The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

Asynchronous Teacher Guided Bit-wise Hard Mining for Online Hashing

Sheng Jin [1,2], Qin Zhou[2], Hongxun Yao*[1], Yao Liu[2], Xian-Sheng Hua[2]

[1]The Harbin Institute of Technology,
[2]Alibaba DAMO Academy, Alibaba Group
jsh.hit.doc@gmail.com, h.yao@hit.edu.cn.

Abstract

Online hashing for streaming data has attracted increasing attention recently. However, most existing algorithms focus on batch inputs and instance-balanced optimization, which is limited in the single datum input case and does not match the dynamic training in online hashing. Furthermore, constantly updating the online model with new-coming samples will inevitably lead to the catastrophic forgetting problem. In this paper, we propose a novel online hashing method to handle the above-mentioned issues jointly, termed **A**synchronus **T**eacher-Guided Bit-wise **H**ard Mining for **O**nline **H**ashing. Firstly, to meet the needs of datum-wise online hashing, we design a novel binary codebook that is discriminative to separate different classes. Secondly, we propose a novel semantic loss (termed *bit-wise attention loss*) to dynamically focus on hard samples of each bit during training. Last but not least, we design an asynchronous knowledge distillation scheme to alleviate the catastrophic forgetting problem, where the teacher model is delaying updated to maintain the old knowledge, guiding the student model learning. Extensive experiments conducted on two public benchmarks demonstrate the favorable performance of our method over the state-of-the-arts.

Introduction

In the big data era, large-scale image retrieval is widely used in many practical applications. One promising family of these retrieval methods is based on hashing, which merits in both low storage and efficient computation (Andoni and Indyk 2006; Zhang et al. 2014; Cao et al. 2018). Most existing methods are designed for offline training with a given collection of data, which may not well adapt to online streaming data. To address this issue, online hashing has emerged as a hot topic for its capacity in processing streaming data. Representative works include Online Kernel-based Hashing (OKH) (Huang, Yang, and Zheng 2013, 2017), Online Supervised Hashing (OSH) (Cakir, Bargal, and Sclaroff 2017), Mutual Information Hashing (MIH) (Cakir et al. 2017), Hadamard Codebook based Online Hashing (HCOH) (Lin et al. 2018), Balanced Similarity for Online Discrete Hashing (BSODH) (Lin et al. 2019), and Online Hashing with Efficient Updating (OHWEU) (Weng and Zhu 2020). In this paper, to better adapt to the practical scenario, we focus on the task of datum-wise online hashing.

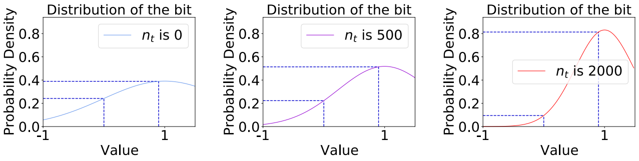

Figure 1: Illustration of the accumulated probability density for a certain bit during training along the timeline. With more and more training iterations, we can obtain a more compact distribution (with smaller standard deviation), which indicates the improvement of the fitting degree.

Existing datum-wise online hashing generally follows a two-stage protocol. Firstly, a codeword is generated for each class, then a hashing function is learned to fit this codeword. Therefore, the retrieval performance is closely related to the predefined codewords. In the early stage, OSH (Cakir, Bargal, and Sclaroff 2017) adopts random construction to generate codewords which degenerates the retrieval performance. Intuitively, a good codebook should be well separated among the codewords in the hamming space. From this perspective, the Hadamard (Lin et al. 2018) is proposed to maintain orthogonality among codewords. Orthogonality guarantees a good separability among the codewords, significantly boosting the recognition performance. However, the Hadamard is limited by the dimension constraint, which is strictly restricted as the power of 2. To extend it into flexible dimensions, LSH (Andoni and Indyk 2006) is introduced to project the initialized codewords, damaging the discriminative property of Hadamard. In addition to online hashing, the codebook is also adopted in some offline hashing methods. For example, Central Similarity Quantization (Yuan et al. 2020) constructs codewords by randomly sampling from Bernoulli distributions, maintaining sufficient mutual distances among codewords in expectation.

To construct well-spreading codewords in the binary space, in this paper, we propose to generate codewords by utilizing the singular matrix, which avoids the dimension constraints. Since the singular matrix is mutually orthogonal among rows and columns, it naturally preserves good separability among different codewords.

*Correspondence Author

Another drawback of existing OH methods is that all bits of the hashing codes are equally treated, which neglects the fact that for each bit, the fitting degree evolves with time as shown in Fig 1. And for each streaming input, it should be dynamically re-weighted to reflect the accumulated fitting degree (*e.g.*, a hard sample should be assigned with a bigger loss weight when it deviates from the ground truth binary code). To address this issue, we propose a bit-wise attention loss to pay more attention to the hard samples of each bit (especially for the well-fitted ones). This is reasonable since correcting the hard samples helps to generate a more compact distribution centered at the ground truth binary code. Specially, a gaussian distribution is calculated to simulate the fitting degree of each bit (centered at its ground truth binary code). In such a case, the standard deviation of the gaussian distribution can be utilized to indicate the fitting degree (*e.g.*, smaller std reflects a better fitting degree). Then hard samples deviate from the center are affiliated with higher weights to improve the OH learning.

