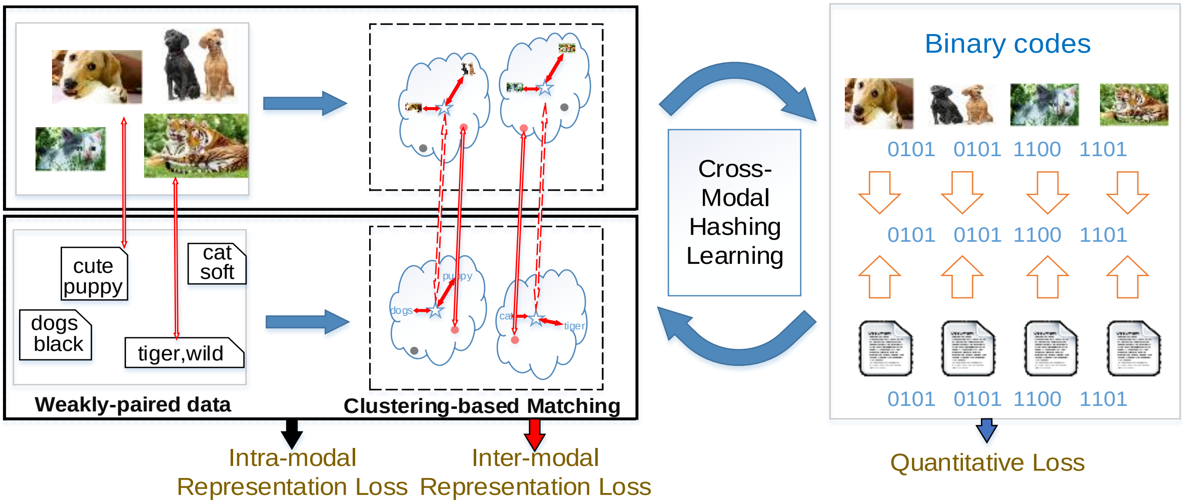


Fig. 1. Workflow of the proposed FlexCMH (Flexible Cross-Modal Hashing). FlexCMH includes two parts: (1) A clustering-based matching strategy to explore the matched clusters and samples therein across modalities; (2) A unified objective function to jointly account for the inter-modal representation loss, the intra-modal representation loss, and the quantitative loss to learn adaptive hashing functions. The intra-modality presentation loss aims at exploring the clusters and centroids of respective modalities. The inter-modal representation loss aims at preserving the proximity between samples of different modalities using matched samples. The quantitative loss aims at quantifying the hashing loss from the high-dimensional vectors to the compact binary codes.

TABLE I
NOTATION.

b	Hashing code length
k	Number of clusters
$\mathbf{X}^m \in \mathbb{R}^{d_m \times N_m}$	Feature data matrix of the m-th modality
$\mathbf{Z}^m \in \mathbb{R}^{d_m \times k}$	Cluster centroid matrix of the m-th modality
$\mathbf{H}^m \in \mathbb{R}^{k \times N_m}$	Cluster indicator matrix of the m-th modality
$\mathbf{\Gamma}^{mm'} \in \mathbb{R}^{N_m \times N_{m'}}$	Permutation matrix for two modalities
$\mathbf{X}_c^m$, $\mathbf{Z}_c^m$, $\mathbf{H}_c^m$	Feature, cluster centroid and cluster indicator matrices of the m-th modality with respect to the c-th cluster
$\mathbf{B} \in \mathbb{R}^{b \times n}$	Hashing code matrix

vector of sample i, and $\mathbf{x}_i^2$ is the tag vector of this sample. The notation used is summarized in Table I. To enable cross-modal hashing, we need to learn two hashing functions, F_1: $\mathbb{R}^{d_1} \rightarrow \{-1, 1\}^b$ and F_2: $\mathbb{R}^{d_2} \rightarrow \{-1, 1\}^b$, where b is the length of the binary hash codes. These two hashing functions are expected to map $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$ from the respective modality onto a common Hamming space, and to preserve the proximity of the original data.

This canonical cross-modal hashing assumes that training samples in different modalities have a complete correspondence. However, the samples may be weakly-paired only. For example, consider the scenario in which, due to a temporary sensor failure, $\mathbf{x}_i^1$ and $\mathbf{x}_i^2$ do not describe the same object from different feature views. Instead, $\mathbf{x}_i^1$ and $\mathbf{x}_j^2$ $(i \neq j)$ depict the same object from different views. Several efforts have been made to leverage paired and unpaired objects for cross-modal hashing [27], [35], [38], but they assume the training objects are identically paired. Furthermore, they cannot handle training objects whose correspondence is completely unknown, and modalities with different numbers of objects.

To achieve an effective cross-modal hashing on such weakly-paired (or totally unpaired) multi-modal data, we introduce a flexible solution (FlexCMH), and provide its overall workflow in Fig. 1. FlexCMH first introduces a clustering-based matching strategy to leverage the cluster centroids and the local structure around the centroids to explore the potential correspondence between clusters (and samples within) across different modalities. Next, it defines a permutation matrix based on the explored correspondence to align the index of the same samples across modalities. Based on the aligned index, it introduces a unified objective function to simultaneously account for cross-modal similarity preserving loss, the intra-modal representation loss, and the quantitative hashing loss. An alternative optimization technique is also proposed to jointly optimize the correspondence and the hash functions, and to reinforce the reciprocal effects of these two objectives. The following subsections elaborate on the above process.

B. *Clustering-based cross-modal matching strategy*

Unlike single-modal hashing, the correspondence between samples is crucial for multi-modal data fusion and retrieval. For completely matched samples, the correspondence is known and can be used, along with the inter(intra)-modality similarity between samples across modalities, to learn cross-modal hashing functions. But for weakly-paired data, since the correspondence is only partially known, the computation of similarities between samples of different modalities is a non-trivial task. A remedy is to divide the samples into different groups based on their labels, and impose constraints (i.e., concerning the similarity between different classes) on the coding vectors [44], [21]. In the representation space, the within-class data would cluster together although they are from different modalities, and the between-class data would be placed far apart from each other. In other words, the data vectors of the same class (different classes) from different modalities should be similar (dissimilar) [39]. We can approximate the similarity between different classes using the centroids of the respective groups [22]. However, considering only centroids may be insufficient, and the neighborhood objects around a centroid may also be helpful. Furthermore, incomplete labels of training data restrict the quality of groups. SPDH takes known paired samples as anchors to augment latent matched samples.

As such, it needs sufficient anchors for reliable matches, and cannot be applied to multi-modality data whose correspondence is completely unknown.

Given these observations, we introduce a novel clustering-based matching strategy to leverage the centroids of clusters and the local structure around the centroids. This strategy can explore the correspondence between clusters (and samples therein) between different modalities. We illustrate the clustering-based matching strategy in the center of Fig. 1, where the stars represent centroids of clusters in different modalities, and the red points indicate the objects with known correspondence in another modality. The likelihood that two clusters will match increases with the similarity of their centroids and with the similarity of the local structure around the centroids. To achieve this goal, we define the following matching function:

$$s_{cc'}^{mm'} = \sum_{g=1}^{n_s} (||\mathbf{x}_{c_g}^m - \mathbf{z}_c^m||_F^2 - \alpha||\mathbf{x}_{c'_g}^{m'} - \mathbf{z}_{c'}^{m'}||_F^2)^2 \qquad (1)$$

where $\mathbf{z}_c^m$ and $\mathbf{z}_{c'}^{m'}$ are the centroids of the c-th cluster in the m-th modality and the c'-th cluster in the m'-th modality; n_s is the user specified number of nearest samples to the centroids; $\mathbf{x}_{c_g}^m$ is the g-th nearest sample of $\mathbf{z}_c^m$; and $\alpha = ||\mathbf{z}_c^m||_F^2/||\mathbf{z}_{c'}^{m'}||_F^2$ is a scalar coefficient to balance the scale difference between two modalities. To seek the correspondence between clusters of different modalities, Eq. (1) evaluates the similarity of two clusters by measuring the consistency of ordered nearest neighbors. Therefore, both the centroids and neighborhood samples around the centroids are used to match clusters and samples therein, and thus to facilitate the follow-up cross-modal hashing. The smaller $s_{cc'}^{mm'}$ is, the more similar the two clusters are. In contrast, existing solutions only match centroids using labeled samples and ignore informative local patterns [18], [22]. Our matching function neither requires matched clusters to have the same number of samples, nor the same number of samples across modalities. It can also be applied to multi-modality data whose label information and correspondence are completely unknown. These advantages contribute to the flexibility of FlexCMH.

