Fall 12-2022

Towards Privacy and Security Concerns of Adversarial Examples in Deep Hashing Image Retrieval

Yanru Xiao
Old Dominion University, sugarruy@gmail.com

TOWARDS PRIVACY AND SECURITY CONCERNS OF ADVERSARIAL EXAMPLES IN DEEP HASHING IMAGE RETRIEVAL

by

Yanru Xiao
B.E. June 2017, Central South University, China

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
December 2022

Approved by:

Cong Wang (Director)

Ravi Mukkamala (Co-Director)

Rui Ning (Member)

Lusi Li (Member)

Chunsheng Xin (Member)

ABSTRACT

TOWARDS PRIVACY AND SECURITY CONCERNS OF ADVERSARIAL EXAMPLES IN DEEP HASHING IMAGE RETRIEVAL

Yanru Xiao
Old Dominion University, 2022
Director: Dr. Cong Wang
Co-Director: Dr. Ravi Mukkamala

With the explosive growth of images on the internet, image retrieval based on deep hashing attracts spotlights from both research and industry communities. Empowered by deep neural networks (DNNs), deep hashing enables fast and accurate image retrieval on large-scale data. However, inheriting from deep learning, deep hashing remains vulnerable to specifically designed input, called adversarial examples. By adding imperceptible perturbations on inputs, adversarial examples fool DNNs to make wrong decisions. The existence of adversarial examples not only raises security concerns for real-world deep learning applications, but also provides us with a technique to confront malicious applications.

In this dissertation, we investigate privacy and security concerns in deep hashing image retrieval systems related to adversarial examples. Starting with a privacy concern, we stand on users side to preserve privacy information in images, which can be extracted by adversaries by retrieving similar images in image retrieval systems. Existing image processing-based privacy-preserving methods suffer from a trade-off of efficacy and usability. We propose a method introducing imperceptible adversarial perturbations on original images to prevent them from being retrieved. Users upload protected adversarial images instead of the original images to preserve privacy while maintaining usability. Then we shift to the security concerns. We act as attackers, proactively providing adversarial images to retrieval systems. These adversarial examples are embedded to specific targets so that the user retrieval results contain our unrelated adversarial images, e.g., users query with a "Husky dog" image,

but retrieve adversarial "dog food" images in the result. A transferability-based attack is proposed for black-box models. We improve black-box transferability with the random noise as the proxy in optimization, achieving state-of-the-art success rate. Finally, we stand on retrieval systems side to mitigate the security concerns of adversarial attacks in deep hashing image retrieval. We propose a detection method that detects adversarial examples in the inference time. By studying unique adversarial behaviors in deep hashing image retrieval, our proposed method is constructed on criterions of these adversarial behaviors. The proposed method detects most of the adversarial examples with minimum overhead.

ACKNOWLEDGMENTS

I could not finish my Ph.D journey without help from many people.

Firstly, I would like to thank my PhD advisor, Dr. Cong Wang, for advising me with the greatest patience during my Ph.D years. Dr. Wang gave me the maximum freedom to explore whatever topics I am interested in and supported me through mentorship and scholarship. Under his advice, I stepped into the research area of AI security, from a naive undergraduate student to a mature Ph.D researcher. Beyond advice and scholarship, his passion for both life and work impacts me tremendously. I could never have had the chance to finish my study without Dr. Wang's mentorship.

I would like to thank my co-advisor, Dr. Ravi Mukkamala, and committee members, Dr. Chunsheng Xin, Dr. Lusi Li, and Dr. Rui Ning, for giving me advice through this dissertation process.

I would like to thank all faculty and student researchers in Cybersecurity Center (CCSER) at ODU, for having me and providing me with a charming research atmosphere.

I would like to thank Dr. Luming Liang and all the labmates at Microsoft, for a wonderful summer internship working together on novel research topics. This internship expanded my horizons of research, and helped me conduct high-impact and practical research.

Finally and most importantly, I would like to thank my parents for giving me birth, raising me up, funding me into college, and supporting me with their selfless love throughout my life.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

Deep neural networks (DNNs) become the *de-facto* standard tools when dealing with a large amount of vision data in artificial intelligence (AI) systems. With the development of both network architectures [1–5] and training paradigms [6–8], DNNs-based methods not only achieve higher performance compared to traditional machine learning methods, but also eliminate the requirements of domain knowledge to design the features. Universal backbone DNN models have been adopted to different vision tasks and have dominated in computer vision fields [1–5, 9, 10] : With a classification head and a set of single-class labeled image data, a classifier [3] surpasses the human ability to distinguish pictures in ImageNet; With a localization head and bounding box labeled image data, an object detector [11, 12] is empowered to locate objects fast and accurately; With a deep hashing head and relation-labeled image data, an image retrieval [13–18] enables efficient similarity content-based image search in large scale.

Despite their great success, DNNs are found to be vulnerable to some specific kinds of perturbations on the inputs [19]. These perturbations are called adversarial perturbations and the perturbed inputs are called adversarial examples. Adversarial perturbations are imperceptible to human eyes but they are highly capable to make the model misclassification into any other label (*untargeted attack*) or a specific label (*targeted attack*), which makes them an obvious security concern for model safety, but also makes them an advanced image processing tool to work against models. Several works have studied adversarial examples in classification [19–22], object detection [23–25], and semantic segmentation [23, 26]. However, adversarial examples in image retrieval have not been systematically studied before.

1.1 DEEP HASHING IMAGE RETRIEVAL

Image retrieval is one of the most important applications on the internet. Several companies, including Google [27], Bing [28], Amazon [29], Alibaba [30] have deployed their own

version of image retrievals and opened their interfaces to the public. In general, an image retrieval system contains three key participants: *Database*, *Model*, and *User*:

- *Database.* A database stores images and their corresponding descriptors (features) for retrieval. It also collects new images from users and from the internet.
- *Model.* A model converts images into descriptors and performs searches to find similar images from the database. The model also opens an interface for users to query the database with images.
- *User.* A user uses the interface provided by the model, by querying an image to get similar images from the database. The user also provides images to the database passively or proactively.

The most important participant among them is *model*. Before the era of deep learning, traditional hashing methods [31–34] were widely used in image retrieval. They quantize images into low-dimensional binary hash codes, making high-efficient retrieval in hash space possible. However, they are subject to low-precise feature descriptors, resulting in low retrieval precision. Empowered by DNNs, large-scale image retrieval has been dominated by deep hashing based methods [13–18]. Compared to the traditional hashing methods, deep hashing methods learn well-presented features from data in an end-to-end style, so that they achieve remarkably better retrieval performance.

However, deep learning is a double-edged sword. On the one hand, it enables end-to-end learning for deep hashing image retrieval, improving retrieval performance. On the other hand, the vulnerability to adversarial examples [19] is also inherited from DNNs to deep hashing.

1.2 PRIVACY AND SECURITY CONCERNS

Privacy Concern. Images contain lots of information. A picture shared incidentally by users may be found by adversaries to extract private information, including family members, location, income, personal interest, or even sexual orientation for accurate contextual advertising [35–38], or spear phishing [39]. With the help of image retrieval systems, adversaries

are even more powerful. Privates images could be passively collected by image retrieval systems without the owner's consent, and they could be retrieved when adversaries query with similar images.

We investigate the privacy concern and the method to preserve the privacy via adversarial examples in Chapter 2.

Security Concern. Adversarial examples in image retrieval open a new attack vector to invade the system. Attackers could generate a bunch of adversarial examples based on natural images, provide them to the database, and hope they could be retrieved by other users. This attack could improve the exposure of images from attackers, letting more users see them in their retrieval results, which is similar to Search Engine Optimization (SEO) [40] for regular search engines. This attack can be used to override product search results in online shopping, to make users view free "advertisement" images, or even to propagate political banners.

We investigate the security concern in this dissertation. We study a method to enhance the success rate of this attack in Chapter 3. We also study a detection mechanism to mitigates the attack in Chapter 4.

1.3 CONTRIBUTIONS

In this dissertation, the following contributions have been made.

Stashing Image in Deep Hashing Image Retrieval for Privacy Preserving. In Chapter 2, we stand on the *User* side, to alleviate the privacy concern. Our objective is to prevent private images from being retrieved and extracted by adversaries in image retrieval. We proposed a method that introduces an imperceptible adversarial perturbation on the original image. These new images keep the perceptual similarity to original images, but they are stashed into the hash space that maximizes the hamming distances to all the other samples, which prevents them from being retrieved by any other images, including adversaries queries. Compared to other method [41], our proposed method hardens adversaries' efforts by 2 to 7 orders of magnitudes, with negligible computational overhead and perceptual degradation. We also extend this to a more realistic black-box setting, where the model

used in retrieval systems is unknown, demonstrating 30-60% transferability in hash space.

Black-box Transferability Adversarial Attack in Deep Hashing Image Retrieval. The white-box setting, where attackers have perfect model information, yields to a high success rate of adversarial examples in image retrieval. A more practical counterpart is the black-box setting, which assumes that the model is unknown. However, it also makes adversarial examples difficult to generate with limited information. Transferability is an interesting property of adversarial examples that enables black-box attacks. Attackers could generate adversarial examples on a local surrogate model, and hope they remain adversarial in the black-box model. In Chapter 3, we act as attackers to apply a black-box adversarial attack. Our objective is to insert adversarial examples into user retrieval results in deep hashing image retrieval with the black-box setting by using transferability. A targeted attack is needed to make sure that our adversarial examples are mapped into specific categories. By noticing the existence of vulnerable pairs that transfer easily, we utilize these vulnerable pairs based on hamming distance from a surrogate model to enhance the success rate. Then we explore the implications of transferability and find out that the tolerance to random noise of an adversarial example is related to its black-box transferability. We design a targeted attack to generate adversarial examples that are robust to random noise, to enhance its transferability. In extensive experiments, we demonstrate that our proposed attack yields a boost of transferability by $1.2 - 3\times$ on PGD [42], and $1.5\times$ on the diversity techniques [43].

Adversarial Detection in Deep Hashing Image Retrieval. While a set of adversarial attack methods [41, 44–46] has been proposed recently in deep hashing image retrieval, the defending techniques in deep hashing are still in shortage. In classification, adversarial detection methods [47, 48, 48–54] deployed in the inference time distinguish the adversarial examples by their adversarial behaviors. A study on adversarial behaviors in deep hashing is critical to defend against adversarial attacks. In Chapter 4, we act on the model side, designing an adversarial detection method in deep hashing image retrieval. We first deduce the hamming distance distribution from the untargeted adversarial examples to other categories, proposing a new criterion to identify adversarial behaviors of untargeted attacks. Then we

analyze the objectives of targeted attacks, which reduce quantization loss close to zero as a side effect, inducing a criterion to identify targeted attacks. We combine these two criterions with a denoising-based detection method, which measures the disagreement between an input and its denoised transformation. Our proposed method is an unsupervised detection method based on these three criterions. It only relies on the natural images to set thresholds so that it is not biased towards any specific adversarial attacks. The extensive experiments show that our method surpasses defenses adopted from classifications [49, 50, 54, 55], with negligible overhead in the inference time.

1.4 DISSERTATION OUTLINE

The rest of this dissertation is organized into following chapters:

- Chapter 2: We propose a privacy preserve method based on adversarial examples in deep hashing image retrieval, to stash private images in hash space.
- Chapter 3: We explore the transferability of adversarial examples in deep hashing image retrieval, and propose a transfer-based adversarial attack with the proxy of random noise.
- Chapter 4: We study the adversarial behaviors in deep hashing image retrieval, and propose an unsupervised detection method based on three criterions deduced from adversarial behaviors.
- Chapter 5: Conclusions.

CHAPTER 2

EVADE DEEP IMAGE RETRIEVAL BY STASHING PRIVATE IMAGES IN THE HASH SPACE

With the rapid growth of visual content, deep learning to hash is gaining popularity in the image retrieval community recently. Although it greatly facilitates search efficiency, privacy is also at risks when images on the web are retrieved at a large scale and exploited as a rich mine of personal information. An adversary can extract private images by querying similar images from the targeted category for any usable model. Existing methods based on image processing preserve privacy at a sacrifice of perceptual quality. In this chapter, we propose a new mechanism based on adversarial examples to "stash" private images in the deep hash space while maintaining perceptual similarity. We first find that a simple approach of hamming distance maximization is not robust against brute-force adversaries. Then we develop a new loss function by maximizing the hamming distance to not only the original category, but also the centers from all the classes, partitioned into clusters of various sizes. The extensive experiment shows that the proposed defense can harden the attacker's efforts by 2-7 orders of magnitude, without significant increase of computational overhead and perceptual degradation. We also demonstrate 30-60% transferability in hash space with a black-box setting.

2.1 INTRODUCTION

A picture is worth a thousand words. The rapid growth of large image and video collections has made content-based image retrieval possible at a large scale, e.g. Google [27], Pinterest [56], Bing [28] and TinEye [57]. Powered by deep learning, they have been increasingly built into social networks [58], e-commerce [30, 59](e.g., Pailitao from Taobao [60]) and fashion design [61] to capture semantic similarities from visual queries for finer results. Social media, e-commerce websites and even the user's query are utilized as a rich mine of images to train these systems. For instance, 100M photos and videos are uploaded everyday

on Instagram [62]; more than 1B products are listed on Ebay [63]. Google also claims a 7-day storage of queried and uploaded images and utilizes them for further analysis [64].

Although legislation (e.g. GDPR [65]) imposes restrictions on the usage of personal data, for the exploding volume of visual content, there still remains a vague definition of ownership as well as a weak legal boundary between what can be learned and what cannot, from an image. Further, users' awareness of their privacy remains quite subjective towards latent, but sensitive information in their images. The resourceful visual content can be exploited in bewildering ways to learn private information such as family member, location, income, personal interest or even sexual orientation for accurate contextual advertising [35–38] or spear phishing [39]. For example, Facebook has patented a new application of predicting household demographics based on image data [66]. Though these applications expedite search efficiency and product offering, they also compromise user privacy and make privacy trampling easier at a large scale. These issues stretch beyond social media and search engines: any platform with content-based image retrieval shares the same risk of privacy leakage.

Unfortunately, there is less incentive for the platforms to implement privacy guarantees, as long as they are faltering in the grey area of legislation. It is always up to the users to protect their own privacy. Previous approaches utilize image processing such as blurring, darkening and occlusion to evade face recognition [67] or disassociate friend tagging [68], at a sacrifice of degraded visual quality. Another thread of work is to establish a privacy-respecting protocol by an identifiable tag [69, 70], so anyone wearing the privacy tag is excluded from the image. The success of these systems relies on building sophisticated, trusted protocols between the users and the platform, that demands commitments from both sides.

These techniques may be fragile in the eyes of deep learning, which can still extract useful information from the local descriptors. The state-of-the-art image retrieval adopts *deep hashing* for efficient similarity search [13, 14, 16, 17, 71]. It quantizes images in the database into low-dimensional binary codes during training, computes the *hamming distance* from the queried image, and returns relevant images (inadvertently) gathered by the database. A well-trained model would return images with high similarity (usually from the same category).

With some categorical information, e.g., gathering a few images from the targeted category, an adversary can query the database and retrieve all the images including those private ones. Thus, to evade retrieval, privacy preservation entails opening the box of deep hashing while maintaining perceptual similarity.

In this chapter, we aim to minimize the chances for the private images to be extracted by introducing a small, crafted perturbation on the original image. Studied in [19–22], deep neural networks are vulnerable to adversarial inputs - perturbations that are inconspicuous to human eyes can be added to cause misclassification. In principle, deep hashing should inherit these vulnerabilities by design. A recent work shows that maximizing the hamming distance from the original image in hash space would make the system return an irrelevant image to the query, which can be utilized directly to protect the private images. Nevertheless, by implementing the strategy, we find that it can only defend weak adversaries, who only exploit the original category. Strong adversaries are more common in reality; they could enumerate all the categories and expose the private images in brute force. To tackle this challenge, we propose a new *cluster-based weighted distance maximization* that can transform the hash code into the subspaces away from all the categories.

The main contributions are summarized below. First, we propose to utilize adversarial techniques for privacy preservation and identify the limitations of the existing approach against strong adversaries. Second, we develop a new mechanism to stash samples into the hash space that maximizes the hamming distance to all the classes, while maintaining perceptual similarity. Finally, we conduct experiments on various datasets and demonstrate that the proposed mechanism successfully hardens the attack efforts by 1-3 orders of magnitude compared to [41], while achieving minimal perceptual dissimilarity. We show that 30-60% of the protected images can successfully transfer to an unknown model in a blackbox setting.

The rest of the chapter is organized as follows. Section 2.2 introduces the related works. Section 2.3 motivates this study by defining the threat model and identifying the limits of the existing approach. Section 2.4 presents a new defense against strong adversaries. Section 2.5 evaluates the proposed mechanism and Section 2.6 concludes this work.

2.2 BACKGROUND AND RELATED WORKS

In this section, we briefly review past efforts on deep image retrieval, adversarial examples, and privacy preserving, from which inspires our innovations.

2.2.1 DEEP IMAGE RETRIEVAL

Traditional image retrieval works on a vector of hand-crafted visual descriptors [72, 73], followed by a separate process of projection and quantization to encode feature vectors into binary codes. Propelled by the success of deep learning, the new deep image retrieval enables learning of pairwise similarity from end-to-end [13,14,16,17]. It transforms high-dimensional real-valued inputs into the binary hash codes so similarity search can be performed efficiently by calculating the *hamming distance*. These systems typically consist of a *database* and a *model*. The database contains a finite set of images as the retrieval results; the model accepts query and returns retrieved images. The objective is to learn a nonlinear hash function to map input $x \rightarrow h(x) \in \{-1, +1\}^m$ into an m-bit binary code. A typical range of m is between 16 to 128 depending on the application requirements, which is made less than the original image dimension.

In addition to the convolutional and densely connected layers, a hash layer is introduced for the binarization process, in order to mitigate the quantization error. It converts a continuous representation z into discrete hash code by the sign function $sgn(z)$.

Since the sign function is not compatible with back propagation due to non-smoothness, the key is to build a function for continuous approximation. For example, HashNet [14] adopts the hyperbolic tangent function, $sgn(z) = \lim_{\beta \rightarrow \infty} \tanh(\beta z)$. By tuning the scaling parameter β during the learning process, the function converges to the sign function when $\beta \rightarrow \infty$. Similar to deep features in their floating point format, hashing concentrates similar images into a Hamming ball. The system usually defines a *retrieval threshold* so any image with smaller hamming distance would be returned as the query results. We refer to the survey [71] for more details.

2.2.2 ADVERSARIAL EXAMPLES

In contrast to their super-human capabilities, neural networks are highly vulnerable to

small perturbations, where purposely crafted perturbations added to the input can make the system misbehave at run-time [19–22]. An efficient attack is the *fast gradient sign method* [20]. It takes a large step in the gradient directions to maximize the loss function, by finding a perturbed image x' with small additive noise ϵ such that $f(x') \neq f(x)$.

$$x' = x + \epsilon \cdot sgn(\nabla_x L(\theta, x, y)), \tag{1}$$

where $L(\cdot)$ is the loss function. θ is the model parameter. ∇ is the gradient. x is the data and y is the true label. Instead of making one-step gradient ascent, the method is extended in [21] as the *basic iterative method* to apply (1) multiple times and clip the image within the ϵ-constraint. Empirical experiments demonstrate that these adversarial examples can not only "fool" the classifier, but also transfer between different models for black-box attacks [74, 75].

2.2.3 PRIVACY PRESERVING

Previous efforts of preserving privacy online mainly focus on web analytics [35,36], mobile advertising [37, 38] and behavioral tracking [76, 77]. To balance privacy and utility, a popular approach is through differential privacy that introduces noise to answers so the service provider cannot detect the presence or absence of a user. Though these mechanisms offer provable foundations on a statistical basis, they are not specialized in protecting inference of a single record, such as the private image being retrieved from the database. Privacy is a growing concern with the wide adoption of deep learning based search methods. Only a few works have utilized adversarial examples for privacy preservation. In [78], a strategy based on adversarial examples is developed to disable the object detections so it cannot identify objects at the first place. An adversarial technique is also developed in [41] to corrupt semantic relationships and make the retrieval system return irrelevant images. Our work extends [41] to tackle strong and adaptive adversaries.

2.3 MOTIVATION

This section motivates the research by defining the threat model and investigating the mechanism in [41] as defense.

2.3.1 THREAT MODEL

We first present the scenario and assumptions made in this chapter. Platforms such as social networks and search engines usually collect user information including profile, email, IP address and most importantly, *pictures*. The platform has deployed a deep image retrieval system such as HashNet-ResNet50 [14] to match imagery content from visual queries for marketing purposes. For profit, the platform also opens an interface for third-party advertisers or data brokers (escalated by calling them *adversaries*) [56, 57], who can match and retrieve similar images from the database for accurate advertising [76, 77]. Since the service is rated per query, the platform does not impose any limit on the number of queries but the adversaries have a fixed amount of budget. Users (*defenders*) have no control over the privacy policy, therefore, they introduce perturbation to prevent personal images from being returned as the retrieval results. The flowchart is illustrated in Fig.1.

To maximize retrieval quality, the adversary collects a dataset (*attack set*) to resemble the database. Similarly, the user also collects a dataset to facilitate the generation of the perturbations. We assume both data sets are independent and identically distributed (i.i.d) with the training set. For simplicity, in this chapter, it is implemented by random selections from the test set. As a first proof of concept in the hash space, we assume the user has complete knowledge about the model (white-box) as [22, 41], including the information of category, structure, parameters, hashing mechanism and loss function. Then we demonstrate the existence of black-box transferability of the proposed mechanism in hash space, when users estimate the model architecture and parameters at the best effort.