Catastrophic forgetting is also demonstrated to severely influence the performance in an online setting (Weng and Zhu 2020), since the model is only updated with temporary input. The similarity relationship between the streaming input and existing database is introduced (Lin et al. 2019), which can not well adapt to the dynamic changes in online hashing, since hashing codes of database change with the model update. To address this problem, we introduce knowledge distillation (Hinton, Vinyals, and Dean 2014) into hashing and propose an asynchronous Teacher-Student training scheme.

The proposed Asynchronous Teacher Guided Hashing is based on the assumption that hashing codes generated by different sets of codewords should share similar semantic relationships, and the asynchronous updating scheme can preserve data distribution on previous data to alleviate the catastrophic forgetting problem. Please note here, different from transferring instances (Ba and Caruana 2014) or local semantic relationships (Yu et al. 2019), our method focuses on the distillation of global semantic relationships.

Overall, in this paper, we propose a combo to jointly solve the problem of bit-wise hard sample mining and catastrophic forgetting in datum-wise online hashing, termed **A**sychronous **T**eacher-Guided **H**ard Mining for **O**nline **H**ashing (**ATHOH**).

The main contributions of **ATHOH** are three-folds:

- We propose a singular-vectors-based method to generate discriminative codewords, which can be flexibly generalized to a flexible number of bits.
- We propose a novel bit-wise attention loss to reweight each bit of each streaming input, aiming to dynamically lower weight easy samples and focus on hard samples considering their fitting differences.
- We introduce knowledge distillation into the datum-wise online hashing to address the catastrophic forgetting problem, where global semantic relationships are exploited for better distillation, and asynchronous update strategy is utilized to preserve the previous knowledge.

Related Work

We introduce the most related works from two main aspects: *Online Hashing* and *Knowledge Distillation*.

Online Hashing has become a hot topic in recent years since it merits in updating the hash functions efficiently by using the streaming data online. Existing OH work can be further categorized into either supervised methods or unsupervised ones. Compared with the unsupervised methods (Leng et al. 2015; Chen, King, and Lyu 2017), supervised methods leverage the label information to learn hashing codes. OKH (Huang, Yang, and Zheng 2013) proposes a structured similarity loss function and learns the hash functions via an online passive-aggressive strategy (Crammer et al. 2006). Similar to OKH, BSODH (Lin et al. 2019) further propose a balanced similarity to reweight positive and negative pairs. However, these methods take pairs or batches as input. To consider the case of an extreme input (a single datum). Cakir *et al.* propose the online supervised hashing (OSH) which is inspired by the Error-Correcting Output Codes (ECOCs) (Dietterich and Bakiri 1994). Very recently, HCOH (Lin et al. 2018) introduce the Hadamard into online hashing to boost retrieval performance due to the orthogonality of the Hadamard. However, the strict dimension constraint of Hadamard limits its performance in some cases.

Knowledge Distillation is propose by Hinton *et al.* (Hinton, Vinyals, and Dean 2014) based on a teacher-student framework, which can be further subdivided into two categories: the absolute teacher (Zhou et al. 2018) and the relative one (Liu et al. 2019; Yu et al. 2019). The absolute teacher-based methods focus on transferring instance features from the teacher to the student, including regressing logits before the Softmax layer (Ba and Caruana 2014), the instance features of intermediate layers (Romero et al. 2015), the attention maps (Zagoruyko and Komodakis 2017) and so on (Lee, Kim, and Song 2018). However, the absolute teacher (Koratana et al. 2019) requires student networks has very similar architectures with their corresponding teacher, which is a crucial limitation. Very recently, some relative teacher based methods (Tung and Mori 2019; Tian, Krishnan, and Isola 2020) are proposed to be able to transfer semantic relationships to the student. However, these methods utilize pairwise similarity for semantic relationships, which is captured from a local perspective. Different from the above-mentioned methods, we propose a global semantic relationship-based knowledge distillation loss, where feature proximities calculated on the teacher model are transferred to guide the learning process of the student model.

Methodology

Problem Definition

Given n image samples $\mathcal{X} = \{x_1, ..., x_n\}$ with their corresponding labels $\mathcal{Y} = \{y_1, ..., y_n\}$, hashing methods aim to encode the given data point into a k-bit binary codes $b \in \{-1, 1\}$, which can preserve their semantic information. Following (Cakir, Bargal, and Sclaroff 2017; Lin et al. 2019), we adopt the linear projection-based hash functions, which is defined as:

$$b = sgn(W^T x) \tag{1}$$

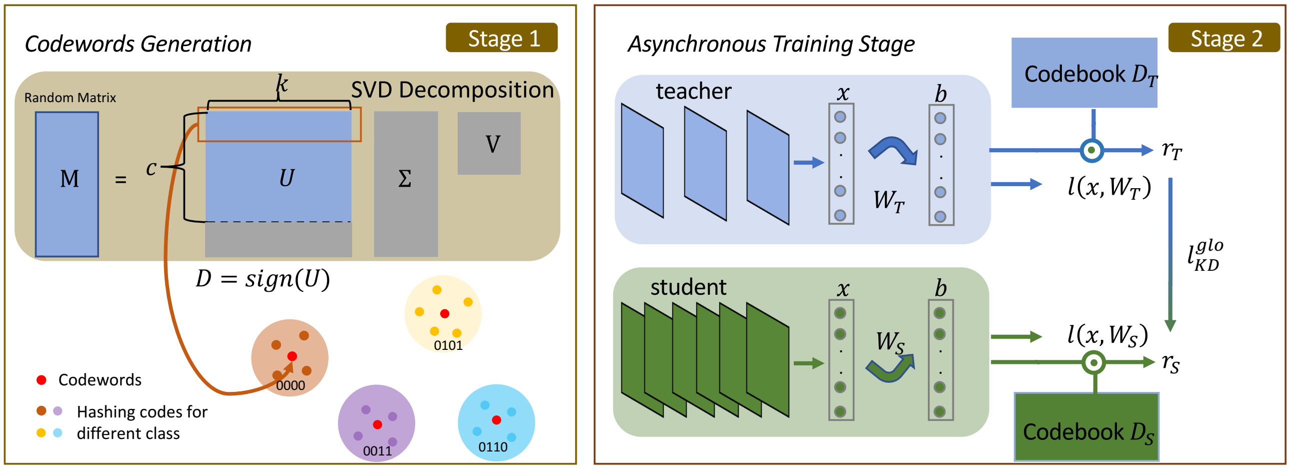

Figure 2: The ATHOH is a two-stage OH method. We first generate discriminative codewords for each label by using singular vectors. Then each input sample is fed into the model to generate their predicted binary coding. In our method, we construct two codewords sets for the teacher and student respectively. (1) To optimize the model effectively, we reweight each bit of streaming input such that focus more on hard samples. (2) To overcome the catastrophic forgetting problem, we asynchronous update the teacher and the student, where global semantic relationships are exploited to preserve the previous knowledge.

where $W = [w_1, w_2, \cdots, w_k] \in \mathcal{R}^{d\times k}$ is the projection matrix to be learned with $w_i \in \mathcal{R}^{1\times d}$ being responsible for the i-th hash bit, d denotes the dimensionality of features. The sign function $sgn(x)$ returns $+1$ if input variable $x > 0$, and returns -1 otherwise. For the online learning problem, the data comes in a streaming fashion. The current mapping matrix W are learned on the t-th round input streaming data $\mathcal{X}^t = \{x_1^t, ..., x_{n_t}^t\}$. In this paper, for the single datum input case, n_t is fixed as 1.

Singular Vector-based Codebook

The proposed algorithm is a two-step hashing method: (1) for each class, we assign a codeword for it. (2) each new input with the same labels is encoded to fit this codeword. For the multi-label case, the target codes for the input datum is the mean of the codewords of its related class labels. The objective function can be rewritten as:

$$l(x^t, W) = ||sgn(W^T x^t) - sgn(\sum_{y^t\in\mathcal{Y}^t} D_{y^t})||_F^2 \quad (2)$$

where we denotes D as the codebook, $\mathcal{Y}_t$ is the class label set of x_t and D_{y^t} is the y^t-th column of codebooks.

For this method, as illustrated in the introduction, a good codebook is important for hashing learning, and discriminative codewords can significantly benefit the performance. To generate valid codewords, we propose to construct the codebook by designing a unitary matrix. The unitary matrix has the property that the row and column vectors of each unitary matrix are mutually orthogonal, thus the codewords are well-spread in the binary space. The detailed construction of the unitary matrix is presented in the following.

In this paper, the unitary matrix is obtained by the singular value decomposition of a randomly generated matrix. We first initialize a random matrix $M \in \mathcal{R}^{m\times m}$, where $m = max(k, c)$ and c is the number of classes. Then we normalize this matrix by row. And the unitary matrix U is obtained by SVD of M. Finally, we obtain D by:

$$D = sgn(U[:, 1:k]) \quad (3)$$

where U is the singular matrixes of M. Given the codebook D defined in Eq. 3, we aim to optimize the objective function as follows. SGD optimization can be used to iteratively update for Eq. 3. To make the SGD feasible, we relax the non-convex and non-smooth sign function $sgn()$ as below:

$$l(x^t, W) = ||W^T x^t - sgn(\sum_{y^t\in\mathcal{Y}^t} D_{y^t})||_F^2 \quad (4)$$

In the t-round, the partial derivative of $l(x^t, \mathcal{W})$ with regard to w_i can be derived as:

$$w_i^{t+1} = w_i^t - \eta(w_i^t x - sgn(\sum_{y^t\in\mathcal{Y}^t} \mathcal{D}_{y^t i}))x^t \quad (5)$$

Discussion. In this section, we briefly summarize existing codewords generation methods, which is divided into two types: the random sampling-based and the orthogonality-based methods. The first type maximizes mutual distances among codewords in expectation. Each bit of codewords is sampled from a symmetrical distribution, such as Bernoulli distribution, uniform distribution, standard Gaussian distribution, and beta distribution, which is formulated as:

$$D = sgn(U > u_0) \quad (6)$$

where U is a random matrix sampled from a symmetrical distribution and u_0 is the mean of this distribution. We can easily prove that the distance between these codewords is $K/2$ in expectation, where k is the length of hashing codes.

