Generalized Debiased Semi-Supervised Hashing for Large-Scale Image Retrieval

Xingbo Liu[1], Xuening Zhang[2]*, Xiushan Nie[1,3], Yang Shi[4], Yilong Yin[4]

[1]School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China
[3]Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Co., Ltd, Jinan, China
[4]School of Software, Shandong University, Jinan 250101, China
{sclxb, shiyang}@mail.sdu.edu.cn, yukizhang0527@outlook.com, niexsh@hotmail.com, ylyin@sdu.edu.cn

Abstract

Semi-supervised hashing has shown promising efficacy in large-scale image retrieval, which learns similarity-preserving codes from both labeled and unlabeled data. To enable the use of advanced supervised hashing techniques, pseudo labels are widely applied. However, existing methods typically suffer from a biased learning issue due to pseudo label noise, which can be further aggravated during optimization. Although such a bias can adversely affect hashing accuracy, it has not been investigated sufficiently. In view of this, we present a comprehensive discussion on potential causes of biases, involving processes of pseudo-labeling, hash learning and optimization. Accordingly, a novel Generalized Debiased Semi-supervised Hashing (GDSH) method is proposed as a unified solution to mitigate the biases. Specifically, reliable pseudo labels are first predicted via a robust label completion strategy. Secondly, a debiased hash learning module is designed by combining label denoising and similarity updating. This can not only refine the supervision, but also obtain hash codes that are semantically debiased in both category and sample levels. Finally, a discrete semi-supervised hashing algorithm is proposed to alleviate the bias arising from optimization. Experimental results on three single-label and three multi-label image benchmarks demonstrate that GDSH remarkably outperforms the state-of-the-arts in different semi-supervised settings.

Introduction

With the development of data acquisition technology and the popularization of the Internet and social media, the amount of image data has increased exponentially (Wu et al. 2024; Zhu et al. 2020; Liu et al. 2022). To manage large-scale image database efficiently, hashing provides a practical solution by mapping high-dimensional visual features into compact binary codes while maintaining the original similarities. Therefore, the storage cost of massive data is substantially reduced, and the retrieval task can be accomplished quickly and effectively. Owing to these promising properties, hashing has become a highlighted topic in the field of image retrieval (Cui et al. 2024; Shen et al. 2024; Tang et al. 2024). Typically, existing approaches can generate high-quality hash codes when there are sufficient correctly-labeled data for training (Shen et al. 2015; Gui et al. 2018; Sun et al. 2024b). However, this requirement can hardly be met in practice. For one thing, due to the intensive labor of manual labeling, the observed data set often contains plenty of unlabeled instances (Wang et al. 2023b). For another, the given labels can be insufficient (or even wrong) to describe the corresponding image, especially in multi-label scenarios (Chua et al. 2009; Xia et al. 2023), *e.g.*, on photo-sharing websites and apps like Flickr and Instagram. Under these circumstances, training only on labeled samples may not guarantee high accuracy.

To reduce the dependency on labeled data, semi-supervised hashing approaches (Hu et al. 2019; Zhang and Peng 2019; Wang et al. 2023a) have been proposed by exploiting both labeled and unlabeled samples to promote retrieval performance. According to how the extensive unlabeled data are used, existing methods can be categorized into regularization-based ones (Wang, Kumar, and Chang 2012; Wu et al. 2013; Pan et al. 2015) and pseudo label-based ones (Song et al. 2017; Zhang and Peng 2019). The former typically formulate the hash learning problem as an empirical error minimization over labeled data, along with an unsupervised regularization over all the data (Wu et al. 2013; Kan et al. 2014). In such a formulation, unlabeled data provide helpful distribution information using their features, while the more discriminative category information remains unexplored. To address this gap, pseudo label-based approaches have been investigated (Song et al. 2017; Cheng et al. 2022; Wang et al. 2022a). By completing categorization information behind unlabeled instances, the semi-supervised setting is expanded into a supervised one, which supports the use of various supervised hashing techniques. Therefore, pseudo label-based semi-supervised hashing possesses high generality and adaptability.

However, it has been revealed lately that the pseudo-labeling stage is prone to generate unreliable or even false predictions(Chen et al. 2022; Kage et al. 2024). Simply treating pseudo labels as ground truth can result in a biased semi-supervised learning (Wang et al. 2022b). In order to alleviate this, debiased learning scheme has been designed for self-training (Chen et al. 2022) and imbalanced learning (Wang et al. 2022b). Nonetheless, it has not received enough attention in context of semi-supervised hashing. In fact, for

*Corresponding author.

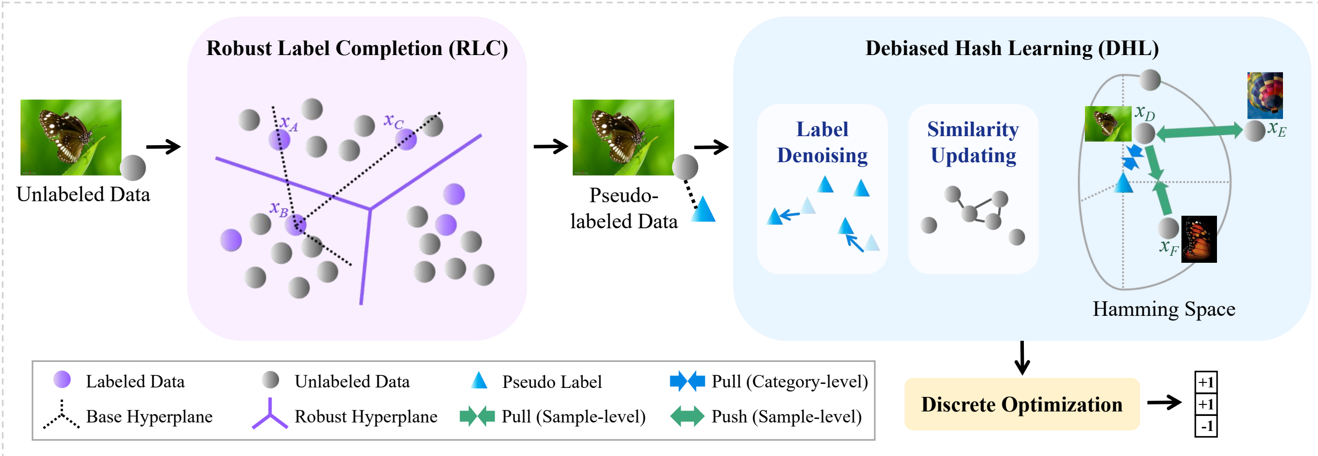


Figure 1: Framework of the proposed GDSH. The biases are reduced in the process of pseudo-labeling, hash learning and optimization. First, the RLC module generates more reliable pseudo labels based on robust classification hyperplane. Second, the DHL module calibrates the hash code representation by using two levels of debiased supervision, including denoised pseudo labels at the category level and updated similarities at the sample level. Finally, the binary codes are optimized discretely, so as to avoid bias caused by relaxation.

pseudo label-based semi-supervised hashing, biases can be born in every stage, including pseudo-labeling, hash learning and optimization. First, the noisy pseudo labels generally serve as supervision in existent hashing framework (Song et al. 2017; Zhang and Peng 2019), which would lead to biased hash codes. Although some recent work is carried out to solve pseudo label noise, they are generally based on sample weighting, which requires considerable training time and computational memory (Shi et al. 2021; Cheng et al. 2022; Fan et al. 2023). Second, in hash learning stage, unreasonable objectives can also produce training bias. Besides, the optimization of binary codes incurs bias when applying relaxation strategies (Shi et al. 2021). These biases, although implicit, are likely to limit the performance of semi-supervised hashing in large-scale data sets.

