S3ACH: Semi-Supervised Semantic Adaptive Cross-Modal Hashing

Liu Yang, Kaiting Zhang, Yinan Li, Yunfei Chen, Jun Long and Zhan Yang

October 23, 2023

S3ACH: Semi-Supervised Semantic Adaptive Cross-modal Hashing

Liu Yang[1], Kaiting Zhang[1], Yinan Li[2], Yunfei Chen[2], Jun Long[2], and Zhan Yang[2]⋆

[1] School of computer science and engineering, Central South University, Changsha, Hunan, China {yangliu,software-zkt}@csu.edu.cn
[2] Big data institute, Central South University, Changsha, Hunan, China {liyinan,yunfeichen,junlong,zyang22}@csu.edu.cn

Abstract. Hash learning has been a great success in large-scale data retrieval field because of its superior retrieval efficiency and storage consumption. However, labels for large-scale data are difficult to obtain, thus supervised learning-based hashing methods are no longer applicable. In this paper, we introduce a method called **S**emi-**S**upervised **S**emantic **A**daptive **C**ross-modal **H**ashing (S3ACH), which improves performance of unsupervised hash retrieval by exploiting a small amount of available label information. Specifically, we first propose a higher-order dynamic weight public space collaborative computing method, which balances the contribution of different modalities in the common potential space by invoking adaptive higher-order dynamic variable. Then, less available label information is utilized to enhance the semantics of hash codes. Finally, we propose a discrete optimization strategy to solve the quantization error brought by the relaxation strategy and improve the accuracy of hash code production. The results show that S3ACH achieves better effects than current advanced unsupervised methods and provides more applicable while balancing performance compared with the existing cross-modal hashing.

Keywords: Hashing · Cross-modal retrieval · Semi-Supervised.

1 Introduction

With the rapid development of Big Data, plenty of multimodal data have been produced, such as text, images, audio, etc. Many practiced scenarios require multimodal data processing, and the cross-modal retrieval techniques that have turned a hot research theme, which rely on semantic similarity calculation between data. Hash-based methods become a practical solution to deal with massive heterogeneous data, aiming to reduce data dimensions to a binary code while retaining the original semantic information. This can reduce time and memory overhead. The basic principle of hash methods is that multimodal data be projected into a uniform low-dimensional Hamming space, so that achieve efficient similarity search with Hamming distance.

⋆ Corresponding author.

Traditional hash methods led by Local Sensitive Hash series [21] are data-independent. They generate hash codes by random projection without considering data distribution, so it is difficult to keep high accuracy with short coding at the same time. As machine learning techniques develop rapidly, data-dependent methods come into mainstream, while multimodal hashing gradually become the most promising research direction, and many cross-modal hash methods have been offered [1, 31]. The key to cross-modal hashing is retaining similarity of hash code in multimodal. These methods [19, 24] exploit semantic information for supervised learning and show very good performance, but ignore the high time and labor cost of acquiring labels. In contrast, unsupervised cross-modal hashing [7, 16, 13] is able to find modalities relationships without label information. Despite the significant progress of these methods, they still have the limitation of lacking label supervision. The reality is that only a very tiny part of data is labeled, so semi-supervised methods using few labels is the most realistic solution. However, semi-supervised hash retrieval methods currently compensate for the lack of supervised capacity by deep learning [30], which is not only expensive but also difficult to reuse and interpret.

To address the above limitations, we propose Semi-supervised Semantic Adaptive Cross-modal Hashing (S3ACH). Firstly, we design a public potential space learning framework with higher-order dynamic weights to collaborative computing the contributions of different modalities. To fully utilize semantic information from sparse labels, we design an adaptive label enhancement module to enhance representation learning. Meanwhile, S3ACH obtains a potentially consistent representation of different modalities based on a matrix decomposition strategy with dynamic weights, which ensures the semantic completeness among modalities. Moreover, an efficient iterative optimization algorithm is offered for discrete constraints with direct one-step hash coding. Summarily, the core contributions of this work include:

1. A higher-order dynamic variable collaborative computing method is proposed to adaptively balance the different modalities' contribution, so as to improve the learning stability of common potential representation.
2. A label enhancement framework is proposed to directly exploit the labeled data to mine semantic information and maximize similarity differences between modalities.
3. A fast iterative optimization method for solving discrete constrained problems is proposed, where the time consumption of the method scales linearly related to data size.
4. Extensive experiments are conducted on the MIRFlickr and NUS-WIDE datasets, and our proposed S3ACH shows better retrieval performance and higher applicability in real-world scenarios with large-scale data compared to state-of-the-art methods.

The rest of the paper includes the following. Reviews the work related to cross-mode hash retrieval in section 2, and section 3 details our proposed S3ACH. In section 4, we conduct comparative and analytical experiments. Finally, the conclusion drawn in section 5.

2 Related Works

Cross-mode hashing mainly consists of (un)supervised methods. Unsupervised cross-modal hashing methods generate hash codes from data distribution, rather than semantic labeling information, and focus on inter-modal and intra-modal correlations. They can be classified into shallow and deep methods. In some early studies, CMFH [6] handles different modalities by collective matrix decomposition method, and LSSH [32] preserves specific properties by constructing inter-modal sparse representations. In recent years, the similarity in the common subspace is optimized from different perspectives. For example, CUH [18] uses a novel optimization strategy for multi-modal clustering and hash learning. RUCMH [2] preserves both the deterministic continuous shared space and discrete hamming space. JIMFH [17] retains both the shared properties and the properties specific to each modality. FUCMSH [25] ensures inter-modal consistent representation and intra-modal specific potential representation by shared matrix decomposition and individual self-coding, respectively.

In addition, the latest research in deep cross-modal hashing [14, 20] has showed superior performance thanks to the strong nonlinear representation of deep learning. But the massive resource consumption also becomes an obvious drawback. Therefore, one of the focuses of this paper is to design an interpretable objective function and an effective discrete optimization method based on shallow method.

Supervised cross-modal hashing, in contrast, enhances the correlation between modalities by supervising the learning process using semantic labeling information. For example, SRLCH [12] transforms class labels in the Hamming subspace into relational information. LEADH [23] designs a label-binary mutual mapping architecture to fully exploit and utilize multi-label semantic information, and SPECH [22] uses a likelihood loss technique to measure the semantic similarity of paired data. However, artificial semantic annotation is costly with massive data, and the tiny percentage of labeled data lead to the unavailability of supervised methods in real-world situations. We consider this problem and enhance the process by a label learning framework that utilizes as few labels as possible, so as to improve the semi-supervised learning.

Research in semi-supervised cross-modal hashing, for example, SCH-GAN [30] fits the correlation distribution of unlabeled data by adversarial networks, and UMCSH [3] uses uncertainty estimation methods to select label information which gives discriminative features to unlabeled data, but they are both deep methods. Recent research in shallow methods such as FlexCmh [27] allows learning hash codes from weakly paired data, and WASH [28] uses weakly supervised enhanced learning by regularizing the noise label matrix. In this paper, we propose shallow semi-supervised cross-modal hashing to satisfies real-world scenarios concisely and effectively.

