

# Robust Hashing With Bilinear Drift for Image-Text Retrieval

Huan Zhao<sup>ID</sup>, Zeyi Li<sup>ID</sup>, Song Wang<sup>ID</sup>, Zixing Zhang<sup>ID</sup>, *Senior Member, IEEE*, and Keqin Li<sup>ID</sup>, *Fellow, IEEE*

**Abstract**—Supervised hashing models for image-text retrieval are fundamental and versatile in social media analysis and cross-lingual web search. Among them, supervised bilinear drift hashing is one of the most popular approaches. However, it still faces several challenges. For instance, how to leverage the power of bilinear drift hashing to distinguish similar and dissimilar data samples effectively; how to strengthen the semantic relationship between similar data and supervision. To solve these problems, we propose Robust Hashing with Bilinear Drift (RHBD) to improve the accuracy and robustness of the supervised model. The key idea of this work is to generate effective hash codes between image-text feature representations by combining robust data distributions and multiple supervision information. The benefits of bilinear drift with robust hashing, which enhance the discrimination of hash binary, are manifested mainly in two ways: (1) RHBD employs a semantic autoencoder with a linear drift to get a discriminative common feature representation between image and text modalities; (2) RHBD explores iteration quantization with a linear drift to well generate similarity-preserving hash codes. Moreover, we introduce multiple supervision learning to promote the consistency between data information and supervision knowledge for semantic complementarity. Results on three public datasets show that RHBD is effective in image-text retrieval, consistently outperforming other state-of-the-art models with comparable training efficiency to competitive baselines.

**Index Terms**—Bilinear drift, image-text retrieval, robust hashing, supervised hashing.

## I. INTRODUCTION

WITH the continuous exponential growth of data on social networks, matching the similarity between original instances in high-dimensional space is impractical for image-text retrieval research [4], [19], [20], [25], [29], [45]. To achieve efficient retrieval, many well-designed hashing methods [10], [11], [12], [21], [33] have been explored and successfully applied in various fields, including online

recommendation [3], multimedia analysis [38], web query representation [44], etc.

Hashing-based methods improve the retrieval efficiency and reduce the storage requirements by transforming the high-dimensional data into low-dimensional compact binary codes [48], [49], [60]. Based on this binary representation, such approaches rapidly construct the hashing models by producing the hash codes or hash functions for similarity search. In retrieval process, similarity search is simply achieved by computing the Hamming distance between the hash code of the original data and that of the query sample. Since the Hamming distance can be quickly calculated by leveraging bitwise XOR operations, the whole retrieval process can be conducted efficiently. Numerous studies have shown that supervised methods can yield better results than unsupervised ones, which have become the main research hotspot in image-text retrieval. Specifically, supervised hashing-based image-text retrieval approaches [1], [8], [13], [35], [37], [41], [52] can be categorized into two types: traditional supervision and deep supervised ones. The former optimizes the hash codes via semantic supervision knowledge as additional information, while the latter adopts distinct network branches to correlate the image-text instances. Although yielding great success, deep supervised hashing methods are limited by complex optimization objectives and inefficiency. Thus, our work concentrates on traditional supervised hashing for image-text retrieval.

Generally, most excellent traditional supervised methods [6], [30], [36], [40], [57] effectively utilize the data distributions and supervised knowledge to generate high-quality hash codes for image-text retrieval. Despite great achievements in this learning paradigm, there are still two challenges that need further consideration. (1) **Ineffective representation of original data.** Most supervised hashing methods [30], [41], [42], [57] always strive to explore a wider variety of supervised knowledge, as well as common and unique attributes of data, and the strong correlation between common and unique data to design the optimal hashing framework. However, the noise and outliers mixed in the data collection stage have been rarely addressed in the literature yet. Furthermore, such inherent noise and outliers from instances tend to disrupt the distribution of image-text data, bringing in inefficient data representations and a subsequent decline in accuracy. (2) **The weak correlation between similar data and supervision.** After delivering similar data instances, the common way is to correlate the similar data features

Received 1 June 2024; revised 13 August 2024 and 10 January 2025; accepted 22 February 2025. Date of publication 25 February 2025; date of current version 6 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62076092 and in part by the Special Project of Foshan Science and Technology Innovation Team under Grant FS0AA-KJ919-4402-0069. This article was recommended by Associate Editor S. Bakshi. (*Corresponding author: Song Wang.*)

Huan Zhao, Zeyi Li, Song Wang, and Zixing Zhang are with the College of Computer Science and Electronic Engineering, Hunan University, Hunan 410082, China (e-mail: hzhao@hnu.edu.cn; zeyili@hnu.edu.cn; swang17@hnu.edu.cn; zixingzhang@hnu.edu.cn).

Keqin Li is with the College of Computer Science and Electronic Engineering, Hunan University, Hunan 410082, China, and also with the Department of Computer Science, State University of New York at New Paltz, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TCSVT.2025.3545643

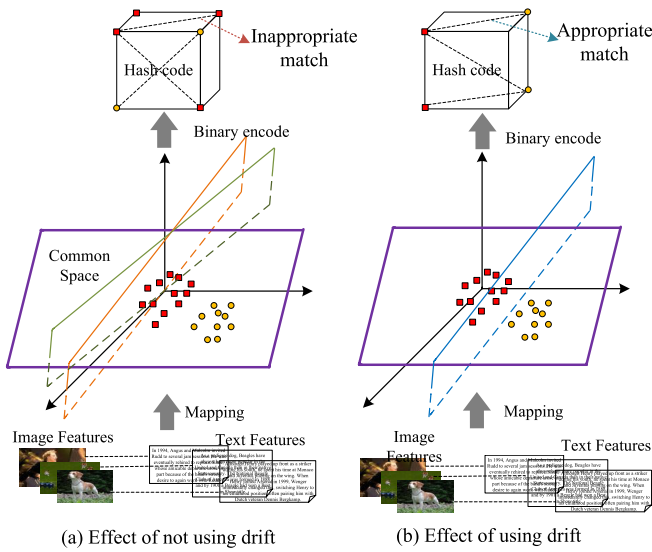


Fig. 1. The graphical descriptions between the conventional and our methods over the drift. (a) is the negative effect without drift, which fails to effectively separate the sample data. (b) shows the positive effect achieving a better separation effect due to the drift.

and supervision. For example, one type of method [32], [36], [51] combines single supervision and data descriptor for image-text retrieval. Another type of method [7], [28], [59] employs the original data characteristics and nonlinear multiple supervised knowledge to provide the hash codes. However, these two approaches either lack abundant supervised information, or generate multiple supervision with high complexity, leading to an imbalance between supervision and data, and ultimately resulting in inaccurate hash code representations.

To address these two problems, we propose a novel supervised hashing model called Robust Hashing with Bilinear Drift (RHBD) for image-text retrieval. Specifically, the RHBD constructs a cross-modal semantic autoencoder with linear drift to obtain a common feature representation. Then, it exploits multiple supervision learning to formulate the hash code. Next, we introduce the iteration quantization optimization with linear drift to establish the relationship between the hash code and common feature representation. These two linear drift constitute our proposed robust hashing with bilinear drift. Figure 1 illustrates a description of the positive effect of drift. Figure 1 (a) shows the separation effect of samples from different classes without drift, while Figure 1 (b) is the separation effect with drift. Drift can be equivalent to the intercept of an equation with one variable. By adding drift, the proposed RHBD effectively separates data samples using different classification functions generated by this intercept. This learning paradigm not only optimizes the generation quality of hash codes, but also eliminates the mismatching information and further enhances the robustness of hash codes, thereby achieving the accuracy of model. RHBD includes robust hashing with bilinear drift part (semantic autoencoder with linear drift and iteration quantization with linear drift), and multiple supervision learning part. The framework of RHBD is shown in Figure 2. To sum up, this paper contributes in the following aspects:

- We propose a novel supervised hashing learning framework dubbed RHBD by integrating multiple supervised knowledge and efficient data descriptors to achieve shared hash representations for image-text instances.
- The introduced robust hashing with bilinear drift learning strategy effectively separates the similar and dissimilar samples, thereby improving the discrimination of hash codes during training.
- Extensive experiments on three public cross-modal datasets demonstrate that the proposed RHBD outperforms several state-of-the-art hashing methods in retrieval performance.

The rest of the paper is arranged: Section II reviews the hashing. Section III presents our proposed RHBD. Section IV illustrates the experiments and Section V concludes the paper.

## II. RELATED WORK

This section divides the related work into three categories: unsupervised hashing, supervised hashing, and hashing with linear drift. The unsupervised one implements the search tasks within the original feature distributions. Typical examples are CMFH [5], FSH [23], JIMFH [37], and DRMFH [52], which primarily obtain the query hash code matrix through collective matrix factorization or by common and individual feature characteristics. Our research also incorporates these paradigms, representing the image and text feature matrices while improving the accuracy in cross-modal applications.

In supervised hashing, numerous excellent methods leverage various supervision knowledge for image-text retrieval such as SMFH [32], SePH [22], LCMFH [36], SRLCH [30], LFMH [58], RDMH [56], SDMSA [57], ALECH [17], ESGEH [53], and IMADS [43]. Compared with these approaches, we find an interesting point that the above approaches conduct the training models jointly taking into account distinct feature distributions and multiple supervision information. Another notable aspect is that such methods adopt multiple collective matrix factorization and mapping learning strategies to enhance the effectiveness of hashing models. However, this strategy does not consider the problem of imbalanced data distribution caused by similar samples. To handle this, a few hashing models [24], [33], [40] have attempted to improve the search accuracy by leveraging linear projection with drift to obtain the high-quality hash codes. For instance, FDCH [24], ACQH [40], WASH [55], ROHLSE [18], and FADCH [33] mainly explore the relevance of similar samples by classify the original data with a linear drift or an auxiliary matrix variable to construct the hashing models. However, these hashing models do not fully a good correlation between feature information, hash learning, and multiple supervised knowledge, which obtain limited retrieval performance. Hence, our bilinear model can be viewed as a generalization of hashing with linear drift approach, utilizing the feature, dual supervision, and separate linear drift information with application to image-text search.

