A Smart Adversarial Attack on Deep Hashing Based Image Retrieval

Junda Lu*
University of New South Wales
Sydney, Australia
junda.lu@student.unsw.edu.au

Mingyang Chen*
University of New South Wales
Sydney, Australia
mingyang.chen1@unsw.edu.au

Yifang Sun
Northeastern University
Shenyang, China
sunyifang@cse.neu.edu.cn

Wei Wang†
The Hong Kong University of Science and Technology
HKSAR, China
weiwcs@ust.hk

Yi Wang‡
Dongguan University of Technology
Dongguan, China
wangyi@dgut.edu.cn

Xiaochun Yang
Northeastern University
Shenyang, China
yangxc@mail.neu.edu.cn

ABSTRACT

Deep hashing based retrieval models have been widely used in large-scale image retrieval systems. Recently, there has been a surging interest in studying the adversarial attack problem in deep hashing based retrieval models. However, the effectiveness of existing adversarial attacks is limited by their poor perturbation management, unawareness of ranking weight, and only laser-focusing on the attack image. These shortages lead to high perturbation costs yet low AP reductions. To overcome these shortages, we propose a novel adversarial attack framework to improve the effectiveness of adversarial attacks. Our attack designs a dimension-wise surrogate Hamming distance function to help with wiser perturbation management. Further, in generating adversarial examples, instead of focusing on a single image, we propose to collectively incorporate relevant images combined with an AP-oriented (average precision) weight function. In addition, our attack can deal with both untargeted and targeted adversarial attacks in a flexible manner. Extensive experiments demonstrate that, with the same attack performance, our model significantly outperforms state-of-the-art models in perturbation cost on both untargeted and targeted attack tasks.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision problems**;
• **Security and privacy** → *Software and application security*.

KEYWORDS

image retrieval; adversarial attack; deep hashing

*The first two authors have equal contributions.
†Corresponding author.
‡ORCID of Yi Wang: 0000-0002-8448-8570.

ICMR '21, August 21–24, 2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8463-6/21/08.
https://doi.org/10.1145/3460426.3463640

ACM Reference Format:
Junda Lu, Mingyang Chen, Yifang Sun, Wei Wang, Yi Wang, and Xiaochun Yang. 2021. A Smart Adversarial Attack on Deep Hashing Based Image Retrieval. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460426.3463640

1 INTRODUCTION

Image retrieval is a long-established research task for retrieving content-similar images in massive image collections. Over the past few years, deep hashing have become a predominant image retrieval technique as being proved successful in terms of both efficiency and accuracy. Deep hashing models typically build upon powerful Deep Neural Networks (DNNs) [13, 18, 28, 29] to automatically generate informative embedding vectors for the images, and incorporate various hashing functions to generate *binary* hash codes. Relying on the efficient Hamming search along with corresponding filtering techniques, a single retrieval could be done in sub-linear time. Some well-known examples, including Pinterest [35] and Alibaba [36], have shown promising performance and become the back-bone for real industrial applications.

Like other DNNs-based models (e.g., image classification), DNNs-based hashing retrieval models are also proved to be vulnerable to adversarial attacks [34]. This means the outputs of deep hashing retrieval models would be readily misled or affected by an adversarial image $\tilde{\mathbf{q}}$ which added a small perturbation on the original query image $\mathbf{q}$. Functionally, the adversarial attacks could be further classified into *untargeted attack* [26, 34, 38] and *targeted attack* [3, 30], according to whether certain malicious content or labels are specified by the attackers.

Regardless of being effective, existing adversarial attack algorithms suffer from high image perturbation costs which makes the attacks easier to be noticed. The drawbacks of existing approaches are specified in the following aspects:

i) For the objective function, the simple inner product based surrogate distance function employed by existing approaches will blindly reward larger Hamming distances while ignoring image perturbation cost. In fact, *each* dimension of a hash vector is only capable of contributing one unit of Hamming distance, once achieved, it is in vain to make a further modification. This is also in accordance with the principle of diminishing marginal utility – a good

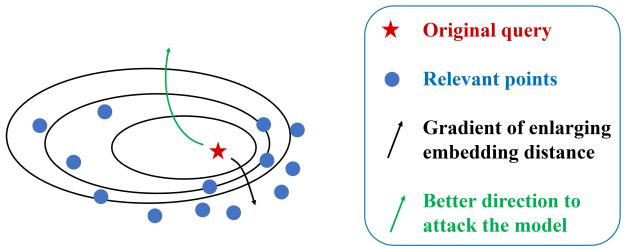

Figure 1: An example to show that considering relevant images of query is important and blindly pushing the embedding of $\tilde{q}$ away from that of q can be ineffective.

attack model should make wise perturbation management that judiciously gives more priorities to the dimensions to gain maximal Hamming distances yet minimal perturbation costs.

ii) In generating an adversarial example for an image, existing untargeted attack models only consider the query image itself while forgetting its relevant images. Intuitively, use relevant images could help with a more effective adversarial example by rectifying errors of anomaly single image to facilitate better AP reductions. An example is shown in Figure 1.

iii) Existing adversarial attack models are not ranking-sensitive. In fact, the retrieval result is a ranking list, the ranking of each relevant image has a different weight on the computing of AP. As it is shown in Figure 2, a weighting schema manifested in computing AP. One shall get a further AP reduction by taking the ranking weights into consideration.

In this paper, we propose a novel adversarial attack method (namely, SDHA, *Smart Deep Hashing Attack*). Our method employs a dimension-wise Hamming distance surrogate function, which rewards the modifications on these dimensions whose potential gains in Hamming distances are worth paying for the perturbation. Our surrogate function is of a better perturbation management strategy to lower the image perturbation cost (address i). In computing surrogate Hamming distance of untargeted attack, instead of just considering the query image itself, we take its all relevant images into consideration. In this way, the AP could be greatly reduced as relevant images contribute to the AP (address ii). As the image ranks higher contributes more to AP, we further design an AP-oriented weight function that assigns different weight to different ranking position, which makes the attack focuses on the images with higher rankings to decrease AP (address iii). As such, our model could achieve a lower image perturbation cost with more AP reduction. Moreover, our model is flexible enough to readily support both untargeted and targeted attacks.

Our main contributions can be summarized as follow:

- We propose a novel adversarial attack framework for deep hashing based image retrieval systems, which aims to generate adversarial queries with small perturbation in the image space.
- We propose a sophisticated surrogate objective function which considers the influence of relevant images, and the different contribution of embedding dimensions. We further propose an AP-oriented weight function to enhance the performance.

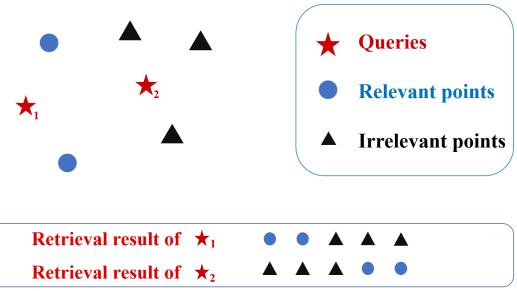

Figure 2: An example to show that the ranking of relevant images is important to the retrieval performance. The relevant images with higher ranking contribute more to the retrieval performance. (E.g., The average precision (AP) of top-5 points with two relevant points of q_1 is $\frac{1}{2}(1+1+0+0+0) = 1$, while the AP of q_2 is $\frac{1}{2}\left(0+0+0+\frac{1}{4}+\frac{2}{5}\right) = 0.325$.)