Two clusters (c and c'), and their respective centroids z_c^m and $z_{c'}^{m'}$, are matched if $s_{cc'}^{mm'}$ is the smallest among all pairwise clusters from two modalities. We can align the objects in the respective modalities by reordering their indices, and then use the 'matched' (aligned) objects in different modalities for cross-modality hashing. To this end, we define a permutation matrix $\mathbf{\Gamma}^{mm'} \in \mathbb{R}^{N_m \times N_{m'}}$ to align samples as follows:

$$\mathbf{\Gamma}_{ij}^{mm'} = \begin{cases} 1, & s_{cc'}^{mm'} \text{ is the smallest or } \mathbf{P}_{ij}^{mm'} = 1 \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

where $\mathbf{P}_{ij}^{mm'} = 1$ indicates that the i-th sample in the m-th modality is paired with the j-th sample in the m'-th modality. In this way, our cluster-based matching strategy also incorporates the known matched samples from different modalities. $\mathbf{\Gamma}_{ij}^{mm'} = 1$ if $\mathbf{x}_i^m$ belongs to the c-th cluster and $\mathbf{x}_j^{m'}$ belongs to the c'-th cluster (with the same order to their centroids), and $s_{cc'}^{mm'}$ is the smallest among all pairwise clusters from two modalities. These conditions indicate that the indices of $\mathbf{x}_i^m$ and $\mathbf{x}_j^{m'}$ should be reordered for alignment. We observe that our matching strategy is different from the typical network alignment, which aims at finding identical sub-networks [36], [30]. In contrast, we aim at matching samples within the explored clusters, which describe the same object in different feature views. In addition, a sample in one modality can be paired with more than one sample in another modality. The follow-up cross-modal hashing functions can be learned using the found correspondence.

C. *Cross-modal hashing*

To compute the matching loss, we must first identify the centroids of the respective clusters. WMCA [18] and MMPDL [22] both aim at addressing cross-model learning with weakly-paired samples, but they obtain clusters using only labeled samples. In practice, the labels of samples may not be sufficient, or just unavailable. As such, these methods have restricted flexibility. To find the centroids, we adopt Semi-Nonnegative Matrix Factorization (SemiNMF) [6] as follows:

$$\mathbf{L}_s = \sum_{m=1}^{M} ||\mathbf{X}^m - \mathbf{Z}^m\mathbf{H}^m||_F^2, \quad s.t.\ \mathbf{H}^m \geq 0 \qquad (3)$$

where $\mathbf{Z}^m \in \mathbb{R}^{d_m \times k}$ can be viewed as the latent representation of k cluster centroids of the m-th modality, and $\mathbf{H}^m \in \mathbb{R}^{k \times N_m}$ is the cluster assignment of samples in the latent space. Since clustering can explore the data distribution and achieve dimensionality reduction, the above equation can also quantify the *intra-modality* representation loss by clustering. Therefore, $\mathbf{Z}^m$ can be used for the clustering-based matching. $\mathbf{H}^m$ is the indicator matrix, which represents the probability that N_m samples belong to different clusters, and can be used for hashing code learning.

To achieve sample-to-sample cross-modal retrieval, based on the matched clusters and samples from Eq. (2), we further minimize the difference between the matched pairs to encourage them to be as similar as possible. Specifically, the indicator vectors ($\mathbf{H}^m$) of two samples from two different modalities should be similar if they have the same cluster label, and dissimilar otherwise. To this end, we quantify the relationship between two different modalities by minimizing the deviation of the indicator vectors of pairwise objects from different modalities as follows:

$$\mathbf{L}_c = \sum_{c=1}^{k} \sum_{m=1, m'>m}^{M} ||\mathbf{H}_c^m - \mathbf{H}_c^{m'}\mathbf{\Gamma}_c^{mm'}||_F^2 \qquad (4)$$

where $\mathbf{H}_c^m$ reorders the samples in $\mathbf{X}^m$ in descending order based on their association probabilities with respect to the c-th cluster. $\mathbf{\Gamma}_c^{mm'} \in \mathbb{R}^{N \times N}$ is the permutation matrix, which shuffles the sample indices in $\mathbf{H}_c^{m'}$ to align the samples according to the same indices in $\mathbf{H}_c^m$, which can be obtained using Eq. (2) and $\mathbf{Z}^m$. As such, the samples of $\mathbf{H}_c^{m'}$ can be matched with $\mathbf{H}_c^m$. In practice, we choose the top N ($< N_m$) samples which belong to the c (c') class to setup $\mathbf{H}_c^m$ and $\mathbf{H}_c^{m'}$, and to achieve the cross-modal matching. As a result, our matching strategy can accommodate the case in which the number of samples belonging to the same cluster (class) in different modalities is different. In this way, we can achieve cross-modal retrieval on multi-modal data, with partially or completely unknown matched samples, and with different numbers of samples in the matched clusters.

Next, we splice $\mathbf{H}_c^m$ and remove the repeated samples. As a result, we obtain $\mathbf{Y}^m \in \mathbb{R}^{k \times n}$ with the same order of matched samples across different modalities, and $n = \min\{N_m\}_{m=1}^M$ when modalities have a different number of samples. $\mathbf{Y}^m$ can be viewed as a k-dimensional new representation of samples in the m-th modality with respect to k centroids in a latent space. We further transform $\mathbf{Y}^m$ into a binary hashing coding matrix $\tilde{\mathbf{H}}^m \in \{-1|1\}^{b \times n}$ to facilitate compatible hashing codes for cross-modal hashing. More specifically, we first use k-means to obtain b-dimensional one-hot codes on $\mathbf{Y}^m$, and then set the largest entry of each row of $\mathbf{Y}^m$ to 0. We repeat k-means until the clustering results do not change anymore. Next, we merge all the one-hot codes of $\mathbf{Y}_i^m$ into a b-dimensional hashing code $\tilde{\mathbf{H}}_i^m$. In this way, we transform $\mathbf{Y}^m$ into binary hashing codes, which reflect the structure information of each cluster. After this transformation, we update $\tilde{\mathbf{H}}^m = 2\tilde{\mathbf{H}}^m - \mathbf{1}_{b \times n}$, and seek the common hamming hash coding matrix $\mathbf{B} \in \{-1|1\}^{b \times n}$ as follows:

$$\mathbf{L}_q = \sum_{m=1}^{M} ||\mathbf{B} - \tilde{\mathbf{H}}^m||_F^2 \tag{5}$$

where $\mathbf{B}$ can be viewed as the common Hamming space across all data modalities. It can be used for cross-modal retrieval, along with the $\tilde{\mathbf{H}}^m$ of the respective modalities. Eq. (5) is also called the hashing quantitative loss.

