Adversarial Attack and Defense in Deep Ranking

Mo Zhou, *Student Member, IEEE,* Le Wang, *Senior Member, IEEE,* Zhenxing Niu, *Member, IEEE,* Qilin Zhang, *Member, IEEE,* Nanning Zheng, *Fellow, IEEE,* and Gang Hua, *Fellow, IEEE*

Abstract— Deep Neural Network classifiers are vulnerable to adversarial attacks, 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. Then a triplet-like objective function is designed to obtain the optimal perturbation. Conversely, an anti-collapse triplet defense is proposed to improve the ranking model robustness against all proposed attacks, where the model learns to prevent the adversarial attack from pulling the positive and negative samples close to each other. To comprehensively measure the empirical adversarial robustness of a ranking model with our defense, we propose an empirical robustness score, which involves a set of representative attacks against ranking models. Our adversarial ranking attacks and defenses are evaluated on MNIST, Fashion-MNIST, CUB200-2011, CARS196, and Stanford Online Products datasets. Experimental results demonstrate that our attacks can effectively compromise a typical deep ranking system. Nevertheless, our defense can significantly improve the ranking system's robustness and simultaneously mitigate a wide range of attacks.

Index Terms—Deep Ranking, Deep Metric Learning, Adversarial Attack, Adversarial Defense, Ranking Model Robustness.

1 Introduction

Despite the successful application in computer vision tasks such as image classification [1], Deep Neural Networks (DNNs) have been found vulnerable to adversarial attacks. In particular, the DNN's prediction can be arbitrarily changed by applying an imperceptible perturbation to the input image [2], [3]. Moreover, such adversarial attacks can effectively compromise the recent state-of-the-art DNNs such as Inception [4] and ResNet [1]. This poses a serious security risk on many DNN-based applications such as face recognition, where recognition evasion and impersonation can be easily achieved [5], [6], [7].

Previous adversarial attacks primarily focus on *classification*. However, we speculate that DNN-based image ranking systems [8], [9] also suffer from a similar vulnerability. Taking the image-based product search as an example, a fair ranking system should rank the database 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 own product by adding perturbation to the image (CA+, row 2), or lower the rank of their competitor's product (CA-, row 3); Besides, a "man-in-the-middle" attacker (*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 image classifiers where labels are predicted independently, a ranking model determines a candidate's rank according to the query as well as all other candidates based on their relative distance. Therefore, the existing adversarial attacks against the DNN classifier are incompatible with deep ranking models. To this end, we need to study the *adversarial ranking attack* thoroughly.

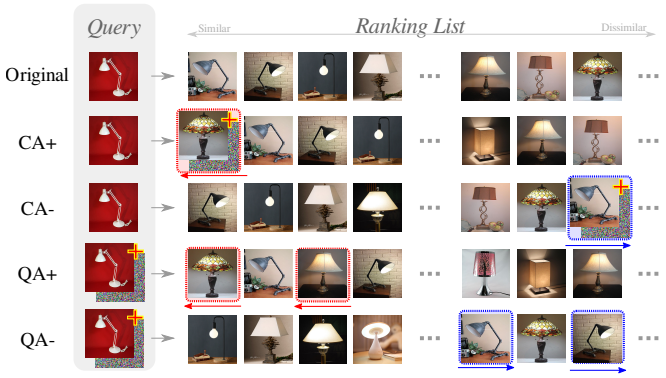


Fig. 1. Adversarial ranking attack that can *raise* or *lower* the rank of the chosen candidates by adversarial perturbations. In Candidate Attack, adversarial perturbation is added to the candidate image, and its rank will be *raised* (CA+) or *lowered* (CA-). In Query Attack, adversarial perturbation is added to the query image, making the ranks of the chosen candidates to be *raised* (QA+) or *lowered* (QA-). An ideal robust ranking model should be resistant to any of these attacks.

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, the adversarial ranking attack can be achieved by performing CA

- *Mo Zhou, Le Wang, and Nanning Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. Email: cdluminate@gmail.com, {lewang,nnzheng}@mail.xjtu.edu.cn. (Corresponding author: Le Wang.)*
- *Zhenxing Niu is with School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi 710071, China. Email: zhenxingniu@gmail.com.*
- *Qilin Zhang is with Apple, Cupertino, CA 95014, USA. This research work was carried out before his joining of Apple. Email: samqzhang@gmail.com.*
- *Gang Hua is with Wormpex AI Research, Bellevue, WA 98004, USA. Email: ganghua@gmail.com.*

on each $c \in C$, or QA on each $q \in Q$. In practice, the choice of CA or QA depends on the accessibility of the candidate or query, respectively. Namely, CA is feasible for a modifiable candidate, while QA is feasible for a modifiable query.

An effective implementation of these attacks is proposed in this paper. As known, a typical deep ranking model maps samples (*i.e.*, queries and candidates) to a common embedding space, where their distances determine the final ranking order. Predictably, a sample's position in the embedding space will be changed by adding a perturbation. Therefore, the essence of the adversarial ranking attack is to find a proper perturbation, which could push the sample to a desired position that leads to the expected ranking order. Specifically, we first represent the expected ranking order as a set of inequalities. Subsequently, a triplet-like objective function is designed according to those inequalities, which can be combined with Projected Gradient Descent (PGD) to obtain the desired adversarial perturbation efficiently.

As opposed to these attacks, the *adversarial ranking defense* is worth investigating, especially for security-sensitive applications. Until now, adversarial training [10] remains one of the most effective defense methods for classification. However, the defense for deep ranking remains almost uncharted. Moreover, we empirically discover the direct adaptation of adversarial training [10] suffers from model collapse [11]. Thus, a new defense for deep ranking needs to be designed.

To this end, an *Embedding-Shifted Triplet* (EST) defense is proposed to defend against all attacks simultaneously. Note that shifting the embedding position of a sample is the key to any ranking attack. Although different attacks prefer distinct shift directions (*e.g.*, CA+ and CA- often prefer opposed shifting directions), a notable shift distance is usually required. If the shift distance of embeddings incurred by adversarial perturbation can be reduced, all attacks are expected to be simultaneously defended. Specifically, we first create adversarial examples with a maximized shift distance off their original locations in the embedding space. Then, the original training samples are directly replaced with their corresponding adversarial examples during adversarial training. Although this defense can moderately improve the model robustness, it suffers from misleading gradient and inefficient mini-batch exploitation, which collectively lead to slow convergence and poor generalization.

To address the problems identified from EST, and further improve ranking model robustness, we propose another adversarial training defense method named "Anti-Collapse Triplet" (ACT). In particular, for each sample triplet (anchor, positive, negative), the positive and negative samples are pulled close to each other via adversarial attack, while the model learns to separate them. Thus, the ranking model is forced to learn robust representations [12] to differentiate different samples better, lest the adversarial attack collapses them together again. This leads to a significant performance improvement over EST defense.

In practice, a deep ranking model has zero prior knowledge of the type of adversarial attack it will confront. Thus, the generic robustness against all types of known attacks is important for a practical defense for real-world applications. This also requires a defense not to be coupled with any specific attack. In this paper, we also propose an Empirical Robustness Score (ERS) for deep ranking models, which is absent from the literature. It involves evaluating a model with a group of adversarial attacks representative of all existing attacks against deep ranking, including but not limited to the proposed CA and QA.

Experimental results on five datasets manifest that our proposed CA and QA can significantly compromise a deep ranking model and successfully achieve the attack goals. Besides, our proposed EST defense can moderately improve model robustness. The new ACT defense significantly outperforms the EST defense in terms of adversarial robustness against a wide range of attacks (hence achieving a high ERS) and generalization performance on benign (*i.e.*, unperturbed) samples. Thus, ranking models with our ACT defense are generically robust, as such model is already resistant to a wide range of white-box attacks.

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

1) The adversarial ranking attack is defined and implemented, which can intentionally change the ranking result and raise or lower the ranks of a set of selected candidates.
2) Two adversarial ranking defense methods are proposed to improve the ranking model robustness and mitigate all the proposed attacks simultaneously.
3) A comprehensive empirical adversarial robustness evaluation metric for deep ranking models is proposed.

This paper is an extension to the previous conference paper [13]. The new major contributions or changes include:

1) An Anti-Collapse Triplet (ACT) defense, which achieves $60\% \sim 540\%$ robustness improvement on all datasets compared to the EST defense and generalizes better.
2) An Empirical Robustness Score (ERS) to comprehensively evaluate the robustness of deep ranking models. To the best of our knowledge, this is the first work of such kind.
3) Experiment and discussion sections are re-written to better focus on defense and adversarial robustness. Our experimental settings are adjusted to be more compatible with existing related deep metric learning works.

This paper is self-contained, and thus potential readers can skip reading the conference version [13].

2 Related Works

Deep Ranking is generally formulated as a deep metric learning (DML) problem [14], [15], which is important for a wide range of tasks such as image retrieval [9], cross-modal retrieval [16], [17], and face/person recognition [11], [18]. Different from the traditional "learning to rank" [19] methods, deep ranking methods embed samples into a common space, and subsequently determine the ranking order based on a defined distance metric. Recent works in DML mainly focus on loss functions, such as triplet loss [11], [9], lifted structured loss [20], and margin loss [21]; or data mining methods, such as semi-hard [11] and distance-weighted [21] mining. More details can be found in surveys [14], [15].

Adversarial Attack. Szegedy *et al.* [2] find DNN susceptible to imperceptible adversarial due to its intriguing "blind spot" property. Then Goodfellow *et al.* [3] attribute this to the "local linearity" of neural networks. Following

these, stronger white-box (*i.e.*, model details such as model architecture and parameters are known to the adversary) attacks [22], [23], [10], [24], [25], [26] are proposed to effectively compromise the state-of-the-art DNN classifiers. Nevertheless, all the white-box attacks rely on the white-box assumption, which can be easily broken in practice. Thus, recent works usually use white-box attacks for robustness evaluation [24]. However, adversarial examples are transferable [27], [28], [29], [30], [31] among different models, which can be seen as model-agnostic. Similarly, image-agnostic universal perturbations are also discovered [32], [33]. These attacks are more practical than white-box attacks when the accessible information is limited. In the extreme case where only the logits or the prediction labels are known, score-based and decision-based black-box attacks are still feasible and effective [34] in overcoming the practical limitations of inaccessible gradients. Moreover, It is even possible to create physical adversarial examples [35], [36], [7].

Adversarial Attack in Deep Ranking. For information retrieval systems, the risk of malicious ranking manipulation consistently exists [37], [38], and so does deep ranking. Some existing attacks against deep ranking aim to incur mismatching top-ranked items [39], [40], [41] as long as they mismatch with the expected ones. The others lead to more specific ranking results [13], [42], [7], [43] beyond a mere mismatch. These attacks will be reviewed in detail in Section 5 in terms of adversarial robustness evaluation.

Adversarial Defense consistently engages in an arms race with adversarial attacks [34], [44], because attacks aim to make models perform worse, while defenses aim to make models retain the original performance when the model is attacked. Various defense methods are proposed to counter the attacks but are continuously compromised by newer attacks. As a simple and straightforward method, gradient masking-based defenses can be circumvented [45]. Defensive distillation [46] is proposed but subsequently compromised by C&W [23]. An ensemble of weak defenses is insufficient [47] against adversarial examples. Other types of defenses may involve input transformation [48], input reconstruction [49], input replacement [50], randomization [51], [52], and feature denoising [53], but many of them are still susceptible to adaptive attack [44]. As an early defense [2], adversarial training [3], [10], [54], [55], [56], [57] remains very effective [34] to date. The existing works for adversarial defense or adversarial robustness mostly focus on the classification task. They can not be easily adopted in other tasks, such as deep ranking.

Adversarial Defense in Deep Ranking remains mostly uncharted. In this paper, we identify problems in the defense [13] proposed in the conference version of this paper and attempt to eliminate the limitations. Meanwhile, we propose an Empirical Robustness Score to evaluate the robustness of the deep ranking model.

3 Adversarial Ranking Attack

Generally, a DNN-based ranking task could be formulated as a metric learning problem [8]. Given the query q and candidate set $X = \{c_1, c_2, \ldots, c_n\}$, deep ranking aims to learn a mapping f, which is usually implemented by a DNN to map all candidates and the 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 the candidate c_j, it is encouraged for the mapping f to satisfy the inequality $\|f(q) - f(c_i)\| < \|f(q) - f(c_j)\|$, where $\|\cdot\|$ denotes ℓ_2 norm (where embedding vectors are projected onto the unit hypersphere following common practice [14]). For brevity, we denote the Euclidean distance $\|f(q) - f(c_i)\|$ as $d(q, c_i)$.

Therefore, the adversarial ranking attack should find a proper adversarial perturbation that changes the ranking order as expected. For example, if a less relevant c_j is expected to be ranked *ahead* of a relevant c_i, a proper perturbation r should be found 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 following text, 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.*, $\tilde{c} = c + r$. The size of the set Q is w.

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

$$\begin{aligned} r &= \underset{r \in \Gamma}{\operatorname{argmin}} \sum_{q \in Q} \text{Rank}_X(q, c + r), \\ \Gamma &= \{r \mid \|r\|_\infty \leqslant \varepsilon; r, c + r \in [0, 1]^N\}, \end{aligned} \tag{1}$$

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 [58]; ℓ_1 [59]; and ℓ_2 [23] variants), we use the ℓ_∞ constraint following [3], [22], [10].

However, the optimization problem Eq. (1) cannot be directly solved due to the discrete nature of the rank value $\text{Rank}_X(q, c)$. Instead, a surrogate objective is needed.

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

$$L_{\text{trip}}(q, c_p, c_n) = [\beta + d(q, c_p) - d(q, c_n)]_+ , \tag{2}$$

where $[\cdot]_+$ denotes $\max(0, \cdot)$, and β is a pre-defined margin constant. This loss function is widely known as the triplet ranking loss [8], [9], [11].

