Dual Self-Paced Cross-Modal Hashing

Yuan Sun[1,2], Jian Dai[3], Zhenwen Ren[4], Yingke Chen[5], Dezhong Peng[1,2], Peng Hu[1 *]

[1] College of Computer Science, Sichuan University, Chengdu, China
[2] National Innovation Center for UHD Video Technology, Chengdu, China
[3] Department of Automation, Tsinghua University, Beijing, China
[4] School of National Defense Science and Technology, Southwest University of Science and Technology, Mianyang, China
[5] Department of Computer and Information Sciences, Northumbria University, UK
sunyuan_work@163.com, daijian1000@163.com, rzw@njust.edu.cn, yke.chen@gmail.com, pengdz@scu.edu.cn, penghu.ml@gmail.com

Abstract

Cross-modal hashing (CMH) is an efficient technique to retrieve relevant data across different modalities, such as images, texts, and videos, which has attracted more and more attention due to its low storage cost and fast query speed. Although existing CMH methods achieve remarkable processes, almost all of them treat all samples of varying difficulty levels without discrimination, thus leaving them vulnerable to noise or outliers. Based on this observation, we reveal and study dual difficulty levels implied in cross-modal hashing learning, *i.e.*, instance-level and feature-level difficulty. To address this problem, we propose a novel Dual Self-Paced Cross-Modal Hashing (DSCMH) that mimics human cognitive learning to learn hashing from 'easy' to 'hard' in both instance and feature levels, thereby embracing robustness against noise/outliers. Specifically, our DSCMH assigns weights to each instance and feature to measure their difficulty or reliability, and then uses these weights to automatically filter out the noisy and irrelevant data points in the original space. By gradually increasing the weights during training, our method can focus on more instances and features from 'easy' to 'hard' in training, thus mitigating the adverse effects of noise or outliers. Extensive experiments are conducted on three widely-used benchmark datasets to demonstrate the effectiveness and robustness of the proposed DSCMH over 12 state-of-the-art CMH methods.

Introduction

With the rapid development of multimedia technology, cross-modal retrieval (Jing et al. 2020; Hu et al. 2023; Qin et al. 2023a) has attracted increasing attention from both academic and industrial communities. However, due to the massive growth of multimedia data (Qin et al. 2023b; Qin, Pu, and Wu 2023; He et al. 2022), continuous-value methods suffer from high storage costs and computation time. To solve this issue, cross-modal hashing (CMH) methods (Tan et al. 2022; Zhang et al. 2022; Yang et al. 2023b) have been proposed to achieve efficient performance. The essential key of CMH is to map multimodal data into discriminative binary codes while eliminating the cross-modal gap.

To learn common hash representations, numerous CMH methods (Liu et al. 2023; Tan et al. 2023; Zhu et al. 2023) have been proposed to project different modalities into a common Hamming space, which can be roughly categorized into unsupervised and supervised CMH methods. Specifically, unsupervised CMH methods often exploit data structures without semantic labels to learn hash codes. The representative methods include unsupervised contrastive cross-modal hashing (Hu et al. 2022), collective reconstructive embeddings (Hu et al. 2019a), unsupervised cross-modal hashing (Tu et al. 2023), and deep graph-neighbor coherence preserving network (Yu et al. 2021). Different from unsupervised methods, supervised CMH methods use semantic labels to learn discriminative hash codes. Benefiting from semantic information, supervised methods can learn more discriminative hash codes and achieve better performance (Hu et al. 2021, 2019b; Sun et al. 2023a).

Although these methods have achieved promising performance, most existing cross-modal hashing methods implicitly assume the collected multi-modal data (Yang et al. 2023a) is ideally clean without noise, which is hard to hold in real-world scenarios. In practice, multimodal data inevitably exist noise due to occlusion, equipment fault, and other open-world anomalies, producing noisy points or outliers. Therefore, data-driven methods will undoubtedly suffer from disturbances and get stuck into bad local minima during training, thereby remarkably degrading the performance. To tackle this problem, self-paced learning (SPL) (Kumar, Packer, and Koller 2010; Shao et al. 2022; Huang et al. 2021; Pan et al. 2020) was proposed to train the model from 'easy' to 'hard' samples inspired by human cognitive learning, which has been proven to be beneficial in alleviating the noise/outlier problem (Li et al. 2021). However, almost all CMH methods treat all instances and features equally during learning hash codes, while ignoring the difficulty differences caused by noise or outliers. Based on this observation, we expect to mitigate the effect of noise by learning from 'easy' to 'hard', thus enhancing robustness.

To achieve this, we propose a novel Dual Self-paced Cross-Modal Hashing (DSCMH) that enables our model to eliminate noise/outliers in the latent space. Our DSCMH elaborately mimics human cognitive learning that starts with easy instances and features, and gradually progresses to more difficult ones. As shown in Fig.1, our key idea is to gradually learn latent hash representations from 'easy' to 'hard' instances and features, respectively, thereby enhanc-

*Corresponding author.

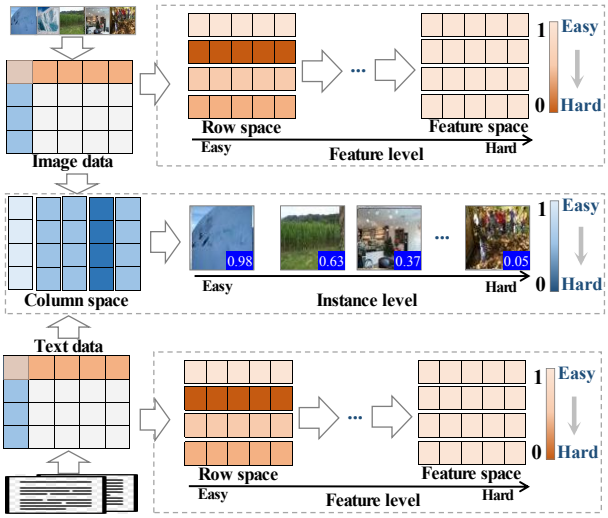


Figure 1: The frameworks of DSCMH. We perform the instance- and feature-level SPL in the row and column space, such that the consistent latent representation is learned from both 'easy' to 'hard' instances and features.

ing the robustness of dealing with noise/outliers. Specifically, DSCMH first evaluates the reliability/easiness of both instances and features dynamically in the learning process. Then, we use the dual self-paced regularize to adaptively optimize the model from 'easy' to 'hard', thereby making the model focus on reliable/easy instances and features. Finally, we adopt cross-space consensus learning to learn reliable hash codes. In general, we summarize the main contributions as follows:

- To alleviate the negative effect of noise/outliers in the learning process, we propose a hashing model with SPL to learn robust and discriminative hash codes.
- We reveal and study the instance-level and feature-level difficulty, and learn to hash from 'easy' to 'hard' in both instance- and feature-level manners. To the best of our knowledge, this could be the first work that utilizes instances- and features-level SPL for CMH.
- Extensive experiments demonstrate our DSCMH outperforms the state-of-the-art CMH methods on three widely-used benchmark datasets.

