SSAH: Semi-Supervised Adversarial Deep Hashing with Self-Paced Hard Sample Generation

Sheng Jin,[*1,2] **Shangchen Zhou,**[3] **Yao Liu,**[2] **Chao Chen,**[2]
Xiaoshuai Sun,[4] **Hongxun Yao,**[†1] **Xian-Sheng Hua**[2]
[1]The Harbin Institute of Technology, [2]Alibaba DAMO Academy, Alibaba Group
[3]Nanyang Technological University, [4]Xiamen University
jsh.hit.doc@gmail.com, h.yao@hit.edu.cn.

Abstract

Deep hashing methods have been proved to be effective and efficient for large-scale Web media search. The success of these data-driven methods largely depends on collecting sufficient labeled data, which is usually a crucial limitation in practical cases. The current solutions to this issue utilize Generative Adversarial Network (GAN) to augment data in semi-supervised learning. However, existing GAN-based methods treat image generations and hashing learning as two isolated processes, leading to generation ineffectiveness. Besides, most works fail to exploit the semantic information in unlabeled data. In this paper, we propose a novel **S**emi-supervised **S**elf-pace **A**dversarial **H**ashing method, named **SSAH** to solve the above problems in a unified framework. The SSAH method consists of an adversarial network (A-Net) and a hashing network (H-Net). To improve the quality of generative images, first, the A-Net learns hard samples with multi-scale occlusions and multi-angle rotated deformations which compete against the learning of accurate hashing codes. Second, we design a novel self-paced hard generation policy to gradually increase the hashing difficulty of generated samples. To make use of the semantic information in unlabeled ones, we propose a semi-supervised consistent loss. The experimental results show that our method can significantly improve state-of-the-art models on both the widely-used hashing datasets and fine-grained datasets.

Introduction

In the big data era, large-scale image retrieval is widely used in many practical applications, yet it remains a challenge because of the large computational cost and high accuracy requirement. To address the efficiency and effectiveness issues, hashing methods have become a hot research topic. A great number of hashing methods are proposed to map images into a hamming space, including *traditional hashing methods* (Andoni and Indyk 2006; Lin et al. 2018; 2019) and *deep hashing methods* (Cao et al. 2017; 2018b; Liu et al. 2018; Sheng et al. 2018). Compared with traditional ones, deep hashing methods usually achieve better retrieval performance due to its powerful ability of feature representation and nonlinear mapping.

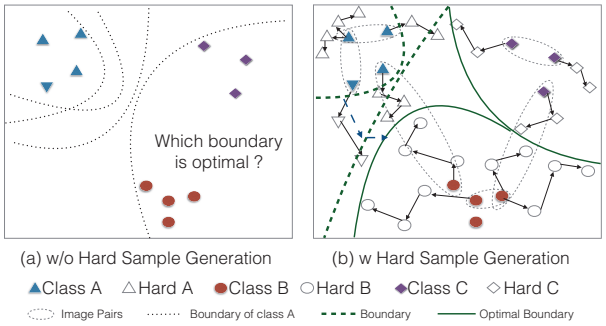

Figure 1: To obtain the optimal boundary for points with similar hashing codes, we propose a novel self-paced deep adversarial hashing to generate hard samples, as shown in (b). Intuitively, these samples can help the network learn more optimal classification boundaries.

Although great efforts have been devoted to deep learning-based algorithms, the label-hungry property makes it intractable in practice. Contrarily, for some retrieval tasks, unlabeled data is always enough. To make use of unlabeled data, several semi-supervised methods are proposed, include *graph-based methods* like SSDH (Zhang and Peng 2017) and BGDH (Yan, Zhang, and Li 2017), and *generation methods* like DSH-GANs (Qiu et al. 2017), HashGAN (Cao et al. 2018a) and SSGAH (Wang et al. 2018). Graph-based works like SSDH and BGDH use graph structure to mine unlabeled data. However, constructing the graph model of large-scale data is expensive computation and time-consuming, and using batch data instead may lead to a suboptimal result. Currently, GAN (Goodfellow et al. 2014) is proved to be effective in generation tasks and then this novel technical are introduced into hashing. Existing GAN-based methods restricted by two crucial problems, *i.e.*, **generation ineffectiveness** and **unlabeled-data underutilization**.

In terms of **generation ineffectiveness**, the existing *GAN-based* methods train the generation network solely based on label information. This setting leads to ineffective genera-

[*]This work was done as a research intern in Alibaba Group.
[†]Correspondence Author

tions that are either too hard or easy for hashing code learning, which unable to match the dynamic training of the hashing network. In terms of **unlabeled-data underutilization**, most existing works like DSH-GANs (Qiu et al. 2017) only exploit unlabeled data to synthesize high-quality images, while the unsupervised data is not utilized when learning hashing codes. We argue that, the above two issues are not independent. In particular, the ineffecitive generation policy makes triplet-wise methods like SSGAH (Wang et al. 2018) failed to make the most of unlabeled data, since these algorithms heavily depend on hard triplets.

In this paper, we propose a novel deep hashing method as a solid solution for **generation ineffectiveness** and **unlabeled-data underutilization** termed semi-supervised self-paced deep adversarial hashing (**SSAH**). The main idea of SSAH is depicted in Figure 1.

