Contrastive Self-Supervised Hashing With Dual Pseudo Agreement

YANG LI, YAPENG WANG, ZHUANG MIAO, JIABAO WANG, AND RUI ZHANG

Command and Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China

Corresponding author: Rui Zhang (3959966@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61806220, and in part by the Frontier Science and Technology Innovation Project of the Army Engineering University of PLA.

ABSTRACT
Recently, unsupervised deep hashing has attracted increasing attention, mainly because of its potential ability to learn binary codes without identity annotations. However, because the labels are predicted by their pretext tasks, unsupervised deep hashing becomes unstable when learning with noisy labels. To mitigate this issue, we propose a simple but effective approach to self-supervised hash learning based on dual pseudo agreement. By adding a consistency constraint, our method can prevent corrupted labels and encourage generalization for effective knowledge distillation. Specifically, we use the refined pseudo labels as a stabilization constraint to train hash codes, which can implicitly encode semantic structures of the data into the learned Hamming space. Based on the stable pseudo labels, we propose a self-supervised hashing method with mutual information and noise contrastive loss. Throughout the process of hash learning, the stable pseudo labels and data distributions collaboratively work together as teachers to guide the binary codes learning process. Extensive experiments on three publicly available datasets demonstrate that the proposed method can consistently outperform state-of-the-art methods by large margins.

INDEX TERMS
Unsupervised hashing, self-supervised learning, pseudo labels, mutual information.

I. INTRODUCTION
Hash learning [1]–[5] aims to accelerate the retrieval system’s speed when querying an image from a large-scale images database. This is a challenging task because only limited binary features are available to preserve the semantic relationship of the original high dimensional data [6], [7]. Recently, significant advances [8]–[19] have been made to tackle this problem using the ideas of deep hash learning.

Generally, deep hash learning methods can be roughly divided into two categories: supervised deep hashing [8]–[13] and unsupervised deep hashing [14]–[19]. Supervised deep hashing methods propose to learn deep networks as well as binary codes simultaneously with labeled data. However, all the datasets require time-consuming annotations, which are impractical or expensive in practice. Meanwhile, it is reported that deep supervised hashing models trained on one source domain have a significant performance drop on a new target dataset. This is because the data-bias existing between the source and target datasets.

Since it is unfeasible to annotate all images in the target dataset, unsupervised deep hashing methods [14]–[19] are therefore proposed to learn hash codes from pixels themselves without relying on semantic annotations. But how to transform high-dimensional images data into low-dimensional binary codes without labels and effectively preserve the structural information between images has become a new issue in this field [20]. State-of-the-art unsupervised deep hashing methods [14]–[19] for image retrieval always use a pretext task to learn representations on unlabeled data. Therefore, this kind of unsupervised methods can also be called self-supervised hash learning, which is based on the same core idea of creating labels by predicting them on their own.

Although the pretext tasks [21] such as generative adversarial networks (GAN) [22]–[24] and clustering can conduct alternatively to refine the binary codes to some extent, the training of the neural network is still substantially hindered by the inevitable label noise. Recently, some works try to select clean instances out of the noisy ones [25]. For example, DistillHash [18] treated the initial similarity relationship as noisy similarity signals and learned a distilled
dataset to perform confidence unsupervised hash codes learning. PSCH [25] used soft multi-part corresponding similarity and refined pseudo labels to further enhance the accuracy of hash codes. However, the performances of these unsupervised approaches are still far behind their fully-supervised counterparts [10], [12]. The main reason is that the noise labels are varied, which can be derived from the imperfect results of the clustering algorithm, the unknown number of target-domain identities, and the limited transferability of source-domain features.

To effectively address the problem of noisy labels in unsupervised deep hashing, in this article, we propose an simple but effective approach to self-supervised hash learning based on dual pseudo agreement (illustrated in Fig. 1). Specifically, we use two deep networks with identical architecture to produce refined pseudo labels. By adding a consistency constraint, our method can prevent corrupted labels which encourages generalization for effective knowledge distillation. Furthermore, we use the refined pseudo labels as a stabilization constraint to train hash codes, thereby ensuring the error flow from the noisy label would not be accumulated in the process of hash learning.

The main contributions of our method can be summarized as follows:

- We propose a simple but effective dual pseudo agreement labelling paradigm, which allows us to learn deep binary codes robustly even with extremely noisy labels. Essentially, this conclusion is almost trivial from a statistical perspective, but in practice, it significantly eases learning hash codes by lifting requirements on the availability of clean data.
- Based on the stable pseudo labels, we propose a self-supervised hashing method with mutual information and noise contrastive loss. Throughout the process of hash learning, the stable pseudo labels and data distributions collaboratively work together as teachers to guide the binary codes learning process.
- Extensive experiments on three benchmark datasets show the robustness of the proposed method is superior to many state-of-the-art approaches. Furthermore, the ablation studies clearly demonstrate the reliability of dual pseudo agreement and contrastive loss.

II. RELATED WORK

Deep hashing methods have been broadly studied in previous work, both theoretically and empirically [8]–[10], [12], [26]. In this section, we focus on the recent progress on unsupervised deep hashing methods. Since unsupervised deep learning is a challenging task, it also brings new challenges for unsupervised deep hashing.