Unlike the aforementioned ones, deep hashing methods have taken central in image-text retrieval [16], [26], [34]. Although these methods generally outperform the shallow methods,

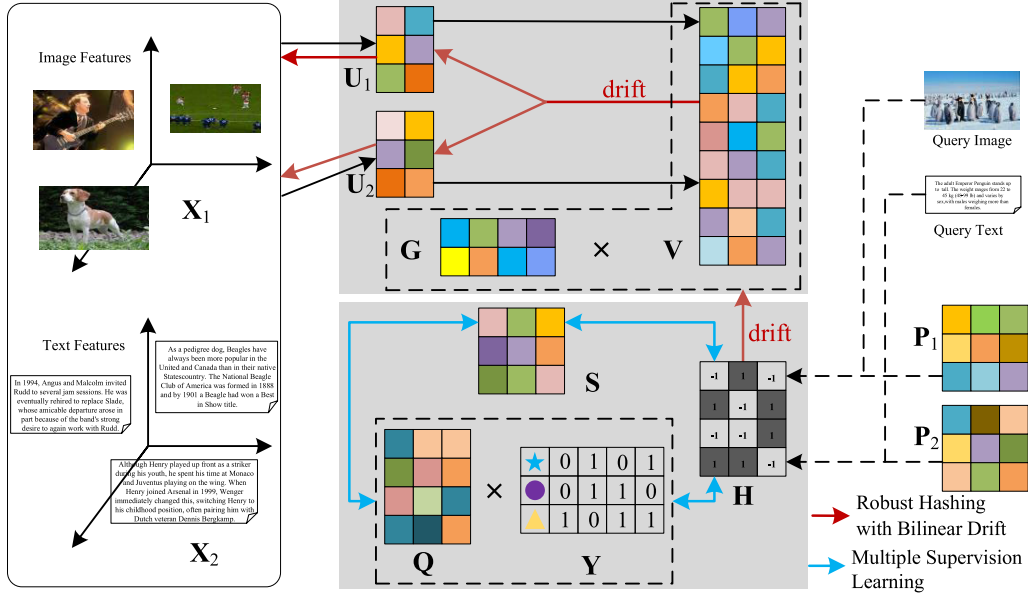


Fig. 2. The pipeline of RHBD contains two learning modules: robust hashing with bilinear drift (red arrows) to obtain common representation  $\mathbf{V}$  and multiple supervision learning (blue arrows) to produce the hash code  $\mathbf{H}$ . RHBD takes the image-text features  $\mathbf{X}_1, \mathbf{X}_2$  as the model input, and then achieves a unified supervised hashing model by connecting the relationship between the matrices  $\mathbf{V}$  and  $\mathbf{H}$ . And this model during training stage produces the final hash code  $\mathbf{H}$ , hash functions  $\mathbf{P}_1, \mathbf{P}_2$  for the querying stage as the model output.

their training is very time-consuming and requires heavy storage cost of experimental equipment. After comprehensive consideration and important development, this paper focuses on the shallow image-text retrieval field.

### III. METHODOLOGY

#### A. Notation and Problem Definition

Assuming there is a collection of  $n$  image-text sample pairs, represented by image feature matrix  $\mathbf{X}_1 = \{x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(n)}\} \in \mathbb{R}^{d_1 \times n}$  and text feature matrix  $\mathbf{X}_2 = \{x_2^{(1)}, x_2^{(2)}, \dots, x_2^{(n)}\} \in \mathbb{R}^{d_2 \times n}$ , where  $d_1$  and  $d_2$  denote the feature dimensions of images and texts ( $d_1 \neq d_2$ ). To clear representation, we explain that  $\|\cdot\|_F$  is the Frobenius norm of a matrix,  $\text{tr}(\cdot)$  is the trace of a matrix, and  $\text{sgn}(\cdot)$  indicates the sign function, namely  $\text{sgn}(x) = 1$  when  $x > 0$ , and  $\text{sgn}(x) = -1$  when  $x \leq 0$ . The frequently-used matrix variables are defined in Table I.

In multiple supervision learning, we utilize two kinds of supervised knowledge: label and semantic supervision. (1) For label supervision, we express  $\mathbf{Y} = \{y_1, y_2, \dots, y_n\} \in \mathbb{R}^{c \times n}$  as the label matrix, where  $c$  mean the number of classes. The feature vector of sample  $i$  is defined as  $y_i = \{y_{i1}, y_{i2}, \dots, y_{ic}\} \in \{0, 1\}^c$ . If sample  $i$  belongs to class  $j$ ,  $y_{ij} = 1$ ; otherwise,  $y_{ij} = 0$ . (2) For semantic supervision, most methods [7], [33], [41], [49] leverage a pairwise similarity matrix  $\mathbf{S}$  (size  $n \times n$ ) achieved by the label matrix  $\mathbf{Y}$  to compare the similarity between samples. When samples  $i$  and  $j$  are similar,  $\mathbf{S}_{ij} = 1$ ; otherwise,  $\mathbf{S}_{ij} = -1$ . However, these approaches have negative effects on computational complexity because  $\mathbf{S}$  requires a large amount of computational space. To address this, cosine similarity is employed to quantify the similarity between two samples:

$$\tilde{\mathbf{S}}_{ij} = \frac{y_i \cdot y_j}{\|y_i\|_2 \|y_j\|_2}. \quad (1)$$

TABLE I  
THE MAIN SYMBOLS USED IN RHBD

Notations	Definition
$\mathbf{X}_t$	Image or text feature matrix
$\mathbf{U}_t$	Basic matrices for training data
$\mathbf{V}$	Common representation
$\mathbf{S}$	Semantic pairwise similarity matrix
$\mathbf{Q}, \mathbf{G}$	Mapping matrices
$\mathbf{Y}$	Shared Label matrix
$\mathbf{H}$	To-be-generated hash code matrix
$\mathbf{l}_t, \mathbf{m}$	Two linear drifts

Here, the label is normalized as:

$$\tilde{\mathbf{Y}} = \left\{ \frac{\mathbf{Y}_{\cdot 1}}{\|\mathbf{Y}_{\cdot 1}\|_2}, \frac{\mathbf{Y}_{\cdot 2}}{\|\mathbf{Y}_{\cdot 2}\|_2}, \dots, \frac{\mathbf{Y}_{\cdot n}}{\|\mathbf{Y}_{\cdot n}\|_2} \right\}, \quad (2)$$

where  $\mathbf{Y}_{\cdot i}$  is label vector of sample  $x_i$ . To well optimize time consumption, the final similarity matrix is defined by  $\mathbf{S} = 2\tilde{\mathbf{Y}}^\top \tilde{\mathbf{Y}} - \mathbf{1}_n^\top \mathbf{1}_n$ .

In the domain of image-text retrieval, kernelization possesses a crucial role in facilitating nonlinear relationships by converting their respective data into feature representations that exhibit similar properties [30], [41], [57]. The Radial Basis Function (RBF) is adopted as the kernel function in this paper, which is defined as:

$$\phi(x) = \left[ \exp\left(-\frac{\|x - \alpha_1\|^2}{2\sigma^2}\right), \dots, \exp\left(-\frac{\|x - \alpha_m\|^2}{2\sigma^2}\right) \right]^\top, \quad (3)$$

where  $\{\alpha_i\}_{i=1}^m$  represents a set of randomly chosen anchor points, and  $\sigma = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m (x_i - \alpha_j)$  denotes the kernel width. For enhance readability, the transformed feature representations of  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are denoted as  $\mathbf{X}_1 = \phi(\mathbf{X}_1)$  and  $\mathbf{X}_2 = \phi(\mathbf{X}_2)$ , respectively.

The core of this work is to get the hash code matrix  $\mathbf{H} \in \{-1, 1\}^{k \times n}$  to characterize image-text data pairs when

given binary length  $k$ . Matrix  $\mathbf{H}$  is obtained by the learned optimal common feature representation and multiple supervision knowledge.

### B. Robust Hashing With Bilinear Drift

The main contribution of the proposed RHBD lies in the formulation of robust hashing with bilinear drift, as opposed to the traditional signal linear drift learning. The learning module includes a semantic autoencoder with linear drift and iterative quantization with linear drift.

1) *Semantic Autoencoder With Linear Drift*: To incorporate potential information from original data, we use collective matrix factorization technique accepted by most methods [5], [30], [36], to obtain common representation  $\mathbf{V}$ . However,  $\mathbf{V}$  may introduce redundant information from data collection. To tackle this, we extend the encoder module from the original decoder component to construct a semantic autoencoder. Further study [39], [46], [50] shows that the encoder mapping learning brings in irrelevant information such as noise from data pairs. Thus, we elaborately design a novel semantic autoencoder with linear drift term to handle this. The concrete formula is defined:

$$\min_{\mathbf{V}} \sum_{t=1}^2 \lambda_t \underbrace{\|\mathbf{X}_t - \mathbf{U}_t^\top \mathbf{V}\|_F^2}_{\text{Decode}} + \alpha \sum_{t=1}^2 \underbrace{\|\mathbf{V} - \mathbf{U}_t \mathbf{X}_t + \mathbf{I}_t \mathbf{1}_n^\top\|_F^2}_{\text{Encode}}, \quad (4)$$

where  $\mathbf{U}_t$  is projection matrix, and  $\mathbf{I}_t$  acts as an intercept to impact the relative importance of encoder, and  $\mathbf{1}_n$  is one vector with  $n \times 1$ . Generally,  $\sum_{t=1}^2 \lambda_t = 1$ .

2) *Iterative Quantization With Linear Drift*: Many conventional methods [9], [30], [41] utilize rotary quantization learning to directly obtain the hash codes. However, this linear mapping is essentially a classification strategy to differentiate between similar and dissimilar samples, which can lead to substantial quantization errors. To further eliminate redundant information in common representation  $\mathbf{V}$ , we optimize this strategy by proposing an iterative quantization with linear drift constraint term to obtain a high quality hash code  $\mathbf{H}$  with discriminative power via:

$$\min_{\mathbf{H}, \mathbf{G}} \eta \|\mathbf{H} - \mathbf{G}\mathbf{V} - \mathbf{m}\mathbf{e}_n\|_F^2, \quad \text{s.t. } \mathbf{H} \in \{-1, 1\}^{k \times n}, \quad (5)$$

where  $\mathbf{G}$  is projection matrix,  $\mathbf{e}_n$  denotes identity vector with  $1 \times n$ , and drift  $\mathbf{m}$  discriminates the samples in a stable way.

To summarize, the bilinear drift module consisting of semantic autoencoder and iterative optimization shows two positive benefits: (1) it removes irrelevant information from the instances to maximize common representation  $\mathbf{V}$ ; and (2) strengthens the association constraints between the hash code  $\mathbf{H}$  and common representation  $\mathbf{V}$ .

### C. Multiple Supervision Learning

After obtaining common the representation  $\mathbf{V}$  and  $\mathbf{H}$  in an unsupervised way, we further improve hash code generation utilizing supervisory knowledge. Previous methods [7], [30], [36], [59] typically rely on label supervision to guide hash

code learning or construct a pairwise similarity matrix  $\mathbf{S}$  to deliver the hash code  $\mathbf{H}$ :

$$\min_{\mathbf{H}} \|k\mathbf{S} - \mathbf{H}^\top \mathbf{H}\|_F^2, \quad \text{s.t. } \mathbf{H} \in \{-1, 1\}^{k \times n}, \quad (6)$$

which compares the similarity measurements between samples with the inner products of the hash codes. By minimizing this discrepancy, it ensures that similar samples have smaller inner products between hash codes.

