

Relaxed Energy Preserving Hashing for Image Retrieval

Yuan Sun¹, Jian Dai, Zhenwen Ren², *Member, IEEE*, Qilin Li, and Dezhong Peng

Abstract—Image retrieval is the eye of industrial robots, which determines the performance of machine visual search, street view search, and object grasping. Learning to hash, as a promising technique, has attracted much attention. Existing image hashing methods often directly learn hash codes by a single hash function. Despite their success, they suffer from the following limits: 1) It is difficult to perfectly preserve the intrinsic structure of the data using a single-layer hash function to generate discriminative hash codes; 2) they unconsciously ignore the main energy information of the original data, which lead to severe information loss of low-dimensional hash codes. To alleviate these issues, we propose a concise yet effective Relaxed Energy Preserving Hashing (REPH) method. Specifically, we utilize a two-layer hash function to provide more flexibility, thereby learning discriminant hash codes. The first-layer hash function projects the image data into a transition space, and the second-layer hash function narrows the semantic gap between features and hash codes. Then, we propose an energy preserving strategy to retain the energy of the original data in the transition space, thereby alleviating the energy loss of hash projecting. Moreover, the semantic reconstruction mechanism is proposed to guarantee the semantic information can be well preserved into hash codes. Extensive experiments demonstrate the superior performance of the proposed REPH on five real-world image datasets. Our source code has been released at https://github.com/sunyuan-cs/REPH_main.

Index Terms—Similarity retrieval, energy preserving, learning to hash, transition space.

Manuscript received 25 March 2023; revised 13 September 2023 and 22 November 2023; accepted 2 January 2024. Date of publication 25 January 2024; date of current version 2 July 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U19A2078 and Grant 62372315; in part by the Sichuan Science and Technology Planning Project under Grant 2023YFG0033, Grant 2023ZHCG0016, Grant 2023YFQ0020, and Grant 2023ZYD0143; in part by the Chengdu Science and Technology Project under Grant 2023-XT00-00004-GX and Grant 2021-JB00-00025-GX; in part by the Base Strengthening Program of National Defense Science and Technology under Grant 2022-JCJQ-JJ-0292; in part by the Project of Key Laboratory of Film and Television (TV) Media Technology of Zhejiang Province under Grant 2020E10015; and in part by the Mianyang Science and Technology Program under Grant 2022ZYDF089. The Associate Editor for this article was H. Huang. (*Corresponding author: Qilin Li.*)

Yuan Sun is with the College of Computer Science, Sichuan University, Chengdu 610044, China (e-mail: sunyuan_work@163.com).

Jian Dai is with the Southwest Automation Research Institute, China South Industries Group Corporation, Mianyang 621000, China.

Zhenwen Ren is with the Department of National Defence Science and Technology, Southwest University of Science and Technology, Mianyang 621000, China.

Qilin Li is with State Grid Sichuan Electric Power Company, Chengdu 610045, China (e-mail: li_qi_lin@163.com).

Dezhong Peng is with the College of Computer Science, Sichuan University, Chengdu 610044, China, also with Chengdu Ruibei Yingte Information Technology Company Ltd., Chengdu 610054, China, and also with the National Innovation Center for UHD Video Technology, Chengdu 610095, China.

Digital Object Identifier 10.1109/TITS.2024.3351841

I. INTRODUCTION

WITH the rapid development of machine vision technology, the speed, accuracy, and intelligent characteristics of vision searching technology [1], [2], [3] have been widely used in all walks of life in modern industry. Visual search uses real-world visual images as the stimuli for searches [4], [5], [6], thereby effectively retrieving visual information needed by users. As shown in Fig. 1 (Note these pictures come from the Internet), some typical industrial application scenarios include machine target search, underwater visual search, street view search, and manipulator target recognition [7], [8]. Visual search can guide the navigation behavior of smart driving to capture the most relevant parts of the environment. Boston Dynamics adopts visual recognition to quickly and accurately find the parts under test, confirm the target position, and guide the robot manipulator to grasp accurately. Moreover, visual search in the e-commerce industry can suggest thematic or style-related items to shoppers. However, system data in industrial applications is often massive and the value density of available information is very low [9], [10], [11], [12]. It is difficult that how to extract semantic information quickly from large-scale data. Therefore, there is an urgent need for an algorithm to meet the needs of massive data retrieval and storage.

Due to significant advantages in terms of storage cost and query speed, image hashing has become a widely used industrial solution for fast similarity retrieval [13], [14], [15]. Image hashing aims to project data into Hamming space to preserve the similarity of the original data. Existing hashing methods are mainly divided into two categories: supervised hashing [16], [17], [18], [19] and unsupervised hashing [20], [21]. Unsupervised hashing methods do not use label information and utilize image data to learn hash functions. The research direction of traditional unsupervised hashing methods mainly focuses on hash function learning and hash code quantization problems. Supervised hashing methods apply the prior information of semantic label to obtain better performance than unsupervised methods.

Although existing supervised image hashing methods have obtained promising performance [22], they unconsciously ignore the information loss of hashing projecting, which leads to difficulty in preserving major energy in the original space. To tackle the above problems, we propose a novel **Relaxed Energy Preserving Hashing (REPH)** for large-scale image retrieval. As shown in Fig. 2, we first construct a

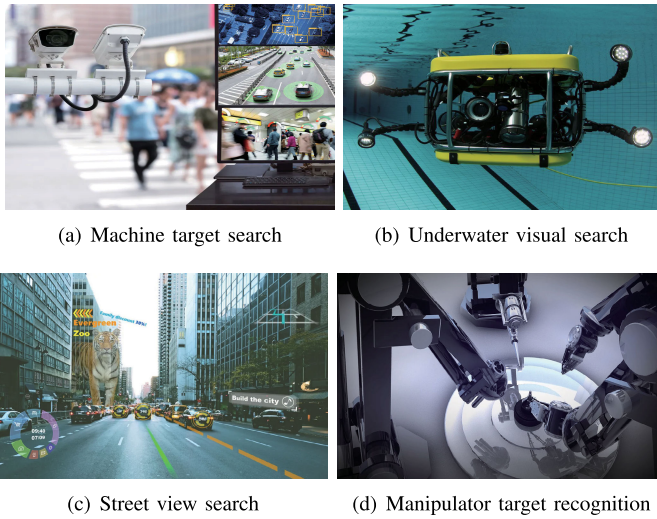


Fig. 1. Typical industrial application scenarios of image retrieval, such as machine target search, underwater visual search, street view search, and manipulator target recognition.

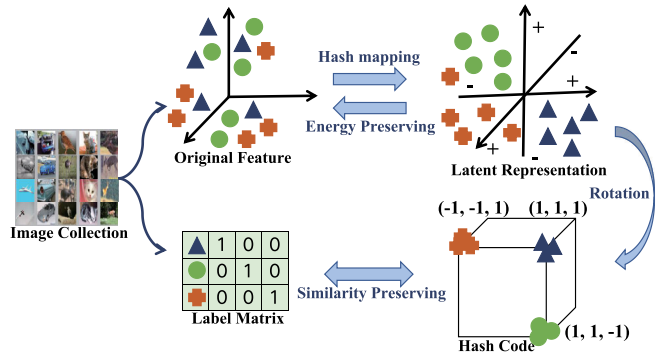


Fig. 2. Illustration of the proposed REPH. REPH adopts the energy preserving strategy to better extract the latent information, and then learns hash codes by rotating operation. Further, REPH utilizes similarity preserving to guide hash codes.

transition space to learn the latent representation from the high-dimensional features. Further, an energy preserving strategy is proposed that enforces latent representation to preserve the primary energy of the data in the obtained transition space. Moreover, we propose a semantic mapping and reconstruction mechanism to preserve semantic information. In summary, our main contributions have the following perspectives:

- We propose a novel image retrieval method, namely REPH. For the first time, REPH formulates a well-designed transition space to dexterously bridge the semantic connection between the kernel space and Hamming space, meanwhile preserving the major energy of the original data to reduce information loss of hashing projecting.
- To reduce the semantic loss of hash codes, we propose a semantic mapping and reconstruction mechanism that introduces an orthogonal matrix to build a latent bi-direction semantic connection between class labels and hash codes.

- Extensive experiments on five large-scale datasets are performed to demonstrate that our REPH outperforms several state-of-the-art methods.

The rest parts of this paper is organized as follows. Section III states the proposed method. Section IV shows the experimental results and analysis. Finally, we give the conclusion in section V.

II. RELATED WORK

In recent years, many promising hashing methods [23] have been explored, which project the high-dimensional data into the efficient binary representation while preserving the similarity of the original space. Existing hashing methods [24], [25] can be divided into two main types: unsupervised hashing and supervised hashing. Unsupervised hashing [26] does not exploit label information to learn hash codes, but rather leverages correlations between training data. Supervised hashing [27] makes full use of labels to guide the learning of hash codes. Generally speaking, supervised hashing can achieve better performance. Therefore, many satisfactory supervised hashing methods [28] are proposed. For example, fast supervised discrete hashing (FSDH) [29] transforms labels into hash codes to speed up the algorithm, and achieves large-scale image retrieval. Then, some researchers propose to use a similarity matrix to preserve semantic information. However, the large similarity matrix [30] with size $n \times n$ inevitably leads to high storage space, which is difficult to adapt to large-scale image retrieval tasks. Fast scalable supervised hashing (FSSH) [31] proposes a pre-computed intermediate term to avoid the use of such a similarity matrix. In addition, probability ordinal-preserving semantic hashing (POSH) [32] proposes a novel ordinal-preserving hashing concept to learn high-quality hash codes. And ordinal-preserving latent graph hashing (OLGH) [33] learns the latent representation to generate hash codes, while preserving the high-order local structure information. To enhance semantic discrimination ability, feature and semantic views consensus hashing (FSVCH) [34] proposes feature and semantic views consensus to further extract common hidden information. Since the one-step drop of the dimension can lead to severe information loss, hierarchical hashing learning (HHL) [35] adopts a layer-wise fashion to gradually extract the discriminative features.