To tackle **generation ineffectiveness**, first, our method tries to generate proper hard images to gradually improve hashing learning. The generation network is designed to produce hard samplesL[1] with multi-scale masks and multi-angle rotations. Second, we apply the key idea of SPL[2] to our framework aiming to control the hard generations in the dynamic training procedure. To tackle **unlabeled-data underutilization**, we propose a consistent loss by encouraging consistent binary codes for the input image (both labeled and unlabeled data) and its corresponding hard generations.

Specially, **SSAH** consists of an adversarial generation network (A-Net) and a hashing network (H-Net). The loss function contains four components, including a self-paced adversarial loss, a semi-supervised consistent loss, a supervised semantic loss, and a quantization loss, which guide the training of two networks in an adversarial manner. The A-Net learns deformations and masks to generate hard images, where the quality of these generative images are evaluated by the H-Net. Then the H-Net is trained by both the input images and the generated hard samples. In the test phase, we only use the H-Net to produce hashing codes.

The main contributions of **SSAH** are three-fold:

- To generate samples properly, we propose a novel hashing framework by integrating the self-paced adversarial mechanism into hard generations and hashing codes learning. Our generation network takes both masking and deformation into account.
- To make better use of unlabeled data, a novel consistent loss is proposed to exploit semantic information of all the data in a semi-supervised manner.
- Experimental results on both general and fine-grained datasets demonstrate the superior performance of our method in comparison with many state-of-art methods.

[1]We define the samples which are difficult for current retrieval as hard samples.

[2]The SPL theory (Kumar, Packer, and Koller 2010) is inspired by the learning process of human, where samples are involved in learning from easy to gradually complex ones.

Related Work

We introduce the most related works from two aspects: *Semi-supervised Hashing* and *Hard Example Generation*.

Semi-supervised Hashing Based on whether labeled data is used in the training process, hashing methods can be divided into unsupervised (Liu et al. 2011), semi-supervised (Yan, Zhang, and Li 2017), and supervised ones (Xia et al. 2014). Semi-supervised hashing is effective when a small amount of labeled data and enough unlabeled data is available. SSH (Wang, Kumar, and Chang 2010) 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 a traditional shallow method, which leads to unsatisfying performance compared with deep hashing methods like SSDH (Zhang and Peng 2017), BGDH (Yan, Zhang, and Li 2017), which are discussed in the introduction.

Very recently, some semi-supervised hashing methods are proposed, which use GAN (Goodfellow et al. 2014) to augment data. Deep Semantic Hashing (DSH-GANs) (Qiu et al. 2017) is the first hashing method that introduces GANs into hashing. But it can only incorporate pointwise label which is often unavailable in online image retrieval applications. Cao *et al.* propose a novel conditional GANs based on pairwise supervised information, named HashGAN (Cao et al. 2018a) to solve the insufficient sample problem. However, the sample generation is independent of hashing codes learning. Wang *et al.* (Wang et al. 2018) propose SSGAH which utilizes triplet-labels and specifically designs a GANs model which can be well learned with limited supervised information. For cross-model hashing, Zhang *et al.* (Zhang, Lai, and Feng 2018) mines the attention region in an adversarial way. However, all the mentioned GAN-based methods try to generate as much as possible images, which is not an effective and even feasible solution in most cases.

Hard Example Generation. Hard example generation is currently used in training deep models effectively for many computer vision tasks, including object detection (Wang, Shrivastava, and Gupta 2017), retrieval (Huang et al. 2018), and other tasks (Peng et al. 2018). Zhong *et al.* (Zhong et al. 2017) propose a parameter-learning free method, termed Random Erasing which randomly selects a rectangle region in an image and erases its pixels with random values. However, hard example generations of Random Erasing is still isolated with network training. The generated images may not be consistent with the dynamic training status. Very recently, Wang *et al.* (Wang, Shrivastava, and Gupta 2017) introduce the adversarial mechanism to synthesize hard samples. This method incorporates pointwise supervised information, *e.g.* class labels, which is often unavailable in online image retrieval applications. Different from previous methods, we propose a novel architecture using the pairwise label to generate hard samples for learning better hashing codes. What's more, our proposed generation networks learn hard samples in a self-paced adversarial learning manner.

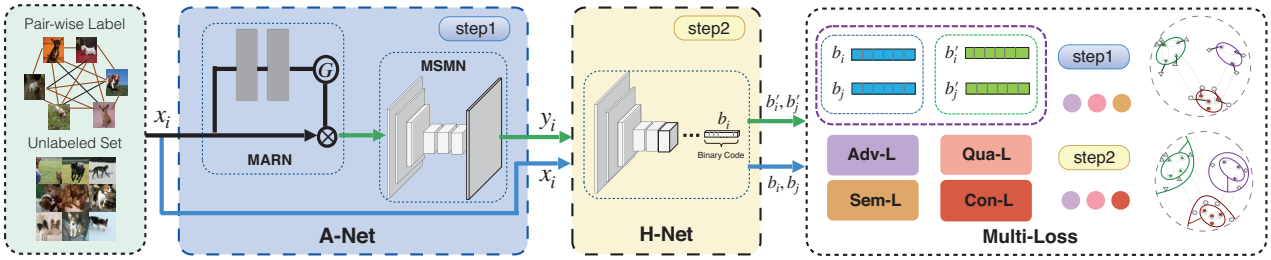

Figure 2: An illustration of self-paced adversarial hashing (**SSAH**). **SSAH** is comprised of two main components: (1) an adversarial network (A-Net) for hard sample generation, (2) a hashing network (H-Net) based on AlexNet for learning hashing codes. In the step1, the A-Net takes as input the training images and pairwise similarity to learn a hard mask. This mask is learned under the criterion that the hashing codes of masked image pairs become contrary to their pairwise label. In the step2, the H-Net take both training images and the generated hard samples as input to learn more accurate binary codes. In the training stage, hard generation and hashing codes are learned in an adversarial way.