In the second category, the orthogonality among codewords helps to generate discriminative codebooks. For example, a mutually orthogonal binary matrix is an optimal

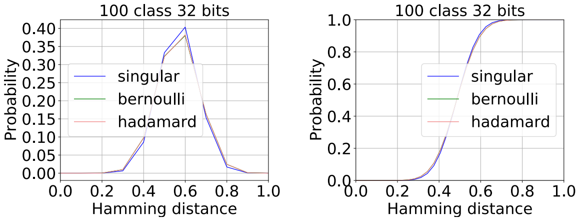

Figure 3: Statistics of the hamming distance of 32-bit codewords for 100 classes by adopting singular vectors or sampled from various symmetrical distribution. (a) Corresponding histogram distribution. (b) Cumulative distribution.

codebook. In this situation, codewords have a fixed distance $k/2$ to each other. Hadamard (Lin et al. 2018) is a special case when the dimension of the matrix is strictly restricted in the power of 2. The proposed method belongs to this category. However, different from existing methods (e.g., Hadamard matrix), we utilize a unitary matrix to retain orthogonality, which avoids the dimension limitation.

To compare the quality of these above-mentioned codebooks, we synthesize the codebooks for 10^4 times and then calculate the hamming distance among codeword pairs. Fig. 3 left is the corresponding distance histogram over all codewords pairs, and Fig. 3 right shows the distance cumulative histogram. It can be seen that singular vectors-based codewords are more stable and discriminative.

Bit-wise Attention Loss

The novel dynamic attention loss is designed to perform hard sample mining for each bit according to its fitting degree among bits during training. In the single-label case, by decomposition Eq.4 into bit-level, we can obtain the objective related the i-th bit as:

$$l(x^t, W) = \sum_i (sgn(w_i^T x^t) - D_{y^t i})^2 \tag{7}$$

where $D_{y^t i}$ is the i-th bits of the codeword D_{y^t}.

From Eq.7, we can see that each upcoming sample is equally weighted during the training process. As analyzed in the introduction part, more attention should be paid to the hard sample that deviates from the ground truth binary code. To address this issue, we propose a fitting degree guided weights to better guide the learning process.

In this paper, we calculate the accumulated statistics to reflect the fitting degree of each bit. The mean is set as its ground truth binary code and the standard deviation is calculated during the training stage. For the coming input, if its calculated $(sgn(Wx))$ deviates from the ground truth, then it is regarded as a hard sample on that bit. In this case, its density score will be small, therefore we utilize $1 - p(b_i|y)$ as the loss weight, to impose a huge punishment on the hard samples. The details of the information-aware loss are given below. First, we assume the bit follows a gaussian distribution based on their label, which is formulated as:

$$p(b_i|y) \sim \mathcal{N}(D_{yi}, \sigma_{yi}) \tag{8}$$

where b_i denotes the i th-bit of b, D_{yi} the i th-bit of the codeword D_y and σ the std. The value of σ_{yi} is updated in the training stage, which is computed as below:

$$\sigma^t_{yi} = \frac{1}{n^t_y}(n^{t-1}_y * \sigma^{t-1}_{yi} + (w^{t-1}_i x - D_{y^t i})^2) \tag{9}$$

where n^t_y denotes the numbers of samples with label y in the round t. In the practical experiments, the variance is updated at intervals.

Then we obtain the formulation of the loss weight for the i-th bit at the t-th iteration as follows::

$$\alpha^t_i = max((T - p(b_i|y))/T, 0) \tag{10}$$

where $T > 0$ is a constant, which rescales the value of weights within $[0, 1]$, since the value of probability density can be very large in some extreme cases.

Finally, the bit-wsie attention loss is formulated as:

$$l(x^t, W) = \sum_i \alpha_i (sgn(w_i^T x^t) - D_{y^t})^2 \tag{11}$$

where α_i is the balanced weight.

The gradient of the attention loss is formulated as:

$$w^{t+1}_i = w^t_i - \eta \alpha^t_i (w^t_i x - D_{y^t i}) x^t \tag{12}$$

where η is the learning rate.

For the multi-label case, we compute the weight of x^t under their class label set $\mathcal{Y}^t$, and then the overall weight is obtained by minimum over these values:

$$\alpha^t = \min_{y^t \in \mathcal{Y}^t} (\alpha') \tag{13}$$

Using minimum value in the multi-label case can help the hash function focus more on the shared bits among codewords.

Asynchronous Knowledge Distillation

For the online tasks, the previous samples cannot be used to retrain models, which leads to catastrophic forgetting. To solve the catastrophic forgetting in online hashing, we propose an asynchronous teacher-student framework by introducing a novel delaying teaching strategy, as illustrated in Figure 2. Specially, we first generate different codebooks for the teacher and the student, respectively. Then we train the hashing functions guided by these codebooks. To preserve the knowledge from former periods, the teacher model is updated and fixed between two updates at intervals.

As for what knowledge to transfer from the teacher to the student model, existing work focus on absolute knowledge such as instance-features or local semantic relationships (*e.g.*, pairwise similarity). However, in the datum-wise online hashing case, the local semantic knowledge is not available due to the single datum input. And different codebooks in our method result in the inconsistent output between the teacher and the student, rendering absolute knowledge transfer invalid. Therefore, we propose a novel knowledge transfer strategy, which models the global semantic relationship. Illustration of the differences among all the mentioned distillation losses are presented in Fig. 4.

