ECCV'20 ONLINE

23-28 AUGUST 2020

16TH EUROPEAN CONFERENCE ON

COMPUTER VISION

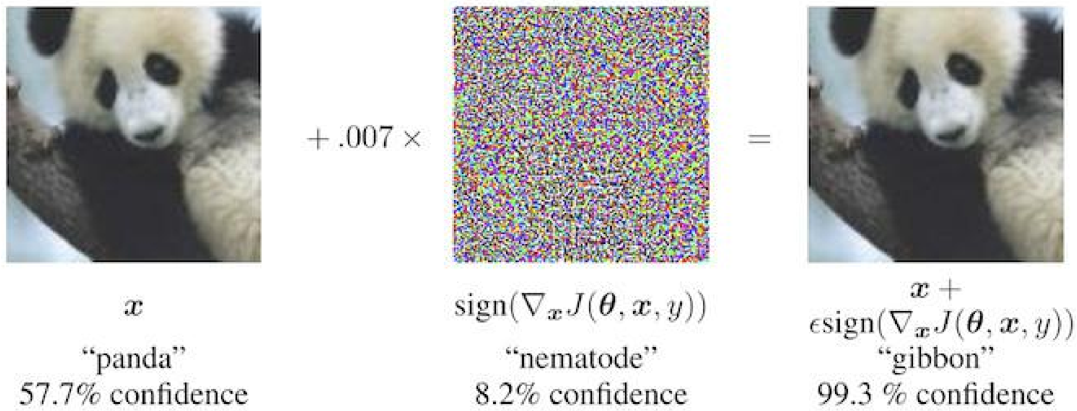
WWW.ECCV2020.EU

Adversarial Ranking Attack and Defense

M. Zhou, Z. Niu, L. Wang, Q. Zhang, G. Hua

https://arxiv.org/abs/2002.11293 Github:

Adversarial Example



Deep Neural Network (DNN) classifiers are vulnerable to adversarial attack, where an imperceptible perturbation could result in misclassification.

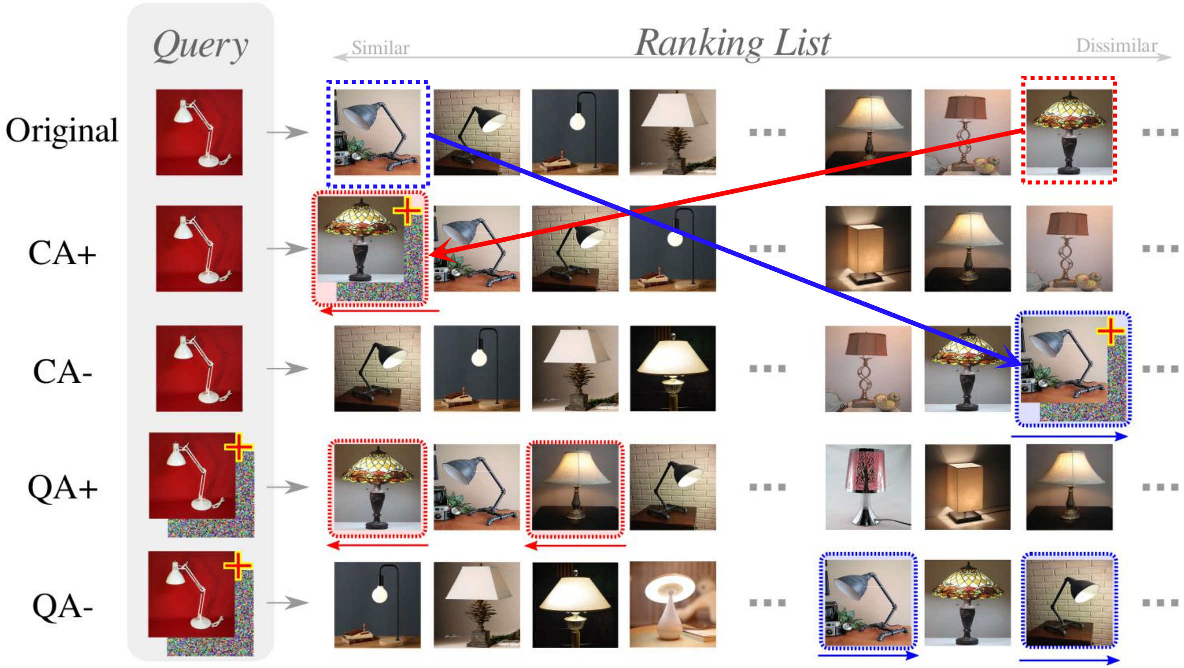
However, the vulnerability of DNN-based image ranking systems remains **under-explored**.