To obtain high retrieval precision, the existing methods typically learn default long hash codes [36] (much larger than $\log_2 c$, such as 64 bits or 128 bits), which are usually employed in the high-performance device. It is obvious that flexibility and scalability are greatly limited. An important factor of hashing learning [37] is that the learned hash codes should be as compact and short as possible, which is helpful to deal with large-scale data. For a dataset with c classes, the length l of hash codes should be short and an integer slightly larger than $\log_2 c$. This means that short hash codes have faster training and query speed, which are more suitable for devices with limited computing resources [38]. Nevertheless, short hash codes often lead to disappointing performance due to severe information loss. How to learn high-quality short hash codes for scalable image retrieval is

a great challenge [39]. To meet the performance of short hash codes in practical applications, supervised short length hashing (SSLH) [40] proposes a mutual reconstruction strategy to reduce the loss of semantic information, and a robust estimator term to enhance the robustness of short hash codes. Reinforced short length hashing (RSLH) [41] unifies mutual regression, semantic pairwise similarity, and a relaxed strategy for learning high-quality short hash codes. However, because of discrete cyclic coordinate descent (DCC) optimization, these two short hashing methods are not flexible and scaleable for large-scale image retrieval.

In general, the existing methods often directly construct the connection between the features and hash codes. However, since the single-layer hash function has a huge burden between the features and hash codes, the optimal Hamming space could be difficult to search. It is difficult to achieve the complex mapping transformation from the original space to the optimal Hamming space. In addition, the existing methods largely ignore the main energy loss. Therefore, the inherent semantic structure can not perfectly preserved to be effectively extract more discriminative features.

III. PROPOSED METHODOLOGY

In this section, we first give some used notations. Then we elaborate on the details of the relaxed energy preserving hashing (REPH) and some analysis including complexity and convergence.

A. Notations

For the sake of presentation, we adopt the uppercase bold letters and the lowercase bold letters to denote the matrices and feature vectors (*i.e.*, \mathbf{V} and \mathbf{v}), respectively. Moreover, the i -th column feature vector represents as \mathbf{v}_i . $\|\mathbf{V}\|_F^2$ and \mathbf{V}^\top represent Frobenius norm and the transpose of the matrix \mathbf{V} , respectively. \mathbf{I} indicates the identity matrix with the corresponding dimensionality.

For image retrieval, we are typically given n image samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$, where d is the feature dimension of each image. For the supervised image retrieval problem, all images have a corresponding ground-truth class label matrix, *i.e.*, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in 0, 1^{c \times n}$, where c is the number of classes. The goal of hashing learning is to learn the hash function to project the high-dimensional image features into the more compact hash codes $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n] \in \{-1, 1\}^{l \times n}$; meanwhile, preserving the intrinsic similarities from the original feature space. Without loss of generality, we suppose all image instances to be zero centered, *i.e.*, $\sum_{i=1}^n \mathbf{x}_i = 0$. Then, we adopt the kernel trick (such as the radial basis function) to capture the local structure of the image data. Specifically, we randomly select k samples as anchor points from the training images, *i.e.*, $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$. The k -dimension kernel features can be represented as follows

$$\phi(\mathbf{v}_i) = [\exp(\frac{\|\mathbf{v}_i - \mathbf{a}_1\|_2^2}{-2\sigma^2}), \dots, \exp(\frac{\|\mathbf{v}_i - \mathbf{a}_k\|_2^2}{-2\sigma^2})]^\top \quad (1)$$

where σ is the kernel width and can be calculated by the average Euclidean distances between the samples. To make easy presentation, we use \mathbf{X} to represent $\phi(\mathbf{V})$.

B. Problem Formulation

Existing image hashing methods [6] often directly transform high-dimensional data into low-dimensional hash codes by a single-layer hash function. This sudden drop of dimension could cause severe semantic information loss. To overcome this problem, we mainly deal with it from three aspects. First, we propose the progressive learning strategy to perfectly preserve the intrinsic structure of the data. Then, we propose energy preserving to retain the main energy information of the original data to alleviate severe information loss during hashing mapping. Finally, to further preserve the semantic similarities in original space, we propose the semantic reconstruction scheme.

Specifically, since the single-layer hash function has a huge burden to compromise between the features and hash codes, it could result in large the discriminative information loss. Hence, the hashing mapping could fail to search an optimal Hamming space. It is difficult to achieve the complex mapping transformation from the original space to the optimal Hamming space. Therefore, such a mapping structure cannot perfectly preserve the inherent semantic structure to effectively extract more discriminative features. To overcome this issue, we propose a progressive learning strategy, which utilizes two relaxed hashing projection matrices to search an optimal Hamming space. Thereupon, the hash codes \mathbf{B} can be learned through the following formula

$$\min_{\mathbf{B}, \mathbf{R}, \mathbf{Q}} \|\mathbf{B} - \mathbf{R}\mathbf{Q}\mathbf{X}\|_F^2 \text{ s.t. } \mathbf{B} \in \{-1, 1\}^{l \times n}, \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \quad (2)$$

where $\mathbf{R} \in \mathbb{R}^{l \times k}$ and $\mathbf{Q} \in \mathbb{R}^{l \times l}$ are the two-level hash functions, which can endow \mathbf{Q} more freedom to extract the most discriminative features. The orthogonal constraint of \mathbf{R} can avoid trivial solutions and boost the discriminative power. The proposed progressive learning strategy can construct the transition space to bridge the semantic connection between the feature space and Hamming space. Therefore, the underlying structure of data can be further captured, thereby enhancing the representation capability of hash codes. In theory, we gradually shrink the solution set to a very small range by building the transition space. Hence, we can easily obtain an optimal solution.

To further reduce the information loss of feature mapping, we desire the transition space can preserve sufficient energy to recover original data. Specifically, we introduce a recovering matrix \mathbf{P} to achieve data reconstruction in the obtained transition space. The problem can be described as

$$\min_{\mathbf{P}, \mathbf{Q}} \|\mathbf{X} - \mathbf{P}\mathbf{Q}\mathbf{X}\|_F^2 \text{ s.t. } \mathbf{P}^\top \mathbf{P} = \mathbf{I} \quad (3)$$

where $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$ is the orthogonal constraint, which can prevent trivial solution. Note here that \mathbf{P} can be viewed as a reverse projection matrix. It can restore the aggregated discriminative features in the transition space to the original space and avoid energy loss and information loss. We hold the main energy in the original space instead of the transition space since the original space contains more semantic information.

However, this transition space lacks adequate flexibility to extract the intrinsic semantic knowledge. To this end, we hope

to inherit the intrinsic semantic information from original images. Typically, supervised discrete hashing (SDH) [42] proposes to project hash codes \mathbf{B} into label matrix \mathbf{Y} , thereby preserving label semantic information. It can be formulated as

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

To reduce time complexity and stabilize hashing learning, fast supervised discrete hashing (FSDH) [29] proposes to attempt to map label matrix \mathbf{Y} into hash codes \mathbf{B} as follows

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

To fully utilize intrinsic semantic similarity information between hash codes and class labels, we combine these two strategies to enhance coding ability. The problem can be written as follows:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{Y} - \mathbf{W}^\top \mathbf{B}\|_F^2 + \lambda \|\mathbf{B} - \mathbf{W}\mathbf{Y}\|_F^2 \\ s.t. \quad & \mathbf{B} \in \{-1, 1\}^{l \times n} \end{aligned} \quad (6)$$

where λ is the regularization parameter. Clearly, a new parameter λ is introduced, which is not user-friendly. This will cause us to spend more effort adjusting the most suitable parameters for a specific dataset. We thus impose the constraint, *i.e.*, $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$. Both $\|\mathbf{Y} - \mathbf{W}^\top \mathbf{B}\|_F^2$ and $\|\mathbf{B} - \mathbf{W}\mathbf{Y}\|_F^2$ can easily be satisfied. Such similarity preserving strategy can be viewed as a latent linear auto-encoder process. Therefore, the mapping and reconstruction mechanism builds a latent bi-direction semantic connection to preserve the semantic similarity. The problem can further be simplified as

$$\min_{\mathbf{W}} \|\mathbf{B} - \mathbf{W}\mathbf{Y}\|_F^2 \quad s.t. \quad \mathbf{W}^\top \mathbf{W} = \mathbf{I} \quad (7)$$

The overall optimization objective of REPH can be formulated as

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{B}, \mathbf{W}, \mathbf{R}, \mathbf{Q}} \quad & \|\mathbf{B} - \mathbf{R}\mathbf{Q}\mathbf{X}\|_F^2 + \alpha \|\mathbf{X} - \mathbf{P}\mathbf{Q}\mathbf{X}\|_F^2 \\ & + \beta \|\mathbf{B} - \mathbf{W}\mathbf{Y}\|_F^2 \\ s.t. \quad & \mathbf{B} \in \{-1, 1\}^{l \times n}, \mathbf{R}^\top \mathbf{R} = \mathbf{I} \\ & \mathbf{P}^\top \mathbf{P} = \mathbf{I}, \mathbf{W}^\top \mathbf{W} = \mathbf{I} \end{aligned} \quad (8)$$

In summary, our proposed method possesses the following properties:

- The first term uses two-level hash function instead of single hash function, which progressively projects the original feature space into the optimal Hamming space through two-level transformations. Hence, the solution set with a smaller range can be obtained.
- The second term is the energy preserving strategy that guarantees the main energy loss of the original space can be mitigated. It utilizes two energy transformation matrices to preserve the main discriminant information. Therefore, more flexible feature extraction can be achieved for the discriminative features.
- The third term is a latent linear auto-encoder process that builds the bi-direction semantic connection between label and hash codes to preserve the semantic similarity by the orthogonal matrix.