A. UNSUPERVISED DEEP HASHING METHODS

Unsupervised deep hashing methods can learn binary codes without semantic annotations, which are widely adopted to accelerate the approximate nearest neighbor (ANN) search for large-scale database [27]. We briefly group the unsupervised deep hashing methods into two categories: generative model based deep hashing and pseudo label learning based deep hashing.

Generative model based deep hashing mainly explore image synthesis task in an unconditioned manner for hash learning [22]. For example, stochastic generative hashing (SGH) [16] proposed a novel generative approach based on the minimum description length principle to learn binary hash codes. HashGAN [28] proposed a new deep unsupervised hashing network, which consists of three networks: a generator, a discriminator, and an encoder. Binary generative adversarial network (BGAN) [14] proposed to simultaneously learn a binary representation and generate an image plausibly similar to the original one. However, unsupervised hashing based on generative models are limited for their complexity. Furthermore, these methods can not capture semantic relationships between different data points by the reconstruction loss.
Recently, pseudo label [29], [30] learning based deep hashing is widely used and leads to better performance. In general, these methods use pseudo labels to convert the unsupervised problem into a supervised problem. For example, Hu et al. [31] proposed the first two-step pseudo label based unsupervised deep discriminative hashing algorithm. In the first stage, they cluster images via k-means and the cluster labels are treated as pseudo labels. In the second stage, the pseudo labels are used as soft supervisions to train a deep hashing network and minimize the classification loss and quantization loss. Unsupervised triplet hashing (UTH) [32] proposed to construct the image triplet by an anchor image, a rotated image, and a random image. Similarity-adaptive deep hashing (SADH) [13] proposed to train hashing model alternatively over three modules, which substantially improves the robustness of binary codes optimization. Semantic structure based unsupervised deep hashing (SSDH) [17] analyzed cosine distance distribution of image features and constructed the similarity matrix through the pairwise distances. DistillHash [18] improved the performance of SSDH [17] by using the initial similarity relationship as noisy similarity signals and learned a distilled dataset to perform confidence unsupervised hash codes learning. However, most of these works ignore the label noise, which inevitably yields negative pairs that share similar semantic meaning in the embedding space. Such problem can also lead to class collision [33] and hurt hash codes learning.

### B. Deep Learning with Noisy Labels

Previous works have demonstrated that noisy labels can severely degrade the performance of supervised deep learning algorithms [34]. Several methods have been proposed to address this problem by outlier detection [35], thresholding [36] or reweighting [37].

For example, MentorNet [38] proposed to learn a data-driven curriculum dynamically with StudentNet, which can significantly improve the generalization performance of deep networks trained on corrupted training data. Co-teaching [39] proposed to train two deep neural networks simultaneously, and identify the examples with small loss as clean examples. Noise-tolerant [40] proposed a novel training paradigm, which uses angular margin based loss to reflect the probability of noisy labels. PurifyNet [41] proposed to refine the annotated labels and optimizes the neural networks by progressively adjusting the predicted logits. Li et al. [42] proposed a distillation framework, which uses “side” information to “hedge the risk” of learning from noisy labels. Shu et al. [43] proposed a meta-learning method, which can adaptively learn hyper-parameter in robust loss functions. JoCoR [44] proposed a joint training method to select small-loss examples and reduce the diversity of two networks during training. DivideMix [45] proposed to model the per-sample loss distribution with a mixture model, which can dynamically divide the training data into a labeled set with clean samples and an unlabeled set with noisy samples in a semi-supervised manner.

In unsupervised deep hashing domain, DistillHash [18] treated the initial similarity relationship as noisy similarity signals and learned a distilled dataset to perform confidence unsupervised hash codes learning. PSCH [25] used soft multi-part corresponding similarity and refined pseudo labels to further enhance the accuracy of hash codes. Despite these important findings, how to further improve the performances of these unsupervised approaches stills an open question. In this article, we provide further insights into the unsupervised hash learning by investigating refined pseudo labels and robust loss functions.

### III. Proposed Method

In this section, we first introduce the dual pseudo agreement labelling paradigm and then provide a self-supervised hashing method based on the contrastive loss.

### A. Dual Pseudo Agreement

Suppose we are given a dataset \( X = \{x_i\}_{i=1}^n \) with \( n \) images, where \( x_i \) denotes one image in the image retrieval dataset. The goal of deep image hashing is to learn a nonlinear hash function \( H(x_i) \) by using the initial similarity relationship as noisy similarity signals and learned a distilled dataset to perform confidence unsupervised hash codes learning.

Base on the pairwise similarities, the key idea of deep hashing lies in constructing a similarity matrix \( S \in \{0, 1\}^{N \times N} \), so that when \( S_{ij} = 1 \) denotes that \( x_i \) and \( x_j \) are similar, \( S_{ij} = -1 \) denotes that \( x_i \) and \( x_j \) are dissimilar. According to whether extra labels information \( Y = \{y_i\}_{i=1}^N \) are used, existing deep hashing methods can be categorized into either supervised or unsupervised learning. In the supervised setting, instance level annotations (e.g., classification tags) are used to measure whether two images are similar. In the unsupervised setting, no additional labels are available and hence the raw features are the only source of judgment.

Obviously, deep supervised hashing methods can provide more accurate estimations on \( S \) than unsupervised methods. However, annotating all the images in the dataset is time-consuming, which is impractical or expensive in practice. Hence, we focus on improving the performance of unsupervised deep hashing methods that are easier to be deployed to real-world scenarios.

