A. Proofs of Theorems

A.1. Proof of Theorem 1

Proof. From Eq. (7) and Eq. (8), for $\forall\delta \in \{\delta \mid \delta \in X, \|\delta\|_\infty \leq \epsilon\}$, the following holds:

$$\sum_{c\in C} \mathbb{1}\Big[\|f(q+\delta) - f(\hat{c})\|_2 < \|f(q+\delta) - f(c)\|_2\Big] \geq \sum_{c\in C} \mathbb{1}\Big[\overline{d}_{\hat{c}}(q) < \underline{d}_c(q)\Big], \quad (26)$$

$$\sum_{c\in C} \mathbb{1}\Big[\|f(q+\delta) - f(c)\|_2 < \|f(q+\delta) - f(\hat{c})\|_2\Big] \geq \sum_{c\in C} \mathbb{1}\Big[\overline{d}_c(q) < \underline{d}_{\hat{c}}(q)\Big] \quad (27)$$

where $\hat{c} = \mathrm{IR}_f(q, C)_j$. Since Eq. (26) indicates that the lower bound of the number of candidate images that are more dissimilar to $q+\delta$ than $\hat{c}$ is $\sum_{c\in C} \mathbb{1}\Big[\overline{d}_{\hat{c}}(q) < \underline{d}_c(q)\Big]$, we obtain Eq. (9). Since Eq. (27) indicates that the lower bound of the number of candidate images that are more similar to $q+\delta$ than $\hat{c}$ is $\sum_{c\in C} \mathbb{1}\Big[\overline{d}_c(q) < \underline{d}_{\hat{c}}(q)\Big]$, we obtain Eq. (10). □

A.2. Proof of Theorem 2

Proof. From Eq. (7) and Eq. (8), for $\tilde{C} = \{\mathrm{IR}_f(q, C)_i + \delta_i\}_{i=1}^N$ where $\forall\delta_1, ..., \forall\delta_N \in \{\delta \mid \delta \in X, \|\delta\|_\infty \leq \epsilon\}$, the following holds:

$$\sum_{c+\delta\in\tilde{C}} \mathbb{1}\Big[\|f(q) - f(\hat{c}+\delta_j)\|_2 < \|f(q) - f(c+\delta)\|_2\Big] \geq \sum_{c\in C} \mathbb{1}\Big[\overline{d}_q(\hat{c}) < \underline{d}_q(c)\Big], \quad (28)$$

$$\sum_{c+\delta\in\hat{C}} \mathbb{1}\Big[\|f(q) - f(c+\delta)\|_2 < \|f(q) - f(\hat{c}+\delta_j)\|_2\Big] \geq \sum_{c\in C} \mathbb{1}\Big[\overline{d}_q(c) < \underline{d}_q(\hat{c})\Big] \quad (29)$$

where $\hat{c} = \mathrm{IR}_f(q, C)_j$. Since Eq. (28) indicates that the lower bound of the number of perturbed candidate images in $\tilde{C}$ that are more dissimilar to q than $\hat{c}+\delta_j$ is $\sum_{c\in\hat{C}} \mathbb{1}\Big[\overline{d}_q(\hat{c}) < \underline{d}_q(c)\Big]$, we obtain Eq. (11). Since Eq. (29) indicates that the lower bound of the number of perturbed candidate images in $\tilde{C}$ that are more similar to q than $\hat{c}+\delta_j$ is $\sum_{c\in\hat{C}} \mathbb{1}\Big[\overline{d}_q(c) < \underline{d}_q(\hat{c})\Big]$, we obtain Eq. (12). □

A.3. Proof of Theorem 3

Proof. When Eq. (13) is satisfied,

$$-\alpha \leq \mathrm{Rank}_f(q+\delta, \mathrm{IR}_f(q, C)_j, C) - j \leq \alpha$$

is satisfied for $\forall\delta \in \{\delta \mid \delta \in X, \|\delta\|_\infty \leq \epsilon\}$ from Theorem 1, which is equivalent to Eq. (5). □

A.4. Proof of Theorem 4

Proof. When Eq. (14) is satisfied,

$$-\alpha \leq \mathrm{Rank}_f(q, \mathrm{IR}_f(q, C)_j + \delta_j, \tilde{C}) - j \leq \alpha$$

is satisfied for $\tilde{C} = \{\mathrm{IR}_f(q, C)_i + \delta_i\}_{i=1}^N$ where $\forall\delta_1, ..., \forall\delta_N \in \{\delta \mid \delta \in X, \|\delta\|_\infty \leq \epsilon\}$ from Theorem 2, which is equivalent to Eq. (6). □

A.5. Proof of Theorem 5

Proof. Since $\underline{f}(x_1)_i \leq f(x_1+\delta)_i \leq \overline{f}(x_1)_i$ holds for $\forall\delta \in \{\delta \mid \delta \in X, \|\delta\|_\infty \leq \epsilon\}$ from the definitions of $\underline{f}(x)_i$ and $\overline{f}(x)_i$, the following holds:

$$\max_{\delta\in X, \|\delta\|_\infty\leq\epsilon} (f(x_1+\delta)_i - f(x_2)_i)^2 \leq \max\big\{|\overline{f}(x_1)_i - f(x_2)_i|, |f(x_2)_i - \underline{f}(x_1)_i|\big\}^2.$$

Then, we obtain

$$\max_{\delta\in X, \|\delta\|_\infty\leq\epsilon} \|f(x_1+\delta) - f(x_2)\|_2 = \max_{\delta\in X, \|\delta\|_\infty\leq\epsilon} \Big(\sum_{i\in\{1,..,d\}} \big(f(x_1+\delta)_i - f(x_2)_i\big)^2\Big)^{\frac{1}{2}} \leq \Big(\sum_{i\in\{1,..,d\}} \max\{|\overline{f}(x_1)_i - f(x_2)_i|, |f(x_2)_i - \underline{f}(x_1)_i|\}^2\Big)^{\frac{1}{2}}.$$

□

A.6. Proof of Theorem 6

Proof. Since $\underline{f}(x_1)_i \leq f(x_1+\delta)_i \leq \overline{f}(x_1)_i$ holds for $\forall\delta \in \{\delta \mid \delta \in X, \|\delta\|_\infty \leq \epsilon\}$ from the definitions of $\underline{f}(x)_i$ and $\overline{f}(x)_i$, the following holds:

$$\min_{\delta\in X, \|\delta\|_\infty\leq\epsilon} (f(x_1+\delta)_i - f(x_2)_i)^2 \geq \min\big\{0, \overline{f}(x_1)_i - f(x_2)_i, f(x_2)_i - \underline{f}(x_1)_i\big\}^2$$

Then, we obtain

$$\min_{\delta\in X, \|\delta\|_\infty\leq\epsilon} \|f(x_1+\delta) - f(x_2)\|_2 = \min_{\delta\in X, \|\delta\|_\infty\leq\epsilon} \Big(\sum_{i\in\{1,..,d\}} \big(f(x_1+\delta)_i - f(x_2)_i\big)^2\Big)^{\frac{1}{2}} \geq \Big(\sum_{i\in\{1,..,d\}} \min\big\{0, \overline{f}(x_1)_i - f(x_2)_i, f(x_2)_i - \underline{f}(x_1)_i\big\}^2\Big)^{\frac{1}{2}}.$$

□

B. Comparison Methods

We compare our proposed robustness training Eq. (20) (TBT) and Eq. (21) (FCTB) with three existing methods: (i) triplet Loss (Triplet) [6], (ii) anti-collapse triplet (ACT), which is an adversarial training for CBIR to improve empirical robustness [9], (iii) training for classification using interval bound propagation (C-IBP) to improve certified robustness for the classification task [2].

Triplet is one of the loss functions commonly used in metric learning. Let $D_t = \{(a, p, n)_i\}_{i=1}^M$ be a training data set where p belongs to the same class as a, and n belongs to a different class than a. Then, Triplet trains the feature extraction DNN f by minimizing the following loss function:

$$\sum_{(a,p,n)\in D_t} \max\{\|f(a)-f(p)\|_2 - \|f(a)-f(n)\|_2 + m, 0\}, \tag{30}$$

where m is a margin parameter.

ACT trains feature extraction DNN f on generated adversarial examples. Let $D_t = \{(a, p, n)_i\}_{i=1}^M$ be a training data set. For each triplet $(a, p, n) \in D_t$, ACT generate $p + \delta_p$ and $n + \delta_n$ so that the distance $\|f(p+\delta_p) - f(n+\delta_n)\|_2$ is small. Specifically, ACT minimize triplet loss with the triplet $(a, p + \delta_p, n + \delta_n)$ as follows:

$$\sum_{(a,p,n)\in D_t} \max\{\|f(a) - f(p+\delta_p)\|_2 - \|f(a) - f(n+\delta_n)\|_2 + m, 0\}, \tag{31}$$

where

$$\delta_p, \delta_n = \underset{\substack{\delta_p,\delta_n\in X,\\ \|\delta_p\|_\infty\le\epsilon, \|\delta_n\|_\infty\le\epsilon}}{\arg\min} \|f(p+\delta_p) - f(n+\delta_n)\|_2 \tag{32}$$

In our experiments, we minimize Eq. (32) by using PGD [4] with the step size of $\frac{\epsilon}{10}$ and the number of updates of 20.

C-IBP trains the classifier f_c by simultaneously minimizing the original cross-entropy loss and cross-entropy loss due to the upper and lower bounds of the logits calculated by IBP. Let $\hat{f}_c^y(x)$ be the upper and lower bounds of the logits $f_c(x)$ where the logit of true class y is equal to its lower bound and the other logits are equal to their upper bounds. Then, C-IBP trains $f_c(x)$ by minimizing the following loss function with training data $D_t = \{(x, y)_i\}_{i=1}^M$:

$$\sum_{(x,y)\in D_t} \kappa \cdot CE(f_c(x), y) + (1-\kappa)\cdot CE(\hat{f}_c^y(x), y). \tag{33}$$

where CE represents Cross-Entropy loss. Note that we use the classifier trained with IBP without the final layer (logit layer) as a feature extractor in our experimentation.

Table 3: The model architectures of feature extraction DNNs in our experiments. Conv-f-k-s-p denotes a convolutional layer with a number of filters f of size $k \times k$, stride size is s, and padding size is p. Linear-d denotes a linear layer whose output dimension is d. When training C-IBP, we add one more linear layer to Small and Large to compute the logits. Note that there is a ReLU between each layer.

Small	Large
Conv-16-4-2-1	Conv-64-3-1-1
Conv-32-4-1-1	Conv-64-3-1-1
Linear-128	Conv-128-3-2-1
	Conv-128-3-1-1
	Conv-128-3-1-1
	Linear-128

C. Experimental Settings for MNIST, FMNIST, and CIFAR10

C.1. Architectures

In our experiments for MNIST, FMNIST, and CIFAR10, we train the feature extractor f of embedding dimensionality 128 in two different model architectures, as shown in the Table 3. We refer to each model as Small (3-layer CNN) and Large (6-layer CNN), respectively.

C.2. Hyperparameters

The total number of training epochs is 100 for MNIST and FMNIST and 200 for CIFAR10. We use the Adam optimizer [3] with a batch size of 100 and an initial learning rate of 0.001. We decay the learning rate by times 0.1 at 25 and 42 epochs for MNIST and FMNIST and times 0.5 every 10 epochs between 130 and 200 epochs for CIFAR10. The margin of triplet loss is set to $m = 1.0$. As data augmentation, we use random crop and random horizontal flip when training f on CIFAR10.

When training with TBT, to stabilize training, we use scheduling strategy for ϵ and κ proposed in [2]. Specifically, ϵ is gradually increased from 0.0 to ϵ_e, and the κ is gradually decreased from 1.0 to κ_e. We use $\epsilon_e = 0.2$ for MNIST and FMNIST, and $\epsilon_e = \frac{2}{255}$ for CIFAR10, respectively. We use $\kappa_e = 0.5$ for Table 1, Table 2, Table 4, Table 5, Table 6, and Table 7. Then, we linearly increase ϵ and decrease κ between $2K$ and $10K$ steps. The results of other κ_e are shown in Appendix G.

When training with FCTB, we fine-tune the pre-trained feature extractor with TBT. We set fixed ϵ to 0.2 for MNIST and FMNIST and $\frac{2}{255}$ for CIFAR10. We set fixed κ to 0.2 for MNIST and 0.1 for FMNIST and CIFAR10.

When training with ACT, we set the fixed maximum perturbation size of the adversarial examples as $\epsilon = 0.2$ for

MNIST and FMNIST and $\epsilon = \frac{2}{255}$ for CIFAR10. Then we generate them by using PGD [4] with the step size of $\frac{\epsilon}{10}$ and the number of updates of 20.

When training with C-IBP, to stabilize training, we also use scheduling strategy for ϵ and κ proposed in [2]. Specifically, ϵ is gradually increased from 0.0 to ϵ_e, and the κ is gradually decreased from 1.0 to κ_e. We use $\epsilon_e = 0.2$ for MNIST and FMNIST, and $\epsilon_e = \frac{2}{255}$ for CIFAR10, respectively. We use $\kappa_e = 0.5$ for Table 1, Table 2, Table 4, Table 5, Table 6, and Table 7. Then, we linearly increase ϵ and decrease κ between $2K$ and $10K$ steps. The results of other κ_e are shown in Appendix G.

D. Experimental Settings for CUB-200-2011

D.1. Dataset

CUB-200-2011 is a bird species dataset consisting of 200 classes [8]. We use the first 100 classes as training data and the remaining 100 classes as test data. We train feature extractors f on the training set and evaluate f using the test set. Let $Q = \{(q_i, y_{q_i})\}_{i=1}^{|Q|}$ and $C = \{(c_i, y_{c_i}) \in X\}_{i=1}^{|C|}$ be the annotated set of query and candidate images, respectively. We randomly select Q and C without duplication from the test set so that $|Q| = 1000$ and $|C| = 1000$. We resize images to 224×224. Pixel values of images are in $[0, 1]$.

D.2. Architectures

To train feature extractor f for CUB-200-2011, we use VGG11 architecture [7] pre-trained on ImageNet [1] with three linear layers replaced by a single linear layer. We obtain the pre-trained model from torchvision library in PyTorch [5]. The feature dimension of f is 128.

D.3. Hyperparameters of TBT and C-IBP

The total number of training epochs is 200. We use the Adam optimizer [3] with a batch size of 100 and an initial learning rate of 0.00001. We decay the learning rate by times 0.5 every 10 epochs between 130 and 200 epochs. The margin of triplet loss is set to $m = 1.0$. As data augmentation, we use random crop and random horizontal flip. We normalize each image channel with mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$.

When training with TBT, to stabilize training, we use scheduling strategy for ϵ and κ proposed in [2]. Specifically, ϵ is gradually increased from 0.0 to ϵ_e, and the κ is gradually decreased from 1.0 to κ_e. We use $\epsilon_e = \frac{1}{255}$. We use $\kappa_e = 0.5$. Then, we linearly increase ϵ and decrease κ between $2K$ and $10K$ steps.

When training with C-IBP, to stabilize training, we also use same scheduling strategy as TBT training. We use $\epsilon_e = \frac{1}{255}$ for CIFAR10. We use $\kappa_e = 0.5$. Then, we linearly increase ϵ and decrease κ between $2K$ and $10K$ steps.

E. Experimental Results for Large Models with Perturbation Size $\epsilon = 0.1$ and $\epsilon = \frac{3}{255}$

Table 4 shows the results for Large with perturbation size $\epsilon = 0.1$ for MNIST and FMNIST and $\epsilon = \frac{3}{255}$ for CIFAR10. From Table 4, we can see similar results for perturbation sizes $\epsilon = 0.2$ and $\epsilon = \frac{2}{255}$.

F. Experimental Results for Small Models

Table 5, Table 6 and Table 7 show the results of Recall@K, ER-Recall@K, and CR-Recall@K for Small. Table 6 represents results when we use $\epsilon = 0.2$ for MNIST and FMNIST and $\epsilon = \frac{2}{255}$ for CIFAR10. Table 7 represents results when we use $\epsilon = 0.1$ for MNIST and FMNIST and $\epsilon = \frac{3}{255}$ for CIFAR10. Note that we use the same hyperparameters as Large for training Small. From Table 5, Table 6 and Table 7, We can see similar results to results of Large.

G. Effect of hyper parameter κ

TBT and C-IBP can control trade-off between accuracy and certified robustness by changing $\kappa \in [0, 1.0]$ in Eq. (20) and Eq. (33). Here, we validate the trade-off when only κ_{end} is changed and the other hyperparameters are fixed. Figure 1 and Figure 2 show Recall@20 and CR-Recall@20 of TBT and C-IBP for Large and Small when we use $\epsilon = 0.2$ (MNIST and FMNIST) and $\epsilon = \frac{2}{255}$ (CIFAR10) and change $\kappa_{end} \in \{0.0, 0.3, 0.5, 0.7\}$. Note that we omit the results of TBT when its training collapse, which means trained feature extraction models return the same values for any test data. From Figure 1 and Figure 2, we can confirm that for smaller κ_{end}, TBT reduces Recall more than C-IBP, but TBT can significantly improve CR-Recall than C-IBP. These results also suggest that TBT is more successful than C-IBP in tightening Eq. (17) and Eq. (18).

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[2] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[4] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning

Table 4: Comparison of empirical robust (ER) Recall@K and certified robust (CR) Recall@K (Large). QA and CA represents query attack and candidate attack, respectively. For calculating ER-Recall@K and CR-Recall@K, we use $\epsilon = 0.1$ (MNIST and FMNIST) and $\epsilon = \frac{3}{255}$ (CIFAR10). Each value is rounded off to two decimal places.

		ER-Recall@K (QA)				CR-Recall@K (QA)				ER-Recall@K (CA)				CR-Recall@K (CA)			
	K	1	10	20	40	1	10	20	40	1	10	20	40	1	10	20	40
MNIST	Triplet	0.00	0.12	0.19	0.27	0.00	0.00	0.00	0.00	0.25	0.60	0.71	0.81	0.00	0.00	0.00	0.00
	ACT	0.99	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.99	1.00	1.00	1.00	0.00	0.00	0.00	0.00
	C-IBP	0.97	0.99	1.00	1.00	0.00	0.04	0.11	0.29	0.96	0.99	0.99	1.00	0.00	0.01	0.06	0.29
	TBT	0.94	0.98	0.98	0.99	0.15	0.66	0.80	0.89	0.94	0.98	0.99	0.99	0.12	0.92	0.98	0.98
	TBT+FCTB	0.93	0.98	0.98	0.99	0.16	0.66	0.78	0.89	0.93	0.98	0.98	0.99	0.12	0.92	0.97	0.98
FMNIST	Triplet	0.00	0.11	0.17	0.22	0.00	0.00	0.00	0.00	0.03	0.09	0.11	0.17	0.00	0.00	0.00	0.00
	ACT	0.80	0.97	0.98	0.99	0.00	0.00	0.00	0.00	0.72	0.96	0.98	0.99	0.00	0.00	0.00	0.00
	C-IBP	0.72	0.97	0.98	0.99	0.01	0.16	0.28	0.42	0.71	0.96	0.98	0.99	0.00	0.06	0.20	0.49
	TBT	0.61	0.93	0.97	0.98	0.11	0.44	0.59	0.71	0.59	0.93	0.97	0.98	0.01	0.49	0.76	0.94
	TBT+FCTB	0.63	0.93	0.97	0.99	0.11	0.47	0.59	0.70	0.61	0.93	0.96	0.98	0.02	0.47	0.80	0.94
CIFAR10	Triplet	0.07	0.56	0.69	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.17	0.00	0.00	0.00	0.00
	ACT	0.36	0.81	0.90	0.95	0.00	0.00	0.00	0.00	0.07	0.45	0.68	0.91	0.00	0.00	0.00	0.00
	C-IBP	0.40	0.87	0.94	0.98	0.00	0.01	0.02	0.06	0.32	0.83	0.93	0.98	0.00	0.01	0.01	0.06
	TBT	0.19	0.78	0.91	0.96	0.01	0.12	0.20	0.33	0.14	0.77	0.92	0.96	0.00	0.09	0.21	0.38
	TBT+FCTB	0.20	0.82	0.93	0.97	0.01	0.12	0.22	0.36	0.15	0.78	0.92	0.97	0.00	0.11	0.25	0.48

Table 5: Comparison of Recall@K (Small). Each value is rounded off to two decimal places.

	MNIST				FMNIST				CIFAR10			
K	1	10	20	40	1	10	20	40	1	10	20	40
Triplet	0.99	1.00	1.00	1.00	0.86	0.98	0.98	0.99	0.48	0.90	0.95	0.97
ACT	0.99	1.00	1.00	1.00	0.82	0.96	0.97	0.99	0.53	0.90	0.95	0.97
C-IBP	0.94	0.99	0.99	1.00	0.73	0.96	0.98	0.99	0.40	0.88	0.95	0.98
TBT	0.88	0.96	0.97	0.99	0.60	0.92	0.95	0.98	0.25	0.77	0.91	0.97
TBT+FCTB	0.86	0.96	0.98	0.99	0.60	0.91	0.95	0.98	0.25	0.76	0.91	0.97

models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[6] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[8] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

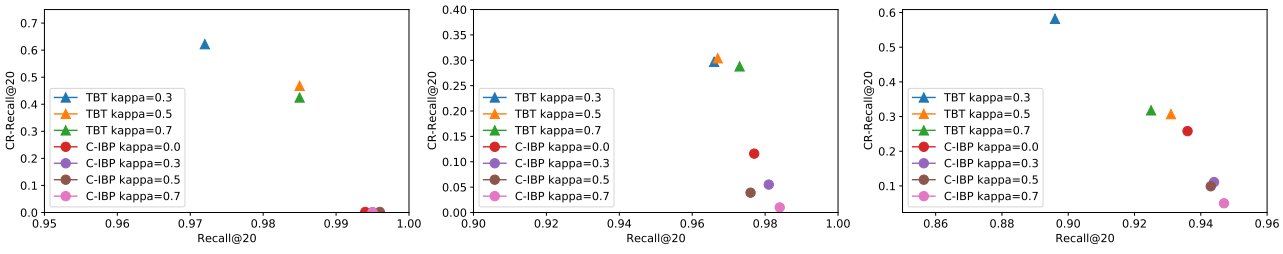
[9] Mo Zhou, Le Wang, Zhenxing Niu, Qilin Zhang, Nanning Zheng, and Gang Hua. Adversarial attack and defense in deep ranking. *arXiv preprint arXiv:2106.03614*, 2021.

Table 6: Comparison of empirical robust (ER) Recall@K and certified robust (CR) Recall@K (Small). QA and CA represents query attack and candidate attack, respectively. For calculating ER-Recall@K and CR-Recall@K, we use $\epsilon = 0.2$ (MNIST and FMNIST) and $\epsilon = \frac{2}{255}$ (CIFAR10). Each value is rounded off to two decimal places.

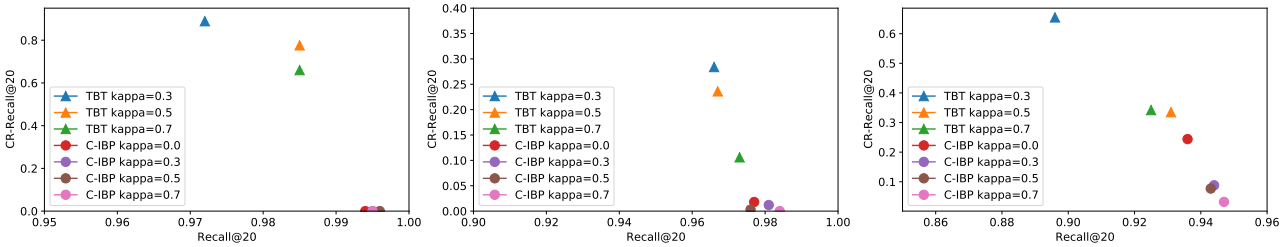
		ER-Recall@K (QA)				CR-Recall@K (QA)				ER-Recall@K (CA)				CR-Recall@K (CA)			
	K	1	10	20	40	1	10	20	40	1	10	20	40	1	10	20	40
MNIST	Triplet	0.00	0.09	0.14	0.23	0.00	0.00	0.00	0.00	0.06	0.13	0.19	0.33	0.00	0.00	0.00	0.00
	ACT	0.95	0.99	1.00	1.00	0.00	0.00	0.00	0.00	0.94	0.99	1.00	1.00	0.00	0.00	0.00	0.00
	C-IBP	0.91	0.98	0.99	1.00	0.00	0.00	0.00	0.01	0.87	0.98	0.99	1.00	0.00	0.00	0.00	0.00
	TBT	0.85	0.97	0.98	0.99	0.01	0.19	0.35	0.56	0.86	0.96	0.97	0.99	0.00	0.15	0.41	0.77
	TBT+FCTB	0.83	0.96	0.98	0.99	0.01	0.21	0.35	0.57	0.84	0.96	0.97	0.99	0.00	0.16	0.45	0.81
FMNIST	Triplet	0.00	0.11	0.17	0.24	0.00	0.00	0.00	0.00	0.02	0.05	0.06	0.09	0.00	0.00	0.00	0.00
	ACT	0.78	0.95	0.97	0.99	0.00	0.00	0.00	0.00	0.42	0.84	0.93	0.97	0.00	0.00	0.00	0.00
	C-IBP	0.71	0.96	0.97	0.99	0.00	0.02	0.05	0.14	0.62	0.94	0.97	0.99	0.00	0.00	0.01	0.03
	TBT	0.57	0.91	0.95	0.98	0.04	0.19	0.27	0.37	0.54	0.91	0.94	0.98	0.00	0.08	0.23	0.51
	TBT+FCTB	0.57	0.92	0.95	0.98	0.04	0.18	0.26	0.36	0.56	0.91	0.94	0.97	0.00	0.08	0.25	0.57
CIFAR10	Triplet	0.26	0.80	0.90	0.95	0.00	0.00	0.00	0.00	0.02	0.27	0.54	0.83	0.00	0.00	0.00	0.00
	ACT	0.41	0.86	0.93	0.97	0.00	0.00	0.00	0.00	0.13	0.66	0.85	0.95	0.00	0.00	0.00	0.00
	C-IBP	0.38	0.87	0.95	0.98	0.00	0.04	0.08	0.18	0.34	0.85	0.94	0.98	0.00	0.03	0.08	0.18
	TBT	0.24	0.79	0.91	0.97	0.04	0.34	0.52	0.71	0.22	0.76	0.89	0.97	0.02	0.34	0.58	0.82
	TBT+FCTB	0.22	0.77	0.90	0.98	0.04	0.32	0.51	0.74	0.23	0.74	0.90	0.97	0.03	0.36	0.60	0.84

Table 7: Comparison of empirical robust (ER) Recall@K and certified robust (CR) Recall@K (Small). QA and CA represents query attack and candidate attack, respectively. For calculating ER-Recall@K and CR-Recall@K, we use $\epsilon = 0.1$ (MNIST and FMNIST) and $\epsilon = \frac{3}{255}$ (CIFAR10). Each value is rounded off to two decimal places.

		ER-Recall@K (QA)				CR-Recall@K (QA)				ER-Recall@K (CA)				CR-Recall@K (CA)			
	K	1	10	20	40	1	10	20	40	1	10	20	40	1	10	20	40
MNIST	Triplet	0.09	0.44	0.60	0.71	0.00	0.00	0.00	0.00	0.31	0.73	0.82	0.91	0.00	0.00	0.00	0.00
	ACT	0.98	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.98	1.00	1.00	1.00	0.00	0.00	0.00	0.00
	C-IBP	0.93	0.99	0.99	1.00	0.00	0.01	0.04	0.15	0.91	0.99	0.99	1.00	0.00	0.00	0.03	0.13
	TBT	0.87	0.97	0.98	0.99	0.09	0.58	0.75	0.89	0.87	0.96	0.97	0.99	0.04	0.71	0.91	0.96
	TBT+FCTB	0.86	0.96	0.98	0.99	0.08	0.58	0.75	0.88	0.85	0.96	0.97	0.99	0.05	0.73	0.91	0.96
FMNIST	Triplet	0.00	0.14	0.21	0.31	0.00	0.00	0.00	0.00	0.02	0.05	0.08	0.16	0.00	0.00	0.00	0.00
	ACT	0.78	0.95	0.97	0.99	0.00	0.00	0.00	0.00	0.66	0.94	0.96	0.98	0.00	0.00	0.00	0.00
	C-IBP	0.74	0.96	0.98	0.99	0.02	0.19	0.32	0.48	0.69	0.95	0.98	0.99	0.00	0.07	0.23	0.54
	TBT	0.59	0.92	0.95	0.98	0.14	0.41	0.49	0.60	0.57	0.91	0.94	0.98	0.04	0.42	0.69	0.89
	TBT+FCTB	0.60	0.91	0.95	0.97	0.12	0.40	0.49	0.59	0.58	0.91	0.95	0.97	0.04	0.44	0.69	0.90
CIFAR10	Triplet	0.15	0.68	0.83	0.91	0.00	0.00	0.00	0.00	0.00	0.06	0.19	0.49	0.00	0.00	0.00	0.00
	ACT	0.33	0.82	0.91	0.95	0.00	0.00	0.00	0.00	0.03	0.37	0.62	0.88	0.00	0.00	0.00	0.00
	C-IBP	0.38	0.87	0.95	0.98	0.00	0.01	0.02	0.05	0.30	0.83	0.93	0.98	0.00	0.01	0.02	0.05
	TBT	0.22	0.79	0.90	0.97	0.02	0.21	0.39	0.56	0.21	0.75	0.89	0.97	0.01	0.21	0.40	0.66
	TBT+FCTB	0.23	0.76	0.91	0.98	0.02	0.23	0.37	0.58	0.22	0.73	0.90	0.97	0.01	0.22	0.45	0.71

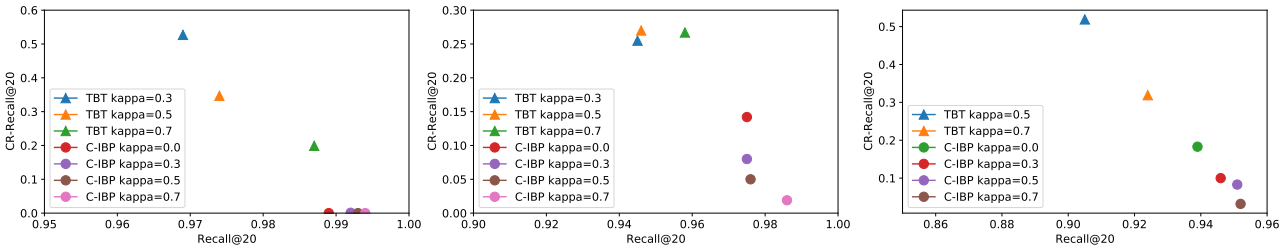


(a) MNIST (QA) (b) FMNIST (QA) (c) CIFAR10 (QA)

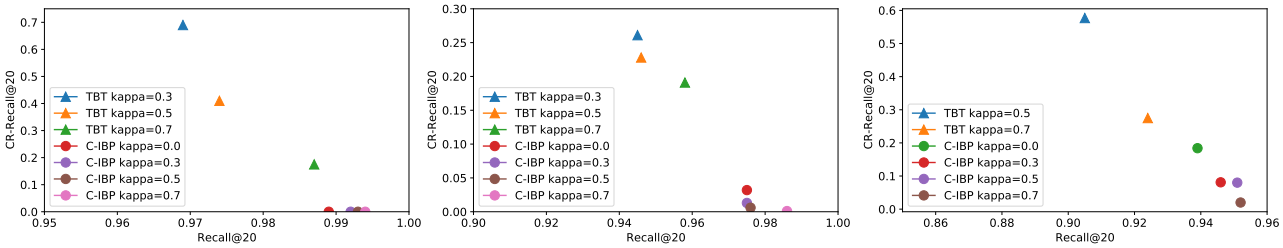


(d) MNIST (CA) (e) FMNIST (CA) (f) CIFAR10 (CA)

Figure 1: Trade-off between Recall@20 vs. CR-Recall@20 of TBT and C-IBP with different $\kappa_{end} \in \{0.0, 0.3, 0.5, 0.7\}$ (Large). QA and CA represents query attack and candidate attack, respectively. For calculating ER-Recall@20 and CR-Recall@20, we use $\epsilon = 0.2$ (MNIST and FMNIST) and $\epsilon = \frac{2}{255}$ (CIFAR10).



(a) MNIST (QA) (b) FMNIST (QA) (c) CIFAR10 (QA)



(d) MNIST (CA) (e) FMNIST (CA) (f) CIFAR10 (CA)

Figure 2: Trade-off between Recall@20 vs. CR-Recall@20 of TBT and C-IBP with different $\kappa_{end} \in \{0.0, 0.3, 0.5, 0.7\}$ (Small). QA and CA represents query attack and candidate attack, respectively. For calculating ER-Recall@20 and CR-Recall@20, we use $\epsilon = 0.2$ (MNIST and FMNIST) and $\epsilon = \frac{2}{255}$ (CIFAR10).