Adversarial Ranking Attack

Definition: Raise or lower the rank of chosen candidates with respect to a specific query set

Candidate Attack (CA): Raise (CA+) or lower (CA-) the rank by perturbing candidates.

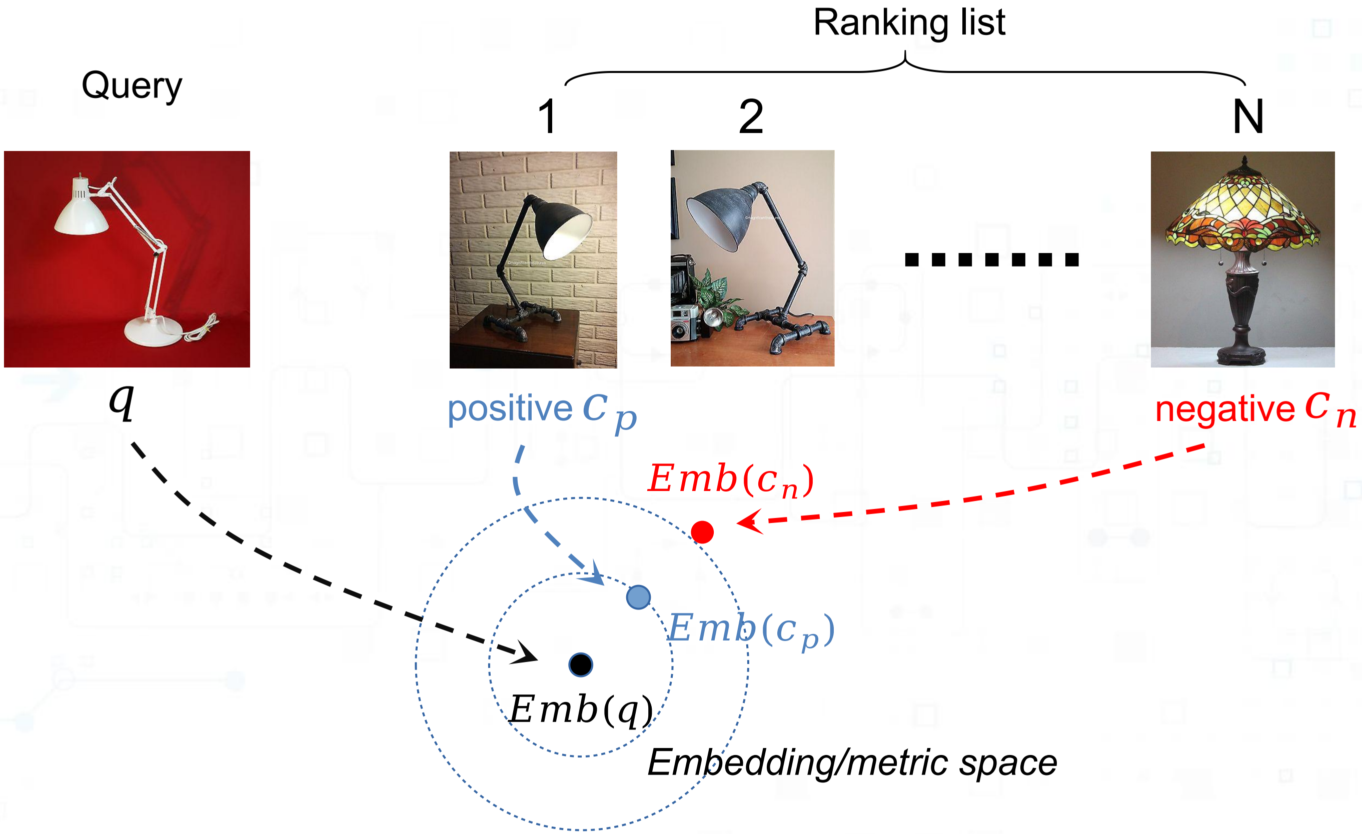
Query Attack (QA): Raise (QA+) or lower (QA-) the rank by perturbing queries.