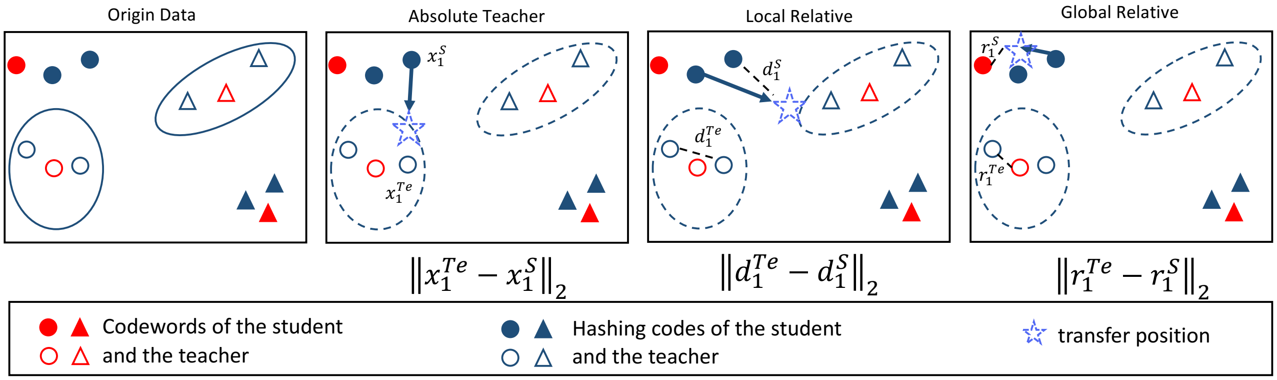

Figure 4: Illustration of the difference between absolute, local relative, and global relative teacher to student knowledge distillation. (left) Example of four data points from two classes (indicated by circular and triangle) and their corresponding codewords (colored red) in the embedding space of the teacher (indicated by hollow marks) and the student (indicated by solid marks), respectively. The transfer direction is given in a solid line and the dashed line represents semantic relationships. (Right three) show the absolute, local- and global-relative loss for student models.

The proposed semantic relationship-based knowledge distillation allows the student to have different network architecture with the teacher, making it more flexible. The global semantic loss is defined as:

$$\mathcal{L}_{KD}^{glo} = \lambda||r^S - r^{Te}||_2 \tag{14}$$

where r^S and r^{Te} are the distances between hashing codes of the input datum x^t and the pregenerated codebooks from the student and teacher model respectively:

$$\begin{aligned} r^S &= (k - W_{stu}^T x^t D_{stu})/2k \\ r^{Te} &= (k - W_{tea}^T x^t D_{tea})/2k \end{aligned} \tag{15}$$

where D_{stu} and D_{tea} is the codebooks for the teacher and student, respectively.

$$\frac{\partial \mathcal{L}_{KD}^{glo}}{\partial W_{stu}} = \lambda W_{stu}(r^S - r^{Te})D_{stu}^T \tag{16}$$

where λ is the weight parameters.

Optimization

The overall framework of the proposed **ATHOH** is a teacher-student framework. The teacher model is trained by the objective loss defined in Eq. 4 and updated by:

$$W_{tea}^{t+1} = W_{tea}^t - \eta x^t(\alpha_{tea}^t \cdot ({x^t}^T W_{tea}^t - D_{teay^t}^T)) \tag{17}$$

where $\alpha_{tea}^t = [\alpha_{tea1}^t, \alpha_{tea2}^t, \cdots, \alpha_{teak}^t] \in \mathcal{R}^{1\times k}$ is the weight vector and α_{teai}^t is defined in Eq. 10. $\cdot$ represents the element-wise multiplication.

The student model is trained by the objective loss defined in Eq. 4 and the novel global relative knowledge distillation loss defined in Eq 14, which is updated by:

$$\begin{aligned} W_{stu}^{t+1} =& W_{stu}^t - \eta x^t(\alpha_{stu}^t \cdot ({x^t}^T W_{stu}^t - D_{stuy^t}^T)) \\ & - \lambda W_{stu}(r^S - r^{Te})D_{stu}^T \end{aligned} \tag{18}$$

where r^{Te} and r^S are defined in Eq. 15 and α_{stu}^t is defined in the same way as α_{tea}^t. The whole learning procedure is summarized in Alg.1.

Algorithm 1 Asychronous Teacher-guided Bit-wise Hard Mining for Online Hashing (ATHOH)

```
Input: Training set $\mathcal{X}$ with their label $\mathcal{Y}$, the number of
    hash bits $k$, the learning rate $\eta$, parameters $\lambda$, the total
    number $n$ of training samples and the interval size $n_t$;
Output: The projection coefficient matrix $W_{stu}$.
Initialize $W_{stu}$ and $W_{tea}$ with the normal Gaussian distribution. Set Flag = 1.
Generate teacher codebooks $D_{tea}$ and student $D_{tea}$ as stated in Sec. Singular vectors-based Codebooks.
for $t = 1 : n$ do
    if $t <= n_t$ then
        Update $W_{stu}$ and $W_{tea}$ by Eq. 5.
    end if
    if Flag == 1 then
        Update $W_{tea}$ by Eq. 17.
    end if
    Update $W_{stu}$ by Eq. 18.
    if $t\%n_t == 0$ then
        Flag = -Flag
    end if
end for
return $W_{stu}$.
```

Experiment

Dataset and Evaluation Metric

CIFAR-10 (Krizhevsky and Hinton 2009) contains 60K 32 × 32 images in 10 classes. We randomly select 1000 images to form the test set and 20K instances from the rest images to form a train set. The rest images are used as the database.