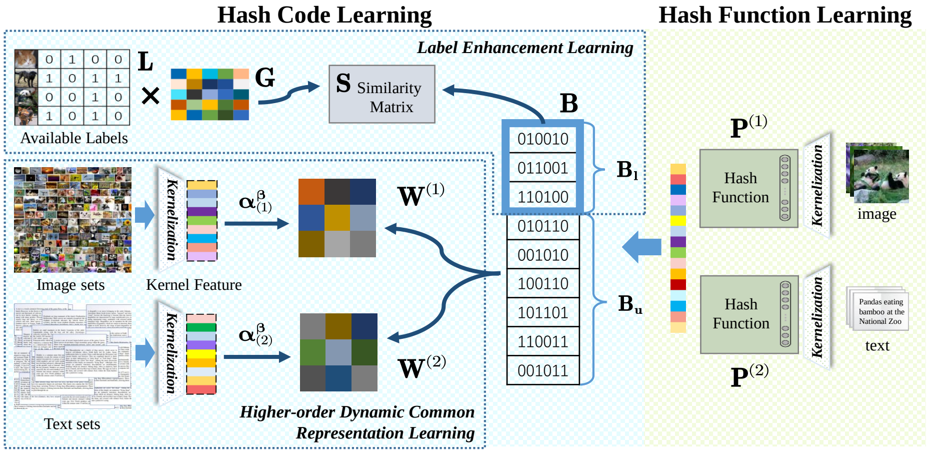


Fig. 1. The framework of S3ACH.

3 The Proposed Method

3.1 Notation and Problem Formulation

Support $\mathbb{O} = \{\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, ..., \mathbf{x}_n^{(v)}\} \in \mathbb{R}^{d^{(v)} \times n}$ denote the cross-modal dataset, n is the total instances amount, $d^{(v)}$ refers to the v-th modality. Specifically, $\mathbf{X}^{(v)}$ is decomposed into two parts: the labeled instances $\mathbf{X}_l^{(v)} \in \mathbb{R}^{d^{(v)} \times n_l}$ and the unlabeled instances $\mathbf{X}_u^{(v)} \in \mathbb{R}^{d^{(v)} \times n_u}$, where n_l and n_u are the numbers of labeled and unlabeled datasets, respectively. $\mathbf{B} \in \mathbb{R}^{k \times n}$ is the binary codes, k is the hash codes length. The aim of this paper is to learn a hash function $\mathcal{H} : \mathbf{x} \rightarrow \mathbf{b} \in \{-1, 1\}^k$, where $\mathbf{x}$ is the input instance. The framework of this paper is shown in Fig. 1, which consists of two main processes: hash code learning process include Higher-order Dynamic Common Representation and Label Enhancement Learning, and hash function learning process rely on the generated hash code. All two processes will be kernelized initially.

In this paper, bold lowercase and uppercase letters denote vectors and matrices, respectively, and ordinary lowercase denote Scalars. $tr(\mathbf{U})$ represents the trace, and $\mathbf{U}^\top$ represents the transposition of matrix $\mathbf{U}$.

3.2 S3ACH method

Kernelization Kernelization is one of the frequently used nonlinear relationship modeling techniques in machine learning domain. Thus, for more accurate represent the potentially complex relationships between different modalities, we use the RBF-based kernel technique to process raw features of different modalities. For a instance $x_i^{(v)}$, the kernelized features $\phi(x_i^{(v)})$ can be expressed as,

$$\phi(x_i^{(v)}) = \left[\exp(\frac{-||x_i^{(v)} - a_1^{(v)}||_2^2}{2\sigma^2}), ..., \exp(\frac{-||x_i^{(v)} - a_q^{(v)}||_2^2}{2\sigma^2})\right]^\top \tag{1}$$

where $a_q^{(v)}$ represents the randomly selected q anchor instances of the v-th modality and σ is the width.

Higher-order Dynamic Common Representation Learning Since the "heterogeneous gap" between different modalities, it is not possible to fuse them directly, and it is necessary to find a common subspace (i.e., the learned hash codes) $\mathbf{B}$ to bridge two modals. As shown in Fig. 1, we formulate the following objective function to learn the binary codes,

$$\begin{aligned} &\min_{\mathbf{W}^{(v)},\mathbf{B}} || \mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^2 + \mathcal{R}(\mathbf{W}^{(v)}), \\ &s.t.\ \mathbf{B} = [\mathbf{B}_l; \mathbf{B}_u] \in \{-1, 1\}^{k\times n}, \end{aligned} \tag{2}$$

where $\mathbf{W}^{(v)} \in \mathbb{R}^{q\times k}$ is modality-specific mapping matrix for the v-th modality and $\mathcal{R}(\cdot) = \delta||\cdot||_F^2$ is the regularization term. It is noteworthy that there exist an opposite learning manner [11], i.e., $\ell(\phi(\mathbf{X}^{(v)})\mathbf{W}^{(v)}, \mathbf{B})$, which encoding different modalities information into a common latent representation. But this means that each modality can be individually encode common latent representation, which weaken the completeness of the information substantially.

However, the above approach treats the contributions of all modalities to the common latent representation as the same. In fact, the contributions of different modalities to the common representation should be different, thus we introduce a self-learning dynamic weight parameter α_v to represent the contribution of v-th modality. Therefore, Eq. (2) can be rewritten as,

$$\begin{aligned} &\min_{\alpha_v,\mathbf{W}^{(v)},\mathbf{B}} \alpha_v||\mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^2 + \mathcal{R}(\mathbf{W}^{(v)}), \\ &s.t.\ \sum_v^V \alpha_v = 1, \alpha_v \geq 0,\ \mathbf{B} = [\mathbf{B}_l; \mathbf{B}_u] \in \{-1, 1\}^{k\times n} \end{aligned} \tag{3}$$

Although the solution of Eq. (3) can fix the problem of unbalance contribution, there are still an issue need to fix, i.e., if the feature values of a modality $\mathbf{X}^{(v)}$ are sparse, then it will make the corresponding weight parameter take the maximum value, i.e., $\alpha_v = 1$. In other word, other modalities will be ignored directly. Therefore, we introduce a smooth exponential factor $\beta > 0$ to avoid the problem, that is,

$$\begin{aligned} &\min_{\alpha_v,\mathbf{W}^{(v)},\mathbf{B}} \alpha_v^\beta||\mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^2 + \mathcal{R}(\mathbf{W}^{(v)}), \\ &s.t.\ \sum_v^V \alpha_v = 1, \alpha_v \geq 0, \beta > 0,\ \mathbf{B} = [\mathbf{B}_l; \mathbf{B}_u] \in \{-1, 1\}^{k\times n} \end{aligned} \tag{4}$$

where $\mathbf{X}^{(v)}$ is composed of labeled instances $\mathbf{X}_l^{(v)}$ and unlabeled instances $\mathbf{X}_u^{(v)}$.

Label Enhancement Learning Since some labeled data exists in real scenarios, it is important to make full use of labeled data to improve a recognition of the learned hash codes. As shown in Fig. 1, we construct a pairwise similar matrix $\mathbf{S} \in \mathbb{R}^{n_l \times n_l}$, i.e., $\mathbf{S} = 2\mathbf{L}^\top \mathbf{L} - \mathbf{1}_{n_l}\mathbf{1}_{n_l}^\top$, where $\mathbf{L} \in \mathbb{R}^{c \times n_l}$ represents label matrix and $\mathbf{1}_{n_l}$ denotes an all-one column vector with length n_l. Note that, the implementation of this approach can solve the problem of high time consumption caused by the direct use of pairwise similarity matrices. Then we build a bridge to link the semantic information and the corresponding hash codes $\mathbf{B}_l$. Inspired by a popular symmetric framework KSH [10] method, which the definition is $\min\limits_{\mathbf{B}} ||\mathbf{B}^\top \mathbf{B} - k\mathbf{S}||^2,\ s.t.\ \mathbf{B} \in \{-1, 1\}^{k \times n}$. This strategy, however, is a difficult quadratic optimization to solve. Luckily, a few researchers [5, 26] propose asymmetric learning frameworks to address this problem in terms of accuracy and speed. Therefore, we design an asymmetric learning framework as follows:

$$\begin{aligned} &\min_{\mathbf{B}_l, \mathbf{G}} \gamma||k\mathbf{S} - {\mathbf{B}_l}^\top (\mathbf{G}\mathbf{L})||_F^2 + \rho||\mathbf{B}_l - \mathbf{G}\mathbf{L}||_F^2, \\ &s.t.\ \mathbf{B}_l \in \{-1, 1\}^{k \times n_l}, \end{aligned} \tag{5}$$

where γ, ρ are the parameters to balance the asymmetric learning framework, and $\mathbf{G} \in \mathbb{R}^{k \times c}$ is a transformation matrix.