Case1: a malicious seller may attempt to raise the rank of his own product (CA+), or lower the rank of his competitor's product (CA-);

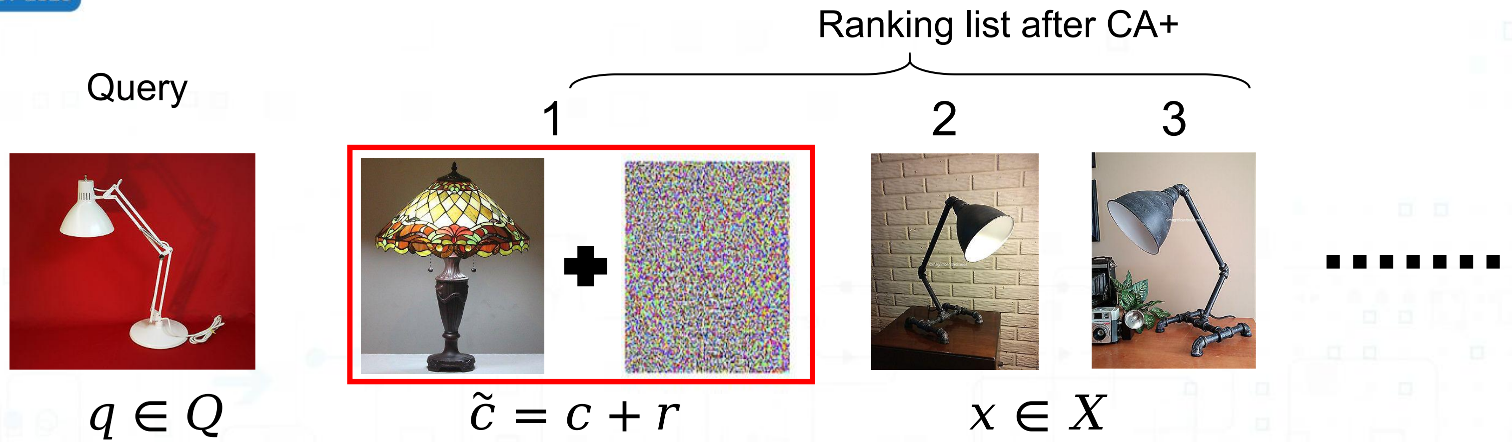
Case2: a "man-in-the-middle" attacker could hijack the query image in order to promote (QA+) or impede (QA-) the sales of specific products.

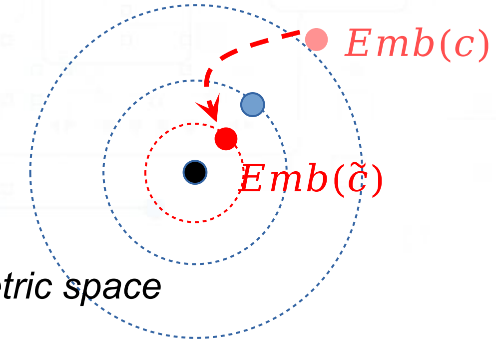
Recall: Metric learning for ranking



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

Metric learning for CA+





Embedding/metric space

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

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

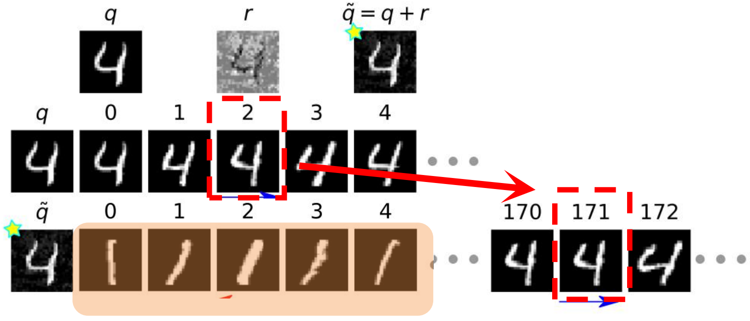
Optimize with Projected Gradient Descent (PGD)

QA- with semantic preserving

$$L_{\text{QA-}}(q, C; X) = \sum_{c \in C} \sum_{x \in X} \left[-d(q,c) + d(q,x) \right]_+,$$
$$r = \arg\min_{r \in \Gamma} L_{\text{QA-}}(q + r, C; X),$$

However

Semantics is affected by the perturbation.

