IEICE TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2023EDP7229

Publicized:2024/09/05

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Deterministic and Probabilistic Certified Defenses for Content-Based Image Retrieval*

Kazuya KAKIZAKI[†,††a)], **Kazuto FUKUCHI**[††,†††], ***Nonmembers**, **and*** **Jun SAKUMA**[††††,†††], ***Member***

SUMMARY This paper develops certified defenses for deep neural network (DNN) based content-based image retrieval (CBIR) against adversarial examples (AXs). Previous works put their effort into certified defense for classification to improve certified robustness, which guarantees that no AX to cause misclassification exists around the sample. Such certified defense, however, could not be applied to CBIR directly because the goals of adversarial attack against classification and CBIR are completely different. To develop the certified defense for CBIR, we first define the new certified robustness of CBIR, which guarantees that no AX that changes the ranking results of CBIR exists around the input images. Then, we propose computationally tractable verification algorithms that verify whether a given feature extraction DNN satisfies the certified robustness of CBIR at given input images. Our proposed verification algorithms are achieved by evaluating the upper and lower bounds of distances between feature representations of perturbed and non-perturbed images in deterministic and probabilistic manners. Finally, we propose robust training methods to obtain feature extraction DNNs that increase the number of inputs that satisfy the certified robustness of CBIR by tightening the upper and lower bounds. We experimentally show that our proposed certified defenses can guarantee robustness deterministically and probabilistically on various datasets.

key words: *adversarial example, certified defense, content-based image retrieval*

1. Introduction

Content-based image retrieval (CBIR) is a task that retrieves visually similar images to a given query image from a set of candidate images. Modern CBIR performs retrieval by ranking the similarity between the query image and candidate images based on feature extraction deep neural networks (DNNs) trained by metric learning [1]. However, recent studies reveal that such DNN-based CBIR is vulnerable to small human-imperceptible perturbation to the input data, called *adversarial examples (AXs)* [2]–[10]. Such AXs can be input to DNN-based CBIR as the query or candidate images and maliciously modify the ranking results by manipulating the output of the feature extraction DNNs. Since the DNN-based CBIR is often involved in security-critical systems such as person re-identification [11], defense methods for DNN-based CBIR against AXs are necessary.

A great deal of effort has been devoted to empirical defense methodologies for the classification task. *Adversarial training* [12], which trains DNNs using AXs as training data, is one of the most effective empirical defense methodologies for classification. Adversarial training has also been shown to be effective in CBIR empirically [2], [3]. While these empirical defense methods achieve robustness against conventional attacks, they often suffer from adaptive attacks [13], which assume the attacker is aware of the strategy of the defense method. Since there is no guarantee that these empirical defense methods are effective against adaptive attacks, defense methods with theoretical guarantees of robustness are needed to deal with adaptive attacks.

To overcome adaptive attacks, many studies have worked to establish defense with *certified robustness* of classification [14]. Certified robustness means that there is no AX to cause misclassification within an l_p-ball centered on a given sample. This type of defense is referred to as *certified defense*. Certified defense generally consists of (i) a verification algorithm to verify whether a given classifier satisfies certified robustness at a given sample and (ii) robust training for classifiers to increase the number of samples that can be verified by the corresponding verification algorithm. Since exact verification is known to be reduced to an NP-complete problem [15], [16], the verification algorithms alternatively evaluate a sufficient condition of certified robustness that depend on the upper and lower bounds of the classifier's predictions against AXs in the l_p-ball. Then, the bounds are computed by computationally tractable deterministic or probabilistic algorithms [17]–[23]. While using the bounds makes the verification computationally tractable, the results can include false negatives, i.e., given samples are determined to be not robust, even when they actually achieve certified robustness. Considering that the looseness of the bounds causes this gap, robust training to make this bound tighter has been introduced. By training the classifier in this way, we can expect to reduce the number of cases where robust samples are misjudged to be non-robust.

1.1 Related Work

Some studies proposed certified defenses for classification, which guarantee certified robustness deterministically. This type of certified defense includes a verification algorithm that computes the upper and lower bounds of logits in a deterministic manner. [20], [24] utilize the Lipschitz constant of neural networks to calculate the bounds. [17]–[19] calculate

[†] The author is with NEC Corporation.
[††] The author is with University of Tsukuba.
[††††] The author is with Tokyo Institute of Technology.
[†††] The author is with RIKEN AIP.
[*] This paper is an extension of the authors' previous work presented at IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2023.
a) E-mail: kazuya1210@nec.com

the bounds by linear relaxations of ReLU activations. [21], [22] propose a straightforward but effective method called interval bound propagation (IBP) that propagates the upper and lower bounds for each layer. These deterministic certified defenses can guarantee certified robustness for low-resolution images such as CIFAR10 (32x32) with small false negatives. However, they are known to be difficult to scale to high-resolution images such as ImageNet (224x224) [14] because the evaluation of the upper and lower bounds is too loose.

Another line of certified defense for classification is probabilistic. This type of certified defense is also called randomized smoothing (RS). RS achieves the verification for certified robustness by estimating the upper and lower bounds of the classifier's prediction in a probabilistic manner. Concretely, RS smooths the classifier with Gaussian [23], [25] or Laplace [26] distribution and theoretically derives the upper and lower bounds on the prediction of the smoothed model. Then, the bounds are probabilistically estimated by Monte Carlo estimation and hypothesis testing. As a result, different from the deterministic verification algorithms, the result of RS contains not only false negatives but also false positives with small probability. However, RS can scale to high-resolution images; RS can provide the meaningful evaluation of the upper and lower bounds for high-resolution images.

Although certified defense for classification has been investigated extensively, less attention has been paid to certified defense for CBIR. Moreover, the existing certified defenses for classification cannot be directly applied to CBIR because the goals of the adversarial attack against classification and CBIR are completely different. Specifically, the adversarial attacks against classification aim to change the predicted class label of the classifier, whereas the adversarial attacks against CBIR aim to change the ranking results of CBIR calculated by feature extraction DNNs. Thus, new definitions of certified robustness tailored to CBIR and certified defense to guarantee it should be considered. Only [27] have proposed a certified defense for CBIR, named RetrievalGuard, which guarantees that no AX changes the top-1 ranking results of CBIR. However, since RetrievalGuard does not guarantee the invariance of any ranking result other than top-1, its use is limited.

1.2 Our Contributions

In this paper, we develop two types of certified defenses for CBIR that guarantee robustness deterministically or probabilistically. Our contribution is four-fold. First, we define the new certified robustness of CBIR, named (l_p, α, ϵ)-robustness. (l_p, α, ϵ)-robustness means that no AX exists within l_p-balls with radius $\epsilon \in \mathbb{R}$ centered on the query or candidate images that changes the ranking result of a specific candidate image more than $\alpha \in \mathbb{N}$. Different from the existing certified robustness of CBIR that guarantees the invariance of top-1 ranking results [27], (l_p, α, ϵ)-robustness can guarantee the invariance of any ranking results.

Second, we introduce a tractable sufficient condition for (l_p, α, ϵ)-robustness. To verify whether a given feature extraction DNN satisfies (l_p, α, ϵ)-robustness at given a query and candidate images, we need to evaluate the exact maximum and minimum distances in the feature space between AXs in the l_p-balls and the benign images. That makes the verification computationally intractable. To alleviate this, we derive the sufficient condition for (l_p, α, ϵ)-robustness using the upper and lower bounds of the distances. Then, we achieve certified defenses that guarantee $(l_\infty, \alpha, \epsilon)$-robustness and (l_2, α, ϵ)-robustness by evaluating the upper and lower bounds of the distances with our proposed deterministic and probabilistic tractable methods, respectively.

Third, we present a certified defense for CBIR, which guarantees $(l_\infty, \alpha, \epsilon)$-robustness in a deterministic manner. To this end, we first propose a tractable verification algorithm, which evaluates the derived sufficient condition by applying interval bound propagation (IBP) [22] to feature extraction DNNs. Concretely, we evaluate the upper and lower bounds of the distances by propagating bounds from the input space to the feature space. Since the computational complexity of IBP is equivalent to two forward propagations of DNNs, we can evaluate the derived sufficient conditions in polynomial time. Moreover, we also propose robustness training methods of feature extraction DNNs that attain tighter evaluation of the upper and lower bounds of the distances. When the bounds are loose, our verification algorithms can judge truly robust inputs as non-robust. To decrease such misjudging, we introduce new objective functions to train feature extraction DNNs that encourage tighter bounds of distances evaluated by IBP. We experimentally confirmed that the robustness training method can increase the number of samples verified by our proposed deterministic verification algorithm for MNIST [28], Fashion-MNIST [29], and CIFAR10 [30].

Fourth, we present a certified defense for CBIR, which guarantees (l_2, α, ϵ)-robustness probabilistically. We experimentally confirmed that our proposed deterministic certified defense does not scale to high-resolution images such as CUB (224x224) [31]. To overcome this limitation, inspired by the success of randomized smoothing (RS) for classification setting, we propose a verification algorithm to evaluate the derived sufficient condition using the upper and lower bounds of *Gaussian smoothed distances*, of which input is smoothed with Gaussian distributions. Specifically, we theoretically derive the upper and lower bounds on the Gaussian smoothed distance and probabilistically estimate them by our proposed Monte Carlo algorithms. Moreover, we propose to use Gaussian data augmentation [23], [27] to training data of feature extraction DNNs for obtaining tighter bounds of the Gaussian smoothed distances. We theoretically and experimentally show that our probabilistic certified defense, different from the deterministic one, includes false positives as well as false negatives but can scale to higher-resolution images.

Note that this is an extension of our conference paper [32]. The main content extended from the conference paper is the proposal and experimental evaluations of the probabilistic certified defense.

The paper is organized as follows. Section 2 describes the background of this study and defines the new certified robustness of CBIR, (l_p, α, ϵ)-robustness. Section 3 derives the sufficient conditions of (l_p, α, ϵ)-robustness using the upper and lower bounds of the distances. Section 4 proposes deterministic certified defense: deterministic verification algorithms and their robustness training. Section 5 proposes probabilistic certified defense: probabilistic verification algorithms and their robustness training. In Section 6, we evaluate our proposed deterministic and probabilistic certified defenses. Section 7 provides potential directions for future research focusing on enhancing the certified defense for CBIR. Section 8 concludes the paper.

2. Preliminaries

2.1 Content-Based Image Retrieval (CBIR)

CBIR is a task to find images similar to a query image in a set of candidate images. Let X be the instance space. Let $q \in X$ be a query image and $C = \{c_i | c_i \in X\}_{i=1}^{|C|}$ be a set of candidate images. Let $f : X \to \mathbb{R}^d$ be a feature extractor where d is the feature dimension. Then, CBIR ranks $\forall c \in C$ with distance $d(f(q), f(c))$ and retrieves the top-k similar images to q in C. In this paper, $\text{Rank}(q, c, C)$ represents the rank of $c \in C$ in terms of the similarity to q. We omit f from the augments of $\text{Rank}(q, c, C)$ for notational simplicity when it is obvious from the context.

2.2 Adversarial Attacks against CBIR

In recent years, many studies have focused on adversarial attacks on CBIR [2]–[10]. These attacks can be categorized into two types of attacks, *query attack (QA)* and *candidate attack (CA)*, depending on whether the AX is given as a query image or a candidate image.

Query Attack (QA). Let $C_t \subset C$ be the target candidates in C specified by the adversary. The adversary aiming at QA perturbs a source query image q_s to raise or lower the rank of the candidates in C_t. When the attacker's goal is to raise the rank of the candidates in C_t, adversarial perturbation δ for QA is obtained by solving the following optimization problem:

$$\min_{\delta \in X, \|\delta\|_p \leq \epsilon} \sum_{t \in C_t} \text{Rank}(q_s + \delta, t, C), \tag{1}$$

where $\|\cdot\|_p$ is l_p norm and $\epsilon \in \mathbb{R}_{\geq 0}$ is a constant that bounds the size of the perturbation. Eq. (1) cannot be solved directly due to the discrete nature of $\text{Rank}(\cdot)$. Instead, [2], [3] minimizes the following objective function:

$$\min_{\substack{\delta \in X, \\ \|\delta\|_p \leq \epsilon}} \sum_{t \in C_t} \sum_{c \in C} [d(f(q_s{+}\delta), f(t)){-}d(f(q_s{+}\delta), f(c))]_+. \tag{2}$$

Minimization in Eq. (1) is changed to maximization when the attacker's goal is to lower the rank of the candidates in C_t.

Candidate Attack (CA). Let $Q_t = \{q_i \in X\}_{i=1}^{M}$ be a set of target query images specified by the adversary. The adversary aiming at CA perturbs a source candidate image $c_s \in C$ so that the rank of perturbed c_s is raised or lowered when $\forall q \in Q_t$ is issued as a query. When the attacker's goal is to raise the rank of the perturbed c_s, adversarial perturbation for CA is obtained by the following minimization problem with respect to δ:

$$\min_{\delta \in X, \|\delta\|_p \leq \epsilon} \sum_{t \in Q_t} \text{Rank}(t, c_s + \delta, C). \tag{3}$$

where $\|\cdot\|_p$ is l_p norm and $\epsilon \in \mathbb{R}_{\geq 0}$ is a constant that bounds the size of the perturbation. Since optimization in Eq. (3) is intractable, [2], [3] optimizes the following objective function instead:

$$\min_{\substack{\delta \in X, \\ \|\delta\|_p \leq \epsilon}} \sum_{t \in Q_t} \sum_{c \in C} \Big[d(f(t), f(c_s + \delta)) - d(f(t), f(c))\Big]_+, \tag{4}$$

As well as QA, minimization in Eq. (3) is changed to maximization when the attacker's goal is to lower the rank of the perturbed c.