According to the above analysis, we can see that the difference between supervised and unsupervised deep hashing methods is not so big. Actually, the supervised hashing methods do not require real manually labeled information, but only need to know the similarity between different images. Under the unsupervised manner, the similarity matrix \( S \) can also be obtained form calculating the distance between their feature vectors. However, due to the similarity of feature vectors are not as accurate as label information, the estimated \( S \) always contain some noisy information. In other words, if we can estimate the similarity matrix \( S \) more accurately, we can further improve the accuracy of unsupervised deep hashing.
In this article, we use two different deep neural networks denoted as \( f(\mathbf{x}, \Theta_1) \) and \( f(\mathbf{x}, \Theta_2) \) to estimated the refined similarity matrix \( S \). As illustrated in Fig.1, we first used two different pre-trained deep neural networks to extract two deep features set \( \mathcal{F}_1 = \{f^1_i\}_{i=1}^N \) and \( \mathcal{F}_2 = \{f^2_i\}_{i=1}^N \). Then, we can roughly label the training data based on their features centroid matrix \( C_1 \) and \( C_2 \). It means that we assume the corruption of labels is a random variable distributed according to the clean target. This kind of assumption is an abstract approximation to the real-world corruption of labels [35].

Inspired by the deep clustering theory [46], iterating the \( k \)-means centers of \( C_1 \) and \( C_2 \), the deep image features will be divided into \( k \) different groups by optimizing the following problems:

\[
\begin{align*}
\min_{c_1 \in \mathbb{R}^d \times k} & \quad \frac{1}{N} \sum_{n=1}^{N} \min_{y_i \in \{0, 1\}^k} \| f(\mathbf{x}_i, \Theta_1) - c_1^{y_i} \|_2^2 \\
\min_{c_2 \in \mathbb{R}^d \times k} & \quad \frac{1}{N} \sum_{n=1}^{N} \min_{y_i \in \{0, 1\}^k} \| f(\mathbf{x}_i, \Theta_2) - c_2^{y_i} \|_2^2
\end{align*}
\]

(1) 

(2)

where \( d \) is dimension of the feature vectors, \( y_i \) is the label associated with \( \mathbf{x}_i \). After solving Eq.(1) and Eq.(2), we can use these assignments as pseudo labels \( \hat{Y}_1 = \{y_i^1\}_{i=1}^N \) and \( \hat{Y}_2 = \{y_i^2\}_{i=1}^N \).

However, due to the domain gaps, the pre-trained networks \( f(\mathbf{x}, \Theta_1) \) and \( f(\mathbf{x}, \Theta_2) \) will produce lots of noisy labels, which always located at the edge away from the center of the sample features. In consequence, we use two distance thresholds to reject the noisy samples:

\[
\begin{align*}
1 - \frac{f(\mathbf{x}_i, \Theta_1) \cdot C_1^{y_i}}{\| f(\mathbf{x}_i, \Theta_1) \| \cdot \| C_1^{y_i} \|} & \leq \mu_1^{d} + \sigma_1^{d} \quad \text{(3)} \\
1 - \frac{f(\mathbf{x}_i, \Theta_2) \cdot C_2^{y_i}}{\| f(\mathbf{x}_i, \Theta_2) \| \cdot \| C_2^{y_i} \|} & \leq \mu_2^{d} + \sigma_2^{d} \quad \text{(4)}
\end{align*}
\]

where \( \mu_1^{d} \) and \( \mu_2^{d} \) are the mean of the cosine distances between the feature and the centroid feature, \( \sigma_1^{d} \) and \( \sigma_2^{d} \) are the standard deviation of the cosine distance distribution.

Based on the two distance thresholds in Eq.(3) and Eq.(4), we can get two refined pseudo labels set. From a statistical viewpoint, we calculate the common parts of the two sets as maximum likelihood (ML) estimation, and formulate the similarity matrices:

\[ S_{ij} = \left( 4 - \| f_i^1 - f_j^1 \|_2 - \| f_i^2 - f_j^2 \|_2 \right) / 4 \]  

(5)

where \( \| f_i - f_j \|_2 \) is the Euclidean distance between two feature vectors which lies in the range of \([0, 2]\) for two normalized vectors. Furthermore, we consider two different modal information in Eq.(5). And due to the refined step, our final labeled dataset has \( m \) samples, which is smaller than the original dataset \( N \).

**B. CONTRASTIVE SELF-SUPERVISED HASHING**

Training accurate hash codes in the presence of noisy labels is very important for unsupervised hash learning. Our contrastive self-supervised hashing method is based on the dual pseudo agreement labels and distribution as introduced in Section III-A, where the final pseudo labels are refined by two different pre-trained deep networks.

Our hashing framework used the final pseudo labels and data-to-data relations to train a deep siamese network [47] with the same structures and parameters. Siamese network is a common architecture to reduce the model size, which is widely used in deep learning field [48], [49]. For the backbone of deep siamese network, we use the PSCH architecture [25] for hash codes learning. The overall of our method is illustrated in Fig. 2. In our framework, the main intuition behind our method is to learn the binary representations that encode the underlying shared information between different images. Therefore, our loss function has three complementary parts: contrastive loss, hashing loss, and balance loss.
1) CONTRASTIVE LOSS
In order to alleviate label noise, we propose a contrastive loss that is not dependent on the labels. By introducing the smoothness of decision boundaries along the data manifold, this contrastive loss can improve the robustness of hash codes. Formally, for any data point $x$, contrastive loss aims to learn an encoder $\Phi$ such that:
\[
\text{score}(\Phi(x), \Phi(x^+)) \gg \text{score}(\Phi(x), \Phi(x^-))
\]
where $x$ is an “anchor” data point, $x^+$ is a positive sample, and $x^-$ is a negative sample. The score function is a metric that measures the similarity between two features. From Eq.(6), we can see that the contrastive loss aims to bring closer samples from the same instance and separate samples from different instances.

