Collaboratively Semantic Alignment and Metric Learning for Cross-modal Hashing

Jiaxing Li, Wai Keung Wong, Lin Jiang, Kaihang Jiang, Xiaozhao Fang, Shengli Xie, *Fellow, IEEE*, Jie Wen

***Abstract*—Cross-modal retrieval is a promising technique nowadays to find semantically similar instances in other modalities while a query instance is given from one modality. However, there still exists many challenges for reducing heterogeneous modality gap by embedding label information to discrete hash codes effectively, solving the binary optimization when generating unified hash codes and reducing the discrepancy of data distribution efficiently during common space learning. In order to overcome the above-mentioned challenges, we propose a Collaboratively Semantic alignment and Metric learning for cross-modal Hashing (CSMH) in this paper. Specifically, by a kernelization operation, CSMH firstly extracts the non-linear data features for each modality, which are projected into a latent subspace to align both marginal and conditional distributions simultaneously. Then, a maximum mean discrepancy-based metric strategy is customized to mitigate the distribution discrepancies among features from different modalities. Finally, semantic information obtained from the label similarity matrix, is further incorporated to embed the latent semantic structure into the discriminant subspace. Experimental results of CSMH and baseline methods on four widely-used datasets show that CSMH outperforms some state-of-the-art hashing baseline methods for cross-modal retrieval on efficiency and precision.**

***Index Terms*—Cross-modal hashing, semantic alignment, maximum mean discrepancy, metric learning, information retrieval.**

I. Introduction

RECENTLY, massive volume of multimedia data with high dimension and heterogeneous modalities are generated by users on the world wide web [1], [2]. The demands for people retrieving multimedia data from different modalities are also increasing dramatically. Actually, people hope the model for retrieving tasks in reality can handle cross-modal retrieval. For example, when people see an object they do not know or remember, they will try to retrieve the relevant textual or audio descriptions from the object's photos, and vice versa. However, retrieving multimedia data in large scale datasets from various modalities will be a great challenge, as the volume of multimedia information is growing explosively.

Single modality retrieval can no longer meet the increasing demands of users, as multimedia stream contains data from various modalities [3]. Thus, the cross-modal retrieval is of increasing important to handle the daily information retrieving tasks for users [4]. As an intuitive and important technique for information retrieval, nearest neighbor searching is widely applied in computer vision, data mining, etc. However, it is with huge difficulty to find exact nearest neighbors for a query to large scale datasets within acceptable complexity on time and space. Fortunately, hashing technique owns the potential ability to solve the problems mentioned above, because its promising accuracy, efficiency and storage-friendly feature for cross-modal retrieval [5], [6], [7], [8].

To accelerate the cross-modal retrieval, existing hashing-based methods project data from different modalities into a common space, which makes cross-modal similarity search easier and faster. Although many hashing-based methods are proposed to achieve an efficient retrieval for multimedia data [9], [10], it is still insufficient to generate high quality hash codes by using only linear embedding of original features. Meanwhile, there exits a huge difference for data distributions between different modalities. Moreover, how to reduce the difference of data distributions and extract the common data features will be crucial to improve the effectiveness of modal for cross-modal retrieval.

Common space learning is an essential solution for reducing this difference, but most of the existing common space learning-based methods do not consider the distances of instances within or between different modalities. Therefore, a simple but effective non-linear embedding is urgently needed to capture the non-linear structure of data from cross-modal modalities. As a powerful non-linear feature extraction tool, kernelization has been explored in some related works on cross-modal retrieval, though it is not yet a widely adopted approach and its full potential in this domain remains unexplored. Some existing related works (e.g., [11] and [12], etc.) have demonstrated the effectiveness of kernelization in handling non-linear issues in cross-modal retrieval tasks, but there still have room for further investigating its applications. In this paper, kernelization plays a significant role in addressing the challenge of dimensionality discrepancy between different

Jiaxing Li is with the School of Artificial Intelligence, Guangzhou University, Guangzhou 510006, China (email: jiaxing.li.cs@gmail.com).

Wai Keung Wong and Kaihang Jiang are with the School of Fashion and Textiles, The Hong Kong Polytechnic University, and Laboratory for Artificial Intelligence in Design, Hong Kong SAR (email: calvin.wong@polyu.edu.hk, jkh1650290810@163.com).

Lin Jiang is with the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China. (email: linn_jiang@163.com).

Xiaozhao Fang is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Key Laboratory of Intelligent Detection and The Internet of Things in Manufacturing (GDUT), Ministry of Education, Guangzhou 510006, China (email: xzhfang168@126.com).

Shengli Xie is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing (GDUT), Guangzhou 510006, China (email: shlxie@gdut.edu.cn).

Jie Wen is with the Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen 518055, China, and also with the Shenzhen Key Laboratory of Visual Object Detection and Recognition, Shenzhen 518055, China (email: jiewen_pr@126.com)

(Corresponding authors: Wai Keung Wong, Xiaozhao Fang and Jie Wen.)

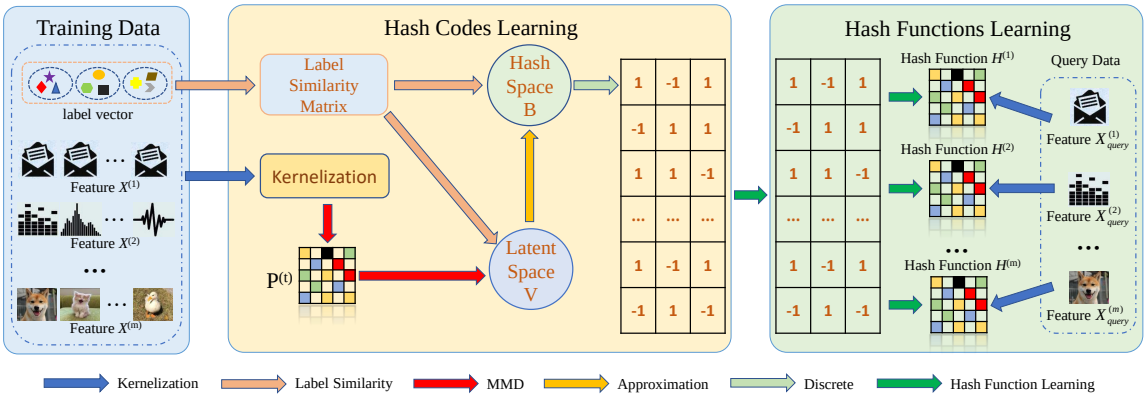


Fig. 1. The framework of the proposed CSMH. Firstly, the kernelization operation is introduced to extract non-linear features of the original data. Then, a MMD-based metric strategy is introduced to mitigate the distribution discrepancies among features from different modalities. Specifically, it projects the features into a latent subspace to simultaneously align both marginal and conditional distributions. This strategy involves projecting the features into a latent subspace, aligning both marginal and conditional distributions simultaneously. Thereafter, semantic information, obtained from the proposed label similarity matrix, is incorporated to embed the latent semantic structure into the training of the discriminant subspace. Finally, hash codes with high-quality are generated for data from different modalities, by efficiently embedding of semantic information and modality similarities into the discriminant latent subspace.

modalities. By mapping data from different modalities into a common, high-dimensional latent space, kernelization facilitates the extraction of shared features and simplifies the learning process.

However, kernelization alone is insufficient for fully aligning the data distributions across different modalities. Although it can capture complex non-linear relationships, the distributional discrepancy between modalities may still persist, leading to the limitation on model's ability to align the data from different modalities of the same object. Fortunately, maximum mean discrepancy (MMD) which is widely-used in the filed of cross-domain adaptation [13], [14], [15], [16] has the potential to overcome it. Specifically, MMD is able to measure the distance between the central data points (means of the data) of every two modalities. Thereafter, the difference of data distribution can be reduced by minimizing the distance of instances in the same modality while maximizing the distance of instances in the different modalities. That is, MMD effectively reduces the distributional gap between modalities while simultaneously preserving the non-linear relationships learned through kernelization. The combination of kernelization and MMD is a unique dual-strategy approach that allows our model to address both the non-linear relationship and distribution discrepancy challenges in cross-modal retrieval, enabling it to perform effectively in complex cross-modal scenarios. Thus, this paper attempts to apply the idea of MMD-based metric with kernelization and semantic alignment for hashing-based cross-modal retrieval.

There exists some challenges to provide a more efficient and accurate hashing-based cross-modal retrieval mentioned above. Firstly, heterogeneous modality gap is one of the key challenge, as the similarity of instances form different modalities cannot be measured directly. Secondly, it is an intuitive challenge to customize a MMD-based method for boosting the common space learning in hashing-based cross-modal retrieval, since the exploration of MMD in the research areas of hashing-based cross-modal retrieval is almost vacant. Furthermore, the optimization problem for minimizing the MMD loss in CSMH is also an intuitive challenge. Finally, generating high-quality hash codes by solving the binary optimization problem is also a key issue.

In order to overcome the challenges mentioned above, CSMH is proposed to provide a more efficient and effective hashing-based cross-modal retrieval method. The framework of CSMH is presented in Fig. 1. The process consists of the following key steps: Kernelization: Non-linear features of each modality are extracted using a kernelization operation, allowing for complex data relationships to be captured. Projection into Latent Space: The kernelized features are projected into a common latent space to align marginal and conditional distributions, enabling shared feature learning across modalities. MMD-Based Metric: An MMD-based strategy is applied to reduce distributional discrepancies, minimizing distances within the same modality and maximizing distances between different modalities. Embedding Semantic Information: Semantic information from the label similarity matrix is embedded to enhance the discriminative power of the learned features. Optimization and Hash Code Generation: An optimization algorithm is used to generate high-quality hash codes, ensuring that both modality and label similarities are preserved.

The contributions of this paper are summarized as follows.

- Unlike previous cross-modal hashing methods where hash codes are generated from linear embedding of original features by the common space learning, CSMH customizes a MMD-based metric strategy which is able to boost the common space learning based on the non-linear data features extracted from the kernelization operation, for improving the efficiency and precision of hashing-based cross-modal retrieval.
- The MMD-based metric strategy in CSMH projects the features into a latent subspace to simultaneously align both marginal and conditional distributions. The strategy can reduce the discrepancy of data distribution between

modalities, by shortening the distance of instances in the same modality while enlarging the distance of instances in the different modalities.
- By the semantic alignment and the MMD-based metric strategy in CSMH, both the label and modality similarities can be embedded into the features in the latent common space, which is approximated to the hash space for generating hash codes and functions with high quality.

The reminder of this paper will be organized as follows. Section II introduces related works on cross-modal hashing retrieval from supervised and unsupervised perspectives. Section III introduces CSMH in detail and section IV proposed an algorithm for solving the optimization problem in CSMH. Section VI provides records and analyses for the experimental results. Section VII concludes this paper.

II. Related Works

A. Hashing-based Cross-modal Retrieval

Cross-modal hashing normally utilizes the hashing technique to mitigate the heterogeneous modality gap, which can boost the performance of cross-modal retrieval by generating more compact and discriminative hash codes. For instance, collective matrix factorization hashing (CMFH) leverages collective matrix factorization (CMF) and a latent factor model for learning unified hash codes [17]. To fully exploit the semantic information in labels, semantic preserving hashing (SePH) minimizes the KL-divergence for narrowing the gap in semantic and makes the affinity matrix probabilistic [18]. However, with the increasing of the training instances, the computational complexity and memory overhead increase dramatically, leading to SePH without ability to handle the large-scale datasets. To relax the above constraints, supervised multi-modal hashing with semantic correlation maximization (SCM) continuously learns hash codes through using semantic label similarity [19]. Discrete cross-modal hashing (DCH) utilizes a discrete optimization model for learning the binary hash codes [20]. However, DCH only considers the situation of dealing paired multi-modal data. In order to handle the unpaired multi-modal data, generalized semantic preserving hashing (GSPH) factorizes the affinity matrix to deal different cross-modal scenarios [21]. Scalable discrete matrix factorization hashing (SCRATCH) [22] and the enhanced discrete multi-modal hashing (EDMH) [23] synchronously learn the hash codes and functions, leading to an inflexible and complex optimization. Scalable asymmetric discrete cross-modal hashing (BATCH) embeds semantic information into hash codes by the common semantic latent space learning [11]. Average approximate hashing (AAH) learns the unified hash codes by leveraging a strategy of average approximation [12]. Joint specifics and consistency hash learning (SCLCH) leveraged an asymmetric framework for fully exploiting the supervised information to learn discriminative hash codes [24]. Adaptive label correlation based asymmetric discrete hashing (ALECH) learns hash codes by adaptively exploiting the high order semantic label correlation to generate discrete hash code [25].

B. Metric Learning

Metric learning learns a distance metric as similarity measurement for instances, where the metric can practically and effectively reveal the relationship between instance. By embedding data into a suitable representation space, metric learning can capture the semantic or other useful relationships for instances accurately based on the distance metric in that space. For instance, kernel-distance metric learning it utilized in person-reidentification, which thereafter gains ability to cope with complex problem without the vector representation [26]. Transformation based fuzzy rule interpolation (TFZI) utilizes Mahalanobis distance as metric for improving the fuzzy interpolative reasoning performance [27]. Joint category compactness and disturbance reduction (CCDR) leverages the metric learning to eliminate the discrimination between the nearest samples from different category [28]. Clip-based knowledge distillation hashing (CKDH) integrates intra and inter similarities of data from different modalities as metric to distill the knowledge for mitigating the semantic gaps [29]. Except the methods mentioned above, maximum mean discrepancy (MMD) is one of the well-known and efficient methods in metric learning, and it can minimize the distribution divergence between data from different modalities. For instance, domain invariant and class discriminative (DICT) feature learning utilizes MMD to learn both domain-invariant and class-discriminative representation for features [30]. Locality preserving joint transfer (LPJT) simultaneously selects the landmark and preserves the locality structure to transfer knowledge at both feature and sample levels [31]. Deep weibull hashing (DWH) leverages MMD as metric to optimize the neighborhood structure by an error minimizing quantification, for learning hash codes with high quality [32]. In summary, MMD has attracted great interest from many researchers in the literature, as it has significant interpretability and can achieve considerable experimental results. Finally, as an effective metric, MMD has great potentials in capturing the semantic relationship between modalities for mitigating the heterogeneous modality gaps in cross-modal retrieval.

III. The Proposed Method

A. Problem Statement

This paper aims to learn hash functions for generating hash codes of data from different modalities, in which the cross-modal data with similar semantics are close in the shared latent Hamming space. Suppose that there are n instances in the training datasets, and let $\mathbf{X}^{(t)} = \{\mathbf{x}_i^{(t)}\} \in \Re^{d^{(t)} \times n}$ be the matrix of the $d^{(t)}$-dimensional data from the t-th modality, where $t \in \{1, 2, \cdots, m\}$ and m is the number of modalities. Let $\mathbf{L} \in \{0, 1\}^{c \times n}$ be the label matrix, in which c is the number of categories. $\mathbf{L}_{ij} = 1$ indicates that the j-th instance is of category i, and vice versa. Moreover, let binary codes $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n] \in \{-1, 1\}^{r \times n}$ be the hash codes, where r is the length of $\mathbf{B}$. In addition, $||\cdot||$ indicates a vector in 2-norm, while $||\cdot||_F$ indicates a matrix in F-norm. Based on the assumption mentioned above, the objective of this paper is to find an unified hash code $\mathbf{B}$ for data from various modalities, through customizing a MMD-based method to boost common

space learning. As such, similarities of instances from cross-modal can be directly obtained by calculating the distances among their hash codes.

