Adversarial Ranking Attack and Defense

Mo Zhou[1], Zhenxing Niu[2], Le Wang[1]*, Qilin Zhang[3], and Gang Hua[4]

[1]Xi'an Jiaotong University, [2]Alibaba DAMO MIIL, [3]HERE Technologies, [4]Wormpex AI Research

Abstract. Deep Neural Network (DNN) classifiers are vulnerable to adversarial attack, where an imperceptible perturbation could result in misclassification. However, the vulnerability of DNN-based image ranking systems remains under-explored. In this paper, we propose two attacks against deep ranking systems, *i.e.*, Candidate Attack and Query Attack, that can raise or lower the rank of chosen candidates by adversarial perturbations. Specifically, the expected ranking order is first represented as a set of inequalities, and then a triplet-like objective function is designed to obtain the optimal perturbation. Conversely, a defense method is also proposed to improve the ranking system robustness, which can mitigate all the proposed attacks simultaneously. Our adversarial ranking attacks and defense are evaluated on datasets including MNIST, Fashion-MNIST, and Stanford-Online-Products. Experimental results demonstrate that a typical deep ranking system can be effectively compromised by our attacks. Meanwhile, the system robustness can be moderately improved with our defense. Furthermore, the transferable and universal properties of our adversary illustrate the possibility of realistic black-box attack.

1 Introduction

Despite the successful application in computer vision tasks such as image classification [31, 21], Deep Neural Networks (DNNs) have been found vulnerable to adversarial attacks. In particular, the DNN's prediction can be arbitrarily changed by just applying an imperceptible perturbation to the input image [69, 17]. Moreover, such adversarial attacks can effectively compromise the state-of-the-art DNNs such as Inception [67, 68] and ResNet [21]. This poses a serious security risk on many DNN-based applications such as face recognition, where recognition evasion or impersonation can be easily achieved [12, 64, 30, 72].

Previous adversarial attacks primarily focus on *classification*, however, we speculate that DNN-based image ranking systems [3, 6, 70, 29, 52, 15, 35] also suffer from similar vulnerability. Taking the image-based product search as an example, a fair ranking system should rank the products according to their visual similarity to the query, as shown in Fig. 1 (row 1). Nevertheless, malicious sellers may attempt to raise the rank of their product by adding perturbation to the image (CA+, row 2), or lower the rank of his competitor's product (CA-, row 3);

* Corresponding author.

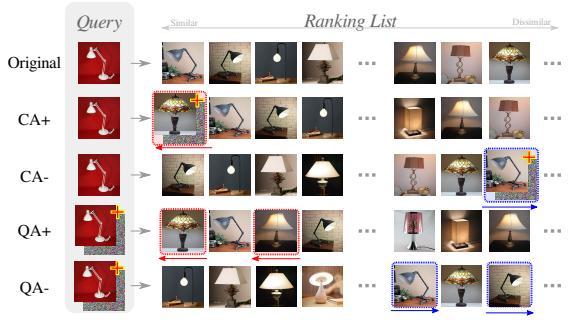

Fig. 1. Adversarial ranking attack that can *raise* or *lower* the rank of chosen candidates by adversarial perturbations. In Candidate Attack, adversarial perturbation is added to the candidate and its rank is *raised* (CA+) or *lowered* (CA-). In Query Attack, adversarial perturbation is added to the query image, and the ranks of chosen candidates are *raised* (QA+) or *lowered* (QA-).

Besides, "man-in-the-middle" attackers (*e.g.*., a malicious advertising company) could hijack and imperceptibly perturb the query image in order to promote (QA+, row 4) or impede (QA-, row 5) the sales of specific products.

Unlike classification tasks where images are predicted independently, the rank of one candidate is related to the query as well as other candidates for image ranking. The relative relations among candidates and queries determine the final ranking order. Therefore, we argue that the existing adversarial classification attacks are incompatible with the ranking scenario. Thus, we need to thoroughly study the *adversarial ranking attack*.

In this paper, adversarial ranking attack aims to *raise* or *lower* the ranks of some chosen candidates $C = \{c_1, c_2, \ldots, c_m\}$ with respect to a specific query set $Q = \{q_1, q_2, \ldots, q_w\}$. This can be achieved by either Candidate Attack (CA) or Query Attack (QA). In particular, CA is defined as to raise (*abbr.* CA+) or lower (*abbr.* CA-) the rank of a single candidate c with respect to the query set Q by perturbing c itself; while QA is defined as to raise (*abbr.* QA+) or lower (*abbr.* QA-) the ranks of a candidate set C with respect to a single query q by perturbing q. Thus, adversarial ranking attack can be achieved by performing CA on each $c \in C$, or QA on each $q \in Q$. In practice, the choice of CA or QA depends on the accessibility to the candidate or query respectively, *i.e.*, CA is feasible for modifiable candidate, while QA is feasible for modifiable query.

An effective implementation of these attacks is proposed in this paper. As we know, a typical DNN-based ranking model maps objects (*i.e.*, queries and candidates) to a common embedding space, where the distances among them determine the final ranking order. Predictably, the object's position in the embedding space will be changed by adding a perturbation to it. Therefore, the essential of adversarial ranking attack is to find a proper perturbation, which could push the object to a desired position that leads to the expected ranking order. Specifically, we first represent the expected ranking order as a set of in-

equalities. Subsequently, a triplet-like objective function is designed according to those inequalities, and combined with Projected Gradient Descent (PGD) to efficiently obtain the desired adversarial perturbation.

Opposed to the proposed attacks, *adversarial ranking defense* is worth being investigated especially for security-sensitive deep ranking applications. Until now, the Madry defense [45] is regarded as the most effective method for classification defense. However, we empirically discovered a primary challenge of diverging training loss while directly adapting such mechanism for ranking defense, possibly due to the generated adversarial examples being too "strong". In addition, such defense mechanism needs to defend against distinct ranking attacks individually, but a *generic* defense method against all CA+, CA-, QA+ and QA- attacks is preferred.

To this end, a shift-distance based ranking defense is proposed, which could simultaneously defend against all attacks. Note that the position shift of objects in the embedding space is the key for all ranking attacks. Although different attacks prefer distinct shift directions (*e.g.*, CA+ and CA- often prefer opposed shifting directions), a large shift distance is their common preference. If we could reduce the shift distance of embeddings incurred by adversarial perturbation, all attacks can be simultaneously defensed. Specifically, we first propose a shift-distance based ranking attack, which aims to push the objects as far from their original positions as possible. And then, the adversarial examples generated from such attack is involved in the adversarial training. Experimental results manifest that our ranking defense can converge and moderately improve model robustness.

