You See What I Want You to See: Exploring Targeted Black-Box Transferability Attack for Hash-based Image Retrieval Systems

Yanru Xiao and Cong Wang
Old Dominion University, Norfolk, VA
{yxiao002,c1wang}@odu.edu

Abstract

With the large multimedia content online, deep hashing has become a popular method for efficient image retrieval and storage. However, by inheriting the algorithmic backend from softmax classification, these techniques are vulnerable to the well-known adversarial examples as well. The massive collection of online images into the database also opens up new attack vectors. Attackers can embed adversarial images into the database and target specific categories to be retrieved by user queries. In this paper, we start from an adversarial standpoint to explore and enhance the capacity of targeted black-box transferability attack for deep hashing. We motivate this work by a series of empirical studies to see the unique challenges in image retrieval. We study the relations between adversarial subspace and black-box transferability via utilizing random noise as a proxy. Then we develop a new attack that is simultaneously adversarial and robust to noise to enhance transferability. Our experimental results demonstrate about 1.2-3× *improvements of black-box transferability compared with the state-of-the-art mechanisms. The code is available at: https://github.com/SugarRuy/CVPR21_Transferred_Hash.*

1. Introduction

With the exponential growth of visual content on the Internet, deep learning to hash (*deep hashing*) [46, 9, 25] has emerged as a leading technique in content-based image retrieval. By mapping semantically similar images into close proximity in the Hamming space, it enables efficient nearest neighbor search and storage of large-scale multimedia data. Powered by deep hashing, from a photo of a product taken in the real world, without knowing its name, customers could extract similar products online. Service providers, such as search engines (Google [2], Bing [1]), social networks (Pinterest [6], e-commerce (Taobao [5]) and fashion designers([16]), are investing largely into this technology to complement the traditional text query.

Unfortunately, by inheriting the backend from classification networks, deep hashing is also vulnerable to the well-known adversarial examples [27, 44, 38, 42], that purposely crafted perturbations with minimal perceptual difference can cause misclassification into any other label (*untargeted attack*) or a specific label (*targeted attack*). Targeted attacks are strictly more difficult given the complex inter-class semantics [8, 26]. While white-box attacks almost guarantee success, service providers do not reveal their models publicly, which remain a black box to the attacker. Because of the resemblance of decision boundaries, adversarial examples can still transfer to the black-box models, but at a much less chance to accomplish targeted attacks [26].

Rather than causing a wrong decision, system designers face a slightly different attack surface in image retrieval systems, in which images from the database are returned to match user's query. For better results, a growing database is typically maintained via automated crawling, indexing of online images [4] and caching user queries [3]. However, this may also inadvertently include private/inappropriate/upsetting content such as protected copyright, violence, pornography, racism or advertising spam into the database. By designing adversarial perturbations into the inappropriate images, attackers can launch targeted attacks against benign search queries, and visually display those images to the victims. To exploit this vulnerability, competitors can override the product search results in online shopping; advertisers can make customers view their advertisements for free; conspirators can divert images of political banners into racism or violence. Attackers can further target the content in the top searching list to reap high visibility.

The previous works have shown high success rate of untargeted white-box attacks for image retrieval [27, 44, 38]. E.g., [44] shows that by maximizing the hamming distance of a perturbed image to its original category in the hash space, the network retrieves an irrelevant image. Nevertheless, the most challenging targeted attacks are yet to be fully explored in the black-box setting and they also carry higher practical value as attackers can mislead the results into specific categories. A trivial way to accomplish black-box transferability is to increase the level of perturbation [26], at the cost of degrading visual quality and being detected. In fact, our preliminary experiment indicates drastically small transferability under 1%, even the state-of-the-

art mechanism [42] is implemented for deep hashing. However, such low transferability does not necessarily translate into a blessing in security before we fully understand the attacker's capacity.

In this paper, we explore and improve targeted transferable attack in deep hashing. Similar to susceptible classes in classification [32], our first discovery is the existence of vulnerable pairs that transfer more easily than the rest. They could be explicitly mined based on the hamming distance from the white-box model, where attackers can utilize these pairs to enhance the success rate. Then we look into different attacks to find implications of their transferrable capacity. We design an algorithm to utilize additive Gaussian random noise as a proxy to estimate the generated adversarial region, and show that it is indeed related to black-box transferability, i.e., an adversarial example with higher tolerance to random noise is more prone to transfer to black-box models. Based on this finding, we further devise a new attack to look for perturbations that are simultaneously adversarial and robust to random noise, i.e., both adversarial and noise-corrupted adversarial images are retrievable by querying the target images.

The main contributions are summarized below. First, this work aims to bridge the two areas of adversarial attacks and image retrieval. By studying the most challenging targeted black-box transferability attack, it opens up a new dimension to realize an array of realistic attacks in image retrieval systems. Second, we point out useful information from the white-box model that implies black-box transferability: a) the existence of vulnerable pairs; b) the relation between transferability and white-box adversarial region. We propose an algorithm to estimate the adversarial region by introducing random noise, which is used to assess the capacity of different attacks. Then we design a new attack to search for a perturbation for potentially higher transferability. Finally, we conduct extensive experiments and demonstrate that the proposed attack can boost the black-box transferability by $1.2-3\times$, compared to PGD [29], and $1.5\times$ compared to the diversity techniques [42]. We also demonstrate case studies of crafting out-of-distribution images to target normal queries with high successful rates.

2. Background and Related Work

2.1. Black-Box Adversarial Attacks

Fast Gradient Sign Method (FGSM) [15] and Projected Gradient Descent (PGD) [29] are the two baseline methods. FGSM takes a large step in the gradient directions to maximize the probability of the target class, by finding a perturbed image within the η-norm ball. The PGD attack initializes the adversarial search from a random point within the norm ball, and conducts several iterations towards the target class. The existing works take two directions in a black-box setting.