To optimize this property, we introduce a mutual information loss, called InfoNCE [50] for hashing learning. Different from [50], our method is from the perspective of resistance to label noise. Furthermore, our loss can encourage the score function to assign large values to positive examples and small values to negative examples:
\[
\mathcal{L}_{MI} = -\log \frac{\exp (\Phi(x)^T \Phi(x^+))}{\exp (\Phi(x)^T \Phi(x^+)) + \sum_{j=1}^{N-1} \exp (\Phi(x)^T \Phi(x_j))}
\]
(7)

As we can see, the contrastive objective in Eq.(7) is a kind of mutual information (MI) loss, which explicitly considers data-to-data relationships. Specifically, minimizing the InfoNCE loss can maximize a lower bound on the mutual information between $\Phi(x)$ and $\Phi(x^+)$ [51]. Furthermore, the MI loss is better than the traditional classification loss due to the separation in entropy and conditional entropy.

2) HASHING LOSS
In order to preserve the learned semantic structure, we can define the hashing loss function based on the hash layer output $b_i$ and $b_j$:
\[
\mathcal{L}_H = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \|S_{ij} (H_{ij} - S_{ij})^2
\]
\[
\text{s.t. } H_{ij} = \frac{1}{K} b_i \cdot b_j, \quad b_i \in \{-1, +1\}^K
\]
(9)
where $H_{ij}$ is the similarity structure in the hash code space, and $S_{ij}$ is defined by the final pseudo labels in Eq.(5). According to Eq.(9), we can see that hashing loss can minimize the loss between semantic structure and hash similarity structure.

During hash learning, we relax the binary codes $b_i = \tanh(\Phi(x_i))$ to be the output of the neural network. Therefore, we can integrate the above loss function into our deep architectures. However, this relaxed strategy may be harmful to the performance due to the relaxation error [52]. To reduce relaxation error, it is a natural choice to introduce the balance loss when solving the relaxed problem.

3) BALANCE LOSS
Balance loss originally proposed for balancing hash codes, it is also an effective way to make the binary value close to the output value of the network. Balance loss has also been proved to be robust to label noise, and is defined as:
\[
\mathcal{L}_B = \sum_{i=1}^{K} (\|b_i - \text{sgn}(h_i)\|^2 + \|\frac{2b_i - K}{2K}\|^2)
\]
(10)
where $\|b_i - \text{sgn}(h_i)\|^2$ measures the quantization error between the binary code $b_i$ and the deep activation $h_i$. And $\|\frac{2b_i - K}{2K}\|^2$ will constraint the binary codes fulfill the binary-valued requirement more appropriately.

4) FINAL LOSS FUNCTION
Inspired by the benefit of complementary learning [53], we propose to combine the strength of the three loss functions in Eq. (7), Eq. (9), and Eq. (10). The final loss function can leverage their complementary aspects for both robust and sufficient learning. Formally, the final loss function of our method is defined as:
\[
\mathcal{L} = \mathcal{L}_{MI} + \alpha \cdot \mathcal{L}_H + \beta \cdot \mathcal{L}_B
\]
(11)
where $\alpha > 0$ and $\beta > 0$ are two hyper-parameters to balance the hash loss and balance loss. With proper balancing between the three terms, our final loss function can leverage both the convergence and the robustness advantages.

Note that, based on Eq.(11), we can solve the vanishing gradient problem by using a smooth activation function $b_i = \tanh(h_i)$ to approximate the non-smooth activation function $b_i = \text{sgn}(h_i)$ by continuation. Therefore, the final loss can be optimized by the standard stochastic gradient descent (SGD) method. During training, we randomly select 48 images from the training set to update the parameters. During the testing process, we can predict image binary codes just by forward propagation.

IV. EXPERIMENTS
Following common practice in hash learning, we provide comprehensive experiments to demonstrate the effectiveness of our method. We first introduce three public datasets and implementation details. Following these settings, we compare our method with some representative prior works. Finally, parameter sensitivity, training complexity, and retrieval results are further reported.

A. BENCHMARK DATASETS
We evaluate the performance of our method on three popular benchmark datasets: CIFAR-10, NUSWIDE, and FLICKR25K. All images except for the query images are used as the retrieval set. The description of these datasets are as follows.

- CIFAR-10. This dataset is a popular subset of 80M tiny images, which composed of 60,000 color 32 × 32 images

\[^{1}\text{http://www.cs.toronto.edu/~kriz/cifar.html}\]
from 10 classes (each class contain 6,000 images). Following experimental settings in [14], [17], [18], we randomly select 500 images in each category as the training data, and 1,000 images in each category as the query dataset.

- **NUSWIDE**. This dataset contains 269,648 color images referring to 81 semantic concepts. Following experimental settings in [18], we also use the subset of the 10 most popular concepts, where each concept contains of at least 5,000 images. We randomly select 100 query images and 1,000 training images in each class respectively.