- We show that the proposed framework is flexible. It can be generated to both untargeted and targeted adversarial attack problems, by slightly changing the objective function.
- We demonstrate that the proposed framework consistently generates better adversarial queries than the state-of-the-art methods on different datasets with different output dimensions, which implies the effectiveness of our design.

We will elaborate on each point in the following sections. We review related works in Section 2 and introduce preliminaries in Section 3. We propose our framework in Section 4. Experimental results are shown in Section 5 and Section 6 makes a conclusion.

2 RELATED WORK

2.1 Deep Hashing Based Retrieval Systems

Deep retrieval models are based on DNNs with triplet loss [5] or approximated AP loss [4] to train the models to get high retrieval performance. While deep hashing models use binary hash codes as the embedding to represent objects thus got high efficiency for large-scale and high-dimensional data [14, 20, 24, 25]. It benefits from both the retrieval efficiency of hashing and the powerful extractability of DNNs, thus it got promising retrieval performance. Hashing methods consist of supervised [25, 31, 32, 37] and unsupervised methods [10, 17, 22, 24].

DHN [40] is the first end-to-end supervised deep hashing method and benefits from the powerful extractability of DNNs. It used pairwise loss and it also controlled the quantization error. HashNet [6] balanced the positive and negative pairs in the training dataset and it uses the continuation technique to get lower quantization error. DCH [5] used Cauchy distribution to design a pairwise cross-entropy loss, which penalized significantly on relevant image pairs with Hamming distance larger than a threshold, to encourage relevant images to be located in a small Hamming Ball. Unsupervised deep hashing models are training with unlabelled data. DeepBit [22] used DNNs to learn the non-linear projection functions to compute the binary codes which have minimal quantization loss. Distill-Hash [33] learned a distilled dataset consisted of data pairs with confidence similarity signals to improve the performance.

2.2 Adversarial Attack on Retrieval Systems

In recent years, studies found that deep classification models are vulnerable to adversarial attacks which add a small well-designed perturbation on the input to cause a wrong classification prediction. Various attacks had been proposed, such as Fast Gradient Sign Methods (FGSM) [12], DeepFool [27], PGD [19], and CW attack [8].

Adversarial attacks on deep retrieval systems also attract interest [1, 2]. The lacking of classification confidence value makes the attack on retrieval systems harder to design. It is even harder for deep hashing retrieval models as their embeddings are binary codes and hard to calculate the gradient. Currently, researchers have proposed their solutions, but most of their ideas are intuitive. PIRE [26] tried its best to push the adversarial query image away from the original query in the embedding space to generate an untargeted attack. UAP [21] used a similar idea but train the adversarial attack on another dataset to generate a universal adversarial perturbation. UAA-GAN [38] pushed the query away and used GAN [11] to make the perturbation hard to be noticed by humans. [39] modified either query or candidate images to change the ranking of specific retrieval images. Targeted Mismatch Attack [30] moved the query's embedding close to that of a target image to generate a targeted attack. For the attack on deep hashing based retrieval models, HAG [34] is the state-of-the-art untargeted method, it solved the vanishing gradient problem but it still used the idea of pushing the embedding of adversary query away from that of the original one. For the targeted attack, DHTA [3] improved [30] by calculating an anchor code and then move the adversarial query close to it.

3 PRELIMINARIES

In this paper, we consider a deep neural network based image retrieval system, where the DNNs converts each image into a binary embedding vector (also known as the hash code). When receiving a query image, the retrieval system returns T database images whose embedding vectors are the closest to the query's embedding. Hamming distance is used to rank the embeddings and MAP is used to evaluate the performance of the retrieval system.

DNNs Based Image Retrieval System. Let $\mathbf{q}$ be the query image, $\mathcal{X}$ be the database that contains all the images, and $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$ be the set of images in $\mathcal{X}$ that are relevant to $\mathbf{q}$. Let $E(\cdot)$ be the deep neural network, then the binary hash code of $\mathbf{x}$ is obtained as $h(\mathbf{x}) = E(\mathbf{x}) = sgn(g(\mathbf{x}))$, such that $h(\mathbf{x}) \in \{1, -1\}^k$, where $sgn(\cdot)$ is the last layer of $E(\cdot)$, $g(\cdot)$ is the penultimate layer and k is the output dimensionality of $E(\cdot)$. $sgn(x) = 1$ if $x > 0$, and $sgn(x) = -1$ otherwise. In addition, current deep hashing methods usually approximate $sgn(\cdot)$ with $tanh(\cdot)$ to alleviate the gradient vanishing problem [7] during the training process. Therefore, in this paper, we define $f(\mathbf{x}) = tanh(g(\mathbf{x}))$ as the approximation of $h(\mathbf{x})$, and $f(\mathbf{x}) \in (-1, 1)$.

Mean Average Precision. Mean Average Precision (MAP) is one of the most popular measurements to evaluate the quality of a retrieval system. It is computed based on a set of queries Q, more specifically, it is the mean of the AP (average precision) computed for each constituent query $\mathbf{q} \in Q$, defined as

$$AP(\mathbf{q}) = \frac{1}{n^+} \sum_{i=1}^{n} prec(i) \cdot \mathbb{1}(G_i = 1),$$

where $\mathbb{1}(G_i = 1)$ is the indicator function on whether the i-th element in the ranked result list is relevant to $\mathbf{q}$ (i.e., $\mathbb{1}(G_i = 1)$ equals 1 if G_i is relevant to $\mathbf{q}$, and equals 0 otherwise), $prec(i)$ is the precision at cut-off i in the ranked result list, and n^+ is the total number of relevant points in the ranked result list.

4 SMART DEEP HASHING ATTACK

In this section, we will first consider the untargeted setting by defining the attack problem and then propose our framework in details. We will then present the extension of our solution to the targeted adversarial attack setting. Finally, we will discuss the implementation details when solving the proposed optimization problems.

4.1 Problem Definition

Consider a deep hashing based image retrieval system as described in Section 3, given a query image $\mathbf{q}$, the attacker aims to generate an adversarial query $\tilde{\mathbf{q}}$ such that:

(1) the difference between $\tilde{\mathbf{q}}$ and $\mathbf{q}$ is imperceptible, and
(2) the quality of the retrieval result of $\tilde{\mathbf{q}}$ satisfies the attacker specified objectives.

To make sure $\tilde{\mathbf{q}}$ do not differ from $\mathbf{q}$ visually, we constrain both the Euclidean distance and the ℓ^∞ distance between them.

- The Euclidean distance between $\tilde{\mathbf{q}}$ and $\mathbf{q}$ should be as small as possible.
- The ℓ^∞ distance between $\tilde{\mathbf{q}}$ and $\mathbf{q}$ should not exceed predefined threshold ϵ.

