Semi-supervised Generative Adversarial Hashing for Image Retrieval

Guan'an Wang[1,3], Qinghao Hu[2,3], Jian Cheng[2,3,4], and Zengguang Hou[1,3,4](✉)

[1] The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{wangguanan2015,zengguang.hou}@ia.ac.cn

[2] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{qinghao.hu,jcheng}@nlpr.ia.ac.cn

[3] University of Chinese Academy of Sciences, Beijing, China

[4] Center for Excellence in Brain Science and Intelligence Technology, Beijing, China

Abstract. With explosive growth of image and video data on the Internet, hashing technique has been extensively studied for large-scale visual search. Benefiting from the advance of deep learning, deep hashing methods have achieved promising performance. However, those deep hashing models are usually trained with supervised information, which is rare and expensive in practice, especially class labels. In this paper, inspired by the idea of generative models and the minimax two-player game, we propose a novel semi-supervised generative adversarial hashing (SSGAH) approach. Firstly, we unify a generative model, a discriminative model and a deep hashing model in a framework for making use of triplet-wise information and unlabeled data. Secondly, we design novel structure of the generative model and the discriminative model to learn the distribution of triplet-wise information in a semi-supervised way. In addition, we propose a semi-supervised ranking loss and an adversary ranking loss to learn binary codes which preserve semantic similarity for both labeled data and unlabeled data. Finally, by optimizing the whole model in an adversary training way, the learned binary codes can capture better semantic information of all data. Extensive empirical evaluations on two widely-used benchmark datasets show that our proposed approach significantly outperforms state-of-the-art hashing methods.

Keywords: Information retrieval · Hashing · Deep learning · GANs

1 Introduction

With explosive growth of image and video data on the Internet, large-scale image retrieval task has attracted more and more attention in recent years. One of traditional methods applied to this task is Nearest Neighbor Search (NNS), where the first k images with the smallest distance between the query one are returned

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11219, pp. 491–507, 2018.
https://doi.org/10.1007/978-3-030-01267-0_29

as results. However, for large-scale images with high-dimensional feature, NNS is extremely expensive in terms of space and time. Hashing technique [25,26] is a popular Approximate Nearest Neighbor Search due to its both computation efficiency and high retrieval accuracy by calculating the Hamming distance between binary codes.

Hashing methods can be mainly grouped into traditional hashing methods and deep hashing methods. In traditional hashing methods, images are firstly represented with the hand-crafted visual descriptors (e.g. SIFT [15], GIST [20], HOG [3]), and then hash functions and quantization algorithm are separately learned to encode the features into binary codes. Based on whether the supervised information is adopted, traditional methods can be categorized into unsupervised hashing models (LSH [4], SH [29], ITQ [10], AGH [14]) and supervised hashing models (SSH [27], BRE[8], MLH [18], KSH [13], SDH [24]). Deep hashing methods (CNNH [30], NINH [9], DPSH [12], DHN [35], DSDH[11]) simultaneously learn feature representation and hash functions based on deep networks and usually are trained with supervised information. Due to its powerful ability of feature representation and nonlinear mapping, deep hashing methods have shown their better performance than traditional ones.

Although encouraging performance reported in the models above, obtaining labeled data is expensive. Contrarily, unlabeled data is always enough and free. Thus, semi-supervised hashing method is a good solution where a small amount of labeled data and lots of unlabeled data are utilized to learn better binary codes. Semi-supervised Hashing (SSH) [27] is proposed to minimize the empirical error over labeled data and maximize the information entropy of binary codes over both labeled and unlabeled data. However, SSH is implemented without deep networks, which leads to unsatisfying performance compared with deep hashing methods. Deep Semantic Hashing with GANs (DSH-GANs) [21] minimizes the empirical error over synthetic data generated conditioned on class labels based on deep architecture. However, class labels are more difficult and expensive to obtain than triplet-wise labels [9]. Semi-supervised Deep Hashing (SSDH) [34] and Deep Hashing with a Bipartite Graph (BGDH) use graph structure to model unlableled data. However, constructing the graph model of large-scale data is extremely expensive in terms of time and space, and using batch data instead may lead to a suboptimal result.

To solve the problem above, we propose a novel semi-supervised generative adversarial hashing (SSGAH), which utilizes a generative model to model unlabeled data and uses triplet-wise labels as supervised information. Specifically, our SSGAH includes a generative model, a discriminative model and a deep hashing model, where all three models are optimized together in an adversarial framework. The generative model can well learn the triplet-wise information in a semi-supervised way. Benefiting from both adversary learning and the generative model, the deep hashing model is able to learn binary codes which not only preserve semantic similarity but also capture the meaningful triplet-wise information. Main contributions of our proposed approach are outlined as below:

(1) We propose a novel semi-supervised generative adversarial hashing (SSGAH) approach to make full use of triplet-wise information and unlabeled data. Our approach unifies generative, discriminative and deep hashing models in an adversarial framework, where the generative and discriminative models are carefully designed for capturing the distribution of triplet-wise information in a semi-supervised way, all of which contribute to semantic preserving binary codes.

(2) We propose novel semi-supervised ranking loss and adversary ranking loss to learn better binary codes that capturing semantic information of both labeled and unlabeled data. For semi-supervised ranking loss, we propose to preserve relative similarity of real and synthetic samples. For adversary ranking loss, we propose to make the deep hashing model and generative model improve each other in a two-player minimax game.

(3) Extensive evaluations on two widely-used datasets demonstrate that our SSGAH approach significantly outperforms the state-of-the-art methods, and component analysis verifies the effectiveness of each part of our model.