In addition, our ranking attacks have some good properties for realistic applications. First, our adversary is transferable, *i.e.*, the adversary obtained from a known DNN ranker can be directly used to attack an unknown DNN ranker (*i.e.*, the network architecture and parameters are unknown). Second, our attacks can be extended to *universal* ranking attacks with slight performance drop, *i.e.*, we could learn a *universal* perturbation to all candidates for CA, or a *universal* perturbation to all queries for QA. Such properties illustrate the possibility of practical black-box attack.

To the best of our knowledge, this is the first work that thoroughly studies the adversarial ranking attack and defense. In brief, our contributions are:

1. The adversarial ranking attack is defined and implemented, which can intentionally change the ranking results by perturbing the candidates or queries.
2. An adversarial ranking defense method is proposed to improve the ranking model robustness, and mitigate all the proposed attacks simultaneously.

2 Related Works

Adversarial Attacks. Szegedy *et al.* [69] claimed that DNN is susceptible to imperceptible adversarial perturbations added to inputs, due to the intriguing "blind spot" property, which was later ascribed to the local linearity [17] of neural networks. Following these findings, many white-box (model architecture

and parameters are known to the adversary) attacking methods [50, 57, 32, 5, 8, 10, 61, 66, 45, 74, 7, 16] are proposed to effectively compromise the state-of-the-art DNN classifiers. Among them, PGD [45] is regarded as one of the most powerful attacks [1]. Notably, adversarial examples are discovered to be transferable [56, 55] among different neural network classifiers, which inspired a series of black-box attacks [65, 73, 76, 41, 11, 24]. On the other hand, universal (*i.e.*, image-agnostic) adversarial perturbations are also discovered [49, 37].

Deep Ranking. Different from the traditional "learning to rank" [38, 27] methods, DNN-based ranking methods often embed data samples (including both queries and candidates) of all modalities into a common embedding space, and subsequently determine the ranking order based on distance. Such workflow has been adopted in distance metric learning [6, 70, 53, 26], image retrieval [3], cross-modal retrieval [52, 15, 35, 29], and face recognition [62].

Adversarial Attacks in Deep Ranking. For information retrieval and ranking systems, the risk of malicious users manipulating the ranking always exists [19, 23]. However, only a few research efforts have been made in adversarial attacks in deep ranking. Liu *et al.* [42] proposed adversarial queries leading to incorrect retrieval results; while Li *et al.* [36] staged similar attack with universal perturbation that corrupts listwise ranking results. None of the aforementioned research efforts explore the *adversarial ranking attack*. Besides, adaptation of distance-based attacks (*e.g.* [61]) are unsuitable for our scenario.

Adversarial Defenses. Adversarial attacks and defenses are consistently engaged in an arms race [77]. Gradient masking-based defenses can be circumvented [2]. Defensive distillation [54, 58] has been compromised by C&W [5, 4]. As claimed in [22], ensemble of weak defenses are insufficient against adversarial examples. Notably, as an early defense method [69], adversarial training [17, 45, 25, 13, 33, 63, 71, 78, 51, 46] remains to be one of the most effective defenses. Other types of defenses include adversarial detection [43, 48], input transformation/reconstruction/replacement [60, 44, 20, 47, 14], randomization [40, 39], network verification [28, 18], *etc.* However, defense in deep ranking systems remains mostly uncharted.

3 Adversarial Ranking

Generally, a DNN-based ranking task could be formulated as a metric learning problem. Given the query q and candidate set $X = \{c_1, c_2, \ldots, c_n\}$, deep ranking is to learn a mapping f (usually implemented as a DNN) which maps all candidates and query into a common embedding space, such that the relative distances among the embedding vectors could satisfy the expected ranking order. For instance, if candidate c_i is more similar to the query q than candidate c_j, it is encouraged for the mapping f to satisfy the inequality $\|f(q) - f(c_i)\| < \|f(q) - f(c_j)\|$[1], where $\|\cdot\|$ denotes ℓ_2 norm. For brevity, we denote $\|f(q) - f(c_i)\|$ as $d(q, c_i)$ in following text.

[1] Sometimes cosine distance is used instead.

Therefore, adversarial ranking attack is to find a proper adversarial perturbation which leads the ranking order to be changed as expected. For example, if a less relevant c_j is expected to be ranked *ahead* of a relevant c_i, it is desired to find a proper perturbation r to perturb c_j, *i.e.*$\tilde{c}_j = c_j + r$, such that the inequality $d(q, c_i) < d(q, c_j)$ could be changed into $d(q, c_i) > d(q, \tilde{c}_j)$. In the next, we will describe Candidate Attack and Query Attack in detail.

3.1 Candidate Attack

Candidate Attack (**CA**) aims to raise (*abbr.* **CA+**) or lower (*abbr.* **CA-**) the rank of a *single* candidate c with respect to a set of queries $Q = \{q_1, q_2, \ldots, q_w\}$ by adding perturbation r to the candidate itself, *i.e.*$\tilde{c} = c + r$.

Let $\mathrm{Rank}_X(q, c)$ denote the rank of the candidate c with respect to the query q, where X indicates the set of all candidates, and a smaller rank value represents a higher ranking. Thus, the **CA+** that *raises* the rank of c with respect to every query $q \in Q$ by perturbation r could be formulated as the following problem,

$$r = \arg\min_{r \in \Gamma} \sum_{q \in Q} \mathrm{Rank}_X(q, c + r), \tag{1}$$

$$\Gamma = \{r \big| \|r\|_\infty \leqslant \varepsilon; r, c + r \in [0, 1]^N\}, \tag{2}$$

where Γ is a ℓ_∞-bounded ε-neighbor of c, $\varepsilon \in [0, 1]$ is a predefined small positive constant, the constraint $\|r\|_\infty \leqslant \varepsilon$ limits the perturbation r to be "visually imperceptible", and $c + r \in [0, 1]^N$ ensures the adversarial example remains a valid input image. Although alternative "imperceptible" constraints exist (*e.g.*, ℓ_0 [66, 9], ℓ_1 [8] and ℓ_2 [5, 50] variants), we simply follow [17, 32, 45] and use the ℓ_∞ constraint.

However, the optimization problem Eq. (1)–(2) cannot be directly solved due to the discrete nature of the rank value $\mathrm{Rank}_X(q, c)$. In order to solve the problem, a surrogate objective function is needed.

In metric learning, given two candidates $c_p, c_n \in X$ where c_p is ranked ahead of c_n, *i.e.*$\mathrm{Rank}_X(q, c_p) < \mathrm{Rank}_X(q, c_n)$, the ranking order is represented as an inequality $d(q, c_p) < d(q, c_n)$ and formulated in triplet loss:

$$L_{\mathrm{triplet}}(q, c_p, c_n) = \left[\beta + d(q, c_p) - d(q, c_n)\right]_+, \tag{3}$$

where $[\cdot]_+$ denotes $\max(0, \cdot)$, and β is a manually defined constant margin. This function is known as the triplet (*i.e.*pairwise ranking) loss [6, 62].