Transferability Attack exploits the similarity of decision boundaries between different models on the same data, and utilizes the gradients from the source model to generate adversarial examples, in the hope that they transfer to the unknown target model. In the worst case, gradient directions from the source and target models could be orthogonal to each other [26], which makes the source model less effective. A handful of studies ascribe the difficulty of black-box transferability to the overfitting on the source model and misalignment of decision boundaries [42, 12, 39, 35]. Therefore, enhancing diversity has been taken at different levels of input image [42, 12], model ensemble [39] and gradient trajectory [35]. Rather than using a single image, in [42], random affine transformation of the input image is adopted in each iteration to enhance input diversity. Similarly, an ensemble of shifted images are used to maximize the loss objectives for better transferability [12]. Both gradient ascent and descent are combined for more diversity [35]. Another thread of works focus on the feature level to improve transferability [45, 23]. The intuition is to induce a similar intermediate feature via perturbing image pixels, by assuming that different models generate identical feature-level representations. Intermediate loss is introduced to optimize l_2 norm between feature maps from all layers in [45, 23]. Our work taps into this line to enhance black-box transferability for image retrieval systems and will compare with these techniques in Sec. 6.

Query-based Attack. These techniques treat the targeted model as an oracle and adjust the perturbation in iterative steps based on the system output of probability [22, 10] or decision (label-only) [13, 7, 8]. E.g., [10] utilizes the changes from the softmax output to estimate the gradients. [22] adopts the natural evolutionary strategy to estimate the gradient under the search distribution. [7] only relies on the final decision of the model, which iteratively draws random distribution from a proposed distribution while staying adversarial and [8] further optimizes such distribution. Though considerable effort is devoted to enhance query efficiency, it is still very difficult to estimate the gradient of high dimensions with limited information: several thousands of queries are typically required to craft an adversarial example. Since the image retrieval system could be metered by the number of queries, these strategies are less cost-effective for budget-limited attackers. To this end, we focus on transferability attacks that the attackers can economically generate a large number of adversarial examples and wait for them to be matched and retrieved by the users.

2.2. Deep Learning to Hash

Similar to metric learning, deep hashing also learns pairwise similarity from end-to-end through the maximum likelihood estimation, and transforms real-valued inputs into binary hash codes [46, 9, 25]. Hence, similarity search can be performed efficiently by calculating the *hamming distance*. In addition to the feature extraction layers, a hash layer is introduced to map input $x \rightarrow h(x) \in \{-1, +1\}^K$ into a K-bit binary code (the sign function $sgn(\cdot)$). To remain differentiation with backpropagation, continuous ap-

proximation for the non-smooth sign function is performed, e.g., HashNet [9] adopts the hyperbolic tangent function, $sgn(z) = \lim_{\beta\to\infty} \tanh(\beta z)$, by tuning β; the function converges to the sign function when $\beta \to \infty$. As a result, hashing aggregates similar images into a Hamming ball. The system typically relies on a *retrieval threshold* and any image with smaller hamming distance is returned as matched results.

Deep hashing inherits the vulnerability to adversarial examples from the classification model [44, 27, 41], but triggers in a slightly different way. Targeted attacks in classification redirect the original label to a target label in a closed set of discrete classes; targeted attacks in deep hashing push the adversarial image into the retrieval threshold of the target class (image), so that whenever an image in the target class is queried, the adversarial image is matched and returned. [44] fools deep hashing to maximize the distance between a perturbed image and the original one, such that the hamming distance exceeds the retrieval threshold for that category. [27] follows with a similar optimization objective to design adversarial queries. [41] designs a new optimization problem to prevent private images in the database from queried by curious third parties. [38] crafts adversarial images to conceal sensitive queries while still retrieving the targeted images. Most of these works focus on re-designing the adversarial objectives in a white-box setting, but have yet to explore the design space of the more challenging targeted black-box attacks.

3. Motivation

In this section, we introduce basic definitions and motivate this work by important observations.

Definition 1. (Hamming Distance) Deep hashing transforms inputs x_i and x_j into hash codes $h(x_i), h(x_j) \in \{-1, +1\}^{1\times K}$. The hamming distance between them, $D_h(x_i, x_j)$ can be computed from the inner product, $\frac{1}{2}(K - h(x_i)h(x_j)^\top)$.

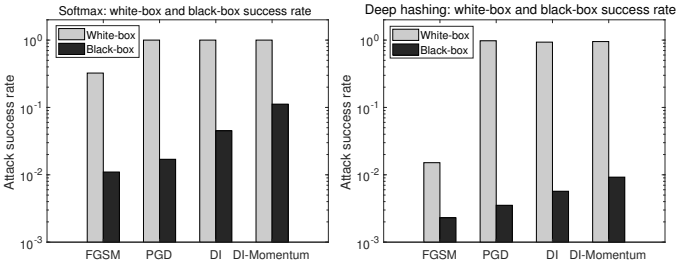
Definition 2. (Retrieval) For a queried image x_i, all x_j satisfying $D_h(x_i, x_j) \leq T_h$ (T_h is the retrieval threshold) are returned as the results.

Definition 3. (Class) Though deep hashing characterizes a weak notion of class, samples from the same class often result high similarity. For targeted attacks, we retain the concept of class here and define that if an input retrieves more than N_r samples from a class, the input belongs to that class. An input with various contents could be mapped to different classes, resulting the multi-label situation [25, 9].

Definition 4. (Targeted Attack) For an input x and the images in the targeted class $x_t \in \mathcal{C}_t$, the attacker's goal is to minimize the hamming distance via adjusting $x + \epsilon = x'$ under the η-norm bound[1],

$$\min_{x', x_t \in \mathcal{C}_t} D_h(x', x_t), \tag{1}$$

[1]Since the sign function is non-differentiable, we take the penultimate output from HashNet instead of directly optimizing on the hashcodes.



(a) (b)

Figure 1: Targeted white-box and black-box attack success rate (a) Softmax classification; (b) Deep hashing. See summary in *Observation 1*.

$$s.t. \, ||x - x'||_\infty < \eta. \tag{2}$$

Definition 5. (Query Symmetry) Hamming distance is symmetric: if an image x_i can be queried via the adversarial input x', then querying x_i also returns x'. The attacker can take advantage of this property to embed x' in the database. Once $x_t \in \mathcal{C}_t$ is queried by a user, x' will be returned and visualized by the user.

