Practical Relative Order Attack in Deep Ranking

Mo Zhou[1] Le Wang[1*] Zhenxing Niu[2] Qilin Zhang[3] Yinghui Xu[2] Nanning Zheng[1] Gang Hua[4]
[1]Xi'an Jiaotong University [2]Alibaba Group [3]HERE Technologies [4]Wormpex AI Research
{cdluminate,zhenxingniu,samqzhang,ganghua}@gmail.com lewang@xjtu.edu.cn renji.xyh@taobao.com nnzheng@mail.xjtu.edu.cn

Abstract

Recent studies unveil the vulnerabilities of deep ranking models, where an imperceptible perturbation can trigger dramatic changes in the ranking result. While previous attempts focus on manipulating absolute ranks *of certain candidates, the possibility of adjusting their* relative order *remains under-explored. In this paper, we formulate a new adversarial attack against deep ranking systems,* i.e.*, the* Order Attack*, which covertly alters the* relative order *among a selected set of candidates according to an attacker-specified permutation, with limited interference to other unrelated candidates. Specifically, it is formulated as a triplet-style loss imposing an inequality chain reflecting the specified permutation. However, direct optimization of such white-box objective is infeasible in a real-world attack scenario due to various black-box limitations. To cope with them, we propose a* Short-range Ranking Correlation *metric as a surrogate objective for black-box* Order Attack *to approximate the white-box method. The* Order Attack *is evaluated on the Fashion-MNIST and Stanford-Online-Products datasets under both white-box and black-box threat models. The black-box attack is also successfully implemented on a major e-commerce platform. Comprehensive experimental evaluations demonstrate the effectiveness of the proposed methods, revealing a new type of ranking model vulnerability.*

1. Introduction

Thanks to the widespread applications of deep neural networks [26, 21] in the learning-to-rank tasks [51, 41], deep ranking algorithms have witnessed significant progress, but unfortunately they have also inherited the long-standing adversarial vulnerabilities [46] of neural networks. Considering the "search by image" application for example, an imperceptible adversarial perturbation to the query image is sufficient to intentionally alter the ranking results of candidate images. Typically, such adversarial examples can be designed to cause the ranking model to "misrank" [32, 29] (*i.e.*, rank items incorrectly), or purposefully raise or lower the ranks of selected candidates [59].

Since "misranking" can be interpreted as deliberately lowering the ranks of well-matching candidates, previous attacks on ranking models unanimously focus on changing the *absolute ranks* of a set of candidates, while neglecting the manipulation of *relative order* among them. However, an altered *relative order* can be disruptive in some applications, such as impacting sales on e-commerce platforms powered by content-based image retrieval [44], where potential customers attempt to find merchandise via image search.

As shown in Fig. 1, an attacker may want to adversarially perturb the query image and thus change the *relative order* among products A, B, C, D, and E into A $\prec$ E $\prec$ D $\prec$ C $\prec$ B in the search-by-image result. The sales of a product closely correlates to its Click-Through Rate (CTR), while the CTR can be significantly influenced by its ranking position [10, 38] (which also decides the product pagination on the client side). Hence, subtle changes in the *relative order* of searches can be sufficient to alter CTR and impact the actual and relative sales among A to E.

Such vulnerability in a commercial platform may be exploited in a malfeasant business competition among the top-ranked products, *e.g.*, via promoting fraudulent web pages containing adversarial example product images generated in advance. Specifically, the attack of changing *relative order* does not aim to incur significant changes in the *absolute ranks* of the selected candidates (*e.g.*, moving from list bottom to top), but it intentionally changes the *relative order* of them subtly without introducing conspicuous abnormality. Whereas such goal cannot be implemented by *absolute rank* attacks such as [59], the *relative order* vulnerability may justify and motivate a more robust and fair ranking model.

Specifically, we propose the *Order Attack* (OA), a new adversarial attack problem in deep ranking. Given a query image $\boldsymbol{q} \in [0,1]^D$, a set of selected candidates $\mathbb{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$, and a predefined permutation vector $\mathbf{p} = [p_1, p_2, \ldots, p_k]$, *Order Attack* aims to find an imperceptible perturbation $\boldsymbol{r}$ ($\|\boldsymbol{r}\|_\infty \leqslant \varepsilon$ and $\tilde{\boldsymbol{q}} = \boldsymbol{q} + \boldsymbol{r} \in [0,1]^D$), so that $\tilde{\boldsymbol{q}}$ as the adversarial query can convert the *relative order* of the selected candidates into $\boldsymbol{c}_{p_1} \prec \boldsymbol{c}_{p_2} \prec \cdots \prec \boldsymbol{c}_{p_k}$. For example, a successful OA with $\mathbf{p} = [1,5,4,3,2]$ will result in $\boldsymbol{c}_1 \prec \boldsymbol{c}_5 \prec \boldsymbol{c}_4 \prec \boldsymbol{c}_3 \prec \boldsymbol{c}_2$, as shown in Fig. 1.

*Corresponding author.

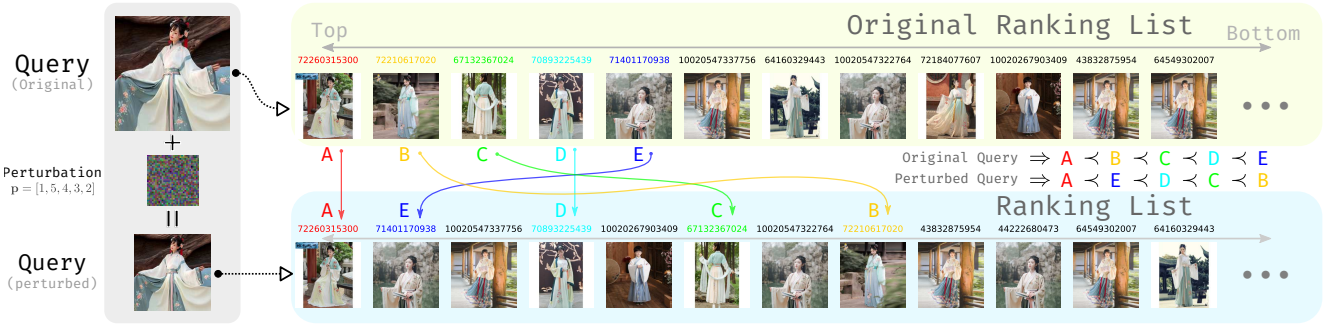

Figure 1: Showcase of a practical *Order Attack* (OA) against "JD SnapShop", a major online retailing e-commerce platform. The query image is "Han Chinese clothing". Numbers atop candidate images are Stock Keep Unit (SKU) IDs.

To implement OA, we first assume the *white-box* threat model (*i.e.*, the ranking model details, *incl.* the gradient are accessible to the attacker). Recall that a conventional deep ranking model [51, 58, 41, 25] maps the query and candidates onto a common embedding space, and determines the ranking list according to the pairwise similarity between the query and these candidates. Thus, OA can be formulated as the optimization of a triplet-style loss function based on the inequality chain representing the desired *relative order*, which simultaneously adjusts the similarity scores between the query and the selected candidates. Additionally, a semantics-preserving penalty term [59] is also included to limit conspicuous changes in ranking positions. Finally, the overall loss function can be optimized with gradient methods such as PGD [33] to find the adversarial example.

However, in a real-world *black-box* attack scenario, practical limitations (*e.g.*, gradient inaccessibility) invalidate the proposed method. To accommodate them and make OA practical, we propose a "Short-range Ranking Correlation" (SRC) metric to measure the alignment between a desired permutation and the actual ranking result returned to clients by counting concordant and discordant pairs, as a practical approximation of the proposed triplet-style white-box loss. Though non-differentiable, SRC can be used as a surrogate objective for black-box OA and optimized by an appropriate black-box optimizer, to achieve similar effect as the white-box OA. SRC can also be used as a performance metric for the white-box method, as it gracefully degenerates into Kendall's ranking correlation [24] in white-box scenario.