Similarly, the attacking goal of **CA+** in Eq. (1) can be readily converted into a series of inequalities, and subsequently turned into a sum of triplet losses,

$$L_{\mathrm{CA+}}(c, Q; X) = \sum_{q \in Q} \sum_{x \in X} \big[d(q, c) - d(q, x)\big]_+. \tag{4}$$

In this way, the original problem in Eq. (1)–(2) can be reformulated into the following constrained optimization problem:

$$r = \arg\min_{r \in \Gamma} L_{\mathrm{CA+}}(c + r, Q; X). \tag{5}$$

To solve the optimization problem, Projected Gradient Descent (PGD) method [45, 32] (*a.k.a* the iterative version of FGSM [17]) can be used. Note that PGD is one of the most effective first-order gradient-based algorithms [1], popular among related works about adversarial attack.

Specifically, in order to find an adversarial perturbation r to create a desired adversarial candidate $\tilde{c} = c + r$, the PGD algorithm alternates two steps at every iteration $t = 1, 2, \ldots, \eta$. Step one updates $\tilde{c}$ according to the gradient of Eq. (4); while step two clips the result of step one to fit in the ε-neighboring region Γ:

$$\tilde{c}_{t+1} = \mathrm{Clip}_{c,\Gamma}\big\{\tilde{c}_t - \alpha \mathrm{sign}(\nabla_{\tilde{c}_t} L_{\mathrm{CA}+}(\tilde{c}_t, Q, X))\big\}, \tag{6}$$

where α is a constant hyper-parameter indicating the PGD step size, and $\tilde{c}_1$ is initialized as c. After η iterations, the desired adversarial candidate $\tilde{c}$ is obtained as $\tilde{c}_\eta$, which is optimized to satisfy as many inequalities as possible. Each inequality represents a pairwise ranking sub-problem, hence the adversarial candidate $\tilde{c}$ will be ranked ahead of other candidates with respect to every specified query $q \in Q$.

Likewise, the **CA-** that *lowers* the rank of a candidate c with respect to a set of queries Q can be obtained in similar way:

$$L_{\text{CA-}}(c, Q; X) = \sum_{q \in Q} \sum_{x \in X} \big[- d(q, c) + d(q, x) \big]_+. \tag{7}$$

3.2 Query Attack

Query Attack (**QA**) is supposed to raise (*abbr.* **QA+**) or lower (*abbr.* **QA-**) the rank of a set of candidates $C = \{c_1, c_2, \ldots, c_m\}$ with respect to the query q, by adding adversarial perturbation r to the query $\tilde{q} = q + r$. Thus, **QA** and **CA** are two "symmetric" attacks. The **QA-** for *lowering* the rank could be formulated as follows:

$$r = \arg\max_{r \in \Gamma} \sum_{c \in C} \mathrm{Rank}_X(q + r, c), \tag{8}$$

where Γ is the ε-neighbor of q. Likewise, this attacking objective can also be transformed into the following constrained optimization problem:

$$L_{\text{QA-}}(q, C; X) = \sum_{c \in C} \sum_{x \in X} \big[- d(q, c) + d(q, x) \big]_+, \tag{9}$$

$$r = \arg\min_{r \in \Gamma} L_{\text{QA-}}(q + r, C; X), \tag{10}$$

and it can be solved with the PGD algorithm. Similarly, the **QA+** loss function $L_{\mathrm{QA}+}$ for *raising* the rank of c is as follows:

$$L_{\mathrm{QA}+}(q, C; X) = \sum_{c \in C} \sum_{x \in X} \big[d(q, c) - d(q, x) \big]_+. \tag{11}$$

Unlike **CA**, **QA** perturbs the *query* image, and hence may drastically change its semantics, resulting in abnormal retrieval results. For instance, after perturbing a "lamp" query image, some unrelated candidates (*e.g.*, "shelf", "toaster",

etc) may appear in the top return list. Thus, an ideal query attack should preserve the query semantics, *i.e.*, the candidates in $X \setminus C$ [2] should retain their original ranks if possible. Thus, we propose the Semantics-Preserving Query Attack (**SP-QA**) by adding the **SP** term to mitigate the semantic changes q, *e.g.*,

$$L_{\text{SP-QA-}}(q, C; X) = L_{\text{QA-}}(q, C; X) + \xi L_{\text{QA+}}(q, C_{\text{SP}}; X), \tag{12}$$

where $C_{\text{SP}} = \big\{c \in X \setminus C | \text{Rank}_{X \setminus C}(q, c) \leqslant G\big\}$, *i.e.*, C_{SP} contains the top-G most-relevant candidates corresponding to q, and the $L_{\text{QA+}}(q, C_{\text{SP}}; X)$ term helps preserve the query semantics by retaining some C_{SP} candidates in the retrieved ranking list. Constant G is a predefined integer; and constant ξ is a hyper-parameter for balancing the attack effect and semantics preservation. Unless mentioned, in the following text **QA** means **SP-QA** by default.

3.3 Robustness & Defense

Adversarial defense for classification has been extensively explored, and many of them follows the adversarial training mechanism [25, 33, 45]. In particular, the adversarial counterparts of the original training samples are used to replace or augment the training samples. Until now, Madry defense [45] is regarded as the most effective [71, 2] adversarial training method. However, when directly adapting such classification defense to improve ranking robustness, we empirically discovered a primary challenge of diverging training loss, possibly due to the generated adversarial examples being too "strong". Moreover, such defense mechanism needs to defend against distinct attacks individually. Therefore, a *generic* defense against all the proposed attacks is preferred.

Note that the underlying principle of adversarial ranking attack is to shift the embeddings of candidates/queries to a proper place, and a successful attack depends on a large shift distance as well as a correct shift direction. A large shift distance is an indispensable objective for all the **CA+**, **CA-**, **QA+** and **QA-** attacks. Predictably, a reduction in shift distance could improve model robustness against all attacks simultaneously.

To this end, we propose a "maximum-shift-distance" attack that pushes an embedding vector as far from its original position as possible (resembles Feature Adversary [61] for classification), $r = \arg\max_{r \in \Gamma} d(c + r, c)$. Then we use adversarial examples obtained from this attack to replace original training samples for adversarial training, hence reduce the shift distance incurred by adversarial perturbations.

A ranking model can be normally trained with the defensive version of the triplet loss:

$$\begin{aligned} L_{\text{d-t}}(q, c_p, c_n) = L_{\text{triplet}}\Big(q + \arg\max_{r \in \Gamma} d(q + r, q), c_p + \arg\max_{r \in \Gamma} d(c_p + r, c_p), \\ c_n + \arg\max_{r \in \Gamma} d(c_n + r, c_n)\Big). \end{aligned} \tag{13}$$

[2] The complement of the set C.