Definition 6. (Black-Box Transferability) Without prior knowledge and access to the black-box model M_b, the attacker crafts adversarial examples x' based on a white-box source model M_w.

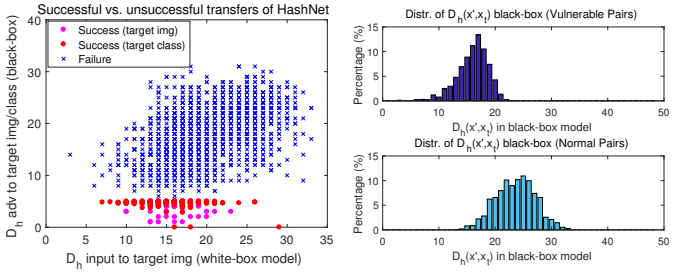
Definition 7. (Criteria of Successful Attack) Attack success can be measured by the number of images returned in the target class $\mathcal{C}_t$ from model M_b, when the adversarial image x' is queried. We further define that an attack is successful if it is larger than a certain number N_t, e.g., retrieving 10 images from the target class.

3.1. Targeted Black-box Attacks to Image Retrieval

To see the success rate of targeted black-box attacks, we conduct some preliminary experiments to transfer adversarial examples generated from ResNet152 to ResNet50 on the ImageNet (other model combinations also indicate similar numerical gaps). We set the retrieval threshold $T_h = 5$ and iterate four state-of-the-art attacking methods: FGSM [15], PGD [29], Iterative FGSM with Diversity (DI) [42] and its momentum integration (DI-Momentum), originally designed for the softmax classification models. The key observations are summarized below.

Observation 1. There exists a large gap between the targeted white-box and black-box attack success rates (Fig. 1). Compared with softmax, which delivers around 10% black-box success, adversarial images rarely transfer with deep hashing: the overall success rate is below 1%.

Such low transferability is expected: rather than selecting $\arg\max$ from the softmax probabilities, successful retrieval requires the hashcode to be mapped into the vicinity of T_h in the vast open hash space. This leads to a large fraction of the adversarial hash codes lying in the non-retrievable region (away from all the classes) in the black-box model. The training paradigm with randomized pairing also induces more uncertainty. Different from a closed



(a) (b)

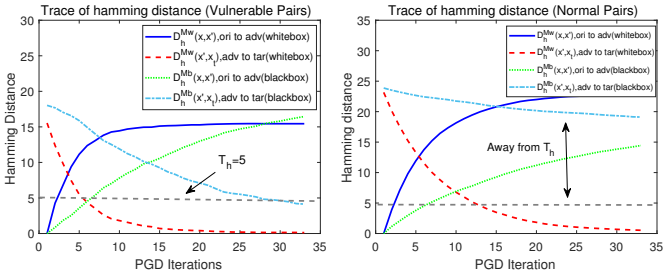
Figure 2: Illustration of vulnerable pairs. (a) Relations of hamming distance between input and target images in the white-box source model, and adversarial input to targeted images (class) in the black-box model; (b) Distribution of hamming distance from adversarial to target image in black-box model of vulnerable and normal pairs.

set of categories in one-hot encoding, deep hashing are more fluid to map pairwise similarity relations into binary hash codes. However, the low transferability should not be treated as a security benefit. We discover an intriguing persistence of *vulnerable pairs* as illustrated below.

Observation 2. (Vulnerable Pairs $\mathcal{V}$) There exists a large number of heterogenous input pairs $(x, x_t) \in \mathcal{V}$, such that the hamming distance between input x to target x_t, $T_h < D_h^{M_w}(x, x_t) \leq T_d$ in the white-box model M_w (T_d is a threshold larger than T_h). Then the probability of success on the black-box model M_b, $P\{D_h^{M_b}(x+\epsilon, x_t) < T_h | (x, x_t) \in \mathcal{V}\}$, is much higher than the rest of the normal pairs $(x, x_t) \notin \mathcal{V}$.

To see this, we pretend as if we could access the black-box model and demonstrate the relations between hamming distance $D_h^{M_w}(x, x_t)$ and $D_h^{M_b}(x+\epsilon, x_t)$ for successful and unsuccessful transfers in Fig.2(a). There are two ways of successful transfers:1) the adversarial image can directly retrieve the target image; 2) it retrieves similar images from the targeted class (other than the targeted image itself), where these images may come from a different intra-class cluster. It is observed that most of the successful transfers concentrate in a narrow distance range between 10-20 (Fig.2(a), x-axis), though a large number of unsuccessful transfers are also found for the same range. Fig.2(b) further compares the distribution between vulnerable and normal pairs on black-box model. It confirms that the vulnerable pairs are much closer to the target images with the mean around 16 vs.25 of the normal pairs.

We also trace the hamming distance vs. PGD iterations for the vulnerable and normal pairs in Fig.3. For the white-box setting, there is no doubt that PGD can push the adversarial inputs close to the target under the L_∞ bound, which corresponds to the adversarial image being driven away from the original input in hash space. However, the reflection on the black-box is divergent - PGD just succeeded at the end of 30 iterations for vulnerable pairs, whereas the normal pairs are far from success. The trend of the slope suggests more iterations for better transferability [26], which also brings higher perturbation and risks of violating the η-bound. Recall from Fig.2(a), even for the vulnerable pairs, only a minority can succeed, so is there a way to craft more transferable adversarial examples? We answer this question by exploring the adversarial subspace that enables transfer between different models.



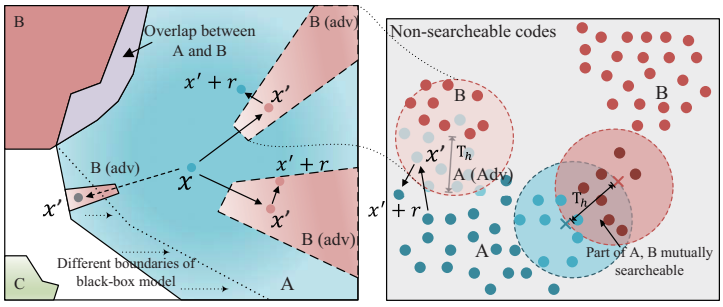
(a) (b)

Figure 3: Trace of hamming vs. PGD iterations. (a) vulerable pairs; (b) normal pairs.