D. Unified objective function and optimization

Based on the above analysis, we can assemble the three losses into a unified objection function:

$$\begin{aligned} L(\mathbf{Z}^m, \mathbf{H}^m, \mathbf{B}) = \underset{\mathbf{Z}^m, \mathbf{H}^m, \mathbf{B}}{argmin} \sum_{c=1}^{k} \sum_{m=1, m'>m}^{M} ||\mathbf{H}_c^m - \mathbf{H}_c^{m'} \mathbf{\Gamma}_c^{mm'}||_F^2 \\ + \sum_{m=1}^{M} ||\mathbf{X}^m - \mathbf{Z}^m \mathbf{H}^m||_F^2 + \lambda \sum_{m=1}^{M} ||\mathbf{B} - \tilde{\mathbf{H}}^m||_F^2 \end{aligned} \tag{6}$$

where the first term quantifies the cross-modal matching loss from the cluster-wise and the inter-modal representation loss; the second term measures the intra-modal representation loss and seeks the clusters per-modality; and the third term measures the hashing code quantitative loss. Typically, the first two terms are equally weighted, since the cross-modal retrieval has to jointly preserve the inter- and intra-modal similarity [48]. For simplicity, we only use a scalar parameter λ to balance the hashing code learning, and inter- and intra-modality similarity preserving. By simultaneously optimizing the above three losses, we jointly account for the matching and the hash functions, and thus reinforce the reciprocal effects of the two objectives. This joint optimization can avoid the misleading impact on subsequent cross-modal hashing of initially not well-matched clusters and samples. Our experimental results confirm this conjecture.

The binary code of a new sample which is not in the training set can be easily generated. For example, let's consider a query instance in the first modality $\mathbf{x}^1$. Since we have obtained the clustering center $\mathbf{z}^1$ during the training process, we can compute its soft-cluster indicator vector $\mathbf{h}^1$ via Eq. (3), and determine its corresponding hash code as $\mathbf{b}^1 = \text{sign}(\tilde{\mathbf{h}}^1)$, where $\text{sign}(x) = 1$ if $x > 0$, $\text{sign}(x) = -1$ otherwise. $\tilde{\mathbf{h}}^1$ can be derived from the indicator vector $\mathbf{h}^1$ using the k-means clustering strategy given in the last paragraph of Section III-C.

We observe that the loss function in Eq. (6) is actually a sum of the cross-modal matching and retrieval loss, the intra-modal representation loss, and the hashing quantitative loss. Once $\mathbf{Z}^m$ is fixed, we can directly obtain $\mathbf{\Gamma}_c^{mm'}$ using Eq. (2). We can solve Eq. (6) via the Alternating Direction Method of Multipliers (ADMM) [1], which alternatively optimizes one of $\mathbf{Z}^m$, $\mathbf{H}^m$, and $\mathbf{B}$, while keeping the other two fixed.

Optimize $\mathbf{H}^m$ with $\mathbf{Z}^m$ and B fixed: We utilize stochastic gradient descent (SGD) to learn $\mathbf{H}^m$ Here, Eq. (6) is transformed into k independent optimization sub-problems for consistency and easy computation. The c-th sub-problem is:

$$\min \sum_{m=1, m'>m}^{M} ||\mathbf{H}_c^m - \mathbf{H}_c^{m'} \mathbf{\Gamma}_c^{mm'}||_F^2 + \sum_{m=1}^{M} ||\mathbf{X}_c^m - \mathbf{Z}^m \mathbf{H}_c^m||_F^2 \tag{7}$$

where $\mathbf{X}_c^m$ has the same size and sample order as $\mathbf{H}_c^m$. For any class, the derivative of Eq. (7) with respect to the indicator matrix $\mathbf{H}_c^m$ is:

$$\frac{\partial L}{\partial \mathbf{H}_c^m} = 2(\mathbf{Z}^{mT} \mathbf{Z}^m \mathbf{H}_c^m - \mathbf{Z}^{mT} \mathbf{X}_c^m) + \sum_{m'>m}^{M} 2(\mathbf{H}_c^m - \mathbf{H}_c^{m'} \mathbf{\Gamma}_c^{mm'}) \tag{8}$$

We can then take $\frac{\partial L}{\partial \mathbf{H}_c^m}$ to update the indicator matrix $\mathbf{H}_c^m$ using SGD. Similarly, we can also update $\mathbf{H}_c^{m'}$ based on the derivative $\frac{\partial L}{\partial \mathbf{H}_c^{m'}}$. After $\mathbf{H}_c^m$ is updated, the b-dimensional binary matrix $\tilde{\mathbf{H}}^m$ is consequently determined, and is used to optimize $\mathbf{B}$.

Optimize $\mathbf{Z}^m$ with $\mathbf{H}^m$ and B fixed: Considering two modalities as an example, since $\mathbf{\Gamma}_c^{mm'}$ depends on $\mathbf{Z}_c^m$ and $\mathbf{X}_c^m$, we can compute the derivative of Eq. (6) with respect to $\mathbf{\Gamma}_c^{mm'}$ and $\mathbf{Z}^m$ as follows:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{Z}^m} &= \frac{dL}{d\mathbf{Z}^m} + \frac{dL}{d\mathbf{\Gamma}_c^{mm'}} \frac{d\mathbf{\Gamma}_c^{mm'}}{d\mathbf{Z}^m} \\ &= 2\mathbf{Z}^m \mathbf{H}_c^m (\mathbf{H}_c^m)^T - 4\mathbf{X}_c^m (\mathbf{H}_c^m)^T + 2\mathbf{X}_c^m (\mathbf{\Gamma}_c^{mm'})^T (\mathbf{H}_c^{m'})^T \end{aligned} \tag{9}$$

We can then use these derivatives to update the centroid matrix $\mathbf{Z}^m$. Similarily, in each iteration, after the centroids in $\mathbf{Z}^m$ are updated, $\mathbf{\Gamma}_c^{mm'}$ is also determined. We consequently update $\mathbf{\Gamma}_c^{mm'}$ based on Eqs. (1-2), and use the updated $\mathbf{\Gamma}_c^{mm'}$ for the next optimization round. The optimization of Eq. (9) is given in the supplementary file.

Optimize B with $\mathbf{H}^m$ and $\mathbf{Z}^m$ fixed: Once $\mathbf{Z}^m$ and $\mathbf{H}^m$ are fixed, we can determine $\tilde{\mathbf{H}}^m$. The minimization in Eq. (6) is equivalent to the following maximization problem:

$$\max_{\mathbf{B}} tr(\mathbf{B}^T(\lambda \sum_{m=1}^{M} \tilde{\mathbf{H}}^m) = tr(\lambda \mathbf{B}^T \mathbf{U}) = \sum_{i,j} \mathbf{B}_{ij} \mathbf{U}_{ij} \tag{10}$$

where $\mathbf{B} \in \{-1, +1\}^{b \times N}$ and $\mathbf{U} = \lambda \sum_{m=1}^{M} \tilde{\mathbf{H}}^m$. It's easy to see that the binary code $\mathbf{B}_{ij}$ should have the same sign as $\mathbf{U}_{ij}$. Therefore, we have:

$$\mathbf{B} = sign(\mathbf{U}) = sign(\lambda \sum_{m=1}^{M} \tilde{\mathbf{H}}^m) \tag{11}$$

By iteratively applying Eqs. (8-11), we can jointly optimize the correspondence and the hash functions, and thus reinforce the reciprocal effects of the two objectives. The whole FlexCMH procedure and the alternative optimization for solving Eq. (6) are summarized in Algorithm 1. The convergence of the alternative optimization is studied in Section IV-C

