INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

A fast discrete two-step learning hashing for scalable cross-modal retrieval

Huan Zhao, Kaili Ma

College of Computer Science and Electronic Engineering, Hunan University, China
hzhao@hnu.edu.cn, makaili89@163.com

Abstract

Recently, some cross-modal hashing methods are proposed to search data for different modality effectively. Hashing has received wide attention because of its low storage and high efficiency. Hashing-based methods project the data instances from different modalities into a Hamming space to learn hash codes for retrieval between different modality. Although obtaining promising performance, hashing-based methods have still several common limitations. First, they learn the hash codes by constructing semantic similarity matrices, resulting in the loss of information. Second, most existing methods simultaneously learn the hash codes and the hash functions, which bring a high computational complexity. Third, they utilize the relaxation-based optimization strategy to generate the hash codes which leads to the large quantization error of the hash codes. To solve the above problems, we propose a novel fast supervised hashing method, termed Fast Discrete Two-Step Learning Hashing (FDTLH) for scalable cross-modal retrieval, which learns the discriminative hash codes by adopting a effective two-step learning scheme. Extensive experiments show that the FDTLH outperforms several state-of-the-art hashing methods in terms of retrieval performance and learning efficiency.

Index Terms: hash codes, hash function, cross-modal retrieval

1. Introduction

In this paper, we choose image and text for discussion. The proposed method can be extended to other modalities such as speech or video. To achieve fast image-text retrieval, several hashing-based methods are proposed. These methods map the original data of different modalities to a hamming space which consists of simplified hash codes. Based on such binary representation, these methods achieve excellent performance with low storage and high efficiency [1]. In addition, there are also several researches about neural network. Although obtaining promising retrieval performance, such methods have high computational cost and adjustment of super-parameter in the experiment. Therefore, we mainly focus on the shallow supervised hashing for image-text.

Although supervised hashing methods have achieved desirable results, there are still several obvious common problems. (1) Many hashing methods yield suboptimal retrieval performance owing to high computational loads when performing training. In [2], [3], [4], they construct a sematic similarity matrix based on supervised information to learn the unified hash codes for retrieval tasks. With the increase of training set, the dimension of semantic similarity matrix also increases, which requires a large amount of storage space and computational overhead. Therefore, these methods cannot be applied to large-scale data retrieval scenarios. (2) Most supervised hashing methods have high model complexity within many constraints in their models. In [5], [6], they learn the hash codes and the hash functions simultaneously during the training phase. However, there are many matrix variables in the objective function, which requires a lot of calculation costs. The hash function is trained in the loop, it increases unnecessary calculation amount. Eventually, all methods have high computational complexity in the training process. (3) Several existing hashing methods have limited retrieval accuracy, making the overall performance suboptimal [7], [8], [9], [10], [11], [12], [13], [14]. They utilize relaxation optimization strategies to generate the hash codes. They first relax the discrete constraint of the hash codes and solve the continuous solution. The hash codes are obtained by rounding the continuous solution of approximate binary codes. Although they can generate the hash codes, adopting relaxation strategy leads to quantitative loss and reduces the retrieval accuracy.

To tackle the above issues, we propose a novel supervised hashing method, termed Fast Discrete Two-Step Learning Hashing (FDTLH). The concrete framework of the proposed FDTLH is shown in Fig. 1. Specifically, the main contributions of our work are summarized as follows: (1) To reduce the computational complexity of our previous method Supervised Matrix Factorization Hashing(SMFH-QL) [6], a novel fast supervised hashing method is proposed, termed Fast Discrete Two-Step Learning Hashing (FDTLH). This method adopts a two-step learning scheme to learn the discriminative hash codes, which improves the retrieval performance for image-text retrieval. (2) A discrete optimization strategy [6] is cited to solve the proposed FDTLH instead of the relaxation strategy. It contributions to obtain a closed-form solution for all matrix variables in the objective function, and also short the training time of the whole optimization for enhancing the performance of retrieval tasks. (3) Extensive experimental results show that the proposed FDTLH outperforms several state-of-the-art hashing methods on three representative data sets with retrieval performance and learning efficiency.