4. Explore Adversarial Subspace

In this section, we propose a mechanism to efficiently estimate the transferable adversarial subspace given the white-box model. The adversarial subspace is typically described as a contiguous multi-dimensional subspace close to the data manifold [36, 15, 37] and notably difficult when it comes to quantitative analysis, e.g., some literatures employ intrinsic dimensionality [28] and orthogonal adversarial directions [40]. Only a few connects adversarial subspace with transferability: [40] finds the maximal number of orthogonal adversarial directions that induce a significant increase in loss, and demonstrates that transferability is proportional to this number on small-scale datasets. Nevertheless, the curse of high dimensionality quickly dampens such effort for large networks.

We propose an efficient method to utilize random noise as a proxy, and feedbacks from the white-box model to predict transferability. Random noise injection finds deep roots in the defense literatures to certify classifier robustness, e.g., learning a smoothed classifier that returns the most probable class under Gaussian noise [11, 34]. We draw a close connection to adversarial examples, which are found to form a cone-shape structure surrounded by natural classes [33, 19]. We conjecture their presence in deep hashing has a similar geometry sketched in Fig.4(a), but with a slight variation: classes may have minor overlaps due to multi-labeling (A and B have some overlaps), which are mapped to the vicinity of similar hash codes in the hash space (Fig. 4(b)). For adversarial image x' in class A, it is pushed into the retrieval threshold of class B in hash space. Most of the unsuccessful transfers to the black-box model are due to samples being mapped to different sets of hash codes. Out of the retrieval range, x' crafted from the source model often results a hash code with no retrieval results at all from the black-box model. We formally define the adversarial sphere below.



(a) Adv in the pixel space (b) Adv in the hash space

Figure 4: Illustration of adversarial examples (a) image pixel space (b) hash space.

Algorithm 1 Estimation of Adversarial Sphere

Input: An attack strategies, $x' = x + \epsilon$ for targeted class $\mathcal{C}_t$, $\mathcal{A} \leftarrow \{x'\}$, $\min \leftarrow 0$, $\max \leftarrow R$, $R = l_\infty$ bound, $0 < \beta \leq 1$
while $\min < \max$ **do**
  $m \leftarrow \frac{\min+\max}{2}$, $r \sim \mathcal{N}(0, \sigma^2 I)$, $\sigma = 1/3$.
  $r \leftarrow r - (r \cdot \frac{\epsilon}{|\epsilon|})\frac{\epsilon}{|\epsilon|}$, $r \leftarrow \frac{r}{||r||_\infty} \cdot m$.
  $x'' \leftarrow x' + r$, query x'' to model M_s.
  $y_i = \mathbb{1}(D_h(x_i', x_t \in \mathcal{C}_t) \leq T_h), \forall i \in \mathcal{A}$.
  if $\sum_{i \in \mathcal{A}} y_i < \beta|\mathcal{A}|$ **then**
    $\max \leftarrow m$.
  else
    $\min \leftarrow m$.
  end if
end while
Output: m (corresponds to the volume of adversarial space).

Definition 7. (Adversarial Sphere) For each data point $x \in \mathbb{R}^d$, $\mathbb{S}$ is the adversarial sphere embedded in $\mathbb{R}^d$ with dimension n and radius r_n. The radius is defined as the shortest distance from the centroid to the boundaries such that x' remains adversarial.

The volume of the adversarial sphere grows exponentially to the dimension n and radius r_n. Samples successfully transfer when the source and target models share part of the adversarial sphere for the same class [40]. It is not difficult to conjecture the following property.

Property 1. Transferability is proportional to the volume of adversarial sphere generated by an attack strategy.

Finding the closed-form representation of adversarial sphere seems difficult. Thus, we pursue an implicit measure by extending the defense method from [19]. Neural nets are known to be robust to random noises, but surprisingly sensitive to small, purposely crafted perturbations. According to [14], the magnitude of random noise required for misclassification is $\Theta(\sqrt{d/n}||\epsilon||_2)$, where $||\epsilon||_2$ is the amount of perturbation. Considering the adversarial image x', it indicates that by adding random noise on x', if the noise level is well beyond the order of $\sqrt{d/n}$, x' could be driven out of the adversarial sphere [19] (see Fig.4(a)). In other words, the adversarial sphere has to be large enough to keep x' adversarial when random noise are injected. To estimate the adversarial sphere, we propose an algorithm via reject sampling as described below.

First, we form a sample set $\mathcal{A}$ by paring (x, x_t) and candidate noise levels $[0, \cdots, R]$ (in the sense of l_∞). Then we adopt an attack strategy to generate adversarial samples $x' = x + \epsilon$, where ϵ is within the η-norm bound. Second, we sample i.i.d. random noise from the isotropic Gaussian distribution $r \sim \mathcal{N}(0, \sigma^2 I)$ that is orthogonal to the perturbation ϵ, i.e., projecting the Gaussian noise to the perturbation, $r \leftarrow r - (r \cdot \frac{\epsilon}{|\epsilon|})\frac{\epsilon}{|\epsilon|}$ and rescale by $\frac{1}{2}R$. We query the output $x'' = x' + r$ to the (white-box) source model M_w. The queries sum up the number of successful attacks from $\mathcal{A}$. If it is more than $\beta|\mathcal{A}|$ (more than β fraction of x'' succeed, $0 < \beta \leq 1$), we increase the noise level to $\frac{3}{4}R$ following the binary search rule; otherwise, we reduce it to $\frac{1}{4}R$. The iteration continues until an appropriate R is found such that β-ratio x'' remains in the adversarial sphere. The procedures are summarized in Algorithm 1 and it takes $\mathcal{O}(\log R|\mathcal{A}|)$ queries to the source model.

	Noise R	0	4	8	16	32	64
$\eta = 16$	PGD	96.7	96.1	93.2	69.5	24.0	4.5
	DI	95.7	95.2	93.3	76.6	29.0	6.0
	DI-Mom	99.4	99.3	99.3	98.9	96.5	38.6
$\eta = 32$	PGD	100.0	100.0	100.0	99.8	79.4	12.4
	DI	100.0	100.0	100.0	99.3	82.3	15.7
	DI-Mom	100.0	100.0	100.0	100.0	99.6	70.2

Table 1: White-box attack success rate when $\eta_\infty = 16, 32$ and $R \in [0, 64]$.