However, constructing the matrix  $\mathbf{S}$  has both temporal and spatial complexities of  $O(n^2)$ , which adversely affects the performance of the model. To mitigate this, we propose a multiple supervision learning module utilizing a latent representation space  $\mathbf{QY}$  to replace one hash code  $\mathbf{H}$  in Eq. (6). This linear strategy not only reduces the temporal complexity from  $O(n^2)$  to  $O(n)$ , but also degrades the information loss of hash codes through relaxation. We then get the hash code under the supervision of both  $\mathbf{QY}$  and  $\mathbf{S}$ :

$$\min_{\mathbf{Q}, \mathbf{H}} \beta \|\mathbf{kS} - \mathbf{H}^\top \mathbf{QY}\|_F^2 + \epsilon \|\mathbf{H} - \mathbf{QY}\|_F^2, \quad \text{s.t. } \mathbf{H} \in \{-1, 1\}^{k \times n}. \quad (7)$$

### D. Overall Objective Function

By integrating the linear weight-based formulations from Eqs. (4), (5), and (7), we derive a unified framework:

$$\min_{\mathbf{U}_t, \mathbf{V}, \mathbf{QY}, \mathbf{G}, \mathbf{I}_t, \mathbf{m}} J(\mathbf{U}_t, \mathbf{V}, \mathbf{QY}, \mathbf{G}, \mathbf{I}_t, \mathbf{m}), \quad (8)$$

where the overall objective function of RHBD is:

$$\begin{aligned} J = & \sum_{t=1}^2 \lambda_t \|\mathbf{X}_t - \mathbf{U}_t^\top \mathbf{V}\|_F^2 + \alpha \sum_{t=1}^2 \|\mathbf{V} - \mathbf{U}_t \mathbf{X}_t + \mathbf{I}_t \mathbf{1}_n^\top\|_F^2 \\ & + \beta \|\mathbf{kS} - \mathbf{H}^\top \mathbf{QY}\|_F^2 + \epsilon \|\mathbf{H} - \mathbf{QY}\|_F^2 \\ & + \eta \|\mathbf{H} - \mathbf{G}\mathbf{V} - \mathbf{m}\mathbf{e}_n\|_F^2 + \gamma \text{Reg}(\mathbf{U}_t, \mathbf{V}, \mathbf{QY}, \mathbf{G}), \\ \text{s.t. } & \sum_{t=1}^2 \lambda_t = 1, \mathbf{H} \in \{-1, 1\}^{k \times n}, \end{aligned} \quad (9)$$

where  $\lambda_t, \alpha, \beta, \epsilon$ , and  $\gamma$  are parameters.  $\text{Reg}(\cdot) = \|\cdot\|_F^2$  serves as a regularization term to prevent model overfitting.

### E. Optimization Algorithm

This part employs an alternating optimization strategy to achieve the analytical solution of each matrix variable. The solution procedure of our RHBD is shown as follows.

1) **Updating  $\mathbf{U}_t$** : By fixing  $\mathbf{H}, \mathbf{Q}, \mathbf{V}, \mathbf{G}, \mathbf{I}_t$ , and  $\mathbf{m}$ , setting the derivative of Eq. (9) with respect to  $\mathbf{U}_t$  equal to zero and we have:

$$\begin{aligned} & \sum_{t=1}^2 (\lambda_t \mathbf{V}\mathbf{V}^\top + \gamma \mathbf{D}) \mathbf{U}_t + \sum_{t=1}^2 \alpha \mathbf{U}_t \mathbf{X}_t \mathbf{X}_t^\top \\ & = \sum_{t=1}^2 (\lambda_t \mathbf{V} + \alpha \mathbf{V} + \alpha \mathbf{I}_t \mathbf{1}_n^\top) \mathbf{X}_t^\top, \end{aligned} \quad (10)$$

which is a Sylvester equation updated by the Bartels-Stewart algorithm and  $\mathbf{U}_t$  can be obtained.

2) **Updating V**: By fixing  $\mathbf{U}_t, \mathbf{H}, \mathbf{Q}, \mathbf{G}, \mathbf{I}_t$ , and  $\mathbf{m}$ , and then taking the partial derivative of Eq. (9) over  $\mathbf{V}$  equal to zero, we get:

$$\mathbf{V} = [\sum_{t=1}^2 \lambda_t \mathbf{U}_t \mathbf{U}_t^\top + (\alpha + \gamma) \mathbf{I} + \eta \mathbf{G}^\top \mathbf{G}]^{-1} [\sum_{t=1}^2 (\lambda_t + \alpha) \mathbf{U}_t \mathbf{X}_t - \alpha \sum_{t=1}^2 \mathbf{I}_t \mathbf{1}_n^\top + \eta \mathbf{G}^\top (\mathbf{H} - \mathbf{m} \mathbf{e}_n)]. \quad (11)$$

3) **Updating Q**: By fixing  $\mathbf{U}_t, \mathbf{H}, \mathbf{V}, \mathbf{G}, \mathbf{I}_t$ , and  $\mathbf{m}$ , and letting the derivative of Eq. (9) for  $\mathbf{Q}$  equal to zero, we obtain:

$$\mathbf{Q} = [\beta \mathbf{H} \mathbf{H}^\top + (\varepsilon + \gamma) \mathbf{I}]^{-1} [k \beta \mathbf{H} \mathbf{S} \mathbf{Y}^\top + \varepsilon \mathbf{H} \mathbf{Y}^\top] [\mathbf{Y} \mathbf{Y}^\top]^{-1}. \quad (12)$$

4) **Updating G**: Fixing  $\mathbf{U}_t, \mathbf{H}, \mathbf{V}, \mathbf{Q}, \mathbf{I}_t$ , and  $\mathbf{m}$  and conducting the partial derivative of Eq. (9) for  $\mathbf{G}$  equal to zero, an analytical solution of  $\mathbf{G}$  is generated:

$$\mathbf{G} = [\eta (\mathbf{H} \mathbf{V}^\top - \mathbf{m} \mathbf{e}_n \mathbf{V}^\top)] [\eta \mathbf{V} \mathbf{V}^\top + \gamma \mathbf{I}]^{-1}. \quad (13)$$

5) **Updating m**: With  $\mathbf{U}_t, \mathbf{H}, \mathbf{V}, \mathbf{Q}, \mathbf{I}_t$ , and  $\mathbf{G}$  fixed, and by achieving the derivative of Eq. (9) for  $\mathbf{m}$  equal to zero, we compute  $\mathbf{m}$  by:

$$\mathbf{m} = [(\mathbf{H} - \mathbf{G} \mathbf{V}) \mathbf{e}_n^\top] / n. \quad (14)$$

6) **Updating  $\mathbf{I}_t$** : As  $\mathbf{m}$  does, we get:

$$\mathbf{I}_t = [(\mathbf{U}_t \mathbf{X}_t - \mathbf{V}) \mathbf{1}_n] / n. \quad (15)$$

7) **Updating H**: Since  $\mathbf{H}$  is a discrete value, we cannot directly derive it. Here, we simplify the F-norm matrix as the trace form and then Eq. (9) is reduced as:

$$\begin{aligned} \min_{\mathbf{H}} & (-2\varepsilon \text{tr}(\mathbf{H}^\top \mathbf{Q} \mathbf{Y}) - 2\eta (\text{tr}(\mathbf{H}^\top \mathbf{G} \mathbf{V}) + \text{tr}(\mathbf{H}^\top \mathbf{m} \mathbf{e}_n)) \\ & - 2k\beta \text{tr}(\mathbf{H}^\top \mathbf{Q} \mathbf{Y} \mathbf{S}^\top + \beta \text{tr}(\mathbf{H}^\top \mathbf{Q} \mathbf{Y} (\mathbf{Q} \mathbf{Y})^\top \mathbf{H}) + \text{const}), \end{aligned} \quad (16)$$

in which *const* belongs to irrelevant term. Because the term in Eq. (16) contains a hash matrix constraint  $\text{tr}(\mathbf{H}^\top \mathbf{Q} \mathbf{Y} (\mathbf{Q} \mathbf{Y})^\top \mathbf{H})$ , directly discarding this may reduce the quality of  $\mathbf{H}$ . Based on the setting in [41], we introduce two auxiliary matrices  $\mathbf{A}$  and  $\mathbf{B}$  for solving. Because the calculation process of these two matrices is simple and linear, the training efficiency of the proposed RHBD is evidently reduced. As  $\mathbf{m}$  does, we can acquire  $\mathbf{A} = \text{sgn}(-\lambda_1 (\mathbf{Q} \mathbf{Y}) (\mathbf{Q} \mathbf{Y})^\top \mathbf{H} + \omega \mathbf{H} + \mathbf{B})$ , and  $\mathbf{B} = \mathbf{B} + \omega (\mathbf{H} - \mathbf{A})$ . Finally, we calculate  $\mathbf{H}$  via:

$$\mathbf{H} = \text{sgn}[2\varepsilon \mathbf{Q} \mathbf{Y} + 2\eta (\mathbf{G} \mathbf{V} + \mathbf{m} \mathbf{e}_n) + 2k\beta \mathbf{Q} \mathbf{Y} \mathbf{S}^\top - \beta \mathbf{Q} \mathbf{Y} (\mathbf{Q} \mathbf{Y})^\top \mathbf{A} + \omega \mathbf{A} - \mathbf{B}]. \quad (17)$$

#### F. Learning Hash Functions

It is very necessary to learn the hash functions of image and text modalities for querying samples depending on matrix  $\mathbf{H}$ . When a new instance appears, we can utilize the obtained hash functions to conduct the tasks of image-text retrieval. In this paper, we employ a linear regression strategy that transforms

#### Algorithm 1 The RHBD Training Procedure

**Input:** Matrices  $\mathbf{X}_1, \mathbf{X}_2, \mathbf{Y}$ , parameters  $\alpha, \beta, \varepsilon, \eta, \gamma, \omega$ .

**Output:** Hash codes  $\mathbf{H}$ .

- 1: Randomly initialize  $\mathbf{U}_1, \mathbf{U}_2, \mathbf{V}, \mathbf{Q}, \mathbf{G}, \mathbf{m}$ .
- 2: repeat
- 3:   Update  $\mathbf{U}_1$  and  $\mathbf{U}_2$  based on Eq. (10).
- 4:   Update  $\mathbf{V}$  based on Eq. (11).
- 5:   Update  $\mathbf{Q}$  based on Eq. (12).
- 6:   Update  $\mathbf{G}$  based on Eq. (13).
- 7:   Update  $\mathbf{m}$  based on Eq. (14).
- 8:   Update  $\mathbf{I}_t$  based on Eq. (15).
- 9: until convergence.
- 10: Generate hash code  $\mathbf{H}$  based on Eq. (17).
- 11: Learn projection matrices  $\mathbf{P}_1$  and  $\mathbf{P}_2$  by Eq. (19).

input instances into the hash code to produce the projection matrix  $\mathbf{P}_t$  by:

$$\min_{\mathbf{P}_t} \sum_{t=1}^2 \|\mathbf{H} - \mathbf{P}_t \mathbf{X}_t\|_F^2 + \gamma \|\mathbf{P}_t\|_F^2, \quad (18)$$

where  $\gamma$  is an equilibrium parameter. By minimizing Eq. (18) and then taking the derivative as  $\mathbf{P}_t$  equal as zero, we can get:

$$\mathbf{P}_t = \mathbf{H} \mathbf{X}_t^\top (\mathbf{X}_t \mathbf{X}_t^\top + \gamma \mathbf{I})^{-1}. \quad (19)$$

Next, we use the sign function to binarize the product of the obtained hash functions and the query samples to generate the hash codes of the query samples by:

$$f(x_t) = \text{sgn}(\mathbf{P}_t x_t). \quad (20)$$

In training procedure, the proposed RHBD calculates the Hamming distance between matrix  $\mathbf{H}$  and  $f(x_t)$ . Then the similarity score of the hash code  $\mathbf{H}$  (original data) and query-hash code can be easily acquired in retrieval procedure.

## IV. EXPERIMENTS AND RESULTS

### A. Datasets

1) *Wiki* [22]: It is a collection of 2,866 image-text pairs sourced from Wikipedia. These pairs are classified into 10 semantic categories. Each image is reduced as a 128-dimensional SIFT feature vector, while the text is a 10-dimensional LDA feature vector. As [36] does, Wiki is divided into a query set, comprising 693 randomly selected pairs, and a retrieval and training set including the remaining 2,173 pairs.