2. The proposed FDTLH

In this section, we introduce the problem formulation, the details of the proposed FDTLH and the discrete optimization strategy. The main symbols are involved in Table 1.

2.1. Problem formulation

In this paper, we employ uppercase bold letters to represent matrices, lowercase bold letters to represent vectors, and italicized letters to represent scalars. Let $\mathcal{O} = \{o_i\}_{i=1}^{n}$ represents the training set of cross-modal data, where n is the total number of the data instances. For each instance $o_i = (x_i, y_j, l)$, $x_i \in \Re^{d_1}$ is the ith image feature vector, $y_j \in \Re^{d_2}$ is the ith text feature vector. $l_i \in \{0, 1\}$ is the corresponding class label. d_1 and d_2 are the dimensions of image features and text features respectively, c is the total number of semantic categories. FDTLH aims to learn the effective modality-specific hash functions by projecting the features of the isomerous data into a common Hamming space and generating the hash codes for image-

http://dx.doi.org/10.21437/Interspeech.2021-287

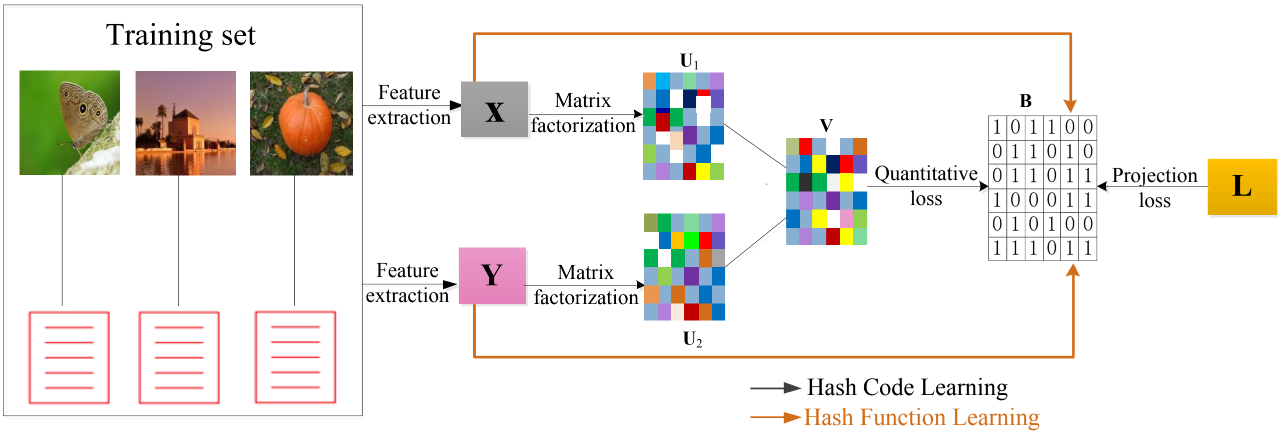

Fig. 1: *In our proposed frame, feature extraction is carried out for the given training set to obtain the feature matrix firstly. The feature matrix is decomposed to obtain the basic matrix and common representation. Then generate the hash codes based on semantic labels, and finally learn the hash codes and the hash functions respectively.*

Table 1: *Important notations in FDTLH*

Notations	Definition
$\mathbf{X},\mathbf{Y}$	Image, text feature matrices, respectively.
$\mathbf{L}$	Semantic label matrix.
$\mathbf{B}$	Hash code matrix of training instances.
$\mathbf{U}_1,\mathbf{U}_2$	Basic matrices for matrix factorization
$\mathbf{P}_1,\mathbf{P}_2$	Mapping matrices for the hash functions
$\mathbf{V}$	Shared latent representation.
$\mathbf{W}$	Projection matrix for the classification loss function.
n	Number of training instances
c	Number of label classes
h	Length of hash codes
m	The number of anchors
ε	Width of kernel

text retrieval. Specifically, the hash functions are expressed as: $f(X) : \Re^{d_1} \rightarrow \{0,1\}^h$ and $f(Y) : \Re^{d_2} \rightarrow \{0,1\}^h$.