Semi-supervised Self-paced Adversarial Hash

Given a dataset consist of labeled and unlabeled data. Since each labeled image in the dataset owns a unique class label, image pairs can be further labeled as similar or dissimilar $S_{ij} = 0$ or 1. where S_{ij} denotes the pairwise label of images (x_i, x_j). Our goal is to learn more accurate hashing codes b_i in a semi-supervised way. To this end, we will present **S**emi-supervised **S**elf-paced **A**dversarial **H**ashing (**SSAH**) in details. Since discrete optimization is difficult to be solved by deep networks, we firstly ignore the binary constraint and concentrate on binary-like code μ_i. Then we obtain b_i from μ_i. Figure 2 illustrates an overview of our architecture, which consists of an A-Net for hard samples generation and an H-Net for binary codes learning. The generation network takes labeled and unlabeled images as inputs and produces hard masks and rotated-related parameters as the outputs. Then the H-Net learns compact binary hashing codes from both generative hard samples and training images.

Adversarial Network

The A-Net is designed to generate hard images. Our method generates hard images in two main methods. The first method is to change the rotation angle and scale of the whole image. Here we propose Multi-Angle Rotation Network, termed **MARN**. The second method is to generate masks to change the value of the pixel. Here we propose Multi-Scale Mask Network, termed **MSMN**.

Multi-Angle Rotation Network. Motivated by STN (Jaderberg et al. 2015), we propose the MARN to create multi-angle rotations on the training images. The reason is we want to preserve the location correspondence between image pixels and landmark coordinates. Otherwise, we might hurt the localization accuracy once the intermediate feature maps are disturbed. However, the Single-angle RN will often predict the largest angle. To improve the diversity of generated images, the MARN is designed to produce n hard generations with size $d_0 \times d_0 \times (3 \times n)$, where d_0 is the scale of images x_0. Each generated hard samples are required in different ranges of angles. More specially, the rotation degree of the first image is constrained within 10° clockwise and anticlockwise. The n-th generated image is constrained within $[10^\circ \times (n-1), 10^\circ \times n]$ clockwise and anticlockwise. For each input image x_i, the MARN produces n hard generated images, which is denoted as ${y_i}^{(n)}$.

Multi-Scale Mask Network. The object of MSMN is to produce multi-scale masks to make training images harder. In the hashing learning stage, we can obtain the convolutional features from different layers in H-Net. These features represent different spatial scales region of the original image. Correspondingly, we generate multi-scale additive and multiplicative masks for these selected features.

The framework of MSMN is shown in Figure 3. Specially, for each selected convolutional layer m, we extract features f_m with size $d_m \times d_m \times c_m$, where d_m is the spatial dimension and c_m represents the number of channels. Given this feature, our MSMN predicts an additive hard mask AM_m and a multiplicative hard mask PM_m.

We use the sigmoid function as the activation function for additive masks and tanh function for multiplicative masks. The value of AM_m with $d_m \times d_m$ is in range of $[0, 1]$ and that of PM_m is in range of $[-1, 1]$. The corresponding features of hard samples, which is denoted as F_m, is obtained by

$$F_m = (I - sigmoid(PM_m)) \cdot f_m + tanh(AM_m), \quad (1)$$

When the value of m is 0, f_m represents the images ${y_i}^{(n)}$.

Hashing Learning Network

We directly adopt a pre-trained AlexNet (Krizhevsky, Sutskever, and Hinton 2012) as the base model of the H-Net. The raw image pixel, from either the original images and the generated images, is the input of the hashing model. The learned additive masks AM_m and multiplicative masks PM_m are only required in the coding process of generated images y_i. The hard feature maps are computed according to Eq (1). For each generated images y_i, μ_i are learned:

$${\mu'}_i^{(n)} = \textit{H-Net}({y_i}^{(n)} | AM_m, PM_m, \forall m > 0). \quad (2)$$

The output layer of AlexNet is replaced by a hashing layer where the dimension is defined based on the length of the required hashing code.

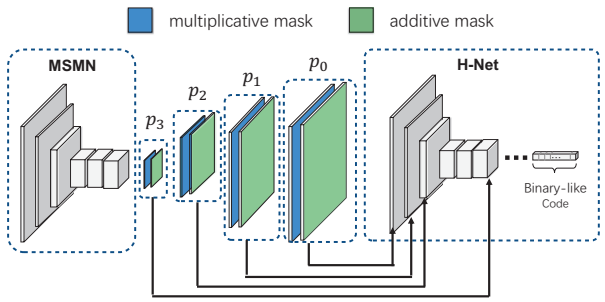

Figure 3: The framework of MSMN, where p_i denotes multi-scale masks for the selected layer.

Loss Functions

In this section, we design the adversarial loss function, including a self-paced adversarial loss and a supervised semantic loss, for the A-Net to guide the generations of hard samples. Besides, a hashing loss function for the H-Net, including the supervised semantic loss and the semi-supervised consistent loss, is proposed to learn better codes by capturing semantic information of all data.

Self-paced Adversarial Loss. Most existing GAN based hashing methods try to generate images to augment data. However, these methods can not ensure the quality of generative samples, which may obtain bad samples: (1) too difficult or too easy, (2) unable to match the dynamic training.