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,

$$L_{\text{CA+}}(\tilde{c}, Q; X) = \sum_{q \in Q} \sum_{x \in X} \left[d(q, \tilde{c}) - d(q, x)\right]_+. \tag{3}$$

Note, the margin β of Eq. (2) is unnecessary for CA+ loss, because $\text{Rank}_X(q, c_n)$ will be less than $\text{Rank}_X(q, c_p)$ as long

as $d(q, c_n) < d(q, c_p)$. Thus, as a zero margin is already satisfactory for our goal, the notation β is omitted. It will be omitted in other similar equations for the same reason.

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

$$r = \underset{r \in \Gamma}{\operatorname{argmin}}\, L_{\text{CA+}}(c + r, Q; X). \tag{4}$$

To solve the optimization problem, Projected Gradient Descent (PGD) method [10] (*a.k.a* the iterative version of FGSM [3]) can be used. Note that the PGD is one of the most effective first-order gradient-based algorithms [34], 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. (3); while step two clips the result of step one to fit in the ε-neighboring region Γ:

$$\tilde{c}_{t+1} = \text{Clip}_{c,\Gamma}\{\tilde{c}_t - \alpha\, \text{sign}(\nabla_{\tilde{c}_t} L_{\text{CA+}}(\tilde{c}_t, Q, X))\}, \tag{5}$$

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 as:

$$L_{\text{CA-}}(\tilde{c}, Q; X) = \sum_{q \in Q} \sum_{x \in X} \big[-d(q, \tilde{c}) + d(q, x) \big]_+. \tag{6}$$

3.2 Query Attack

Query Attack (**QA**) aims 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 an adversarially perturbed query $\tilde{q} = q + r$. The size of the set C is m. Thus, **QA** and **CA** are two "symmetric" attacks. The **QA-** for *lowering* the rank could be formulated as follows:

$$r = \underset{r \in \Gamma}{\operatorname{argmax}} \sum_{c \in C} \text{Rank}_X(q + r, c), \tag{7}$$

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

$$L_{\text{QA-}}(\tilde{q}, C; X) = \sum_{c \in C} \sum_{x \in X} \big[-d(\tilde{q}, c) + d(\tilde{q}, x) \big]_+, \tag{8}$$

$$r = \underset{r \in \Gamma}{\operatorname{argmin}}\, L_{\text{QA-}}(q + r, C; X). \tag{9}$$

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

$$L_{\text{QA+}}(\tilde{q}, C; X) = \sum_{c \in C} \sum_{x \in X} \big[d(\tilde{q}, c) - d(\tilde{q}, x) \big]_+. \tag{10}$$

Unlike **CA**, the **QA** perturbs the *query* q. Hence, **QA** may drastically change its semantics, resulting in abnormal retrieval results. For instance, after perturbing a "lamp" query image, some totally unrelated candidates (*e.g.*, "shelf", "toaster", *etc.*) may appear in the top return list, which is undesired. Thus, an ideal query attack should preserve the query semantics, *i.e.*, the candidates in the set $X \backslash C$ [1] should retain their original ranks if possible. To this end, we propose the Semantics-Preserving Query Attack (**SP-QA**) by adding an **SP** term to suppress the semantic changes of the adversarial query $\tilde{q}$, *i.e.*,

$$L_{\text{SP-QA-}}(\tilde{q}, C; X) = L_{\text{QA-}}(\tilde{q}, C; X) + \xi L_{\text{QA+}}(\tilde{q}, C_{\text{SP}}; X), \tag{11}$$

where $C_{\text{SP}} = \{c \in X \backslash C | \text{Rank}_{X \backslash C}(q, c) \leqslant G\}$, *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. The constant G is a predefined integer; the constant ξ balances the attack effect and semantics preservation.

The SP term in Eq. (11) is expected to be negligible at the beginning of optimization. Nevertheless, the ranks of C_{SP} are prone to be sacrificed later in order to optimize the previous loss term. As drastic changes in the query semantics are strongly undesired, we set ξ as an exponentially changing variable that does not involve in back-propagation, *i.e.*,

$$\xi = \min\Big(10^9,\ \exp\big(\zeta \times L_{\text{QA+}}(\tilde{q}, C_{\text{SP}}; X)\big)\Big). \tag{12}$$

where ζ is a hyper-parameter, the exponential is clipped to 10^9 for numerical stability. This differs from the conference version [13], which employs a constant ξ. This change makes the semantics preservation adaptive and stronger.

4 Defense for Deep Ranking

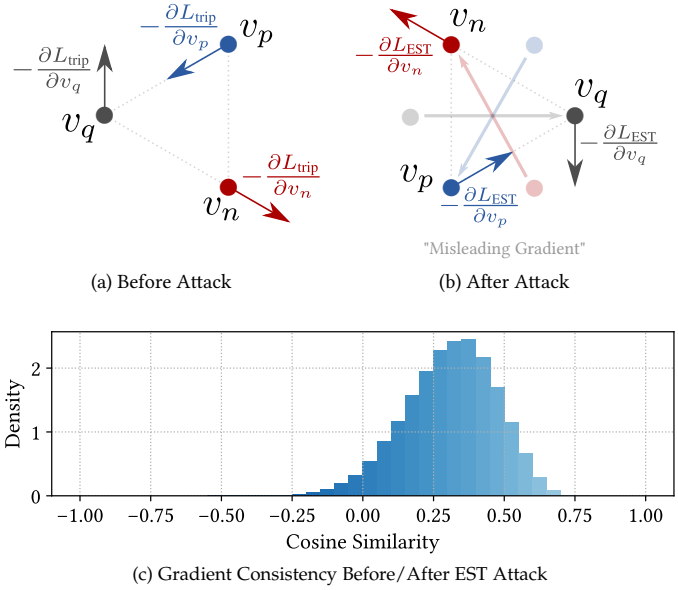
Adversarial training [54], [10] is a commonly used defense for classification. For instance, the Madry defense [10] replaces or augments the original training samples with their adversarial counterparts, which is regarded as one of the most effective [55], [45], [34] defense methods. However, when directly adapting such defense to improve the ranking robustness, we empirically discover a primary challenge of excessively hard (adversarial) training samples causing model collapse [11] and failing to generalize. Therefore, a new *generic* defense for deep ranking is preferred.

4.1 Defense with Embedding-Shifted Triplet

Recall that the principle of an attack against deep ranking is to shift the embedding of a sample to a proper position. Moreover, a successful attack depends on a considerable shift distance and a correct direction. Predictably, as a notable shift distance is indispensable for all types of **CA** and **QA**, a reduction in the embedding shift distance that adversarial perturbation could incur will lead to a more robust model against all of them simultaneously.

To this end, we propose an "Embedding-Shifted Triplet" (EST) defense to adversarially train the ranking model with adversarial examples with the maximum embedding shift, namely the distance off their original locations in the embedding space, *e.g.*, $r = \operatorname{argmax}_{r \in \Gamma} d(q + r, q)$ (resembles Feature Adversary [60] for classification). Then we replace original training samples with such adversarial examples at each training iteration for adversarial training following Madry *et al.* [10]. Once a model can generalize on these

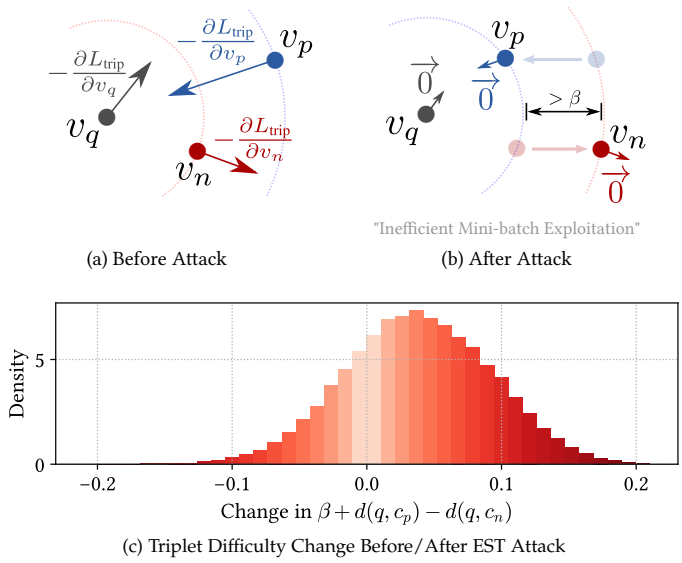
1. The complement of the set C.



(a) Before Attack (b) After Attack

(c) Gradient Consistency Before/After EST Attack

Fig. 2. Misleading Gradient in EST defense [13]. With the samples moving far off their original locations in an arbitrary direction, the loss gradient with respect to the embeddings may point to the wrong sample cluster.



(a) Before Attack (b) After Attack

(c) Triplet Difficulty Change Before/After EST Attack

Fig. 3. Inefficient Mini-batch Exploitation in EST defense [13]. With the samples moving far off their original locations in an arbitrary direction, the initially hard examples from which the model should learn will not be involved in the gradients of the loss function.

adversarial examples, the shift distance that adversarial perturbation can incur is expected to be implicitly reduced.

In brief, a model can be trained with EST as follows:

$$L_{\text{EST}}(q, c_p, c_n) = L_{\text{trip}}\Big(q + \operatorname*{argmax}_{r\in\Gamma} d(q+r, q), \; c_p + \operatorname*{argmax}_{r\in\Gamma} d(c_p+r, c_p), \; c_n + \operatorname*{argmax}_{r\in\Gamma} d(c_n+r, c_n)\Big), \quad (13)$$

which only changes the loss function in the standard deep ranking model training. Empirically, this method can converge without causing model collapse. The idea can also be adapted to other deep metric learning loss functions.

Note that reducing the maximum embedding shift distance caused by adversarial perturbation minimizes a neural network's Lipschitz constant [61]. Not only does Lipschitz constant bound the generalization error of a neural network [62], but also tightly connects with adversarial robustness [63], [64], [65], [66], [67]. These works unanimously focus on classification. Similarly, even if not explicitly built upon the Lipschitz constant, a defense method for deep ranking is expected to indirectly affect the Lipschitz constant of a model, as will be reflected by experiments measuring the embedding shift distance.

4.1.1 Limitations of Embedding-Shifted Triplet

Although EST can moderately improve ranking model robustness, we find it slow to converge and poor in performance on unperturbed examples. Further examination suggests that EST greatly suffers from misleading gradients and inefficient mini-batch exploitation.

We denote the embeddings of a sample triplet (*i.e.*, an anchor, a positive, and a negative sample) as $v_q = f(q)$, $v_p = f(c_p)$, and $v_n = f(c_n)$, respectively. When $L_{\text{trip}}(q, c_p, c_n) > 0$, the gradients of the triplet loss with respect to these sample embeddings v_q, v_p, and v_n are respectively:

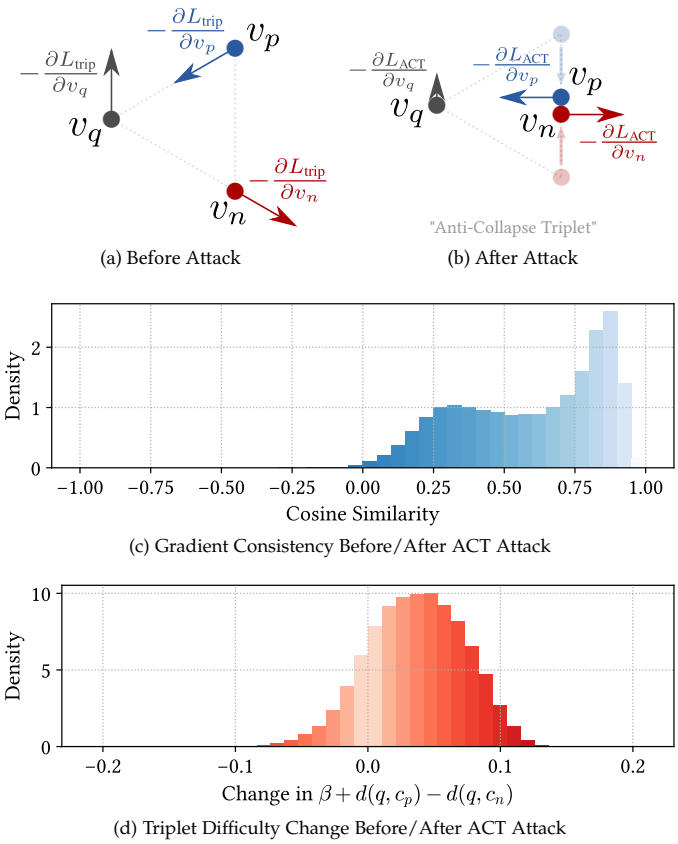
$$\frac{\partial L_{\text{trip}}}{\partial v_q} = \frac{v_q - v_p}{\|v_q - v_p\|} - \frac{v_q - v_n}{\|v_q - v_n\|}, \quad (14)$$

$$\frac{\partial L_{\text{trip}}}{\partial v_p} = \frac{v_p - v_q}{\|v_q - v_p\|}, \quad (15)$$

$$\frac{\partial L_{\text{trip}}}{\partial v_n} = \frac{v_q - v_n}{\|v_q - v_n\|}. \quad (16)$$

Misleading Gradient. The embeddings v_q, v_p, and v_n of adversarial examples are moved off their original positions without any direction restrictions. As a result, the embeddings may be near to the cluster of any other sample class, as shown in Fig. 2. In this case, the gradients may point at wrong directions (*e.g.*, negative gradient of v_n points towards the cluster of v_q and v_p in Fig. 2 (b)). In other words, the gradient vectors can point at "arbitrary" directions as there is no restriction on the shifting directions of embedding vectors. Such "arbitrary" directions can even vary across training iterations. As a result, the embeddings are moving towards "arbitrary" directions during the training process, leading to a prolonged convergence rate and poor generalization on benign (*i.e.*, unperturbed) examples.