Related Work

Recently, many CMH methods are proposed to solve cross-modal retrieval. Supervised CMH methods utilize labels to guide the generation of hash codes, which generally outperform unsupervised ones. Label consistent matrix factorization hashing (LCMFH) (Wang et al. 2018) extracts the shared attributes from heterogeneous data and uses labels to preserve the semantic similarities. To improve the retrieval performance, discrete latent factor model based cross-modal hashing (DLFH) (Jiang and Li 2019) proposes a discrete scheme to directly learn hash codes without continuous relaxation. In practice, the feature dimensions of multi-modal data are often different, and using equal-length hash codes could sacrifice the representation scalability. Therefore, matrix tri-factorization hashing (MTFH) (Liu et al. 2019) makes the first attempt to learn different-length hash codes for multi-modal data. Since some methods use a large similarity matrix to preserve semantic information, they have difficulty adapting to large-scale datasets. To this end, scalable asymmetric discrete cross-modal hashing (BATCH) (Wang et al. 2021c) proposes to minimize a distance-distance difference. Considering large-scale retrieval applications and unequal hash length encoding scenarios, discrete asymmetric hashing (DAH) (Zhang et al. 2023b) proposes a flexible framework. To enhance the discrimination of hash codes, adaptive marginalized semantic hashing (AMSH) (Luo et al. 2023) proposes the adaptive margin matrices to alleviate the rigid zero-one linear regression. However, these methods unconsciously ignore the influence of noise and outliers in the hashing learning process. Robust and discrete matrix factorization hashing (RDMH) (Zhang and Wu 2022) use ℓ_{21} norm to alleviate the adverse impact of noises/outliers, thus improving robustness against noise. Besides, some research shows that SPL can resist noise interference. Prototype-supervised adversarial network (ProS-GAN) (Wang et al. 2021a) utilizes SPL to replace the previous Hamming distance loss, thereby optimizing the target adversarial samples. Cognitive multi-modal consistent hashing (CMCH) (An et al. 2022) uses SPL to achieve feature aggregation gradually and fuse multi-modal data into a common latent space. Different from these methods, we reveal the inter-instance and inter-feature differences and explore SPL in both instance and feature levels.

Methodology

Problem Definition

In this paper, suppose that $\boldsymbol{O}^t = [\boldsymbol{o}_1^t, \boldsymbol{o}_2^t, \cdots, \boldsymbol{o}_n^t] \in \mathbb{R}^{h^t \times n}$ $(t = 1, 2, \cdots, m)$ are the collected multi-modal data with n instances from m modalities, where h^t is the feature dimensionality of the t-th modality. These multi-modal data shares the same ground-truth label $\boldsymbol{Y} \in \{0, 1\}^{c \times n}$, where c is the number of the classes. For each instance, $\boldsymbol{Y}_{ij} = 1$ if the j-th data pair belongs to the i-th class and otherwise $\boldsymbol{Y}_{ij} = 0$. To better capture the nonlinear structure of multi-modal data, we use radial basis function (RBF) kernel mapping (Sun et al. 2022, 2023b) to generate kernel features. Specifically, we randomly choose d samples from each modality as anchors $\boldsymbol{a}_i^t$ and use the Gaussian kernel function to obtain the nonlinear features. Therefore, the kernel features of the t-th modality can be represented as $\boldsymbol{X}_i^t = [exp(\frac{\|\boldsymbol{o}_i^t - \boldsymbol{a}_1^t\|_2^2}{-2\sigma^2}), exp(\frac{\|\boldsymbol{o}_i^t - \boldsymbol{a}_2^t\|_2^2}{-2\sigma^2}), \cdots, exp(\frac{\|\boldsymbol{o}_i^t - \boldsymbol{a}_d^t\|_2^2}{-2\sigma^2})]^T$, where σ is the kernel width. Cross-modal hashing aims at learning l-bit hash codes $\boldsymbol{B} \in \{-1, 1\}^{l \times n}$ in Hamming space while preserving the intrinsic similarities from the original feature space. It is a great challenge how to learn high-quality hash codes from multi-modal data for cross-modal retrieval due to the impact of noise and outliers. For the sake of convenience in the presentation, we just focus on two modalities, *i.e.*, image and text.

Problem Formulation

For multi-modal data, since different modal features jointly describe the same instances, they should contain the similar or same latent representation. Therefore, we can adopt matrix factorization to excavate more hidden semantic information in the latent space. Further, we impose orthogonal and balanced constraints on the latent representation $\boldsymbol{V} \in \mathbb{R}^{l \times n}$. To be specific, to minimize the intra-modality redundancy, we adopt the orthogonal constraint to facilitate the factors of $\boldsymbol{V}$ independent. Besides, to preserve more discriminative information, we utilize the balanced constraint to guarantee $\boldsymbol{V}$ uniformly distributed. As a result, the objective function can be defined as follows:

$$\min_{\boldsymbol{U}^t, \boldsymbol{V}} \sum_{t=1}^{2} \|\boldsymbol{X}^t - \boldsymbol{U}^t \boldsymbol{V}\|_F^2 s.t. \, \boldsymbol{V}\boldsymbol{1} = \boldsymbol{0}, \boldsymbol{V}\boldsymbol{V}^T = n\boldsymbol{I}, \tag{1}$$

where $\boldsymbol{U}^t \in \mathbb{R}^{d \times l}$ is the modality-specific latent factors.

Similar to the manner of human cognitive learning, SPL gradually learns from easy concepts of the task to difficult ones, which can relieve the influence of noise or outliers. Hence, we introduce SPL into cross-modal hashing to improve robust learning ability. Different from prior SPL methods, we discover that such a cognitive mechanism is appropriate for not only the instance dimension but also the feature dimension when generating the latent space. On the one hand, easy instances and features can be helpful in learning latent representation; on the other hand, with the learning process, more and more instances and features become easy for learning, which are gradually fed into the model to train. To this end, we simultaneously perform the instance- and feature-level SPL to gradually train the hashing model from 'easy' to 'hard' until it is powerful enough to handle the complex ones. To be precise, in the training phase, the model is first trained with only easy instances and features. Then, as the model is trained, hard instances and features are gradually incorporated. Finally, the problem can be formulated as follows

$$\begin{aligned} \min_{\boldsymbol{U}^t, \boldsymbol{V}, \boldsymbol{f}^t, \boldsymbol{s}} & \sum_{t=1}^{2} \|\boldsymbol{E}_f^t (\boldsymbol{X}^t - \boldsymbol{U}^t \boldsymbol{V}) \boldsymbol{E}_s\|_F^2 + f(\eta, \boldsymbol{f}^t, \boldsymbol{s}) \\ & s.t. \, \boldsymbol{V}\boldsymbol{1} = \boldsymbol{0}, \boldsymbol{V}\boldsymbol{V}^T = n\boldsymbol{I}, \\ & \boldsymbol{E}_f^t = diag\left(\sqrt{f_1^t}, \sqrt{f_2^t}, \cdots, \sqrt{f_d^t}\right), \\ & \boldsymbol{E}_s = diag\left(\sqrt{s_1}, \sqrt{s_2}, \cdots, \sqrt{s_n}\right). \end{aligned} \tag{2}$$