Joint Hash Learning Framework Combing Eq. (4) and Eq. (5), we obtain the overall objective function of S3ACH as follows.

$$\begin{aligned} &\min_{\alpha_v, \mathbf{W}^{(v)}, \mathbf{G}, \mathbf{B}_u, \mathbf{B}_l} \sum_v^V \alpha_v^\beta ||\mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^2 + \mathcal{R}(\mathbf{W}^{(v)}, \mathbf{G}\mathbf{L}) \\ &+ \gamma||k\mathbf{S} - {\mathbf{B}_l}^\top (\mathbf{G}\mathbf{L})||_F^2 + \rho||\mathbf{B}_l - \mathbf{G}\mathbf{L}||_F^2, \\ &s.t.\ \sum_v^V \alpha_v = 1, \alpha_v \geq 0, \beta > 0, \mathbf{B} = [\mathbf{B}_l; \mathbf{B}_u] \in \{-1, 1\}^{k \times n}. \end{aligned} \tag{6}$$

3.3 Optimization

Eq. (6) is an NP-hard issue due to the multivariate and discrete constraints. Therefore, some strategies involve initially approximating the discrete variables using the *sgn* function. Such an approach can cause huge quantization errors and affect the quality of hash code generation. Some strategies use the DCC (Discrete Cyclic Coordinate Descent) strategy to optimize each hash bit in hash code by circular iteration. Although this approach does not cause the problem of quantization loss, the time consumption of optimization is proportional to the hash code length and the solution is inefficient. This paper uses discrete optimization methods to learn complete hash codes in one step, addressing issues with the mentioned strategies. Specifically, we solve for the other variables by fixing one of them. The overall optimization process of Eq. (6) is as follows.

α_v-step We fix $\mathbf{G}, \mathbf{B}_l, \mathbf{B}_u, \mathbf{W}^{(v)}$ variables, the updating for variable α_v can be reformulated as,

$$\begin{aligned} &\min_{\alpha_v} \alpha_v^\beta ||\mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^2, \\ &s.t. \sum_v^V \alpha_v = 1, \alpha_v \geq 0. \end{aligned} \tag{7}$$

The Lagrange multiple scheme is used to construct the Lagrange arithmetic formulation, that is,

$$\min_{\alpha_v} \alpha_v^\beta ||\mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^2 - \nu(\mathbf{1}^\top \alpha - 1), \tag{8}$$

where $\alpha = [\alpha_1, \alpha_2, ..., \alpha_V]^\top \in \mathbb{R}^V$ is the vector of weights for the related modalities, and ν is the Lagrange arithmetic.

Setting the derivative with respect to α_v and ν to 0, we get,

$$\alpha_v = \frac{||\mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^{2\ 1/1-\beta}}{\sum_v^V (||\mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^{2\ 1/1-\beta})}. \tag{9}$$

$\mathbf{W}^{(v)}$-step We fix $\mathbf{G}, \mathbf{B}_l, \mathbf{B}_u, \alpha_v$ variables, the updating for variable $\mathbf{W}^{(v)}$ can be reformulated as,

$$\min_{\mathbf{W}^{(v)}} \alpha_v^\beta ||\mathbf{W}^{(v)}\mathbf{B} - \phi(\mathbf{X}^{(v)})||_F^2 + \delta||\mathbf{W}^{(v)}||_F^2. \tag{10}$$

Then Eq. (10) can be simplified as,

$$\begin{aligned} &\min_{\mathbf{W}^{(v)}} \alpha_v^\beta tr(\mathbf{W}^{(v)}\mathbf{B}\mathbf{B}^\top {\mathbf{W}^{(v)}}^\top - 2\phi(\mathbf{X}^{(v)})\mathbf{B}^\top {\mathbf{W}^{(v)}}^\top) \\ &+ \delta tr(\mathbf{W}^{(v)}{\mathbf{W}^{(v)}}^\top). \end{aligned} \tag{11}$$

Setting the derivative with respect to $\mathbf{W}^{(v)}$ to 0, we get,

$$\mathbf{W}^{(v)} = \alpha_v^\beta \phi(\mathbf{X}^{(v)})\mathbf{B}^\top(\alpha_v^\beta \mathbf{B}\mathbf{B}^\top + \delta\mathbf{I})^{-1}. \tag{12}$$

G-step We fix $\mathbf{W}^{(v)}, \mathbf{B}_l, \mathbf{B}_u, \alpha_v$ variables, the updating for variable $\mathbf{G}$ can be reformulated as,

$$\min_{\mathbf{G}} \delta||\mathbf{G}\mathbf{L}||_F^2 + \gamma||k\mathbf{S} - {\mathbf{B}_l}^\top(\mathbf{G}\mathbf{L})||_F^2 + \rho||\mathbf{B}_l - \mathbf{G}\mathbf{L}||_F^2, \tag{13}$$

Then Eq. (13) can be simplified as,

$$\begin{aligned} &\min_{\mathbf{G}} \delta\ tr(\mathbf{G}\mathbf{L}\mathbf{L}^\top\mathbf{G}^\top) + \gamma\ tr(-2k\mathbf{S}\mathbf{L}^\top\mathbf{G}^\top\mathbf{B}_l + \mathbf{B}_l^\top\mathbf{G}\mathbf{L}\mathbf{L}^\top\mathbf{G}^\top\mathbf{B}_l) \\ &+ \rho\ tr(-2\mathbf{B}_l\mathbf{L}^\top\mathbf{G}^\top + \mathbf{G}\mathbf{L}\mathbf{L}^\top\mathbf{G}^\top). \end{aligned} \tag{14}$$

Setting the derivative with respect to $\mathbf{G}$ to 0, we get,

$$\mathbf{G} = ((\delta + \rho)\mathbf{I} + \gamma\mathbf{B}_l\mathbf{B}_l^\top)^{-1}(\gamma k\mathbf{B}_l\mathbf{S}\mathbf{L}^\top + \rho\mathbf{B}_l\mathbf{L}^\top)(\mathbf{L}\mathbf{L}^\top)^{-1}. \tag{15}$$

$\mathbf{B}_{\boldsymbol{l}}$,$\mathbf{B}_{\boldsymbol{u}}$-step We fix $\mathbf{W}^{(v)}, \mathbf{G}, \mathbf{B}_u, \alpha_v$ variables, the updating for variable $\mathbf{B}_l$ can be reformulated as,

$$\begin{aligned}&\min_{\mathbf{B}_l} \sum_v^V \alpha_v^\beta ||\mathbf{W}^{(v)}\mathbf{B}_l - \phi(\mathbf{X}_l^{(v)})||_F^2 + \gamma||k\mathbf{S} - \mathbf{B}_l{}^\top(\mathbf{GL})||_F^2 + \rho||\mathbf{B}_l - \mathbf{GL}||_F^2,\\ &s.t.\ \mathbf{B}_l \in \{-1,1\}^{k\times n_l}.\end{aligned} \tag{16}$$