We validate *Property 1* by adopting the algorithm to estimate the adversarial subspaces and assess whether the results align with Fig. 1. Table 1 shows the white-box attack success rate when $R \in [0, 64]$. Higher success rate indicates larger adversarial sphere. Horizontally, when R is increased, we see that the success rate declines monotonically, which validates that large random noise tends to push x' out of the adversarial sphere. Vertically, the white-box success rate is generally consistent with the ranking order of black-box transferability in Fig.1. Thus, random noise can be used as an effective measure to estimate adversarial sphere and predict black-box transferability.

5. Exploit Transferable Subspace

In this section, we answer the next fundamental question: Can we craft perturbation in a way to land x' in a robust adversarial region, so as to potentially improve the black-box transferability? We develop a new mechanism to make both x' and $x' + r$ adversarially retrievable. Denote $f_{M_b}(x)$ as an oracle that returns the number of samples retrieved from a black-box model M_b, when x is queried. The ultimate goal is to maximize the black-box transferability by crafting $\epsilon^*(\sigma)$ on the source model, as well as selecting an appropriate input noise level σ from a candidate set $\mathcal{R}$),

$$\max \mathbb{E}_{\sigma \in \mathcal{R}} \left[f_{M_b}\left(x + \epsilon^*(\sigma)\right) \right], \tag{3}$$

Algorithm 2 Noise-induced Adversarial Generation (NAG)

Input: Target pairs $(x, x_t \in \mathcal{C}_t)$, candidate set of noise levels $\mathcal{R}$, initialize λ_0, $x_0 = x$, learning rate α.
for each $\sigma \in \mathcal{R}$ **do**
  for iteration $k = 1, 2, \cdots$ **do**
    Sample $r_i \sim \mathcal{N}(0, \sigma^2 I)$, $i \leftarrow \{1, \cdots, M\}$. Update:
    $x'_k = proj_{x',\epsilon}\big(x'_{k-1} + \alpha \cdot sgn\big(\nabla_x \mathcal{L}_\rho(x'_{k-1}, \lambda_{k-1}, r)\big)\big)$.
    $\lambda_k = \lambda_{k-1} + \alpha \frac{\partial \mathcal{L}(x'_k, \lambda_{k-1}, r)}{\partial \lambda}$.
  end for
  Input x'_k to Algorithm 1 and output m, $\mathcal{M} \leftarrow \mathcal{M} + m$.
end for
Output $x' \leftarrow \arg\max_{x' \in \mathcal{X}'} \mathcal{M}$.

where

$$\epsilon^*(\sigma) = \underset{\epsilon = ||x'-x||_\infty < \eta}{\arg\min} D_h(x', x_t), \tag{4}$$

s.t.

$$\mathbb{E}_{r \sim \mathcal{N}(0,\sigma^2 I)} \Big[D_h(x' + r, x_t) \Big] \leq T_h, \tag{5}$$

The inner optimization (4) aims to find the optimal perturbation that minimizes the hamming distance between x' and target x_t. (5) stipulates an additional constraint to keep $x' + r$ targeting at x_t as well, where r is drawn from isotropic Gaussian distribution with the input variance σ^2.

Optimization. We solve the inner optimization (4) first. This constrained optimization problem can be solved via Lagrangian relaxation and dual gradient ascent. Denote $g(x') = \mathbb{E}_{r \sim \mathcal{N}(0,\sigma^2 I)}[D_h(x' + r, x_t)] - T_h$. The dual problem is,

$$\max_{\lambda} \min_{x'} \Big(D_h(x', x_t) + \lambda^\top g(x') \Big). \tag{6}$$

Denote $\mathcal{L}$ as the Lagrangian. x' can be optimized with projected gradient descent, and alternatively updating λ with gradient ascent:

$$\begin{aligned} x'_k &= proj_{x',\epsilon}\Big(x'_{k-1} + \alpha \cdot sgn\big(\nabla_{x'} \mathcal{L}(x'_{k-1}, \lambda_{k-1}, r)\big)\Big) \\ \lambda_k &= \lambda_{k-1} + \alpha \frac{\partial \mathcal{L}(x'_k, \lambda_{k-1}, r)}{\partial \lambda} \end{aligned} \tag{7}$$

Note that calculating the exact gradient of $g(x')$ in $\nabla_{x'} \mathcal{L}$ involves high-dimensional integrals. Thus, we approximate the gradient with Monte Carlo sampling,

$$\nabla_{x'} g(x') \approx \nabla_{x'} \Big(\frac{1}{M} \sum_{i=1}^{M} D_h(x' + r_i, x_t) \Big) \tag{8}$$

by taking M samples. For the outer optimization, since $f_{M_b}(x)$ is unknown, we maximize its expectation based on *Property 1* in the white-box model. For all x' generated by input noise $\sigma \in \mathcal{R}$, we utilize Algorithm 1 to evaluate the adversarial sphere and keep those x' with the largest adversarial sphere. This sanity check is necessary because: 1) though the Lagrangian relaxation allows the optimization problem to be efficiently handled in an unconstrained fashion, the penalty only works as a soft constraint and does not guarantee constraint satisfaction [30]; 2) we only obtain an approximation of $\nabla_{x'} g(x')$. We cannot increase the number of samples M indefinitely since each one requires a network query. In fact, our experiment indicates that $M = 1, 4, 8$ all work well with great convergence as shown in Sec.6.

6. Evaluation

Experimental Setup. We conduct the experiments on ImageNet. Following [9], we randomly select 100 categories and use all the images from these categories in the training and test set as the database and query, respectively. Six networks are considered: ResNet101, ResNet152 [17], ResNext101 [43], SeResNet50 [18], ResNet34 and DenseNet161 [20]. We develop HashNet structure into these networks and the result accuracies are: 76.5, 76.1, 77.5, 64.2, 67.3, 64.9% respectively. Though other networks are also available such as VGG/Inception, the accuracy of their HashNet-integration is below 50% so we focus on these six networks.

