Targeted Attack for Deep Hashing based Retrieval

Jiawang Bai[1,2,*], Bin Chen[1,2,*], Yiming Li[1,*], Dongxian Wu[1,2], Weiwei Guo[3], Shu-Tao Xia[1,2], and En-Hui Yang[4]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University
[2] PCL Research Center of Networks and Communications, Peng Cheng Laboratory
[3] vivo AI Lab
[4] Department of Electrical and Computer Engineering, University of Waterloo

Abstract. The deep hashing based retrieval method is widely adopted in large-scale image and video retrieval. However, there is little investigation on its security. In this paper, we propose a novel method, dubbed deep hashing targeted attack (DHTA), to study the targeted attack on such retrieval. Specifically, we first formulate the targeted attack as a *point-to-set* optimization, which minimizes the average distance between the hash code of an adversarial example and those of a set of objects with the target label. Then we design a novel *component-voting scheme* to obtain an *anchor code* as the representative of the set of hash codes of objects with the target label, whose optimality guarantee is also theoretically derived. To balance the performance and perceptibility, we propose to minimize the Hamming distance between the hash code of the adversarial example and the anchor code under the ℓ^∞ restriction on the perturbation. Extensive experiments verify that DHTA is effective in attacking both deep hashing based image retrieval and video retrieval.

Keywords: Targeted Attack, Deep Hashing, Adversarial Attack, Similarity Retrieval

1 Introduction

High-dimension and large-scale data approximate nearest neighbor (ANN) retrieval has been widely adopted in online search engines, *e.g.*, Google or Bing, due to its efficiency and effectiveness. Within all ANN retrieval methods, hashing-based methods [42] have attracted a lot of attentions due to their compact binary representations and rapid similarity computation between hash codes with Hamming distance. In particular, deep learning based hashing methods [28,2,8,19,35,24] have shown their superiority in performance since they generally learn more meaningful semantic hash codes through learnable hashing functions with deep neural networks (DNNs).

[*]Equal contribution.
Correspondence to: Bin Chen (cb17@mails.tsinghua.edu.cn).

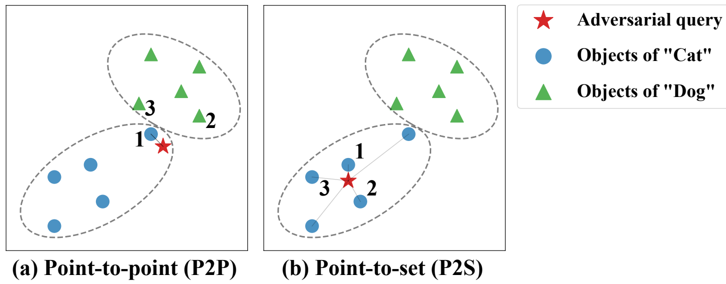


Fig. 1. The comparison between the P2P attack paradigm and proposed P2S paradigm. There are two object classes (*i.e.* 'Cat' and 'Dog') as shown above, where the target label being attack is 'Cat'. In the P2P paradigm, a object with the target label is randomly selected as the reference to generate the adversarial query. But if the selected object is close to the category boundary (dotted lines in the figure) or is an outlier, the attack performance will be poor. In this example, the 'targeted attack success rate' of P2P and P2S is 33.3% and 100%, respectively.

Recent studies [39,16,48,14,1,6] revealed that DNNs are vulnerable to adversarial examples, which are crafted by adding intentionally small perturbations to benign examples and fool DNNs to confidently make incorrect predictions. While deep retrieval systems take advantage of the power of DNNs, they also inherit the vulnerability to adversarial examples [15,25,40,52]. Previous research [52] only paid attention to design a non-targeted attack in deep hashing based retrieval, *i.e.*, returning retrieval objects with incorrect labels. Compared with non-targeted attacks, targeted attacks are more malicious since they make the adversarial examples misidentified as a predefined label and can be used to achieve some malicious purposes [5,13,32]. For example, a hashing based retrieval system may return violent images when a child queries with an intentionally perturbed cartoon image by the adversary. Accordingly, it is desirable to study the targeted attacks on deep hashing models and address their security concerns.

This paper focuses on the targeted attack in hashing based retrieval. Different from classification, retrieval aims at returning multiple relevant objects instead of one result, which indicates that the query has more important relationship with the set of relevant objects than with other objects. Motivated by this fact, we formulate the targeted attack as a *point-to-set* (P2S) optimization, which minimizes the average distance between the compressed representations (*e.g.*, hash codes in Hamming space) of the adversarial example and those of a set of objects with the target label. Compared with the *point-to-point* (P2P) paradigm [40] which directs the adversarial example to generate a representation similar to that of a randomly chosen object with the target label, our proposed point-to-set attack paradigm is more stable and efficient. The detailed comparison between P2S and P2P attack paradigm is shown in Figure 1. In particular, when minimizing the average Hamming distances between a hash code and those of an object set, we prove that the globally optimal solution (dubbed *anchor code*) can be achieved through a simple component-voting scheme, which is a gift from the nature of hashing-based retrieval. Therefore, the anchor code can be naturally chosen as a targeted hash code to direct the generation of adversarial query. To further balance the attack performance and the imperceptibility, we propose a novel attack method, dubbed ***d**eep **h**ashing **t**argeted **a**ttack* (DHTA), by minimizing the Hamming distance between the hash code of adversarial query and the anchor code under the ℓ^∞ restriction on the adversarial perturbations.

In summary, the main contribution of this work is four-fold:

- We formulate the targeted attack on hashing retrieval as a point-to-set optimization instead of the common point-to-point paradigm considering the characteristics of retrieval tasks.
- We propose a novel component-voting scheme to obtain an anchor code as the representative of the set of hash codes of objects with the target label, whose theoretical optimality of proposed attack paradigm with average-case point-to-set metric is discussed.
- We develop a simple yet effective targeted attack, the DHTA, which efficiently balances the attack performance and the perceptibility. This is the first attempt to design a targeted attack on hashing based retrieval.
- Extensive experiments verify that DHTA is effective in attacking both deep hashing based image retrieval and video retrieval.