2 Related Work

Traditional Hashing Methods. Conditioning whether labeled data is used in training process, traditional hashing methods can be divided into unsupervised and supervised ones. Unsupervised hashing methods employ unlabeled data even no data. Local Sensitive Hashing (LSH) [4] uses random linear projections to map similar samples to nearby binary codes. Spectral Hashing (SH) [29] tries to keep hash functions balanced and uncorrelated. Iterative Quantization (ITQ) [10] proposes an alternating minimization algorithm to minimize the quantization error. Anchor Graph Hashing (AGH) [14] preserves the neighborhood structures by anchor graphs. Supervised hashing methods utilize labeled information to improve binary codes. Semi-supervised Hashing (SSH) [27] minimizes the empirical error and maximizes information entropy. Binary Reconstruction Embedding (BRE) [8] minimizes the reconstruction error between original distances and reconstructed distances in the Hamming space. Minimal Loss Hashing (MLH) [18] minimizes loss between the learned Hamming distance and quantization error. Supervised Hashing with Kernels (KSH) [13] utilizes the equivalence between optimizing codes inner products and Hamming distance. Supervised Discrete hashing (SDH) [24] integrates hash codes generation and classifier training.

Deep Hashing Methods. Recently, deep learning has shown its powerful ability in various domains including classification, object detection, semantic segmentation and so on. Inspired by powerful feature representation learning of deep networks, many hashing methods based on deep networks have been proposed. Most deep hashing methods are trained in a supervised way. Convolutional Neural Network Hashing (CNNH) [30] firstly learns binary codes by similarity matrix decomposition, then utilizes convolutional neural networks to simultaneously learn good feature representation and hash functions guided by those binary codes. Network in Network Hashing (NINH) [9] straightly learns feature representation and hash functions for binary codes which preserve relative similarity of raw samples in an end-to-end manner. Deep Pairwise-Supervised Hashing (DPSH) [12] performs simultaneous feature learning and binary codes learning with pair-wise labels. Deep Hashing Network (DHN) [35] simultaneously optimizes pair-wise cross entropy loss and pair-wise quantization loss to remit quantization error. Deep Supervised Discrete Hashing (DSDH) [11] straightly optimize binary codes without relaxation by proposing a iterative optimization algorithm. Supervised Semantics-preserving Deep Hashing [32] constructs hash functions as a latent layer in a deep network and trains the model over classification error. Semi-supervised Deep Hashing (SSDH) [34] and BGDH [31] minimizes empirical error on labeled data and exploits unlabeled data through a graph construction method. Different from SSDH and BGDH, which use the graph to model unlabeled data and is time and space consuming, our approach utilizes generative models to learn unlabeled data. DSH-GANs [21] utilizes class labels to train AC-GANs [19] for synthetic images, and then use those synthetic images to train a deep hashing model. Different from DSH-GANs, our approach utilizes triplet-labels which are more common, and specifically designs a GANs model which can be well learned with limited supervised information. What's more, we import adversarial learning between the generative model and the deep hashing model for better binary codes, while DSH-GANs not.

Generative Models. The generative model is a kind of important model in machine learning, which can understand data by learning its distribution. Goodfellow proposes an efficient yet straightforward framework for estimating generative models named Generative Adversarial Networks (GANs) [5] by making a generative model and a discriminative model play a two-player minimax game. Conditional Generative Adversarial Networks (cGANs) [17] extends GANs to its conditional version by using class labels to limit the generator and the discriminator. Different from existed generative models, the generative model in our approach is particularly designed for triplet-wise information and unlabeled data.

3 Semi-supervised Generative Adversarial Hashing

Given a dataset $\mathcal{X}$ that is composed of unlabeled data $\mathcal{X}^u = \{x_i^u | i = 1, \ldots, m\}$ in form of individual images and labeled data $\mathcal{X}^l = \{(x^q, x^p, x^n)_i | i = 1, \ldots, n\}$ with triplet-wise information where the query image x^q is more similar to positive one x^p than to negative one x^n , our primal goal is to learn a mapping function $\mathcal{B}(\cdot)$ which encodes the input image $x \subset \mathcal{X}$ into k-bit binary codes $\mathcal{B}(x) \in \{0, 1\}^k$, while preserves relative semantic similarity of images in the dataset $\mathcal{X}$.

As shown in Fig. 1, our SSGAH approach consists of a generative model G, a discriminative model D and a deep hashing model H. The generative model is to learn the distribution of triplet-wise information in a semi-supervised way and generates synthetic triplets $\{(x, x_{syn}^q, x_{syn}^n)_i | i = 1, 2, \cdots, m + n\}$, where the real

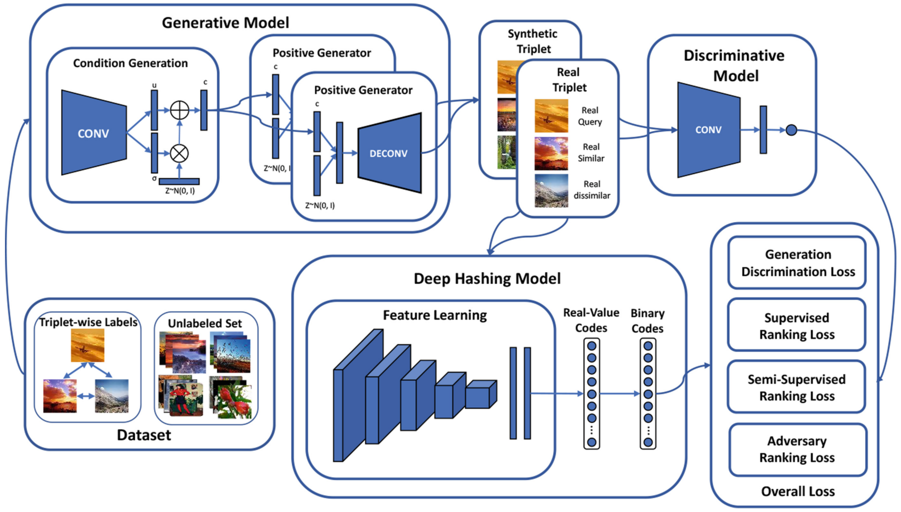

Fig. 1. Overview of our proposed SSGAH approach, where the curves indicate the data flow and the straight lines indicate the fully connected layers. Our model consists of a generative model, a discriminative model and a deep hashing model. The generative model is to capture the distribution of triplet-wise information and unlabeled data, the discriminative model is to distinguish the real triplets from synthetic ones, the deep hashing model is to learn binary codes which not only preserve semantic similarity but also can identificate small difference between real and synthetic images. The whole model is trained in an adversary way.

image x is more similar to the synthetic image x^q_{syn} than to another synthetic one x^n_{syn}. The discriminative model is to distinguish the real triplets from synthetic ones. The deep hashing model is to learn binary codes which preserve semantic similarity on the whole dataset. All three models are trained together in an adversary way.