Algorithm 1 FlexCMH: Flexible Cross-Modal Hashing

```
Input: $M$ modality data matrices $\mathbf{X}^m$, $m \in \{1, 2, \cdots, M\}$;
    the matched samples indicator matrix $\mathbf{P}^{mm'}$ (optional).
Output: Clustering centroid matrices $\mathbf{Z}^m$ and indicator matrices $\mathbf{H}^m$, binary code matrix $\mathbf{B}$.
Initialize centroid matrices $\mathbf{Z}^m$ , indicator matrices $\mathbf{H}^m$, the number of classes $k$ and the number of iterations $iter$, $t = 1$.
while $t < iter$ or Eq. (6) has not converged do
    for $c = 1 \rightarrow k$ do
        Update $\mathbf{H}_c^m$ using Eq. (8);
    end for
    Update $\mathbf{H}^m$ using Eq. (8) and then the permutation matrix $\mathbf{\Gamma}^{mm'}$ using Eqs. (1-2);
    Update $\mathbf{Z}^m$ using Eq. (9);
    Update $\mathbf{B}$ using Eq. (11);
    $t = t + 1$.
end while
```

E. Complexity analysis

To facilitate the time complexity analysis, we assume a simple extreme case with M modalities, k clusters, and t iterations. For any modality, we have n samples and the extreme pairing case is considered. The time complexity of the proposed method is composed of three parts. The time cost of updating $\mathbf{H}_c^m$ in Eq. (8) is $O(kM(k^2d+k^2n+kdn+(k^2d)(M-1)/2))$; the time cost of updating $\mathbf{Z}_c^m$ in Eq. (9) is $O(M(4dkn + nk^2))$; and the time cost of updating $\mathbf{\Gamma}^{mm'}$ in Eq. (2) is $O(k^2nd^2(M(M-1))/2)$. Since the complexity of the third part is larger than other two in each iteration, the overall complexity of FlexCMH is $O(tk^2n^2d(M(M-1))/2)$.

IV. EXPERIMENTS

A. Experimental setup

Datasets: Three widely used benchmark datasets (Nus-wide, Wiki, and Mirflicker) are collected to evaluate the performance of FlexCMH. Each dataset includes two modalities, image and text, although FlexCMH can also be directly applied to cases with more than two data modalities. Nus-wide[1] contains 269,648 web-text pairs. Each image is annotated with one or more labels taken from 81 concept labels. Each text is represented as a 1,000-dimensional bag-of-words vector. The hand-crafted feature of each image is a 500-dimensional bag-of-visual words (BOVW) vector. Wiki[2] is generated from a group of 2,866 Wikipedia documents. Each document is an image-text pair, can be annotated with 10 semantic labels, and is represented by a 128-dimensional SIFT feature vector. The text articles are represented as probability distributions over 10 topics, which are derived from a Latent Dirichlet Allocation (LDA) model. Mirflickr[3] originally contained 25,000 instances collected from Flicker. Each instance consists of an image and its associated textual tags, and is manually annotated with one or more labels, from a total of 24 semantic labels. Each text is represented as a 1,386-dimensional bag-of-words vector, and each image is represented by a 512-dimensional GIST feature vector.

Comparing methods: Eleven related and representative methods are adopted for comparison, which were introduced in the related work Section. (i) CMSSH (Cross-modal Similarity Sensitive Hashing) [2]; (ii) SCM (Semantic Correlation Maximization) [46]; (iii) CMFH (Collective Matrix Factorization Hashing) [7]; (iv) SePH (Semantics Preserving Hashing) [20]; (v) WMCA (Weakly-paired Maximum Correlation Analysis) [18]; (vi) MMPDL (Muti-Modal Projection Dictionary Learning) [22]; (vii) CCQ (Composite Correlation Quantization) [27]; (viii) SPDH (Semi-Paired Discrete Hashing) [35]; (ix) FDCH (Fast Discrete Cross-modal Hashing) [25]; (x) GSS-SL (Generalized Semi-supervised and Structured Subspace Learning) [47]; (xi) GSPH (Generalized Semantic Preserving Hashing) [5]. The codes of the baselines are available from the authors, and the input parameter values are set according to the guidelines given by the authors in their respective papers. We implemented SPDH, since its code is not available. WMCA, MMPDL, and GSS-SL are not hashing methods; thus, for these approaches we obtain the hashing codes by substituting the classification with the ordinary hashing function sign$(\cdot)$. For FlexCMH, we fix λ in Eq. (6) to 1, $k = 10$ on Wiki, $k = 25$ on Mirflickr, and $k = 80$ on Nus-wide; the number of nearest neighbors n_s in Eq. (1) is fixed to 5 and N in Eq. (4) is fixed to $min\{N_m, N_{m'}\}$. Our study shows that FlexCMH is robust to the input values of n_s and N. The number of iterations for optimizing Eq. (6) is set to 500. We empirically found that FlexCMH generally converges within 400 iterations on all the datasets. The sensitivity with respect to parameters λ and k is studied in the supplementary file.

B. Results in different practical settings

(i) **Settings:** We conducted experiments in three different settings: (1) completely-paired, (2) weakly-paired, and (3) completely-unpaired. In each type of experiment, all methods are run ten times, and we report the average MAP (mean average precision) results. The standard deviations of MAP results of the compared methods are quite small (less than 2%) across all datasets. To save space, we do not report the standard deviations in these tables. The best results are **boldfaced**. All the experimental settings with different scenarios are summarized in Table II.

For the completely-paired experiments, the clustering-based matching process of FlexCMH is excluded, and each comparing method uses all the paired samples for training (70%) and the rest for validation (30%). Table III reports the MAP results on Mirflickr, Nus-wide, and Wiki datasets. In the Table, 'Image vs. Text' denotes the setting where the query is an image and the database is text, and the vice versa for 'Text vs. Image'.

For the weakly-paired experiments, we investigate two different settings: (2a) 50% of the image-text pairs of the training set (70% of the whole dataset) are kept, and the other pairs are randomly shuffled. (2b) As in (2a), all the images in

---

[1] https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html

[2] https://www.wikidata.org/wiki/Wikidata

[3] http://press.liacs.nl/mirflickr/mirdownload.html

TABLE III

RESULTS (MAP) ON THREE DATASETS WITH COMPLETELY-PAIRED DATA.

