Collapse-Aware Triplet Decoupling for Adversarially Robust Image Retrieval

Qiwei Tian[1] Chenhao Lin[1] Zhengyu Zhao[1] Qian Li[1] Chao Shen[1]

Abstract

Adversarial training has achieved substantial performance in defending image retrieval against adversarial examples. However, existing studies in deep metric learning (DML) still suffer from two major limitations: *weak adversary* and *model collapse*. In this paper, we address these two limitations by proposing **C**ollapse-**A**ware **TRI**plet **DE**coupling (CA-TRIDE). Specifically, TRIDE yields a stronger adversary by spatially decoupling the perturbation targets into the anchor and the other candidates. Furthermore, CA prevents the consequential model collapse, based on a novel metric, collapseness, which is incorporated into the optimization of perturbation. We also identify two drawbacks of the existing robustness metric in image retrieval and propose a new metric for a more reasonable robustness evaluation. Extensive experiments on three datasets demonstrate that CA-TRIDE outperforms existing defense methods in both conventional and new metrics. *Codes are available at* https://github.com/michaeltian108/CA-TRIDE.

1. Introduction

Thanks to the massive image data and the development of Deep Neural Networks (DNNs), image retrieval has experienced substantial advancements. Despite their effectiveness, deep image retrieval is known to be vulnerable to adversarial examples, i.e. test samples that cause erroneous model behaviour (Szegedy et al., 2013). Existing work on adversarial attacks against DNN-based image retrieval has explored deep metric learning (DML) models (Liu et al., 2019; Li et al., 2021; Chen et al., 2021; Zhou et al., 2021a) and deep hashing models (Bai et al., 2020; Lu et al., 2021).

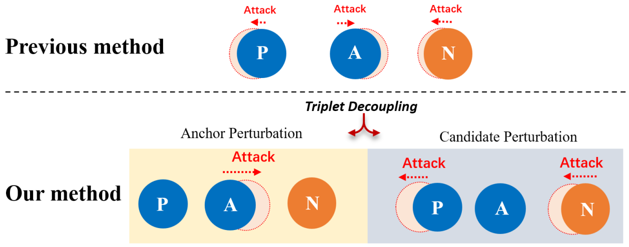


Figure 1. Our defense vs. the previous defense. Specifically, we address the *weak adversary* by decoupling the updates of perturbation into anchors (A) and positive (P)/negative (N) candidates to maximize embedding shifts.

To defend DML models against adversarial examples, recent studies (Zhou et al., 2021b; Zhou & Patel, 2022) have modified the commonly-used adversarial training (Madry et al., 2017) from the domain of image classification into the domain of image retrieval. However, these methods still suffer from two major limitations:

- **Weak adversary** is a widely overlooked issue since existing defenses (Zhou & Patel, 2022; Zhou et al., 2021b) directly adopt the perturbation method from the image classification domain without fully exploiting the triplet structure, which consists of three components: positives, negatives, and anchors. As shown in Figure 1, the existing method perturbs all three components simultaneously and allocates the desired embedding shifts among each component, as represented by the small red shadows, which leads to smaller average embedding shifts. A detailed qualitative analysis is presented in Section A of the Appendix.
- **Model Collapse** is a notorious challenge in DML, impeding researchers from training models with hard samples (Xuan et al., 2020). A collapsed model embeds all samples disastrously close and cannot retrieve semantically similar examples appropriately. Model collapse becomes inevitable in adversarial training as it aims at increasing sample hardness. The current state-of-the-art adversarial training method, Hardness Manipulation (Zhou & Patel, 2022), avoids model collapse by restricting perturbation strength, consequently lowering its gained robustness.

In this paper, we address the above limitations by propos-

[1] Xi'an JiaoTong University, Xi'an, China. Correspondence to: Chenhao Lin <linchenhao@xjtu.edu.cn>, Chao Shen <chaoshen@mail.xjtu.edu.cn>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

ing a novel adversarial training approach to robust image retrieval, called **C**ollapseness-**A**ware **TRI**plet **DE**coupling (**CA-TRIDE**). A comparison between our CA-TRIDE and the previous method is illustrated in Figure 1. Generally, our CA-TRIDE can be applied to any triplet-based DML in a plug-and-play manner, without making major modifications.

First, our TRIDE aims to address the *weak adversary* by decoupling the perturbation updates on the triplets (anchors, positive candidates, and negative candidates) into two phases: anchor perturbation (ANP) and candidate perturbation (CAP). Specifically, in the ANP phase, only the anchor is perturbed, while in the CAP phase, the positive and negative candidates are perturbed. By alternating between CAP and ANP, TRIDE incurs larger embedding shifts and yields a stronger adversary, as represented by the red shadows in Figure 1. Then, to tackle model collapse caused by our stronger adversary, CA is proposed to use a novel metric 'collapseness' (denoted as $\mathcal{C}$) to capture intermediate model states, preventing impending model collapse by orienting the optimization of TRIDE through $\mathcal{C}$.

In sum, our paper makes the following three contributions:

- We propose CA-TRIDE, a novel approach to adversarial robust image retrieval based on collapse-aware (CA) and triplet decoupling (TRIDE). In particular, a new metric called 'collapseness' is proposed to capture the intermediate model states w.r.t. model collapse.
- We validate that CA-TRIDE addresses the two limitations in existing defense: *weak adversary* and *model collapse*, showing that TRIDE yields noticeably larger embedding shifts, more shrinkage in embedding space and lower separability, without causing model collapse.
- We identify two drawbacks of the commonly used robustness evaluation metric, ERS (Zhou et al., 2021b), and propose a new metric, ARS. Experimental results on three datasets show that CA-TRIDE outperforms existing methods in both ERS and ARS by 4$\sim$5%.

2. Related Work

2.1. Image Retrieval and Deep Metric Learning

Image retrieval (Wang et al., 2014) has been thriving as a surging amount of visual content has become available to the public. Deep metric learning (DML) is one of the popular methods to realize such tasks. Two main focuses of current work in DML are loss functions and data sampling methods, both of which have a crucial impact on image retrieval performance (Musgrave et al., 2020). For loss functions, among other newly proposed losses, such as lifted structure loss (Oh Song et al., 2016), N-pair loss (Sohn, 2016) and Multi-Similarity loss (Wang et al., 2019), triplet loss (Schroff et al., 2015; Yu et al., 2018) has been popular due to its simplicity and performance. For sampling methods, current research focuses on improving the diversity of samples within a batch to enhance the overall performance. Popular sampling strategies include random sampling, semi-hard sampling (Schroff et al., 2015), soft-hard sampling (Roth et al., 2020), distance-weighted sampling (Wu et al., 2017), etc. In this paper, we adopt the commonly used triplet loss for training DML models.

2.2. Adversarial Attacks in Deep Metric Learning

Adversarial attacks (AAs) against image retrieval primarily seek to lower the R@1 of image retrieval (Liu et al., 2019; Tolias et al., 2019; Wang et al., 2020; Feng et al., 2020). These attacks, such as DAIR (Chen et al., 2021) and QAIR (Li et al., 2021), are mostly black-box and realized through repetitive queries to get a subset of the victim model's training set, which is subsequently used for optimizing AAs. Universal adversarial perturbations (UAPs) have also been explored for image retrieval, which works by utilizing rank-wise relationships to increase the transferability of AAs (Li et al., 2019). In addition, Zhou et al. (2021a) explores a ranking attack that compromises the ranking results, i.e. improves or decreases the rank of certain candidates.

2.3. Adversarial Defenses in Deep Metric Learning

Current DML defenses against AAs are relatively less explored. Similar to common adversarial defense practices in image classification (Zhong & Deng, 2019; Zhang et al., 2019; Picot et al., 2022; Pang et al., 2020; Zhu et al., 2021), existing defense methods in DML (Zhou et al., 2021b; Zhou & Patel, 2022) also rely on adversarial training with a triplet loss. Specifically, Zhou et al. (2021b) proposed an indirect perturbation method, anti-collapse triplet (ACT), which pulls positive and negative candidates in the triplet closer to make them harder to distinguish, which consequently causes limited robustness. In their later work (Zhou & Patel, 2022), a direct adversary called hardness manipulation (HM) (Zhou & Patel, 2022) was proposed and achieved a better result than ACT. However, as a certain number of adversaries were found to cause the model to collapse, the authors had to resort to weak adversaries to prevent such an issue. In this paper, we specifically address the model collapse and weak adversary problems in existing defenses with our proposed CA-TRIDE.