Fig. 2 (b) only illustrates an ideal case. To better support the insight about misleading gradient, we measure the gradient consistency of $\partial L_{\text{trip}}/\partial v_q$, $\partial L_{\text{trip}}/\partial v_p$, and $\partial L_{\text{trip}}/\partial v_n$ on the Fashion [68] dataset with a randomly initialized model. The gradient consistency is calculated as the cosine similarity of the gradients before and after the EST attack. Namely, the model will suffer less from misleading gradients when the cosine similarity is closer to 1.0, or suffer more when the cosine similarity is closer to -1.0. By traversing the training dataset for one epoch, we obtain a histogram of the cosine similarity, as shown in Fig. 2 (c). Specifically, the cosine similarity is statistically 0.31 ± 0.16. As the cosine similarity is 1.00 ± 0.00 when there is no attack, it suggests the gradients



(a) Before Attack (b) After Attack

(c) Gradient Consistency Before/After ACT Attack

(d) Triplet Difficulty Change Before/After ACT Attack

Fig. 4. Gradient Direction of Anti-Collapse Triplet (ACT). ACT does not suffer from misleading gradients or inefficient mini-batch exploitation. It does not create excessively hard adversarial examples either.

after EST significantly deviate from the original direction. Note, although the absolute quantity may greatly vary across different datasets and models, a method that can effectively mitigate the "misleading gradient" issue should manifest a relatively large quantity that is close to 1.00. Potential mitigations will be discussed in the following subsection.

Inefficient Mini-batch Exploitation. As maximizing the embedding shift distance is not an adversarially opposite goal to minimizing triplet loss, the adversarial examples can lead to either a larger or smaller loss value. Namely, the initially easy samples with $L_{\text{trip}} = 0$ can be turned into hard samples (with a large loss). Although sufficiently learned by the model, these samples will be moved along "arbitrary" directions, which is unnecessary. Besides, the initially hard samples with a large L_{trip} can be turned into easy samples (with a small loss), as shown in Fig. 3. In this way, the training samples from which it should learn will not be involved in the gradients of the loss function. In brief, the EST cannot help the model efficiently exploit the information from mini-batches. As a result, the model will converge slowly and generalize poorly due to low-quality gradients.

Fig. 3 (b) only illustrates an ideal case. To better support the insight about inefficient mini-batch exploitation, we measure the change in triplet difficulty on the Fashion dataset with a randomly initialized model. The change is calculated as the difference of $\beta + d(q, c_p) - d(q, c_n)$ (equivalent to L_{trip} without the $[\cdot]_+$ operation) after and before the EST attack. The model will suffer more from the inefficiency when the change has a larger variance, because more easy samples will be turned excessively hard, and more hard samples will be turned excessively easy. By traversing the training dataset for one epoch, we obtain a histogram of the triplet difficulty change, as shown in Fig. 3 (c). Specifically, the change is statistically 0.038 ± 0.054. As the change should be 0.000 ± 0.000 when there is no attack, it suggests EST will make many samples excessively harder or simpler based on the observed variance. Note, although the absolute quantity of triplet difficulty change may vary greatly across different datasets and models, a method that can effectively mitigate the "inefficient mini-batch exploitation" issue should manifest a relatively small variance that is close to 0.000. Potential mitigations will be discussed in the following subsection.

In short, we learn that an adversarial training-based defense for deep ranking should be free from misleading gradients and inefficient exploitation of mini-batches. Meanwhile, it should not create excessively hard adversarial examples to trigger ranking model collapse [11]. How can these three conditions be satisfied simultaneously to seek a better defense for deep ranking?

4.1.2 Mitigation of Limitations of EST

To alleviate the problem of misleading gradient, we slightly modify the EST defense into the "Revised EST" (REST) defense, where the query sample q is not replaced with its adversarial counterpart:

$$L_{\text{REST}}(q, c_p, c_n) = L_{\text{trip}}\Big(q, \; c_p + \operatorname*{argmax}_{r \in \Gamma} d(c_p + r, c_p), \; c_n + \operatorname*{argmax}_{r \in \Gamma} d(c_n + r, c_n)\Big). \quad (17)$$

According to Eqs. (14)–(16), the gradients will be stabilized by the v_q of benign example. They will sometimes point at a proper direction (*e.g.*, negative gradient of v_n will not point towards the cluster of v_q and v_p). A positive effect is expected from such a subtle and careful change.

To improve the efficiency of mini-batch exploitation for EST, a simple and straightforward mitigation is to increase the margin hyper-parameter β in the triplet loss. In this way, more samples from which the model should learn will involve in the gradient computation.

In addition, a defense to directly "Suppress the Embedding Shift" (SES)[2] can circumvent both limitations:

$$L_{\text{SES}} = L_{\text{trip}}(q, c_p, c_n) + \sum_{x \in \{q, c_p, c_n\}} \max_{r \in \Gamma} d(x + r, x). \quad (18)$$

All these mitigations will be examined in Section 7.

4.2 Defense with Anti-Collapse Triplet

Instead of mitigating the limitations of EST, we attempt to address them from the root. In this paper, we present a new defense method that adversarially trains deep ranking models with "Anti-Collapse Triplet" (ACT), where the positive and negative sample embeddings are "collapsed" (*i.e.*, pulled close) together through the adversarial attack, and then the

2. This is discussed in the supplementary material of Zhou *et al.* [13].

TABLE 1
Empirical Comparison between EST and ACT in the Gradient Consistency and Triplet Difficulty Change.

Defense	Gradient Consistency		Triplet Difficulty Change	
None	1.00 ± 0.00		0.000 ± 0.000	
EST [13]	0.31 ± 0.16	(Fig. 2(c))	0.038 ± 0.054	(Fig. 3(c))
ACT	0.61 ± 0.25	(Fig. 4(c))	0.038 ± 0.036	(Fig. 4(d))

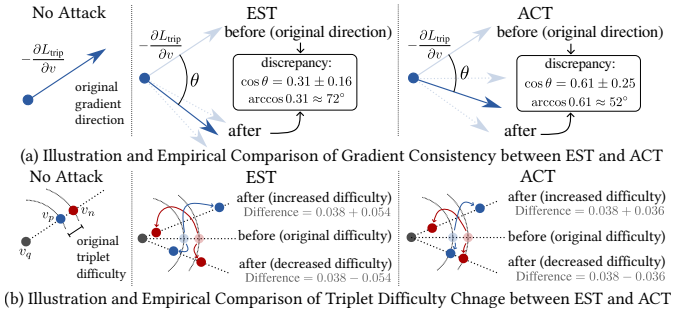


Fig. 5. Illustration and Empirical Comparisons for Quantities in Table 1. (a) A higher mean value of cosine similarity means the gradient of ACT deviates less from the original direction compared to EST, and hence mitigates the "misleading gradients" issue. (b) A smaller variance means ACT refrains from creating excessively hard or simple adversarial example triplets compared to EST, and hence mitigates the "inefficient mini-batch exploitation" issue.

ranking model is trained to separate them with the triplet loss. The ACT satisfies the three conditions in Section 4.1.1.

Specifically, given a triplet (q, c_p, c_n), we first find a pair of adversarial positive and negative examples $(\overrightarrow{c_p}, \overleftarrow{c_n}) = (c_p + \overrightarrow{r_p}, c_n + \overleftarrow{r_n})$, so that their embedding vectors are "collapsed" together (the distance between them is minimized):

$$(\overrightarrow{r_p}, \overleftarrow{r_n}) = \underset{r_p \in \Gamma_{c_p}, r_n \in \Gamma_{c_n}}{\operatorname{argmin}} \|f(c_p + r_p) - f(c_n + r_n)\|. \quad (19)$$

Subsequently, $(\overrightarrow{c_p}, \overleftarrow{c_n})$ and the original query q are fed into the triplet loss function as the ACT defense:

$$L_{\text{ACT}}(q, c_p, c_n) = L_{\text{trip}}(q, \overrightarrow{c_p}, \overleftarrow{c_n}). \quad (20)$$

During the training process, the model is forced to learn robust representations [12] to differentiate and separate the collapsed positive and negative samples, lest they be collapsed again through non-robust representations by the next round of adversarial attack. Meanwhile, the robust feature will also help generalization on benign examples.

Unlike EST, ACT will suffer less from "misleading gradients". According to Eqs. (15)–(16), the negative gradients for v_p and v_n will largely point at a proper direction (*e.g.*, that of v_p points toward v_q of unperturbed query; that of v_n points as opposed to v_q) whether the adversarial attack successfully "collapse" v_p and v_n together or not, because the v_p and v_n are moved along a fixed direction. When the "collapse" is successful, even if the directions of v_p and v_n slightly deflected due to the optimization algorithm (*i.e.*, the projection step of PGD), the norm of gradient for v_q will be negligible ($\partial L_{\text{ACT}}/\partial v_q \approx 0$) regardless of the gradient direction. When $(\overrightarrow{c_p}, \overleftarrow{c_n})$ are merely slightly pulled closer to each other, the model is already somewhat resistant to the attack. Expectedly, the gradient directions will remain approximately correct in most cases during training.

To justify this, we also measure the gradient consistency with ACT, following the same setup in Section 4.1.1. The corresponding histogram is shown in Fig. 4 (c). The cosine similarity is statistically 0.61 ± 0.25. Compared to EST (0.31 ± 0.16), the gradient consistency of ACT is much higher. These quantities are also illustrated and compared in Table 1 and Fig. 5. The results show that ACT will suffer much less from misleading gradients than EST in practice.

The ACT suffers less from inefficient mini-batch exploitation, as shown in Fig. 4. It will turn both easy and hard samples into moderately hard samples (so that $L_{\text{ACT}} > 0$) without creating excessively hard examples because the attack stops when $v_p = v_n$ (where $L_{\text{ACT}} = \beta$). Thus, the gradient quality will not degrade, as the model will not omit any sample from which it should learn. Automatically stopping at $v_p = v_n$ ($L_{\text{ACT}} = \beta$) also means that ACT will be unlikely to incur model collapse [11].

To justify this, we also measure the change in triplet difficulty with ACT. The corresponding histogram is shown in Fig. 4 (d). The change is statistically 0.038 ± 0.036. Compared to EST (0.038 ± 0.054), the change of ACT has a much smaller variance. These quantities are also illustrated and compared in Table 1 and Fig. 5. The results indicate that ACT will refrain from making samples excessively easier or harder compared to EST, and hence suffer less from inefficient mini-batch exploitation in practice.

5 Adversarial Robustness Evaluation

In real-world ranking applications, the ranking model has zero prior knowledge of the exact type of attack it will confront. Thus, a practical defense should not couple with any specific attack. It should be generically robust to a wide range of attacks [12]. As a robustness evaluation metric for deep ranking is absent from the literature, we propose an "Empirical Robustness Score" (ERS) to evaluate the empirical adversarial robustness comprehensively.

In particular, ERS adopts the following attacks:

1) **CA+ (w=1).** CA+ with $w > 1$ will be more difficult, hence showing lower efficacy. Expectedly, a model resistant to CA+ (w=1) will be more resistant to CA+ (w>1). Thus, CA+ (w=1) is representative for the CA+ family. The constant w is the size of set Q as defined in Section 3.1.
2) **CA- (w=1).** It is chosen for similar reasons.
3) **QA+ (m=1).** The semantics-preserving term is discarded to make the attack much easier. Expectedly, a model resistant to QA+ will be more resistant to SP-QA+. The constant m is the size of set C as defined in Section 3.2.
4) **QA- (m=1).** It is chosen for similar reasons.
5) **TMA,** namely the Targeted Mismatch Attack using global descriptor [40], which aims to increase the cosine similarity between $\tilde{q}$ and a randomly chosen "target" q_t:

$$L_{\text{TMA}}(\tilde{q}, q_t) = 1 - f^{\mathsf{T}}(\tilde{q}) f(q_t). \quad (21)$$

6) **ES,** namely the Embedding Shift attack used in the EST defense, where an adversarial query $\tilde{q} = q + \operatorname{argmax}_{r \in \Gamma} d(q, q+r)$ is fed into the model to incur a large embedding shift distance, and sometimes a mismatching top-1 retrieval as well.

7) **LTM,** namely the Learning-To-Misrank [41] attack, which aims to perturb the ranking system output by minimizing the distance of unmatched pairs while maximizing the distance of matched pairs, as follows:

$$L_{\text{LTM}}(\tilde{q}) = [\max_{c_n \in X_n} d(\tilde{q}, c_n) - \min_{c_p \in X_p} d(\tilde{q}, c_p)]_+, \tag{22}$$

where X_n and X_p are the sets containing all candidates of different class and the same class, respectively.

8) **GTM,** a new "Greedy Top-1 Misranking" attack which aims to reduce the distance between the adversarial query $\tilde{q}$ and the most confusing negative sample (*i.e.*, the closest candidate to q in a different class):

$$L_{\text{GTM}}(\tilde{q}) = d\big(\tilde{q}, \operatorname*{argmin}_{c_n \in X_n} d(q, c_n)\big). \tag{23}$$

The efficacy of the attack is measured with Recall@1 as well.