Eq. (16) can be reformulated as

$$\begin{aligned}&\min_{\mathbf{B}_l} \sum_v^V \alpha_v^\beta tr\ (\mathbf{W}^{(v)}\mathbf{B}_l\mathbf{B}_l{}^\top\mathbf{W}^{(v)}{}^\top - 2\phi(\mathbf{X}_l^{(v)})\mathbf{B}_l{}^\top\mathbf{W}^{(v)}{}^\top)\\ &+ \gamma tr\ (-2k\mathbf{B}_l{}^\top\mathbf{GLS}^\top + \mathbf{B}_l{}^\top\mathbf{GLL}^\top\mathbf{G}^\top\mathbf{B}_l) + \rho tr\ (-2\mathbf{GLB}_l{}^\top),\\ &s.t.\ \mathbf{B}_l \in \{-1,1\}^{k\times n_l}.\end{aligned} \tag{17}$$

The discrete constraints of the discrete variables to be solved make the above problem difficult to solve. Therefore, we use the ALM (Augmented Lagrange Multiplier method) to separate the discrete variables to be solved, i.e., we introduce an auxiliary discrete variable $\mathbf{K}_l$ to substitute the first $\mathbf{B}_l$ in $\mathbf{W}^{(v)}\mathbf{B}_l\mathbf{B}_l{}^\top\mathbf{W}^{(v)}{}^\top$ and $\mathbf{B}_l{}^\top\mathbf{GLL}^\top\mathbf{G}^\top\mathbf{B}_l$. We obtain,

$$\begin{aligned}&\min_{\mathbf{B}_l} \sum_v^V \alpha_v^\beta tr\ (\mathbf{W}^{(v)}\mathbf{K}_l\mathbf{B}_l{}^\top\mathbf{W}^{(v)}{}^\top - 2\phi(\mathbf{X}_l^{(v)})\mathbf{B}_l{}^\top\mathbf{W}^{(v)}{}^\top)\\ &+ \gamma tr\ (-2k\mathbf{B}_l{}^\top\mathbf{GLS}^\top + \mathbf{B}_l{}^\top\mathbf{GLL}^\top\mathbf{G}^\top\mathbf{K}_l) + \rho tr\ (-2\mathbf{GLB}_l{}^\top)\\ &+ \frac{\xi}{2}||\mathbf{B}_l - \mathbf{K}_l + \frac{\mathbf{H}_l}{\xi}||_F^2,\\ &s.t.\ \mathbf{B}_l \in \{-1,1\}^{k\times n_l},\end{aligned} \tag{18}$$

where $\mathbf{H}$ denotes the differences between $\mathbf{B}_l$ and $\mathbf{K}_l$, and the last term $\frac{\xi}{2}||\mathbf{B}_l - \mathbf{K}_l + \frac{\mathbf{H}_l}{\xi}||_F^2$ can be rewritten as

$$\min_{\mathbf{B}_l} tr\ (-\xi\mathbf{K}_l\mathbf{B}_l{}^\top + \mathbf{H}_l\mathbf{B}_l{}^\top). \tag{19}$$

Then we optimize the function for $\mathbf{B}_l$ rewritten as

$$\begin{aligned}&\max_{\mathbf{B}_l} tr\ (\sum_v^V(2\alpha_v^\beta\mathbf{W}^{(v)}{}^\top\phi(\mathbf{X}_l^{(v)}))\mathbf{B}_l{}^\top + 2k\gamma\mathbf{GLS}^\top\mathbf{B}_l{}^\top + 2\rho\mathbf{GLB}_l{}^\top\\ &+ \xi\mathbf{K}_l\mathbf{B}_l{}^\top - \alpha_v^\beta\mathbf{W}^{(v)}{}^\top\mathbf{W}^{(v)}\mathbf{K}_l\mathbf{B}_l{}^\top - \gamma\mathbf{GLL}^\top\mathbf{G}^\top\mathbf{K}_l\mathbf{B}_l{}^\top - \mathbf{H}_l\mathbf{B}_l{}^\top).\end{aligned} \tag{20}$$

The closed-solution of $\mathbf{B}_l$ as

$$\begin{aligned}\mathbf{B}_l =& sgn(2\sum_v^V \alpha_v^\beta\mathbf{W}^{(v)}{}^\top\phi(\mathbf{X}_l^{(v)}) + 2k\gamma\mathbf{GLS}^\top + 2\rho\mathbf{GL}\\ &+ \xi\mathbf{K}_l - \sum_v^V(\alpha_v^\beta\mathbf{W}^{(v)}{}^\top\mathbf{W}^{(v)})\mathbf{K}_l - \gamma\mathbf{GLL}^\top\mathbf{G}^\top\mathbf{K}_l - \mathbf{H}_l).\end{aligned} \tag{21}$$

Similarly, the variable $\mathbf{B}_u$ can be computed as

$$\mathbf{B}_u = sgn(2\sum_{v}^{V}\alpha_v^{\beta}\mathbf{W}^{(v)^{\top}}\phi(\mathbf{X}_u^{(v)}) + \xi\mathbf{K}_u - \sum_{v}^{V}(\alpha_v^{\beta}\mathbf{W}^{(v)^{\top}}\mathbf{W}^{(v)})\mathbf{K}_u - \mathbf{H}_u). \tag{22}$$

Finally, the learned hash codes $\mathbf{B}$ can be obtained by $\mathbf{B} = [\mathbf{B}_l; \mathbf{B}_u]$.

$\mathbf{K}_l, \mathbf{K}_u$-step We fix other variables, and the updating for variables $\mathbf{K}_l, \mathbf{K}_u$ can be reformulated as,

$$\begin{aligned}&\min_{\mathbf{K}_l} tr\ (\sum_{v}^{V}\alpha_v^{\beta}\mathbf{W}^{(v)}\mathbf{K}_l{\mathbf{B}_l}^{\top}\mathbf{W}^{(v)^{\top}} + \gamma{\mathbf{B}_l}^{\top}\mathbf{G}\mathbf{L}\mathbf{L}^{\top}\mathbf{G}^{\top}\mathbf{K}_l)\\ &+ \frac{\xi}{2}||\mathbf{B}_l - \mathbf{K}_l + \frac{\mathbf{H}_l}{\xi}||_F^2,\\ &s.t.\ \mathbf{K}_l \in \{-1, 1\}^{k\times n_l},\end{aligned} \tag{23}$$

Then we optimize the function for $\mathbf{K}_l$ rewritten as

$$\begin{aligned}&\min_{\mathbf{K}_l} tr\ ((\sum_{v}^{V}(\alpha_v^{\beta}\mathbf{W}^{(v)^{\top}}\mathbf{W}^{(v)})\mathbf{B}_l + \gamma\mathbf{G}\mathbf{L}\mathbf{L}^{\top}\mathbf{G}^{\top}\mathbf{B}_l - \xi\mathbf{B}_l - \mathbf{H}_l){\mathbf{K}_l}^{\top})\\ &s.t.\ \mathbf{K}_l \in \{-1, 1\}^{k\times n_l},\end{aligned} \tag{24}$$