2.3 Certified Robustness

Here, we briefly review the existing definition of the certified robustness and verification algorithms for classification. Then, we define two new certified robustness of CBIR.

2.3.1 Certified Robustness of Classification.

The adversarial attacks against the classifier aim to change the predicted label of the classifier to an untargeted or targeted label by perturbing the input images [12], [33]. The certified robustness of classification guarantees that predicted labels are kept invariant when the size of adversarial perturbation is limited within a specified range:

Definition 1 (Certified Robustness of Classification [14])**.** *Let $x \in X$ be a input image and $t \in \{1, \ldots, C\}$ be corresponding label to x. Let $f_c : X \to \mathbb{R}^C$ and $f_c(x)_j$ be the vector of logits and the logit of class $j \in \{1, \ldots, C\}$ for x, respectively. Let $\epsilon \in \mathbb{R}_{\geq 0}$. Then, classifier $F_c(x) := \arg\max_{j \in \{1,\ldots,C\}} f_c(x)_j$ is certified robust at x if $F_c(x + \delta) = t$ for $\forall \delta \in \{\delta \mid \delta \in X, \|\delta\|_p \leq \epsilon\}$.*

Verification Algorithms for Classification. When classifier F_c is DNN with ReLUs, verifying whether given F_c satisfies certified robustness at given x is reduced to an NP-complete problem [15], [16]. To make the verification computationally tractable, existing verification algorithms use the lower bounds of margins between logits $\underline{m}_i(x) \leq \min_{\delta, \|\delta\|_p \leq \epsilon} f_c(x + \delta)_t - f_c(x + \delta)_{i:i \neq t}$ or class probabilities $\underline{m}_i(x) \leq \min_{\delta, \|\delta\|_p \leq \epsilon} \Pr[F_c(x+\delta) = t] - \Pr[f_c(x+\delta) = i]_{i:i\neq t}$ against AXs in the l_p-ball. Then, the bounds are computed by computationally tractable deterministic methods such as interval bound propagation [22] or estimated by probabilistic methods such as randomized smoothing [23]. Then, if $\underline{m}_i(x) > 0$, the verification algorithms determine the classifier satisfies certified robustness at x. In this paper, we call the verification algorithms *deterministic* and *probabilistic verification algorithms* when the bounds are computed through deterministic and probabilistic methods, respectively. We remark that the results of the deterministic verification algorithms do not include false positives but false negatives because $\underline{m}_i(x) > 0$ is a sufficient condition for Definition 1. On the other hand, the results of the probabilistic verification algorithms include not only false negatives but also false positives with a certain probability. However, the probabilistic verification algorithms have the advantage of scaling to large-scale images such as ImageNet [34], which is difficult to be verified by the deterministic verification algorithms [23].

2.3.2 Certified Robustness of CBIR.

Definition 1 is not suitable for CBIR because the goals of adversarial attacks against classification and CBIR are different: the adversarial attacks against classification aim to change the predicted class label of the classifier, whereas QA and CA aim to change the rank of the candidates. Thus, in certified defense for CBIR, we need to consider rank invariance rather than label invariance against AXs. We define the certified robustness of CBIR against QA and CA as follows, respectively:

Definition 2 ((l_p, α, ϵ)-Robustness against QA)**.** *Let $f : X \to \mathbb{R}^d$ be a feature extractor. Let $q \in X$ and $C = \{c_i | c_i \in X\}_{i=1}^N$ be a query image and a set of candidate images, respectively. Let $\alpha \in \mathbb{N}_0$ and $\epsilon \in \mathbb{R}_{\geq 0}$. Then, for $\forall \delta \in \{\delta \mid \delta \in X, \|\delta\|_p \leq \epsilon\}$, f satisfies (l_p, α, ϵ)-robust against QA at q, $c_i \in C$, and C if*

$$|\mathrm{Rank}(q+\delta, c_i, C) - \mathrm{Rank}(q, c_i, C)| \leq \alpha. \tag{5}$$

Definition 3 ((l_p, α, ϵ)-Robustness against CA)**.** *Let $f : X \to \mathbb{R}^d$ be a feature extractor. Let $q \in X$ and $C = \{c_i | c_i \in X\}_{i=1}^N$ be a query image and a set of candidate images, respectively. Let $\alpha \in \mathbb{N}_0$ and $\epsilon \in \mathbb{R}_{\geq 0}$. Then, for $\tilde{C} = \{c_i + \delta_i\}_{i=1}^N$ where $\forall \delta_1, ..., \forall \delta_N \in \{\delta \mid \delta \in X, \|\delta\|_p \leq \epsilon\}$, f satisfies (l_p, α, ϵ)-robust against CA at q, $c_i \in C$, and C if*

$$|\mathrm{Rank}(q, c_i + \delta_i, \tilde{C}) - \mathrm{Rank}(q, c_i, C)| \leq \alpha. \tag{6}$$

In both robustness definitions, we introduced α to relax the strictness of the guarantee because requiring complete rank invariance can be too strict.

3. Sufficient Conditions for (l_p, α, ϵ)-Robustness

Since verifying whether given inputs satisfy (l_p, α, ϵ)-robustness against QA and CA (Definition 2 and Definition 3) are computationally intractable, the key challenge of designing the verification algorithms is to make them relax and computationally efficient. Our idea to recover tractability is to introduce computationally tractable sufficient conditions for them. Unfortunately, the existing sufficient condition for certified robustness for classifier described in Section 2.3 ($\underline{m}_i(x) > 0$) cannot be used directly because (l_p, α, ϵ)-robustness against QA and CA depend on the distances in the feature space rather than the margins of logits or class probabilities. Thus, in this section, we first derive sufficient conditions for (l_p, α, ϵ)-robustness against QA and CA using the upper and lower bounds of the distances against AXs in the l_p-ball, assuming that the bounds can be obtained in a tractable way. Then, in Section 4 and Section 5, we introduce computationally tractable algorithms to obtain the upper and lower bounds of the distances in a deterministic and probabilistic manner, respectively.

Let $x_1, x_2 \in X$ and $\epsilon \in \mathbb{R}_{\geq 0}$. Then, we define upper and lower bounds of distance against AXs in the l_p-ball as follows:

$$\overline{d}_{x_2}(x_1) \geq \max_{\delta, \|\delta\|_p \leq \epsilon} d(f(x_1+\delta), f(x_2)), \tag{7}$$

$$\underline{d}_{x_2}(x_1) \leq \min_{\delta, \|\delta\|_p \leq \epsilon} d(f(x_1+\delta), f(x_2)). \tag{8}$$

We omit f from the augments of $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$ for notational simplicity when it is obvious from the context.

To derive sufficient conditions for Definition 2 and Definition 3, we first derive upper and lower bounds of $\mathrm{Rank}(q+\delta, c_i, C)$ in Eq. (5) and $\mathrm{Rank}(q, c_i + \delta_i, \tilde{C})$ in Eq.(6) by comparing $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$:

Lemma 1 (Upper and Lower Bounds of Rank against QA)**.** *For $\forall \delta \in \{\delta \mid \delta \in X, \|\delta\|_p \leq \epsilon\}$, the following holds:*

$$|C| - \sum_{c \in C} \mathbb{1}[\overline{d}_{c_i}(q) < \underline{d}_c(q)] \geq \mathrm{Rank}(q+\delta, c_i, C) \tag{9}$$

$$\sum_{c \in C} \mathbb{1}[\overline{d}_c(q) < \underline{d}_{c_i}(q)] + 1 \leq \mathrm{Rank}(q+\delta, c_i, C). \tag{10}$$

Proof. The proof is shown in Appendix A.1. □

Lemma 2 (Upper and Lower Bounds of Rank against CA)**.** *For $\tilde{C} = \{c_i + \delta_i\}_{i=1}^N$ where $\forall \delta_1, ..., \forall \delta_N \in \{\delta \mid \delta \in X, \|\delta\|_p \leq \epsilon\}$, the following holds:*

$$|C| - \sum_{c \in C} \mathbb{1}[\overline{d}_q(c_i) < \underline{d}_q(c)] \geq \mathrm{Rank}(q, c_i + \delta_i, \tilde{C}) \tag{11}$$

$$\sum_{c \in C} \mathbb{1}[\overline{d}_q(c) < \underline{d}_q(c_i)] + 1 \leq \mathrm{Rank}(q, c_i + \delta_i, \tilde{C}). \tag{12}$$

Proof. The proof is shown in Appendix A.2. □

From Theorem 1 and Theorem 2, we can also immediately obtain the upper and lower bounds of

$|\mathrm{Rank}(q+\delta, c_i, C) - \mathrm{Rank}(q, c_i, C)|$ and $|\mathrm{Rank}(q, c_i+\delta_j, \tilde{C}) - \mathrm{Rank}(q, c_i, C)|$ in Eq. (5) and Eq. (6), respectively. We can derive sufficient condition for Definition 2 and Definition 3 by comparing the bounds with α:

Theorem 1 (Sufficient Condition for (α, ϵ)-Robustness against QA)**.** *Feature extractor f satisfies (l_p, α, ϵ)-robustness against QA at q, $c_i \in C$, and C if*

$$\alpha \geq |C| - \sum_{c \in C} \mathbb{1}\left[\overline{d}_{c_i}(q) < \underline{d}_c(q)\right] - \mathrm{Rank}(q, c_i, C) \tag{13}$$
$$\wedge -\alpha \leq \sum_{c \in C} \mathbb{1}\left[\overline{d}_c(q) < \underline{d}_{c_i}(q).\right] + 1 - \mathrm{Rank}(q, c_i, C).$$

Proof. The proof is shown in Appendix A.3 □

Theorem 2 (Sufficient Condition for (α, ϵ)-Robustness against CA)**.** *Feature extractor f satisfies satisfies (l_p, α, ϵ)-robust against CA at q, $c_i \in C$, and C if*

$$\alpha \geq |C| - \sum_{c \in C} \mathbb{1}\left[\overline{d}_q(c_i) < \underline{d}_q(c)\right] - \mathrm{Rank}(q, c_i, C) \tag{14}$$
$$\wedge -\alpha \leq \sum_{c \in C} \mathbb{1}\left[\overline{d}_q(c) < \underline{d}_q(c_i)\right] + 1 - \mathrm{Rank}(q, c_i, C).$$

Proof. The proof is shown in Appendix A.4. □

From Theorem 1 and Theorem 2, verifying (l_p, ϵ, α)-robustness against QA and CA is computationally tractable if the evaluation of $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$ is computationally tractable.

4. Deterministic Certified Defense for CBIR

In this section, we propose deterministic certified defenses for CBIR, which guarantee $(l_\infty, \epsilon, \alpha)$-robustness against QA and CA deterministically.

4.1 Deterministic Verification Algorithms

In this subsection, we propose verification algorithms, which verify whether given inputs satisfy $(l_\infty, \epsilon, \alpha)$-robustness against QA and CA in a tractable deterministic manner. Our deterministic verification algorithms evaluate the derived sufficient conditions Eq. (13) and Eq. (14). Since they are sufficient conditions, they do not necessarily hold for inputs truly satisfying (l_p, ϵ, α)-robustness against QA and CA. Whether they can hold depends on the tightness of $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$. For this reason, we need to obtain meaningfully tight evaluation of $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$. To obtain a meaningfully tight evaluation of $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$, we utilize interval bound propagation (IBP) [22], [35]. IBP is an tractable algorithm for calculating the upper and lower bounds of logits when a l_∞-ball is given as input. IBP is used for deterministic verification for classification and is known to give a meaningfully tight bound for this purpose.

Original interval bound propagation (IBP). Given, $x \in X$, $\epsilon \in \mathbb{R}_{\geq 0}$, and L-layer classifier f_c, original IBP evaluates the upper and lower bounds of $f_c(x+\delta)$ for $\forall \delta \in \{\delta \mid \delta \in X, \|\delta\|_\infty \leq \epsilon\}$. Let $z^l = W^l h^{l-1} + b^l$ be the l-th affine layer (e.g. fully connected layer and convolution layer) and $h^{l-1} = \sigma(z^{l-1})$ be a monotonic activation function (e.g. ReLU) where $l \in \{1, ..., L\}$ and $h^0 = x$. Then, IBP provides upper and lower bounds on the outputs of l-th affine layers as follows:

$$\overline{z}^l = W^l \frac{\overline{h}^{l-1} + \underline{h}^{l-1}}{2} + |W^l| \frac{\overline{h}^{l-1} - \underline{h}^{l-1}}{2} + b^l, \tag{15}$$

$$\underline{z}^l = W^l \frac{\overline{h}^{l-1} + \underline{h}^{l-1}}{2} - |W^l| \frac{\overline{h}^{l-1} - \underline{h}^{l-1}}{2} + b^l, \tag{16}$$

where $|\cdot|$ represents the element-wise absolute value operator, $\overline{h}^{l-1} = \sigma(\overline{z}^{l-1})$, $\underline{h}^{l-1} = \sigma(\underline{z}^{l-1})$, $\overline{h}^0 = x + \epsilon 1$, and $\underline{h}^0 = x - \epsilon 1$.