To improve the **effectiveness of generations**, in this paper, we leverage the H-Net to guide the training of the A-Net by a novel definition termed **hard degree**. Firstly, we compute the similarity degree of image pairs $(x_i,\ x_j)$ by using $\frac{\mu_i^T * \mu_j + k}{2k}$, where μ_i represents binary-like codes of the input images. The distance of the pairwise label and the estimated similarity degree is denoted as d_{ij}:

$$d_{ij} = S_{i,j} - (2 * S_{i,j} - 1) * \frac{\mu_i^T * \mu_j + k}{2k}. \tag{3}$$

d'_{ij} can be obtained similarly, where μ'_i represents the binary-like codes of their corresponding generated images. .

Since the hard samples learned by the A-Net may increase the H-Net loss, the value of d'_{ij} is required to be larger than that of d_{ij}. The **hard degree** $hard(y_i, y_j)$ of generative image pairs $(y_i,\ y_j)$ is defined by using the difference between $d_i j$ and d'_{ij}, which can be formulated as:

$$hard(y_i, y_j) = d'_{ij} - d_{ij} = \frac{(2S_{i,j} - 1)}{2k}(\mu'^T_i \mu'_j - \mu_i^T \mu_j). \tag{4}$$

We adopt a positive dynamic margin $(1 - d_{ij})\omega$ to define the self-paced adversarial loss, which can be concluded as:

$$l_{adv}(y_i, y_j, \omega) = \max(\omega(1 - d_{ij}) - hard(y_i, y_j), 0). \tag{5}$$

where $\omega > 0$ is a fixed constant.

Discussion. The self-paced adversarial loss has two merits: *considering inter-pair difference* and *generating hard samples gradually*. In terms of *inter-pair difference*, the value of d_{ij} is large, when original image pairs (x_i, x_j) is hard to distinguish. In this situation, the hard degree of (y_i, y_j) is not necessarily large. By contrast, if (x_i, x_j) is easy to be distinguished, the hard degree of (y_i, y_j) need to be relatively large. To meet difference requirements, $(1 - d_{ij})$ adapts the margin of hard degree. In terms of *hard generations policy*, with the training of H-Net, better codes are learned and then d_{ij} become small gradually. $1 - d_{ij}$ can be used to obtain a larger margin, leading to harder generations.

Similarly, we use x_i, y_j as cross-domain image pairs, where x_i is the input image and y_i is its corresponding hard samples. The **hard degree** of $(x_i,\ y_j)$ is define as $hard(x_i, y_j)$, which can be formulated as:

$$hard(x_i, y_j) = \frac{(2S_{i,j} - 1)}{2k}(\mu_i^T \mu'_j - \mu_i^T \mu_j). \tag{6}$$

Then self-paced adversarial loss of $(x_i,\ y_j)$ is defined as $l_{adv}(x_i, y_j, \omega/2)$, where the constant $\omega > 0$ is used in Eq (5). The whole self-paced adversarial loss is written as:

$$l_{sp} = \sum_{i,j}\sum_{n} l_{adv}(y_i^{(n)}, y_j^{(n)}, \omega) + l_{adv}(x_i, y_j^{(n)}, \omega/2). \tag{7}$$

Supervised Semantic Loss. Similar to other hashing methods, we adopt a pairwise semantic loss to ensure the hashing codes preserve relevant class information. The similarity degree of image pairs is computed by $\frac{\mu_i^T * \mu_j + k}{2k}$. Then the value of similarity degree is required to near the pairwise label $S_{i,j}$. Since the H-Net is trained by both labeled image and their corresponding hard generations, the supervised semantic loss can be written as:

$$\begin{aligned} l_{sem} &= \sum_{i,j} l_{pair}(x_i, x_j) + l_{pair}(x_i, y_j) + l_{pair}(y_i, y_i) \\ &= \sum_{i,j} \frac{\mu_i^T * \mu_j}{2k} + \frac{\mu'^T_i * \mu_j}{2k} + \frac{\mu'^T_i * \mu'_j}{2k} + \frac{3}{2}. \end{aligned} \tag{8}$$

Besides, when training the A-Net, the supervised semantic loss is also adopted as a **regular term** to preserve class information of the generative hard samples.

Semi-supervised Consistent Loss. In some related computer vision tasks like semi-supervised classification, pseudo-ensembles methods[2] (Bachman, Alsharif, and Precup 2014; Laine and Aila 2016; Tarvainen and Valpola 2017) are proved to be effective. These methods encourage consistent networks output for each image with and without noise. Motivated by the success of these works, we propose a novel consistent loss to improve the **utilization of unlabeled data**.

However, compared with existing pseudo-ensembles methods, which always adopt random and data-independent noise, our proposed method designs the A-Net to generate more proper noise for the inputs, including multi-scale masks and multi-angle rotation. Then the H-Net is required to learn consistent binary codes of the training images x_i (including labeled and unlabeled images) and their corresponding hard samples y_i, by taking the hamming distance

[2]These methods develop from the cognitive ability of human. When a percept is changed slightly, a human typically still consider it to be the same object.