2) *Flickr25K* [42]: It possesses 25,000 image-text pairs sourced from Flickr, containing 24 distinct classes. The images are derived as a 512-dimensional GIST feature vector and the texts are depicted a 1,386-dimensional BoW vector. As done in [15], Flickr25K randomly assigns 2,000 instances as the query set, and the resting 18,015 for training and retrieval.

3) *NUS-WIDE* [41]: It includes 269,684 image-text pairs, encompassing 81 distinct semantic concepts. The images and texts are respectively reduced as a 500-dimensional SIFT vector and a 1,000-dimensional vector in the 10 most frequent categories. Analogous to [15], 2,000 instances are randomly designated as the query set, 184,577 for retrieval, and 10,000 for training purpose.

TABLE II  
THE mAP RESULTS OF ALL COUNTERPARTS ON THE SELECTED THREE BENCHMARK DATASETS (B MEANS BITS)

Tasks	Methods	Wiki				Flickr25K				NUS-WIDE			
		16 b	32 b	64 b	128 b	16 b	32 b	64 b	128 b	16 b	32 b	64 b	128 b
ItoT	CMFH [5]	0.2324	0.2422	0.2445	0.2446	0.6439	0.6383	0.6361	0.6304	0.4640	0.4777	0.4786	0.4714
	SePH [22]	0.2418	0.2642	0.2563	0.2642	0.7033	0.7010	0.7078	0.7139	0.5816	0.5881	0.5920	0.5947
	FDCH [24]	0.2308	0.2392	0.2444	0.2485	0.7441	0.7738	0.7973	0.8038	0.6714	0.7058	0.7254	0.7336
	SRLCH [30]	0.2416	0.2563	0.2665	0.2627	0.7852	0.7913	0.8179	0.8116	0.6927	0.7294	0.7261	0.7482
	JIMFH [37]	0.2235	0.2398	0.2435	0.2413	0.6487	0.6639	0.6605	0.6660	0.5169	0.5188	0.5259	0.5459
	DRMFH [52]	0.2374	0.2359	0.2571	0.2536	0.6801	0.6870	0.7056	0.7139	0.5683	0.5714	0.5829	0.5927
	LFMH [58]	0.1965	0.2005	0.2108	0.2241	0.7563	0.7686	0.7800	0.7775	0.6573	0.6571	0.6693	0.6633
	SDMSA [57]	0.2386	0.2476	0.2424	0.2381	0.7467	0.7534	0.7566	0.7551	0.6331	0.6061	0.6329	0.6518
	FADCH [33]	0.2580	0.2738	0.2694	0.2801	0.7929	0.8101	0.8174	0.8211	0.6727	0.7027	0.7149	0.7284
	IMADS [43]	0.2356	0.2598	0.2661	0.2813	0.7300	0.7947	0.7744	0.8105	0.5948	0.6189	0.6793	0.6834
	RHBD	<b>0.2607</b>	<b>0.2901</b>	<b>0.2769</b>	<b>0.2917</b>	<b>0.8275</b>	<b>0.8468</b>	<b>0.8782</b>	<b>0.8866</b>	<b>0.7137</b>	<b>0.7531</b>	<b>0.7625</b>	<b>0.7502</b>
TtoI	CMFH [5]	0.5953	0.6107	0.6215	0.6279	0.6930	0.7062	0.7304	0.7428	0.4684	0.4904	0.4963	0.4978
	SePH [22]	0.6770	0.6784	0.6876	0.6831	0.7927	0.8017	0.8079	0.8192	0.7389	0.7476	0.7558	0.7598
	FDCH [24]	0.6310	0.6380	0.6396	0.6422	0.8658	0.8876	0.8949	0.9010	0.8264	0.8466	0.8495	0.8507
	SRLCH [30]	0.6544	0.6682	0.6439	0.6651	0.8406	0.8349	0.8867	0.8549	0.8286	0.8235	0.8388	0.8309
	JIMFH [37]	0.5953	0.6127	0.6291	0.6363	0.7003	0.7199	0.7361	0.7444	0.5619	0.5746	0.6076	0.6225
	DRMFH [52]	0.5548	0.5597	0.6056	0.6060	0.7071	0.7363	0.7722	0.7880	0.6268	0.6384	0.6588	0.6808
	LFMH [58]	0.6664	0.6600	0.6825	0.6690	0.8066	0.8154	0.8387	0.7919	0.5369	0.7591	0.7668	0.7885
	SDMSA [57]	0.6554	0.6700	0.6752	0.6618	0.8614	0.8654	0.8622	0.8563	0.7941	0.7829	0.8008	0.8133
	FADCH [33]	0.6891	0.6913	0.6898	0.6895	0.8801	0.8865	0.8889	0.8859	0.8269	0.8380	0.8411	0.8444
	IMADS [43]	0.6881	0.6778	0.6811	0.6884	0.8437	0.8916	0.8854	0.9067	0.7882	0.8194	0.8302	0.8465
	RHBD	<b>0.6910</b>	<b>0.7039</b>	<b>0.7014</b>	<b>0.6947</b>	<b>0.8851</b>	<b>0.9058</b>	<b>0.9125</b>	<b>0.9234</b>	<b>0.8517</b>	<b>0.8630</b>	<b>0.8717</b>	<b>0.8629</b>

### B. Evaluation Metric

We utilize four widely-used metrics [15]: mean Average Precision (**mAP**), normalized discounted cumulative gain(**NDCG**), **topN-Precision** curve, and **Precision-Recall** curve to assess the performance of RHBD. For mAP and NDCG, we both set the search point to 100. A higher value or coverage area illustrates better model performance. We conduct comparative experiments on both 32-bit and 64-bit lengths over **topN-Precision** and **Precision-Recall** owing to limited space.

### C. Baselines and Study Details

This work completes two kinds of tasks: ItoT (images search similar texts) and TtoI (texts search similar images). We finish three main comparative experiments of RHBD and eight state-of-the-art baselines, namely unsupervised ones CMFH [5], JIMFH [37], DRMFH [52] and supervised ones SePH [22], FDCH [24], SRLCH [30], LFMH [58], SDMSA [57], FADCH [33], and IMADS [43]. The source codes of baselines can be found in their publication papers. For fairness, we follow the same parameters of their papers on identical datasets while performing tuning on distinct datasets. The whole experiment scores are 25 randomly repeated operations and output the averaged results.

We set the parameters  $\{\alpha = 10^{-6}, \beta = 10^{-4}, \epsilon = 10^2, \eta = 10^1, \gamma = 10^{-2}, \omega = 10^{-7}\}$  on Wiki,  $\{\alpha = 10^{-3}, \beta = 10^{-1}, \epsilon = 10^4, \eta = 10^{-1}, \gamma = 10^{-2}, \omega = 10^{-1}\}$  on Flickr25K and  $\{\alpha = 10^{-3}, \beta = 10^{-5}, \epsilon = 10^1, \eta = 10^{-3}, \gamma = 10^{-2}, \omega = 10^{-2}\}$  on NUS-WIDE. Our work is conducted on platform (Intel(R) CPU @ 3.3 GHz, 10 cores, 128 GB memory) and MATLAB 2021a software.

### D. Results and Discussion

1) *Search Performance on mAP Metric* : Table II reports the mAP results of RHBD and ten baseline methods

across two image-text retrieval tasks on three datasets, where we select representative 16-bit, 32-bit, 64-bit, and 128-bit code lengths. It is evident from Table II that RHBD outperforms all counterparts across three datasets on both ItoT and TtoI retrieval tasks. Specifically, taking the mAP results on Flickr25K as an example, compared to the latest IMADS method, RHBD has achieved a great improvement in the average mAP values for ItoT and TtoI by up to 8.24% and 2.49%, respectively. Compared to the SDMSA, the average mAP values of RHBD can be improved to 4.94% on ItoT and 2.68% on TtoI tasks. Compared to the top-performing SRLCH method on Flickr25K and NUS-WIDE, RHBD achieved an average improvement of 5.83% (ItoT) and 5.24% (TtoI) on Flickr25K, and an average improvement of 2.08% (ItoT) and 3.19% (TtoI) on NUS-WIDE. These results strongly demonstrate the superior performance of the proposed RHBD in terms of retrieval accuracy.

2) *Search Performance on NDCG Metric* : Table III depicts the NDCG comparisons of all baselines on all datasets. It is observed that our RHBD possesses the best NDCG scores than almost all other baselines about two tasks on three selected datasets. Specifically, it has the best NDCG scores in 11 out of 12 cases for ItoT retrieval task and 9 out of 12 cases for TtoI retrieval task. Using Flickr25K as an example, compared to the latest IMADS, RHBD outperforms IMADS with all code lengths on ItoT task, with an average improvement of 5.37%. Although RHBD falls behind by 1.13% and 0.70% when using 64-bit and 128-bit hash code lengths on TtoI task, there is a significant improvement of 5.09% and 3.79% for 16-bit and 32-bit code lengths, respectively. RHBD demonstrates an average improvement of 4.06% on ItoT and 2.45% on TtoI tasks against the supervised linear drift FADCH method. Compared to SDMSA, RHBD possesses an average improvement of 7.17% on ItoT and 4.60% on TtoI tasks. To summarize, the achieved

TABLE III  
THE NDCG RESULTS OF ALL COUNTERPARTS ON THE SELECTED THREE BENCHMARK DATASETS (B MEANS BITS)