To obtain debiased hash codes in the general semi-supervised setting, we propose a novel Generalized Debiased Semi-supervised Hashing (GDSH) method. The proposed GDSH decouples the semi-supervised hash learning into two processes: robust label completion and debiased hash learning. Specifically, the pseudo label for unsupervised data is firstly predicted with a delicate strategy which is scalable for large-scale data. Then the hash codes for semi-supervised data are learned in a label-denoising fashion. Contributions of this work are summarized as follows:

- We identify the issue and discuss the causes of implicit bias in pseudo label-based semi-supervised hashing, and propose a unified framework to handle different biases.
- We design a robust label completion strategy to reliably annotate the unlabeled data, which can be well generalized to weakly-labeled collections with various label rates and numbers of categories.
- We propose a novel debiased hash learning method, where pseudo label noise is alleviated with lower complexity, and hash codes are semantically debiased in both category and sample levels.
- Theoretical analysis demonstrates the stability for learning the bridge between hash codes and semantics.

The Proposed Method

This section will detail the proposed GDSH method, including model formulation, learning algorithm, stability and complexity analysis. The framework of GDSH is shown in Fig. 1, which provides a unified and comprehensive solution to the implicitly biased semi-supervised hashing.

Notations and Problem Formulation

Given n training images $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n$, where $\mathbf{x}_i$ represents the feature vector of the i-th sample. $\mathbf{V} = [\mathbf{X}, \mathbf{X}_u] \in \mathbb{R}^{d\times n}$ denotes the feature matrix of the whole training set, comprising n_l labeled samples $\mathbf{X} \in \mathbb{R}^{d\times n_l}$ and n_u unlabeled samples $\mathbf{X}_u \in \mathbb{R}^{d\times n_u}$. The labels for $\mathbf{X}$ are given as $\mathbf{L} \in \{0,1\}^{c\times n_l}$, where c is the number of categories, $L_{ji} = 1$ if $\mathbf{x}_i$ belongs to the j-th category and 0 otherwise. Besides the labels, the semi-supervised training set also includes a pairwise similarity matrix $\mathbf{S} \in \{-1,0,1\}^{n\times n}$. $\mathbf{S}_{ij} = 1$ means that sample i and sample j are semantically similar (*e.g.* share at least one common label), and $\mathbf{S}_{ij} = -1$ means they are dissimilar. If either of the two samples is unlabeled, the semantic relationship is unknown, and $\mathbf{S}_{ij}$ is set to 0. The goal of GDSH is to learn the r-bit hash code representation $\mathbf{B} \in \{-1,1\}^{r\times n}$, which well preserves the semantic similarities among the semi-supervised data.

Throughout this article, matrices are denoted by boldface uppercase letters (like $\mathbf{A}$), vectors are denoted by lowercase bold letters (like $\mathbf{a}$ and $\mathbf{a}_i$), $\mathbf{A}_{ij}$ gives the element at the i-th row and j-th column in $\mathbf{A}$. $\mathbf{I}$ and $\mathbf{0}$ denote the identity matrix and matrix with all zeros, respectively. $\mathbf{A}^{-1}$, $\mathbf{A}^T$ and $Tr(\mathbf{A})$ represent the inverse, transpose and the trace of $\mathbf{A}$, respectively. A variety of norms on matrices and vectors will be used, including the ℓ_2-norm $\|\cdot\|$, ℓ_1-norm $\|\cdot\|_1$, $\ell_{2,1}$-norm $\|\cdot\|_{2,1}$ and the Frobenius norm $\|\cdot\|_F$. $sgn(\cdot)$ is the element-wise sign function which outputs 1 for positive elements and -1 otherwise. $max\{\cdot,\cdot\}$ and $|\cdot|$ return the maximum and the absolute value, respectively.

Robust Label Completion

Pseudo-labeling is a general solution to semi-supervised learning (Song et al. 2017; Shi et al. 2021; Zhang et al. 2024). It predicts the underlying associations between unlabeled data and semantic classes, so as to provide fully annotated data for downstream machine learning tasks. A basic method is to map the corresponding features to label space, formulated as $\mathbf{L}_u = \mathbf{P}\mathbf{X}_u$, where $\mathbf{L}_u \in \{0,1\}^{c\times n_u}$ denotes the pseudo labels, $\mathbf{P} \in \mathbb{R}^{c\times d}$ is the projection matrix. It is expected that $\mathbf{P}$ can discover the true semantic distribution, even when there is only few labeled data for training. Toward this end, we propose a robust modeling method by preserving the sparse structure of label matrix. The motivation is illustrated in Fig. 1, where a base hyperplane is presented as comparison. The base model aims at fitting the observed labeled data precisely, *i.e.*, $\mathbf{L} = \mathbf{P}\mathbf{X}$. However, such hyperplane crosses over samples x_A, x_B, x_C that are associated with multiple categories. Moreover, it can vary a lot depending on the selected data set, leaving pseudo labels less reliable. For the pursuit of robustness, the $\ell_{2,1}$-norm regularization (Nie et al. 2010) is introduced to keep the sparsity of label vectors. $\mathbf{P}$ is optimized with

$$\min_{\mathbf{P}} \|\mathbf{L}-\mathbf{P}\mathbf{X}\|^2 + \delta \|\mathbf{P}\mathbf{X}\|_{2,1}, \tag{1}$$

where δ is a trade-off parameter. Eq. (1) encourages assigning less number of categories to a single data. This improves the discriminative ability of classification hyperplane, reducing the probability of false predictions.

Apart from sparsity preserving property, it is also desirable that the projections are insensitive to data variation, so an extra regularization term is imposed on $\mathbf{P}$. The objective for learning the robust hyperplane becomes

$$\min_{\mathbf{P}} \|\mathbf{L}-\mathbf{P}\mathbf{X}\|^2 + \delta_1 \|\mathbf{P}\mathbf{X}\|_{2,1} + \delta_2 \|\mathbf{P}\|^2. \tag{2}$$

In practice, by adjusting the hyperparameters δ_1 and δ_2, an optimal fitting degree can be achieved. Therefore, the RLC model in Eq. (2) can adapt to data collections with different label rates and different numbers of categories.

For the convex programming in Eq. (2), a global optimum solution can be reached at the stationary point (Markovsky and Van Huffel 2007). Setting the derivative *w.r.t* $\mathbf{P}$ equal to zero arrives at

$$2\mathbf{P}\mathbf{X}\mathbf{X}^T - 2\mathbf{L}\mathbf{X}^T + 2\delta_2\mathbf{P} + 2\delta_1\mathbf{D}\mathbf{P}\mathbf{X}\mathbf{X}^T = \mathbf{0}, \tag{3}$$

where $\mathbf{D}$ is a diagonal matrix dependent on $\mathbf{P}$. Let $(\mathbf{P}\mathbf{X})^i$ be the i-th row of $\mathbf{P}\mathbf{X}$, then the i-th diagonal element of $\mathbf{D}$ is $(2\|(\mathbf{P}\mathbf{X})^i\|)^{-1}$.

$\mathbf{P}$ can be obtained via solving Sylvester equation (Golub and Loan 1996) below:

$$(\delta_1\mathbf{D}+\mathbf{I})\mathbf{P} + \delta_2\mathbf{P}(\mathbf{X}\mathbf{X}^T)^{-1} = \mathbf{L}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}. \tag{4}$$

The optimization of RLC model alternatively proceeds over the variables $\mathbf{P}$ and $\mathbf{D}$ till convergence. With the learned projections, predictions for unlabeled data are easily given by $\mathbf{P}\mathbf{X}_u$. In the following discussion, we denote the complete labels of semi-supervised data as $\mathbf{Y} = [\mathbf{L}, \mathbf{L}_u] \in \{0,1\}^{c\times n}$, where $\mathbf{L}_u$ keeps the top-k predicted categories.

After pseudo-labeling, existing methods (Song et al. 2017; Zhang and Peng 2019) usually leverage the complete supervision to guide hash learning. However, pseudo labels inevitably contain noise due to practical factors, such as random data sampling and utilized feature representations (Chen et al. 2022). Such noisy supervision can mislead learning process and eventually cause bias in hash codes. Therefore, it is urgent to denoise the labels for better performance.