2.3.2 HAMMING DISTANCE MAXIMIZATION AS A DEFENSE

The work of [41] fools the hash-based image retrieval system by adversarial examples, which can be also leveraged as a privacy-preserving technique. The objective is to maximize the distance between the perturbed image and the original one, such that the hamming

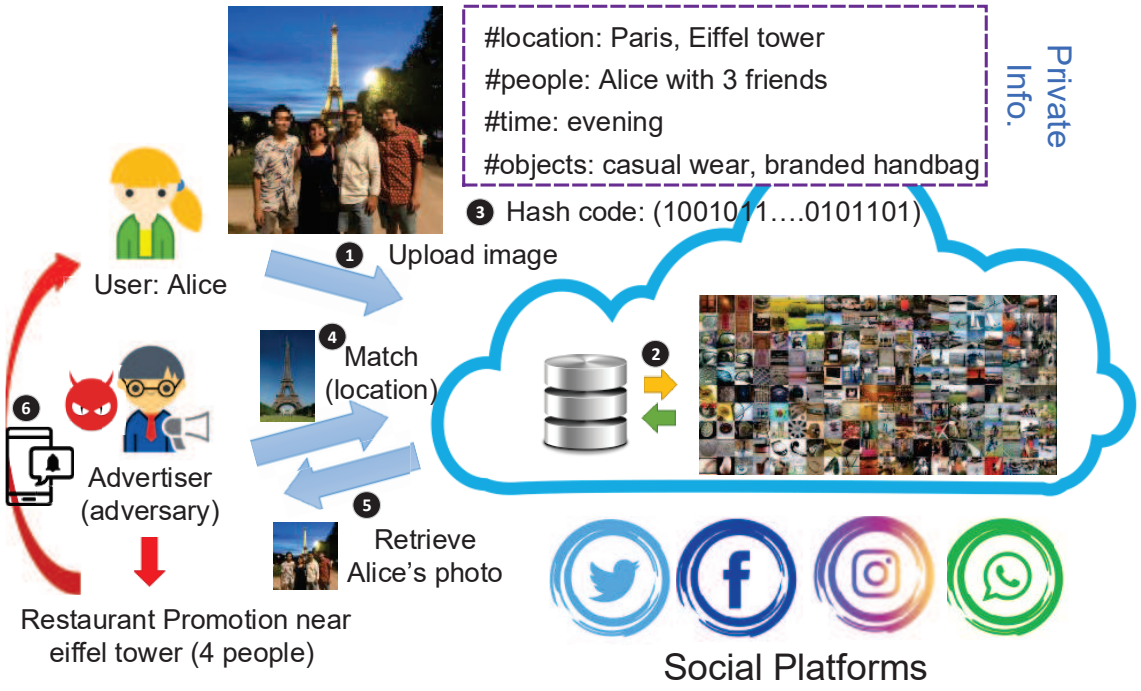


FIG. 1. Illustration of the attack flow: (1) user uploads a photo to the social platform; (2)(3) platform adds the photo into the database, generates hash code; (4)(5) advertiser matches the image via an identical query; (6) advertiser exploits location privacy from the image and pushes nearby promotions onto the user's mobile (even though she has disabled location access on her phone).

distance exceeds the retrieval threshold for that category. More formally, it transforms x into x' by maximizing their hamming distance $\mathcal{D}_h(x, x')$. $\mathcal{D}_h(x, x')$ can be deduced from the inner product of the m-bit hash code [79],

$$\mathcal{D}_h(x, x') = \frac{1}{2}(m - h(x)h(x')^\top), h_i(x) \in \{1, -1\}^{1\times m} \tag{2}$$

where $i \in [1, m]$ and $m = 48$ bits for the HashNet-ResNet50 architecture. The goal is to adjust x' such that the hamming distance is maximized, $\max_{x'} \mathcal{L}(x', x) = -\frac{1}{m}h(x)h(x')^\top$. The problem can be re-written into a least-square style minimization function [41, 79], and shift the negative hash code by $+1$ to $\{0, 2\}$. The ϵ-constraint maintains the perceptual similarity between x and x'.

$$\min_{x'} \mathcal{L}_h(x', x) = \|\frac{1}{m}h(x)h(x')^\top + 1\|_2^2, \tag{3}$$

$$s.t.\ |x - x'| < \epsilon. \tag{4}$$

Though effective against trivial queries targeting at the *original category* of the protected image, the defense is vulnerable when the adversary enumerates through the rest categories and extracts the protected image by brute force. This is because simple maximization of hamming distance from the original image may unwittingly push the perturbed image into the vicinity of other categories. Fig.2 visualizes such cases in t-SNE on the MNIST dataset. As observed, simply hiding the private images into the subspaces of some irrelevant categories is still susceptible to stronger and adaptive adversaries. To gain more insights, we present some preliminary results based on MNIST [80] and CIFAR10 [81] in Fig. 3.

2.3.3 KEY OBSERVATIONS

The adversary could expose all private images by enumerating through the entire attack set. Since the adversary is budget-limited, he wants to minimize such effort. Thus, we evaluate the average number of queries for the adversary to extract the private images, when

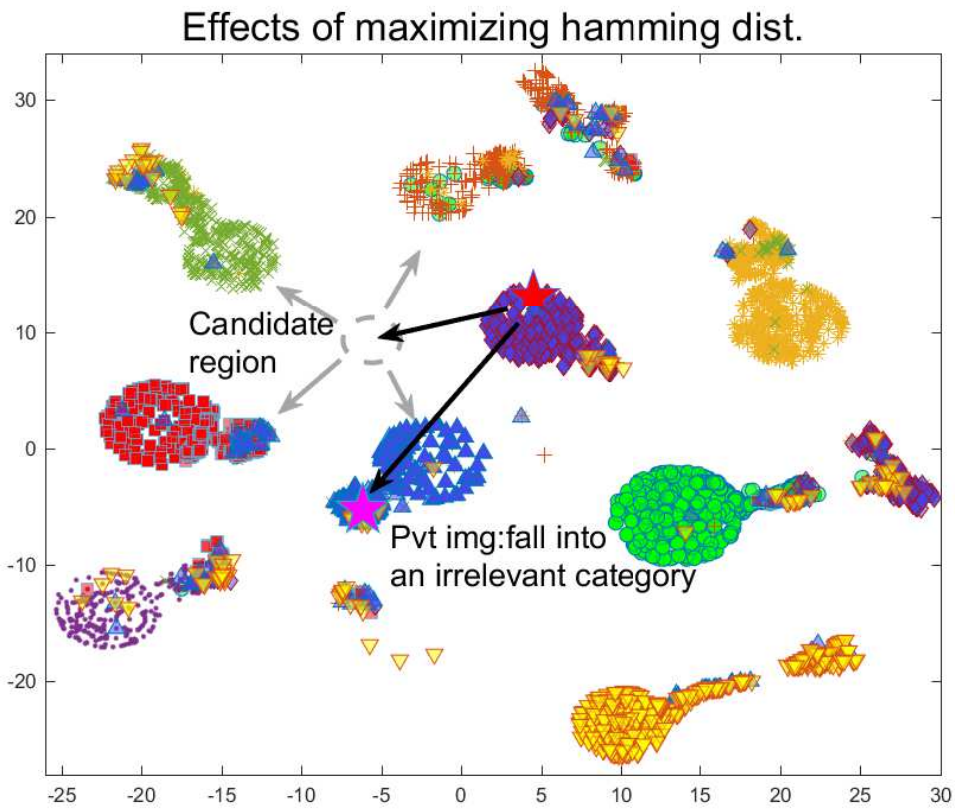
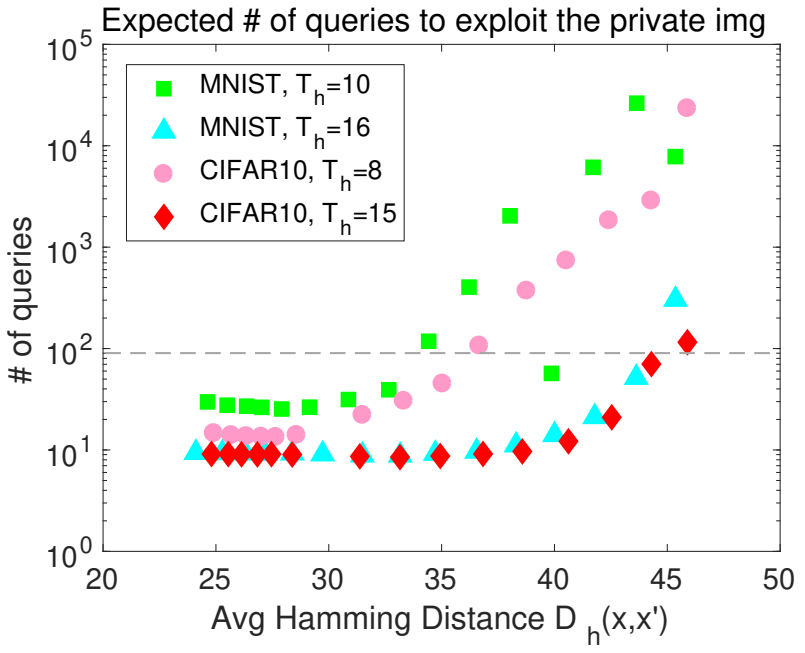


FIG. 2. t-SNE visualization of learned hash codes from MNIST: hamming distance maximization has (accidentally) driven the private image into an irrelevant category.

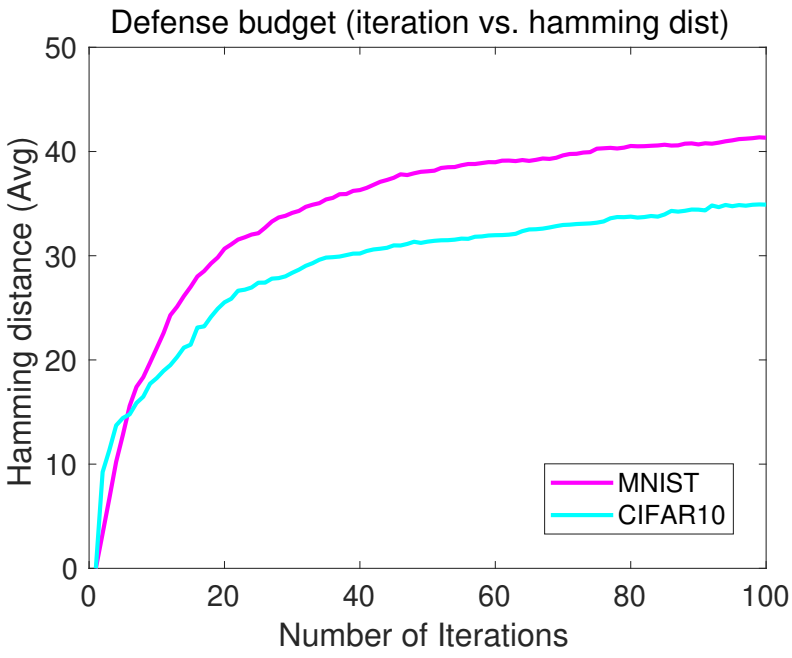
a random image is queried from the attack set each time. If a private image is mapped to the vicinity of n images in the attack set of size N, the probability of retrieving this image is n/N. The expected number of queries is N/n.

Fig.3 shows the expected number of queries against strong attackers and the defense efforts in terms of iterations to generate the crafted perturbations [41]. The retrieval threshold T_h is selected according to the best F-1 score and precision.

Observation 1. The attack efforts trend up parabolically with the increasing hamming distance between x and x'. However, a strong adversary can still extract the private images within 100 queries for most of the hamming distances.



(a)



(b)

FIG. 3. Brute-force attacks as a defense (a) expected number of queries to extract private images; (b) defense budget (# iterations).

Observation 2. The average hamming distance is difficult to maximize further after a certain number of iterations. For example, its average saturates around 40 and 35 after 100 iterations on MNIST and CIFAR10 as observed in Fig.3(b), leaving a large gap to the total hash bits of $m = 48$.

Observation 3. When the categorical features are more dispersed in the hamming space, the protected image is more prone to fall into the retrieval threshold of some samples. It is validated in Fig. 3 given that the attacks on CIFAR10 require less effort than MNIST, due to higher intra-class diversity of CIFAR10. This makes the defense using hamming distance maximization flimsy in the real world, where data has complex and high intra/inter-class diversity.

We can see from these observations that defense is challenging against strong adversaries. Instead of naive maximization from the original category, the optimization should be guided within a narrow subspace to avoid being: 1) exposed from the original category; 2) extracted via querying the rest categories; 3) degrading visual quality. To meet these requirements, we propose a new mechanism in the next section.

2.4 CLUSTER-BASED WEIGHTED DISTANCE MAXIMIZATION

We propose a new mechanism called *cluster-based weighted distance maximization.* The idea is parallel to the center loss [82], which aims to enhance the discrimination of inter-class features and pull the intra-class features towards their centers for better classification. Here, however, we are learning through the adversarial lens for generating a hashcode via perturbing the input image, such that the distance to the hash centers is maximized. To account for intra-class variations, we represent each class with several centers, rather than a single one [82]. The hamming distance to the centers also exhibits heterogenous distributions across various categories. Samples may have high density around the center for some categories while others may scatter more evenly. Thus, the optimization should be aware of the intra-class distributions and their hamming distance to the center; otherwise, the protected image may fall into high-density regions, where all the samples have similar hash codes. The attacker can easily exploit these regions to retrieve the private image with high chances.

Our Mechanism. To address intra-class variations, we further partition the hash codes by a clustering method. For the set of hash codes $\{h(x_i)\}_{i=1,\cdots,N}$, we re-organize them into a number of k distinct clusters $\mathcal{C}_i$, $1 \leq i \leq k$. Existing clustering techniques such as k-means [83] and density-based DBSCAN [84] can be adopted (their pros and cons will be compared in Sec. 2.5.1).

After the clusters are found, we develop a weighted loss function to characterize the in-cluster hamming distance distribution. The goal is to push x' away from the cluster centers such that the number of samples returned by a query using x' is minimized. Because hamming distance is symmetric, this is equivalent to our original intention that maximizes $\mathcal{D}_h(x, x')$, so x' is not returned by the query, when the attacker queries any image x from $\mathcal{C}_i$. Define $F_i(d)$ as the cumulative distribution of the number of samples with distance d from center c_i. For a total number of k clusters, the new objective minimizes a new loss function $\mathcal{L}_c$ defined as,

$$\min_{x'} \mathcal{L}_c(x') = \sum_{i=1}^{k} \|F_i\big(\frac{1}{2}(m - h(x')h(c_i)^\top)\big)\|_2^2, \tag{5}$$

$$s.t. |x - x'| < \epsilon. \tag{6}$$

where $h(c_i)$ is the hash code of the i-th cluster center c_i, m is the total hash bits ($m = 48$).

Optimization. Set the initial image of x' as x, x' can be updated in an iterative manner,

$$x' = clip_{x,\epsilon}\big(x' + \epsilon \cdot \nabla_x \mathcal{L}_c(\theta, h(x'), \{h(c_i)\}_{i=1}^k)\big) \tag{7}$$

The gradient of the loss function can be calculated as,

$$\begin{aligned} \frac{\partial \mathcal{L}_c}{\partial h(x')} &= \sum_{i=1}^{k} 2F_i\big(\mathcal{D}_h(x', c_i)\big) \frac{\partial F_i(\mathcal{D}_h(x', c_i))}{\partial \mathcal{D}_h(x', c_i)} \frac{\partial \mathcal{D}_h(x', c_i)}{\partial h(x')} \\ &= -\sum_{i=1}^{k} F_i\big(\mathcal{D}_h(x', c_i)\big) \frac{\partial F_i\big(\mathcal{D}_h(x', c_i)\big)}{\partial \mathcal{D}_h(x', c_i)} h(c_i). \end{aligned} \tag{8}$$

To use gradient-based optimization, $F_i(\cdot)$ should be a differentiable function. We learn a least square regression for each cluster $1 \leq i \leq k$, based on the hamming distance j to the

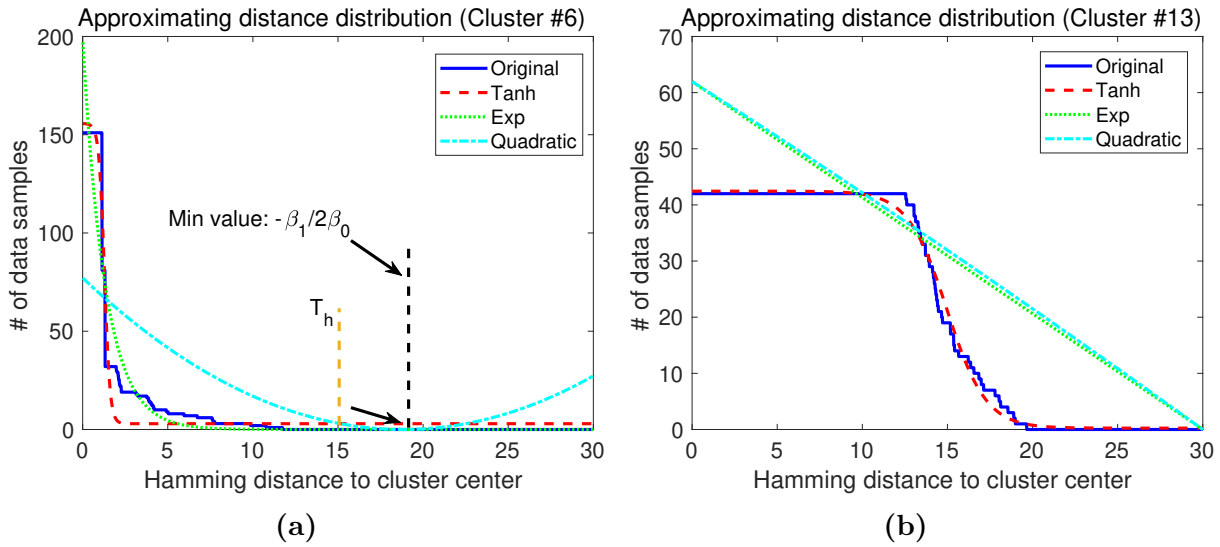


FIG. 4. Least square approximation of in-cluster sample distributions using hyperbolic tangent, exponential and quadratic functions on CIFAR10 with k-means clustering (a) Cluster #6; (b) Cluster #13.

center ($j \in [1, 30]$) and the number of samples y_j,

$$\hat{F}_i = \arg\min_{F_i} \sum_{j=1}^{m} \|F_i(d_j) - y_j\|_2^2. \tag{9}$$

The parameters can be derived by a closed form solution, $\hat{\boldsymbol{\beta}} = (\boldsymbol{d}^T\boldsymbol{d})^{-1}\boldsymbol{d}^T\boldsymbol{y}$.

To examine the effectiveness of regression, we plot the relationships between the d and y (shown as "Original") in Fig.4 for CIFAR10. In most of the cases, the images are concentrated around the cluster centers (Fig.4(a)). There are also some clusters that samples are more scattered (Fig.4(b)). To minimize the square error, it is tempting to adopt high-order polynomials for better characterization, but they would slow down the defense process due to high computations. For training stability, we adopt quadratic regression and compare them with nonlinear regressions of hyperbolic tangent and exponential in Fig. 4. The quadratic regression demonstrates empirical advantages summarized by the following properties.

Property 1. The convexity of quadratic function facilitates the convergence of the loss

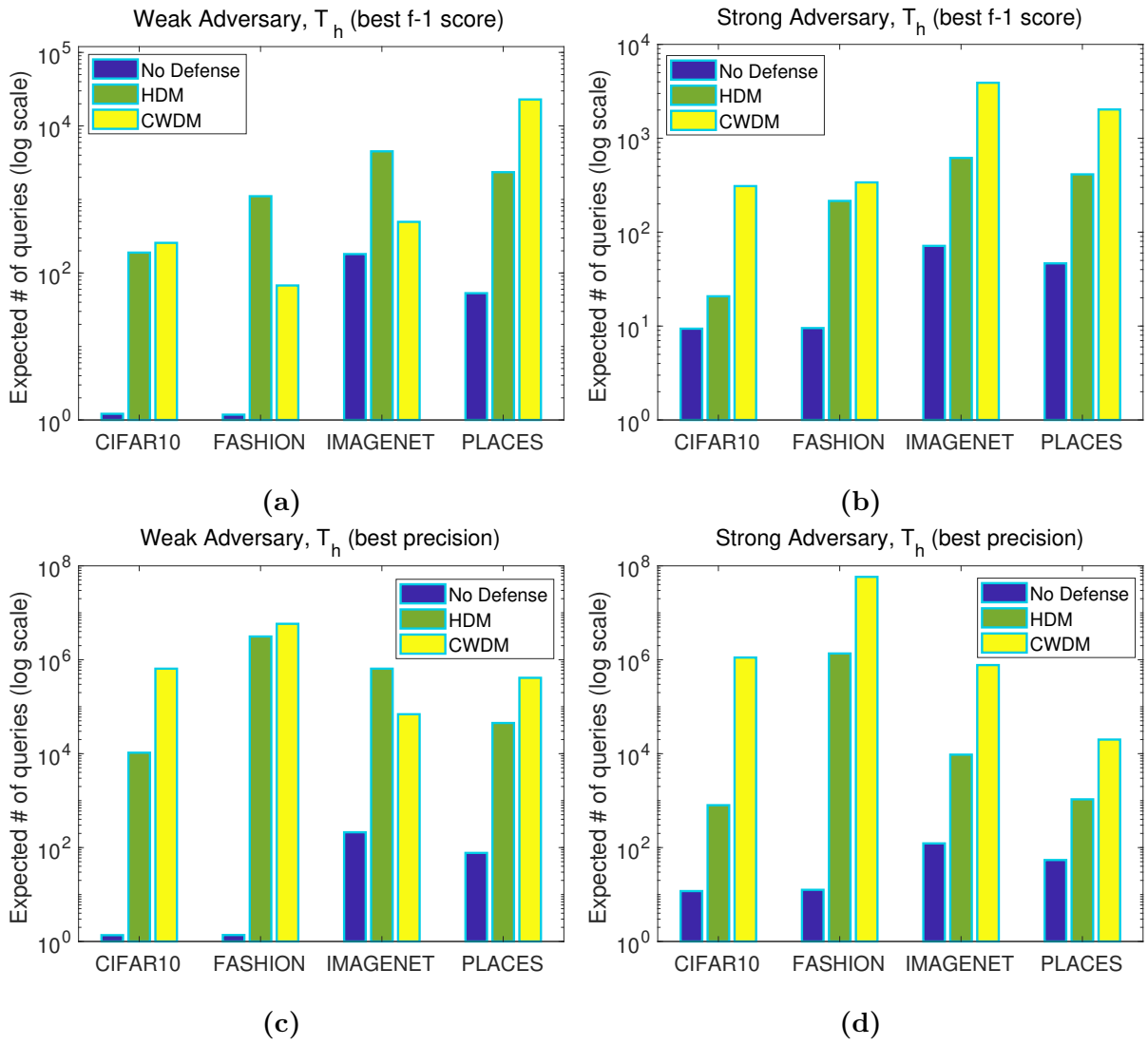


FIG. 5. Expected number queries to expose the private images with strong and weak adversaries (larger indicates higher robustness) (a) Weak adversary (T_h with best f-1 score); (b) Strong adversary (T_h with best f-1 score); (c) Weak adversary (T_h with best precision); (d) Strong adversary (T_h with best precision).

function. Though both hyperbolic tangent and exponential functions fit the distribution better (almost perfectly for tanh), they are not stable during training.