B. Similarity Matrix Construction

In order to generate high quality hash codes, similarity matrix construction is utilized to capture the semantic relationship of data from different modalities, and thus the similarity information is able to be efficiently and accurately embedded into hash codes. However, directly using the label matrix may not embed the semantic information in the discrete hash codes well. Therefore, in this paper, we adopt the following similarity matrix by using the label matrix $\mathbf{L}$.

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

where $\mathbf{S} \in \Re^{n\times n}$, and $\mathbf{G} = \frac{\mathbf{L}}{||\mathbf{L}||}$ is the integrated matrix. Moreover, $\mathbf{1}_n$ is a column vector to fulfill with ones in n-bits length.

C. Latent Common Space Learning

In order to discover the unified hash codes satisfying for data from all modalities, CSMH attempts to learn a latent common semantic space for eliminating the inconsistency among various modalities. For this propose, a non-linear embedding is necessary for capturing the features of data from different modalities, because the linear embedding is insufficient to use the modality information efficiently when capturing the original feature. As a powerful nonlinear embedding method, guassian kernel mapping is simple and widely-used in many existing works. Specifically, in the guassian kernel mapping $\widetilde{\mathbf{X}}^{(t)}$, the sampled anchor point is set to d_t for the t-th modality data, that is $\widetilde{\mathbf{X}}^{(t)} \in \Re^{d_t\times n}$. Thereafter, the latent common space learning problem can be defined as follows.

$$\begin{aligned} &\min_{\mathbf{V}} ||\sum_{t=1}^{m}\lambda_t\mathbf{P}^{(t)^T}\widetilde{\mathbf{X}}^{(t)} - \mathbf{V}||_F^2, \\ &s.t. \quad \mathbf{P}^{(t)} \in \Re^{d_t\times r}, \widetilde{\mathbf{X}}^{(t)} \in \Re^{d_t\times n}, \\ &\qquad \sum_{t=1}^{m}\lambda_t = 1, \mathbf{V} \in \Re^{r\times n}. \end{aligned} \tag{2}$$

Note that $\mathbf{P}^{(t)}$ is a hash mapping matrix for the t-th modality and $\mathbf{V}$ is a common space projected from $\widetilde{\mathbf{X}}^{(t)}$ by $\mathbf{P}^{(t)}$. Moreover, constraint $\sum_{t=1}^{m}\lambda_t = 1$ is the balance parameter for m modalities and $||\cdot||_F^2$ indicates the Frobenius norm, in which $||\mathbf{X}||_F^2 = Tr(\mathbf{X}^T\mathbf{X})$ and $Tr(\cdot)$ is the trace of a matrix.

D. Maximum-Mean-Discrepancy Minimization

To narrow the distances between the instances in the latent common space $\mathbf{V}$ projected from the same modality. While enlarging the distances between the instances in space $\mathbf{V}$ projected from the different modalities, a distance loss in the customized MMD-based method is formulated as follows.

$$\begin{aligned} &\sum_{t=1}^{m}\sum_{\mathbf{x}_i\in\widetilde{\mathbf{X}}^{(t)}}\max_{\mathbf{x}_j\in\widetilde{\mathbf{X}}^{(t)}}||\mathbf{P}^{(t)^T}\mathbf{x}_i - \mathbf{P}^{(t)^T}\mathbf{x}_j||^2 - \\ &\qquad \min_{\mathbf{x}_k\notin\widetilde{\mathbf{X}}^{(t)}}||\mathbf{P}^{(t)^T}\mathbf{x}_i - \mathbf{P}^{(t)^T}\mathbf{x}_k||^2 \\ &s.t. \quad \mathbf{P}^{(t)} \in \Re^{d_t\times r}, \widetilde{\mathbf{X}}^{(t)} \in \Re^{d_t\times n}. \end{aligned} \tag{3}$$

The first term of the Eq. (3) is to approximate instances from the same modality by shortening their distances. Meanwhile, the second term is to discriminate instances from the different modalities by enlarging their distances. Then, Eq. (3) can be further transferred into the following trace form.

$$\begin{aligned} &\min_{\mathbf{P}^{(t)}}\sum_{t=1}^{m}Tr(\mathbf{P}^{(t)^T}\widetilde{\mathbf{X}}^{(t)}(\mathbf{W}_{same}^{(t)} - \mathbf{W}_{diff}^{(t)})\widetilde{\mathbf{X}}^{(t)^T}\mathbf{P}^{(t)}) + \\ &\qquad ||\mathbf{P}^{(t)}||_F^2, \\ &s.t. \quad \mathbf{P}^{(t)} \in \Re^{d_t\times r}, \widetilde{\mathbf{X}}^{(t)} \in \Re^{d_t\times n}. \end{aligned} \tag{4}$$

in which the second term $||\mathbf{P}^{(t)}||_F^2$ is to prevent over-fitting of $\mathbf{P}^{(t)}$. $\mathbf{W}_{same}^{(t)} \in \Re^{n\times n}$ is the MMD matrix for the t-th modality, which can shorten distances for instances of the t-th modality. Meanwhile, $\mathbf{W}_{diff}^{(t)} \in \Re^{n\times n}$ is the MMD matrix for different modalities, which can enlarge distances between the instances from the t-th modality and that from other modalities. Furthermore, these two MMD matrices can be defined by the following Eq. (5) and Eq. (6).

$$(\mathbf{W}_{same}^{(t)})_{ij} =$$

$$\begin{cases} I(\mathbf{x}_i \in \widetilde{\mathbf{X}}^{(t)}) + \\ \quad \sum\limits_{\mathbf{x}_q\in\widetilde{\mathbf{X}}^{(t)}} I(\mathbf{x}_i = \arg\max\limits_{\mathbf{x}_k\in\widetilde{\mathbf{X}}^{(t)}} ||\mathbf{P}^{(t)^T}\mathbf{x}_q - \mathbf{P}^{(t)^T}\mathbf{x}_k||^2), \\ if \quad i = j; \\ -I(\mathbf{x}_j \in \widetilde{\mathbf{X}}^{(t)}, \mathbf{x}_i = \arg\max\limits_{\mathbf{x}_k\in\widetilde{\mathbf{X}}^{(t)}} ||\mathbf{P}^{(t)^T}\mathbf{x}_j - \mathbf{P}^{(t)^T}\mathbf{x}_k||^2) \\ -I(\mathbf{x}_i \in \widetilde{\mathbf{X}}^{(t)}, \mathbf{x}_j = \arg\max\limits_{\mathbf{x}_k\in\widetilde{\mathbf{X}}^{(t)}} ||\mathbf{P}^{(t)^T}\mathbf{x}_i - \mathbf{P}^{(t)^T}\mathbf{x}_k||^2), \\ if \quad i \neq j. \end{cases} \tag{5}$$

where $I(\cdot)$ is an indicator function. $\mathbf{W}_{same}^{(t)}$ represents the distance matrix between the samples of the same category. That is, the distances of features for a pair of samples belonging to the same category will be maximized, and thereby enhancing the discrepancy between the samples with the same category. In Eq. (5), the $i = j$ part represents the diagonal elements in $\mathbf{W}_{same}^{(t)}$, i.e., the maximum similarity of the same sample. This part can be represented by maximizing the distances between them and other samples with the same category. The $i \neq j$ part represents the case of different samples, where the maximum distance between two different samples corresponding to row i and column j and belonging to the same class, will be calculated.

$$(\mathbf{W}_{diff}^{(t)})_{ij} =$$

$$\begin{cases} I(\mathbf{x}_i \in \widetilde{\mathbf{X}}^{(t)}) + \\ \quad \sum\limits_{\mathbf{x}_q\in\mathbf{X}^{(t)}} I(\mathbf{x}_i = \arg\max\limits_{\mathbf{x}_k\notin\widetilde{\mathbf{X}}^{(t)}} ||\mathbf{P}^{(t)^T}\mathbf{x}_q - \mathbf{P}^{(t)^T}\mathbf{x}_k||^2), \\ if \quad i = j; \\ -I(\mathbf{x}_j \in \widetilde{\mathbf{X}}^{(t)}, \mathbf{x}_i = \arg\max\limits_{\mathbf{x}_k\notin\widetilde{\mathbf{X}}^{(t)}} ||\mathbf{P}^{(t)^T}\mathbf{x}_j - \mathbf{P}^{(t)^T}\mathbf{x}_k||^2) \\ -I(\mathbf{x}_i \in \widetilde{\mathbf{X}}^{(t)}, \mathbf{x}_j = \arg\max\limits_{\mathbf{x}_k\notin\widetilde{\mathbf{X}}^{(t)}} ||\mathbf{P}^{(t)^T}\mathbf{x}_i - \mathbf{P}^{(t)^T}\mathbf{x}_k||^2), \\ if \quad i \neq j. \end{cases} \tag{6}$$

where $I(\cdot)$ is an indicator function. $\mathbf{W}_{diff}^{(t)}$ represents the distance matrix between samples of different category. Eq. (6) is used to calculate the minimum distance of data from different categories. In Eq. (6), the $i = j$ part represents the diagonal elements, which represents the minimum distance between the sample itself and other samples with different category. It can be used to indicate the degree of closeness between the sample itself and other samples with different categories. The part where $i \neq j$ indicates the case of samples between different categories, and their minimum distance is calculated to further highlight the differences across categories.

In summary, both the matrices are used to construct the similarity and difference between samples, in order to maximize the similarity of samples of the same category while minimizing the similarity of samples of different categories in cross-modal tasks. In this way, MMD can help align the data distribution, allowing samples with the same category to cluster in the latent space and effectively separate samples of different categories. That is, $\mathbf{W}_{same}^{(t)}$ enhances the differences between similar samples by maximizing the distance between them, enabling the model to better distinguish between similar samples. Meanwhile, $\mathbf{W}_{diff}^{(t)}$ is used to reduce the distance between samples of different categories and enhance the separation effect of heterogeneous samples. Thus, through the synergistic effect if these two matrices, the model can better learn the commonalities and differences between modal data in cross modal tasks, optimize feature alignment and classification performance.

E. Hash Codes Learning

In order to obtain the hash codes with high quality, a connection needs to be built between $\mathbf{B}$ and $\mathbf{V}$. Thus, the useful and powerful semantic information that containing in $\mathbf{V}$ can be efficiently transferred into the binary hash codes $\mathbf{B}$ by the following Eq. (7).

$$\begin{aligned} &\min_{\mathbf{B}} ||r\mathbf{S} - \mathbf{V}^T\mathbf{B}||_F^2, \\ s.t. \quad &\mathbf{V} \in \Re^{r\times n}, \mathbf{B} \in \{-1, 1\}^{r\times n}, \\ &\mathbf{V}^T\mathbf{V} = n\mathbf{I}, \mathbf{V}\mathbf{1}_n = \mathbf{0}_r. \end{aligned} \tag{7}$$

where $\mathbf{S}$ is the similarity matrix generated in Eq. (1) and $\mathbf{0}_r$ is a column zero vector with r-bits length. Notably, constraint $\mathbf{V}^T\mathbf{V} = n\mathbf{I}$ ensures that the generated hash code $\mathbf{B}$ is bit-irrelevant, as it can ensure eigenvalue of $\mathbf{B}$ is largest and the rank of $\mathbf{B}$ is full. Meanwhile, constraint $\mathbf{V}\mathbf{1}_n = \mathbf{0}_r$ ensures $\mathbf{B}$ is more balanced by avoiding $\mathbf{B}$ is full of -1 or 1. In addition, it is an asymmetric strategy to generate $\mathbf{B}$ from $\mathbf{V}$ in Eq. (7), which has been proven in [33], [34] that it is able to achieve better performance on accuracy as a more accurate representation on cross-modal queries is able to promote a similarity research better. As such, semantic information can be preserved as much as possible.

F. Objective Function

Combining Eq. (2), Eq. (4) and Eq. (7), the overall objective function of the proposed CSMH is as follows.

$$\begin{aligned} &\min_{\mathbf{P}^{(t)},\mathbf{V},\mathbf{B}} ||\sum_{t=1}^{m} \lambda_t \mathbf{P}^{(t)^T} \widetilde{\mathbf{X}}^{(t)} - \mathbf{V}||_F^2 + \alpha||\sum_{t=1}^{m} \mathbf{P}^{(t)}||_F^2 + \\ &\alpha Tr(\sum_{t=1}^{m} \mathbf{P}^{(t)^T} \widetilde{\mathbf{X}}^{(t)} (\mathbf{W}_{same}^{(t)} - \mathbf{W}_{diff}^{(t)}) \widetilde{\mathbf{X}}^{(t)^T} \mathbf{P}^{(t)}) + \\ &\beta||r\mathbf{S} - \mathbf{V}^T\mathbf{B}||_F^2. \\ s.t. \quad &\mathbf{P}^{(t)} \in \Re^{d_t\times r}, \widetilde{\mathbf{X}}^{(t)} \in \Re^{d_t\times n}, \mathbf{V}^T\mathbf{V} = n\mathbf{I}, \\ &\mathbf{V}\mathbf{1}_n = \mathbf{0}_r, \mathbf{B} \in \{-1, 1\}^{r\times n}. \end{aligned} \tag{8}$$

where α, β are the balanced parameters.

IV. Optimization

For solving the problem mentioned above, an efficient algorithm (as shown in **Algorithm 1**) is customized in this section. The process for optimizing CSMH can be described as follows.