2 Related Work

2.1 Deep Hashing based Similarity Retrieval

Hashing methods can map semantically similar objects to similar compact binary codes in Hamming space, which are widely adopted to accelerate the ANN retrieval [42]. The classical version of data-dependent hashing consists of two parts, including hash function learning and binary inference [26,29,43,34].

Recently, more and more deep learning techniques were introduced to the traditional hashing-based retrieval methods and reach state-of-the-art performance, thanks to the powerful feature extraction of deep neural networks. The first deep hashing method was proposed in [47] focusing on image retrieval. Recent works showed that learning hashing mapping in an end-to-end manner can greatly improve the quality of the binary codes [23,28,3,2]. The above-mentioned methods can be easily extended to multi-label image retrieval, *e.g.*, [54,44]. Depending on the availability of unlabeled images, other researchers devoted to design novel hashing methods to cope with the lack of labeled images, *e.g.*, unsupervised deep hashing method [51], and semi-supervised one [50]. Different from deep image hashing methods, deep video hashing usually first extract frame features by a convolutional neural network (CNN), then fuse them to learn global hashing function. Among various kinds of fusion methods, recurrent neural network (RNN) architecture is the most common choice, which can well model the temporal structure of videos [17]. Moreover, some of the unsupervised video hashing methods were also proposed [46,27], which organize the hash code learning in a self-taught manner to reduce the time and labor consuming labeling.

2.2 Adversarial Attack

DNNs can be easily fooled to confidently make incorrect predictions by intentional and human-imperceptible perturbations. The process of generating adversarial examples is called *adversarial attack*, which was initially proposed by Szegedy *et al*. [39] in the image classification task. To achieve such adversarial examples, the fast gradient sign method (FGSM) [16] aims to maximize the loss along the gradient direction. After that, projected gradient descent (PGD) [22] was proposed to reach better performance. Deepfool finds the smallest perturbation by exploring the nearest decision boundary [31]. Except for the aforementioned attacks, many other methods [4,10,53,45] have also been developed to find the adversarial perturbation in the image classification problem.

Besides, there are also other DNN based tasks that inherit the vulnerability to adversarial examples [49,11,30,12]. Especially for the deep learning based similarity retrieval, it raises wide concerns on its security issues. For feature-based retrieval, Li *et al*. [25] focused on non-targeted attack by adding universal adversarial perturbations (UAPs), while targeted mismatch adversarial attack was explored in [40]. In [15], adversarial queries for deep product quantization network are generated by perturbing the overall soft-quantized distributions. However, for hashing based retrieval, one of the most important retrieval methods, its robustness analysis is left far behind. There is only one previous work in attacking deep hashing based retrieval [52], which paid attention to the non-targeted attack, *i.e.*, returning retrieval objects with the incorrect label. The targeted attack in such retrieval a system remains unaddressed.

3 The Proposed Method

3.1 Preliminaries

In this section, we briefly review the process of deep hashing based retrieval. Suppose $\boldsymbol{X} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ indicates a set of N sample collection labeled with C classes, where $\boldsymbol{x}_i$ indicates the retrieval object, *e.g.*, a image or a video, and $\boldsymbol{y}_i \in \{0,1\}^C$ corresponds to a label vector. The c-th component of indicator vector $\boldsymbol{y}_i^c = 1$ means that the sample $\boldsymbol{x}_i$ belongs to class c. Let $\boldsymbol{X}^{(t)} = \{(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{X} \mid \boldsymbol{y} = \boldsymbol{y}_t\}$ be a subset of $\boldsymbol{X}$ consisting of those objects with label $\boldsymbol{y}_t$.
Deep Hashing Model. The hash code of a query object $\boldsymbol{x}$ of deep hashing model is generated as follows:

$$\boldsymbol{h} = F(\boldsymbol{x}) = \text{sign}\left(f_\theta(\boldsymbol{x})\right), \tag{1}$$

where $f_\theta(\cdot)$ is a DNN. In general, $f_\theta(\cdot)$ consists of a feature extractor followed by the fully-connected layers. Specifically, the feature extractor is usually specified by a CNN for image retrieval [3,2,7], while CNN stacked with RNN is widely adopted for video retrieval [17,37,27]. In particular, the sign$(\cdot)$ function is approximated by the tanh$(\cdot)$ function during the training process in deep hashing based retrieval methods to alleviate the *gradient vanishing* problem [3].

Similarity-based Retrieval. Given a deep hashing model $F(\cdot)$, a query object $\boldsymbol{x}$ and a object database $\{\boldsymbol{x}_i\}_{i=1}^{M}$, the retrieval process is as follows. Firstly, the query $\boldsymbol{x}$ is fed into the deep hashing model and binary code $F(\boldsymbol{x})$ can be obtained through Eq. (1). Secondly, the Hamming distance between the hash code of query $\boldsymbol{x}$ and that of each object $\boldsymbol{x}_i$ in the database is calculated, denoted as $d_H(F(\boldsymbol{x}), F(\boldsymbol{x}_i))$. Finally, the retrieval system returns a list of objects, which is produced by sorting their Hamming distances.

3.2 Deep Hashing Targeted Attack

Problem Formulation. In general, given a benign query $\boldsymbol{x}$, the objective of targeted attack in retrieval is to generate an attacked version $\boldsymbol{x}'$ of $\boldsymbol{x}$, which would cause the targeted model to retrieve objects with the target label $\boldsymbol{y}_t$. This objective can be achieved through minimizing the distance between the hash code of the attacked sample $\boldsymbol{x}'$ and those of the object subset $\boldsymbol{X}^{(t)}$ with the target label $\boldsymbol{y}_t$, *i.e.*,

$$\min_{\boldsymbol{x}'} d\left(F(\boldsymbol{x}'), F(\boldsymbol{X}^{(t)})\right), \tag{2}$$

where $F(\boldsymbol{X}^{(t)}) = \{F(\boldsymbol{x}) | \boldsymbol{x} \in \boldsymbol{X}^{(t)})\}$, and $d(\cdot, \cdot)$ denotes a point-to-set metric.