To validate the white-box and black-box OA, we conduct comprehensive experiments on Fashion-MNIST and Stanford-Online-Product datasets. To illustrate the viability of the black-box OA in practice, we also showcase successful attacks against the "JD SnapShop" [23], a major retailing e-commerce platform based on content-based image retrieval. Extensive quantitative and qualitative evaluations illustrate the effectiveness of the proposed OA, and reveals a new type of ranking model vulnerability.

To the best of our knowledge, this is the first work that tampers the *relative order* in deep ranking. We believe our contributions include, **(1)** the formulation of *Order Attack* (OA), a new adversarial attack that covertly alters the *relative order* among selected candidates; **(2)** a triplet-style loss for ideal-case white-box OA; **(3)** a *Short-range Ranking Correlation* (SRC) metric as a surrogate objective approximating the triplet-style loss for practical black-box OA; **(4)** extensive evaluations of OA including a successful demonstration on a major online retailing e-commerce platform.

2. Related Works

Adversarial Attack. Szegedy *et al*. [46] find the DNN classifiers susceptible to imperceptible adversarial perturbations, which leads to misclassification. This attracted research interest among the community, as shown by subsequent works on adversarial attacks and defenses [14, 12, 48]. In particular, the attacks can be categorized into several groups: (1) White-box attack, which assumes the model details including the gradient are fully accessible [19, 27, 33, 35, 7, 2, 3, 13]. Of these methods, PGD [33] is the most popular one; (2) Transfer-based attack, which is based on the transferability of adversarial examples [15, 54, 16]. Such attack typically transfers adversarial examples found from a locally trained substitute model onto another model. (3) Score-based attack, which only depends on the soft classification labels, *i.e.*, the logit values [22, 49, 31, 9, 1]. Notably, [22] proposes a black-box threat model for classification that is similar to our black-box ranking threat model; (4) Decision-based attack, a type of attack that requires the least amount of information from the model, *i.e.*, the hard label (one-hot vector) [6, 8, 11, 17, 43, 28]. All these extensive adversarial attacks unanimously focus on classification, which means they are not directly suitable for ranking scenarios.

Adversarial Ranking. In applications such as web retrieval, documents may be promoted in rankings by intentional manipulation [20]. Likewise, the existence of aforementioned works inspired attacks against deep rank-

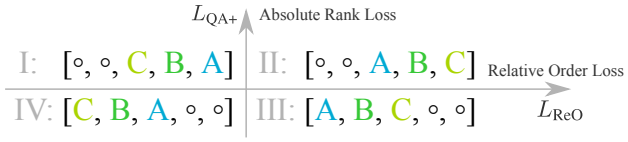

Figure 2: Relative order attack *v.s.* absolute rank attack.

ing, but it is still insufficiently explored. In light of distinct purposes, ranking attacks can be divided into *absolute rank* attacks and *relative order* attacks. Most *absolute rank* attacks attempt to induce random "misranking" [47, 29, 32, 55, 56, 50, 57, 5, 18, 30]. Some other attacks against ranking model aim to incur purposeful changes in *absolute rank*, *i.e.*, to raise or lower the ranks of specific candidates [4, 59]. On the contrary, the *relative order* attacks remains under-explored. And this is the first work that tampers the *relative order* in deep ranking.

As shown in Fig. 2, *relative order* is "orthogonal" to *absolute rank*. Let "∘" denote any uninterested candidate. Suppose the selected candidates $\mathbb{C}$ and permutation $\mathbf{p}$ are [A, B, C] and $[3, 2, 1]$, respectively. The exemplar *absolute rank* loss $L_{\text{QA+}}$ [59] is ignorant to the difference in *relative order* comparing I and II, or IV and III. The *relative order* loss L_{ReO} proposed in this paper is ignorant to the difference in *absolute rank* comparing I and IV, or II and III. Although focusing on different aspects of ranking, the two types of loss functions can be combined and jointly optimized.

3. Adversarial Order Attack

Typically, a deep ranking model is built upon deep metric learning [51, 41, 58, 25, 40]. Given a query $\boldsymbol{q}$ and a set of candidates $\mathbb{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ selected from database $\mathbb{D}$ ($\mathbb{C} \subset \mathbb{D}$), a deep ranking model f evaluates the distance between every pair of query and candidate, *i.e.*, $f : \mathcal{I} \times \mathcal{I} \mapsto \mathbb{R}$ where $\mathcal{I} = [0, 1]^D$. Thus, by comparing all the pairwise distances $\{f(\boldsymbol{q}, \boldsymbol{c}_i) | i = 1, 2, \ldots, k\}$, the whole candidate set can be ranked with respect to the given query. For instance, the model outputs the ranking list $\boldsymbol{c}_1 \prec \boldsymbol{c}_2 \prec \cdots \prec \boldsymbol{c}_k$ if it determines $f(\boldsymbol{q}, \boldsymbol{c}_1) < f(\boldsymbol{q}, \boldsymbol{c}_2) < \cdots < f(\boldsymbol{q}, \boldsymbol{c}_k)$.

Based on these, *Order Attack* (OA) aims to find an imperceptible perturbation $\boldsymbol{r}$ ($\|\boldsymbol{r}\|_\infty \leqslant \varepsilon$ and $\tilde{\boldsymbol{q}} = \boldsymbol{q} + \boldsymbol{r} \in \mathcal{I}$), so that $\tilde{\boldsymbol{q}}$ as the adversarial query can convert the *relative order* of the selected candidates into $\boldsymbol{c}_{p_1} \prec \boldsymbol{c}_{p_2} \prec \cdots \prec \boldsymbol{c}_{p_k}$, where $\mathbf{p} = [p_1, p_2, \ldots, p_k]$ is a permutation vector predefined by the attacker. In particular, we assume that the attacker is inclined to select the candidate set $\mathbb{C}$ from the top-N ranked candidates $\mathbb{X}$, as the ranking lists returned to the clients are usually "truncated" (*i.e.*, only the top-ranked candidates will be shown). We call the length N ($N \geqslant k$) of the "truncated" list as a "visible range". The white-box and black-box OA will be discussed in Sec.3.1 and Sec.3.2 respectively. For sake of brevity, we let $\Omega_{\boldsymbol{q}} = \{\boldsymbol{r} | \ \boldsymbol{q} + \boldsymbol{r} \in \mathcal{I}, \ \|\boldsymbol{r}\|_\infty \leqslant \varepsilon\}$.

3.1. Triplet-style Loss Function for White-Box OA

During the training process, a typical deep ranking model f involves a triplet (anchor $\boldsymbol{q}$, positive $\boldsymbol{c}_{\text{p}}$, negative $\boldsymbol{c}_{\text{n}}$) in each iteration. In order to rank $\boldsymbol{c}_{\text{p}}$ ahead of $\boldsymbol{c}_{\text{n}}$, the model is penalized when $f(\boldsymbol{q}, \boldsymbol{c}_{\text{p}}) + \gamma < f(\boldsymbol{q}, \boldsymbol{c}_{\text{n}})$ does not hold. This inequality can be reformulated exploiting the form of a hinge loss [39], resulting in the triplet ranking loss function [41] $L_{\text{triplet}}(\boldsymbol{q}, \boldsymbol{c}_{\text{p}}, \boldsymbol{c}_{\text{n}}) = [\gamma + f(\boldsymbol{q}, \boldsymbol{c}_{\text{p}}) - f(\boldsymbol{q}, \boldsymbol{c}_{\text{n}})]^+$, where $[\cdot]^+ = \max(0, \cdot)$, and γ denotes the margin hyper-parameter.