Note here that $f(\eta, \boldsymbol{f}^t, \boldsymbol{s})$ is the self-paced regularizer that controls the age parameter η. η is a scalar that controls the features or instances to be selected for each learning phase to gradually incorporate more ones into the training. f_i^t and s_i are respectively the weights of i-th feature-dimension and i-th instance-dimension that estimate the reliability along different features and instances. $\boldsymbol{E}_r$ and $\boldsymbol{E}_s$ are two dynamical diagonal matrices consisting of f_i^t and s_i, respectively. To achieve learning from 'easy' to 'hard', we use the following formula (Xu, Tao, and Xu 2015; An et al. 2022)

$$f(\eta, \boldsymbol{f}^t, \boldsymbol{s}) = \sum_{t=1}^{2} f(\eta, \boldsymbol{f}^t) + f(\eta, \boldsymbol{s}), \tag{3}$$

where we have

$$\begin{aligned} f(\eta, \boldsymbol{f}^t) = & \sum_{i=1}^{d} \left(1 + e^{-\eta} - f_i^t\right) \ln\left(1 + e^{-\eta} - f_i^t\right) \\ & + f_i \ln f_i^t - \eta f_i^t, \end{aligned} \tag{4}$$

$$\begin{aligned} f(\eta, \boldsymbol{s}) = & \sum_{i=1}^{n} \left(1 + e^{-\eta} - s_i\right) \ln\left(1 + e^{-\eta} - s_i\right) \\ & + s_i \ln s_i - \eta s_i, \end{aligned} \tag{5}$$

where $f(\eta, \boldsymbol{f}^t, \boldsymbol{s})$ aims at making $\boldsymbol{E}_f^t$ and $\boldsymbol{E}_s$ gradually increase from 'low' to 'high', thereby driving instance- and feature-level learning from 'easy' to 'hard' with increasing iterations. To simulate human cognitive learning, the weight of hard instances or features should be assigned a smaller value when learning begins. The loss decreases gradually with the learning process, making the weights (*i.e.*, f^t and s) larger. Finally, all instances or features are involved in the model. Hence, we can regard the weight as the easiness/contribution of each feature and instance, thereby learning from 'easy' to 'hard' as the number of iterations increases. Note here, we will give a more theoretical analysis about f^t and s in the 'Optimization' section.

To obtain discriminative hash codes from heterogeneous data, we propose cross-space consensus learning that utilizes labels to minimize the similarities between Hamming space and latent space, thereby preserving the semantic similarities. Thus, we can easily obtain the following formula:

$$\min_{\boldsymbol{B}} \|\boldsymbol{B}^T \boldsymbol{V} - l\boldsymbol{Y}^T \boldsymbol{Y}\|_F^2 \, s.t. \, \boldsymbol{B} \in \{-1, 1\}^{l \times n}. \tag{6}$$

By combining Eq.2 and Eq.6, the following overall objective function can be written as follows

$$\begin{aligned} \min_{\boldsymbol{U}^t, \boldsymbol{V}, \boldsymbol{f}^t, \boldsymbol{s}} & \sum_{t=1}^{2} \|\boldsymbol{E}_f^t (\boldsymbol{X}^t - \boldsymbol{U}^t \boldsymbol{V}) \boldsymbol{E}_s\|_F^2 + f(\eta, \boldsymbol{f}^t, \boldsymbol{s}) \\ & + \alpha \|\boldsymbol{B}^T \boldsymbol{V} - l\boldsymbol{Y}^T \boldsymbol{Y}\|_F^2 \\ & s.t. \, \boldsymbol{V}\boldsymbol{1} = \boldsymbol{0}, \boldsymbol{V}\boldsymbol{V}^T = n\boldsymbol{I}, \boldsymbol{B} \in \{-1, 1\}^{l \times n}, \\ & \boldsymbol{E}_f^t = diag\left(\sqrt{f_1^t}, \sqrt{f_2^t}, \cdots, \sqrt{f_d^t}\right), \\ & \boldsymbol{E}_s = diag\left(\sqrt{s_1}, \sqrt{s_2}, \cdots, \sqrt{s_n}\right), \end{aligned} \tag{7}$$

where α is the trade-off parameter.

Optimization

To solve the optimization problem Eq.7, we adopt the alternative optimization algorithm that solves a variable in each iteration while fixing others. Therefore, we can convert the objective function into four sub-problems to be solved.

***s*-Step:** Fixing other variables, the sub-problem $\boldsymbol{s}$ can be updated by the following formula

$$\begin{aligned} & \min_{\boldsymbol{s}} \|\boldsymbol{E}_f^t (\boldsymbol{X}^t - \boldsymbol{U}^t \boldsymbol{V}) \boldsymbol{E}_s\|_F^2 + f(\eta, \boldsymbol{s}) \\ & s.t. \, \boldsymbol{E}_s = diag(\sqrt{s}_1, \sqrt{s}_2, \cdots, \sqrt{s}_n). \end{aligned} \tag{8}$$

To solve the weight of the i-th instance, we can simplify Eq.8 as the following problem

$$\min_{\boldsymbol{s}_i \in (0,1)} \boldsymbol{s}_i l s^i + f(\eta, \boldsymbol{s}_i). \tag{9}$$

where $ls^i = \|(\boldsymbol{E}_f^t(\boldsymbol{X}^t - \boldsymbol{U}^t\boldsymbol{V}))_{:,i}\|_F^2$ is the quantization loss of the i-th instances. Afterwards, we can obtain the solution by letting the partial derivative with regard to $\boldsymbol{s}$ to zero, *i.e.*,

$$s_i = \frac{1 + e^{-\eta}}{1 + e^{ls^i - \eta}}. \tag{10}$$

Thereupon, the weights (*i.e.*, $\boldsymbol{s} = \{s_i\}_{i=1}^n$) of all instances can be solved.

$\boldsymbol{f}^t$-Step: Fixing other variables, the sub-problem $\boldsymbol{r}$ can be reduced as

$$\begin{aligned} &\min_{\boldsymbol{f}^t} \|\boldsymbol{E}_f^t(\boldsymbol{X}^t - \boldsymbol{U}^t\boldsymbol{V})\boldsymbol{E}_s\|_F^2 + f(\eta, \boldsymbol{f}^t) \\ &s.t.\ \boldsymbol{E}_f^t = diag\left(\sqrt{f_1^t}, \sqrt{f_2^t}, \cdots, \sqrt{f_d^t}\right). \end{aligned} \tag{11}$$

By setting the partial derivative with regard to $\boldsymbol{f}^t$ to zero, we have

$$f_i^t = \frac{1 + e^{-\eta}}{1 + e^{lf_i^t - \eta}}, \tag{12}$$

where $lf_i^t = \|((\boldsymbol{X}^t - \boldsymbol{U}^t\boldsymbol{V})\boldsymbol{E}_s)_{i,:}\|_F^2$ is the quantization loss of the i-th feature. Thereupon, the weights (*i.e.*, $\boldsymbol{f}^t = \{f_i^t\}_{i=1}^d$) of all features can be solved.