NUS-WIDE (Chua et al. 2009) contains nearly 270k images with 81 classes. For NUS-WIDE, we follow (Weng and Zhu 2020) to use the images associated with the 21 most frequent concepts as the subset. We randomly select 2000 images as the test set and the remaining images are used as the training set and the database.

Dataset	CIFAR-10						NUSWIDE					
	8 bits	16 bits	32 bits	48 bits	64 bits	128 bits	8 bits	16 bits	32 bits	48 bits	64 bits	128 bits
OSH	0.123	0.126	0.129	0.131	0.127	0.125	0.402	0.458	0.482	0.498	0.508	0.524
MIHash	0.512	0.640	0.675	0.668	0.667	0.664	0.551	0.608	0.632	0.634	0.635	0.617
HCOH	0.536	0.688	0.698	0.707	0.724	0.734	0.595	0.645	0.692	0.691	0.694	0.698
BSODH	0.564	0.604	0.689	0.656	0.709	0.711	0.602	0.650	0.655	0.658	0.660	0.672
OHWEU	0.528	0.619	0.680	0.683	0.705	0.702	0.610	0.632	0.658	0.661	0.665	0.682
ATHOH	0.613	0.713	0.734	0.745	0.751	0.762	0.645	0.691	0.717	0.724	0.731	0.739

Table 1: Mean Average Precision results for different number of bits on two widely-used datasets.

Dataset	CIFAR-10						NUSWIDE					
	8 bits	16 bits	32 bits	48 bits	64 bits	128 bits	8 bits	16 bits	32 bits	48 bits	64 bits	128 bits
OSH	0.138	0.150	0.150	0.152	0.154	0.157	0.501	0.558	0.572	0.590	0.605	0.619
MIHash	0.560	0.703	0.744	0.743	0.739	0.745	0.668	0.714	0.732	0.735	0.734	0.729
HCOH	0.636	0.752	0.756	0.762	0.772	0.779	0.714	0.772	0.794	0.798	0.807	0.818
BSODH	0.623	0.709	0.730	0.742	0.749	0.767	0.703	0.722	0.728	0.735	0.758	0.772
OHWEU	0.632	0.723	0.745	0.749	0.768	0.773	0.724	0.770	0.782	0.788	0.795	0.801
ATHOH	0.697	0.786	0.791	0.795	0.805	0.812	0.762	0.807	0.825	0.836	0.844	0.853

Table 2: Precision@top500 results for different number of bits on two widely-used datasets.

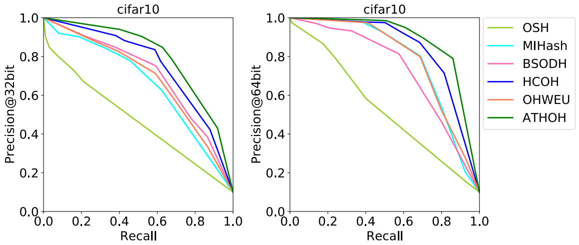

Figure 5: Precision-recall curves on CIFAR10 with 32 and 64 hash bits.

Each image in **CIFAR-10** and **NUS-WIDE** is represented by 4096 dimension features (Simonyan and Andrew 2015). For fair comparisons with existing work, all methods use identical training and test sets. We adopt Mean Average Precision (**MAP**), the precision of the top 500 retrieved examples (**Precision@top500**), Precision-Recall curves, Precision@top-N for quantitative evaluation, and mAP vs. different sizes of training instances curves.

Experimental Settings

We compare our method (**ATHOH**) with several state-of-the-art OH methods, including Online Supervised Hashing (OSH) (Cakir, Bargal, and Sclaroff 2017), OH with Mutual Information (MIHash) (Cakir et al. 2017), Hadamard Codebook based Online Hashing (HCOH)(Lin et al. 2018), Balanced Similarity for Online Discrete Hashing (BSODH) (Lin et al. 2019) and Online Hashing with Efficient Updating (OHWEU)(Weng and Zhu 2020).

Parameter Settings. The interval size n_i is 200. The learning rates η is 0.2 and the parameters λ to balance informatic semantic loss and global knowledge distillation loss is 0.1, which is discussed in the Supplementary.

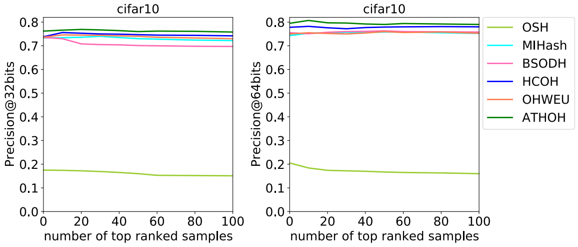

Figure 6: TopN-precision curves on CIFAR10 with 32 and 64 hash bits.