The three proposed components are all designed to solve the problem of information and semantic loss in the single-layer hash mapping process. The three components complement each other and work together to improve the quality of hash codes. Overall, we formalize the proposed progressive learning, energy preserving, and semantic reconstruction into a unified hashing framework, which coordinates with each other to mitigate information and semantic loss, thereby boosting retrieval performance.

C. Optimization

In this section, we give detailed optimization steps to solve the proposed REPH method. Specifically, we solve all variables of problem (23) by an alternating strategy. That is, updating one variable when fixing the remaining variables, and repeating this process several times until convergence.

Q-Step: We fix other variables and drop the irrelevant terms. The simplified problem can be obtained:

$$\min_{\mathbf{Q}} \|\mathbf{B} - \mathbf{R}\mathbf{Q}\mathbf{X}\|_F^2 + \alpha \|\mathbf{X} - \mathbf{P}\mathbf{Q}\mathbf{X}\|_F^2 \quad (9)$$

Setting the derivative with respect to \mathbf{Q} to zero, and we can obtain

$$\mathbf{R}^\top \mathbf{R} \mathbf{Q} \mathbf{X} \mathbf{X}^\top + \alpha \mathbf{P}^\top \mathbf{P} \mathbf{Q} \mathbf{X} \mathbf{X}^\top = \mathbf{R}^\top \mathbf{B} \mathbf{X}^\top + \alpha \mathbf{P}^\top \mathbf{X} \mathbf{X}^\top \quad (10)$$

Because of $\mathbf{R}^\top \mathbf{R} = \mathbf{I}$ and $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$, the result of this problem can be obtained as follows

$$\mathbf{Q} = (\mathbf{R}^\top \mathbf{B} \mathbf{X}^\top + \alpha \mathbf{P}^\top \mathbf{X} \mathbf{X}^\top) ((1 + \alpha) \mathbf{X} \mathbf{X}^\top)^{-1} \quad (11)$$

P-Step: Fixing other variables, we update \mathbf{P} by solving the following subproblem

$$\min_{\mathbf{P}} \alpha \|\mathbf{X} - \mathbf{P}\mathbf{Q}\mathbf{X}\|_F^2 \quad s.t. \quad \mathbf{P}^\top \mathbf{P} = \mathbf{I} \quad (12)$$

It is obviously that problem (12) is an orthogonal procrustes problem (OPP) [43], whose closed form solution can be calculated by SVD, *i.e.*,

$$\mathbf{U} \mathbf{\Lambda} \mathbf{V}^\top = \mathbf{X} \mathbf{X}^\top \mathbf{Q}^\top \quad (13)$$

where \mathbf{U} and \mathbf{V} are the left-singular and right-singular vectors, respectively. The solution thus is obtained by

$$\mathbf{P}^* = \mathbf{U} \mathbf{V}^\top \quad (14)$$

R-Step: Fixing other variables, we update \mathbf{R} by solving the following subproblem

$$\min_{\mathbf{R}} \|\mathbf{B} - \mathbf{R}\mathbf{Q}\mathbf{X}\|_F^2 \quad s.t. \quad \mathbf{R}^\top \mathbf{R} = \mathbf{I} \quad (15)$$

Similar to problem (12), \mathbf{R} can be solved by

$$\mathbf{U} \mathbf{\Lambda} \mathbf{V}^\top = \mathbf{B} \mathbf{X}^\top \mathbf{Q}^\top \quad (16)$$

W-Step: Fixing other variables, we update \mathbf{W} by solving the following subproblem

$$\min_{\mathbf{W}} \beta \|\mathbf{B} - \mathbf{W}\mathbf{Y}\|_F^2 \quad s.t. \quad \mathbf{W}^\top \mathbf{W} = \mathbf{I} \quad (17)$$

Similar to problem (12), the optimal solution for W can be derived by

$$U \Lambda V^T = BY^T \quad (18)$$

B-Step: We can obtain the following optimization problem:

$$\begin{aligned} \min_B \quad & \|B - RQX\|_F^2 + \beta \|B - WY\|_F^2 \\ \text{s.t.} \quad & B \in \{-1, 1\}^{l \times n} \end{aligned} \quad (19)$$

With a simple manipulation, we can further obtain the trace norm maximization problem

$$\begin{aligned} \max_B \quad & B^T(RQX + \beta WY) \\ \text{s.t.} \quad & B \in \{-1, 1\}^{l \times n} \end{aligned} \quad (20)$$

Therefore, the close-form solution of hash codes can be obtained:

$$B = \text{sign}(RQX + \beta WY) \quad (21)$$

With the above analysis, our proposed algorithm iteratively solves each variable until the objective value satisfies $\|B^{t+1} - B^t\|_F^2 / \|B^t\|_F^2 \leq \epsilon$ or reaches the maximum iteration number set in advance. The optimization scheme of our proposed REPH is summarized in Algorithm 1.

Algorithm 1 Energy Preserving Short Hashing for Image Retrieval

Require: Training set V with label matrix Y , query set X^q , hash length l , and the number of anchors k

- 1: Parameter: α and β ;
- 2: Initialize: W , Q , P , R , and B ;
- 3: Obtain kernel features X by kernel trick;
- 4: **repeat**
- 5: Update Q via (11);
- 6: Update P via (13);
- 7: Update R via (16);
- 8: Update W via (18);
- 9: Update B via (21);
- 10: **until** Satisfy the stop criteria;
- 11: Learn hash codes H of query set via (22);

Ensure: Perform image retrieval.

D. Out-of-Sample Extension

According to our learning algorithm, the optimal hash codes B for the training set X are easily obtained. When facing the newcome samples X^q , we need to further obtain a set of hash codes for out-of-sample images by the hashing projection, i.e.,

$$H = \text{sign}(RQX^q) \quad (22)$$

Afterward, we can calculate the similarity between hash codes by XOR operation to achieve the fast similarity search [44].

E. Computational Complexity Analysis

The computational complexity of Algorithm 1 mainly lies in five subproblems. The main time cost of these steps is computing the inverse of the square matrix and performing SVD. More precisely, for the inverse operation of an $k \times k$ square matrix, its computational complexity is $\mathcal{O}(k^3)$; for the SVD operation of an $k \times l$ matrix, its computational complexity is $\mathcal{O}(l^3)$. Generally, for updating Q , its computational cost is mainly the matrix inverse transformation, which is about $\mathcal{O}(k^3)$. For updating P , R , and W , their computational complexities of the SVD computation are $\mathcal{O}(l^3)$, $\mathcal{O}(k^3)$ and $\mathcal{O}(c^3)$, respectively. For updating B , its computational complexity is $\mathcal{O}(nl^2)$. For the REPH method, the overall computational complexity of each iteration is about $\mathcal{O}(2k^3 + l^3 + c^3 + nl^2)$. In practice, $k \ll n$ and the values of c and l are small. Moreover, the proposed REPH has fast convergence property, which means the number of iteration t is small. Hence, the total computational complexity is close to $\mathcal{O}(n)$.

F. Convergence Analysis

To better perform convergence analysis, we denote $\mathcal{L}(Q, P, R, W, B)$ as the objective function (23). It is obvious that each subproblem always monotonically decreases in each iteration and has a closed-form solution. Therefore, we can obtain the following formula

$$\begin{aligned} & \mathcal{L}(Q^{t+1}, P^{t+1}, R^{t+1}, W^{t+1}, B^{t+1}) \\ & \leq \mathcal{L}(Q^{t+1}, P^{t+1}, R^{t+1}, W^{t+1}, B^t) \\ & \leq \mathcal{L}(Q^{t+1}, P^{t+1}, R^{t+1}, W^t, B^t) \\ & \dots \\ & \leq \mathcal{L}(Q^t, P^t, R^t, W^t, B^t) \end{aligned} \quad (23)$$

In general, the value of the objective function is monotonously decreasing. And it has a local optimal point due to the simple summation of these F -norms. According to the bounded monotone convergence rule, our REPH method is convergent. Further, we also perform the convergence experiments in the next section.

G. Connections to Other Methods

Although our REPH method also uses regression mapping to learn hash codes, we are fundamentally different from these existing methods, including SDH [42] and FSDH [29]. Essentially, we have two differences: 1) Different from the single-level hash function, we gradually project the original feature space into the optimal Hamming space through a two-level hash transformation. 2) To mitigate the main energy and information loss of the original space, an energy preserving strategy is proposed.