Update $\mathbf{P}^{(t)}$: Fixed $\mathbf{V}$ and $\mathbf{B}$, $\mathbf{P}^{(t)}$ can be updated by minimizing the following equation.

$$\begin{aligned} &\min_{\mathbf{P}^{(t)}} ||\sum_{t=1}^{m} \lambda_t \mathbf{P}^{(t)^T} \widetilde{\mathbf{X}}^{(t)} - \mathbf{V}||_F^2 + \alpha||\sum_{t=1}^{m} \mathbf{P}^{(t)}||_F^2 + \\ &\alpha Tr(\sum_{t=1}^{m} \mathbf{P}^{(t)^T} \widetilde{\mathbf{X}}^{(t)} (\mathbf{W}_{same}^{(t)} - \mathbf{W}_{diff}^{(t)}) \widetilde{\mathbf{X}}^{(t)^T} \mathbf{P}^{(t)}) \\ &s.t. \quad \mathbf{P}^{(t)} \in \Re^{d_t\times r}. \end{aligned} \tag{9}$$

Then, Eq. (9) can be transferred into the trace form as follows.

$$\begin{aligned} &\min_{\mathbf{P}^{(t)}} Tr(\sum_{t=1}^{m} \lambda_t^2 \widetilde{\mathbf{X}}^{(t)} \widetilde{\mathbf{X}}^{(t)^T} \mathbf{P}^{(t)} \mathbf{P}^{(t)^T} - \\ &2\sum_{t=1}^{m} \lambda_t \mathbf{V}^T \mathbf{P}^{(t)^T} \widetilde{\mathbf{X}}^{(t)} + \mathbf{V}^T\mathbf{V} + \alpha \sum_{t=1}^{m} \mathbf{P}^{(t)^T} \mathbf{P}^{(t)} + \\ &\alpha \sum_{t=1}^{m} \mathbf{P}^{(t)^T} \widetilde{\mathbf{X}}^{(t)} (\mathbf{W}_{same}^{(t)} - \mathbf{W}_{diff}^{(t)}) \widetilde{\mathbf{X}}^{(t)^T} \mathbf{P}^{(t)}) \\ &s.t. \quad \mathbf{P}^{(t)} \in \Re^{d_t\times r}. \end{aligned} \tag{10}$$

Taking the derivative of $\mathbf{P}^{(t)}$ in Eq. (10) to zero, we can obtain the following equation.

$$\begin{aligned} &\sum_{t=1}^{m} \lambda_t^2 \widetilde{\mathbf{X}}^{(t)} \widetilde{\mathbf{X}}^{(t)^T} \mathbf{P}^{(t)} + \alpha \sum_{t=1}^{m} \mathbf{P}^{(t)} + \\ &\alpha \sum_{t=1}^{m} \widetilde{\mathbf{X}}^{(t)} (\mathbf{W}_{same}^{(t)} - \mathbf{W}_{diff}^{(t)}) \widetilde{\mathbf{X}}^{(t)^T} \mathbf{P}^{(t)} = \sum_{t=1}^{m} \lambda_t \widetilde{\mathbf{X}}^{(t)} \mathbf{V}^{\mathbf{T}} \end{aligned} \tag{11}$$

Finally, $\mathbf{P}^{(t)}$ can be updated by the following equation.

$$\begin{aligned} &\mathbf{P}^{(t)} = (\lambda_t^2 \widetilde{\mathbf{X}}^{(t)} \widetilde{\mathbf{X}}^{(t)^T} + \alpha\mathbf{I} + \\ &\alpha \widetilde{\mathbf{X}}^{(t)} (\mathbf{W}_{same}^{(t)} - \mathbf{W}_{diff}^{(t)}) \widetilde{\mathbf{X}}^{(t)^T})^{-1} (\lambda_t \widetilde{\mathbf{X}}^{(t)} \mathbf{V}^{\mathbf{T}}) \end{aligned} \tag{12}$$

Update $\mathbf{V}$: Fixed $\mathbf{P}^{(t)}$ and $\mathbf{B}$, $\mathbf{V}$ can be updated by minimizing the following equation.

$$\begin{aligned} &\min_{\mathbf{V}} ||\sum_{t=1}^{m} \lambda_t \mathbf{P}^{(t)^T} \widetilde{\mathbf{X}}^{(t)} - \mathbf{V}||_F^2 + \beta||r\mathbf{S} - \mathbf{V}^T\mathbf{B}||_F^2. \\ &s.t. \quad \mathbf{V}\mathbf{V}^T = n\mathbf{I}, \mathbf{V}\mathbf{1}_n = \mathbf{0}_r. \end{aligned} \tag{13}$$

Due to the constrained $\mathbf{V}\mathbf{V}^T = n\mathbf{I}$ on $\mathbf{V}$, the equation can be transferred to the following trace form.

$$\begin{aligned} &\max_{\mathbf{V}} \sum_{t=1}^{m} Tr(\mathbf{V}^T(\lambda_t \mathbf{P}^{(t)^T}\widetilde{\mathbf{X}}^{(t)} + r\mathbf{B}\mathbf{S}^T)). \\ &s.t. \quad \mathbf{V}\mathbf{V}^T = n\mathbf{I}, \mathbf{V}\mathbf{1}_n = \mathbf{0}_r. \end{aligned} \tag{14}$$

In order to address Eq. (14), a singular value decomposition (SVD) method $\mathbf{J}^T\mathbf{K}\mathbf{J}$, where $\mathbf{K} = \mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n^T$ and $\mathbf{J} = \sum_{t=1}^{m} \lambda_t \mathbf{P}^{(t)^T}\widetilde{\mathbf{X}}^{(t)} + r\mathbf{B}\mathbf{S}^T$, can be presented as follows.

$$\mathbf{J}^T\mathbf{K}\mathbf{J} = \begin{bmatrix}\mathbf{Q} & \hat{\mathbf{Q}}\end{bmatrix} \begin{bmatrix}\mathbf{\Omega}^2 & 0 \\ 0 & 0\end{bmatrix} \begin{bmatrix}\mathbf{Q} & \hat{\mathbf{Q}}\end{bmatrix}^T. \tag{15}$$

Let r' be the rank of matrix $\mathbf{J}^T\mathbf{K}\mathbf{J}$, $\mathbf{\Omega} \in \Re^{r'\times r'}$ is the diagonal matrix of the positive eigenvalues, $\mathbf{Q} \in \Re^{r\times r'}$ is the corresponding eigenvectors, and $\hat{\mathbf{Q}} \in \Re^{r\times(r-r')}$ is the other eigenvector corresponding to the zero eigenvalue. Let $\mathbf{Z} = \mathbf{K}\mathbf{J}^T\mathbf{Q}\mathbf{\Omega}$, and $\bar{\mathbf{Z}} \in \Re^{n\times(r-r')}$ is random and orthogonal matrix. Finally, according to [35], $\mathbf{V}$ can be updated by the following equation.

$$\mathbf{V} = \sqrt{n}\begin{bmatrix}\mathbf{Q} & \bar{\mathbf{Q}}\end{bmatrix}\begin{bmatrix}\mathbf{Z} & \bar{\mathbf{Z}}\end{bmatrix}^T, \tag{16}$$

where orthogonal matrix $\bar{\mathbf{Q}} \in \Re^{r\times(r-r')}$ can be obtained by the Gram-Schmidt orthogonalization on $\hat{\mathbf{Q}}$.

Update B: Fixed $\mathbf{P}^{(t)}$ and $\mathbf{V}$, $\mathbf{B}$ can be updated through minimizing the following equation.

$$\begin{aligned} &\min_{\mathbf{B}} \beta||r\mathbf{S} - \mathbf{V}^T\mathbf{B}||_F^2. \\ &s.t. \quad \mathbf{B} \in \{-1, 1\}^{r\times n}. \end{aligned} \tag{17}$$

Similar to updating $\mathbf{V}$, the Eq. (11) can be simplified and transferred into the trace form as follows.

$$\begin{aligned} &\max_{\mathbf{B}} Tr(r\beta\mathbf{B}^T\mathbf{V}\mathbf{S}). \\ &s.t. \quad \mathbf{B} \in \{-1, 1\}^{r\times n}. \end{aligned} \tag{18}$$

As such, the optimal solution of the objective function can be obtained as follows.

$$\mathbf{B} = sgn(r\beta\mathbf{V}\mathbf{S}), \tag{19}$$

in which $sgn(\cdot)$ denotes a sign function.

V. Hash Function Learning

A. Hash Function Learning

This paper realizes the hash function learning in a linear regression way as follows.

$$\min_{\mathbf{H}^{(t)}} ||\mathbf{B} - \sum_{t=1}^{m} \mathbf{H}^{(t)}\widetilde{\mathbf{X}}^{(t)}||_F^2 + ||\sum_{t=1}^{m} \mathbf{H}^{(t)}||_F^2, \tag{20}$$

where $\mathbf{H}^{(t)} \in \Re^{d_t\times n}$ is the hash function for the t-th modality. Taking the derivative of $\mathbf{H}^{(t)}$ in Eq. (20) to zero, we can obtain the following equation.

$$\mathbf{H}^{(t)} = \mathbf{B}\widetilde{\mathbf{X}}^{(t)^T}(\widetilde{\mathbf{X}}^{(t)^T}\widetilde{\mathbf{X}}^{(t)} + \mathbf{I})^{-1}. \tag{21}$$

Thereafter, hash code of the query data $\mathbf{X}_{query}^{(t)}$ can be obtained by the following equation.

$$\mathbf{B}_{query}^{(t)} = sgn(\mathbf{H}^{(t)}\widetilde{\mathbf{X}}_{query}^{(t)}). \tag{22}$$

Algorithm 1 The optimization of CSMH

procedure
Input: Training data $\mathbf{X}^{(t)}$, label matrix $\mathbf{L}$, hash code length r, maximum number of iterations T, parameters λ_t, d_t, α and β.
Initialization: Initializing $\mathbf{B}^{(t)}$, $\mathbf{P}^{(t)}$ and $\mathbf{V}$ by random matrices.
1. Projecting $\mathbf{X}^{(t)}$ into $\widetilde{\mathbf{X}}^{(t)}$ as the kernelized features.
2. Boosting the common space learning by customizing a MMD-based method.
while not converged do
3. Update $\mathbf{P}^{(t)}$ with Eq. (12);
4. Update $\mathbf{V}$ with Eq. (16);
5. Update $\mathbf{B}$ with Eq. (19);
End while
Return: The hash codes matrix $\mathbf{B}$.
End procedure

TABLE I
Characteristics Statistics of Four Benchmark datasets.

Statistics	Wiki	IAPRCT-12	MIRFlickr-25K	UCI	NUS-WIDE
Dataset Size	2866	20000	16738	2000	186577
Query Size	693	2000	2000	1500	1867
Retrieval Size	2173	18000	14738	500	184710
Training Size	2173	18000	14738	500	20000
Number of Categories	10	255	24	10	10
Dim. of Image Features	128	512	150	76	500
Dim. of Text Features	10	2912	500	64	1000

B. Complexity Analysis

As mentioned above, there are three steps in the training phase, namely latent space learning, maximum-mean-discrepancy minimization and hash codes learning. Specifically, for solving Eq. (12), the computational complexity is about $\mathcal{O}(T\sum_{t=1}^{m}(d_t^3 + nd_t^2 + rnd_t + rnc + nd_td^{(t)}))$, including $\mathcal{O}(\sum_{t=1}^{m} d_t^3)$ for calculating a $d_t \times d_t$ matrix from the inverse operation and $\mathcal{O}(\sum_{t=1}^{m} nd_td^{(t)})$ for the kernelization operation. For solving Eq. (16), the computational complexity is about $\mathcal{O}(mT(r^3 + r^2 + r^2n + rnc))$, including $\mathcal{O}(mT(r^3+r^2n))$ for calculating $\mathbf{J}^T\mathbf{K}\mathbf{J}$. For solving Eq. (19), the computational complexity is about $\mathcal{O}(T\sum_{t=1}^{m} r^2nd_t)$. The total computational complexity is $\mathcal{O}(T\sum_{t=1}^{m}(n(d_t^2 + d_t(r + d^{(t)}+r^2)+r(c+mr+mc))+d_t^3+mr^2(r+1)))$. Notably, T is the number of iteration, d_t is dimension of kernel features, r is the length of hash code, m is the number of modalities, $d^{(t)}$ is the dimension of original features while n is the number of training instances. Since T, d_t, r, m, $d^{(t)} \ll n$, the overall computational cost is linear with n. Thus, the proposed CSMH is highly efficient. The computational complexity is also verified by experiments on time cost in section VI-D8.

VI. Performance Analysis

A. Datasets

In order to verify the performance of CSMH, extensive experiments are conducted on four benchmark datasets (i.e., Wiki, IAPRTC-12, MIRFlickr25k and UCI Handwritten digit),

which are widely-accepted in literature of cross-modal hashing. In addition, characteristic statistics of them are summarized in Table I, and the brief introductions of them are as follows.

Wiki collected from Wikipedia, is made of 2,866 image-text pairs including 2,173 training instances and 693 query instances [36]. Each instance contains one pair of a text and a image, which are labeled by one of 10 semantic categories. The text is denoted as a 10-dimensional topic vector generated by the Latent-Dirichlet-Allocation (LDA), while the image is represented by the 128-dimensional bag of visual SIFT feature vector. Specifically, 693 pairs were selected as the query set randomly, while the left 2,173 pairs were selected as the training and retrieval sets.

IAPRTC-12 is made of 20,000 image-text pairs, and each of them is annotated by 255 labels. The text in a pair is represented as a 2,912-dimensional bag of words vector, while the image in a pair is represented as a 512-dimensional GIST feature vector. In this paper, 2,000 pairs were picked as the query set randomly, and rest of them were treated as the training and retrieval sets.

MIR-Flickr25K collected from Flickr website, is made of 25,000 instances of 24 unique categories [37]. Each of the instance is annotated by more than one of the 24 semantic labels corresponding to the categories. The text in an instance is represented as a vector whose dimension is 500, and the image in an instance is treated as an edge histogram feature vector whose dimension is 150. As in [18], textual tags appearing more than 20 times were selected, while the instances without textual tags were removed. As such, there were 16,738 instances left. In this paper, 2,000 instances were selected as the query set randomly, while rest of 14,738 instances were treated as the training and retrieval sets.

UCI Handwritten digit is made of 2,000 instances with handwritten number (0-9) features of 10 categories in multiple modalities. As in [38], image features of these instances are 76 Fourier coefficients of the character shapes, while text feature of these instance are 64 Karhunen-Loève coefficients. Finally, 1,500 instances were picked as the training set, and rest of 500 instances were treated as the training and retrieval sets.

NUS-WIDE is made of 269648 text-image pairs with 81 categories, in which each image instance is represented as an 500 dimensions SIFT vector and each text instance is represented as an 1000 dimensions binary tagged vector. For processing, samples with 10 most frequent categories in original 81 ones were selected, and therefore 186577 labeled pairs were available in the experiments.

B. Evaluation Metrics

Two cross-modal retrieval tasks are conducted in this section, namely Image-to-Text and Text-to-Image tasks. Image-to-Text (I→T) task utilizes images as queries to search texts, while Text-to-Image (T→I) task utilizes texts as queries to search images. Normally, the performances of these two tasks are evaluated through the widely-accepted Mean-Average-Precision (MAP) metric, which is the mean of Average-Precision (AP). Given a query, the AP can be calculated as

$$AP = \frac{1}{s_g}\sum_{\varepsilon=1}^{s_r} P(\varepsilon)\xi(\varepsilon), \tag{23}$$

in which s_g is the number of ground-true neighboring instances, and s_r is the number of retrieval instances. For a query, the ground-true neighboring instances can be defined as those sharing more than one common labels. Moreover, $P(\varepsilon)$ is the precision of the top ε retrieval instances. $\xi(\varepsilon) = 1$ when the ε-th retrieval instance is the ground-true neighboring instance, and vice versa. As such, the MAP can be calculated as follows.

$$MAP = \frac{1}{s_q}\sum_{m=1}^{s_q} AP(m), \tag{24}$$

in which s_q is the number of queries. Following the MAP definition, it can be seen that the larger the MAP result, the better the performance is.

Another two evaluation metrics adopted in this paper are precision-recall curve and top-N precision curve. The precision-recall curve is able to reveal the relationship between precision and recall, while the top-N precision curves can reflect the changing of precision in terms of the number for retrieved instances. Similar to MAP, for the precision-recall curves and the top-N precision curves, the larger their values, the better the performances are.