Finally, the optimal solution of $\mathbf{K}_l$ can be obtained as

$$\mathbf{K}_l = sgn(-\sum_{v}^{V}(\alpha_v^{\beta}\mathbf{W}^{(v)^{\top}}\mathbf{W}^{(v)})\mathbf{B}_l - \gamma\mathbf{G}\mathbf{L}\mathbf{L}^{\top}\mathbf{G}^{\top}\mathbf{B}_l + \xi\mathbf{B}_l + \mathbf{H}_l). \tag{25}$$

Similarly, the variable $\mathbf{K}_u$ can be obtained as

$$\mathbf{K}_u = sgn(-\sum_{v}^{V}(\alpha_v^{\beta}\mathbf{W}^{(v)^{\top}}\mathbf{W}^{(v)})\mathbf{B}_u + \xi\mathbf{B}_u + \mathbf{H}_u). \tag{26}$$

$\mathbf{H}_l, \mathbf{H}_u$-step According to ALM scheme, the variables $\mathbf{H}_l, \mathbf{H}_u$ can be updated by,

$$\mathbf{H}_l = \mathbf{H}_l + \xi(\mathbf{B}_l - \mathbf{K}_l), \tag{27}$$

$$\mathbf{H}_u = \mathbf{H}_u + \xi(\mathbf{B}_u - \mathbf{K}_u). \tag{28}$$

3.4 Hash Function Learning

In hash function learning process, we need to use the hash codes learned in previous sections for hash function generation. The hash function can be a linear function, deep neural network, support vector machine and other models. Due to the consideration of training time, we use a linear model as base model of the hash function in this paper, and in fact other deep nonlinear models are trained in a similar way. Specifically, the hash function can be learned by the following solution,

$$\min_{\mathbf{P}^{(v)}} ||\mathbf{B} - \mathbf{P}^{(v)}\phi(\mathbf{X}^{(v)})||_F^2 + \omega||\mathbf{P}^{(v)}||_F^2, \tag{29}$$

where ω is a regularization parameter. The optimal solution of the variable $\mathbf{P}^{(v)} \in \mathbb{R}^{k\times q}$ is,

$$\mathbf{P}^{(v)} = \mathbf{B}\phi(\mathbf{X}^{(v)})^\top(\phi(\mathbf{X}^{(v)})\phi(\mathbf{X}^{(v)})^\top + \omega\mathbf{I})^{-1}. \tag{30}$$

The overall training procedures of S3ACH are described in Algorithm 1.

Algorithm 1 S3ACH

```
Input: Training labeled instances $\mathbf{X}_l^{(v)}$, label matrix $\mathbf{L}$, Training unlabeled instances $\mathbf{X}_u^{(v)}$, parameter $k, \beta, \delta, \rho, \gamma, \xi, \omega$, maximum iteration number $\mathbf{I}$.
Output: Binary codes $\mathbf{B}$.
Procedure:
1.Construct $\phi(X^{(v)})$ with randomly selected q anchors;
2.Initialize $\mathbf{B}$, $\mathbf{W}^{(v)}$, $\mathbf{G}$, $\mathbf{K}^{(v)}$ randomly with a standard normal distribution;
3.Initialize $\mathbf{H}^{(v)}$ = $\mathbf{B}$ - $\mathbf{K}^{(v)}$;
4.Initialize $\mathbf{S} = 2\mathbf{L}^\top\mathbf{L} - \mathbf{1}\mathbf{1}^\top$;
% step 1: Hash code learning
5.Repeat
    $\alpha_v$-step: Update $\alpha_v$ via Eq. (9).
    $\mathbf{W}^{(v)}$-step: Update $\mathbf{W}^{(v)}$ via Eq. (12).
    $\mathbf{G}$-step: Update $\mathbf{G}$ via Eq. (15).
    $\mathbf{B}$-step: Update $\mathbf{B}_l$,$\mathbf{B}_u$ via Eq. (21) and Eq. (22).
    Obtain $\mathbf{B} = [\mathbf{B}_l; \mathbf{B}_u]$.
    $\mathbf{K}_l, \mathbf{K}_u$-step: Update $\mathbf{K}_l, \mathbf{K}_u$ via Eq. (25) and Eq. (26).
    $\mathbf{H}_l, \mathbf{H}_u$-step: Update $\mathbf{H}_l, \mathbf{H}_u$ via Eq. (27) and Eq. (28).
 Until up to $\mathbf{I}$
6.End
% step 2: Hash function learning
7.Learn the hash mapping matrix $\mathbf{P}^{(v)}$ via Eq. (30).
Return Hash function
```

3.5 Time Cost Analysis

In this subsection, we analyze the time consumption required to close the solution with different parameters of S3ACH, i.e., $\mathbf{W}^{(v)}$, $\mathbf{G}$ and $\mathbf{B}_l, \mathbf{B}_u$. Specifically, the training time of $\mathbf{W}^{(v)}$ is $\mathcal{O}(qnk + k^2n + k^3 + k^2q)$, the training time of $\mathbf{G}$ is $\mathcal{O}(k^2n_l + k^3 + kcn_l + c^3 + k^2c)$, the training time of $\mathbf{B}_l$ and $\mathbf{B}_u$ are $\mathcal{O}(kqn_l + kcn_l + k^2q + k^2n_l + kcn_l)$ and $\mathcal{O}(kqn_u + kcn_l + k^2q + k^2n_u)$ respectively. It can be seen that the overall time complexity of S3ACH in training parameters is linearly proportional to the number of samples, i.e., n, indicating that our proposed optimization algorithm can satisfy efficient learning under large-scale data environment.

4 Experiments

4.1 Datasets

We validate our proposed S3ACH with two publicly available datasets of MIRFlickr [8] and NUS-WIDE [4].

1. **MIRFlickr dataset** consists of 25,000 instances tagged in 24 categories. Each instance contains a pair of image modality and text modality. The image modality uses 512-dim GIST features and the text modality uses 1,386-dim Bag-of Words (BoW) features. We select 20,015 instances and randomly choose 18,015 of these pairs as the train set and the rest 2,000 as the test set.
2. **NUS-WIDE dataset** contains 269,648 instances with 81 different tags. For each pair, the image modality uses a 500-dim SIFT vector and the text modality uses 1,000-dim BoW features. We select 186,577 instances of 10 of these common concept labels and randomly choose 1867 of them as test set and remaining ones as training set. In addition, we cannot load total training data at once due to the limitations in our experimental conditions, so we split them equally into four subsets for parallel experiments and take the average as the result.

4.2 Compared Baselines and Evaluation Metrics

Compared with unsupervised learning methods, our method makes full use of a small number of labels for semi-supervised hash learning, which is more applicable in real world. To evaluate the effectiveness, therefore, we select some classical and advanced unsupervised cross-modal hashing methods for comparison, including: CVH [9], IMH [15], CMFH [6], LSSH [32], UGACH [29], RUCMH [2], JIMFH [17], CUH [18], FUCMSH [25]. We focus on two cross-modal retrieval tasks, such as text retrieval by image (I→T) and image retrieval by text (T→I). We use the mean average precision (mAP) and top-k precision (P@k), and these evaluation metrics can evaluate retrieval performance. It should be noted that retrieving instances returned number in mAP is set to 100 in our experiment, while the top k is set from 0 to 1000 with 50 per step.