Debiased Hash Learning

Due to its intrinsic discriminative and binary properties, label is commonly regarded as another kind of data representation (Liu and Lu 2016; Liu et al. 2019). Thus, it is natural to regress the class labels to hash codes for building a bridge between them. For example, $\mathbf{B} = \mathbf{A}\mathbf{Y}$, where $\mathbf{A} \in \mathbb{R}^{r\times c}$ represents the to-be-learnt projection matrix. Besides, the pairwise similarities can contribute to achieving the similarity maintaining goal of hash learning (Wang, Kumar, and Chang 2012; Zhu et al. 2023). In the proposed model, we define the affinities among samples according to their labels, and construct matrix $\mathbf{S} \in \{-1,1\}^{n\times n}$ using $sgn(\mathbf{Y}^T\mathbf{Y})$. The semantic preserving objective is formulated as

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{S}-\mathbf{B}^T\mathbf{B}\|^2 + \gamma\|\mathbf{B}-\mathbf{A}\mathbf{Y}\|^2 + \theta\|\mathbf{A}\|^2, \tag{5}$$

where γ and θ are balance parameters. Note that the term $\|\mathbf{S}-\mathbf{B}^T\mathbf{B}\|^2$ can be substituted by $\|\mathbf{S}-\mathbf{B}^T\mathbf{A}\mathbf{Y}\|^2$, where the real-valued $\mathbf{A}\mathbf{Y}$ is promising to embed more precise similarity information. Then, Eq. (5) is rewritten as

$$\min_{\mathbf{A},\mathbf{B}} \|\mathbf{S}-\mathbf{B}^T\mathbf{A}\mathbf{Y}\|^2 + \gamma\|\mathbf{B}-\mathbf{A}\mathbf{Y}\|^2 + \theta\|\mathbf{A}\|^2. \tag{6}$$

The major focus is to alleviate the bias brought by the pseudo label noise, and meanwhile devise reasonable objective for debiased hashing. As a feasible solution, recent methods have adopted sample weighting(Shi et al. 2021; Cheng et al. 2022), which examines the quality of each pseudo-labeled data. However, this requires extra computation and memory space, which is unacceptable in large-scale applications. To solve this problem, we design a valid and low-cost DHL approach in this paper. As shown in Fig. 1, the supervision is refined with a category-level debiasing strategy and a sample-level one. The former aims at denoising the original pseudo labels and recovering the clean label information. To this end, we assume there exists a ground-truth label matrix $\mathbf{Y}_u$ that can be obtained with the sum of pseudo label matrix $\mathbf{L}_u$ and a noise correction matrix $\mathbf{E}$, *i.e.*, $\mathbf{Y}_u = \mathbf{L}_u + \mathbf{E}$. By minimizing the pseudo-labeling error $\mathbf{E}$, the label denoising problem is formulated as follows:

$$\min_{\mathbf{E},\mathbf{Y}_u} \|\mathbf{E}\|_1, \text{ s.t. } \mathbf{Y}_u = \mathbf{L}_u + \mathbf{E}, \mathbf{Y}_u \in \{0,1\}^{c\times n_u}. \tag{7}$$

The ℓ_1-norm constraint guarantees that label noise is sparse, which is natural since $\mathbf{E}$ is a sub-matrix of label matrix $\mathbf{Y}_u$.

Once $\mathbf{Y}_u$ is optimized, the semantic relations among training samples change. To make hash codes more compatible with the debiased semantic distribution, samples associated with the same label (*e.g.* x_D and x_F in Fig. 1) need to be pulled together, while samples with different labels (*e.g.* x_D and x_E) must be pushed away. Therefore, an extra sample-level debiasing scheme is proposed. More specifically, the pairwise similarities in $\mathbf{S}$ are dynamically updated to be consistent with new labels $\mathbf{Y} = [\mathbf{L}, \mathbf{Y}_u]$ and then used to calibrate the hash model.

Based on the label denoising in Eq. (7) and the subsequent similarity updating, we can derive the following unified objective function:

$$\min_{\mathcal{W}} \|\mathbf{S} - \mathbf{B}^T\mathbf{A}\mathbf{Y}\|^2 + \gamma\|\mathbf{B} - \mathbf{A}\mathbf{Y}\|^2 + \mu\|\mathbf{E}\|_1 + \theta\|\mathbf{A}\|^2,$$
$$\text{s.t.} \mathbf{Y}_u = \mathbf{L}_u + \mathbf{E}, \mathbf{Y}_u \in \{0,1\}^{c\times n_u}, \mathbf{B} \in \{-1,1\}^{r\times n}, \tag{8}$$

where $\mathcal{W} = \{\mathbf{E}, \mathbf{Y}_u, \mathbf{A}, \mathbf{B}, \mathbf{S} = sgn(\mathbf{Y}^T\mathbf{Y})\}$, μ is a trade-off parameter. Eq. (8) collectively optimizes the supervisory information and hash codes, which has two main advantages. For one thing, the pseudo semantics can be refined without adding much memory complexity. For another, the codes $\mathbf{B}$ can be semantically debiased in both category and sample levels, thus reflecting more accurate similarities among semi-supervised samples.

Optimizing Eq. (8) can suffer from huge time complexity if all supervised information in $\mathbf{S}$ is used during training (Kang, Li, and Zhou 2016). Hence, a relatively small subset of the whole training set is selected for sample-level debiasing. Concretely, we adopt a sampling ratio $\beta (0 < \beta < 1)$ and randomly select βn samples to represent the entire set. Accordingly, $\widetilde{\mathbf{S}} \in \{-1,1\}^{\beta n\times n}$ can be obtained based on the row sampling of $\mathbf{S}$, and $\widetilde{\mathbf{B}}$ denotes the corresponding codes. Reformulate Eq. (8) with the sampled similarities, we get the optimization problem for GDSH:

$$\min_{\mathcal{W}} \|\widetilde{\mathbf{S}} - \widetilde{\mathbf{B}}^T\mathbf{A}\mathbf{Y}\|^2 + \gamma\|\mathbf{B} - \mathbf{A}\mathbf{Y}\|^2 + \mu\|\mathbf{E}\|_1 + \theta\|\mathbf{A}\|^2,$$
$$\text{s.t.} \mathbf{Y}_u = \mathbf{L}_u + \mathbf{E}, \mathbf{Y}_u \in \{0,1\}^{c\times n_u}, \mathbf{B} \in \{-1,1\}^{r\times n}. \tag{9}$$

Optimization

To eliminate the equality constraint in Eq. (9), we consider its augmented Lagrangian function:

$$\begin{aligned}&\mathcal{L}(\mathbf{E}, \mathbf{Y}_u, \mathbf{A}, \mathbf{B}, \mathbf{G}, \rho)\\ =&\|\widetilde{\mathbf{S}} - \widetilde{\mathbf{B}}^T\mathbf{A}\mathbf{Y}\|^2 + \gamma\|\mathbf{B} - \mathbf{A}\mathbf{Y}\|^2 + \mu\|\mathbf{E}\|_1 + \theta\|\mathbf{A}\|^2\\ &+ Tr(\mathbf{G}^T(\mathbf{Y}_u - \mathbf{L}_u - \mathbf{E})) + \frac{\rho}{2}\|\mathbf{Y}_u - \mathbf{L}_u - \mathbf{E}\|^2,\end{aligned} \tag{10}$$

where $\mathbf{G}$ is the Lagrange multiplier, ρ is a positive penalty parameter. It is nontrivial to minimize $\mathcal{L}(\mathbf{E}, \mathbf{Y}_u, \mathbf{A}, \mathbf{B}, \mathbf{G}, \rho)$ *w.r.t* the variables simultaneously, while this separable objective function can be solved via ADM (Gabay and Mercier 1976). Specifically, at each iteration, $\mathbf{E}$, $\mathbf{Y}_u$ (together with $\mathbf{S}$), $\mathbf{G}$, $\mathbf{A}$ and $\mathbf{B}$ are optimized in an alternating order by holding the others constant. Consequently, we have the following iterative updating schemes:

Update E: The optimal solution $\mathbf{E}^*$ is given by

$$\begin{aligned}\mathbf{E}^* =& \arg\min_{\mathbf{E}} \ \mu\|\mathbf{E}\|_1 + Tr((\mathbf{Y}_u - \mathbf{L}_u - \mathbf{E})^T\mathbf{G})\\ &+ \frac{\rho}{2}\|\mathbf{Y}_u - \mathbf{L}_u - \mathbf{E}\|^2\\ =& \arg\min_{\mathbf{E}} \ \mu\|\mathbf{E}\|_1 + \frac{\rho}{2}\|\mathbf{E} - (\mathbf{Y}_u - \mathbf{L}_u + \frac{1}{\rho}\mathbf{G})\|^2\\ =& Soft_{\frac{\mu}{\rho}}(\mathbf{Y}_u - \mathbf{L}_u + \frac{1}{\rho}\mathbf{G}),\end{aligned} \tag{11}$$

where $Soft(\cdot)$ is the soft-thresholding (shrinkage) operator for ℓ_1-min problems (Beck and Teboulle 2009). When applied to matrices, it defines element-wise as follows:

$$Soft_{\frac{\mu}{\rho}}(x) = sgn(x)max\{|x| - \frac{\mu}{\rho}, 0\}. \tag{12}$$

Update $\mathbf{Y}_u$: The sub-problem reads

$$\begin{aligned}&\min_{\mathbf{Y}_u} \|\widetilde{\mathbf{S}}_u - \widetilde{\mathbf{B}}^T\mathbf{A}\mathbf{Y}_u\|^2 + \gamma\|\mathbf{B}_u - \mathbf{A}\mathbf{Y}_u\|^2\\ &\quad + Tr(\mathbf{G}^T(\mathbf{Y}_u - \mathbf{L}_u - \mathbf{E})) + \frac{\rho}{2}\|\mathbf{Y}_u - \mathbf{L}_u - \mathbf{E}\|^2,\\ &\text{s.t.} \quad \mathbf{Y}_u \in \{0,1\}^{c\times n_u},\end{aligned} \tag{13}$$

where $\widetilde{\mathbf{S}}_u$ and $\mathbf{B}_u$ denote the last n_u columns of $\widetilde{\mathbf{S}}$ and $\mathbf{B}$, respectively. With other variables fixed, Eq. (13) can be simplified as

$$\begin{aligned}&\min_{\mathbf{Y}_u} \quad Tr(\mathbf{Y}_u^T\mathbf{Q}_1\mathbf{Y}_u) - 2Tr(\mathbf{Y}_u^T\mathbf{Q}_2),\\ &\text{s.t.} \quad \mathbf{Y}_u \in \{0,1\}^{c\times n_u},\end{aligned} \tag{14}$$

where $\mathbf{Q}_1 = \mathbf{A}^T\widetilde{\mathbf{B}}\widetilde{\mathbf{B}}^T\mathbf{A} + \gamma\mathbf{A}^T\mathbf{A} + \frac{\rho}{2}\mathbf{I}$, $\mathbf{Q}_2 = \mathbf{A}^T\widetilde{\mathbf{B}}\widetilde{\mathbf{S}}_u + \gamma\mathbf{A}^T\mathbf{B}_u - \frac{1}{2}\mathbf{G} + \frac{\rho}{2}(\mathbf{L}_u + \mathbf{E})$. Given $\mathbf{Q}_1$ is symmetric, the objective in Eq. (14) has a closed-form solution at $\widehat{\mathbf{Y}}_u = \mathbf{Q}_1^{-1}\mathbf{Q}_2$. Regarding the intractable constraint $\mathbf{Y}_u \in \{0,1\}^{c\times n_u}$, it is suitable to replace the set $\{0,1\}$ by its convex hull, *i.e.*, the interval $\{y|0 \le y \le 1\}$ (Mandal, Chaudhury, and Biswas 2019). Then,

$$\mathbf{Y}_u = Proj_{[0,1]}(\mathbf{Q}_1^{-1}\mathbf{Q}_2), \tag{15}$$

where $Proj(\cdot)$ is defined by

$$Proj_{[0,1]}(x) = \begin{cases}0, if\ x < 0,\\ x, if\ x \in [0,1],\\ 1, if\ x > 1.\end{cases} \tag{16}$$

Update S: Once $\mathbf{Y}_u$ is obtained, the complete labels $\mathbf{Y}$ and the pairwise similarities $\mathbf{S}$ are accordingly updated using

$$\mathbf{Y} = [\mathbf{L}, \mathbf{Y}_u], \mathbf{S} = sgn(\mathbf{Y}^T\mathbf{Y}). \tag{17}$$

Update G: The Lagrange multiplier is updated through the gradient ascent on the dual problem:

$$\mathbf{G}^{(t+1)} = \mathbf{G}^{(t)} + \rho(\mathbf{Y}_u^{(t+1)} - \mathbf{L}_u - \mathbf{E}^{(t+1)}). \tag{18}$$

Update A: By fixing all the variables but $\mathbf{A}$, we reach

$$\min_{\mathbf{A}} \|\widetilde{\mathbf{S}} - \widetilde{\mathbf{B}}^T\mathbf{A}\mathbf{Y}\|^2 + \gamma\|\mathbf{B} - \mathbf{A}\mathbf{Y}\|^2 + \theta\|\mathbf{A}\|^2. \quad (19)$$

After setting the derivative of Eq. (19) *w.r.t* $\mathbf{A}$ equal to zero, we get the equation below:

$$(\gamma\mathbf{I} + \widetilde{\mathbf{B}}\widetilde{\mathbf{B}}^T)\mathbf{A} + \theta\mathbf{A}(\mathbf{Y}\mathbf{Y}^T)^{-1} = (\gamma\mathbf{B}\mathbf{Y}^T + \widetilde{\mathbf{B}}\widetilde{\mathbf{S}}\mathbf{Y}^T)(\mathbf{Y}\mathbf{Y}^T)^{-1}. \quad (20)$$

This is a Sylvester equation (Golub and Loan 1996), and $\mathbf{A}$ can be solved with an analytical solution using existing tools in MATLAB.

Update B: Notice that two separable parts of $\mathbf{B}$ are involved in the optimization problem Eq. (9), *i.e.*, $\widetilde{\mathbf{B}}$ and the matrix excluding $\widetilde{\mathbf{B}}$. Let $\widetilde{\mathbf{Y}}$ denote the label vectors corresponding to $\widetilde{\mathbf{B}}$, $\overline{\mathbf{Y}}$ be the complement of $\widetilde{\mathbf{Y}}$. Then, the latter part of $\mathbf{B}$ can be computed by $sgn(\mathbf{A}\overline{\mathbf{Y}})$. While for $\widetilde{\mathbf{B}}$, we have the following discrete optimization problem:

$$\min_{\widetilde{\mathbf{B}}} \|\widetilde{\mathbf{S}} - \widetilde{\mathbf{B}}^T\mathbf{A}\mathbf{Y}\|^2 + \gamma\|\widetilde{\mathbf{B}} - \mathbf{A}\widetilde{\mathbf{Y}}\|^2, \quad \text{s.t.} \quad \widetilde{\mathbf{B}} \in \{-1,1\}^{r\times\beta n}, \quad (21)$$

which can be simplified as

$$\min_{\widetilde{\mathbf{B}}} \|\mathbf{Q}_3^T\widetilde{\mathbf{B}}\|^2 - 2Tr(\widetilde{\mathbf{B}}^T\mathbf{Q}_4), \text{s.t.}\, \widetilde{\mathbf{B}} \in \{-1,1\}^{r\times\beta n}, \quad (22)$$

where $\mathbf{Q}_3 = \mathbf{A}\mathbf{Y}$, $\mathbf{Q}_4 = \gamma\mathbf{A}\widetilde{\mathbf{Y}} + \mathbf{A}\mathbf{Y}\widetilde{\mathbf{S}}^T$.