For tanh, the gradient vanishes for most of the clusters and the loss function is unable to converge. We conjecture that the failure is due to the original distance distribution having a

high concentration of samples close to the cluster center and a flat, long tail with gradients almost equal to zero. Since tanh fits such distribution perfectly, the flat tail is causing the gradient to vanish and no subsequent updates from the backpropagation. For the exponential function, it tends to overfit when d is small (overshoots around the cluster center with small distance). Our test indicates that when $d \to 0$, $F_i(d) \to \infty$ for some clusters and this brings instability to the backpropagation process.

Property 2. Denote the quadratic parameters as $(\beta_0^i, \beta_1^i, \beta_2^i), 1 \leq \forall i \leq k$, and the largest hamming distance to the center (radius) as, $r_i = \max \mathcal{D}_h(x, c_i), x \in \mathcal{C}_i$. If $-\frac{\beta_1^i}{2\beta_0^i} > r_i + T_h$ and x is mapped to x' such that $\mathcal{L}_c$ is minimized, it is guaranteed that x' will not be returned as query results.

In Fig.4(a), $-\frac{\beta_1^i}{2\beta_0^i}$ is the distance corresponds to the minimum value of the quadratic function, which is the optimization goal. If it is larger than the sum of the retrieval threshold and the radius, using any samples from the cluster will not be able to fetch x'. This condition holds for most clusters because the samples tend to concentrate in high density around the centers. For the rest of clusters like shown in Fig.4(b), though optimization is able to reach the minimum value (around 30 in hamming distance), it has to balance the influence from other clusters as well. That is, maximizing the distance to a single cluster may accidently push the protected image into the proximity of other clusters. Our loss function is designed in a way to mitigate such effects based on the in-cluster distributions.

2.5 EVALUATION

The main goal of evaluation is to investigate: 1) effectiveness of the proposed mechanism in both white-box and black-box settings; 2) defense budget in terms of computational efforts; 3) perceptual similarity from the original image.

Dataset. We conduct the experiments on four datasets: CIFAR10 [81], Fashion-MNIST [85], ImageNet [86] and Places365 [87]. Places365 mimics the scenario when privacy is exploited from location similarity. Following [14], we randomly select 10% categories of ImageNet and Places365.

Implementation Details. We train HashNet-ResNet50 for CIFAR10/Fashion and

HashNet-ResNet152 for ImageNet/Places365. We randomly select 500 images from the test set as the *private* images to be protected and use the rest of the test set as the *attack set*. The retrieval threshold T_h is selected when the best F-1 score ($T_h = 15, 16, 12, 10$) and the best precision ($T_h = 8, 6, 8, 8$) are achieved for the four datasets, respectively.

Baselines. We compare our mechanism with a combination of baselines: *no defense* and *hamming distance maximization* [41] against the *weak adversary* and *strong adversary*. The weak adversary has some knowledge about the private image so he only queries the original category. The strong adversary enumerates through the entire test set of all categories.

Metrics. Based on the threat model, the adversary randomly picks images from the attack set to expose the private images. The mechanisms are evaluated thoroughly based on the following metrics: 1) Expected number of queries of weak adversary E_w,

$$E_w = \frac{\text{total \# attack images}}{\text{avg \# img retrieved (same class)}}.$$

2) Expected number of queries of strong adversary E_s,

$$E_s = \frac{\text{total \# attack images}}{\text{avg \# img retrieved (all class)}}.$$

These metrics quantify the efforts from the attacker. 3) Defense effort in terms of the number of iterations and computational time using a sole Nvidia GTX1070 GPU. 4) Perceptual difference between x and x' by the two metrics, a) *mean square error*, $\text{MSE} = \sum_i (x'_i - x_i)^2/N$, where x_i, x'_i are the normalized pixel values of the original and protected images and N is the dimensionality of the image; b) *Structural similarity index* that captures structural similarities to emulate human visual [88].

2.5.1 ATTACK EFFORTS

Fig. 5 compares the attack efforts of the cluster-based weighted hamming distance maximization (CWDM) with the hamming distance maximization (HDM) [41] and the no defense baseline. We can see that with "no defense", the adversary can simply extract the private images from the database within 10 queries for CIFAR10/Fashion and 100 queries for ImageNet/Places365. For weak adversary, our mechanism CWDM is a little worse or on par with HDM. This is because the hash code found by CWDM is not as far as HDM

from the original image, because CWDM has to consider distance from the rest categories to defend strong adversary. As a result, for strong adversary, CWDM effectively hardens the attack effort by 1-3 orders of magnitude than HDM, and 2-7 orders of magnitude than "no defense". E.g, for best precision, 1.1M, 58M, 0.77M and 20K number of queries are required on average, which are prohibitive for attackers with finite resources. In practice, adversaries may not know exactly what categories the private images are from, so a viable way is to explore all possible categories. CWDM successfully enlarges the attack efforts in this case.

Clustering Techniques. We assess the impact from the clustering techniques in Fig. 6 between k-means [83] and DBSCAN [84]. For k-means, we increase the number of clusters k from 15 to 30 for CIFAR10/Fashion, 150 to 300 for ImageNet and 54 to 108 for Places365; for DBSCAN, we increase EPS (maximum distance between two samples in order to be clustered) from 0.5 to 3.5. With a larger k, k-means tends to result more compact clusters with less intra-cluster distance, which leads to a general trend of higher robustness against strong adversaries. DBSCAN is more sensitive to distribution density and the value of EPS (e.g., the surge when $eps = 2.5$). k-means offers better predictability, and performance than DBSCAN by almost an order of magnitude. The main reason is because DBSCAN explicitly categorizes samples with distance larger than the EPS as outliers; CWDM does not account for these outliers during optimization thus leaving some risks, if the attack sample is identical to the outliers. An exception is Places365 where the learned hashcodes of selected categories are more concentrated than ImageNet and the outliers are less. This makes DBSCAN better than k-means on Places365.

2.5.2 DEFENSE EFFORTS

Defense efforts are measured by the hardness of finding the adversarial example in hash space. HDM only pays attention to the original category, thus should be much easier to optimize in general (which takes about 20 iterations to reach equilibrium as shown in Fig. 3(b)). On the other hand, CWDM balances the influence from all the class distributions and the quadratic regression introduce additional computation overhead. Fig. 7 traces the convergence of hamming distance to the original class, average and minimum distance to

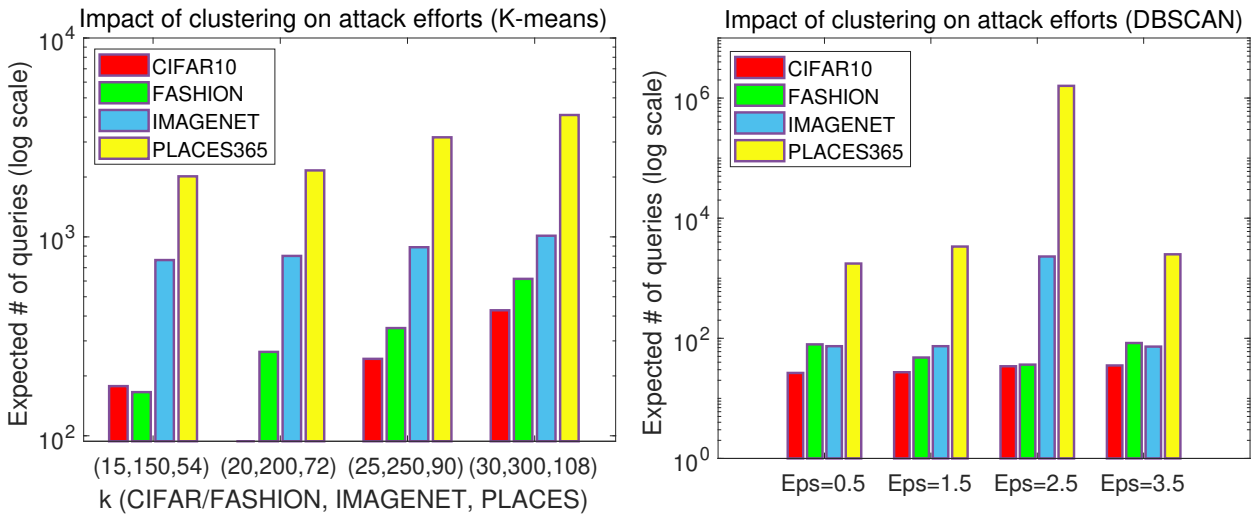


FIG. 6. Impact of clustering techniques on attack efforts (a) k-means; (b) DBSCAN.

all classes. It is observed that CWDM quickly enlarges the distance from the original class beyond the retrieval threshold (preventing retrieval from weak adversaries). "Minimum distance" tracks the class with the minimum hamming distance. It converges a little slower than the original class, because it represents those hard classes during optimization. This explains why a few samples from irrelevant classes could still fall into the retrieval threshold and lead to the success of strong adversaries. Overall, the "average distance" summarizes the convergence from all classes, and reaches a value larger than the retrieval threshold so most of the queries should return no result for protected images. Computationally, using the Nvidia GTX1070 GPU, an image takes about 4s with 100 iterations, which is quite practical in real applications. A speed-up strategy is to increase the learning rate, but at a cost of degraded success rate of generating the protected image.

2.5.3 USABILITY

Perturbation Artifacts. We compare the amount of perturbations introduced onto the private images with some examples in Fig. 8. To visualize the noise clearly, we scale up

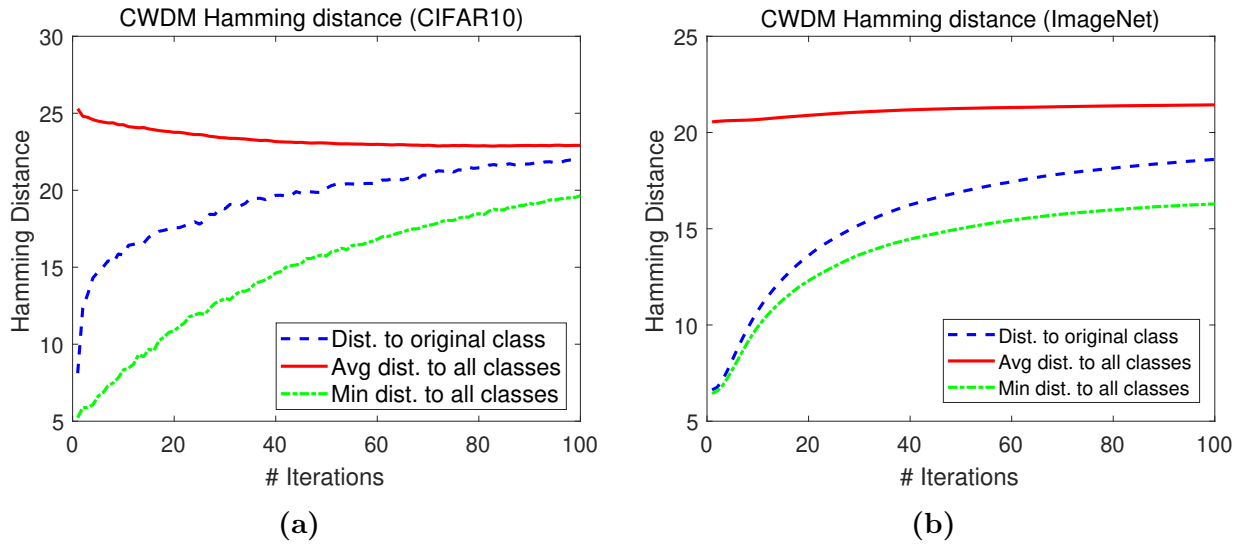


FIG. 7. Defense budget (convergence) - hamming distance from the protected image to different classes (a) CIFAR10; (b) ImageNet.

TABLE 1. Perturbations measured by MSE and SSIM

	MSE(per pixel 10^{-5})		SSIM ($[0,1]$)	
	HDM	CWDM	HDM	CWDM
CIFAR10	3.4071	2.0022	0.8971	0.9751
Fashion	2.9107	2.0757	0.8038	0.8907
ImageNet	3.0957	3.2244	0.9614	0.9611
Places365	2.3470	3.0031	0.9721	0.9628

their values by four times with a 0.5 uplift to offset any negative adversarial values. We can see that noise from CWDM concentrates more around the object, whereas HDM tends to distribute the noise across the entire image. To quantify the nuances, we further evaluate the average MSE and SSIM from the original image in Table 1. The MSE is averaged per pixel value by dividing $224 \times 224 \times 3$. SSIM falls in the range of $[0, 1]$, where 1 means the image is identical to the original one, and a less value means the distortion is higher. For

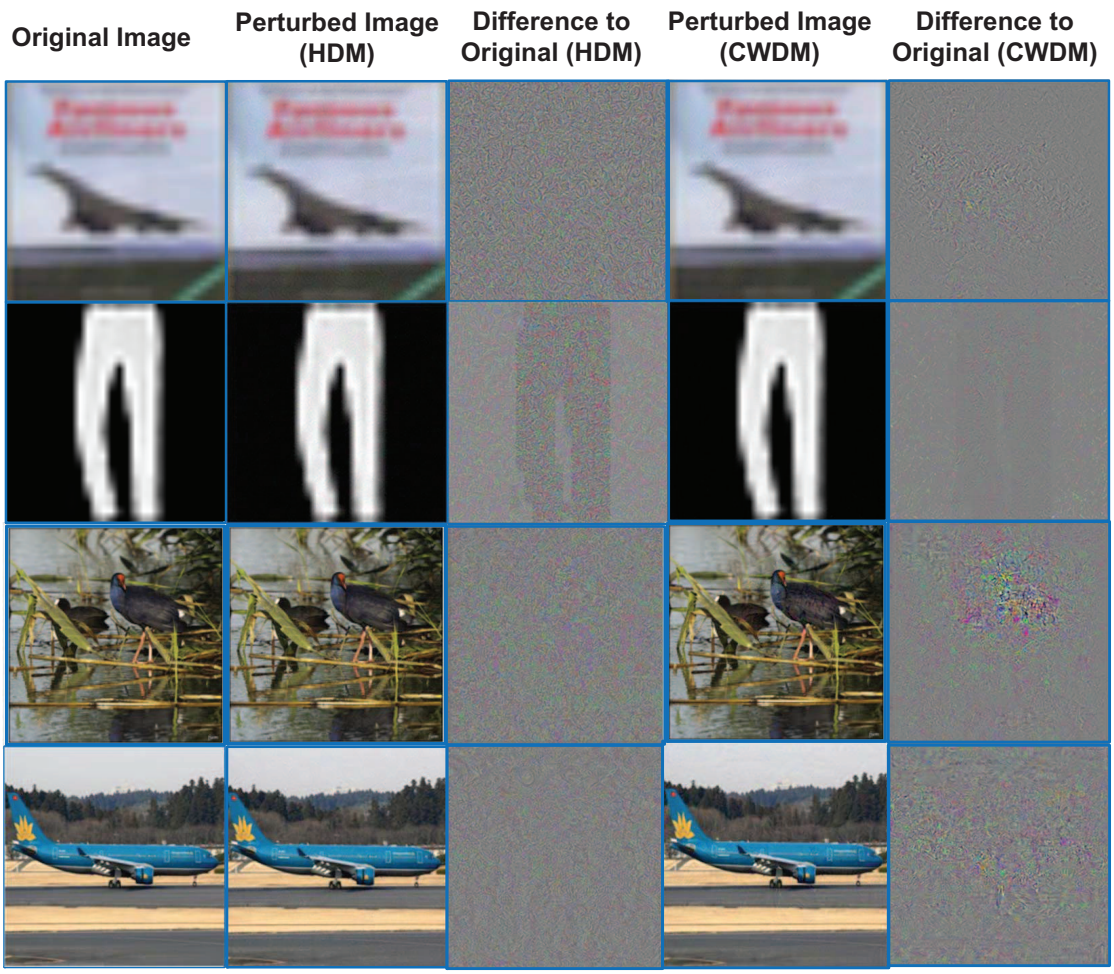


FIG. 8. Perturbed image using HDM, CWDM and their normalized difference to the original image.

CWDM, the MSE is 37% less than HDM for CIFAR10/Fashion and SSIM is almost identical to the original image by reaching a score over 0.9 on average. This is because the objective of maximized hamming distance would push the protected image further away from the original sample in hash space, whereas CWDM is more moderately looking for a subspace not far from the decision boundaries (retrieval threshold). The noise for ImageNet/Places365 is slightly higher because more scattered samples and clusters make it harder to find a subspace to hide, thereby demanding strengthened perturbations. Fortunately, the additive noise does not turn out to be significant measured by SSIM (last two columns).

TABLE 2. Evaluation of potential accuracy loss on classification tasks.

	Original		Adversarial (CWDM))	
	HashNet	Softmax	HashNet	Softmax
CIFAR10	0.870	*0.904*	0.097	**0.831**
Fashion	0.896	*0.936*	0.191	**0.934**
ImageNet	0.882	*0.908*	0.008	**0.817**
Places365	0.862	*0.853*	0.143	**0.731**

Classification Tasks. Social platforms also provide functions such as automatic photo classification, object and text recognition. These tasks typically adopt different loss functions (e.g., softmax). Since the perturbations are applied globally, we show that they do not transfer to the normal feature space, and mislead softmax classifications. Table 2 demonstrates the accuracy loss of classification tasks by applying CWDM samples to the original models. The first two columns are the baselines of the original retrieval accuracy and softmax classification respectively (100 random categories for ImageNet). The third column shows the effectiveness of CWDM that reduces retrieval accuracy below 20%. When the protected image are applied to softmax classification (the fourth column), the result does not render significant accuracy loss (compared to the 2nd column). It is interesting to see that, though the hash space perturbations have influence in the normal feature space, neural networks can treat them as random noise in general so their existence should not impact other smart applications.

2.5.4 BLACK-BOX DEFENSE

The previous subsections evaluate the scenario when users have full knowledge of the architecture and parameters of the model on the server (white-box). In practice, the proprietary model usually remains a *black box* to the users, such that they can only make their best guess of the *target model*. Empirical evidence has shown that adversarial perturbations can *transfer* across models in normal feature space [74, 75]. Here, we demonstrate transferability

of our mechanism in the hash space.

We fix the target model (server side) and generate the protected image using different source models (user side). Black-box transferability is difficult given that the source and target models usually have different decision boundaries. Strong adversary from the server side can further take advantages of any nuance in such boundaries to expose the protected image. Thus, we consider the black-box scenario to be successful, as long as there are less than n samples that can be exploited to extract a protected image. We define the *defense success rate* as the ratio between the number of protected images that have less than n retrieval results in the target model and the total number of protected images. We set $n = 100$ here since it would take the adversary considerable efforts to find these 100 images from the 50K/60K/100K/36K attack sets. We adopt different architectures on the four datasets due to the performance gap from the original HashNet, i.e., ResNet50 and architectures of less complexity exhibit much lower accuracy on the ImageNet. Thus, we set ResNet50* and ResNet152* as the target models for CIFAR10/Fashion and ImageNet/Places365 respectively, use ResNet18, ResNet34 and VGG16 as the source model for CIFAR10/Fashion, and ResNet50, ResNet101 and ResNext101 [4] for ImageNet/Places365.

Table 3 shows the success rate of transferability to target models (col.2-4) and benchmarks the results with the "no defense" baseline as the lower bound (col.5). For CIFAR10/Fashion, CWDM-protected image can successfully transfer with a rate of 40-60% within the ResNet family, possibly due to similar decision boundaries. Transferability also exists for ImageNet/Places365 about 20-40%. VGG16 has less chance to transfer. Thus, in a blackbox setting, if the user makes the correct guess of the target architecture, her images can be protected almost perfectly based on the previous white-box experiments; if the guess falls off a little, she still enjoys nearly 30-50% on average, which offers considerable improvement over "no defense" (col. 5).