3. Methodology

3.1. Preliminaries

Triplet Loss. Given a triplet $\mathbb{T} = (\mathbf{A}, \mathbf{P}, \mathbf{N})$, the triplet loss $L_{\mathcal{T}}$ is defined as follows:

$$L_{\mathcal{T}}(\mathbb{T}) = d(\mathbf{A}, \mathbf{P}) - d(\mathbf{A}, \mathbf{N}) + \beta_{\mathcal{T}} \qquad (1)$$

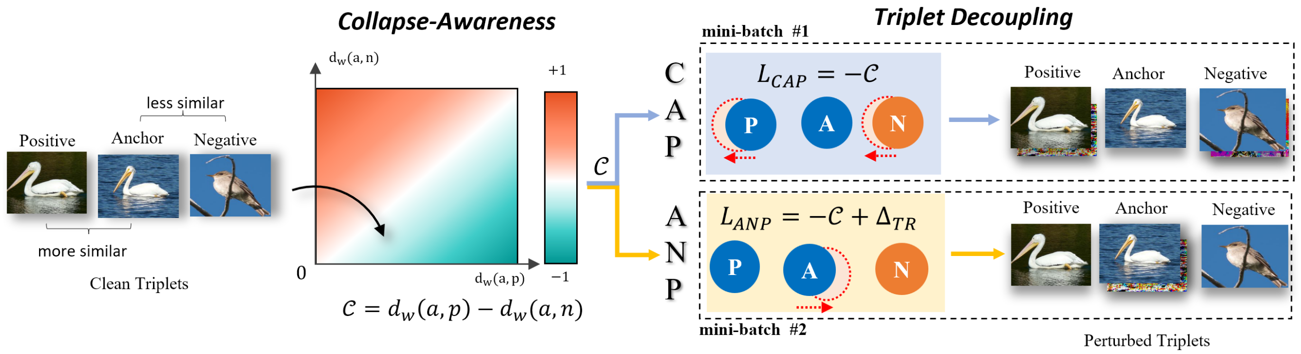


Figure 2. The working pipeline of our **CA-TRIDE** defense on a per mini-batch basis. (1) **Collapseness-Awareness (CA)** first calculates the proposed collapseness $\mathcal{C}$ on the clean triplets to capture model states and then incorporates $\mathcal{C}$ into subsequent perturbation optimization. (2) **Triplet Decoupling (TRIDE)** decouples the perturbation targets on the triplets into candidate perturbation (CAP) and anchor perturbation (ANP), which are alternatively implemented across mini-batches during adversarial training, starting with CAP.

where $\mathbf{A}$, $\mathbf{P}$, and $\mathbf{N}$ represent the anchors, positive, and negative examples, respectively. $d(\cdot,\cdot)$ calculates the average distance between two samples of the given triplet. $\beta_{\mathcal{T}}$ is the margin that defines how far the model needs to separate $\mathbf{P}$ from $\mathbf{N}$, usually set within $[0.2, 0.8]$.

$L_{\mathcal{T}}$ helps the model learn an embedding space that locates semantically similar examples as closely as possible. This is achieved by minimizing $d(\mathbf{A},\mathbf{P})$ and maximizing $d(\mathbf{A},\mathbf{N})$ until the margin $\beta_{\mathcal{T}}$ is met. Consequently, the goal of adversarial training is otherwise: adversarially maximizing $d(\mathbf{A},\mathbf{P})$ while minimizing $d(\mathbf{A},\mathbf{N})$.

Hardness. Proposed in the recent work (Zhou & Patel, 2022), the hardness of a triplet $\mathbb{T}$ is defined as follows:

$$H(\mathbf{A},\mathbf{P},\mathbf{N}) = d(\mathbf{A},\mathbf{P}) - d(\mathbf{A},\mathbf{N}), \quad H \in [-2,2] \qquad (2)$$

Hardness determines how difficult a triplet is for a DML model to distinguish. A negative H signifies easy triplets, while a positive H denotes hard triplets. The magnitude of H determines how hard or easy the triplet is.

Adversarial Training (AT) in DML. In contrast to the standard training, the goal of AT is to learn a robust model on the adversarial triplet $\tilde{\mathbb{T}}$, which is acquired by adding perturbations δ into the clean triplet $\tilde{\mathbb{T}} = (\tilde{\mathbf{A}}, \tilde{\mathbf{P}}, \tilde{\mathbf{N}}) = (\mathbf{A}+\delta, \mathbf{P}+\delta, \mathbf{N}+\delta)$.

Specifically, the perturbation δ is optimized by maximizing hardness H:

$$\arg\max_{\delta} H(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}, \tilde{\mathbf{N}}) \qquad (3)$$

where δ is bounded by an l_p norm ϵ for imperceptibility. The specific methods for maximizing H could vary across different methods. For example, the afore-mentioned HM uses $L_{HM} = ||H_D - \tilde{H}_O||_2^2$ to generate perturbation to increase orginal hardness from H_O to destination hardness H_D.

After the generation of perturbed triplets, these triplets are used for training an adversarially robust model Θ by optimizing:

$$\arg\min_{\theta} L_{\mathcal{T}}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}, \tilde{\mathbf{N}}; \Theta) \qquad (4)$$

Our CA-TRIDE is built on AT, as shown in the per mini-batch working pipeline in Figure 2. Overall, given a mini-batch of triplets, CA first evaluates model states on clean triplets using collapseness $\mathcal{C}$. The calculated collapseness is then fed into subsequent TRIDE to incorporate collapse awareness for the optimization of perturbations to prevent model collapse. Specifically, TRIDE starts with CAP and alternates between CAP and ANP to perturb the triplets with the dedicated adversarial losses, L_{CAP} and L_{ANP}. These perturbed triplets are then fed into the model for training, followed by another mini-batch of CA-TRIDE until the training is finished. The full process of generating perturbed triplets using CA-TRIDE is described in Algorithm 1. Note that our CA-TRIDE does not incur any extra epochs but only changes the number of perturbed samples per epoch.

3.2. Collapse-Aware Adversary (CA)

To incorporate collapse awareness in the optimization of perturbation and prevent model collapse, we first introduce *collapseness* as a novel metric to capture model states proactively and accurately.

Collapseness. Hard triplets (i.e. $H > 0$) are considered the main cause of model collapse due to their difficulty in optimization, according to current research (Xuan et al., 2020; Schroff et al., 2015). Furthermore, the collapsing speed and severity depend on how many and how hard these triplets are. Although an intuitive practice is to use H as a metric, it is not a feasible option as H only 'reports' model collapse after it occurs and cannot 'forecast' it beforehand, not to mention capturing an impending collapse. Thus,

Algorithm 1 Generating Adversarial Triplets in **CA-TRIDE**

```
Require: Clean triplets $\mathbb{T} = (\mathbf{A}, \mathbf{P}, \mathbf{N})$, maximum PGD steps $M$, PGD step size $\alpha$
Ensure: Adversarial Triplet $\tilde{\mathbb{T}}$
Initialize $\tilde{\mathbb{T}}_0 \leftarrow \mathbb{T}$
if CAP then
    for $i \leftarrow 1$ to $M$ do
        $\delta_i \leftarrow \alpha \nabla_\delta L_{CAP}(\tilde{\mathbb{T}}_{i-1})$
        $\tilde{\mathbb{T}}_i \leftarrow \big(\mathbf{A}, \tilde{\mathbf{P}}_{i-1} + \delta_i, \tilde{\mathbf{N}}_{i-1} + \delta_i\big)$
    end for
else if ANP then
    for $i \leftarrow 1$ to $M$ do
        $\delta_i \leftarrow \alpha \nabla_\delta L_{ANP}(\tilde{\mathbb{T}}_{i-1})$
        $\tilde{\mathbb{T}}_i \leftarrow (\tilde{\mathbf{A}}_{i-1} + \delta_i, \mathbf{P}, \mathbf{N})$
    end for
end if
return $\tilde{\mathbb{T}} \leftarrow \tilde{\mathbb{T}}_M$
```

to guide subsequent counter-measurement against model collapse, we propose *Collapsenss* as a proactive metric for model states evaluation, denoted as $\mathcal{C}$:

$$\mathcal{C}(\mathbf{A}, \mathbf{P}, \mathbf{N}) = d_\omega(\mathbf{A}, \mathbf{P}) - d_\omega(\mathbf{A}, \mathbf{N}) \tag{5}$$

$d_\omega(\cdot, \cdot)$ is a weighting function focusing on anchor-proximity samples. which is calculated as follows:

$$d_\omega(\mathbf{A}, \mathbf{P}) = \frac{\sum_i^{\mathbf{A}, \mathbf{P}} \big(w_{p_i} \cdot d(a_i, p_i)\big)}{\sum_i^{\mathbf{P}} w_{p_i}} \tag{6}$$

$$w_{p_i} = exp\Big(- \lambda \big(d(a_i, p_i) - \min_{\forall a_i \in \mathbf{A}, p_i \in \mathbf{P}} d(a_i, p_i) \big) \Big) \tag{7}$$

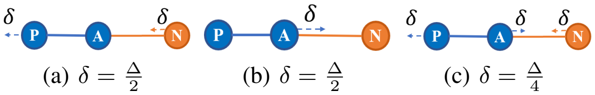
where λ is the attention factor to determine how much attention $\mathcal{C}$ should pay to anchor-proximity samples, and $exp(\cdot)$ is conventionally adopted to map all inputs into $[0, 1]$, with a larger weight for closer samples and a smaller value for others. $d_\omega(\mathbf{A}, \mathbf{N})$ can be calculated similarly. **Positive $\mathcal{C}$ signals an impending model collapse, while negative $\mathcal{C}$ suggests a moderate hardness for the model.**

As a cause of model collapse (Xuan et al., 2020), our $\mathcal{C}$ can detect undergoing model collapse and track the model state proactively by focusing more on neighbouring samples. We also evaluate both H and $\mathcal{C}$ w.r.t. tracking model states to demonstrate the superiority of $\mathcal{C}$, presented in Section 4.4.