Algorithm 1 Self-paced Deep Adversarial Hashing

Input: Training set and their corresponding class labels.
Output: H-Net function: *H-Net*$(x|\Theta_2)$ and A-Net function: *A-Net*$(x|\Theta_1)$.

```
For the entire training set, construct the pairwise label matrix S.
for t = 1 : T epoch do
    Compute B accordi.ng to Eq (2).
    Fixing Θ2, update Θ1 according to Eq (11).
    Fixing Θ1, update Θ2 according to Eq (12).
end for
return HN(x|Θ2), AN(x|Θ1).
```

between the hashing codes μ_i and ${\mu'}_i$. The consistent loss can be formulated as:

$$l_{con} = \sum_i \left\| \frac{k - {\mu'}_i^T * \mu_i}{2k} \right\|_2. \tag{9}$$

Quantization Loss. Due to the ignorant of the binary constraint, a widely-used quantization loss is used to pull the value of μ_i and that of b_i together, which is written as:

$$l_{quan} = \sum_i \|\mu'_i - b'_i\|_1 + \|\mu_i - b_i\|_1, \tag{10}$$

Alternating Optimization

Our network consists of two sub-networks: an adversarial network, termed A-Net, for hard image generation and a hashing network, termed H-Net, for compact hashing codes learning. As shown in Algorithm. 1, we train the A-Net and the H-Net iteratively. The overall training objective of the A-Net integrates the semantic loss defined in Eq (8), the self-paced adversarial loss of three types of image pairs defined in Eq (7) and the quantization loss defined in Eq (10). The A-Net is trained by the following loss:

$$\min_{\Theta_1} \quad \alpha l_{sp} + \lambda_1 l_{sem} + \beta l_{quan}. \tag{11}$$

By minimizing this term, the A-Net is trained to generate proper hard samples, leading to better hashing codes.

For the shared hashing model, the H-Net is trained by the input data, including labeled and unlabeled images, and their corresponding hard genrative samples. The supervised semantic loss, semi-supervised consistent loss, and the quantization loss are used to train the H-Net. We update its parameters according to the following overall loss:

$$\min_{\Theta_2} \quad \lambda_1 l_{sem} + \lambda_2 l_{con} + \beta l_{quan}. \tag{12}$$

By minimizing this term, the shared H-Net is trained to learn effective hashing codes.

Experiment

To test the performance of our proposed **SSAH** method, we conduct experiments on general hashing datasets, *i.e.* CIFAR-10 and NUS-WIDE, to verify the effectiveness of our method. Then we conduct experiments on two fine-grained datasets, *i.e.* CUB Bird and Stanford Dogs-120, to prove that our method is still robust and effective for more complex fine-grained retrieval tasks. We also conduct some analytical experiments to further verify our method.

Table 1: The mAP scores for different number of bits on CIFAR-10 and NUSWIDE datasets.

Dataset	CIFAR-10			NUSWIDE		
	12 bits	24 bits	48 bits	12 bits	24 bits	48 bits
ITQ-CCA	0.435	0.435	0.435	0.526	0.575	0.594
KSH	0.556	0.572	0.588	0.618	0.651	0.682
SDH	0.558	0.596	0.614	0.645	0.688	0.711
CNNH	0.439	0.476	0.489	0.611	0.618	0.608
HashGAN	0.655	0.709	0.727	0.708	0.722	0.730
SSDH	0.801	0.813	0.814	0.773	0.779	0.778
BGDH	0.805	0.824	0.833	0.803	0.818	0.828
SSGAH	0.819	0.837	0.855	0.835	0.847	0.865
SSAH	**0.862**	**0.878**	**0.886**	**0.872**	**0.884**	**0.898**

Datasets

We conduct our experiments on two general datasets, namely CIFAR-10 and NUSWIDE. CIFAR-10 (Krizhevsky and Hinton 2009) is a small image dataset including 60k 32 $\times$ 32 images in 10 classes. Each image belongs to one class (6000 images per class). NUS-WIDE (Chua et al. 2009) contains nearly 270k images with 81 semantic concepts. For NUS-WIDE, we follow (Liu et al. 2011) to use the images associated with the 21 most frequent concepts, where each of these concepts associated with at least 5,000 images. Following (Liu et al. 2011; Wang et al. 2018), we randomly sample 100 images per class as the test set, and the others are as a database. In the training process, we randomly sample 500 images per class from the database as labeled data, and the others are as unlabeled data.

We further verify our experiments on two widely-used fine-grained datasets, namely Stanford Dogs-120 and CUB Bird. **Stanford Dogs-120** (Nilsback and Zisserman 2006) dataset consists of 20,580 images in 120 mutually classes. Each class contains about 150 images. **CUB Bird** (Wah et al. 2011) includes 11,788 images in mutually 200 classes. We directly use test set defined in these datasets. The train set is used as a database. In the training process, we randomly sample 50% images per class s from the database as labeled data, and the others are as unlabeled data.

Comparative Methods and Evaluation Metrics

For the general datasets, including CIFAR-10 and NUSWIDE dataset, we compare our method (**SSAH**) with three supervised deep hashing methods: CNNH (Xia et al. 2014), HashGAN (Cao et al. 2018a), three semi-supervised deep hashing methods: SSDH (Zhang and Peng 2017), BGDH (Yan, Zhang, and Li 2017), SSGAH (Wang et al. 2018) and three shallow methods: ITQ-CCA (Gong et al. 2013), KSH (Liu et al. 2012), SDH (Shen et al. 2015). For the fine-grained datasets, our method is further compared with DSaH (Sheng et al. 2018), which is the first hashing method designed for fine-grained retrieval.

For a fair comparison between traditional and deep hashing methods, we conduct these methods on features ex-

tracted from the fc7 layer of AlexNet which is pre-trained on ImageNet. For deep hashing methods, we use as input the original images, and adopt AlexNet (Krizhevsky, Sutskever, and Hinton 2012) as the backbone architecture.