Once the problem is formulated as objective (2), the remaining problem is how to define the point-to-set metric. In this paper, we use the most widely used point-to-set metric, the *average-case metric*, as shown in Definition 1.

Definition 1. *Given a point $\boldsymbol{h}_0 \in \{-1, +1\}^K$ and a set of points $\mathcal{A}$ in $\{-1, +1\}^K$ and point-to-point metric d_H, the average-case point-to-set metric is defined as follows:*

$$d_{Ave}\left(\boldsymbol{h}_0, \mathcal{A}\right) \triangleq \frac{1}{|\mathcal{A}|} \sum_{\boldsymbol{h} \in \mathcal{A}} d_H(\boldsymbol{h}_0, \boldsymbol{h}). \tag{3}$$

Remark 1. If average-case point-to-set metric is adopted, the objective function (2) is specified as

$$\min_{\boldsymbol{h}'} \frac{1}{|\mathcal{A}|} \sum_{\boldsymbol{h} \in F(\boldsymbol{X}^{(t)})} d_H(\boldsymbol{h}', \boldsymbol{h}), \tag{4}$$

where $\boldsymbol{h}'$ is the hash code corresponding to the adversarial example $\boldsymbol{x}'$.

In particular, there exists an analytical optimal solution (dubbed *anchor code*) of the optimization problem (4) obtained through a *component-voting scheme*, which is a property arising from the nature of Hamming distance of hashing-based retrieval. The *component-voting scheme* is shown in Algorithm 1, and the optimality of anchor code is verified in Theorem 1. The proof is shown in the **Appendix**.

Theorem 1. *Anchor code $\boldsymbol{h}_a$ calculated by Algorithm 1 is the binary code achieving the minimal sum of Hamming distances with respect to $\boldsymbol{h}_i$, $i = 1, \ldots, n_t$, i.e.,*

$$\boldsymbol{h}_a = \arg \min_{\boldsymbol{h} \in \{+1,-1\}^K} \sum_{i=1}^{n_t} d_H(\boldsymbol{h}, \boldsymbol{h}_i). \tag{5}$$

Algorithm 1 Component-voting Scheme

Input: K-bits hash codes $\{\boldsymbol{h}_i\}_{i=1}^{n_t}$ of objects with the target label t.
Output: Anchor code $\boldsymbol{h}_a$.

1: **for** $j = 1 : K$ **do**

2: Conduct voting process through counting up the number of $+1$ and -1, denoted by N_{+1}^j and N_{-1}^j, respectively. For the j-th component among $\{\boldsymbol{h}_i\}_{i=1}^{n_t}$, *i.e.*,

$$N_{+1}^j = \sum_{i}^{n_t} \mathbb{I}(\boldsymbol{h}_i^j = +1), \qquad N_{-1}^j = \sum_{i=1}^{n_t} \mathbb{I}(\boldsymbol{h}_i^j = -1), \tag{6}$$

where $\mathbb{I}(\cdot)$ is an indicator function.

3: Determine the j-th component of anchor code $\boldsymbol{h}_a^j$ as

$$\boldsymbol{h}_a^j = \begin{cases} +1, & \text{if } N_{+1}^j \geqslant N_{-1}^j \\ -1, & \text{otherwise} \end{cases}. \tag{7}$$

4: **end for**

5: **return** Anchor code $\boldsymbol{h}_a$.

Overall Objective Function. Due to the optimal representative property of anchor code for the set of hash codes of objects with the target label (Theorem 1), we can naturally choose the anchor code as a targeted hash code to direct the generation of the adversarial query. However, the attacked object corresponding to the anchor code may be far different from the original one visually, which would cause the attacked object to be easily detectable. To solve this problem, we introduce the ℓ^∞ restriction on the adversarial perturbations by minimizing the Hamming distance between the hash code of attacked object and that of the anchor code as follows:

$$\min_{\boldsymbol{x}'} d_H(\text{sign}(f_\theta(\boldsymbol{x}')), \boldsymbol{h}_a) \quad s.t. \;\; ||\boldsymbol{x}' - \boldsymbol{x}||_\infty \leq \epsilon, \tag{8}$$

where ϵ denotes the maximum perturbation strength, $\boldsymbol{h}_a$ is the anchor code of object set with the target label.

Besides, given a pair of binary codes $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$, since $d_H(\boldsymbol{h}_i, \boldsymbol{h}_j) = \frac{1}{2}(K - \boldsymbol{h}_i^\top \boldsymbol{h}_j)$, we can equivalently replace Hamming distance with inner product in the objective function. In particular, similar to deep hashing methods [3], we adopt the hyperbolic tangent (tanh) function to approximate sign function for the adversarial generation. Similar to [52], we also introduce the factor α to address the *gradient vanishing* problem. In summary, the overall optimization objective of proposed method is as follows:

$$\min_{\boldsymbol{x}'} \; -\frac{1}{K}\boldsymbol{h}_a^\top \tanh(\alpha f_\theta(\boldsymbol{x}')) \quad s.t. \;\; ||\boldsymbol{x}' - \boldsymbol{x}||_\infty \leq \epsilon, \tag{9}$$

where the hyper-parameter $\alpha \in [0, 1]$, $\boldsymbol{h}_a$ is the anchor code.

The overall process of proposed DHTA is shown in Figure 2.