3.1 Model Architecture Structure

Generative and Discriminative Models. The goals of our generative and discriminative models are to learn the distribution of unlabeled data $\mathcal{X}^u = \{x^u_i | i = 1, 2, \ldots, n\}$ and labeled data $\mathcal{X}^l = \{(x^q, x^p, x^n)_i | i = 1, 2, \ldots, n\}$, and then synthesize realistic meaningful triplets. Specifically, given a real sample $x \in \{\mathcal{X}^u, \mathcal{X}^l\}$, a synthetic triplet $(x, x^p_{syn}, x^n_{syn})$ should be generated, where x is more similar to the positive synthetic one x^p_{syn} than to negative synthetic one x^n_{syn}, and both synthetic ones are realistic. Compared with common conditionally image generation tasks, several difficulties exist in ours. Firstly, in common tasks, their conditions are low-dimensional and meaningful (e.g. class labels) [17,19], or their real and synthetic images share similar structure [6,33]. In our task, the conditions are raw images which are high-dimensional and noisy, and our synthetic images share only semantic information with raw ones, neither structure nor texture. What's more, our labeled data is limited.

To mitigate the problem above, we propose novel generative and discriminative models, whose architecture are shown in Fig. 1. Firstly, to filter the noise and produce meaningful conditions, images are feed into a stack of convolutional layers followed by a fully connected layer to extract short, meaningful features ν. Secondly, in order to ease the lack of labeled data, the final conditions are randomly sampled from an independent Gaussian distribution $\mathcal{N}(\mu(\nu), \Sigma(\nu))$, where the mean $\mu(\nu)$ and diagonal covariance matrix $\Sigma(\nu)$ are learned from the ν. What's more, for better understanding of the semantic relationship of triplets, we improve the discriminative model to distinguish real triplets (x, x^p, x^n) from synthetic ones $(x, x^p_{syn}, x^n_{syn})$, and design extra triplets (x, x^n, x^p) as negative samples beside synthetic triplets.

The architecture of the two models are shown in Fig. 1. Firstly, a condition vector c is generated through the condition generation module given an real image. Secondly, a random vector z sampled from standard normal distribution concatenated with the generated condition is feed into the positive generator G_p and the negative generator G_n to generate triplets $(x, x^p_{syn}, x^n_{syn})$ where the x is more similar to x^p_{syn} than to x^n_{syn}. Then, the discriminative model determines the probability that input triplets are real. Finally, the generative model and the discriminative model can be optimized by playing a two-player minimax game with value function $\mathcal{L}_{GD}$.

$$\begin{aligned}\underset{G}{min}\ \underset{D}{max}\ \mathcal{L}_{GD} = & E_{(x^q,x^p,x^n)\in\mathcal{X}^l}\{logD(x^q, x^p, x^n) + log[1 - D(x^q, x^n, x^p)]\} \\ & + E_{x\in\{\mathcal{X}^u,\mathcal{X}^l\}}\{log[1 - D(x, G_p(x), G_n(x))]\} \\ & + D_{KL}(\mathcal{N}(\mu(\nu), \Sigma(\nu)) \mid\mid \mathcal{N}(0, I))\end{aligned} \tag{1}$$

where $D(\cdot,\cdot,\cdot)$ is the probability that input triplet is from labeled data $\mathcal{X}^l$, $G_p(x)$ and $G_n(x)$ is the synthetic images generated by the two generators, D_{KL} is a regularization term which means the Kullback-Leibler divergence between standard Gaussian distribution and conditioning Gaussian distribution.

Deep Hashing Model. For easy comparison with other hashing algorithms, we adopt AlexNet [7] as our basic network. AlexNet contains 5 convolutional layers $(conv1-conv5)$ with max-pooling operations followed by 2 fully connected layers $(fc6 - fc7)$ and an output layer. In the convolutional layers, units are organized into feature maps and are connected locally to patches in the outputs of the previous layer. The fully connected layers $(fc6 - fc7)$ are activated by rectified linear units (Relu) for faster training.

AlexNet is designed particularly for multi-class classification task, so the amount of units in the output layer is equal to class amounts. To adapt AlexNet to our deep hashing architecture, we replace the output layer with a fully connected layer f_h and activate it by a sigmoid function, through which the high dimensional feature of the $fc7$ layer can be projected to k-bits hash real-value in $[0, 1]$. The formulation is in Eq. (2), where $f(x)$ is the feature representation in $fc7$ layer of AlexNet, W^h and b^h denote weights and bias in hash layer f_h, σ is sigmoid function. Since the output of the neural network is continuous, we transfer the $\mathcal{H}(x) \in [0, 1]^k$ to binary codes $\mathcal{B}(x) \in \{0, 1\}^k$ with Eq. (3)

$$\mathcal{H}(x) = \sigma(f(x)W^h + b^h) \tag{2}$$

$$\mathcal{B}(x) = (sgn(\mathcal{H}(x) - 0.5) + 1)/2 \tag{3}$$

3.2 Objective Function

Existing deep hashing methods usually design the objective function to preserve the relative semantic similarity of samples in labeled data, but ignore the unlabeled data. To address the problem, we propose novel semi-supervised ranking loss and adversary ranking loss to exploit the relative similarity of samples in both triplet-wise label and unlabeled data. By jointly minimizing the supervised ranking loss, semi-supervised ranking loss, as well as adversary ranking loss, the learned binary codes can better capture semantic information of all data.