Collapse-aware adversary. With our proposed $\mathcal{C}$, the adversarial goal of our subsequent perturbation optimization shifts from directing maximizing H into maximizing $\mathcal{C}$:

$$\arg\max_\delta \mathcal{C}(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}, \tilde{\mathbf{N}}) \tag{8}$$

The definition of our collapse-aware adversary is intuitive as it utilizes the proposed $\mathcal{C}$ to optimize its adversary, while $\mathcal{C}$ also provides feedback for the optimization of the adversary to keep it aware of how well the model handles these perturbed samples.



(a) $\delta = \frac{\Delta}{2}$ (b) $\delta = \frac{\Delta}{2}$ (c) $\delta = \frac{\Delta}{4}$

Figure 3. An illustration of how the same desired hardness change Δ is transferred into sample-wise perturbation δ using different perturbation methods: (a) CAP, (b) ANP, and (c) existing methods (i.e. HM and ACT). Our CAP and ANP double the averaged embedding shift in this specific case.

In general, the maximization goal in Equation 8 determines the strength of AT. Since our primary goal of introducing a collapse-aware adversary is to prevent model collapse during training, we intuitively keep $\mathcal{C} \leq 0$ during the optimization of all perturbations to ensure that the adversary does not cause a severe model collapse while remaining reasonably strong to bring sufficient adversarial robustness.

3.3. Triplet Decoupling (TRIDE)

To address the weak adversary, we propose a new AT paradigm dubbed TRIDE to decouple the current perturbation method into two stages, candidate perturbation (CAP) and anchor perturbation (ANP), as shown in Figure 1. Combining with our collapse-aware adversary, Equation 8 can be rewritten as:

$$\arg\max_\delta \begin{cases} \mathcal{C}(\tilde{\mathbf{A}}, \mathbf{P}, \mathbf{N}), & ANP \\ \mathcal{C}(\mathbf{A}, \tilde{\mathbf{P}}, \tilde{\mathbf{N}}), & CAP \end{cases} \tag{9}$$

As depicted in Figure 2, CAP and ANP are alternated mini-batch-wise, i.e. one perturbation method per mini-batch.

Given a desired hardness change Δ, the generated perturbation δ varies according to perturbation methods. Besides, the angular relationships between samples could also influence δ. To eliminate the influence of angle and provide intuition on how perturbation methods impact δ, we use a special case in Figure 3 to illustrate the difference between CAP, ANP, and the existing method. Analysis of more general cases can be found in Section A of the Appendix.

We now introduce the adversarial loss for CAP and ANP, which is designed following the inherent traits of each stage.

Candidate perturbation (CAP). CAP perturbs all candidates (positives and negatives) while keeping the anchors fixed during perturbation optimization. L_{CAP} is intuitively designed to maximize the aforementioned 'collapsness' $\mathcal{C}$ in Equation 5, by minimizing the following loss function:

$$L_{CAP} = \max\big(-\mathcal{C}, 0\big) \tag{10}$$

where the $\max(\cdot)$ term clips the loss to 0 to terminate the optimization once $\mathcal{C} \geq 0$. In this way, the optimization of CAP becomes collapse-aware and properly stops once the adversary becomes unacceptably strong for the model, thus preventing models from confronting severe model collapse.

CAP corrupts the global embedding (all negatives and positives) and helps the model defend against Ranking attacks (i.e. manipulating the ranks of one or more samples) by enhancing **global robustness**.

Anchor perturbation (ANP). ANP directly pushes the anchor away from the positives and towards the negatives. Similarly to CAP, the goal of L_{ANP} is also to maximize $\mathcal{C}$, given below:

$$L_{ANP} = \max\big(-\mathcal{C} + \Delta_{TR}, 0\big) \tag{11}$$

ANP specializes in efficiently corrupting the local (anchor-proximity) embeddings, making it a practical implementation for black-box attacks against retrieval performance (Chen et al., 2021; Li et al., 2021), i.e. Recall attacks. We propose a top-rank pair, consisting of a top-rank term Δ_{TR} and a top-rank triplet L_{TR}, to fully exploit such traits and help models acquire better resistance against such attacks through stronger **local** (top-rank) **robustness**.

Top-rank pair. The top-rank term Δ_{TR} is designed to push anchors towards the closest negatives further and maximize the embedding shift of anchors as $\mathcal{C}$ approaches 0, to reinforce local corruption:

$$\Delta_{TR} = e^{\max(\mathcal{C},0)}\big(d(\mathbf{A}, \mathbf{N}_{\upsilon}) - d(\mathbf{A}, \mathbf{A_0})\big) \tag{12}$$

where $\mathbf{N}_{\upsilon}$ represents the top half of negatives, ranked by their distances to anchors, and $\mathbf{A_0}$ stands for original unperturbed anchors. The $exp(\cdot)$ term ensures Δ_{TR} only kicks in as $\mathcal{C}$ approaches 0, i.e. $\mathbf{A}$ approaches $\mathbf{N}$.

The top-rank triplet L_{TR} is similarly defined using the triplet loss $L_{\mathcal{T}}$, pairing with Δ_{TR} to help models capture locality:

$$L_{TR} = \gamma\big(d(\mathbf{A}, \mathbf{P}_{\upsilon}) - d(\mathbf{A}, \mathbf{N}_{\upsilon}) + \beta_{TR}\big) \tag{13}$$

where γ is a pre-determined coefficient, and $\mathbf{P}_{\upsilon}$ is similarly defined as $\mathbf{N}_{\upsilon}$, *i.e.*, the top half of positives, ranked by their distances to the anchor.

In sum, our top-rank pair, Δ_{TR} and L_{TR}, works collaboratively to further boost the local robustness of models against prevailing black-box attacks deployed on anchors (queries) (Chen et al., 2021; Li et al., 2021). Δ_{TR} reinforces local corruption for stronger, more targeted adversaries, while L_{TR} helps the model to capture locality.

Model training. As delineated in Equation 4, perturbed triplets generated through TRIDE are subsequently used for training a robust model by optimizing:

$$\arg\min_{\Theta} \begin{cases} L_{\mathcal{T}}(\tilde{\mathbf{A}}, \mathbf{P}, \mathbf{N}; \Theta) + L_{TR}, & ANP \\ L_{\mathcal{T}}(\mathbf{A}, \tilde{\mathbf{P}}, \tilde{\mathbf{N}}; \Theta), & CAP \end{cases} \tag{14}$$

In general, our TRIDE follows the lead of $\mathcal{C}$ to push the model to its limit by alternatively perturbing the triplets through CAP and ANP, enhancing its **global robustness** and **local robustness** respectively, without incurring disastrous model collapse. More implementation details can be found in Appendix C.

3.4. Robustness evaluation metrics

Existing metrics. Existing works use ERS as the robustness evaluation metric, including 10 attacks for evaluation (details can be found in (Zhou et al., 2021b)), which we categorize into **Ranking attacks** and **Recall attacks**. Ranking attacks, such as CA+ and CA-, evaluate the global robustness (i.e. overall ranking), while Recall attacks focus on evaluating local robustness by corrupting the proximity of anchors to lower overall retrieval performance (i.e. R@k).

However, there are issues in ERS that make it infeasible for reasonable robustness evaluation: (1) *Inconsistent similarity metrics.* ERS includes TMA (Tolias et al., 2019), the only cosine-similarity-based attack, as one of the attacks for robustness evaluation, which is not appropriate for evaluating models trained in the Euclidean space. Experimental results also indicate that TMA yields contradictory results against all other attacks (see Table.2 and Table.6 in (Zhou & Patel, 2022)). (2) *Initial state variations.* ERS is calculated using after-attack results, without considering the variation in the initial states before attacks. For example, in the CA+ attack (which raises the rank of a candidate over a query), despite the **randomly** selected targets, ERS ignores the variation in their before-attack ranks, *e.g.*, 50 ± 1. This introduces noise into ERS and makes it inaccurate. Similar issues persist for Recall attacks: since ERS calculates a joint total score, clean R@1 cannot become a reference for justification.