Inspired by this, to implement the OA, we decompose the inequality chain prescribed by the predefined permutation vector $\mathbf{p}$, namely $f(\tilde{\boldsymbol{q}}, \boldsymbol{c}_{p_1}) < f(\tilde{\boldsymbol{q}}, \boldsymbol{c}_{p_2}) < \cdots < f(\tilde{\boldsymbol{q}}, \boldsymbol{c}_{p_k})$ into $\binom{k}{2} = k(k-1)/2$ inequalities, *i.e.*, $f(\tilde{\boldsymbol{q}}, \boldsymbol{c}_{p_i}) < f(\tilde{\boldsymbol{q}}, \boldsymbol{c}_{p_j})$, $i, j = 1, 2, \ldots, k, \ i < j$. Reformulation of these inequalities into the hinge loss form leads to the *relative order* loss,

$$L_{\text{ReO}}(\tilde{\boldsymbol{q}}; \mathbb{C}, \mathbf{p}) = \sum_{i=1}^{k} \sum_{j=i}^{k} \left[f(\tilde{\boldsymbol{q}}, \boldsymbol{c}_{p_i}) - f(\tilde{\boldsymbol{q}}, \boldsymbol{c}_{p_j}) \right]^+. \quad (1)$$

Subsequently, given this loss function, the OA can be cast as a constrained optimization problem,

$$\boldsymbol{r}^* = \arg\min_{\boldsymbol{r} \in \Omega_{\boldsymbol{q}}} L_{\text{ReO}}(\boldsymbol{q} + \boldsymbol{r}; \mathbb{C}, \mathbf{p}), \quad (2)$$

which can be solved with first-order-gradient-based methods such as Projected Gradient Descent (PGD) [33], *i.e.*,

$$\boldsymbol{r}_{t+1} = \text{Clip}_{\Omega_{\boldsymbol{q}}} \{ \boldsymbol{r}_t - \eta \text{sign} \left[\nabla_{\boldsymbol{r}} L_{\text{ReO}}(\tilde{\boldsymbol{q}}; \mathbb{C}, \mathbf{p}) \right] \}, \quad (3)$$

where η is the PGD step size, and $\boldsymbol{r}_0$ is initialized as a zero vector. PGD stops at a predefined maximum iteration T, and the final $\boldsymbol{r}_T$ is the desired adversarial perturbation.

It is worth noting that query image semantics can be drastically changed even with a very slight perturbation [59]. As a result, candidates $\mathbb{C}$ are prone to be excluded from the topmost part of ranking, and become invisible when the ranking result is "truncated". To mitigate such side effect, we follow [59] and introduce a semantics-preserving term $L_{\text{QA+}}(\tilde{\boldsymbol{q}}, \mathbb{C})$ to keep $\mathbb{C}$ within the topmost part of the ranking by raising their *absolute ranks*, *i.e.*, to keep $\boldsymbol{c} \in \mathbb{C}$ ranked ahead of other candidates. Finally, the *relative order* loss term $L_{\text{ReO}}(\cdot)$ and the *absolute rank* loss term $L_{\text{QA+}}(\cdot)$ are combined to form the complete white-box OA loss L_{OA},

$$L_{\text{OA}}(\tilde{\boldsymbol{q}}; \mathbb{C}, \mathbf{p}) = L_{\text{ReO}}(\tilde{\boldsymbol{q}}; \mathbb{C}, \mathbf{p}) + \xi L_{\text{QA+}}(\tilde{\boldsymbol{q}}, \mathbb{C}), \quad (4)$$

where ξ is a positive constant balancing factor between the *relative order* and *absolute rank* goals.

Despite the formulation of Eq. (4), an ideal $\tilde{\boldsymbol{q}}$ that fully satisfy the desired relative order of $\mathbb{C}$ does not necessarily exist. Consider a Euclidean embedding space, where candidates $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$ lie consecutively on a straight line. It is impossible to find a query embedding that leads to $\boldsymbol{c}_1 \prec \boldsymbol{c}_3 \prec \boldsymbol{c}_2$. That

indicates the compatibility between the specified relative order and the factual geometric relations of the candidate embeddings affects the performance upper-bound of OA. In cases like this, our algorithm can still find an inexact solution that satisfies as many inequalities as possible to approximate the specified *relative order*. In light of this, Kendall's ranking correlation τ [24] between the specified relative order and the real ranking order appears to be a more reasonable performance metric than the success rate for OA.

3.2. Short-range Ranking Correlation

A concrete triplet-style implementation of OA is present in Sec. 3.1, but it is infeasible in a real-world attack scenario. In particular, multiple challenges are present for black-box OA, including (1) *Gradient inaccessibility*. The gradient of the loss *w.r.t* the input is inaccessible, as the network architecture and parameters are unknown; (2) *Lack of similarity (or distance) scores*. Exact similarity scores rarely appear in the truncated ranking results; (3) *Truncated ranking results*. In practice, a ranking system only presents the top-N ranking results to the clients; (4) *Limited query budget*. Repeated, intensive queries within a short time frame may be identified as threats, *e.g.*, Denial of Service (DoS) attack. Therefore, it is preferable to construct adversarial examples within a reasonable amount of queries. These restrictions collectively invalidate the triplet-style method.

To address these challenges, we present the "Short-range Ranking Correlation" (SRC; denoted as $\tau_{\mathcal{S}}$) metric, a practical approximation of the L_{OA} (Eq. 4) as a surrogate loss function for black-box OA.

Specifically, to calculate $\tau_{\mathcal{S}}$ given $\mathbb{C}$, $\mathbf{p}$ and the top-N retrieved candidates $\mathbb{X}$ *w.r.t* query $\tilde{\boldsymbol{q}}$, we first initialize a $(k \times k)$-shaped zero matrix $\boldsymbol{S}$, and permute $\mathbb{C}$ into $\mathbb{C}_{\mathbf{p}} = \{\boldsymbol{c}_{p_1}, \boldsymbol{c}_{p_2}, \dots, \boldsymbol{c}_{p_k}\}$. Assuming $\forall \boldsymbol{c}_i, \boldsymbol{c}_j \in \mathbb{C}_{\mathbf{p}}$ $(i > j, i \neq j)$ exist in $\mathbb{X}$, we define $(\boldsymbol{c}_i, \boldsymbol{c}_j)$ as a *concordant* pair as long as $\text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_i)$ and $\text{R}_{\mathbb{X}}(\boldsymbol{c}_i)$ are simultaneously greater or smaller than $\text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_j)$ and $\text{R}_{\mathbb{X}}(\boldsymbol{c}_j)$, respectively, where $\text{R}_{\mathbb{X}}(\boldsymbol{c}_i)$ denotes the integer rank value of $\boldsymbol{c}_i$ in $\mathbb{X}$, *i.e.*, $\text{R}_{\mathbb{X}}(\boldsymbol{c}_i) := \arg_m\{\boldsymbol{c}_i = \boldsymbol{x}_m\}$. Otherwise, $(\boldsymbol{c}_i, \boldsymbol{c}_j)$ is defined as a *discordant* pair. Namely a *concordant* matches a part of the specified permutation, and could result in a zero loss term in Eq. 1, while a *discordant* pair does not match, and could result in a positive loss term in Eq. 1. Thus, in order to approximate the *relative order* loss L_{ReO} (Eq. 1), a *concordant* pair and a *discordant* pair will be assigned a score of $S_{i,j} = +1$ (as reward) and $S_{i,j} = -1$ (as penalty), respectively. Apart from that, when $\boldsymbol{c}_i$ or $\boldsymbol{c}_j$ does not exist in $\mathbb{X}$, $S_{i,j}$ will be directly assigned with an "out-of-range" penalty -1, which approximates the semantics-preserving term in Eq. 4. Finally, after comparing the ordinal relationships of every pair of candidates and assigning penalty values in $\boldsymbol{S}$, the average score of the lower triangular of $\boldsymbol{S}$ excluding the diagonal is the $\tau_{\mathcal{S}}$, as summarized in Algo. 1.

Algorithm 1: Short-range Ranking Correlation $\tau_{\mathcal{S}}$.