For the second condition, we consider two different attack scenarios, namely *untargeted* attack and *targeted* attack. More specifically,

- let $\mathbf{y}_u$ be the true label of the original query image $\mathbf{q}$, *untargeted* attack aims to reduce the average precision of $\tilde{\mathbf{q}}$ with respect to $\mathbf{y}_u$, such that the AP of the returned retrieval result of $\tilde{\mathbf{q}}$ is sufficiently *low*.
- Let $\mathbf{y}_t$ be the user specified target label, *targeted* attack aims to improve the average precision of $\tilde{\mathbf{q}}$ with respect to $\mathbf{y}_t$, such that the AP of the returned retrieval result of $\tilde{\mathbf{q}}$ is sufficiently *high*.

We consider the white-box attack scenario where the attacker can access all the parameters of $E(\cdot)$ and all the images $\mathbf{x} \in \mathcal{X}$.

4.2 Attack Framework

We propose to solve both untargeted and targeted adversarial attack with a flexible attack framework. The main idea is to generate the adversarial query by solving the optimization problem as below:

$$\begin{aligned} \underset{\tilde{q}}{\text{minimize}} \quad & \|\tilde{\mathbf{q}} - \mathbf{q}\|_2 \\ \text{subject to} \quad & \|\tilde{\mathbf{q}} - \mathbf{q}\|_\infty \le \epsilon, \\ & \|\tilde{\mathbf{q}}\|_\infty \in [0, 255], \\ & AP(\tilde{\mathbf{q}}) \text{ opt } \delta. \end{aligned}$$

Where the objective function corresponds to the first condition as defined in Section 4.1, and the first two constraints correspond

to the restriction of the ℓ^∞ distance between $\tilde{\mathbf{q}}$ and $\mathbf{q}$. The δ in the last constraint corresponds to the AP requirements for the attack, we let opt be $\leq$ for untargeted adversarial attack and $\geq$ for targeted adversarial attack, respectively.

Using a positive Lagrange multiplier α, we can lift the AP constraint into the objective function. In practice, we can use binary search to find the appropriate α value to optimize, similar to [8]. However, even with this trick, the optimization problem is still hard to solve numerically due to the discrete nature of $AP(\tilde{\mathbf{q}})$.

Hence, we design a surrogate objective function, $F(\tilde{\mathbf{q}})$, to replace $AP(\tilde{\mathbf{q}})$ based on the following intuition: if the Hamming distance between $\tilde{\mathbf{q}}$ and a relevant image, $\mathbf{r}_i$, of $\mathbf{q}$ in the embedding space is large, then $\mathbf{r}_i$ is expected to rank low or even outside the retrieved results of $\tilde{\mathbf{q}}$. Thus, if $\tilde{\mathbf{q}}$ is faraway from *all* of the relevant images, then its AP will be low, and vice versa.

We now formulate the below optimization problem for both untargeted and targeted adversarial attack:

$$\begin{aligned} \underset{\tilde{q}}{\text{minimize}} \quad & \|\tilde{\mathbf{q}} - \mathbf{q}\|_2 + \alpha \cdot F(\tilde{\mathbf{q}}) \\ \text{subject to} \quad & \|\tilde{\mathbf{q}} - \mathbf{q}\|_\infty \leq \epsilon, \\ & \|\tilde{\mathbf{q}}\|_\infty \in [0, 255]. \end{aligned} \tag{1}$$

Where the detail of the surrogate objective function $F(\tilde{\mathbf{q}})$ will be discussed in Section 4.3. Now, we have formulate the optimization problem of attack which optimize both AP (e.g., $F(\tilde{\mathbf{q}})$) and l_2 perturbation (e.g., $\|\tilde{\mathbf{q}} - \mathbf{q}\|_2$). While the previous method (e.g., HAG) only moves $f(\tilde{\mathbf{q}})$ away from $f(\mathbf{q})$, which do not consider measurement metric and l_2 distance.

4.3 Modelling the Surrogate Objective Function

In this subsection, we first use an untargeted adversarial attack to motivate and illustrate the design of $F(\tilde{\mathbf{q}})$, and then we extend the idea to solve the targeted adversarial attack.

4.3.1 Untargeted Adversarial Attack. As we discussed, the objective of $F(\tilde{\mathbf{q}})$ is to enlarge the Hamming distance between $\mathbf{q}$ and all $\mathbf{r}_i$, $F_u(\tilde{\mathbf{q}})$ for untargeted attack can be naively defined as:

$$F_u(\tilde{\mathbf{q}}) = -\frac{1}{n_u} \sum_{i=1}^{n_u} d_u(\tilde{\mathbf{q}}, \mathbf{r}_i), \tag{2}$$

where n_u is the total number of relevant images to $\mathbf{q}$, $d_u(\cdot)$ is the Hamming distance surrogate function. We add " $-$ " sign so as to minimize it by enlarging the surrogate Hamming distance. In the following subsection, we will discuss the designing the sophisticated Hamming distance surrogate function, and the designing of the AP-oriented weight function. We firstly analyze the drawbacks of the current surrogate function and then propose our approaches.

Design Hamming Distance Surrogate Function. Because the embedding vector of $\tilde{\mathbf{q}}$ (i.e., $h(\tilde{\mathbf{q}}) = sgn(f(\tilde{\mathbf{q}}))$) is discrete (i.e., $h(\tilde{\mathbf{q}}) \in \{1, -1\}^k$), we aim at $f(\tilde{\mathbf{q}})$ which is a real vector to calculate the surrogate Hamming distance. The previous method uses the inner product based function of $f(\tilde{\mathbf{q}})$ and $h(\mathbf{q})$ (e.g., $d_{inner}(\tilde{\mathbf{q}}, \mathbf{q}) = \frac{k - f(\tilde{\mathbf{q}})^\top h(\mathbf{q})}{2}$) as the surrogate Hamming distance and push $f(\tilde{\mathbf{q}})$ away from $h(\mathbf{q})$.

This inner product based surrogate function is inefficient because of the following 2 aspects, i) it does not consider any of the related points $\mathbf{r}_i$ which contribute to the AP, and ii) it does not consider the different contributions of different dimensions to the Hamming distance, and it assigns each dimension with the same gradient value.

For the aspect i, it does not consider $\mathbf{r}_i$ which contributes to AP. Thus it has less relationship with the attack goal (e.g., decreasing AP). Without the guidance by relevant images, it might result in an inefficient process which is shown in Figure 1, where the distance increases fast but AP decreases slowly. For the aspect ii, as each dimension of $f(\tilde{\mathbf{q}})$ can at most contribute one unit to change the Hamming distance, once achieved, it is wasteful to continually modify this dimension. In addition, it assigns the same gradient value on each dimension. However, the embedding's dimensions with different values (e.g., $f(\tilde{\mathbf{q}})_i = 0.1$ and $f(\tilde{\mathbf{q}})_j = 0.9$) have different vulnerability because they have different distances to the boundary (e.g., 0) to change the $sgn()$.

As an attack algorithm wish to achieve the attack goal with less cost (e.g., less image perturbation), the algorithm should focus on the vulnerable dimensions instead of treating all dimensions equally. To address this problem, we firstly analysis the ideal solution and then proposed our approach to design the surrogate function.