We set retrieval thresholds $T_h = 5$, and $T_d = 18$ for vulnerable pairs. For targeted attack, we randomly select 500 images from the test set as the source images (query) to target all 100 classes (one target image from each class). Depending on T_d, we randomly sample 10% vulnerable pairs from the total 500×100 pairs and discard those pairs with hamming distance already less than T_h, and keep normal pairs at the same number. An attack is considered to be successful if it retrieves at least 10 images from the target class. We set l_∞ to 32, step size $\alpha = 1$ and 32 iterations for crafting the adversarial examples. λ is initialized as 1 in Algorithm 2. We compare the proposed Noise-induced Adversarial Generation (NAG) with four benchmarks: FGSM [15], PGD [29], Feature-level Activation Attack(AA) [23], Diversity Inputs (DI) and Diversity Inputs with Momentum (DI-Mom) [42] on targeted attacks[2].

6.1. Black-Box Transferability

We first demonstrate the attack success rate in Table 2 on the six networks. The vertical and horizontal axes represent the source and black-box models respectively. The diagonal blocks are the white-box success rates. For vulnerable pairs, NAG can boost the black-box transferability by $1.2 - 3\times$, with an average of 16.85% success compared with the diversity/diversity-momentum method [42] at 11.33/11.61%, PGD [29] at 11.05% and AA [23] at 9.22%. For normal pairs, NAG generates an average of 1.82% success compared with 0.516%, 0.546%, 0.87% and 0.22% of the four benchmarks respectively. Some model combinations achieve phenomenal improvements such as ResNet152 $\rightarrow$ ResNet101, which yields almost $2 - 3\times$ performance boost.

Note that DI/DI-Mom attempt to reduce overfitting of the adversarial example to the white-box model via input diversity. This may undermine their white-box performance

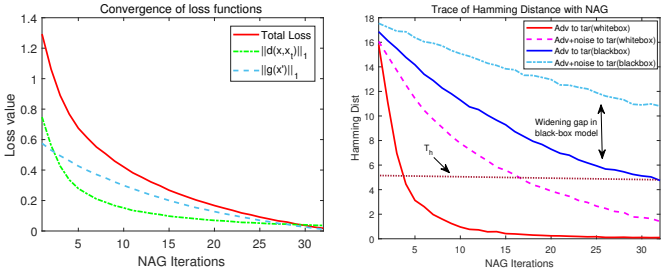
[2]We do not compare with the *Universal Adversarial Perturbation* (UAP) attack here [31, 24], since it is designed for untargeted attack.

		ResNet101		ResNet152		ResNext101		SeResNet50		ResNet34		DenseNet161	
		Vul	Normal	Vul	Normal	Vul	Normal	Vul	Normal	Vul	Normal	Vul	Normal
ResNet101	FGSM	20.6	0.0	8.1	0.0	6.8	0.0	2.5	0.0	5.6	0.0	1.7	0.0
	PGD	98.0	93.1	12.3	1.6	11.5	0.6	1.5	0.0	14.2	0.4	2.4	0.0
	AA	**99.1**	**98.5**	11.0	0.8	10.2	0.0	0.6	0.0	13.6	0.2	3.8	0.0
	DI	97.2	90.1	13.7	3.0	12.9	0.2	2.3	0.0	15.8	0.4	3.2	0.0
	DI-Mom	97.3	88.1	21.7	2.5	11.2	**4.0**	**5.5**	**1.5**	12.1	**1.0**	4.2	**1.2**
	NAG(ours)	98.7	90.8	**22.3**	**3.1**	**14.4**	0.6	5.4	0.3	**18.2**	0.6	**4.7**	0.0
ResNet152	FGSM	7.6	0.8	28.9	4.6	3.7	0.0	2.3	0.0	9.3	0.0	2.9	0.0
	PGD	13.6	3.6	**99.9**	**100.0**	4.9	0.8	3.3	0.1	9.3	0.2	4.9	0.3
	AA	11.6	2.3	99.3	99.9	3.6	0.3	2.5	0.0	8.9	0.1	5.5	0.0
	DI	14.1	3.8	99.6	98.9	4.0	0.6	2.6	0.0	10.1	0.3	4.9	0.8
	DI-Mom	18.3	1.6	99.6	99.5	4.2	1.6	**7.2**	0.5	10.3	0.3	6.1	1.0
	NAG(ours)	**24.5**	**14.4**	**99.9**	99.9	**12.5**	**5.7**	6.6	**1.5**	**15.1**	**3.9**	**8.4**	**1.6**
ResNext101	FGSM	10.1	0.0	11.5	0.0	34.5	0.1	7.2	0.0	13.0	0.0	1.6	0.0
	PGD	11.9	1.2	11.6	0.8	**99.9**	99.8	9.2	0.1	19.0	0.0	3.6	0.0
	AA	10.3	0.1	10.7	0.0	99.2	**99.9**	10.4	0.0	21.6	0.33	2.3	0.0
	DI	10.0	2.1	12.1	1.1	99.1	97.2	9.0	0.0	20.1	0.1	2.8	0.0
	DI-Mom	12.6	2.1	13.9	0.6	98.7	98.4	9.1	**1.4**	17.9	0.4	2.5	0.2
	NAG(ours)	**21.5**	**4.0**	**21.3**	**4.1**	**99.9**	**99.9**	**15.5**	0.3	**26.5**	**2.7**	**6.1**	**1.0**
SeResNet50	FGSM	8.6	0.0	13.2	0.0	11.3	0.0	32.1	0.1	14.2	0.1	5.0	0.0
	PGD	8.4	0.0	9.9	0.0	10.1	0.0	99.5	99.3	15.0	0.0	3.5	0.0
	AA	11.5	**0.4**	15.6	**0.6**	14.1	0.0	**99.9**	**99.9**	13.9	**0.4**	6.1	0.0
	DI	8.0	0.0	11.9	0.0	13.3	0.0	99.0	95.6	14.2	0.0	5.0	0.0
	DI-Mom	5.0	0.0	13.1	0.0	12.0	0.0	99.3	97.0	9.2	0.0	3.2	0.0
	NAG(ours)	**11.8**	0.0	**20.5**	0.0	**20.1**	**0.1**	99.3	98.5	**18.6**	0.0	**6.2**	**0.4**
ResNet34	FGSM	11.4	0.0	7.9	0.0	11.3	0.0	7.0	0.0	42.1	3.2	2.0	0.0
	PGD	12.9	0.0	8.8	1.0	17.8	0.4	5.5	0.1	100.0	**100.0**	5.7	0.0
	AA	11.5	0.0	6.8	0.0	9.4	0.0	3.0	0.0	98.9	99.0	3.3	0.0
	DI	11.2	0.0	9.3	0.5	17.9	0.4	4.9	0.1	100.0	98.6	5.8	0.0
	DI-Mom	9.1	0.3	7.8	0.1	21.6	0.5	7.8	0.4	100.0	99.1	4.0	0.0
	NAG(ours)	**24.1**	**1.8**	**22.2**	**2.4**	**25.4**	**1.5**	**11.0**	**3.7**	**100.0**	99.1	**9.1**	**0.1**
DenseNet161	FGSM	7.9	0.0	7.3	0.0	7.2	0.0	6.3	0.0	12.9	0.0	7.9	0.0
	PGD	20.2	**0.4**	**30.0**	0.0	17.5	0.0	9.8	3.9	23.4	0.0	94.8	84.4
	AA	3.8	0.0	9.0	0.0	12.4	0.0	11.6	0.0	18.1	0.0	99.6	**99.8**
	DI	26.6	0.0	17.6	0.0	19.0	0.0	10.2	3.0	**27.6**	0.0	**100.0**	84.8
	DI-Mom	26.8	0.0	21.9	0.0	21.7	0.0	8.9	**5.0**	19.6	0.0	**100.0**	89.0
	NAG(ours)	**32.0**	0.0	19.4	0.0	**23.2**	0.0	**11.7**	0.8	27.5	0.0	**100.0**	79.2