DPH [45], PGH [46], and the proposed REPH utilize progressive learning strategy for image retrieval. However, they have obvious differences. In DPH, hash codes are learned by progressively expanding salient regions. In PGH, hash codes are fed into GANs to progressively excavate the semantic relations. By contrast, our REPH progressively projects the original feature space into the optimal Hamming space

through two-level transformations. Besides, TBH [47] is a unsupervised hashing framework with an auto-encoding twin-bottleneck scheme. Specifically, the binary and continuous bottlenecks collaboratively generate discriminative hash codes. Then, according to the similarities by decoding, TBH adopts the auto-encoding scheme to determine the reward of the encoding quality on the code-driven graph. Different from the auto-encoder scheme, the energy preserving strategy adopts a recovering matrix to retain the energy of the original data in the transition space, thereby alleviating the energy loss of hash projecting during progressive learning.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we perform various experiments with eight image hashing methods on five large-scale datasets. All comparison methods are conducted on a desktop Windows PC with an Intel Core i9 CPU and 64GB RAM.

A. Datasets

Industrial applications often generate large amounts of industrial image data. To evaluate the performance of the proposed REPH, we perform experiments on five real-world large-scale image datasets, which cover image target, hand-written digital, industrial scenes, video surveillance. General statistics of five datasets are given in Table I.

CIFAR-10 [48] consists of 60k images from 10 classes and each class has 6k images. We represent each image using a 512-dimension GIST feature vector extracted from the original color image of size 32×32 .

MNIST [49] consists of 70k handwritten digital images from 10 classes (*i.e.*, from 0 to 9). In our experiments, we crop and normalize each image to 28×28 , and then use a 784-dimensional feature vector to represent each image.

NUS-WIDE [50] is collected from the real-world large-scale multi-label image dataset Flickr. We select the 21 most frequent tags containing 195834 images to evaluate the effect. In this multi-label dataset, two images are semantically similar if they share at least one common label, and vice versa. In our experiments, we used a 500-dimensional bag-of-words feature vector to represent each image.

Caltech-256 [33] contains 29780 samples from 256 classes. Each category has at least 80 images. We use CNN pre-trained on ImageNet to extract the 1024-dimensional deep features.

Place205 [51] has 2.5M images from 205 classes used for scene recognition. We extract CNN features pre-trained on ImageNet and obtain 128-dimensional by PCA.

For CIFAR-10, MNIST, and Caltech-256, we randomly select 1k images as the query set and the remains as the training set. For NUS-WIDE, we randomly select 2.1k images as the query set and the remaining images as the training set. For Place205, we randomly select 100k images as the training set and 4.1k images as the query set.

B. Experimental Setting

To demonstrate the effectiveness, we compare REPH with some advanced hashing methods, which are SDH [42],

TABLE I
GENERAL STATISTICS OF FIVE DATASETS

Dataset	Training size	Query size	Classes	Dimension
CIFAR-10	59000	1000	10	512
MNIST	69000	1000	10	784
NUS-WIDE	193734	2100	21	500
Caltech-256	28780	1000	256	1024
Place205	100000	4100	205	128

FSDH [29], FSSH [31], SSLH [40], RSLH [41], SCDH [44], POPSH [32], OLGH [33], and SASH [52]. Due to the high computation complexity of SSLH and RSLH, we randomly choose 5000 images on the CIFAR-10, MNIST, and Caltech-256 datasets as the training set, and 10500 images on the NUS-WIDE and Place205 dataset as one. We use the published code of the original authors for all comparison methods and adopt the suggested parameters from their published papers. In our method, empirically, we set the number of anchors $k = 1500$. For two main hyper-parameters α and β , we perform parameter sensitivity analysis and set $\{10^{-4}, 10^{-3}\}$ for the first three datasets, $\{10^{-3}, 10^{-4}\}$ for the Caltech-256 dataset, $\{10^{-3}, 10^{-3}\}$ for the Place205 dataset, respectively. The existing image hashing methods often report the performance within 128 bits of hash codes. To better explore the effect of hash length on retrieval performance, we analyse the performance of hash codes with 8-1024 bits.

C. Evaluation Criteria

In order to demonstrate the effectiveness of our proposed method, we use several classic evaluation criteria, including mean average precision (mAP), top-50 precision (Precision@50), NDCG@100, and precision-recall curve. The mAP is the mean of the average precision scores for all query samples. Precision@50 represents mAP scores for the top 50 samples. NDCG@100 represents normalized discounted cumulative gain at rank 100. The precision-recall curve can be used for evaluating the precision and recall.

D. mAP Results

The mAP performance results of all compared methods are summarized on these five datasets in table II. From the experimental results, the proposed REPH obvious outperforms existing hashing methods for almost all hash lengths. The results show that the proposed relaxed energy preserving strategy can reduce semantic information loss and enhance the quality of hash codes, thereby improving the overall retrieval performance on all datasets. For example, when hash length is 64 bits, the retrieval performance of our method improves 1.05%, 0.8%, 2.02%, 3.1%, and 1.05%, respectively. Moreover, as the bit length increases, the performance of our method gradually improves. The performance of our REPH stabilizes when the hash length is greater than or equal to 256 bits. It is worth noting that some comparison methods show performance degradation when the hash length

TABLE II
THE MAP RESULTS (%) ON THE FIVE DATASETS. THE 1ST/2ND BEST VALUES ARE SHOWN IN BOLD/UNDERLINED

Dataset	Method	8 bits	16 bits	32 bits	64 bits	128 bits	256 bits	512 bits	1024 bits
CIFAR-10	SDH [42]	32.90	39.50	43.82	46.47	47.62	48.28	48.55	/
	FSDH [29]	34.15	38.02	42.75	45.47	47.20	48.52	48.25	48.65
	FSSH [31]	56.89	61.78	65.38	68.25	69.06	70.67	71.00	70.68
	SSLH [40]	26.68	33.60	37.86	39.80	37.79	40.25	39.04	/
	RSLH [41]	21.34	28.81	34.06	34.86	36.22	37.32	37.71	37.76
	SCDH [44]	64.52	<u>67.75</u>	<u>70.43</u>	<u>71.63</u>	<u>72.14</u>	<u>71.89</u>	<u>72.48</u>	<u>72.73</u>
	POPSH [32]	60.75	64.33	67.94	69.14	70.15	71.28	71.60	71.30
	OLGH [33]	61.27	66.40	68.31	64.22	49.83	38.65	31.40	/
	SASH [52]	19.97	21.84	21.85	26.65	25.61	17.21	15.13	/
	Our REPH	<u>62.81</u>	70.33	72.66	72.68	73.89	74.06	74.29	74.85
MNIST	SDH [42]	60.58	69.02	68.60	68.54	68.86	69.12	/	/
	FSDH [29]	89.71	90.89	93.70	94.39	94.64	94.88	94.77	94.98
	FSSH [31]	92.83	94.14	94.96	95.29	95.40	95.63	95.80	95.90
	SSLH [40]	71.29	81.71	91.68	92.21	92.80	92.52	92.76	/
	RSLH [41]	80.01	89.31	90.41	92.31	92.78	93.19	93.27	93.71
	SCDH [44]	91.57	91.42	91.68	91.30	91.29	91.77	91.80	91.97
	POPSH [32]	<u>92.88</u>	<u>95.02</u>	<u>96.03</u>	<u>96.26</u>	<u>96.31</u>	<u>96.76</u>	<u>96.62</u>	<u>96.66</u>
	OLGH [33]	91.29	95.00	95.85	95.82	94.77	89.53	77.70	/
	SASH [52]	75.76	74.78	86.22	90.21	87.62	70.10	/	/
	Our REPH	95.17	96.32	96.49	97.07	97.06	97.17	97.07	97.07
NUS-WIDE	SDH [42]	53.18	53.18	53.18	53.18	53.18	53.18	/	/
	FSDH [29]	31.44	31.44	31.44	31.44	31.45	31.44	31.44	31.44
	FSSH [31]	54.36	55.37	<u>60.86</u>	61.08	<u>64.04</u>	<u>64.13</u>	<u>64.37</u>	<u>64.52</u>
	SSLH [40]	38.84	41.93	42.23	43.32	43.84	42.78	40.92	/
	RSLH [41]	38.26	41.23	42.49	41.74	43.35	43.73	43.72	73.71
	SCDH [44]	46.38	46.64	47.04	47.75	47.71	47.99	48.07	48.13
	POPSH [32]	<u>54.95</u>	<u>57.46</u>	60.80	<u>61.14</u>	62.38	62.47	63.00	62.88
	OLGH [33]	49.47	53.64	56.80	55.67	53.80	48.69	44.24	/
	SASH [52]	34.61	34.93	35.93	39.28	/	/	/	/
	Our REPH	57.63	60.07	61.89	63.16	64.15	64.23	64.62	64.88
Caltech-256	SDH [42]	12.64	24.86	36.00	44.98	51.24	54.96	57.30	/
	FSDH [29]	14.88	29.44	39.78	47.09	53.44	56.74	58.95	59.80
	FSSH [31]	22.85	41.14	52.73	62.50	68.41	70.49	71.45	72.39
	SSLH [40]	7.25	9.30	9.73	12.90	38.98	39.77	46.23	/
	RSLH [41]	10.94	20.89	30.25	38.32	45.03	49.85	52.20	53.70
	SCDH [44]	25.92	43.54	55.65	63.55	67.91	69.69	70.05	70.82
	POPSH [32]	27.86	<u>51.86</u>	<u>62.25</u>	<u>68.80</u>	<u>74.48</u>	<u>75.80</u>	<u>76.80</u>	<u>77.11</u>
	OLGH [33]	25.69	47.77	60.12	62.00	56.82	51.54	46.59	/
	SASH [52]	12.87	20.89	27.81	33.79	36.47	32.73	35.53	/
	Our REPH	<u>26.94</u>	52.07	65.32	71.90	76.22	77.26	78.34	78.55
Place205	SDH [42]	5.21	12.34	18.72	23.50	26.58	28.27	/	/
	FSDH [29]	5.53	13.14	18.72	23.45	26.57	28.08	29.14	29.37
	FSSH [31]	16.08	29.65	40.47	46.7	51.50	53.20	54.43	54.81
	SSLH [40]	2.22	1.51	2.52	4.21	15.13	17.49	21.19	/
	RSLH [41]	2.80	6.89	11.28	15.42	19.26	21.44	22.83	23.49
	SCDH [44]	<u>18.91</u>	27.74	27.84	28.04	28.06	28.14	28.12	28.05
	POPSH [32]	21.35	<u>36.55</u>	<u>46.64</u>	<u>51.35</u>	<u>53.34</u>	<u>54.73</u>	<u>55.54</u>	<u>55.95</u>
	OLGH [33]	19.74	34.41	43.57	37.41	27.06	20.44	16.47	/
	SASH [52]	4.77	6.17	10.17	13.05	11.24	11.50	/	/
	Our REPH	17.08	36.01	46.70	52.40	54.38	55.98	56.52	56.98