		Mirflickr				Nus-wide				Wiki			
	Methods	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
Image vs. Text	CMSSH	0.5616	0.5555	0.5513	0.5484	0.3414	0.3336	0.3282	0.3261	0.1694	0.1523	0.1447	0.1434
	SCM-orth	0.5721	0.5607	0.5535	0.5482	0.3623	0.3646	0.3703	0.3721	0.1577	0.1434	0.1376	0.1358
	SCM-seq	0.6041	0.6112	0.6176	0.6232	0.4651	0.4714	0.4822	0.4851	0.2341	0.2411	0.2443	0.2564
	CMFH	0.6232	0.6256	0.6268	0.6293	0.4752	0.4793	0.4812	0.4866	0.2578	0.2591	0.2603	0.2612
	SePH	0.6573	0.6603	0.6616	0.6637	0.4787	0.4869	0.4888	0.4932	0.2836	0.2859	0.2879	0.2863
	WMCA	0.5834	0.5847	0.5856	0.5873	0.4396	0.4415	0.4433	0.4436	0.2243	0.2271	0.2283	0.2312
	MMPDL	0.6126	0.6135	0.6141	0.6128	0.4635	0.4658	0.4661	0.4672	0.2731	0.2745	0.2768	0.2801
	CCQ	0.6139	0.6152	0.6178	0.6221	0.5019	0.5027	0.5051	0.5073	0.2389	0.2412	0.2433	0.2446
	SPDH	0.6348	0.6356	0.6369	0.6391	0.5172	0.5189	0.5212	0.5231	0.2512	0.2533	0.2563	0.2578
	FDCH	0.6516	**0.6735**	**0.6814**	0.6691	0.4982	0.5031	0.5062	0.5055	0.2697	0.2755	0.2811	0.2817
	GSS-SL	0.6317	0.6353	0.6395	0.6472	0.4926	0.4957	0.4986	0.5013	0.2431	0.2463	0.2492	0.2514
	GSPH	0.6514	0.6579	0.6628	0.6692	0.5026	0.5057	0.5091	0.5116	0.2819	0.2842	0.2865	0.2879
	FlexCMH	**0.6639**	0.6674	0.6691	**0.6724**	**0.5211**	**0.5232**	**0.5249**	**0.5257**	**0.2846**	**0.2889**	**0.2912**	**0.2935**
Text vs. Image	CMSSH	0.5616	0.5551	0.5506	0.5475	0.3392	0.3321	0.3272	0.3256	0.1578	0.1384	0.1331	0.1256
	SCM-orth	0.5694	0.5611	0.5544	0.5497	0.3412	0.3459	0.3472	0.3539	0.1521	0.1561	0.1371	0.1261
	SCM-seq	0.6055	0.6154	0.6238	0.6299	0.4370	0.4428	0.4504	0.2235	0.2257	0.2459	0.2482	0.2518
	CMFH	0.6205	0.6237	0.6259	0.6286	0.4349	0.4387	0.4412	0.4425	0.2872	0.2891	0.2907	0.2923
	SePH	0.6481	0.6521	0.6545	0.6534	0.4489	0.4539	0.4587	0.4621	0.5345	0.5351	0.5471	0.5506
	WMCA	0.5847	0.5861	0.5886	0.5903	0.4179	0.4192	0.4221	0.4235	0.2089	0.2104	0.2131	0.2156
	MMPDL	0.6124	0.6142	0.6156	0.6172	0.4225	0.4232	0.4237	0.4256	0.2821	0.2824	0.2836	0.2861
	CCQ	0.6079	0.6096	0.6121	0.6145	0.5011	0.5025	0.5037	0.5046	0.2328	0.2339	0.2351	0.2371
	SPDH	0.6233	0.6254	0.6267	0.6283	0.5112	0.5124	0.5136	0.5146	0.2563	0.2584	0.2596	0.2607
	FDCH	0.6539	**0.6672**	**0.6744**	**0.6879**	0.4967	0.5025	0.5084	0.5173	0.4143	0.4115	0.4184	0.4202
	GSS-SL	0.6305	0.6346	0.6376	0.6431	0.4893	0.4935	0.4973	0.5011	0.2435	0.2472	0.2515	0.2535
	GSPH	0.6478	0.6492	0.6507	0.6523	0.4931	0.4977	0.5014	0.5021	**0.5812**	**0.5836**	**0.5849**	**0.5867**
	FlexCMH	**0.6601**	0.6632	0.6648	0.6676	**0.5137**	**0.5149**	**0.5172**	**0.5188**	0.2812	0.2836	0.2857	0.2869

TABLE IV

RESULTS (MAP) ON THREE DATASETS WITH WEAKLY-PAIRED DATA.

		Mirflickr				Nus-wide				Wiki			
	Methods	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
	50% image-text pairs are paired, all methods use all the paired and unpaired data												
Image vs. Text	CMSSH	0.5216	0.5238	0.5244	0.5249	0.2715	0.2731	0.2757	0.2766	0.1011	0.1023	0.1035	0.1031
	SCM-orth	0.5398	0.5401	0.5406	0.5412	0.2953	0.2968	0.2991	0.3012	0.1107	0.1112	0.1125	0.1128
	SCM-seq	0.5404	0.5413	0.5430	0.5442	0.3343	0.3358	0.3372	0.3395	0.1126	0.1138	0.1149	0.1168
	CMFH	0.5405	0.5422	0.5438	0.5447	0.3409	0.3428	0.3442	0.3462	0.1157	0.1165	0.1179	0.1182
	SePH	0.5411	0.5436	0.5467	0.5501	0.3561	0.3582	0.3610	0.3612	0.1235	0.1267	0.1284	0.1302
	WMCA	0.5456	0.5463	0.5471	0.5489	0.3721	0.3746	0.3758	0.3761	0.1575	0.1593	0.1611	0.1635
	MMPDL	0.5778	0.5792	0.5814	0.5846	0.4117	0.4136	0.4137	0.4136	0.2342	0.2361	0.2375	0.2341
	CCQ	0.5806	0.5812	0.5826	0.5833	0.4359	0.4368	0.4372	0.4386	0.2253	0.2274	0.2281	0.2294
	SPDH	0.5784	0.5803	0.5826	0.5832	0.4312	0.4326	0.4335	0.4342	0.2342	0.2363	0.2375	0.2389
	FDCH	0.5622	0.5732	0.5767	0.5787	0.4125	0.4163	0.4219	0.4240	0.2193	0.2237	0.2269	0.2273
	GSS-SL	0.5771	0.5753	0.5795	0.5832	0.4210	0.4239	0.4277	0.4296	0.2223	0.2256	0.2289	0.2311
	GSPH	0.5651	0.5649	0.5681	0.5712	0.4062	0.4091	0.4123	0.4135	0.2519	0.2531	0.2546	0.2571
	FlexCMH	**0.5867**	**0.5891**	**0.5925**	**0.5973**	**0.4473**	**0.4496**	**0.4515**	**0.4531**	**0.2629**	**0.2647**	**0.2655**	**0.2687**
Text vs. Image	CMSSH	0.5121	0.5135	0.5142	0.5136	0.2563	0.2607	0.2622	0.2741	0.0989	0.1002	0.1011	0.1020
	SCM-orth	0.5211	0.5226	0.5237	0.5242	0.2855	0.2879	0.2893	0.2921	0.1118	0.1124	0.1121	0.1128
	SCM-seq	0.5235	0.5238	0.5241	0.5250	0.3211	0.3234	0.3269	0.3274	0.1206	0.1209	0.1214	0.1221
	CMFH	0.5314	0.5335	0.5356	0.5372	0.3382	0.3397	0.3421	0.3442	0.1231	0.1255	0.1269	0.1293
	SePH	0.5431	0.5441	0.5453	0.5459	0.3531	0.3554	0.3560	0.3579	0.1238	0.1242	0.1247	0.1264
	WMCA	0.5456	0.5461	0.5458	0.5472	0.3612	0.3648	0.3679	0.3712	0.1437	0.1445	0.1458	0.1473
	MMPDL	0.5631	0.5647	0.5648	0.5655	0.3872	0.3891	0.3911	0.3924	0.2132	0.2141	0.2155	0.2135
	CCQ	0.5732	0.5743	0.5755	0.5763	0.4267	0.4281	0.4299	0.4312	0.2216	0.2237	0.2253	0.2263
	SPDH	0.5715	0.5736	0.5751	0.5768	0.4213	0.4234	0.4267	0.4286	0.2365	0.2379	0.2393	0.2411
	FDCH	0.5652	0.5683	0.5712	0.5729	0.4155	0.4183	0.4211	0.4246	0.2336	0.2418	0.2502	0.2533
	GSS-SL	0.5692	0.5713	0.5746	0.5791	0.4218	0.4245	0.4283	0.4311	0.2215	0.2258	0.2384	0.2491
	GSPH	0.5613	0.5635	0.5649	0.5692	0.4205	0.4237	0.4262	0.4291	0.2522	0.2539	**0.2573**	**0.2639**
	FlexCMH	**0.5801**	**0.5825**	**0.5836**	**0.5859**	**0.4431**	**0.4456**	**0.4479**	**0.4412**	**0.2538**	**0.2541**	0.2557	0.2563
	50% image-text pairs are paired, the number of image samples and that of text samples are different												
Image vs. Text	FlexCMH(nJ)	0.5421	0.5435	0.5467	0.5485	0.3878	0.3892	0.3905	0.3936	0.2231	0.2245	0.2256	0.2273
	FlexCMH(nC)	0.5259	0.5286	0.5304	0.5327	0.3618	0.3643	0.3666	0.3647	0.2015	0.2057	0.2076	0.2088
	FlexCMH	**0.5635**	**0.5641**	**0.5653**	**0.5668**	**0.4333**	**0.4351**	**0.4367**	**0.4383**	**0.2577**	**0.2593**	**0.2612**	**0.2635**
Text vs. Image	FlexCMH(nJ)	0.5224	0.5237	0.5244	0.5256	0.4115	0.4123	0.4143	0.4150	0.2235	0.2256	0.2271	0.2293
	FlexCMH(nC)	0.5183	0.5104	0.5131	0.5142	0.3832	0.3845	0.3873	0.3907	0.2056	0.2074	0.2093	0.2108
	FlexCMH	**0.5589**	**0.5624**	**0.5643**	**0.5658**	**0.4327**	**0.4335**	**0.4354**	**0.4388**	**0.2503**	**0.2515**	**0.2534**	**0.2542**