- **FLICKR25K**. This dataset is a popular dataset that contains 25,000 images collected from Flickr website. Each image is annotated with one of 38 unique labels. We randomly selected 4,000 images for training and 1,000 query images for testing.

### B. IMPLEMENTATION DETAILS

The implementation of our method is based on the TensorFlow [54] framework. For non-deep learning-based hashing methods [55]–[60], we utilize the 4096-D deep features extracted from the last fully-connected layer of the VGG16 network to learn hash codes. For deep hashing methods, we use raw images as inputs and adopt the VGG16 as the backbone architectures to extract the deep representations.

During training, we also use mini-batch SGD algorithm with 0.9 momentum to optimize the network. The learning rate is 0.001 and the decline rate of learning is 0.1 after every 100 iterations. We set the batch size to 48, $\alpha = 0.6$ and $\beta = 0.01$. Detail analysis of these hyper-parameters is shown in the following sections. Our experiments are implemented on Windows 10 (64-Bit) platform with a Geforce RTX 2080 Ti GPU.

In order to evaluate the performance of our method, we use three widely employed evaluation criteria [61]: Mean Average Precision (MAP), TopN-precision curves, and Precision-recall (PR). Detailed description of the above evaluation criteria can be referred to [62].

### C. COMPARISON WITH STATE-OF-THE-ART

Our method is systematically compared with several state-of-the-art hashing methods, including six traditional unsupervised hashing methods (LSH [55], SH [56], ITQ [57],

---

**TABLE 1.** The retrieval results of MAP with different length hash codes on CIFAR-10, NUSWIDE, and FLICKR25K datasets. The first and second best results are shown in boldface and underline.

| Method      | CIFAR-10 (MAP) | NUSWIDE (MAP) | FLICKR25K (MAP) |
|-------------|----------------|---------------|-----------------|
|             | 16 bits | 32 bits | 64 bits | 128 bits | 16 bits | 32 bits | 64 bits | 128 bits | 16 bits | 32 bits | 64 bits | 128 bits |
| LSH [55]    | 0.132   | 0.158   | 0.167   | 0.179   | 0.432   | 0.441   | 0.443   | 0.482   | 0.583   | 0.589   | 0.593   | 0.601   |
| SH [56]     | 0.161   | 0.158   | 0.151   | 0.154   | 0.446   | 0.454   | 0.493   | 0.500   | 0.592   | 0.592   | 0.602   | 0.621   |
| ITQ [57]    | 0.194   | 0.209   | 0.215   | 0.219   | 0.528   | 0.532   | 0.532   | 0.542   | 0.619   | 0.632   | 0.635   | 0.648   |
| PCAH [58]   | 0.143   | 0.159   | 0.173   | 0.184   | 0.463   | 0.453   | 0.464   | 0.492   | 0.609   | 0.611   | 0.603   | 0.607   |
| DSH [59]    | 0.162   | 0.188   | 0.192   | 0.206   | 0.520   | 0.523   | 0.535   | 0.537   | 0.607   | 0.612   | 0.612   | 0.615   |
| Sh [60]     | 0.144   | 0.167   | 0.178   | 0.184   | 0.453   | 0.460   | 0.496   | 0.513   | 0.611   | 0.603   | 0.634   | 0.625   |
| DeepBit [15]| 0.220   | 0.241   | 0.252   | 0.253   | 0.454   | 0.463   | 0.476   | 0.492   | 0.593   | 0.593   | 0.620   | 0.635   |
| SGH [16]    | 0.180   | 0.183   | 0.189   | 0.190   | 0.494   | 0.483   | 0.487   | 0.498   | 0.616   | 0.628   | 0.625   | 0.621   |
| SSDG [17]   | 0.257   | 0.256   | 0.259   | 0.260   | 0.623   | 0.629   | 0.632   | 0.649   | 0.662   | 0.673   | 0.673   | 0.677   |
| BGAN [14]   | 0.525   | 0.531   | 0.562   | –       | 0.684   | 0.714   | 0.730   | –       | –       | –       | –       | –       |
| DVB [63]    | 0.403   | 0.422   | 0.446   | –       | 0.604   | 0.632   | 0.665   | –       | –       | –       | –       | –       |
| DistillHash [18] | 0.284   | 0.285   | 0.287   | 0.290   | 0.667   | 0.675   | 0.677   | 0.675   | 0.696   | 0.706   | 0.708   | 0.700   |
| **Ours**    | **0.540** | **0.618** | **0.634** | **0.645** | **0.699** | **0.742** | **0.757** | **0.759** | **0.814** | **0.839** | **0.846** | **0.845** |

**FIGURE 3.** TopN-precision and precision-recall curves on NUSWIDE with 16 and 32 hash bits.
Y. Li et al.: Contrastive Self-Supervised Hashing With Dual Pseudo Agreement

FIGURE 4. TopN-precision and precision-recall curves on FLICKR25K with 16 and 32 hash bits.

FIGURE 5. Hyper-parameter $\alpha$ on CIFAR-10 with different lengths hash codes.

FIGURE 6. Hyper-parameter $\beta$ on CIFAR-10 with different lengths hash codes.

PCAH [58], DSH [59], SpH [60]) and six recently proposed unsupervised deep hashing methods (DeepBit [15], SGH [16], SSDH [17], BGAN [14], DVB [63], DistillHash [18]).