We can easily observe that the values of f_i^t and s_i are always between zero and one, *i.e.*, $f_i^t \in [0, 1]$ and $s_i \in [0, 1]$. η controls the rate of change of weight relative to the loss. Eq.10 and Eq.12 assign each instance and each feature the probabilities of being 'easy', respectively. In other words, we can use weights as a measure of easiness. Thus, if the weights are higher, the instance or feature can be viewed as easier. To be specific, when $ls_i < \eta$ or $lf_i^t < \eta$, we implicitly regard the i-th instance and i-th feature as 'easy', otherwise, viewed as 'hard'. It indicates that $ls_i = \eta$ or $lf_i^t = \eta$ is the threshold to divide 'easy' and 'hard' instances or features. Moreover, for a given fixed η, the weights will vary rapidly within a certain interval, which could affect hard instances or features. Afterward, we normalize each loss ls_i or lf_i^t to control the range of values, thereby keeping the loss of each instance or feature with the rapidly varying interval for η. Hence, we can describe as follows:

$$\begin{aligned} ls_i &:= \frac{m * ls_i}{\max\{ls_1, ls_2, \ldots, ls_n\}}, \\ lf_i^t &:= \frac{m * lf_i^t}{\max\{lf_1^t, lf_2^t, \ldots, lf_d^t\}} \end{aligned} \tag{13}$$

where m is a constant. To speed up the changes of the weights, we set $\eta = r * \eta$. From Eq.10 and Eq.12, we can observe that s_i and r_i^t are proportional to η. In other words, the weight will increase as η increases until approaching one, thereby paying more attention to hard instances and features. Thanks to the property, our model could gradually consider more hard samples from 'easy' to 'hard' to prevent fitting on noise or outliers first, thus embracing stronger robustness.

$\boldsymbol{U}^t$-Step: We fix the remaining variables to solve $\boldsymbol{U}^t$. The Eq.7 can be simplified as

$$\min_{\boldsymbol{U}^t} \|\boldsymbol{E}_f^t(\boldsymbol{X}^t - \boldsymbol{U}^t\boldsymbol{V})\boldsymbol{E}_s\|_F^2. \tag{14}$$

Afterwards the Eq.14 can be converted into the following trace form

$$\min_{\boldsymbol{U}^t} Tr((\boldsymbol{E}_f^t(\boldsymbol{X}^t - \boldsymbol{U}^t\boldsymbol{V})\boldsymbol{E}_s)(\boldsymbol{E}_f^t(\boldsymbol{X}^t - \boldsymbol{U}^t\boldsymbol{V})\boldsymbol{E}_s)^\top). \tag{15}$$

Then we set the derivative of 15 w.r.t. $\boldsymbol{U}^t$ to zero, and the solution can be obtained by

$$\boldsymbol{U}^t = \boldsymbol{X}^t\boldsymbol{E}_{s^2}\boldsymbol{V}^T(\boldsymbol{V}\boldsymbol{E}_{s^2}\boldsymbol{V}^T)^{-1}, \tag{16}$$

where $\boldsymbol{E}_s^2 = \boldsymbol{E}_s\boldsymbol{E}_s^T$.

$\boldsymbol{V}$-Step: We fix the other variables except $\boldsymbol{V}$, and Eq.7 can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{V}} Tr(\boldsymbol{V}(\boldsymbol{E}_{s^2}(\boldsymbol{X}^t)^T\boldsymbol{E}_{f^2}^t\boldsymbol{U}^t)) \\ &s.t.\ \boldsymbol{V}\boldsymbol{1} = \boldsymbol{0}, \boldsymbol{V}\boldsymbol{V}^T = n\boldsymbol{I} \end{aligned} \tag{17}$$

where $\boldsymbol{E}_{f^2}^t = \boldsymbol{E}_f^t(\boldsymbol{E}_f^t)^T$.

Let $\boldsymbol{Z} = \boldsymbol{E}_{s^2}(\boldsymbol{X}^t)^T\boldsymbol{E}_{f^2}^t\boldsymbol{U}^t$, Eq.17 can be transformed as $\max tr(\boldsymbol{Z}^T\boldsymbol{V})$. We adopt the approximate maximization algorithm (Liu et al. 2014) to update the solution $\boldsymbol{V} = \sqrt{d}[\boldsymbol{K}\ \bar{\boldsymbol{K}}][\boldsymbol{D}\ \bar{\boldsymbol{D}}]^T$. Here, we can use SVD to update D and $\bar{D}$, *i.e.*, $Z^TJZ = [\boldsymbol{D}\ \bar{\boldsymbol{D}}]\begin{bmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{bmatrix}[\boldsymbol{D}\ \bar{\boldsymbol{D}}]^T$, where $\boldsymbol{J} = \boldsymbol{I} - \frac{1}{d}\boldsymbol{1}\boldsymbol{1}^T$. For $\bar{\boldsymbol{K}}$, we use the Gram-Schmidt process (Björck 1994) to define it as a random orthogonal matrix, and obtain $\boldsymbol{K} = \boldsymbol{J}\boldsymbol{Z}\boldsymbol{D}\Sigma^{-1}$.

$\boldsymbol{B}$-Step: We fix the other variables and easily get the closed-form solution by the following formula

$$\boldsymbol{B} = sign(\alpha l\boldsymbol{V}\boldsymbol{Y}^T\boldsymbol{Y}). \tag{18}$$

Hash Function Learning

Our DSCMH main has two stages in the training phase, including hash codes learning and hash functions learning. Generally, to balance the accuracy and the efficiency, we learn hash codes and hash function separately, thereby endowing hash function more flexibility. Specifically, given the learned hash codes $\boldsymbol{B}$ and the kernel features $\boldsymbol{X}^t$, we can calculate the hash function for each modality by the following problem, *i.e.*,

$$\min_{\boldsymbol{W}^t} \sum_{t=1}^{2} \|\boldsymbol{B} - \boldsymbol{W}^t\boldsymbol{X}^t\|_F^2 + \lambda\|\boldsymbol{W}^t\|_F^2 \tag{19}$$

where λ is a trade-off parameter. Thereupon, we can obtain $\boldsymbol{W}^t$ as follows

$$\boldsymbol{W}^t = \boldsymbol{B}(\boldsymbol{X}^t)^T(\boldsymbol{X}^t(\boldsymbol{X}^t)^T + \lambda\boldsymbol{I})^{-1}). \tag{20}$$

Further, we can obtain hash codes $\boldsymbol{H}^t$ of query multi-modal data $\boldsymbol{Q}^t$ by hash functions, *i.e.*,

$$\boldsymbol{H}^t = sign(\boldsymbol{W}^t\boldsymbol{Q}^t). \tag{21}$$

Hence, we can use Hamming distance to calculate the similarities between training data and query data, thereby achieving cross-modal retrieval.