/ denotes that the running time are too high.

is greater than 128 bits. For large-scale data sets (NUS-WIDE Place205), retrieval performance is limited and short hash and Place205) or large classes of data sets (Caltech256 and codes cannot be used to advantage. Besides, we observe that

TABLE III

THE PRECISION@50 RESULTS (%) ON THE CIFAR-10 DATASET. THE 1ST/2ND BEST VALUES ARE SHOWN IN BOLD/UNDERLINED

Dataset	Method	8 bits	16 bits	32 bits	64 bits	128 bits	256 bits	512 bits	1024 bits
CIFAR-10	SDH [42]	43.20	50.52	53.26	54.49	53.39	55.43	55.92	/
	FSDH [29]	42.38	48.27	51.83	53.67	54.83	55.45	55.30	55.71
	FSSH [31]	45.03	55.37	60.55	62.13	60.79	63.72	63.81	63.73
	SSLH [40]	26.68	44.49	47.97	49.48	50.76	53.78	55.01	/
	RSLH [41]	30.64	39.88	44.93	46.47	47.58	48.44	48.45	48.59
	SCDH [44]	60.08	<u>61.86</u>	<u>63.70</u>	<u>65.58</u>	<u>65.53</u>	65.15	65.85	<u>66.15</u>
	POPSH [32]	55.66	59.71	63.29	63.77	64.63	<u>65.68</u>	<u>65.87</u>	65.65
	OLGH [33]	56.15	60.44	63.56	62.11	53.40	44.07	37.56	/
	SASH [52]	22.59	28.49	30.30	39.23	39.09	41.19	42.86	/
	Our REPH	<u>57.14</u>	64.85	66.04	66.02	67.06	67.12	67.27	68.04
MNIST	SDH [42]	66.48	72.54	72.34	72.81	74.07	76.37	/	/
	FSDH [29]	91.18	92.32	93.59	94.42	94.40	94.27	93.90	94.71
	FSSH [31]	90.23	93.27	93.12	93.87	94.60	94.52	94.60	94.70
	SSLH [40]	76.54	83.31	92.66	93.41	93.95	94.56	94.68	/
	RSLH [41]	86.86	91.89	92.46	93.66	93.67	93.94	93.80	94.11
	SCDH [44]	90.28	90.28	90.18	90.00	89.96	90.15	90.10	90.30
	POPSH [32]	<u>92.72</u>	<u>94.21</u>	<u>95.04</u>	<u>95.36</u>	<u>95.22</u>	<u>95.68</u>	<u>95.55</u>	<u>95.53</u>
	OLGH [33]	90.41	94.19	94.92	95.07	95.02	93.27	87.99	/
	SASH [52]	81.04	79.27	89.44	91.95	92.52	93.14	/	/
	Our REPH	94.51	95.59	95.49	96.24	96.27	96.30	96.10	96.10
NUS-WIDE	SDH [42]	36.34	36.35	36.35	36.35	36.35	36.35	/	/
	FSDH [29]	36.35	36.35	36.35	36.35	36.18	36.35	36.35	35.71
	FSSH [31]	51.89	<u>63.87</u>	<u>69.52</u>	<u>69.58</u>	<u>70.88</u>	<u>70.78</u>	<u>71.60</u>	<u>72.63</u>
	SSLH [40]	44.74	47.76	42.23	43.32	43.84	49.65	47.70	/
	RSLH [41]	43.17	46.42	48.47	49.51	50.67	51.63	52.42	52.47
	SCDH [44]	42.02	42.14	41.34	41.12	40.93	40.97	41.03	41.07
	POPSH [32]	<u>55.72</u>	61.81	64.55	65.12	66.00	65.48	66.24	66.26
	OLGH [33]	41.83	47.54	52.47	50.16	52.53	52.60	51.07	/
	SASH [52]	37.21	39.65	41.90	45.67	/	/	/	/
	Our REPH	64.03	63.97	70.58	70.99	71.48	70.95	71.78	73.14
Caltech-256	SDH [42]	15.64	34.87	48.69	58.52	63.90	66.90	68.38	/
	FSDH [29]	18.42	39.69	53.44	61.4	65.04	67.71	69.50	69.84
	FSSH [31]	18.86	36.65	52.21	60.28	63.92	66.34	67.03	67.95
	SSLH [40]	7.87	10.56	11.33	15.69	53.68	56.19	61.18	/
	RSLH [41]	10.94	30.69	43.44	53.53	59.76	63.74	65.33	66.29
	SCDH [44]	24.54	41.59	53.44	60.96	64.09	65.02	65.48	66.05
	POPSH [32]	26.24	<u>49.80</u>	<u>59.91</u>	<u>66.70</u>	<u>71.08</u>	<u>72.09</u>	<u>72.85</u>	<u>73.03</u>
	OLGH [33]	24.09	45.80	58.58	62.87	62.90	61.24	59.36	/
	SASH [52]	15.95	27.62	36.29	44.59	53.53	55.34	57.80	/
	Our REPH	<u>25.51</u>	50.26	63.46	69.82	72.86	73.40	74.87	74.69
Place205	SDH [42]	8.47	23.81	32.91	36.94	39.09	40.28	/	/
	FSDH [29]	9.14	25.13	32.9	37.01	39.22	39.89	40.41	40.58
	FSSH [31]	14.37	27.8	37.49	42.67	46.27	46.91	47.48	47.42
	SSLH [40]	3.28	2.07	4.44	9.39	31.11	33.73	36.23	/
	RSLH [41]	4.94	15.77	25.55	31.46	35.48	36.84	37.63	38.17
	SCDH [44]	<u>18.18</u>	26.82	27.02	27.05	27.02	27.09	27.05	27.05
	POPSH [32]	19.50	<u>33.66</u>	<u>42.52</u>	<u>45.86</u>	<u>46.99</u>	<u>47.45</u>	<u>48.15</u>	<u>48.33</u>
	OLGH [33]	17.72	31.89	40.64	39.51	36.42	33.57	31.59	/
	SASH [52]	7.63	10.50	18.06	24.27	30.22	31.27	/	/
	Our REPH	14.96	33.77	43.77	48.22	48.53	49.55	49.52	49.79

/ denotes that the running time are too high.

the results of our REPH with 8 bits are not the best on some datasets. The shorter the length of hash codes, the less image feature information it can represent, which means more information loss. Especially, in some datasets with large categories