The general MAP results of our method and other methods with different hash codes lengths are shown in Table 1. The first and second best results are highlighted by bold and underline. As can be seen in Table 1, traditional unsupervised hashing methods (ITQ [57] and DSH [59]) sometimes can surpass unsupervised deep hashing methods (DeepBit [15] and SGH [16]). This indicates that some deep hashing methods may achieve unsatisfactory performance by label noisy and over-fitting. However, by leveraging a reduction of risk minimization under noisy labels, deep hashing methods (DistillHash [18] and our method) achieve more promising results. Compared with other unsupervised deep hashing methods, our method consistently achieves the best performance across different lengths of hash codes for all three datasets. Compared with DistillHash [18], the MAP improvement of our method is over 13% in all cases on the FLICKR25K dataset. Furthermore, our method can outperform BGAN [14] over 6% on CIFAR-10 dataset and 2% on NUSWIDE dataset.

Additionally, we also provide the TopN-precision and precision-recall curves for a more comprehensive comparison on NUSWIDE and FLICKR25K datasets. As shown in Fig. 3 and Fig. 4, the curves of our method can consistently above all other curves, which demonstrates that the performance of our method can achieve higher precision than all other methods.
Y. Li et al.: Contrastive Self-Supervised Hashing With Dual Pseudo Agreement

FIGURE 8. Image retrieval results on the CIFAR-10 dataset by using 32 or 48 bits hash codes. For each row, the image on the left-hand side is the query. Retrieved results with top-10 similarity are shown on the right. The samples marked with red borders indicate false positives.

TABLE 2. Our dual pseudo agreement vs. single pseudo label on CIFAR-10.

| Method              | 16 bits | 32 bits | 64 bits | 128 bits |
|---------------------|---------|---------|---------|----------|
| Single label        | 0.418   | 0.432   | 0.457   | 0.461    |
| Dual agreement      | 0.540   | 0.618   | 0.634   | 0.645    |

TABLE 3. Our dual pseudo agreement vs. single pseudo label on NUSWIDE.

| Method              | 16 bits | 32 bits | 64 bits | 128 bits |
|---------------------|---------|---------|---------|----------|
| Single label        | 0.669   | 0.681   | 0.693   | 0.694    |
| Dual agreement      | 0.699   | 0.742   | 0.757   | 0.759    |

D. EFFECTIVENESS OF DUAL PSEUDO AGREEMENT
To validate the effectiveness of dual pseudo agreement mechanism in our self-labelling procedure, we compare our method with the single pseudo method: single pseudo label + hash function learning. The retrieval results about MAP are shown in Table 2 and Table 3. It can be observed that dual pseudo agreement effectively improves the retrieval performance. This happens because more corrupted labels are prevented with the consistency constraint. Moreover, dual pseudo agreement mechanism connects the learning of hash function and binary codes, which makes them more compatible and easier to preserve the updated semantic similarity.

E. PARAMETER SENSITIVITY
To investigate the sensitivity of hyper-parameters $\alpha$ and $\beta$, we conducted experiments with different parameter settings. Fig. 5 and Fig. 6 show the precision score of our method, when $\alpha$ and $\beta$ are within a range at 32 bits and 64 bits on CIFAR-10 dataset. As we can see, for both 32 bits and 64 bits, there is a significant performance drop in terms of MAP when $\alpha$ and $\beta$ become very small or very large. This is because $\alpha$ and $\beta$ are designed to balance the hashing loss and balance loss. Setting $\alpha$ and $\beta$ to small or to large will lead to unbalance between these two terms. The best result is obtained when $\alpha = 0.6$ and $\beta = 0.01$, so we fix $\alpha = 0.6$ and $\beta = 0.01$ in our other experiments.

F. EFFICIENCY OF THE PROPOSED METHOD
In this subsection, we evaluate the efficiency of our method. Specifically, we conduct experiment on CIFAR-10 dataset and compare the encoding time of different unsupervised methods, including: Deepbit [15], BGAN [14], SH [56], DistillHash [18], and our method.

As illustrated in Fig. 7, our method outperforms Deepbit [15], BGAN [14], SH [56], and DistillHash [18]. In particular, our method can run 1.37x faster than the widely used BGAN [14] in testing stage, which suggests our method is indeed fast in real applications.

G. QUALITATIVE RESULTS
To further demonstrate the retrieval performance of our method, we provide some qualitative examples (queries and top-10 matches) on CIFAR-10 dataset in Fig. 8. From Fig. 8 we can see that most of the retrieval examples are true positives results, which demonstrate that our hash features are robust to changes in cropping, scale, and viewpoint. However, there are still some failure cases when the hash feature is distracted by irrelevant objects.

V. CONCLUSION
In this article, we present a contrastive self-supervised method that addresses the fundamental limitations of corrupted labels for unsupervised hash learning. Based on dual pseudo agreement labels, we propose a self-supervised hashing method with mutual information and noise contrastive loss. Throughout the process of hash learning, the stable pseudo labels and data distributions collaboratively work together as teachers to guide the binary codes learning.
process. Extensive experiments on three benchmark datasets show the robustness of the proposed method is superior to many state-of-the-art approaches. Furthermore, the ablation studies clearly demonstrate the reliability of dual pseudo agreement and contrastive loss.

In the near future, we would like to explore new loss functions that not only are effective for sufficient learning but also have theoretically guaranteed robustness further. Furthermore, we would like to test different architectures to further decrease the model size and improve the training speed.