Supervised Ranking Term. For most existing hashing methods, class labels [32], pair-wise labels [30], and triplet-wise labels [9] are most frequently used as supervised information. Among the three kinds of labels, class labels contain the most accurate information, followed by pair-wise ones and triplet-wise ones. In contrast, the most easily available labels are triplet-wise labels, followed by pair-wise ones and class ones [9]. Considering easy acquirement in practice, we choose triplet-wise labels as our supervised information. Specially, given labeled data $\mathcal{X}_s = \{(x_i^q, x_i^p, x_i^n) | i = 1, \ldots, n\}$, the supervised ranking loss can be formulated in Eq. (4), where $|| \cdot ||_H$ denotes Hamming distance, $\mathcal{B}(x)$ is the binary codes of x, and m_{sr} is the margin between match pairs and the mismatch pairs.

$$\begin{aligned} \underset{H}{min}\, \hat{\mathcal{L}}_{sr} &= \sum_{i=1}^{n} \hat{\mathcal{L}}_{triplet}(m_{sr}, (x^q, x^p, x^n)_i) \\ &= \sum_{i=1}^{n} max(0, m_{sr} - (||\mathcal{B}(x^q) - \mathcal{B}(x^n)||_H - ||\mathcal{B}(x^q) - \mathcal{B}(x^p)||_H)_i) \end{aligned} \tag{4}$$

Semi-supervised Ranking Term. Training the deep hashing model solely based on supervised information usually leads to an unsatisfying result for that limited labeled data can't accurately reflect similarity relation of samples in unlabeled data. In order to address the problem, we propose to leverage Generative Adversarial Networks (GANs) to learn distribution of real data, which is composed of limited labeled data and lots of unlabeled data, and in return synthetic samples generated by GANs are used to train the deep hashing model to learn better feature representation and more discriminative binary codes.

Accordingly, we propose a novel semi-supervised ranking loss. On the one hand, to learn more discriminative binary codes, we use a synthetic sample x_{syn}^p which is similar with the query one x^q in a real triplet or a synthetic sample x_{syn}^n which is dissimilar with x^q to replace corresponding real one. Through this method, labeled data can be augmented without losing supervision information. On the other hand, for better utilizing the unlabeled data, given an unlabeled sample, we generate a synthetic triplet where the given real sample is more similar to a synthetic positive sample than to a synthetic negative one.

Specifically, given a real triplet (x^q, x^p, x^n), we can get synthetic triplets (x^q, x^p_{syn}, x^n) and (x^q, x^p, x^n_{syn}), where x^p_{syn} and x^n_{syn} are generated by positive generator G_p and negative generator G_n respectively conditioned on real sample x^q and random noise z. For a real unlabeled sample $x^u \in \mathcal{X}_u$, similar procedure can be performed to generate a synthetic triplet $(x^u, x^p_{syn}, x^n_{syn})$. Hence the semi-supervised ranking term can be defined in Eq. (5), where $\hat{\mathcal{L}}_{triplet}(\cdot, (\cdot, \cdot, \cdot))$ is defined in Eq. (4).

$$\begin{aligned} \min_H \hat{\mathcal{L}}_{ssr} = & \sum_{i=1}^{n} [\hat{\mathcal{L}}_{triplet}(m_{ssr}, (x^q, x^p_{syn}, x^n)_i) + \hat{\mathcal{L}}_{triplet}(m_{ssr}, (x^q, x^p, x^n_{syn})_i)] \\ & + \sum_{i=1}^{m} \hat{\mathcal{L}}_{triplet}(m_{ssr}, (x^u, x^p_{syn}, x^n_{syn})_i) \end{aligned} \tag{5}$$

Adversary Ranking Term. Wang et al. [28] has shown simultaneously learning a generative retrieval model and a discriminative retrieval model in an adversary way is able to improve both models and achieve a better performance than separately training them. Inspired by the idea, we also introduce the idea of minimax two-player game between the generative and deep hashing models and propose a novel adversary ranking loss. Specifically, in the minimax two-player game, the deep hashing model try to learn binary codes that can identificate small difference between (x, x^p) and (x, x^p_{syn}), and the generative model is to make the binary codes of x, x^p and x^p_{syn} distinguishable. Given real triplets $\{(x^q, x^p, x^n)_i | i = 1, 2, \ldots, n\}$ and corresponding synthetic triplets $\{(x^q, x^p_{syn}, x^n_{syn})_i | i = 1, 2, \ldots, n\}$, the minimax two-player game can be formulated in Eq. (6), where $\hat{\mathcal{L}}_{triplet}(\cdot, (\cdot, \cdot, \cdot))$ is defined in Eq. (4).

$$\min_H \max_G \hat{\mathcal{L}}_{ar} = \sum_{i=1}^{n} \hat{\mathcal{L}}_{triplet}(m_{ar}, (x^q, x^p, x^p_{syn})) \tag{6}$$

3.3 Overall Objective Function and Adversary Learning

The overall objective function of our semi-supervised generative adversarial hashing approach integrates loss in Eq. (1), supervised ranking loss in Eq. (4), semi-supervised ranking loss in Eq. (5) and adversary ranking loss in Eq. (6). Hence, the overall objective function $\hat{\mathcal{L}}$ can be formulated in Eq. (7).

$$\min_G \max_{D,H} \hat{\mathcal{L}} = \mathcal{L}_{GD} - \hat{\mathcal{L}}_{sr} - \hat{\mathcal{L}}_{ssr} - \hat{\mathcal{L}}_{ar} \tag{7}$$

Considering the the mapping function $\mathcal{B}(\cdot)$ is discrete and Hamming distance $||\cdot||_H$ is not differentiable, natural relaxation are utilized on Eq. (7) by changing the integer constraint to a range constraint and replacing Hamming distance with Euclidean Distance $||\cdot||_2$. Using the supervised ranking term as an example, relaxed term $\mathcal{L}_{sr}$ is in Eq. (8).