Input: Selected candidates $\mathbb{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_k\}$,
permutation vector $\mathbf{p} = [p_1, p_2, \dots, p_k]$,
top-N retrieval $\mathbb{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ for $\tilde{\boldsymbol{q}}$.
Note that $\mathbb{C} \subset \mathbb{D}$, $\mathbb{X} \subset \mathbb{D}$, and $N \geqslant k$.
Output: SRC coefficient $\tau_{\mathcal{S}}$.
Permute candidates as $\mathbb{C}_{\mathbf{p}} = \{\boldsymbol{c}_{p_1}, \boldsymbol{c}_{p_2}, \dots, \boldsymbol{c}_{p_k}\}$;
Initialize score matrix $\boldsymbol{S} = \mathbf{0}$ of size $k \times k$;
for $i \leftarrow 1, 2, \dots, k$ **do**
  for $j \leftarrow 1, 2, \dots, i-1$ **do**
    if $\boldsymbol{c}_i \notin \mathbb{X}$ [1] or $\boldsymbol{c}_j \notin \mathbb{X}$ **then**
      $S_{i,j} = -1$ `// out-of-range`
    else if $\left[\text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_i) > \text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_j) \text{ and } \text{R}_{\mathbb{X}}(\boldsymbol{c}_i) > \text{R}_{\mathbb{X}}(\boldsymbol{c}_j)\right]$
     or $\left[\text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_i) < \text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_j) \text{ and } \text{R}_{\mathbb{X}}(\boldsymbol{c}_i) < \text{R}_{\mathbb{X}}(\boldsymbol{c}_j)\right]$
     then
      $S_{i,j} = +1$ `// concordant`
    else if $\left[\text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_i) > \text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_j) \text{ and } \text{R}_{\mathbb{X}}(\boldsymbol{c}_i) < \text{R}_{\mathbb{X}}(\boldsymbol{c}_j)\right]$
     or $\left[\text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_i) < \text{R}_{\mathbb{C}_{\mathbf{p}}}(\boldsymbol{c}_j) \text{ and } \text{R}_{\mathbb{X}}(\boldsymbol{c}_i) > \text{R}_{\mathbb{X}}(\boldsymbol{c}_j)\right]$
     then
      $S_{i,j} = -1$ `// discordant`
return $\tau_{\mathcal{S}} = \sum_{i,j} S_{i,j} / \binom{k}{2}$

The value of $\tau_{\mathcal{S}} \in [-1, 1]$ reflects the real ranking order's alignment to the order specified by $\mathbf{p}$, where a semantics-preserving penalty is spontaneously incorporated. When the specified order is fully satisfied, *i.e.*, any pair of $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ is *concordant*, and none of the elements in $\mathbb{C}$ disappear from $\mathbb{X}$, $\tau_{\mathcal{S}}$ will be 1. In contrast, when every candidate pair is *discordant* or absent from the top-N result $\mathbb{X}$, $\tau_{\mathcal{S}}$ will be -1. Overall, $(\tau_{\mathcal{S}} + 1)/2$ percent of the candidate pairs are *concordant*, and the rest are *discordant* or "out-of-range".

Maximization of $\tau_{\mathcal{S}}$ leads to the best alignment to the specified permutation, as discordant pairs will be turned into concordant pairs, while maintaining the presence of $\mathbb{C}$ within the top-N visible range. Thus, although non-differentiable, the $\tau_{\mathcal{S}}$ metric can be used as a practical surrogate objective for black-box OA, *i.e.*, $\boldsymbol{r}^* = \arg\max_{\boldsymbol{r} \in \Omega_{\boldsymbol{q}}} \tau_{\mathcal{S}}(\tilde{\boldsymbol{q}}; \mathbb{C}, \mathbf{p})$, which achieves a very similar effect to the white-box OA.

Particularly, when $\forall \boldsymbol{c} \in \mathbb{C}$ exists in $\mathbb{X}$, $\tau_{\mathcal{S}}$ degenerates into Kendall's τ [24] between $\mathbf{p}$ and the permutation of $\mathbb{C}$ in $\mathbb{X}$. Namely, $\tau_{\mathcal{S}}$ also degenerates gracefully to τ in the white-box scenario because the whole ranking is visible. However, τ is inapplicable on truncated ranking results.

$\tau_{\mathcal{S}}$ does not rely on any gradient or any similarity score, and can adapt to truncated ranking results. When optimized with an efficient black-box optimizer (*e.g.*, NES [22]), the limited query budget can also be efficiently leveraged. Since all the black-box challenges listed at the beginning of this section are handled, it is practical to perform black-box OA by optimizing $\tau_{\mathcal{S}}$ in real-world applications.

[1] $\nexists m \in \{1, 2, \dots, N\}$ so that $\boldsymbol{c}_i = \boldsymbol{x}_m$.

	$k=5$					$k=10$					$k=25$				
ε	0	$\frac{2}{255}$	$\frac{4}{255}$	$\frac{8}{255}$	$\frac{16}{255}$	0	$\frac{2}{255}$	$\frac{4}{255}$	$\frac{8}{255}$	$\frac{16}{255}$	0	$\frac{2}{255}$	$\frac{4}{255}$	$\frac{8}{255}$	$\frac{16}{255}$
Fashion-MNIST $N=\infty$															
τ_S	0.000	0.286	0.412	0.548	0.599	0.000	0.184	0.282	0.362	0.399	0.000	0.063	0.108	0.136	0.149
mR	2.0	4.5	9.1	12.7	13.4	4.5	7.4	10.9	15.2	17.4	12.0	16.1	17.6	18.9	19.4
Stanford Online Products $N=\infty$															
τ_S	0.000	0.396	0.448	0.476	0.481	0.000	0.263	0.348	0.387	0.398	0.000	0.125	0.169	0.193	0.200
mR	2.0	5.6	4.9	4.2	4.1	4.5	12.4	11.2	9.9	9.6	12.0	31.2	28.2	25.5	25.4

Table 1: White-box order attack on Fashion-MNIST and SOP datasets with various settings.

ξ	0	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}	10^{4}	10^{5}	10^{6}	10^{7}
Fashion-MNIST $k=5, N=\infty, \varepsilon=4/255$										
τ_S	0.561	0.467	0.451	*0.412*	0.274	0.052	0.043	0.012	0.007	0.002
mR	27.2	22.7	18.3	*9.1*	4.9	3.2	2.8	2.7	2.7	2.7
Stanford Online Products $k=5, N=\infty, \varepsilon=4/255$										
τ_S	0.932	0.658	0.640	0.634	0.596	*0.448*	0.165	0.092	0.013	0.001
mR	973.9	89.8	48.1	22.4	7.5	*4.9*	2.9	2.8	2.8	2.7

Table 2: Searching for balancing factor ξ on both datasets.

4. Experiments

To evaluate the white-box and black-box OA, we conduct experiments on the Fashion-MNIST [53] and the Stanford-Online-Products (SOP) [36] datasets which comprise images of retail commodity. Firstly, we train a CNN with 2-convolution-1-fully-connected network on Fashion-MNIST, and a ResNet-18 [21] without the last fully-connected layer on SOP following [59] that focuses on the *absolute rank* attack. Then we perform OA with the corresponding test set as the candidate database $\mathbb{D}$. Additionally, we also qualitatively evaluate black-box OA on "JD SnapShop" [23] to further illustrate its efficacy. In our experiments, the value of rank function $\mathrm{R}_{\mathbb{X}}(\cdot)$ starts from 0, *i.e.*, the k-th ranked candidate has the rank value of $k-1$.

Selection of $\mathbb{C}$ and $\mathbf{p}$. As discussed, we assume that the attacker is inclined to select the candidate set $\mathbb{C}$ from the top-N ranked candidates given the visible range limit. For simplicity, we only investigate the (k, N)-OA, *i.e.*, OA with the top-k-within-top-N ($k \leq N$) candidates selected as as $\mathbb{C}$. It is representative because an OA problem with some candidates randomly selected from the top-k results as $\mathbb{C}$ is a sub-problem of (k, N)-OA. Namely, our attack will be effective for any selection of $\mathbb{C}$ as long as the (k, N)-OA is effective. For white-box OA, we conduct experiments with $N{=}\infty$, and $k{\in}\{5, 10, 25\}$. For black-box attack, we conduct experiments with $N{=}\{\infty, 50, k\}$, and $k{=}\{5, 10, 25\}$. A random permutation vector $\mathbf{p}$ is specified for each query.