Let $f(\tilde{\mathbf{q}})_j$ be the j-th dimension of $f(\tilde{\mathbf{q}})$ and $h(\mathbf{r}_i)_j$ be the j-th dimension of $h(\mathbf{r}_i)$ where $h(\mathbf{r}_i)$ is the binary hash code of $\mathbf{r}_i$, we consider the following three **cases**:

(1) $f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j$ is positive, meaning the signs of $f(\tilde{\mathbf{q}})_j$ and $h(\mathbf{r}_i)_j$ are the same. In this case, we want to enlarge the Hamming distance between $h(\tilde{\mathbf{q}})_j$ and $h(\mathbf{r}_i)_j$ such that the sign of $f(\tilde{\mathbf{q}})_j$ can be changed. Thus, the gradient should push $f(\tilde{\mathbf{q}})_j$ away from $h(\mathbf{r}_i)_j$ in order to change the sign.
(2) $f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j$ is negative, meaning the signs of $f(\tilde{\mathbf{q}})_j$ and $h(\mathbf{r}_i)_j$ are different. Thus, this dimension already contributes one unit to Hamming distance, and any change will not contribute more to our objective. Therefore, we should keep it in this way and assign a small or zero gradient.
(3) We also consider a special case when $f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j$ is positive and $f(\tilde{\mathbf{q}})_j$ is close to 0, meaning $f(\tilde{\mathbf{q}})_j$ is located at the boundary. In this case, we wish the gradient is relatively larger to make $f(\tilde{\mathbf{q}})_j$ across 0 fast and change its sign. It is efficient to focus on the dimensions in this case because the $f(\tilde{\mathbf{q}})_j$ only needs a small distance to change its sign and change the Hamming distance.

Therefore, to satisfied the first two cases, given $\tilde{\mathbf{q}}$ and $\mathbf{r}_i$, we formulate the following data-oriented Hamming distance surrogate objective function for the untargeted attack:

$$\begin{aligned} d_u(\tilde{\mathbf{q}}, \mathbf{r}_i) = & \sum_{j=1}^{k} \mathbb{1}(f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j > 0) \cdot \frac{\left| f(\tilde{\mathbf{q}})_j - h(\mathbf{r}_i)_j \right|}{2} \\ & + \sum_{j=1}^{k} \left(1 - \mathbb{1}(f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j > 0)\right) \cdot 1, \end{aligned} \tag{3}$$

where $\mathbb{1}(x > 0)$ is the indicator function of whether $x > 0$ or not. It defines whether the binary code of this dimension needs to be changed. The $\frac{\left| f(\tilde{\mathbf{q}})_j - h(\mathbf{r}_i)_j \right|}{2}$ is an approximation of the Hamming distance of the j-th dimension when $f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j > 0$, which aims to assign a gradient on $f(\tilde{\mathbf{q}})_j$. The second part indicates that

if $f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j \leq 0$, we give it a Hamming distance 1. Now, this function surrogates the Hamming distance and indicates whether the value on this dimension need to be change.

Then, we use the logistic function to replace the indicator function in Eq. (3), i.e., $\mathbb{1}(x > 0) \approx \frac{1}{1+e^{-\sigma \cdot x}}$ where σ is a hyper-parameter to control the steepness of the curve. It makes the function smoother and has a larger gradient value around 0, which satisfied the third case. Hence we have the Hamming distance surrogate function:

$$d_u(\tilde{\mathbf{q}}, \mathbf{r}_i) = \frac{1}{2} \sum_{j=1}^{k} \frac{\left| f(\tilde{\mathbf{q}})_j - h(\mathbf{r}_i)_j \right|}{1 + e^{-\sigma \cdot f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j}} + \sum_{j=1}^{k} \left(1 - \frac{1}{1 + e^{-\sigma \cdot f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j}} \right) \cdot 1. \tag{4}$$

We show the value and gradient of the proposed function and the previous inner product based function d_{inner} in Figure 3. In the right side figure, when $f(\tilde{\mathbf{q}})_j$ has the same $sgn()$ with $h(\mathbf{r}_i)_j$ (e.g., C_1 for Case 1), our function wishes to change the $sgn()$ of $f(\tilde{\mathbf{q}})_j$ and assign a gradient on it. When $f(\tilde{\mathbf{q}})_j$ is around 0 (e.g., C_3), its gradient is relatively larger than the others, and the gradient decreases to 0 fast when $f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{r}_i)_j$ have different $sgn()$ (e.g., C_2). However, the gradient of the inner product based function stay at the same value.

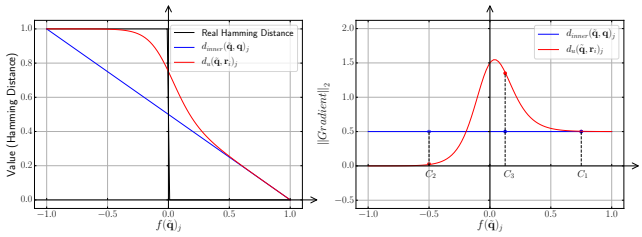

Figure 3: The value and gradient on *each* dimension (i.e., the j-th dimension) of $d_u(\tilde{\mathbf{q}}, \mathbf{r}_i)$ and $d_{inner}(\tilde{\mathbf{q}}, \mathbf{q})$. In this example, $h(\mathbf{r}_i)_j$ and $h(\mathbf{q})_j$ have a fixed value 1. σ is set at 10.

Design AP-oriented Weight Function. We also observed that the relevant images contribute differently to the average precision. As shown in Figure 2, if the hash vector of the relevant image $\mathbf{r}_i$ is closer to $\tilde{\mathbf{q}}$, then it contributes more to the AP of $\tilde{\mathbf{q}}$. Therefore, in the untargeted adversarial attack, we want the algorithm to focus more on those relevant images with a smaller Hamming distance to $\tilde{\mathbf{q}}$. Motivated by this, we propose the following weight function for each $\mathbf{r}_i$:

$$w(\tilde{\mathbf{q}}, \mathbf{r}_i) = \left(\frac{1}{d_H(h(\tilde{\mathbf{q}}), h(\mathbf{r}_i))} \right)^z, \tag{5}$$

where $d_H(h(\tilde{\mathbf{q}}), h(\mathbf{r}_i))$ is the Hamming distance between $h(\tilde{\mathbf{q}})$ and $h(\mathbf{r}_i)$, z is a non-negative hyper-parameter to control the steepness of the curve. For the case when $h(\tilde{\mathbf{q}}) = h(\mathbf{r}_i)$ which results in a Hamming distance 0 in the denominator, we let $d_H(h(\tilde{\mathbf{q}}), h(\mathbf{r}_i)) = 0.5$.