TABLE V

RESULTS (MAP) ON THREE DATASETS WITH COMPLETELY-UNPAIRED DATA.

		Mirflickr				Nus-wide				Wiki			
	Methods	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
Image vs. Text	WMCA	0.5214	0.5231	0.5245	0.5263	0.3559	0.3574	0.3591	0.3604	0.1276	0.1295	0.1310	0.1336
	MMPDL	0.5535	0.5542	0.5567	0.5588	0.3963	0.3984	0.4004	0.4015	0.2210	0.2231	0.2254	0.2268
	GSPH	0.5582	0.5597	0.5616	0.5633	0.4179	0.4183	0.4199	0.4237	0.2438	0.2436	0.2453	0.2475
	FlexCMH	**0.5693**	**0.5704**	**0.5723**	**0.5749**	**0.4215**	**0.4235**	**0.4259**	**0.4273**	**0.2511**	**0.2534**	**0.2548**	**0.2563**
Text vs. Image	WMCA	0.5256	0.5263	0.5278	0.5293	0.3414	0.3438	0.3467	0.3481	0.1335	0.1344	0.1358	0.1381
	MMPDL	0.5489	0.5503	0.5531	0.5547	0.3635	0.3678	0.3691	0.3713	0.2015	0.2038	0.2074	0.2098
	GSPH	0.5515	0.5542	0.5576	0.5601	0.4013	0.4047	0.4065	0.4093	0.2402	0.2426	0.2455	0.2476
	FlexCMH	**0.5631**	**0.5652**	**0.5681**	**0.5694**	**0.4131**	**0.4158**	**0.4183**	**0.4212**	**0.2437**	**0.2459**	**0.2483**	**0.2501**

TABLE II
EXPERIMENTAL SETTINGS WITH DIFFERENT SCENARIOS. 'T/I' GIVES THE PROPORTION OF TRAINING IMAGES AND TEXTS, RESPECTIVELY; 'P/U' GIVES THE PROPORTION OF PAIRED AND UNPAIRED SAMPLES.

scenarios	T/I	P/U
completely-paired	0.7/0.7	1/0
weakly-paired(2a)	0.7/0.7	0.5/0.5
weakly-paired(2b)	0.7/0.6	0.5/0.5
completely-unpaired	0.7/0.7	0/1

the training set are used for training, but 10% of the text samples in the training set is randomly removed. As such, the number of images is different from the number of text samples across modalities and clusters. For the setting (2b), all the comparing methods cannot be applied, so we only report the MAP results of our FlexCMH and its variants (FlexCMH(nJ) and FlexCMH(nC)). FlexCMH(nJ) first seeks the potential image-text pairs, and then executes the follow-up cross-modal hashing, without jointly optimizing the matched clusters (samples) and hashing functions in a coherent fashion. FlexCMH(nC) uses the label information to obtain the correspondence between samples (as done by MMPDL), instead of our proposed clustering-based matching strategy. Table IV reports the MAP values of the compared methods in these settings.

For the completely-unpaired experiments, besides randomly partitioning the data into training (70%) and testing (30%) sets, we randomly shuffle the index of images and the index of text samples in the training set. As a result, the images and the text samples are almost completely unpaired. CCQ, SPDH, GSS-SL, and GSPH all require paired samples for training, so they cannot be run in this setting. For this type of experiments, only WMCA, MMPDL, and GSPH can be used for comparison. Table V reports the MAP values of the three methods.

(ii) **Completely-paired:** Table III shows that our FlexCMH achieves the best performance in most cases. This is because FlexCMH not only jointly models the cross-modal and intra-modal similarity preserving losses, to build a more faithfully semantic projection, but also models the quantitative loss to learn adaptive hashing codes. We observe that SePH and GSPH obtain better results for 'Text vs. Image' retrieval on the small Wiki dataset. This is possible because they consider different data distributions for different modalities, while FlexCMH adopts a consistent clustering-based strategy for all modalities. An unexpected observation is that the performance of CMSSH and SCM-Orth decreases as the length of hash codes increases. This might be caused by the imbalance between bits in the hash codes learned by singular value or eigenvalue decomposition. These experimental results show the effectiveness of FlexCMH for the canonical cross-modal hashing, where training samples from different modalities are completely paired.

(iii) **Weakly-paired:** Compared with the results in Table III, all the methods manifest reduced MAP values in Table IV, where weakly-paired training sets are used. This observation suggests the correspondence information is of paramount importance for cross-modal hashing. SePH is a probability-based method that learns unified hashing codes across all views; it performs well on small text datasets. However, when dealing with weakly-paired data, where the important pair information between samples across modalities is destroyed, it cannot maintain well-unified hashing codes across views, and has significantly compromised results. WMCA, MMPDL, CCQ, SPDH, GSS-SL, GSPH, and FlexCMH give better results than other comparing methods. That is because they adopt different techniques to augment matched samples, which boost the performance of cross-modal hashing. MMPDL, CCQ, and GSS-SL are outperformed by SPDH, CCQ, GSPH, and FlexCMH, since they are not targeted to hashing, and the adopted $sign$ to convert their numeric outputs into hashing codes is not ideal. Although SPDH, CCQ, GSPH, and FlexCMH all try to augment paired samples, FlexCMH still outperforms the former three in most cases. This is due to: (1) its novel clustering-based matching approach to explore the matched clusters and samples therein, and (2) a unified objective function to optimize, in a coordinated manner, the matching between clusters and samples, and the cross-modal hashing functions with the matched clusters and samples. Results (reported in the Supplementary file) obtained using another evaluation metric, Precision with Hamming radius (PH) [24], drive similar observations and conclusions.

FlexCMH gives slightly reduced MAP values when the numbers of samples (images and texts) in different modalities are not the same, and only 50% of the image-text is paired. In 'Image vs. Text' retrieval, the MAP results of FlexCMH are generally lower than those in 2(a). This is because 10% of the text samples in the text modality is removed. We also study the performance of FlexCMH with other ratios of unpaired samples (from 30% to 90%) and of removed samples (from 10% to 40%), and report the results in the Supplementary file.