Tasks	Methods	Wiki				Flickr25K				NUS-WIDE			
		16 b	32 b	64 b	128 b	16 b	32 b	64 b	128 b	16 b	32 b	64 b	128 b
ItoT	CMFH [5]	0.2005	0.2150	0.2181	0.2217	0.3717	0.3704	0.3658	0.3637	0.3610	0.3757	0.3750	0.3715
	SePH [22]	0.2088	0.2202	0.2162	0.2096	0.4055	0.4018	0.4146	0.4199	0.4548	0.4657	0.4764	0.4745
	FDCH [24]	0.2297	0.2382	0.2441	0.2483	0.4233	0.4374	0.4495	0.4602	0.4903	0.5073	0.5245	0.5385
	SRLCH [30]	0.2409	0.2305	0.2511	0.2584	0.4432	0.4572	0.5007	0.5083	0.4920	0.5090	0.5586	0.5597
	JIMFH [37]	0.2062	0.2126	0.2154	0.2193	0.3881	0.3907	0.3941	0.3922	0.4257	0.4206	0.4382	0.4362
	DRMFH [52]	0.2195	0.2196	0.2287	0.2280	0.3953	0.4097	0.4176	0.4241	0.4138	0.4568	0.4681	0.4906
	LFMH [58]	0.2031	0.1958	0.2102	0.2107	0.4261	0.4576	0.4362	0.4505	0.4568	0.4617	0.4967	0.5261
	SDMSA [57]	0.2381	0.2526	0.2425	0.2441	0.4167	0.4315	0.4309	0.4400	0.4804	0.5006	0.5108	0.5254
	FADCH [33]	0.2536	0.2677	0.2799	0.2807	0.4504	0.4708	0.4822	0.4908	0.5152	0.5238	0.5462	0.5581
	IMADS [43]	0.2394	0.2651	0.2788	0.2636	0.3962	0.4164	0.4520	0.5262	0.5009	0.5158	0.5478	<b>0.5731</b>
	RHBD	<b>0.2659</b>	<b>0.2827</b>	<b>0.2943</b>	<b>0.3005</b>	<b>0.4895</b>	<b>0.5040</b>	<b>0.5133</b>	<b>0.5496</b>	<b>0.5311</b>	<b>0.5615</b>	<b>0.5699</b>	0.5667
TtoI	CMFH [5]	0.5623	0.5776	0.5850	0.5892	0.3948	0.4056	0.4262	0.4354	0.3679	0.3859	0.3935	0.3956
	SePH [22]	0.6473	0.6524	0.6552	0.6623	0.4635	0.4713	0.4851	0.4923	0.5869	0.5973	0.6166	0.6175
	FDCH [24]	0.6303	0.6374	0.6392	0.6421	0.5190	0.5373	0.5416	0.5542	0.6513	0.6676	0.6746	0.6819
	SRLCH [30]	0.6443	0.6743	0.6595	0.6886	0.5066	0.5048	0.5326	0.5524	0.6106	0.6305	0.6631	0.6534
	JIMFH [37]	0.5664	0.5817	0.5927	0.6024	0.4002	0.4199	0.4294	0.4380	0.4510	0.4387	0.4822	0.4761
	DRMFH [52]	0.5363	0.5639	0.5594	0.5674	0.4063	0.4468	0.4664	0.4787	0.4402	0.5001	0.5324	0.5508
	LFMH [58]	0.6346	0.6281	0.6764	0.6625	0.4011	0.4898	0.4858	0.5078	0.4952	0.5581	0.5770	0.5873
	SDMSA [57]	0.6694	0.6653	0.6696	0.6645	0.5060	0.5254	0.5274	0.5244	0.6408	0.6571	0.6594	0.6719
	FADCH [33]	0.6722	0.6835	0.6857	0.6873	0.5279	0.5479	0.5546	0.5607	0.6697	0.6792	0.6908	0.6965
	IMADS [43]	0.6645	0.6894	0.6876	0.6791	0.4921	0.5201	<b>0.5642</b>	<b>0.6202</b>	0.6489	0.6721	0.6920	<b>0.7171</b>
	RHBD	<b>0.6846</b>	<b>0.7070</b>	<b>0.6981</b>	<b>0.6976</b>	<b>0.5542</b>	<b>0.5577</b>	0.5636	0.6132	<b>0.6855</b>	<b>0.7021</b>	<b>0.7007</b>	0.7066

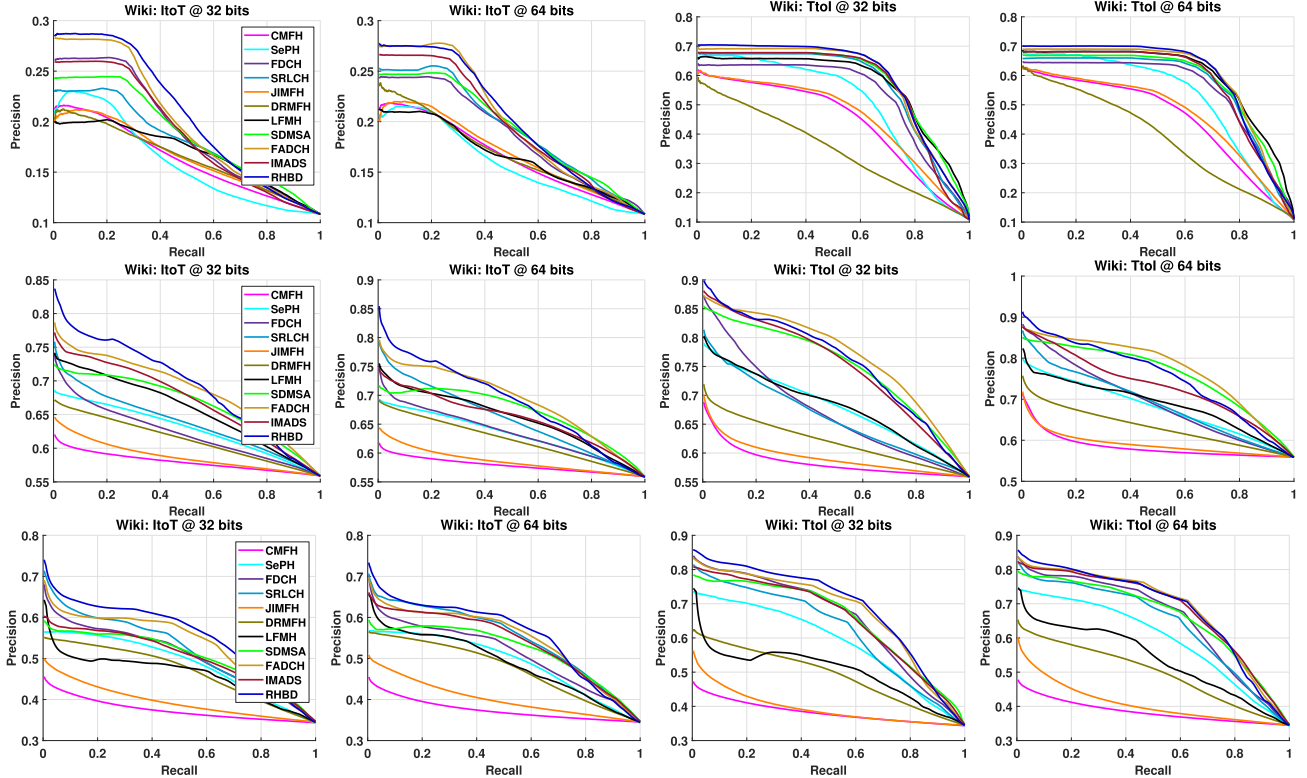


Fig. 3. The Precision-Recall curves of all alternatives with 32 and 64 code lengths on the selected three datasets.

observations illustrate the effectiveness of our RHBD in NDCG metric.

3) *Search Performance on Precision–Recall Metric* : Figure 3 represents the Precision-Recall curves of RHBD and ten benchmark methods on Wiki, Flickr25K, and NUS-WIDE. We observe that RHBD consistently conducts the best precision than other ten baselines in 10 out of 22 cases for the

TtoI task on Flickr25K. This is mainly attributed to the fact that RHBD effectively utilizes multiple supervisory knowledge to generate high-quality hash codes. Besides, the reduced redundant feature can decrease the errors in hash bits and further enhance the robustness of hash codes.

4) *Search Performance on topN – Precision Metric* : Figure 4 shows the topN-Precision curves of RHBD along

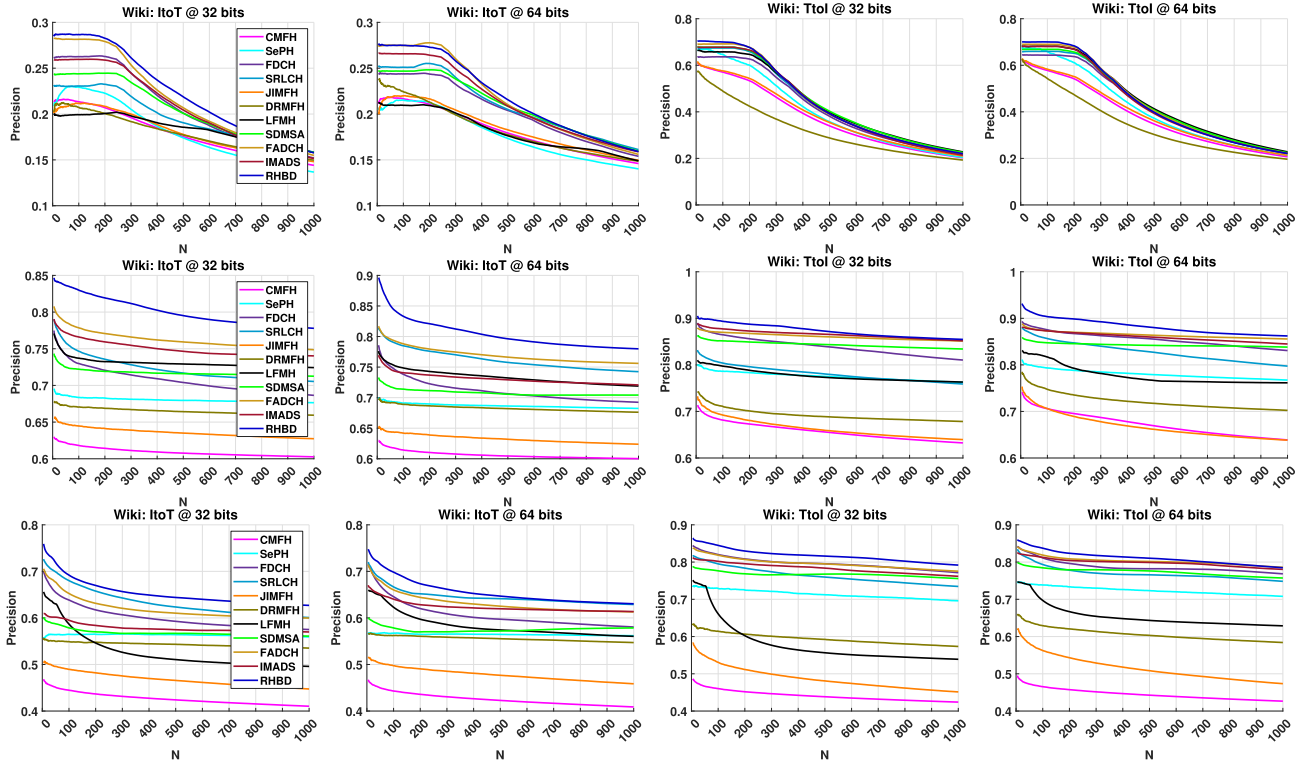


Fig. 4. The topN-Precision curves of all alternatives with 32 and 64 code lengths on the selected three datasets.