Evaluation Metric. We use Mean Average Precision (**mAP**) for quantitative evaluation. The Precision@top-N curves and Precision-Recall curves are shown in supple materials.

Implementation Details

Network Design. As shown in Figure 2, our model consists of an A-Net including MSMN, MARN, and a H-Net. For MSMN, we adopt a lightweight version of generator in GANimation (Pumarola et al. 2018). This network contains a stride-2 convolution, three residual blocks (He et al. 2016), and a 1/2-strided convolution. Similar to Johnson *et al.* (Johnson, Alahi, and Fei-Fei 2016), we use instance normalization (Ulyanov, Vedaldi, and Lempitsky 2016). The figurations of MSMN are shown in the supplematerials. MARN is built upon the STN (Jaderberg et al. 2015). Different from STN, our transformer network produces n sets of affine parameters. We adopt AlexNet (Krizhevsky, Sutskever, and Hinton 2012) as the encoder of H-Net, fine-tune all layers but the last one are copied from the pre-trained AlexNet.

Trainind Details. Our **SSAH** is implemented on PyTorch and the deep model is trained by batch gradient descent. In the training stage, images are regarded as input in the form of batch and every two images in the same batch construct an image pair. Practically, we train A-Net before H-Net. If we first train H-Net, A-Net might output semantic-irrelevant hard generated images, which would be a bad sample and guide the training of hashing model in the wrong direction.

Network Parameters. The value of hyper-parameter λ_1 is 1.0, λ_2 is 0.5, α is 0.5 and β is 0.1. We use the mini-batch stochastic gradient descent with 0.9 momentum. We set the value of the margin parameters ω as 0.1, which increases 0.02 every 5 epochs. The mini-batch size of images is fixed as 32 and the weight decay parameter as 0.0005. The value of the number of the rotated hard samples n is 3.

Quantitative Results

Performance on general hashing datasets. The Mean Average Precision (mAP,%) results of different methods for different numbers of bits on NUSWIDE and CIFAR-10 dataset are shown in Table 1. Experimental results show that **SSAH** outperforms state-of-the-art SSGAH (Wang et al. 2018) by 3.75%, 3.55% on CIFAR10, and NUSWIDE, respectively. According to the experimental results, **SSAH** can be seen to be more effective for traditional hashing task.

Performance on fine-grained hashing datasets. The fine-grained retrieval task requires methods describing fine-grained objects that share similar overall appearance but have a subtle difference. To meet this requirement, there will be greater demand for collecting and annotating data, where professional knowledge is required in some cases. Since it is more difficult to generate fine-grained objects, GAN-based hashing methods are also not effective due to the scarcity of supervised data. However, the experimental results show that our method is still robust to this task.

The mAP results of different methods on fine-grained datasets are shown in Table 2. The proposed **SSAH** method substantially outperforms all the comparison methods. Compared with existing best retrieval performance (DSaH (Alexnet)), **SSAH** achieves absolute increases of 4.55%, 6.24% on CUB Brid datasets and on Stanford dog datasets, respectively. Compared with the mAP results on traditional hashing task, our method is proved to achieve a significant improvement in fine-grained retrieval. What's more, we only use the H-Net to produce binary codes in the test phase and the DSaH method need to highlight the salient regions before the encoding process. Thus, our method is also more efficient in time.

Table 2: The mAP scores for different number of bits on Stanford Dogs-120 and CUB Bird datasets.

Dataset	Stanford Dogs-120			CUB Bird		
	12 bits	24 bits	48 bits	12 bits	24 bits	48 bits
HashGAN	0.029	0.172	0.283	0.020	0.0542	0.123
SSGAH	0.127	0.233	0.329	0.073	0.1321	0.247
DSaH	0.244	0.287	0.408	0.091	0.2087	0.285
SSAH	**0.273**	**0.343**	**0.478**	**0.141**	**0.265**	**0.359**

Performance of Unseen Classes. To further evaluate our SSAH approach, we adopt the evaluation protocol from (Sablayrolles et al. 2017). In the training process, 75% of classes (termed set 75) are known, and the remaining 25% classes (termed set 25) are used to for evaluation. The set 75 and set 25 are further divided into the training set and test set. Data amount in train and test set are the same. Following settings in (Zhang and Peng 2017), we use train75 as the training set and test25 as the test set. For general hashing retrieval, 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.

The mAP scores under the retrieval of unseen classes are shown in Table 3. Our SSAH method achieves the best result when retrieving unseen classes, which means that our method achieves better generalization performance to unseen class. The experimental results on fine-grained datasets are shown in supple materials.

Ablation Study

To further verify our method, we conduct some analysis experiments including: (1) the effectiveness of hard samples generation, (2) the analysis of each loss component, (3) the effectiveness of self-paced generation policy.

Component Analysis of the Network. We compare our MARN and MSMN with random image rotation/random mask generation strategy in training stage using the AlexNet architecture. (1) Random Image Rotation: For each image, we obtain three rotated images, where each image are rotated in the specified angle range. (2) Random Mask Generation: The values of multiplicative masks are in range of $[0, 1]$ and that of additive masks are in range of $[-1, 1]$. The 90% values of multiplicative and additive masks are required in the range of $[-0.1, 0.1]$.

Table 3: The mAP scores under retrieval of unseen classes on CIFAR-10 and NUSWIDE datasets.