4.1.1 Evaluation for $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$ via IBP

We propose tractable methods to evaluate $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$ when using Euclidean distance as $d_{x_2}(x_1)$. Let $f(x)_i$ be the i-th element of $f(x)$. Let $\overline{f}(x)_i$ and $f(x)_i$ be upper and lower bounds of $f(x)_i$ calculated by IBP, respectively. Then, we can evaluate $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$ by the following theorems:

Theorem 3. $\max_{\delta \in X, \|\delta\|_\infty \leq \epsilon} \|f(x_1+\delta) - f(x_2)\|_2$ *is upper bounded by*

$$\sqrt{\sum_{i \in \{1,..,d\}} \max\{|\overline{f}(x_1)_i - f(x_2)_i|, |f(x_2)_i - \underline{f}(x_1)_i|\}^2}. \tag{17}$$

Proof. The proof is shown in A.5. □

Theorem 4. $\min_{\delta \in X, \|\delta\|_\infty \leq \epsilon} \|f(x_1+\delta) - f(x_2)\|_2$ *is lower bounded by*

$$\sqrt{\sum_{i \in \{1,..,d\}} \min\{0, \overline{f}(x_1)_i - f(x_2)_i, f(x_2)_i - \underline{f}(x_1)_i\}^2}. \tag{18}$$

Proof. The proof is shown in A.6. □

Evaluating Eq. (17) and Eq. (18) to determine if the derived sufficient condition Eq. (13)/Eq. (14) is satisfied, we can obtain our tractable deterministic verification algorithms to verify $(l_\infty, \alpha, \epsilon)$-robustness against QA and CA as follows:

$$\mathsf{Ver}_{\alpha,\epsilon}(q, c_i, C) = \begin{cases} \text{True} & \text{if Eq. (13)/(14) is True} \\ \text{False} & \text{otherwise.} \end{cases} \tag{19}$$

We omit f from the augments of $\mathsf{Ver}_{\alpha,\epsilon}(q, c_i, C)$ for notational simplicity when it is obvious from the context.

The computational costs of calculating the upper bound in Eq. (17) and the lower bound in Eq. (18) for a single pair of (x_1, x_2) is comparable to three forward propagation of DNNs. The total number of forwards in evaluating Eq. (13) and Eq. (14) is equivalent to $|C|+3$ and $3|C|+1$, respectively.

4.2 Robust Training Methods

In this subsection, we propose robustness training methods that increase the number of samples verified by our deterministic verification algorithms, Eq.(19). We experimentally confirm that Eq.(13) and Eq.(14) are always not satisfied for all q, c_i, and C used in our experiments when using f trained by conventional metric learning (See Section 6 for details). This is because the upper and lower bounds calculated by Eq. (17) and Eq. (18) can be too loose to satisfy the sufficient conditions Eq. (13) and Eq. (14). To increase the number of inputs that satisfy Eq. (13) and Eq. (14), we need to train f so that attains tighter evaluation of $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$.

To this end, we propose two new objective functions to train feature extractor for CBIR. One is training of general feature extractor that attains tighter bounds in Eq. (17) and Eq. (18) without knowledge of query and candidate images. The other is fine tuning of feature extractor given that candidate images for the target CBIR are provided. We remark that both algorithms are independent, and the latter algorithm can be applied to the feature extractor trained with the former algorithm.

4.2.1 Training General Feature Extractor for Robust CBIR

Recall that tighter evaluation of the upper bound in Eq. (17) and the lower bound in Eq. (18) is needed to attain certified robustness in a meaningful way. Our idea is to train f by simultaneously minimizing conventional objective function (e.g., triplet loss [36]) and the regularization term to make the bounds in Eq. (17) and Eq. (18) tighter.

Let $D_{train} = \{(a, p, n)_i\}_{i=1}^{M}$ be a training data set where p belongs to the same class as a, and n belongs to a different class than a. Here, the training dataset and query/candidate images of CBIR are mutually exclusive. Then, our objective function is given as follows:

$$\min_f \sum_{(a,p,n)\in D_{train}} \kappa\cdot \mathrm{T}(a, p, n)+(1-\kappa)\cdot \sum_{x\in\{p,n\}} \mathrm{Reg}(a, x), \tag{20}$$

where $\mathrm{Reg}(a, x) = \max\{|d(f(a), f(x)) - \overline{d}_x(a)|, |d(f(a), f(x)) - \underline{d}_x(a)|\}$ and $\mathrm{T}(a, p, n)$ is the triplet loss [36] often used in metric learning, which affects the performance of CBIR. $\mathrm{Reg}(a, x)$ is a regularization term to encourage that the upper and lower bound of $\|f(a + \delta) - f(x)\|_2$ are close to $\|f(a) - f(x)\|_2$. $\kappa \in [0, 1]$ is a hyper parameter to adjust the trade-off between performance of CBIR and $(l_\infty, \alpha, \epsilon)$-robustness of CBIR against QA and CA. We call the training with Eq. (20) as Tightly Bounding Training (TBT).

4.2.2 Fine-tuning DNNs to Candidate Images

The feature extractor obtained by Eq. (20) is independent of the CBIR query and candidate set. In this subsection, assuming that the candidates images for the target CBIR are given, we show a method to fine tune the feature extractor to the set of candidate images. The objective of this fine-tuning is to reduce the gap between Definition 3 and the corresponding sufficient condition in Eq. (14) by adjusting f with the given candidate images. To achieve this, we update f so that tighter evaluation of Eq. (17) and Eq. (18) is attained with given candidate images while maintaining the performance of CBIR.

Let $C = \{c_i | c_i \in X\}_{i=1}^{N}$ be the set of candidate images. Let f_0 be the pre-trained feature extractor before fine-tuning. Then, our objective function for fine-tuning is given as follows:

$$\min_f \sum_{c_1,c_2\in C} \Big(\kappa \cdot d(f_0(c_1), f(c_1)) + (1-\kappa)\cdot \mathrm{Reg}(c_1, c_2)\Big). \tag{21}$$

The first term maintains the accuracy of the CBIR by ensuring that the difference between the features calculated by f and f_0 is small. The second term is a regularization term to encourage that the upper and lower bound of $\|f(c_1+\delta)-f(c_2)\|_2$ are close to $\|f(c_1)-f(c_2)\|_2$. $\kappa \in [0, 1]$ is a hyperparameter to adjust the trade-off between the performance of CBIR and $(l_\infty, \alpha, \epsilon)$-robustness against CA. We call fine-tuning with Eq. (21) as Fine-tuning to Candidates with Tighter Bounds (FCTB).

5. Probabilistic Certified Defense

In this section, we propose probabilistic certified defenses for CBIR, which guarantee (l_2, ϵ, α)-robustness against QA and CA probabilistically.

5.1 Probabilistic Verification Algorithms

In this subsection, we propose algorithms to verify whether given inputs satisfy (l_2, ϵ, α)-robustness against QA and CA in a probabilistic manner. We experimentally confirm that our deterministic verification algorithms, Eq.(19), can not verify high-resolution images even if we learn f with our robust training method, Eq.(20) (See Section 6 for details). To overcome the limitation, we utilize randomized smoothing (RS) [23], [25] to evaluate $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$. RS is a tractable algorithm to estimate the upper and lower bounds of the class probability of the classifier against AXs in an l_2-ball. RS is used for probabilistic verification algorithms for classification and is known to scale to high-resolution images [14].

Original randomized smoothing (RS). Let $F_c : X \to [C]$ be a classifier. Let $N(0, \sigma^2 I)$ be a Gaussian distribution with mean 0 and standard deviation σ. Gaussian smoothed classifier $G_{F_c}(x) : X \to [C]$ is defined as follows:

$$G_{F_c}(x) := \underset{j \in [C]}{\arg\max} \Pr_{\xi \sim N(0,\sigma^2 I)} (F_c(x + \xi) = j). \quad (22)$$

Then, randomized smoothing estimates the upper and lower bounds of the class probability of Gaussian smoothed classifier $\Pr[G_{F_c}(x + \delta) = i]$ for $\forall \delta \in \{\delta \mid \delta \in X, \|\delta\|_2 \leq \epsilon\}$ with a certain probabilistic guarantee. Precisely, randomized smoothing uses Monte Carlo sampling and hypothesis testing to estimate the upper and lower bounds $\Phi(\Phi^{-1}(\Pr[G_{F_c}(x) = i]) - \epsilon/\delta) \leq \Pr[G_{F_c}(x + \delta) = i] \leq \Phi(\Phi^{-1}(\Pr[G_{F_c}(x) = i]) + \epsilon/\delta)$ theoretically derived by the Lipschitz continuity of Gaussian smoothed functions:

Theorem 5 (Lipschitz continuity of Gaussian Smoothed Functions [25], [37]). *For any function $h : X \to [0, 1]$, $\Phi^{-1}(\mathbb{E}_{\xi \sim N(0,\sigma^2 I)}[h(x + \xi)])$ is $\frac{1}{\sigma}$-Lipschitz in terms of l_2 distance, where Φ^{-1} is the inverse of standard Gaussian CDF.*

5.1.1 Evaluation for $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$ via RS

We utilize randomized smoothing to estimate $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$. Different from the original randomized smoothing, which smooths the classifier, we smooth the distance with a Gaussian distribution. Precisely, we use the following *Gaussian smoothed distance* $sd_{x_2}(x_1)$ as the distance (i.e., $d_{x_1}(x_2)$) for computing the CBIR ranking results and estimate the upper and lower bounds of $sd_{x_2}(x_1)$:

Definition 4. *Let $x_1, x_2 \in X$ be inputs and $f : X \to [0, 1]^d$ be a feature extractor normalized so that $\|f(x)\|_2 = 1$. Then, Gaussian smoothed distance $sd_{x_2}(x_1)$ is defined as follows:*

$$sd_{x_2}(x_1) := \mathbb{E}_{\xi \sim N(0,\sigma^2 I)} \frac{1}{2} \|f(x_1 + \xi) - f(x_2)\|_2. \quad (23)$$

Upper and lower bound of $sd_{x_2}(x_1)$. Since the range of $sd_{x_2}(x_1)$ is [0, 1], $\Phi^{-1}(sd_{x_2}(x_1))$ is $\frac{1}{\sigma}$-Lipschitz continuity in terms of l_2 distance from Theorem 5. Thus, we can immediately derive the upper and lower bounds of the Gaussian smoothed distance against AXs in l_2-ball (i.e., $\overline{d}_{x_2}(x_1)$ and $\underline{d}_{x_2}(x_1)$) as follows:

$$\Phi(\Phi^{-1}(sd_{x_2}(x_1)) + \frac{\epsilon}{\sigma}) \geq \max_{\delta, \|\delta\|_2 \leq \epsilon} sd_{x_2}(x_1 + \delta), \quad (24)$$

$$\Phi(\Phi^{-1}(sd_{x_2}(x_1)) - \frac{\epsilon}{\sigma}) \leq \min_{\delta, \|\delta\|_2 \leq \epsilon} sd_{x_2}(x_1 + \delta). \quad (25)$$

Estimation of upper and lower bound of $sd_{x_2}(x_1)$. The upper and lower bound of $sd_{x_2}(x_1)$ in Eq.(24) and Eq.(25) can not be calculated directly because $sd_{x_2}(x_1)$ is not computable in practice. Thus, we estimate Eq.(24) and Eq.(25) with a certain probabilistic guarantee by estimating $sd_{x_2}(x_1)$ with Monte Carlo sampling. Let $\hat{s}d_{x_2}(x_1) = \frac{1}{N} \sum_{\xi_1,\ldots,\xi_N} \frac{1}{2} \|f(x_1+\xi_i) - f(x_2)\|_2$ be estimation of $sd_{x_2}(x_1)$ where $\xi_1, \ldots, \xi_N$ are sampled from $\mathcal{N}(0, \sigma^2 I)$. Then, given query image $q \in X$ and candidate images $C = \{c_i | c_i \in X\}_{i=1}^{N}$, for $\forall c \in C$, the gap between $sd_c(q)$ and $\hat{s}d_c(q)$ (or $sd_q(c)$ and $\hat{s}d_q(c)$) can be guaranteed by utilizing Hoeffding's inequality [38] as follows:

Lemma 3 (Theoretical Guarantee of Gap between $sd_{x_2}(x_1)$ and $\hat{s}d_{x_2}(x_1)$). [†] *Let $q \in X$ and $C = \{c_i | c_i \in X\}_{i=1}^{N}$ be a query image and a set of candidate images, respectively. Let $\beta \in [0, 1]$. For $\forall c \in C$, the following holds with at least probability $1 - \beta$:*

$$\hat{s}d_c(q) + \sqrt{\frac{1}{2N} \log(\frac{2|C|}{\beta})} > sd_c(q) \quad (26)$$

$$\hat{s}d_c(q) - \sqrt{\frac{1}{2N} \log(\frac{2|C|}{\beta})} < sd_c(q). \quad (27)$$

Eq. (26) and Eq. (27) also hold if $\hat{s}d_c(q)$ and $sd_c(q)$ are replaced with $\hat{s}d_q(c)$ and $sd_q(c)$, respectively.