ε	CA+				CA-				QA+				QA-			
	$w=1$	2	5	10	$w=1$	2	5	10	$m=1$	2	5	10	$m=1$	2	5	10
(CT) Cosine Distance, Triplet Loss (R@1=99.1%)																
0	50	50	50	50	2.1	2.1	2.1	2.1	50	50	50	50	0.5	0.5	0.5	0.5
0.01	44.6	45.4	47.4	47.9	3.4	3.2	3.1	3.1	45.2	46.3	47.7	48.5	0.9	0.7	0.6	0.6
0.03	33.4	37.3	41.9	43.9	6.3	5.9	5.7	5.6	35.6	39.2	43.4	45.8	1.9	1.4	1.1	1.1
0.1	12.7	17.4	24.4	30.0	15.4	14.9	14.8	14.7	14.4	21.0	30.6	37.2	5.6	4.4	3.7	3.5
0.3	**2.1**	**9.1**	**13.0**	**17.9**	**93.9**	**93.2**	**93.0**	**92.9**	**6.3**	**11.2**	**22.5**	**32.1**	**8.6**	**6.6**	**5.3**	**4.8**

Table 1. Adversarial ranking attack on vanilla model with MNIST. The "+" attacks (*i.e.*CA+ and QA+) raise the rank of chosen candidates towards 0 (%); while the "-" attacks (*i.e.*CA- and QA-) lower the ranks of chosen candidates towards 100 (%). Applying $\varepsilon = 0.01, 0.03, 0.1, 0.3$ QA+ attacks on the model, the SP term keeps the ranks of C_{SP} no larger than $3.6\%, 5.7\%, 7.7\%, 7.7\%$, respectively, regardless of m. With the QA- counterpart, the ranks of C_{SP} are kept no larger than $1.6\%, 1.6\%, 1.5\%, 1.5\%$, respectively, regardless of m. For all the numbers in the table, "%" is omitted.

4 Experiments

To validate the proposed attacks and defense, we use three commonly used ranking datasets including MNIST [34], Fashion-MNIST [75], and Stanford Online Product (SOP) [53]. We respectively train models on these datasets with PyTorch [59], and conduct attacks[3] on their corresponding test sets (used as X).

Evaluation Metric. Adversarial ranking attack aims to change the ranks of candidates. For each candidate c, its *normalized* rank is calculated as $R(q,c) = \frac{\text{Rank}_X(q,c)}{|X|} \times 100\%$ where $c \in X$, and $|X|$ is the length of full ranking list. Thus, $R(q,c) \in [0,1]$, and a top ranked c will have a small $R(q,c)$. The attack effectiveness can be measured by the magnitude of change in $R(q,c)$.

Performance of Attack. To measure the performance of a single CA attack, we average the rank of candidate c across every query $q \in Q$, *i.e.*, $R_{\text{CA}}(c) = \sum_{q \in Q} R(q,c)/w$. Similarly, the performance of a single QA attack can be measured by the average rank across every candidate $c \in C$, *i.e.*, $R_{\text{QA}}(q) = \sum_{c \in C} R(q,c)/m$. For the overall performance of an attack, we conduct T times of independent attacks and report the mean of $R_{\text{CA}}(c)$ or $R_{\text{QA}}(q)$, accordingly.

CA+ & QA+. For CA+, the query set Q is randomly sampled from X. Likewise, for QA+, the candidate set C is from X. Without attack, both the $R_{\text{CA}}(c)$ and $R_{\text{QA}}(q)$ will approximate to 50%, and the attacks should significantly *decrease* the value.

CA- & QA-. In practice, the Q for CA- and the C for QA- cannot be randomly sampled, because the two attacks are often to lower some top ranked candidates. Thus, the two sets should be selected from the top ranked samples (top-1% in our experiments) in X. Formally, given the candidate c for CA-, we randomly sample the w queries from $\{q \in X | R(c,q) \leqslant 1\%\}$ as Q. Given the query q for QA-, m candidates are randomly sampled from $\{c \in X | R(q,c) \leqslant 1\%\}$ as C. Without attack, both the $R_{\text{CA}}(c)$ and $R_{\text{QA}}(q)$ will be close to 0%, and the attacks should significantly *increase* the value.

[3] Specifically, we use PGD without random starts [45].

ε	CA+				CA-				QA+				QA-			
	$w=1$	2	5	10	$w=1$	2	5	10	$m=1$	2	5	10	$m=1$	2	5	10
(CTD) Cosine Distance, Triplet Loss, Defensive (R@1=98.3%)																
0	50	50	50	50	2.0	2.0	2.0	2.0	50	50	50	50	0.5	0.5	0.5	0.5
0.01	48.9	49.3	49.4	49.5	2.2	2.2	2.2	2.1	49.9	49.5	49.5	49.7	0.5	0.5	0.5	0.5
0.03	47.4	48.4	48.6	48.9	2.5	2.5	2.4	2.4	48.0	48.5	49.2	49.5	0.6	0.6	0.5	0.5
0.1	42.4	44.2	45.9	46.7	3.8	3.6	3.5	3.4	43.2	45.0	47.4	48.2	1.0	0.8	0.7	0.7
0.3	**30.7**	**34.5**	**38.7**	**40.7**	**7.0**	**6.7**	**6.5**	**6.5**	**33.2**	**37.2**	**42.3**	**45.1**	**2.4**	**1.9**	**1.6**	**1.5**

Table 2. Adversarial ranking defense with MNIST. Applying $\varepsilon = 0.01, 0.03, 0.1, 0.3$ QA+ attacks on model, the ranks of candidates in C_{SP} are kept no larger than $0.5\%, 0.5\%, 0.5\%, 0.5\%$, respectively, regardless of m. With the QA- counterpart, the ranks of C_{SP} are kept less than $0.4\%, 0.4\%, 0.4\%, 0.4\%$, respectively, regardless of m.

Hyper-Parameters. We conduct CA with $w \in \{1, 2, 5, 10\}$ queries, and QA with $m \in \{1, 2, 5, 10\}$ candidates, respectively. In QA, we let $G = 5$. The SP balancing parameter ξ is set to 1 for QA+ , and 10^2 for QA-. In addition, We investigate attacks of different strength ε, *i.e.* $0.01, 0.03, 0.1, 0.3$ on MNIST and Fashion-MNIST following [45], and $0.01, 0.03, 0.06$ on SOP following [33]. The PGD step size is empirically set to $\alpha = \min(\max(\frac{\varepsilon}{10}, \frac{1}{255}), 0.01)$, and the number of PGD iterations to $\eta = \min(\max(10, \frac{2\varepsilon}{\alpha}), 30)$. We perform $T = |X|$ times of attack to obtain the reported performance.

Adversarial Defense. Ranking models are trained using Eq. (13) with the strongest adversary following the procedure of Madry defense [45].