Evaluation Metric. Since τ_S is equivalent to τ when $N=\infty$, we use τ_S as the performance metric for both white-box and black-box OA. Specifically, in each experiment, we conduct $T = 10^4$ times of OA attack. In each attack, we randomly draw a sample from $\mathbb{D}$ as the query $\boldsymbol{q}$. In the end, we report the average τ_S over the T trials. Also, when $N=\infty$, we additionally calculate the mean rank of the candidate set $\mathbb{C}$ (demoted as "mR", which equals $[\sum_i^k \mathrm{R}_{\mathbb{X}}(\boldsymbol{c}_i)]/k$), and report the average mean rank over the T attacks. Larger τ_S value and smaller mR value are preferable.

Parameter Settings. We set the perturbation budget as $\varepsilon \in \{\frac{2}{255}, \frac{4}{255}, \frac{8}{255}, \frac{16}{255}\}$ following [27] for both white-box and black-box attacks. The query budget Q is set to $1.0{\times}10^3$. For white-box OA, the PGD step size η is set to $\frac{1}{255}$, the PGD step number to 24. The balancing parameter ξ is set as 10^1 and 10^3 for Fashion-MNIST and SOP respectively. The learning rates of black-box optimizer NES [22] and SPSA [49] are both set to $2/255$. See supplementary for more details of the black-box optimizers.

Search Space Dimension Reduction. As a widely adopted trick, dimension reduction of the adversarial perturbation search space has been reported effective in [14, 9, 43, 28]. Likewise, we empirically reduce the space to $(3\times32\times32)$ for black-box OA on Stanford-Online-Products dataset and "JD SnapShop". In fact, a significant performance drop in τ_S can be observed without this trick.

4.1. White-Box Order Attack Experiments

The first batch of the experiments is carried out on the Fashion-MNIST dataset, as shown in the upper part of Tab. 1. With the original query image ($\varepsilon = 0$), the expected τ_S performance of $(5, \infty)$-OA is 0.000, and the $\mathbb{C}$ retains their original ranks as the mR equals 2.0. With a $\varepsilon = 2/255$ adversarial perturbation budget, our OA achieves $\tau_S = 0.286$, which means on average 64.3% [2] of the inequalities reflecting the specified permutations are satisfied by the adversarial examples. Meanwhile, the mR changes from 2.0 to 4.5, due to adversarial perturbation can move the query embedding off its original position [59] while seeking for a higher τ_S. Nevertheless, the mR value of 4.5 indicates that the $\mathbb{C}$ are still kept visible in the topmost part of the ranking result by the loss term $L_{\mathrm{QA+}}(\cdot)$. With larger perturbation budget ε, the τ_S metric increases accordingly, *e.g.*, τ_S reaches 0.599 when $\varepsilon = 16/255$, which means nearly 80% of the inequalities are satisfied. Likewise, the experimental results on SOP are available in the lower part of Tab. 1, which also demonstrate the effectiveness of our method under different settings.

[2] Solution of $\frac{(n_{\text{concordant}}-n_{\text{discordant}})}{\binom{k}{2}}=0.286$; $\frac{(n_{\text{concordant}}+n_{\text{discordant}})}{\binom{k}{2}}=1.0$.

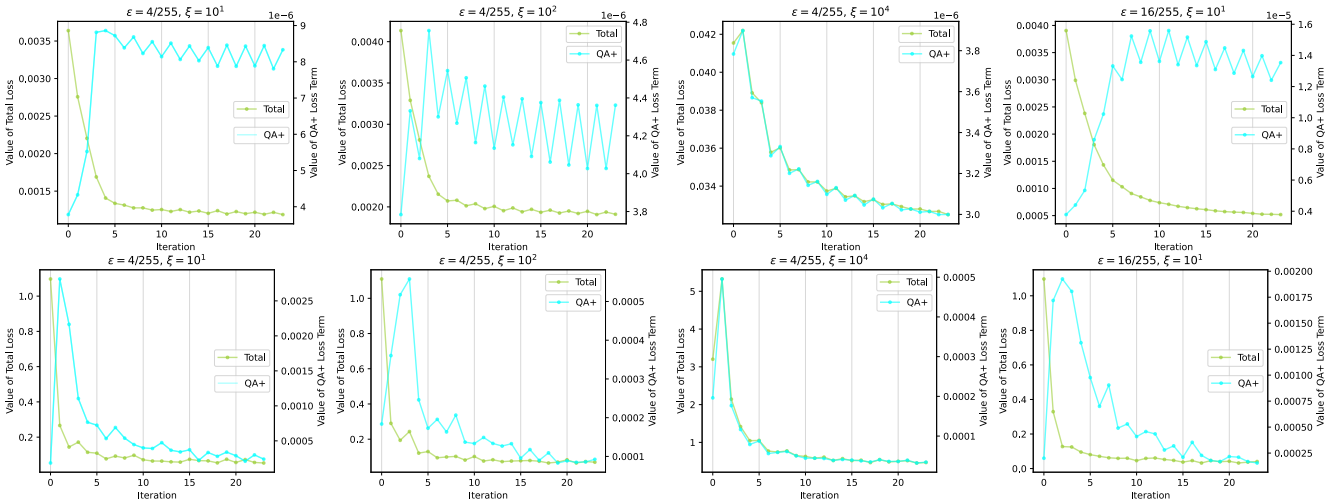

Figure 3: Curves of total loss L_{OA} (left y-axis) and the $L_{\text{QA+}}$ term (right y-axis) during the optimization procedures under different ε and ξ settings. The first row is for Fashion-MNIST, while the second row is for SOP dataset.

Besides, we note that different balancing parameter ξ for $L_{\text{QA+}}(\cdot)$ leads to distinct results, as shown in Tab. 2. We conduct $(5, \infty)$-OA with $\varepsilon = 4/255$ with different ξ values ranging from 0 to 10^7 on both datasets. Evidently, a larger ξ leads to a better (smaller) mR value, but meanwhile a worse $\tau_{\mathcal{S}}$ as the weighted $L_{\text{QA+}}(\cdot)$ term dominates the total loss. There is a trade-off between the $\tau_{\mathcal{S}}$ and mR, which is effectively controlled by the tunable constant parameter ξ. Hence, we empirically set ξ as 10^1 and 10^3 for Fashion-MNIST and SOP respectively, in order to keep the mR of most experiments in Tab. 1 below a sensible value, *i.e.*, $50/2$.

Additionally, Tab. 1 reveals that the mR trends w.r.t. ε on the two datasets differ. To investigate this counter-intuitive phenomenon, we plot loss curves in Fig. 3. In the $\varepsilon = 4/255$, $\xi = 10$ case for Fashion-MNIST, the total loss decreases but the $L_{\text{QA+}}$ surges at the beginning and then plateaus. After increasing ξ to 10^2, the $L_{\text{QA+}}$ rises more smoothly. The curve eventually decreases at $\xi = 10^4$, along with a small mR and a notable penalty on $\tau_{\mathcal{S}}$ as a result. Besides, the "sawtooth-shaped" $L_{\text{QA+}}$ curves also indicate that the L_{ReO} term is optimized while sacrificing the mR as a side-effect at the even steps, while the optimizer turns to optimize $L_{\text{QA+}}$ at the odd steps due to the semantics-preserving penalty, causing a slight increase in L_{ReO}. These figures indicate that optimizing L_{ReO} without sacrificing $L_{\text{QA+}}$ is difficult. Moreover, perturbation budget is irrelevant. Comparing the first and the fourth sub-figures, we find a larger budget ($\varepsilon{=}\frac{16}{255}$) unhelpful in reducing optimization difficulty as the $L_{\text{QA+}}$ curve still soars and plateaus. Based on these cues, we speculate that the different curve patterns of mR stem from the optimization difficulty due to a *fixed PGD step size* that cannot be smaller[3], and *different dataset properties*.