Adversarial Resistance Score. To solve the issues above, we remove TMA from evaluation for fairness consideration, and ES:D is also removed as it only considers attack-incurred embedding shifts and does not necessarily equal robustness. Specifically, the Adversarial Resistance Score (ARS) of an attack $\mathcal{A}$ against a model $\mathbf{M}$ is calculated based on the actual impact it makes compared to the intention of the attack, rather than directly using the results, defined as follows:

$$\mathbb{R}_{\mathbf{M},\mathcal{A}} = \big(1 - \frac{\mathcal{O}_r - \mathcal{O}_i}{\mathcal{O}_g - \mathcal{O}_i}\big) \times 100\%, \tag{15}$$

where $\mathcal{O}_i$ refers to the initial value before attacks, $\mathcal{O}_r$ is the

Table 1. Robustness of ACT, HM, and our CA-TRIDE under our proposed metric, Adversarial Resistance Score (ARS).

Dataset	Defense	PGD	Benign Example Evaluation				Adversarial Example Evaluation (ARS scores)								Overall
	Method	steps	R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↑	QA+ ↑	QA- ↑	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	ARS (%) ↑
CUB	N/A	N/A	58.9	66.4	26.1	59.5	3.3	0.0	0.0	0.0	0.0	0.0	23.9	0.0	3.5
	ACT	32	27.5	38.2	12.2	43.0	31.0	62.9	30.2	68.5	40.3	34.2	54.2	1.0	40.3
	HM	32	34.9	45.0	19.8	47.1	31.0	62.9	33.2	69.8	51.3	47.9	**78.2**	2.9	47.2
	Ours	**16**	34.9	45.1	19.6	45.6	**32.6**	**68.5**	**41.8**	**79.2**	**61.9**	**59.0**	64.8	**5.1**	**51.6**
CARS	N/A	N/A	63.2	75.3	36.6	55.6	0.4	0.0	0.0	3.6	0.0	0.0	21.2	0.0	2.8
	ACT	32	43.4	54.6	11.8	42.9	36.0	68.4	35.0	70.2	37.6	35.3	47.7	1.6	41.4
	HM	32	60.2	71.6	33.9	51.2	**38.6**	74.8	39.2	75.1	50.3	61.0	**76.4**	8.8	52.9
	Ours	**16**	60.7	71.2	34.6	49.4	36.0	**81.0**	**47.0**	**87.5**	**64.4**	**66.9**	60.8	**13.7**	**57.2**
SOP	N/A	N/A	62.9	68.5	39.2	87.4	0.2	0.6	0.3	0.9	0.0	0.0	10.0	0.0	1.5
	ACT	32	47.5	52.6	25.5	84.9	48.2	90.4	45.4	91.5	44.6	45.5	58.5	15.3	54.9
	HM	32	46.8	51.7	24.5	84.7	64.0	96.8	67.4	**98.0**	83.5	85.0	81.0	45.6	77.7
	Ours	**16**	48.3	53.3	25.9	84.9	**65.8**	**97.1**	**71.4**	97.9	**89.4**	**93.4**	**82.4**	**53.1**	**81.3**

Table 2. Robustness of ACT, HM, and our CA-TRIDE under the conventional metric, Empirical Robustness Score (ERS) (Zhou et al., 2021b).

Dataset	Defense	PGD	Benign Example Evaluation				Adversarial Example Evaluation (ERS scores)										Overall
	Method	steps	R@1 ↑	R@2 ↑	mAP ↑	NMI ↑	CA+ ↑	CA- ↓	QA+ ↑	QA- ↓	TMA ↓	ES:D ↓	ES:R ↑	LTM ↑	GTM ↑	GTT ↑	ERS↑
CUB	N/A	N/A	58.9	66.4	26.1	59.5	0.0	100	0.0	99.9	0.883	1.76	0.0	0.0	14.1	0.0	3.8
	ACT	32	27.5	38.2	12.2	43.0	15.5	37.7	15.1	32.2	**0.47**	0.82	11.1	9.4	14.9	1.0	33.9
	HM	32	34.9	45.0	19.8	47.1	15.5	37.7	16.6	30.9	0.75	0.50	17.9	16.7	**27.3**	2.9	36.0
	Ours	**16**	34.9	45.1	19.6	45.6	**16.7**	**31.1**	**20.9**	**21.1**	0.97	**0.16**	**21.6**	**20.6**	22.6	**5.1**	**38.6**
CARS	N/A	N/A	63.2	75.3	36.6	55.6	0.2	100.0	0.1	97.3	0.87	1.82	0.0	0.0	13.4	0.0	3.6
	ACT	32	43.4	54.6	11.8	42.9	18.0	32.3	17.5	30.5	**0.38**	0.76	16.3	15.3	20.7	1.6	38.6
	HM	32	60.2	71.6	33.9	51.2	**19.3**	25.9	19.6	25.7	0.65	0.45	30.3	36.7	**46.0**	8.8	46.1
	Ours	**16**	60.7	71.2	34.6	49.4	17.7	**20.3**	**23.5**	**12.9**	0.96	**0.13**	**39.1**	**40.6**	36.9	**13.7**	**47.7**
SOP	N/A	N/A	62.9	68.5	39.2	87.4	0.1	99.3	0.2	99.1	0.85	1.69	0.0	0.0	6.3	0.0	4.0
	ACT	32	47.5	52.6	25.5	84.9	24.1	10.5	22.7	9.4	**0.25**	0.53	21.2	21.6	27.8	15.3	50.8
	HM	32	46.8	51.7	24.5	84.7	32.0	4.2	33.7	3.0	0.61	0.20	39.1	39.8	37.9	45.6	61.6
	Ours	**16**	48.3	53.3	25.9	84.9	**32.3**	**3.7**	**36.0**	**2.6**	0.80	**0.14**	**43.2**	**45.1**	**39.8**	**53.1**	**62.4**

actual attacking result, and $\mathbb{O}_g$ is the intended result of the attack, respectively.

Detailed calculations of ASR for different types of attacks can be found in Appendix B.

Unlike ERS, our ARS calculates the percentage of change over the attack's intention, eliminating the variation in initial conditions for a more reasonable robustness evaluation.

4. Experiments

In this section, we conduct comprehensive experiments to demonstrate the effectiveness of our CA-TRIDE, involving its comparison to other defense baselines, the validation of CA in preventing model collapse and TRIDE in solving the weak adversary, and ablation studies on the main components of our CA-TRIDE.

4.1. Experimental Settings

Models and datasets. We follow the setting of Zhou et al. (2021b) and use a pre-trained ResNet-18 (He et al., 2015) with the last layer changed to N=512 as our baseline model. The triplet margin $\beta_{\mathcal{T}}$ is set as 0.2 for all datasets. Evaluations are on three popular datasets in image retrieval tasks, i.e. CUB-200-2011 (Welinder et al., 2010), Cars-196 (Krause et al., 2013), and SOP (Oh Song et al., 2016). We train our models using ADAM(Kingma & Ba, 2014) optimizer with a 1.0×10^{-3} learning rate, a mini-batch size of 112, and training epochs of 100 under the above three datasets. For the top-rank pair, $\gamma = 0.5$ and the triplet margin in L_{TR} β_{TR} is 0.04.

Adversaries. Adversarial perturbation is generated through PGD (Madry et al., 2017) with an optimization step $\alpha = 1/255$, 16 iterations and clipped by an l_∞ norm of $\epsilon = 8/255$. A progressive PGD step size α is also deployed to help the model balance between accuracy and robustness. All CA-TRIDE implementation details are the same unless specified. Details are given in Section C of the Appendix.

Metrics. Benign results are given as R@1, R@2, mAP, and NMI following the setting in Zhou & Patel (2022), while robustness scores are calculated using both the conventional metric, Empirical Robustness Score (ERS)(Zhou et al., 2021b), and our proposed metric, Adversarial Resistance Score (ARS).

4.2. CA-TRIDE vs. Other Defense Methods

We compare our CA-TRIDE to the existing SOTA methods, HM (Zhou & Patel, 2022) and ACT (Zhou et al., 2021b), regarding their performance on both benign and adversarial examples, based on both ARS and ERS. Results for ARS are presented in Table 1, and ERS results are given in Table

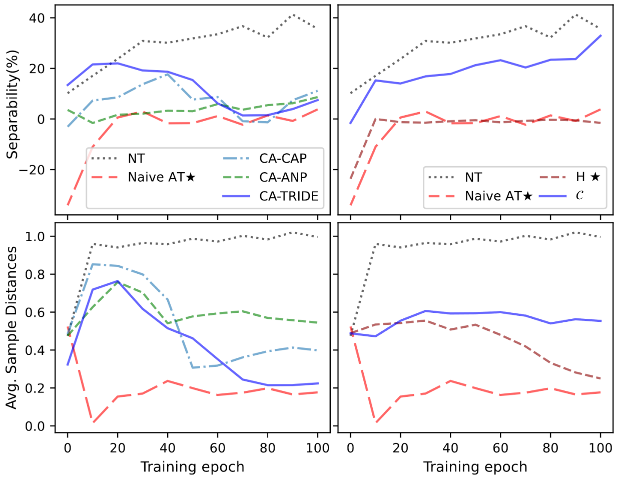


Figure 4. **Left Column:** Our TRIDE leads to lower separability and larger shrunk embedding distances compared with CAP/ANP, but the model remains uncollapsed due to our CA. **Right Column:** Our collapseness $\mathcal{C}$ outperforms hardness H in preventing model collapse. ★ denotes model collapse, i.e. separability ≈ 0 and heavily shrunk embeddings. The dataset is CUB.