Quantitative Results

We first show the experimental results of MAP and Precision@topN on CIFAR-10 and NUS-WIDE. The results are shown in Tab. 1 and Tab. 2. Generally, the proposed ATHOH performs consistently better in terms of **MAP** and **Precision@top500** on all two benchmarks. For a depth analysis, in terms of MAP, compared with the second-best method, *i.e.*, HCOH, the proposed method achieves improvements of 9.3%, 3.4% on CIFAR10 and NUSWIDE, respectively. As for Precision@top500, compared with HCOH, the proposed method improves by 13.73%, 5.58% on two benchmarks, respectively. The large performance gain in terms of MAP and Precision@top500 validates the effectiveness of ATHOH.

Moreover, to illustrate the hash lookup results, we also evaluate Precision-Recall curves and Precision@top100 curves under the hash bit of 32 and 64 on CIFAR10. The results are shown in Fig. 5, Fig. 6 and Fig. 7. It clearly shows that our ATHOH obtains better results in most cases, which is consistent with the observations in Tab. 1 and Tab. 2.

Ablation Study

To further verify our method, we conduct experiments including (1) the analysis of each proposed component, (2) the effectiveness of singular vectors-based codewords, (3) the

Methods	CIFAR-10		NUSWIDE	
	32 bits	48 bits	32 bits	48 bits
Guassian	0.692	0.709	0.689	0.698
Bernouli	0.695	0.706	0.686	0.695
Hadamard	0.704	0.707	0.692	0.691
Singular vectors	0.702	0.716	0.690	0.701
+Bit-wise Attention	0.728	0.738	0.708	0.717
$+l_{KD}^{glo}$	0.725	0.735	0.704	0.712
Ours(full)	0.734	0.745	0.717	0.724

Table 3: The mAP scores of ATHOH using components.

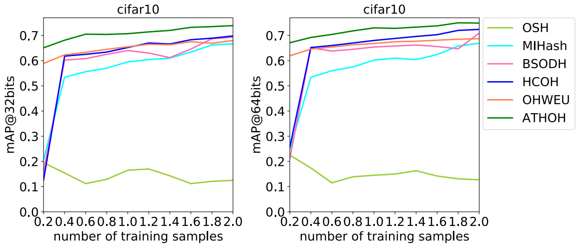

Figure 7: mAP vs. different sizes of training instances curves on CIFAR10 with 32 and 64 hash bits.

effectiveness of asynchronous teacher-student framework, (4) the effectiveness of global knowledge distillation loss.

Component Analysis of the ATHOH. Our proposed ATHOH method consists of two major components: *bit-wise attention loss* and *the asynchronous teacher-student framework*. To evaluate each component, we conduct an ablation study on retrieval performance. The experimental results are shown in TABLE 3. The *bit-wise attention loss* and *the knowledge distillation framework* alone can improve the performance by a large margin (almost 3%) on both multi-labels and single-label cases. Among which the *bit-wise attention loss* brings more improvement, since the reweight strategy is calculated on the existing database, which solves catastrophic forgetting in some degree. More importantly, altogether, the proposed ATHOH achives the best result.

Evaluation on Codewords Generation Approach. Since discriminant codewords are validated important for the datum-wise online hashing. We conduct experiments by using different strategies for codebooks construction, including sampling from Bernoulli distribution and Gaussian distribution, using the hadamada, and adopting singular vectors. The experimental results are shown in TABLE 3. First, when the value of bits is the power of 2, the singular vectors-based methods are comparable to directly using hadamada. Both hadamada- and singular vectors-based methods perform better with codewords random sampled from the symmetrical distribution. Second, when the length of bits is more general, singular-vectors based methods outperform others.

Evaluation on Asynchronous Teacher-Student Framework. To verify the effectiveness of the asynchronous training strategy, we compare our proposed teacher-student framework with two settings: (1) teacher and student are trained synchronously. (2) We design a short-term teacher to compare with our proposed model. The update frequency of teacher is faster than that of data distribution in Eq. 9. The experimental results are shown in TABLE 4. First, among these teacher models, the synchronous teacher is relatively worse. Since the student is trained in step with the teacher, the proposed loss degrades to a general distillation loss. Second, our proposed teacher model is better than its short-term counterpart. Since the short-term teacher can not preserve sufficient information of the existing database. Some parameters experimental results on the updata frequency of teacher are shown in Supplementary.

Methods	CIFAR-10		NUSWIDE	
	32 bits	48 bits	32 bits	48 bits
absolute	0.706	0.720	0.688	0.701
center-s	0.707	0.722	0.691	0.698
shared	0.713	0.725	0.692	0.704
center	0.715	0.728	0.697	0.705
synchronous	0.712	0.722	0.687	0.696
short-term	0.720	0.727	0.694	0.703
Ours(full)	0.725	0.735	0.704	0.712

Table 4: The mAP scores of ATHOH by using different teacher-student framework.

Evaluation on Global Semantic Knowledge. We evaluate the effectiveness of our proposed knowledge loss. First, we train the teacher and the student with shared codewords. In this setting, we transfer three types of knowledge, including instance hashing codes (*absolute*), the similarity of hashing codes, and their target codewords (*center*), and our proposed loss (*shared*). Besides, we also conduct comparative experiments with *center knowledge loss* where the teacher and student are trained by different codewords. First, using different codewords is better than using the same codewords to train the teacher and student, respectively, which means that the former has diverse semantic information. Second, our novel global semantic distillation loss is better than the *center* knowledge. Since the proposed loss transfers the relationship between hashing codes and the other codewords besides their target codeword. Finally, all the relative teachers are better than the absolute one.