C. Baseline Methods & Implementation Details

1) Baseline Methods: In this paper, CSMH is compared with 11 classic and widely-used baseline methods which are FSH_{2017} [38], SCRATCH_{2018} [22], GSPH_{2019} [21], EDMH_{2020} [23], BATCH_{2020} [11], AAH_{2021} [12], SCLCH_{2022} [24], ALECH_{2023} [25], ROH_{2023} [39], EDH_{2024} [40] and TASPH_{2024} [41]. Notably, FSH_{2017} is an unsupervised method, and the rest of them and the proposed CSMH are supervised methods. In addition, all of them have been introduced in section II.

2) Implementation Details: All baselines in this paper were implemented and conducted carefully with the codes and the suggested parameters provided by the authors. Meanwhile, all experiments in this paper were conducted on a workstation running Windows 10 professional system with AMD Ryzen 9 5900X 12-Core CPU @ 3.70 GHz, 64GB RAM and NVIDIA GeForce RTX 3090 GPU. The IDE for running CSMH and the baseline methods is Matlab 2021a, while the experiments of deep hashing-based baselines were conducted under python 3.9.8. In addition, the version of pytorch is 1.10.1, and the version of CUDA is 11.3. The iteration number in the experiments of CSMH is set to 10 for each run.

Both ranges of parameters α and β are set to $\{0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000\}$, and the range of d_t is from 50 to 1500 in interval of 50. Moreover, only sensitivity of λ_1 is analyzed, as we only discuss two modalities (i.e., image and text) in this paper. Thereafter, the range of λ_1 is set to $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, and $\sum_{t=1}^{m}\lambda_t = 1$. Notably, parameters α, β and λ_1 were selected based on empirical testing combined

TABLE II
MAP RESULTS OF CSMH AND BASELINES ON WIKI, AND THE BEST PERFORMANCE IS IN BOLDFACE.

Task	Method	Wiki			
		16 bits	32 bits	64 bits	128 bits
I→T	FSH	0.2288	0.2513	0.2590	0.2679
	SCRATCH	0.3474	0.3539	0.3672	0.3736
	GSPH	0.3068	0.3173	0.3289	0.3307
	EDMH	0.3409	0.3684	0.3746	0.3540
	BATCH	0.3520	0.3693	0.3794	0.3852
	AAH	0.3313	0.3383	0.3480	0.3495
	SCLCH	0.3625	0.3686	0.3897	0.3956
	ALECH	0.3631	0.3689	0.3901	0.3944
	ROH	0.3634	0.3691	0.3904	0.3949
	EDH	0.3638	0.3697	0.3906	0.3951
	TASPH	0.3644	0.3705	0.3909	0.3958
	CSMH	**0.3662**	**0.3733**	**0.3921**	**0.3982**
T→I	CMFH	0.4898	0.5167	0.5312	0.5380
	SePH	0.6276	0.6557	0.6663	0.6733
	DCH	0.7419	0.7404	0.7580	0.7571
	FSH	0.2364	0.2472	0.2548	0.2568
	SCRATCH	0.6919	0.6996	0.7168	0.7157
	GSPH	0.6530	0.6703	0.6814	0.6895
	EDMH	0.6878	0.7195	0.7209	0.6985
	BATCH	0.7541	0.7633	0.7659	0.7681
	AAH	0.7121	0.7327	0.7444	0.7464
	SCLCH	0.7276	0.7477	0.7623	0.7602
	ALECH	0.7302	0.7481	0.7636	0.7644
	ROH	0.7313	0.7499	0.7545	0.7656
	EDH	0.7327	0.7506	0.7557	0.7670
	TASPH	0.7364	0.7521	0.7584	0.7682
	CSMH	**0.7545**	**0.7645**	**0.7688**	**0.7709**

TABLE III
MAP RESULTS OF CSMH AND BASELINES ON IAPRTC-12, AND THE BEST PERFORMANCE IS IN BOLDFACE.

Task	Method	IAPRTC-12			
		16 bits	32 bits	64 bits	128 bits
I→T	FSH	0.3644	0.3684	0.4299	0.4404
	SCRATCH	0.4677	0.4803	0.4976	0.4889
	GSPH	0.4488	0.4756	0.4926	0.5052
	EDMH	0.4786	0.5035	0.5243	0.5308
	BATCH	0.4799	0.5031	0.5246	0.5372
	AAH	0.4558	0.4765	0.4831	0.4912
	SCLCH	0.4758	0.5041	0.5228	0.5380
	ALECH	0.4777	0.5043	0.5204	0.5329
	ROH	0.4782	0.5004	0.5212	0.5344
	EDH	0.4705	0.4989	0.5195	0.5323
	TASPH	0.4784	0.5012	0.5201	0.5346
	CSMH	**0.4805**	**0.5040**	**0.5247**	**0.5378**
T→I	CMFH	0.3866	0.3954	0.4042	0.4110
	SePH	0.4773	0.4884	0.5048	0.5147
	DCH	0.5098	0.5251	0.5313	0.5555
	FSH	0.3619	0.3732	0.3882	0.3938
	SCRATCH	0.5498	0.5837	0.6109	0.6141
	GSPH	0.4973	0.5366	0.5624	0.5791
	EDMH	0.5870	0.6053	0.6353	0.6495
	BATCH	0.5758	0.6185	0.6504	0.6683
	AAH	0.5203	0.5567	0.5664	0.5781
	SCLCH	0.5584	0.5992	0.6328	0.6572
	ALECH	0.5752	0.6163	0.6472	0.6672
	ROH	0.5755	0.6149	0.6531	0.6734
	EDH	0.5471	0.6060	0.6397	0.6696
	TASPH	0.5626	0.6165	0.6542	0.6743
	CSMH	**0.5762**	**0.6208**	**0.6583**	**0.6765**

TABLE IV
MAP RESULTS OF CSMH AND BASELINES ON MIRFLICKR-25K, AND THE BEST PERFORMANCE IS IN BOLDFACE.

Task	Method	MIR-Flickr25K			
		16 bits	32 bits	64 bits	128 bits
I→T	FSH	0.5954	0.6029	0.6088	0.6128
	SCRATCH	0.7034	0.7105	0.7205	0.7252
	GSPH	0.6708	0.6818	0.6895	0.6949
	EDMH	0.7324	0.7372	0.7435	0.7485
	BATCH	0.7375	0.7438	0.7452	0.7494
	AAH	0.7127	0.7116	0.7230	0.7234
	SCLCH	0.7335	0.7433	0.7545	0.7564
	ALECH	0.7317	0.7356	0.7393	0.7406
	ROH	0.7363	0.7515	0.7539	0.7562
	EDH	0.7310	0.7504	0.7547	0.7566
	TASPH	0.7313	0.7519	0.7548	0.7576
	CSMH	**0.7393**	**0.7526**	**0.7559**	**0.7592**
T→I	CMFH	0.5973	0.5962	0.5963	0.5949
	SePH	0.7349	0.7481	0.7519	0.7567
	DCH	0.7474	0.7471	0.7678	0.7964
	FSH	0.6005	0.6107	0.6180	0.6222
	SCRATCH	0.7806	0.7941	0.8056	0.8171
	GSPH	0.7109	0.7370	0.7455	0.7498
	EDMH	0.8169	0.8270	0.8309	0.8361
	BATCH	0.8217	0.8274	0.8319	0.8376
	AAH	0.8171	0.8192	0.8317	0.8309
	SCLCH	0.8132	0.8296	0.8376	0.8413
	ALECH	0.8063	0.8157	0.8203	0.8234
	ROH	0.8191	0.8226	0.8304	0.8391
	EDH	0.8098	0.8219	0.8292	0.8385
	TASPH	0.8206	0.8263	0.8301	0.8392
	CSMH	**0.8229**	**0.8285**	**0.8328**	**0.8407**

TABLE V
MAP RESULTS OF CSMH AND BASELINES ON UCI HANDWRITTEN DIGIT, AND THE BEST PERFORMANCE IS IN BOLDFACE.

Task	Method	UCI Handwritten digit			
		16 bits	32 bits	64 bits	128 bits
I→T	FSH	0.6163	0.6881	0.6916	0.7195
	SCRATCH	0.8396	0.8403	0.8454	0.8435
	GSPH	0.8417	0.8461	0.8560	0.8532
	EDMH	0.7338	0.7871	0.8085	0.7939
	BATCH	0.8512	0.8619	0.8628	0.8661
	AAH	0.4667	0.4397	0.3944	0.4756
	SCLCH	0.8478	0.8560	0.8707	0.8750
	ALECH	0.8481	0.8575	0.8710	0.8754
	ROH	0.8489	0.8682	0.8713	0.8758
	EDH	0.8493	0.8694	0.8716	0.8777
	TASPH	0.8505	0.8696	0.8719	0.8783
	CSMH	**0.8598**	**0.8794**	**0.8744**	**0.8819**
T→I	CMFH	0.4695	0.4784	0.4936	0.5110
	SePH	0.9706	0.9754	0.9795	0.9762
	DCH	0.9109	0.9037	0.9189	0.9312
	FSH	0.6299	0.6962	0.7013	0.7168
	SCRATCH	0.9514	0.9554	0.9557	0.9532
	GSPH	0.9682	0.9696	0.9695	0.9712
	EDMH	0.9118	0.9145	0.9275	0.9043
	BATCH	0.9744	0.9753	0.9778	0.9789
	AAH	0.8255	0.8333	0.9171	0.9138
	SCLCH	0.9705	0.9742	0.9769	0.9780
	ALECH	0.9711	0.9748	0.9774	0.9791
	ROH	0.9713	0.9766	0.9782	0.9794
	EDH	0.9717	0.9771	0.9786	0.9797
	TASPH	0.9720	0.9789	0.9794	0.9801
	CSMH	**0.9757**	**0.9846**	**0.9828**	**0.9835**

with cross-validation, as suggested in related works on hashing methods [11], [12], [24], etc. For α and β, we fixed the intervals of d_t at 50 and of λ_1 at 0.1, then performed a grid search over a range of values to identify the optimal balance between model complexity and retrieval performance. Once the optimal α and β values were obtained, we fixed them and conducted a similar tuning process for d_t and λ_1 to finalize the settings for optimal performance. As for Wiki, the parameters for achieving the best performance are $\lambda_1 = 0.1$, $\alpha = 1$, $\beta = 0.1$ and $d_t = 1150$. As for IAPRTC-12, the parameters for achieving the best performance are $\lambda_1 = 0.1$, $\alpha = 0.1$, $\beta = 0.0001$ and $d_t = 1050$. As for MIR-Flickr25K, the parameters for achieving the best performance are $\lambda_1 = 0.4$, $\alpha = 0.1$, $\beta = 0.0001$ and $d_t = 1150$. As for UCI Handwritten digit, the parameters for achieving the best performance are $\lambda_1 = 0.2$, $\alpha = 10$, $\beta = 0.0001$ and $d_t = 850$. As for NUS-WIDE, the parameters for achieving the best performance are $\lambda_1 = 0.3$, $\alpha = 0.1$, $\beta = 1$ and $d_t = 1100$.

D. Experimental Results & Discussions

The MAP results of CSMH and all baseline methods on four datasets (i.e., Wiki, IAPRTC-12, MIR-Flickr25K and UCI Handwritten digit) are presented in Table II, III, IV, V and

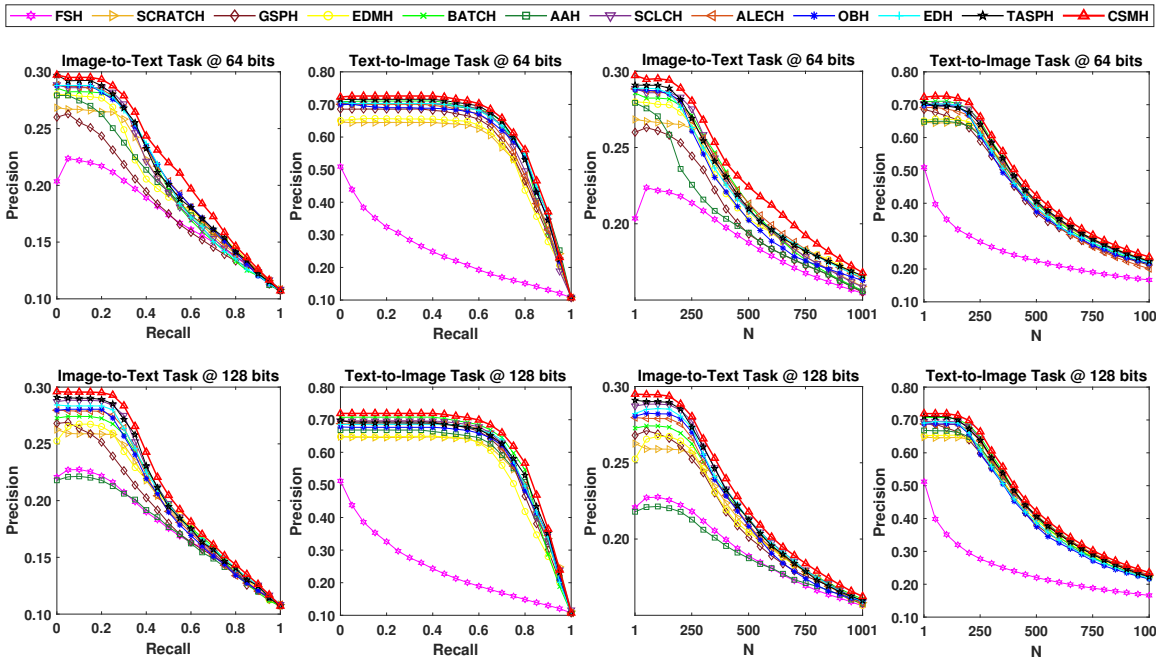


Fig. 2. Precision-Recall curves and Top-N curves with 64 bits and 128 bits hash codes of CSMH and all baseline methods on Wiki.

TABLE VI
MAP RESULTS OF CSMH AND BASELINES ON NUS-WIDE, AND THE BEST PERFORMANCE IS IN BOLDFACE.

Task	Method	NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits
I→T	FSH	0.4921	0.4998	0.5064	0.5082
	SCRATCH	0.6215	0.6322	0.6454	0.6498
	GSPH	0.5859	0.6030	0.6101	0.6124
	EDMH	0.6382	0.6515	0.6544	0.6593
	BATCH	0.6274	0.6505	0.6669	0.6685
	AAH	0.6201	0.6319	0.6409	0.6462
	SCLCH	0.6592	0.6715	0.6739	0.6816
	ALECH	0.6598	0.6724	0.6745	0.6824
	ROH	0.6683	0.6761	0.6813	0.6861
	EDH	0.6507	0.6553	0.6627	0.6809
	TASPH	0.6693	0.6776	0.6810	0.6868
	CSMH	**0.6789**	**0.6832**	**0.6866**	**0.6894**
T→I	CMFH	0.4019	0.4090	0.4135	0.4152
	SePH	0.5334	0.5437	0.5499	0.5561
	DCH	0.7189	0.7277	0.7159	0.7436
	FSH	0.5158	0.5295	0.5320	0.5297
	SCRATCH	0.7441	0.7631	0.7723	0.7757
	GSPH	0.6771	0.6958	0.7055	0.7064
	EDMH	0.7649	0.7761	0.7764	0.7825
	BATCH	0.7596	0.7782	0.7815	0.7837
	AAH	0.7273	0.7254	0.7416	0.7457
	SCLCH	0.7837	0.7915	0.8001	0.8109
	ALECH	0.7841	0.7928	0.8014	0.8112
	ROH	0.7860	0.7942	0.8047	0.8116
	EDH	0.7802	0.7996	0.8053	0.8123
	TASPH	0.7874	0.8008	0.8055	0.8124
	CSMH	**0.7997**	**0.8054**	**0.8099**	**0.8163**

VI, including I→T (Image-to-Text) and T→I (Text-to-Image) retrieval tasks. The length of the hash codes varies from 16 to 128 bits and the N in top-N precision is set from 1 to 1001, while the best MAP results are shown in boldface. Precision-recall and top-N precision curves are also presented from Fig. 2 to Fig. 5, in the case of 64 and 128 bits. Notably, it can be observed from Table II, III, IV, V and VI that, the longer the hash code, the better the MAP results are. Moreover, based on the results presented in the table and figures, we have observations as follows.