2. From both tables, we can draw the following conclusions: (1) CA-TRIDE significantly outperforms HM and ACT in almost all attacks under both metrics, which demonstrates the effectiveness of our CA-TRIDE. This suggests that **CA-TRIDE models can undertake stronger AT without model collapse**. (2) CA-TRIDE uses only half the PGD steps of the previous methods but exhibits higher robustness, at the cost of little or no drop in benign performances. This implies that **our TRIDE settings provide stronger adversaries and a more efficient AT paradigm.**

As for the relatively lower performance of our models in TMA scores, it does not necessarily mean lower performance because all models are trained in the Euclidean space while TMA is evaluated using cosine similarity. Moreover, it is noticeable that models with higher overall robustness (i.e. HM and ours) constantly score lower in TMA than less robust models (ACT), which contradicts all other attacks. As for GTM, this reflects the trade-off between emphasizing local robustness (top-rank samples) and global robustness (overall samples). In CA-TRIDE, we propose a top-rank pair to emphasize local robustness. However, GTM naively pushes the anchor towards the *top-1 closest* negative example rather than a subset of top-rank samples, which contradicts the goal of our top-rank pair that prioritizes top-half negatives, thus leading to relatively lower scores. The weakness of GTM is also implied by the non-zero attacking results on undefended models in Table 1 and Table 2.

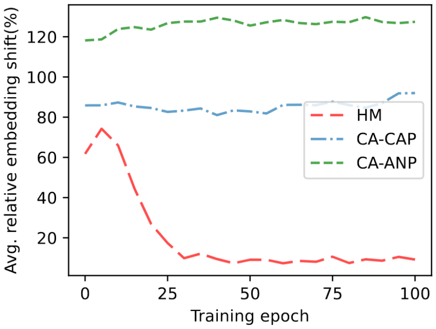


Figure 5. Our CA-ANP and CA-CAP cause substantially larger embedding shifts than HM. The dataset is CUB.

4.3. Validation of CA and TRIDE

In this section, we provide analyses to validate the individual effectiveness of our CA and TRIDE. To this end, we train the following models: an NT model trained on vanilla data, a Naive AT model trained using a conventional adversary, a CA-CAP model, a CA-ANP model, and a CA-TRIDE model (i.e. our full implementation).

CA prevents model collapse. To visualize model states, we calculate the per mini-batch average sample distances $\bar{d}$ and use *separability*, defined as $\frac{d(a,n)-d(a,p)}{\bar{d}}$, to evaluate how well the model separates positive and negative samples. In particular, normalization by $\bar{d}$ eliminates the influence of overall embedding changes.

As shown in the left column of Figure 4, benign training (black dotted line) maximizes the separation between positives and negatives, without incurring a decrease in average sample distances (i.e. $\bar{d}$). However, naive AT yields a drastic decrease in $\bar{d}$ and harms separability by making it largely negative and fluctuating around 0. We thus summarise *embedding shrinkage* and *entangled samples* as two representative manifestations of model collapse. The separability of CA-CAP and CA-ANP remains positive during training, and $\bar{d}$ undergoes a mild shrinkage. This shows the effectiveness of collapse awareness in stopping model collapse, i.e. preventing embedding shrinkage and entangled samples.

Finally, to validate the superiority of $\mathcal{C}$ as a novel metric to track model states proactively, we train two models using CA-TRIDE but with different metrics: our collapse-aware adversary and hardness-aware adversary, denoted as $\mathcal{C}$ and H, respectively. Progressive α is disabled for a fairer comparison. Results are given in the right column of Figure 4. The behavior of the H-oriented model resembles that of Naive AT, both of which confront model collapse. The C-oriented model, on the other hand, behaves similarly to NT, implying the superiority of our $\mathcal{C}$.

TRIDE solves the weak adversary. To validate the effectiveness of our TRIDE, we compare the averaged embedding shifts caused by perturbations generated using three

Table 3. Ablation study on CA-TRIDE vs. CA-CAP/-ANP. The dataset is CUB.

Defense	R@1↑	Adversarial Example Evaluation (ARS scores)								Overall	Overall
Method		CA+↑	CA-↑	QA+↑	QA-↑	ES:R ↑	LTM↑	GTM ↑	GTT ↑	ERS↑	ARS (%) ↑
CA-ANP	34.2	27.4	56.7	35.6	73.3	57.3	61.1	65.8	5.1	34.0	47.8
CA-CAP	33.8	34.2	68.0	52.2	70.8	51.2	47.6	60.7	3.1	37.9	48.5
CA-TRIDE	34.9	32.6	68.5	41.8	79.2	61.9	59.0	64.8	5.1	**38.6**	**51.6**

Table 4. Ablation study on the top-rank pair (TPR), i.e. Δ_{TR}+L_{TR}. The dataset is CUB.

Defense	R@1↑	Adversarial Example Evaluation (ARS scores)								Overall	Overall
Method		CA+↑	CA-↑	QA+↑	QA-↑	ES:R ↑	LTM↑	GTM ↑	GTT ↑	ERS↑	ARS (%) ↑
w/o TPR	32.7	27.8	60.2	36.6	69.1	46.2	26.3	47.4	0.7	33.3	39.3
w/o L_{TR}	34.6	**34.6**	**70.0**	**43.4**	79.1	57.8	54.9	60.7	4.0	38.4	50.6
CA-TRIDE	34.9	32.6	68.5	41.8	**79.2**	**61.9**	**59.0**	**64.8**	**5.1**	**38.6**	**51.6**

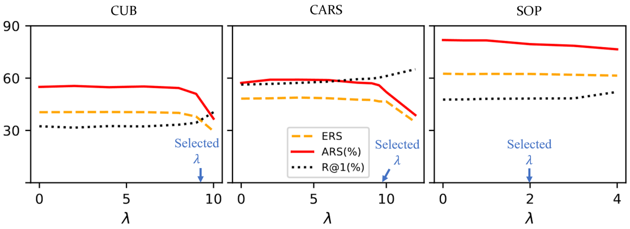


Figure 6. Ablation study on the attention factor λ.

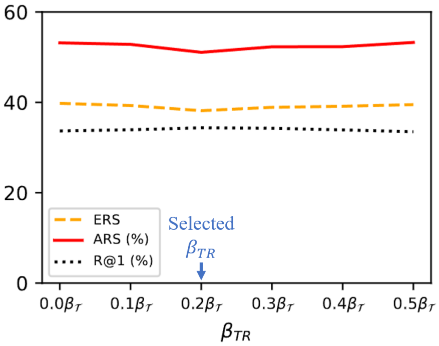


Figure 7. Ablation study on the triplet loss margin β_{TR} in L_{TR}. The dataset is CUB.

methods: HM, CAP, and ANP. All values are normalized similarly as separability to eliminate the influence of embedding shrinkage. As shown in Figure 5, our CA-CAP and CA-ANP consistently yield larger embedding shifts than HM, especially in the later stage of the training. Our TRIDE surpasses HM regarding both average and overall embedding shifts. Here the overall embedding shift equals the averaged embedding shift multiplied by its corresponding number of perturbed components, i.e., 1 for ANP, 2 for CAP, and 3 for HM. This demonstrates the efficacy of TRIDE in maximizing embedding shifts. Qualitative analysis also aligns with this result, as discussed in Appendix A. To further investigate the individual effectiveness of TRIDE, we train three model variants without using CA, i.e. a naive CAP model, a naive ANP model, and a naive TRIDE model. Their R@1 results are 4.4%, 7.8% and 0.8% respectively, indicating that, as expected, both CAP/ANP/TRIDE could yield strong adversaries sufficient to cause model collapse.

4.4. Ablation Studies

In this section, we verify the effectiveness of CA-TRIDE by providing the ablation experiment results in Table 3 and 4.

TRIDE vs CAP/ANP. As shown in Table 3, TRIDE does not necessarily reduce the overall strength of AT, and CA-CAP/ANP models emphasize differently on robustness. CA-CAP exhibits better robustness in almost all ranking attacks than CA-ANP, while the latter is more robust against Recall attacks, aligning with our intention of designing CAP and ANP. Finally, combining CAP and ANP yields well-rounded robustness across all attacks and achieves the best overall robustness.