$$\begin{aligned}\mathcal{L}_{sr} &= \sum_{i=1}^{n} \mathcal{L}_{triplet}(m_{sr}, (x^q, x^p, x^n)_i) \\ &= \sum_{i=1}^{n} max(0, m_{sr} - (||\mathcal{H}(x^q) - \mathcal{H}(x^n)||_2^2 - ||\mathcal{H}(x^q) - \mathcal{H}(x^p)||_2^2)_i)\end{aligned} \tag{8}$$

Then, the relaxed semi-supervised ranking loss $\mathcal{L}_{ssr}$ and adversary ranking loss $\mathcal{L}_{ar}$ and be derived similarly. Finally, we apply mini-batch gradient descent, in conjunction with back propagation [16] to network training in Eq. (9).

$$\underset{G}{min} \ \underset{D,H}{max} \ \mathcal{L} = \mathcal{L}_{GD} - \mathcal{L}_{sr} - \mathcal{L}_{ssr} - \mathcal{L}_{ar} \tag{9}$$

3.4 Image Retrieval

After the optimization of SSGAH, one can compute binary codes of a new image and find its similar images. Firstly, a query image x is fed into the deep hashing model and real-value codes $\mathcal{H}(x)$ can be obtained through Eq. (2). Secondly, binary codes $\mathcal{B}(x)$ can be calculated by quantization process via Eq. (3). Finally, the retrieval list of images is produced by sorting the Hamming distances of binary codes between the query image and images in search pool.

4 Experiment

4.1 Dataset

We conduct our experiments on two widely-used datasets, namely CIFAR-10 and NUS-WIDE. CIFAR-10 is a small image dataset including 60,000 32 × 32 color images in 10 categories with 6000 images per class. NUS-WIDE [2] contains nearly 270,000 images collected from Flickr associated with one or multiple labels in 91 semantic concepts. For NUS-WIDE, we follow [9] to use the images associated with the 21 most frequent concepts, where each of these concepts associates with at least 5,000 images.

Following [9,30], we randomly sample 100 images per class to construct query set, and the others are as the base set. In training process, we randomly sample 500 images per class from the base set as labeled data, and the others are as unlabeled data. Triplets are generated from the labeled set conditioned on corresponding labels. Specifically, (x^q, x^p, x^n) are constructed where x^q shares at least one label with x^p and no label with x^n.

4.2 Evaluation Protocol and Baseline Methods

We adopt mean Average Precision (mAP) to measure the performance of hashing methods, and mAP on the NUS-WIDE dataset is calculated with the top 5,000 returned neighbors. Based on the evaluation protocol, we compare our SSGAH with nine state-of-the-art hashing methods, including four traditional hashing methods LSH [4], SH [29], ITQ [10], SDH [24], two supervised deep hashing methods CNNH [30], NINH [9], and three semi-supervised deep hashing methods DSH-GANs [21], SSDH [34] and BGDH [31].

Following the settings in [9], hand-crafted features for traditional hashing methods are presented by 512-dimensional GIST [20] features in the CIFAR-10 dataset and by 500-dimensional bag-of-words features in the NUS-WIDE dataset. Besides, for a fair comparison between traditional and deep hashing methods, we also construct traditional methods on features extracted from the $fc7$ layer of AlexNet which is pre-trained on ImageNet. For deep hashing methods, we adopt raw pixels as input.

4.3 Implementation Details

We implement our SSGAH based on the open-source Tensorflow [1] framework. The generative and discriminative models are implemented and optimized under the guidance of DCGANs [22]. Specifically, we use fractional-strided convolutions and ReLU activation for the generative model, strided convolutions and Leaky ReLU activation for the discriminative model, and both models utilize batch normalization and are optimized by ADAM with learning rate 0.0002 and β_1

Table 1. Mean Average Precision (mAP) scores for different methods on two datasets. The best mAP scores are emphasized in boldface. Note that the mAP scores on NUS-WIDE dataset is calculated based on the top 5,000 returned neighbors.

Methods	CIFAR-10				NUS-WIDE			
	12bits	24bits	32bits	48bits	12bits	24bits	32bits	48bits
SSGAH(*Ours*)	**0.819**	**0.837**	**0.847**	**0.855**	**0.835**	**0.847**	**0.859**	**0.865**
BGDH	0.805	0.824	0.826	0.833	0.803	0.818	0.822	0.828
SSDH	0.801	0.813	0.812	0.814	0.773	0.779	0.778	0.778
DSH-GANs	0.745	0.789	0.793	0.811	0.807	0.820	0.831	0.834
NINH	0.535	0.552	0.566	0.558	0.581	0.674	0.697	0.713
CNNH	0.439	0.476	0.472	0.489	0.611	0.618	0.625	0.608
SDH+CNN	0.363	0.528	0.529	0.542	0.520	0.507	0.591	0.610
ITQ+CNN	0.212	0.230	0.234	0.240	0.728	0.707	0.689	0.661
SH+CNN	0.158	0.157	0.154	0.151	0.620	0.611	0.620	0.591
LSH+CNN	0.134	0.157	0.173	0.185	0.438	0.586	0.571	0.507
SDH	0.255	0.330	0.344	0.360	0.414	0.465	0.451	0.454
ITQ	0.162	0.169	0.172	0.175	0.452	0.468	0.472	0.477
SH	0.124	0.125	0.125	0.126	0.433	0.426	0.426	0.423
LSH	0.116	0.121	0.124	0.131	0.404	0.421	0.426	0.441

0.5. The hyper-parameters m_{sr}, m_{ssr}, m_{ar} are set $\frac{k}{4}$, $\frac{k}{8}$ and 1 respectively via cross validation, where k is code length. The deep hashing model is optimized by stochastic gradient descent with learning rate 0.0001 and momentum 0.9. The mini-batch size of images is 64. For faster convergence, we firstly train the generative and discriminative models under the Eq. (1), and then optimize the whole model under the Eq. (9).