Task	Method	MIRFlickr				IAPR-TC12				NUS-WIDE			
		8	16	32	64	8	16	32	64	8	16	32	64
I → T	RFDH	57.53	58.14	57.78	58.07	35.32	44.85	45.53	45.83	34.54	47.33	57.76	58.32
	LCMFH	68.08	69.73	69.28	70.27	32.69	42.73	44.70	45.69	55.52	63.18	64.21	64.87
	DLFH	71.24	75.56	76.20	76.62	44.90	48.29	50.27	54.19	61.72	62.13	64.68	66.17
	MTFH	67.40	71.09	71.28	73.43	47.14	48.32	50.45	51.98	/	/	/	/
	FCMH	74.48	75.22	76.16	76.12	46.25	49.48	51.70	53.20	64.66	65.82	66.40	67.20
	FDDH	72.91	73.00	76.13	75.83	44.37	48.04	52.29	53.89	59.75	62.07	65.79	68.60
	BATCH	74.01	75.73	75.88	76.43	44.91	48.05	50.40	52.62	63.17	65.72	66.49	67.47
	EDMH	74.26	75.07	75.59	76.08	46.39	49.86	50.85	52.43	64.56	65.83	67.16	67.44
	DAH	72.47	74.60	75.48	75.74	43.40	44.72	48.15	52.01	62.63	63.58	66.29	66.31
	ALECH	73.90	75.34	76.14	76.62	45.68	48.30	50.35	52.07	65.02	66.08	67.85	68.22
	WASH	73.01	74.35	74.53	74.64	46.75	48.25	51.00	53.45	62.45	64.04	64.18	63.34
	AMSH	74.06	75.39	76.49	76.96	46.81	49.05	51.62	53.66	64.63	65.37	67.60	67.34
	DSCMH	**76.68**	**77.35**	**78.03**	**78.76**	**49.34**	**52.40**	**54.81**	**56.75**	**66.80**	**67.72**	**68.86**	**69.26**
T → I	RFDH	58.07	58.25	58.03	57.06	34.83	45.52	46.40	57.54	35.48	53.66	58.22	62.73
	LCMFH	72.97	75.31	75.40	76.95	34.69	49.86	53.68	56.42	58.43	67.08	72.23	73.64
	DLFH	77.25	80.36	80.50	81.42	46.24	50.44	54.85	63.01	67.75	70.50	73.02	75.27
	MTFH	74.62	79.19	80.01	80.54	52.27	57.36	60.92	62.33	/	/	/	/
	FCMH	79.56	80.16	81.93	82.38	53.47	58.50	61.92	65.13	75.57	77.64	78.84	80.76
	FDDH	76.87	77.60	80.92	81.38	49.33	55.16	61.14	65.00	70.20	74.79	77.98	81.58
	BATCH	79.61	80.31	81.75	82.44	52.75	57.77	61.85	64.88	76.57	77.58	79.41	80.20
	EDMH	80.22	80.84	81.59	82.12	53.61	58.70	60.53	63.53	73.12	78.50	79.61	79.64
	DAH	78.63	78.85	81.03	81.63	49.80	54.75	58.17	61.17	73.82	77.45	78.05	79.09
	ALECH	79.15	80.43	81.73	82.02	52.55	57.74	61.44	64.61	76.26	77.64	78.89	79.77
	WASH	77.42	78.74	79.55	79.66	50.89	54.25	61.50	65.02	73.31	77.70	80.39	**81.09**
	AMSH	80.31	81.36	82.43	83.07	53.89	58.87	62.98	66.32	77.05	78.46	80.12	80.83
	DSCMH	**81.41**	**82.29**	**83.30**	**83.59**	**55.14**	**60.43**	**64.48**	**67.21**	**79.28**	**80.11**	**80.95**	80.99

Table 1: The mAP results (%) with different bit lengths on the three datasets. The best results are in bold.

Method	I → T				T → I			
	8	16	32	64	8	16	32	64
RFDH	54.92	55.63	55.79	55.80	57.10	57.38	57.45	57.60
LCMFH	57.98	58.02	58.57	58.27	70.54	73.80	76.20	77.52
DLFH	60.54	61.32	62.05	63.12	71.45	75.67	76.59	78.13
MTFH	59.80	60.02	60.15	61.03	70.84	73.54	75.01	76.03
FCMH	60.14	61.34	62.53	62.49	71.74	76.79	79.34	79.56
FDDH	63.10	63.25	63.61	63.75	73.69	74.05	76.02	76.02
BATCH	60.21	60.62	60.73	61.12	79.53	80.02	81.63	82.10
EDMH	62.94	63.06	63.17	64.21	79.47	80.03	81.68	82.37
DAH	59.12	59.43	59.55	60.52	76.80	79.28	79.42	80.93
ALECH	59.50	59.78	60.06	60.24	79.08	80.81	81.63	81.97
WASH	60.05	60.05	60.85	60.90	78.28	79.68	80.62	80.99
AMSH	59.55	60.17	61.17	61.76	78.83	80.61	82.43	82.34
DSCMH	**66.27**	**66.38**	**66.23**	**66.76**	**80.00**	**82.03**	**82.59**	**82.90**

Table 2: The mAP results (%) with noise on MIRFlickr. The best results are in bold.