TABLE IV  
THE mAP VALUES OF OUR RHBD AND RELATED VARIANTS ON THE SELECTED THREE BENCHMARK DATASETS

Tasks	Methods	Wiki				Flickr25K				NUS-WIDE			
		16 b	32 b	64 b	128 b	16 b	32 b	64 b	128 b	16 b	32 b	64 b	128 b
ItoT	RHBD-1	0.2597	0.2840	<b>0.2842</b>	0.2892	0.7988	0.8319	0.8755	0.8810	0.7161	0.7332	0.7593	<b>0.7614</b>
	RHBD-2	0.2518	0.2696	0.2688	0.2749	0.7947	0.8327	0.8772	0.8657	0.6936	0.7437	0.7558	0.7548
	RHBD-3	0.2377	0.2787	0.2798	0.2849	0.7924	0.8253	0.8668	0.8741	0.6971	0.7476	0.7424	0.7523
	RHBD-4	0.2581	0.2711	0.2733	0.2818	0.7815	0.8031	0.8219	0.8473	0.6539	0.7436	0.7108	0.7333
	RHBD-5	0.1477	0.2061	0.2588	0.2884	0.7752	0.8329	0.8421	0.8578	<b>0.7294</b>	0.7477	0.7282	0.7327
	RHBD-6	0.1555	0.1514	0.1511	0.1537	0.5747	0.5805	0.5786	0.5791	0.3709	0.3721	0.3740	0.3712
	RHBD	<b>0.2607</b>	<b>0.2901</b>	0.2769	<b>0.2917</b>	<b>0.8275</b>	<b>0.8468</b>	<b>0.8782</b>	<b>0.8866</b>	0.7137	<b>0.7531</b>	<b>0.7625</b>	0.7502
TtoI	RHBD-1	0.6792	0.6787	0.6856	0.6928	<b>0.8888</b>	0.8985	0.9061	0.9142	0.8438	0.8579	0.8620	0.8595
	RHBD-2	0.6720	0.6884	0.6869	0.6901	0.8805	0.8960	0.9066	0.9173	0.8260	0.8619	0.8625	0.8613
	RHBD-3	0.6778	0.6768	0.6799	0.6898	0.8698	0.8904	0.9115	0.9166	0.8435	0.8615	0.8589	0.8586
	RHBD-4	0.6898	0.6941	0.6841	0.6900	0.8808	0.8869	0.9011	0.9143	0.8172	0.8484	0.8419	0.8499
	RHBD-5	0.1595	0.3377	0.6803	0.6929	0.8631	0.8787	0.8892	0.9121	0.8218	0.8305	0.8521	0.8563
	RHBD-6	0.1647	0.1700	0.1945	0.2259	0.5747	0.5781	0.5822	0.5879	0.3732	0.3751	0.3731	0.3778
	RHBD	<b>0.6910</b>	<b>0.7039</b>	<b>0.7014</b>	<b>0.6947</b>	0.8851	<b>0.9058</b>	<b>0.9125</b>	<b>0.9234</b>	<b>0.8517</b>	<b>0.8630</b>	<b>0.8717</b>	<b>0.8629</b>

with ten baseline methods. It is clearly noticed that our RHBD outperforms all comparison methods on all retrieval tasks. Although the precision scores of RHBD over the ToI retrieval task on Wiki are not outstanding as the retrieval points  $N$  increases, the coverage area enclosed by the curve and the axes remains still the largest. Therefore, the result of this experiment on topN-Precision criterion further demonstrates the effectiveness of our proposed model.

#### E. Ablation Work

To substantiate the efficacy of each module over RHBD, we establish six variant types for comparisons. Concretely, RHBD-1 eliminates the drift  $I_t$ , RHBD-2 omits the drift  $m$ ,

RHBD-3 removes both drifts concurrently. RHBD-4 is a type that only utilizes label supervision ( $\beta = 0$ ), while RHBD-5 solely employs semantic supervision ( $\epsilon = 0$ ), RHBD-6 does not use any supervision ( $\beta = 0$  and  $\epsilon = 0$ ). Table IV shows the results of the ablation experiments on all datasets. Evidently, we observe that the mAP scores of RHBD outperform other types of tasks on the ItoT task (9 of 12), and it also performs better than other types of tasks on the TtoI task (11 of 12). Thus, these results show the efficacy of the RHBD modules.

#### F. Parameter Analysis

Among five important parameters in Eq. (9),  $\alpha$  represents the impact of the encoder part on the shared representation  $\mathbf{V}$ ,

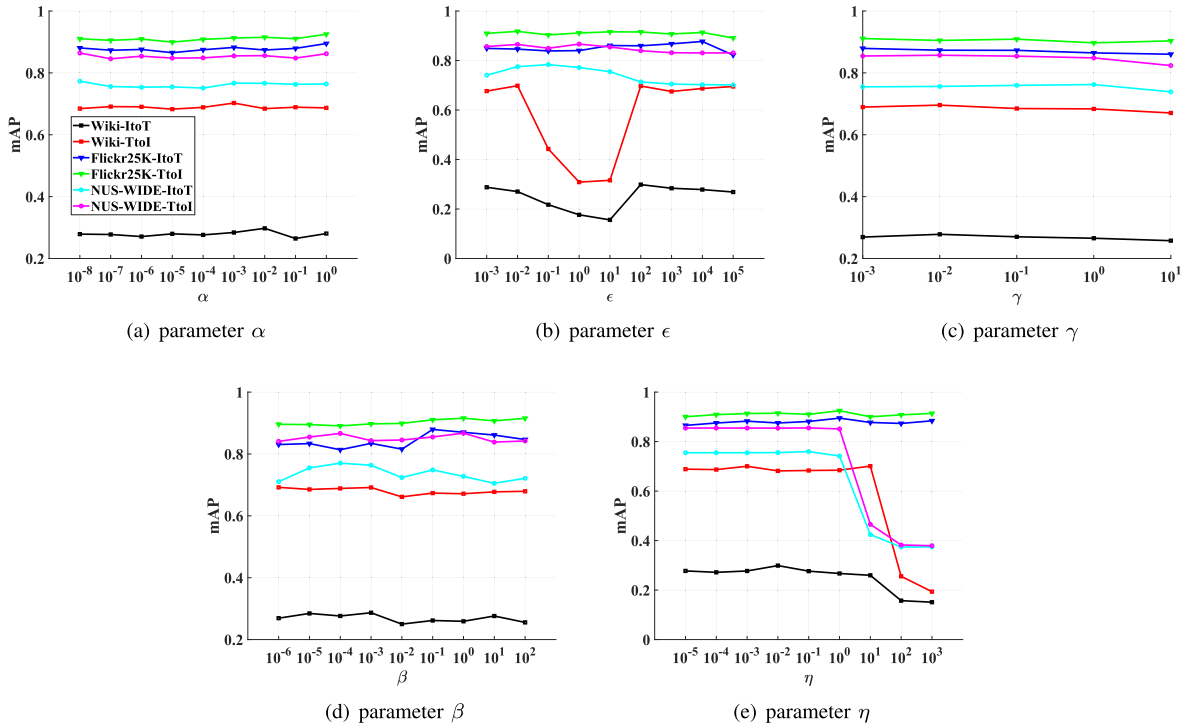


Fig. 5. Model parameters analysis of RHBD @ 64-bit. (a)-(e) are parameter variations on the selected three datasets.

$\beta$  measures the influence of the label supervision in hash code learning,  $\epsilon$  indicates the quality of hash codes under semantic supervision,  $\eta$  reflects the effects of hash function learning, and  $\gamma$  is regularization to prevent overfitting. To understand the effects of these parameters on the model performance across different datasets, we conduct a sensitivity analysis experiment. We set  $\lambda_1 = \lambda_2 = 0.5$ . The results in Figure 5, consist of five smaller diagrams illustrating the mAP values under different values of the five main parameters. We explore the relationships between  $\beta$  and  $\epsilon$ , as well as  $\beta$  and  $\eta$  for 64-bit code lengths. These experiments help refine the final parameter settings of the model.

In addition, we observe that for Figure 5 (b), for the  $\epsilon$  parameter on Wiki, values within the range of  $[10^{-1}, 10]$  resulted in lower mAP values. This is possibly attributed to the model's tendency to overlook intermodal relationships when the parameter approaches zero, particularly in a small dataset like Wiki where the association between text and imagery may not be effectively utilized, which can lead to reduced mAP values. Additionally, from Figure 5 (e), we find that on both Wiki and NUS-WIDE, when the value of  $\eta$  is greater than 1, the mAP values are smaller. This may be due to differences in dataset characteristics, distributions, and the influence of scale and variability and the data distribution of Flickr25K appears to be more suitable for model training and optimization procedures.

#### G. Kernelization Research

To explore the influence of different kernel numbers on performance, we conduct mAP results to evaluate changes in kernel anchors, which is shown in Figure 8. The results

indicate that RHBD has better results when the anchors number falls within the range of 1,000 to 2,000 on all datasets. RHBD achieves superior performance when the anchors number is in the respective ranges of 2,500 to 3,500 and 2,000 to 3,000. To ensure the model have satisfactory and robust accuracy, we respectively take 2,000, 3,000 and 2,000 as the final kernel number on all datasets.

#### H. Visualization

To visually show the query results of RHBD, we conduct a visualization experiment using 64-bit on Wiki. Figure 7 displays the top 10 results of RHBD about different classes on two kinds of retrieval tasks, where a green box represents a similar sample and a red box represents a dissimilar sample. We observe from Figure 7 the following: 1) As for ItoT task, the retrieval results achieve the majority of similar samples in almost situations (9 of 10) except for two dissimilar biology and royalty classes. 2) RHBD obtains similar samples in almost all cases (29 of 30) on three distinct query classes while music has a dissimilar sample. Despite the dissimilar query samples, they often rank top 9 and 10 which are in the bottom positions. These findings illustrate that RHBD efficiently conduct the image-text retrieval task while maintaining high querying accuracy.

#### I. Convergence Validation

Figure 9 displays the convergence curves of RHBD with 64-bit length on all datasets. The ordinate of this figure is normalized, meaning that the objective function value is divided by the maximum value. The convergence curves on all datasets drop sharply and eventually reach stability as the

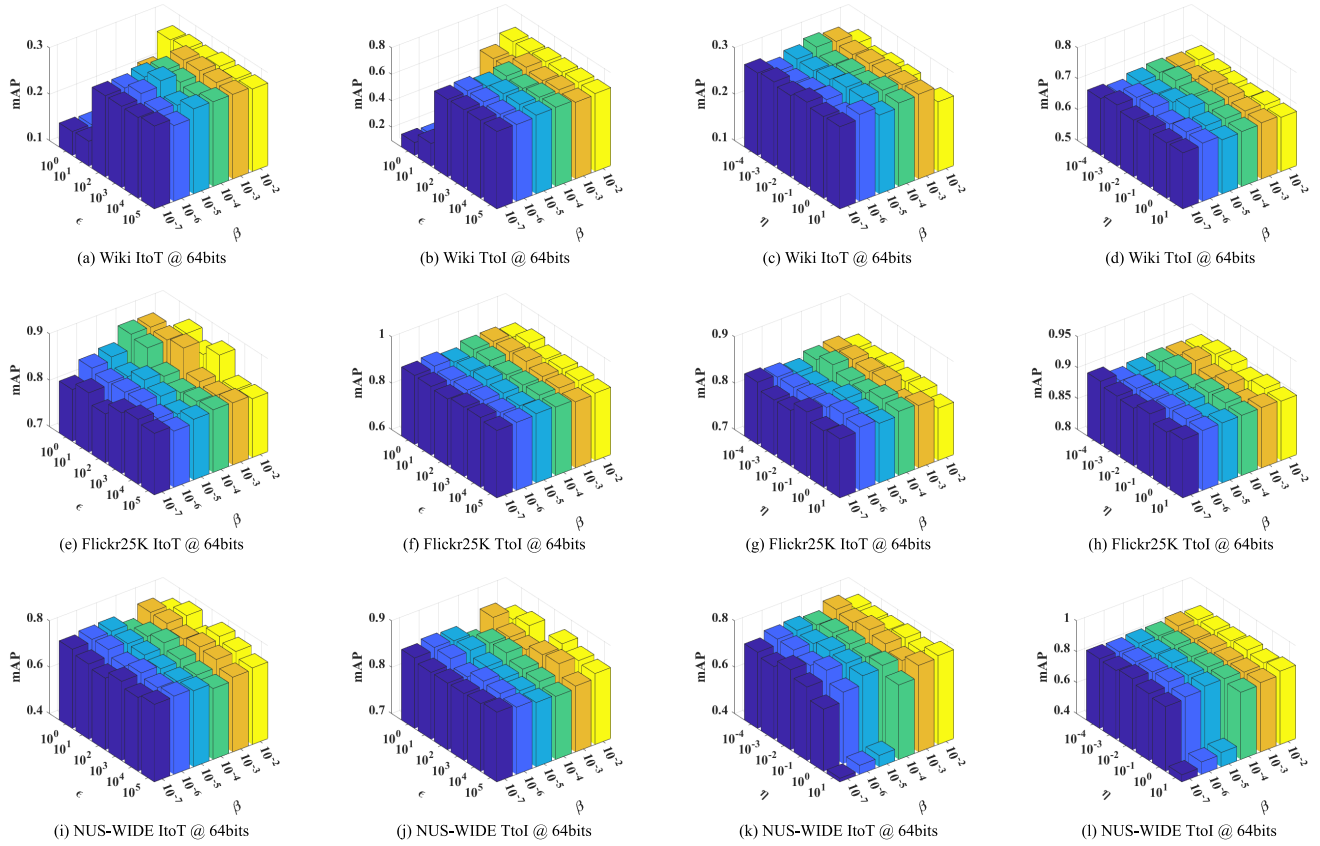


Fig. 6. The mAP scores of RHBD with varying parameters on the selected three datasets.



















































































Task	Query	TOP1 → TOP10									
ItoT											
	biology										
	geography										
	royalty										
TtoI											
	literature										
	music										
	sport										

Fig. 7. The visualization results of RHBD against distinct query classes on Wiki.

iterations number increases. Notably, RHBD keep robust about 5 iterations on Wiki, and within the range of 15 iterations on Flickr25K and NUS-WIDE. Thus, we have chosen 15 as the optimal number of iterations for our RHBD.