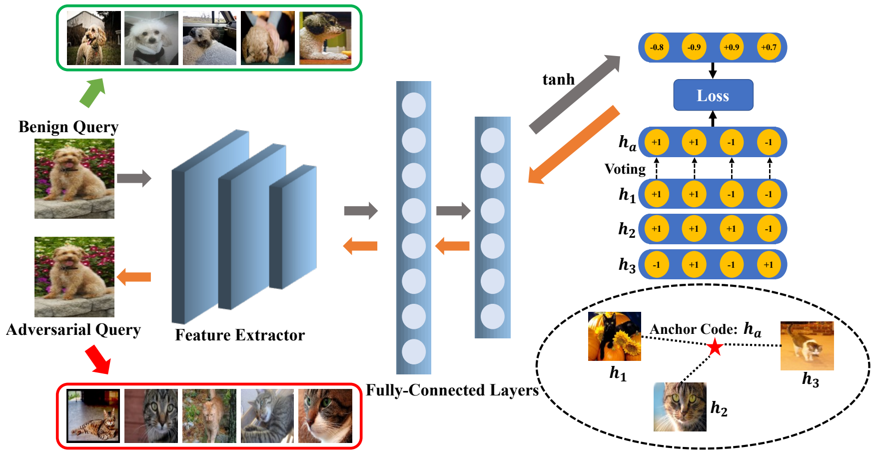


Fig. 2. The pipeline of proposed DHTA method, where the gray and orange arrows indicate forward and backward propagation, respectively. The adversarial query is generated through minimizing the loss calculated by its hash code and the anchor code of the set of objects with the target label. The anchor code h_a is calculated through the component-voting scheme (*i.e.* an entry-wise voting process). In this toy example, $\boldsymbol{h}_1$, $\boldsymbol{h}_2$ and $\boldsymbol{h}_3$ are three 4 bits hash codes of objects with the target label "Cat".

4 Experiments

4.1 Benchmark Datasets and Evaluation Metrics

Four benchmark datasets, including ImageNet [33], NUS-WIDE [9], JHMDB [20], and UCF-101 [38], are adopted in our experiments. The first two datasets are used for image retrieval, while the last two are used for video retrieval. More details about datasets are described in the **Appendix**.

For the evaluation of targeted attacks, we define the targeted mean average precision (t-MAP) as the evaluation metric, which is similar to mean average precision (MAP) widely used in information retrieval [55]. Specifically, the referenced label of t-MAP is the targeted label instead of the original one of the query object in MAP. The higher the t-MAP, the better the targeted attack performance. In image hashing, we evaluate t-MAP on top 5,000 and 1,000 retrieved images on NUS-WIDE and ImageNet, respectively. We evaluate t-MAP on all retrieved videos in video hashing. Besides, we also present the precision-recall curves (PR curves) of different methods for more comprehensive comparison.

4.2 Overall Results on Image Retrieval

Evaluation Setup. For image hashing, we adopt VGG-11 [36] as the backbone network pre-trained on ImageNet to extract features, then replace the last fully-connected layer of softmax classifier with the hashing layer. The detailed settings of training image hashing models are illustrated in the **Appendix**. For each

Table 1. t-MAP (%) of targeted attack methods and MAP (%) of query with benign objects ('Original') with various code lengths on two image datasets.

Method	Metric	ImageNet				NUS-WIDE			
		16bits	32bits	48bits	64bits	16bits	32bits	48bits	64bits
Original	t-MAP	3.80	1.36	1.64	1.98	37.62	36.03	38.32	38.69
Noise	t-MAP	3.29	1.24	1.89	2.10	37.34	36.15	38.25	38.57
P2P	t-MAP	44.35	58.32	62.50	65.61	75.45	78.59	81.40	81.28
DHTA	t-MAP	**63.68**	**77.76**	**82.31**	**82.10**	**82.35**	**85.66**	**86.80**	**88.84**
Original	MAP	51.02	62.70	67.80	70.11	76.93	80.37	82.06	81.62

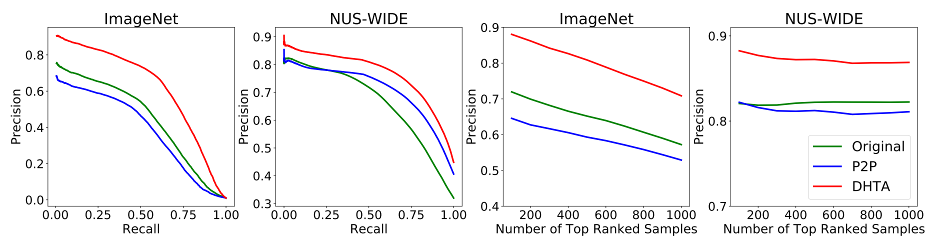


Fig. 3. Precision-recall and precision curves under 48 bits code length in image retrieval. P2P attack and DHTA are evaluated based on the target label, while the result of 'Original' is calculated based on the label of query object.

dataset, we randomly select 100 samples from the query set as benign queries to evaluate the performance of attack. For each generation, we randomly select a label as the target label different from the label of query. When generating an anchor code, we randomly sample images from objects in the database with the target label to form the hash code set. For all adversarial examples, the perturbation magnitude ϵ of normalized data and n_t is set to 0.032 and 9, respectively. We adopt PGD [22] to optimize the proposed attack. We attack image hashing models with learning rate 1 and the number of iterations is set to 2,000. Following [52], the parameter α is set as 0.1 during the first 1,000 iterations, and is updated every 200 iterations according to [0.2, 0.3, 0.5, 0.7, 1.0] during the last 1,000 iterations. We compare DHTA with targeted attack with P2P paradigm [40], which is specified as DHTA with $n_t = 1$. We also show the t-MAP results of images with additive noise sampled from the uniform distribution $U(-\epsilon, +\epsilon)$.