Dataset	CIFAR-10			NUSWIDE		
	12 bits	24 bits	48 bits	12 bits	24 bits	48 bits
ITQ-CCA	0.157	0.165	0.201	0.488	0.493	0.503
SDH	0.185	0.193	0.213	0.471	0.490	0.507
CNNH	0.210	0.225	0.231	0.445	0.463	0.477
DRSCH	0.219	0.223	0.251	0.457	0.464	0.360
NINH	0.241	0.249	0.272	0.484	0.483	0.487
SSDH	0.285	0.291	0.325	0.510	0.533	0.551
SSGAH	0.309	0.323	0.339	0.539	0.553	0.579
SSAH	**0.338**	**0.370**	**0.379**	**0.569**	**0.571**	**0.596**

We report our results for using MARN and MSMN in Table 4. For the AlexNet architecture, the mAP of our implemented baseline is $75.1\% \sim 79.2\%$ and $76.2\% \sim 80.1\%$ on CIFAR-10 and NUS-WIDE datasets. Based on this setting, joint learning with our MARN model improves baseline by 4.5% and 4.4%, respectively on these datasets. Joint learning with the MSMN model improves baseline by 7.0% and 6.9%. As both methods are complementary to each other, combining MARN and MSMN into our model gives another boost to $86.2\% \sim 88.6\%$ and $87.2\% \sim 89.8\%$ on CIFAR-10 and NUS-WIDE datasets, respectively.

Table 4: The mAP scores of SSAH using different network components (MARN and MSMN).

Methods	CIFAR-10		NUSWIDE	
	12 bits	48 bits	12 bits	48 bits
baseline	0.751	0.792	0.762	0.801
random rotate	0.788	0.810	0.797	0.828
+MARN	0.801	0.831	0.810	0.842
random mask	0.792	0.813	0.806	0.811
+MSMN (+)	0.811	0.838	0819	0.832
+MSMN (×)	0.820	0.847	0.831	0.849
+MSMN	0.830	0.853	0.840	0.861
random rotate+mask	0.796	0.816	0.810	0.833
Ours(full)	**0.862**	**0.886**	**0.872**	**0.898**

Component Analysis of the Loss Functions. Our loss function consists of three major components: l_{sem}, l_{con} and l_{sp}. l_{sp} includes that of hard and cross-domain image pairs, which are denoted as l_{sph} and l_{spc}. To evaluate the contribution of each loss, we study the effect of different loss combinations on retrieval performance. From Table 5, when we use l_{sem}, l_{con} and l_{quan} to train H-Net, and l_{sp} and l_{sem} to train A-Net, the retrieval performance is best. For the H-Net, the self-paced adversarial loss l_{sp} may destroy the training procedure of accurate binary codes. Combined with l_{con}, our method further improves about 2%, which shows H-Net capture the semantic information of unlabeled data. We also show some typical visualization results of hard masks using different loss components of the A-Net in Figure 4.

As shown in Figure 4, for the A-Net, if we only use the semantic loss, the generated mask would avoid the object. if we only use the self-paced adversarial loss, the generated mask occludes the object in some cases. However, using the combination of l_{adv} and l_{sem} can obtain a proper mask. The object is partially occluded and it still can be recognized.

The effectiveness of Self-paced Hard Generation Policy.

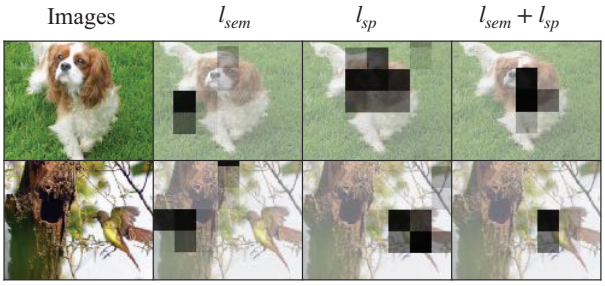

Figure 4: Typical Examples of the learned mask using different loss components on fine-grained dataset.

Table 5: The mAP scores of SSAH on CIFAR-10 dataset using different combinations of loss functions.

A-Net	H-Net	CIFAR-10		
		12 bits	24 bits	48 bits
l_{sem}	$l_{sem} + l_{con}$	0.809	0.823	0.85
l_{sph}		0.821	0.843	0.862
l_{spc}		0.836	0.850	0.870
l_{sp}		0.848	0.862	0.872
$l_{sem} + l_{sp}$	l_{sem}	0.841	0.855	0.867
	$l_{sem} + l_{con}$	**0.862**	**0.878**	**0.886**
	$l_{sem} + l_{con} + l_{sp}$	0.843	0.852	0.8612

In this section, we evaluate SSAH on the impact of generation policy by self-paced vs. fixed-paced loss. To define the fixed-paced loss, the dynamic margin $(1 - d_{ij})\omega$ in Eq (5) is replace by a fixed parameter ω. The fixed-paced loss is formulated as $l_{fixed}(y_i, y_j, \omega) = \max(\omega - hard(y_i, y_j), 0)$.

As shown in Table 6, compare with fixed-paced loss, the self-paced adversarial loss is more effective and also robust to the margin parameter ω. A possible reason is that the fixed-paced loss can not match the dynamic training procedure, and balance hard pairs and simple ones.

Table 6: The mAP scores of SSAH on CIFAR-10 using self-paced loss and fixed-paced loss with different margin ω.