2.2. The framework overview

In this section, we introduce the overall objective function of FDTLH including the parts of the hash code learning and the hash function learning, and give its optimization procedure.

2.2.1. Overall Objective Function

Given the feature matrices $\mathbf{X} \in \Re^{d_1 \times n}$ and $\mathbf{Y} \in \Re^{d_2 \times n}$, we first obtain the m-dimension row vectors for the kernel input [15], where $\phi(\mathbf{x})$, $\phi(\mathbf{y})$ via the Gaussian kernel $\phi(\mathbf{x}) = [exp(\|\mathbf{x}-\mathbf{p}_1\|^2/\varepsilon), ..., exp(\|\mathbf{x}-\mathbf{p}_m\|^2/\varepsilon)], \phi(\mathbf{y}) = [exp(\|\mathbf{y}-\mathbf{p}_1\|^2/\varepsilon), ..., exp(\|\mathbf{y}-\mathbf{p}_m\|^2/\varepsilon)]$. The variable $\{\mathbf{p}_j\}_{j=1}^m$ is m anchor points randomly, which is selected from the training instance, and ε is the Gaussian kernel dimension. The main idea of FDTLH adopts a two-step learning scheme to learn the discriminative hash codes. Specifically, the mathematical formulation of FDTLH is summarized as below:

$$\begin{aligned}\min_{\mathbf{U}_1,\mathbf{U}_2,\mathbf{W},\mathbf{V}} & \lambda\|\phi(\mathbf{X})-\mathbf{U}_1\mathbf{V}\|_F^2+\lambda\|\phi(\mathbf{Y})-\mathbf{U}_2\mathbf{V}\|_F^2 \\ & +\beta\|\mathbf{L}-\mathbf{W}\mathbf{B}\|_F^2+\alpha\|\mathbf{B}-\mathbf{V}\|+\gamma Re(\mathbf{U}_1,\mathbf{U}_2,\mathbf{W},\mathbf{V}),\end{aligned} \quad (1)$$

$$\min_{\mathbf{P}_1,\mathbf{P}_2} \|\mathbf{B}-\mathbf{P}_1\phi(\mathbf{X})\|_F^2+\|\mathbf{B}-\mathbf{P}_2\phi(\mathbf{Y})\|_F^2+\mu Re(\mathbf{P}_1,\mathbf{P}_2), \quad (2)$$

where $Re(\cdot) = \|\cdot\|_F^2$ is the regulation term to avoid the over fitting problem.

We divide the objective function into two parts, Eq. (1) and Eq. (2), which represent the hash code learning and the hash function learning respectively.

Hash code learning: In the hash code learning process, we generate the unified discriminative hash codes $\mathbf{B}$ by employing the proposed two-step learning hashing model in Eq. (1).

The first term of Eq. (1) is common representation learning to generate the common representation of the original data. Concretely, we utilize collective matrix decomposition technique [8] to preserve the semantic similarity between modalities. The definition formula is as follows:

$$\min_{\mathbf{U}_1,\mathbf{U}_2,\mathbf{V}} \|\phi(\mathbf{X})-\mathbf{U}_1\mathbf{V}\|_F^2+\|\phi(\mathbf{Y})-\mathbf{U}_2\mathbf{V}\|_F^2. \quad (3)$$

The second term of Eq. (1) is the classification loss function that is applied to keep the consistency between the learned hash codes and the semantic labels. This term is used to obtain the hash codes by a linear mapping based on the class labels. And the corresponding form is represented as:

$$\min_{\mathbf{B},\mathbf{W}} \|\mathbf{L}-\mathbf{W}\mathbf{B}\|_F^2,\ s.t.\ \mathbf{B} \in \{-1,1\}^{h\times n}. \quad (4)$$