Results. The general attack performance of different methods is shown in Table 1. The t-MAP values of query with benign objects (dubbed *Original*) or query with noisy objects (dubbed *Noise*) are relatively small on both ImageNet and NUS-WIDE datasets. Especially on ImageNet dataset, the t-MAP values of two aforementioned methods are closed to 0, which indicates that query with benign images or images with noise can not successfully retrieve objects with the target labels as expected. In contrast, designed targeted attack methods (*i.e.* P2P and DHTA) can significantly improve the t-MAP values. For example, compared

Fig. 4. An example of image retrieval with benign query and its adversarial query on ImageNet. Retrieved objects with top-10 similarity are shown in the box. The tick and cross indicate whether the retrieved object is consistent with the desired label (the original label for benign query and the target label for adversarial query).

Table 2. t-MAP (%) of targeted attack methods and MAP (%) of query with benign objects ('Original') with various code lengths on two video datasets.

Method	Metric	JHMDB				UCF-101			
		16bits	32bits	48bits	64bits	16bits	32bits	48bits	64bits
Original	t-MAP	6.73	6.26	6.48	6.89	1.69	1.67	1.79	1.86
Noise	t-MAP	6.67	6.13	6.50	6.94	1.69	1.72	1.87	1.85
P2P	t-MAP	39.67	42.37	44.78	44.38	55.57	53.49	55.27	51.88
DHTA	t-MAP	**56.47**	**62.04**	**63.02**	**66.06**	**67.84**	**66.18**	**69.72**	**67.83**
Original	MAP	35.18	42.46	45.80	45.50	55.16	55.25	56.56	56.79

with the t-MAP of benign query on ImageNet dataset, the improvement of P2P methods is over 40% in all cases. Especially under the relatively large code length (64 bits), the improvement even goes to 63%. Among two targeted attack methods, the proposed DHTA method achieves the best performance. Compared with P2P, the t-MAP improvement of DHTA is over 16% (usually over 19%) in all cases on the ImageNet dataset. Moreover, the t-MAP values of targeted attacks increase as the number of bits, which is probably caused by the extra information introduced in the longer code length. In particular, an interesting phenomenon is that the t-MAP value of DHTA is even significantly higher than the MAP value of 'Original', which suggests that the attack performance of DHTA is not hindered by the performance of the original hashing model (*i.e.* threat model) to some extent. An example of the results of query with a benign image and an adversarial image is displayed in Figure 4.

Furthermore, we also provide the precision-recall and precision curves for a more comprehensive comparison. As shown in Figure 3, the curves of DHTA are always above all other curves, which demonstrates that the performance of DHTA does better than all other methods.

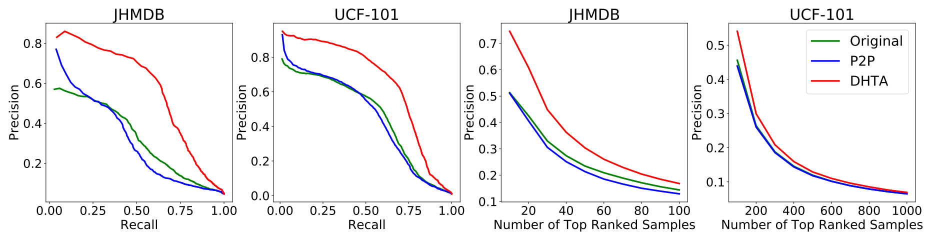


Fig. 5. Precision-recall and precision curves under 48 bits code length in video retrieval. P2P attack and DHTA are evaluated based on the target label, while the result of 'Original' method is calculated based on the label of query object.

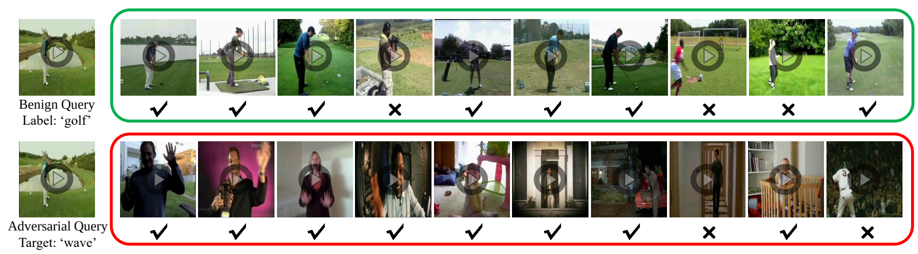


Fig. 6. An example of video retrieval with benign query and its adversarial query on JHMDB. Retrieved objects with top-10 similarity are shown in the box. The tick and cross indicate whether the retrieved object is consistent with the desired label (the original label for benign query and the target label for adversarial query).

4.3 Overall Results on Video Retrieval

Evaluation Setup. According to model architectures of the state-of-the-art deep video retrieval methods [17,37,27], we adopt AlexNet [21] to extract spatial features and LSTM [18] to fuse the temporal information. The detailed settings of training video hashing model are presented in the **Appendix**. For attacking video hashing, the number of iterations is 500, and the parameter α is fixed at 0.1. Other settings are the same as those used in Section 4.2.

Results. The attack performance in video retrieval is shown in Table 2. Similar to the image scenario, query with benign videos or videos with noise can not successfully retrieve objects with the target label, thus fails to attack the deep hashing based retrieval. In contrast, deep hashing based video retrieval can be easily attacked by designed targeted attacks, especially the DHTA proposed in this paper. For example, the t-MAP value of DHTA is 59% over query with benign videos, and 21% over P2P attack paradigm on the JHMDB dataset with code length 64 bits. The precision-recall and the precision curves also verify the superiority of DHTA over other methods, as shown in Figure 5. Especially on JHMDB dataset, there exists a significantly large gap between the PR curve of DHTA and those of other methods. In addition, the t-MAP value of DHTA is again significantly larger than the MAP of the benign query (the 'Original'). An