1) Results on Wiki: The MAP results of CSMH and all other baseline methods on Wiki are summarized in Table II. It can be observed from the table that, MAP results of all methods on Wiki are relative poor compared with other datasets, mainly because the extremely abstract textual features and the very low dimension of data features. Moreover, the data in Wiki is single labeled, which greatly limits its contribution on learning the more efficient hash codes. Thus, it is difficult to learn model for exploiting useful information on Wiki.

Under this circumstance, CSMH still achieves the best performance with all code lengths in both I→T and T→I retrieval tasks. The reason is that, the MMD operation reduces the data distribution difference and preserves more modality similarity, while the kernelization operation extracts the nonlinear structure of data features. This indicates that, CSMH has advantages to handle the extremely abstract textual features and the low dimensional feature, for Wiki with small scale and single-label data.

The curves of precision-recall and Top-N precision with 64 and 128 bits on Wiki are plotted in Fig. 2. It can be observed from the figure that, there is an obvious gap between the curves of CSMH and other baseline methods in terms of I→T retrieval tasks with both 64 and 128 bits. Meanwhile, the curves of CSMH is close to the baseline method BATCH in terms of T→I retrieval tasks with both 64 and 128 bits. These results are consistent with the MAP results, indicating that CSMH outperforms other baseline methods on Wiki.

2) Results on IAPRTC-12: The MAP results of CSMH and all other baseline methods on IAPRTC-12 are summarized in Table III. It can be observed from the table that, the MAP result of all methods on IAPRTC-12 is passable compared with other datasets, as the high dimension of data feature. Moreover, the label information of IAPRTC-12 is complex (255 categories), which leads to the label information to be

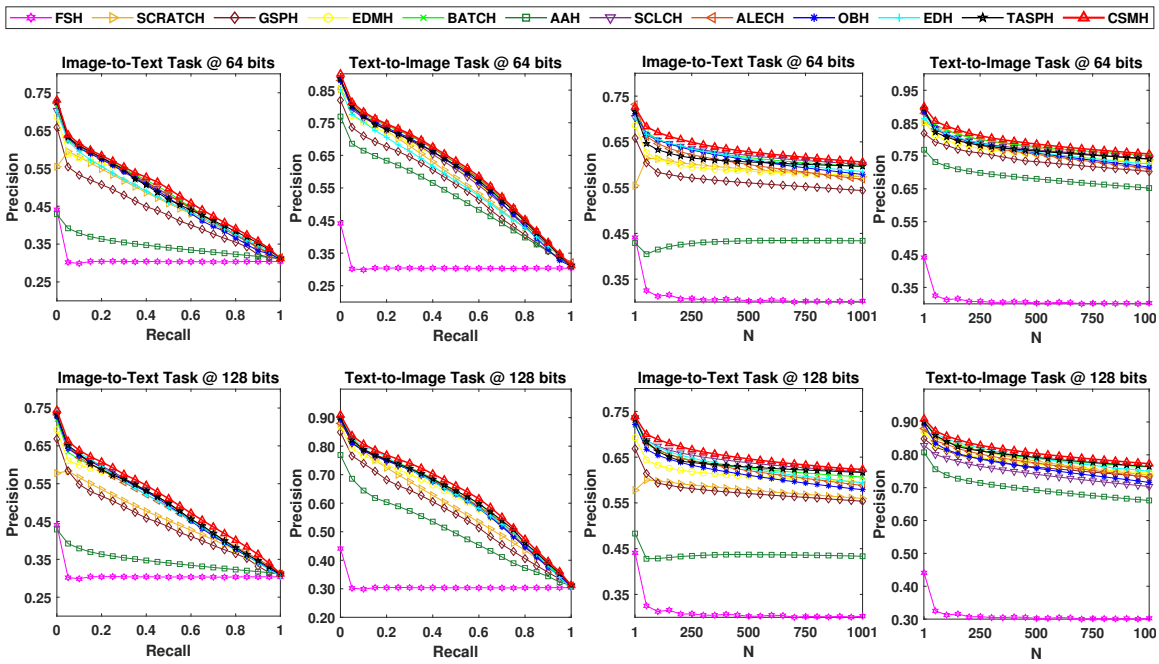


Fig. 3. Precision-Recall curves and Top-N curves with 64 bits and 128 bits hash codes of CSMH and all baseline methods on IAPRTC-12.

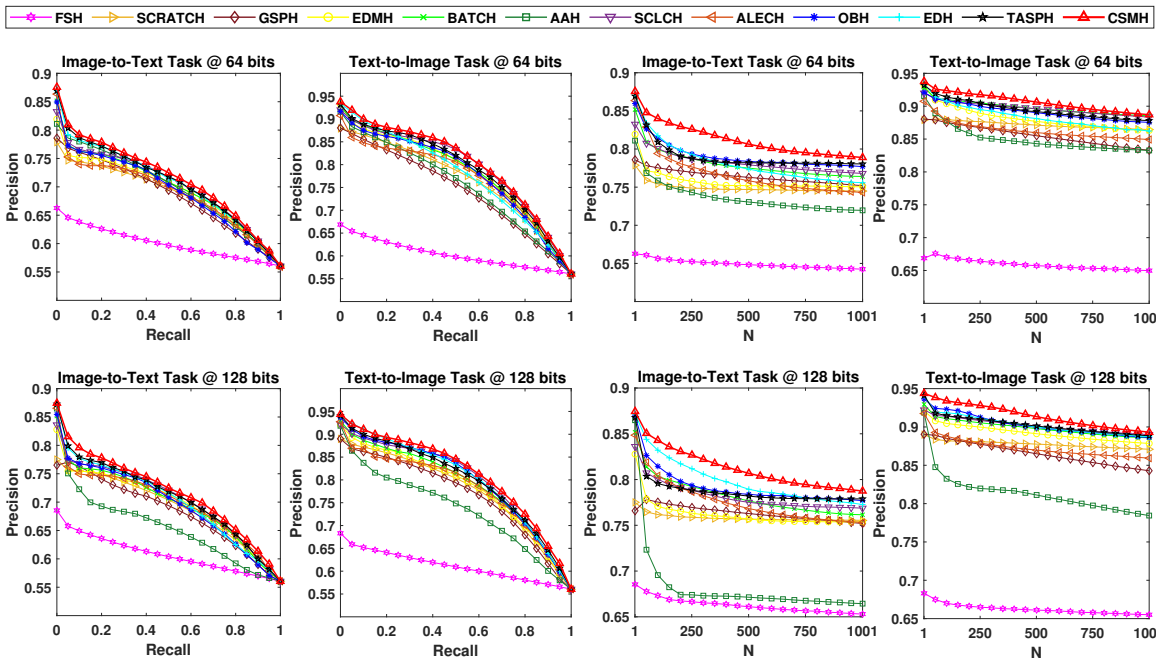


Fig. 4. Precision-Recall curves and Top-N curves with 64 bits and 128 bits hash codes of CSMH and all baseline methods on MIR-Flickr25K.

hard to be embed into the binary hash codes.

CSMH can still achieve the best performance among the baseline methods. To some extent, the kernelization operation reduces the dimension of data feature in IAPRTC-12 while the MMD operation can further improve the performance, which makes CSMH achieve the best performances in all cases. This indicates that, CSMH has advantages to handle the high dimensional data features with complex label information of large number of categories in IAPRTC-12.

The curves of precision-recall and Top-N precision with 64 and 128 bits on IAPRTC-12 are plotted in Fig. 3. It can be observed that, there is an obvious gap between the curves of CSMH and other baseline methods in terms of I→T retrieval tasks, especially with 128 bits. At the same time, the curves of CSMH are close to that of BATCH in terms of T→I retrieval tasks with both 64 and 128 bits. These results indicates that CSMH outperforms all other baseline methods on IAPRTC-12, which is consistent with the MAP results.

3) Results on MIR-Flickr25K: The MAP results of CSMH and all other baseline methods on MIR-Flickr25K are summa-

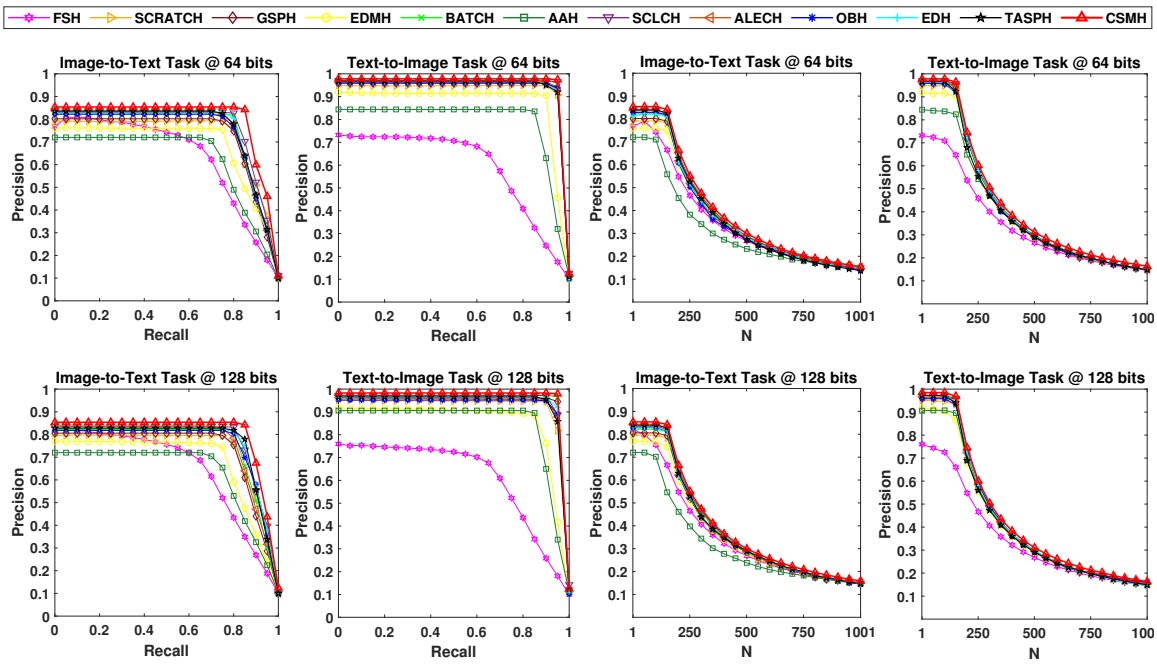


Fig. 5. Precision-Recall curves and Top-N curves with 64 bits and 128 bits hash codes of CSMH and all baseline methods on UCI Handwritten digit.

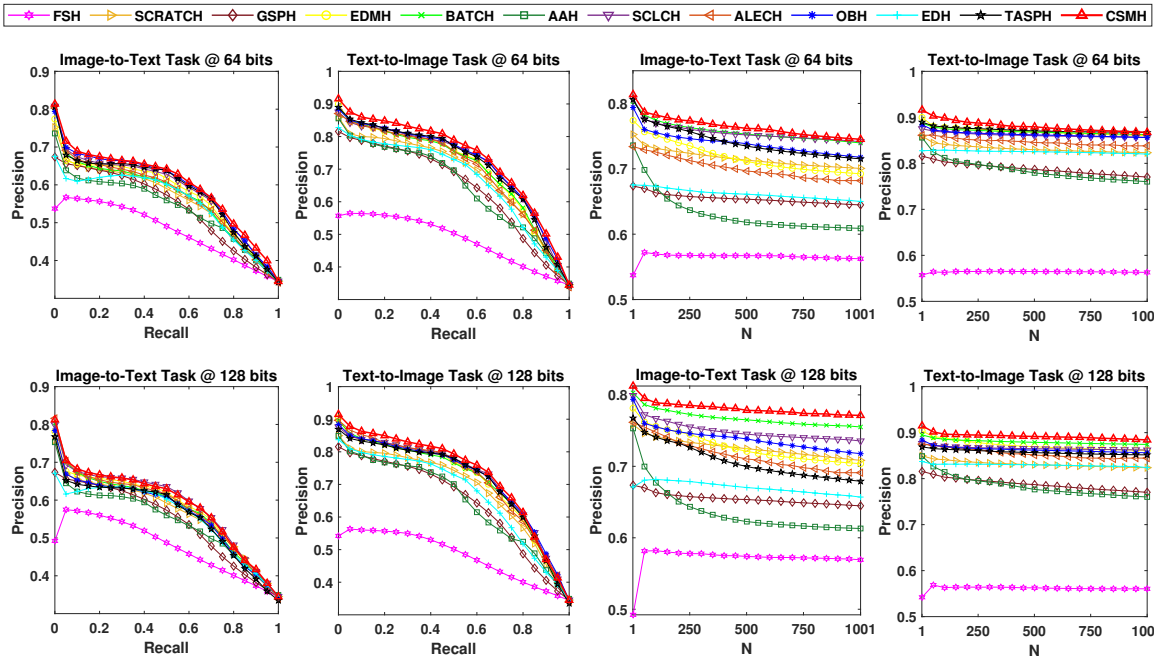


Fig. 6. Precision-Recall curves and Top-N curves with 64 bits and 128 bits hash codes of CSMH and all baseline methods on NUS-WIDE.

rized in Table IV. It can be observed that, all methods achieve mediocre MAP results on MIR-Flickr25K. A possible reason may be that the features of image and text in MIR-Flickr25K are related to each other in terms of semantic. Another one is that, it has more categories (i.e., 24 tags) than other datasets, which can be better described the complex instances.

CSMH still can achieve the best performance in all cases, mainly because the MMD operation preserves the category information and reduces distribution difference of data for each modality. To some extent, the kernelization operation also extracts the non-linear structure of data features to gain better results. In this aspect, it can better reflect the effectiveness of the MMD and kernelization operations in CSMH, as the characteristics statistics of MIR-Flickr25K are balanced and mediocre compared with other three datasets (it can be seen in Table I). This indicates that, CSMH has the advantages to handle textual features and image features with strong semantic relevance in MIR-Flickr25K.

The curves of precision-recall and Top-N precision with 64 and 128 bits on MIR-Flickr25K are plotted in Fig. 4. It can

be observed from the figure that, there has a significant gap between the curves of CSMH and all other baseline methods in both I→T and T→I retrieval tasks with both 64 and 128 bits. These results are consistent with the MAP results and indicates that CSMH outperforms all other baseline methods on MIR-Flickr25K.