9) **GTT,** a new "Greedy Top-1 Translocation" attack simplified from [43], which aims to move the top-1 candidate out of the top-ranked items with the following objective:

$$L_{\text{GTT}}(\tilde{q}) = L_{\text{QA-}}(\tilde{q}, \{\operatorname*{argmin}_{c \in X} d(q, c)\}; X). \tag{24}$$

Similar to the robustness of a deep classifier [34], that of a deep ranking model can also be reflected by the reduction in the efficacy of the above adversarial attacks. Namely, a robust ranking model should prevent **CA+** and **QA+** from moving selected candidates towards the topmost part of the ranking list; **CA-** and **QA-** from moving selected candidates towards the bottommost part of the ranking list; **TMA** from achieving a high cosine similarity; **ES** from incurring a large embedding shift distance or reducing the recall performance; **LTM** and **GTM** from reducing the recall performance; **GTT** from moving the original top-1 candidate out from the top-k results (extremely difficult with a small k). The concrete evaluation protocol will be detailed in Section 6.

After evaluating each attack against the model, the score of the corresponding attack (*e.g.*, success rate) will be normalized within $[0, 100]$. The detail of score normalization is a part of our evaluation protocol, which is discussed in Section 6.1.4. Finally, the ERS is calculated as the average score across all attacks. Thus, a high ERS is preferred for a robust ranking model resistant to a wide range of attacks (even including unknown attacks).

Although there are related attacks focusing on transferability [41], universal perturbation [39], and even black-box attacks [43], they assume that the model architecture and parameters are inaccessible, which is opposite to the default white-box assumption. Based on the significant difference in the amount of information available for the attacker, finding a transferable, universal, or black-box perturbation with an optionally complicated goal is much more difficult than finding a per-model, per-sample, or white-box perturbation with a simplified goal. Thus, ERS only involves simple white-box attacks representing or simplified from them, but not any attack in a complicated form. When a model is already resistant to the simpler attack, a more complicated attack (with extra loss terms or black-box optimizers) is empirically expected to perform worse. In brief, the performance of the simple white-box attacks is representative of adversarial robustness evaluation. Additional experiments using black-box attacks are provided in Section 7.4 to support this claim.

In the end, we review all existing attacks against deep ranking and discuss which attacks in ERS can represent them for robustness evaluation:

1) QAIR [43] (CVPR'21) is a black-box attack to subvert the top-k retrieval results, where the union between the original top-k samples and the top-k samples with an adversarial query is expected to be an empty set. We simplify QAIR into GTT to examine whether a model can retain the original top-1 sample within the top-k results.
2) Bai *et al.* [69] (T-PAMI'21) propose metric attacks, where the "non-targeted" attack is identical to ES, while the "targeted" attack can be represented by TMA or GTM.
3) Learning-To-Misrank [41] (CVPR'20) is directly adopted as a part of ERS evaluation in its simplest form.
4) DPQN [70] (AAAI'20) presents an attack whose goal formulation is fully identical to that of ES.
5) AdvPattern [7] (ICCV'19) introduces two concepts without their concrete white-box implementations. The "Evading Attack" to demote the rank of the selected candidate can be represented by CA-, QA-, ES, and LTM due to resemblance of the goal. The "Impersonation Attack" to promote a selected candidate's rank while demoting another candidate's rank can be represented by CA+, QA+, TMA, and GTM for the resemblance of goal.
6) Targeted Mismatch Attack [40] (ICCV'19) is directly adopted in its simplest form (with global descriptor) as a part of our ERS evaluation, because it is "suitable when all parameters of the retrieval system are known".
7) Li *et al.* [39] (ICCV'19) propose a universal perturbation to "corrupt as many similarity relationships as possible in the data distribution", which can be represented by ES. Besides, their formulation for "corrupting pair-wise relationship" is very similar to LTM.

A model achieving a high ERS is expected to be robust against all attacks mentioned above.

6 Experiments

To validate the proposed attacks and defenses, and evaluate the ranking models with ERS, we use five ranking datasets including MNIST [71], Fashion-MNIST (Fashion) [68], CUB200-2011 (CUB) [72], CARS196 (CARS) [73], and Stanford Online Product (SOP) [20]. For MNIST and Fashion, we use the same dataset split as in [13]. For CUB, CARS, and SOP, we use the same dataset split as in [14]. Note, the dataset split for CUB and CARS is zero-shot [14], where the classes in the training set do not overlap with those in the test set.

We conduct experiments with Nvidia RTX3090 GPUs and Intel Xeon 6226R CPU. Our PyTorch [74]-based code implementation of the attacks, defenses, and the empirical robustness score is available as a Python library at https://cdluminate.github.io/robrank.

6.1 Evaluation Protocol

6.1.1 Baseline Deep Ranking Model

For MNIST and Fashion, we train a CNN model with 2 convolution layers and 2 fully-connected layers (*abbr.* C2F2),

TABLE 2
Adversarial Ranking Attack & Defense with Ranking Models on Various Datasets.

Dataset	Model	Loss	Defense	Benign Example				ε	CA+ $\uparrow^{(50)}$				CA- $\downarrow_{(0)}$				SP-QA+ $\uparrow^{(50)}$					SP-QA- $\downarrow_{(0)}$				
				R@1↑	R@2↑	mAP↑	NMI↑		w=1	2	5	10	w=1	2	5	10	m=1	2	5	10	$\hat{R}_{\text{GT}}$	m=1	2	5	10	$\hat{R}_{\text{GT}}$
MNIST	C2F2	Triplet	✗	**99.0**	**99.4**	**98.7**	84.7	8/255	41.8	43.7	45.1	45.6	4.9	4.6	4.5	4.5	44.8	46.5	47.9	48.8	0.3	1.7	1.4	1.2	1.1	0.3
								77/255	3.3	10.3	14.1	15.9	69.9	69.6	69.2	69.1	29.2	35.7	41.6	44.8	0.7	2.7	2.2	1.9	1.8	0.7
			✓	98.3	99.0	91.3	80.7	8/255	41.1	41.9	42.2	42.3	3.0	2.7	2.5	2.5	46.2	47.4	48.7	49.0	0.2	1.5	1.3	1.2	1.2	0.4
								77/255	6.8	11.5	14.6	15.8	36.0	33.9	32.2	31.7	32.6	37.9	43.2	45.6	0.7	3.6	2.9	2.4	2.3	0.8
			★	98.6	99.1	98.1	**86.4**	8/255	48.2	48.3	48.9	49.2	2.7	2.7	2.6	2.6	48.8	48.6	49.4	49.6	0.0	0.7	0.6	0.6	0.6	0.0
								77/255	33.4	37.2	41.3	43.9	7.6	7.5	7.5	7.5	36.7	39.8	44.2	46.1	0.4	2.7	2.2	1.9	1.8	0.6
Fashion	C2F2	Triplet	✗	**87.6**	**92.7**	**84.9**	**77.8**	8/255	31.7	35.8	38.1	38.9	12.7	12.4	12.3	12.3	44.5	47.0	48.7	49.0	0.3	1.7	1.4	1.2	1.1	0.3
								77/255	1.0	13.2	20.2	22.8	95.0	95.0	95.0	95.0	40.3	42.6	46.2	47.9	0.7	2.5	2.4	2.2	2.2	0.8
			✓	78.6	86.8	64.6	64.9	8/255	42.0	43.2	43.7	43.9	3.6	3.2	3.0	3.0	46.7	48.3	49.3	49.4	0.4	1.9	1.6	1.4	1.4	0.6
								77/255	12.1	19.3	23.2	24.5	32.7	31.8	31.1	30.9	42.2	44.7	47.1	48.4	0.5	1.9	1.5	1.3	1.3	0.4
			★	79.4	87.9	71.6	69.6	8/255	47.7	48.3	48.9	49.2	2.1	2.0	2.0	2.0	48.6	49.0	49.4	49.7	0.1	0.9	0.8	0.7	0.6	0.1
								77/255	34.7	38.8	42.5	44.6	11.3	10.9	10.9	10.9	43.3	45.4	47.4	48.5	0.3	2.2	1.8	1.5	1.4	0.3
CUB	RN18	Triplet	✗	**53.9**	**66.4**	**26.1**	**59.5**	2/255	0.2	5.5	13.5	17.8	99.6	99.3	98.9	98.7	13.4	23.3	32.2	39.1	0.4	17.2	11.5	7.0	6.8	0.6
								8/255	0.0	5.0	13.2	17.6	100.0	100.0	100.0	100.0	23.5	29.2	37.1	42.1	0.8	11.9	10.8	6.7	6.0	0.8
			✓	8.5	13.0	2.6	25.2	2/255	8.8	13.3	18.8	21.4	79.2	77.2	75.4	74.7	11.4	16.5	26.2	32.6	0.4	40.4	30.4	19.0	13.7	0.5
								8/255	2.7	8.4	14.8	18.0	97.9	97.6	97.2	97.1	13.6	17.6	28.1	33.7	0.5	37.9	28.9	16.6	10.7	0.7
			★	27.5	38.2	12.2	43.0	2/255	39.0	42.1	44.2	45.1	5.4	4.8	4.5	4.5	41.2	44.1	46.1	47.8	0.1	1.9	1.4	1.1	0.9	0.1
								8/255	15.5	21.9	28.8	32.5	37.7	34.1	31.8	31.2	25.9	32.9	39.9	43.4	0.1	7.1	4.0	2.7	2.1	0.2
CARS	RN18	Triplet	✗	**62.5**	**74.0**	**23.8**	**57.0**	2/255	0.5	10.9	21.9	26.6	99.6	99.2	98.8	98.6	31.6	37.6	43.2	45.9	0.3	10.0	7.6	5.6	4.8	0.6
								8/255	0.2	10.6	21.5	26.2	100.0	100.0	99.9	99.8	45.4	46.9	47.8	49.0	0.3	3.8	3.8	2.8	2.7	0.6
			✓	30.7	41.0	5.6	31.8	2/255	18.7	22.4	25.6	27.0	39.0	35.7	33.7	33.2	24.7	31.7	39.6	43.5	0.1	11.6	6.7	3.8	2.9	0.2
								8/255	1.2	3.4	6.7	8.5	98.1	97.5	97.3	97.1	6.9	12.2	22.8	32.7	0.2	44.1	33.8	19.5	12.8	0.4
			★	43.4	54.6	11.8	42.9	2/255	40.2	43.5	45.6	46.9	5.1	4.6	4.4	4.3	42.0	44.7	46.9	48.3	0.1	1.7	1.2	0.9	0.8	0.1
								8/255	18.0	25.6	33.5	37.2	32.2	28.7	27.0	26.3	32.2	37.8	43.0	45.8	0.1	4.8	3.2	2.3	2.0	0.1
SOP	RN18	Triplet	✗	**62.9**	**68.5**	**39.2**	**87.4**	2/255	5.0	10.6	18.4	22.7	55.7	48.3	43.1	41.2	12.5	20.2	30.1	36.6	0.1	18.9	12.8	7.6	6.3	0.3
								8/255	0.1	3.2	10.1	15.0	99.3	98.7	98.1	97.9	19.9	27.6	35.4	39.6	0.2	19.0	14.0	11.0	10.9	1.1
			✓	46.0	51.4	24.5	84.7	2/255	39.8	43.0	45.4	46.4	4.6	3.7	3.4	3.3	37.6	41.1	44.6	46.7	0.0	1.9	1.4	1.0	0.9	0.0
								8/255	12.5	18.5	25.2	28.7	43.6	36.4	32.0	30.7	12.1	19.7	30.2	36.8	0.1	22.2	13.4	8.7	7.1	0.2
			★	47.5	52.6	25.5	84.9	2/255	45.0	47.9	49.4	49.9	1.7	1.5	1.4	1.3	42.4	44.7	46.8	48.0	0.0	1.0	0.8	0.7	0.6	0.0
								8/255	24.1	29.9	36.9	40.1	10.5	7.8	6.3	5.9	22.8	28.8	36.9	41.3	0.0	8.2	5.3	3.6	3.0	0.1

For SP-QA+ and SP-QA-, we denote the worst rank percentile of C_{SP} among four settings of m as $\hat{R}_{\text{GT}}$ (the lower the better). The mark "↑" means larger values are preferred for a robust model, while "↓" means smaller values are preferred for a robust model. The optional superscript or subscript to the arrows means the upper or lower bound of the value, respectively. The definition of defense method marks can be found in Section 6.1.4.

which is the same as the model used in [10] except that the output dimension is changed to D.

For CUB, CARS, and SOP, we train a ResNet-18 [1] (*abbr.* RN18) model with the output dimension of the last fully-connected layer changed as D following [14].

We follow the standard deep ranking model training procedure. All embeddings are projected onto the unit hypersphere [14]. In particular, models are fed with "SPC-2" mini-batches [14], where every mini-batch contains (at least) 2 samples for each sampled class.

The embedding space dimension is set as $D = 512$ for all models. The margin β is set as 0.2 by default following [14]. The C2F2 model is trained with the Adam [75] optimizer for 16 epochs with batch-size set as 128, and a constant learning rate 1.0×10^{-3}. The ResNet-18 model is trained with Adam optimizer for 150 epochs with batch-size 112 following [14], and a constant learning rate 1.0×10^{-5}. The model performance on benign (*i.e.*, unperturbed) examples is evaluated in Recall@1 (R@1), Recall@2 (R@2), mAP, and NMI following [14]. All these metrics are scaled to $[0, 100]$, where higher values are preferred.

6.1.2 Evaluation of Candidate Attack & Query Attack

We conduct attacks on the corresponding test dataset (used as X). For each candidate c, its *normalized* rank (*i.e.*, ranking percentile) is calculated as $R(q, c) = \text{Rank}_X(q,c)/|X| \times 100\%$ where $c \in X$, and $|X|$ is the length of the 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 $R(q, c)$. We omit the percent sign "%" for brevity.

Performance Metric of CA&QA. To measure the performance of a single CA, 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 can be measured by the average rank across every candidate $c \in C$, *i.e.*, $R_{\text{QA}}(q) = \sum_{c \in C} R(q,c)/m$. Then, the overall performance of an attack is reported as the mean of $R_{\text{CA}}(c)$ or $R_{\text{QA}}(q)$ over T times of independent trials, accordingly.