The binary constraints imposed in Eq. (22) make the problem NP-hard. To facilitate optimization, previous methods (Wang, Kumar, and Chang 2012; Kan et al. 2014; Zhang and Peng 2019; Cheng et al. 2022; Shi et al. 2021) discard the constraints and solve a relaxed version. However, the introduced quantization error can enlarge and reinforce the potential bias in hash codes, especially for long codes (Liu et al. 2024). To alleviate this issue, GDSH optimizes $\widetilde{\mathbf{B}}$ discretely with discrete cyclic coordinate descent (DCC) (Shen et al. 2015) method. Denote the i-th row of $\widetilde{\mathbf{B}}$, $\mathbf{Q}_3$ and $\mathbf{Q}_4$ as $\mathbf{b}^T$, $\mathbf{q}_3^T$ and $\mathbf{q}_4^T$, respectively. Matrices excluding the i-th row are $\overline{\mathbf{B}}$, $\overline{\mathbf{Q}}_3$ and $\overline{\mathbf{Q}}_4$. Then, each of the r bits can be optimized with the closed-form solution below:

$$\mathbf{b} = sgn(\mathbf{q}_4 - \overline{\mathbf{B}}^T\overline{\mathbf{Q}}_3\mathbf{q}_3). \quad (23)$$

Out-of-sample Extension

Based on Algorithm 1, all training images are mapped to binary codes without reference to unseen images. To accommodate any query image, it is necessary to learn hash functions based on training data $\mathbf{V}$ and its hash code matrix $\mathbf{B}$. Here a linear hash model is adopted for efficiency, which is determined by the following regression problem:

$$\min_{\mathbf{W}} \|\mathbf{B} - \mathbf{W}^T\mathbf{V}\|^2 + \alpha\,\|\mathbf{W}\|^2, \quad (24)$$

where the projection matrix $\mathbf{W} \in \mathbb{R}^{d\times r}$ can be solved with $(\mathbf{V}\mathbf{V}^T + \alpha\mathbf{I})^{-1}\mathbf{V}\mathbf{B}^T$, and the hash code representation of a given query $\mathbf{x}$ is obtained by $sgn(\mathbf{W}^T\mathbf{x})$.

Theoretical Analysis

Stability As a semi-supervised hashing algorithm, the proposed GDSH is expected to be stable, *i.e.*, the output will not vary significantly when a training sample is replaced by an independent and identically distributed (iid) one (Gui et al. 2018). Thus, the learning of hash model is robust against the variations in training set partitioning. Let $\mathcal{S}$ represent the sample set for training GDSH, and $\mathcal{S}^i$ be identical to $\mathcal{S}$ except that the i-th sample be replaced by an iid one. Without loss of generality, assume that Algorithm 1 stops with T iterations and in the t-th iteration ($t = 1, \cdots, T$), $\mathbf{B}^{(t-1)}$, $\mathbf{Y}^{(t)}$, $\mathbf{S}^{(t)}$, $\widetilde{\mathbf{S}}^{(t)}$ and $\widetilde{\mathbf{B}}^{(t)}$ are obtained. It can be proved that GDSH is stable when learning the projection matrix $\mathbf{A}$, which connects semantic labels and hash codes.

Theorem 1. *Let $\mathbf{A}_{\mathcal{S}}^{(t)}$ and $\mathbf{A}_{\mathcal{S}^i}^{(t)}$ represent the learned projections by employing the samples $\mathcal{S}$ and $\mathcal{S}^i$, respectively. In addition, assume that for any learned $\mathbf{Y}$, $\mathbf{B}$ and $\mathbf{A}$, we have $\|sgn(\mathbf{y}^T\mathbf{y}) - \mathbf{b}^T\mathbf{A}\mathbf{y}\| \le M_1$, $\|\mathbf{b} - \mathbf{A}\mathbf{y}\| \le M_2$, where M_1 and M_2 are two constants. Then the proposed Algorithm 1 satisfies the following stability inequality:*

$$\|\mathbf{A}_{\mathcal{S}}^{(t)} - \mathbf{A}_{\mathcal{S}^i}^{(t)}\|_F \le \frac{2c^2}{\theta n}\left(\frac{1+\beta}{\beta}r^2M_1 + \gamma M_2\right). \quad (25)$$

The proof of Theorem 1 is provided in the appendix.

Complexity Analysis For the proposed RLC, computing $\mathbf{P}$ and $\mathbf{D}$ respectively need $\mathcal{O}(n_lcd + n_ld^2 + cd^2 + d^3 + c^3)$ and $\mathcal{O}(cdn_l)$. Predicting pseudo labels $\mathbf{L}_u$ requires $\mathcal{O}(cdn_u)$, which can be completed before hash learning. For the proposed DHL, updating $\mathbf{E}$, $\mathbf{Y}_u$, $\mathbf{S}$ and $\mathbf{G}$ using Eqs. (11), (15) and (18) require $\mathcal{O}(cn_u)$, $\mathcal{O}(n_u\beta nc + n_uc^2 + n_ucr + \beta ncr + c^2r + c^3)$, $\mathcal{O}(n^2c)$ and $\mathcal{O}(cn_u)$, respectively. Updating $\mathbf{A}$ based on Eq. (20) needs $\mathcal{O}(\beta n^2r + nc^2 + ncr + \beta nr^2 + c^3 + rc^2 + r^3)$. Optimizing $\mathbf{B}$ based on Eq. (23) and $sgn(\mathbf{A}\overline{\mathbf{Y}})$ respectively require $\mathcal{O}(n^2r^2 + \beta n^2r + ncr)$ and $\mathcal{O}(ncr - \beta ncr)$. Seeing that $r, c, d \ll n, \beta < 1, n_l + n_u = n$, the overall training complexity is $\mathcal{O}(n^2(ct + r^2)T)$, where t and T are the maximum inner and outer iteration numbers, respectively. In practice, a few iterations are enough to guarantee convergence as well as satisfactory performance. For out-of-sample extension, generating hash code for a given query needs only a constant $\mathcal{O}(rd)$, which is efficient for large-scale applications.

Compared with previous strategies for pseudo label denoising (Shi et al. 2021; Cheng et al. 2022), the proposed DHL is free from the weighting of massive pseudo-labeled samples, saving at least a $\mathcal{O}(cn_uT)$ time complexity and a $\mathcal{O}(n_u)$ memory cost. Owing to the row sampling, the pairwise similarity preserving loss in Eq. (9) reduces the $\mathcal{O}(n^2)$ computational complexity to $\mathcal{O}(\beta n^2)$. Here β can be set to 0.3, since it generally guarantees satisfactory efficiency as well as accuracy.