Margin Parameters ω	Self-paced Loss		Fixed-paced Loss	
	12 bits	48 bits	12 bits	48 bits
0.01	0.790	0.803	0.791	0.815
0.05	0.843	0.855	0.819	0.842
0.3	0.855	0.873	0.838	0.852
0.5	0.846	0.861	0.789	0.813
1.0	0.849	0.852	0.775	0.801
Ours(0.1)	**0.862**	**0.886**	**0.839**	**0.861**

Conclusion

To solve the data insufficiency problem, we propose a **S**emi-supervised **S**elf-paced **A**dversarial **H**ashing (**SSAH**) method, consisting of an adversarial network (A-Net) and a hashing network (H-Net). To exploit the semantic information in images, and their corresponding hard generative images, we adopt a supervised semantic loss and a novel semi-supervised consistent loss to train the H-Net. Then the H-Net is used to guide the training of A-Net by a novel self-paced

adversarial loss to produce multi-scale masks and some sets of deformation parameters. The A-Net and the H-Net are trained iteratively in an adversarial way. Extensive experimental results demonstrate the effectiveness of **SSAH**.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Project No. 61772158, 61702136, and U1711265.

References

Andoni, A., and Indyk, P. 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS'06. 47th Annual IEEE Symposium on*. IEEE.

Bachman, P.; Alsharif, O.; and Precup, D. 2014. Learning with pseudo-ensembles. In *Nips*, 3365–3373.

Cao, Y.; Long, M.; Wang, J.; and Liu, S. 2017. Deep visual-semantic quantization for efficient image retrieval. In *CVPR*.

Cao, Y.; Liu, B.; Long, M.; Wang, J.; and KLiss, M. 2018a. Hashgan: Deep learning to hash with pair conditional wasserstein gan. In *CVPR*, 1287–1296.

Cao, Y.; Long, M.; Bin, L.; and Wang, J. 2018b. Deep cauchy hashing for hamming space retrieval. In *CVPR*.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *MM*. ACM.

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

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Nips*, 2672–2680.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, H.; Li, D.; Zhang, Z.; Chen, X.; and Huang, K. 2018. Adversarially occluded samples for person re-identification. In *CVPR*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. In *Nips*, 2017–2025.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 694–711. Springer.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Nips*.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *Nips*, 1189–1197.

Laine, S., and Aila, T. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*.

Lin, M.; Ji, R.; Liu, H.; and Wu, Y. 2018. Supervised online hashing via hadamard codebook learning. In *MM*. ACM.

Lin, M.; Ji, R.; Liu, H.; Sun, X.; Wu, Y.; and Wu, Y. 2019. Towards optimal discrete online hashing with balanced similarity. *arXiv preprint arXiv:1901.10185*.

Liu, W.; Wang, J.; Kumar, S.; and Chang, S.-F. 2011. Hashing with graphs. In *Proceedings of the 28th international conference on machine learning*, 1–8.

Liu, W.; Wang, J.; Ji, R.; Jiang, Y.-G.; and Chang, S.-F. 2012. Supervised hashing with kernels. In *CVPR*. IEEE.

Liu, B.; Cao, Y.; Long, M.; Wang, J.; and Wang, J. 2018. Deep triplet quantization. *MM, ACM*.

Nilsback, M.-E., and Zisserman, A. 2006. A visual vocabulary for flower classification. In *CVPR*, volume 2, 1447–1454. IEEE.

Peng, X.; Tang, Z.; Yang, F.; Feris, R. S.; and Metaxas, D. 2018. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *CVPR*, 2226–2234.

Pumarola, A.; Agudo, A.; Martinez, A. M.; Sanfeliu, A.; and Moreno-Noguer, F. 2018. Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*, 818–833.

Qiu, Z.; Pan, Y.; Yao, T.; and Mei, T. 2017. Deep semantic hashing with generative adversarial networks. In *ACM SIGIR*, 225–234. ACM.

Sablayrolles, A.; Douze, M.; Usunier, N.; and Jégou, H. 2017. How should we evaluate supervised hashing? In *ICASSP*. IEEE.

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

Sheng, J.; Xiaoshuai, S.; Hongxun, Y.; Shangchen, z.; Lei, Z.; and Xiansheng, H. 2018. Deep saliency hashing for fine-grained retrieval. *arXiv preprint arXiv:1807.01459*.

Tarvainen, A., and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Nips*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, CIT.

Wang, G.; Hu, Q.; Cheng, J.; and Hou, Z. 2018. Semi-supervised generative adversarial hashing for image retrieval. In *ECCV*.

Wang, J.; Kumar, S.; and Chang, S.-F. 2010. Semi-supervised hashing for scalable image retrieval. In *CVPR*.

Wang, X.; Shrivastava, A.; and Gupta, A. 2017. A-fast-rcnn: Hard positive generation via adversary for object detection. In *CVPR*.

Xia, R.; Pan, Y.; Lai, H.; Liu, C.; and Yan, S. 2014. Supervised hashing for image retrieval via image representation learning. In *AAAI*, volume 1, 2156–2162.

Yan, X.; Zhang, L.; and Li, W.-J. 2017. Semi-supervised deep hashing with a bipartite graph. In *IJCAI*, 3238–3244.

Zhang, J., and Peng, Y. 2017. Ssdh: semi-supervised deep hashing for large scale image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology* 29(1):212–225.

Zhang, X.; Lai, H.; and Feng, J. 2018. Attention-aware deep adversarial hashing for cross-modal retrieval. In *ECCV*.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2017. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*.