Sampling for CA+ & QA+. For CA+, the query set Q is randomly sampled from X. Likewise, for QA+, the candidate set C is randomly sampled from X. Before attack, both the $R_{\text{CA}}(c)$ and $R_{\text{QA}}(q)$ will approximate to 50%. Expectedly, the attacks should significantly *decrease* the value towards 0%.

Sampling for CA- & QA-. We expect an attacker in practice prefers to lower some top-ranked candidates than further lowering candidates that are already away from the top part of the ranking list. Thus, the Q for CA- and the C for QA- 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

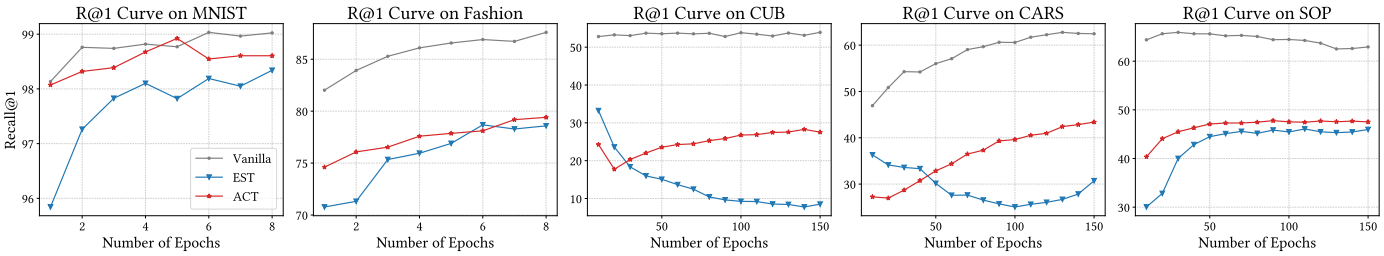


Fig. 6. R@1 Curves of Defense Methods on Various Datasets. ACT consistently converges faster and generalizes better than EST on any dataset.

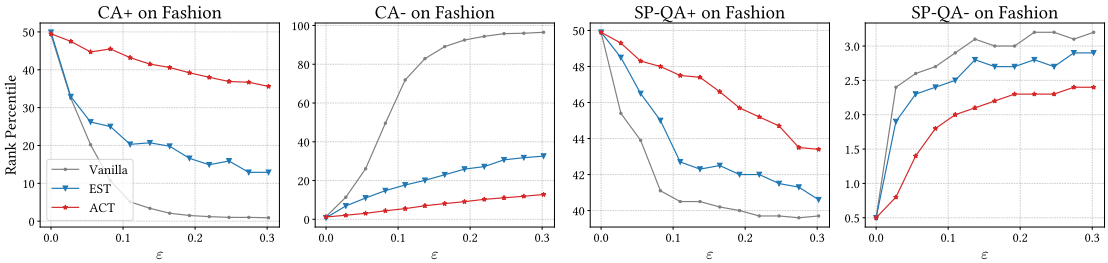


Fig. 7. Adversarial Ranking Attacks on Different Models with Varying ε. ACT consistently manifests better robustness than EST against any attack.

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 towards 100%.

Parameters for CA&QA. We conduct CA with $w \in \{1,2,5,10\}$ queries, and QA with $m \in \{1,2,5,10\}$ candidates, respectively. In SP-QA, we let $G = 5$. The parameter ζ in SP is empirically set to $2e4$ for MNIST; $4e4$ for Fashion and CUB; and $7e4$ for CARS and SOP in order to keep C_{SP} within the top-1% ranked samples. We perform $T = |X|$ times of attack to obtain the reported performance.

6.1.3 Hyper-Parameters for Projected Gradient Descent

We use PGD [10] as the optimizer for any attack mentioned in this paper. Specifically, we investigate attacks with different $\varepsilon \in \{8/255, 77/255\}$ on MNIST and Fashion; $\varepsilon \in \{2/255, 8/255\}$ on CUB, CARS and SOP following [13], [10], [54]. The number of PGD iterations is a constant $\eta = 32$, where the size of each step is a constant $\alpha = 3/255$.

6.1.4 Defense & Empirical Robustness Score

Adversarial Defense. Following the procedure of Madry defense [10], we use a strong adversary (*i.e.*, $\varepsilon = 77/255$ on MNIST and Fashion datasets; $\varepsilon = 8/255$ on CUB200, CARS196, and SOP) to create adversarial examples for adversarial training based on standard deep ranking [14].

Scores and Normalization for ERS. Let z be the performance score of any attack. Specifically, **(1)** we evaluate CA and QA as described in Section 6.1.2. The ranking percentile of CA- or QA- is normalized as $100 - z$, because $z \in [0, 100]$ for the two attacks while an ideally robust model leads to 0. The score of CA+ or QA+ is normalized as $2z$, because $z \in [0, 50]$ while an ideally robust model leads to 50; **(2)** The TMA is evaluated in cosine similarity. It is normalized as $100(1 - z)$ as its value lies within $[0, 1]$ in most cases, while an ideal model leads to a small value; **(3)** The ES is evaluated in embedding shift distance (denoted as ES:D) and R@1 (denoted as ES:R). For ES:D, the shift distance is normalized as $100 * (1 - z/2)$, because $z \in [0, 2]$ while an ideal model leads to a small value. For ES:R, the R@1 is directly used since it ranges within $[0, 100]$; **(4)** The LTM and GTM are both evaluated in R@1, which is directly used; **(5)** The GTT is measured in the success rate (scaled to $[0, 100]$ so directly used for ERS) that the top-1 sample is retained within the top-k ($k = 4$) result after the attack. We do not choose $k = 1$ in order to lower the difficulty. The score of any attack is averaged over $T = |X|$ times of trials. Finally, the average of normalized scores across all attacks is ERS. Thus, a higher value indicates better model robustness.

Marks. To ease the comparison among different defenses in the following tables, we mark the vanilla (*i.e.*, without any defense) model as "✗", EST as "✓", REST as "✓*", SES as "○", and ACT as "★". Superscript "$^{\beta}$" to a mark indicates the usage of a non-default margin β.

6.2 Candidate Attack & Query Attack

On MNIST Dataset. We first train a vanilla (*i.e.*, without defense) C2F2 ranking model. Its retrieval performance can be found in the "Benign Example" columns of Table 2. Then we conduct adversarial ranking attacks against this model.

The attack results are presented in Table 2. For example, a strong **CA+** with $\varepsilon = 77/255$ and $w = 1$ can raise the rank $R_{\text{CA}}(c)$ from 50% to 3.3%, which is close to the top part of ranking list. As expected, the attack has a weaker effect with a $\varepsilon = 8/255$ constraint, but remains effective. Likewise, the rank of c can be raised to 10.3%, 14.1%, 15.9% for $w = 2, 5, 10$ chosen queries, respectively. This means that CA+ with more selected queries is more difficult. We speculate such difficulty mainly stems from geometric restriction[3] in the embedding space and optimization difficulty[4].

Meanwhile, a strong **CA-** for $w = 1$ can lower the rank $R_{\text{CA}}(c)$ from 2.0% to 69.9%. The **CA-** for $w = 2, 5, 10$ are

3. For instance, a candidate c cannot be close to q_1 and q_2 simultaneously when the distance between q_1 and q_2 is large.
4. The PGD optimizer is based on the first-order gradient.

TABLE 3
Adversarial Robustness Evaluation for Deep Ranking Models on Various Datasets.

Dataset	Model	Loss	Defense	Benign Example				White-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
MNIST	C2F2	Triplet	✗	**99.0**	**99.4**	**98.7**	84.7	3.3	69.9	3.7	83.8	0.940	1.314	0.6	21.7	10.5	0.0	13.3
			✓	98.3	99.0	91.3	80.7	6.8	36.0	15.3	51.3	0.920	0.572	78.4	58.6	31.6	0.0	40.5
			★	98.6	99.1	98.1	**86.4**	**33.4**	**7.6**	**35.7**	**3.8**	**0.145**	**0.259**	**93.2**	**96.6**	**96.1**	**1.1**	**78.6**
Fashion	C2F2	Triplet	✗	**87.6**	**92.7**	**84.9**	**77.8**	1.0	95.0	0.5	94.2	0.993	1.531	0.1	0.8	6.7	0.0	4.5
			✓	78.6	86.8	64.6	64.9	12.1	32.7	19.6	49.6	0.955	**0.381**	57.2	22.4	17.6	0.0	36.4
			★	79.4	87.9	71.6	69.6	**34.7**	**11.3**	**39.1**	**9.0**	**0.216**	0.450	**58.5**	**66.2**	**68.0**	**0.5**	**67.7**
CUB	RN18	Triplet	✗	**53.9**	**66.4**	**26.1**	**59.5**	0.0	100.0	0.0	99.9	0.883	1.762	0.0	0.0	14.1	0.0	3.8
			✓	8.5	13.0	2.6	25.2	2.7	97.9	0.4	97.3	0.848	1.576	1.4	0.0	4.0	0.0	5.3
			★	27.5	38.2	12.2	43.0	**15.5**	**37.7**	**15.1**	**32.2**	**0.472**	**0.821**	**11.1**	**9.4**	**14.9**	**1.0**	**33.9**
CARS	RN18	Triplet	✗	**62.5**	**74.0**	**23.8**	**57.0**	0.2	100.0	0.1	99.6	0.874	1.816	0.0	0.0	13.4	0.0	3.6
			✓	30.7	41.0	5.6	31.8	1.2	98.1	0.4	91.8	0.880	1.281	2.9	0.7	8.2	0.0	7.3
			★	43.4	54.6	11.8	42.9	**18.0**	**32.3**	**17.5**	**30.5**	**0.383**	**0.763**	**16.3**	**15.3**	**20.7**	**1.6**	**38.6**
SOP	RN18	Triplet	✗	**62.9**	**68.5**	**39.2**	**87.4**	0.1	99.3	0.2	99.1	0.845	1.685	0.0	0.0	6.3	0.0	4.0
			✓	46.0	51.4	24.5	84.7	12.5	43.6	10.6	34.8	0.468	0.830	9.6	7.2	17.3	3.8	31.7
			★	47.5	52.6	25.5	84.9	**24.1**	**10.5**	**22.7**	**9.4**	**0.253**	**0.532**	**21.2**	**21.6**	**27.8**	**15.3**	**50.8**

The mark "↑" means larger values are preferred for a robust model, while "↓" means smaller values are preferred for a robust model. The ERS value is calculated as the average normalized score across all attacks involved, as described in Section 6.1.4.

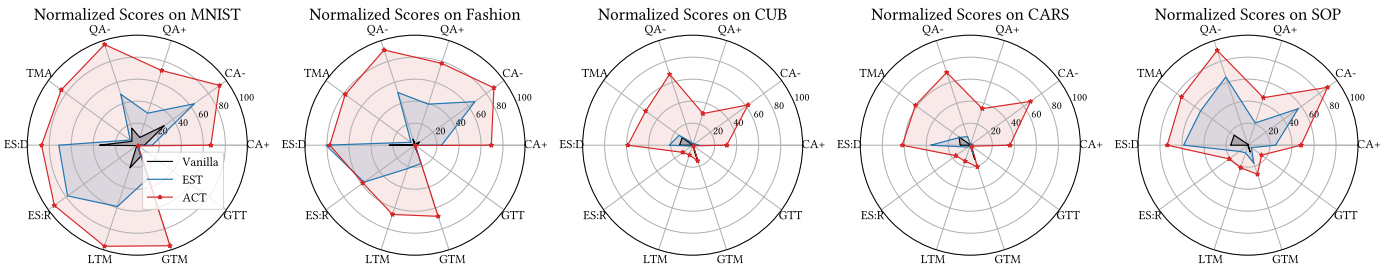


Fig. 8. Comparison of Individual Normalized Attack Scores among Different Defense Methods. Every score is normalized within $[0, 100]$.

similarly effective. A larger w makes the attack more difficult due to the same reasons as **CA+**. Note, the **CA-** performance drop with a large w is relatively small because the selected queries Q are highly correlated due to sampling method.

The results of **SP-QA+** and **SP-QA-** are also shown in Table 2. For instance, the **SP-QA+** ($m = 1$) can raise the rank of c from 50% to 29.2%, while keeping the rank of C_{SP} at 0.7%. The **SP-QA-** ($m = 1$) can lower the rank of c from 0.5% to 2.7%, while retaining C_{SP} in the top-ranked candidates. This means **SP-QA** can raise or lower the rank of C (with a less dramatic effect compared to **CA**) while largely retaining the query semantics. The difficulty stems from **SP** term in Eq. (11). The **SP-QA** effectiveness is inversely correlated with ζ. Predictably, the effect can be boosted with a smaller ζ, at the cost of a larger change in query semantics. Such a trade-off for **SP-QA** between the extent of rank change and the extent of semantics change depends on the attacker.

On the Other Datasets. We train the vanilla deep ranking models with C2F2 architecture on Fashion, and RN18 architecture on CUB, CARS, and SOP. Their corresponding ranking performance on benign examples can be found in Table 2. Then we conduct attacks against these models.

The attack results are available in Table 2. As shown, the CA+ and CA- consistently achieve a better effect on datasets harder than MNIST. For example, in a strong **CA+** ($w = 1$) on SOP, the rank $R_{\text{CA}}(c)$ can be raised to 0.1%, almost reaching the top. In a strong **CA-** ($w = 1$) on SOP, the rank of c can be lowered to 99.3%, almost reaching the bottom. We speculate that the dataset difficulty partly contributes to the effectiveness of CA, since it is difficult for a model to converge into an ideal state. Moreover, the input dimension (*i.e.*, $1 \times 28 \times 28$ for C2F2, $3 \times 224 \times 224$ for RN18) is another reason why CA is more effective on RN18 than C2F2. According to Goodfellow *et al.* [3], it is easier to drive the neural network output into a "locally linear area" with a higher input dimension. Thus, CA can easily succeed on models with a high dimensional input. The models like C2F2 are actually not easy to attack.