2.5.5 DISCUSSION

Without knowing our defense, the attacker strives to collect a dataset that resembles the training set to extract the private, similar images. Our mechanism successfully defends

TABLE 3. Defense success rate of black-box transferability from ResNet50* and ResNet152* to different architectures (%).

ResN50*	ResN18	ResN34	VGG16	No def.
CIFAR10	44.13	40.96	22.39	9.8
Fashion	56.99	60.32	51.85	5.8
ResN152*	ResN50	ResN101	ResNext101	No def.
ImageNet	22.40	36.40	33.40	13.20
Places365	45.09	40.36	30.79	11.86

against these attackers when their attack sets are i.i.d. with the training set. Due to space limit, we will conduct more experiments to evaluate active attackers when they deviate from the i.i.d settings. The accuracy from the original HashNet also affects the effectiveness of the defense. For example, ResNet18-50 do not achieve satisfied accuracy to map semantically similar images into identical and compact hash codes on ImageNet/Places365. The learned codes are more scattered, thereby squeezing the optimization space to successfully perturb the image and prevent from retrieval. Since the service providers typically finetune their models, we expect the proposed mechanism to be effective against the production models with high accuracy.

2.6 CHAPTER SUMMARY

In this chapter, we describe efforts to protect private images from malicious deep image retrieval. We first identify and experimentally validate the effectiveness of using adversarial perturbations as a defense in the hash space. By showing vulnerabilities against strong adversaries, we propose a new mechanism to find an alternative subspace that maximizes the weighted hamming distance to all the classes. We evaluate the efforts from both the attack and defense perspectives, usability, and black-box transferability with extensive experimental results.

CHAPTER 3

YOU SEE WHAT I WANT YOU TO SEE: EXPLORING TARGETED BLACK-BOX TRANSFERABILITY ATTACK FOR HASH-BASED IMAGE RETRIEVAL SYSTEMS

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 chapter, 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.

3.1 INTRODUCTION

With the exponential growth of visual content on the Internet, deep learning to hash (*deep hashing*) [13, 14, 16] 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 [27], Bing [28]), social networks (Pinterest [56], e-commerce (Taobao [30])

and fashion designers([85]), 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 [41, 43, 89, 90], 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 [91, 92]. 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 [92].

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 [93] and caching user queries [64]. 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 [41, 89, 90]. E.g., [41] 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 [92], 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 [43] 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 chapter, we explore and improve targeted transferable attack in deep hashing. Similar to susceptible classes in classification [94], 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 transferable 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 [42], and $1.5\times$ compared to the diversity techniques [43]. We also demonstrate case studies of crafting out-of-distribution images to target normal queries with

high successful rates.

3.2 BACKGROUND AND RELATED WORK

In this section, we summarily introduce previous works on black-box adversarial attacks and deep learning to hash.

3.2.1 BLACK-BOX ADVERSARIAL ATTACKS

Fast Gradient Sign Method (FGSM) [20] and Projected Gradient Descent (PGD) [42] 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 [92], 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 [43, 95–97]. Therefore, enhancing diversity has been taken at different levels of input image [43, 95], model ensemble [96] and gradient trajectory [97]. Rather than using a single image, in [43], 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 [95]. Both gradient ascent and descent are combined for more diversity [97]. Another thread of works focus on the feature level to improve transferability [98, 99]. 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 [98, 99]. Our work taps into this line to enhance black-box transferability

for image retrieval systems and will compare with these techniques in Sec. 3.6.

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 [100, 101] or decision (label-only) [91, 102, 103]. E.g., [101] utilizes the changes from the softmax output to estimate the gradients. [100] adopts the natural evolutionary strategy to estimate the gradient under the search distribution. [103] only relies on the final decision of the model, which iteratively draws random distribution from a proposed distribution while staying adversarial and [91] 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.

3.2.2 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 [13, 14, 16]. 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 approximation for the non-smooth sign function is performed, e.g., HashNet [14] adopts the hyperbolic tangent function, $sgn(z) = \lim_{\beta \rightarrow \infty} \tanh(\beta z)$, by tuning β; the function converges to the sign function when $\beta \rightarrow \infty$. 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 [41, 89, 104], 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. [41] 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. [89] follows with a similar optimization objective to design adversarial queries. [104] designs a new optimization problem to prevent private images in the database from queried by curious third parties. [90] 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.

3.3 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 [13, 14].

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],

$$\min_{x', x_t \in \mathcal{C}_t} D_h(x', x_t), \tag{10}$$

$$s.t. ||x - x'||_\infty < \eta. \tag{11}$$

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.

3.3.1 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 [20], PGD [42], Iterative FGSM with Diversity (DI) [43] 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. 9). Compared with softmax, which delivers around 10% black-box success, adversarial images rarely transfer with deep hashing: the overall success rate is

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

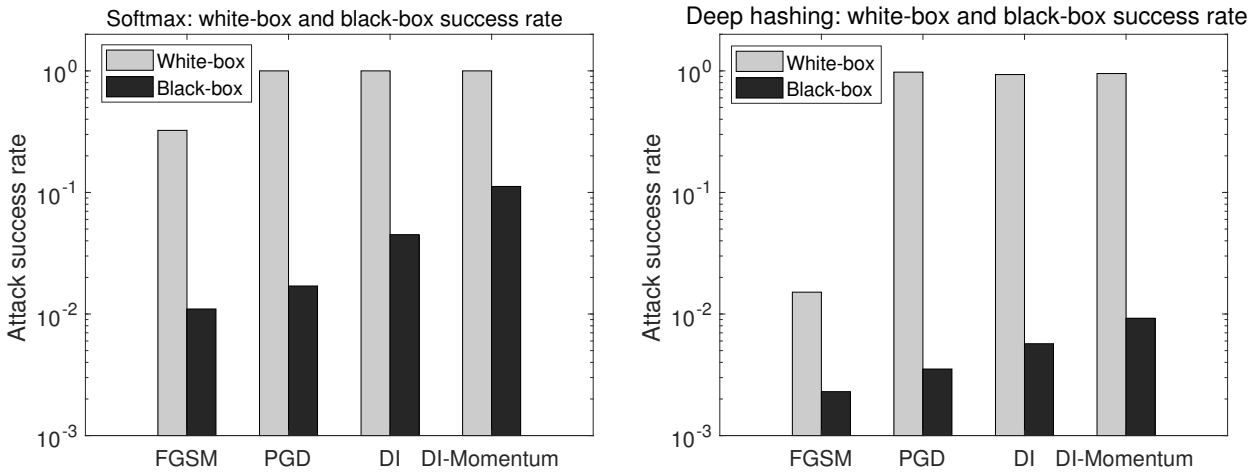
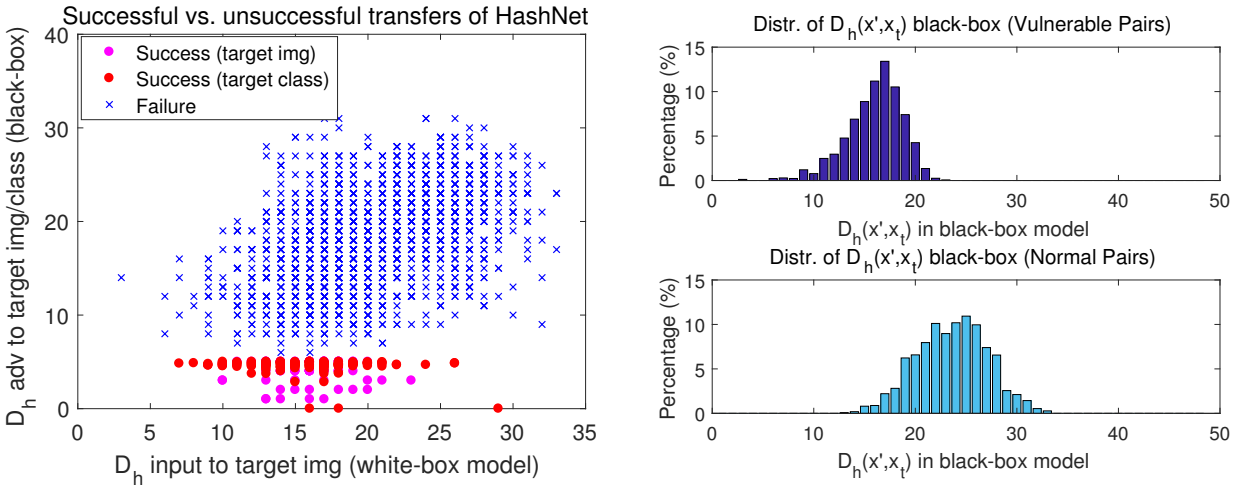


FIG. 9. Targeted white-box and black-box attack success rate (a) Softmax classification; (b) Deep hashing. See summary in *Observation 1.*

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 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 $(x, x_t) \notin \mathcal{V}$.



(a) **(b)**

FIG. 10. Illustration of vulnerable pairs. (a) Relations of hamming distance between input and target images in white-box source model, and adversarial input to targeted images (class) in black-box model; (b) Distribution of hamming distance from adversarial to target image in black-box model of vulnerable and normal pairs.

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.10(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.10(a), x-axis), though a large number of unsuccessful transfers are also found for the same range. Fig.10(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.11. 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

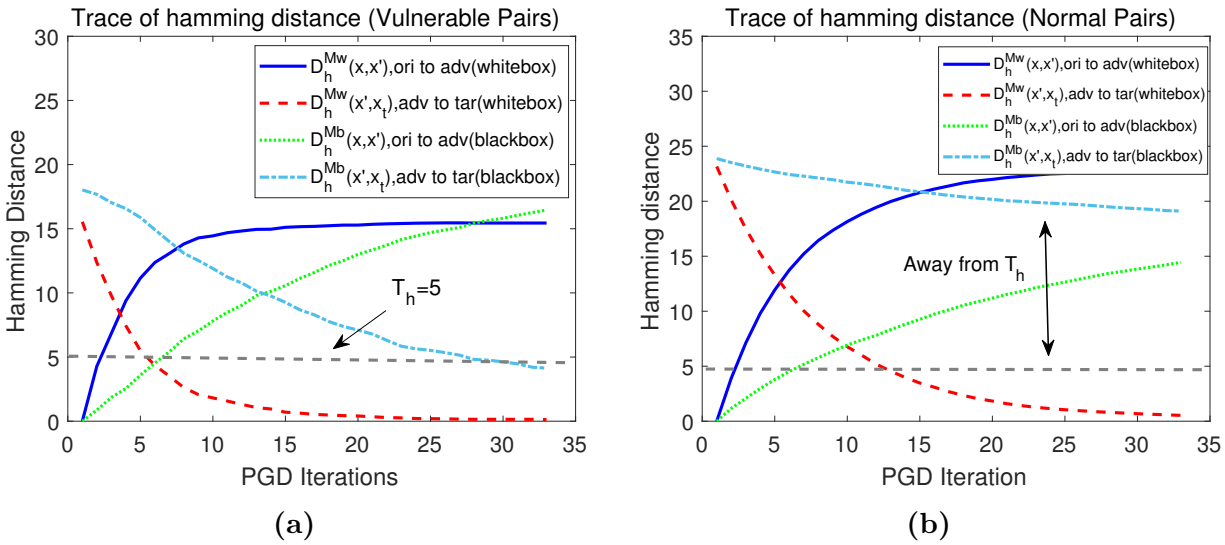


FIG. 11. Trace of hamming vs. PGD iterations. (a) vulnerable pairs; (b) normal pairs.

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 [92], which also brings higher perturbation and risks of violating the η-bound. Recall from Fig.10(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.

3.4 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 [19, 20, 105] and notably difficult when it comes to quantitative analysis, e.g., some literatures employ intrinsic dimensionality [50] and orthogonal adversarial directions [75]. Only a few connects adversarial subspace with transferability: [75] 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.

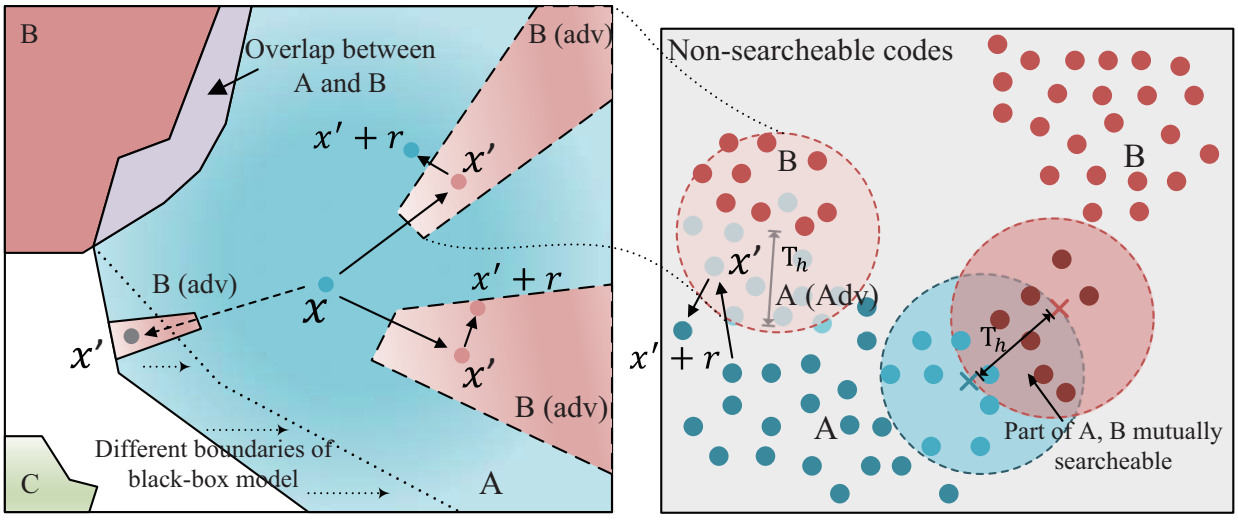
We propose an efficient method to utilize random noise as a proxy, and feedbacks from the white-box model to predict transferability. Random noise injection finds deep roots in the defense literatures to certify classifier robustness, e.g., learning a smoothed classifier that returns the most probable class under Gaussian noise [106, 107]. We draw a close connection to adversarial examples, which are found to form a cone-shape structure surrounded by natural classes [108,109]. We conjecture their presence in deep hashing has a similar geometry sketched in Fig.12(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. 12(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.

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 [75]. 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 [109]. Neural nets are known to be robust to random noises, but surprisingly sensitive to small, purposely crafted



(a) Adv in the pixel space

(b) Adv in the hash space

FIG. 12. Illustration of adversarial examples (a) image pixel space (b) hash space.

perturbations. According to [110], 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 [109] (see Fig.12(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

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**

  $m \leftarrow \frac{\text{min} + \text{max}}{2}$, $r \sim \mathcal{N}(0, \sigma^2 I)$, $\sigma = 1/3$.

  $r \leftarrow r - (r \cdot \frac{\epsilon}{|\epsilon|})\frac{\epsilon}{|\epsilon|}$, $r \leftarrow \frac{r}{||r||_\infty} \cdot m$.

  $x'' \leftarrow x' + r$, 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).

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.

TABLE 4. White-box attack success rate when $\eta_\infty = 16, 32$ and $R \in [0, 64]$.

	Noise R	0	4	8	16	32	64
$\eta_\infty = 16$	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
$\eta_\infty = 32$	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

We validate *Property 1* by adopting the algorithm to estimate the adversarial subspaces and assess whether the results align with Fig. 9. Table 4 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.9. Thus, random noise can be used as an effective measure to estimate adversarial

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 all $\sigma \in \mathcal{R}$ **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 $x' \leftarrow \arg\max_{x' \in \mathcal{X}'} \mathcal{M}$.

sphere and predict black-box transferability.

3.5 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 $\mathcal{R}$),

$$\max \mathbb{E}_{\sigma \in \mathcal{R}} \left[f_{M_b}\big(x + \epsilon^*(\sigma)\big) \right], \tag{12}$$

where

$$\epsilon^*(\sigma) = \underset{\epsilon = ||x' - x||_\infty < \eta}{\arg\min} D_h(x', x_t), \tag{13}$$

s.t.

$$\mathbb{E}_{r \sim \mathcal{N}(0, \sigma^2 I)} \Big[D_h(x' + r, x_t) \Big] \leq T_h, \tag{14}$$

The inner optimization (13) aims to find the optimal perturbation that minimizes the hamming distance between x' and target x_t. (14) 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 (5) 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,

$$\max_{\lambda}\min_{x'}\Big(D_h(x', x_t) + \lambda^\top g(x')\Big). \tag{15}$$

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

$$\begin{aligned} x'_k &= proj_{x',\epsilon}\Big(x'_{k-1} + \alpha\cdot sgn\big(\nabla_{x'}\mathcal{L}(x'_{k-1}, \lambda_{k-1}, r)\big)\Big) \\ \lambda_k &= \lambda_{k-1} + \alpha\frac{\partial\mathcal{L}(x'_k, \lambda_{k-1}, r)}{\partial\lambda} \end{aligned} \tag{16}$$

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,

$$\nabla_{x'} g(x') \approx \nabla_{x'}(\frac{1}{M}\sum_{i=1}^{M} D_h(x'+r_i, x_t)) \tag{17}$$

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 [111]; 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.3.6.

3.6 EVALUATION

Experimental Setup. We conduct the experiments on ImageNet. Following [14], 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 [3], ResNext101 [4], SeResNet50 [112], ResNet34 and DenseNet161 [113]. 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 [20], PGD [42], Feature-level Activation Attack(AA) [99], Diversity Inputs (DI) and Diversity Inputs with Momentum (DI-Mom) [43] on targeted attacks[2].

3.6.1 BLACK-BOX TRANSFERABILITY

We first demonstrate the attack success rate in Table 5 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 [43] at 11.33/11.61%, PGD [42] at 11.05% and AA [99] at 9.22%. For normal pairs, NAG generates an average of 1.82% success compared

[2]We do not compare with the *Universal Adversarial Perturbation* (UAP) attack here [114, 115], since it is designed for untargeted attack.

TABLE 5. Attack success rates (%) of vulnerable/normal pairs. The diagonal blocks indicate the white-box success rates.

		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

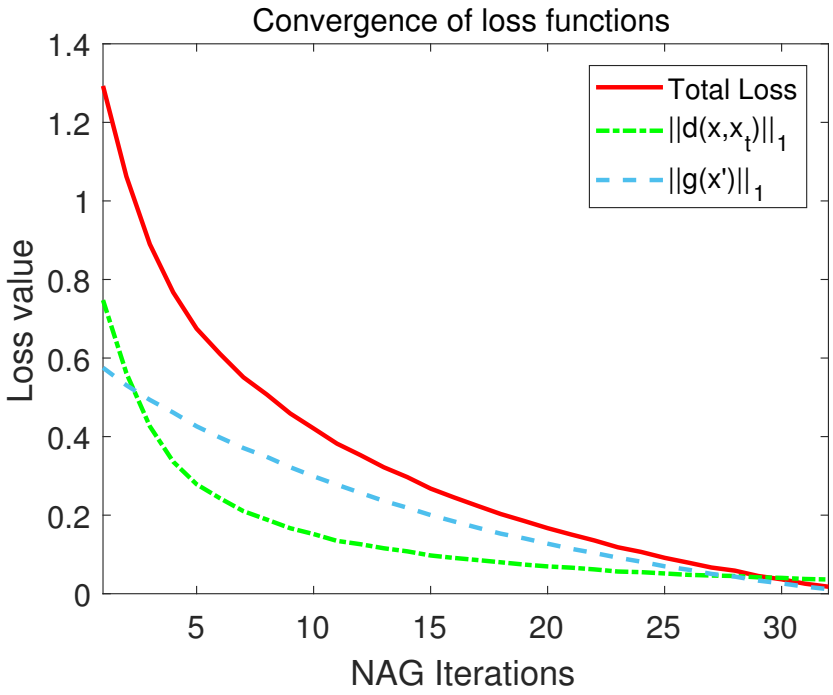
with 0.516%, 0.546%, 0.87% and 0.22% of the four benchmarks respectively. Some model combinations achieve phenomenal improvements such as ResNet152 $\rightarrow$ ResNet101, which yields almost $2-3\times$ performance boost.

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 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 [43, 92] because the cosine similarity of gradient directions is much higher than that of a different family [92]. 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.

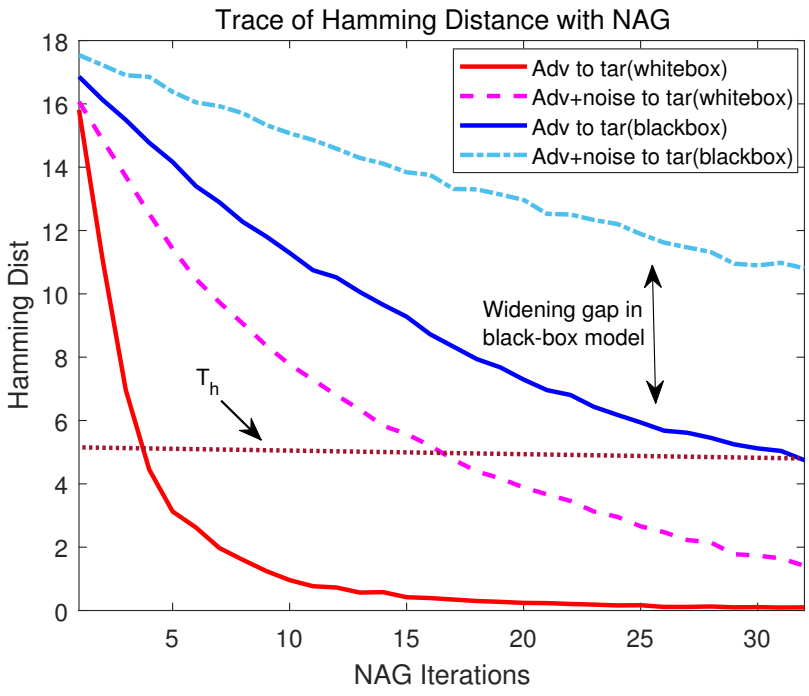
Convergence of Adversarial Loss. To see how NAG meets the objectives, we pick a representative model pair and trace the loss convergence by averaging the generation of all the adversarial examples shown in Fig.13(a). For clarity, we plot the normalized $||d(x', x_t)||_1$, $||g(x')||_1$ and the total loss in Eq. (15), 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 (6) 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.13(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 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 [96].

3.6.2 CASE STUDIES OF SIMULATING REAL-WORLD ATTACKS

We also conduct some case studies to simulate the real-world attacks that the adversarial



(a)



(b)

FIG. 13. 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.

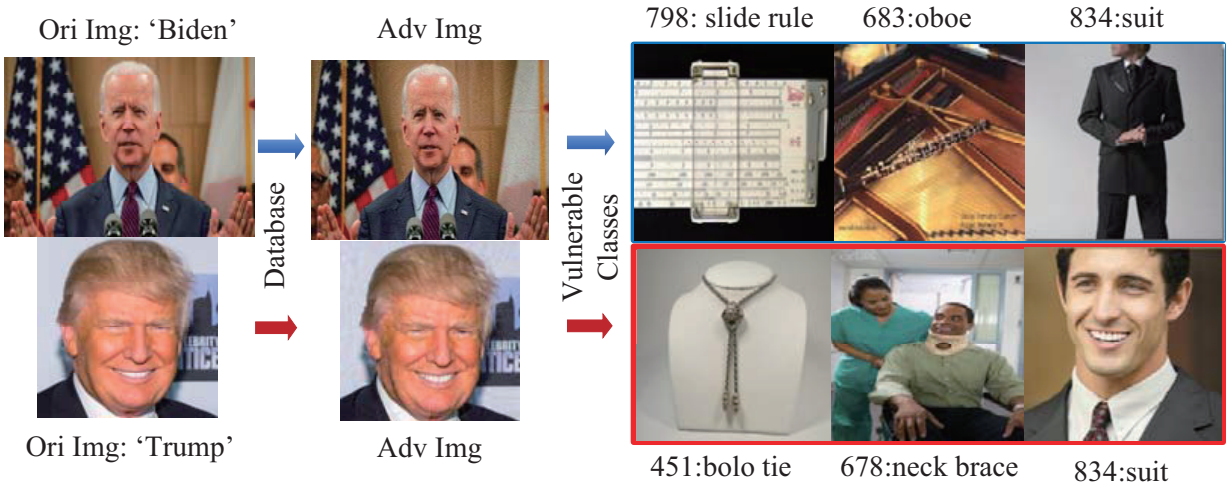


FIG. 14. Case study of retrieving the adversarially-crafted, out-of-distribution images of presidential candidates from normal queries.

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.14 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.

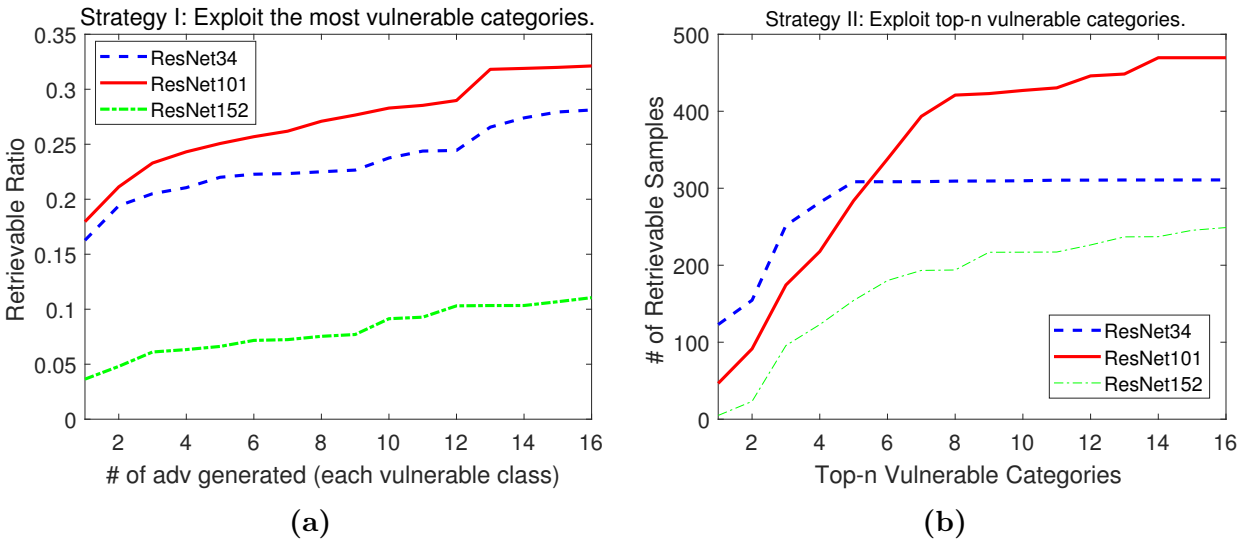
Case Study II: Advertising for Free. To evaluate at a larger scale, we conduct another case study to utilize the advertisement dataset [116] with a similar objective to make those out-of-distribution advertisement retrievable from user's normal queries. Here, we assume the attacker is resource-limited 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.15 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.15(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.15(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.

3.7 ADDITIONAL RESULTS

3.7.1 PERFORMANCE ON SOFTMAX CLASSIFICATION

We conduct additional experiments to evaluate the proposed mechanism on softmax classification tasks using the same setting of ImageNet and compare with the aforementioned benchmarks [42, 43]. Since softmax returns the arg max-label as the classification result, we



(a) **(b)**

FIG. 15. Evaluation of retrievable ratio/number of normal queries (a) S(I): Exploit the most vulnerable categories. (b) S(II): Exploit top-n categories.

revise the objective of NAG slightly,

$$\epsilon^*(\sigma) = \underset{||x'-x||_\infty<\eta}{\arg\min} \Big(\mathbb{E}_{r\sim\mathcal{N}(0,\sigma^2 I)} \left(\mathcal{L}_{CE}(x', x_t) + \lambda ||d(x'+r, x')||_1 \right) \Big) \tag{18}$$

Given the target label x_t, the first term is the cross-entropy loss of targeting x' at x_t. The second term is the l_1 distance between the output of $x'+r$ and x', that the goal is to keep noise-injected $x'+r$ close to x' so that it remains adversarial [109]. λ is a scaling parameter to balance the two losses.

We randomly select one from the 100 categories as the target class (other than the original source class) and show the transferability rates in Table 6. We add Inception_v3 into the model combination because of improved accuracy on the softmax classification task. We set $\lambda = 50$ to weigh more on the second loss and the noise level is set to 32. From Table 6, the average black-box transferability is 0.93%, 1.11%, 4.19% and 4.69% for PGD, DI, DI-Mom and NAG, i.e., both DI-Mom and NAG offer 4× targeted transferability compared to PGD and DI. Note that DI/DI-Mom were originally proposed and tested on softmax classification, which defeat the winners of NIPS 2017 adversarial competition by a large margin. NAG is slightly better or on par with DI-Mom in softmax.

Discussion. We notice some interesting phenomenons during our experimentation on the hashing and classification networks. The first one is their response to the injected random noise: softmax classification is more robust to random noise, such that: a) the convergence of (18) is much faster than hashing; b) for the same level of random noise, NAG is more effective in deep hashing than softmax (our mechanism is comparable to DI-Mom in softmax, but with more than 15% improvements in deep hashing). In contrast to softmax, hashing learns similarity relations from pairwise inputs. The difference could be investigated along the direction of attentive regions/feature structures learned by deep hashing and softmax. We also notice that the variation of loss curvature in deep hashing is higher than softmax during training, which indicates that softmax may have a smaller Lipchitiz constant overall (sensitivity of the network to perturbations). This may partially illustrate why random noise is less effective on softmax classification. We also notice that the distributions of the perturbation generated by NAG visually retain a Gaussian distribution. Since the solutions are mostly found among the vertices of the l_∞ ball, the rest of the additive Gaussian noise is preserved. The fundamental questions of how much random noise would help learn a randomized smoothing classifier [107], improve black-box transferability (compared to the competitive input diversity methods [43]), and resolve the relations between adversarial perturbations and random noise in different learning tasks (softmax/metric/hashing) are worth future research efforts.

3.7.2 MORE EXAMPLES FROM "ADVERTISING FOR FREE"

Figs.16 and 17 visualize the two strategies to advertise for free. Recall that in Strategy I, we randomly pick a fixed number of images from the most vulnerable category and generate the corresponding adversarial images (one of them is depicted with the perturbation). It is seen that the vulnerable categories do not appear to be purely random - some of them have obvious semantic relations, e.g., the advertisement of "beer/soda" (third row) is closest to "lotion", because most lotion images contain bottle(s). This allows NAG to realize black-box transfer attacks more easily. Similarly, the fourth advertisement of beverage features a dog in the image and "Welsh Springer" came as the vulnerable category (not directly retrievable

TABLE 6. Targeted attack success rates of softmax classification (%): The diagonal blocks indicate the white-box success rates.

		Res34	Res50	Res101	Res152	Next101	SeRes50	Inc_v3	Dense161
Res34	PGD	**100.0**	1.3	1.1	1.0	1.2	1.0	0.9	0.3
	DI	99.9	1.3	1.2	1.1	1.6	0.5	0.5	0.4
	DI-M	**100.0**	4.5	3.7	4.5	4.3	2.9	2.5	2.7
	NAG	98.5	**7.8**	**6.2**	**6.5**	**5.6**	**3.3**	**5.3**	**4.2**
Res50	PGD	1.2	**100.0**	1.7	1.3	1.4	0.9	0.3	0.3
	DI	1.6	**100.0**	1.8	1.7	1.6	1.1	1.0	0.6
	DI-M	**6.5**	99.8	6.0	7.1	5.0	**4.1**	2.6	4.1
	NAG	5.0	99.7	**7.2**	**8.8**	**6.7**	**4.1**	**3.2**	**4.3**
Res101	PGD	1.1	2.1	**100.0**	1.8	1.6	0.9	0.7	0.6
	DI	1.2	2.5	**100.0**	2.6	2.1	0.7	1.2	0.8
	DI-M	**7.4**	**8.8**	**100.0**	**11.9**	6.3	3.7	2.6	**4.0**
	NAG	4.9	8.7	99.6	9.1	**7.5**	**3.8**	**3.6**	3.5
Res152	PGD	1.5	2.3	3.6	**100.0**	1.9	0.9	0.9	0.5
	DI	1.2	3.2	3.8	**100.0**	2.2	0.9	1.0	1.0
	DI-M	**8.0**	8.9	8.0	99.9	5.7	4.9	3.1	4.9
	NAG	7.3	**11.9**	**10.4**	99.8	**9.9**	**5.3**	**4.9**	**6.1**
Next101	PGD	0.8	1.1	1.5	1.1	**100.0**	0.7	0.7	0.3
	DI	1.5	1.5	2.0	1.3	99.9	0.3	0.6	1.0
	DI-M	6.2	6.3	4.4	6.5	**100.0**	**3.3**	2.6	4.2
	NAG	**6.4**	**9.5**	**7.9**	**9.0**	**100.0**	2.9	**3.4**	**7.6**
SeRes50	PGD	0.4	0.4	0.5	0.8	0.7	**100.0**	0.7	0.3
	DI	0.4	0.9	0.7	0.5	0.5	**100.0**	0.9	0.6
	DI-M	3.1	3.0	1.9	**3.0**	**2.8**	**100.0**	1.9	2.5
	NAG	**3.6**	**4.2**	**2.9**	2.8	2.6	99.6	**3.6**	**2.7**
Inc_v3	PGD	0.5	0.3	0.5	0.4	0.4	0.5	**100.0**	0.3
	DI	0.8	0.3	0.4	0.3	0.6	0.2	99.9	0.4
	DI-M	**2.9**	**2.6**	**2.1**	**2.5**	**1.9**	**2.0**	**100.0**	2.0
	NAG	2.1	2.2	1.9	2.0	1.8	**2.0**	91.4	**2.2**
Dense161	PGD	0.8	0.8	0.6	0.8	0.9	0.6	0.4	**100.0**
	DI	0.8	0.8	1.2	0.8	0.8	0.8	0.9	99.9
	DI-M	**4.3**	**2.8**	**2.1**	**3.2**	**2.3**	**1.9**	**2.1**	**100.0**
	NAG	0.6	0.6	1.0	0.6	0.6	0.5	0.8	90.7

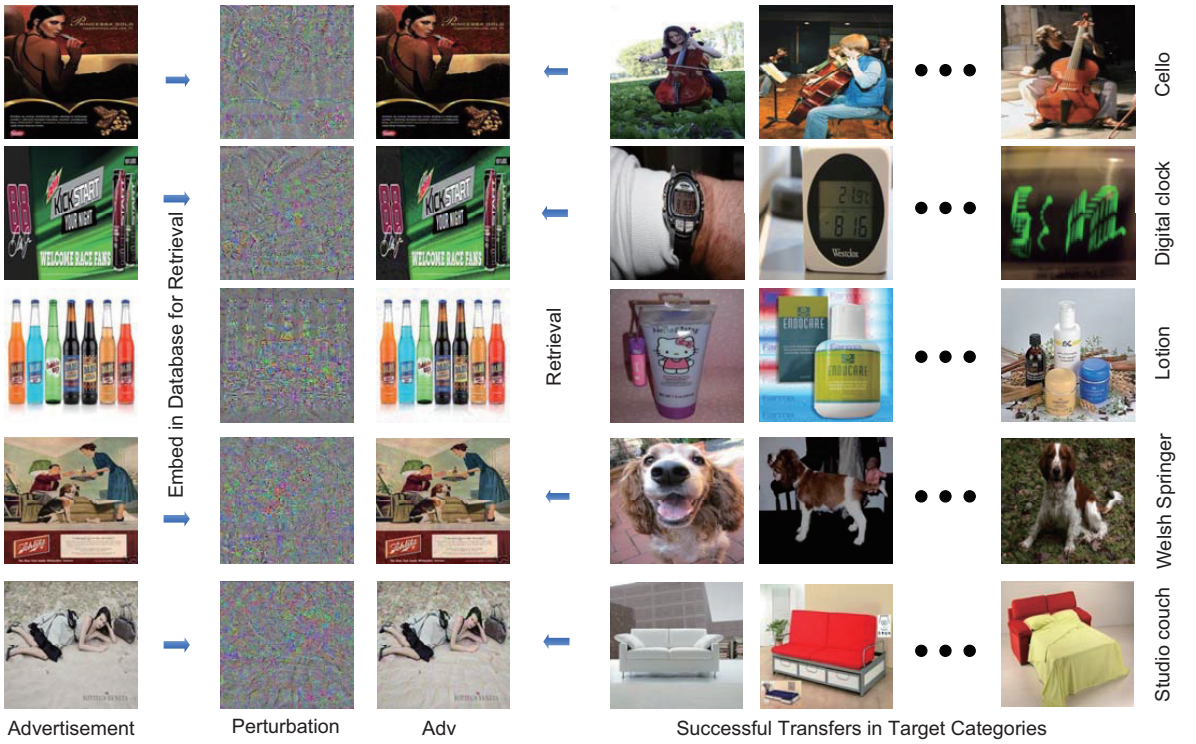


FIG. 16. Visualization of Strategy I: exploit the most vulnerable categories. For each advertisement image, randomly pick a fixed number of images from the most vulnerable category and generate corresponding adversarial examples.

within T_h). It is interesting to see that the last advertisement of handbag includes the posture of lying on the side and "studio couch" emerges as the vulnerable class, which also has some connections. Strategy II exploits the top-n most vulnerable categories and selects one image from each category to generate adversarial examples. Except a few vulnerable categories with semantic relations, the rest seem quite random, e.g., one may ask why leopard/clog is in any way similar to the last advertisement of an Android phone. In our experiment, we found that these categories actually have lower chance to succeed.

3.8 CHAPTER SUMMARY

In this chapter, we study the targeted black-box transferability attack in deep hashing. We connect transferability to the adversarial subspace and propose an implicit technique to

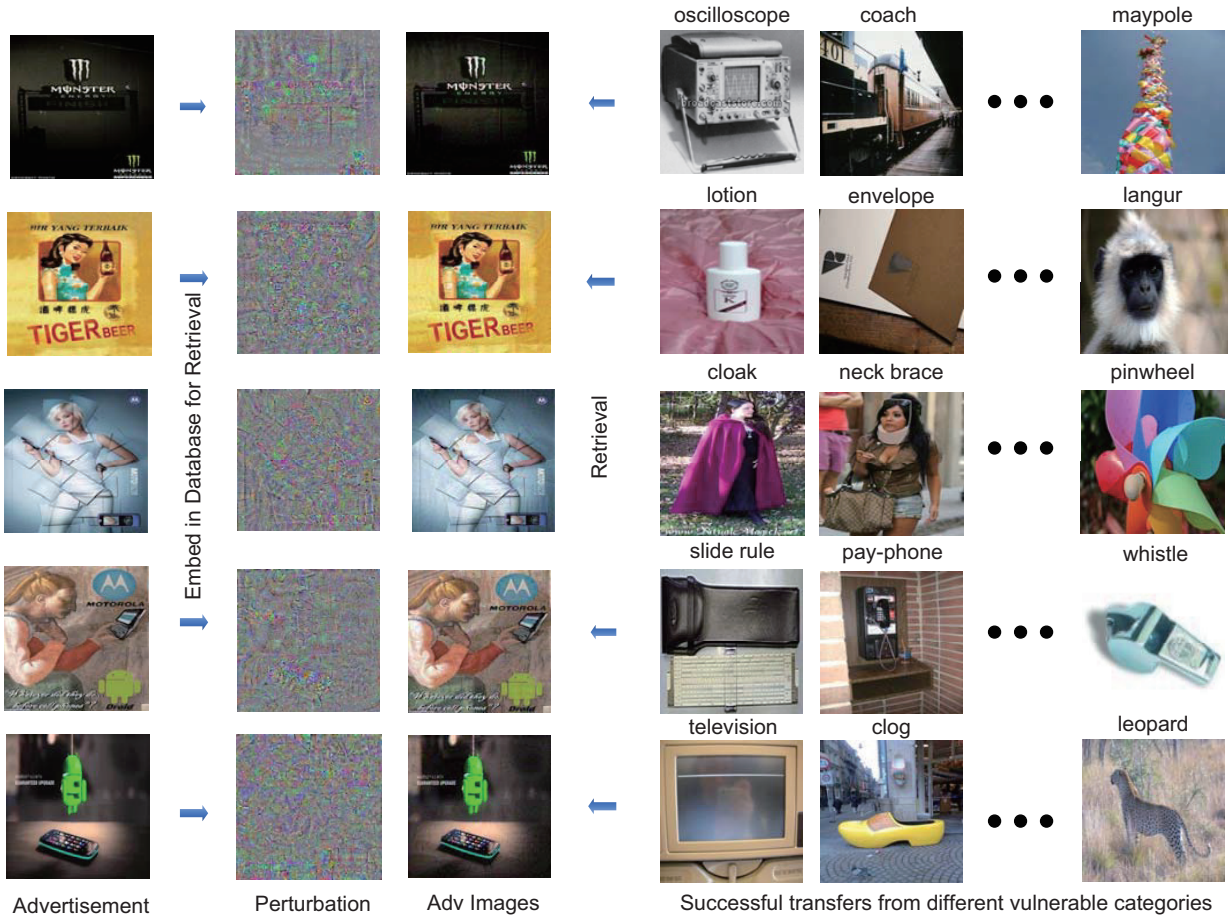


FIG. 17. Visualization of Strategy II: exploit top-n vulnerable categories. For each advertisement images, pick the most n vulnerable categories according to the hamming distance and generate an adversarial example for each category.

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.

CHAPTER 4

FAST AND EFFICIENT DETECTION OF ADVERSARIAL EXAMPLES IN DEEP HASHING BASED IMAGE RETRIEVAL

The vulnerability in the deep learning supply chain has imposed new challenges to image retrieval systems in the downstream. As deep hashing is gaining popularity, a handful of attacks are recently proposed to disrupt normal image retrieval. Unfortunately, the defense strategies in softmax classification are not readily available to be applied in the image retrieval domain. In this chapter, we propose an efficient detection scheme to identify unique adversarial behaviors in the hamming space. In particular, we design three criterions from different perspectives of hamming distance, quantization loss and denoising to defend against both untargeted and targeted attacks, which also effectively limit the action space of white-box attackers. The extensive experiments on four datasets demonstrate more than 20% improvements of detection rates with minimum computational overhead for real-time image queries.

4.1 INTRODUCTION

Powered by neural networks, deep hashing enables image retrieval at a large scale [13–18]. By representing high-dimensional images with compact binary codes, retrieval becomes an efficient similarity computation of Hamming distance. Google [27], Bing [28], Pinterest [56], Taobao [60] have all incorporated image query as part of their products. Despite of its great success, deep hashing also inherits the vulnerabilities from neural networks [19] with new attack vectors and effects. By introducing adversarial perturbations either on the query or database images, normal requests can be diverted to an irrelevant (*untargeted attack*) [41] or specific category (*targeted attack*) [44–46], e.g., turning a query of "husky dog" into retrieving a branded "dog food" so the attacker can advertise their products for free.

With a handful of efforts on the attack side [41, 44–46], deep hashing still falls short to

defend against adversarial examples in the hamming space. *Adversarial training* and *detection* are the two common defenses in softmax classification. Yet, adversarial training has to deal with the non-trivial trade-off between robustness and accuracy [117]. According to our implementation (see additional results), finding the min-max saddle points becomes even more difficult under the hash function, which makes them suffer from a large accuracy loss. On the other hand, detection aims to unveil the adversarial behaviors on different levels of raw pixel [47, 48], feature distribution [48–50], softmax probabilities [51] and frequency components [52] in a supervised [53] or unsupervised manner [54]. Based on the prior knowledge of attack methods, supervised detection trains a classifier to distinguish the adversarial images, but is hard to extrapolate to the unknown attacks. Thus, in this chapter, we pursue the direction of unsupervised anomaly detection. Different from softmax classification on a closed set of class probabilities, deep hashing maps similar/dissimilar images into binary codes in an open Hamming space. Thus, the focus of our work is to tap into the unique adversarial behaviors in deep hashing to detect both untargeted and targeted attacks.

Starting from the untargeted attacks [41], we first theoretically deduce the hamming distance distribution from the adversarial image to other categories, which asymptotically approaches a Gaussian distribution. For targeted attacks, the adversarial objective [44, 45] aims to produce the exact hash code of a specific category(image), that also brings the quantization loss close to zero as a side effect. Thus, we first develop two thresholding methods that take hamming distance and quantization loss as the proxies for detection. Then we combine the two criterions with a denoising-based detection to measure the disagreement between an input and its denoised transformation. We demonstrate that this combination can successfully defend both gray-box and white-box attacks. The overall framework is shown in Fig. 18.

The main contributions of this chapter are summarized below. To the best of our knowledge, this is the first effort to defend against adversarial attacks in deep hashing based image retrieval. We propose three criterions to reveal adversarial behaviors of targeted and untargeted attacks in the hamming space and demonstrate their complementing relations to detect white-box attackers. The extensive experiments on CIFAR-10, ImageNet, MS-COCO and

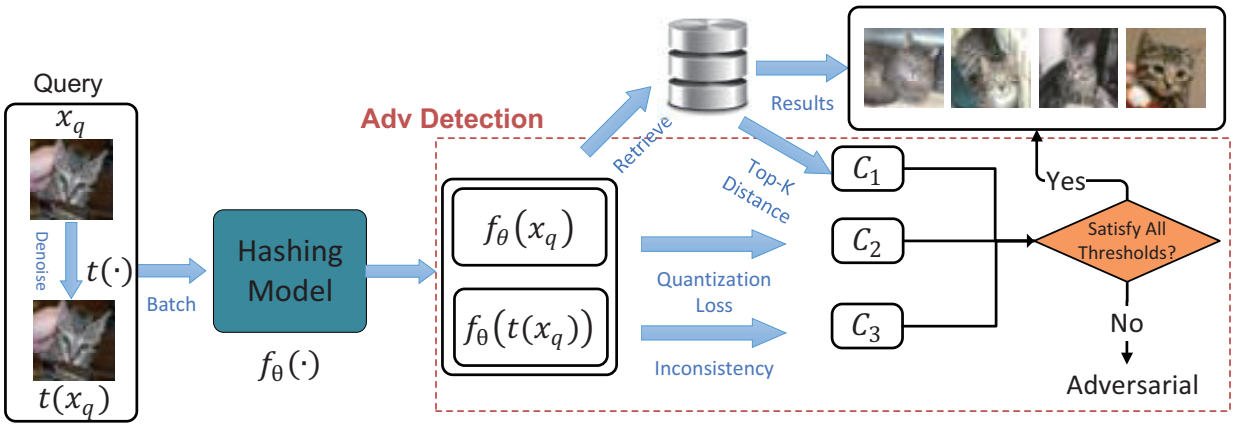


FIG. 18. The proposed detection framework: highlighted by the dash lines.

NUSWIDE datasets show that the proposed method surpasses the state-of-the-art defenses by more than 20% in detection rates with negligible computational overhead for real-time image queries.

4.2 PRELIMINARY

This section illustrates the fundamentals of deep hashing and adversarial attacks.

4.2.1 DEEP HASHING

Given a dataset of N samples $X = \{x_1, x_2, \ldots, x_N\}$, $x_i \in \mathbb{R}^D$ and their corresponding labels $Y = \{y_1, y_2, \ldots, y_N\}$, $y_i \in \mathbb{R}^C$, where x_i is the i-th sample and $y_{c,i} = 1$ if the i-th image is associated with class c. Deep hashing learns a function $f_\theta(x)$ that maps the input image x into a K-bit binary code $h(x)$ via a sign operation,

$$h(x) = sign(f_\theta(x)) \in \{-1, +1\}^K, \tag{19}$$

where θ are the parameters learned from minimizing the weighted combination of the similarity loss L_S and quantization loss L_Q [13–18],

$$\theta = \arg\min_\theta L_S + \lambda L_Q. \tag{20}$$

L_S represents the hamming distance $D_h(h(x_i), h(x_j))$ between two images x_i and x_j with

their similarity $s(y_i, y_j)$,

$$s(y_i, y_j) = \begin{cases} +1, & \text{if } y_i y_j^T > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{21}$$

L_Q is the quantization loss to minimize the difference between the continuous output of $f_\theta(x)$ and its binary code $h(x)$. The objective is to minimize the hamming distance $D_h(h(x_i), h(x_j))$ between two samples x_i and x_j when they are similar, maximize the hamming distance when they are dissimilar, and meanwhile, represent the continuous $f_\theta(x)$ as binary codes. Both $D_h(h_1, h_2)$ and $h(x)$ are non-differentiable regarding their inputs. A common technique is to use the differentiable form of $D_h(h_1, h_2)$ noted as $\frac{1}{2}(K - {h_1}^T h_2)$ during back propagation, where h_1, h_2 are the continuous floating point representation in $[-1, +1]$, and the binary hash codes $h(x)$ are represented by the continuous output of $f_\theta(x)$. The gap between such continuous and binary representations is considered as the quantization loss L_Q, which is minimized in Eq. (20).

Deep hashing consists of two main components, a *database* and a *model*. The database stores the images and their pre-computed hash codes. Given a query image x with hash code $h(x)$, the system returns the top-k images from the database which are $h(x)$'s k-nearest neighbors determined by hamming distance. The retrieval performance is calculated by the mean average precision (mAP), which is the ratio of images similar to x. In this chapter, we base the hashing framework on the state-of-the-art method called Central Similarity Quantization (CSQ) [18]. CSQ pre-determines the optimal hash codes based on the Hadamard matrix and randomly selects a set of hash codes with sufficient distances from each other as the hash centers from the Hadamard matrix (or from a random binary matrix if the Hadamard matrix is not available). Since different hashing techniques share the general objective of Eq. (20), our defense applies to other techniques as well [13–17].

4.2.2 ADVERSARIAL ATTACKS

Untargeted Attack [41] finds an adversarial image x' by maximizing the hamming distance between the hash codes of adversarial examples and original images, subject to the

L_∞ bound of ϵ.

$$\begin{aligned} &\max_{x'} D_h\big(h(x'), h(x)\big) \\ &\text{s.t. } ||x - x'||_\infty \leq \epsilon \end{aligned} \tag{22}$$

It works effectively to reduce the mAP by pushing the original image towards the furthest hamming distance in the hash space.

Targeted Attack [44–46] attempts to minimize the hamming distance from x' to the targeted hash code h_t of a specific category,

$$\begin{aligned} &\min_{x'} D_h(h(x'), h_t) \\ &\text{s.t. } ||x - x'||_\infty \leq \epsilon \end{aligned} \tag{23}$$

Once the attacker has embedded the adversarial images in the database, targeted attacks enable image retrieval from a specific category upon user queries. For example, as illustrated in [46], the database could mistakenly return the advertisements of branded beer from the database upon the query of facial lotions.

Attack Model. Attackers can carry out both untargeted and targeted attacks. In particular, we consider two types of *gray-box* and *white-box* attackers. *Gray-box* attackers have access to all the information including network architecture, weights and data, but are not aware of the existence of adversarial detection. The stronger *white-box* attackers are aware of both the model function/parameters and the existence of the detection. So they implement different bypassing strategies as discussed in Section 4.4.

4.3 ADVERSARIAL BEHAVIORS IN THE HAMMING SPACE

Among a variety of artifacts left by adversarial images in classification networks, one of the most evident "adversarial behaviors" is from the softmax function [49, 51]. Due to the fast-growing exponentiation, it magnifies small changes in the logits [51] and becomes overconfident in the presence of adversarial images by regularizing other categories [49]. In contrast to softmax, that the decision is made from a closed set of categories, hashing maps similar images into compact hamming balls in an open hamming space of $\{-1, 1\}^K$. In this

section, we define three criterions to identify adversarial behaviors in the hamming space.

4.3.1 DETECTING UNTARGETED ATTACKS (C_1)

We start with untargeted attacks that maximize the hamming distance between x' and x [41]. Though such behavior is straightforward to discern, we seek a theoretical answer to the distribution of $h(x')$ when the attacking capacity is maximized. Assume the network is capable of learning *perfect* hash codes with the minimum intra-class distance (i.e., equals to zero) and maximum margin between each other, what is the hamming distance from $h(x')$ to the rest of the hash codes? To answer this question, we first establish the distribution of the maximum inter-class distance as illustrated in the following Lemma.

Lemma 1. Given a number of C classes in the K-bit hamming space with (ideally) compact hash codes, the inter-class hamming distance follows a Binomial Distribution of $\mathcal{X} \sim B(K, p)$, where $p = \frac{C}{2(C-1)}$ and $p \approx \frac{1}{2}$ when C is large.

Proof. For all C classes, consider only one bit location at a time. The maximum hamming distance is achieved when there is an equal number of $\frac{C}{2}$, $\{+1\}$ and $\{-1\}$ codes among the C classes. The hamming distance between two bits is either 0 or 1. Thus, among $\binom{C}{2}$ selection of pairs, the probability that the hamming distance equals to 1 is $(\frac{C}{2} \cdot \frac{C}{2})/\binom{C}{2} = \frac{C}{2(C-1)}$. Since all K bits can be selected independently, the probability of the inter-class hamming distance between h_i and h_j equals to d is,

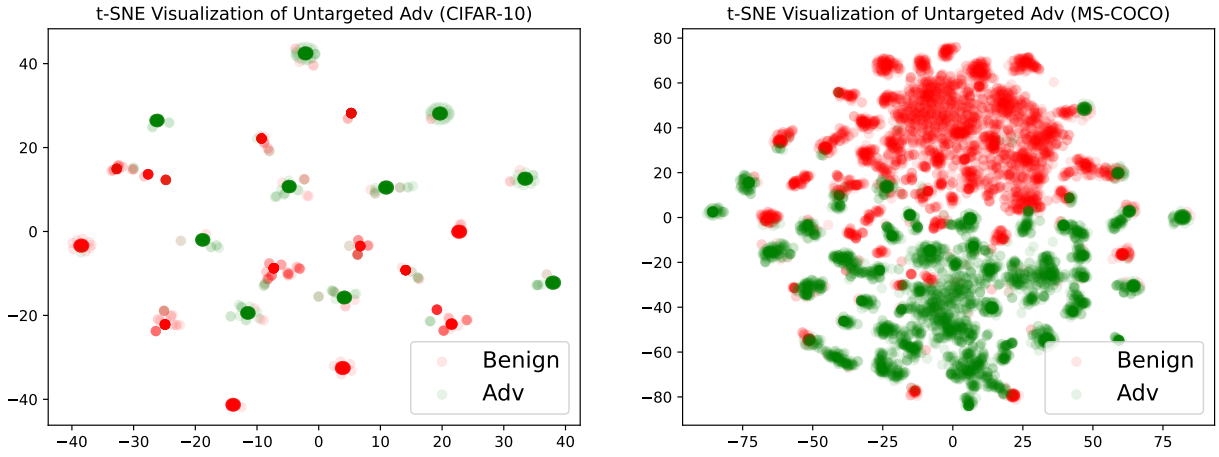
$$Pr\big(D_h(h_i, h_j) = d\big) = \binom{K}{d} p^d (1-p)^{K-d}, \quad p = \frac{C}{2(C-1)}, \tag{24}$$

in which the mean value is Kp and the variance is $Kp(1-p)$. □

From *Lemma 1*, we can further deduce the next theorem.

Theorem 1. For untargeted attacks, the hamming distance from the adversarial image to any other classes follows a Gaussian distribution $\mathcal{N} \sim (K(1-p), Kp(1-p))$.

Proof. In the ideal situation, the untargeted attack maximizes the hamming distance from $D_h(h(x'), h(x_i))$ to K. Thus, for any other hash codes $h(x_j)$, the hamming distance is $D_h(h(x'), h(x_j)) = K - D_h(h_i, h_j)$, which is also a Binomial distribution with the mean



(a) **(b)**

FIG. 19. t-SNE visualization of untargeted adversarial images vs. original images of different datasets (a) CIFAR-10. (b) MS-COCO.

of $K(1-p)$ and the same variance. When the hash bits K is a large value, it can be approximated by a Gaussian distribution $\mathcal{N} \sim (K(1-p), Kp(1-p))$ [118]. □

Example. When $K = 64$ bits, and C is large ($p \to \frac{1}{2}$), using the three-sigma rule, the confidence interval is $(K(1-p) - 3\sqrt{Kp(1-p)}, K(1-p) + 3\sqrt{Kp(1-p)})$. In other words, there is 99.73% confidence that the hamming distance from an untargeted adversarial image to any other classes would be within the $(20, 44)$ interval with the mean of $K/2 = 32$, which is sufficiently distinguishable in the hamming space. Note that the above analysis serves an upper bound for the detection because the ideal case of achieving optimal min/max intra and inter-class distance is still an ongoing effort [18, 119]. Thus, we demonstrate the t-SNE visualization of untargeted adversarial images vs. benign images on CIFAR-10 and MS-COCO in Fig. 19. It is observed that despite of a few samples, the majority of the adversarial images are sufficiently distinguishable based on hamming distance. Hence, we design the first detection criterion.

Criterion 1 (Hamming Distance). For query x, collect the set of top-k hash codes

$\mathcal{H}_k$ and calculate the average hamming distance to $h(x)$.

$$C_1 = \frac{1}{|\mathcal{H}_k|} \sum_{h(x_k) \in \mathcal{H}_k} D_h\big(h(x_k), h(x)\big) \tag{25}$$

C_1 is the average hamming distance of the top-k retrieval results, i.e., a scalar value and we can compare it with a threshold T_1 calculated on benign samples. The computational process of C_1 follows the normal retrieval procedures using the top-k hash codes. To detect targeted attacks, we develop the next criterion.

4.3.2 DETECTING TARGETED ATTACKS (C_2)

While untargeted attacks attempt to induce a bit flip that makes $h(x') = -h(x)$, targeted attacks minimize the hamming distance between $h(x')$ and an arbitrary target code h_t (e.g., such as computed from consensus voting [44] of a category). To find an appropriate metric to identify them, we have the following observation.

Observation 1. For the quantization loss of benign images L_Q^b and targeted images L_Q^t, the relation $L_Q^b > L_Q^t \approx 0$ holds.

Recall that the original targeted objective in Eq. (23) is not differentiable regarding the targeted binary code of x'. The implementation approximates via a continuous relaxation and the goal is to minimize the distance between the continuous output from the $\tanh(\cdot)$ function and the target code [14]. As more gradient descent steps are taken, the quantization loss $L_Q^t \to 0$, when their inter-distance is minimized. This is in close analogy with the adversarial images on softmax classifications while the targeted probabilities become overconfident [49]. In contrast, for all the benign samples, it is difficult to find the optimal model parameters to push L_Q^b towards zero during the training process. An example of ImageNet is shown in Fig. 20(a) as the targeted attacks leave a distinguishable gap from benign samples. It is also interesting to compare with the quantization loss of untargeted attacks in Fig. 20(a), which is larger than zero. This is because for untargeted attacks, finding an adversarial subspace that reduces the mAP to zero can be achieved without flipping all the bits, i.e., much easier than targeted attacks. Based on these observations, we develop the second detection criterion.

Criterion 2 (Quantization Loss). Calculate the l_p distance from the output of network

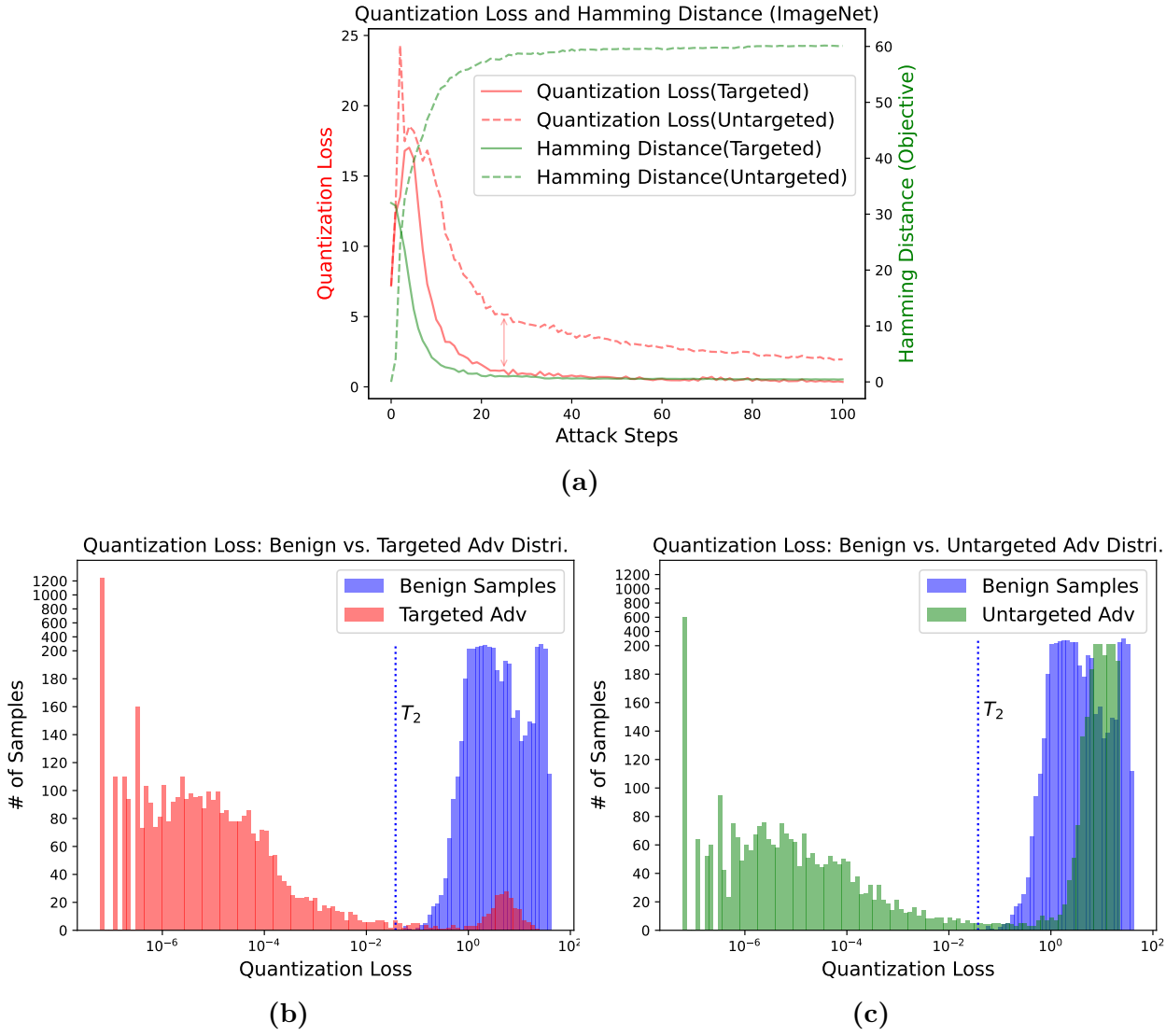


FIG. 20. Example of identifying targeted attacks based on quantization loss on ImageNet. (a) The quantization loss for targeted attacks concentrates around zero vs. the benign samples. (b) Targeted attacks push the quantization loss to zero compared to untargeted attacks. (c) 60% of the untargeted attacks also concentrate around zero.

$f_\theta(x)$ (logits before the sign function) and its hash code $h(x)$,

$$C_2 = \|h(x) - f_\theta(x)\|_p \tag{26}$$

C_2 is the quantization loss of query x. Here, we use the l_1 distance ($p = 1$) and obtain a threshold T_2 on benign samples offline. Figs. 20(b)(c) show the distribution of quantization

loss between the adversarial and the benign images. As $C_2 \to 0$ for targeted attacks, we can see that using T_2 can effectively identify most of the attacks; using C_2 also identifies about 60% of the untargeted attacks,

4.3.3 DETECTING PREDICTION INCONSISTENCY (C_3)

C_1 and C_2 alone are not sufficient. In principle, detection works by limiting the attacker's action space in a confined region. Perturbation can be generally treated as an artificial noise with high-frequency components [52]. Thus, a common approach is to apply local, non-local smoothing filters [54], auto-encoder denoiser [120], color bit reduction [54], quantization [55], and measure the response sensitivity to the denoised images. The adversarial images are more prone to produce a different result, while the benign samples are less sensitive. These denoising operations reduce the entropy (randomness) and the input dimensions of adversarial space that the perturbations can act upon.

We extend this principle in deep hashing to formulate Criterion 3. Denote the transformation [54, 55, 120] as $t(\cdot)$. For query x, C_3 measures the hamming distance between a transformed $t(x)$ and x based on the output before the sign function.

Criterion 3 (Prediction Inconsistency).

$$C_3 = D_h\big(f_\theta(t(x)), f_\theta(x)\big) \tag{27}$$

In other words, C_3 quantifies the disagreement between the original and transformed inputs, which can be evaluated against a threshold T_3 calculated offline on benign samples.

4.3.4 PUT EVERYTHING TOGETHER

The overall detection combines the three criterions: given a query image x, we calculate $\{C_1, C_2, C_3\}$ and compare with thresholds $\{T_1, T_2, T_3\}$. If (a) $C_1 < T_1$; (b) $C_2 > T_2$; (c) $C_3 < T_3$, the input is considered as benign; otherwise, if any of them is not satisfied, the input is rejected as an adversarial example. The computation time is bounded by C_3 since it requires two retrievals. To minimize the compute time, the system can combine the original

query and its denoised copy into a batched query. In case the GPU has sufficient resources, it should have minimum overhead as discussed in Section 4.4.5.

4.4 EXPERIMENTS

We introduce our experimental setting and extensive experimental results in this section.

4.4.1 IMPLEMENTATION

We evaluate our mechanism on the CIFAR-10, ImageNet, MS-COCO and NUSWIDE datasets that are commonly used for deep hashing [13–18] and adopt CSQ [18] with ResNet50 [3] as the base model. The RMSProp optimizer [121] with learning rate 10^{-5} is used for training of 150 epochs. The weight of the quantization loss is set to 10^{-4}. For the four datasets, the mAPs are 0.854, 0.883, 0.884, and 0.843.

We compare with several benchmarks originally designed for softmax classification: Local Intrinsic Dimensionality(LID) [50], reduction of color Bit Depth (FS-Bit) [54], Median Smoothing (FS-Median), Non-local Means (FS-NLM) [54], FS-Adaptive [55] and MeanBlur [49]. FS-Adaptive uses the entropy of input as a metric to adaptively reduce the input space using scalar quantization and smoothing spatial filter. We select MeanBlur as the denoising technique for our method and use True Positive Rate (TPR) with fixed False Positive Rate (FPR) as the detection metric, i.e., a successfully detected adversarial input is counted as a true positive, while a misidentified benign sample is counted as false positive.

All detection methods are evaluated against both untargeted [41] and targeted [44] PGD [42] and untargeted CW [22] attacks[1]. For PGD, the step size is set to 1.0 and the L_∞ norm of perturbation ϵ is set to 8, 16, and 32 with 100 steps. For CW attack, the learning rate is set to 0.1 and 0.01 denoted as CW-a and CW-b with 500 steps.

4.4.2 DETECTION OF GRAY-BOX ATTACKS

Our Method. Table 7 shows the TPRs of different detection methods when we fix the

[1]The CW attack is initially designed for softmax classification and only a few logits are optimized. But for deep hashing, all hashing bits need to be optimized which rarely succeed for targeted CW attacks. Thus, we focus on the untargeted CW attack in deep hashing.

TABLE 7. Detection rate of adversarial examples with 0.05 FPR on benign samples.

		PGD($\epsilon = 8$)		PGD($\epsilon = 16$)		PGD($\epsilon = 32$)		CW-a	CW-b
		Untgt	Tgt	Untgt	Tgt	Untgt	Tgt	Untgt	Untgt
CIFAR-10	LID [50]	0.8510	0.9600	0.9020	0.9850	0.9520	0.9890	0.8016	0.8783
	FS-BitDepth [54]	0.0000	0.0000	0.0570	0.0000	0.0000	0.0000	0.0250	0.4790
	FS-Median [54]	0.8660	0.4570	0.9800	0.6160	**0.9900**	0.8420	0.9653	0.9760
	FS-NLM [54]	0.8120	0.8190	0.7990	0.8250	0.8650	0.8270	0.8044	0.8559
	FS-Adaptive [55]	0.4900	0.5170	0.5670	0.4930	0.6550	0.5260	0.4189	0.4745
	MeanBlur [49]	0.8840	0.7310	0.9060	0.7260	0.9610	0.8040	0.9584	0.9760
	Ours	**0.9250**	**0.9960**	**0.9540**	**0.9930**	0.9880	**0.9950**	**0.9986**	**0.9985**
ImageNet	LID [50]	0.7822	0.9340	0.8628	0.9594	0.8846	0.9630	0.3017	0.2405
	FS-BitDepth [54]	0.0104	0.0018	0.0004	0.0000	0.0000	0.0000	0.5649	0.9858
	FS-Median [54]	0.9146	0.8080	0.8628	0.7518	0.9218	0.8134	**0.9684**	0.9858
	FS-NLM [54]	0.9738	0.9660	0.9762	0.9622	0.9658	0.9556	0.8386	0.9198
	FS-Adaptive [55]	0.9594	0.9570	0.9618	0.9578	0.9430	0.9430	0.8140	0.8632
	MeanBlur [49]	0.9924	0.9546	0.9676	0.9066	0.9530	0.8708	0.9509	0.9906
	Ours	**0.9958**	**0.9966**	**0.9916**	**0.9942**	**0.9918**	**0.9956**	0.9579	**0.9953**
MS-COCO	LID [50]	0.9676	0.9880	0.9906	**0.9980**	0.9966	0.9992	0.4362	0.4541
	FS-BitDepth [54]	0.0176	0.0076	0.0002	0.0004	0.0002	0.0000	0.4179	0.9545
	FS-Median [54]	0.9974	0.9290	0.9930	0.8854	0.9952	0.9246	**1.0000**	**1.0000**
	FS-NLM [54]	**1.0000**	0.9828	**1.0000**	0.9784	0.9995	0.9746	0.9989	0.9992
	FS-Adaptive [55]	**1.0000**	0.9716	**1.0000**	0.9656	0.9996	0.9492	0.9960	0.9992
	MeanBlur [49]	**1.0000**	0.9788	0.9990	0.9578	0.9974	0.9412	**1.0000**	**1.0000**
	Ours	**1.0000**	**0.9888**	0.9998	0.9798	**0.9998**	0.9784	**1.0000**	**1.0000**
NUSWIDE	LID [50]	0.7466	**0.9992**	0.8376	**0.9914**	0.8514	**0.9933**	0.3390	0.5204
	FS-BitDepth [54]	0.0110	0.0067	0.0000	0.0000	0.0000	0.0000	0.2336	0.9082
	FS-Median [54]	0.9648	0.8367	0.9352	0.7857	0.9471	0.8695	0.9972	**1.0000**
	FS-NLM [54]	**1.0000**	0.9710	**1.0000**	0.9710	**1.0000**	0.9705	**1.0000**	**1.0000**
	FS-Adaptive [55]	**1.0000**	0.9543	0.9995	0.9514	0.9995	0.9476	0.9886	**1.0000**
	MeanBlur [49]	0.9981	0.9571	0.9852	0.9257	0.9800	0.9138	**1.0000**	**1.0000**
	Ours	**1.0000**	0.9871	0.9976	0.9800	0.9967	0.9848	**1.0000**	**1.0000**

FPR at 0.05. It is observed that our method makes robust detection of targeted attacks with most of the TPRs over 0.95, which surpasses the detection rates of the 6 benchmarks by 0.248 on average (33% improvement). Untargeted attacks are relatively easier to detect, thus the benchmark TPRs are higher than targeted attacks. Our method can still improve the baselines by 0.206 on average (26% improvement).

Compare with LID and Color Bit Depth. Although LID performs well on most of

the targeted PGD attacks, it does not generalize to the CW attacks. Furthermore, different from the softmax networks that the last few layers often offer better detections [50], LID is quite sensitive to which layers should those features be extracted in deep hashing, which adds extra configuration overhead. On the other hand, it is interesting to see that the reduction of color bit depth (FS-BitDepth) does not provide defense against PGD attacks at all as most of the detection rates are around zero, but it performs contrastively better on CW-b. This is because bit depth reduction is only effective to filter out small noise/perturbations from the CW attacks, but not for large noise from the PGD attacks. Once the small perturbations from CW attacks are removed, the adversarial input and its denoised copy would yield large difference and get detected. This is also validated from CW-a and CW-b: since CW-b has lower learning rate than CW-a, it results in smaller perturbations and the detection rates are much higher than CW-a. In sum, LID and Color Bit Depth reduction are only effective against a single type of attacks.

Compare with Spatial Denoising Methods. Reduction of bit depth does not take advantage of the spatial information like the rest four benchmarks using non-local mean, median, etc. As shown in Table 7, though they generally have over 0.9 detection rates on untargeted attacks, there is a 10-20% gap in detecting targeted attacks. Such a gap can be explained by the attack mechanisms as targeted PGD attacks take more gradient steps. This lands the image deep into the adversarial space, which is more robust to pixel-level modifications such as denoising [46, 109]. However, our method provides an extra layer of defense from C_2 to specifically monitor the value of quantization loss as targeted attacks bring it to zero, thereby complementing C_3 when targeted PGD attacks push the inputs deep into the adversarial space.

4.4.3 DETECTION OF WHITE-BOX ATTACKS

White-box Attacks. Next, we consider white-box attackers who can conduct countermeasures to evade detection. Since the effects from denoising techniques are similar, we use MeanBlur as the representative here and adopt *backward pass differentiable approximation* [122] to estimate the gradients. The white-box attacker can apply different strategies

against C_1, C_2 and C_3:

- C_1 relies on the hamming distance between hash codes to detect outliers. Thus, an effective evasion is to generate adversarial examples with the same binary hash codes as benign images.

- C_2 detects near-zero quantization loss by accessing the logits before the sign function. To bypass this detection, the attacker can optimize the following objective. The first part minimizes the hamming distance to the targeted hash code. The second part (including the minus sign) attempts to enlarge the quantization loss, which amortizes the adversarial behavior identified from C_2.

$$\min_{x'} D_h(f_\theta(x'), h_t) - \lambda_1 ||h(x') - f_\theta(x')||_1 \tag{28}$$

- C_3 detects the disagreement between $f_\theta(x')$ and the denoised copy $f_\theta(t(x'))$. Thus, a countermeasure is to introduce a regularized term to minimize such difference by bringing $f_\theta(x')$ and $f_\theta(t(x'))$ to the same target code h_t,

$$\min_{x'} D_h(f_\theta(x'), h_t) + \lambda_2 D_h(f_\theta(t(x')), h_t). \tag{29}$$

By combining them together, the white-box attacker constructs a joint optimization objective,

$$\min_{x'} \underbrace{D_h(f_\theta(x'), h_t)}_{\text{adv loss}} - \underbrace{\lambda_1 ||h(x') - f_\theta(x')||_1}_{\text{quantization loss}} + \underbrace{\lambda_2 D_h(f_\theta(t(x')), h_t)}_{\text{denoised adv loss}}. \tag{30}$$

For λ_1, a larger value penalizes C_2 more, and vice versa. To reduce the searching effort, λ_2 is set to 1 since the two terms in Eq. (29) carry equal importance.

Detection of White-box Attacks. Fig. 21 demonstrates the AUC scores with different λ_1 values of CIFAR-10 and NUSWIDE datasets. As highlighted, for a fixed λ_1, there is at least one of the criterions with high AUC scores (larger than 0.9), which guarantees our method is robust against white-box attackers. E.g., in CIFAR-10, when λ_1 increases from 0.0075 to 0.008, C_2 drops from 0.9 to near zero; however, C_3 quickly gaps up above 0.9 to compensate the weaknesses from C_2, and C_1 also increases. This phenomenon reveals that the white-box attacker can barely optimize C_2 and C_3 at the same time, i.e., it is difficult

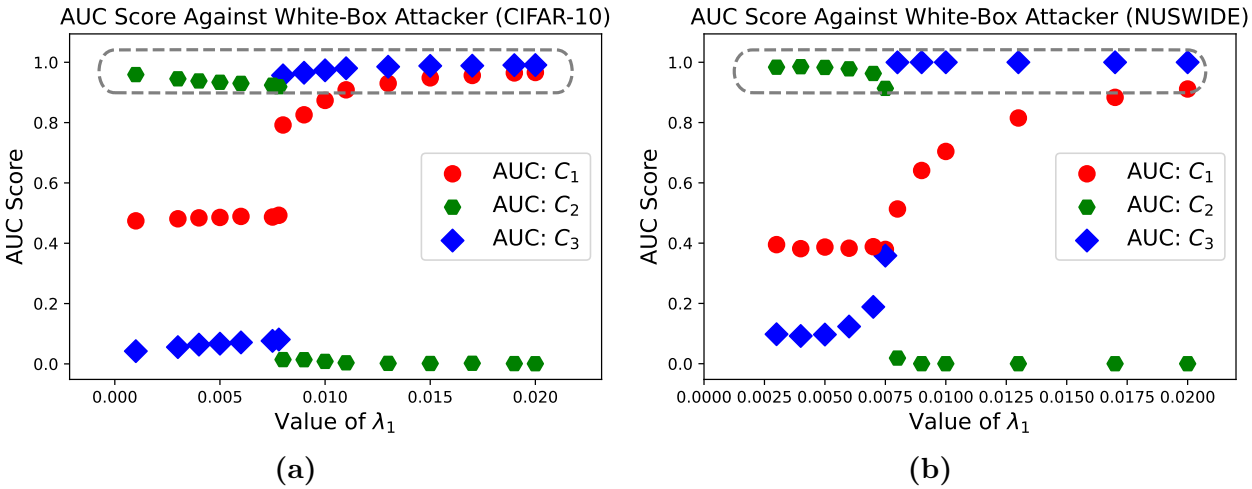


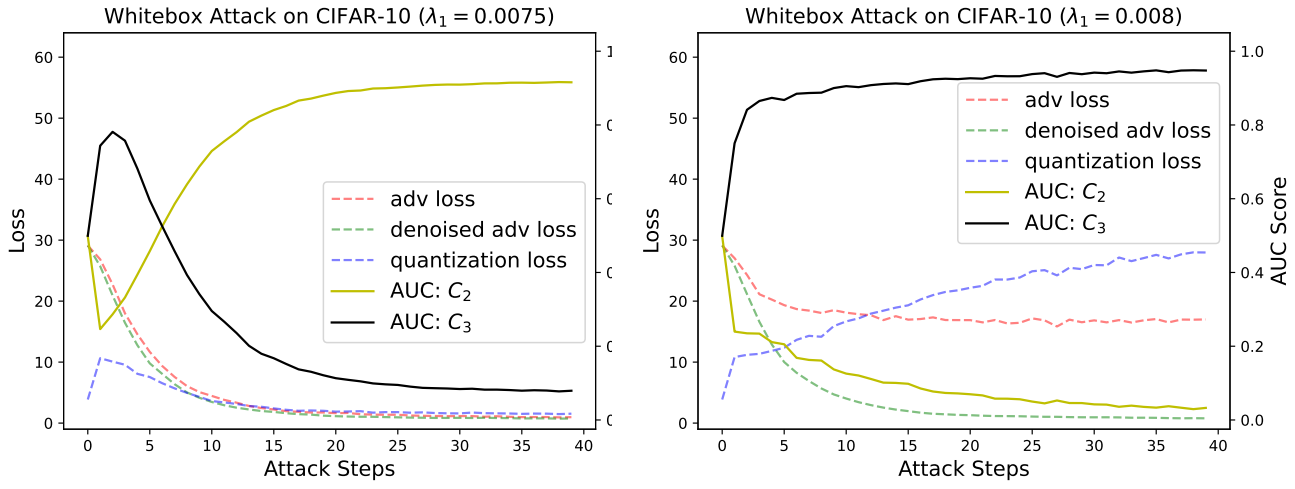
FIG. 21. AUC Scores from different criterions by tuning λ_1 against white-box attackers: a) CIFAR-10; b) NUSWIDE.

to find an adversarial space that satisfies both: 1) its quantization loss is much larger than zero; 2) the adversarial example and its denoised copy yield the same targeted hash code.

Recall that the attacker's goal is to minimize both *adv loss* and *denoised adv loss*, but to maximize *quantization loss*. Fig. 22 further demonstrates the trace of the optimization process for each sub-objective in Eq. (30) when λ_1 changes from 0.0075 to 0.008 (the threshold from above). When $\lambda_1 = 0.0075$, the *quantization loss* is not being maximized compared to *adv* and *denoised adv loss*. Hence, we see that white-box attacker is detected by C_2. When λ_1 increases to 0.008, the optimization starts to weigh more on the *quantization loss* but the original *adv loss* is no longer minimal, which leads to successful detection from C_3 instead.

4.4.4 ABLATION STUDY

We present the ablation study to quantify the contribution of each criterion in the overall detection. We use C_3 alone as the baseline and add C_1 and C_2 with their averaged gain shown in Table 8. The result is consistent with the defense objectives as the addition of C_1 and C_2 helps improve the detection rates of the untargeted and targeted attacks by 0.0533 and 0.1377, respectively. Meanwhile, C_1 and C_2 contribute almost independently on the overall



(a) **(b)**

FIG. 22. Trace of the white-box attack process of sub-objectives between two cases on the threshold (a) $\lambda_1 = 0.0075$. (b) $\lambda_2 = 0.008$.

detection, e.g., $C_1 + C_3 = 0.0533$ plus $C_2 + C_3 = 0.0161$ is equal to $C_1 + C_2 + C_3 = 0.0692$ for untargeted attacks and the same also holds for targeted attacks. This validates that all three criterions act as indispensable parts to detect adversarial in deep hashing.

TABLE 8. Ablation study: detection rates of different combinations ($\epsilon = 32$)

	CIFAR-10		ImageNet		MS-COCO		NUSWIDE		*Avg.*	*Gain*
	Untgt	Tgt	Untgt	Tgt	Untgt	Tgt	Untgt	Tgt	Untgt	Tgt
C_3 Alone	0.8160	0.7460	0.9110	0.8338	0.9954	0.9076	0.9757	0.8904	–	–
$C_1 + C_3$	0.9870	0.7580	0.9522	0.8460	0.9966	0.9098	0.9757	0.8904	*0.0533*	*0.0066*
$C_2 + C_3$	0.8170	0.9830	0.9504	0.9828	0.9992	0.9784	0.9961	0.9847	*0.0161*	*0.1377*
$C_1 + C_2 + C_3$	0.9880	0.9950	0.9916	0.9956	0.9992	0.9784	0.9961	0.9847	*0.0692*	*0.1439*

4.4.5 COMPUTATIONAL TIME

Finally, we evaluate the computational overhead of the detection mechanism. In practice,

the system can accumulate queries into a batch to enhance the utilization of GPU resources and reduce cost. Fig. 23 shows the average retrieval time per sample/batch. First, it is observed that the average time per sample is under 0.05s and further reduced as we increase the batch size. Once the batch size is small, detection introduces negligible overhead because the GPU is underutilized; as the batch size increases, an additional retrieval from the denoised copy in C_3 enlarges the gap between normal retrieval since the GPU resources have been fully utilized. Thus, our detection introduces minimum overhead when the image retrieval system accumulates relatively small batch and responds to queries in real-time.

4.5 ADDITIONAL RESULTS

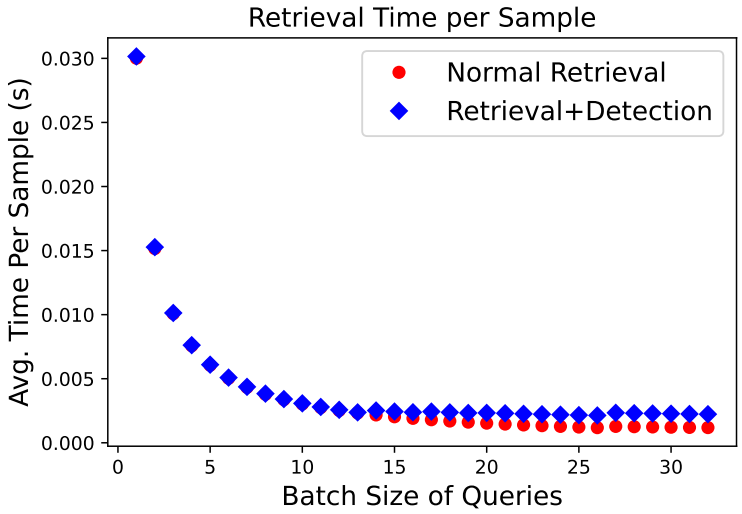
We provide additional experiments of adversarial training and white-box attacks in this section.

4.5.1 ADVERSARIAL TRAINING

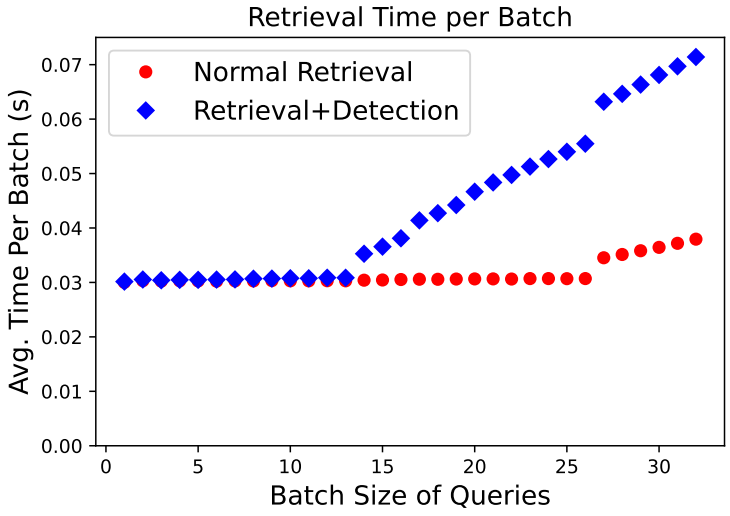
As a proactive defense, adversarial training aims to solve a min-max optimization problem, that the inner optimization finds adversarial examples to maximize the loss function, whereas the outer optimization minimizes the overall loss. Thus, the trade-off between model accuracy and robustness becomes the key in adversarial training. In the following, we show that such gap becomes even larger in deep hashing and is non-trivial to handle with the conventional adversarial training method.