4.1 MNIST Dataset

Following conventional settings with the MNIST [34] dataset, we train a CNN ranking model comprising 2 convolutional layers and 1 fully-connected layer. This CNN architecture (denoted as C2F1) is identical to the one used in [45] except for the removal of the last fully-connected layer. Specifically, the ranking model is trained with cosine distance and triplet loss. The retrieval performance of the model is Recall@1=99.1% (R@1), as shown in Tab. 1 in grey highlight.

Attacking results against this vanilla model (*i.e.*, the ranking model which is not enhanced with our defense method) are presented in Tab. 1. For example, a strong **CA+** attack (*i.e.*, $\varepsilon = 0.3$) for $w = 1$ can raise the rank $R_{\text{CA}}(c)$ from 50% to 2.1%. Likewise, the rank of C can be raised to 9.1%, 13.0%, 17.9% for $w = 2, 5, 10$ chosen queries, respectively. On the other hand, a strong **CA-** attack for $w = 1$ can lower the rank $R_{\text{CA}}(c)$ from 2.1% to 93.9%. The results of strong **CA-** attacks for $w = 2, 5, 10$ are similar to the $w = 1$ case.

The results of **QA+** and **QA-** are also shown in Tab. 1. the rank changes with **QA** attacks are less dramatic (but still significant) than **CA**. This is due to the additional difficulty introduced by **SP** term in Eq. (12), and the **QA** attack effectiveness is inversely correlated with ξ. For instance, a strong **QA-** for $m = 1$ can only lower the rank $R_{\text{QA}}(q)$ from 0.5% to 8.6%, but the attacking effect can be further boosted by decreasing ξ. More experimental results are presented in following discussion. In brief, our proposed attacks against the vanilla ranking model is effective.

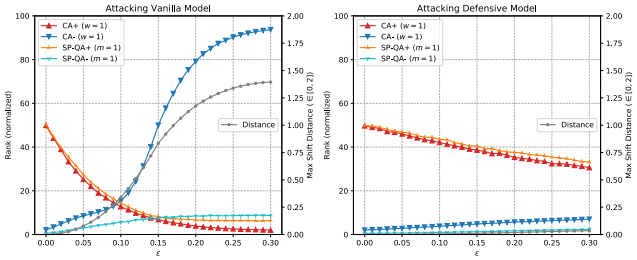

Fig. 2. Comparison of Attacks on vanilla and defensive models. Apart from the ranks of chosen candidates, We also measure the maximum shift distance of embedding vectors that adversarial perturbation could incur.

Next, we evaluate the performance of our defense method. Our defense should be able to enhance the robustness of a ranking model, which can be measured by the difference between the attack effectiveness with our defense and the attack effectiveness without our defense. As a common phenomenon of adversarial training, our defense mechanism leads to a slight retrieval performance degradation for unperturbed input (highlighted in blue in Tab. 2), but the attacking effectiveness is clearly mitigated by our defense. For instance, the same strong **CA+** attack for $w = 1$ on the defensive model (*i.e.*, the ranking model which is enhanced by our defense method) can only raise the rank $R_{\mathrm{CA}}(c)$ from 50% to 30.7%, compared to its vanilla counterpart raising to 2.1%. Further analysis suggests that the weights in the first convolution layer of the defensive model are closer to 0 and have smaller variance than those of the vanilla model, which may help resist adversarial perturbation from changing the layer outputs into the local linear area of ReLU [17].

To visualize the effect of our attacks and defense, we track the attacking effect with ε varying from 0.0 to 0.3 on the vanilla and defensive models, as shown in Fig. 2. It is noted that our defense could significantly suppress the maximum embedding shift distance incurred by adversarial perturbation to nearly 0, but the defensive model is still not completely immune to attacks. We speculate the defensive model still has "blind spots" [69] in some local areas that could be exploited by the attacks.

In summary, these results and further experiments suggest that: (1) deep ranking models are vulnerable to adversarial ranking attacks, no matter what loss function or distance metric is selected; (2) vanilla models trained with contrastive loss are more robust than those trained with triplet loss. This is possibly due to contrastive loss explicitly reducing the intra-class embedding variation. Additionally, our defense method could consistently improve the robustness of all these models; (3) Euclidean distance-based models are harder to defend than cosine distance-based ones. Beyond these experiments, we also find that the margin hyper-parameter β of triplet loss and the dimensionality of the embedding space have marginal influences on model robustness.

ε	CA+				CA-				QA+				QA-			
	$w=1$	2	5	10	$w=1$	2	5	10	$m=1$	2	5	10	$m=1$	2	5	10
(CT) Cosine Distance, Triplet Loss (R@1=88.8%)																
0	50	50	50	50	1.9	1.9	1.9	1.9	50	50	50	50	0.5	0.5	0.5	0.5
0.01	36.6	39.9	43.2	44.8	5.6	5.1	4.9	4.8	39.4	42.0	45.3	47.1	2.1	1.6	1.2	1.1
0.03	19.7	25.4	31.7	35.6	15.5	14.8	14.4	14.3	21.7	28.2	35.7	40.6	5.6	4.1	3.3	2.9
0.1	3.7	10.5	17.3	22.7	87.2	86.7	86.3	86.3	7.1	12.4	23.6	32.5	10.9	8.3	6.7	6.0
0.3	**1.3**	**9.4**	**16.0**	**21.5**	**100.0**	**100.0**	**100.0**	**100.0**	**6.3**	**10.8**	**21.8**	**31.7**	**12.6**	**9.4**	**7.5**	**6.6**
(CTD) Cosine Distance, Triplet Loss, Defensive (R@1=79.6%)																
0	50	50	50	50	1.2	1.2	1.2	1.2	50	50	50	50	0.5	0.5	0.5	0.5
0.01	48.9	48.9	49.3	49.3	1.4	1.4	1.4	1.4	49.4	49.9	49.9	50.0	0.5	0.5	0.5	0.5
0.03	47.1	47.9	48.3	48.3	2.0	1.9	1.8	1.8	48.3	49.1	49.5	49.8	0.7	0.6	0.6	0.6
0.1	42.4	43.5	44.5	44.8	4.6	4.2	4.0	3.9	45.4	47.2	48.7	49.2	1.4	1.2	1.1	1.1
0.3	**32.5**	**35.4**	**37.5**	**38.2**	**11.2**	**10.5**	**10.1**	**10.0**	**39.3**	**42.6**	**46.5**	**47.8**	**3.9**	**3.3**	**3.0**	**2.9**

Table 3. Adversarial ranking attack and defense on Fashion-MNIST. The lowest ranks of C_{SP} are $3.0\%, 5.2\%, 7.8\%, 8.3\%$ in QA+, and $1.9\%, 1.9\%, 1.9\%, 1.8\%$ for QA+, respectively.

4.2 Fashion-MNIST Dataset

Fashion-MNIST [75] is an MNIST-like but more difficult dataset, comprising $60,000$ training examples and $10,000$ test samples. The samples are 28×28 greyscale images covering 10 different fashion product classes, including "T-shirt" and "dress", *etc.* We train the vanilla and defensive models based on the cosine distance and triplet loss and conduct attack experiments.