Experiments

To verify the superiority of the proposed method, we carried out experiments using six widely-used image benchmarks,

Method	CALTECH-101				CIFAR-10				ImageNet				MS-COCO				NUS-WIDE			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
SH	0.2746	0.3134	0.3187	0.3219	0.2799	0.2906	0.2924	0.2978	0.0418	0.0576	0.0589	0.0662	0.6745	0.6577	0.6584	0.6718	0.5977	0.6030	0.5902	0.5830
PCA-ITQ	0.1158	0.1579	0.2059	0.2391	0.3114	0.3467	0.3452	0.3472	0.0277	0.0353	0.0409	0.0467	0.5839	0.6536	0.6942	0.7086	0.3476	0.5233	0.5877	0.6149
PCA-RR	0.1419	0.2029	0.2393	0.2952	0.2853	0.3050	0.3151	0.3236	0.0348	0.0370	0.0463	0.0515	0.6368	0.6478	0.6832	0.6974	0.4898	0.5757	0.6140	0.6062
MFH	0.2673	0.3220	0.3582	0.3732	0.2820	0.2963	0.3079	0.3169	0.0422	0.0602	0.0736	0.0847	0.6256	0.6486	0.6498	0.6612	0.5627	0.5909	0.6058	0.6115
SDH	0.2414	0.3416	0.4034	0.4426	0.4223	0.4781	0.4941	0.5026	0.0543	0.0853	0.1689	0.1880	0.7502	0.7708	0.7580	0.7652	0.6413	0.6796	0.6862	0.6715
NSH	0.3673	0.4285	0.4661	0.4859	0.3707	0.4050	0.4194	0.4274	0.1430	0.2304	0.2824	0.3154	0.7886	0.8078	0.8159	0.8247	0.7189	0.7494	0.7663	0.7767
FSDH	0.3580	0.4437	0.4862	0.4851	0.4134	0.4854	0.5022	0.5088	0.1635	0.2670	0.3240	0.3348	0.8140	0.8320	0.8391	0.8468	0.7485	0.7735	0.7912	0.7979
SDHMLR	0.3074	0.4005	0.4631	0.4803	0.4473	0.4933	0.5031	0.5100	0.1352	0.2518	0.3040	0.3316	0.8174	0.8274	0.8296	0.8316	0.7442	0.7709	0.7851	0.7950
SCDH	0.4238	0.4271	0.4493	0.4620	0.5918	0.5913	0.5986	0.5994	0.1127	0.1042	0.1170	0.1697	0.8318	0.8475	0.8562	0.8584	0.7645	0.7888	0.7968	0.8038
LBSE	0.4072	0.4585	0.4820	0.4956	0.5725	0.5981	0.6081	0.6113	0.1002	0.1699	0.2264	0.2613	0.8212	0.8396	0.8438	0.8503	0.7462	0.7704	0.7843	0.7926
DSPH	0.2958	0.3149	0.3252	0.3279	0.4165	0.4544	0.4610	0.4719	0.0531	0.0832	0.1047	0.1213	0.6727	0.6898	0.6981	0.7082	0.6268	0.6509	0.6630	0.6669
REPH	0.2894	0.3045	0.3193	0.3254	0.4280	0.4519	0.4657	0.4689	0.0512	0.0780	0.1044	0.1167	0.6517	0.7009	0.6982	0.7129	0.6207	0.6456	0.6557	0.6608
SSH-orth	0.2792	0.3465	0.3871	0.4196	0.2836	0.3098	0.3202	0.3277	0.0798	0.1298	0.1572	0.1689	0.7392	0.7291	0.7269	0.7233	0.6316	0.6659	0.6909	0.6911
SSH-nonorth	0.3527	0.3500	0.3442	0.3360	0.3275	0.3281	0.3180	0.3120	0.0914	0.1008	0.0920	0.0863	0.7609	0.7601	0.7612	0.7596	0.6970	0.6976	0.6985	0.6983
SPLH	0.2719	0.3040	0.2896	0.2870	0.2994	0.3069	0.3063	0.3361	0.0726	0.0975	0.0823	0.0999	0.7393	0.6030	0.5542	0.5521	0.6469	0.5574	0.5254	0.4618
KRSHSC	0.4064	0.4412	0.4626	0.4755	0.4268	0.4493	0.4499	0.4657	0.1735	0.1820	0.1919	0.1948	0.7789	0.8004	0.8083	0.8385	0.7000	0.7515	0.7473	0.7688
SMH	0.3954	0.4509	0.4845	0.5092	0.6187	0.5700	0.6100	0.5887	0.1818	0.2526	0.2404	0.2762	0.8132	0.8261	0.8259	0.8209	0.7602	0.7655	0.7670	0.7470
MLAGH	0.4273	0.4439	0.4472	0.4462	0.5684	0.5689	0.5737	0.5744	0.1802	0.2779	0.2799	0.2808	0.8072	0.8329	0.8306	0.8291	0.7416	0.7668	0.7717	0.7716
GDSH	**0.4301**	**0.4835**	**0.5054**	**0.5163**	**0.6197**	**0.6422**	**0.6515**	**0.6565**	**0.1961**	**0.2822**	**0.3254**	**0.3473**	**0.8370**	**0.8581**	**0.8629**	**0.8678**	**0.7655**	**0.7890**	**0.7983**	**0.8047**

Table 1: Performance in terms of mAP scores on five datasets with 30% supervision.

Method	mAP				Precision			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
SSH-orth	0.7383	0.7346	0.7268	0.7134	0.6725	0.6545	0.6469	0.6334
SSH-nonorth	0.8006	0.7611	0.7245	0.7041	0.7283	0.6791	0.6534	0.6418
SPLH	0.7786	0.6248	0.5827	0.5796	0.6963	0.5850	0.5633	0.5614
KRSHSC	0.7382	0.7650	0.7728	0.8078	0.7141	0.7291	0.7324	0.7554
SMH	0.8039	0.8716	0.7344	0.8685	0.7139	0.7744	0.6760	0.8129
MLAGH	0.8926	0.8833	0.8954	0.8996	0.7854	0.7836	0.8024	0.8198
GDSH	**0.9086**	**0.9242**	**0.9287**	**0.9359**	**0.8484**	**0.8628**	**0.8695**	**0.8808**

Table 2: mAP comparison with semi-supervised baselines on MIRFlickr using CLIP features.

Module	Variant	CALTECH		CIFAR-10		ImageNet		MS-COCO		NUS-WIDE	
		48b	64b	48b	64b	32b	64b	24b	32b	24b	32b
RLC	BaseLC	49.94	51.75	63.40	63.90	23.70	31.51	83.47	84.29	75.82	76.10
	RLC-reg	50.49	52.12	65.24	65.86	27.57	34.69	85.13	86.21	78.03	78.33
	RLC-sparse	50.56	51.39	64.88	65.39	28.00	34.48	85.45	85.73	77.65	79.05
DHL	BaseHL	49.91	51.05	64.02	64.00	27.21	33.28	84.29	84.76	75.33	76.71
	DHL-S	50.52	51.43	64.09	64.52	28.62	34.07	84.54	85.74	75.48	77.34
	DHL-E	50.26	51.30	64.09	65.24	27.29	33.85	84.49	84.85	76.37	77.04
GDSH-relax		47.57	48.88	63.65	64.44	28.33	34.57	85.57	85.88	77.81	79.08
GDSH		**50.81**	**52.13**	**65.38**	**65.90**	**28.69**	**34.94**	**86.06**	**86.45**	**78.82**	**79.47**

Table 3: Ablation study results (%) of the proposed RLC and DHL modules.

including three single-label datasets, CALTECH-101(Fei-Fei, Fergus, and Perona 2007), CIFAR-10(Krizhevsky and Hinton 2009), ImageNet, and three multi-label datasets, MS-COCO(Lin et al. 2014), NUS-WIDE(Chua et al. 2009), MIRFlickr(Huiskes and Lew 2008).

Baselines and Implementation Details

The proposed GDSH is compared with 18 state-of-the-art hashing approaches, in which SH (Weiss, Torralba, and Fergus 2008), PCA-ITQ (Gong et al. 2013), PCA-RR (Gong et al. 2013), MFH (Ding, Guo, and Zhou 2014) are unsupervised methods, SDH (Shen et al. 2015), NSH (Liu and Lu 2016), FSDH (Gui et al. 2018), SDHMLR (Liu et al. 2019), SCDH (Chen et al. 2020), LBSE (Liu et al. 2022), DSPH (Sun et al. 2024b) and REPH (Sun et al. 2024a) are supervised ones, and SSH-orth (Wang, Kumar, and Chang 2012), SSH-nonorth (Wang, Kumar, and Chang 2012), SPLH (Wang, Kumar, and Chang 2012), KRSHSC (Pan et al. 2015), SMH (Song et al. 2017), MLAGH (Hu et al. 2019) are semi-supervised baselines. The hyperparameters of all baselines are initialized according to the suggestion in the original publications. To make a fair comparison, we used kernelized features as the input of all the hashing methods on CALTECH-101, CIFAR-10, ImageNet, MS-COCO, and NUS-WIDE. In addition, the labeled subsets for all six benchmarks were kept the same throughout the experiments. Note that supervised approaches cannot directly handle the unlabeled samples, so we trained them only with labeled data.