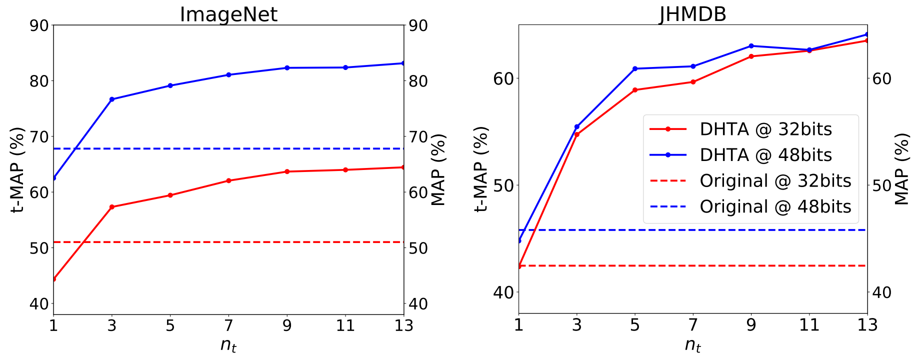


Fig. 7. t-MAP (%) of DHTA and MAP (%) of query with benign objects ('Original') with different n_t and code length on ImageNet and JHMDB.

example of the results of query with a benign video and an adversarial video is displayed in Figure 6.

4.4 Discussion

Effect of n_t. To analyze the effect of the size of object set for generating the anchor code (*i.e.*, n_t), we discuss the t-MAP of DHTA under different values of $n_t \in \{1, 3, 5, 7, 9, 11, 13\}$. Other settings are the same as those used in Section 4.2-4.3. We use ImageNet and JHMDB as the representative for analysis.

As shown in Figure 7, the t-MAP value increase as the increase of n_t under different code lengths. The MAP of corresponding query with benign objects (*i.e.* the 'Original') can be regarded as the reference of the retrieval performance. We observe that the t-MAP is higher than the MAP of its corresponding 'Original' method in all cases when $n_t \geq 3$. In other words, DHTA can still have satisfying performance with relatively small n_t. This advantage is critical for attackers, since the bigger the n_t, the higher the cost of data collection and adversarial generation for an attack. It is worth noting that the attack performance degrades significantly when $n_t = 1$, which exactly corresponds to the P2P attack paradigm.

Effect of the Number of Iterations. Table 3-4 present the t-MAP of DHTA with different iterations on ImageNet and JHMDB datasets. Except for the iterations, other settings are the same as those used in Section 4.2-4.3.

As expected, the t-MAP values increase with the number of iterations. Even with relatively few iterations, the proposed DHTA can still achieve satisfying performance. For example, with 100 iterations, the t-MAP values are over 50% under all code lengths. Especially on the ImageNet dataset, the t-MAP is over 70% with relatively larger code length (≥48 bits). These results consistently verify the high-efficiency of our DHTA method.

Table 3. t-MAP (%) of DHTA with different iterations on ImageNet.

Iteration	16bits	32bits	48bits	64bits
100	52.99	66.29	70.65	72.43
500	55.18	68.30	74.47	76.15
1000	56.96	68.36	75.03	76.25
1500	62.81	74.11	79.28	78.71
2000	63.68	77.76	82.31	82.10

Table 4. t-MAP (%) of DHTA with different iterations on JHMDB.

Iteration	16bits	32bits	48bits	64bits
10	28.51	23.88	22.84	23.21
50	48.69	48.18	47.01	48.97
100	53.21	54.91	55.94	58.28
500	56.47	62.04	63.02	66.06

Table 5. MAP (%) of different methods on ImageNet and JHMDB. The best results are marked with boldface, while the second best results are marked with underline.

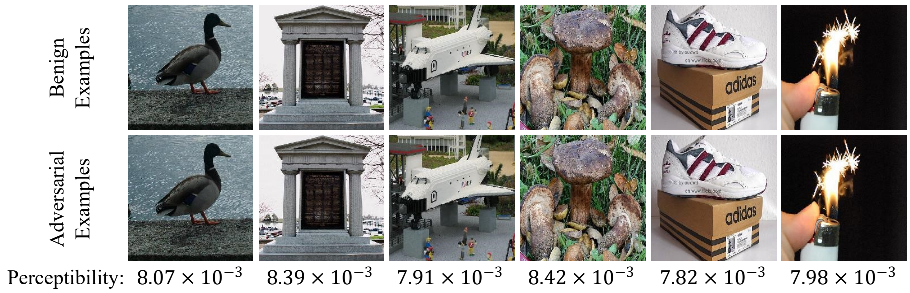
Method	ImageNet				JHMDB			
	16bits	32bits	48bits	64bits	16bits	32bits	48bits	64bits
Original	51.02	62.70	67.80	70.11	35.18	42.46	45.80	45.50
Noise	50.94	62.52	66.69	69.85	35.04	42.15	45.67	45.63
P2P	3.36	**2.48**	<u>2.45</u>	3.93	7.71	8.20	8.14	10.19
HAG	<u>1.88</u>	4.96	3.89	<u>2.34</u>	**3.52**	**3.58**	**3.42**	**3.34**
DHTA	**0.54**	<u>5.64</u>	**2.30**	**1.70**	<u>6.76</u>	<u>7.23</u>	<u>6.56</u>	<u>7.55</u>

Evaluation from the Perspective of Non-targeted Attack. Targeted attack can be regarded as a special non-targeted attack, since the target label is usually different from the one of query object. In this part, we compare the targeted attacks (P2P and DHTA) with other methods, including additive noise and HAG [52] (which is the state-of-the-art non-targeted attack), in the non-targeted attack scenario.