The third part of Eq. (1) is quantitative loss function. To directly generate the binary hash codes, a semantic relationship is established between $\mathbf{B}$ and $\mathbf{V}$ to avoid using a continuous relaxation, and the matrix $\mathbf{V}$ can be accurately represented by the learned hash codes. In addition, the difference of the square loss measure between $\mathbf{B}$ and $\mathbf{V}$ is minimized to minimize the quantitative loss:

$$\min_{\mathbf{B},\mathbf{V}} \|\mathbf{B}-\mathbf{V}\|_F^2,\ s.t.\ \mathbf{B} \in \{-1,1\}^{h\times n}. \quad (5)$$

Hash function learning: When generating the pre-learned hash codes $\mathbf{B}$, we can guide the hash functions procedure by these binary codes. To be specific, we obtain the modality-specific hash functions from different modalities by a linear regression form. The concrete formula is defined as:

$$\begin{aligned}&\min_{\mathbf{P}_1,\mathbf{P}_2,\mathbf{B}} \|\mathbf{B}-\mathbf{P}_1\phi(\mathbf{X})\|_F^2+\|\mathbf{B}-\mathbf{P}_2\phi(\mathbf{Y})\|_F^2, \\ &s.t.\ \mathbf{B} \in \{-1,1\}^{h\times n}.\end{aligned} \quad (6)$$

2.3. Optimization Strategy

The optimization problem of Eq. (1) is a non-convex for all matrix variables [16]. A iteration optimization strategy is proposed to solve the FDTLH model. Specifically, the optimization procedure of Eq. (1) consist of step 1-4 and Eq. (2) can be solved by step5 iteratively.

2.3.1. Optimization procedure of Eq. (1)

Step 1 : Updating $\mathbf{U}_1, \mathbf{U}_2$. By fixing the other variables, $\mathbf{U}_1$ and $\mathbf{U}_2$ can be derived as follows:

$$\min_{\mathbf{U}_1} \lambda \|\phi(\mathbf{X}) - \mathbf{U}_1\mathbf{V}\|_F^2 + \gamma\|\mathbf{U}_1\|_F^2, \tag{7}$$

$$\min_{\mathbf{U}_2} \lambda \|\phi(\mathbf{X}) - \mathbf{U}_2\mathbf{V}\|_F^2 + \gamma\|\mathbf{U}_2\|_F^2. \tag{8}$$

We compute the derivation of Eq. (1) with respect to $\mathbf{U}_1$, $\mathbf{U}_2$ and set it to 0. And then we obtain that:

$$\mathbf{U}_1 = \lambda\phi(\mathbf{X})\mathbf{V}^T \left(\lambda\mathbf{V}\mathbf{V}^T + \gamma\mathbf{I}\right)^{-1}, \tag{9}$$

$$\mathbf{U}_2 = \lambda\phi(\mathbf{Y})\mathbf{V}^T \left(\lambda\mathbf{V}\mathbf{V}^T + \gamma\mathbf{I}\right)^{-1}, \tag{10}$$

where $\mathbf{I}$ is the identity matrix.

Step 2 : Updating $\mathbf{W}$. By fixing the other variables, the optimization for W can be solved as follows:

$$\min_{\mathbf{W}} \beta\|\mathbf{L} - \mathbf{W}\mathbf{B}\|_F^2 + \gamma\|\mathbf{W}\|_F^2, \tag{11}$$

where we compute the derivation of Eq. (1) with respect to $\mathbf{W}$ and set it to 0. And we can obtain:

$$\mathbf{W} = \left(\beta\mathbf{B}\mathbf{B}^T + \gamma\mathbf{I}\right)^{-1} \beta\mathbf{B}\mathbf{L}^T. \tag{12}$$