Top-rank pair. As discussed in Section 3.3, the top-rank pair is incorporated to reinforce local corruption in ANP and help the model capture top-rank robustness against Recall attacks. According to the results presented in Table 4, the model with only Δ_{TR} already gains over 10% of robustness boost against all attacks compared to its counterpart without our top-rank pair, with an impressive enhancement of performance on clean examples. This indicates the novelty of Δ_{TR} regarding the AT efficiency. Furthermore, when paired with the top-rank triplet L_{TR}, our model further acquires a noticeable increase in robustness against all Recal attacks, but at the cost of a marginal drop w.r.t. ranking attacks, implying a trade off between global and local robustness.

Hyperparameters. We also investigate how our models behave under different hyperparameters, i.e. the attention

Table 5. Statistical results of inter-/intra-class distances on all datasets, evaluated using our CA-TRDIE models. Inter/Intra quantifies the level of entanglement of data distribution.

Dataset	Inter-Class Distances	Intra-Class Distances	Entanglement	λ
CUB	0.287	0.226	0.79	10.0
CARS	0.325	0.256	0.79	9.5
SOP	0.664	0.438	0.66	2.0

factors λ in $\mathcal{C}$ and the triplet loss margin β_{TR} in L_{TR}. λ determines how much attention $\mathcal{C}$ should pay to anchor-proximity samples, as introduced in Equation 5. Larger λ means more focus on samples closer to anchors and less focus on the other samples. Note that when $\lambda = 0$, our $\mathcal{C}$ falls back to H by treating all examples identically. Results are presented in Figure 6. Overall, our CA-TRIDE is insensitive to λ, while a too-large λ could further increase the clean accuracy but harm the robustness. Hence, we choose the value of λ that yields the closest R@1 performance to HM for a fair comparison, i.e. 10 for CUB, 9.5 for CARS, and 2.0 for SOP.

Moreover, as shown in Table 5, we find our settings of λ on different datasets are correlated with their levels of entanglement, i.e., d_{intra}/d_{inter}, where d_{intra} (d_{inter}) is the average intra (inter)-class distance of CA-TRIDE trained models. This correlation aligns with the intuition that more entangled datasets (i.e., CUB and CARS) require more attention on the anchor-proximity samples to avoid model collapse, while less entangled datasets such as SOP do not require much attention to handle such samples.

For β_{TR} in L_{TR}, it functions similarly to the $\beta_{\mathcal{T}}$ in the triplet loss: determining how hard the model should keep top-rank positives and negatives separated. Likewise, L_{TR} is essentially a triplet loss variant calculated using only top-rank samples instead of all samples, and we thus evaluate the β_{TR} as a proportion of $\beta_{\mathcal{T}}$, ranging from 0 to 50%. As shown in Figure 7, no significant changes in robustness are found as β_{TR} varies. Therefore, our method is also insensitive to the hyperparameter β_{TR}. We thus choose the value of β_{TR} that yields the closest R@1 performance as HM for a fair comparison, i.e. $\beta_{TR} = 0.2\beta_{\mathcal{T}}$.

4.5. Training Time and Computational Cost

Since CA-TRIDE introduces extra operations during AT, we further compare it to HM regarding training time and computational cost. Results imply that both are reduced. We conducted 5 runs of HM and CA-TRIDE on the CUB dataset with an RTX3090 GPU. On average, CA-TRIDE takes 15% less training time: 470 vs 550 minutes. We attribute this efficiency increase of CA-TRIDE to its triplet decoupling and the halved PGD steps, i.e., 16 (CA-TRIDE) vs. 32 (HM). For the computational cost, since CAP and ANP only use 2/3 (P and N) and 1/3 (A) of the triplets respectively, our CA-TRIDE yields only a $\frac{1}{2} \times (\frac{1}{3} + \frac{2}{3}) = \frac{1}{2}$ computational cost of HM (and ACT).

4.6. Compatibility of CA-TRIDE

Our CA-TRIDE is a plug-and-play method for other triplet-based deep metric learning, regardless of datasets, models, and methods. Specifically, our CA-TRIDE does not change adversarial training at the system level but at the data sample level, by simply applying a weight to all samples (via our CA) and decoupling the perturbation update on them (via our TRIDE). In particular, it is worth mentioning that the only tunning possibly required is to find the attention factor λ for the new dataset or model (if applicable) because as validated in Table 5, λ is dataset-/model-dependent as it correlates with the level of entanglement.

5. Limitations and Future Work

Globality vs locality. As presented in Table 1 and 2, our CA-TRIDE achieves lower results than HM w.r.t. GTM attacks on both CUB and CARS. This might be due to the conflict between the design of our top-rank pair and the attacking mechanism of GTM. A more sophisticated design of the top-rank pair may potentially mitigate these gaps. Besides, to trade off globality and locality, we empirically choose the top-half positives and negatives as top-rank samples when designing L_{TR}. It is possible to explore further refinement to such a trade-off in the future.

Attention factor λ. Through our observation in Table 5, the level of entanglement, i.e., d_{intra}/d_{inter}, and our chosen attention factor λ are found to be correlated. Therefore, a more rigorous and adaptive collapse-aware mechanism can be designed to adjust λ better accordingly.

6. Conclusion

In this paper, we have proposed CA-TRIDE, a novel approach to training image retrieval models with stronger adversarial robustness. CA-TRIDE addresses two overlooked limitations: weak adversary and model collapse. Specifically, TRIDE yields a strong adversary by spatially decoupling the optimization of perturbation on the triplets into ANP and CAP, while CA captures intermediate model states using a novel metric called *collapseness* and integrates it into the subsequent optimization of perturbations. For evaluation metrics, we examine the conventional robustness evaluation metric ERS and identify two drawbacks. Consequently, we propose a new metric called ARS to address these drawbacks accordingly for reasonable robustness evaluations. Extensive experiments on three datasets validate the superiority of our CA-TRIDE in both ERS and ARS.

Acknowledgments

This research is supported by the National Key Research and Development Program of China (2021YFB3100700), the National Natural Science Foundation of China (62376210, 62161160337, 62132011, U21B2018, U20A20177, 62206217), the Shaanxi Province Key Industry Innovation Program (2023-ZDLGY-38, 2021ZDLGY01-02).

Impact Statement

This paper presents work that aims to advance the field of adversarial defense in image retrieval. Potential societal consequences of our work are less influential and have become scientific consensus, none of which we feel must be specifically highlighted here.

References

Bai, J., Chen, B., Li, Y., Wu, D., Guo, W., Xia, S., and Yang, E. Targeted attack for deep hashing based retrieval. *CoRR*, abs/2004.07955, 2020. URL https://arxiv.org/abs/2004.07955.

Chen, M., Lu, J., Wang, Y., Qin, J., and Wang, W. Dair: A query-efficient decision-based attack on image retrieval systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pp. 1064–1073, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462887. URL https://doi.org/10.1145/3404835.3462887.

Feng, Y., Chen, B., Dai, T., and Xia, S. Adversarial attack on deep product quantization network for image retrieval. *CoRR*, abs/2002.11374, 2020. URL https://arxiv.org/abs/2002.11374.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Li, J., Ji, R., Liu, H., Hong, X., Gao, Y., and Tian, Q. Universal perturbation attack against image retrieval. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4898–4907, 2019. doi: 10.1109/ICCV.2019.00500.

Li, X., Li, J., Chen, Y., Ye, S., He, Y., Wang, S., Su, H., and Xue, H. QAIR: practical query-efficient black-box attacks for image retrieval. *CoRR*, abs/2103.02927, 2021. URL https://arxiv.org/abs/2103.02927.

Liu, Z., Zhao, Z., and Larson, M. A. Who's afraid of adversarial queries? the impact of image modifications on content-based image retrieval. *CoRR*, abs/1901.10332, 2019. URL http://arxiv.org/abs/1901.10332.

Lu, J., Chen, M., Sun, Y., Wang, W., Wang, Y., and Yang, X. A smart adversarial attack on deep hashing based image retrieval. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, ICMR '21, pp. 227–235, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384636. doi: 10.1145/3460426.3463640. URL https://doi.org/10.1145/3460426.3463640.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Musgrave, K., Belongie, S., and Lim, S.-N. A metric learning reality check. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pp. 681–699. Springer, 2020.

Oh Song, H., Xiang, Y., Jegelka, S., and Savarese, S. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

Pang, T., Yang, X., Dong, Y., Xu, K., Zhu, J., and Su, H. Boosting adversarial training with hypersphere embedding. *Advances in Neural Information Processing Systems*, 33:7779–7792, 2020.

Picot, M., Messina, F., Boudiaf, M., Labeau, F., Ayed, I. B., and Piantanida, P. Adversarial robustness via fisher-rao regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022. doi: 10.1109/tpami.2022.3174724. URL https://doi.org/10.1109%2Ftpami.2022.3174724.