4) Results on UCI Handwritten Digit: The MAP results of CSMH and all other baseline methods on UCI Handwritten digit are summarized in Table V. Almost all methods can achieve better performances on UCI Handwritten digit than other datasets. The reason is that, the dimension and size of data in UCI Handwritten digit are much smaller than other datasets, and the semantic gap between different modalities in UCI Handwritten digit are relatively small.

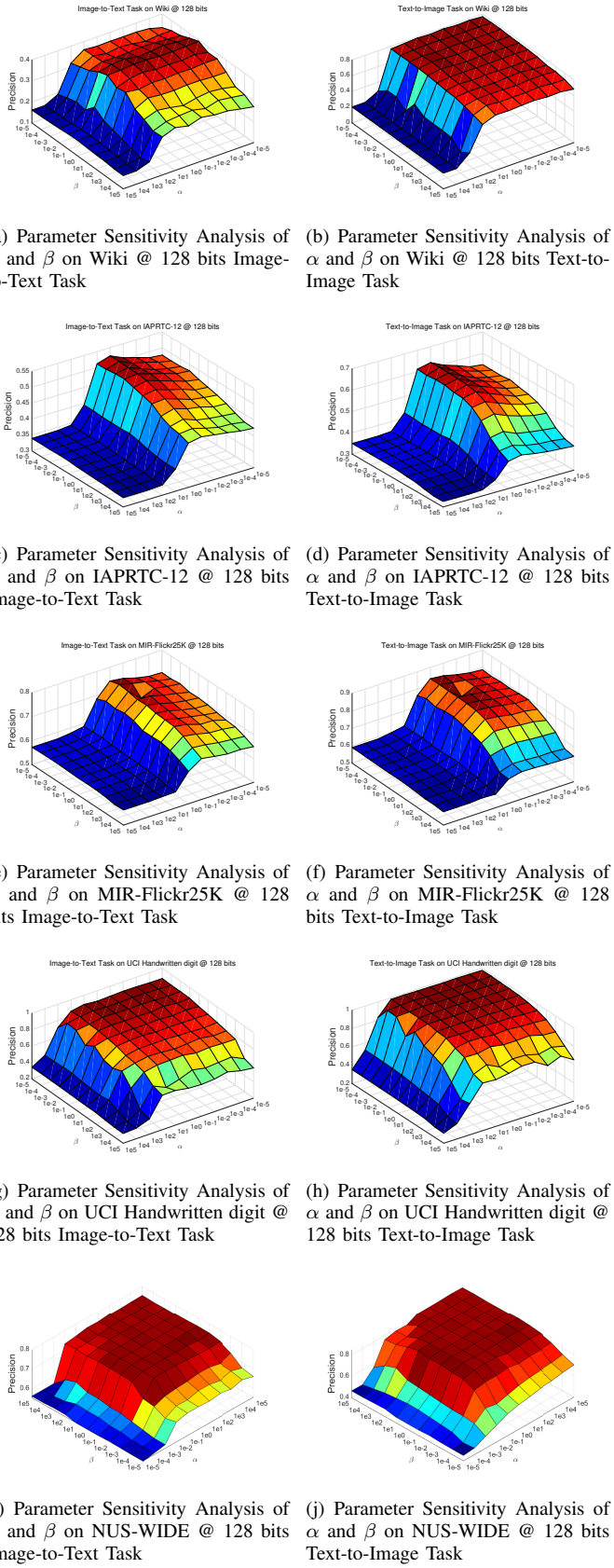
CSMH is still able to achieve the best performance in all cases. The MMD operation instead of kernelization operation makes the main contribution to the performance, as the dimension of data feature is low in UCI Handwritten digit. However, to some extent, the kernelization operation still can extract the non-linear structure of low-dimensional data in UCI Handwritten digit. In this aspect, CSMH is able to perform well on datasets with small scale like UCI Handwritten digit. This indicates that, CSMH has the advantages to handle the data with extremely low dimension and huge semantic gap in UCI Handwritten digit.

The curves of precision-recall and Top-N precision with 64 and 128 bits on UCI Handwritten digit are plotted in Fig. 5. It can be observed that, there is a significant gap between the curves of CSMH and all other baseline methods in terms of I→T retrieval tasks with 64 and 128 bits. At the same time, the curves of CSMH are close to that of baseline method BATCH in terms of T→I retrieval tasks with both 64 and 128 bits. These results indicates that CSMH outperforms all other baseline methods on UCI Handwritten digit, which is consistent with the MAP results.

5) Results on NUS-WIDE: The MAP results of CSMH and all other baseline methods on NUS-WIDE are summarized in Table VI. It can be observed that, all methods achieve better performances on T→I tasks than that on I→T tasks, in terms of MAP results. The reason may be that, the textual features in NUS-WIDE are able to describe the contents of the samples better, compared with the hand-crafted image features. A possible reason may be that the features of image and text in NUS-WIDE are related to each other in terms of semantic. Another one is that, it has more categories (i.e., 24 tags) than other datasets, which can be better described the complex instances.

CSMH achieves an obvious improvement compared with some recent baseline methods, in both I→T and T→I tasks. The reason may be that CSMH is able to excavate more intense semantic correlations from cross-modal data, by learning the sparse but essential feature representations. The results on NUS-WIDE can better reflect the effectiveness of CSMH on handling the large-scale cross-modal retrieval tasks in reality. This indicates that, CSMH has the advantages to handle the real data with more categories in NUS-WIDE, in which textual features can be better described than image features.

As shown in Fig. 6, there is an obvious gap between the



(a) Parameter Sensitivity Analysis of α and β on Wiki @ 128 bits Image-to-Text Task

(b) Parameter Sensitivity Analysis of α and β on Wiki @ 128 bits Text-to-Image Task

(c) Parameter Sensitivity Analysis of α and β on IAPRTC-12 @ 128 bits Image-to-Text Task

(d) Parameter Sensitivity Analysis of α and β on IAPRTC-12 @ 128 bits Text-to-Image Task

(e) Parameter Sensitivity Analysis of α and β on MIR-Flickr25K @ 128 bits Image-to-Text Task

(f) Parameter Sensitivity Analysis of α and β on MIR-Flickr25K @ 128 bits Text-to-Image Task

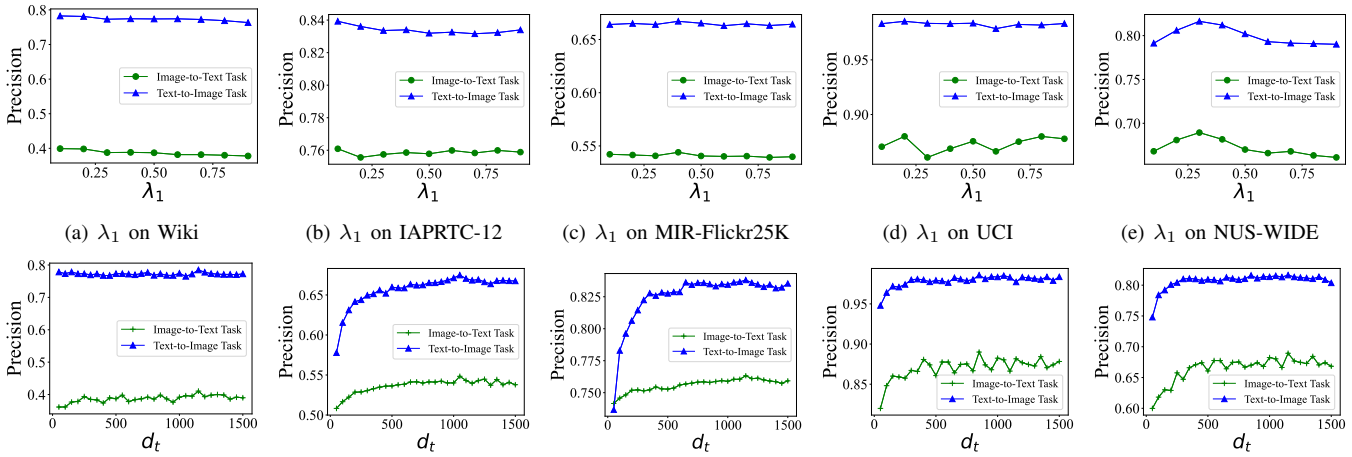
(g) Parameter Sensitivity Analysis of α and β on UCI Handwritten digit @ 128 bits Image-to-Text Task

(h) Parameter Sensitivity Analysis of α and β on UCI Handwritten digit @ 128 bits Text-to-Image Task

(i) Parameter Sensitivity Analysis of α and β on NUS-WIDE @ 128 bits Image-to-Text Task

(j) Parameter Sensitivity Analysis of α and β on NUS-WIDE @ 128 bits Text-to-Image Task

Fig. 7. Parameter sensitivity analysis of CSMH in terms of α and β with 128 bits hash codes.



(a) λ_1 on Wiki (b) λ_1 on IAPRTC-12 (c) λ_1 on MIR-Flickr25K (d) λ_1 on UCI (e) λ_1 on NUS-WIDE

(f) d_t on Wiki (g) d_t on IAPRTC-12 (h) d_t on MIR-Flickr25K (i) d_t on UCI (j) d_t on NUS-WIDE

Fig. 8. Parameter sensitivity analysis of CSMH in terms of λ_1 and d_t with 128 bits hash codes.

curves of CSMH and other baseline methods, in both I→T and T→I tasks with 64 and 128 bits. The results are consistent with that of MAP, meaning that CSMH is able to outperform all baseline methods on NUS-WIDE.

6) Parameter Sensitivity Analysis: Parameter sensitivity analysis of CSMH was conducted on all datasets with 128 bits hash codes, including I→T (Image-to-Text) and T→I (Text-to-Image) retrieval tasks. Specifically, α controls the influence of the MMD term in the objective function, and β controls the influence of common space learning and semantic embedding for hash codes learning. Moreover, λ_1 balances the influence of each modality for generating hash codes, while d_t controls the dimension of kernelization operation in CSMH.

As shown in Fig. 7, parameters α and β have significant impacts to the performance of CSMH, in both I→T and T→I tasks. It can be easily seen that CSMH achieves the best performance on Wiki with respect to $\alpha = 1$ and $\beta = 0.1$ from Fig. 7(a) and Fig. 7(b), on IAPRTC-12 with respect to $\alpha = 0.1$ and $\beta = 0.0001$ from Fig. 7(c) and Fig. 7(d), on MIR-Flickr25K with respect to $\alpha = 0.1$ and $\beta = 0.0001$ from Fig. 7(e) and Fig. 7(f), on UCI Handwritten digit with respect to $\alpha = 1$ and $\beta = 0.1$ from Fig. 7(g) and Fig. 7(h), on NUS-WIDE with respect to $\alpha = 0.1$ and $\beta = 1$ from Fig. 7(i) and Fig. 7(j).

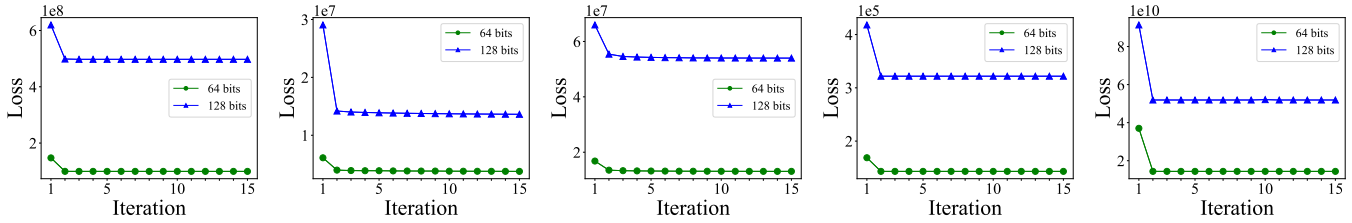
As shown in Fig. 8, the CSMH achieves the best performance with corresponding values of λ_1 (i.e., 0.2 for Wiki, 0.1 for IAPRTC-12, 0.4 for MIR-Flickr25K, 0.2 for UCI Handwritten digit and 0.3 for NUS-WIDE), although the influence of λ_1 is slight. Meanwhile, the CSMH achieves the best performance with the corresponding values of d_t (i.e., 1150 for Wiki, 1050 for IAPRTC-12, 1150 for MIR-Flickr25K, 850 for UCI Handwritten digit and 1100 for NUS-WIDE). Moreover, d_t has significant influences on CSMH performance, especially in the cases of IAPRTC-12, MIR-Flickr25K and UCI Handwritten digit.

7) Convergence Analysis: As shown in Fig. 9, experiments on convergence analysis were conducted by recording the objective function's normalized values on four datasets with 64 and 128 bits hash codes. It only shows the results in the first 15 iterations, owing to CSMH converges in all the cases very quickly.

From the figure, CSMH converges quickly in the first two iterations on datasets Wiki and UCI Handwritten digit, and also converges quickly in the first three iterations on datasets IAPRTC-12, MIR-Flickr25K and NUS-WIDE. Then, the objective function's normalized values of the proposed CSMH on all five datasets are becoming stable and convergent, indicating that the objective function of CSMH is well designed and CSMH can efficiently achieve the collaboratively semantic alignment for cross-modal hashing. Thus, the iteration number of CSMH is set to 10 in all experiments, as it is enough to reflect the stable performances of CSMH.

8) Time Cost Analysis: In section V-B, the computational complexity of CSMH is analyzed. To further verify the complexity analysis of CSMH, training time of CSMH and baseline methods on four datasets with 16 to 128 bits length of hash codes, are presented in Table VII and VIII. As shown in the tables, time cost of CSMH on IAPRTC-12 is the lowest among all baseline methods, while the time costs of CSMH on Wiki, MIR-Flickr25K, UCI Handwritten digit and NUS-WIDE are acceptable (it is in the same order of magnitude as the optimal result). Note that, the reason why all methods need a lot of time on IAPRTC-12 is that, the dimensions of image and text feature of IAPRTC-12 are highest compared with other datasets (dimension of datasets can be seen from Table I). Meanwhile, the reason why time costs of CSMH on Wiki and UCI Handwritten digit are not the lowest is that, the kernelization operation increases the dimensions of data feature.

Apparently, it can be seen from the time cost of CSMH on Wiki, MIR-Flickr25K, UCI Handwritten digit and NUS-WIDE compared with that on IAPRTC-12, the time spending on MMD and common space learning operations is acceptable and reasonable, under the premise that dimension of data features has been increased by kernelization operation. In addition, the time cost of CSMH is stable with the increase of the length of hash codes, as the kernelization operation in CSMH projects the data features into the latent common space



(a) Wiki (b) IAPRTC-12 (c) MIR-Flickr25K (d) UCI (e) NUS-WIDE

Fig. 9. Convergence curves of CSMH on five datasets in terms of 64 bits and 128 bits hash codes.

TABLE VII
TRAINING TIME (SECONDS) OF CSMH AND BASELINES ON IAPRTC-12, MIRFLICKR-25K AND NUS-WIDE WITH VARIOUS HASH CODE LENGTH.