FlexCMH(nJ) isolates the optimization of matching samples and of hash functions, and its MAP values are lower than those of FlexCMH. This observation proves that jointly optimizing the hashing functions and the matched clusters and samples enables a mutual boost of the two objectives. FlexCMH(nC) simply adopts the label information to set the correspondence between samples, and it also loses to FlexCMH. This fact proves that our proposed clustering-based matching strategy can more reliably find the matching between samples across modalities.

(iv) **Unpaired:** The MAP results of all methods in Table V are inferior to those of Table III. Still, FlexCMH achieves the best results, which proves the effectiveness of FlexCMH on completely-unpaired data. From these results, we can state that the matching information of samples across modalities is crucial for cross-modal hashing. Our clustering-based matching strategy can reliably explore paired samples, and it boosts the performance of cross-modal hashing on weakly-paired (or completely unpaired) samples.

We further studied the performance of FlexCMH with pre-learnt deep features and on datasets with more than two modalities, and performed parameter sensitivity analysis. The results (reported in the Supplementary file) show that FlexCMH outperforms the competing methods also in scenarios with more than three modalities and with pre-learnt deep features.

In summary, our experimental results prove that FlexCMH can learn cross-modal hashing codes more effectively than representative comparing methods. FlexCMH can be applied in a variety of practical settings, where paired samples across modalities are either partially available or completely unknown, and the numbers of samples in different modalities (and matched clusters) are also different. To the best of our knowledge, existing cross-modal hashing methods [41], [43] cannot be applied in these settings, or can work only for the weakly-paired setting [35], [27], [5].

C. Convergence curves and runtime analysis

We further plot the unified objective function loss ($L(\mathbf{Z}^m, \mathbf{H}^m, \mathbf{B})$ in Eq. (6) under different iterations in Fig. 2. FlexCMH reaches a convergence state after 30, 100, and 400 iterations on Wiki, Mirflickr and Nus-wide, respectively. In addition, we report the runtime costs of the compared methods on three datasets in Table VI, where the experimental settings are the same as in (2a) of Section IV-B. All the experiments were conducted on a server with Intel E5-2650v3, 256GB RAM, and Ubuntu 16.04.01 OS.

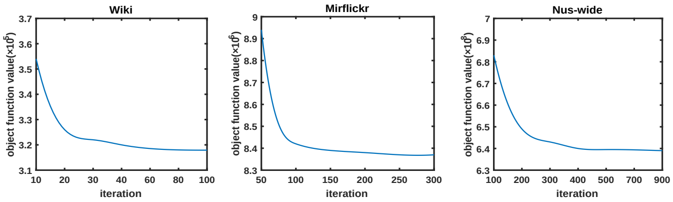


Fig. 2. Objective function loss of FlexCMH on three datasets as the number of iterations increases.

TABLE VI

RUNNING TIMES (IN SECONDS) WITH CODE LENGTH FIXED TO 16.

	Wiki		Mirflickr		Nus-wide	
	train	search	train	search	train	search
CMSSH	113.51	1.36	1147.56	5.86	1038103.08	532.15
SCM-seq	10.13	0.11	75.16	0.72	59154.34	37.89
SCM-orth	16.41	0.15	113.42	1.15	128752.64	155.36
SePH	30.41	0.33	211.75	1.76	434575.12	134.67
WMCA	15.39	0.19	108.47	1.02	121759.26	135.81
WMPDL	75.61	1.06	876.31	3.15	948972.64	373.56
CCQ	19.28	0.21	185.34	1.55	165491.36	110.34
FDCH	18.93	0.26	212.15	1.61	181345.15	131.21
GSS-SL	45.17	0.81	302.87	2.56	673215.43	212.54
FlexCMH	15.31	0.12	96.42	0.86	74163.55	46.13

We can see that FlexCMH costs 15.31 seconds on Wiki, 96.42 seconds on Mirflickr, and 74163.55 seconds on Nus-wide. FlexCMH is the second most efficient method, despite the fact that it seeks the matching between samples across modalities. This demonstrates the efficiency of the proposed alternative optimization procedure for the unified objective function. WMCA, WMPDL, and GSS-SL also seek the matching between samples across modalities, but they run slower than FlexCMH. This is because FlexCMH simultaneously achieves the matching between samples and inter-modal similarity preservation, whereas WMCA, WMPDL, and GSS-SL pursue the two goals separately, and thus they have to spend more time computing intermediate variables. SCM-seq uses sequential learning to optimize the hash bits, and runs faster than all the other comparing methods. These results demonstrate that our proposed FlexCMH performs weakly-paired cross-modal retrieval efficiently.

V. CONCLUSIONS

We proposed a flexible cross-modal hashing solution (FlexCMH) to learn effective hashing functions from weakly-paired (or completely-unpaired) data across modalities. FlexCMH uses a clustering-based matching strategy to explore the potential correspondence between clusters and their member samples. In addition, we introduced a unified objective function to jointly optimize the cross-modal matching loss, the intra(inter)-modal representation loss, and the quantitative loss to learn adaptive hashing codes in a coherent way. Our extensive experiments have shown that FlexCMH outperforms several competitive hashing methods on completely-paired, weakly-paired, and completely-unpaired multi-modality data. In the future, we will incorporate deep learning into cross-modal hashing on weakly-paired data.

REFERENCES

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[2] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, pages 3594–3601, 2010.

[3] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *ICDM*, pages 410–419, 2008.

[4] C. Deng, E. Yang, T. Liu, and D. Tao. Two-stream deep hashing with class-specific centers for supervised image search. *TNNLS*, 31(6):2189–2201, 2020.

[5] S. B. Devraj Mandal, Kunal N. Chaudhury. Generalized semantic preserving hashing for cross-modal retrieval. *TIP*, 28(1):102–112, 2019.

[6] C. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *TPAMI*, 32(1):45–55, 2010.

[7] G. Ding, Y. Guo, and J. Zhou. Collective matrix factorization hashing for multimodal data. In *CVPR*, pages 2083–2090, 2014.

[8] K. Ding, C. Huo, B. Fan, S. Xiang, and C. Pan. In defense of locality-sensitive hashing. *TNNLS*, 29(1):87–103, 2018.

[9] Y. Fang, H. Zhang, and Y. Ren. Unsupervised cross-modal retrieval via multi-modal graph regularized smooth matrix factorization hashing. *Knowledge-Based Systems*, 171:69–80, 2019.

[10] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 106(2):210–233, 2014.

[11] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan. Supervised discrete hashing with relaxation. *TNNLS*, 29(3):608–617, 2018.

[12] Z. Hu, X. Liu, X. Wang, Y.-m. Cheung, N. Wang, and Y. Chen. Triplet fusion network hashing for unpaired cross-modal retrieval. In *ICMR*, pages 141–149, 2019.

[13] Q. Y. Jiang and W. J. Li. Deep cross-modal hashing. In *CVPR*, pages 3270–3278, 2017.

[14] L. Jin, K. Li, Z. Li, F. Xiao, G. J. Qi, and J. Tang. Deep semantic-preserving ordinal hashing for cross-modal similarity search. *TNNLS*, 30(5):1429–1440, 2019.

[15] Z. Jin, Y. Hu, Y. Lin, D. Zhang, S. Lin, D. Cai, and X. Li. Complementary projection hashing. In *ICCV*, pages 257–264, 2013.

[16] C. Kang, S. Xiang, S. Liao, C. Xu, and C. Pan. Learning consistent feature representation for cross-modal multimedia retrieval. *TMM*, 17(3):370–381, 2015.