We then define $F_u(\tilde{\mathbf{q}})$ as a weighted average of the data-oriented function between $\tilde{\mathbf{q}}$ and $\mathbf{r}_i$:

$$F_u(\tilde{\mathbf{q}}) = -\frac{1}{n_u} \sum_{i=1}^{n_u} \left(d_u(\tilde{\mathbf{q}}, \mathbf{r}_i) \cdot w(\tilde{\mathbf{q}}, \mathbf{r}_i) \right). \tag{6}$$

4.3.2 Targeted Adversarial Attack. For targeted adversarial attack, let $\mathbf{t}_i$ be the i-th image with the target label $\mathbf{y}_t$, our goal is to improve the AP of $\tilde{\mathbf{q}}$ with respect to $\mathbf{t}_i$. Similar to $F_u(\tilde{\mathbf{q}})$, we define $F_t(\tilde{\mathbf{q}})$ as follows:

$$F_t(\tilde{\mathbf{q}}) = \frac{1}{n_t} \sum_{i=1}^{n_t} \left(d_t(\tilde{\mathbf{q}}, \mathbf{t}_i) \cdot w(\tilde{\mathbf{q}}, \mathbf{t}_i) \right), \tag{7}$$

where n_t is the total number of user specified relevant images. Different from Eq. (6), there is no " − " sign in Eq. (7). Because for targeted attack, we wish to minimal $F_t(\tilde{\mathbf{q}})$ to decrease the weighted surrogate Hamming distance between $\tilde{\mathbf{q}}$ and its relevant images.

For $d_t(\tilde{\mathbf{q}}, \mathbf{t}_i)$, the intuition is opposite to that of the untargeted adversarial attack. We want to make $sgn(f(\tilde{\mathbf{q}})_j) = h(\mathbf{t}_i)_j$ for each $j \in \{1, \ldots, k\}$. Therefore, we need to push $f(\tilde{\mathbf{q}})_j$ close to $h(\mathbf{t}_i)$, especially when the signs of $f(\tilde{\mathbf{q}})_j$ and $h(\mathbf{t}_i)$ are different. In order to do that, we change the indicator function in Eq. (3) into $\mathbb{1}(f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{t}_i)_j < 0)$, and change the $\cdot 1$ into $\cdot 0$ hence the following data-oriented objective function is formulated for the targeted adversarial attack:

$$\begin{aligned} d_t(\tilde{\mathbf{q}}, \mathbf{t}_i) &= \frac{1}{2} \sum_{j=1}^{k} \frac{\left| f(\tilde{\mathbf{q}})_j - h(\mathbf{t}_i)_j \right|}{1 + e^{\sigma \cdot f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{t}_i)_j}} + \sum_{j=1}^{k} \left(1 - \frac{1}{1 + e^{\sigma \cdot f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{t}_i)_j}} \right) \cdot 0 \\ &= \frac{1}{2} \sum_{j=1}^{k} \frac{\left| f(\tilde{\mathbf{q}})_j - h(\mathbf{t}_i)_j \right|}{1 + e^{\sigma \cdot f(\tilde{\mathbf{q}})_j \cdot h(\mathbf{t}_i)_j}}. \end{aligned} \tag{8}$$

On the other hand, the formulation of $w(\tilde{\mathbf{q}}, \mathbf{t}_i)$ remains the same as in Eq. (5) but changes z into $-z$ to get better performance. It is because if $\tilde{\mathbf{q}}$ is far away from $\mathbf{t}_i$, then reducing their distance contributes more to the increment of AP of $\tilde{\mathbf{q}}$, hence should have a larger weight.

4.4 Solving the Optimization Problem

We could use any widely used optimizers (e.g., Adam [15]) to solve the optimization problems. The algorithm will stop when the number of iterations exceeds the pre-defined threshold. In addition, for the untargeted attack, we use $AP = 0$ as the early terminate condition since the AP cannot be reduced anymore. Similarly, for the targeted attack, we use $AP = 1$ as the early termination condition since the AP cannot be improved anymore.

In addition, similar to [34], in order to deal with the vanishing gradient problem in deep hashing models, we multiple $g(\mathbf{q}')$ with a hyperparameter $\gamma \in [0, 1]$ to make the gradient stay at a relatively large value (i.e., $f(\tilde{\mathbf{q}}) = tanh(\gamma g(\tilde{\mathbf{q}}))$).

5 EXPERIMENTS

5.1 Datasets and Evaluation Protocol

Datasets and Model. We perform experiments on three standard image retrieval datasets, namely CIFAR-10 [16], MS-COCO [23], and NUS-WIDE [9]. We use DCH [5] as the retrieval model, as it is one of the state-of-the-art deep hashing based retrieval model and got promising retrieval performance. We follow DCH to train the models and generate image databases for retrieving.

- CIFAR-10 contains 60,000 images with 10 labels. We randomly select 5,000 images as the training set, 1,000 images as the testing set, and the rest images are used as the database.

Table 1: The original MAPs on three datasets of models with different k.

Datasets	CIFAR-10				MS-COCO				NUS-WIDE			
k	16	32	48	64	16	32	48	64	16	32	48	64
Original MAP	0.788	0.775	0.718	0.736	0.658	0.698	0.691	0.685	0.704	0.727	0.724	0.707

Table 2: The result of image perturbation (l_2) when algorithms achieve MAP=0.1 on three datasets of models with different k for untargeted attack. Smaller value means better performance. The underlined number indicates the algorithm can not achieve MAP=0.1.

Datasets	CIFAR-10				MS-COCO				NUS-WIDE			
k	16	32	48	64	16	32	48	64	16	32	48	64
SDHA	**0.707**	**1.096**	**1.074**	**0.942**	**2.408**	**2.081**	**2.014**	**1.950**	**1.860**	**1.705**	**2.063**	**2.486**
SDHA-UW	1.102	1.844	1.574	1.879	2.642	2.513	2.513	2.548	2.401	2.348	3.057	3.072
HAG	1.231	2.004	1.849	2.086	3.291	2.812	2.768	2.712	2.618	2.424	3.287	3.306

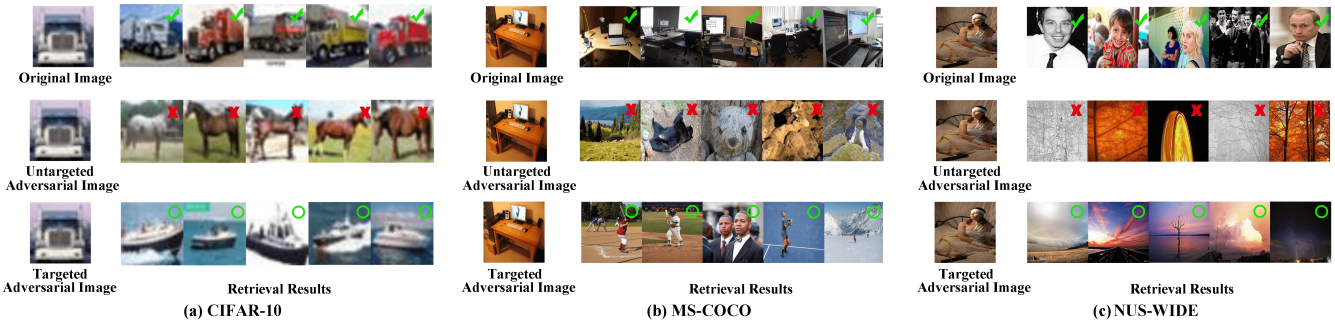

Figure 4: The result of both untargeted and targeted attacks of SDHA on three datasets for k=64. The first row represents the original images with their top-5 retrieved images. The second row and the third row are the untargeted images and targeted images with their retrieval results, respectively. The tick indicates the image shares the same label with the original image and the cross indicates the otherwise. The circle indicates that the label of image is same to the target label.