[3]Every element of perturbation should be an integral multiple of $1/255$.

The intra-class variance of the simple Fashion-MNIST dataset is smaller than that of SOP, which means sample embeddings of the same class are densely clustered. As each update can change the $\mathbb{X}$ drastically, it is difficult to adjust the query embedding position in a dense area with a fixed PGD step for a higher $\tau_{\mathcal{S}}$ without significantly disorganizing the ranking list (hence a lower mR). In contrast, a larger intra-class variance of the SOP dataset makes $L_{\text{QA+}}$ easier to be maintained, as shown in the 2nd row of Fig. 3.

4.2. Black-Box Order Attack Experiments

To simulate a real-world attack scenario, we convert the ranking models trained for Sec. 4.1 into black-box versions, which are subject to limitations discussed in Sec. 3.2. Black-box OA experiments are conducted on these models.

To optimize the surrogate loss $\tau_{\mathcal{S}}$, we adopt and compare several black-box optimizers: (1) Random Search (Rand), which independently samples every dimension of $\boldsymbol{r}$ from uniform distribution $\mathcal{U}(-\varepsilon, +\varepsilon)$, then clips it to $\Omega_{\boldsymbol{q}}$; (2) Beta-Attack (Beta), a modification of $\mathcal{N}$-Attack [31] that generates the $\boldsymbol{r}$ from an iteratively-updated Beta distribution (instead of Gaussian) per dimension; (3) Particle Swarm Optimization (PSO) [42], a classic meta-heuristic black-box optimizer with an extra step that clips the adversarial perturbation to $\Omega_{\boldsymbol{q}}$; (4) NES [22, 52], which performs PGD [33] using estimated gradient; (5) SPSA [49, 45], which can be interpreted as NES using a different sampling distribution.

We first investigate the black-box $(5, \infty)$-OA, as shown in the upper part ($N{=}\infty$) of Tab. 3 and Tab. 5. In these cases, $\tau_{\mathcal{S}}$ does not pose any semantics-preserving penalty since $N{=}\infty$, which is similar to white-box attack with $\xi{=}0$. With the Rand optimizer, $\tau_{\mathcal{S}}$ can be optimized to 0.309 with $\varepsilon{=}\frac{4}{255}$ on Fashion-MNIST. As ε increases, we obtain better $\tau_{\mathcal{S}}$ results, and larger mR values as an expected side-effect.

Algorithm	$k=5$				$k=10$				$k=25$			
	$\varepsilon=\frac{2}{255}$	$\varepsilon=\frac{4}{255}$	$\varepsilon=\frac{8}{255}$	$\varepsilon=\frac{16}{255}$	$\varepsilon=\frac{2}{255}$	$\varepsilon=\frac{4}{255}$	$\varepsilon=\frac{8}{255}$	$\varepsilon=\frac{16}{255}$	$\varepsilon=\frac{2}{255}$	$\varepsilon=\frac{4}{255}$	$\varepsilon=\frac{8}{255}$	$\varepsilon=\frac{16}{255}$
None	0.0, 2.0	0.0, 2.0	0.0, 2.0	0.0, 2.0	0.0, 4.5	0.0, 4.5	0.0, 4.5	0.0, 4.5	0.0, 12.0	0.0, 12.0	0.0, 12.0	0.0, 12.0
Fasion-MNIST $N=\infty$												
Rand	0.211, 2.1	0.309, 2.3	0.425, 3.0	0.508, 7.7	0.172, 4.6	0.242, 5.0	0.322, 6.4	0.392, 12.7	0.084, 12.3	0.123, 13.1	0.173, 15.8	0.218, 25.8
Beta	0.241, 2.1	0.360, 2.6	0.478, 4.6	0.580, 19.3	0.210, 4.8	0.323, 5.7	0.430, 9.6	0.510, 30.3	0.102, 12.4	0.163, 13.8	0.237, 19.7	0.291, 42.7
PSO	0.265, 2.1	0.381, 2.3	0.477, 4.4	0.580, 21.1	0.239, 4.8	0.337, 5.7	0.424, 9.7	0.484, 34.0	0.131, 12.7	0.190, 14.6	0.248, 21.7	0.286, 54.2
NES	0.297, 2.3	**0.416, 3.1**	**0.520, 8.7**	**0.630, 46.3**	**0.261, 5.0**	0.377, 6.6	0.473, 14.3	0.518, 55.6	**0.142, 13.0**	0.217, 15.9	0.286, 28.3	0.312, 74.3
SPSA	**0.300, 2.3**	0.407, 3.2	0.465, 7.1	0.492, 16.3	0.249, 5.0	**0.400, 6.6**	**0.507, 12.8**	**0.558, 27.5**	0.135, 12.9	**0.236, 16.3**	**0.319, 27.1**	**0.363, 46.4**
Fashion-MNIST $N=50$												
Rand	0.207	0.316	0.424	0.501	0.167	0.242	0.321	0.378	0.083	0.123	0.165	0.172
Beta	0.240	0.359	0.470	0.564	0.204	0.323	0.429	0.487	0.103	0.160	0.216	0.211
PSO	0.266	0.377	0.484	0.557	0.239	0.332	0.420	0.458	0.134	0.183	0.220	0.203
NES	**0.297**	**0.426**	**0.515**	**0.584**	**0.262**	0.378	0.463	0.458	**0.141**	0.199	0.223	0.185
SPSA	0.292	0.407	0.468	0.490	0.253	**0.397**	**0.499**	**0.537**	0.131	**0.214**	**0.260**	**0.275**
Fashion-MNIST $N=k$												
Rand	0.204	0.289	0.346	0.302	0.146	0.181	0.186	0.124	0.053	0.062	0.049	0.021
Beta	0.237	0.342	0.372	0.275	0.183	0.236	0.218	0.106	0.072	0.079	0.058	0.020
PSO	0.252	0.342	0.388	0.284	**0.198**	0.240	0.219	0.081	**0.080**	0.082	0.046	0.013
NES	**0.274**	**0.360**	0.381	0.282	**0.198**	0.234	0.213	0.113	0.071	0.076	0.055	0.016
SPSA	**0.274**	**0.360**	**0.412**	**0.427**	0.188	**0.251**	**0.287**	**0.298**	0.067	**0.086**	**0.091**	**0.095**

Table 3: Black-box OA on Fashion-MNIST dataset. In the $N=\infty$ experiments, ($\tau_{\mathcal{S}}$, mR) are reported in each cell, while only $\tau_{\mathcal{S}}$ is reported in the cells when N equals 50 or k. A larger k and a smaller N make the attack harder.

Dataset	k	Rand	Beta	PSO	NES	SPSA	SRC Time
Fashion-MNIST	5	0.195	0.386	0.208	0.208	0.202	0.080
Fashion-MNIST	10	0.206	0.404	0.223	0.214	0.213	0.087
Fashion-MNIST	25	0.228	0.435	0.249	0.236	0.235	0.108
SOP	5	1.903	2.638	1.949	1.882	1.783	0.091
SOP	10	1.923	2.720	1.961	1.954	1.836	0.095
SOP	25	1.936	2.745	1.985	1.975	1.873	0.117

Table 4: Run time (second) per $(k, 50)$-OA adversarial example for different black-box methods. $\varepsilon=\frac{4}{255}$ and $Q=10^3$.

Since all the queries of Rand are independent, one intuitive way towards better performance is to leverage the historical query results and adjust the search distribution. Following this, we modify $\mathcal{N}$-Attack [31] into Beta-Attack, replacing its Gaussian distributions into Beta distributions, from which each element of $\boldsymbol{r}$ is independently generated. All the Beta distribution parameters are initialized as 1, where Beta degenerates into Uniform distribution (Rand). During optimization, the probability density functions are modified according to $\tau_{\mathcal{S}}$ results, in order to increase the expectation of the next adversarial perturbation drawn from these distributions. Results in Tab. 3 suggest Beta's advantage against Rand, but it also shows that the Beta distributions are insufficient for modeling the perturbation $\boldsymbol{r}$ for OA.