Step 3 : Update $\mathbf{V}$. We learn the latent semantic representation of $\mathbf{V}$ for image and text data by fixing other variables and we compute the derivation of Eq. (1) with respect to $\mathbf{V}$ and set it to 0, then we obtain that:

$$\begin{aligned} \mathbf{V} = & \left[\lambda(\mathbf{U}_1)^T\mathbf{U}_1 + \lambda(\mathbf{U}_2)^T\mathbf{U}_2 + \alpha\mathbf{I}\right]^{-1} \\ & \times \left[\lambda(\mathbf{U}_1)^T\phi(\mathbf{X}) + \lambda(\mathbf{U}_2)^T\phi(\mathbf{Y}) + \alpha\mathbf{B}\right]. \end{aligned} \tag{13}$$

Step 4 : Updating $\mathbf{B}$. The unified binary codes B is optimized by fixing other variables of Eq. (1):

$$\min_{\mathbf{B}} \beta\|\mathbf{L} - \mathbf{W}\mathbf{B}\|_F^2 + \alpha\|\mathbf{B} - \mathbf{V}\|_F^2. \tag{14}$$

Specifically, Eq. (14) can be rewritten as:

$$\begin{aligned} \min_{\mathbf{B}} & \beta tr\left[\left(\mathbf{L}^T - \mathbf{B}^T\mathbf{W}^T\right)\left(\mathbf{L} - \mathbf{W}\mathbf{B}\right)\right] \\ & + \alpha tr\left[\left(\mathbf{B}^T\mathbf{V}^T\right)\left(\mathbf{B}\mathbf{V}\right)\right]. \end{aligned} \tag{15}$$

Specifically, subject to $\mathbf{B} \in \{-1, 1\}^{h\times n}$, $tr(\mathbf{B}^T\mathbf{B})$ is a constant, and then Eq. (15) becomes:

$$\min_{\mathbf{B}} -\beta tr\left(\mathbf{L}^T\mathbf{W}\mathbf{B}\right) - \alpha\left(\mathbf{V}^T\mathbf{B}\right). \tag{16}$$

Then the unified hash codes $\mathbf{B}$ can be solved as follows:

$$\mathbf{B} = sgn(\alpha\mathbf{V} + \beta\mathbf{W}^T\mathbf{L}), \tag{17}$$

according to Eq. (17), we can obtain a closed solution of $\mathbf{B}$, and then learn all bits of each binary codes through semantic labels. The hash codes are obtained by Eq. (17), where $sgn(\cdot)$ is a sign function which transforms continuous data into the hash codes.

2.3.2. Optimization procedure of Eq. (2)

Step 5 : Updating $\mathbf{P}_1$ and $\mathbf{P}_2$. By fixing the other variables of Eq. (2), $\mathbf{P}_1$ and $\mathbf{P}_2$ are optimized as follows:

$$\min_{\mathbf{P}_1} \|\mathbf{B} - \mathbf{P}_1\phi(\mathbf{X})\|_F^2 + \mu\|\mathbf{P}_1\|_F^2, \tag{18}$$

$$\min_{\mathbf{P}_2} \|\mathbf{B} - \mathbf{P}_2\phi(\mathbf{Y})\|_F^2 + \mu\|\mathbf{P}_2\|_F^2. \tag{19}$$

The solutions are:

$$\mathbf{P}_1 = \mathbf{B}(\phi(\mathbf{X}))^T \left(\phi(\mathbf{X})(\phi(\mathbf{X}))^T + \mu\mathbf{I}\right)^{-1}, \tag{20}$$

$$\mathbf{P}_2 = \mathbf{B}(\phi(\mathbf{Y}))^T \left(\phi(\mathbf{Y})(\phi(\mathbf{Y}))^T + \mu\mathbf{I}\right)^{-1}. \tag{21}$$

3. Experiments and Discussion

In this section, to evaluate the retrieval performance and the learning efficiency of the proposed FDTLH, a series of quantitative experiments are conducted on three benchmark data sets. We first introduce the data sets, the evaluation metrics, baselines in the experiments. Then we compare the proposed FDTLH with several state-of-the-art cross-modal hashing approaches and analyze the results.