4.3 Implementation Details

There are parameters to be set to implement our proposed S3ACH, where β is the smoothing parameter, γ and ρ are used to balance the different terms, δ and ξ are used to optimize the solution process, and ω is used for the regularization of the hash function. We perform a grid search for all parameters (β from 2 to 9 and the rest from 10^{-5} to 10^{5}, with 10 times per step) and set anchor $q = 2500$ for kernelization, if not specified. In our experiments, the optimal parameter combinations is obtained, when $\{\beta = 9, \gamma = 10^{4}, \rho = 10^{-1}, \delta = 10^{5}, \xi = 10^{4}, \omega = 10^{5}\}$ and $\{\beta = 9, \gamma = 10^{-2}, \rho = 10^{5}, \delta = 10^{5}, \xi = 10^{3}, \omega = 10^{-5}\}$, respectively, corresponding to MIRFlickr and NUS-WIDE datasets. Note that our method is selected to use 20% labeled data for the experiments.

In addition, we implement FUCMSH, JIMFH and RUCMH ourselves with the parameters they provided. For the other baseline methods, we implement them directly with open source codes. The environment for experiments is a server with an Intel Xeon Gold 5220R @2.20 GHz with 24 cores and 64G RAM.

4.4 Results

We get the results that the mAP scores of our method and the comparison methods on two datasets with hash code lengths from 16 bits to 128 bits, as shown in Table 1, while Fig. 2 shows P@k curves of these methods with a hash code length of 64 bit. It can be seen as follows:

1. Our method outperform all comparative baselines on both datasets. Compared to the best baseline FUCMSH, our method improves on average 19.9% and 7.2% for I2T task, 5.8% and 2.7% for T2I task, respectively on MIRFlickr and NUS-WIDE datasets. These improvements in mAP scores indicate that our method enhances the learning process using a small amount of labeled data. In general, semi-supervised methods compared to unsupervised ones can achieve better retrieval performance.
2. The mAP scores of these methods are related to the hash code length. This may be because that longer hash code lengths are able to distinguish more semantic information. Nevertheless, the retrieval performance does not improve significantly when the encoding length is increased to a certain level, and the reason might be the longer hash codes enlarge the accumulation of quantization errors or other factors. In addition, our method result have no advantage at 16 and 32 bits length on T2I task of NUS-WIDE. This is probably due to the less quantity of label categories, which leads to insufficient use of semantic information at short hash codes.
3. Compared with graph-based methods (CVH, IMH), the matrix factorization based methods (including the remaining baseline and S3ACH) can better extract potential representations of multimodal data.
4. The trend of the P@k curve is similar to that of the mAP score, and our method outperforms the comparative baseline in retrieval performance. From Fig. 2, it can be seen that the effect of our method gradually approaches the optimal baseline FUCMSH with increasing the number of retrieval instances. This might be caused by the limited performance improvement of label-enhanced learning, where too many retrieval instances weaken the semi-supervised effect.

Table 1. The mAP results for all methods on MIRFlickr and NUS-WIDE datasets.

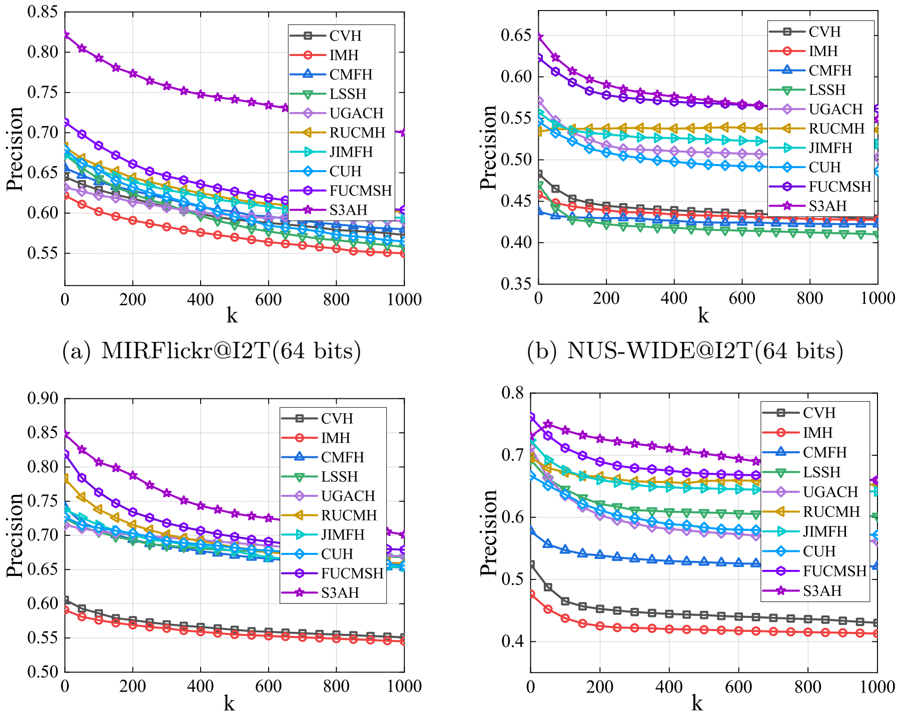
task	method	MIRFlikr				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
I→T	CVH	0.6317	0.6356	0.6229	0.6003	0.5201	0.5162	0.4899	0.4610
	IMH	0.6018	0.6089	0.6012	0.5981	0.4952	0.4871	0.4813	0.4309
	CMFH	0.6295	0.6232	0.6231	0.6045	0.4249	0.4601	0.4778	0.4592
	LSSH	0.5993	0.6123	0.6407	0.6471	0.4592	0.4251	0.4738	0.4606
	UGACH	0.5906	0.6206	0.6218	0.6178	0.5367	0.5545	0.5562	0.5580
	RUCMH	0.6447	0.6588	0.6567	0.6513	0.5433	0.5675	0.5691	0.5593
	JIMFH	0.6371	0.6572	0.6504	0.6489	0.5469	0.5710	0.5593	0.5672
	CUH	0.6312	0.6513	0.6433	0.6542	0.5375	0.5361	0.5449	0.5326
	FUCMSH	0.6679	0.6702	0.6809	0.6845	0.5532	0.5799	0.6108	0.6320
	S3ACH	**0.7931**	**0.7954**	**0.8186**	**0.8350**	**0.5635**	**0.6081**	**0.6481**	**0.7110**
T→I	CVH	0.6322	0.6317	0.6206	0.6111	0.5423	0.5271	0.4909	0.4647
	IMH	0.6227	0.6231	0.6119	0.6105	0.4979	0.4991	0.4821	0.4395
	CMFH	0.6893	0.7110	0.7331	0.7428	0.4048	0.5552	0.5805	0.5872
	LSSH	0.6779	0.7151	0.7412	0.7486	0.6631	0.6668	0.6805	0.6983
	UGACH	0.6390	0.6420	0.6472	0.6524	0.6259	0.6406	0.6673	0.6679
	RUCMH	0.6907	0.7364	0.7536	0.7501	0.6727	0.6859	0.6922	0.6846
	JIMFH	0.6885	0.7302	0.7476	0.7600	0.6614	0.6937	0.6998	0.7177
	CUH	0.6818	0.6773	0.6541	0.6570	0.6584	0.6527	0.6489	0.6417
	FUCMSH	0.7288	0.7460	0.7681	0.7745	**0.6822**	**0.7153**	0.7228	0.7289
	S3ACH	**0.7666**	**0.7792**	**0.8187**	**0.8292**	0.6482	0.7056	**0.7664**	**0.8065**

4.5 Ablation Experiments

In order to further proof the effectiveness of our method, ablation studies are conducted as follows.

Effects of Kernelization We design S3ACH-K, which uses the original data features instead of kernel-based features, i.e., replacing $\phi(\mathbf{X}^{(v)})$ in Eq. (6) with $\mathbf{X}^{(v)}$. As shown in Table 2, the comparison shows that the performance after kernelization is better than the one without kernelization.