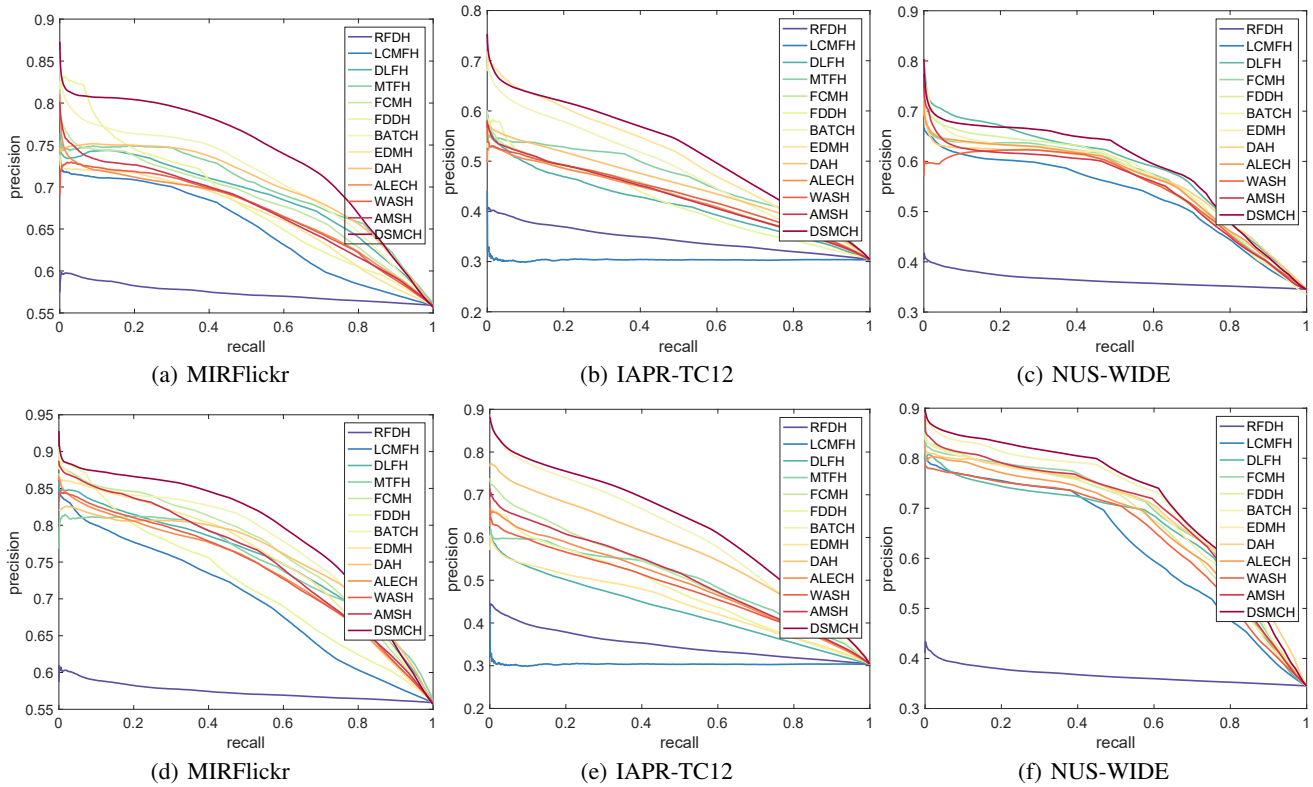
Complexity Analysis

In the hash codes learning stage, the main complexity of each iteration mainly includes $\mathcal{O}(\sum_{t=1}^{2} tdl^2)$ for updating Eq.10, $\mathcal{O}(dln)$ for updating Eq.12, $\mathcal{O}(\sum_{t=1}^{2} t(dnl + l^2n + l^3))$ for updating Eq.16, $\mathcal{O}(dln + l^2n)$ for updating Eq.17, and $\mathcal{O}(c^2ln + cln + ln)$ for updating Eq.18, respectively. In the hash functions learning stage, the complexity is about $\mathcal{O}(d^2 + d^2l + d^2n + dln)$. Since $d, l, t \ll n$, we can observe the overall complexity approximates to $\mathcal{O}(n)$, i.e., linear to the size of the training set.

Experiments

Datasets

To evaluate the performance of our DSCMH, we compare it with thirteen baselines on three used-widely benchmark datasets, *i.e.*, MIRFlickr, IAPR-TC12, and NUS-WIDE. **MIRFlickr** includes 25000 images marked by one or more of 24 textual tags, which crawled from the Flickr website. We select data with more than 20 tags, resulting in 20015 image-text pairs. Image-text pairs are represented by the 512-dimensional GIST feature vector and the 1386-dimensional BOW vectors, respectively. **IAPR-TC12** has 20000 image-text pairs with 255 labels. Image-text pairs are represented by the 512-dimensional GIST feature vectors and the 2912-dimensional BOW vectors, respectively. **NUS-WIDE** consists of 269648 instances of 81 concepts, with the largest ten concepts corresponding to 186577 instances. Image-text pairs are represented by the 500-dimensional SIFT vectors and the 1000-dimensional BOW vectors, respectively. We randomly select 2000 image-text pairs as the query set on MIRFlickr and IAPR-TC12, and choose 1867 image-text pairs as the query set on NUS-WIDE.



(a) MIRFlickr (b) IAPR-TC12 (c) NUS-WIDE

(d) MIRFlickr (e) IAPR-TC12 (f) NUS-WIDE

Figure 2: Precision-recall curves with 64 bits on MIRFlickr. (a-c) and (d-f) is the I $\rightarrow$ T task and the T $\rightarrow$ I task, respectively.

Baselines and Implementation

To verify the performance of our method, we compare DSCMH with thirteen state-of-the-art cross-modal hashing methods, including RFDH (Wang, Wang, and Gao 2017), LCMFH (Wang et al. 2018), DLFH (Jiang and Li 2019), MTFH (Liu et al. 2019), FCMH (Wang et al. 2021b), FDDH (Liu, Wang, and Cheung 2021), BATCH (Wang et al. 2021c), EDMH (Chen et al. 2022), DAH (Zhang et al. 2023b), ALECH (Li et al. 2023), WASH (Zhang et al. 2023a), and AMSH (Luo et al. 2023).

In our experiments, we conduct two cross-modal retrieval tasks, including searching image modality data by text modality data (T $\rightarrow$ I) and searching text modality data by image modality data (I $\rightarrow$ T). Further, we use the mean Average Precision (mAP) and precision-recall curve to evaluate the performance. For the sake of fairness, all comparison methods use the codes provided by the authors and the suggested parameters in the original paper. In the experiments, we empirically set $m = 15$, $q = 1.2$, and $d = 1500$. From the parameter analysis, we set α and λ as $\{10^{-3}, 10^{-2}\}$, $\{10^{-2}, 10^{-3}\}$, and $\{10^{-3}, 10^{-4}\}$ on three datasets, respectively.

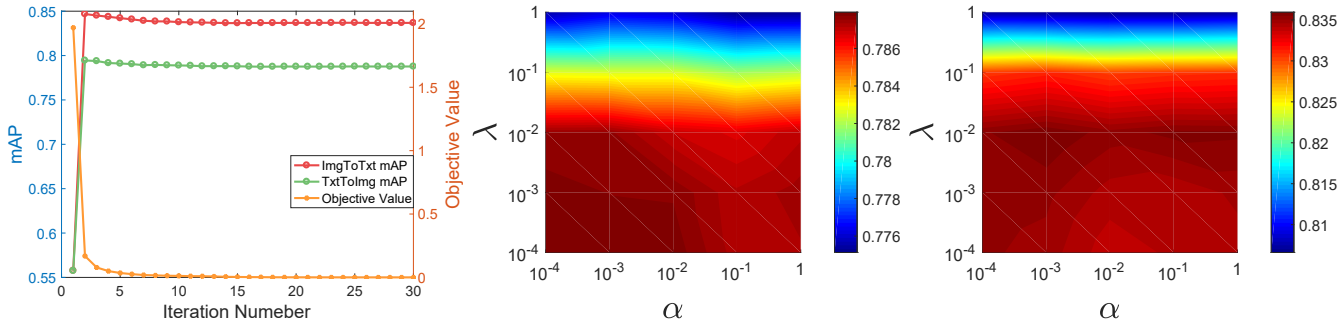
Comparison with State-of-the-Art Methods

Table 1 shows the mAP scores of all comparison methods on the three datasets. From these tables, we can observe the following conclusion: (1) The proposed DSCMH outperforms all baselines for two retrieval tasks on all datasets, which indicates the effectiveness of the dual self-paced cross-modal hashing framework. Specifically, compared to the best baseline, DSCMH achieves a relative improvement of 2.2%, 1.62%,1.54%, and 1.8% for the T $\rightarrow$ I task on MIRFlickr, respectively. (2) Since the textual features possess more discriminative semantic information, almost all baselines perform better for the T $\rightarrow$ I task than the I $\rightarrow$ T task. (3) The performance of all baseline methods increases synchronously with the bit length. This is because long hash codes can preserve more discriminative information. (4) Since MTFH has the large $n \times n$ similarity matrix, it leads to not enough storage space on the large dataset (*i.e.*, NUS-WIDE), making the program unable to execute.