- MS-COCO contains 82,783 training images and 40,504 validation images. Each image is labeled by some of the 80 labels. We combine the training and validation images and randomly select 5,000 images from them as the testing set, and the rest of them are databases. 10,000 sampled images from the database are used to train the model.
- NUS-WIDE contains 269,648 images with multiple labels related to 81 different concepts. We select the subset of 10 the most popular concepts, which consists of 165,798 images. 10,000 images are randomly selected as the training set, 5,000 images are used as the testing set, and the rest images are used as the image database.

We randomly reserve 100 queries from each testing set as the query workload for both untargeted and targeted attacks.

Evaluated Methods. We evaluate the following four methods:

- **SDHA** is the proposed adversarial attack method.
- **SDHA-UW** is the proposed adversarial attack method which sets the weight function $w(\tilde{\mathbf{q}}, \mathbf{r}_i) = 1$, i.e., all the $\mathbf{r}_i$ contribute the same weight to the objective function.
- **HAG** [34] is the state-of-the-art *untargated* deep hashing retrieval system attack. The main idea of HAG is to push the adversarial query away from the original query in the embedding space. The objective function of HAG is $\left\| \frac{1}{k'} f'(\tilde{\mathbf{q}})^\top f'(\mathbf{q}) + 1 \right\|^2$, where $f'(\cdot)$ is same to $f(\cdot)$ with a mask [34] and k' is the number of nonzero elements of the mask. We apply the same early termination condition (i.e., AP = 0) described in Section 4.4 on it, such that the algorithm can stop earlier and avoid unnecessarily large perturbations.
- **DHTA** [3] is the state-of-the-art *targeted* deep hashing retrieval system attack. The core idea is to move the adversarial query close to a pre-computed anchor code $\mathbf{h}_a$ in the embedding space. The objective function of DHTA is $-\frac{1}{k} f(\tilde{\mathbf{q}})^\top \mathbf{h}_a$ where $\mathbf{h}_a$ is computed as having the minimum average distance to the hash codes of the images with target label $\mathbf{y}_t$. The early termination condition for targeted attack (i.e., AP = 1) in Section 4.4 is also applied.

Measurements. We consider two measurements in our experiments: MAP and image perturbation distance.

For untargeted attack, same to HAG, the MAP is calculated on top-1000 retrieved images. For targeted attack, the images shared at least one label with the user specified *targeted* labels are treat as relevant images. We followed DHTA to set the targeted labels. We do not explicitly report MAP, instead, we set a reasonable MAP as the goal (e.g., ≤ 0.1 for untargeted attack and ≥ 0.7 for targeted attack). We then fine tune the algorithms such that they can achieve the preset MAP and evaluate the image perturbation distance under this situation.

We use Euclidean distance to measure the image perturbation between $\tilde{\mathbf{q}}$ and $\mathbf{q}$. The smaller image perturbation means the better performance of the algorithm as it is harder to be noticed. The pixel value of images are all normalized to [0, 1] and the ℓ^∞ threshold ϵ is set to 0.039.

Implement detail of the retrieval models. For threat deep hashing retrieval models, we follow the DCH's setting guide to train the threat deep hashing retrieval models. The input images are resized to 256×256. The scale parameter of the Cauchy distribution is set as 30. Mini-batch Stochastic Gradient Descent (SGD) with 0.9 momentum is used and batch size is 128. The learning rates on CIFAR-10, MS-COCO, and NUS-WIDE are set to 0.005, 0.001, and 0.005, respectively. The training iteration number is set to 2000.

The MAPs of trained DCH models of three datasets with different output dimensionalities are shown in Table 1. Therefore in our targeted attack experiment, we set the objective MAPs of CIFAR-10, MS-COCO, and NUS-WIDE as 0.8, 0.7, and 0.75, respectively. The objective MAPs are slightly higher than the current MAPs.

Implement detail of four methods. For HAG, we follow its setting and use SGD with momentum 0.9 as the optimizer. The learning rate is 500 and the maximum iteration number is 2000. For the generation of each single query, the algorithm terminates when all bits of the query's embedding are flipped.

For DHTA, we follow its setting and use PGD as the optimizer with learning rate 1. The parameter n_t in DHTA to calculate the anchor point is set to 9 and the maximum iteration is set to 2000.

For SDHA and SDHA-UW, we use Adam [15] as the optimizer for both untargeted attack and targeted attack with the learning rate at 0.01. The momentum β_1 is set to 0.9 and the second-order momentum β_2 is set to 0.999. The maximum iteration number is set to 1500. The σ of surrogate function is set as 5 and 10 for untargeted and targeted attack, respectively. The z of weight function is set at 0.5 and -0.3 for untargeted and targeted attack, respectively. They control the steepness of their corresponding functions. The Lagrange multiplier α is influenced by the model dimension k and different datasets. Similar to [8], we tune and set it from $\frac{5}{k}$ to $\frac{100}{k}$ on CIFAR-10, and from $\frac{25}{k}$ to $\frac{100}{k}$ for MS-COCO and NUS-WIDE on both untargeted and targeted attacks.

For all methods, similar to the HAG's setting, the parameter γ is set as 0.1 for the first half iterations. For the last half iterations, the value changes to the values in the list [0.2, 0.3, 0.5, 0.7, 1.0] for every 10% increment iterations. We calculate the MAP every 10 iterations during the generation process of adversarial examples.

5.2 Results

5.2.1 Result of Untargeted Adversarial Attack. Table 2 shows the image perturbation for the three algorithms on different datasets with different output dimensionalities for untargeted attack. The preset threshold of the MAP is no more than 0.1.

It can be observed that the proposed method SDHA consistently costs much smaller perturbation distance than HAG among all settings. The image perturbation of SDHA decreases 34% of that of HAG on average of all the evaluated models. In addition, SDHA-UW also consistently performs better than HAG. This is because our methods consider the relevant images of the query in the objective functions, rather than simply pushing the adversarial query away from the original query as in HAG. Another reason is that, unlike SDHA and SDHA-UW, HAG does not explicitly consider the perturbation distance in its algorithm. In addition, HAG fails to achieve $MAP \leq 0.1$ on the harder NUS-WIDE dataset when $k = 64$ while our methods guarantee the successful attacks in all settings.

When comparing SDHA with SDHA-UW, we observe that the image perturbations of SDHA decrease 27% of that of the SDHA-UW on average. This implies the effectiveness of the AP-oriented weight function, as it uses the distance between $h(\tilde{\mathbf{q}})$ and $h(\mathbf{r}_i)$ to indicates the importance of different relevant images.

5.2.2 Result of Targeted Adversarial Attack. Table 3 summarizes the image perturbations of targeted attacks when they achieve the preset objective MAPs. The objective MAPs with respect to user specified label $\mathbf{y}_t$ are set to be higher than the original MAP with respect the label of the original query $\mathbf{q}$.

It can be observed that both SDHA and SDHA-UW manage to improve the MAPs with significantly less image perturbation than DHTA. For example, SDHA decreases more than 55% of image perturbations of DHTA on average of all datasets with all models. SDHA-UW performs slightly worse than SDHA, but still reduces the image perturbation of DHTA by 49% on average. This improvement is own by the design of our loss function and it proves that our framework is suitable for both untargeted and targeted attacks.