Conclusion

In this paper, we present a novel **A**synchronous **T**eacher-Guided Bit-wise **H**ard Mining for **O**nline **H**ashing (ATHOH) method. The ATHOH aims to jointly address three challenging problems in datum-wise online hashing including *the discriminative codebook design*, *online bit-wise hard mining*, and *catastrophic forgetting*. Specifically, *singular vectors* is utilized to generate well-spreading codebook for datum-wise hashing. *Bit-wise Attention Loss* is introduced to reweight each sample to pursue more compact fitting to the ground truth during training. Last but not least, the asynchronous scheme is proposed to alleviate the *catastrophic forgetting* by delaying update the teacher model. Extensive experiments demonstrate the effectiveness of our method (ATHOH).

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Project No. 61772158, U1711265 and by National Key Research and Development Project, Ministry of Science and Technology, China under Project No. 2018AAA0101300.

References

Andoni, A.; and Indyk, P. 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, 459–468.

Ba, J.; and Caruana, R. 2014. Do deep nets really need to be deep? In *Advances in neural information processing systems*, 2654–2662.

Cakir, F.; Bargal, S. A.; and Sclaroff, S. 2017. Online supervised hashing. *Computer Vision and Image Understanding* 156: 162–173.

Cakir, F.; He, K.; Adel Bargal, S.; and Sclaroff, S. 2017. Mihash: Online hashing with mutual information. In *International Comference on Computer Vision*, 437–445.

Cao, Y.; Long, M.; Bin, L.; and Wang, J. 2018. Deep Cauchy Hashing for Hamming Space Retrieval. In *International Conference on Computer Vision and Pattern Recogintion*.

Chen, X.; King, I.; and Lyu, M. R. 2017. FROSH: FasteR Online Sketching Hashing. In *The Conference on Uncertainty in Artificial Intelligence*.

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.

Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; and Singer, Y. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7(Mar): 551–585.

Dietterich, T. G.; and Bakiri, G. 1994. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research* 2: 263–286.

Hinton, G.; Vinyals, O.; and Dean, J. 2014. Distilling the knowledge in a neural network. *Advances in neural information processing systems* .

Huang, L.-K.; Yang, Q.; and Zheng, W.-S. 2013. Online hashing. In *International Joint Conference on Artificial Intelligence*.

Huang, L.-K.; Yang, Q.; and Zheng, W.-S. 2017. Online hashing. *IEEE transactions on neural networks and learning systems* 29(6): 2309–2322.

Koratana, A.; Kang, D.; Bailis, P.; and Zaharia, M. 2019. Lit: Learned intermediate representation training for model compression. In *International Conference on Machine Learning*, 3509–3518.

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

Lee, S. H.; Kim, D. H.; and Song, B. C. 2018. Self-supervised knowledge distillation using singular value decomposition. In *Europeon Conference on Computer Vision*, 339–354.

Leng, C.; Wu, J.; Cheng, J.; Bai, X.; and Lu, H. 2015. Online sketching hashing. In *International Conference on Computer Vision and Pattern Recogintion*, 2503–2511.

Lin, M.; Ji, R.; Liu, H.; Sun, x.; Wu, Y.; and Wu, Y. 2019. Towards optimal discrete online hashing with balanced similarity. *the Association for the Advance of Artificial Intelligence* .

Lin, M.; Ji, R.; Liu, H.; and Wu, Y. 2018. Supervised online hashing via hadamard codebook learning. In *ACM International Conference on Multimedia*, 1635–1643.

Liu, Y.; Cao, J.; Li, B.; Yuan, C.; Hu, W.; Li, Y.; and Duan, Y. 2019. Knowledge distillation via instance relationship graph. In *International Conference on Computer Vision and Pattern Recogintion*, 7096–7104.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. Fitnets: Hints for thin deep nets. *International Conference on Learning Representations* .

Simonyan, K.; and Andrew, Z. 2015. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations* .

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive representation distillation. *International Conference on Learning Representations* .

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *International Comference on Computer Vision*, 1365–1374.

Weng, Z.; and Zhu, Y. 2020. Online Hashing with Efficient Updating of Binary Codes. In *the Association for the Advance of Artificial Intelligence*, 12354–12361.

Yu, L.; Yazici, V. O.; Liu, X.; Weijer, J. v. d.; Cheng, Y.; and Ramisa, A. 2019. Learning metrics from teachers: Compact networks for image embedding. In *International Conference on Computer Vision and Pattern Recogintion*, 2907–2916.

Yuan, L.; Wang, T.; Zhang, X.; Tay, F. E.; Jie, Z.; Liu, W.; and Feng, J. 2020. Central Similarity Quantization for Efficient Image and Video Retrieval. *International Conference on Computer Vision and Pattern Recogintion* .

Zagoruyko, S.; and Komodakis, N. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *International Conference on Learning Representations* .

Zhang, P.; Zhang, W.; Li, W.-J.; and Guo, M. 2014. Supervised hashing with latent factor models. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 173–182.

Zhou, G.; Fan, Y.; Cui, R.; Bian, W.; Zhu, X.; and Gai, K. 2018. Rocket launching: A universal and efficient framework for training well-performing light net. In *the Association for the Advance of Artificial Intelligence*.