We further draw the precision-recall curves with 64 bits of all comparison methods on the three datasets. As shown in Fig.2, we can notice that our DSCMH can achieve the best results on these datasets, which demonstrates the effectiveness of our dual self-paced hashing framework. Moreover, we can observe that precision-recall curves of the proposed method have slower downward trends than all comparison methods, which shows the stability and superiority on the cross-modal retrieval tasks.

Robustness Analysis

To demonstrate the robustness of the proposed DSCMH against the noise/outliers, we conduct robustness experiments by simulating noise. Specifically, we randomly select half of the image and text data. And we randomly add 20% impulsive noise and flip the value of one/two bits for images and texts, respectively. Table 2 shows the mAP results



(a) Convergence results (b) I → T (c) T → I

Figure 3: Convergence results and the mAP results from varying values of α and λ with 64 bits on MIRFlickr.

of all methods on MIRFlickr. We can see that our method achieves the best retrieval performance both without noise and with noise which demonstrates its strong robustness against noise.

Convergence Experiments

To observe the proposed DSCMH, we plot the convergence and mAP curves with 64 bits on MIRFlickr. In Fig.3 (a), we can observe that the objective function fast converges to a stable value within 5 iterations. Besides, we can find that the mAP scores gradually increases with the number of iterations, and tends to be stable as the objective function converges. In general, the fast and stable convergence property can be experimentally demonstrated.

Parameter Sensitivity Analysis

To evaluate the parameter sensitivity, we first set the code length to 64 bits, and then adopt grid search to set the values of α and λ from $[10^{-4}, 10^{-3}, \cdots, 1]$. As shown in Fig.3 (b) and (c), on MIRFlickr, the different values of these parameters have only small fluctuations in the retrieval performance, which indicates that the proposed DSCMH is very stable. Hence, we can easily obtain the optimum parameters to achieve the best performance.

Time Cost Comparison

To evaluate the efficiency of the proposed method, we record the training time with different bits on NUS-WIDE as shown in Table 3. Obviously, the training time increases accordingly as the bit length increases. Although our method takes some time on the reliability/easiness evaluation of features and instances, our DSCMH still has some advantages in terms of time cost compared with the benchmark baselines. This is because DSCMH is a two-step method that learns hash codes and hash function separately, thereby reducing the computation complexity.

Ablation Analysis

To further analyze the effectiveness of our DSCMH, we perform the ablation study on MIRFlickr. DSCMH has two variations including DSCMH-1 and DSCMH-2. Thereinto, DSCMH-1 denotes learning specific-view latent representation instead of learning consensus latent representation.

Method	Training time			
	8	16	32	64
RFDH	170.53	203.52	302.83	621.01
LCMFH	12.49	12.59	13.72	15.42
DLFH	4.86	8.42	20.45	55.86
FCMH	146.34	147.82	152.55	163.23
FDDH	61.54	64.04	68.65	77.42
BATCH	85.44	86.86	92.23	100.59
EDMH	11.80	12.94	15.20	20.74
DAH	0.81	1.43	2.73	5.36
ALECH	3.46	3.65	4.60	6.07
WASH	11.58	12.03	13.25	15.10
AMSH	36.85	38.84	41.67	47.48
DSCMH	30.82	31.05	31.83	32.46

Table 3: The training time (second) on NUS-WIDE.

Method	I → T				T → I			
	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bit
DSCMH-1	74.59	75.86	77.45	78.24	80.57	81.42	82.15	82.34
DSCMH-2	73.07	74.22	75.33	76.14	78.78	80.22	81.49	81.86
DSCMH	**76.68**	**77.35**	**78.03**	**78.76**	**81.41**	**82.29**	**83.30**	**83.59**

Table 4: Ablation results (mAP: %) on MIRFlickr.

DSCMH-2 denotes to discard dual self-paced learning. Table 4 shows the ablation experimental results. It can be seen that dual self-paced cross-modal hashing can enhance retrieval performance.

Conclusion

In this paper, we propose a novel dual self-paced cross-modal hashing (DSCMH) for learning from both feature and instance levels, which can alleviate the negative effect of the hard instances and features in the learning process. Specifically, inspired by SPL, our DSCMH first estimates the reliability of each instance and feature by the instance-level and feature-level weighting. Then we simultaneously conduct the instance- and feature-level SPL that gradually train the hashing model by starting from 'easy' to 'hard' ones. Comprehensive experiments on three benchmark datasets demonstrate our DSCMH outperforms 12 state-of-the-art CMH methods in terms of the effectiveness and robustness.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U19A2078, 62372315, and 62102274), Sichuan Science and Technology Planning Project (2023YFG0033, 2023ZHCG0016, 2023YFQ0020, 2023ZYZYTS0077), Chengdu Science and Technology Project (2023-XT00-00004-GX, 2021-JB00-00025-GX), the SCU-LuZhou Sciences and Technology Coorperation Program (2023CDLZ-16), and Fundamental Research Funds for the Central Universities under Grant YJ202140.

References

An, J.; Luo, H.; Zhang, Z.; Zhu, L.; and Lu, G. 2022. Cognitive multi-modal consistent hashing with flexible semantic transformation. *Information Processing & Management*, 59(1): 102743.

Björck, Å. 1994. Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications*, 197: 297–316.

Chen, Y.; Zhang, H.; Tian, Z.; Wang, J.; Zhang, D.; and Li, X. 2022. Enhanced Discrete Multi-Modal Hashing: More Constraints Yet Less Time to Learn. *IEEE Transactions on Knowledge and Data Engineering*, 34(3): 1177–1190.

He, R.; Han, Z.; Lu, X.; and Yin, Y. 2022. Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14585–14594.

Hu, M.; Yang, Y.; Shen, F.; Xie, N.; Hong, R.; and Shen, H. T. 2019a. Collective Reconstructive Embeddings for Cross-Modal Hashing. *IEEE Transactions on Image Processing*, 28(6): 2770–2784.

Hu, P.; Huang, Z.; Peng, D.; Wang, X.; and Peng, X. 2023. Cross-Modal Retrieval With Partially Mismatched Pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 9595–9610.

Hu, P.; Peng, X.; Zhu, H.; Lin, J.; Zhen, L.; and Peng, D. 2021. Joint Versus Independent Multiview Hashing for Cross-View Retrieval. *IEEE Transactions on Cybernetics*, 51(10): 4982–4993.

Hu, P.; Wang, X.; Zhen, L.; and Peng, D. 2019b. Separated variational hashing networks for cross-modal retrieval. In *Proceedings of the 27th ACM International Conference on Multimedia*, 1721–1729.

Hu, P.; Zhu, H.; Lin, J.; Peng, D.; Zhao, Y.-P.; and Peng, X. 2022. Unsupervised contrastive cross-modal hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3877–3889.

Huang, Z.; Ren, Y.; Pu, X.; and He, L. 2021. Non-linear fusion for self-paced multi-view clustering. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3211–3219.

Jiang, Q.; and Li, W. 2019. Discrete latent factor model for cross-modal hashing. *IEEE Transactions on Image Processing*, 28(7): 3490–3501.