According to the (k, ∞)-OA results in Tab. 3 and Tab. 5, NES and SPSA outperform Rand, Beta and PSO. This means black-box optimizers based on estimated gradient are still the most effective and efficient for (k, ∞)-OA. A larger ε leads to a unanimous increase in $\tau_{\mathcal{S}}$ metric, and a side effect of worse (larger) mR. Predictably, when N=50 or k, the algorithms may confront a great penalty due to the absence of the selected candidates from the top-N visible range.

Further results of $(k, 50)$-OA and (k, k)-OA confirm our speculation, as shown in the middle (N=50) and bottom (N=k) parts of Tab. 3 and Tab. 5. With N=50 and a fixed ε, algorithms that result in a small mR (especially for those with mR$<\frac{50}{2}$) also perform comparably as in (k, ∞)-OA. Conversely, algorithms that lead to a large mR in (k, ∞)-OA are greatly penalized in $(k, 50)$-OA. The results also manifest a special characteristic of OA, that $\tau_{\mathcal{S}}$ peaks at a certain small ε, and does not positively correlate with ε. This is rather apparent in difficult settings such as (k, k)-OA on SOP dataset. In brief, the optimizers based on estimated gradients still perform the best in $(k, 50)$-OA and (k, k)-OA, and an excessively large perturbation budget is not necessary.

All these experiments demonstrate the effectiveness of optimizing the surrogate objective $\tau_{\mathcal{S}}$ to conduct black-box OA. As far as we know, optimizers based on gradient estimation are the most reliable choices. Next, we adopt SPSA and perform practical OA in real-world applications.

Time Complexity. Although the complexity of Algo. 1 is $\mathcal{O}(k^2)$, the actual run time of our Rust[4]-based SRC implementation is short, as measured with Python cProfile with a Xeon 5118 CPU and a GTX1080Ti GPU. As shown in Tab. 4, SRC calculation merely consumes 0.117 seconds on average across the five algorithms for an adversarial example on SOP with $k=25$. The overall time consumption is dominated by sorting and PyTorch [37] model inference. Predictably, in real-world attack scenarios, It is highly likely that the time consumption bottleneck stems from other factors irrelevant to our method, such as network delays and bandwidth, or the query-per-second limit set by the service provider.

[4]Rust Programming language. See https://www.rust-lang.org/.

Algorithm	$k=5$				$k=10$				$k=25$			
	$\varepsilon=\frac{2}{255}$	$\varepsilon=\frac{4}{255}$	$\varepsilon=\frac{8}{255}$	$\varepsilon=\frac{16}{255}$	$\varepsilon=\frac{2}{255}$	$\varepsilon=\frac{4}{255}$	$\varepsilon=\frac{8}{255}$	$\varepsilon=\frac{16}{255}$	$\varepsilon=\frac{2}{255}$	$\varepsilon=\frac{4}{255}$	$\varepsilon=\frac{8}{255}$	$\varepsilon=\frac{16}{255}$
None	0.0, 2.0	0.0, 2.0	0.0, 2.0	0.0, 2.0	0.0, 4.5	0.0, 4.5	0.0, 4.5	0.0, 4.5	0.0, 12.0	0.0, 12.0	0.0, 12.0	0.0, 12.0
Stanford Online Product $N=\infty$												
Rand	0.187, 2.6	0.229, 8.5	0.253, 85.8	0.291, 649.7	0.167, 5.6	0.197, 13.2	0.208, 92.6	0.222, 716.4	0.093, 14.1	0.110, 27.6	0.125, 146.7	0.134, 903.7
Beta	0.192, 3.3	0.239, 15.3	0.265, 176.7	0.300, 1257.7	0.158, 6.2	0.186, 19.9	0.207, 139.0	0.219, 992.5	0.099, 15.5	0.119, 37.1	0.119, 206.5	0.132, 1208.5
PSO	0.122, 2.1	0.170, 3.0	0.208, 13.3	0.259, 121.4	0.135, 4.8	0.177, 6.5	0.206, 22.8	0.222, 166.5	0.104, 12.7	0.122, 16.7	0.137, 49.5	0.140, 264.2
NES	**0.254, 3.4**	0.283, 15.6	**0.325, 163.0**	**0.368, 1278.7**	**0.312, 7.2**	**0.351, 26.3**	0.339, 227.1	0.332, 1486.7	**0.242, 18.0**	**0.259, 51.5**	0.250, 324.1	0.225, 1790.8
SPSA	0.237, 3.5	**0.284, 11.9**	0.293, 75.2	0.318, 245.1	0.241, 7.8	0.325, 22.2	**0.362, 112.7**	**0.383, 389.0**	0.155, 18.1	0.229, 41.9	**0.286, 185.6**	**0.306, 557.8**
Stanford Online Product $N=50$												
Rand	0.180	0.216	0.190	0.126	0.163	0.166	0.119	0.055	0.092	0.055	0.016	0.003
Beta	0.181	0.233	0.204	0.119	0.153	0.168	0.116	0.054	0.084	0.057	0.021	0.003
PSO	0.122	0.173	0.183	0.153	0.135	0.164	0.137	0.081	0.093	0.083	0.042	0.011
NES	**0.247**	0.283	0.246	0.152	**0.314**	0.295	0.195	0.077	**0.211**	**0.136**	0.054	0.013
SPSA	0.241	**0.287**	**0.297**	**0.303**	0.233	**0.298**	**0.298**	**0.292**	0.125	0.130	**0.114**	**0.103**
Stanford Online Product $N=k$												
Rand	0.148	0.100	0.087	0.026	0.094	0.044	0.018	0.001	0.023	0.009	0.002	0.001
Beta	0.136	0.106	0.053	0.025	0.076	0.040	0.010	0.004	0.021	0.004	0.001	0.001
PSO	0.102	0.098	0.059	0.031	0.088	0.049	0.022	0.007	0.040	0.015	0.006	0.001
NES	**0.185**	0.139	0.076	0.030	**0.173**	0.097	0.036	0.008	**0.071**	**0.027**	0.007	0.005
SPSA	0.172	**0.154**	**0.141**	**0.144**	0.107	**0.104**	**0.085**	**0.069**	0.026	0.025	**0.017**	**0.016**

Table 5: Black-box OA on Stanford Online Product dataset. In the $N=\infty$ experiments, (τ_S, mR) are reported in each cell, while only τ_S is reported in the cells when N equals 50 or k. A larger k and a smaller N make the attack harder.

Algorithm	ε	k	Q	T	Mean τ_S	Stdev τ_S	Max τ_S	Min τ_S	Median τ_S
SPSA	1/255	5	100	204	0.390	0.373	1.000	-0.600	0.400
SPSA	1/255	10	100	200	0.187	0.245	0.822	-0.511	0.200
SPSA	1/255	25	100	153	0.039	0.137	0.346	-0.346	0.033

Table 6: Quantitative $(k, 50)$-OA Results on JD Snapshop.

Algorithm	ε	k	Q	T	Mean τ_S	Stdev τ_S	Max τ_S	Min τ_S	Median τ_S
SPSA	8/255	5	100	105	0.452	0.379	1.000	-0.400	0.600
SPSA	8/255	10	100	95	0.152	0.217	0.733	-0.378	0.156
SPSA	8/255	25	100	93	0.001	0.141	0.360	-0.406	0.010

Table 7: $(k, 50)$-OA Results on Bing Visual Search API.

4.3. Practical Black-Box Order Attack Experiments

"JD SnapShop" [23] is an e-commerce platform based on content-based image retrieval [44]. Clients can upload query merchandise images via an `HTTP`-protocol-based API, and then obtain the top-50 similar products. This exactly matches the setting of $(k, 50)$-OA. As the API specifies a file size limit ($\leqslant$ `3.75MB`), and a minimum image resolution (128×128), we use the standard (224×224) size for the query. We merely provide limited evaluations because the API poses a hard limit of 500 queries per day per user.