4.4 Experiment Results and Analysis

The mAP scores of hashing methods on CIFAR-10 and NUS-WIDE datasets with different code length k are shown in Table 1. From Table 1 we can see that our proposed SSGAH substantially outperforms the other methods. Specifically, the performance of traditional hashing method with hand-crafted features is poor, where SDH achieves only 36.0% on CIFAR-10 datasets, and ITQ achieves only 47.7% on the NUS-WIDE dataset. Traditional methods with CNN features achieve better performance, which shows that features learned from deep neural networks capture more semantic information.

Among deep methods except ours, the best performance on the two datasets are achieved by BGDH [31] and DSH-GANs [21] with 83.3% and 83.4% respectively. The semi-supervised methods (BGDH, SSDH, DSH-GANs) outperforms the supervised ones (NINH, CNNH), which demonstrates that unlabeled data indeed improves the performance of binary codes. Finally, our SSGAH approach outperforms all methods on two datasets by 2.2% and 3.1% and achieves 85.5% and 86.5% correspondingly.

Note that the two graph-based models BGDH and SSDH achieve good performance on CIFAR-10, but a common performance on NUS-WIDE, DSH-GANs obtains an opposite result, and our SSGAH achieves the best performance on both datasets. The reason may be that compared with complex multi-label images on NUS-WIDE, a graph is easier to capture the structure of simple images

Table 2. Mean Average Precision (mAP) scores under retrieval of unseen classes on two datasets. The best mAP scores are emphasized in boldface. Following settings in [34], mAP scores are calculated based on all returned images.

Methods	CIFAR-10				NUS-WIDE			
	12bits	24bits	32bits	48bits	12bits	24bits	32bits	48bits
SSGAH(*Ours*)	**0.309**	**0.323**	**0.341**	**0.339**	**0.539**	**0.553**	**0.565**	**0.579**
SSDH	0.285	0.291	0.311	0.325	0.510	0.533	0.549	0.551
NINH	0.241	0.249	0.253	0.272	0.484	0.483	0.485	0.487
DRSCH	0.219	0.223	0.242	0.251	0.457	0.464	0.469	0.460
CNNH	0.210	0.225	0.227	0.231	0.445	0.463	0.471	0.477
SDH+CNN	0.185	0.193	0.199	0.213	0.471	0.490	0.489	0.507
ITQ+CNN	0.157	0.165	0.189	0.201	0.488	0.493	0.508	0.503
SH+CNN	0.134	0.127	0.126	0.124	0.416	0.386	0.380	0.379
LSH+CNN	0.107	0.119	0.125	0.138	0.341	0.358	0.371	0.373

on CIFAR-10. Contrarily, DSH-GANs is able to generate meaningful and plentiful images conditioned on multi-label on the NUS-WIDE dataset, but easily lead to lack of diversity of synthetic samples conditioned on limited discrete labels on the CIFAR-10 dataset. Different from graph and DSH-GANs, our approach extracts continuous conditions through a Condition Generation module, and can well capture the distribution of triplet-wise information. The experiments on two datasets demonstrate the efficiency of our SSGAH approach.

4.5 Retrieval of Unseen Classes

To further evaluate our SSGAH approach, we additionally adopt the evaluation protocol from [23], where 75% of classes are known during the training process, and the remaining 25% classes are used to for evaluation. Specifically, the dataset is divided into four folds $train75$, $test75$, $train25$ and $test25$, where the $set75$ ($train75 + test75$) includes data of 75% classes, the $set25$ ($train25 + test25$) contains data of 25% classes, and data amount in $train$ and $test$ set are the same. Following settings in [34], we use $train75$ as training set, $test25$ as query set, and $train25 + test75$ as database set.

The specific experiment settings are as below. The $set75$ of CIFAR-10 and NUS-WIDE consist of 7 classes and 15 classes respectively, results are calculated by the average of 5 different random splits, mAP scores are calculated based on all returned images, and the non-deep methods use features extracted from $fc7$ layer of pre-trained AlexNet as inputs.

The mAP scores under the retrieval of unseen classes are shown in Table 2. As we can see, the gaps between unsupervised methods and supervised ones reduce under retrieval of unseen classes, which is because the unsupervised methods learn on the whole dataset and own better generalization performance, but the supervised methods easily overfit labeled data. SSDH achieves a good performance, which demonstrates that unlabeled data can improve the binary codes. Our SSGAH approach achieves the best result when retrieving unseen classes, which is because the generative model in our framework can capture triplet-wise information of unlabeled data, and our semi-supervised ranking loss and adversary ranking loss can make the learned binary codes not only preserve the semantic similarity of labeled data but also capture underlying relationship of data. Thus our approach achieves better generalization performance to unseen class.

Table 3. Mean Average Precision scores (mAP) under different components of our model.

Methods	CIFAR-10				NUS-WIDE			
	12bits	24bits	32bits	48bits	12bits	24bits	32bits	48bits
SSGAH	**0.819**	**0.837**	**0.847**	**0.855**	**0.835**	**0.847**	**0.859**	**0.865**
w/ssr	0.799	0.819	0.836	0.846	0.810	0.819	0.834	0.835
w/ar	0.776	0.804	0.820	0.829	0.787	0.794	0.810	0.812
baseline	0.744	0.771	0.782	0.789	0.759	0.780	0.794	0.803

4.6 Component Analysis

To further analyze the affect of each component in our SSGAH, we report the results of two variants of our model and a baseline method. For simplicity, we use G, D and H to represent the generative model, discriminative model and deep hashing model respectively. For the baseline method, we only train H under the supervised ranking loss $\mathcal{L}_{sr}$. For the first variant, we train the G, D and H together, but remove the semi-supervised ranking loss $\mathcal{L}_{ssr}$ from Eq. (9), and mark it as *w/ ar*. For the second variant, we first train the G and D together under Eq. (1), and then train H under the supervised ranking loss $\mathcal{L}_{sr}$ and semi-supervised ranking loss $\mathcal{L}_{ssr}$, and mark it as w/ssr. Finally, SSGAH achieves the best performance, which demonstrates the effectiveness of our proposed approach.