TABLE IV
THE NDCG@100 RESULTS (%) ON THE CIFAR-10 DATASET. THE 1ST/2ND BEST VALUES ARE SHOWN IN BOLD/UNDERLINED

Dataset	Method	8 bits	16 bits	32 bits	64 bits	128 bits	256 bits	512 bits	1024 bits
CIFAR-10	SDH [42]	43.59	50.47	53.10	54.60	55.47	55.68	56.18	/
	FSDH [29]	43.20	47.65	52.21	53.45	54.74	55.82	55.54	55.77
	FSSH [31]	50.70	54.99	58.71	61.44	62.15	63.58	63.73	63.74
	SSLH [40]	33.62	44.31	47.97	49.76	50.86	53.34	54.15	/
	RSLH [41]	26.99	39.85	44.79	46.63	47.90	48.68	48.68	48.59
	SCDH [44]	60.23	<u>61.92</u>	63.69	<u>65.58</u>	<u>65.51</u>	65.15	65.85	<u>66.14</u>
	POPSH [32]	55.95	60.61	63.29	63.86	65.02	<u>65.68</u>	<u>65.98</u>	66.01
	OLGH [33]	56.12	60.50	<u>63.80</u>	63.16	59.10	53.71	48.72	/
	SASH [52]	22.94	27.97	39.39	38.23	37.92	37.24	38.29	/
	Our REPH	<u>57.19</u>	64.75	66.05	66.01	67.04	67.12	67.25	68.04
MNIST	SDH [42]	67.04	73.44	74.14	73.69	75.04	76.24	/	/
	FSDH [29]	90.79	92.33	94.12	94.21	94.40	94.43	94.27	94.72
	FSSH [31]	91.83	93.07	93.97	94.15	94.25	94.5	94.6	94.70
	SSLH [40]	77.48	83.80	92.85	93.31	93.89	94.18	94.27	/
	RSLH [41]	86.51	91.84	92.34	93.71	93.77	93.77	93.91	94.22
	SCDH [44]	90.30	90.28	90.18	90.00	89.96	90.14	90.10	90.30
	POPSH [32]	<u>93.24</u>	<u>94.31</u>	<u>95.08</u>	<u>95.33</u>	95.23	<u>95.67</u>	<u>95.55</u>	<u>95.57</u>
	OLGH [33]	90.27	94.21	94.94	95.23	<u>95.44</u>	95.20	94.70	/
	SASH [52]	81.28	78.99	89.52	91.90	90.03	91.54	/	/
	Our REPH	94.52	95.61	95.47	96.25	96.28	96.30	96.10	96.10
NUS-WIDE	SDH [42]	27.73	27.73	27.73	27.73	27.73	36.35	/	/
	FSDH [29]	27.73	27.75	27.73	27.73	27.73	27.73	27.73	27.61
	FSSH [31]	42.64	46.49	52.38	52.94	55.28	54.00	54.30	54.48
	SSLH [40]	34.44	36.51	37.14	38.29	39.34	39.50	38.60	/
	RSLH [41]	32.51	35.22	36.86	37.87	38.81	39.45	39.94	39.95
	SCDH [44]	33.07	33.97	33.48	33.49	33.19	33.41	33.48	33.48
	POPSH [32]	<u>42.37</u>	45.81	<u>48.02</u>	<u>50.19</u>	<u>52.58</u>	<u>52.33</u>	<u>53.31</u>	<u>52.43</u>
	OLGH [33]	35.30	38.60	35.18	36.72	41.26	41.70	41.46	/
	SASH [52]	28.46	30.16	32.07	34.57	/	/	/	/
	Our REPH	46.10	<u>45.61</u>	50.06	51.44	52.41	52.13	53.48	53.91
Caltech-256	SDH [42]	15.62	30.47	42.23	51.23	56.77	59.54	61.15	/
	FSDH [29]	18.83	35.45	46.27	53.04	58.49	60.88	62.39	62.81
	FSSH [31]	21.73	39.24	50.80	59.63	65.38	66.54	67.24	68.10
	SSLH [40]	7.87	10.59	11.15	15.03	45.94	47.30	52.98	/
	RSLH [41]	14.20	26.34	36.76	45.42	51.78	55.96	57.49	58.60
	SCDH [44]	24.62	41.69	53.55	61.13	64.28	65.21	65.63	66.25
	POPSH [32]	26.67	<u>50.06</u>	<u>60.04</u>	<u>66.89</u>	<u>71.20</u>	<u>72.29</u>	<u>73.03</u>	<u>73.26</u>
	OLGH [33]	24.53	45.99	58.21	61.22	58.83	55.24	51.90	/
	SASH [52]	15.42	25.16	32.37	39.33	44.72	44.90	47.20	/
	Our REPH	<u>26.03</u>	50.08	63.23	69.55	72.59	73.00	74.51	74.40
Place205	SDH [42]	8.75	22.96	32.14	36.48	38.48	39.49	/	/
	FSDH [29]	9.45	24.44	32.27	36.59	38.76	39.27	39.82	39.91
	FSSH [31]	14.30	27.46	37.50	42.67	46.29	46.90	47.49	47.42
	SSLH [40]	3.35	2.11	4.34	8.97	29.42	32.06	34.43	/
	RSLH [41]	5.03	14.93	23.84	29.85	33.82	35.17	36.00	36.39
	SCDH [44]	18.10	26.82	27.02	27.06	27.02	27.09	27.05	28.05
	POPSH [32]	19.68	<u>33.68</u>	<u>42.52</u>	<u>45.86</u>	<u>46.96</u>	<u>47.46</u>	<u>48.12</u>	<u>48.33</u>
	OLGH [33]	17.82	31.87	40.65	39.03	35.01	31.49	29.07	/
	SASH [52]	7.46	10.10	16.92	22.35	27.18	28.19	/	/
	Our REPH	14.87	33.74	43.76	48.23	48.54	49.54	49.52	49.79

/ denotes that the running time are too high.

(such as Caltech256 and Place205), the differences between images from different categories can be very subtle. In this case, shorter hash codes may not provide enough discrimination, resulting in lower retrieval performance. In general, the

TABLE V
THE TRAINING TIME (SECONDS) ON NUS-WIDE

Methods	8 bits	16 bits	32 bits	64 bits	128 bits
SDH [42]	16.63	22.68	50.03	143.26	686.49
FSDH [29]	16.09	24.53	24.72	27.78	34.98
FSSH [31]	11.10	12.83	13.10	13.26	19.52
SSLH [40]	15.42	32.81	91.36	278.94	817.28
RSLH [41]	103.14	114.11	159.84	195.02	260.24
SCDH [44]	48.01	56.98	59.48	172.54	831.81
POPSH [32]	11.01	12.03	12.91	17.14	20.52
OLGH [33]	10.19	14.88	14.56	14.31	15.62
SASH [52]	2678.89	6574.30	8399.81	11701.05	/
REPH	17.39	18.98	19.85	21.65	27.18

/ denotes that the running time are too high.

results demonstrates the effectiveness of the relaxed energy preserving hashing framework.

E. Precision@50 Results

The Precision@50 results of all compared methods are summarized in table III. From the experimental results, we can obtain the following observations: The proposed REPH achieves satisfactory Precision@50 results on these five datasets for different hash codes. And REPH obtain the more retrieval performance with hash length varying from 64 bits to 256 bits. Specifically, compared to competitors, for 64 bits, we obtain absolute boosts of 0.44%, 0.88%, 1.41%, 3.15%, and 2.36%, respectively. This superior performance is due to the fact that the proposed REPH can preserve the main energy to mitigate information loss. As the hash length increases, the Precision@50 scores of all methods gradually increases until it becomes stable. We can find that the retrieval performance will be worse when the data size is larger and the number of categories is more. However, our REPH can achieve a delightful retrieval performance.

F. NDCG@100 Results

The NDCG@100 results on the five datasets are summarized in table IV. From these tables, we can clearly see that our proposed REPH obtains the best scores with different hash lengths. The best results obtained by our method mainly benefit from energy preserving, which can mitigate of information loss. According to these tables, The NUS-WIDE, Caltech-256, and Place205 datasets are quite challenging because of large scale and large categories. For the Caltech-256 dataset, its features is deep features extracted by CNN pre-trained on ImageNet. Hence, the performance on the Caltech-256 dataset is significantly better than the performance of the Place205 dataset.

G. Precision-Recall

Precision-Recall curves on five datasets with different hash lengths (32 and 64 bits) are plotted in Fig.3. From these figures, we can obviously observe that the retrieval performance of the proposed REPH over the best competitor is significant.

It shows that our REPH can generate more discriminative hash codes to enhance retrieval performance. In general, the effectiveness of REPH is consistent with the mAP and Precision@50 scores and is further demonstrated.

H. Training Time

The time cost of existing image hashing methods mainly depends on the training step. Hence, we compare the training efficiency of these methods in the large-scale NUS-WIDE dataset when using different hash lengths in Table V. For several hashing methods (such as SSLH and RSLH), they need more training time due to learning hash codes bit by bit, and greatly improves dramatically as the length of bits increases. Since the similarity matrix consumes more training time to make similarity approximation, SCDH costs more training time. According to the experimental results, we can see that the training time grows, as the hash length increases. REPH can obtain acceptable training time and is scalable for large-scale data.

I. Convergence Experiments

As mentioned before, the alternating updates of different variables make the objective function value monotonically decreasing. To further investigate the convergence property, we conduct an empirical study for 8 bits hash length. As shown in Fig.4, five convergence curves are monotonically decreased close to a stable point within five iterations.

J. Parameter Sensitivity Analysis

We further use a grid search to perform parameter sensitivity analysis on these five datasets, such that the retrieval performance of our proposed REPH can be showed. The two trade-off parameters are α and β . The mAP scores with 8 bits hash length are drawn in Fig.5. We can see that these parameters possess some impact on the performance of our REPH. And REPH works well within a wide range of α and β . It indicates our method is very stable.