Method	IAPRCT-12				MIR-Flickr25K				NUS-WIDE			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
FSH	95.811	102.214	103.323	104.344	10.156	10.306	11.214	11.872	18.749	22.024	25.328	27.911
SCRATCH	78.300	84.454	87.362	94.013	4.361	4.842	6.000	8.026	10.384	12.138	14.646	16.297
GSPH	419.634	810.616	1332.089	2392.494	101.462	180.740	337.064	655.981	260.812	417.421	597.706	775.465
EDMH	24.828	26.771	31.833	41.537	3.413	4.316	7.177	11.976	9.558	11.974	13.351	15.410
BATCH	68.418	72.727	80.280	83.363	3.156	3.532	4.410	6.006	9.023	11.547	12.989	14.836
AAH	1698.217	1702.894	1709.449	1725.697	1063.491	1064.226	1064.298	1075.151	1324.476	1397.615	1440.223	1487.924
SCLCH	8.420	12.801	19.836	25.455	0.929	1.287	2.174	3.135	1.996	2.213	3.544	4.232
ALECH	10.801	14.987	21.024	26.965	1.188	1.406	2.223	3.384	2.431	2.906	3.185	3.987
ROH	21.574	24.335	27.755	30.081	11.727	13.049	14.586	15.219	13.625	15.749	17.666	18.753
EDH	20.795	23.552	26.117	28.869	9.384	9.738	10.281	10.985	11.444	12.992	14.007	15.832
TASPH	8.422	22.243	89.470	388.113	2.895	12.032	53.752	277.924	3.583	13.857	83.732	390.829
CSMH	19.828	19.938	20.586	21.779	8.815	8.831	8.875	8.985	10.997	11.890	13.015	14.983

TABLE VIII
TRAINING TIME (SECONDS) OF CSMH AND BASELINES ON WIKI AND UCI HANDWRITTEN DIGIT WITH VARIOUS HASH CODE LENGTH.

Method	Wiki				UCI Handwritten digit			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
FSH	0.659	0.668	0.759	0.883	0.535	0.536	0.560	0.665
SCRATCH	0.249	0.278	0.444	0.722	0.272	0.315	0.373	0.541
GSPH	3.196	5.580	9.472	16.784	1.118	1.535	2.728	5.299
EDMH	0.562	0.585	0.849	1.273	0.178	0.321	0.351	0.959
BATCH	0.266	0.273	0.390	0.525	0.147	0.166	0.207	0.288
AAH	3.191	3.349	3.569	4.334	1.344	1.387	1.489	2.016
SCLCH	0.976	1.092	1.172	1.344	0.379	0.394	0.517	0.611
ALECH	0.988	1.134	1.205	1.389	0.408	0.442	0.558	0.634
ROH	0.957	1.087	1.161	1.291	0.483	0.498	0.511	0.587
EDH	0.833	0.991	1.086	1.178	0.491	0.509	0.597	0.623
TASPH	0.798	0.979	1.677	2.016	0.476	0.513	0.895	1.182
CSMH	0.729	0.737	0.746	0.761	0.466	0.469	0.474	0.479

TABLE IX
PRIMARY COMPUTATIONAL COMPLEXITY OF CSMH AND SOME SOTA BASELINE METHODS.

Methods	Primary Computational Complexity
SCRATCH	$\mathcal{O}(n(r^2+q^2+mqr)c)$
EDMH	$\mathcal{O}((m+n)lq+d_x^3+(m+l)d_x^2)$
BATCH	$\mathcal{O}(\sum_{l=1}^{m} k_l^2 n)$
AAH	$\mathcal{O}(dn^2T)$
SCLCH	$\mathcal{O}(n\sum_{t=1}^{m} d_t k_t)$
ALECH	$\mathcal{O}(\sum_{k=1}^{v}(nd_k^2+d_k^3))$
ROH	$\mathcal{O}(\sum_{t=1}^{\mathcal{K}}\sum_{m=1}^{o}(n_t d_m^2+d_m^3))$
EDH	$\mathcal{O}(\sum_{t=1}^{m}((r^2+k_t^2)n+k_t^3))$
TASPH	$\mathcal{O}((m^2+q^2)n)$
CSMH (ours)	$\mathcal{O}(\sum_{t=1}^{m} nd_t^2)$

with closed dimension. Although the time costs of CSMH on Wiki, MIR-Flickr25K, UCI Handwritten digit and NUS-WIDE are not the lowest, CSMH still outperforms all baseline methods on MAP, precision-recall and top-N precision.

To gain deeper insights from the complexity analysis of CSMH in subsection V-B and training time of CSMH and baselines in Table VII and VIII, we summarized the primary computational complexity of the proposed CSMH and some SOTA baseline methods as shown in Table IX. Notably, all the computational complexity of the baseline methods that was provided by their authors. The primary computational complexity indicates that the main contribution terms for causing the computation consumption. For example, in the proposed CSMH, we can have $\mathcal{O}(\sum_{t=1}^{m} nd_t^2T)$, in which the run-time of CSMH is mainly derived from the n and d_t as $n \gg d_t > T, r, m, d^{(t)}$. From Table IX, it can be found that the primary computational complexity of most SOTA baseline methods is linearly approximated to $\mathcal{O}(n)$ except one (i.e., AAH) is approximated to $\mathcal{O}(n^2)$, which is consistent with the training time recorded in Table VII and VIII. Thus, it can be concluded that the proposed CSMH delivers the best retrieval accuracy with acceptable run-time and computational complexity compared with some SOTA baseline methods.

9) Ablation Analysis: In order to gain insights of CSMH deeply, ablation experiments were conducted on several aspects, i.e., without kernelization, without common space learning and without maximum-mean-discrepancy minimization. Specifically, we dropped the corresponding modules in the proposed method and redid the optimization. Results of ablation experiments and MAP on four datasets with various code lengths can be seen in Table X and XI, in which '–' indicates dropping the corresponding module in the proposed CSMH. Specifically, *'kernelization –'* means training CSMH with original features instead of kernelled features. *'common space –'* means training CSMH without the common space learning procedure. *'MMD –'* means training CSMH without maximum-mean-discrepancy minimization. It can be

TABLE X
THE ABLATION RESULTS OF CSMH ON IAPRTC-12, MIR-FLICKR25K AND NUS-WIDE WITH VARIOUS CODE LENGTHS. THE BEST PERFORMANCE IS SHOWN IN BOLDFACE.

Task	Variants	IAPRCT-12				MIR-Flickr25K				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
I→T	kernelization –	0.4597	0.4798	0.4978	0.5123	0.7195	0.7240	0.7273	0.7290	0.6576	0.6628	0.6755	0.6781
	common space –	0.4712	0.4833	0.5092	0.5177	0.7380	0.7519	0.7581	0.7588	0.6680	0.6724	0.6759	0.6787
	MMD –	0.4588	0.4795	0.4998	0.5096	0.7377	0.7515	0.7528	0.7579	0.6493	0.6532	0.6579	0.6583
	CSMH	**0.4805**	**0.5040**	**0.5247**	**0.5378**	**0.7393**	**0.7526**	**0.7559**	**0.7592**	**0.6789**	**0.6832**	**0.6866**	**0.6894**
T→I	kernelization –	0.5569	0.5935	0.6246	0.6437	0.8222	0.8239	0.8307	0.8329	0.7784	0.7841	0.7987	0.8055
	common space –	0.5607	0.6021	0.6319	0.6572	0.8224	0.8275	0.8314	0.8393	0.7792	0.7813	0.7972	0.7997
	MMD –	0.5559	0.5923	0.6258	0.6405	0.8182	0.8273	0.8302	0.8328	0.7681	0.7757	0.7886	0.7913
	CSMH	**0.5762**	**0.6208**	**0.6583**	**0.6765**	**0.8229**	**0.8285**	**0.8328**	**0.8407**	**0.7997**	**0.8054**	**0.8099**	**0.8163**

TABLE XI
THE ABLATION RESULTS OF CSMH ON WIKI AND UCI HANDWRITTEN DIGIT WITH VARIOUS CODE LENGTHS. THE BEST PERFORMANCE IS SHOWN IN BOLDFACE.

Task	Variants	Wiki				UCI Handwritten digit			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
I→T	kernelization –	0.3107	0.3594	0.3542	0.3623	0.7718	0.7593	0.7992	0.8210
	common space –	0.3518	0.3692	0.3756	0.3835	0.8466	0.8618	0.8659	0.8777
	MMD –	0.3385	0.3472	0.3584	0.3747	0.7845	0.7900	0.8062	0.8001
	CSMH	**0.3662**	**0.3733**	**0.3921**	**0.3922**	**0.8598**	**0.8794**	**0.8744**	**0.8819**
T→I	kernelization –	0.7152	0.7294	0.7295	0.7445	0.8971	0.9261	0.9238	0.9254
	common space –	0.7486	0.7518	0.7556	0.7635	0.9641	0.9767	0.9811	0.9818
	MMD –	0.6993	0.7175	0.7294	0.7471	0.8845	0.9216	0.9078	0.9331
	CSMH	**0.7545**	**0.7645**	**0.7688**	**0.7709**	**0.9757**	**0.9846**	**0.9828**	**0.9835**

TABLE XII
THE MAP RESULTS AND TRAINING TIME (SECONDS) OF CSMH WITH CNN FEATURES AND DEEP HASHING BASELINES ON MIR-FLICKR25K. THE BEST PERFORMANCE IS SHOWN IN BOLDFACE.

Method	Dataset	MIR-Flickr25K			
	Task	16-bits	32-bits	64-bits	128-bits
DCMH	Image-to-Text	0.7376	0.7466	0.7469	0.7547
	Text-to-Image	0.7628	0.7733	0.7792	0.7857
	Time(seconds)	14589	15745	15950	16481
AGAH	Image-to-Text	0.7600	0.7847	0.7951	0.7985
	Text-to-Image	0.7498	0.7793	0.7899	0.7950
	Time(seconds)	4116	4345	4742	4899
DADH	Image-to-Text	0.8124	0.8152	0.8295	0.8331
	Text-to-Image	0.7968	0.8059	0.8102	0.8123
	Time(seconds)	13259	14072	14137	14222
SKDCH	Image-to-Text	0.8216	0.8335	0.8382	0.8403
	Text-to-Image	0.7814	0.7985	0.8067	0.7982
	Time(seconds)	11549	11981	12186	12384
CKDH	Image-to-Text	0.8276	0.8379	0.8419	0.8508
	Text-to-Image	0.8011	0.8098	0.8113	0.8116
	Time(seconds)	5068	5210	5558	5795
CSMH_{cnn}	Image-to-Text	**0.8388**	**0.8473**	**0.8512**	**0.8532**
	Text-to-Image	**0.8079**	**0.8161**	**0.8194**	**0.8215**
	Time(seconds)	**1.303**	**1.307**	**1.323**	**1.389**

observed and analyzed from the Table X and XI that,

- *'kernelization –'* indicates that the kernelization operation improves the performance to a certain extent stably, as it is able to capture non-linear structure of data features. However, it also expenses a lot of time at kernelization in the training step.
- *'common space –'* indicates that the latent common space learning operation slightly boosts the performance of CSMH, as the operation embeds more discriminated class information to the hash codes.
- *'MMD –'* has the worst performance in most of the cases, which means MMD is the most critical part in CSMH to generate high quality hash codes. Moreover, it also indicates that the MMD operation efficiently embeds the relationship among labels into the latent common space, and reduces the data distribution difference in the latent common space for each modality. In this aspect, it can further verify the effectiveness of the MMD term in objective function of CSMH.
- Training the CSMH model without the MMD minimization leads to a noticeable performance drop. The reason may be that, there is significant inherent similarity between data points from the same modal. By minimizing the discrepancy between such data, the shared characteristics of the modal are captured, which enables the hashing mapping matrix $\mathbf{P}$ to preserve essential information from the original data. As such, the well-trained matrix $\mathbf{P}$ plays a crucial role in learning consistent hashing codes in the cross-modal retrieval tasks.
- Removing the semantic alignment strategy significantly weakens the model's performance. Thus, the semantic alignment strategy we introduced is essential for reducing the inconsistency between different modalities, ensuring that the model can learn coherent representation across them.
- By combining the MMD minimization and semantic alignment strategies, the proposed CSMH can effectively capture both intra-modal and inter-modal shared information. This dual approach is crucial for generating consistent hash codes, which in turn improves retrieval performance.

In summary, the above ablation study has verified the effectiveness of each module in CSMH, indicating that CSMH can outperform all of its variants.

10) Comparison with Deep Hashing: Recently, deep learning-based methods have attracted the interests from researchers, and achieved excellent performances in the literature of cross-modal hashing. For evaluating the performance of CSMH further, CSMH is compared with four widely-used cross-modal deep hashing methods, whose methodologies are introduced briefly as follows.

- $\mathbf{DCMH}_{2017}$ [42] integrates hash codes learning and feature extraction into a framework. It is an end-to-end learning framework with deep neural networks, one for each modality, to perform feature learning from scratch. The iteration number of DCMH is 500.
- $\mathbf{AGAH}_{2019}$ [43] enhances the feature learning ability for cross-modal retrieval by employing an adversarial learning guided multi-label attention module. It utilizes

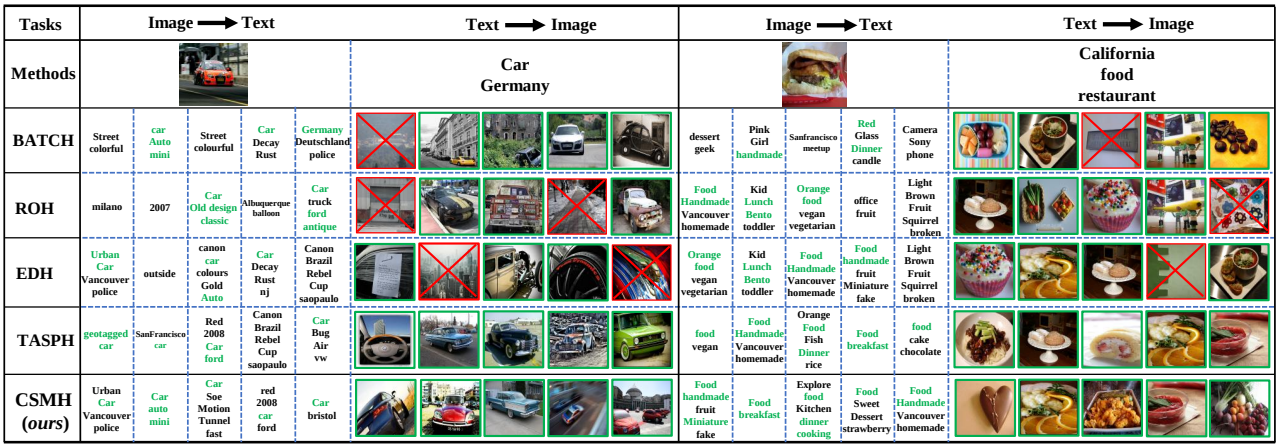


Fig. 10. Visualization of retrieved examples of CSMH and some SOTA baseline methods on MIRFlickr-25K. Images and texts with green boarders are marked as relevant, while that with red crosses are irrelevant.

a multi-label binary code map to learn hash codes better, and preserves Hamming space similarity by adopting a new triplet-margin constraint and a cosine quantization technique. The iteration number of AGAH is 300.