Roth, K., Milbich, T., Sinha, S., Gupta, P., Ommer, B., and Cohen, J. P. Revisiting training strategies and generalization performance in deep metric learning. In *International Conference on Machine Learning*, pp. 8242–8252. PMLR, 2020.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015. URL http://arxiv.org/abs/1503.03832.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tolias, G., Radenovic, F., and Chum, O. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. *CoRR*, abs/1908.09163, 2019. URL http://arxiv.org/abs/1908.09163.

Wang, H., Wang, G., Li, Y., Zhang, D., and Lin, L. Transferable, controllable, and inconspicuous adversarial attacks on person re-identification with deep mis-ranking. *CoRR*, abs/2004.04199, 2020. URL https://arxiv.org/abs/2004.04199.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1386–1393, 2014.

Wang, J., Zhou, F., Wen, S., Liu, X., and Lin, Y. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Wang, X., Han, X., Huang, W., Dong, D., and Scott, M. R. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5022–5030, 2019.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010. URL /se3/wp-content/uploads/2014/09/WelinderEtal10_CUB-200.pdf,http://www.vision.caltech.edu/visipedia/CUB-200.html.

Wu, C.-Y., Manmatha, R., Smola, A. J., and Krahenbuhl, P. Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision*, pp. 2840–2848, 2017.

Xuan, H., Stylianou, A., Liu, X., and Pless, R. Hard negative examples are hard, but useful. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. (eds.), *Computer Vision – ECCV 2020*, pp. 126–142, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58568-6.

Yu, B., Liu, T., Gong, M., Ding, C., and Tao, D. Correcting the triplet selection bias for triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 71–87, 2018.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Zhong, Y. and Deng, W. Adversarial learning with margin-based triplet embedding regularization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6549–6558, 2019.

Zhou, M. and Patel, V. M. Enhancing adversarial robustness for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15325–15334, 2022.

Zhou, M., Wang, L., Niu, Z., Zhang, Q., Xu, Y., Zheng, N., and Hua, G. Practical relative order attack in deep ranking. *CoRR*, abs/2103.05248, 2021a. URL https://arxiv.org/abs/2103.05248.

Zhou, M., Wang, L., Niu, Z., Zhang, Q., Zheng, N., and Hua, G. Adversarial attack and defense in deep ranking. *CoRR*, abs/2106.03614, 2021b. URL https://arxiv.org/abs/2106.03614.

Zhu, J., Zhang, J., Han, B., Liu, T., Niu, G., Yang, H., Kankanhalli, M. S., and Sugiyama, M. Understanding the interaction of adversarial training with noisy labels. *CoRR*, abs/2102.03482, 2021. URL https://arxiv.org/abs/2102.03482.

A. Theoretical Analysis of Weak Adversary

In this section, we provide a qualitative analysis of how and why the weak adversary leads to minimized embedding shifts, which necessitates our triplet decoupling (TRIDE) mechanism.

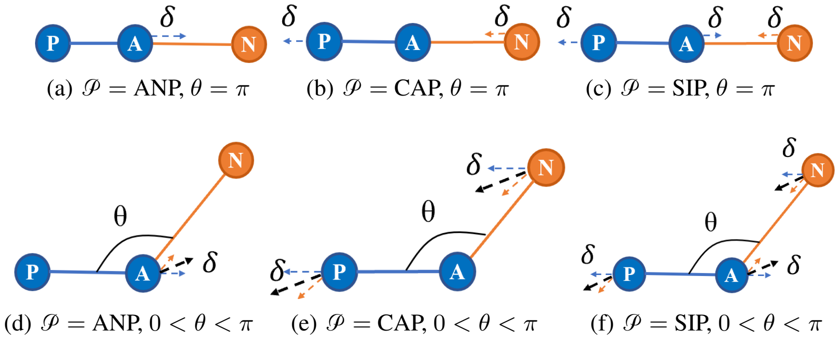
As discussed in the main text of the paper, the current adversarial perturbation δ is acquired by:

$$\arg\max_{\delta} H(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}, \tilde{\mathbf{N}}) \tag{16}$$

Essentially, although the specific adversarial losses L_{adv} vary across methods, the general goal of L_{adv} is to maximize H adversarially. Thus, we denote the consequential change caused by the perturbation as Δ_H, defined as follows:

$$\Delta_H = H(\tilde{\mathbf{A}}, \tilde{\mathbf{P}}, \tilde{\mathbf{N}}) - H(\mathbf{A}, \mathbf{P}, \mathbf{N}) \tag{17}$$

As mentioned in the main paper, unlike AT in image classification, AT in DML has multiple choices for perturbation targets (anchors **A**, positives **P**, and negatives **N**). The average embedding shift of the perturbed targets depends on two factors: the angle between $\bar{an}$ and $\bar{ap}$, denoted as θ (Wang et al., 2017), and the perturbation methods, denoted as $\mathscr{P}$. We will then determine how these factors influence the overall embedding shifts.



(a) $\mathscr{P}$ = ANP, $\theta = \pi$ (b) $\mathscr{P}$ = CAP, $\theta = \pi$ (c) $\mathscr{P}$ = SIP, $\theta = \pi$

(d) $\mathscr{P}$ = ANP, $0 < \theta < \pi$ (e) $\mathscr{P}$ = CAP, $0 < \theta < \pi$ (f) $\mathscr{P}$ = SIP, $0 < \theta < \pi$

Figure 8. Demonstration on scenarios of different angle θ and perturbation method. Δ denotes the average embedding shifts of perturbed samples. ANP for anchor perturbation (only perturbs anchors), CAP for candidates perturbation (only perturbs candidates), and SIP for simultaneous perturbation (perturbs all the triplets). $\pi < \theta < 2\pi$ is omitted due to symmetry.

As shown in Figure 8, θ influences **how the perturbation changes the overall hardness**. In other words, with the same embedding shifts, θ determines the proportion of embedding shifts that transfer to Δ_H. To calculate this proportion, we calculate a function γ_θ of θ. Hence, given $\gamma(\theta)$ and an embedding shift δ, Δ_H is given by:

$$\Delta_H = \gamma_\theta \cdot \delta \tag{18}$$

According to Figure 8, γ_θ could range from 0 ($\theta = 0$, *i.e.*, $\Delta_H = 0$) to 2 ($\theta = \pi$, *i.e.*, $\Delta_H = 2\delta$.)

A.1. Perturbation targets

Perturbation methods $\mathscr{P}$ determine the total number of selected targets, (*i.e.*, 1, 2 or 3), influencing how the desired embedding shift Δ_H is allocated to all perturbation targets. A general way to understand the idea is like assigning a certain amount of tasks (target hardness). The more people (samples) get assigned, the fewer tasks (embedding shifts) each person would have.

For analysis, we assume that all perturbed targets move identically by Δ. In other words, $d_{a,\tilde{a}} = \delta_{ANP}$ for anchor perturbation (ANP), and $d_{p,\tilde{p}} = d_{n,\tilde{n}} = \delta_{CAP}$ for candidate perturbation (CAP), and $d_{a,\tilde{a}} = d_{p,\tilde{p}} = d_{n,\tilde{n}} = \delta_{SIP}$ for simultaneous perturbation (SIP), namely the existing method.

A.2. Calculations

To calculate the averaged embedding shifts given θ and $\mathscr{P}$, we need to determine how much of each perturbed sample needs to move to achieve the required hardness increase $\Delta_H - H(\mathbb{T})$. We further drop the term $H(\mathbb{T})$ as it is identical for the same initial triplet. **Note that to simplify the calculation, we assume all perturbed samples have identical embedding shifts.** We can now qualitatively demonstrate how different perturbation methods $\mathscr{P}$ and θ determine the averaged overall embedding shifts δ given the same Δ_H, in a special-to-general manner:

(1) *Special case*: $\theta = \pi$. For $\mathscr{P} =$ ANP, as shown in Figure 8(a), the gradient to increase $d(a, p)$ aligns with the gradient to decrease $d(a, n)$, which doubles the hardness shift caused by δ_{ANP}. Thus, we can obtain the average embedding shift δ_{ANP} of ANP follows:

$$\begin{aligned} \gamma_\theta \cdot d_{a,\tilde{a}} = \Delta_H \\ 2\delta_{ANP} = \Delta_H \\ \delta_{ANP} = \frac{\Delta_H}{2} \end{aligned} \tag{19}$$

Turning to $\mathscr{P} =$ CAP, as demonstrated in Figure 8(b), $\gamma_\theta = 1$, and δ_{CAP} given as follows:

$$\begin{aligned} \gamma_\theta \cdot d_{p,\tilde{p}} + \gamma_\theta \cdot d_{n,\tilde{n}} = \Delta_H \\ \delta_{CAP} + \delta_{CAP} = \Delta_H \\ \delta_{CAP} = \frac{\Delta_H}{2} \end{aligned} \tag{20}$$

For their combination, SIP leads to the least average embedding shift (as shown in Figure 8(c)), which can be similarly calculated :

$$\begin{aligned} \gamma_\theta \cdot d_{a,\tilde{a}} + \gamma_\theta \cdot d_{p,\tilde{p}} + \gamma_\theta \cdot d_{n,\tilde{n}} = \Delta_H \\ 2\delta_{SIP} + \delta_{SIP} + \delta_{SIP} = \Delta_H \\ \delta_{SIP} = \frac{\Delta_H}{4} \end{aligned} \tag{21}$$

In this case, by comparing Equation 19, Equation 20, and Equation 21, the overall average embedding shift of ANP and CAP is twice the average embedding shift of SIP.

(2) *General cases:* $0 < \theta < \pi$. For ANP, as shown in Figure 8(d), γ_θ becomes a function of θ. Through geometric calculation, we obtain the approximation of γ_θ, given as follows:

$$\gamma_\theta = 2cos(\frac{\pi - \theta}{2}) \tag{22}$$

Then, δ_{ANP} of ANP is calculated as:

$$\begin{aligned} \gamma_\theta \cdot d_{a,\tilde{a}} =& \Delta_H \\ 2cos(\frac{\pi - \theta}{2}) \cdot \delta_{ANP} =& \Delta_H \\ \delta_{ANP} =& \frac{\Delta_H}{2cos(\frac{\pi - \theta}{2})} \end{aligned} \tag{23}$$

This result can be verified by simply inserting $\theta = \pi$ into Equation 23, which gives the same result in Equation 19.

Similar to ANP, CAP can be regarded as applying an equivalent perturbation to **P** and **N**, with γ_θ becoming half of ANP, (*i.e.*, half of $2cos(\frac{\pi - \theta}{2})$) :

$$\begin{aligned} \gamma_\theta \cdot d_{p,\tilde{p}} + \gamma_\theta \cdot d_{n,\tilde{n}} =& \Delta_H \\ 2 \times cos(\frac{\pi - \theta}{2}) \cdot \delta_{CAP} =& \Delta_H \\ \delta_{CAP} =& \frac{\Delta_H}{2cos(\frac{\pi - \theta}{2})} \end{aligned} \tag{24}$$

For SIP, δ_{SIP} can once again be calculated by combining CAP and ANP:

$$\begin{aligned} \gamma_\theta \cdot d_{a,\tilde{a}} + \gamma_\theta \cdot d_{p,\tilde{p}} + \gamma_\theta \cdot d_{n,\tilde{n}} &= \Delta_H \\ 4cos(\frac{\pi-\theta}{2})\delta_{SIP} &= \Delta_H \\ \delta_{SIP} &= \frac{\Delta_H}{4cos(\frac{\pi-\theta}{2})} \end{aligned} \tag{25}$$

Finally, we can compare SIP with CAP + ANP in more general cases ($0 < \theta < \pi$) by comparing Equation 23, Equation 24 and Equation 25. Ideally, with the exactly identical θ, $\Delta_{CAP,ANP}$ is still two times of δ_{SIP}. However, in practical scenarios, there will be a phase difference most of the time. Hence, the magnitude ratio of CAP + ANP over SIP in more general cases can be given as follows:

$$\frac{\Delta_{CAP,ANP}}{\delta_{SIP}} = \frac{2cos(\frac{\pi-\theta_1}{2})}{cos(\frac{\pi-\theta_2}{2})} \tag{26}$$

For $0 < \theta < \pi$, the denominator of the ratio can be regarded as a coefficient ranging from 0 to 1, with most of the time less than 1. This implies that this denominator mostly acts as an amplifier, making the overall ratio even larger. Due to symmetry, the scenario when $\pi < \theta < 2\pi$ is identical to what has been discussed. In other words, despite some variation, Equation 26 aligns with the experimental results shown in Figure 5, which suggests that **our CAP + ANP setting outperforms the extant SIP regarding maximizing the average embedding shift under the same perturbation.**

A.3. Conclusions

Our analysis is also validated experimentally in Section 4.2. Our Tride consistently outperforms SIP w.r.t. the average embedding shift under perturbation, with the largest gap being almost 4 times the averaged embedding shifts of existing methods (SIP).

Our theoretical analysis and experimental results demonstrate that the existing method leads to a significantly decreased perturbation-caused embedding shift compared to our TRIDE (CAP+ANP) setting. Consequently, the perturbation generated likewise is much weaker than the perturbation optimized using TRIDE.

Another reason for TRIDE is that simultaneous perturbation is not a practical method considering many existing attacks against DML. In black-box scenarios, most existing attacks are achieved by ANP(Li et al., 2021; Chen et al., 2021; Zhou et al., 2021a; Li et al., 2019), which is more practical and effective.

B. Calculations Details of Adversarial Resistance Scores

Ranking attacks. Ranking attacks intend to manipulate the rank of candidates through adversarial perturbations. Given a ranking attack $\mathscr{A}_{rank}$, for the ith candidate, we denote its initial rank r for the jth query as $r_{i,j}$, and its after-attack rank as $\tilde{r}_{i,j}$. Taking CA+ as an example, the goal of which is to elevate $r_{i,j}$ as much as possible, *i.e.*, $r = 0$. In other words, for this trial of CA+ attack, $\mathscr{O}_g = 0$, $\mathscr{O}_i = r_{ij}$ and $\mathscr{O}_r = \tilde{r}_{i,j}$. Hence, $\mathbb{R}_{\mathbf{M},CA_+}$ can be calculated as follows:

$$\begin{aligned} \mathbb{R}_{\mathbf{M},CA_+} &= (1 - \frac{\mathscr{O}_r - \mathscr{O}_i}{\mathscr{O}_g - \mathscr{O}_i}) \times 100\%, \\ &= \big(1 - \frac{|r_{i,j} - \tilde{r}_{i,j}|}{r_{i,j}}\big) \times 100\% \end{aligned} \tag{27}$$

$$\tag{28}$$

Similarly, AR all rank attacks (CA+, CA-, QA+, QA-) can be calculated likewise:

$$\mathbb{R}_{\mathbf{M},(CA_+,CA_-,QA_+,QA_-)} = \big(1 - \frac{|r_{i,j} - \tilde{r}_{i,j}|}{r_{i,j}}\big) \times 100\% \tag{29}$$

Note that the final ARS of a ranking attack is calculated by the average of $\mathbf{N}$ selected candidates (CA) or queries (QA), calculated as follows:

$$\mathbb{R}_{\mathbf{M},\mathcal{A}_{rank}} = \frac{1}{N}\sum_{i,j}^{\mathbf{N}} \big(1 - \frac{|r_{i,j} - \tilde{r}_{i,j}|}{r_{i,j}}\big) \times 100\% \tag{30}$$

Details of CA and QA attacks can be found in (Zhou et al., 2021b).

Recall attacks. The evaluation of Recall attacks is much simpler as these attacks seek to lower the R@1. Thus, given a model $\mathbf{M}$ with initial R@1 $\mu_{\mathbf{M}}$ and its after-attack R@1 $\tilde{\mu}_{\mathbf{M}}$, all Rrecall attacks' intention is to lower $\mu_{\mathbf{M}}$ to 0, *i.e.*, $\mathbb{O}_g = \mu_{\mathbf{M}}$, $\mathbb{O}_a = \tilde{\mu}_{\mathbf{M}}$. Hence, $\mathbb{R}_{\mathbf{M},(ES,LTM,GTM)}$ can be calculated as follows:

$$\begin{aligned}\mathbb{R}_{\mathbf{M},(ES,LTM,GTM)} &= \big(1 - \frac{\mu_{\mathbf{M}} - \tilde{\mu}_{\mathbf{M}}}{\mu_{\mathbf{M}}}\big) \times 100\% \\ &= \frac{\tilde{\mu}_{\mathbf{M}}}{\mu_{\mathbf{M}}} \times 100\% \end{aligned} \tag{31}$$

In essence, our proposed ARS quantifies the robustness of a model based on how well the attack achieves its goal on this model. Initial state variation is eliminated by calculating the difference instead of evaluating pure results.

C. Implementation Details

To help the model balance between learning meaningful information and acquiring robustness, we apply a simple epoch-wise adjustment strategy to mini-batch sampling and perturbation adding.

For mini-batch sampling, we follow the semi-hard sampling (Schroff et al., 2015), which samples hard examples within a given range $d(a,p) < d(a,n) < d(a,p) + \eta$, and apply an epoch-wise strategy to η:

$$\eta = \eta_0(1 - (\frac{n}{2 \times n_{total}})^2) \tag{32}$$

where η_0 is the preset margin of semi-hard sampling, n stands for the current number of epochs, and n_{total} stands for the total number of epochs.

For progressive step size, we similarly multiply the PGD step size α by $\frac{n}{n_{total}}$ to enable a gradually increasing adversary strength.