3.1. Data sets

We adopt Wiki [3], MIRFILCKR-25k [17] and NUS-WIDE [18] to evaluate the cross-modal hashing methods.

Table 2: *The details of three benchmark data sets.*

Data set	Wiki	MIRFlICKR-25K	NUS-WIDE
Training Set	2,173	20,015	186,577
Quary Set	693	2,000	1,867
Image Feature	128-D	512-D	500-D
Text Feature	10-D	1,386-D	1,000-D

3.2. Baseline methods and Evaluation metrics

To demonstrate the performance of the FDTLH, several state-of-the-art hashing methods are selected as the baselines to compare with our method. All comparison methods include unsupervised hashing (Latent semantic sparse hashing(LSSH) [8], Collective Matrix Factorization Hashing(CMFH) [10]), and supervised hashing (Supervised Matrix Factorization Hashing(SMFH) [2], Semantic Correlation Maximization(SCM) [3], Semantics-Preserving Hashing(SePH) [4], Label Consistent Matrix Factorization Hashing(LCMFH) [5], and SMFH-QL [6]). To validate the effectiveness of FDTLH, two popular evaluation metrics are adopted: Mean average precision (mAP) [3], [19], Precision-Recall(PR) [20], [21]. A larger value of metric indicates the better retrieval performance.

3.3. Retrieval Performance comparison

To demonstrate the retrieval performance of the proposed FDTLH, we implement the comparison experiments including the training time analysis and the effect of training size. When calculating the training time of all methods, the number of iterations we select is the number of iterations when the objective function of this method converges.

The training time analysis: Due to the Wiki data set is so small, the experimental results are not comparable. We only conduct a set of experiments on the MIRFLICKR-25K data set and obtain the training time of each method when it converges, as shown in Table 4. Compared with the related method SMFH-QL, the training time of FDTLH is far lower, which is about 3-times faster than SMFH-QL [6]. Compared with other earlier methods, the training time of FDTLH is greatly shortened. For example, when the hash code length is 256 bits, the training time of the FDTLH is 50-times faster than supervised hashing

Table 3: *The mAP comparison of all methods on three data sets.*

Task	Methods	Wiki				MIRFILCKR-25K				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
I2T	LSSH	0.2321	0.2412	0.2355	0.2350	0.6287	0.6399	0.6443	0.6559	0.4949	0.5239	0.5305	0.5409
	CMFH	0.2457	0.2540	0.2598	0.2609	0.6477	0.6509	0.6500	0.6449	0.4966	0.5150	0.5146	0.5086
	SMFH	0.2284	0.2480	0.2670	0.2575	0.6054	0.6255	0.6640	0.6821	0.6315	0.6355	0.6424	0.6484
	SCM	0.2474	0.2363	0.2403	0.2602	0.6387	0.6501	0.6588	0.6653	0.5496	0.5642	0.5425	0.5369
	SePH	0.2115	0.2545	0.2663	0.2486	0.6352	0.6394	0.6417	0.6437	0.5379	0.5284	0.5410	0.5515
	LCMFH	0.2468	0.2484	0.2520	0.2459	0.6963	0.7096	0.7061	0.6476	0.6124	0.6126	0.6318	0.6378
	SMFH-QL	0.3223	0.3775	0.3828	**0.3940**	**0.7069**	0.7030	0.7166	0.7193	**0.6174**	0.6369	0.6501	0.6622
	FDTLH	**0.3379**	**0.3881**	**0.3920**	0.3914	0.7048	**0.7208**	**0.7185**	**0.7268**	0.6121	**0.6463**	**0.6541**	**0.6624**