On the contrary, DHTA simply pushing $f(\tilde{\mathbf{q}})$ close to $\mathbf{h}_a$, not matter $sgn(f(\tilde{\mathbf{q}})_j)$ equals to $\mathbf{h}_{aj}$ or not. For example, when $f(\tilde{\mathbf{q}})_j = 0.5$ and $\mathbf{h}_{aj} = 1$, DHTA still tends to increase the value of $f(\tilde{\mathbf{q}})_j$ such that the margin can be reduced. But it will not change the distance between $h(\tilde{\mathbf{q}})$ and $\mathbf{h}_a$, hence will not increase the AP either. Therefore, it leads to a waste of the perturbation cost and results in a large perturbation.

We also notice that DHTA does not always achieve the preset MAPs objective (e.g., the underlined numbers in Table 3, even they have already perturbed much more than SDHA and SDHA-UW). After fine tuning of the hyper parameters (e.g., increasing the iteration number and the learning rate), it still not achieved. It happens when the dimension increased. One possible reason is that DHTA use the negative inner product between $f(\tilde{\mathbf{q}})$ and $\mathbf{h}_a$ as the objective function and do not distinguish the difference between dimensions. The gradients of different dimensions might influence each other.

5.2.3 Results with Preset l_2 Budgets and Final Performance. In Table 4, we present the MAPs on 64 bits models when different attacks achieve preset l_2 perturbation budgets and the performance at the maximum iterations. It can be observed that our proposed methods can achieve comparable or better (i.e., lower) MAPs for untargeted attack and achieve better (i.e., higher) MAPs for targeted attack with same image perturbation budget. After running maximum iterations, our proposed methods also achieve better performance on the final MAPs and image perturbations in most cases.

5.3 Visualization

We show examples of SDHA for both untargeted and targeted attacks in Figure 4 on three datasets. It can be seen that, for untargeted attack, none of the retrieved images has the same label with the

Table 3: The result of image perturbation (l_2) when algorithms achieve preset MAPs on three datasets of different k for targeted attack. Smaller value means better performance. Underlined numbers indicate the algorithm can not achieve the preset MAPs.

Datasets	CIFAR-10				MS-COCO				NUS-WIDE			
k	16	32	48	64	16	32	48	64	16	32	48	64
SDHA	**1.504**	**2.123**	**1.743**	**1.890**	**0.946**	**0.843**	**0.678**	**0.713**	**1.689**	**1.474**	**1.267**	**1.494**
SDHA-UW	1.707	2.549	2.125	2.504	1.203	1.078	0.924	0.986	2.298	2.118	1.542	1.610
DHTA	2.516	4.402	4.555	4.913	2.279	3.410	4.016	3.862	3.914	4.104	4.971	5.001

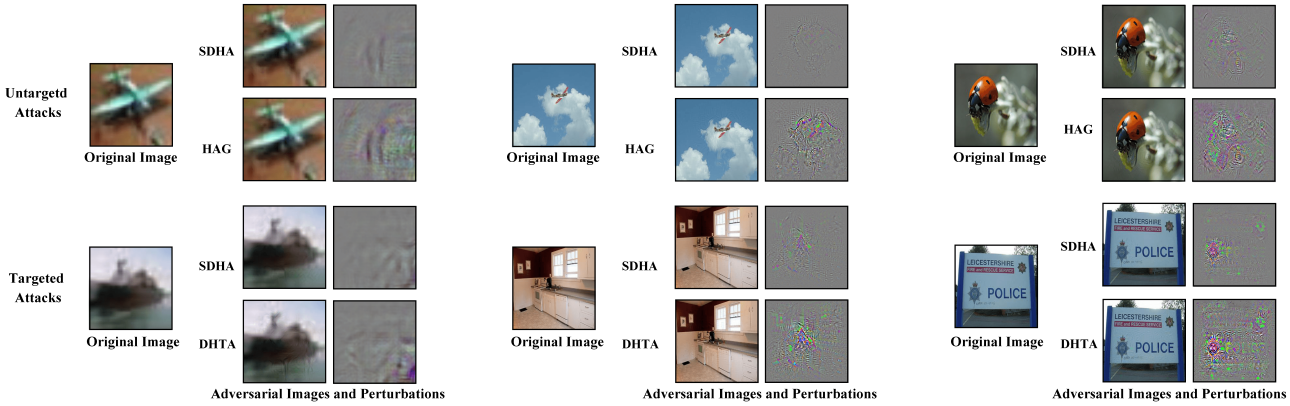

Figure 5: The adversarial images and perturbations of SDHA, HAG, and DHTA on three datasets for k=64. The left side of each set shows the original image. The first column of the right side shows the images and the second column shows the perturbations. The perturbation of SDHA is less noticeable than DHTA and DHTA. Best viewed in color and zoom.

Table 4: The result of MAPs of different algorithms on three datasets for $k = 64$ with different l_2 budgets, and the final performance at the maximum iteration. Smaller value (i.e., smaller MAP) is better for untargeted attack and larger value (i.e., larger MAP) is better for targeted attack. For algorithms that do not cost the preset l_2 budgets, we fill the form with its best MAPs.

		CIFAR-10				MS-COCO				NUS-WIDE			
		$l_2 = 0.5$	$l_2 = 1.0$	$l_2 = 1.5$	Final MAP/l_2	$l_2 = 1.5$	$l_2 = 2.0$	$l_2 = 2.5$	Final MAP/l_2	$l_2 = 2.0$	$l_2 = 2.5$	$l_2 = 3.0$	Final MAP/l_2
Untargeted	SDHA	**0.605**	**0.076**	**0.076**	0.076/**0.908**	0.316	**0.089**	**0.056**	0.056/**2.063**	**0.228**	**0.087**	**0.056**	0.056/**2.168**
	SDHA-UW	0.639	0.454	0.235	0.040/2.252	0.403	0.235	0.105	**0.038**/2.980	0.338	0.224	0.127	**0.022**/3.208
	HAG	0.620	0.496	0.282	**0.002**/2.992	**0.313**	0.201	0.130	0.066/3.482	0.287	0.217	0.169	0.150/3.304
		$l_2 = 1.0$	$l_2 = 1.5$	$l_2 = 2.0$	Final MAP/l_2	$l_2 = 0.5$	$l_2 = 1.0$	$l_2 = 1.5$	Final MAP/l_2	$l_2 = 1.0$	$l_2 = 1.5$	$l_2 = 2.0$	Final MAP/l_2
Targeted	SDHA	0.109	**0.197**	0.823	0.823/**1.745**	**0.599**	**0.732**	**0.732**	**0.732**/**0.780**	**0.462**	**0.751**	0.751	0.745/**1.499**
	SDHA-UW	**0.119**	0.182	**0.919**	**0.919**/1.971	0.547	0.725	0.725	0.725/0.954	0.450	0.524	**0.754**	**0.751**/1.623
	DHTA	0.112	0.191	0.214	0.477/4.910	0.542	0.624	0.683	0.728/4.751	0.427	0.448	0.465	0.692/4.999

original query. For targeted attack, all the returned images have the same user specified target label.