#### J. Failure Case and Next Exploration

Although achieving good retrieval performance across four main evaluation metrics and other ablation studies, our RHBD

has some limitations in large cross-modal applications. For instance, our method consumes more training time than competing SDMSA and FADCH methods in Figure 10. The training efficiency of the RHBD model is mainly influenced by the hyperparameters and the optimization process during training. The main reasons are as follows: (1) Our overall objective function composed of multiple constraint terms, improves the retrieval accuracy but increases training time.

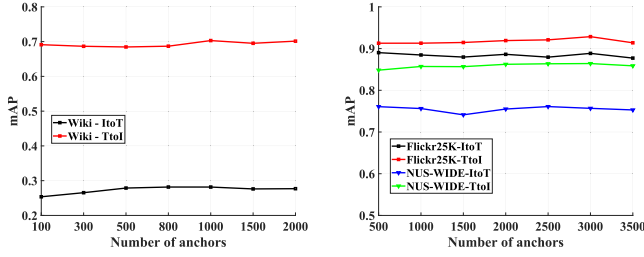


Fig. 8. The kernelization curves of our RHBD on three datasets.

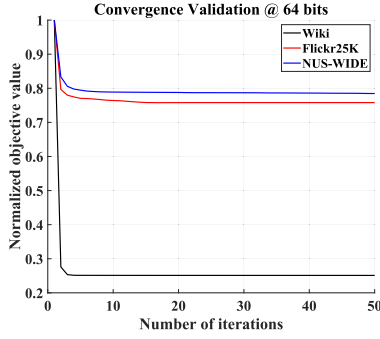
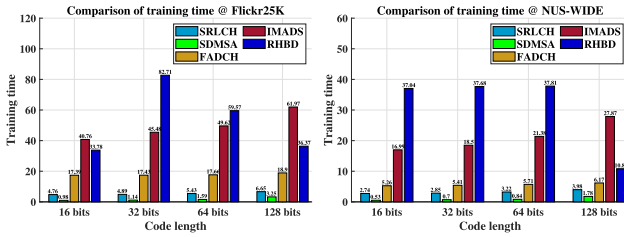


Fig. 9. The convergence curves of RHBD on public benchmark datasets.



(a) Training time @ Flickr25K (b) Training time @ NUS-WIDE

Fig. 10. Training time results of RHBD and four competitors @ on larger Flickr25K and NUS-WIDE datasets.

Specifically, Eq. (9) mainly includes four parameters  $\alpha, \beta, \epsilon, \eta, \gamma$ . Thereinto, We set  $\lambda_1 = \lambda_2 = 0.5$  for the experiment validation, and  $\gamma$  has a wide parameter range with minimal impact.  $\omega$  is an auxiliary parameter with a small weight. (2) We mitigate the redundancy of supervisory information by introducing auxiliary matrices or linear biases. Although reducing the matrix dimension of multiple supervised knowledge, RHBD inevitably sacrifices training efficiency.

To sum up, we struggle to reduce the impacts of multi-source supervision and multiple hyperparameters to enhance the performance of RHBD, while striving to maintain comparable training efficiency to competitive alternatives. Meanwhile, the convex optimization strategy will be a focus of future work.

## V. COMPARISON WITH DEEP HASHING METHODS

To further verify the superiority of RHBD in deep feature application, eleven deep hashing baselines are selected to compare with RHBD on Flickr25K. Concretely, we exact a 4096-dimensional image deep feature and 1386-dimensional BoW text feature as the model input. For baselines, the

TABLE V  
THE mAP COMPARISONS BETWEEN OUR RHBD AND DEEP HASHING METHODS ON FLICKR25K

Methods	ItoT			TtoI		
	16 b	32 b	64 b	16 b	32 b	64 b
DCMH [14]	0.7410	0.7465	0.7485	0.7827	0.7900	0.7932
SSAH [16]	0.7820	0.7900	0.8000	0.7910	0.7950	0.8030
EGDH [31]	0.7569	0.7729	0.7959	0.7787	0.7939	0.7985
MLCAH [26]	0.7960	0.8080	0.8150	0.7940	0.8050	0.8050
DADH [2]	0.8020	0.8072	0.8179	0.7920	0.7959	0.8064
CPAH [47]	0.7890	0.7960	0.7950	0.7780	0.7860	0.7850
MLSPH [61]	0.8076	0.8235	0.8337	0.7852	0.8041	0.8146
DMFH [27]	0.7802	0.7919	0.7946	0.7978	0.8097	0.8101
DDCHms [54]	0.7394	0.7450	0.7575	0.7596	0.7742	0.7847
UCCH [13]	0.7498	0.7569	0.7587	0.7507	0.7575	0.7596
GCDH [1]	0.8352	0.8553	0.8625	<b>0.8159</b>	<b>0.8226</b>	<b>0.8322</b>
RHBDcnn	<b>0.8413</b>	<b>0.8531</b>	<b>0.8655</b>	0.7924	0.7923	0.8018

mAP scores are obtained by their original paper. For ease of comparison, we set  $R$  as the size of the query set on mAP metric. The completed deep variant is called RHBDcnn. Figure (V) reports the mAP scores between RHBDcnn and eleven deep hashing methods. We observe that RHBDcnn get the best over others on ItoT task and the results of RHBDcnn are slightly lower than that of MLSPH, DMFH, and GCDH on TtoI task. The possible explanation is that such three methods train image and text features online in an end-to-end manner, while our variation extracts shallow features offline. In addition, the training time of RHBDcnn consumes about 23 seconds and baselines needs more than 3 hours on the training time. Thus, the deep variant RHBDcnn has comparable accuracy to the deep baselines and the best training efficiency, which shows the superiority of our RHBD method in terms of deep framework.

## VI. CONCLUSION

We present a novel supervised hashing model (RHBD) for image-text retrieval, demonstrating its superiority over several state-of-the-art baselines while achieving training efficiency comparable to the competitive methods and validating the effectiveness of the proposed learning modules across various cross-modal datasets. Specifically, introducing the well-designed bilinear drift hashing component to effectively distinguish similar and dissimilar original image-text instances, as well as multiple supervision component to obtain abundant supervision knowledge. Then, RHBD has promoted the representation ability of hash codes by fully integrating beneficial features and supervision properties during training. The future study will utilize discrete optimal transport theory to design a supervised hashing algorithm for incomplete image-text data pairs while eliminating redundant features.

## REFERENCES

- [1] C. Bai, C. Zeng, Q. Ma, and J. Zhang, "Graph convolutional network discrete hashing for cross-modal retrieval," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4756–4767, Apr. 2024.
- [2] C. Bai, C. Zeng, Q. Ma, J. Zhang, and S. Chen, "Deep adversarial discrete hashing for cross-modal retrieval," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2020, pp. 525–531.
- [3] P. D. V. Chaves, B. L. Pereira, and R. L. T. Santos, "Efficient online learning to rank for sequential music recommendation," in *Proc. ACM Web Conf.*, Lyon, France, Apr. 2022, pp. 2442–2450.