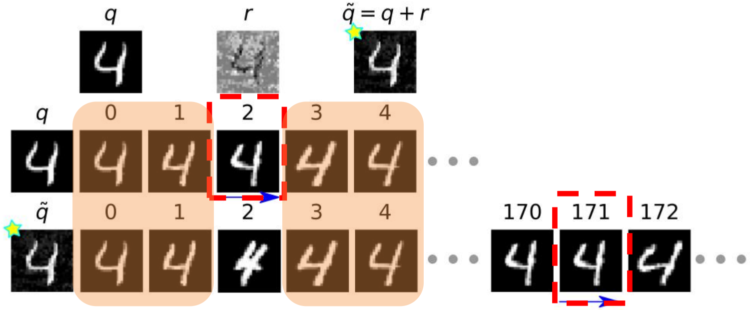


Abnormal

Naive QA- could lead to **abnormal** ranking results.

Retain all candidates except for the targeted one

$$L_{\text{SP-QA-}}(q, C; X) = L_{\text{QA-}}(q, C; X) + \xi L_{\text{QA+}}(q, C_{\text{SP}}; X),$$
$$C_{\text{SP}} = \{ c \in X \setminus C \,|\, \text{Rank}_{X \setminus C}(q, c) \leqslant G \},$$



Semantics-Preserving QA-

Adversarial Defense

* How to make our ranking system robust ?

Adversarial defense for classification task

- Many specific methods w.r.t different attackers
- Adversarial training is the most effective method:
  - generate adversarial examples first, and training a new model with all original training examples as well as adversarial ones

Cannot be directly used to defense against AdvRank

- diverging training loss
- needs to defend against distinct attacks individually

Adversarial Ranking Defense

- Underlying principle of AdvRank is to shift the embeddings of candidates/queries to a proper place, and a successful attack depends on *a large shift distance as well as a correct shift direction*

- A large shift distance is an indispensable objective for all CA+, CA-, QA+, and QA- attacks

- Our approach
  - propose a “maximum-shift-distance” attack to generate adversarial examples, and then conduct adversarial training procedure

Experiments (Attack + Defense)

For a random candidate, its average rank is at middle (i.e., 50%) of ranking list

After CA+, with perturbation 0.3, its average rank is raised to the top (i.e., 2.1%) of ranking list

With a robust system (with defense), the CA+ can only raise its average rank to the 30.7% of ranking list