Effects of Label Enhancement Learning We conduct experiments on different numbers of labels, and introduce a factor τ to measure the percentage of labeled data among all data. Note that when $\tau = 0$ means no labeled data, the label-enhanced learning framework is invalidated, i.e., the $\gamma||k\mathbf{S} - {\mathbf{B}_l}^\top(\mathbf{GL})||_F^2 + \rho||\mathbf{B}_l - \mathbf{GL}||_F^2$ in Eq. (6), when our method is degenerated to simple unsupervised learning. As shown in Table 2, comparing the unsupervised case, our label-enhanced framework plays a significant role in the effect, which prove its effectiveness. Moreover, the effect improves as the percentage of labeled data



(a) MIRFlickr@I2T(64 bits)

(b) NUS-WIDE@I2T(64 bits)

(c) MIRFlickr@T2I(64 bits)

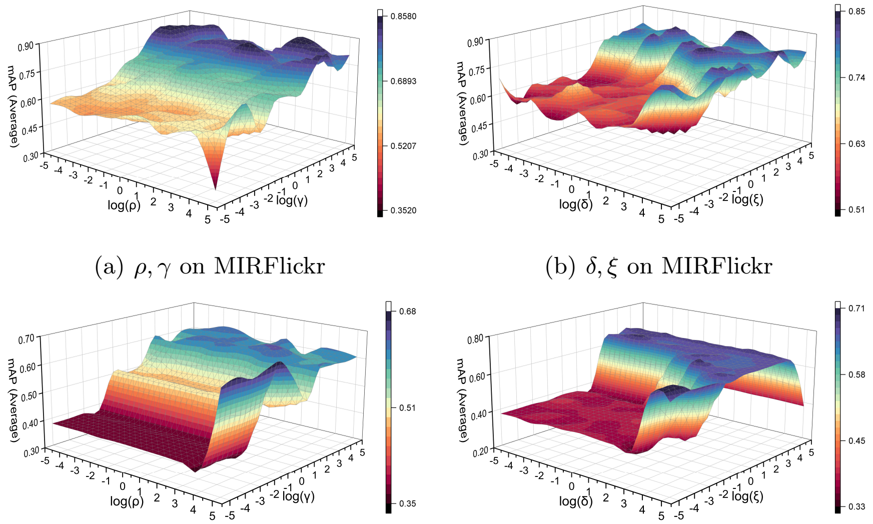
(d) NUS-WIDE@T2I(64 bits)

Fig. 2. Top-k precision curves on MIRFlickr and NUS-WIDE datasets.

increases, and it indicates that the framework can adapt to varying amounts of labeled data.

4.6 Parameter Sensitivity Analysis

In this subsection, we conduct a sensitivity analysis of the parameters in the method, and we divide them into three groups: (1) β is the smoothing parameter, which has a relatively small impact on other parameters and is therefore experimented independently; (2) γ and ρ are used to balance different parts of the asymmetric learning framework, and their values interact with each other, so a grid search is performed; (3) δ and ξ are parameters introduced to solve discrete constraint, and a grid search is conducted to observe their comprehensive impact on the results. By varying the parameter values in each group while keeping all other parameters constant, we perform these three sets of experiments on MIRFlickr and NUS-WIDE datasets with 64 bits hash codes. The search ranges of β is $\{2,3,4...,8,9\}$, that of γ, ρ, δ and ξ is $\{10^{-5},10^{-4},...,10^{3},10^{4},10^{5}\}$. For convenience of observation, we take the average mAP values of the two tasks (I2T and T2I) obtained from the experiments and plot them in Fig. 3 and Fig. 4.



(a) ρ, γ on MIRFlickr

(b) δ, ξ on MIRFlickr

(c) ρ, γ on NUS-WIDE

(d) δ, ξ on NUS-WIDE

Fig. 3. Sensitivity analysis of parameters sets.

From the figure, we observe that β exhibits different fluctuations within the specified range due to the characteristics of the dataset, achieving relatively stable results at 2, 9 and 6, 9 on MIRFlickr and NUS-WIDE, respectively. Additionally, we can observe that ρ has a relatively small impact on the overall performance within a large range, while γ shows a generally increasing trend in a stepwise manner as it increases. Furthermore, it is evident that δ and ξ have an interaction, and they yield better results when their values differ significantly. Through comprehensive observation, it can be concluded that the parameter values exhibit certain regularities within the global range and are not sensitive to the retrieval performance within the given range. This also indicates the robustness and practical applicability of our proposed method in real-world deployments.

4.7 Convergence Analysis

In this subsection, we use the NUS-WIDE dataset as an example to analyze the convergence of our proposed S3ACH . The experiments are performed with different hash code lengths, and the objective values obtained from each iteration of Eq. (6) are calculated. The first 50 values are selected and plotted in Fig. 5. It should be noted that to better demonstrate the convergence of different hash code lengths, we normalize these objective values. From the figure, we observe that S3ACH achieves rapid convergence within 10 iterations, confirming the effectiveness of the optimization algorithm.

Table 2. The mAP results of S3ACH for different labeling ratio.

task	method	MIRFlikr				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
I→T	S3ACH-K	0.7187	0.7549	0.7739	0.7826	0.4920	0.5317	0.5899	0.6625
	S3ACH$_{(\tau=0)}$	0.5744	0.5588	0.5682	0.5984	0.3752	0.3747	0.3751	0.3788
	S3ACH$_{(\tau=0.2)}$	0.7931	0.7954	0.8186	0.8350	0.5635	0.6081	0.6781	0.7110
	S3ACH$_{(\tau=0.5)}$	0.8153	0.8266	0.8406	0.8554	0.6309	0.7133	0.7574	0.7496
T→I	S3ACH-K	0.7905	0.7853	0.8111	0.8215	0.5184	0.5851	0.6631	0.7427
	S3ACH$_{(\tau=0)}$	0.5671	0.5516	0.5543	0.5871	0.3759	0.3762	0.3805	0.3947
	S3ACH$_{(\tau=0.2)}$	0.7666	0.7792	0.8187	0.8292	0.6482	0.7056	0.7764	0.8065
	S3ACH$_{(\tau=0.5)}$	0.8069	0.8397	0.8724	0.8833	0.7604	0.8244	0.8578	0.8587

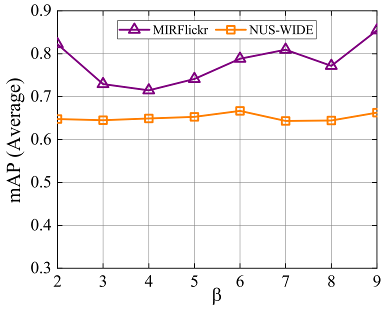


Fig. 4. Sensitivity analysis of parameter β on MIRFlickr and NUS-WIDE datasets.

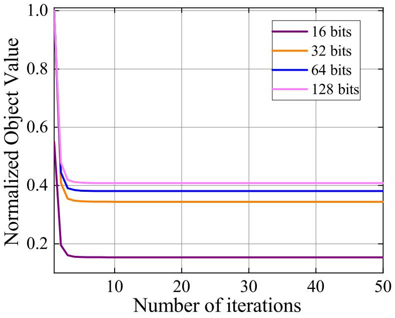


Fig. 5. Convergence curves on NUS-WIDE dataset.