The attack and defense results are available in Tab. 3. From the table, we note that our attacks could achieve better effect compared to experiments on MNIST. For example, in a strong **CA+** for $w = 1$, the rank $R_{\text{CA}}(c)$ can be raised to 1.3%. On the other hand, despite the moderate improvement in robustness, the defensive model performs worse in unperturbed sample retrieval. The performance degradation is more pronounced on this dataset compared to MNIST. We speculate the differences are related to the increased dataset difficulty.

4.3 Stanford Online Products Dataset

Stanford Online Products (SOP) dataset [53] contains 120k images of 23k classes of real online products from eBay for metric learning. We use the same dataset split as used in the original work [53]. We also train the same vanilla ranking model using the same triplet ranking loss function with Euclidean distance, except that the GoogLeNet [67] is replaced with ResNet-18 [21]. The ResNet-18 achieves better retrieval performance.

Attack and defense results on SOP are present in Tab. 4. It is noted that our attacks are quite effective on this difficult large-scale dataset, as merely 1% perturbation ($\varepsilon = 0.01$) to any candidate image could make it ranked ahead or behind of nearly all the rest candidates (as shown by the CA+ and CA- results with $w = 1$). QA on this dataset is significantly effective as well. On the other hand, our defense method leads to decreased retrieval performance, *i.e.* R@1 from 63.1% to 46.4%, which is expected on such a difficult dataset. Meanwhile, our defense could moderately improve the model robustness against relatively weaker adversarial examples (*e.g.* $\varepsilon = 0.01$), but improving model robustness on this dataset is more difficult, compared to experiments on other datasets.

ε	CA+				CA-				QA+				QA-			
	$w=1$	2	5	10	$w=1$	2	5	10	$m=1$	2	5	10	$m=1$	2	5	10
(ET) Euclidean Distance, Triplet Loss (R@1=63.1%)																
0	50	50	50	50	1.9	1.9	1.9	1.9	50	50	50	50	0.5	0.5	0.5	0.5
0.01	0.0	0.8	2.0	2.6	99.7	99.6	99.4	99.3	4.8	7.0	16.3	25.8	54.9	40.2	27.1	21.9
0.03	0.0	0.3	1.0	1.5	100.0	100.0	100.0	100.0	1.6	3.3	10.0	19.2	68.1	52.4	36.6	30.1
0.06	**0.0**	**0.2**	**1.0**	**1.5**	**100.0**	**100.0**	**100.0**	**100.0**	**1.1**	**2.7**	**8.8**	**17.6**	**73.8**	**57.9**	**40.3**	**32.4**
(ETD) Euclidean Distance, Triplet Loss, Defensive (R@1=46.4%)																
0	50	50	50	50	2.0	2.0	2.0	2.0	50	50	50	50	0.5	0.5	0.5	0.5
0.01	7.5	12.2	16.5	18.0	66.4	62.6	59.3	57.8	16.1	24.8	36.1	41.4	26.7	18.1	12.2	10.2
0.03	0.7	4.5	8.7	10.4	91.7	90.2	89.1	88.4	7.9	14.5	27.2	35.6	43.4	31.7	21.9	18.1
0.06	**0.1**	**3.8**	**7.9**	**9.7**	**97.3**	**96.8**	**96.4**	**96.2**	**6.9**	**12.5**	**24.3**	**33.4**	**51.4**	**39.0**	**28.0**	**23.5**

Table 4. Adversarial ranking attack and defense on SOP. With different ε, the worst ranks of C_{SP} in QA+ are $0.2\%, 0.7\%, 2.0\%, 3.3\%$, and those for QA- are $0.4\%, 0.7\%, 0.8\%, 1.0\%$, respectively.

By comparing the results among all the three datasets, we find ranking models trained on harder datasets more susceptible to adversarial attack, and more difficult to defend. Therefore, we speculate that models used in realistic applications could be easier to attack, because they are usually trained on larger-scale and more difficult datasets.

5 Discussions

White-box attacks are sometimes limited by data accessibility in practice, but it's possible to circumvent them with adversarial example transferability and universal perturbation, as will be discussed in this section. Such properties reveal the possibility of practical black-box attack.

5.1 Adversarial Example Transferability

As demonstrated in the experiments, deep ranking models can be compromised by our white-box attacks. In realistic scenarios, the white-box attacks are not practical enough because the model to be attacked is often unknown (*i.e.*, the architecture and parameters are unknown). On the other hand, adversarial examples for classification have been found transferable [56, 55] (*i.e.*model-agnostic) between different models with different network architectures. Typically, in this case, adversarial examples are generated from a replacement model [56] using a white-box attack, and are directly used to attack the black-box model.

Adversarial ranking attack could be more practical if the adversarial ranking examples have the similar transferability. Besides the C2F1 model, we train two vanilla models on the MNIST dataset: (1) LeNet [34], which has lower model capacity compared to C2F1; (2) ResNet-18 [21] (denoted as Res18), which has a better network architecture and higher model capacity.

The results are present in Tab. 5. For example, in the **CA+** transfer attack, we generate adversarial candidates from the C2F1 model and directly use them to attack the Res18 model (row 2, column 3, top-left table), and the ranks of the adversarial candidates with respect to the same query is still raised to 31.3%. We also find the **CA-** transfer attack is effective, where the ranks of our adversarial

CA+ Transfer (Black Box), $w = 1$

From \ To	LeNet	C2F1	Res18
LeNet	50→16.6	35.1	34.3
C2F1	28.6	50→2.1	31.3
Res18	24.4	27.0	50→2.2

QA+ Transfer (Black Box), $m = 1$

From \ To	LeNet	C2F1	Res18
LeNet	50→20.5	43.0	45.8
C2F1	43.5	50→6.3	45.4
Res18	41.4	40.4	50→14.1

CA- Transfer (Black Box), $w = 1$

From \ To	LeNet	C2F1	Res18
LeNet	2.5→63.7	2.1→10.0	2.1→9.1
C2F1	2.5→9.1	2.1→93.9	2.1→9.3
Res18	2.5→9.9	2.1→11.8	2.1→66.7

QA- Transfer (Black Box), $m = 1$

From \ To	LeNet	C2F1	Res18
LeNet	0.5→7.0	0.5→1.6	0.5→1.8
C2F1	0.5→1.0	0.5→8.6	0.5→1.9
Res18	0.5→0.8	0.5→1.2	0.5→6.9

Table 5. Transferring adversarial ranking examples generated from one model to another. We report the rank of the same c with respect to the same q across different models to illustrate the transfer attack effectiveness. Transferring adversarial examples to a model itself (the diagonal lines) is equivalent to white-box attack.

candidates are lowered, *e.g.* from 2.1% to 9.3% (row 2, column 3, bottom-left table). Similar results can be observed on the **QA** transfer experiments, and they show weaker effect due to the SP term.