As expected, the SP-QA+ and SP-QA- are also effective on the other datasets. Note, the SP-QA with a large ε (*e.g.*, $8/255$) is sometimes less effective than that with a small ε (*e.g.*, $2/255$). Because it is easier to significantly change the query semantics under a large ε, meanwhile triggering a very strong semantics-preserving penalty that immediately dominates the loss. Thus, the optimizer will temporarily stop raising or lowering C to pull C_{SP} back to the top of the ranking list. As a particular characteristic, the SP-QA performance does not necessarily peak at a large ε.

In summary, the deep ranking models are vulnerable to adversarial ranking attack. Our proposed CA and QA are particularly effective when the input dimension is large, or the corresponding dataset is difficult. Previous study [13] also suggests that CA and QA are effective regardless of the choice of metric learning loss function for training and the

TABLE 4
Robustness with Revised EST (REST) on Various Datasets.

Dataset	Model	Loss	Defense	Benign Example				White-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
MNIST	C2F2	Triplet	✓	98.3	99.0	91.3	80.7	6.8	36.0	15.3	51.3	0.920	0.572	78.4	58.6	31.6	0.0	40.5
			✓*	**98.8**	**99.3**	**98.3**	**87.5**	**26.9**	**9.5**	**32.9**	**6.4**	**0.391**	**0.291**	**88.4**	**90.8**	**89.4**	**0.1**	**71.9**
Fashion	C2F2	Triplet	✓	78.6	86.8	64.6	64.9	12.1	32.7	19.6	49.6	0.955	0.381	**57.2**	22.4	17.6	0.0	36.4
			✓*	**79.8**	**87.7**	**71.1**	**69.4**	**28.3**	**17.0**	**36.8**	**13.6**	**0.701**	**0.360**	51.1	**49.4**	**57.2**	**0.1**	**56.9**
CUB	RN18	Triplet	✓	8.5	13.0	2.6	25.2	2.7	97.9	0.4	97.3	0.848	1.576	1.4	0.0	4.0	0.0	5.3
			✓*	**13.6**	**20.8**	**5.9**	**34.2**	**9.4**	**40.2**	**11.2**	**40.5**	**0.810**	**0.590**	**7.0**	**1.1**	**7.7**	**0.1**	**26.6**
CARS	RN18	Triplet	✓	30.7	41.0	5.6	31.8	1.2	98.1	0.4	91.8	0.880	1.281	2.9	0.7	8.2	0.0	7.3
			✓*	**31.9**	**42.3**	**7.0**	**34.3**	**7.6**	**45.1**	**7.8**	**44.6**	**0.735**	**0.642**	**7.1**	**5.4**	**12.3**	**0.4**	**26.1**
SOP	RN18	Triplet	✓	46.0	51.4	24.5	84.7	12.5	43.6	10.6	34.8	0.468	0.830	9.6	7.2	17.3	3.8	31.7
			✓*	**47.2**	**52.3**	**25.3**	**84.9**	**22.0**	**13.6**	**20.4**	**11.3**	**0.362**	**0.500**	**18.3**	**20.5**	**24.7**	**10.1**	**47.2**

The comparison between REST and EST attests the efficacy of mitigation of misleading gradients.

TABLE 5
Robustness with EST in a Larger Margin on Fashion.

Margin β	Model	Loss	Defense	Benign Example				White-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
0.2	C2F2	Triplet	✓	78.6	86.8	**64.6**	**64.9**	12.1	32.7	19.6	49.6	0.955	**0.381**	57.2	22.4	17.6	0.0	36.4
0.4	C2F2	Triplet	$✓^{\beta}$	**81.0**	**88.6**	64.0	63.5	17.5	36.8	28.1	40.7	0.764	0.662	**63.1**	35.9	22.8	0.0	42.6
0.6	C2F2	Triplet	$✓^{\beta}$	78.0	86.4	62.5	61.5	26.9	28.6	32.0	27.3	0.533	0.649	60.1	50.8	38.7	0.0	52.6
0.8	C2F2	Triplet	$✓^{\beta}$	77.1	85.6	60.7	61.7	**35.7**	**18.5**	**40.0**	**14.4**	**0.232**	0.434	57.9	**53.5**	**53.9**	0.0	**63.9**
1.0	C2F2	Triplet	$✓^{\beta}$	73.9	84.1	57.2	60.3	28.3	26.5	34.5	24.3	0.422	0.565	45.2	45.3	37.9	0.0	53.3

The comparison between different margins (β) based on EST attests the efficacy of mitigation of inefficient mini-batch exploitation in EST.

distance metric. Therefore, we speculate that models used in realistic applications are vulnerable, because they are usually trained on larger-scale and more difficult datasets.

6.3 Defending against CA & QA

As suggested by the above experiments, deep ranking models are vulnerable to adversarial ranking attacks. Whereas attacks may cause security or fairness concerns, a defense to make a ranking model resistant to the attacks is necessary.

To validate the proposed defenses, we train ranking models on the five datasets with them, respectively. The R@1 curves of these defensive models are presented in Fig. 6. As a commonly seen phenomenon, adversarial training leads to a notable performance drop on benign examples, particularly on difficult datasets. For example, while an RN18 without defense can achieve an R@1 of 62.9% on SOP, an RN18 only achieves R@1 of 46.0% with EST, or 47.5% with ACT.

As discussed in Section 4, the EST defense [13] suffers from misleading gradient and inefficient mini-batch exploitation. As a result, the corresponding model is expected to suffer from slow convergence. The R@1 curves in Fig. 6 attest this speculation. Notably, EST leads to a more pronounced performance drop under the zero-shot setting (*i.e.*, the CUB and CARS datasets). From the figure, we also note that the ACT defense, being free from problems identified in EST, converges much faster than EST in terms of R@1, and achieves higher performance on all datasets. The complete ranking performance is presented in Table 2.

Then we examine the defense methods with CA and SP-QA. As shown in Fig. 7, we conduct attacks on the models with ε varying from 0 to $77/255$ (with a $7/255$ interval) on Fashion. As an overall trend, the effect of an attack increases with the ε increasing. Through training with EST, the model gains a moderate robustness against the attacks. Besides, being free from the problems in EST, ACT outperforms EST by a large margin in terms of resistance to these attacks.

Particularly, it is noted in Fig. 7 that the rank percentile values of SP-QA+ and SP-QA- are still far from the ideal value (0.0 and 100.0, respectively), even if ε is large. This is not because the attack is not strong. In fact, it is challenging to simultaneously minimize the QA term (left part of Eq. (11)) and the SP term (right part of Eq. (11)). As discussed in Section 3.2, an optimization step for the QA term may drastically change the semantics of the query. This makes the C_{SP} far from the top of the ranking list, resulting in a large penalty by the SP term. Given a large penalty by the SP term for preserving query semantics, the next optimization step will largely "revert" to the previous optimization step, by moving the C_{SP} back to the top of the ranking list, restoring the query semantics and reduce the SP term penalty. Thus, optimizing the loss of SP-QA is very challenging, as discussed in Section 3.2 and Section 6.2. Fig. 7 aims to demonstrate the effectiveness of defense against the attacks demonstrated in Table 2. However, to better represent the worst-case robustness of a model, the QA for ERS does not involve the SP term as described at the beginning of Section 5. It will be seen in the following sections that the QA+ and QA- performance can be very close to the ideal value without the SP term. Thus, the reason why the SP-QA curves in Fig. 7 are far from the ideal value is not about attack strength.

The complete results on all datasets can be found in Table 2. In general, the EST can achieve a moderate level of robustness against CA and SP-QA, but at a high cost

TABLE 6
Comparison among All Defenses and Their Variants.

Dataset	Model	Loss	Defense	Benign Example				White-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
Fashion	C2F2	Triplet	✓	78.6	86.8	64.6	64.9	12.1	32.7	19.6	49.6	0.955	0.381	57.2	22.4	17.6	0.0	36.4
			$\checkmark^{\beta}$	77.1	85.6	60.7	61.7	35.7	18.5	40.0	14.4	0.232	0.434	57.9	53.5	53.9	0.0	63.9
			$\checkmark^{*}$	**79.8**	87.7	71.1	69.4	28.3	17.0	36.8	13.6	0.701	0.360	51.1	49.4	**57.2**	0.1	56.9
			$\checkmark^{*,\beta}$	79.6	87.6	**72.8**	**70.8**	35.7	16.6	40.5	15.7	0.246	0.369	51.3	49.1	55.3	0.0	63.3
			○	71.2	83.2	56.6	59.5	**38.0**	17.8	**44.7**	13.0	0.964	**0.022**	51.4	49.6	51.2	0.2	59.0
			★	79.4	**87.9**	71.6	69.6	34.7	**11.3**	39.1	**9.0**	**0.216**	0.450	**58.5**	**66.2**	**68.0**	**0.5**	**67.7**

The best result in each column is highlighted in bold font. The second best result is underscored.

TABLE 7
Robustness with ACT in Different Anti-Collapse Strength (Different Margins in ACT).

Margin β	Model	Loss	Defense	Benign Example				White-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
0.2	C2F2	Triplet	★	**79.4**	**87.9**	**71.6**	69.6	34.7	**11.3**	39.1	**9.0**	0.216	0.450	58.5	**66.2**	**68.0**	**0.5**	67.7
0.4	C2F2	Triplet	$★^{\beta}$	78.5	87.0	69.6	68.2	37.3	11.9	40.7	9.2	0.172	**0.399**	**59.0**	63.0	66.8	0.4	**68.7**
0.6	C2F2	Triplet	$★^{\beta}$	78.0	86.4	68.3	**69.7**	**38.6**	11.8	**42.3**	9.4	**0.169**	0.409	52.7	63.2	66.1	0.4	68.6
0.8	C2F2	Triplet	$★^{\beta}$	77.6	85.8	65.1	65.0	35.8	16.2	39.7	13.9	0.217	0.416	50.0	54.7	57.2	0.1	64.0
1.0	C2F2	Triplet	$★^{\beta}$	77.2	86.1	62.9	64.3	35.3	18.5	39.6	15.6	0.205	0.443	42.5	50.9	46.6	0.1	61.3

of ranking performance. On the other hand, the newly proposed ACT can achieve significantly higher robustness, while generalizing better on benign examples (especially under zero-shot settings such as CUB and CARS). ACT overwhelmingly outperforms the EST defense.

Apart from these, the previous study [13] discovers the performance of attacks and defenses varies across different embedding distance metrics (*e.g.*, Euclidean distance or cosine distance), or different metric learning loss functions. We leave further investigation for future study.

6.4 Adversarial Robustness Evaluation

As discussed in Section 5, a practical defense for deep ranking should be resistant to as many types of attacks as possible. Thus, we evaluate the defenses with the proposed ERS.

The performance of all attacks involved in ERS on the ranking models is available in Table 3. These scores are also normalized and visualized in Fig. 8. Take the MNIST dataset as an example, as shown in Table 3. Compared to the EST defense, the ACT defense (1) effectively reduces the efficacy for CA+, CA-, QA+ and QA- to change the rank of selected candidates; (2) only allows the cosine distance between two random samples to increase to 0.145 for TMA; (3) suppresses the maximum embedding shift distance to merely 0.259 for ES:D; (4) retains a much higher R@1 for ES:R, LTM, and GTM; (5) retains the original top-1 candidate within the top-4 results at a success rate of 1.1% in GTT. Notably, ACT is the only method that consistently achieves a non-zero performance in GTT, which is extremely difficult.

From the table and figure, the proposed ACT significantly outperforms EST (state-of-the-art ranking defense) in defending against all attacks involved, while achieving a higher generalization performance on benign examples. By comparing the ERS scores, it is noted that ACT defense achieves at least 60% and at most 540% improvement over the EST defense. This attests our analysis of EST in Section 4.

7 Discussions

In this section, we first conduct further experiments to verify the analysis on the characteristics of EST. Such analysis is also the foundation of the ACT defense. Ultimately, we examine the relationship between adversarial robustness and some commonly concerned factors in deep metric learning.

Apart from these, as discussed in Section 3, an SP loss term is introduced in **SP-QA** to balance semantics preservation and the actual attack goal. By comparing the performance between **SP-QA** ($m = 1$) in Table 2 and that of pure **QA** ($m = 1$) in Table 3, it is clear that **SP-QA** with a large ζ can be much harder than the pure **QA** (*i.e.*, $\xi = 0$) due to the additional SP term.

7.1 Characteristics of EST & ACT

In Section 4.1, we present the mitigations to the misleading gradient and inefficient mini-batch exploitation, correspondingly. The mitigation of the former issue is the Revised EST (REST), while the mitigation of the latter one is to enlarge the margin hyper-parameter β. To validate the underlying ideas, we train defended ranking models with these mitigations, which are subsequently evaluated with ERS.

Mitigation of Misleading Gradient. We train ranking models on all datasets with REST and compare the result with those of EST, as shown in Table 4. Clearly, the sole mitigation of misleading gradient leads to significant improvement in ranking performance on benign examples and the resistance to all attacks. A significantly higher ERS is achieved on every dataset, with at least 49% and at most 402% improvement over EST, nearly reaching that of ACT.

Mitigation of Inefficient Mini-batch Exploitation. We enlarge the margin β and train models on the Fashion dataset. Then we evaluate the resulting models with the proposed ERS. The results can be found in Table 5. Clearly, this mitigation can also significantly improve robustness against all types of attacks, but results in a drop in benign example

TABLE 8
Adversarial Robustness with Different Models or Triplet Sampling Strategies on CUB Dataset.