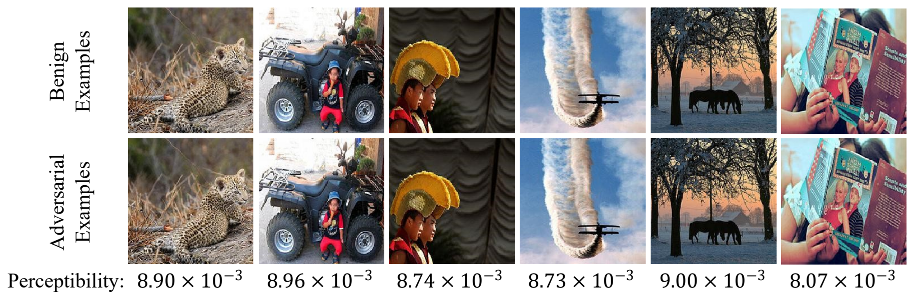
The MAP results of different methods are reported in Table 5. The lower the MAP, the better the non-targeted attack performance. As shown in the table, although targeted attacks are not designed for the non-targeted scenario, they still have competitive performance. For example, the MAP values of DHTA are 50% smaller than those of 'Original' under all code length on ImageNet. Especially for the proposed DHTA, it even has better non-targeted attack performance (*i.e.* smaller MAP) compared with HAG on ImageNet in most cases.

Perceptibility. Except for the attack performance, the *perceptibility* of adversarial perturbations is also important. Following the setting suggested in [39,41], given a benign query $\boldsymbol{x}$, the perceptibility of its corresponding adversarial query $\boldsymbol{x}'$ is defined as $\sqrt{\frac{1}{n}\left\|\boldsymbol{x}'-\boldsymbol{x}\right\|_2^2}$, where n is the size of the object and pixel values are scaled to be in the range [0, 1].

For each dataset, we calculate the average perceptibility over all generated adversarial objects. The perceptibility value of ImageNet and NUS-WIDE datasets is 8.35×10^{-3} and 9.07×10^{-3}, respectively. In video retrieval tasks, the value is 5.81×10^{-3} and 7.72×10^{-3} on JHMDB and UCF-101 datasets, respectively. These results indicate that the adversarial queries are very similar to their orig-



(a) ImageNet



(b) NUS-WIDE

Fig. 8. Visualization examples of generated adversarial examples in image hashing.

inal versions. Some adversarial images are shown in Figure 8, while examples of video retrieval are shown in the **Appendix**.

4.5 Open-set Targeted Attack

Evaluation Setup. In the above experiments, the target label is selected from those of training set. In this section, we use ImageNet dataset as an example to further evaluate the proposed DHTA under a tougher open-set scenario, where the out-of-sample class will be assigned as the target label. This setting is more realistic since the attacker may probably not be able to access the training set of the attacked deep hashing model. For example, the deep hashing model may be downloaded from a third-party open-source platform where the training set is unavailable.

Specifically, we randomly select 10 additional classes different from those used for training a deep hashing model in Section 4.1. These selected images from 10 additional classes will be treated as an open set for our evaluation. When generating the anchor code of objects with the target label (within the open set), we remain our deep hashing model trained on the previous 100 classes.
Results. As shown in Table 6, DHTA still has a certain attack effect even if the target label is out-of-sample. Especially when the n_t and the code length

Table 6. t-MAP (%) of DHTA with out-of-sample target label on ImageNet.

Method	16bits	32bits	48bits	64bits
DHTA ($n_t = 5$)	33.67	46.34	48.91	48.27
DHTA ($n_t = 7$)	34.77	50.92	51.68	49.18
DHTA ($n_t = 9$)	37.34	54.13	55.12	52.17
DHTA ($n_t = 11$)	38.00	54.05	56.93	54.12

tend larger, the t-MAP values of DHTA are over 50%. This phenomenon may reveal that the learned feature extractor did learn some useful low-level features, which represents those objects with the same class in some similar locations in Hamming space, no matter the class is learned or not. In addition, the attack performance is also increasing with the n_t and code length.

5 Conclusion and Future Work

In this paper, we explore the landscape of the targeted attack for deep hashing based retrieval. Based on the characteristics of the retrieval task, we formulate the attack as a point-to-set optimization, which minimizes the average distance between the hash code of the adversarial example and those of a set of objects with the target label. Theoretically, we propose a component-voting scheme to obtain the optimal representative, the anchor code, for the code set of point-to-set optimization. Based on the anchor code, we propose a novel targeted attack method, the DHTA, to balance the performance and perceptibility through minimizing the Hamming distance between the hash code of adversarial example and the anchor code under the ℓ^∞ restriction on the adversarial perturbation. Extensive experiments are conducted, which verifies the effectiveness of DHTA in attacking both deep hashing based image retrieval and video retrieval. To alleviate the proposed threat, we will discuss how to generalize existing adversarial training based methods from P2P to the P2S scheme for the defense. The specific approaches will be further demonstrated in our future works.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800204, the National Natural Science Foundation of China under Grant 61771273, the R&D Program of Shenzhen under Grant JCYJ20180508152204044, the project "PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications (LZC0019)", the Natutal Sciences and Engineering Research Council of Canada under Grant RGPIN203035-16, and the Canada Research Chairs Program. We also thank vivo and Rejoice Sport Tech. co., LTD. for their GPUs.