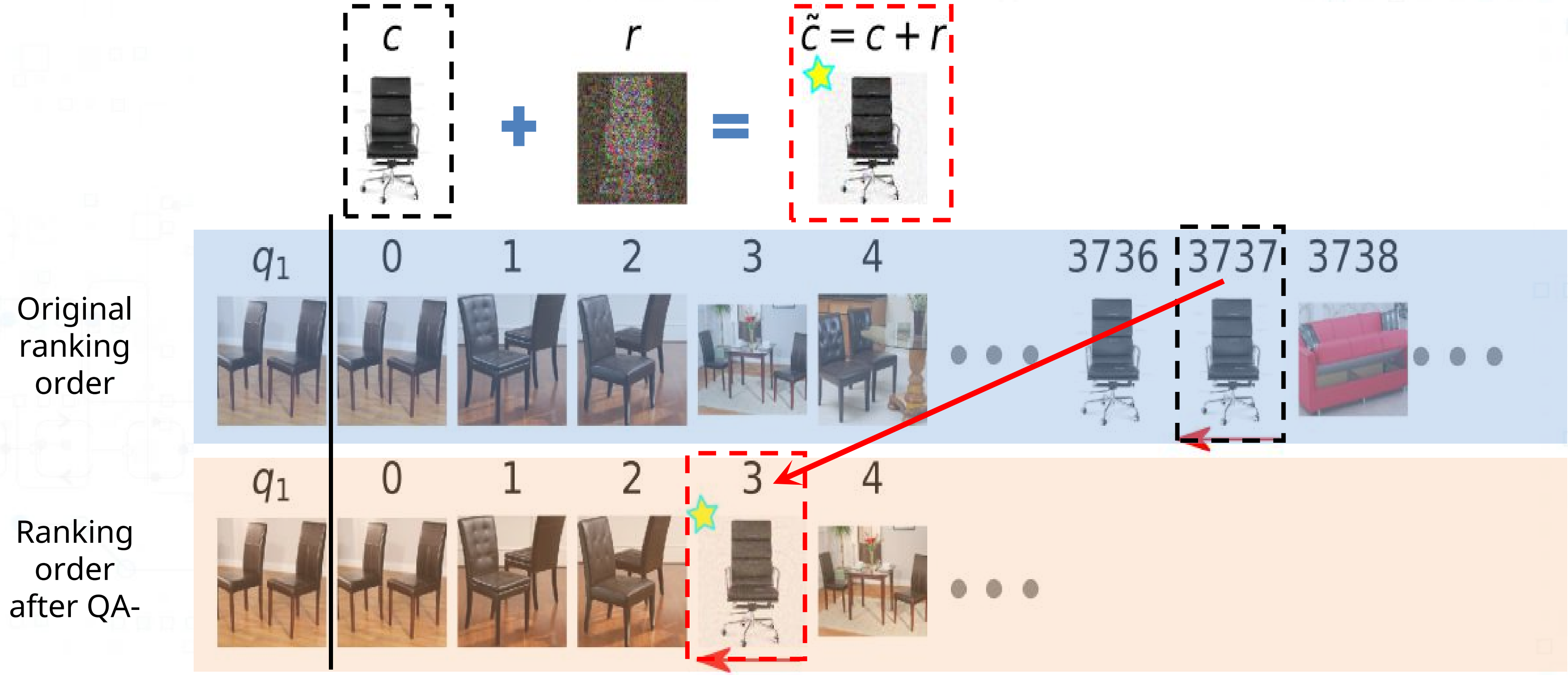
Attack on MNIST

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

Defense on MNIST

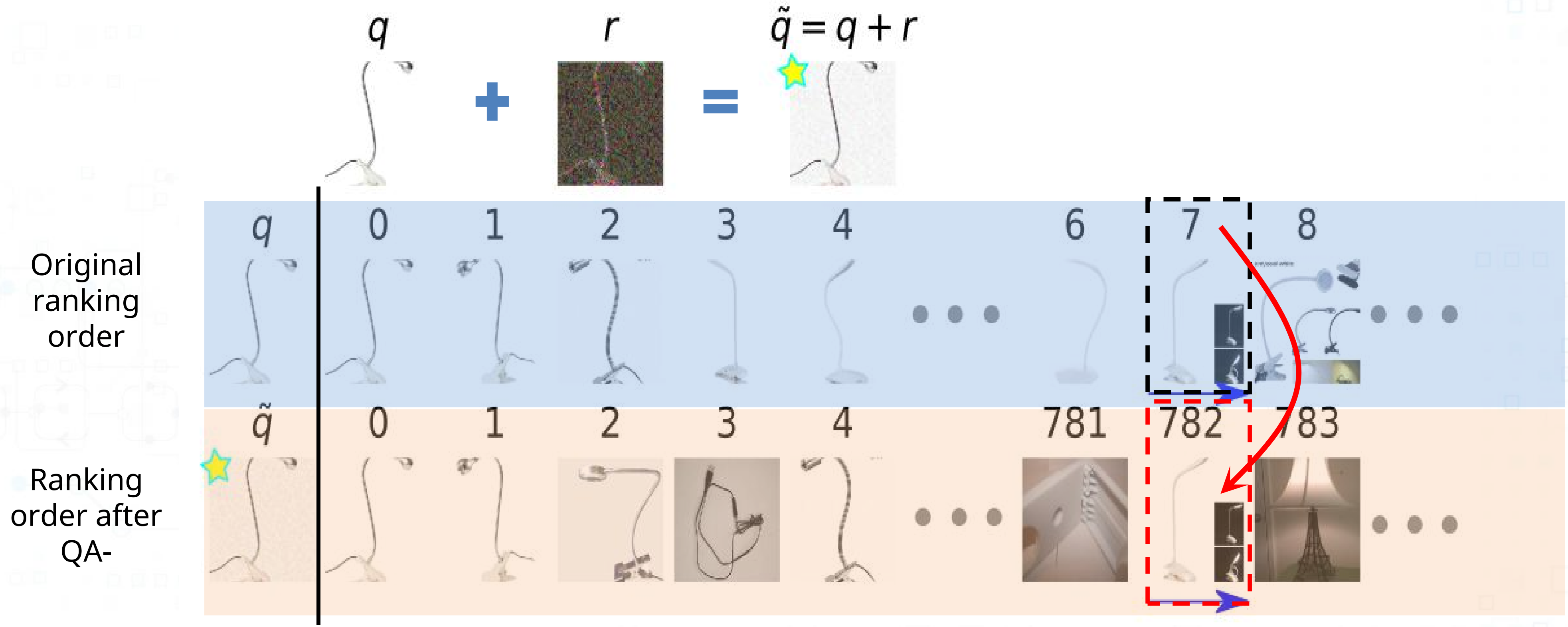
ε	CA+				CA-				QA+				QA-			
	$w=1$	2	5	10	$w=1$	2	5	10	$m=1$	2	5	10	$m=1$	2	5	10
(CTD) Cosine Distance, Triplet Loss, Defensive (R@1=98.3%)																
0	50	50	50	50	2.0	2.0	2.0	2.0	50	50	50	50	0.5	0.5	0.5	0.5
0.01	48.9	49.3	49.4	49.5	2.2	2.2	2.2	2.1	49.9	49.5	49.5	49.7	0.5	0.5	0.5	0.5
0.03	47.4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