[17] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, pages 1360–1365, 2011.

[18] C. H. Lampert and O. Krömer. Weakly-paired maximum covariance analysis for multimodal dimensionality reduction and transfer learning. In *ECCV*, pages 566–579, 2010.

[19] C.-X. Li, Z.-D. Chen, P.-F. Zhang, X. Luo, L. Nie, W. Zhang, and X.-S. Xu. Scratch: A scalable discrete matrix factorization hashing for cross-modal retrieval. In *ACM MM*, pages 1–9, 2018.

[20] Z. Lin, G. Ding, J. Han, and J. Wang. Cross-view retrieval via probability-based semantics-preserving hashing. *IEEE Trans. on Cybernetics*, 47(12):4342–4355, 2017.

[21] H. Liu, W. Feng, X. Zhang, and F. Sun. Weakly-paired deep dictionary learning for cross-modal retrieval. *PRL*, 130:199–206, 2020.

[22] H. Liu, Y. Wu, F. Sun, B. Fang, and G. Di. Weakly paired multimodal fusion for object recognition. *TASE*, 15(2):784–795, 2018.

[23] Q. Liu, G. Liu, L. Li, X. Yuan, M. Wang, and W. Liu. Reversed spectral hashing. *TNNLS*, 29(6):2441–2449, 2018.

[24] X. Liu, Z. Li, C. Deng, and D. Tao. Distributed adaptive binary quantization for fast nearest neighbor search. *TIP*, 26(11):5324–5336, 2017.

[25] X. Liu, X. Nie, W. Zeng, C. Cui, L. Zhu, and Y. Yin. Fast discrete cross-modal hashing with regressing from semantic labels. In *ACM MM*, pages 1662–1669, 2018.

[26] X. Liu, G. Yu, C. Domeniconi, J. Wang, Y. Ren, and M. Guo. Ranking-based deep cross-modal hashing. In *AAAI*, pages 4400–4407, 2019.

[27] M. Long, Y. Cao, J. Wang, and P. S. Yu. Composite correlation quantization for efficient multimodal retrieval. In *SIGIR*, pages 579–588, 2016.

[28] X. Lu, L. Zhu, Z. Cheng, X. Song, and H. Zhang. Efficient discrete latent semantic hashing for scalable cross-modal retrieval. *Signal Processing*, 154:217–231, 2019.

[29] D. Mandal and S. Biswas. Generalized coupled dictionary learning approach with applications to cross-modal matching. *TIP*, 25(8):3826–3837, 2016.

[30] L. Meng, A. Striegel, and T. Milenkovi? Local versus global biological network alignment. *Bioinformatics*, 32(20):3155–3164, 2015.

[31] Y. Peng, X. Huang, and Y. Zhao. An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges. *TCSVT*, 28(9):2372–2385, 2018.

[32] F. Shang, H. Zhang, L. Zhu, and J. Sun. Adversarial cross-modal retrieval based on dictionary learning. *Neurocomputing*, 355:93–104, 2019.

[33] F. Shen, C. Shen, L. Wei, and H. T. Shen. Supervised discrete hashing. In *CVPR*, pages 37–45, 2015.

[34] F. Shen, L. Wei, S. Zhang, Y. Yang, and H. T. Shen. Learning binary codes for maximum inner product search. In *ICCV*, pages 4148–4156, 2015.

[35] X. Shen, F. Shen, Q. S. Sun, Y. Yang, Y. H. Yuan, and H. T. Shen. Semi-paired discrete hashing: Learning latent hash codes for semi-paired cross-view retrieval. *IEEE T. Cybernetics*, 47(12):4275–4288, 2017.

[36] Z. Si and H. Tong. Final: Fast attributed network alignment. In *KDD*, pages 1345–1354, 2016.

[37] D. Song, W. Liu, R. Ji, D. A. Meyer, and J. R. Smith. Top rank supervised binary coding for visual search. In *ICCV*, pages 1922–1930, 2015.

[38] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*, pages 785–796, 2013.

[39] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding for automatic image annotation. In *CVPR*, pages 1643–1650, 2009.

[40] J. Wang, S. Kumar, and S. F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, pages 3424–3431, 2010.

[41] J. Wang, W. Liu, S. Kumar, and S.-F. Chang. Learning to hash for indexing big data – a survey. *Proc. of the IEEE*, 104(1):34–57, 2016.

[42] J. Wang, L. Wei, A. X. Sun, and Y. G. Jiang. Learning hash codes with listwise supervision. In *ICCV*, pages 3032–3039, 2014.

[43] J. Wang, T. Zhang, N. Sebe, and H. T. Shen. A survey on learning to hash. *TPAMI*, 40(4):769–790, 2018.

[44] L. Xiao, M. Song, D. Tao, X. Zhou, C. Chen, and J. Bu. Semi-supervised coupled dictionary learning for person re-identification. In *CVPR*, pages 3550–3557, 2014.

[45] L. Xie, J. Shen, and L. Zhu. Online cross-modal hashing for web image retrieval. In *AAAI*, pages 294–300, 2016.

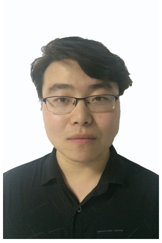
[46] D. Zhang and W. J. Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, pages 2177–2183, 2014.

[47] L. Zhang, B. Ma, G. Li, Q. Huang, and Q. Tian. Generalized semi-supervised and structured subspace learning for cross-modal retrieval. *TMM*, 20(1):128–141, 2017.

[48] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao. Linear cross-modal hashing for efficient multimedia search. In *ACM MM*, pages 143–152, 2013.

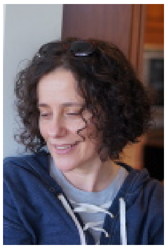
[49] L. Zong, X. Zhang, and X. Liu. Multi-view clustering on unmapped data via constrained non-negative matrix factorization. *Neural Networks*, 108:155–171, 2018.

Guoxian Yu is a Professor at the School of Software, Shandong University, Jinan, China. He received the Ph.D. in Computer Science from South China University of Technology, Guangzhou, China in 2013. His current research interests include data mining and bioinformatics. He severs as reviewers for KDD, ICDM, IJCAI, AAAI, TKDE, TNNLS, TCBB, Bioinformatics and other prestigious conferences and journals.



Xuanwu Liu received the MSc degree from the College of Computer and Information Sciences, Southwest University, Chongqing, China in 2020. He is currently an algorithm engineer with the Alibaba Group, specializing in multi-modality data analysis, hashing and anomaly detection.

Jun Wang is a Professor with the Joint SDU-NTU Centre for Artificial Intelligence Research, Shandong University. She received B.Sc. degree in Computer Science, M.Eng. degree in Computer Science and Ph.D. in Artificial Intelligence from Harbin Institute of Technology, Harbin, China in 2004, 2006 and 2010, respectively. Her current research interests include data mining and their applications in bioinformatics.



Carlotta Domeniconi is an Associate Professor in the Department of Computer Science at George Mason University. Her research interests include machine learning, data mining and bioinformatics. She serves as the Associate Editor of IEEE Transactions on Knowledge and Data Engineering, and Knowledge and Information Systems. She was an Associate Editor of IEEE Transactions on Neural Networks and Learning System.

Xiangliang Zhang is an Associate Professor and directs the Machine Intelligence and Knowledge Engineering (MINE) Laboratory in King Abdullah University of Science and Technology (KAUST). She earned her PhD degree in Computer Science with great honors from INRIA-University Paris-Sud 11, France, in 2010. Her main research interests and experiences are in diverse areas of machine learning and data mining.