We also show the image perturbations of SDHA, HAG and DHTA in Figure 5 of three datasets. It can be seen that the perturbations of SDHA are less noticeable than that of both HAG and DHTA. Smaller image perturbation makes the adversarial images harder to be noticed. For example, on MS-COCO, the image perturbation (in Euclidean distance) of SDHA is 0.805 while it is 2.743 of HAG for untargeted attack. The image perturbations for SDHA and DHTA are 1.503 and 3.430 for targeted attack, respectively. Thus the adversarial images generated by SDHA have better quality than HAG and DHTA.

6 CONCLUSION

In this paper, we studied the attack problems on deep hashing based image retrieval models. To attack retrieval models with less image perturbation, we designed an attack framework which used a dimension-wise surrogate Hamming distance function. It also considered the influence of relevant images and it combined with an AP-oriented weight function. In addition, our attack can deal with both untargeted and targeted attacks. The experimental results shown that our attack generates better adversarial images with smaller image perturbation than the state-of-the-art methods for both untargeted and targeted attacks.

ACKNOWLEDGMENTS

The work was supported in part by the National Key Research and Development Program of China (2020YFB1707900), Natural Science Foundation of China (61876038, 62072088), Ten Thousand Talent Program (ZX20200035), Liaoning Distinguished Professor (XLYC1902057), Dongguan Social Science and Technology Development Key Project (2020507140146), Dongguan University of Technology (KCYKYQD2017003), ARC DPs (180103411, 220101762) and HKUST Red Bird Visiting Scholar Program.

REFERENCES

[1] Laurent Amsaleg, James Bailey, Amélie Barbe, Sarah M. Erfani, Teddy Furon, Michael E. Houle, Miloš Radovanović, and Xuan Vinh Nguyen. 2021. High Intrinsic Dimensionality Facilitates Adversarial Attack: Theoretical Evidence. *IEEE Transactions on Information Forensics and Security* 16 (2021), 854–865. https://doi.org/10.1109/TIFS.2020.3023274

[2] L. Amsaleg, J. Bailey, Dominique Barbe, S. Erfani, M. Houle, Vinh Nguyen, and M. Radovanović. 2017. The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality. *2017 IEEE Workshop on Information Forensics and Security (WIFS)* (2017), 1–6.

[3] Jiawang Bai, Bin Chen, Yiming Li, Dongxian Wu, Weiwei Guo, Shu-Tao Xia, and En-Hui Yang. 2020. Targeted Attack for Deep Hashing based Retrieval. *ArXiv* abs/2004.07955 (2020).

[4] A. Brown, Weidi Xie, Vicky S. Kalogeiton, and Andrew Zisserman. 2020. Smooth-AP: Smoothing the Path Towards Large-Scale Image Retrieval. *ArXiv* abs/2007.12163 (2020).

[5] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. 2018. Deep Cauchy Hashing for Hamming Space Retrieval. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 1229–1237.

[6] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. 2017. HashNet: Deep Learning to Hash by Continuation. *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 5609–5618.

[7] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. 2017. HashNet: Deep Learning to Hash by Continuation. *arXiv preprint arXiv:1702.00758* (2017).

[8] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symp. Secur. Priv.* IEEE, San Jose, CA, USA, 39–57. https://doi.org/10.1109/SP.2017.49

[9] Tat-Seng Chua, J. Tang, R. Hong, Haojie Li, Zhiping Luo, and Y. Zheng. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *CIVR '09*.

[10] B. Dai, R. Guo, S. Kumar, Niao He, and L. Song. 2017. Stochastic Generative Hashing. In *ICML*.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*.

[12] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6572

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 770–778. https://doi.org/10.1109/CVPR.2016.90

[14] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778.

[15] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[16] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images.

[17] Alex Krizhevsky and Geoffrey E. Hinton. 2011. Using very deep autoencoders for content-based image retrieval. In *ESANN*.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (Eds.). 1106–1114. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

[19] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=HJGU3Rodl

[20] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. 2015. Simultaneous feature learning and hash coding with deep neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 3270–3278.

[21] Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. 2019. Universal Perturbation Attack Against Image Retrieval. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 4898–4907. https://doi.org/10.1109/ICCV.2019.00500

[22] Kevin Lin, Jiwen Lu, Chu-Song Chen, and J. Zhou. 2016. Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 1183–1192.

[23] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *ArXiv* abs/1405.0312 (2014).

[24] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. 2015. Deep hashing for compact binary codes learning. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 2475–2483.

[25] Haomiao Liu, R. Wang, S. Shan, and X. Chen. 2019. Deep Supervised Hashing for Fast Image Retrieval. *International Journal of Computer Vision* (2019), 1–18.

[26] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. 2019. Who's Afraid of Adversarial Queries?: The Impact of Image Modifications on Content-based Image Retrieval. *Proceedings of the 2019 on International Conference on Multimedia Retrieval* (2019).

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2574–2582. https://doi.org/10.1109/CVPR.2016.282

[28] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.1556

[29] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[30] Giorgos Tolias, Filip Radenovic, and Ondrej Chum. 2019. Targeted Mismatch Adversarial Attack: Query With a Flower to Retrieve the Tower. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 5036–5045. https://doi.org/10.1109/ICCV.2019.00514

[31] Dayan Wu, Zheng Lin, Bo Li, Mingzhen Ye, and Weiping Wang. 2017. Deep Supervised Hashing for Multi-Label and Large-Scale Image Retrieval. *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval* (2017).

[32] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. 2014. Supervised Hashing for Image Retrieval via Image Representation Learning. In *AAAI*.

[33] Erkun Yang, T. Liu, Cheng Deng, W. Liu, and Dacheng Tao. 2019. DistillHash: Unsupervised Deep Hashing by Distilling Data Pairs. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 2941–2950.

[34] Erkun Yang, Tongliang Liu, Cheng Deng, and Dacheng Tao. 2020. Adversarial Examples for Hamming Space Search. *IEEE Transactions on Cybernetics* 50 (2020), 1473–1484.

[35] Andrew Zhai, Dmitry Kislyuk, Yushi Jing, Michael Feng, Eric Tzeng, Jeff Donahue, Yue Li Du, and Trevor Darrell. 2017. Visual Discovery at Pinterest. In *WWW (Companion Volume)*. ACM, 515–524.

[36] Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, and Rong Jin. 2018. Visual Search at Alibaba. In *KDD*. ACM, 993–1001.

[37] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. 2015. Deep semantic ranking based hashing for multi-label image retrieval. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 1556–1564.

[38] Guoping Zhao, Mingyu Zhang, Jiajun Liu, and Ji-Rong Wen. 2019. Unsupervised Adversarial Attacks on Deep Feature-based Retrieval with GAN. *ArXiv* abs/1907.05793 (2019).

[39] Mo Zhou, Zhenxing Niu, Le Wang, Qilin Zhang, and Gang Hua. 2020. Adversarial Ranking Attack and Defense. *CoRR* abs/2002.11293 (2020). arXiv:2002.11293 https://arxiv.org/abs/2002.11293

[40] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. 2016. Deep Hashing Network for Efficient Similarity Retrieval. In *AAAI*.