For comparison with baselines, we empirically set $\beta = 0.3$, $\mu = 10^5$, $\theta = 10^5$, $\rho = 10^4$, $\alpha = 10^{-6}$. The best choice of γ is equally set to 100 on CALTECH-101, CIFAR-10, MS-COCO, NUS-WIDE, MIRFlickr, and 10^4 on ImageNet. $k = 1$ for single-label datasets, while $k = 2$ for multi-label datasets. The iteration numbers t and T are respectively set to 10 and 4. To maximize the generalization capability of the proposed RLC model, we finely tuned δ_1 and δ_2 via grid search, and use $\delta_1 = 1$ for all datasets. δ_2 is set to 10^{-6} for single-label datasets, and 10^{-3} for multi-label datasets. All the experiments were conducted on a computer with an Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz, 64GB RAM and a 64-bit Windows operating system. To eliminate randomness caused by initialization, the average performance over five runs is reported.

Comparison Results

Table 1 summarizes the results of the proposed method and baselines under semi-supervised setting with 30% supervision on the five benchmarks. It can be observed that the proposed GDSH consistently yields the highest mAP scores across different datasets with various lengths of hash codes. On the challenging ImageNet dataset, GDSH surpasses the best semi-supervised baseline by over 1.4%, 0.4%, 4.5% and 6.6% with 16-bit, 32-bit, 48-bit and 64-bit hash codes, respectively. Additionally, on the multi-label NUS-WIDE dataset, the mAP improvement is over 3.3% with 48-bit GDSH. These results demonstrate its superiority in semi-supervised image retrieval. Note that LBSE (Liu et al. 2022) is also learned under the supervision of semantic labels and the pairwise similarities. The performance advantage of GDSH over LBSE mostly consists in the additional debiasing designs on supervision, *i.e.*, label denoising and similarity updating. Longer hash codes lead to better accuracy in most cases, but not in all cases. This is because that some baselines (such as SSH-nonorth and SPLH) are based on eigen-decomposition of a modified covariance matrix (Wang, Kumar, and Chang 2012), thus their performance may be degraded by redundant hash bits in the long code.

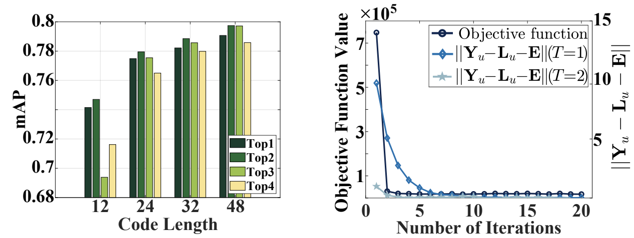
We also compared GDSH against other semi-supervised hashing approaches on multi-modal pre-trained features, with the label rate fixed to 30%. As can be seen from Table 2, GDSH obtains superior mAP and Precision with various lengths of hash codes.

Ablation Study

To evaluate the respective effectiveness of each key component of GDSH, ablation study was conducted in terms of the RLC, DHL strategies and the discrete optimization. The mAP results with 30% supervision are shown in Table 3. To be specific, several variants of the proposed GDSH are designed for contrast, where BaseLC removes the second and third terms in Eq. (2) at the same time. RLC-reg and RLC-sparse are obtained by removing the third term and the second term in Eq. (2), respectively. BaseHL represents GDSH without any debiasing, while DHL-S and DHL-E remove similarity updating and label denoising, respectively. The optimization process of GDSH-relax is the same as GDSH except for the solution of $\widetilde{\mathbf{B}}$. It first discards the discrete constraints in Eq. (22) to learn the continuous solution as $(\mathbf{A}\mathbf{Y}\mathbf{Y}^T\mathbf{A}^T + \gamma\mathbf{I})^{-1}(\mathbf{A}\mathbf{Y}\widetilde{\mathbf{S}}^T + \gamma\mathbf{A}\widetilde{\mathbf{Y}})$. Then, the discrete $\widetilde{\mathbf{B}}$ is obtained via thresholding. As can be found in Table 3, the added robust losses in the proposed RLC method can both help the basic model achieve a better performance. Furthermore, both DHL-S and DHL-E achieve higher mAP than BaseHL, while GDSH obtains the highest scores. These results verify the effectiveness of the two proposed levels of debiasing. GDSH outperforms GDSH-relax with a large margin, showing that the discrete optimization effectively reduces the quantization error and improves the accuracy. Based on above observations, it can be concluded that each robust and debiasing design in the proposed method contributes to the performance improvement.

Parameter Sensitivity Analysis

In order to choose an optimal number of pseudo labels for multi-label datasets, we investigated the impact of parameter k on NUS-WIDE under 30% label rate, as shown in Fig. 2 (a). It can be summarized from Table 1 and Fig. 2 (a) that assigning a single pseudo label can already produce a competitive performance, implying that the proposed label completion method is valid in discovering latent semantics.



(a) Based on NUS-WIDE (b) Based on CIFAR-10

Figure 2: (a) MAP variations with number of pseudo labels ranging from 1 to 4. (b) Convergence curves of the objective function with 30% supervision and 32-bit hash codes.

Convergence Analysis

Fig. 2 (b) plots the convergence curves of GDSH on CIFAR-10. It can observed that the value of the objective function Eq. (9) stably decreases with the increase in the number of iterations, and eventually converges within 10 iterations. As for the inner iteration, the objective value can reach the optimal 0 within the first two global iterations on different datasets. Therefore, the proposed optimization scheme is efficient for large-scale image retrieval.

Conclusion

In this paper, we propose a novel GDSH method for large-scale image retrieval, which provides a unified framework to reduce the biases in pseudo label-based semi-supervised hashing. On one hand, the bias caused by potentially incorrect pseudo labels is alleviated via two modules: robust label completion and debiased hash learning. The former improves the reliability of pseudo labels, while the latter simultaneously refines the supervision and learns semantically debiased hash codes. On the other hand, the bias arising from optimization is addressed through discrete hash code learning. We theoretically prove the stability of the proposed algorithm for learning projections between labels and hash codes. Besides, we experimentally demonstrate the superiority of GDSH and the effectiveness of its debiasing designs under different semi-supervised settings. In future, we will further speed up the proposed algorithm, and investigate self-adaptive techniques for determining the optimal number of pseudo labels.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (U23A20389, 62206160, 62176141, 62176139), Shandong Provincial Natural Science Foundation (ZR2022QF082), Young Talent of Lifting Engineering for Science and Technology in Shandong , China (SDAST2024QTA020), Major Basic Research Project of Shandong Provincial Natural Science Foundation (ZR2024ZD03), the Science and Technology Innovation Program for Distinguished Young Scholars of Shandong Province Higher Education Institutions (2024KJH084), Taishan Scholar Project of Shandong Province (tsqn202103088), Shandong Provincial Natural Science Foundation for Distinguished Young Scholars (ZR2021JQ26), Major science and technology innovation project of Shandong Province (2021CXGC011204), and special funds for distinguished professors of Shandong Jianzhu University.

References

Beck, A.; and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1): 183–202.

Chen, B.; Jiang, J.; Wang, X.; Wan, P.; Wang, J.; and Long, M. 2022. Debiased Self-Training for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 35.

Chen, Y.; Tian, Z.; Zhang, H.; Wang, J.; and Zhang, D. 2020. Strongly Constrained Discrete Hashing. *IEEE Transactions on Image Processing*, 29: 3596–3611.