Jing, M.; Li, J.; Zhu, L.; Lu, K.; Yang, Y.; and Huang, Z. 2020. Incomplete cross-modal retrieval with dual-aligned variational autoencoders. In *Proceedings of the 28th ACM International Conference on Multimedia*, 3283–3291.

Kumar, M.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23.

Li, H.; Zhang, C.; Jia, X.; Gao, Y.; and Chen, C. 2023. Adaptive Label Correlation Based Asymmetric Discrete Hashing for Cross-Modal Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 35(2): 1185–1199.

Li, J.; Kang, Z.; Peng, C.; and Chen, W. 2021. Self-paced two-dimensional PCA. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8392–8400.

Liu, W.; Mu, C.; Kumar, S.; and Chang, S.-F. 2014. Discrete graph hashing. *Advances in neural information processing systems*, 27.

Liu, X.; Hu, Z.; Ling, H.; and Cheung, Y. 2019. MTFH: A matrix tri-factorization hashing framework for efficient cross-modal retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3): 964–981.

Liu, X.; Wang, X.; and Cheung, Y. 2021. FDDH: Fast Discriminative Discrete Hashing for Large-Scale Cross-Modal Retrieval. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15.

Liu, X.; Zeng, H.; Shi, Y.; Zhu, J.; Hsia, C.-H.; and Ma, K.-K. 2023. Deep Cross-modal Hashing Based on Semantic Consistent Ranking. *IEEE Transactions on Multimedia*, 1–12.

Luo, K.; Zhang, C.; Li, H.; Jia, X.; and Chen, C. 2023. Adaptive Marginalized Semantic Hashing for Unpaired Cross-Modal Retrieval. *IEEE Transactions on Multimedia*, 1–14.

Pan, L.; Ai, S.; Ren, Y.; and Xu, Z. 2020. Self-paced deep regression forests with consideration on underrepresented examples. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, 271–287. Springer.

Qin, Y.; Pu, N.; and Wu, H. 2023. Elastic Multi-view Subspace Clustering with Pairwise and High-order Correlations. *IEEE Transactions on Knowledge and Data Engineering*, 1–13.

Qin, Y.; Sun, Y.; Peng, D.; Zhou, J. T.; Peng, X.; and Hu, P. 2023a. Cross-modal Active Complementary Learning with Self-refining Correspondence. *arXiv preprint arXiv:2310.17468*.

Qin, Y.; Tang, Z.; Wu, H.; and Feng, G. 2023b. Flexible Tensor Learning for Multi-View Clustering With Markov Chain. *IEEE Transactions on Knowledge and Data Engineering*, 1–14.

Shao, J.; Wu, Z.; Luo, Y.; Huang, S.; Pu, X.; and Ren, Y. 2022. Self-paced label distribution learning for in-the-wild facial expression recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, 161–169.

Sun, Y.; Peng, D.; Dai, J.; and Ren, Z. 2023a. Stepwise Refinement Short Hashing for Image Retrieval. In *Proceedings of the 31st ACM International Conference on Multimedia*, 6501–6509.

Sun, Y.; Peng, D.; Huang, H.; and Ren, Z. 2022. Feature and semantic views consensus hashing for image set classification. In *Proceedings of the 30th ACM International conference on multimedia*, 2097–2105.

Sun, Y.; Wang, X.; Peng, D.; Ren, Z.; and Shen, X. 2023b. Hierarchical hashing learning for image set classification. *IEEE Transactions on Image Processing*, 32: 1732–1744.

Tan, W.; Zhu, L.; Guan, W.; Li, J.; and Cheng, Z. 2022. Bit-aware semantic transformer hashing for multi-modal retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 982–991.

Tan, W.; Zhu, L.; Li, J.; Zhang, Z.; and Zhang, H. 2023. Partial Multi-Modal Hashing via Neighbor-aware Completion Learning. *IEEE Transactions on Multimedia*, 1–13.

Tu, R.-C.; Jiang, J.; Lin, Q.; Cai, C.; Tian, S.; Wang, H.; and Liu, W. 2023. Unsupervised Cross-modal Hashing with Modality-interaction. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1.

Wang, D.; Gao, X.; Wang, X.; and He, L. 2018. Label consistent matrix factorization hashing for large-scale cross-modal similarity search. *IEEE transactions on pattern analysis and machine intelligence*, 41(10): 2466–2479.

Wang, D.; Wang, Q.; and Gao, X. 2017. Robust and flexible discrete hashing for cross-modal similarity search. *IEEE transactions on circuits and systems for video technology*, 28(10): 2703–2715.

Wang, X.; Zhang, Z.; Wu, B.; Shen, F.; and Lu, G. 2021a. Prototype-supervised adversarial network for targeted attack of deep hashing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16357–16366.

Wang, Y.; Chen, Z.; Luo, X.; Li, R.; and Xu, X. 2021b. Fast Cross-Modal Hashing With Global and Local Similarity Embedding. *IEEE Transactions on Cybernetics*, 1–14.

Wang, Y.; Luo, X.; Nie, L.; Song, J.; Zhang, W.; and Xu, X.-S. 2021c. BATCH: A Scalable Asymmetric Discrete Cross-Modal Hashing. *IEEE Transactions on Knowledge and Data Engineering*, 33(11): 3507–3519.

Xu, C.; Tao, D.; and Xu, C. 2015. Multi-view self-paced learning for clustering. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Yang, X.; Jiaqi, J.; Wang, S.; Liang, K.; Liu, Y.; Wen, Y.; Liu, S.; Zhou, S.; Liu, X.; and Zhu, E. 2023a. Dealmvc: Dual contrastive calibration for multi-view clustering. In *Proceedings of the 31st ACM International Conference on Multimedia*, 337–346.

Yang, Z.; Deng, X.; Guo, L.; and Long, J. 2023b. Asymmetric Supervised Fusion-Oriented Hashing for Cross-Modal Retrieval. *IEEE Transactions on Cybernetics*, 1–14.

Yu, J.; Zhou, H.; Zhan, Y.; and Tao, D. 2021. Deep graph-neighbor coherence preserving network for unsupervised cross-modal hashing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4626–4634.

Zhang, C.; Li, H.; Gao, Y.; and Chen, C. 2023a. Weakly-Supervised Enhanced Semantic-Aware Hashing for Cross-Modal Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 6475–6488.

Zhang, D.; and Wu, X.-J. 2022. Robust and discrete matrix factorization hashing for cross-modal retrieval. *Pattern Recognition*, 122: 108343.

Zhang, D.; Wu, X.-J.; Xu, T.; and Yin, H.-F. 2023b. DAH: Discrete Asymmetric Hashing for Efficient Cross-Media Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 35(2): 1365–1378.

Zhang, Z.; Luo, H.; Zhu, L.; Lu, G.; and Shen, H. T. 2022. Modality-invariant asymmetric networks for cross-modal hashing. *IEEE Transactions on Knowledge and Data Engineering*, 35(5): 5091–5104.

Zhu, L.; Wu, X.; Li, J.; Zhang, Z.; Guan, W.; and Shen, H. T. 2023. Work Together: Correlation-Identity Reconstruction Hashing for Unsupervised Cross-Modal Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 8838–8851.