K. Ablation Analysis

In order to further gain a deep insight for our proposed REPH method, we perform ablation analysis on the NUS-WIDE dataset. Our method has two variations including RH and EPH. Thereinto, RH drops out the energy preserving term and EPH only uses single hash function to learn hash codes. We perform the ablation experiments on NUS-WIDE. The mAP scores of our REPH and two variations are given in table V. We can clearly observe that REPH obtains the best results than the two variations. It shows two-level hash function and energy preserving can reduce semantic information loss, thereby improving the quality of hash codes. Moreover, to show the superiority of our components, we further utilize some commonly used components to replace the proposed structures, thereby obtaining two additional models, *i.e.*, ARH and MRH. Specifically, for ARH, we use the asymmetric

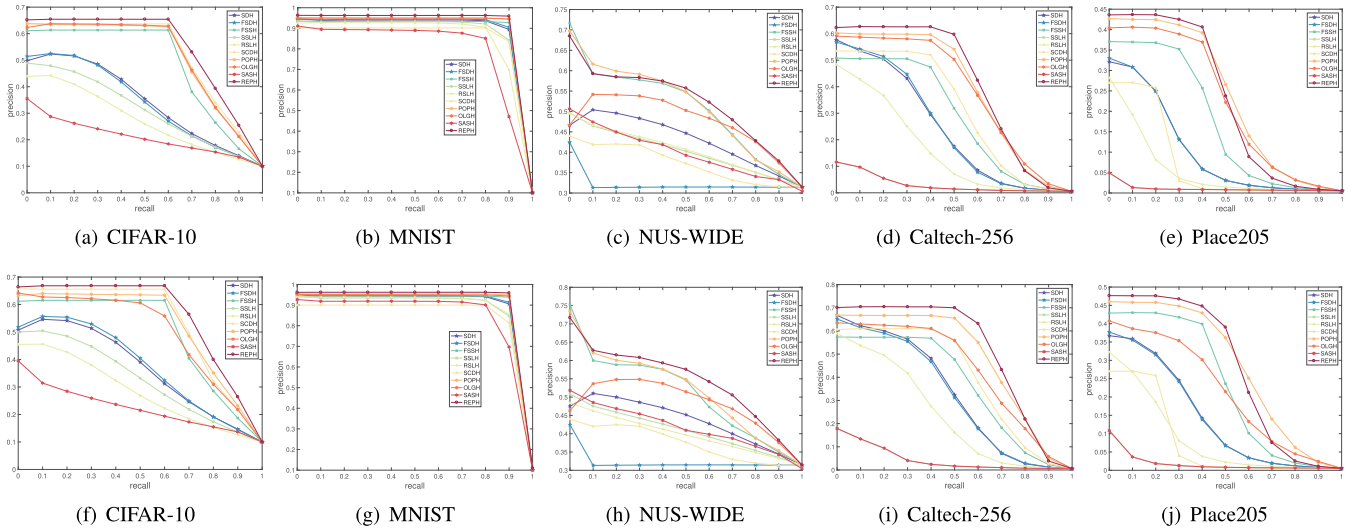


Fig. 3. The precision-recall curves for 32 bits (upper part) and 64 bit (lower part) hash codes on different datasets.

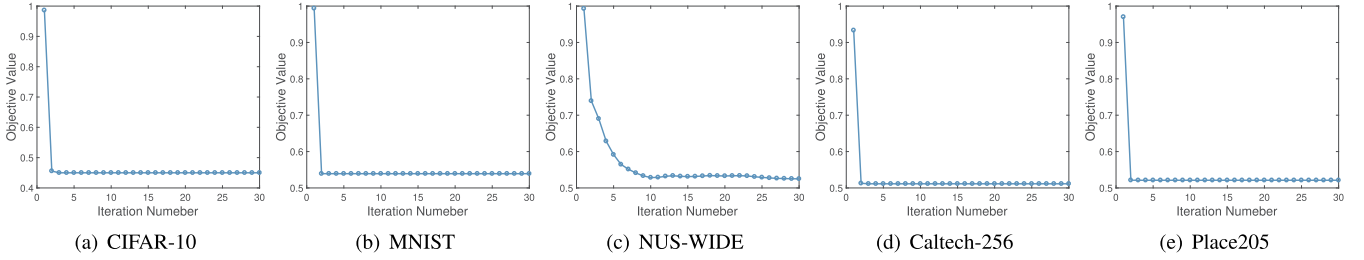


Fig. 4. Convergence curves on different datasets.

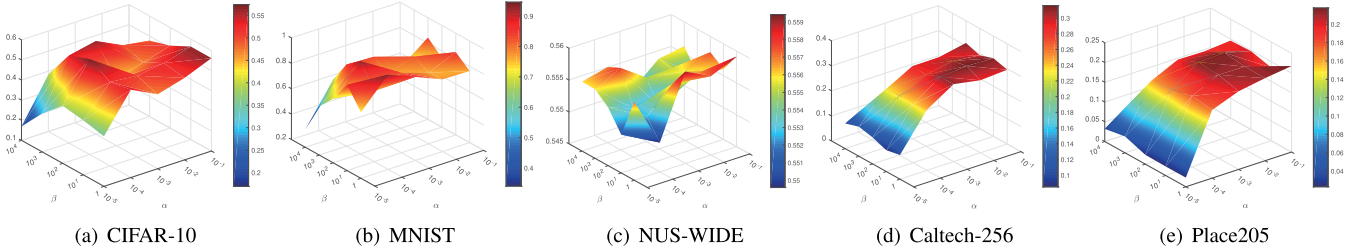


Fig. 5. Parameter analysis results for α and β on different datasets.

inner product with similarity preserving to replace energy preserving, *i.e.*,

$$\begin{aligned}
 \min_{P, B, W, R, Q} & \|B - RQX\|_F^2 + \alpha \|B^T B - IY^T Y\|_F^2 \\
 & + \beta \|B - WY\|_F^2 \\
 \text{s.t. } & B \in \{-1, 1\}^{l \times n}, R^T R = I \\
 & P^T P = I, W^T W = I
 \end{aligned} \quad (24)$$

For MRH, we use mutual regression to replace progressive learning strategy and energy preserving, *i.e.*,

$$\begin{aligned}
 \min_{P, B, W, Q} & \|B - QX\|_F^2 + \alpha \|X - PB\|_F^2 \\
 & + \beta \|B - WY\|_F^2 \\
 \text{s.t. } & B \in \{-1, 1\}^{l \times n}, Q^T Q = I \\
 & P^T P = I, W^T W = I
 \end{aligned} \quad (25)$$

TABLE VI
ABLATION MAP RESULTS (%) ON THE NUS-WIDE

Methods	8 bits	16 bits	32 bits	64 bits	128 bits
RH	53.25	57.98	60.24	61.34	63.04
EPH	55.47	58.21	59.98	61.24	62.15
ARH	53.16	57.85	60.12	62.23	63.07
MRH	50.82	56.93	59.72	60.45	63.28
REPH	57.63	60.07	61.89	63.16	64.15

From table V, we can see that our REPH still obtains the best scores. It indicates that our proposed components exhibits superiority over the commonly used components.

L. Retrieval Results

In Fig. 6, The top-8 image retrieval results of the best two methods (POPSH and REPH) are visualized on the Place205







Query	Retrieval results	Method
		POPSH
		REPH
		POPSH
		REPH

Fig. 6. The top-8 image retrieval results on Place205 dataset.

dataset. Red boxes indicate incorrect retrieval and green boxes indicate correct retrieval. We can observe that our method can search more images related to the query in the top-8 retrieval. In the wrong retrieval results, REPH cannot search the corresponding correct image because the query image and the retrieved image have certain visual similarity.

V. CONCLUSION

In this paper, we propose a novel industrial solution for fast large-scale image retrieval, termed Relaxed Energy Preserving Hashing (REPH). We first construct a flexible and effective transition space to learn latent representation, meanwhile adequately preserving the main energy of the original samples. Then, we propose a semantic mapping and reconstruction mechanism to guarantee similarity preservation, which can further guide to the generation of high-quality hash codes. Comprehensive experiments are performed to demonstrate the effectiveness of the proposed REPH. The experimental results show that REPH achieves satisfactory retrieval performance under different evaluation criteria compared with several state-of-the-art hashing methods.