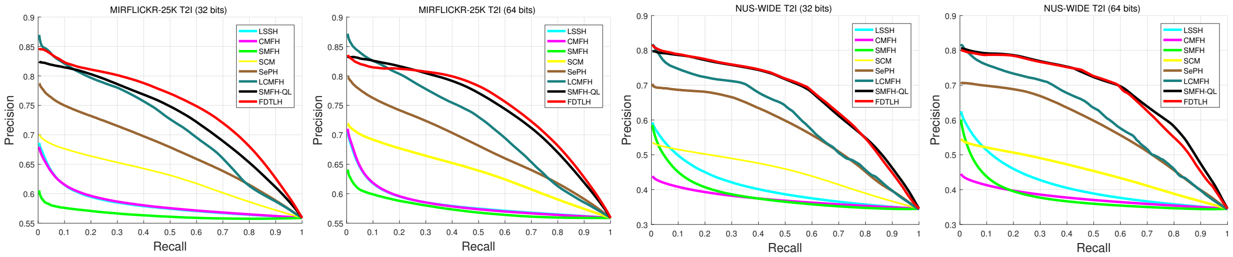

Fig. 2: *PR curves of all baselines @ 64-bit and 32-bit on two data sets.*

Table 4: *Training time comparison of all methods on MIRFLICKR-25K.*

Methods	8 bits	16 bits	32 bits	64 bits	128 bits	256 bits
LSSH	7.25s	21.09s	10.62s	11.52s	13.78s	14.00s
CMFH	42.26s	45.28s	111.48s	58.47s	65.82s	135.18s
SMFH	18.03s	17.12s	17.08s	24.39s	32.22s	34.78s
SCM	6.86s	10.18s	18.45s	36.60s	68.43s	223.74s
SePH	11.28s	13.52s	18.77s	73.73s	85.75s	192.91s
LCMFH	5.91s	6.47s	7.33s	10.44s	7.67s	11.17s
SMFH-QL	3.17s	3.20s	3.76s	4.52s	6.46s	10.13s
FDTLH	**1.31s**	**1.30s**	**1.50s**	**1.96s**	**2.75s**	**5.00s**

SCM [3], which is sufficient to demonstrate that our proposed method is more advantageous on larger data sets.

The effect of training size: We choose to conduct baseline experiments with different data set lengths within 10,000 under the hash codes of different lengths. For the training time results on NUS-WIDE data set are shown in Table 5. It can be seen that with the increase of training set, in addition to SMFH-QL [6], the training time of other baseline methods have a sharp augment which is much higher than the approach we proposed. For example, when the training size is 10,000, FDTLH even is 500-times faster than CMFH [10]. Regardless of the length of any data set, the training time of our method is the shortest which demonstrates that FDTLH obtains the superior retrieval speed over other baselines.

3.4. Retrieval performance comparison

To demonstrate the retrieval performance of FDTLH, we conduct quantitative experiments under two metrics. And we select 30 iterations for the objective function of FDTLH.

Result on mAP and PR: As shown in Table 3, to save space, this table only records the mAP values of the I2T about FDTLH and related comparison methods in three data sets respectively. Four encoding lengths are selected for comparison experiments. Generally, the retrieval performance of our proposed method outperform all the baselines under this metric. Fig. 2 shows the PR curves under T2I of all methods, respectively. Due to the space limitation, we select 32 bits and 64 bits on two data sets to compare our FDTLH and all comparison methods. It can be seen from Fig. 2 that the PR curves of the FDTLH are better than all the baselines under different code lengths on both two tasks, indicating that the advantage of the proposed two-step learning scheme on the two data sets.