From these results, we find that: (1) CNN with better architecture and higher model capacity (*i.e.*, Res18), is less susceptible to adversarial ranking attack. This conclusion is consistent with one of Madry's [45], which claims that higher model capacity could help improve model robustness; (2) adversarial examples generated from the Res18 have the most significant effectiveness in transfer attack; (3) CNN of low model capacity (*i.e.*, LeNet), performs moderately in terms of both adversarial example transferability and model robustness. We speculate its robustness stems from a forced regularization effect due low model capacity. Beyond these, we also noted adversarial ranking examples are transferable disregarding the difference in loss function or distance metric.

Apart from transferability across different architectures, we also investigated the transferability between several independently trained C2F1 models. Results suggest similar transferability between them. Notably, when transferring adversarial examples to a defensive C2F1 model, the attacking effect is significantly mitigated. The result further demonstrates the effectiveness of our defense.

5.2 Universal Perturbation for Ranking

Recently, universal (*i.e.*image-agnostic) adversarial perturbation [49] for classification has been found possible, where a single perturbation may lead to misclassification when added to any image. Thus, we also investigate the existence of universal adversarial perturbation for adversarial ranking attack.

To this end, we follow [49] and formulate the image-agnostic CA+ (*abbr.* **I-CA+**). Given a set of candidates $C = \{c_1, c_2, \ldots, c_m\}$ and a set of queries $Q = \{q_1, q_2, \ldots, q_w\}$, **I-CA+** is to find a *single* universal adversarial perturbation r, so that the rank of every perturbed candidate $\tilde{c} = c + r$ $(c \in C)$ with respect to Q can be raised. The corresponding optimization problem of **I-CA+** is:

$$r = \arg\min_{r \in \Gamma} \sum_{c \in C} L_{\text{CA+}}(c + r, Q; X). \tag{14}$$

CA+	CA-	QA+	QA-
50 → 2.1	2.1 → 93.9	50 → 0.2	0.5 → 94.1
I-CA+	I-CA-	I-QA+	I-QA-
50 → 18.1	0.6 → 9.5	50 → 20.5	2.1 → 7.6
I-CA+ (unseen)	I-CA- (unseen)	I-QA+ (unseen)	I-QA- (unseen)
50 → 18.5	0.7 → 9.4	50 → 21.0	2.2 → 7.4

Table 6. Universal Adversarial Perturbation for Ranking on MNIST. Each pair of results presents the original rank of chosen candidates and that after adding adversarial perturbation. Both w, m are set to 1. Parameter ξ is set to 0 to reduce attack difficulty.

When applied with such universal perturbation, the rank of any candidate *w.r.t* Q is expected to be raised. The objective functions of **I-CA-**, **I-QA+** and **I-QA-** can be obtained in similar way. Note, unlike [36] which aims to find universal perturbation that can make image retrieval system return irrelevant results, our universal perturbations have distinct purposes.

We conduct experiment on the MNIST dataset. For **I-CA+** attack, we randomly sample 5% of X for generating the universal perturbation. Following [49], another non-overlapping 5% examples are randomly sampled from X to test whether the generated perturbation is generalizable on "unseen" (*i.e.*, not used for generating the perturbation) images. Experiments for the other image-agnostic attacks are conducted similarly. Note, we only report the **I-CA-** and **I-QA-** effectiveness on the 1% top ranked samples, similar to **CA-** and **QA-**.

As shown in Tab. 6, our **I-CA** can raise the ranks of C to 18.1%, or lower them to 9.5%. When added to "unseen" candidate images, the universal perturbation could retain nearly the same effectiveness, possibly due to low intra-class variance of the MNIST dataset.

6 Conclusion

Deep ranking models are vulnerable to adversarial perturbations that could intentionally change the ranking result. In this paper, the *adversarial ranking attack* that can compromise deep ranking models is defined and implemented. We also propose an *adversarial ranking defense* that can significantly suppress embedding shift distance and moderately improve the ranking model robustness. Moreover, the transferability of our adversarial examples and the existence of universal adversarial perturbations for ranking attack illustrate the possibility of practical black-box attack and potential risk of realistic ranking applications.

Acknowledgments This work was supported partly by National Key R&D Program of China Grant 2018AAA0101400, NSFC Grants 61629301, 61773312, 61976171, and 61672402, China Postdoctoral Science Foundation Grant 2019M653642, Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001, and Natural Science Foundation of Shaanxi Grant 2020JQ-069.

References

1. Athalye, A., Carlini, N.: On the robustness of the cvpr 2018 white-box adversarial example defenses. arXiv preprint arXiv:1804.03286 (2018)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
3. Bui, T., Ribeiro, L., Ponti, M., Collomosse, J.: Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. CVIU **164**, 27–37 (2017)
4. Carlini, N., Wagner, D.: Defensive distillation is not robust to adversarial examples. arXiv preprint arXiv:1607.04311 (2016)
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE (2017)
6. Chechik, G., Sharma, V., Shalit, U., Bengio, S.: Large scale online learning of image similarity through ranking. JMLR **11**(Mar), 1109–1135 (2010)
7. Chen, J., Jordan, M.I.: Boundary attack++: Query-efficient decision-based adversarial attack. arXiv preprint arXiv:1904.02144 (2019)
8. Chen, P.Y., Sharma, Y., Zhang, H., Yi, J., Hsieh, C.J.: Ead: elastic-net attacks to deep neural networks via adversarial examples. In: AAAI (2018)
9. Croce, F., Hein, M.: Sparse and imperceivable adversarial attacks. In: ICCV. pp. 4724–4732 (2019)
10. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: CVPR (June 2018)
11. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: CVPR. pp. 4312–4321 (2019)
12. Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., Zhu, J.: Efficient decision-based black-box adversarial attacks on face recognition. In: CVPR. pp. 7714–7722 (2019)
13. Dong, Y., Su, H., Zhu, J., Bao, F.: Towards interpretable deep neural networks by leveraging adversarial examples. arXiv preprint arXiv:1708.05493 (2017)
14. Dubey, A., Maaten, L.v.d., Yalniz, Z., Li, Y., Mahajan, D.: Defense against adversarial images using web-scale nearest-neighbor search. In: CVPR. pp. 8767–8776 (2019)
15. Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: Vse++: Improved visual-semantic embeddings. arXiv preprint arXiv:1707.05612 **2**(7), 8 (2017)
16. Ganeshan, A., Babu, R.V.: Fda: Feature disruptive attack. In: ICCV. pp. 8069–8079 (2019)
17. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
18. Gopinath, D., Katz, G., Pasareanu, C.S., Barrett, C.: Deepsafe: A data-driven approach for checking adversarial robustness in neural networks. arXiv preprint arXiv:1710.00486 (2017)
19. Goren, G., Kurland, O., Tennenholtz, M., Raiber, F.: Ranking robustness under adversarial document manipulations. In: ACM SIGIR. pp. 395–404. ACM (2018)
20. Guo, C., Rana, M., Cisse, M., Van Der Maaten, L.: Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117 (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (June 2016)