As shown in Table 3, the best method is SSGAH, followed by w/ssr, w/ar and *baseline*. Firstly, w/ar improves the *baseline* by 3.2% $\sim$ 4.0% and 0.9% $\sim$ 2.8% on CIFAR-10 and NUS-WIDE datasets, which shows that the adversary ranking loss $\mathcal{L}_{ar}$ helps for better binary codes. Secondly, w/ssr improves the *baseline* by 4.8% $\sim$ 5.7% and 3.2% $\sim$ 5.1% on CIFAR-10 and NUS-WIDE datasets, which shows that H can capture the triplet-wise information and the semi-supervised ranking loss $\mathcal{L}_{ssr}$ can significantly improve the binary codes.

4.7 Effect of Supervision Amounts

To further analyze our proposed semi-supervised generative adversarial hashing approach, we report the results of SSGAH and *baseline* (illustrated in Section 4.6) with different supervision amounts on the CIFAR-10 and NUS-WIDE datasets. As shown in Fig. 2, our SSGAH always outperforms the *baseline*, which demonstrates the effectiveness of our approach. What's more, the difference between the two models increases as the supervision amount decreases, which shows that our SSGAH can better utilize the unlabeled data to improve the binary codes.

4.8 Visualization of Synthetic Images

Figure 3 displays the synthetic triplets generated by our SSGAH (green) and its two variants (blue and red). As we can see, our SSGAH can generate color images with size ranging from 32×32 to 64×64. On both datasets, the synthetic images (green) are clear and meaningful, which are indistinguishable from real images. What's more, they successfully acquire the triplet-wise information, i.e. x is more similar to x^p_{syn} than to x^n_{syn}.

Besides the phenomenons above, some extra phenomenons can be observed. Firstly, the red synthetic images are noisy and meaningless and fail to constitute useful triplet, which show that vanilla generative model is hard to capture the distribution of triplet-wise information with limited labeled data. Secondly, the blue images are meaningful, and x are more similar to x^p_{syn} than to x^n_{syn}, which show that our condition generation part contributes to understanding the

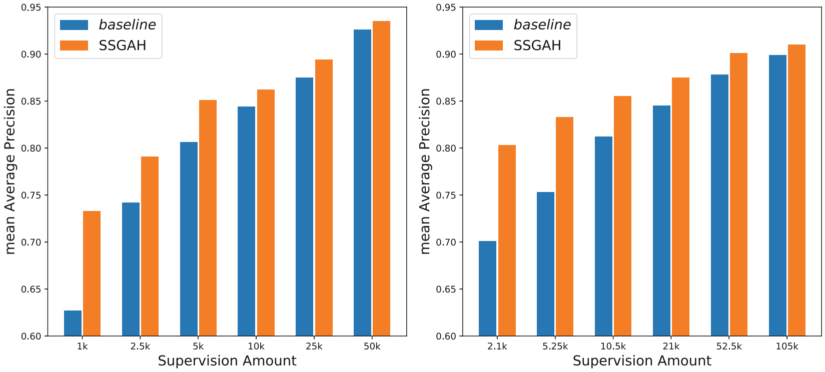

Fig. 2. Mean Average Precision (mAP) scores @48bits of SSGAH and *baseline* with different supervision amounts on the CIFAR10 (left) and NUS-WIDE (right) datasets. Note that our SSGAH always outperforms *baseline*, and the difference between the two models increase as the supervision amount decrease, both of which verify the effectiveness of our proposed approach.

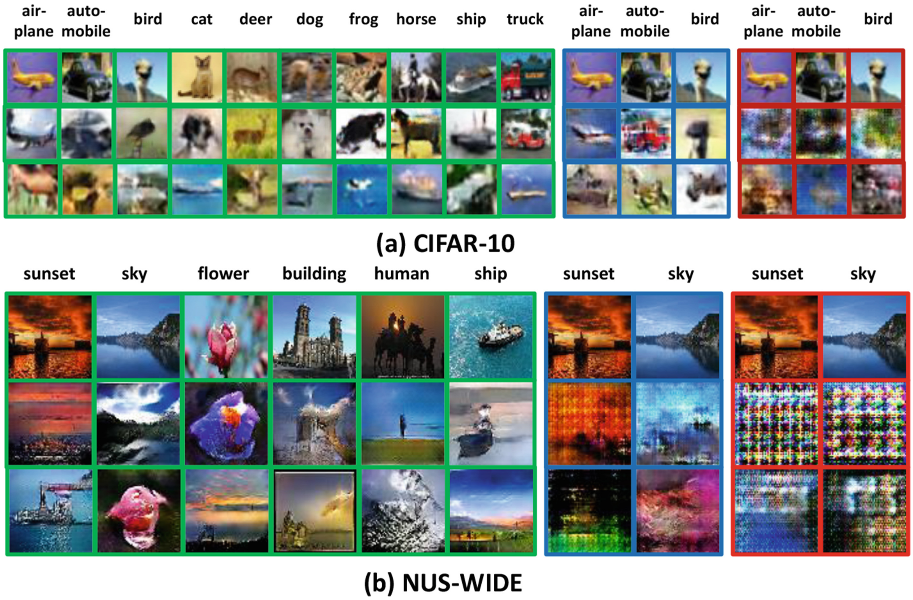

Fig. 3. Visualization of synthetic triplets on (a) CIFAR-10 and (b) NUS-WIDE datasets (better viewed in color). Images in the first row are real images x, followed by synthetic images x^p_{syn} and x^n_{syn}, which are generated by G_p and G_n respectively, thus the three images make up a synthetic triplet $(x, x^p_{syn}, x^n_{syn})$. The green images are generated by our SSGAH, the blue images are generated by our SSGAH without the adversary ranking loss, and the red images are generated by SSGAH without adversary ranking loss and condition generation module.

triplet-wise information. Finally, compared with blue images, the green ones are not only meaningful but also realistic and clear, which verifies that the adversary learning further improves the generative model. Note that compared with synthetic images (blue) on NUSWIDE, those on CIFAR-10 are more clear and meaningful, which is because images on CIFAR-10 dataset are single-labeled and their structures are simple. Thus it's easy to capture the distribution of triplet-relation. The three phenomenons observed above verify the effectiveness of each component of our model and demonstrate that SSGAH can well capture the distribution of labeled and unlabeled data.