- [4] Y. Chen, Y. Fang, Y. Zhang, and I. King, "Bipartite graph convolutional hashing for effective and efficient top-N search in Hamming space," in *Proc. ACM Web Conf.*, Austin, TX, USA, Apr. 2023, pp. 3164–3172.
- [5] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 2083–2090.
- [6] 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 Trans. Cybern.*, vol. 52, no. 11, pp. 11780–11793, Nov. 2022.
- [7] X. Fang, Z. Liu, N. Han, L. Jiang, and S. Teng, "Discrete matrix factorization hashing for cross-modal retrieval," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 10, pp. 3023–3036, Oct. 2021.
- [8] Y. Fang, B. Li, X. Li, and Y. Ren, "Unsupervised cross-modal similarity via latent structure discrete hashing factorization," *Knowledge-Based Syst.*, vol. 218, Apr. 2021, Art. no. 106857.
- [9] X. Gu, G. Dong, X. Zhang, L. Lan, and Z. Luo, "Semantic-consistent cross-modal hashing for large-scale image retrieval," *Neurocomputing*, vol. 433, pp. 181–198, Apr. 2021.
- [10] W. Guan, X. Song, H. Zhang, M. Liu, C.-H. Yeh, and X. Chang, "Bi-directional heterogeneous graph hashing towards efficient outfit recommendation," in *Proc. 30th ACM Int. Conf. Multimedia*, Lisboa, Portugal, Oct. 2022, pp. 268–276.
- [11] W. Guan et al., "Partially supervised compatibility modeling," *IEEE Trans. Image Process.*, vol. 31, pp. 4733–4745, 2022.
- [12] C. Hansen, C. Hansen, J. G. Simonsen, S. Alstrup, and C. Lioma, "Unsupervised multi-index semantic hashing," in *Proc. Web Conf.*, Ljubljana, Slovenia, Apr. 2021, pp. 2879–2889.
- [13] P. Hu, H. Zhu, J. Lin, D. Peng, Y.-P. Zhao, and X. Peng, "Unsupervised contrastive cross-modal hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3877–3889, Mar. 2023.
- [14] Q. Jiang and W. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3270–3278.
- [15] Q.-Y. Jiang and W.-J. Li, "Discrete latent factor model for cross-modal hashing," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3490–3501, Jul. 2019.
- [16] C. Li, C. Deng, N. Li, W. Liu, X. Gao, and D. Tao, "Self-supervised adversarial hashing networks for cross-modal retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4242–4251.
- [17] H. Li, C. Zhang, X. Jia, Y. Gao, and C. Chen, "Adaptive label correlation based asymmetric discrete hashing for cross-modal retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1185–1199, Feb. 2023.
- [18] L. Li, Z. Shu, Z. Yu, and X.-J. Wu, "Robust online hashing with label semantic enhancement for cross-modal retrieval," *Pattern Recognit.*, vol. 145, Jan. 2024, Art. no. 109972.
- [19] W. Li, Z. Ma, L.-J. Deng, X. Fan, and Y. Tian, "Neuron-based spiking transmission and reasoning network for robust image-text retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 7, pp. 3516–3528, Jul. 2023.
- [20] W. Li, Z. Ma, L.-J. Deng, P. Wang, J. Shi, and X. Fan, "Reservoir computing transformer for image-text retrieval," in *Proc. 31st ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2023, pp. 5605–5613.
- [21] W. Li et al., "Semantic constraints matrix factorization hashing for cross-modal retrieval," *Comput. Electr. Eng.*, vol. 100, May 2022, Art. no. 107842.
- [22] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3864–3872.
- [23] H. Liu, R. Ji, Y. Wu, F. Huang, and B. Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6345–6353.
- [24] X. Liu, X. Nie, W. Zeng, C. Cui, L. Zhu, and Y. Yin, "Fast discrete cross-modal hashing with regressing from semantic labels," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1662–1669.
- [25] Z. Liu, F. Chen, J. Xu, W. Pei, and G. Lu, "Image-text retrieval with cross-modal semantic importance consistency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 5, pp. 2465–2476, May 2023.
- [26] X. Ma, T. Zhang, and C. Xu, "Multi-level correlation adversarial hashing for cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 22, no. 12, pp. 3101–3114, Dec. 2020.
- [27] X. Nie, B. Wang, J. Li, F. Hao, M. Jian, and Y. Yin, "Deep multiscale fusion hashing for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 401–410, Jan. 2021.
- [28] J. Qin et al., "Discrete semantic matrix factorization hashing for cross-modal retrieval," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Milan, Italy, Jan. 2021, pp. 1550–1557.
- [29] Y. Shao, J. Sun, Y. Jiang, and J. Li, "Dual-grained text-image olfactory matching model with mutual promotion stages," in *Proc. ACM Web Conf.*, Austin, TX, USA, Apr. 2023, pp. 669–677.
- [30] H. T. Shen et al., "Exploiting subspace relation in semantic labels for cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3351–3365, Oct. 2021.
- [31] Y. Shi, X. You, F. Zheng, S. Wang, and Q. Peng, "Equally-guided discriminative hashing for cross-modal retrieval," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 2019, pp. 4767–4773.
- [32] J. Tang, K. Wang, and L. Shao, "Supervised matrix factorization hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3157–3166, Jul. 2016.
- [33] S. Teng, C. Ning, W. Zhang, N. Wu, and Y. Zeng, "Fast asymmetric and discrete cross-modal hashing with semantic consistency," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 2, pp. 577–589, Apr. 2023.
- [34] R.-C. Tu, X.-L. Mao, J.-N. Guo, W. Wei, and H. Huang, "Partial-softmax loss based deep hashing," in *Proc. Web Conf.*, Ljubljana, Slovenia, Apr. 2021, pp. 2869–2878.
- [35] R.-C. Tu et al., "Unsupervised cross-modal hashing with modality-interaction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 5296–5308, Sep. 2023.
- [36] D. Wang, X. Gao, X. Wang, and L. He, "Label consistent matrix factorization hashing for large-scale cross-modal similarity search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2466–2479, Oct. 2019.
- [37] D. Wang, Q. Wang, L. He, X. Gao, and Y. Tian, "Joint and individual matrix factorization hashing for large-scale cross-modal retrieval," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107479.
- [38] H. Wang et al., "DANCE: Learning a domain adaptive framework for deep hashing," in *Proc. ACM Web Conf.*, Austin, TX, USA, Apr. 2023, pp. 3319–3330.
- [39] H. Wang, M. Yao, G. Jiang, Z. Mi, and X. Fu, "Graph-collaborated auto-encoder hashing for multiview binary clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 10121–10133, Jul. 2024.
- [40] L. Wang, M. Zareapoor, J. Yang, and Z. Zheng, "Asymmetric correlation quantization hashing for cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 24, pp. 3665–3678, 2022.
- [41] S. Wang, H. Zhao, and K. Li, "Discrete joint semantic alignment hashing for cross-modal image-text search," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 8022–8036, Nov. 2022.
- [42] S. Wang, H. Zhao, and K. Nai, "Learning a maximized shared latent factor for cross-modal hashing," *Knowledge-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107252.
- [43] S. Wang, H. Zhao, Z. Zhang, and K. Li, "Individual mapping and asymmetric dual supervision for discrete cross-modal hashing," *Expert Syst. Appl.*, vol. 247, Aug. 2024, Art. no. 123333.
- [44] X. Wang, Y. Lin, and X. Li, "CgAT: Center-guided adversarial training for deep hashing-based retrieval," in *Proc. ACM Web Conf.*, Austin, TX, USA, Apr. 2023, pp. 3268–3277.
- [45] Z. Wang, Z. Gao, K. Guo, Y. Yang, X. Wang, and H. Tao Shen, "Multilateral semantic relations modeling for image text retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2830–2839.
- [46] Y. Wu, S. Wang, and Q. Huang, "Multi-modal semantic autoencoder for cross-modal retrieval," *Neurocomputing*, vol. 331, pp. 165–175, Feb. 2019.
- [47] D. Xie, C. Deng, C. Li, X. Liu, and D. Tao, "Multi-task consistency-preserving adversarial hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 29, pp. 3626–3637, 2020.
- [48] F. Yang, Y. Liu, X. Ding, F. Ma, and J. Cao, "Asymmetric cross-modal hashing with high-level semantic similarity," *Pattern Recognit.*, vol. 130, Oct. 2022, Art. no. 108823.
- [49] F. Yang, Q.-X. Zhang, X.-J. Ding, F.-M. Ma, J. Cao, and D.-Y. Tong, "Semantic preserving asymmetric discrete hashing for cross-modal retrieval," *Int. J. Speech Technol.*, vol. 53, no. 12, pp. 15352–15371, Jun. 2023.
- [50] Z. Yang, X. Deng, and J. Long, "Fast unsupervised consistent and modality-specific hashing for multimedia retrieval," *Neural Comput. Appl.*, vol. 35, no. 8, pp. 6207–6223, Mar. 2023.
- [51] T. Yao et al., "Efficient discrete supervised hashing for large-scale cross-modal retrieval," *Neurocomputing*, vol. 385, pp. 358–367, Apr. 2020.

- [52] T. Yao et al., "Discrete robust matrix factorization hashing for large-scale cross-media retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1391–1401, Feb. 2023.
- [53] T. Yao et al., "Efficient supervised graph embedding hashing for large-scale cross-media retrieval," *Pattern Recognit.*, vol. 145, Jan. 2024, Art. no. 109934.
- [54] E. Yu, J. Ma, J. Sun, X. Chang, H. Zhang, and A. G. Hauptmann, "Deep discrete cross-modal hashing with multiple supervision," *Neurocomputing*, vol. 486, pp. 215–224, May 2022.
- [55] C. Zhang, H. Li, Y. Gao, and C. Chen, "Weakly-supervised enhanced semantic-aware hashing for cross-modal retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6475–6488, Jun. 2023.
- [56] D. Zhang and X.-J. Wu, "Robust and discrete matrix factorization hashing for cross-modal retrieval," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108343.
- [57] D. Zhang and X.-J. Wu, "Scalable discrete matrix factorization and semantic autoencoder for cross-media retrieval," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 5947–5960, Jul. 2022.
- [58] D. Zhang, X.-J. Wu, and J. Yu, "Label consistent flexible matrix factorization hashing for efficient cross-modal retrieval," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 3, pp. 1–18, Aug. 2021.
- [59] P.-F. Zhang, Y. Luo, Z. Huang, X.-S. Xu, and J. Song, "High-order nonlocal hashing for unsupervised cross-modal retrieval," *World Wide Web*, vol. 24, no. 2, pp. 563–583, Mar. 2021.
- [60] X. Zhang, X. Niu, P. Fournier-Viger, and X. Dai, "Image-text retrieval via preserving main semantics of vision," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Brisbane, QLD, Australia, Jul. 2023, pp. 1967–1972.
- [61] X. Zou, X. Wang, E. M. Bakker, and S. Wu, "Multi-label semantics preserving based deep cross-modal hashing," *Signal Process., Image Commun.*, vol. 93, Apr. 2021, Art. no. 116131.



**Song Wang** received the master's degree from Changsha University of Science and Technology, China, in 2017, and the Ph.D. degree in computer science and technology from Hunan University, Changsha, China, in 2022. He is currently a Post-Doctoral Researcher with Hunan University. To date, he has authored more than ten publications in journals and conference proceedings. His research focuses on image processing, multimedia analysis and retrieval, pattern recognition, and computer vision.



**Zixing Zhang** (Senior Member, IEEE) received the master's degree in physical electronics from Beijing University of Posts and Telecommunications (BUPT), China, in 2010, and the Ph.D. degree in computer engineering from the Technical University of Munich (TUM), Germany, in 2015. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, China. From 2017 to 2019, he was a Research Associate with the Department of Computing, Imperial College London (ICL), U.K. Before that, he was a Post-Doctoral Researcher with the University of Passau, Germany. To date, he has authored more than 110 publications in peer-reviewed books, journals, and conference proceedings, leading to more than 5500 citations (H-index 45). His research focuses on human-centred emotion and health computation. He serves as an Associate Editor for IEEE TRANSACTIONS ON AFFECTIVE COMPUTING and the *Frontiers in Signal Processing*, an Editorial Board Member of the *Scientific Reports* (Nature), and a Guest Editor of IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.



**Huan Zhao** received the B.S., M.S., and Ph.D. degrees in computer science and technology from Hunan University, Changsha, China, in 1989, 2004, and 2010, respectively. She is currently a Professor with the School of Information Science and Technology, Hunan University. She has published over 100 research papers in international journals and conferences, including *Information Processing and Management*, *Knowledge-Based Systems*, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and IEEE International Conference on Acoustics, Speech and Signal Processing. Her current research interests include speech signal processing, cross-media retrieval, and natural language processing.



**Zeyi Li** received the bachelor's degree in computer science and technology from Hunan Normal University, where he is currently pursuing the M.S. degree in electronic information. His research interests include cross-modal semantic hashing retrieval.



**Keqin Li** (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University in 1985 and the Ph.D. degree in computer science from the University of Houston in 1990. He is currently a SUNY Distinguished Professor with the State University of New York and a National Distinguished Professor with Hunan University, China. He has authored or co-authored more than 1000 journal articles, book chapters, and refereed conference papers. He received several best paper awards from international conferences, including PDPTA-1996, NAECON-1997, IPDPS-2000, ISPA-2016, NPC-2019, ISPA-2019, and CPSCOM-2022. He holds nearly 75 patents announced or authorized by Chinese National Intellectual Property Administration. He is a member of the SUNY Distinguished Academy. He is an AAAS Fellow, an AAIA Fellow, and an ACIS Founding Fellow. He is an Academician Member and a fellow of the International Artificial Intelligence Industry Alliance. He is a member of Academia Europaea (Academician of the Academy of Europe). He is among the world's top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of the Scopus citation database. He was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis Who's Who in Science and Engineering, Who's Who in America, Who's Who in the World, and Who's Who in American Education for over twenty consecutive years. He received the Distinguished Alumnus Award from the Computer Science Department, University of Houston, in 2018. He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023. He won the IEEE Region 1 Technological Innovation Award (Academic) in 2023.