REFERENCES

- [1] R. Lan, Y. Tan, X. Wang, Z. Liu, and X. Luo, "Label guided discrete hashing for cross-modal retrieval," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25236–25248, Dec. 2022.
- [2] W. Ou, J. Deng, L. Zhang, J. Gou, and Q. Zhou, "Cross-modal generation and pair correlation alignment hashing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3018–3026, Mar. 2023.
- [3] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 284–295, Jan. 2018.
- [4] X. Li, J. Yu, Y. Wang, J.-Y. Chen, P.-X. Chang, and Z. Li, "DAHP: Deep attention-guided hashing with pairwise labels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 933–946, Mar. 2022.
- [5] X. Nie, X. Liu, X. Xi, C. Li, and Y. Yin, "Fast unmediated hashing for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3669–3678, Sep. 2021.
- [6] Y. Cao, S. Chen, J. Gui, H. Qi, Z. Li, and C. Liu, "Hash learning with variable quantization for large-scale retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2624–2637, May 2022.
- [7] Y. Qin, Z. Tang, H. Wu, and G. Feng, "Flexible tensor learning for multi-view clustering with Markov chain," *IEEE Trans. Knowl. Data Eng.*, doi: [10.1109/TKDE.2023.3305624](https://doi.org/10.1109/TKDE.2023.3305624).
- [8] Y. Qin, N. Pu, and H. Wu, "Elastic multi-view subspace clustering with pairwise and high-order correlations," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 2, pp. 556–568, Feb. 2024, doi: [10.1109/TKDE.2023.3293498](https://doi.org/10.1109/TKDE.2023.3293498).
- [9] X. Li, Z. Ren, Q. Sun, and Z. Xu, "Auto-weighted tensor Schatten p -norm for robust multi-view graph clustering," *Pattern Recognit.*, vol. 134, Feb. 2023, Art. no. 109083.
- [10] X. Zang, G. Li, and W. Gao, "Multidirection and multiscale pyramid in transformer for video-based pedestrian retrieval," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8776–8785, Dec. 2022.
- [11] J. Yang, B. Jiang, B. Li, K. Tian, and Z. Lv, "A fast image retrieval method designed for network big data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2350–2359, Oct. 2017.
- [12] J. Ahmad, K. Muhammad, J. Lloret, and S. W. Baik, "Efficient conversion of deep features to compact binary codes using Fourier decomposition for multimedia big data," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3205–3215, Jul. 2018.
- [13] Z. Chen, J. Lu, J. Feng, and J. Zhou, "Nonlinear structural hashing for scalable video search," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1421–1433, Jun. 2018.
- [14] G. Wu, J. Han, Z. Lin, G. Ding, B. Zhang, and Q. Ni, "Joint image-text hashing for fast large-scale cross-media retrieval using self-supervised deep learning," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9868–9877, Dec. 2019.
- [15] X. Luo, Y. Wu, and X.-S. Xu, "Scalable supervised discrete hashing for large-scale search," in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 1603–1612.
- [16] Y. Sun, D. Peng, and Z. Ren, "Discrete aggregation hashing for image set classification," *Expert Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121615.
- [17] P. Hu, X. Peng, H. Zhu, J. Lin, L. Zhen, and D. Peng, "Joint versus independent multiview hashing for cross-view retrieval," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4982–4993, Oct. 2021.
- [18] P. Hu, X. Wang, L. Zhen, and D. Peng, "Separated variational hashing networks for cross-modal retrieval," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1721–1729.
- [19] Y. Sun, Z. Ren, P. Hu, D. Peng, and X. Wang, "Hierarchical consensus hashing for cross-modal retrieval," *IEEE Trans. Multimedia*, doi: [10.1109/TMM.2023.3272169](https://doi.org/10.1109/TMM.2023.3272169).
- [20] 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.
- [21] X. Shen, F. Shen, L. Liu, Y.-H. Yuan, W. Liu, and Q.-S. Sun, "Multiview discrete hashing for scalable multimedia search," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 5, pp. 1–21, Sep. 2018.
- [22] J. Tang and Z. Li, "Weakly supervised multimodal hashing for scalable social image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2730–2741, Oct. 2018.
- [23] Y. Cao, H. Qi, J. Gui, K. Li, Y. Y. Tang, and J. T. Kwok, "Learning to hash with dimension analysis based quantizer for image retrieval," *IEEE Trans. Multimedia*, vol. 23, pp. 3907–3918, 2021.
- [24] S. Wang, C. Li, and H.-L. Shen, "Distributed graph hashing," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1896–1908, May 2019.
- [25] C. Zheng, L. Zhu, Z. Cheng, J. Li, and A.-A. Liu, "Adaptive partial multi-view hashing for efficient social image retrieval," *IEEE Trans. Multimedia*, vol. 23, pp. 4079–4092, 2021.
- [26] R. Ji, H. Liu, L. Cao, D. Liu, Y. Wu, and F. Huang, "Toward optimal manifold hashing via discrete locally linear embedding," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5411–5420, Nov. 2017.
- [27] L. Yuan et al., "Central similarity quantization for efficient image and video retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3080–3089.
- [28] L. Zhu, C. Zheng, X. Lu, Z. Cheng, L. Nie, and H. Zhang, "Efficient multi-modal hashing with online query adaption for multimedia retrieval," *ACM Trans. Inf. Syst.*, vol. 40, no. 2, pp. 1–36, Apr. 2022.
- [29] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 490–496, Feb. 2018.
- [30] Z. Zhang et al., "Scalable supervised asymmetric hashing with semantic and latent factor embedding," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4803–4818, Oct. 2019.
- [31] X. Luo, L. Nie, X. He, Y. Wu, Z.-D. Chen, and X.-S. Xu, "Fast scalable supervised hashing," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 735–744.
- [32] Z. Zhang, X. Zhu, G. Lu, and Y. Zhang, "Probability ordinal-preserving semantic hashing for large-scale image retrieval," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 3, pp. 1–22, Jun. 2021.
- [33] Z. Zhang and C.-M. Pun, "Learning ordinal constraint binary codes for fast similarity search," *Inf. Process. Manage.*, vol. 59, no. 3, May 2022, Art. no. 102919.
- [34] Y. Sun, D. Peng, H. Huang, and Z. Ren, "Feature and semantic views consensus hashing for image set classification," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2097–2105.

- [35] Y. Sun, X. Wang, D. Peng, Z. Ren, and X. Shen, "Hierarchical hashing learning for image set classification," *IEEE Trans. Image Process.*, vol. 32, pp. 1732–1744, 2023.
- [36] H. Liu, R. Ji, J. Wang, and C. Shen, "Ordinal constraint binary coding for approximate nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 941–955, Apr. 2019.
- [37] X. Nie, X. Liu, J. Guo, L. Wang, and Y. Yin, "Supervised discrete multiple-length hashing for image retrieval," *IEEE Trans. Big Data*, vol. 9, no. 1, pp. 312–327, Feb. 2023.
- [38] Y. Luo, Z. Huang, Y. Li, F. Shen, Y. Yang, and P. Cui, "Collaborative learning for extremely low bit asymmetric hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 12, pp. 3675–3685, Dec. 2021.
- [39] W. Hong, Y.-T. Chang, H. Qin, W.-C. Hung, Y.-H. Tsai, and M.-H. Yang, "Image hashing via linear discriminant learning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2531–2539.
- [40] X. Liu, X. Nie, Q. Zhou, X. Xi, L. Zhu, and Y. Yin, "Supervised short-length hashing," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3031–3037.
- [41] X. Liu, X. Nie, Q. Dai, Y. Huang, L. Lian, and Y. Yin, "Reinforced short-length hashing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3655–3668, Sep. 2021.
- [42] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 37–45.
- [43] Z. Ren, Q. Sun, B. Wu, X. Zhang, and W. Yan, "Learning latent low-rank and sparse embedding for robust image feature extraction," *IEEE Trans. Image Process.*, vol. 29, pp. 2094–2107, 2020.
- [44] Y. Chen, Z. Tian, H. Zhang, J. Wang, and D. Zhang, "Strongly constrained discrete hashing," *IEEE Trans. Image Process.*, vol. 29, pp. 3596–3611, 2020.
- [45] J. Bai et al., "Deep progressive hashing for image retrieval," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 208–216.
- [46] Y. Ma, Y. He, F. Ding, S. Hu, J. Li, and X. Liu, "Progressive generative hashing for image retrieval," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 871–877.
- [47] Y. Shen et al., "Auto-encoding twin-bottleneck hashing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2818–2827.
- [48] G. Koutaki, K. Shirai, and M. Ambai, "Hadamard coding for supervised discrete hashing," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5378–5392, Nov. 2018.
- [49] X. Li, D. Hu, and F. Nie, "Large graph hashing with spectral rotation," in *Proc. 31st AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017, pp. 2203–2209.
- [50] Z. Lai, Y. Chen, J. Wu, W. K. Wong, and F. Shen, "Jointly sparse hashing for image retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 6147–6158, Dec. 2018.
- [51] Z. Weng and Y. Zhu, "Online hashing with bit selection for image retrieval," *IEEE Trans. Multimedia*, vol. 23, pp. 1868–1881, 2021.
- [52] Y. Shi, X. Nie, X. Liu, L. Zou, and Y. Yin, "Supervised adaptive similarity matrix hashing," *IEEE Trans. Image Process.*, vol. 31, pp. 2755–2766, 2022.



Yuan Sun received the M.Sc. degree in control science and engineering from the Southwest University of Science and Technology, Mianyang, China, in 2021. He is currently pursuing the Ph.D. degree in electronic information with Sichuan University, Chengdu, China. He has published more than ten papers in IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, AAAI, ACM MM, and other top journals in this field. His research interests include image set classification, information retrieval, and multi-modal learning.



Jian Dai received the Ph.D. degree in electronic and information engineering from the Beijing Institute of Technology in 2023. He is currently a Senior Engineer with China South Industries Group Corporation. He has published more than ten articles. His research interests include computer vision and large model.



Zhenwen Ren (Member, IEEE) received the Ph.D. degree from the Nanjing University of Science and Technology, China, in 2021. He is currently a Professor with the Southwest University of Science and Technology, China. He has published more than 80 peer-reviewed papers, including those in highly regarded journals and conferences, such as CVPR, AAAI, ACM MM, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

His research interests include computer vision, machine learning, deep learning, and industrial software. He serves on the editorial boards for several top tier journals.



Qilin Li received the B.Sc. degree in computer application and the M.Sc. and Ph.D. degrees in computer software and theory and computer application from the University of Electronic Science and Technology of China, Chengdu, China, in 1996, 1999, and 2006, respectively. He is currently a Professorate Senior Engineer with State Grid Sichuan Electric Power Company, Chengdu. His research interests include power systems, big data, and artificial intelligence.



Dezhong Peng received the B.Sc. degree in applied mathematics and the M.Sc. and Ph.D. degrees in computer software and theory from the University of Electronic Science and Technology of China, Chengdu, China, in 1998, 2001, and 2006, respectively. From 2001 to 2007, he was with the University of Electronic Science and Technology of China, as an Assistant Lecturer and a Lecturer. He was a Post-Doctoral Research Fellow with the School of Engineering, Deakin University, Geelong, VIC, Australia, from 2007 to 2009. He is currently a Professor with the College of Computer Science, Sichuan University, Chengdu. His research interests include neural networks and signal processing.