REFERENCES

[1] J. Wang, S. Kumar, and S.-F. Chang, “Semantic-supervised hashing for large-scale search,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[2] X. Lu, L. Song, R. Xie, X. Yang, and W. Zhang, “Deep hash learning for efficient image retrieval,” in Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW), Jul. 2017, pp. 579–584.

[3] Z. Mehmood, M. Rashid, A. Rehman, T. Saba, H. Dawood, and H. Dawood, “Effect of complementary visual words versus complementary features on clustering for effective content-based image search,” J. Intell. Fuzzy Syst., vol. 35, no. 5, pp. 5421–5434, Nov. 2018.

[4] U. Sharif, Z. Mehmood, T. Mahmood, M. A. Javid, A. Rehman, and T. Saba, “Scene analysis and search using local features and support vector machine for effective content-based image retrieval,” Artif. Intell. Rev., vol. 52, no. 2, pp. 901–925, Aug. 2019.

[5] Y. Peng, J. Zhang, and Z. Ye, “Deep reinforcement learning for image hashing,” IEEE Trans. Multimedia, vol. 22, no. 8, pp. 2061–2073, Aug. 2020.

[6] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan, “Instance-aware hashing for multi-label image retrieval,” IEEE Trans. Image Process., vol. 25, no. 6, pp. 2469–2479, Jun. 2016.

[7] L. Jin, X. Shu, K. Li, Z. Li, G.-J. Qi, and J. Tang, “Deep ordinal hashing with spatial attention,” IEEE Trans. Image Process., vol. 28, no. 5, pp. 2173–2186, May 2019.

[8] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, “Supervised hashing for image retrieval via image representation learning,” in Proc. AAAI Conf. Artif. Intell., 2014, pp. 2156–2162.

[9] W. J. Li, S. Wang, and W. C. Kang, “Feature learning based deep supervised hashing with pairwise labels,” in Proc. Int. Joint Conf. Artif. Intell., 2016, pp. 1711–1717.

[10] Y. Li, Z. Miao, M. He, Y. Zhang, and H. Li, “Deep attention residual hashing,” IEICE Trans. Fundam. Electron., Commun. Comput. Sci., vol. E101.A, no. 3, pp. 654–657, 2018.

[11] Y. Li, Z. Miao, J. Wang, Y. Zhang, and H. Li, “Deep discriminative supervised hashing via Siamese network,” IEICE Trans. Inf. Syst., vol. E100.D, no. 12, pp. 3036–3040, 2017.

[12] S. Zhang, J. Li, and B. Zhang, “Semantic cluster unary loss for efficient deep hashing,” IEEE Trans. Image Process., vol. 28, no. 6, pp. 2908–2920, Jun. 2019.

[13] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, “Unsupervised deep hashing with similarity-adaptive and discrete optimization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 12, pp. 3034–3044, Dec. 2018.

[14] J. Song, T. He, L. Gao, X. Xu, A. Hanjalic, and H. T. Shen, “Binary generative adversarial networks for image retrieval,” in Proc. AAAI Conf. Artif. Intell., 2018, pp. 394–401.

[15] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, “Learning compact binary descriptors with unsupervised deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 1183–1192.

[16] B. Dai, R. Guo, S. Kumar, N. He, and L. Song, “Stochastic generative hashing,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 913–922.

[17] E. Yang, C. Deng, T. Liu, W. Liu, and D. Tao, “Semantic structure-based supervised deep hashing,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 1064–1070.

[18] E. Yang, T. Liu, C. Deng, W. Liu, and D. Tao, “Distillhash: Unsupervised deep hashing by distilling data pairs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 2946–2955.

[19] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, “Unsupervised semantic-preserving adversarial hashing for image search,” IEEE Trans. Image Process., vol. 28, no. 8, pp. 4032–4044, Aug. 2019.

[20] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[21] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in Proc. IEEE Int. Conf. Eur. Conf. Comput. Vis., Jun. 2016, pp. 69–84.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[23] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, “Generative adversarial networks: Introduction and outlook,” IEEECACIA I. Automatica Sinica, vol. 4, no. 4, pp. 588–598, 2017.

[24] P. Xiang, L. Wang, F. Wu, J. Cheng, and M. Zhou, “Single-image de-rating with feature-supervised generative adversarial network,” IEEE Signal Process. Lett., vol. 26, no. 5, pp. 650–654, May 2019.

[25] H. Li, Y. Li, X. Xie, S. Gao, and D. Mao, “Pseudo labels and soft multi-part corresponding similarity for unsupervised deep hashing,” IEEE Access, vol. 9, 2020, pp. 53511–53521, 2020.

[26] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saundhi, “A theoretical analysis of contrastive unsupervised representation learning,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 5628–5637.

[27] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. C. Courville, Y. Bengio, and S. Lacoste-Julien, “A closer look at memorization in deep networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 233–242.

[28] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia, “Iterative learning with open-set noisy labels,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8688–8696.

[29] K.-H. Lee, X. He, L. Zhang, and L. Yang, “CleanNet: Transfer learning for scalable image classifier training with label noise,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5447–5456.

[30] T. Liu and D. Tao, “Classification with noisy labels by importance reweighting,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 3, pp. 447–461, Mar. 2016.

[31] L. Jiang, Z. Zhou, T. Zhang, L.-I. Li, and L. Fei-Fei, “MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 2304–2313.