- **DADH**$_{2020}$ [44] learns cross-modal features and makes the distribution of feature representations consistently by using a adversarial learning method. It preserves cross-modal semantic knowledge by introducing a weighted cosine triplet constraint, and learns the hash codes by a discrete strategy. The iteration number of DADH is 300.
- **SKDCH**$_{2023}$ [45] guides a supervised method by knowledge transferred from a semi-supervised model, by making use of teacher-student optimization for propagating knowledge. It supervises student model by utilizing the extensive relevance information exploited from the outputs of the semi-supervised teacher model. The iteration number of SKDCH is 300.
- **CKDH**$_{2024}$ [29] trains a lightweight network (i.e., student network) using knowledge distillation on teacher network to make the trained student network perform well and scalable for large-scale and high-dimensional cross-modal data. The iteration number of CKDH is 300.

As in [42], the 4096-dimensional CNN features of image modality for training CSMH is extracted from the deep CNN-F network [46] which is pre-trained on ImageNet. The MAP results and training time of CSMH with CNN feature (noted as CSMH$_{cnn}$) and baselines on dateset MIR-Flickr25K can be seen in Table XII. Moreover, the results of CSMH$_{cnn}$ are the average of 10 runs with 10 iterations in each run. From Table XII, it can be seen that CSMH outperforms the deep hashing baseline methods, no matter on MAP results or training time.

Compared with deep hashing methods, CSMH$_{cnn}$ can still achieve competitive results, although it is not an end-to-end deep model. It is mainly because the well designed objective function and optimization algorithm of CSMH are significant to generate better discrete binary hash codes and functions. In this aspect, it can also verify the effectiveness of CSMH in term of the objective function and optimization algorithm.

E. Visualization of Retrieval Results

To visually showcase the superiority of the proposed CSMH in retrieval tasks, we compared CSMH with other SOTA baseline methods. Specifically, the length of hash codes is fixed to 128 bits, and the retrieval results for two image-text query pairs are demonstrated in Fig 10. The left column of each row shows the I→T retrieval results, while the right column of each row shows the T→I retrieval results. For each query pair, the top five retrieval images and texts sorted by Hamming instances will be provided. To highlight the relevant retrieval results, we marked the relevant ones with a green boarder and the irrelevant ones with a red cross.

As shown in Fig. 10, the visualization of retrieval results reveals that some SOTA baseline methods (i.e., BATCH, ROH, EDH and TASPH) deliver sub-optimal performances when handling the cross-modal retrieval tasks for user demands. The reason may be that the inherent complexity of image features leading to the simple text vectors fail to adequately represent the image features. For example, some images are closely associated with the concept of a car, but their textual description may not explicitly contain the word of 'car'. These retrieval results reveal the significant semantic gap between the text and image modalities. Thus, it is essential to minimize the semantic gaps for addressing the above-mentioned issues and improving the accuracy of sample retrieval. In this aspect, the proposed CSMH can outperform other baseline methods on retrieving semantically relevant samples to query of both two tasks. It verifies that the MMD-based metric strategy in CSMH can effectively align both marginal and conditional distribution for reducing the discrepancy of data distribution between modalities.

VII. Conclusion

This paper proposed a more efficient and accurate cross-modal hashing retrieval method named CSMH. Unlike the previous cross-modal hashing methods that directly adopting common space learning, CSMH boosts the performance of common space learning by non-linear data features extracted

from the kernelization operation. Meanwhile, CSMH can simultaneously align both marginal and conditional distributions by shortening the distance of instances in the same modality while enlarging the distance of instances in the different modalities. Finally, by aligning the semantic for cross-modal data collaboratively and customizing the MMD-based metric strategy, both the label and modality similarities can be embedded into the features in the latent common space, which is approximated to the hash space. As such, based on the above-mentioned differences or improvements, the proposed CSMH can provide a more efficient and effective hashing-based cross-modal retrieval service. Extensive experimental results of the proposed CSMH outperform all baseline methods on all five widely-used datasets in terms of MAP, precision-recall curve and top-N precision curve, verifying the effectiveness and efficiency of CSMH on hashing-based cross-modal retrieval.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62302112, Grant 62006048, and Grant 62176065; in part by the Guangdong Pearl River Talent Program under Grant 2023QN10X503; in part by the Guang-dong Provincial National Science Foundation under Grant2021A1515012017; in part by the Guangzhou Basic and Applied Basic Research Foundation under Grant 2025A04J3378; and in part by the Laboratory for Artificial Intelligence in Design (Project Code: RP3-4) under the InnoHK Research Clusters, Hong Kong SAR Government.

REFERENCES

[1] Y. Xu, Y. Bin, J. Wei, Y. Yang, G. Wang, and H. T. Shen, "Multi-modal transformer with global-local alignment for composed query image retrieval," *IEEE Transactions on Multimedia*, vol. 25, pp. 8346–8357, 2023.

[2] H. Li, Y. Bin, J. Liao, Y. Yang, and H. T. Shen, "Your negative may not be true negative: Boosting image-text matching with false negative elimination," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 924–934. [Online]. Available: https://doi.org/10.1145/3581783.3612101

[3] J. Zhang and Y. Peng, "Multi-pathway generative adversarial hashing for unsupervised cross-modal retrieval," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 174–187, 2019.

[4] Y. Bin, H. Li, Y. Xu, X. Xu, Y. Yang, and H. T. Shen, "Unifying two-stream encoders with transformers for cross-modal retrieval," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 3041–3050. [Online]. Available: https://doi.org/10.1145/3581783.3612427

[5] Z.-D. Chen, X. Luo, Y. Wang, S. Guo, and X.-S. Xu, "Fine-grained hashing with double filtering," *IEEE Transactions on Image Processing*, vol. 31, pp. 1671–1683, 2022.

[6] X. Liu, J. Yi, Y.-m. Cheung, X. Xu, and Z. Cui, "Omgh: Online manifold-guided hashing for flexible cross-modal retrieval," *IEEE Transactions on Multimedia*, vol. 25, pp. 3811–3824, 2022.

[7] Y. Shi, X. Nie, X. Liu, L. Zou, and Y. Yin, "Supervised adaptive similarity matrix hashing," *IEEE Transactions on Image Processing*, vol. 31, pp. 2755–2766, 2022.

[8] J. Qin, L. Fei, J. Zhu, J. Wen, C. Tian, and S. Wu, "Scalable discriminative discrete hashing for large-scale cross-modal retrieval," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4330–4334.

[9] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 769–790, 2017.

[10] Y. Peng, X. Huang, and Y. Zhao, "An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2372–2385, 2018.

[11] Y. Wang, X. Luo, L. Nie, J. Song, W. Zhang, and X.-S. Xu, "Batch: A scalable asymmetric discrete cross-modal hashing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 11, pp. 3507–3519, 2020.

[12] X. Fang, K. Jiang, N. Han, S. Teng, G. Zhou, and S. Xie, "Average approximate hashing-based double projections learning for cross-modal retrieval," *IEEE Transactions on Cybernetics*, vol. 52, no. 11, pp. 11 780–11 793, 2022.

[13] X. Fang, L. Jiang, N. Han, W. Sun, Y. Xu, and S. Xie, "Cross-domain recognition via projective cross-reconstruction," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 12, pp. 7366–7377, 2022.

[14] S. Li, C. H. Liu, L. Su, B. Xie, Z. Ding, C. P. Chen, and D. Wu, "Discriminative transfer feature and label consistency for cross-domain image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4842–4856, 2020.

[15] N. Han, J. Wu, X. Fang, J. Wen, S. Zhan, S. Xie, and X. Li, "Transferable linear discriminant analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5630–5638, 2020.

[16] N. Han, J. Wu, X. Fang, S. Teng, G. Zhou, S. Xie, and X. Li, "Projective double reconstructions based dictionary learning algorithm for cross-domain recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 9220–9233, 2020.

[17] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2075–2082.

[18] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3864–3872.

[19] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[20] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2494–2507, 2017.

[21] D. Mandal, K. N. Chaudhury, and S. Biswas, "Generalized semantic preserving hashing for cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 102–112, 2019.

[22] 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 *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1–9.

[23] Y. Chen, H. Zhang, Z. Tian, J. Wang, D. Zhang, and X. Li, "Enhanced discrete multi-modal hashing: More constraints yet less time to learn," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1177–1190, 2022.

[24] J. Qin, L. Fei, Z. Zhang, J. Wen, Y. Xu, and D. Zhang, "Joint specifics and consistency hash learning for large-scale cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 31, pp. 5343–5358, 2022.

[25] H. Li, C. Zhang, X. Jia, Y. Gao, and C. Chen, "Adaptive label correlation based asymmetric discrete hashing for cross-modal retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1185–1199, 2023.

[26] B. Nguyen and B. De Baets, "Kernel distance metric learning using pairwise constraints for person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 589–600, 2019.

[27] M. Zhou, C. Shang, G. Li, L. Shen, N. Naik, S. Jin, J. Peng, and Q. Shen, "Transformation-based fuzzy rule interpolation with mahalanobis distance measures supported by choquet integral," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 4, pp. 1083–1097, 2023.

[28] L. Jiang, J. Wu, S. Zhao, J. Li, and S. Ma, "Joint category compactness and disturbance reduction for cross-domain classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–14, 2024, doi: 10.1109/TIM.2024.3368421.

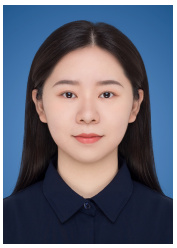
[29] J. Li, W. K. Wong, L. Jiang, X. Fang, S. Xie, and Y. Xu, "CKDH: Clip-based knowledge distillation hashing for cross-modal retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 7, pp. 6530–6541, 2024.

[30] S. Li, S. Song, G. Huang, Z. Ding, and C. Wu, "Domain invariant and class discriminative feature learning for visual domain adaptation," *IEEE transactions on image processing*, vol. 27, no. 9, pp. 4260–4273, 2018.

[31] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, "Locality preserving joint transfer for domain adaptation," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6103–6115, 2019.
[32] H. Feng, N. Wang, and J. Tang, "Deep weibull hashing with maximum mean discrepancy quantization for image retrieval," *Neurocomputing*, vol. 464, pp. 95–106, 2021.
[33] A. Gordo, F. Perronnin, Y. Gong, and S. Lazebnik, "Asymmetric distances for binary embeddings," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 33–47, 2013.
[34] C. Da, S. Xu, K. Ding, G. Meng, S. Xiang, and C. Pan, "Amvh: Asymmetric multi-valued hashing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 736–744.
[35] W. Liu, C. Mu, S. Kumar, and S. Chang, "Discrete graph hashing," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 3419–3427.
[36] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 251–260.
[37] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, 2008, pp. 39–43.
[38] H. Liu, R. Ji, Y. Wu, F. Huang, and B. Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7380–7388.
[39] K. Jiang, W. K. Wong, X. Fang, J. Li, J. Qin, and S. Xie, "Random online hashing for cross-modal retrieval," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023, doi: 10.1109/TNNLS.2023.3330975.
[40] J. Huang, P. Kang, X. Fang, N. Han, S. Xie, and H. Gao, "Efficient discriminative hashing for cross-modal retrieval," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 6, pp. 3865–3878, 2024.
[41] J. Huang, P. Kang, N. Han, Y. Chen, X. Fang, H. Gao, and G. Zhou, "Two-stage asymmetric similarity preserving hashing for cross-modal retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 1, pp. 429–444, 2024.
[42] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3232–3240.
[43] W. Gu, X. Gu, J. Gu, B. Li, Z. Xiong, and W. Wang, "Adversary guided asymmetric hashing for cross-modal retrieval," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR 2019, Ottawa, ON, Canada, June 10-13, 2019*, A. El-Saddik, A. D. Bimbo, Z. Zhang, A. G. Hauptmann, K. S. Candan, M. Bertini, L. Xie, and X. Wei, Eds. ACM, 2019, pp. 159–167.
[44] C. Bai, C. Zeng, Q. Ma, J. Zhang, and S. Chen, "Deep adversarial discrete hashing for cross-modal retrieval," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 2020, pp. 525–531.
[45] M. Su, G. Gu, X. Ren, H. Fu, and Y. Zhao, "Semi-supervised knowledge distillation for cross-modal hashing," *IEEE Transactions on Multimedia*, vol. 25, pp. 662–675, 2023.
[46] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

Jiaxing Li received the Ph.D. degree from the Guangdong University of Technology (GDUT), China, in 2021. From February 2018 to August 2018, he was an intern student, for the joint project of Big Data Integrity Verification, with Nanyang Technological University (NTU), Singapore. From October 2019 to October 2020, he worked toward the joint-Ph.D. program at GDUT and NTU. He was a postdoctoral fellow in the Hong Kong Polytechnic University (PolyU), from October 2021 to October 2023. He is currently with the School of Artificial Intelligence, Guangzhou University, China. His current research interests include cloud computing, cybersecurity, machine learning and AI security.

Wai Keung Wong received the Ph.D. degree a the Hong Kong Polytechnic University, Hong Kong SAR, in 2002. He is a full professor at The Hong Kong Polytechnic University and currently serving as the CEO & Centre Director of the Laboratory for Artificial Intelligence in Design (AiDLab). He has published over 150 scientific articles in refereed journals, including the IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Image Processing, IEEE Transactions on Cybernetics, Pattern Recognition, etc. His recent research interests include pattern recognition, feature extraction, and machine learning.



Lin Jiang received the B.Sc. degree from Chang'AnUniversity, China, in 2018, the M.Sc. degree fromGuangdong University of Technology, China, in2021, and Ph.D. degree from Guangdong Universityof Technology, China, in 2024. She is currently withSchool of Computer Science, Guangdong PolytechnicNormal University, Guangzhou 510665, China.Her current research interests include domain adaptation,pattern recognition and machine learning.

Kaihang Jiang received his M.S. degree in computer science from Guangdong University of Technology, Guangzhou, China, in 2021. He is currently working toward the PhD degree in the Hong Kong Polytechnic University. His present research interests include information retrieval, multi-modal and pattern recognition.

Xiaozhao Fang received the Ph.D. degree in computer science and technology from Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China, in 2016. He is currently with the School of Automation, Guangdong University of Technology, Guangzhou, China. His current research interests include computer vision and machine learning.

Shengli Xie (Fellow, IEEE) received the Ph.D. degree in control theory and applications from the South China University of Technology, Guangzhou, China, in 1997. He is a Full Professor with the School of Automation and the Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangdong University of Technology, Guangzhou.

Jie Wen (Member, IEEE) received the Ph.D. degree in computer science and technology from Harbin Institute of Technology, Shenzhen. He is currently an Assistant Professor with Harbin Institute of Technology. He has published over 50 technical papers at prestigious international journals and conferences, including IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Image Processing, IEEE Transactions on Cybernetics, IEEE Transactions on Multimedia, ECCV, AAAI, IJCAI, and ACM MM. His current research interests include biometrics, pattern recognition, and machine learning. More information please refer to https://sites.google.com/view/jerry-wen-hit/home.