Group	Model	Loss	Defense	Benign Example				White-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
#0	RN18	Triplet	✗	**53.9**	**66.4**	**26.1**	**59.5**	0.0	100.0	0.0	99.9	0.883	1.762	0.0	0.0	14.1	0.0	3.8
			✓	8.5	13.0	2.6	25.2	2.7	97.9	0.4	97.3	0.848	1.576	1.4	0.0	4.0	0.0	5.3
			★	27.5	38.2	12.2	43.0	**15.5**	**37.7**	**15.1**	**32.2**	**0.472**	**0.821**	**11.1**	**9.4**	**14.9**	**1.0**	**33.9**
#1	RN50	Triplet	✗	**53.3**	**65.6**	**26.4**	**58.9**	0.3	100.0	0.1	99.4	0.909	1.766	0.0	0.0	14.5	0.0	3.7
			✓	18.7	26.7	5.6	33.4	3.8	82.2	4.0	79.6	0.859	1.113	4.7	0.3	7.0	0.0	12.4
			★	31.7	40.8	14.3	45.7	**16.9**	**35.0**	**16.0**	**31.3**	**0.410**	**0.842**	**14.9**	**11.0**	**18.1**	**1.2**	**36.2**
	IBN		✗	**53.5**	**66.1**	**27.4**	**59.1**	0.1	100.0	0.1	99.6	0.876	1.752	0.1	0.0	13.7	0.0	3.9
			✓	24.0	33.7	8.3	38.4	3.2	80.6	5.4	64.8	0.950	**0.555**	9.3	0.7	9.3	0.0	16.8
			★	28.9	38.7	12.4	43.3	**17.4**	**36.7**	**17.2**	**29.2**	**0.445**	0.790	**14.6**	**12.3**	**15.8**	**1.7**	**36.4**
	Mnas		✗	**54.2**	**66.2**	**26.5**	**59.9**	0.1	100.0	0.1	99.6	0.860	1.785	0.0	0.0	**13.6**	0.0	3.9
			✓	17.4	25.0	4.4	30.2	0.6	99.0	0.4	98.0	0.828	1.598	1.2	0.0	6.3	0.0	5.0
			★	23.3	32.1	10.0	40.9	**17.4**	**32.3**	**18.3**	**27.2**	**0.482**	**0.736**	**13.6**	**9.0**	12.8	**0.9**	**36.3**
#2	RN18	Triplet (Semihard)	✗	**39.7**	**51.3**	**19.1**	**51.8**	0.2	100.0	0.2	99.8	0.820	1.681	0.4	0.0	12.1	0.0	4.8
			✓	4.5	7.7	1.7	22.9	5.6	99.9	5.9	83.6	0.907	1.821	1.2	0.4	2.1	0.0	6.1
			★	20.3	28.8	9.1	39.3	**21.0**	**27.4**	**22.3**	**21.2**	**0.436**	**0.694**	**11.0**	**9.0**	**12.9**	**1.6**	**39.4**
		Triplet (Softhard)	✗	**53.4**	**65.3**	**25.9**	**60.1**	0.0	100.0	0.0	99.5	0.932	1.353	0.1	0.0	12.5	0.0	5.2
			✓	37.6	48.7	14.6	46.4	1.1	92.2	1.3	80.5	0.984	**0.376**	11.9	3.1	12.0	0.0	14.2
			★	39.4	50.2	18.6	51.3	**6.8**	**61.5**	**5.2**	**60.4**	**0.506**	1.032	**12.8**	**11.3**	**17.7**	**0.3**	**24.2**
		Triplet (Distance)	✗	**48.2**	**60.7**	**23.4**	**56.6**	0.0	100.0	0.0	100.0	0.830	1.747	0.1	0.0	**13.5**	0.0	4.3
			✓	6.3	10.8	2.1	30.2	7.3	99.2	8.1	**29.0**	1.000	**0.013**	**2.6**	0.2	1.5	0.0	**20.6**
			★	18.5	25.4	4.9	29.1	**13.5**	**95.3**	**8.6**	97.5	**0.517**	1.698	1.6	**0.3**	9.4	0.0	12.6

ranking performance. The robustness peaks at $\beta = 0.8$, but the ranking performance does not peak in the meantime.

Mitigations Combined. We combine the two mitigations together, and compare the performance with other defense methods in Table 6. As shown in the 4-th row of the table, the combination achieves a higher ranking performance, and a significant improvement in robustness, but is still outperformed by ACT. This is because ACT eliminates these problems instead of merely mitigating them.

SES Defense. The SES defense method discussed in Section 4.1 can lead to competitive robustness, but meanwhile a significant ranking performance drop. Notably, SES is particularly resistant to the ES attack (embedding shift is suppressed to 0.022), but is relatively weak against some other attacks. Hence, suppressing the embedding shift that adversarial perturbation could incur is not the only condition to achieve a robust model. We speculate that solely reducing the ES may not necessarily introduce enough robustness, because the embedding shift can also be reduced by linearly "shrinking" the embedding space to a smaller scale.

Anti-Collapse Strength of ACT. The margin parameter β has extra meaning in our ACT defense, *i.e.*, the "strength" to separate the adversarially "collapsed" positive and negative samples. To better understand the margin for ACT, we conduct experiments with various margins, as shown in Table 7. From the table, we discover that a slightly larger margin (*e.g.*, 0.4) can further boost the model robustness, as the model is forced to learn more robust representations. However, an excessively large margin will harm the generalization performance as expected in the literature of deep ranking. Thus, there is a trade-off between generalization performance and robustness as they do not peak simultaneously.

In summary, all these experiments attest our analysis of the EST in Section 4. Being free from problems identified in EST, our ACT greatly outperforms EST in various aspects.

7.2 Robustness with Other Models / Triplet Samplers

In deep metric learning, the model architecture and the triplet sampling strategy greatly impact the performance, but their impact on adversarial robustness remains unexplored. To this end, we follow [14] and train models on CUB with different architectures including ResNet-50 (RN50) [1], Inception-BN (IBN) [4], and MnasNet-1.0 (Mnas) [76]); or with different triplet sampling strategies including semi-hard [11], soft-hard [14], and distance-weighted [21] triplet sampling strategy. Then we evaluate the models trained using these settings with ERS, as shown in Table 8. Note, with an aligned number of training epochs across all models in a specific architecture, the models without defense suffer from overfitting, but the models with defense need to be sufficiently trained to gain enough robustness and performance on benign examples. The results in the table suggest that the proposed ACT outperforms EST in virtually any setting.

By comparing group #0 and group #1, we discover that model capacity alone benefits adversarial robustness for adversarial training. For example, compared to RN18 which achieves an ERS of 5.3 for EST or 33.9 for ACT, RN50 achieves an ERS of 12.4 for EST or 36.2 for ACT. A similar effect is also shown by IBN, which has a larger model capacity than RN18. This observation is consistent with Madry's conclusion [10]. On the other hand, although the Mnas model has a lower model capacity than RN18, it achieves comparable robustness due to its better architecture.

By comparing group #0 (uniform sampling) and group #2, we note that the triplet sampling strategy greatly impacts adversarial robustness. Compared to uniform sampling, the negative samples from semi-hard sampler ($-\beta < d(q, c_p) - d(q, c_n) < 0$) are easier to be collapsed with a positive sample, so that the model will learn more robust representations to separate them (hence a higher ERS); The positive and negative samples from the soft-hard sampler are initially

TABLE 9
Defense Methods using Adversarial Examples Created Using FGSM.

Dataset	Model	Loss	Defense	Benign Example				White-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
MNIST	C2F2	Triplet	✓ (FGSM)	98.5	99.1	95.2	93.2	3.3	55.8	3.4	71.5	0.967	0.633	63.2	22.8	8.4	0.0	25.2
			★ (FGSM)	98.9	99.3	98.8	92.9	9.9	46.1	11.1	50.7	0.709	1.241	10.0	66.4	25.9	0.0	31.5
Fashion	C2F2	Triplet	✓ (FGSM)	83.6	89.9	71.8	69.1	2.5	74.5	2.3	83.3	0.980	1.037	16.2	7.2	10.3	0.0	13.5
			★ (FGSM)	83.7	90.1	77.0	74.3	8.5	63.2	11.1	71.5	0.804	1.357	7.2	22.0	17.1	0.0	20.3
CUB	RN18	Triplet	✓ (FGSM)	47.7	60.3	23.6	57.7	0.0	100.0	0.0	99.5	0.918	1.729	0.0	0.0	12.8	0.0	3.5
			★ (FGSM)	31.4	41.9	14.1	46.3	6.6	70.2	4.9	68.0	0.598	1.184	5.6	3.8	13.6	0.1	18.9
CARS	RN18	Triplet	✓ (FGSM)	62.3	73.9	22.3	55.5	0.1	100.0	0.1	99.5	0.905	1.762	0.0	0.0	12.2	0.0	3.4
			★ (FGSM)	43.6	54.9	12.1	42.3	9.1	50.3	8.5	54.4	0.495	0.989	8.5	7.0	17.3	0.4	26.5
SOP	RN18	Triplet	✓ (FGSM)	58.4	63.8	34.8	86.6	1.3	97.3	0.3	95.4	0.837	1.487	0.0	0.0	5.6	0.0	5.8
			★ (FGSM)	53.6	59.0	30.3	85.7	15.9	20.4	14.0	21.2	0.348	0.668	12.1	13.3	20.2	8.9	40.5

TABLE 10
Adversarial Robustness Evaluation by Replacing PGD with NES Algorithm (Black-Box Attack).

Dataset	Model	Loss	Defense	Benign Example				Black-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
Fashion	C2F2	Triplet	✗	**87.6**	**92.7**	**84.9**	**77.8**	20.6	31.4	26.3	40.4	0.665	1.174	14.0	**84.0**	44.3	0.0	43.9
			✓	78.6	86.8	64.6	64.9	26.0	17.9	41.8	40.6	0.929	0.343	30.1	61.8	52.0	0.0	51.1
			★	79.4	87.9	71.6	69.6	**40.7**	**6.0**	**48.3**	**3.9**	**0.140**	**0.263**	**71.7**	77.6	**71.6**	**2.0**	**76.4**

TABLE 11
Adversarial Robustness Evaluation with Adversarial Example Transferability (Black-Box Attack).

Dataset	Model	Loss	Defense	Benign Example				Black-Box Attacks for Robustness Evaluation										ERS ↑
				R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	
Fashion	C2F2	Triplet	✗	**87.6**	**92.7**	**84.9**	**77.8**	39.8	10.6	49.0	10.7	0.456	0.490	51.9	67.1	48.7	1.4	65.5
			✓	78.6	86.8	64.6	64.9	42.8	**1.1**	49.5	1.9	0.792	0.118	69.6	76.0	69.5	24.1	73.6
			★	79.4	87.9	71.6	69.6	**49.6**	1.7	**49.9**	**0.8**	**0.060**	**0.069**	**76.4**	**77.3**	**76.6**	**61.1**	**87.9**

further from each other, hence are less likely to be collapsed together (hence a lower ERS); The distance-weighted sampler has a higher chance of selecting very far negative samples that are even harder to be collapsed with the positive sample (hence an even lower ERS). Namely, the extent of "collapsing" the positive and negative samples affects the model's focus on learning robust representations to separate them.

7.3 Defense with FGSM instead of PGD

Recall that Madry defense [10] involves creating adversarial examples using PGD at every iteration during the adversarial training process. As PGD involves multiple times of model forward and backward propagation, the time consumption for creating adversarial examples can be much higher than training a vanilla model. Hence, we replace the PGD algorithm with FGSM [3], namely its single-iteration version (much faster), and evaluate the defense methods accordingly.

As shown in Table 9, defense with FGSM leads to better performance on benign examples, but meanwhile lower robustness compared to those with PGD in Table 3. Since FGSM is known to be much weaker than PGD in the effect of the attack, we speculate the reason is that the adversarial attack failed to achieve its goal (*e.g.*, "collapse" the positive and negative samples) results in inefficient learning of robust representations. Thus, similar to the discussion on different triplet samplers, the effectiveness of attack for our defense is an important factor that impacts robustness.

7.4 Robustness Evaluation with Black-Box Attack

In Section 5, it is pointed out that black-box attacks are empirically expected to perform worse than the white-box attacks used for ERS evaluation. To support this claim, we replace the PGD optimizer for all the involved attacks with a typical black-box attack method named Natural Evolution Strategy (NES) [77] and conduct experiments on the Fashion dataset. In the experiment, we can still calculate the loss values for the attacks, but the gradient calculation is canceled. NES estimates the gradient direction by the loss values, and performs a descent step based on the estimated gradient.

Specifically, the number of iterations is set as 32 to align with the experimental setting of PGD; the number of samples used for estimating the gradient is 100; the search variance is set as 0.6; the learning rate is set as $3/255$. The experimental results can be found in Table 10. The results show that (1) a black-box attack is empirically weaker than a white-box attack. The white-box attack is more suitable for ERS evaluation; (2) our proposed defense consistently outperforms the previous methods.

Apart from the black-box attacks using the estimated gradient, to further demonstrate the effectiveness of our proposed defense method, we also conduct experiments on the Fashion dataset with a typical transferability-based attack [27]. Typically, a white-box surrogate model is used to create adversarial examples with white-box attacks. It is observed that the resulting samples can also cause

misclassification on other models due to the similarity of their decision boundaries [27]. Following this, we create adversarial examples using a surrogate model in the same architecture, but initialized differently. The surrogate is trained normally using benign examples, based on which the adversarial examples are created using white-box attacks for robustness evaluation.

The results can be found in Table 11. In the deep ranking task, adversarial examples can also manifest transferability on the model without defense. However, such an attack is relatively weaker than the methods using the estimated gradient, because even a different parameter initialization could lead to a completely different embedding space, sharing less similarity than the embedding space of the black-box ranking model. Regarding robustness, ACT still outperforms EST in virtually every metric by a large margin.