References

1. Bai, Y., Zeng, Y., Jiang, Y., Wang, Y., Xia, S.T., Guo, W.: Improving query efficiency of black-box adversarial attack. In: ECCV (2020)
2. Cao, Y., Long, M., Liu, B., Wang, J.: Deep cauchy hashing for hamming space retrieval. In: CVPR (2018)
3. Cao, Z., Long, M., Wang, J., Yu, P.S.: Hashnet: Deep learning to hash by continuation. In: ICCV (2017)
4. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE S&P (2017)
5. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: IEEE S&P Workshops (2018)
6. Chen, W., Zhang, Z., Hu, X., Wu, B.: Boosting decision-based black-box adversarial attacks with random sign flip. In: ECCV (2020)
7. Chen, Y., Lai, Z., Ding, Y., Lin, K., Wong, W.K.: Deep supervised hashing with anchor graph. In: CVPR (2019)
8. Chen, Z., Yuan, X., Lu, J., Tian, Q., Zhou, J.: Deep hashing via discrepancy minimization. In: CVPR (2018)
9. 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: ICMR (2009)
10. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: CVPR (2018)
11. Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., Zhu, J.: Efficient decision-based black-box adversarial attacks on face recognition. In: CVPR (2019)
12. Duan, R., Ma, X., Wang, Y., Bailey, J., Qin, A.K., Yang, Y.: Adversarial camouflage: Hiding physical-world attacks with natural styles. In: CVPR (2020)
13. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust physical-world attacks on deep learning visual classification. In: CVPR (2018)
14. Fan, Y., Wu, B., Li, T., Zhang, Y., Li, M., Li, Z., Yang, Y.: Sparse adversarial attack via perturbation factorization. In: ECCV (2020)
15. Feng, Y., Chen, B., Dai, T., Xia, S.t.: Adversarial attack on deep product quantization network for image retrieval. In: AAAI (2020)
16. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
17. Gu, Y., Ma, C., Yang, J.: Supervised recurrent hashing for large scale video retrieval. In: ACM MM (2016)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
19. Hu, D., Nie, F., Li, X.: Deep binary reconstruction for cross-modal hashing. IEEE Transactions on Multimedia **21**(4), 973–985 (2018)
20. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV (2013)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS. pp. 1097–1105 (2012)
22. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. In: ICLR (2017)
23. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: CVPR (2015)

24. Li, C., Deng, C., Li, N., Liu, W., Gao, X., Tao, D.: Self-supervised adversarial hashing networks for cross-modal retrieval. In: CVPR (2018)
25. Li, J., Ji, R., Liu, H., Hong, X., Gao, Y., Tian, Q.: Universal perturbation attack against image retrieval. In: ICCV (2019)
26. Li, P., Wang, M., Cheng, J., Xu, C., Lu, H.: Spectral hashing with semantically consistent graph for image indexing. IEEE Transactions on Multimedia **15**(1), 141–152 (2012)
27. Li, S., Chen, Z., Lu, J., Li, X., Zhou, J.: Neighborhood preserving hashing for scalable video retrieval. In: ICCV. pp. 8212–8221 (2019)
28. Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: CVPR (2016)
29. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: CVPR (2012)
30. Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J., Lu, F.: Understanding adversarial attacks on deep learning based medical image analysis systems. Pattern Recognition (2019)
31. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: CVPR (2016)
32. Qin, Y., Carlini, N., Cottrell, G., Goodfellow, I., Raffel, C.: Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In: ICML (2019)
33. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
34. Shen, F., Shen, C., Liu, W., Tao Shen, H.: Supervised discrete hashing. In: CVPR (2015)
35. Shen, F., Xu, Y., Liu, L., Yang, Y., Huang, Z., Shen, H.T.: Unsupervised deep hashing with similarity-adaptive and discrete optimization. IEEE transactions on pattern analysis and machine intelligence **40**(12), 3034–3044 (2018)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
37. Song, J., Zhang, H., Li, X., Gao, L., Wang, M., Hong, R.: Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Transactions on Image Processing **27**(7), 3210–3221 (2018)
38. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
39. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2014)
40. Tolias, G., Radenovic, F., Chum, O.: Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In: ICCV (2019)
41. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. In: ICLR (2018)
42. Wang, J., Zhang, T., Sebe, N., Shen, H.T., et al.: A survey on learning to hash. IEEE transactions on pattern analysis and machine intelligence **40**(4), 769–790 (2017)
43. Wang, J., Liu, W., Kumar, S., Chang, S.F.: Learning to hash for indexing big data—a survey. Proceedings of the IEEE **104**(1), 34–57 (2015)
44. Wu, D., Lin, Z., Li, B., Ye, M., Wang, W.: Deep supervised hashing for multi-label and large-scale image retrieval. In: ICMR (2017)
45. Wu, D., Wang, Y., Xia, S.T., Bailey, J., Ma, X.: Skip connections matter: On the transferability of adversarial examples generated with resnets. In: ICLR (2020)

46. Wu, G., Han, J., Guo, Y., Liu, L., Ding, G., Ni, Q., Shao, L.: Unsupervised deep video hashing via balanced code for large-scale video retrieval. IEEE Transactions on Image Processing **28**(4), 1993–2007 (2018)
47. Xia, R., Pan, Y., Lai, H., Liu, C., Yan, S.: Supervised hashing for image retrieval via image representation learning. In: AAAI (2014)
48. Xu, Y., Wu, B., Shen, F., Fan, Y., Zhang, Y., Shen, H.T., Liu, W.: Exact adversarial attack to image captioning via structured output learning with latent variables. In: CVPR (2019)
49. Xu, Y., Wu, B., Shen, F., Fan, Y., Zhang, Y., Shen, H.T., Liu, W.: Exact adversarial attack to image captioning via structured output learning with latent variables. In: CVPR (2019)
50. Yan, X., Zhang, L., Li, W.J.: Semi-supervised deep hashing with a bipartite graph. In: IJCAI (2017)
51. Yang, E., Liu, T., Deng, C., Liu, W., Tao, D.: Distillhash: Unsupervised deep hashing by distilling data pairs. In: CVPR (2019)
52. Yang, E., Liu, T., Deng, C., Tao, D.: Adversarial examples for hamming space search. IEEE transactions on cybernetics **50**(4), 1473–1484 (2018)
53. Yao, Z., Gholami, A., Xu, P., Keutzer, K., Mahoney, M.W.: Trust region based adversarial attack on neural networks. In: CVPR (2019)
54. Zhao, F., Huang, Y., Wang, L., Tan, T.: Deep semantic ranking based hashing for multi-label image retrieval. In: CVPR (2015)
55. Zuva, K., Zuva, T.: Evaluation of information retrieval systems. International journal of computer science & information technology **4**(3), 35 (2012)