CA+ Attack

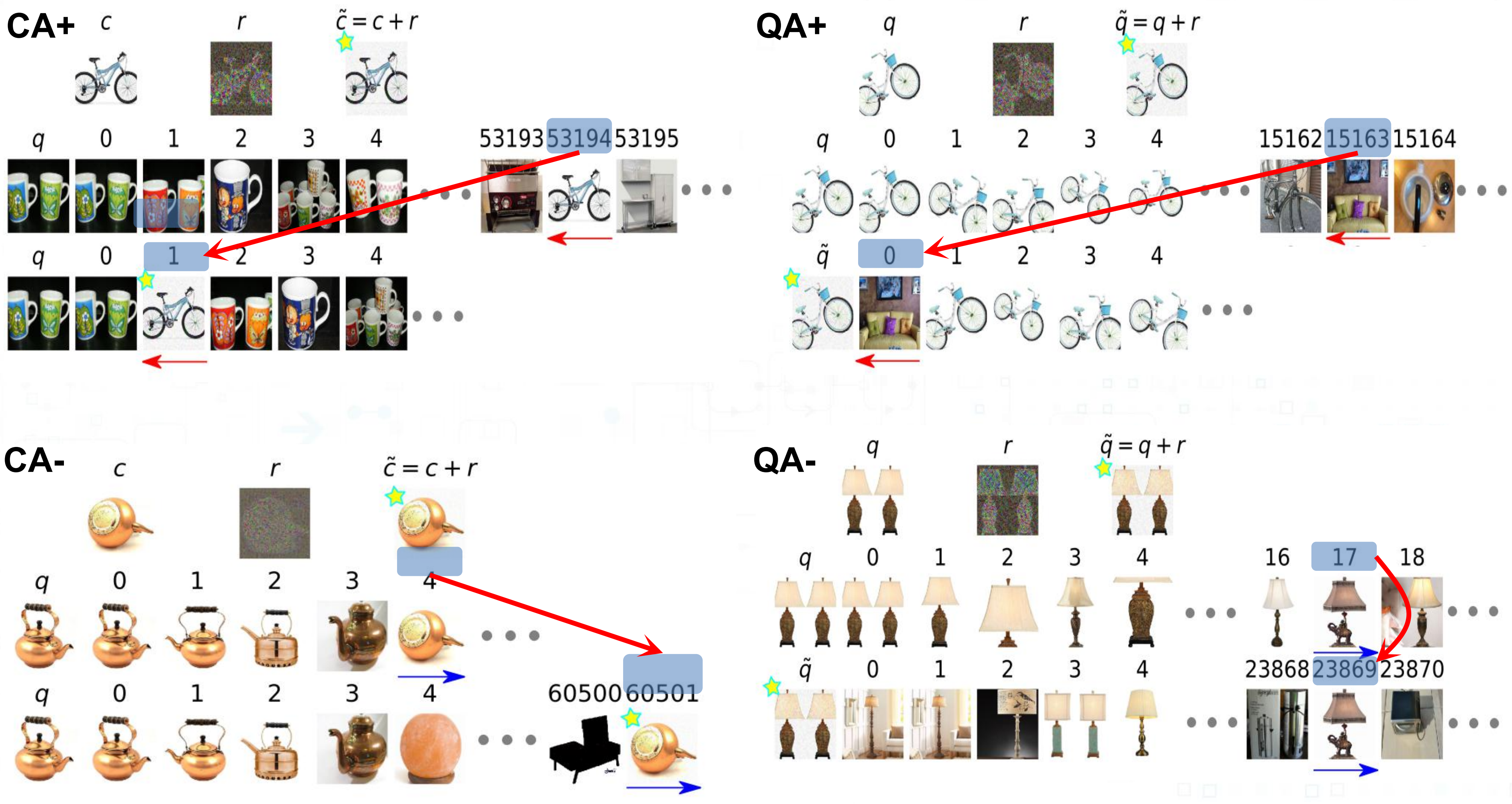


c
r
$\tilde{c} = c + r$
q_1
0
1
2
3
4
3736
3737
3738
Original ranking order
q_1
0
1
2
3
4
Ranking order after QA-