Implementation Details. Our implementation is based on Free Adversarial Training (FreeAT) [123]. It parallelizes parameter update and adversarial generation, thus reducing the overhead from the inner optimization. During training, it replays the same minibatch data m times to update the model parameters as well as generates adversarial examples for the next iteration. Small m has little effect on robustness but large one would impact on the model accuracy. From [123], the best case of softmax classification (when $m = 8$) has 0.46 accuracy gain from the adversarial attacks (PGD and CW) and loses about 0.1 accuracy on CIFAR-10. Next, we validate if similar value holds for deep hashing.

We adopt the FreeAT method in deep hashing using their configuration on CIFAR-10.



(a)



(b)

FIG. 23. Computation time of different batch size: a) per sample; b) per batch.

We set $||\epsilon||_\infty = 8$, with four different repeat times m from 2 to 16, to train four ResNet50 models. For each one of them, we generate PGD adversarial examples with various steps, denoted as PGD-8, PGD-40 and PGD-100 for 8, 40 and 100 steps, respectively.

Results. The first row of Table 9 shows that without adversarial training, PGD attacks have successfully lowered the mAP to near zero. With FreeAT, the mAP improves under

TABLE 9. mAPs of FreeAT for deep hashing on CIFAR-10

Training	Evaluated Against			
	Benign	PGD-8	PGD-40	PGD-100
No FreeAT	0.854	0.028	0.030	0.033
FreeAT $m = 2$	**0.889**	0.070	0.035	0.035
FreeAT $m = 4$	0.838	0.222	0.070	0.055
FreeAT $m = 8$	0.769	**0.328**	**0.146**	**0.130**
FreeAT $m = 16$	0.683	0.278	0.122	0.114

PGD-8, but still performs poorly with large attack steps such as PGD-40 and PGD-100. For the best case here when $m = 8$, PGD attack with 100 steps still manages to lower the mAP to 0.130. Not only having such low fidelity in defense, adversarial training also brings down the model accuracy on benign samples. With more m steps involved, the mAP on benign samples decreases drastically from 0.854 to 0.683[2], while no significant improvement of adversarial robustness is found. Compared to softmax classification, the gap is widening for deep hashing so we pursue adversarial detection in this chapter.

4.5.2 WHITE-BOX EXPERIMENTS

We present additional results of the white-box attacks. Since the attacker need to design specific objectives regarding the detection methods, we choose MeanBlur [49] as a benchmark for comparison. The detection rate is True Positive Rate (TPR) with a certain threshold calculated by False Positive Rate (FPR) on benign samples. We use the PGD attack with $||\epsilon||_\infty = 32$. For the best success rate and ability to bypass the detection, we choose $\lambda_1 = 0.0075$. In Table 10, we show the detection results on four datasets evaluated by TPR. Given FPR= 0.1 and FPR= 0.2, our method achieves an average of 0.7309 and 0.8403 white-box detection rates, whereas the detection rates for MeanBlur is close to zero. This further validates that our method is robust against the strongest white-box attackers. It is also interesting to see that both methods perform relatively better on single-label datasets

[2]The only exception is the model with $m = 2$, which improves the performance on benign samples. It is inline with [6], where adversarial training is found to improve accuracy on clean images for some parameter settings.

(CIFAR-10 and ImageNet) than multi-label datasets (MS-COCO and NUSWIDE), which is different from the gray-box attacks. The reason behind that need to be further studied in the future.

TABLE 10. Detection rate (TPR) of the adversarial examples against white-box attacks (PGD, $\epsilon = 32$, $\lambda_1 = 0.0075$)

Detecotr	FPR	CIFAR10	ImageNet	MS-COCO	NUSWIDE
MeanBlur [49]	0.1	0.0340	0.0218	0.0044	0.0057
	0.2	0.0480	0.0250	0.0076	0.0076
Our Method	0.1	0.9560	0.8824	0.4648	0.6204
	0.2	0.9790	0.9180	0.6742	0.7900

4.6 RELATED WORK

We introduce existing works in deep hashing, adversarial attacks, and adversarial defenses in this section.

4.6.1 DEEP HASHING

Image retrieval uses nearest neighbor search to return the semantically related images of query inputs. Traditionally, it relies on hand-crafted visual descriptors to reduce the computational cost of similarity measure [72, 124]. Recently, empowered by deep learning, end-to-end hash learning improves the performance to a new level [13–18]. They use the similarities between image pairs to train deep hashing models in a supervised manner by transforming the high-dimensional images into compact hash codes, on which neighboring search can be efficiently performed based on hamming distance. To convert the continuous outputs into discrete binary codes, common approaches use continuous relaxation such as sigmoid or hyperbolic tangent functions to approximate the discrete binary thresholding [13–18]. Our work exploits the adversarial behaviors originated from this approximation process, thus can be applied to a variety of deep hashing models.

4.6.2 ADVERSARIAL ATTACKS

Deep neural networks are known to be vulnerable to the non-perceptible perturbations [19]. The Fast Gradient Sign Method (FGSM) [20] generates perturbations in the direction of the signed gradient to maximize the loss function in one-shot computation. The Basic Iterative Method (BIM) [21] and Projected Gradient Descent (PGD) [42] take iterative steps (from random initialization) to achieve higher attack success. There are several other variants [22, 74, 75, 122], e.g., the CW attack aims at minimizing the perturbations to evade detection.

By using the deep learning backends, deep hashing inherits the vulnerability from neural networks. With some slight adaptation, recent works have shown that adversarial attacks can also mislead image retrieval systems [41,44–46,104]. The attacks can be generally categorized into *untargeted* and *targeted attacks*. *Untargeted attacks* divert the query away from the correct results, which make the system retrieve irrelevant images or simply nothing. [41] proposes an untargeted attack to maximize the hamming distance between adversarial and benign samples. [125, 126] craft adversarial examples based on iterative retrievals from a black-box model. [104] hides private images in the database into a non-retrievable subspace by minimizing the number of samples around the private images. *Targeted attacks* make the systems return images from a targeted category, different from the inputs. [44, 45] minimize the average hamming distance between the adversarial examples and a set of images with a target label. [46] enhances targeted transferability to a black-box model via injecting random noise into the adversarial generation. Our work defends against both untargeted and targeted attacks in deep hashing.

4.6.3 ADVERSARIAL DEFENSES

Most of the defense mechanisms are based on softmax classification. As proactive measures, gradient masking [127] and adversarial training [42, 123, 128, 129] aim to learn a more robust model. The early defense of [127] starts with an incorrect conjecture that ascribes adversarial example to high nonlinearity/overfitting, and develops defensive distillation to reduce the variations around input. The method is quickly subverted by [22, 122, 130] as

argued in [20] that the primary cause is due to local linearity of neural networks instead.

Hence, a large body of works focus on adversarial training [42, 123, 128, 129] by solving a min-max saddle point problem. However, it is non-trivial to tackle the trade-off between robustness and accuracy [117], which often leads to significant loss on clean image accuracy, with extensive training efforts. Applying adversarial training into the deep hashing domain suffers from even higher accuracy loss as we have experimented (see additional results). For image retrieval system, as long as the adversarial images are detected at the input, we can equivalently thwart the attacks without accuracy loss and training complexities.

Adversarial detections extract the artifacts left by the adversarial examples at different levels: raw pixels [47, 48], feature distributions [48, 49], softmax distributions [51] and frequency components [52]. By analyzing the contrastive distributions of the adversarial and natural images, a detector can be efficiently trained in a supervised or unsupervised manner. Another thread of works rely on the prediction inconsistency by exploiting denoise method and measuring the disagreement between the results [54, 55, 120]. All these works are based on softmax classification. In this work, we discover adversarial behaviors from the hamming space and propose a set of detection criterions including defending against the strongest white-box attackers.

4.7 CHAPTER SUMMARY

In this chapter, we propose an efficient detection of adversarial examples in deep hashing based image retrieval. We design three criterions to identify adversarial behaviors of both targeted and untargeted attacks in the hamming space and consider white-box attackers who are aware of the existence of defense. The extensive evaluations demonstrate that the proposed detection can surpass previous defense techniques by more than 20% on average and is also robust against white-box attacker by limiting its action space.

CHAPTER 5

CONCLUSIONS

This dissertation focuses on the privacy and security concerns related to adversarial examples in deep hashing image retrieval. A series of works have been proposed from both the attack side and the defense side, standing on different views of image retrieval systems, including users, attackers, and the model.

Specifically, first, a privacy concern that private images may be retrieved by adversaries in image retrieval to extract privacy information is pointed out. We propose a method to introduce an imperceptible adversarial perturbation on the original image to protect it from being retrieved. The perturbed image is stashed in a subspace that maximizes distances to all the other samples in the hash space, to avoid being retrieved and extracted by adversaries.

Then, we explore targeted black-box adversarial attacks in deep hashing image retrieval. This reveals a security concern that we could invade users' retrieval results by proactively providing adversarial examples to image retrieval systems. We have found an implication of transferability, which is robustness to random noise in the surrogate model. Based on this implication, we propose an adversarial attack using random noise as a proxy to enhance black-box transferability.

Finally, we focus on mitigating the security concern brought by adversarial examples. We propose a detection framework to identify adversarial examples in the inference time. We study adversarial behaviors in deep hashing image retrieval. Based on the behaviors, we propose an unsupervised adversarial detection method that ensemble three criterions, which does not rely on specific adversarial examples. Our proposed detection surpasses the previous detection method by more than 20% on average and is efficient to limit the adversarial action space.

To conclude, in this dissertation, we have studied the adversarial examples in deep hashing image retrieval. We have proposed attack and defense adversarial methods to solve or reveal the privacy and security concerns related to adversarial examples in deep hashing. We hope

this work would lead to a significant impact on the research community and it would shed lights on future research.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[6] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 819–828.

[7] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 687–10 698.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[9] C. Wang, Y. Xiao, X. Gao, L. Li, and J. Wang, "Close the gap between deep learning and mobile intelligence by incorporating training in the loop," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1419–1427.

[10] ——, "A framework for behavioral biometric authentication using deep metric learning on mobile devices," *IEEE Transactions on Mobile Computing*, 2021.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[13] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2064–2072.

[14] Z. Cao, M. Long, J. Wang, and P. S. Yu, "Hashnet: Deep learning to hash by continuation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5608–5617.

[15] Y. Cao, M. Long, B. Liu, and J. Wang, "Deep cauchy hashing for hamming space retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1229–1237.

[16] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[17] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 27–35.

[18] L. Yuan, T. Wang, X. Zhang, F. E. Tay, Z. Jie, W. Liu, and J. Feng, "Central similarity quantization for efficient image and video retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3083–3092.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. B. Estrach, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[21] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[22] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[23] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1369–1378.

[24] X. Wei, S. Liang, N. Chen, and X. Cao, "Transferable adversarial attacks for image and video object detection," *arXiv preprint arXiv:1811.12641*, 2018.

[25] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No need to worry about adversarial examples in object detection in autonomous vehicles," *arXiv preprint arXiv:1707.03501*, 2017.

[26] A. Arnab, O. Miksik, and P. H. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 888–897.

[27] "Google image search," https://www.google.com/imghp.

[28] "Bing," https://www.bing.com/visualsearch.

[29] "Amazon image search," https://www.amazon.com/b?node=17387598011.

[30] D. Shankar, S. Narumanchi, H. Ananya, P. Kompalli, and K. Chaudhury, "Deep learning based large scale visual recommendation and search for e-commerce," *arXiv preprint arXiv:1703.02344*, 2017.

[31] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*, 2004, pp. 253–262.

[32] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2916–2929, 2012.

[33] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2074–2081.

[34] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," *Advances in neural information processing systems*, vol. 22, 2009.

[35] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke, "Towards statistical queries over distributed private user data," in *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation NSDI 12)*, 2012, pp. 169–182.

[36] A. Reznichenko and P. Francis, "Private-by-design advertising meets the real world," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 116–128.

[37] M. Hardt and S. Nath, "Privacy-aware personalization for mobile advertising," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 662–673.

[38] S. Nath, F. X. Lin, L. Ravindranath, and J. Padhye, "Smartads: bringing contextual ads to mobile apps," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013, pp. 111–124.

[39] Y. Han and Y. Shen, "Accurate spear phishing campaign attribution and early detection," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. ACM, 2016, pp. 2079–2086.

[40] J. L. Ledford, *Search engine optimization bible*. John Wiley & Sons, 2015, vol. 584.

[41] E. Yang, T. Liu, C. Deng, and D. Tao, "Adversarial examples for hamming space search," *IEEE transactions on cybernetics*, 2018.

[42] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[43] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.

[44] J. Bai, B. Chen, Y. Li, D. Wu, W. Guo, S.-t. Xia, and E.-h. Yang, "Targeted attack for deep hashing based retrieval," in *European Conference on Computer Vision*. Springer, 2020, pp. 618–634.

[45] X. Wang, Z. Zhang, G. Lu, and Y. Xu, "Targeted attack and defense for deep hashing," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 2298–2302.

[46] Y. Xiao and C. Wang, "You see what i want you to see: Exploring targeted black-box transferability attack for hash-based image retrieval systems," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1934–1943.

[47] Z. Gong, W. Wang, and W.-S. Ku, "Adversarial and clean data are not twins," *arXiv preprint arXiv:1704.04960*, 2017.

[48] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[49] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5764–5772.

[50] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," in *International Conference on Learning Representations*, 2018.

[51] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[52] H. Wang, X. Wu, Z. Huang, and E. P. Xing, "High-frequency component helps explain the generalization of convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8684–8694.

[53] F. Carrara, R. Becarelli, R. Caldelli, F. Falchi, and G. Amato, "Adversarial examples detection in features distance spaces," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[54] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[55] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 72–85, 2018.

[56] "Pinterest visual search tool," https://www.pinterest.com/.

[57] "Tin eye," https://www.tineye.com.

[58] D. Lu, X. Liu, and X. Qian, "Tag-based image search by social re-ranking," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1628–1639, 2016.

[59] X. Ji, W. Wang, M. Zhang, and Y. Yang, "Cross-domain image retrieval with attention modeling," in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017, pp. 1654–1662.

[60] "Alibaba's pailitao," http://www.pailitao.com/.

[61] "Deepfashion: attribute prediction dataset," https://bit.ly/2N4ZGQP.

[62] "Instagram by the numbers," https://bit.ly/2wnRfJ1.

[63] "Ebay by the numbers," https://bit.ly/2NzaEif.

[64] "How google uses the picture you search with," https://support.google.com/websearch/answer/1325808.

[65] "Eu general data protection regulation," https://eugdpr.org/.

[66] W. Bullock, L. Xu, and L. Zhou, "Predicting household demographics based on image data," Apr. 30 2019, uS Patent App. 10/277,714.

[67] M. J. Wilber, V. Shmatikov, and S. Belongie, "Can we still avoid automatic face detection?" in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[68] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/off: Preventing privacy leakage from photos in social networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 781–792.

[69] L. Zhang, K. Liu, X.-Y. Li, C. Liu, X. Ding, and Y. Liu, "Privacy-friendly photo capturing and sharing system," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 524–534.

[70] C. Bo, G. Shen, J. Liu, X.-Y. Li, Y. Zhang, and F. Zhao, "Privacy. tag: Privacy concern expressed and respected," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. ACM, 2014, pp. 163–176.

[71] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 769–790, 2017.

[72] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[73] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.

[74] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017, pp. 506–519.

[75] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

[76] I. E. Akkus, R. Chen, M. Hardt, P. Francis, and J. Gehrke, "Non-tracking web analytics," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 687–698.

[77] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, "The web never forgets: Persistent tracking mechanisms in the wild," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2014, pp. 674–689.

[78] Y. Liu, W. Zhang, and N. Yu, "Protecting privacy in shared photos via adversarial examples based stealth," *Security and Communication Networks*, vol. 2017, 2017.

[79] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2074–2081.

[80] "Mnist dataset," http://yann.lecun.com/exdb/mnist/.

[81] "Cifar10 dataset," https://www.cs.toronto.edu/~kriz/cifar.html.

[82] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[83] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[84] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[85] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[86] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255.

[87] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[88] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[89] Z. Liu, Z. Zhao, and M. 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*, 2019, pp. 306–314.

[90] G. Tolias, F. Radenovic, and O. 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.

[91] T. Brunner, F. Diehl, M. T. Le, and A. Knoll, "Guessing smart: Biased sampling for efficient black-box adversarial attacks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4958–4966.

[92] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *International Conference on Learning Representations*, 2017.

[93] "How search organizes information," https://www.google.com/search/howsearchworks/crawling-indexing/.

[94] R. Pan, M. J. Islam, S. Ahmed, and H. Rajan, "Identifying classes susceptible to adversarial attacks," *arXiv preprint arXiv:1905.13284*, 2019.

[95] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4312–4321.

[96] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *International Conference on Learning Representations*, 2018.

[97] Y. Shi, S. Wang, and Y. Han, "Curls & whey: Boosting black-box adversarial attacks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6519–6527.

[98] W. Zhou, X. Hou, Y. Chen, M. Tang, X. Huang, X. Gan, and Y. Yang, "Transferable adversarial perturbations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 452–467.

[99] N. Inkawhich, W. Wen, H. H. Li, and Y. Chen, "Feature space perturbations yield more transferable adversarial examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7066–7074.

[100] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," *International Conference on Machine Learning*, 2018.

[101] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. 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*, 2017, pp. 15–26.

[102] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu, "Efficient decision-based black-box adversarial attacks on face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7714–7722.

[103] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *International Conference on Learning Representations*, 2018.

[104] Y. Xiao, C. Wang, and X. 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*, 2020, pp. 9651–9660.

[105] T. Tanay and L. Griffin, "A boundary tilting persepective on the phenomenon of adversarial examples," *arXiv preprint arXiv:1608.07690*, 2016.

[106] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," *International Conference on Machine Learning*, 2019.

[107] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang, "Provably robust deep learning via adversarially trained smoothed classifiers," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 292–11 303.

[108] K. Roth, Y. Kilcher, and T. Hofmann, "The odds are odd: A statistical test for detecting adversarial examples," in *International Conference on Machine Learning*, 2019, pp. 5498–5507.

[109] S. Hu, T. Yu, C. Guo, W.-L. Chao, and K. Q. Weinberger, "A new defense against adversarial images: Turning a weakness into a strength," in *Advances in Neural Information Processing Systems*, 2019, pp. 1635–1646.

[110] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "Robustness of classifiers: from adversarial to random noise," in *Advances in Neural Information Processing Systems*, 2016, pp. 1632–1640.

[111] P. Márquez-Neila, M. Salzmann, and P. Fua, "Imposing hard constraints on deep networks: Promises and limitations," *CVPR Workshop*, 2017.

[112] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[113] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[114] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[115] J. Li, R. Ji, H. Liu, X. Hong, Y. Gao, and Q. Tian, "Universal perturbation attack against image retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4899–4908.

[116] Z. Hussain, M. Zhang, X. Zhang, K. Ye, C. Thomas, Z. Agha, N. Ong, and A. Kovashka, "Automatic understanding of image and video advertisements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1705–1715.

[117] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7472–7482.

[118] S. M. Ross, *A first course in probability*. Prentice Hall Upper Saddle River, NJ, 2002, vol. 6.

[119] J. T. Hoe, K. W. Ng, T. Zhang, C. S. Chan, Y.-Z. Song, and T. Xiang, "One loss for all: Deep hashing with a single cosine similarity based learning objective," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[120] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 135–147.

[121] T. Tieleman, G. Hinton *et al.*, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, pp. 26–31, 2012.

[122] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018, pp. 274–283.

[123] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *arXiv preprint arXiv:1904.12843*, 2019.

[124] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[125] X. Li, J. Li, Y. Chen, S. Ye, Y. He, S. Wang, H. Su, and H. Xue, "Qair: Practical query-efficient black-box attacks for image retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3330–3339.

[126] M. Chen, J. Lu, Y. Wang, J. Qin, and W. Wang, "Dair: A query-efficient decision-based attack on image retrieval systems," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1064–1073.

[127] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[128] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2206–2216.

[129] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," *arXiv preprint arXiv:2001.03994*, 2020.

[130] N. Carlini and D. Wagner, "Defensive distillation is not robust to adversarial examples," *arXiv preprint arXiv:1607.04311*, 2016.

VITA

Yanru Xiao
Department of Computer Science
Old Dominion University, Norfolk, VA 23529

Education

- Ph.D. Computer Science, Jan 2018 - Dec 2022 (Expected), **Old Dominion University**
- B.Eng. Computer Science and Technology, Sep 2013 - June 2017, **Center South University**

Experience

- Research Intern, May 2022 - Aug 2022, **Microsoft**

Publications

- **Yanru Xiao** and Cong Wang, " *You See What I Want You to See: Exploring Targeted Black-Box Transferability Attack for Hash-based Image Retrieval Systems*," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
- **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 (CVPR), 2020.
- Cong Wang, **Yanru Xiao**, Xing Gao, Li Li, and Jun Wang, "*A Framework for Behavioral Biometric Authentication using Deep Metric Learning on Mobile Devices*," in IEEE Transactions on Mobile Computing (TMC), 2021.
- Cong Wang, **Yanru Xiao**, Xing Gao, Li Li, and Jun Wang, "*Close the Gap between Deep Learning and Mobile Intelligence by Incorporating Training in the Loop*," in Proceedings of the 27th ACM International Conference on Multimedia (ACM MM), 2019.