Table 2: Attack success rates (%) of vulnerable/normal pairs. The diagonal blocks indicate the white-box success rates.

compared to PGD as observed in the diagonal blocks. Nevertheless, NAG does not generally come with such a sacrifice. We also observe that the success rates are essentially higher under the same family of ResNet. This is expected and consistent with the previous works [42, 26] because the cosine similarity of gradient directions is much higher than that of a different family [26]. Finally, note that we adopt a strict retrieval threshold of 5. If the application permits larger T_h, the corresponding hamming ball would be proportionally larger, so as the black-box transferability rates.

Convergence of Adversarial Loss. To see how NAG meets the objectives, we pick a representative model pair and trace the loss convergence by averaging the generation of all the adversarial examples shown in Fig.5(a). For clarity, we plot the normalized $||d(x', x_t)||_1$, $||g(x')||_1$ and the total loss in Eq. (6), which are proportional to the hamming distance. All of them can converge in the white-box source model. Initially, $||g(x')||_1$ is larger than 0, indicating that constraint



(a) (b)

Figure 5: Trace of adversarial loss curves and effectiveness of NAG in white/black box. (a) Trace of loss curves. (b) Trace hamming distance of successful transfers.

(5) has not been satisfied yet, i.e., r pushes x' out of the adversarial region. As learning progresses, the distance between $x' + r$ and x_t approaches T_h. Fig.5(b) shows the trace of hamming distance of (x', x_t) and $(x' + r, x_t)$ in the white-box and black-box models. As NAG attempts to push

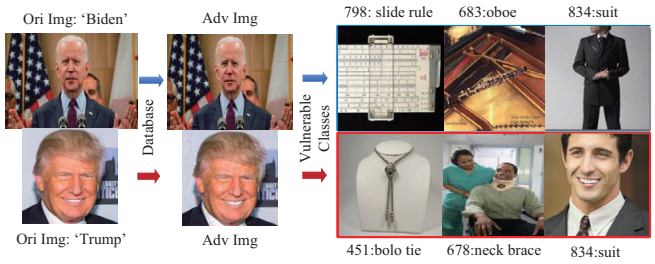


Figure 6: Case study of retrieving the adversarially-crafted, out-of-distribution images of presidential candidates from normal queries.

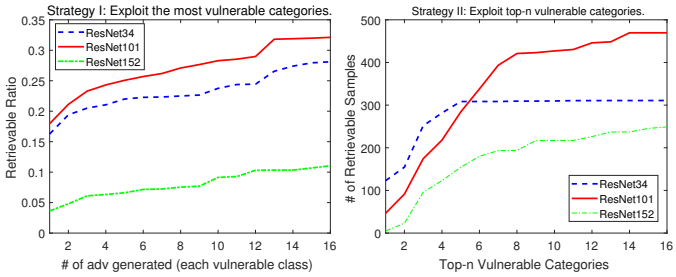
both x' and $x'+r$ within T_h of x_t on the white-box model, the former converges much faster, because the main objective minimizes $D_h(x', x_t)$. The black-box model is more difficult: $D_h(x', x)$ of successful transfers can make to T_h, whereas $D_h(x'+r, x_t)$ is still distant from T_h. Even though we do not expect that $D_h^{M_b}(x'+r, x_t) \leq T_h$, the results suggest further room for improvement if the adversarial sphere is large enough on an ensemble of models [39]. We leave this for future exploration due to space limit.

6.2. Case Studies of Simulating Real-World Attacks

We also conduct some case studies to simulate the real-world attacks that the adversarial images are not part of the training set. This mimics when the malicious images are collected by the database, but not used in training. **Case Study I: Promote Your Favorite Candidate.** First, we present a case study that supporters attempt to advertise their favorable presidential candidate by abusing the vulnerable categories and divert normal queries. We use ResNext101 $\rightarrow$ ResNet152 as an example. Fig.6 shows the original images of "Joe Biden" and "Donald Trump" with adversarial inputs, targeting one of the three vulnerable categories "slide rule", "oboe", "suit" for Biden, and "bolo tie", "neck brace", "suit" for Trump. It is not surprising that "suit" came out as the vulnerable category since the original images share similar content (but does not succeed with direct retrieval $\leq T_h$).

We define the *retrievable ratio* as the percentage of images in a class that would contain the adversarial image as part of its query results, and use this metric to assess the coverage/impact of the attack on the original categories. The retrievable ratios are: 66%, 25% and 13% targeting at "suit", "slide rule", "oboe" for Biden and 41%, 8%, 5% targeting at "neck brace", "suit" and "bolo tie" for Trump, respectively using NAG. We can see that target categories with similar visual appearance enjoy high retrievable ratio - almost half of the images in those categories are impacted by our attack. Other categories with lower visual similarity can be also exploited with 5-25% retrievable ratio, which is still significant to subvert the basic principles of image retrieval systems.

Case Study II: Advertising for Free. To evaluate at a larger scale, we conduct another case study to utilize the advertisement dataset [21] with a similar objective



(a) (b)

Figure 7: Evaluation of retrievable ratio/number of normal queries (a) S(I): Exploit the most vulnerable categories. (b) S(II): Exploit top-n categories.

to make those out-of-distribution advertisement retrievable from user's normal queries. Here, we assume the attacker is resource-limited who only generates a fixed number of adversarial examples. We evaluate the following strategies. S(I): For each advertisement image, pick the most vulnerable category with the minimum hamming distance, and generate n adversarial images for each advertisement image. We can think this as a *depthwise* strategy. S(II): Pick the top-n vulnerable categories and generate one adversarial image for each category. This can be treated as *breadthwise* across multiple vulnerable categories.

Fig.7 compares their effectiveness using 32 advertisement images to generate 32×16 adversarial images ($n = 16$). We use ResNext101 as the white-box source network and transfer to ResNet34/101/152 using NAG. Fig.7(a) shows the retrievable ratios from those targeted categories - the results increase almost linearly with the number of adversarial images generated, with 20-30% images from the original categories impacted by our attack. This translates to: 16 adversarial images per category have corrupted the results of 300-400 normal images. Fig.7(b) shows the total number of images impacted in the top-n vulnerable categories, which amounts a comparable number to S(I). We can see that each strategy has their own advantages: once the most vulnerable class is the top-search class, the attacker may follow S(I) to reap high coverage in those categories; otherwise, if the query patterns are more scattered into multiple categories, S(II) would be more effective.

7. Acknowledgement

This work was supported in part by the U.S. National Science Foundation under grant number 2044841, 2007386 and the State of Virginia Commonwealth Cyber Initiative.

8. Conclusion

In this paper, we study the targeted black-box transferability attack in deep hashing. We connect transferability to the adversarial subspace and propose an implicit technique to estimate its volume using random noise. Then we further develop a new attack to craft more transferable adversarial examples. We evaluate all these efforts with extensive experimental results and demonstrate remarkable improvements compared to the previous works.

References

[1] Bing. https://www.bing.com/. 1

[2] Google image search. https://www.google.com/imghp. 1

[3] How google uses the picture you search with. https://support.google.com/websearch/answer/1325808. 1

[4] How search organizes information. https://www.google.com/search/howsearchworks/crawling-indexing/. 1

[5] Pailitao product search. http://www.pailitao.com/. 1

[6] Pinterest visual search tool. https://www.pinterest.com/. 1

[7] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *International Conference on Learning Representations*, 2018. 2

[8] Thomas Brunner, Frederik Diehl, Michael Truong Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4958–4966, 2019. 1, 2

[9] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE international conference on computer vision*, pages 5608–5617, 2017. 1, 2, 3, 6

[10] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017. 2

[11] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *International Conference on Machine Learning*, 2019. 4

[12] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 2

[13] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7714–7722, 2019. 2

[14] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pages 1632–1640, 2016. 5

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2, 3, 4, 6

[16] M Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. Where to buy it: Matching street clothing photos in online shops. In *Proceedings of the IEEE international conference on computer vision*, pages 3343–3351, 2015. 1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 6

[19] Shengyuan Hu, Tao Yu, Chuan Guo, Wei-Lun Chao, and Kilian Q Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In *Advances in Neural Information Processing Systems*, pages 1635–1646, 2019. 4, 5

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6

[21] Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, Nathan Ong, and Adriana Kovashka. Automatic understanding of image and video advertisements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1705–1715, 2017. 8

[22] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *International Conference on Machine Learning*, 2018. 2

[23] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019. 2, 6

[24] Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. Universal perturbation attack against image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4899–4908, 2019. 6

[25] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2064–2072, 2016. 1, 2, 3

[26] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *International Conference on Learning Representations*, 2017. 1, 2, 4, 7

[27] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. Who's afraid of adversarial queries? the impact of image modifications on content-based image retrieval. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 306–314, 2019. 1, 3

[28] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018. 4

[29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 2, 3, 6

[30] Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Imposing hard constraints on deep networks: Promises and limitations. *CVPR Workshop*, 2017. 6

[31] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. 6

[32] Rangeet Pan, Md Johirul Islam, Shibbir Ahmed, and Hridesh Rajan. Identifying classes susceptible to adversarial attacks. *arXiv preprint arXiv:1905.13284*, 2019. 2

[33] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In *International Conference on Machine Learning*, pages 5498–5507, 2019. 4

[34] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11292–11303, 2019. 4

[35] Yucheng Shi, Siyu Wang, and Yahong Han. Curls & whey: Boosting black-box adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6519–6527, 2019. 2

[36] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna Estrach, Dumitru Erhan, Ian Goodfellow, and Robert Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 4

[37] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016. 4

[38] Giorgos Tolias, Filip Radenovic, and Ondrej Chum. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 3

[39] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *International Conference on Learning Representations*, 2018. 2, 8

[40] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017. 4, 5

[41] Yanru Xiao, Cong Wang, and Xing Gao. Evade deep image retrieval by stashing private images in the hash space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9651–9660, 2020. 3

[42] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 1, 2, 3, 6, 7

[43] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 6

[44] Erkun Yang, Tongliang Liu, Cheng Deng, and Dacheng Tao. Adversarial examples for hamming space search. *IEEE transactions on cybernetics*, 2018. 1, 3

[45] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018. 2

[46] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 1, 2