Table 5: *Training time of all methods with different training size @ 64-bit on NUS-WIDE.*

Methods	Different training sizes				
	1000	2000	3000	5000	10000
LSSH	1.35s	2.29s	3.57s	6.39s	11.70s
CMFH	8.59s	32.51s	63.54s	117.90s	375.40s
SMFH	30.18s	30.83s	32.28s	36.94s	109.24s
SCM	11.34s	16.98s	16.59s	18.40s	15.39s
SePH	15.85s	19.19s	19.66s	23.90s	24.68s
LCMFH	0.35s	0.47s	0.65s	1.04s	2.02s
SMFH-QL	0.24s	0.36s	0.57s	0.76s	1.48s
FDTLH	**0.11s**	**0.14s**	**0.20s**	**0.33s**	**0.67s**

4. Conclusion

In this paper, we adopt a two-step learning scheme to generate the compact hash codes and the distinct modality-specific hash functions for two kinds of retrieval tasks, which reduces the computational complexity of our previous SMFH-QL method and improves the learning efficiency. Extensive experimental results show that our proposed FDTLH outperforms several state-of-the-art hashing methods on three representative data sets. In the future, we plan to combine our method with a self-supervised deep network to learn more informative common representation for the image-text retrieval.

5. References

[1] D. Wang, X. Gao, X. Wang, and L. He, "Semantic topic multimodal hashing for cross-media retrieval," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 3890–3896.

[2] H. Liu, R. Ji, Y. Wu, and G. Hua, "Supervised matrix factorization for cross-modality hashing," *IEEE Transactions on Image Processing*, pp. 1767–1773, 2016.

[3] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 2177–2183.

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

[5] Di, Wang, X-B, Gao, Xiumei, Lihuo, and He, "Label consistent matrix factorization hashing for large-scale cross-modal similarity search." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2466–2479, 2018.

[6] H. Zhao, S. Wang, X. She, and C. Su, "Supervised matrix factorization hashing with quantitative loss for image-text search," *IEEE Access*, vol. 8, no. 99, pp. 102 051–102 064, 2020.

[7] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Intermedia hashing for large-scale retrieval from heterogeneous data sources," *IEEE Transactions on Image Processing*, pp. 785–796, 2013.

[8] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 415–424.

[9] H. Liu, R. Ji, Y. Wu, F. Huang, and B. Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *Computer Vision and Pattern Recognition*, 2017, pp. 7380–7388.

[10] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2083–2090.

[11] Liong, V. Erin, Jiwen, Tan, and Yap-Peng, "Cross-modal discrete hashing," *PATTERN RECOGNITION*, pp. 114–129, 2018.

[12] G. Irie, H. Arai, and Y. Taniguchi, "Alternating co-quantization for cross-modal hashing," in *IEEE International Conference on Computer Vision*, 2016, pp. 1886–1894.

[13] L. Zhang, Y. Zhang, R. Hong, and Q. Tian, "Full-space local topology extraction for cross-modal retrieval," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 24, no. 7, pp. 2212–24, 2015.

[14] 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. PP, no. 5, pp. 2494–2507, 2017.

[15] H. J. Huang, R. Yang, C. X. Li, Y. Shi, and X. S. Xu, "Supervised cross-modal hashing without relaxation," in *IEEE International Conference on Multimedia and Expo*, 2017, pp. 1159–1164.

[16] X. Lu, L. Zhu, Z. Cheng, X. Song, and H. Zhang, "Efficient discrete latent semantic hashing for scalable cross-modal retrieval," *Signal processing*, vol. 154, no. JAN., pp. 217–231, 2019.

[17] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *Acm International Conference on Multimedia Information Retrieval*, 2008, pp. 39–43.

[18] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: A real-world web image database from national university of singapore," in *Acm on Multimedia Conference*, 2009, pp. 48–56.

[19] Taniguchi, Rin-ichiro, Shimada, Atsushi, Xu, Xing, He, Li, Lu, and Huimin, "Learning unified binary codes for cross-modal retrieval via latent semantic hashing," *Neurocomputing*, pp. 191–203, 2016.

[20] G. Ding, Y. Guo, J. Zhou, and Y. Gao, "Large-scale cross-modality search via collective matrix factorization hashing," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5427–5440, 2016.

[21] Xu and Xin-Shun, "Dictionary learning based hashing for cross-modal retrieval," in *Acm on Multimedia Conference*, 2016, pp. 177–181.