4.9 Conclusion

In this paper, we first propose a novel semi-supervised generative adversarial hashing (SSGAH) approach, which unifies the generative model and deep hashing model in minimax two-player game to make full use of a small amount of labeled data and lots of unlabeled data. What's more, we also propose novel semi-supervised ranking loss and adversary ranking loss to learn better binary codes that capturing semantic information of both labeled and unlabeled data. Finally, extensive experiments on two widely-used datasets demonstrate our SSGAH approach outperforms the state-of-the-art hashing mehtods.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grants 61720106012, and 61533016, the Strategic Priority Research Program of Chinese Academy of Science under Grant XDBS01000000, and the Beijing Natural Science Foundation under Grant L172050.

References

1. Abadi, M. et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. CoRR abs/1603.04467 (2016). http://arxiv.org/abs/1603.04467
2. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: Marchand-Maillet, S., Kompatsiaris, Y. (eds.) ACM International Conference on Image and Video Retrieval, p. 48. ACM (2009)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition (CVPR), pp. 886–893. IEEE Computer Society (2005)
4. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: Snoeyink, J., Boissonnat, J. (eds.) Proceedings of the 20th ACM Symposium on Computational Geometry, Brooklyn, New York, USA, June 8–11, 2004, pp. 253–262. ACM (2004)
5. Goodfellow, I. et al.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014). http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
6. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks (2017)

7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Bartlett, P.L., Pereira, F.C.N., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
8. Kulis, B., Darrell, T.: Learning to hash with binary reconstructive embeddings. In: Bengio, Y., Schuurmans, D., Lafferty, J.D., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22, pp. 1042–1050. Curran Associates, Inc. (2009). http://papers.nips.cc/paper/3667-learning-to-hash-with-binary-reconstructive-embeddings.pdf
9. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, June 2015
10. Lazebnik, S.: Iterative quantization: a procrustean approach to learning binary codes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 817–824. IEEE Computer Society (2011)
11. Li, Q., Sun, Z., He, R., Tan, T.: Deep supervised discrete hashing. neural information processing systems, pp. 2482–2491 (2017)
12. Li, W.J., Wang, S., Kang, W.C.: Feature learning based deep supervised hashing with pairwise labels. In: Kambhampati, S. (ed.) International Joint Conference on Artificial Intelligence, pp. 1711–1717. IJCAI/AAAI Press (2016)
13. Liu, W., Wang, J., Ji, R., Jiang, Y.G.: Supervised hashing with kernels. In: Computer Vision and Pattern Recognition, pp. 2074–2081. IEEE Computer Society (2012)
14. Liu, W., Wang, J., Kumar, S., Chang, S.F.: Hashing with graphs. In: Getoor, L., Scheffer, T. (eds.) International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July, pp. 1–8. Omnipress (2011)
15. Lowe, D.G., Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
16. Lcun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
17. Mirza, M., Osindero, S.: Conditional generative adversarial nets. Comput. Sci., 2672–2680 (2014)
18. Norouzi, M., Fleet, D.J.: Minimal loss hashing for compact binary codes. In: Getoor, L., Scheffer, T. (eds.) International Conference on International Conference on Machine Learning, pp. 353–360. Omnipress (2011)
19. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans, pp. 2642–2651 (2017)
20. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001)
21. Qiu, Z., Pan, Y., Yao, T., Mei, T.: Deep semantic hashing with generative adversarial networks. In: Kando, N., Sakai, T., Joho, H., Li, H., de Vries, A.P., White, R.W. (eds.) International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 225–234. ACM (2017)
22. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. Comput. Sci. (2015)
23. Sablayrolles, A., Douze, M., Usunier, N., Jegou, H.: How should we evaluate supervised hashing? In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5–9, 2017, pp. 1732–1736. IEEE (2017)

24. Shen, F., Shen, C., Liu, W., Shen, H.T.: Supervised discrete hashing. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 37–45 (2015)
25. Wang, J., Shen, H.T., Song, J., Ji, J.: Hashing for similarity search: a survey. Comput. Sci. (2014)
26. Wang, J., Zhang, T., Song, J., Sebe, N., Shen, H.T.: A survey on learning to hash. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1–1 (2017)
27. Wang, J., Kumar, S., Chang, S.F.: Semi-supervised hashing for large-scale search. IEEE Trans. Pattern Anal. Mach. Intell. **34**(12), 2393–2406 (2012)
28. Wang, J. et al.: IRGAN: A minimax game for unifying generative and discriminative information retrieval models. In: Kando, N., Sakai, T., Joho, H., Li, H., de Vries, A.P., White, R.W. (eds.) Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7–11, 2017, pp. 515–524. ACM (2017)
29. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) International Conference on Neural Information Processing Systems, pp. 1753–1760. Curran Associates, Inc. (2008)
30. Xia, R., Pan, Y., Lai, H., Liu, C., Yan, S.: Supervised hashing for image retrieval via image representation learning (2014)
31. Yan, X., Zhang, L., Li, W.J.: Semi-supervised deep hashing with a bipartite graph. In: Sierra, C. (ed.) Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 3238–3244. ijcai.org (2017)
32. Yang, H.F., Lin, K., Chen, C.S.: Supervised learning of semantics-preserving hash via deep convolutional neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1–1 (2015)
33. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation (2017)
34. Zhang, J., Peng, Y.: Ssdh: Semi-supervised deep hashing for large scale image retrieval. IEEE Trans. Circuits Syst. Video Technol. (2016)
35. Zhu, H., Long, M., Wang, J., Cao, Y.: Deep hashing network for efficient similarity retrieval. In: Schuurmans, D., Wellman, M.P. (eds.) Thirtieth AAAI Conference on Artificial Intelligence, pp. 2415–2421. AAAI Press (2016)