8 Conclusion

Deep ranking models are vulnerable to adversarial perturbations. In this paper, we present *adversarial ranking attack* that can intentionally change the ranking result. To counter the attacks, two adversarial defense methods are proposed, namely EST and ACT. The EST defense can suppress the embedding shift distance and moderately improve the robustness, but suffers from the misleading gradient and insufficient mini-batch exploitation issues. Being free of these issues, the ACT defense significantly improves adversarial robustness and generalization performance. To comprehensively evaluate a defense, we propose an empirical robustness score involving a wide range of attacks.

In the potential of future works, a better defense can be designed to be more time-efficient while performs better on both benign and adversarial examples. Meanwhile, other metric learning loss functions can also be investigated.

Acknowledgments

This work was supported partly by National Key R&D Program of China Grant 2017YFA0700800, NSFC under Grant 62088102, Natural Science Foundation of Shaanxi Province under Grants 2022JC-41 and 2021JQ-054, and Fundamental Research Funds for the Central Universities under Grants XTR042021005 and XTR072022001.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778. 1, 9, 14

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representations*, 2014. 1, 2, 3

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015. 1, 2, 3, 4, 11, 15

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826. 1, 14

[5] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7714–7722. 1

[6] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM Conf. Computer Commun. Secur.*, 2016, pp. 1528–1540. 1

[7] Z. Wang, S. Zheng, M. Song, Q. Wang, A. Rahimpour, and H. Qi, "advPattern: Physical-world attacks on deep person re-identification via adversarially transformable patterns," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8341–8350. 1, 3, 8

[8] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Research*, vol. 11, no. 36, pp. 1109–1135, 2010. 1, 3

[9] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1386–1393. 1, 2, 3

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018. 2, 3, 4, 9, 10, 14, 15

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823. 2, 3, 4, 6, 7, 14

[12] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 125–136. 2, 7

[13] M. Zhou, Z. Niu, L. Wang, Q. Zhang, and G. Hua, "Adversarial ranking attack and defense," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 781–799. 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13

[14] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen, "Revisiting training strategies and generalization performance in deep metric learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8242–8252. 2, 3, 8, 9, 10, 14

[15] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 681–699. 2

[16] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, "Hierarchical multimodal lstm for dense visual-semantic embedding," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1881–1889. 2

[17] M. Zhou, Z. Niu, L. Wang, Z. Gao, Q. Zhang, and G. Hua, "Ladder loss for coherent visual-semantic embedding," in *Proc. AAAI. Conf. Artif. Intell.*, 2020, pp. 13 050–13 057. 2

[18] L. Zhang, Z. Shi, J. T. Zhou, M.-M. Cheng, Y. Liu, J.-W. Bian, Z. Zeng, and C. Shen, "Ordered or orderless: A revisit for video based person re-identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1460–1466, 2021. 2

[19] T.-Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, pp. 225–331, 2009. 2

[20] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4004–4012. 2, 8

[21] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2840–2848. 2, 14

[22] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. Int. Conf. Learn. Representations Workshops*, 2017. 3

[23] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57. 3

[24] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2206–2216. 3

[25] Y. Yu, X. Gao, and C.-Z. Xu, "LAFEAT: Piercing through adversarial defenses with latent features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021. 3

[26] S. Tang, X. Huang, M. Chen, C. Sun, and J. Yang, "Adversarial attack type I: Cheat classifiers by significant changes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 1100–1109, 2021. 3

[27] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Conf. Computer Commun. Secur.*, 2017, pp. 506–519. 3, 15, 16

[28] Y. Shi, S. Wang, and Y. Han, "Curls & whey: Boosting black-box adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6519–6527. 3

[29] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2730–2739. 3

[30] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4312–4321. 3

[31] Q. Huang, I. Katsman, H. He, Z. Gu, S. Belongie, and S.-N. Lim, "Enhancing adversarial example transferability with an intermediate level attack," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4733–4742. 3

[32] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1765–1773. 3

[33] H. Liu, R. Ji, J. Li, B. Zhang, Y. Gao, Y. Wu, and F. Huang, "Universal adversarial perturbation via prior driven uncertainty approximation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2941–2949. 3

[34] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking adversarial robustness on image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 321–331. 3, 4, 8

[35] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 284–293. 3

[36] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1625–1634. 3

[37] G. Goren, O. Kurland, M. Tennenholtz, and F. Raiber, "Ranking robustness under adversarial document manipulations," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 2018, pp. 395–404. 3

[38] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 2018, pp. 355–364. 3

[39] J. Li, R. Ji, H. Liu, X. Hong, Y. Gao, and Q. Tian, "Universal perturbation attack against image retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4899–4908. 3, 8

[40] G. Tolias, F. Radenovic, and O. Chum, "Targeted mismatch adversarial attack: Query with a flower to retrieve the tower," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5037–5046. 3, 7, 8

[41] H. Wang, G. Wang, Y. Li, D. Zhang, and L. Lin, "Transferable, controllable, and inconspicuous adversarial attacks on person re-identification with deep mis-ranking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 342–351. 3, 8

[42] M. Zhou, L. Wang, Z. Niu, Q. Zhang, Y. Xu, N. Zheng, and G. Hua, "Practical relative order attack in deep ranking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16 393–16 402. 3

[43] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3329–3338. 3, 8

[44] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1633–1645. 3

[45] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283. 3, 4

[46] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2016, pp. 582–597. 3

[47] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial example defense: Ensembles of weak defenses are not strong," in *USENIX Workshop Offensive Technol.*, 2017. 3

[48] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8571–8580. 3

[49] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM Conf. Computer Commun. Secur.*, 2017, pp. 135–147. 3

[50] A. Dubey, L. v. d. Maaten, Z. Yalniz, Y. Li, and D. Mahajan, "Defense against adversarial images using web-scale nearest-neighbor search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8767–8776. 3

[51] X. Liu, Y. Li, C. Wu, and C.-J. Hsieh, "Adv-BNN: Improved adversarial defense through robust bayesian neural network," in *Proc. Int. Conf. Learn. Representations*, 2019. 3

[52] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, "Towards robust neural networks via random self-ensemble," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 369–385. 3

[53] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 501–509. 3

[54] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. Int. Conf. Learn. Representations*, 2017. 3, 4, 10

[55] J. Wang and H. Zhang, "Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6629–6638. 3, 4

[56] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. D. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli, "Adversarial robustness through local linearization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13 842–13 853. 3

[57] D. Wu, S.-T. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 2958–2969. 3

[58] F. Croce and M. Hein, "Sparse and imperceivable adversarial attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4724–4732. 3

[59] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: Elastic-Net attacks to deep neural networks via adversarial examples," in *Proc. AAAI. Conf. Artif. Intell.*, 2018, pp. 10–17. 3

[60] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," in *Proc. Int. Conf. Learn. Representations*, 2016. 4

[61] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3839–3848. 5

[62] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6241–6250. 5

[63] M. Picot, F. Messina, M. Boudiaf, F. Labeau, I. Ben Ayed, and P. Piantanida, "Adversarial robustness via fisher-rao regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 5

[64] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 854–863. 5

[65] K. Roth, Y. Kilcher, and T. Hofmann, "Adversarial training is a form of data-dependent operator norm regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 14 973–14 985. 5

[66] F. Farnia, J. Zhang, and D. Tse, "Generalizable adversarial training via spectral normalization," in *Proc. Int. Conf. Learn. Representations*, 2019. 5

[67] L. Li, T. Xie, and B. Li, "SoK: Certified robustness for deep neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2023. 5

[68] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv:1708.07747*, 2017. 5, 8

[69] S. Bai, Y. Li, Y. Zhou, Q. Li, and P. H. Torr, "Adversarial metric attack and defense for person re-identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2119–2126, 2021. 8

[70] Y. Feng, B. Chen, T. Dai, and S.-T. Xia, "Adversarial attack on deep product quantization network for image retrieval," in *Proc. AAAI. Conf. Artif. Intell.*, 2020, pp. 10 786–10 793. 8

[71] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. 8

[72] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010. 8

[73] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2013, pp. 554–561. 8

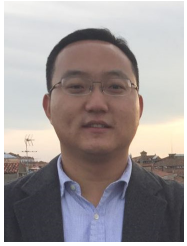
[74] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035. 8

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015. 9

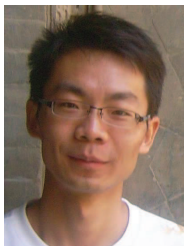
[76] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2820–2828. 14

[77] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2137–2146. 15

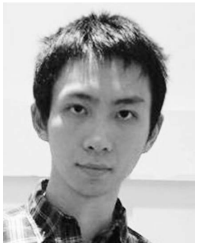
Mo Zhou (Student Member, IEEE) received the B.S. degree in Electromagnetic Fields and Wireless Technology, and the M.S. degree in Pattern Recognition and Intelligent System from Xidian University, Xi'an, China, in 2017 and 2020. He was a Research Assistant with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University, Xi'an, China. His research interests include deep learning, computer vision, cross-modal retrieval, and adversarial attack and defense.



Le Wang (Senior Member, IEEE) received the B.S. and Ph.D. degrees in Control Science and Engineering from Xi'an Jiaotong University, Xi'an, China, in 2008 and 2014, respectively. From 2013 to 2014, he was a visiting Ph.D. student with Stevens Institute of Technology, Hoboken, New Jersey, USA. From 2016 to 2017, he is a visiting scholar with Northwestern University, Evanston, Illinois, USA. He is currently a Professor with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University, Xi'an, China. His research interests include computer vision, pattern recognition, and machine learning. He is an associate editor of Pattern Recognition Letters. He is an area chair of CVPR'2022, ICME'2022, and IAPR'2022, and a senior program committee member of AAAI'2022. He holds 13 China patents and has 17 more China patents pending. He is the author of more than 70 peer-reviewed publications in prestigious international journals and conferences.



Zhenxing Niu (Member, IEEE) received the Ph.D. degree in Control Science and Engineering from Xidian University, Xi'an, China, in 2012. From 2013 to 2014, he was a visiting scholar with University of Texas at San Antonio, Texas, USA. He is a Professor of School of Computer Science and Technology at Xidian University, Xi'an, China. His research interests include computer vision, machine learning, and their application in object discovery and localization. He served as PC member of CVPR, ICCV, and ACM Multimedia. He is an area chair of CVPR'2022.



Qilin Zhang (Member, IEEE) received the B.E. degree in Electrical Information Engineering from the University of Science and Technology of China, Hefei, China, in 2009, and the M.S. degree in Electrical and Computer Engineering from the University of Florida, Gainesville, Florida, USA, in 2011, and the Ph.D. degree in Computer Science from Stevens Institute of Technology, Hoboken, New Jersey, USA, in 2016. He is currently a Computational Imaging Research Engineer at Apple, Cupertino, CA, USA. He was a Senior Research Scientist (2020-2021) with ABB Corporate Research Center, Raleigh, NC, USA. Before that, he was a Senior Research Engineer (2016-2018) and then Lead Research Engineer (2018-2020) with HERE Technologies, Chicago, IL, USA. His research interests include computer vision and signal processing.

Nanning Zheng (Fellow, IEEE) graduated in 1975 from the Department of Electrical Engineering, Xi'an Jiaotong University (XJTU), received the ME degree in Information and Control Engineering from Xi'an Jiaotong University in 1981, and a Ph.D. degree in Electrical Engineering from Keio University in 1985. He is currently a Professor and the Director with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, computational intelligence, and hardware implementation of intelligent systems. Since 2000, he has been the Chinese representative on the Governing Board of the International Association for Pattern Recognition. He became a member of the Chinese Academy Engineering in 1999. He is a fellow of the IEEE.

Gang Hua (Fellow, IEEE) was enrolled in the Special Class for the Gifted Young of Xi'an Jiaotong University (XJTU), Xi'an, China, in 1994 and received the B.S. degree in Automatic Control Engineering from XJTU in 1999. He received the M.S. degree in Control Science and Engineering in 2002 from XJTU, and the Ph.D. degree in Electrical Engineering and Computer Science at Northwestern University, Evanston, Illinois, USA, in 2006. He is currently the Vice President and Chief Scientist of Wormpex AI Research. Before that, he served in various roles at Microsoft (2015-18) as the Science/Technical Adviser to the CVP of the Computer Vision Group, Director of Computer Vision Science Team in Redmond and Taipei ATL, and Principal Researcher/Research Manager at Microsoft Research. He was an Associate Professor at Stevens Institute of Technology (2011-15). During 2014-15, he took an on leave and worked on the Amazon-Go project. He was a Visiting Researcher (2011-14) and a Research Staff Member (2010-11) at IBM Research T. J. Watson Center, a Senior Researcher (2009-10) at Nokia Research Center Hollywood, and a Scientist (2006-09) at Microsoft Live labs Research. He is an associate editor of TIP, TCSVT, CVIU, IEEE Multimedia, TVCJ and MVA. He also served as the Lead Guest Editor on two special issues in TPAMI and IJCV, respectively. He is a general chair of ICCV'2025. He is a program chair of CVPR'2019&2022. He is an area chair of CVPR'2015&2017, ICCV'2011&2017, ICIP'2012&2013&2016, ICASSP'2012&2013, and ACM MM 2011&2012&2015&2017. He is the author of more than 200 peer-reviewed publications in prestigious international journals and conferences. He holds 19 US patents and has 15 more US patents pending. He is the recipient of the 2015 IAPR Young Biometrics Investigator Award for his contribution on Unconstrained Face Recognition from Images and Videos, and a recipient of the 2013 Google Research Faculty Award. He is an IEEE Fellow, an IAPR Fellow, and an ACM Distinguished Scientist.