Cheng, S.; Zhou, Y.; Zhang, W.; Wu, D.; Yang, C.; Li, B.; and Wang, W. 2022. Uncertainty-Aware and Multigranularity Consistent Constrained Model for Semi-Supervised Hashing. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10): 6914–6926.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proceedings of the ACM international conference on image and video retrieval*, 48. ACM.

Cui, H.; Zhao, L.; Li, F.; Zhu, L.; Han, X.; and Li, J. 2024. Effective Comparative Prototype Hashing for Unsupervised Domain Adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8329–8337.

Ding, G.; Guo, Y.; and Zhou, J. 2014. Collective Matrix Factorization Hashing for Multimodal Data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2083–2090.

Fan, W.; Zhang, C.; Li, H.; Jia, X.; and Wang, G. 2023. Three-Stage Semisupervised Cross-Modal Hashing With Pairwise Relations Exploitation. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1): 59–70.

Gabay, D.; and Mercier, B. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1): 17–40.

Golub, G. H.; and Loan, C. F. V. 1996. *Matrix Computations, Third Edition*. Johns Hopkins University Press. ISBN 978-0-8018-5414-9.

Gong, Y.; Lazebnik, S.; Gordo, A.; and Perronnin, F. 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12): 2916–2929.

Gui, J.; Liu, T.; Sun, Z.; Tao, D.; and Tan, T. 2018. Fast supervised discrete hashing. *IEEE transactions on pattern analysis and machine intelligence*, 40(2): 490–496.

Hu, H.; Wang, K.; Lv, C.; Wu, J.; and Yang, Z. 2019. Semi-Supervised Metric Learning-Based Anchor Graph Hashing for Large-Scale Image Retrieval. *IEEE Transactions on Image Processing*, 28(2): 739–754.

Huiskes, M. J.; and Lew, M. S. 2008. The MIR flickr retrieval evaluation. In Lew, M. S.; Bimbo, A. D.; and Bakker, E. M., eds., *Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2008, Vancouver, British Columbia, Canada, October 30-31, 2008*, 39–43. ACM.

Kage, P.; Rothenberger, J. C.; Andreadis, P.; and Diochnos, D. I. 2024. A Review of Pseudo-Labeling for Computer Vision. arXiv:2408.07221.

Kan, M.; Xu, D.; Shan, S.; and Chen, X. 2014. Semisupervised Hashing via Kernel Hyperplane Learning for Scalable Image Search. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(4): 704–713.

Kang, W.-C.; Li, W.-J.; and Zhou, Z.-H. 2016. Column Sampling Based Discrete Supervised Hashing. In *AAAI*, 1230–1236.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, Q.; and Lu, H. 2016. Natural Supervised Hashing. In *International Joint Conferences on Artificial Intelligence*, 1788–1794.

Liu, X.; Kang, X.; Nie, X.; Guo, J.; Wang, S.; and Yin, Y. 2022. Learning Binary Semantic Embedding for Large-Scale Breast Histology Image Analysis. *IEEE Journal of Biomedical and Health Informatics*, 26(7): 3240–3250.

Liu, X.; Li, J.; Nie, X.; Zhang, X.; and Yin, Y. 2024. Fast Unsupervised Cross-Modal Hashing with Robust Factorization and Dual Projection. *ACM Trans. Multimedia Comput. Commun. Appl.*, 20(12).

Liu, X.; Nie, X.; Zhou, Q.; and Yin, Y. 2019. Supervised Discrete Hashing With Mutual Linear Regression. In *ACM Multimedia*, 1561–1568.

Mandal, D.; Chaudhury, K. N.; and Biswas, S. 2019. Generalized Semantic Preserving Hashing for Cross-Modal Retrieval. *IEEE Transactions on Image Processing*, 28(1): 102–112.

Markovsky, I.; and Van Huffel, S. 2007. Overview of total least-squares methods. *Signal Processing*, 87(10): 2283 – 2302.

Nie, F.; Huang, H.; Cai, X.; and Ding, C. 2010. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *24th Annual Conference on Neural Information Processing Systems*, 1813–1821. Vancouver, BC, Canada.

Pan, Y.; Yao, T.; Li, H.; Ngo, C.-W.; and Mei, T. 2015. Semi-supervised hashing with semantic confidence for large scale visual search. In *38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 53–62.

Shen, F.; Shen, C.; Liu, W.; and Tao Shen, H. 2015. Supervised discrete hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 37–45.

Shen, X.; Song, P.; Yuan, Y.-H.; and Zheng, Y. 2024. Distributed Manifold Hashing for Image Set Classification and Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4802–4810.

Shi, W.; Gong, Y.; Chen, B.; and Hei, X. 2021. Transductive semisupervised deep hashing. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8): 3713–3726.

Song, T.; Cai, J.; Zhang, T.; Gao, C.; Meng, F.; and Wu, Q. 2017. Semi-supervised manifold-embedded hashing with joint feature representation and classifier learning. *Pattern Recognition*, 68: 99–110.

Sun, Y.; Dai, J.; Ren, Z.; Li, Q.; and Peng, D. 2024a. Relaxed Energy Preserving Hashing for Image Retrieval. *IEEE Transactions on Intelligent Transportation Systems*.

Sun, Y.; Qin, Y.; Peng, D.; Ren, Z.; Yang, C.; and Hu, P. 2024b. Dual Self-Paced Hashing for Image Retrieval. *IEEE Transactions on Multimedia*.

Tang, L.; Ye, D.; Lv, Y.; Chen, C.; and Zhang, Y. 2024. Once and for All: Universal Transferable Adversarial Perturbation against Deep Hashing-Based Facial Image Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5136–5144.

Wang, G.; Hu, Q.; Yang, Y.; Cheng, J.; and Hou, Z.-G. 2022a. Adversarial Binary Mutual Learning for Semi-Supervised Deep Hashing. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8): 4110–4124.

Wang, H.; Jiang, H.; Sun, J.; Zhang, S.; Chen, C.; Hua, X.-S.; and Luo, X. 2023a. Dior: Learning to hash with label noise via dual partition and contrastive learning. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, J.; Kumar, S.; and Chang, S.-F. 2012. Semi-supervised hashing for large-scale search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12): 2393–2406.

Wang, X.; Wu, Z.; Lian, L.; and Yu, S. X. 2022b. Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14647–14657.

Wang, Y.; Wang, J.; Chen, B.; Zeng, Z.; and Xia, S.-T. 2023b. Contrastive masked autoencoders for self-supervised video hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2733–2741.

Weiss, Y.; Torralba, A.; and Fergus, R. 2008. Spectral hashing. In *International Conference on Neural Information Processing Systems*, 1753–1760.

Wu, C.; Zhu, J.; Cai, D.; Chen, C.; and Bu, J. 2013. Semi-supervised nonlinear hashing using bootstrap sequential projection learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(6): 1380–1393.

Wu, D.; Su, Q.; Li, B.; and Wang, W. 2024. Pairwise-Label-Based Deep Incremental Hashing with Simultaneous Code Expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9169–9177.

Xia, X.; Deng, J.; Bao, W.; Du, Y.; Han, B.; Shan, S.; and Liu, T. 2023. Holistic label correction for noisy multi-label classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1483–1493.

Zhang, J.; and Peng, Y. 2019. SSDH: Semi-Supervised Deep Hashing for Large Scale Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(1): 212–225.

Zhang, X.; Liu, X.; Nie, X.; Kang, X.; and Yin, Y. 2024. Semi-Supervised Semi-Paired Cross-Modal Hashing. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7): 6517–6529.

Zhu, L.; Lu, X.; Cheng, Z.; Li, J.; and Zhang, H. 2020. Deep Collaborative Multi-View Hashing for Large-Scale Image Search. *IEEE Trans. Image Process.*, 29: 4643–4655.

Zhu, L.; Zheng, C.; Guan, W.; Li, J.; Yang, Y.; and Shen, H. T. 2023. Multi-modal Hashing for Efficient Multimedia Retrieval: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 1–20.