Proof. The proof is shown in A.7 □

Combining Eq.(24), Eq.(25), and Lemma 3, we can estimate the upper and lower bounds of $sd_c(q)/sd_q(c)$ for $\forall c \in C$ with a probability of at least $1 - \beta$ as follows:

Corollary 1 (Upper and Lower Bounds of $sd_{x_2}(x_1)$). [†] *Let $q \in X$ and $C = \{c_i | c_i \in X\}_{i=1}^{N}$ be a query image and a set of candidate images, respectively. Then, for $\forall c \in C$ and $\forall \delta \in \{\delta | \delta \in X, \|\delta\|_2 \leq \epsilon\}$, with at least probability $1 - \beta$, the followings hold:*

$$\Phi(\Phi^{-1}(\hat{s}d_c(q) + t) + \frac{\epsilon}{\sigma}) > \max_{\delta, \|\delta\|_2 \leq \epsilon} sd_c(q + \delta), \quad (28)$$

$$\Phi(\Phi^{-1}(\hat{s}d_c(q) - t) - \frac{\epsilon}{\sigma}) < \min_{\delta, \|\delta\|_2 \leq \epsilon} sd_c(q + \delta), \quad (29)$$

where $t = \sqrt{\frac{1}{2N} \log \frac{2|C|}{\beta}}$. Eq. (28) and Eq. (29) also hold if $\hat{s}d_c(q)$ and $sd_c(q + \delta)$ are replaced with $\hat{s}d_q(c + \delta)$ and $sd_q(c)$, respectively.

Proof. The proof is shown in A.8 □

Eq. (28) and Eq. (29) show the upper and lower bounds converge asymptotically to $\hat{s}d_{x_2}(x_1)$ as the sample size of Gaussian noises N and the standard deviation σ increase.

Estimation of CBIR ranking result with $sd_{x_2}(x_1)$. Since exact $sd_{x_2}(x_1)$ is not computable in practice, the exact CBIR ranking results $\mathrm{Rank}(q, c, C), \forall c \in C$ also cannot be calculated. To evaluate the sufficient conditions Eq.(13) and Eq.(14), we estimate $\mathrm{Rank}(q, c, C), \forall c \in C$ instead of the exact ones. Specifically, we estimate the upper and lower bounds of $\mathrm{Rank}(q, c_i, C), \forall c \in C$ by utilizing Lemma 3. The following theorems estimate the upper and lower bounds of ranking results $\mathrm{Rank}(q, c_i, C), \forall c \in C$ with a probability of at least $1 - \beta$ when we use $sd_c(q)$ and $sd_q(c)$ for computing

[†] [37] derives similar results in the proof of Corollary 1 in their paper. [37] proposes a certified defense for saliency map.

the CBIR ranking results, respectively:

Theorem 6 (Upper and Lower Bounds of Ranking Results with $sd_c(q)$)**.** *Let $q \in X$ and $C = \{c_i | c_i \in X\}_{i=1}^N$ be a query image and a set of candidate images, respectively. Then, for $\forall c_i \in C$, with at least probability $1 - \beta$, the followings hold:*

$$|C| - \sum_{c \in C} \mathbb{1}(\hat{sd}_{c_i}(q) + t < \hat{sd}_c(q) - t) \geq \mathrm{Rank}(q, c_i, C) \tag{30}$$

$$\sum_{c \in C} \mathbb{1}(\hat{sd}_c(q) + t < \hat{sd}_{c_i}(q) - t) + 1 \leq \mathrm{Rank}(q, c_i, C), \tag{31}$$

where $t = \sqrt{\frac{1}{2N} \log \frac{2|C|}{\beta}}$.

Proof. The proof is shown in A.9 □

Theorem 7 (Upper and Lower Bounds of Ranking Result with $sd_q(c)$)**.** *Let $q \in X$ and $C = \{c_i | c_i \in X\}_{i=1}^N$ be a query image and a set of candidate images, respectively. Then, for $\forall c_i \in C$, with at least probability $1 - \beta$, the followings hold:*

$$|C| - \sum_{c \in C} \mathbb{1}(\hat{sd}_q(c_i) + t < \hat{sd}_q(c) - t) \geq \mathrm{Rank}(q, c_i, C) \tag{32}$$

$$\sum_{c \in C} \mathbb{1}(\hat{sd}_q(c) + t < \hat{sd}_q(c_i) - t) + 1 \leq \mathrm{Rank}(q, c_i, C), \tag{33}$$

where $t = \sqrt{\frac{1}{2N} \log \frac{2|C|}{\beta}}$.

Proof. The proof is shown in A.9 □

Theorem 6 and 7 show the upper and lower bounds of ranking results can be tighter as the sample size of Gaussian noises N increases.

Probabilistic evaluation of sufficient conditions. We evaluate the sufficient conditions Eq.(13) and Eq.(14) with a probabilistic guarantee by estimating the upper and lower bounds of $sd_{x_2}(x_1)$ and $\mathrm{Rank}(q, c, C)$ by Corollary 1 and Theorem 6 or Theorem 7, respectively:

Theorem 8 (Probabilistic Evaluation of Sufficient Condition for (l_2, α, ϵ)-Robustness against QA)**.** *At least probability $1 - \beta$, feature extractor f satisfies (α, ϵ)-robust against QA at $c_i \in C$, q, and C if*

$$\alpha \geq \left(|C| - \sum_{c \in C} \mathbb{1}\left[\overline{sd}_{c_i}(q) < \underline{sd}_c(q)\right]\right) - \underline{\mathrm{Rank}}(q, c_i, C) \tag{34}$$
$$\wedge\ \alpha \geq \overline{\mathrm{Rank}}(q, c_i, C) - \left(\sum_{c \in C} \mathbb{1}\left[\overline{sd}_c(q) < \underline{sd}_{c_i}(q)\right] + 1\right),$$

where $\overline{sd}_c(q) = \Phi(\Phi^{-1}(\hat{sd}_c(q) + \sqrt{(1/2N)\log(2|C|/\beta)}) + \frac{\epsilon}{\sigma})$, $\underline{sd}_c(q) = \Phi(\Phi^{-1}(\hat{sd}_c(q) - \sqrt{(1/2N)\log(2|C|/\beta)}) - \frac{\epsilon}{\sigma})$, $\overline{\mathrm{Rank}}(q, c, C) = |C| - \sum_{c \in C} \mathbb{1}(\hat{sd}_{c_i}(q) + t < \hat{sd}_c(q) - t)$, and $\underline{\mathrm{Rank}}(q, c, C) = \sum_{c \in C} \mathbb{1}(\hat{sd}_c(q) + t < \hat{sd}_{c_i}(q) - t) + 1$.

Proof. The proof is shown in A.10. □

Theorem 9 (Probabilistic Evaluation of Sufficient Condition for (α, ϵ)-Robustness against CA)**.** *At least probability $1 - \beta$, feature extractor f satisfies (α, ϵ)-robust against QA at $c_i \in C$, q, and C if*

$$\alpha \geq \left(|C| - \sum_{c \in C} \mathbb{1}\left[\overline{sd}_q(c_i) < \underline{sd}_q(c)\right]\right) - \underline{\mathrm{Rank}}(q, c_i, C) \tag{35}$$
$$\wedge\ \alpha \geq \overline{\mathrm{Rank}}(q, c_i, C) - \left(\sum_{c \in C} \mathbb{1}\left[\overline{sd}_q(c) < \underline{sd}_q(c_i)\right] + 1\right),$$

where $\overline{sd}_q(c_i) = \Phi(\Phi^{-1}(\hat{sd}_q(c_i) + \sqrt{(1/2N)\log(2|C|/\beta)}) + \frac{\epsilon}{\sigma})$, $\underline{sd}_q(c_i) = \Phi(\Phi^{-1}(\hat{sd}_q(c_i) - \sqrt{(1/2N)\log(2|C|/\beta)}) - \frac{\epsilon}{\sigma})$, $\overline{\mathrm{Rank}}(q, c_i, C) = |C| - \sum_{c \in C} \mathbb{1}(\hat{sd}_q(c_i) + t < \hat{sd}_q(c) - t)$, and $\underline{\mathrm{Rank}}(q, c_i, C) = \sum_{c \in C} \mathbb{1}(\hat{sd}_q(c) + t < \hat{sd}_q(c_i) - t) + 1$.

Proof. The proof is shown in Appendix A.11. □

To determine if Eq.(34) or Eq.(35) is satisfied, we can obtain our probabilistic verification algorithms for CBIR against QA and CA as follows:

$$\mathsf{Ver}_{\alpha,\epsilon,\beta}(q, c_i, C) = \begin{cases} \text{True if Eq. (34) or (35) is True} \\ \text{False otherwise.} \end{cases} \tag{36}$$

We omit f from the augments of $\mathsf{Ver}_{\alpha,\epsilon,\beta}(q, c_i, C)$ for notational simplicity when it is obvious from the context.

Different from the deterministic verification algorithms, Eq.(19), the results of probabilistic verification algorithms, Eq.(36), include not only false negatives but also false positives with probability at most β. False negatives in Eq.(36) depend on hyperparameters: β, the sample size of noises N, and standard deviation σ. The total number of forwards propagation of DNN in evaluating Eq.(34) and Eq.(35) is equivalent to $N + |C|$ and $1 + N|C|$, respectively. In Section 6, we experimentally confirm that the probabilistic verification algorithms, Eq.(35), can verify high-resolution images (224 × 224) when we use $\beta = 0.01$ and $N = 100000$.

5.2 Robustness Training Methods

In this subsection, we discuss the robust training method for the feature extractor f, which increases the number of inputs verified by our probabilistic verification algorithm Eq.(36). Recall that tighter evaluation of the upper bound in Eq. (28) and the lower bound in Eq. (29) is needed to attain certified robustness in a meaningful way. From Eq. (28) and Eq. (29), we can obtain a tighter evaluation of the upper and lower bounds when using a larger standard deviation σ. However, when σ is large, obtaining meaningful CBIR ranking results is difficult because the estimated smoothed distances $\hat{sd_c}(q)$ or $\hat{sd_q}(c)$ become indistinguishable for $\forall c \in C$. This is because the distribution difference between the training images of f and the input images with Gaussian noises becomes larger. To mitigate the distribution shift, we can utilize Gaussian data augmentation [23], [27], which adds sampled Gaussian noise $\xi \sim N(0, \sigma^2 I)$ to the training data of feature extractor f. In Section 6, we show that training f with Gaussian data augmentation increases the number of inputs verified by our probabilistic verification algorithms.

6. Experiments

In this section, we evaluate our proposed deterministic and probabilistic certified defenses in terms of CBIR accuracy on clean images and robustness against QA and CA. The robustness is evaluated by *empirical robustness*, which is the CBIR accuracy on the generated AXs, and *certified robustness*, which represents how often given inputs achieve $(l_\infty, \alpha, \epsilon)$-robustness and (l_2, α, ϵ)-robustness by our proposed deterministic and probabilistic verification algorithms, respectively.

6.1 Experiments for deterministic certified defenses

6.1.1 Datasets.

We use the following three image datasets, MNIST [28], Fashion-MNIST (FMNIST) [29], CIFAR10 (CIFAR) [30], for evaluating our deterministic certified defense.

- MNIST is a gray-scale image dataset with 60,000 training images and 10,000 test images. The size of each image is 28 × 28 pixels. There are 10 classes.
- Fashion-MNIST is a gray-scale image dataset with 60,000 training images and 10,000 test images. The size of each image is 28 × 28. There are 10 classes.
- CIFAR is an RGB dataset with 50,000 training samples and 10,000 test samples. The size of each image is 32 × 32 pixels. There are 10 classes.

We train feature extractors f on each training set and evaluate f using each 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. We set $|Q| = 1000$ and $|C| = 1000$ for MNIST, FMNIST, and CIFAR. Pixel values of images in all datasets are in $[0, 1]$.

6.1.2 Evaluation Measures.

Performance of CBIR. To evaluate the performance of CBIR, we use Recall@K, which is one of the evaluation measures for CBIR [39], [40]. Recall@K evaluates whether how often any of the top K candidates is similar to the query image. For evaluation purpose, images belonging to the same class are regarded as similar images. Then, Recall@K is defined as follos:

$$\frac{1}{|Q|} \sum_{(q_i, y_{q_i}) \in Q} \begin{cases} 1 & \text{if } \exists (c, y_c) \in C \text{ s.t.} \\ & \text{Rank}(q_i, c, C) \leq K \wedge y_c = y_{q_i} \\ 0 & \text{otherwise.} \end{cases} \tag{37}$$

Empirical Robustness. To evaluate the empirical robustness against QA and CA, we extend recall@K and define empirical robust Recall@K (ER-Recall@K) against QA and CA. ER-Recall@K against QA represents how often any of the top K candidates is similar to the query image under QA:

$$\frac{1}{|Q|} \sum_{(q_i, y_{q_i}) \in Q} \begin{cases} 1 & \text{if } \exists (c, y_c) \in C \text{ s.t.} \\ & \text{Rank}(q_i + \delta_i, c, C) \leq K \wedge y_c = y_{q_i} \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

where $\delta_1, ..., \delta_{|Q|}$ are adversarial perturbations generated with Eq.(2). We randomly select a single target candidate image $C_t = \{(c_t, y_{c_t})\} \subset C$ such that $y_{c_t} \neq y_{q_i}$ for each $(q_i, y_{q_i}) \in Q$. We minimize Eq.(2) by using PGD [12] with the step size of $\frac{\epsilon}{10}$ and the number of updates of 100, where $\epsilon \in \{0.1, 0.2\}$ for MNIST and FMNIST and $\epsilon \in \{\frac{2}{255}, \frac{3}{255}\}$ for CIFAR10, respectively.

ER-Recall@K against CA represents how often any of the top K candidates is similar image to the query image under CA:

$$\frac{1}{|Q|} \sum_{(q_i, y_{q_i}) \in Q} \begin{cases} 1 & \text{if } \exists (c, y_c) \in \tilde{C} \text{ s.t.} \\ & \text{Rank}(q_i, c, \tilde{C}) \leq K \wedge \ y_c = y_{q_i} \\ 0 & \text{otherwise,} \end{cases} \tag{39}$$

where $\tilde{C} = C \backslash C_s \cup \tilde{C_s}$ and $C_s \subset C$ is a set of source candidate images, and $\tilde{C_s} = \{(c_i + \delta_i, y_{c_i}) | (c_i, y_{c_i}) \in C_s\}_{i=1}^{|C_s|}$ is the set of images obtained by adding adversarial perturbation $\delta_1, ..., \delta_{|C_s|}$ to each image in C_s with Eq.(4). We randomly select 100 source candidate images $C_s = \{(c_i, y_{c_i})\}_{i=1}^{100}$ such that $y_{c_i} \neq y_{q_i}$ for each $(q_i, y_{q_i}) \in Q$. We minimize Eq.(4) using PGD with the same step and perturbation size as the QA.

Certified Robustness. To evaluate the certified robustness, we define an extension of recall@K, certified robust Recall@K (CR-Recall@K). Given a set of query images, this measure evaluates how often (i) the retrieved candidate image by the query image has certified robustness against QA or CA, and (ii) are similar to the query image:

$$\frac{1}{|Q|} \sum_{(q, y_q) \in Q} \begin{cases} 1 & \text{if } \exists (c, y_c) \in C \text{ s.t. } y_c = y_q \ \wedge \\ & \text{Rank}(q_i, c, C) \leq K \wedge \mathsf{Ver}_{\alpha,\epsilon}(q, c, C) \\ 0 & \text{otherwise,} \end{cases} \tag{40}$$

where $\mathsf{Ver}_{\alpha,\epsilon}(q, c, C)$ is deterministic verification algorithms defined by Eq. (19). We use $\alpha = K - \text{Rank}(q, c, C)$ for each $c \in C$. Then, $\mathsf{Ver}_{\alpha,\epsilon}(q, c, C)$ verifies whether c still satisfies $\text{Rank}(q_i, c, C) \leq K$ under QA and CA.

Table 1 Comparison of Recall@K for deterministic certifed defense. Each value is rounded off to two decimal places.

	MNIST			FMNIST			CIFAR10		
K	1	10	40	1	10	40	1	10	40
Triplet	0.99	1.00	1.00	0.89	0.98	0.99	0.58	0.93	0.99
ACT	0.99	1.00	1.00	0.83	0.97	0.99	0.63	0.93	0.99
C-IBP	0.97	0.99	1.00	0.75	0.96	0.99	0.39	0.87	0.98
TBT	0.94	0.98	0.99	0.62	0.93	0.98	0.18	0.81	0.97
TBT+FCTB	0.93	0.98	0.99	0.64	0.94	0.98	0.19	0.82	0.97

Table 2 Comparison of empirical robust (ER) Recall@K and certified robust (CR) Recall@K for deterministic certified defense. QA and CA represent query and candidate attack, respectively. 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	40	1	10	40	1	10	40	1	10	40
MNIST	$\epsilon = 0.1$	Triplet	0.00	0.12	0.27	0.00	0.00	0.00	0.25	0.60	0.81	0.00	0.00	0.00
		ACT	0.99	1.00	1.00	0.00	0.00	0.00	0.99	1.00	1.00	0.00	0.00	0.00
		C-IBP	0.97	0.99	1.00	0.00	0.04	0.29	0.96	0.99	1.00	0.00	0.01	0.29
		TBT	0.94	0.98	0.99	0.15	0.66	0.89	0.94	0.98	0.99	0.12	0.92	0.98
		TBT+FCTB	0.93	0.98	0.99	0.16	0.66	0.89	0.93	0.98	0.99	0.12	0.92	0.98
	$\epsilon = 0.2$	Triplet	0.00	0.05	0.14	0.00	0.00	0.00	0.21	0.38	0.58	0.00	0.00	0.00
		ACT	0.97	0.99	1.00	0.00	0.00	0.00	0.98	0.99	1.00	0.00	0.00	0.00
		C-IBP	0.97	0.99	1.00	0.00	0.00	0.01	0.96	0.99	1.00	0.00	0.00	0.00
		TBT	0.92	0.98	0.99	0.03	0.31	0.65	0.93	0.98	0.99	0.01	0.42	0.95
		TBT+FCTB	0.92	0.97	0.99	0.03	0.30	0.64	0.92	0.98	0.99	0.02	0.48	0.96
FMNIST	$\epsilon = 0.1$	Triplet	0.00	0.11	0.22	0.00	0.00	0.00	0.03	0.09	0.17	0.00	0.00	0.00
		ACT	0.80	0.97	0.99	0.00	0.00	0.00	0.72	0.96	0.99	0.00	0.00	0.00
		C-IBP	0.72	0.97	0.99	0.01	0.16	0.42	0.71	0.96	0.99	0.00	0.06	0.49
		TBT	0.61	0.93	0.98	0.11	0.44	0.71	0.59	0.93	0.98	0.01	0.49	0.94
		TBT+FCTB	0.63	0.93	0.99	0.11	0.47	0.70	0.61	0.93	0.98	0.02	0.47	0.94
	$\epsilon = 0.2$	Triplet	0.00	0.09	0.20	0.00	0.00	0.00	0.04	0.11	0.19	0.00	0.00	0.00
		ACT	0.74	0.95	0.98	0.00	0.00	0.00	0.57	0.92	0.99	0.00	0.00	0.00
		C-IBP	0.71	0.96	0.99	0.00	0.01	0.08	0.67	0.95	0.99	0.00	0.00	0.03
		TBT	0.59	0.93	0.98	0.02	0.20	0.45	0.55	0.93	0.98	0.00	0.07	0.64
		TBT+FCTB	0.60	0.93	0.99	0.02	0.22	0.44	0.55	0.93	0.98	0.00	0.09	0.62
CIFAR10	$\epsilon = \frac{2}{255}$	Triplet	0.19	0.70	0.89	0.00	0.00	0.00	0.00	0.09	0.58	0.00	0.00	0.00
		ACT	0.47	0.88	0.97	0.00	0.00	0.00	0.22	0.72	0.96	0.00	0.00	0.00
		C-IBP	0.40	0.87	0.98	0.00	0.04	0.23	0.35	0.84	0.98	0.00	0.02	0.21
		TBT	0.20	0.79	0.97	0.02	0.18	0.48	0.15	0.78	0.96	0.00	0.19	0.58
		TBT+FCTB	0.19	0.81	0.97	0.02	0.20	0.48	0.17	0.80	0.97	0.01	0.21	0.70
	$\epsilon = \frac{3}{255}$	Triplet	0.07	0.56	0.80	0.00	0.00	0.00	0.00	0.00	0.17	0.00	0.00	0.00
		ACT	0.36	0.81	0.95	0.00	0.00	0.00	0.07	0.45	0.91	0.00	0.00	0.00
		C-IBP	0.40	0.87	0.98	0.00	0.01	0.06	0.32	0.83	0.98	0.00	0.01	0.06
		TBT	0.19	0.78	0.96	0.01	0.12	0.33	0.14	0.77	0.96	0.00	0.09	0.38
		TBT+FCTB	0.20	0.82	0.97	0.01	0.12	0.36	0.15	0.78	0.97	0.00	0.11	0.48

6.1.3 Comparison Methods

We compare our proposed robustness training Eq. (20) (TBT) and Eq. (21) (FCTB) with three existing methods: (i) triplet Loss (Triplet) [36], (ii) anti-collapse triplet (ACT), which is an adversarial training for CBIR to improve empirical robustness [3], (iii) robust training for classification using interval bound propagation (C-IBP) to improve certified robustness of classification task [21].

We use Triplet as a baseline which does not have any mechanism for robustness. We compare TBT and FCTB with ACT to show that adversarial training is not sufficient to improve certified robustness of CBIR. We also compare TBT and FCTB with C-IBP to show that robust training for improving certified robustness for the classification task is inadequate to improve certified robustness for CBIR. Detail of each method are explained in Appendix B.

6.1.4 Implementations of Feature extractors

Architectures. In our experiments for probabilistic certified defense, we train the feature extractor f of embedding dimensionality 128 in the following 6-layer CNN model architecture as $\mathrm{Conv}(64, 3, 1, 1) \to \mathrm{Conv}(64, 3, 1, 1) \to \mathrm{Conv}(128, 3, 2, 1) \to \mathrm{Conv}(128, 3, 1, 1) \to \mathrm{Conv}(128, 3, 1, 1)$

→ Linear(128) where Conv(c, k, s, p) denotes a convolutional layer with a number of filters c of size $k \times k$, stride size is s, and padding size is p and Linear(d) denotes a linear layer whose output dimension is d. Note that there is a ReLU between each layer.

Hyperparameters. The total number of training epochs of f is 100 for MNIST and FMNIST and 200 for CIFAR10. We use the Adam optimizer [41] 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 $m = 1.0$. When training with TBT, to stabilize training, we use the scheduling strategy for ϵ and κ proposed in [21]. 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 all datasets. Then, we linearly increase ϵ and decrease κ between $2K$ and $10K$ steps. 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. Other hyperparameters are shown in Appendix B.1.

6.2 Results for deterministic certified defenses

Table 1 and Table 2 show the results for deterministic certified defense proposed in Section 4. We can see that our proposed robust training TBT has less Recall@K than Triplet, ACT, and C-IBP from Table 1. This is presumably due to the fact that the diversity of feature representation is reduced by making the upper and lower bound evaluated tighter. However, the gap in Recall@K between TBT and the existing methods becomes smaller as K increases. Thus, that is not a practical problem in situations where K is large.

We can also confirm both ER-Recall@K and CR-Recall@K of Triplet are significantly lower than the other methods from Table 2. C-IBP and ACT achieve higher ER-Recall@K than Triplet, while their CR-Recall@K is zero or nearly zero, even with larger K. This implies that C-IBP and ACT cannot help to provide certified robustness of CBIR. This is because ACT is training to improve empirical robustness, which is no enough to improve certified robustness. We also conjecture that C-IBP is not sufficient to tighten Eq. (17) and Eq. (18) since it aims at tightening the upper and lower bounds of logits. In contrast, TBT achieves significantly higher CR-recall@K, particularly when K is large. This is because TBT can tighten Eq. (17) and Eq. (18) successfully.

We can also confirm that fine-tuning pre-trained feature extractor with TBT to candidate images (FCTB) improve CR-Recall@K while maintaining Recall@K from Table 1 and Table 2. This implies that FCTB can further reduce the gap between Definition 3 and the corresponding sufficient condition.

Limitations of deterministic certified defense. A drawback of our deterministic verification algorithms are that it does not scale to high-resolution images, which require advanced architecture. We train feature extractor with TBT using CUB and VGG architecture [42]. As a result, its training collapses, which means that the trained feature extractor returns the same value for all test inputs. This is because IBP provides very loose bounds for advanced deep architectures, resulting in extremely large regularization terms in Eq.(20).

6.3 Experiments for probabilistic certified defenses

6.3.1 Experimental settings

Datasets. We use the following three image datasets, CUB-200-2011 (CUB) [31], CARS196 (CAR) [43], and Stanford Online Products (SOP) [44], for evaluating our probabilistic certified defense.

- CUB is an RGB dataset with 11, 788 images labeled with one of 200 classes. We split this dataset into training and test images for the first and last 100 classes.
- CAR is an RGB dataset with 16, 185 images labeled with one of 196 classes. We split this dataset into training and test images for the first and last 98 classes.
- SOP is an RGB dataset with 11, 318 classes 59, 551 training images and 11, 316 classes 60, 501 test images.

We crop and resize the images in these datasets to 224×224, which is a higher resolution than MNIST, FMNIST, and CIFAR10. We train feature extractors f on each training set and evaluate f using each 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. We set $|Q| = 100$ and $|C| = 1000$ for CUB, CAR, and SOP. Pixel values of images in all datasets are in $[0, 1]$.

6.3.2 Evaluation Measures.

Performance of CBIR. To evaluate the performance of CBIR, we use the lower bound of Recall@K Eq. (37). Since the exact Rank(q, c, C) is not computationally tractable when using Gaussian smoothed distance Eq. (23), we cannot also evaluate exact Recall@K. Thus, we evaluate the lower bound of Recall@K by estimating the upper bound of Rank(q, c, C) by Theorem 6 or Theorem 7:

$$\frac{1}{|Q|} \sum_{(q_i, y_{q_i}) \in Q} \begin{cases} 1 & \text{if } \exists (c, y_c) \in C \text{ s.t.} \\ & \overline{\mathrm{Rank}}(q, c, C) \le K \wedge y_c = y_{q_i} \\ 0 & \text{otherwise,} \end{cases} \tag{41}$$

where $\overline{\mathrm{Rank}}(q, c, C)$ is the upper bound of Rank(q, c, C) evaluated by Eq.(30) or Eq.(32).

Certified Robustness. To evaluate the certified robustness, we use the lower bound of CR-Recall@K Eq. (40).

Table 3 Comparison of the lower bounds of Recall@K and certified robust (CR) Recall@K for probabilistic certified defense. All results use $N = 100000$ and $\beta = 0.01$. Note that the results for $\sigma = 0.0$ are a baseline that represents Recall@K when using general Euclidean distance for calculating ranking results. QA and CA represent query and candidate attack, respectively.

				Recall@K (QA)			CR-Recall@K (QA)			Recall@K (CA)			CR-Recall@K (CA)		
		K		1	50	300	1	50	300	1	50	300	1	50	300
CUB		Triplet (Baseline)	$\sigma = 0.0$	0.45	0.96	1.00	-	-	-	0.45	0.96	1.00	-	-	-
	$\epsilon = 0.01$	Triplet	$\sigma = 1.0$	0.00	0.09	0.52	0.00	0.05	0.27	0.00	0.00	0.00	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.03	0.42	0.85	0.02	0.25	0.65	0.00	0.42	0.80	0.00	0.28	0.60
	$\epsilon = 0.05$	Triplet	$\sigma = 1.0$	0.00	0.09	0.52	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.03	0.42	0.85	0.01	0.14	0.42	0.00	0.42	0.80	0.00	0.16	0.39
CAR		Triplet (Baseline)	$\sigma = 0.0$	0.56	0.96	0.98	-	-	-	0.56	0.96	0.98	-	-	-
	$\epsilon = 0.01$	Triplet	$\sigma = 1.0$	0.00	0.07	0.52	0.00	0.00	0.23	0.00	0.00	0.04	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.01	0.30	0.74	0.00	0.20	0.44	0.00	0.22	0.64	0.00	0.04	0.45
	$\epsilon = 0.05$	Triplet	$\sigma = 1.0$	0.00	0.07	0.52	0.00	0.00	0.02	0.00	0.00	0.04	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.01	0.30	0.74	0.00	0.13	0.32	0.00	0.22	0.64	0.00	0.01	0.30
SOP		Triplet (Baseline)	$\sigma = 0.0$	0.75	1.00	1.00	-	-	-	0.75	1.00	1.00	-	-	-
	$\epsilon = 0.01$	Triplet	$\sigma = 1.0$	0.00	0.06	0.21	0.00	0.02	0.10	0.00	0.00	0.02	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.06	0.33	0.62	0.03	0.24	0.42	0.01	0.26	0.61	0.00	0.19	0.39
	$\epsilon = 0.05$	Triplet	$\sigma = 1.0$	0.00	0.06	0.21	0.00	0.00	0.05	0.00	0.00	0.02	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.06	0.33	0.62	0.00	0.06	0.21	0.01	0.26	0.61	0.00	0.06	0.23

As well as the evaluation of the lower bound of Recall@K Eq.(41), we use the upper bound of $\mathrm{Rank}(q, c, C)$ to calculate the lower bound of CR-Recall@K as follows:

$$\frac{1}{|Q|}\sum_{(q,y_q)\in Q}\begin{cases}1 \ \text{if } \exists(c, y_c)\in C \ \text{s.t.} \ y_c = y_q \ \wedge \\ \overline{\mathrm{Rank}}(q,c,C) \le K \wedge \mathsf{Ver}_{\alpha,\epsilon,\beta}(q,c,C) \\ 0 \ \text{otherwise,}\end{cases} \tag{42}$$

where $\mathsf{Ver}_{\alpha,\epsilon,\beta}(q,c,C)$ is probabilistic verification algorithms defined by Eq. (36). We use $\alpha = K - \mathrm{Rank}(q, c, C)$ for each $c \in C$. Then, $\mathsf{Ver}_{\alpha,\epsilon,\beta}(q,c,C)$ verifies whether c still satisfies $\mathrm{Rank}_f(q_i, c, C) \le K$ under QA and CA. We set the false positive rate β to 0.01 for all experiments of probabilistic certified defenses.

6.4 Implementations of Feature extractors

Architectures. We use ResNet50 [45] as model architecture of f. We use the parameters pre-trained on ImageNet [34] obtained from torchvision library in PyTorch [46] as initial parameters. We set the feature dimension is $d = 128$. We use the codes† provided by [47] for training f.

Hyperparameters. We use Triplet loss with margin $m = 0.2$ as the loss function. The total number of training epochs is 150. We use the Adam optimizer [41] with a batch size of 112, an initial learning rate of 0.00001, and a weight decay of 0.0004. We decay the learning rate by times 0.3 at 1000 iterations. When using Gaussian data augmentation, we sample noise from $N(0, \sigma^2 I)$ where $\sigma \in \{0.1, 1.0\}$ at each parameter update and added to the training data. Then, We use the same σ during training and testing.

6.5 Results for probabilistic certified defenses

Table 3 shows the results of the lower bounds of Recall@K and CR-Recall@K for probabilistic certified defense proposed in Section 5. All results in Table 3 use $N = 100000$. Triplet and GA-Triplet represent the results of models trained with Triplet Loss and Triplet Loss with Gaussian data augmentation, respectively. Note that the results for $\sigma = 0.0$ are a baseline that represents Recall@K when using general Euclidean distance for calculating ranking results. From Table 3, we can see that GA-Triplet has higher lower bounds of Recall@K and CR-Recall@K than Triplet. These are due to the fact that Gaussian data augmentation successfully mitigates the distribution shift between the training and test data of feature extractor f, allowing f to compute a meaningfully distinguishable estimated smoothed distance $\hat{sd}_c(q)$ or $\hat{sd}_q(c)$ for $\forall c \in C$. GA-Triplet has less Recall@K than the baseline, but the gap gets smaller as K increases. Thus, that is not a practical problem when K is large.

The effect of standard deviation σ. Table 4 shows the results of the lower bounds of Recall@K and CR-Recall@K for GA-Triplet with different standard deviation σ. All results in Table 3 use $N = 100000$. We can see that σ controls the trade-off between Recall@K and CR-Recall@K. This is because as σ increases, the upper and lower bounds of the smooth distance in Eq. (28) and Eq. (29) become smaller, but it becomes more difficult to compute a meaningfully distinguishable estimated smoothed distance $\hat{sd}_c(q)$ or $\hat{sd}_q(c)$ for $\forall c \in C$.

The effect of sample sizes of Gaussian noises N. Table 5 shows the results of the lower bounds of Recall@K and CR-Recall@K for GA-Triplet with different sample sizes of

†`https://github.com/Confusezius/Revisiting_Deep_Metric_Learning_PyTorch`

Table 4 Comparison of the lower bounds of Recall@K and certified robust (CR) Recall@K for probabilistic certified defense with different standard deviation σ. All results use $N = 100000$ and $\beta = 0.01$. QA and CA represent query and candidate attack, respectively.

				Recall@K (QA)			CR-Recall@K (QA)			Recall@K (CA)			CR-Recall@K (CA)		
		K		1	50	300	1	50	300	1	50	300	1	50	300
CUB	$\epsilon = 0.01$	GA-Triplet	$\sigma = 0.1$	0.14	0.88	1.00	0.00	0.19	0.60	0.13	0.84	0.99	0.00	0.27	0.66
		GA-Triplet	$\sigma = 1.0$	0.03	0.42	0.85	0.02	0.25	0.65	0.00	0.42	0.80	0.00	0.28	0.60
	$\epsilon = 0.05$	GA-Triplet	$\sigma = 0.1$	0.14	0.88	1.00	0.00	0.00	0.00	0.13	0.84	0.99	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.03	0.42	0.85	0.01	0.14	0.42	0.00	0.42	0.80	0.00	0.16	0.39
CAR	$\epsilon = 0.01$	GA-Triplet	$\sigma = 0.1$	0.27	0.91	0.99	0.01	0.41	0.76	0.17	0.84	0.98	0.01	0.37	0.72
		GA-Triplet	$\sigma = 1.0$	0.01	0.30	0.74	0.00	0.20	0.44	0.00	0.22	0.64	0.00	0.04	0.45
	$\epsilon = 0.05$	GA-Triplet	$\sigma = 0.1$	0.00	0.07	0.52	0.00	0.00	0.00	0.17	0.84	0.98	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.01	0.30	0.74	0.00	0.13	0.32	0.00	0.22	0.64	0.00	0.01	0.30
SOP	$\epsilon = 0.01$	GA-Triplet	$\sigma = 0.1$	0.19	0.58	0.86	0.03	0.16	0.39	0.10	0.54	0.78	0.01	0.19	0.47
		GA-Triplet	$\sigma = 1.0$	0.06	0.33	0.62	0.03	0.24	0.42	0.01	0.26	0.61	0.00	0.19	0.39
	$\epsilon = 0.05$	GA-Triplet	$\sigma = 0.1$	0.19	0.58	0.86	0.00	0.00	0.00	0.10	0.54	0.78	0.00	0.00	0.00
		GA-Triplet	$\sigma = 1.0$	0.06	0.33	0.62	0.00	0.06	0.21	0.01	0.26	0.61	0.00	0.06	0.23

Table 5 Comparison of the lower bounds of Recall@K and certified robust (CR) Recall@K for probabilistic certified defense with different sample sizes of Gaussian noises N. All results use $\sigma = 1.0$ and $\beta = 0.01$. QA and CA represent query and candidate attack, respectively.

				Recall@K (QA)			CR-Recall@K (QA)			Recall@K (CA)			CR-Recall@K (CA)		
		K		1	50	300	1	50	300	1	50	300	1	50	300
CUB	$\epsilon = 0.01$	GA-Triplet	$N = 100000$	0.03	0.42	0.85	0.02	0.25	0.65	0.00	0.42	0.80	0.00	0.28	0.60
		GA-Triplet	$N = 1000000$	0.04	0.61	0.92	0.03	0.40	0.82	0.08	0.58	0.94	0.01	0.41	0.80
	$\epsilon = 0.05$	GA-Triplet	$N = 100000$	0.03	0.42	0.85	0.01	0.14	0.42	0.00	0.42	0.80	0.00	0.16	0.39
		GA-Triplet	$N = 1000000$	0.04	0.61	0.92	0.01	0.19	0.54	0.08	0.58	0.94	0.00	0.21	0.47
CAR	$\epsilon = 0.01$	GA-Triplet	$N = 100000$	0.01	0.30	0.74	0.00	0.20	0.44	0.00	0.22	0.64	0.00	0.04	0.45
		GA-Triplet	$N = 1000000$	0.06	0.43	0.88	0.03	0.30	0.72	0.00	0.33	0.82	0.00	0.20	0.62
	$\epsilon = 0.05$	GA-Triplet	$N = 100000$	0.01	0.30	0.74	0.00	0.13	0.32	0.00	0.22	0.64	0.00	0.01	0.30
		GA-Triplet	$N = 1000000$	0.06	0.43	0.88	0.00	0.15	0.40	0.00	0.33	0.82	0.00	0.03	0.34
SOP	$\epsilon = 0.01$	GA-Triplet	$N = 100000$	0.06	0.33	0.62	0.03	0.24	0.42	0.01	0.26	0.61	0.00	0.19	0.39
		GA-Triplet	$N = 1000000$	0.11	0.47	0.76	0.07	0.33	0.61	0.01	0.33	0.71	0.01	0.25	0.56
	$\epsilon = 0.05$	GA-Triplet	$N = 100000$	0.06	0.33	0.62	0.00	0.06	0.21	0.01	0.26	0.61	0.00	0.06	0.23
		GA-Triplet	$N = 1000000$	0.11	0.47	0.76	0.01	0.09	0.27	0.01	0.33	0.71	0.0	0.10	0.29

Gaussian noises N. All results in Table 5 use $\sigma = 1.0$. We can see that Recall@K and CR-Recall@K improve as N increases. This is because the upper and lower bounds of Gaussian smoothed distances and ranking results can be tight as the sample size of Gaussian noises N, as explained in Section 5. However, the computational complexity increases as N increases.

Limitations of probabilistic certified defense. A drawback of our probabilistic verification algorithm is its high computational cost. As denoted in Section 5, in the probabilistic verification algorithms, the total number of forward propagation of DNN in evaluating Eq.(34) and Eq.(35) is equivalent to $N + |C|$ and $1 + N|C|$, respectively. For example, we require a total of 101000 and 100000001 forwards to evaluate Eq.(34) and Eq.(35) when $N = 100000$ and $|C| = 1000$.

7. Future Directions

In this section, we present potential directions for future research focusing on enhancing the certified defense for CBIR. Specifically, we delve into possibilities for improving robust training, reducing the computational complexity of the verification algorithms, and implementing our certified defenses in real-world CBIR systems.

The primary direction is further investigation into more effective robust training of feature extractors, with the aim of enhancing the certified robustness of CBIR. One potential method that could be considered is utilizing adversarial training, which trains DNNs using AXs as training data. [25] has demonstrated that adversarial training can enhance not only empirical robustness but also certified robustness in the classification setting. It is reasonable to explore the application of adversarial training to the training of feature extractors for CBIR, with the expectation of boosting its certified robustness.

Another important direction is to reduce the computational complexity of the verification algorithms used to evaluate the certified robustness of CBIR. As described in Section 5, our probabilistic verification algorithms scale to high-resolution images but bear a high computational cost, especially when the number of candidate images |C| is large. This may limit the practical application of our method in real-world scenarios. One potential solution to mitigate this limitation is to utilize an additional distribution in randomized smoothing. [48] has shown that using an additional distribution can yield higher certified robustness of classification, even with a smaller sampling size (i.e., fewer classification model forwards). Applying this theory and computational methodology to the verification algorithms for CBIR may not be straightforward, but it certainly holds promising potential.

A third vital direction is to assess the effectiveness of our proposed certified defenses in real-world CBIR systems, such as person re-identification, medical image retrieval, and image-based product search. The application of our proposed method to these systems could provide the theoretically guaranteed verification results on ranking invariance against AXs associated with retrieved images, thereby potentially improving the security of these systems. Nonetheless, it ’ s important to note that in real-world CBIR systems, a domain gap between the training and evaluation data often becomes larger due to the frequent addition of new images to the candidate images (e.g., the addition of new products in image-based product search). The impact of such domain gap on the performance of our proposed certified defenses remains largely unexplored and thus is need for further assessment. A detailed investigation into this matter may reveal unique challenges and solutions for deploying certified defenses in real-world CBIR systems.

8. Conclusion

In this study, we proposed deterministic and probabilistic certified defenses for CBIR. Our certified defenses provide certified robustness, which deterministically and probabilistically guarantees that no AX that largely changes the ranking of CBIR exists around the query or candidate images. We theoretically and experimentally confirmed that our deterministic certified defenses are lightweight but do not scale to high-resolution images; on the other hand, the probabilistic one is computationally expensive but scales to high-resolution images. Developing a certified defense that is lightweight and scales to high-resolution images is a future work.

Acknowledgment

This work is partly supported by Japan science and technology agency (JST), CREST JPMJCR23M4, and Japan society for the promotion of science (JSPS), Grantr-in-aid 23H00483.

References

[1] S.R. Dubey, “A decade survey of content based image retrieval using deep learning,” IEEE Transactions on Circuits and Systems for Video Technology, 2021.

[2] M. Zhou, Z. Niu, L. Wang, Q. Zhang, and G. Hua, “Adversarial ranking attack and defense,” Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, pp.781–799, Springer, 2020.

[3] M. Zhou, L. Wang, Z. Niu, Q. Zhang, N. Zheng, and G. Hua, “Adversarial attack and defense in deep ranking,” arXiv preprint arXiv:2106.03614, 2021.

[4] J. Li, R. Ji, H. Liu, X. Hong, Y. Gao, and Q. Tian, “Universal perturbation attack against image retrieval,” Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.4899–4908, 2019.

[5] G. Tolias, F. Radenovic, and O. Chum, “Targeted mismatch adversarial attack: Query with a flower to retrieve the tower,” Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.5037–5046, 2019.

[6] H. Wang, G. Wang, Y. Li, D. Zhang, and L. Lin, “Transferable, controllable, and inconspicuous adversarial attacks on person re-identification with deep mis-ranking,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.342–351, 2020.

[7] M. Zhou, L. Wang, Z. Niu, Q. Zhang, Y. Xu, N. Zheng, and G. Hua, “Practical relative order attack in deep ranking,” Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.16413–16422, 2021.

[8] X. Li, J. Li, Y. Chen, S. Ye, Y. He, S. Wang, H. Su, and H. Xue, “Qair: Practical query-efficient black-box attacks for image retrieval,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.3330–3339, 2021.

[9] Z. Wang, S. Zheng, M. Song, Q. Wang, A. Rahimpour, and H. Qi, “advpattern: physical-world attacks on deep person re-identification via adversarially transformable patterns,” Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.8341–8350, 2019.

[10] S. Bai, Y. Li, Y. Zhou, Q. Li, and P.H. Torr, “Adversarial metric attack and defense for person re-identification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.43, no.6, pp.2119–2126, 2020.

[11] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S.C. Hoi, “Deep learning for person re-identification: A survey and outlook,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[12] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” International Conference on Learning Representations, 2018.

[13] F. Tramer, N. Carlini, W. Brendel, and A. Madry, “On adaptive attacks to adversarial example defenses,” Advances in Neural Information Processing Systems, vol.33, pp.1633–1645, 2020.

[14] L. Li, X. Qi, T. Xie, and B. Li, “Sok: Certified robustness for deep neural networks,” arXiv preprint arXiv:2009.04131, 2020.

[15] G. Katz, C. Barrett, D.L. Dill, K. Julian, and M.J. Kochenderfer, “Reluplex: An efficient smt solver for verifying deep neural networks,” International Conference on Computer Aided Verification, pp.97–117, Springer, 2017.

[16] L. Weng, H. Zhang, H. Chen, Z. Song, C.J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, “Towards fast computation of certified robustness for relu networks,” International Conference on Machine Learning, pp.5276–5285, PMLR, 2018.

[17] L. Weng, H. Zhang, H. Chen, Z. Song, C.J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, “Towards fast computation of certified robustness for relu networks,” International Conference on Machine Learning, pp.5276–5285, PMLR, 2018.

[18] H. Zhang, T.W. Weng, P.Y. Chen, C.J. Hsieh, and L. Daniel, “Efficient neural network robustness certification with general activation func-

tions," Advances in neural information processing systems, vol.31, 2018.

[19] G. Singh, T. Gehr, M. Püschel, and M. Vechev, "An abstract domain for certifying neural networks," Proceedings of the ACM on Programming Languages, vol.3, no.POPL, pp.1–30, 2019.

[20] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks," Advances in neural information processing systems, vol.31, 2018.

[21] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, "On the effectiveness of interval bound propagation for training verifiably robust models," arXiv preprint arXiv:1810.12715, 2018.

[22] S. Gowal, K.D. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, "Scalable verified training for provably robust image classification," Proceedings of the IEEE/CVF International Conference on Computer Vision, pp.4842–4851, 2019.

[23] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," International Conference on Machine Learning, pp.1310–1320, PMLR, 2019.

[24] K. Leino, Z. Wang, and M. Fredrikson, "Globally-robust neural networks," International Conference on Machine Learning, pp.6212–6222, PMLR, 2021.

[25] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang, "Provably robust deep learning via adversarially trained smoothed classifiers," Advances in Neural Information Processing Systems, vol.32, 2019.

[26] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," 2019 IEEE symposium on security and privacy (SP), pp.656–672, IEEE, 2019.

[27] Y. Wu, H. Zhang, and H. Huang, "Retrievalguard: Provably robust 1-nearest neighbor image retrieval," International Conference on Machine Learning, pp.24266–24279, PMLR, 2022.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol.86, no.11, pp.2278–2324, 1998.

[29] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," arXiv preprint arXiv:1708.07747, 2017.

[30] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[32] K. Kakizaki, K. Fukuchi, and J. Sakuma, "Certified defense for content based image retrieval," Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp.4561–4570, January 2023.

[33] I.J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[34] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," 2009 IEEE conference on computer vision and pattern recognition, pp.248–255, Ieee, 2009.

[35] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, and C.J. Hsieh, "Towards stable and efficient training of verifiably robust neural networks," International Conference on Learning Representations, 2019.

[36] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.815–823, 2015.

[37] A. Levine, S. Singla, and S. Feizi, "Certifiably robust interpretation in deep learning," arXiv preprint arXiv:1905.12105, 2019.

[38] W. Hoeffding, "Probability inequalities for sums of bounded random variables," Journal of the American Statistical Association, pp.13–30, 1963.

[39] E. Ramzi, N. Thome, C. Rambour, N. Audebert, and X. Bitot, "Robust and decomposable average precision for image retrieval," Advances in Neural Information Processing Systems, vol.34, 2021.

[40] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," Proceedings of the IEEE international conference on computer vision, pp.2593–2601, 2017.

[41] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[43] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia, 2013.

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

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," Advances in neural information processing systems, vol.32, 2019.

[47] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J.P. Cohen, "Revisiting training strategies and generalization performance in deep metric learning," 2020.

[48] L. Li, J. Zhang, T. Xie, and B. Li, "Double sampling randomized smoothing," International Conference on Machine Learning, pp.13163–13208, PMLR, 2022.

Appendix A: Proofs of Theorems

A.1 Proof of Lemma 1

Proof. From definitions of Eq. (7) and Eq. (8), for $\forall\delta \in \{\delta \mid \delta \in X, \|\delta\|_p \le \epsilon\}$, the following holds:

$$\sum_{c\in C} \mathbb{1}\left[d(f(q+\delta), f(c_i)) < d(f(q+\delta), f(c))\right] \geq \sum_{c\in C} \mathbb{1}\left[\overline{d}_{c_i}(q) < \underline{d}_c(q)\right], \quad \text{(A$\cdot$ 1)}$$

$$\sum_{c\in C} \mathbb{1}\left[d(f(q+\delta), f(c)) < d(f(q+\delta), f(c_i))\right] \geq \sum_{c\in C} \mathbb{1}\left[\overline{d}_{c}(q) < \underline{d}_{c_i}(q)\right]. \quad \text{(A$\cdot$ 2)}$$

Eq. (A$\cdot$ 1) and Eq. (A$\cdot$ 2) represent the lower bounds of the number of candidate images that are more dissimilar and similar to $q + \delta$ than c_i, respectively. Thus, we get the claim. □

A.2 Proof of Lemma 2

Proof. From Eq. (7) and Eq. (8), for $\tilde{C} = \{c_i + \delta_i\}_{i=1}^{N}$ where $\forall\delta_1, ..., \delta_N \in \{\delta \mid \delta \in X, \|\delta\|_p \le \epsilon\}$, the following holds:

$$\sum_{c+\delta\in\tilde{C}} \mathbb{1}\left[d(f(q), f(c_i+\delta_i)) < d(f(q), f(c+\delta))\right]$$

$$\geq \sum_{c \in C} \mathbb{1}[\overline{d}_q(c_i) < \underline{d}_q(c)], \tag{A· 3}$$

$$\sum_{c+\delta \in \hat{C}} \mathbb{1}[d(f(q), f(c+\delta)) < d(f(q), f(c_i+\delta_i))] \geq \sum_{c \in C} \mathbb{1}[\overline{d}_q(c) < \underline{d}_q(c_i)]. \tag{A· 4}$$

Eq. (A· 3) and Eq. (A· 4) represent the lower bounds of the number of perturbed candidate images in $\tilde{C}$ that are more dissimilar and similar to q than $c_i + \delta_i$, respectively. Thus, we get the claim. □

A.3 Proof of Theorem 1

Proof. When Eq. (13) is satisfied, from Theorem 1,

$$-\alpha \leq \mathrm{Rank}(q+\delta, c_i, C) - \mathrm{Rank}(q, c_i, C) \leq \alpha$$

holds for $\forall \delta \in \{\delta \mid \delta \in X, \|\delta\|_p \leq \epsilon\}$, which is equivalent to Eq. (5). □

A.4 Proof of Theorem 2

Proof. When Eq. (14) is satisfied, from Theorem 2,

$$-\alpha \leq \mathrm{Rank}(q, c_i+\delta_i, \tilde{C}) - \mathrm{Rank}(q, c_i, C) \leq \alpha$$

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

A.5 Proof of Theorem 3

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

$$\max_{\delta \in X, \|\delta\|_\infty \leq \epsilon} \Big(\sum_{i \in \{1,..,d\}} (f(x_1+\delta)_i - f(x_2)_i)^2\Big)^{\frac{1}{2}} \leq \Big(\sum_{i \in \{1,..,d\}} \max\{|\overline{f}(x_1)_i - f(x_2)_i|, |f(x_2)_i - \underline{f}(x_1)_i|\}^2\Big)^{\frac{1}{2}}. \tag{A· 5}$$

□

A.6 Proof of Theorem 4

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

$$\min_{\delta \in X, \|\delta\|_\infty \leq \epsilon} \Big(\sum_{i \in \{1,..,d\}} (f(x_1+\delta)_i - f(x_2)_i)^2\Big)^{\frac{1}{2}} \geq \Big(\sum_{i \in \{1,..,d\}} \min\{0, \overline{f}(x_1)_i - f(x_2)_i, f(x_2)_i - \underline{f}(x_1)_i\}^2\Big)^{\frac{1}{2}}. \tag{A· 6}$$

□

A.7 Proof of Lemma 3

Proof. Since $0 \leq sd_{x_2}(x_1) \leq 1$ for $\forall x_1, x_2 \in X$ from definition of smoothed distance Eq. (23), the following holds for any $t > 0$ by Hoeffding's Inequality:

$$\mathbb{P}[|sd_{x_2}(x_1) - \hat{sd}_{x_2}(x_1)| \geq t] \leq 2\exp(-2Nt^2). \tag{A· 7}$$

Then, for $q \in X$ and $C = \{c_i | c_i \in X\}_{i=1}^M$, we have:

$$\mathbb{P}[\cup_{c \in C}(|sd_c(q) - \hat{sd}_c(q)| \geq t)] \leq 2|C|\exp(-2Nt^2). \tag{A· 8}$$

Since $\mathbb{P}[\cup_{c \in C}(|sd_c(q) - \hat{sd}_c(q)| \geq t)] + \mathbb{P}[\cap_{c \in C}(|sd_c(q) - \hat{sd}_c(q)| < t)] = 1$, we have

$$\mathbb{P}[\cap_{c \in C}|sd_c(q) - \hat{sd}_c(q)| < t] \geq 1 - 2|C|\exp(-2Nt^2). \tag{A· 9}$$

Eq. (A· 9) represents that $\hat{sd}_c(q) - t \leq sd_c(q) \leq \hat{sd}_c(q) + t$ holds simultaneously for $\forall c \in C$ with probability at least $1 - 2|C|\exp(-2Nt^2)$. Note that Eq.(A· 9) satisfies if we replace $sd_c(q)$ and $\hat{sd}_c(q)$ with $sd_q(c)$ and $\hat{sd}_q(c)$, respectively. If we set $2|C|\exp(-2Nt^2) = \beta$, we obtain $t = \sqrt{\frac{1}{2N}\log(\frac{2|C|}{\beta})}$. Thus, we get the claim. □

A.8 Proof of Corollary 1

Proof. Since both Φ and Φ^{-1} are monotonic increasing functions, the followings with probability at least $1 - \beta$ hold by combining Lemma 3, Eq.(24), and Eq.(25):

$$\Phi(\Phi^{-1}(sd_{x_2}(x_1)) + \frac{\epsilon}{\sigma}) \leq \Phi(\Phi^{-1}(\hat{sd}_{x_2}(x_1) + \sqrt{\frac{1}{2N}\log(\frac{2|C|}{\beta})}) + \frac{\epsilon}{\sigma}) \tag{A· 10}$$

$$\Phi(\Phi^{-1}(sd_{x_2}(x_1)) - \frac{\epsilon}{\sigma}) \geq \Phi(\Phi^{-1}(\hat{sd}_{x_2}(x_1) - \sqrt{\frac{1}{2N}\log(\frac{2|C|}{\beta})}) - \frac{\epsilon}{\sigma}). \tag{A· 11}$$

Thus, we get the claim. □

A.9 Proof of Theorem 6 and Theorem 7

Proof. Using Lemma 3, the following holds with probability at least $1-\beta$:

$$\sum_{c\in C}\mathbb{1}[sd_{c_i}(q) < sd_c(q)] \geq \sum_{c\in C}\mathbb{1}[\hat{sd}_{c_i}(q)+t < \hat{sd}_c(q)-t], \tag{A· 12}$$

$$\sum_{c\in C}\mathbb{1}[sd_c(q) < sd_{c_i}(q)] \geq \sum_{c\in C}\mathbb{1}[\hat{sd}_c(q)+t < \hat{sd}_{c_i}(q)-t]. \tag{A· 13}$$

where $t=\sqrt{\frac{1}{2N}\log\frac{2|C|}{\beta}}$. Note that both Eq. (A· 12) and Eq. (A· 13) satisfy if we replace $sd_c(q)$ and $\hat{sd}_c(q)$ by $sd_q(c)$ and $\hat{sd}_q(c)$, respectively. Eq. (A· 12) and Eq. (A· 13) represent the lower bounds of the number of candidate images that are more dissimilar and similar to q than c_i, respectively. Thus, we get the claim. □

A.10 Proof of Theorem 8

Proof. From Theorem 6, the following holds with probability at least $1-\beta$:

$$\underline{\mathrm{Rank}}(q,c_i,C) \leq \mathrm{Rank}(q,c_i,C) \leq \overline{\mathrm{Rank}}(q,c_i,C), \tag{A· 14}$$

where $\overline{\mathrm{Rank}}(q,c,C)=|C|-\sum_{c\in C}\mathbb{1}(\hat{sd}_{c_i}(q)+t<\hat{sd}_c(q)-t)$ and $\underline{\mathrm{Rank}}(q,c,C)=\sum_{c\in C}\mathbb{1}(\hat{sd}_c(q)+t<\hat{sd}_{c_i}(q)-t)+1$. Moreover, combining Theorem 1 and Corollary 1, the followings also hold with probability at least $1-\beta$:

$$\mathrm{Rank}(q+\delta,c_i,C) \leq |C|-\sum_{c\in C}\mathbb{1}\left[\overline{sd}_{c_i}(q) < \underline{sd}_c(q)\right] \tag{A· 15}$$

$$\mathrm{Rank}(q+\delta,c_i,C) \geq \sum_{c\in C}\mathbb{1}\left[\overline{sd}_c(q) < \underline{sd}_{c_i}(q)\right]+1,$$

where $\overline{sd}_{x_2}(x_1)=\Phi(\Phi^{-1}(\hat{sd}_{x_2}(x_1)+\sqrt{(1/2N)\log(2|C|/\beta)})+\frac{\epsilon}{\sigma})$ and $\underline{sd}_{x_2}(x_1)=\Phi(\Phi^{-1}(\hat{sd}_{x_2}(x_1)-\sqrt{(1/2N)\log(2|C|/\beta)})-\frac{\epsilon}{\sigma})$. Thus, when Eq. (34) holds, the followings hold with probability at least $1-\beta$ by combining Eq. (A· 16) and Eq.(A· 15):

$$\begin{aligned}
&\alpha \geq \left(N-\sum_{c\in C}\mathbb{1}\left[\overline{sd}_{c_i}(q) < \underline{sd}_c(q)\right]\right)-\underline{\mathrm{Rank}}(q,c_i,C)\\
&\wedge\ \alpha \geq \overline{\mathrm{Rank}}(q,c_i,C)-\left(\sum_{c\in C}\mathbb{1}\left[\overline{sd}_c(q) < \underline{sd}_{c_i}(q)\right]+1\right)\\
&\Leftrightarrow \alpha \geq \left(N-\sum_{c\in C}\mathbb{1}\left[\overline{sd}_{c_i}(q) < \underline{sd}_c(q)\right]\right)-\mathrm{Rank}(q,c_i,C)\\
&\wedge\ \alpha \geq \mathrm{Rank}(q,c_i,C)-\left(\sum_{c\in C}\mathbb{1}\left[\overline{sd}_c(q) < \underline{sd}_{c_i}(q)\right]+1\right)\\
&\Leftrightarrow \alpha \geq \mathrm{Rank}(q+\delta,c_i,C)-\mathrm{Rank}(q,c_i,C)\\
&\wedge\ \alpha \geq \mathrm{Rank}(q,c_i,C)-\mathrm{Rank}(q+\delta,c_i,C)\\
&\Leftrightarrow |\mathrm{Rank}(q+\delta,c_i,C)-\mathrm{Rank}(q,c_i,C)| \leq \alpha
\end{aligned}$$

Thus, we get the claim. □

A.11 Proof of Theorem 9

Proof. From Theorem 7, the following holds probability at least $1-\beta$:

$$\underline{\mathrm{Rank}}(q,c_i,C) \leq \mathrm{Rank}(q,c_i,C) \leq \overline{\mathrm{Rank}}(q,c_i,C), \tag{A· 16}$$

where $\overline{\mathrm{Rank}}(q,c_i,C)=|C|-\sum_{c\in C}\mathbb{1}(\hat{sd}_q(c_i)+t<\hat{sd}_q(c)-t)$, and $\underline{\mathrm{Rank}}(q,c_i,C)=\sum_{c\in C}\mathbb{1}(\hat{sd}_q(c)+t<\hat{sd}_q(c_i)-t)+1$. Combining Theorem 2 and Corollary 1, the followings holds with probability at least $1-\beta$:

$$\mathrm{Rank}(q,c_i+\delta_j,\tilde{C}) \leq |C|-\sum_{c\in C}\mathbb{1}\left[\overline{sd}_q(c_i) < \underline{sd}_q(c)\right] \tag{A· 17}$$

$$\mathrm{Rank}(q,c_i+\delta_i,\tilde{C}) \geq \sum_{c\in C}\mathbb{1}\left[\overline{sd}_q(c) < \underline{sd}_q(c)\right]+1,$$

where $\overline{sd}_{x_2}(x_1)=\Phi(\Phi^{-1}(\hat{sd}_{x_2}(x_1)+\sqrt{(1/2N)\log(2|C|/\beta)})+\frac{\epsilon}{\sigma})$, $\underline{sd}_{x_2}(x_1)=\Phi(\Phi^{-1}(\hat{sd}_{x_2}(x_1)-\sqrt{(1/2N)\log(2|C|/\beta)})-\frac{\epsilon}{\sigma})$. Thus, when Eq. (34) holds, the followings hold with probability at least $1-\beta$ by combining Eq. (A· 16) and Eq.(A· 15):

$$\begin{aligned}
&\alpha \geq \left(N-\sum_{c\in C}\mathbb{1}\left[\overline{sd}_q(c_i) < \underline{sd}_q(c)\right]\right)-\underline{\mathrm{Rank}}(q,c_i,C)\\
&\wedge\ \alpha \geq \overline{\mathrm{Rank}}(q,c_i,C)-\left(\sum_{c\in C}\mathbb{1}\left[\overline{sd}_q(c) < \underline{sd}_q(c_i)\right]+1\right)\\
&\Leftrightarrow \alpha \geq \left(N-\sum_{c\in C}\mathbb{1}\left[\overline{sd}_q(c_i) < \underline{sd}_q(c)\right]\right)-\mathrm{Rank}(q,c_i,C)\\
&\wedge\ \alpha \geq \mathrm{Rank}(q,c_i,C)-\left(\sum_{c\in C}\mathbb{1}\left[\overline{sd}_q(c) < \underline{sd}_q(c_i)\right]+1\right)\\
&\Leftrightarrow \alpha \geq \mathrm{Rank}(q+\delta,c_i,C)-\mathrm{Rank}(q,c_i,C)\\
&\wedge\ \alpha \geq \mathrm{Rank}(q,c_i,C)-\mathrm{Rank}(q+\delta,c_i,C)\\
&\Leftrightarrow |\mathrm{Rank}(q+\delta,c_i,C)-\mathrm{Rank}(q,c_i,C)| \leq \alpha
\end{aligned}$$

Thus, we get the claim. □

Appendix B: Comparison Methods

We compare our proposed robustness training Eq. (20) (TBT) and Eq. (21) (FCTB) with three existing methods: (i) triplet Loss (Triplet) [36], (ii) anti-collapse triplet (ACT), which is an adversarial training for CBIR to improve empirical robustness [3], (iii) training for classification using interval bound propagation (C-IBP) to improve certified robustness for the classification task [21].

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:

$$\sum_{(a,p,n)\in D_t} \max\{\|f(a)-f(p)\|_2-\|f(a)-f(n)\|_2+m, 0\}, \tag{A· 18}$$

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:

$$\sum_{(a,p,n)\in D_t} \max\{\|f(a) - f(p + \delta_p)\|_2 - \|f(a) - f(n + \delta_n)\|_2 + m, 0\}, \tag{A· 19}$$

where

$$\delta_p, \delta_n = \mathop{\arg\min}_{\substack{\delta_p,\delta_n \in X,\\ \|\delta_p\|_\infty \le \epsilon, \|\delta_n\|_\infty \le \epsilon}} \|f(p + \delta_p) - f(n + \delta_n)\|_2. \tag{A· 20}$$

In our experiments, we minimize Eq. (A· 20) by using PGD [12] 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 $D_t = \{(x, y)_i\}_{i=1}^{M}$:

$$\sum_{(x,y)\in D_t} \kappa \cdot CE(f_c(x), y) + (1 - \kappa) \cdot CE(\hat{f}_c^y(x), y), \tag{A· 21}$$

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.

B.1 Hyperparameters of ACT and C-IBP

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 [12] with the step size of $\frac{\epsilon}{10}$ and the number of updates of 20.

When training with C-IBP, we use the scheduling strategy for ϵ and κ proposed in [21] as well as training of TBT. 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 all datasets. Then, we linearly increase ϵ and decrease κ between $2K$ and $10K$ steps.

Kazuya Kakizaki received the B.E. and M.E. degrees from University of Tsukuba, Japan, in 2015 and 2017, respectively. Since 2017, he has worked as a researcher at NEC Corporation, Tokyo, Japan. He has also been a Ph.D. student at University of Tsukuba since 2021. His research interests include machine learning, data privacy, and AI security.

Kazuto Fukuchi received a Ph.D. degree from the University of Tsukuba, Tsukuba, Japan, in 2018. He has been an assistant professor in Institute of Systems and Information Engineering, University of Tsukuba, Japan, since 2019. He has also been a visiting researcher at the Center for Advanced Intelligence Project, RIKEN, Japan, since 2019. His research interests include mathematical statistics, machine learning, and their applications.

Jun Sakuma received a Ph.D. in Engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2003. He has been a professor in the School of Computing of Tokyo Institute of Technology since 2023. He has also been a team leader of the Artificial Intelligence Security and Privacy team in the Center for Advanced Intelligence Project, RIKEN, since 2016. Before that, he worked as a professor in the Department of Computer Science, School of System and Information Engineering, University of Tsukuba, Tsukuba, Japan, since 2016. He worked as an associate professor in the same department at University of Tsukuba (2009–2016). He was an assistant professor in the Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo, Japan (2004–2009). He was a researcher at Tokyo Research Laboratory, IBM, Tokyo, Japan (2003–2004). His research interests are machine learning, data privacy, and AI security. He is a member of the Institute of Electronics, Information, and Communication Engineers of Japan (IEICE).