As noted in Sec. 4.2, the τ_S peaks with a certain small ε. Likewise, an empirical search for ε suggests that $1/255$ and $2/255$ are the most suitable choices for OA against "JD SnapShop". Any larger ε value easily leads to the disappearance of $\mathbb{C}$ from $\mathbb{X}$. This is meanwhile a preferable characteristic, as smaller perturbations are less perceptible.

As shown in Fig. 1, we select the top-5 candidates as $\mathbb{C}$, and specify $\mathbf{p}=[1, 5, 4, 3, 2]$. Namely, the expected *relative order* among $\mathbb{C}$ is $\boldsymbol{c}_1 \prec \boldsymbol{c}_5 \prec \boldsymbol{c}_4 \prec \boldsymbol{c}_3 \prec \boldsymbol{c}_2$. By maximizing the τ_S with SPSA and $\varepsilon=\frac{1}{255}$, we successfully convert the *relative order* to the specified one using 200 times of queries. This shows that performing OA by optimizing our proposed surrogate loss τ_S with a black-box optimizer is *practical*. Some limited quantitative results are presented in Tab. 6, where Q is further limited to 100 in order to gather more data, while a random permutation and a random query image from SOP is used for each of the T times of attacks.

We also conduct OA against Bing Visual Search API [34] following the same protocol. We find this API less sensitive to weak (*i.e.*, $\varepsilon=\frac{1}{255}$) perturbations unlike Snapshop, possibly due to a different data pre-processing pipeline. As shown in Tab. 7, our OA is also effective against this API.

In practice, the adversarial example may be slightly changed by transformations such as the *lossy* JPEG/PNG compression, and resizing (as a pre-processing step), which eventually leads to changes on the τ_S surface. But a black-box optimizer should be robust to such influences.

5. Conclusion

Deep ranking systems inherited the adversarial vulnerabilities of deep neural networks. In this paper, the *Order Attack* is proposed to tamper the *relative order* among selected candidates. Multiple experimental evaluations of the white-box and black-box *Order Attack* illustrate their effectiveness, as well as the deep ranking systems' vulnerability in practice. Ranking robustness and fairness with respect to *Order Attack* may be the next valuable direction to explore. Code: `https://github.com/cdluminate/advorder`.

Acknowledgement. This work was supported partly by National Key R&D Program of China Grant 2018AAA0101400, NSFC Grants 62088102, 61976171, and 61773312, and Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, pages 484–501, 2020. 2

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pages 274–283, 2018. 2

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*, pages 284–293, 2018. 2

[4] Song Bai, Yingwei Li, Yuyin Zhou, Qizhu Li, and Philip HS Torr. Metric attack and defense for person re-identification. *arXiv preprint arXiv:1901.10650*, 2019. 3

[5] Quentin Bouniot, Romaric Audigier, and Angelique Loesch. Vulnerability of person re-identification models to metric adversarial attacks. In *CVPR workshop*, 2020. 3

[6] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *ICLR*, 2018. 2

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. 2

[8] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. HopSkipJumpAttack: a query-efficient decision-based adversarial attack. In *IEEE Symposium on Security and Privacy (SP)*, 2020. 2

[9] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017. 2, 5

[10] Ye Chen and Tak W Yan. Position-normalized click prediction in search advertising. In *SIGKDD*, pages 795–803, 2012. 1

[11] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *ICLR*, 2019. 2

[12] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 2

[13] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, pages 2206–2216, 2020. 2

[14] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In *CVPR*, 2020. 2, 5

[15] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 2

[16] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, pages 4312–4321, 2019. 2

[17] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. *CVPR*, pages 7706–7714, 2019. 2

[18] Yan Feng, Bin Chen, Tao Dai, and Shu-Tao Xia. Adversarial attack on deep product quantization network for image retrieval. In *AAAI*, pages 10786–10793, 2020. 3

[19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015. 2

[20] Gregory Goren, Oren Kurland, Moshe Tennenholtz, and Fiana Raiber. Ranking robustness under adversarial document manipulations. In *ACM SIGIR*, pages 395–404, 2018. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[22] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *ICML*, pages 2137–2146, 2018. 2, 4, 5, 6

[23] JingDong. Snapshop api https://neuhub.jd.com/ai/api/image/snapshop. 2, 5, 8

[24] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, pages 239–251, 1945. 2, 4

[25] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *CVPR*, pages 2288–2297, 2019. 2, 3

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 1

[27] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR workshop*, 2017. 2, 5

[28] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *CVPR*, pages 1221–1230, 2020. 2, 5

[29] Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. Universal perturbation attack against image retrieval. In *ICCV*, pages 4899–4908, 2019. 1, 3

[30] Xiaodan Li, Jinfeng Li, Yuefeng Chen, Shaokai Ye, Yuan He, Shuhui Wang, Hang Su, and Hui Xue. Qair: Practical query-efficient black-box attacks for image retrieval. *CVPR*, 2021. 3

[31] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *ICML*, pages 3866–3876, 2019. 2, 6, 7

[32] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. Who's afraid of adversarial queries?: The impact of image modifications on content-based image retrieval. In *ICMR*, pages 306–314, 2019. 1, 3

[33] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018. 2, 3, 6

[34] Microsoft. Bing visual search api https://docs.microsoft.com/en-us/bing/search-apis/. 8

[35] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, pages 2574–2582, 2016. 2

[36] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016. 5

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. *None*, 2017. 7

[38] Furcy Pin and Peter Key. Stochastic variability in sponsored search auctions: observations and models. In *the 12th ACM conference on Electronic commerce*, pages 61–70, 2011. 1

[39] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? *Neural computation*, pages 1063–1076, 2004. 3

[40] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *ICML*, pages 8242–8252, 2020. 3

[41] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 1, 2, 3

[42] Yuhui Shi and Russell Eberhart. A modified particle swarm optimizer. In *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, pages 69–73, 1998. 6

[43] Satya Narayan Shukla, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Hard label black-box adversarial attacks in low query budget regimes. *arXiv preprint arXiv:2007.07210*, 2020. 2, 5

[44] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE TPAMI*, pages 1349–1380, 2000. 1, 8

[45] James C Spall et al. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, pages 332–341, 1992. 6

[46] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICLR*, 2014. 1, 2

[47] Giorgos Tolias, Filip Radenovic, and Ondrej Chum. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In *ICCV*, pages 5037–5046, 2019. 3

[48] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *NeurIPS*, pages 1633–1645, 2020. 2

[49] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *ICML*, pages 5025–5034, 2018. 2, 5, 6

[50] Hongjun Wang, Guangrun Wang, Ya Li, Dongyu Zhang, and Liang Lin. Transferable, controllable, and inconspicuous adversarial attacks on person re-identification with deep misranking. In *CVPR*, 2020. 3

[51] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393, 2014. 1, 2, 3

[52] Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. Natural evolution strategies. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3381–3387, 2008. 6

[53] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[54] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, pages 2730–2739, 2019. 2

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

[56] Guoping Zhao, Mingyu Zhang, Jiajun Liu, and Ji-Rong Wen. Unsupervised adversarial attacks on deep feature-based retrieval with gan. *arXiv preprint arXiv:1907.05793*, 2019. 3

[57] Zhedong Zheng, Liang Zheng, Zhilan Hu, and Yi Yang. Open set adversarial examples. *arXiv preprint arXiv:1809.02681*, 2018. 3

[58] Mo Zhou, Zhenxing Niu, Le Wang, Zhanning Gao, Qilin Zhang, and Gang Hua. Ladder loss for coherent visual-semantic embedding. In *AAAI*, pages 13050–13057, 2020. 2, 3

[59] Mo Zhou, Zhenxing Niu, Le Wang, Qilin Zhang, and Gang Hua. Adversarial ranking attack and defense. In *ECCV*, pages 781–799, 2020. 1, 2, 3, 5