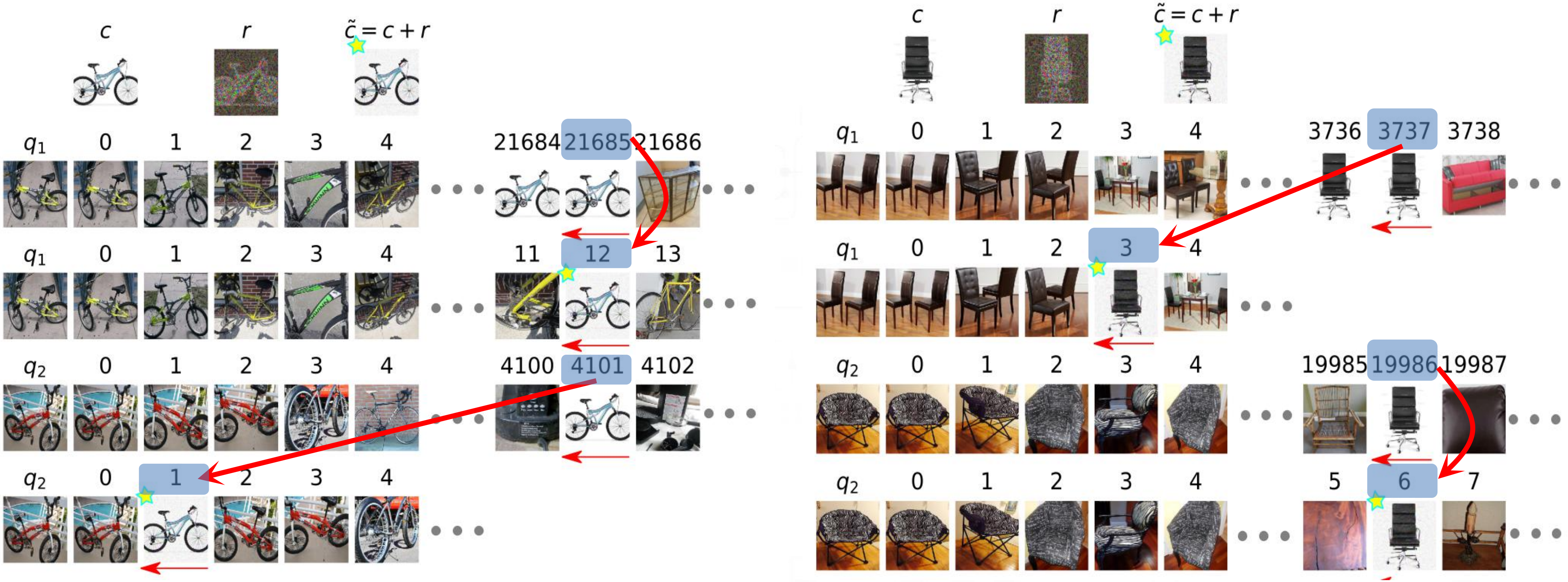
QA- Attack



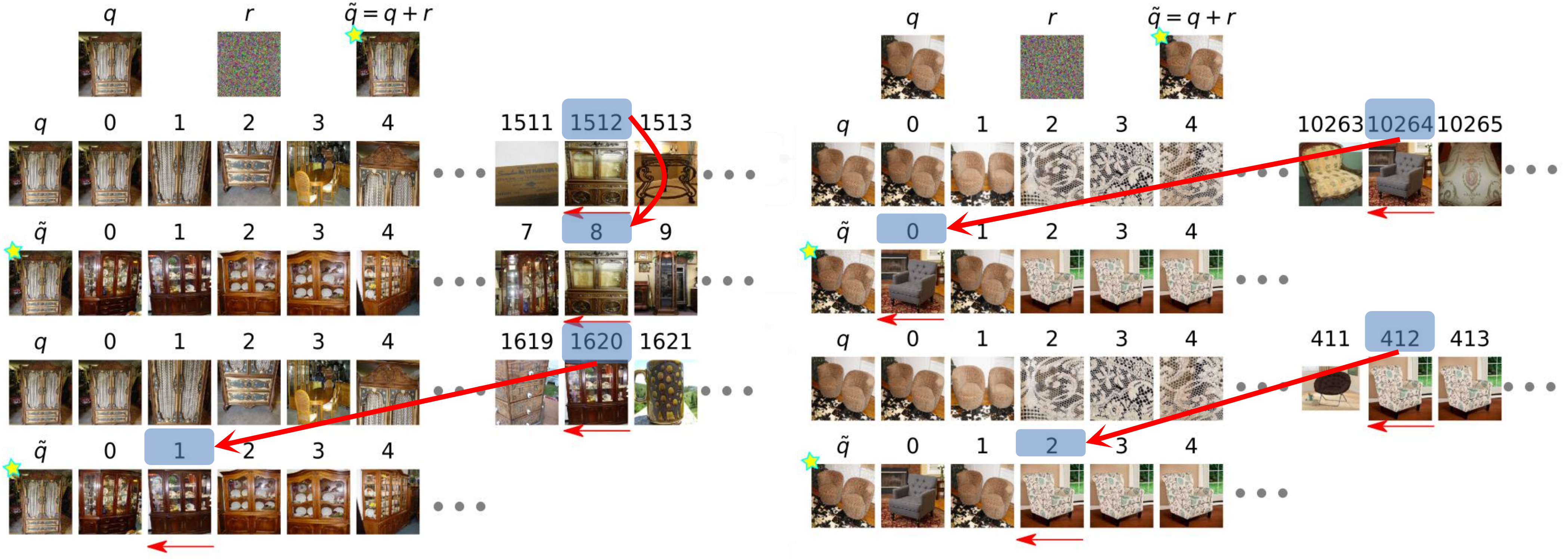
q
r
$\tilde{q} = q + r$
Original ranking order
q
0
1
2
3
4
6
7
8
Ranking order after QA-
$\tilde{q}$
0
1
2
3
4
781
782
783


CA+
c
r
c̃ = c + r
q 0 1 2 3 4
53193 53194 53195
q 0 1 2 3 4
QA+
q
r
q̃ = q + r
q 0 1 2 3 4
15162 15163 15164
q̃ 0 1 2 3 4
CA-
c
r
c̃ = c + r
q 0 1 2 3 4
q 0 1 2 3 4
60500 60501
QA-
q
r
q̃ = q + r
q 0 1 2 3 4
16 17 18
q̃ 0 1 2 3 4
23868 23869 23870

CA+ for a Query Set



QA+ for a Candidate Set



ECCV'20 ONLINE 23-28 AUGUST 2020

Thanks!

Arxiv: (2002.11293)

Github: