Supplementary Material

In this supplementary material, we provide more technical details and additional discussions that are excluded from the manuscript due to space limit.

Contents

A. Additional Information

A.1. Potential Societal Impact

I. Security.

Adversarial defenses alleviate the negative societal impact of adversarial attacks, and hence have positive societal impact.

A.2. Limitations of Our Method

I. Assumptions.

(1) Triplet Training Assumption.

Our method assumes sample triplets are used for training. Our method may not be compatible to other non-triplet DML loss functions. Adversarial training with other DML loss functions is left for future study.

(2) Embedding Space Assumption.

We follow the common setups [6, 10] on the embedding space. Namely, (1) the embedding vectors are normalized onto the real unit hypersphere; (2) the distance function $d(\cdot, \cdot)$ is Euclidean distance. Our formulations are developed upon the two assumptions. It is unknown whether our method method will be effective when embedding vectors are *not* normalized. And it is unknown whether our method will be effective when $d(\cdot, \cdot)$ is replaced as other distance metrics, *e.g.*, cosine distance.

(3) Optimizer Assumption.

Our method assumes PGD [2] is used for optimizing the HM objective to create adversarial examples. The Eq. (4)-(5) may not necessarily hold with other possible optimizers.

II. Performance-Sensitive Factors.

(1) Maximum number of PGD iterations η.

Our method's sensitivity to η has been demonstrated by Tab. 3-6 and Fig. 6-8. A larger η indicates higher training cost, and stronger adversarial examples are created for adversarial training. As a result, a larger η leads to a higher robustness (ERS) and a lower R@1 performance. Our method consistently achieves a higher ERS under different η settings compared to previous methods, and hence are the most efficient defense method. Experiments with η larger than 32 are not necessary because ERS plateaus according to Fig. 6-8.

(2) Source hardness H_S and destination hardness H_D.

The H_S and H_D are the only two adjustable items in HM.

The source hardness H_S depends on triplet sampling strategy. We conduct experiments with existing triplet sampling strategies in order to focus on defense.

The choices for H_D are more flexible than those of H_S, as discussed in Sec. 3.1. In the experiments, we study some possible choices following the discussion and design LGA based on the empirical observations.

(3) Parameters involved in g_LGA.

A constant u is used to normalize the loss value of the previous training iteration ℓ_{t-1} into $\bar{\ell}_{t-1} \in [0, 1]$. The constant is empirically selected as $u = \gamma$ in our experiment. According to our observation, the loss value will quickly decrease below γ, and will remain in the $[0, \gamma]$ range for the whole training process. If we set u to a larger constant than γ, the normalized loss $\bar{\ell}_{t-1}$ will be smaller, and results in stronger adversarial examples through HM[S, g_LGA] and harms the model performance on benign examples, as shown in Tab. 7.

Another parameter in g_LGA is the triplet margin parameter γ in order to align to the hardness range of Semihard triplets, *i.e.*, $-\gamma < g_\mathsf{LGA} < 0$. We follow the common setup [6] for this parameter.

(4) Constant parameter λ for ICS loss term L_ICS.

The weight constant λ is set as 0.5 by default, and 0.05 on the SOP dataset. As demonstrated in Tab. 5, there is a trade-off between robustness and performance on benign examples when tuning the λ parameter.

Additional experiments with $\lambda = 0.5$ on the SOP dataset can be found in Tab. 8. Our method is sensitive to this parameter. An excessively large λ on the SOP datasets leads to worse performance on benign examples and worse robustness.

(5) Backbone deep neural network.

We adopt ResNet-18 following the state-of-the-art de-

Dataset	Defense	η	Benign Example				White-Box Attacks for Robustness Evaluation										ERS↑
			R@1↑	R@2↑	mAP↑	NMI↑	CA+↑	CA-↓	QA+↑	QA-↓	TMA↓	ES:D↓	ES:R↑	LTM↑	GTM↑	GTT↑	
CUB	HM[$\mathcal{S}, g_{\mathsf{LGA}}$]	8	38.0	48.3	21.8	49.3	12.7	46.4	11.6	39.9	0.567	0.783	16.8	11.9	27.9	1.4	32.4
	HM[$\mathcal{S}, g_{\mathsf{LGA}}$] ($u = 2.2$)	8	34.8	45.5	15.2	47.1	13.4	36.0	17.2	26.1	0.934	0.244	20.1	15.9	27.3	3.8	36.1

Table 7. The efficacy of parameter u for clipping loss value ℓ_{t-1}. Stronger adversarial examples will be created for training if the loss value is not clipped (equivalent to setting u to the theoretical upper bound of loss, *i.e.*, 2.2).

Dataset	Defense	η	Benign Example				White-Box Attacks for Robustness Evaluation										ERS↑
			R@1↑	R@2↑	mAP↑	NMI↑	CA+↑	CA-↓	QA+↑	QA-↓	TMA↓	ES:D↓	ES:R↑	LTM↑	GTM↑	GTT↑	
SOP	HM[$\mathcal{S}, g_{\mathsf{LGA}}$]&ICS ($\lambda = 0.5$)	8	42.4	47.1	10.2	84.2	34.9	3.8	36.7	2.3	0.879	0.093	35.3	36.5	35.2	49.1	60.1
	HM[$\mathcal{S}, g_{\mathsf{LGA}}$]&ICS ($\lambda = 0.5$)	32	41.5	46.1	9.9	84.1	36.1	3.1	37.6	2.1	0.873	0.086	35.7	36.8	34.7	50.2	60.8

Table 8. An excessively large λ may lead to worse performance and robustness.

fense [10] for fair comparison. Since our proposed method is independent to the backbone choice, and hence is expected to be effective with different backbone models.

A.3. Use of Existing Assets

(1) Datasets.

All the datasets used in our paper are public datasets, and their corresponding papers are cited. The CUB [8] dataset includes images of birds. The CARS [1] dataset includes images of cars. The SOP [3] dataset includes images of online products.

(2) Code and Implementation.

Our implementation is built upon PyTorch and the public code of the state-of-the-art defense method ACT [10] (License: Apache-2.0).

B. Technical Details & Minor Discussions

B.1. Difference between Existing Defenses & HM

I. Embedding-Shifted Triplet (EST). [9]

Embedding-Shifted Triplet (EST) [9] adopts adversarial counterparts of $\boldsymbol{a}, \boldsymbol{p}, \boldsymbol{n}$ with maximum embedding move distance off their original locations, *i.e.*,

$$L_{\mathsf{EST}} = L_{\mathrm{T}}(\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{p}}, \tilde{\boldsymbol{n}}; \gamma) \tag{1}$$

where $\tilde{\boldsymbol{a}} = \phi(\boldsymbol{A} + \boldsymbol{r}^*)$, and $\boldsymbol{r}^* = \arg\max_{\boldsymbol{r}} d_\phi(\boldsymbol{A} + \boldsymbol{r}, \boldsymbol{A})$. The $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{n}}$ are obtained similarly.

(1) Relationship with HM:

Since EST only aims to maximize the embedding move distance off its original location without specifying any direction, it leads to a random hardness value. The expectation $E[H(\cdot)]$ of its resulting adversarial triplet is expected to be close to $E[H_{\mathsf{S}}]$. Because the perturbed triplet can be either harder or easier than the benign triplet. Namely, EST merely indirectly increase the hardness of the training triplet, and may even decrease its hardness. Thus, EST suffers from inefficiency in adversarial training compared to HM.

II. Anti-Collapse Triplet (ACT). [10]

Anti-Collapse Triplet (ACT) [10] "collapses" the embedding vectors of positive and negative sample, and enforces the model to separate them apart, *i.e.*,

$$L_{\mathsf{ACT}} = L_{\mathrm{T}}(\boldsymbol{a}, \overrightarrow{\boldsymbol{p}}, \overleftarrow{\boldsymbol{n}}; \gamma), \tag{2}$$

$$[\overrightarrow{\boldsymbol{p}}, \overleftarrow{\boldsymbol{n}}] = [\phi(\boldsymbol{P} + \boldsymbol{r}_p^*), \phi(\boldsymbol{N} + \boldsymbol{r}_n^*)] \tag{3}$$

$$[\boldsymbol{r}_p^*, \boldsymbol{r}_n^*] = \underset{\boldsymbol{r}_p, \boldsymbol{r}_n}{\arg\min}\, d_\phi(\boldsymbol{P} + \boldsymbol{r}_p, \boldsymbol{N} + \boldsymbol{r}_n). \tag{4}$$

(1) Relationship with HM:

When ACT successively "collapses" the positive and negative embedding vectors together, the hardness will be zero, *i.e.*, $E[H(\cdot)] = 0$. But ACT is not equivalent to $HM[\cdot, 0]$ because the two methods have different objectives and use different gradients. Besides, in order to avoid the "misleading gradients" [10], ACT fixes the anchor and only perturb the positive and negative samples, which makes the objective for creating adversarial examples more difficult to optimize in practice. In brief, ACT is also indirectly increasing the loss value, suffering from inefficient adversarial learning.

III. Min-max Adversarial Training with Triplet Loss.

The direct formulation of min-max adversarial training [2] for triplet loss-based DML is:

$$\theta^* = \underset{\theta}{\arg\min}[\underset{\boldsymbol{r}_a, \boldsymbol{r}_p, \boldsymbol{r}_n}{\arg\max} L_{\mathrm{T}}(\boldsymbol{A} + \boldsymbol{r}_a, \boldsymbol{P} + \boldsymbol{r}_p, \boldsymbol{N} + \boldsymbol{r}_n)] \tag{5}$$

Previous works [9, 10] point out this method will easily lead to model collapse. Our observation suggests the same.

(1) Relationship with HM:

Maximizing L_{T} is equivalent to maximizing $\tilde{H}_{\mathsf{S}}$. This can be expressed as HM[H_{S}, 2] as discussed in Sec. 3.1.

B.2. Hardness Manipulation (HM)

I. Adjustable Items

The only two adjustable items in HM are H_{S} and H_{D}. They are discussed in the "Performance-Sensitive Factors" part of the previous section.

Dataset	**Defense**	η	**Benign Example**				**White-Box Attacks for Robustness Evaluation**										**ERS↑**
			R@1↑	R@2↑	mAP↑	NMI↑	CA+↑	CA-↓	QA+↑	QA-↓	TMA↓	ES:D↓	ES:R↑	LTM↑	GTM↑	GTT↑	
CUB	HM[$\mathcal{S}, g_{1/2}$]	8	38.7	48.5	22.0	49.2	12.6	48.6	12.3	41.3	0.562	0.825	13.5	12.9	26.9	1.8	31.7
	HM[$\mathcal{S}, g_{\mathsf{LGA}}$]	8	38.0	48.3	21.8	49.3	12.7	46.4	11.6	39.9	0.567	0.783	16.8	11.9	27.9	1.4	32.4
	HM[$\mathcal{S}, g_2$]	8	37.4	48.2	17.1	49.2	12.9	45.1	13.2	39.1	0.599	0.738	17.7	12.8	27.5	1.9	33.1

Table 9. Non-linear Gradual Adversary Examples.

II. Extreme Values of Hardness.

As reflected in Sec. 3.1 and Fig. 2, the range of $H(\cdot)$ is $[-2, 2]$. The embedding vectors have been normalized to the real unit hypersphere as pointed out in the manuscript. And the range of distance between any two points on the hypersphere is $[0, 2]$. Hence the extreme values are:

$$\begin{aligned} &\max H(\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{N}) && (6)\\ =&\max[d_\phi(\boldsymbol{A}, \boldsymbol{P}) - d_\phi(\boldsymbol{A}, \boldsymbol{N})] && (7)\\ =&\max[d_\phi(\boldsymbol{A}, \boldsymbol{P})] - \min[d_\phi(\boldsymbol{A}, \boldsymbol{N})] && (8)\\ =&2 - 0, && (9)\end{aligned}$$

$$\begin{aligned} &\min H(\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{N}) && (10)\\ =&\min[d_\phi(\boldsymbol{A}, \boldsymbol{P}) - d_\phi(\boldsymbol{A}, \boldsymbol{N})] && (11)\\ =&\min[d_\phi(\boldsymbol{A}, \boldsymbol{P})] - \max[d_\phi(\boldsymbol{A}, \boldsymbol{N})] && (12)\\ =&0 - 2. && (13)\end{aligned}$$

Namely $H(\cdot) \in [-2, 2]$. Meanwhile, since

$$L_\text{T}(\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{N}; \gamma) = \max(0, H(\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{N}) + \gamma), \quad (14)$$

we have $L_\text{T} \in [0, 2 + \gamma]$.

B.3. Gradual Adversary

I. Parameters

The parameters, namely u and γ are discussed in the previous section of this supplementary material, see "Performance-Sensitive Factors".

The parameter ξ in g_B is set as 0.1, but it is not an important parameter. The function g_B is only used for demonstrating that "slightly boosting the destination hardness can further increase ERS" as discussed in Sec. 3.1.

II. Non-linear Gradual Adversary

In Sec. 3.2, more complicated designs are left for future work. We provide two Non-linear Gradual Adversary examples, namely g_2 and $g_{1/2}$, as follows:

$$\begin{aligned} g_2(\cdot) &= -\gamma \cdot (\bar{\ell}_{t-1})^2 && \in [-\gamma, 0] && (15)\\ g_{1/2}(\cdot) &= -\gamma \cdot (\bar{\ell}_{t-1})^{1/2} && \in [-\gamma, 0] && (16)\end{aligned}$$

Compared to LGA, g_2 is more "eager" to result in strong adversarial examples in the early phase of training, while $g_{1/2}$ is more "conservative" in creating strong adversarial examples in the early phase of training (adversarial examples from HM are stronger if the function value is closer to 0). The corresponding experiments can be found in Tab. 9.

B.4. Intra-Class Structure (ICS)

I. Parameters

The ICS loss term can be appended to the loss for adversarial training. The only parameter for ICS loss term is the weight constant λ, and has been discussed in the "Performance-Sensitive Factors" part of the previous section.

The margin parameter is set to 0 in order to avoid negative effect in L_ICS, as shown in Tab. 10.

II. Gradients in Fig. 5 (a) in Manuscript

According to [10], when the embedding vectors are normalized onto the real unit hypersphere and Euclidean distance is used, the gradients of the triplet loss with respect to the anchor, positive, and negative embedding vectors are respectively:

$$\begin{aligned} \frac{\partial L_\text{T}}{\partial \boldsymbol{a}} &= \frac{\boldsymbol{a}-\boldsymbol{p}}{\|\boldsymbol{a}-\boldsymbol{p}\|} - \frac{\boldsymbol{a}-\boldsymbol{n}}{\|\boldsymbol{a}-\boldsymbol{n}\|} && (17)\\ \frac{\partial L_\text{T}}{\partial \boldsymbol{p}} &= \frac{\boldsymbol{p}-\boldsymbol{a}}{\|\boldsymbol{a}-\boldsymbol{p}\|} && (18)\\ \frac{\partial L_\text{T}}{\partial \boldsymbol{n}} &= \frac{\boldsymbol{a}-\boldsymbol{n}}{\|\boldsymbol{a}-\boldsymbol{n}\|}, && (19)\end{aligned}$$

when $L_\text{T} > 0$. And the above equations have been reflected in Fig. 5 (a) in terms of vector direction.

III. Alternative Design for Exploiting Sextuplet

Let $\tilde{\boldsymbol{a}}$, $\tilde{\boldsymbol{p}}$, and $\tilde{\boldsymbol{n}}$ be $\phi(\boldsymbol{A}+\hat{\boldsymbol{r}}_a)$, $\phi(\boldsymbol{P}+\hat{\boldsymbol{r}}_p)$ and $\phi(\boldsymbol{N}+\hat{\boldsymbol{r}}_n)$ respectively. In our proposed ICS loss term, only $(\boldsymbol{a}, \tilde{\boldsymbol{a}}, \boldsymbol{p})$ are involved. Other alternative selections of triplets from the sextuplet are possible, but are not as effective as L_ICS for improving adversarial robustness.

(1) $L_\text{T}(\boldsymbol{a}, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{p}})$

As shown in Fig. 5 (c) in the manuscript, the position of $\tilde{\boldsymbol{p}}$ is always further away from both $\boldsymbol{a}$ and $\tilde{\boldsymbol{a}}$ than $\boldsymbol{p}$ due to the gradient direction. Thus, the loss value of $L_\text{T}(\boldsymbol{a}, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{p}})$ will always be smaller than L_ICS, and hence is less effective than L_ICS.

(2) $L_\text{T}(\boldsymbol{a}, \boldsymbol{p}, \boldsymbol{n})$: Mixing regular training and adversarial training.

Dataset	Defense	η	Benign Example				White-Box Attacks for Robustness Evaluation										ERS↑
			R@1↑	R@2↑	mAP↑	NMI↑	CA+↑	CA-↓	QA+↑	QA-↓	TMA↓	ES:D↓	ES:R↑	LTM↑	GTM↑	GTT↑	
CUB	HM[$\mathcal{S}, g_{\text{LGA}}$]&ICS	8	37.2	47.8	21.4	48.4	12.9	40.9	14.7	33.7	0.806	0.487	17.1	13.2	26.3	2.3	33.5
	HM[$\mathcal{S}, g_{\text{LGA}}$]&$L_{\text{ICS}}(\cdots; \gamma = 0.2)$	8	35.8	46.6	16.4	48.3	13.2	39.9	12.7	33.3	0.775	0.507	15.9	14.9	27.2	2.7	33.6

Table 10. The margin parameter is set to 0 in order to avoid negative effect. R@1 performance drops with marginal robustness gain.

According to our observation, mixing regular training and adversarial training leads to better R@1 with drastic robustness degradation for both ACT and our defense.

(3) $L_{\text{T}}(\boldsymbol{p}, \tilde{\boldsymbol{p}}, \boldsymbol{a})$: Symmetric counterpart of L_{ICS}.

Every sample in the training dataset will be used as anchor for once per epoch. Such symmetric loss term is duplicated to L_{ICS} and is not necessary. Experimental results suggest negligible difference compared to L_{ICS}.

(4) $L_{\text{T}}(\boldsymbol{a}, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{n}})$: $\tilde{\boldsymbol{n}}$ is very close to $\boldsymbol{a}$ in Fig. 5.

It enforces *inter*-class structure instead of intra-class structure. Besides, experimental results suggest negligible difference. We speculate this loss term is duplicated to the adversarial training loss term, *i.e.*, $L_{\text{T}}(\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{p}}, \tilde{\boldsymbol{n}})$, which enforces inter-class structure as well in a stronger manner.

(5) $L_{\text{T}}(\boldsymbol{a}, \boldsymbol{p}, \tilde{\boldsymbol{n}})$

According to our observation, it leads to better R@1 performance on benign examples, but drastically reduce the robustness.

B.5. Experiments and Evaluation

I Hardware and Software Configuration

We conduct the experiments with two Nvidia Titan Xp GPUs (12GB of memory each) under the Ubuntu 16.04 operating system with PyTorch 1.8.2 in distributed data parallel.

II. Training Cost

In the manuscript, the training cost of adversarial training is caluclated as $\eta + 1$, which is the number of forward-backward propagation involved in each iteration of the training process, in order to reflect the training efficiency (gain as high robustness as possible given a fixed number of forward-backward calculation for adversarial training) of different defense methods. Specifically, PGD creates adversarial examples from the η times of forward-backward computation. With the resulting adversarial examples, the network requires once more forward-backward computation to update the model parameters.

III. Complete Results for Tab. 2 in Manuscript

Complete experimental results for "Tab. 2: Combinations of Source & Destination Hardness. ..." can be found in Tab. 11 of this supplementary material.

IV. Additional Notes on the Experimental Results

(1) Slight ERS decrease with a larger η

In some cases, *e.g.*, HM[$\mathcal{S}, g_{\text{LGA}}$]&ICS on the CARS dataset reaches a slightly lower ERS with $\eta = 32$ compared to that with $\eta = 8$. We speculate this is because adversarial training suffers from overfitting [4, 5] on adversarial examples, as mentioned in Sec. 2.

B.6. Potential Future Work

I. Faster Adversarial Training with HM

As discussed in Sec. 3.1, one potential adversarial training acceleration method is to incorporate Free Adversarial Training (FAT) [7] (originally for classification) into our DML adversarial training with HM. Besides, directly incorporating FAT into the min-max adversarial training of DML will easily result in model collapse as well, because the FAT algorithm can be interpreted as to maintain a universal (agnostic to sample) perturbation that can maximize the loss. Thus, non-trivial modifications are still required to incorporate FAT FAT into adversarial training with HM. This is left for future work.

II. Better Choice for Destination Hardness

The proposed LGA function incorporates our empirical observation that "adversarial triplets should remain Semihard" based on the results in Tab. 2. However, a better choice for H_{D} may exist between "Semihard" and "Softhard" that can achieve better overall performance. In the manuscript, we only use the existing sampling methods and simple pseudo-hardness functions in order to avoid distraction from our focus.

III. Other Loss Functions

Adversarial trainig with DML loss functions other than triplet loss is insufficiently explored. New metric learning loss functions oriented for adversarial training are also left for future study. Besides, model collapse is an inevitable problem for adversarial training with triplet loss, and it is unknown whether other loss functions could mitigate this issue.

IV. DML & Classification

It is unknown whether DML defenses will improve the robustness in the classification task.

References

[1] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proc.*

Dataset	Defense	η	Benign Example				White-Box Attacks for Robustness Evaluation										ERS
			R@1↑	R@2↑	mAP↑	NMI↑	CA+↑	CA-↓	QA+↑	QA-↓	TMA↓	ES:D↓	ES:R↑	LTM↑	GTM↑	GTT↑	
	HM$[\mathcal{R},\mathcal{R}]$	N/A	53.9	66.4	26.1	59.5	0.0	100.0	0.0	99.9	0.883	1.762	0.0	0.0	14.1	0.0	3.8
	HM$[\mathcal{R},\mathcal{M}]$	8	27.0	36.0	13.2	42.5	19.4	48.0	22.2	32.0	0.535	0.867	11.6	10.4	19.3	2.9	35.1
CUB	HM$[\mathcal{R},\mathcal{S}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{R},\mathcal{D}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{R},\mathcal{H}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{M},\mathcal{R}]$	8	43.9	55.9	27.1	54.2	0.2	100.0	0.4	99.6	0.849	1.640	0.2	0.1	19.1	0.0	5.4
	HM$[\mathcal{M},\mathcal{M}]$	N/A	44.0	56.1	27.2	54.6	0.1	100.0	0.4	99.8	0.843	1.717	0.3	0.3	18.4	0.0	5.0
CUB	HM$[\mathcal{M},\mathcal{S}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{M},\mathcal{D}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{M},\mathcal{H}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{S},\mathcal{R}]$	8	48.3	60.1	30.3	56.3	2.0	84.4	1.4	85.4	0.835	0.895	6.1	1.6	21.0	0.0	13.7
	HM$[\mathcal{S},\mathcal{M}]$	8	38.4	49.7	22.9	50.3	10.9	50.5	10.8	44.6	0.680	0.722	13.3	11.2	25.8	1.2	29.6
CUB	HM$[\mathcal{S},\mathcal{S}]$	N/A	55.7	68.2	35.8	61.0	0.0	100.0	0.0	99.6	0.942	1.272	0.0	0.0	19.2	0.0	6.2
	HM$[\mathcal{S},\mathcal{D}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{S},\mathcal{H}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{D},\mathcal{R}]$	8	52.7	64.0	33.7	58.9	0.0	100.0	0.2	100.0	0.841	1.749	0.1	0.0	19.0	0.0	4.8
	HM$[\mathcal{D},\mathcal{M}]$	8	50.7	63.8	33.1	58.7	0.0	100.0	0.3	99.9	0.841	1.744	0.2	0.0	18.5	0.0	4.8
CUB	HM$[\mathcal{D},\mathcal{S}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{D},\mathcal{D}]$	N/A	51.4	64.2	33.4	59.3	0.0	100.0	0.4	99.9	0.842	1.753	0.0	0.0	19.7	0.0	4.9
	HM$[\mathcal{D},\mathcal{H}]$	8	54.7	66.9	36.2	60.0	0.0	100.0	0.2	99.8	0.874	1.569	0.4	0.0	18.6	0.0	5.4
	HM$[\mathcal{H},\mathcal{R}]$	8	51.0	63.5	32.9	59.2	0.0	100.0	0.2	99.9	0.837	1.750	0.0	0.0	17.7	0.0	4.7
	HM$[\mathcal{H},\mathcal{M}]$	8	52.2	64.0	33.1	58.4	0.0	100.0	0.2	100.0	0.841	1.750	0.0	0.0	18.9	0.0	4.8
CUB	HM$[\mathcal{H},\mathcal{S}]$	8	C	o	l	l	a	p	s	e							
	HM$[\mathcal{H},\mathcal{D}]$	8	52.6	64.9	34.4	60.0	0.0	100.0	0.2	99.9	0.835	1.720	0.1	0.0	20.0	0.0	5.1
	HM$[\mathcal{H},\mathcal{H}]$	N/A	48.9	61.2	30.3	56.5	0.0	100.0	0.4	99.9	0.830	1.752	0.2	0.0	19.5	0.0	5.0

Table 11. Full data sheet for Tab.2 in the Paper. The symbols $\mathcal{R},\mathcal{M},\mathcal{S},\mathcal{D},\mathcal{H}$ denote Random, Semihard, Softhard, Distance and Hardest triplet sampling strategies, respectively. Experiments that end up with model collapse are left with a placeholder in the table.

Int. Conf. Comput. Vis. Workshops, pages 554–561, 2013. 2

[2] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. Int. Conf. Learn. Representations*, 2018. 1, 2

[3] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4004–4012, 2016. 2

[4] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *Proc. Int. Conf. Learn. Representations*, 2021. 4

[5] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *Proc. Int. Conf. Mach. Learn.*, pages 8093–8104, 2020. 4

[6] 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 *Proc. Int. Conf. Mach. Learn.*, pages 8242–8252, 2020. 1

[7] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Proc. Conf. Neural Inf. Process. Syst.*, 2019. 4

[8] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 2

[9] Mo Zhou, Zhenxing Niu, Le Wang, Qilin Zhang, and Gang Hua. Adversarial ranking attack and defense. In *Proc. Eur. Conf. Comput. Vis.*, pages 781–799, 2020. 2

[10] Mo Zhou, Le Wang, Zhenxing Niu, Qilin Zhang, Nanning Zheng, and Gang Hua. Adversarial attack and defense in deep ranking. In *arXiv preprint 2106.03614*, 2021. 1, 2, 3