22. He, W., Wei, J., Chen, X., Carlini, N., Song, D.: Adversarial example defense: Ensembles of weak defenses are not strong. In: 11th USENIX Workshop on Offensive Technologies (WOOT 17) (2017)
23. He, X., He, Z., Du, X., Chua, T.S.: Adversarial personalized ranking for recommendation. In: ACM SIGIR. pp. 355–364. ACM (2018)
24. Huang, Q., Gu, Z., Katsman, I., He, H., Pawakapan, P., Lin, Z., Belongie, S., Lim, S.N.: Intermediate level adversarial attack for enhanced transferability. arXiv preprint arXiv:1811.08458 (2018)
25. Huang, R., Xu, B., Schuurmans, D., Szepesvári, C.: Learning with a strong adversary. CoRR **abs/1511.03034** (2015), http://arxiv.org/abs/1511.03034
26. Jacob, P., Picard, D., Histace, A., Klein, E.: Metric learning with horde: High-order regularizer for deep embeddings. In: ICCV. pp. 6539–6548 (2019)
27. Joachims, T.: Optimizing search engines using clickthrough data. In: ACM SIGKDD. pp. 133–142. ACM (2002)
28. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient smt solver for verifying deep neural networks. In: International Conference on Computer Aided Verification. pp. 97–117. Springer (2017)
29. Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014)
30. Komkov, S., Petiushko, A.: Advhat: Real-world adversarial attack on arcface face id system. arXiv preprint arXiv:1908.08705 (2019)
31. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS. pp. 1097–1105 (2012)
32. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
33. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)
34. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
35. Lee, K.H., Chen, X., Hua, G., Hu, H., He, X.: Stacked cross attention for image-text matching. In: ECCV. pp. 201–216 (2018)
36. Li, J., Ji, R., Liu, H., Hong, X., Gao, Y., Tian, Q.: Universal perturbation attack against image retrieval. In: ICCV. pp. 4899–4908 (2019)
37. Liu, H., Ji, R., Li, J., Zhang, B., Gao, Y., Wu, Y., Huang, F.: Universal adversarial perturbation via prior driven uncertainty approximation. In: ICCV. pp. 2941–2949 (2019)
38. Liu, T.Y., et al.: Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval **3**(3), 225–331 (2009)
39. Liu, X., Cheng, M., Zhang, H., Hsieh, C.J.: Towards robust neural networks via random self-ensemble. In: ECCV. pp. 369–385 (2018)
40. Liu, X., Li, Y., Wu, C., Hsieh, C.J.: Adv-bnn: Improved adversarial defense through robust bayesian neural network. arXiv preprint arXiv:1810.01279 (2018)
41. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770 (2016)
42. Liu, Z., Zhao, Z., Larson, M.: Who's afraid of adversarial queries?: The impact of image modifications on content-based image retrieval. In: ICMR. pp. 306–314. ACM (2019)
43. Lu, J., Issaranon, T., Forsyth, D.: Safetynet: Detecting and rejecting adversarial examples robustly. In: ICCV. pp. 446–454 (2017)
44. Luo, Y., Boix, X., Roig, G., Poggio, T., Zhao, Q.: Foveation-based mechanisms alleviate adversarial examples. arXiv preprint arXiv:1511.06292 (2015)

45. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
46. Mao, C., Zhong, Z., Yang, J., Vondrick, C., Ray, B.: Metric learning for adversarial robustness. In: NeurIPS. pp. 478–489 (2019)
47. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: ACM SIGSAC. pp. 135–147. ACM (2017)
48. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267 (2017)
49. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: CVPR. pp. 1765–1773 (2017)
50. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: CVPR. pp. 2574–2582 (2016)
51. Mummadi, C.K., Brox, T., Metzen, J.H.: Defending against universal perturbations with shared adversarial training. In: ICCV. pp. 4928–4937 (2019)
52. Niu, Z., Zhou, M., Wang, L., Gao, X., Hua, G.: Hierarchical multimodal lstm for dense visual-semantic embedding. In: ICCV. pp. 1881–1889 (2017)
53. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: CVPR. pp. 4004–4012 (2016)
54. Papernot, N., McDaniel, P.: On the effectiveness of defensive distillation. arXiv preprint arXiv:1607.05113 (2016)
55. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277 (2016)
56. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. pp. 506–519. ACM (2017)
57. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 372–387. IEEE (2016)
58. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP). pp. 582–597. IEEE (2016)
59. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. None (2017)
60. Prakash, A., Moran, N., Garber, S., DiLillo, A., Storer, J.: Deflecting adversarial attacks with pixel deflection. In: CVPR. pp. 8571–8580 (2018)
61. Sabour, S., Cao, Y., Faghri, F., Fleet, D.J.: Adversarial manipulation of deep representations. arXiv preprint arXiv:1511.05122 (2015)
62. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR. pp. 815–823 (2015)
63. Shaham, U., Yamada, Y., Negahban, S.: Understanding adversarial training: Increasing local stability of supervised models through robust optimization. Neurocomputing **307**, 195–204 (2018)
64. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: ACM SIGSAC. pp. 1528–1540. ACM (2016)
65. Shi, Y., Wang, S., Han, Y.: Curls & whey: Boosting black-box adversarial attacks. arXiv preprint arXiv:1904.01160 (2019)

66. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation (2019)
67. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. pp. 1–9 (2015)
68. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. pp. 2818–2826 (2016)
69. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
70. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: CVPR. pp. 1386–1393 (2014)
71. Wang, J., Zhang, H.: Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. In: ICCV. pp. 6629–6638 (2019)
72. Wang, Z., Zheng, S., Song, M., Wang, Q., Rahimpour, A., Qi, H.: advpattern: Physical-world attacks on deep person re-identification via adversarially transformable patterns. In: ICCV. pp. 8341–8350 (2019)
73. Wu, L., Zhu, Z., Tai, C., et al.: Understanding and enhancing the transferability of adversarial examples. arXiv preprint arXiv:1802.09707 (2018)
74. Xiao, C., Zhu, J.Y., Li, B., He, W., Liu, M., Song, D.: Spatially transformed adversarial examples. arXiv preprint arXiv:1801.02612 (2018)
75. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
76. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial examples with input diversity. In: CVPR. pp. 2730–2739 (2019)
77. Yuan, X., He, P., Zhu, Q., Li, X.: Adversarial examples: Attacks and defenses for deep learning. IEEE TNNLS (2019)
78. Zhong, Y., Deng, W.: Adversarial learning with margin-based triplet embedding regularization. In: ICCV. pp. 6549–6558 (2019)