[32] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W.-H. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 8536–8546.

[33] W. Hu, Y. Huang, F. Zhang, and R. Li, “Noise-tolerant paradigm for training face recognition CNNs,” in Proc. IEEE Comput. Vis. Pattern Recognit., Jun. 2019, pp. 11879–11888.

[35] M. Ye and P. C. Yuen, “PurifyNet: A robust person re-identification model with noisy labels,” IEEE Trans. Inf. Forensics Security, vol. 15, pp. 2655–2666, 2020.

[36] Y. Li, J. Yang, M. Song, L. Cao, J. Luo, and J.-L. Li, “Learning from noisy labels with distillation,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 1928–1936.
YANG LI received the M.S. degree from the PLA University of Science and Technology, Nanjing, China, in 2010, and the Ph.D. degree from the Army Engineering University of PLA, Nanjing, in 2018. He is currently an Associate Professor with the Army Engineering University of PLA. His current research interests include computer vision, deep learning, and image processing.

YAPENG WANG received the B.S. degree from the PLA University of Science and Technology, Nanjing, China, in 2015. He is currently pursuing the M.S. degree with the Army Engineering University of PLA, Nanjing. His current research interests include digital image processing and deep learning.

ZHUANG MIAO received the Ph.D. degree from the PLA University of Science and Technology, Nanjing, China, in 2007. He is currently an Associate Professor with the Army Engineering University of PLA, Nanjing. His current research interests include artificial intelligence, pattern recognition, and computer vision.

JIABAO WANG received the Ph.D. degree in computational intelligence from the PLA University of Science and Technology, Nanjing, China, in 2013. He is currently an Assistant Professor with the Army Engineering University of PLA, Nanjing. His current research interests include computer vision and machine learning.

RUI ZHANG received the Ph.D. degree from the PLA University of Science and Technology, Nanjing, China, in 2004. He is currently a Professor with the Army Engineering University of PLA, Nanjing. His current research interests include data engineering and information fusion.

**References**

[43] J. Shu, Q. Zhao, K. Chen, Z. Xu, and D. Meng, “Learning adaptive loss for robust learning with noisy labels,” 2020, arXiv:2002.06482. [Online]. Available: http://arxiv.org/abs/2002.06482

[44] H. Wei, L. Feng, X. Chen, and B. An, “Combating noisy labels by agreement: A joint training method with co-regularization,” 2020, arXiv:2003.02752. [Online]. Available: http://arxiv.org/abs/2003.02752

[45] J. Li, R. Socher, and S. C. H. Hoi, “DivideMix: Learning with noisy labels as semi-supervised learning,” 2020, arXiv:2002.07394. [Online]. Available: http://arxiv.org/abs/2002.07394

[46] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in Proc. IEEE Conf. Eur. Conf. Comput. Vis., Jun. 2018, pp. 132–149.

[47] A. He, C. Luo, X. Tian, and W. Zeng, “A twofold Siamese network for real-time object tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4834–4843.

[48] X. Shu, G.-J. Qi, J. Tang, and J. Wang, “Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation,” in Proc. 23rd ACM Int. Conf. Multimedia, 2015, pp. 35–44.

[49] J. Tang, X. Shu, Z. Li, G.-J. Qi, and J. Wang, “Generalized deep transfer networks for knowledge propagation in heterogeneous domains,” ACM Trans. Multimedia Comput., Commun., Appl., vol. 12, no. 4s, pp. 1–22, Nov. 2016.

[50] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2018, arXiv:1807.03748. [Online]. Available: http://arxiv.org/abs/1807.03748

[51] B. Poole, S. Ozair, A. van den Oord, A. Alemi, and G. Tucker, “On variational bounds of mutual information,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 5171–5180.

[52] X. Luo, P.-F. Zhang, Z. Huang, L. Nie, and X.-S. Xu, “Discrete hashing with multiple supervision,” IEEE Trans. Image Process., vol. 28, no. 6, pp. 2962–2975, Jun. 2019.

[53] Y. Kim, J. Yim, J. Yun, and J. Kim, “NLNL: Negative learning for noisy labels,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2019, pp. 101–110.

[54] M. Abadi et al., “TensorFlow: A system for large-scale machine learning,” in Proc. Symp. Oper. Syst. Design Implement., 2016, pp. 265–283.

[55] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions,” Commun. ACM, vol. 51, no. 1, pp. 117–122, Jan. 2008.

[56] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in Proc. Adv. Neural Inf. Process. Syst., 2009, pp. 1753–1760.

[57] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[58] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma, “AnnoSearch: Image annotation by search,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., Jun. 2006, pp. 1483–1490.

[59] Z. Jin, C. Li, Y. Lin, and D. Cai, “Density sensitive hashing,” IEEE Trans. Cybern., vol. 44, no. 8, pp. 1362–1371, Aug. 2014.

[60] J.-F. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon, “Spherical hashing,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 2957–2964.

[61] K. Zuva and T. Zuva, “Evaluation of information retrieval systems,” Int. J. Comput. Sci. Inf. Technol., vol. 4, no. 3, pp. 35–45, 2012.

[62] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[63] Y. Shen, L. Liu, and L. Shao, “Unsupervised binary representation learning with deep variational networks,” Int. J. Comput. Vis., vol. 127, nos. 11–12, pp. 1614–1628, Dec. 2019.