5 Conclusion

In this paper, we propose an semi-supervised adaptive cross-modal hashing method called S3ACH. we add a self-learning dynamic weight parameter to the unsupervised representation of the potential public semantic space for balancing the contributions of each modality. Besides, an asymmetric learning framework is designed for semi-supervised hash learning process, so that it can make full use of limited labels to enhance the accuracy of hash codes. This framework can adapt to different amounts of labeled data. Afterwards, we propose a discrete optimization method to improve hash code learning process. We introduce an augmented Lagrange multiplier to separate the solved discrete variables and converge in fewer times. We perform experimental evaluations on two datasets and the results show that S3ACH is better than the existing advanced baseline methods with stability and high practicality. In the future, we will further focus on the effect of few labels for potential public semantic learning and try to apply the theory to the deep transformation models.

Acknowledgements This work is supported in part by the National Natural Science Foundation of China under the Grant No.62202501, No.62172451 and No.U2003208, in part by the National Key R&D Program of China under Grant No.2021YFB3900902, in part by the Science and Technology Plan of Hunan Province under Grant No.2022JJ40638 and in part by Open Research Projects of Zhejiang Lab under the Grant No.2022KG0AB01.

References

1. Cao, M., Li, S., Li, J., Nie, L., Zhang, M.: Image-text retrieval: A survey on recent research and development. arXiv preprint arXiv:2203.14713 (2022)
2. Cheng, M., Jing, L., Ng, M.K.: Robust unsupervised cross-modal hashing for multimedia retrieval. ACM Transactions on Information Systems **38**(3), 1–25 (2020)
3. Cheng, S., Zhou, Y., Zhang, W., Wu, D., Yang, C., Li, B., Wang, W.: Uncertainty-aware and multigranularity consistent constrained model for semi-supervised hashing. IEEE Transactions on Circuits and Systems for Video Technology **32**(10), 6914–6926 (2022)
4. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z.: Nus-wide: A real-world web image database from national university of singapore. In: Acm International Conference on Image & Video Retrieval (2009)
5. Da, C., Xu, S., Ding, K., Meng, G., Xiang, S., Pan, C.: AMVH: asymmetric multi-valued hashing. In: 2017 IEEE, CVPR. pp. 898–906 (2017)
6. Ding, G., Guo, Y., Zhou, J.: Collective matrix factorization hashing for multimodal data. In: IEEE on CVPR (2014)
7. Hu, P., Zhu, H., Lin, J., Peng, D., Zhao, Y.P., Peng, X.: Unsupervised contrastive cross-modal hashing. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
8. Huiskes, M.J., Lew, M.S.: The mir flickr retrieval evaluation. In: Acm International Conference on Multimedia Information Retrieval. p. 39 (2008)
9. Kumar, S., Udupa, R.: Learning hash functions for cross-view similarity search. In: International Joint Conference on Artificial Intelligence (2011)
10. Liu, W., Wang, J., Ji, R., Jiang, Y., Chang, S.: Supervised hashing with kernels. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012. pp. 2074–2081 (2012)
11. Meng, M., Wang, H., Yu, J., Chen, H., Wu, J.: Asymmetric supervised consistent and specific hashing for cross-modal retrieval. IEEE Trans. Image Process. **30**, 986–1000 (2021)
12. Shen, H.T., Liu, L., Yang, Y., Xu, X., Huang, Z., Shen, F., Hong, R.: Exploiting subspace relation in semantic labels for cross-modal hashing. IEEE Transactions on Knowledge and Data Engineering **33**(10), 3351–3365 (2020)
13. Shi, D., Zhu, L., Li, J., Zhang, Z., Chang, X.: Unsupervised adaptive feature selection with binary hashing. IEEE Transactions on Image Processing (2023)
14. Shi, Y., Zhao, Y., Liu, X., Zheng, F., Ou, W., You, X., Peng, Q.: Deep adaptively-enhanced hashing with discriminative similarity guidance for unsupervised cross-modal retrieval. IEEE Transactions on Circuits and Systems for Video Technology **32**(10), 7255–7268 (2022)
15. Song, J., Yang, Y., Yang, Y., Huang, Z., Shen, H.T.: Inter-media hashing for large-scale retrieval from heterogeneous data sources. ACM (2013)

16. Tu, R.C., Jiang, J., Lin, Q., Cai, C., Tian, S., Wang, H., Liu, W.: Unsupervised cross-modal hashing with modality-interaction. IEEE Transactions on Circuits and Systems for Video Technology (2023)
17. Wang, D., Wang, Q., He, L., Gao, X., Tian, Y.: Joint and individual matrix factorization hashing for large-scale cross-modal retrieval. Pattern Recognition (2020)
18. Wang, L., Yang, J., Zareapoor, M., Zheng, Z.: Cluster-wise unsupervised hashing for cross-modal similarity search. Pattern Recognition **111**(5), 107732 (2021)
19. Wang, Y., Chen, Z.D., Luo, X., Li, R., Xu, X.S.: Fast cross-modal hashing with global and local similarity embedding. IEEE Transactions on Cybernetics **52**(10), 10064–10077 (2021)
20. Wu, F., Li, S., Gao, G., Ji, Y., Jing, X.Y., Wan, Z.: Semi-supervised cross-modal hashing via modality-specific and cross-modal graph convolutional networks. Pattern Recognition **136**, 109211 (2023)
21. Wu, W., Li, B.: Locality sensitive hashing for structured data: A survey. arXiv preprint arXiv:2204.11209 (2022)
22. Yang, F., Ding, X., Liu, Y., Ma, F., Cao, J.: Scalable semantic-enhanced supervised hashing for cross-modal retrieval. Knowledge-Based Systems **251**, 109176 (2022)
23. Yang, F., Han, M., Ma, F., Ding, X., Zhang, Q.: Label embedding asymmetric discrete hashing for efficient cross-modal retrieval. Engineering Applications of Artificial Intelligence **123**, 106473 (2023)
24. Yang, Z., Deng, X., Guo, L., Long, J.: Asymmetric supervised fusion-oriented hashing for cross-modal retrieval. IEEE Transactions on Cybernetics (2023)
25. Yang, Z., Deng, X., Long, J.: Fast unsupervised consistent and modality-specific hashing for multimedia retrieval. Neural Computing and Applications pp. 1–17 (2022)
26. Yang, Z., Raymond, O.I., Huang, W., Liao, Z., Zhu, L., Long, J.: Scalable deep asymmetric hashing via unequal-dimensional embeddings for image similarity search. Neurocomputing **412**, 262–275 (2020)
27. Yu, G., Liu, X., Wang, J., Domeniconi, C., Zhang, X.: Flexible cross-modal hashing. IEEE in TNNLS **33**(1), 304–314 (2022)
28. Zhang, C., Li, H., Gao, Y., Chen, C.: Weakly-supervised enhanced semantic-aware hashing for cross-modal retrieval. IEEE Transactions on Knowledge and Data Engineering (2022)
29. Zhang, J., Peng, Y., Yuan, M.: Unsupervised generative adversarial cross-modal hashing. In: National Conference on Artificial Intelligence (2018)
30. Zhang, J., Peng, Y., Yuan, M.: SCH-GAN: semi-supervised cross-modal hashing by generative adversarial network. IEEE Trans. Cybern. **50**(2), 489–502 (2020)
31. Zhang, P.F., Li, Y., Huang, Z., Yin, H.: Privacy protection in deep multi-modal retrieval. In: ACM SIGIR. pp. 634–643 (2021)
32. Zhou, J., Ding, G., Guo, Y.: Latent semantic sparse hashing for cross-modal similarity search. In: ACM SIGIR. pp. 415–424 (2014)