Robustness Verification for Contrastive Learning

Zekai Wang [1] **Weiwei Liu** [1]

Abstract

Contrastive adversarial training has successfully improved the robustness of contrastive learning (CL). However, the robustness metric used in these methods is linked to attack algorithms, image labels and downstream tasks, all of which may affect the consistency and reliability of robustness metric for CL. To address these problems, this paper proposes a novel Robustness Verification framework for Contrastive Learning (RVCL). Furthermore, we use extreme value theory to reveal the relationship between the robust radius of the CL encoder and that of the supervised downstream task. Extensive experimental results on various benchmark models and datasets verify our theoretical findings, and further demonstrate that our proposed RVCL is able to evaluate the robustness of both models and images. Our code is available at https://github.com/wzekai99/RVCL.

1. Introduction

While neural networks (NNs) have achieved remarkable performance in various applications, many studies (Goodfellow et al., 2015; Madry et al., 2018) have demonstrated that NNs are vulnerable when dealing with imperceptibly perturbed images. A rapidly growing body of work therefore aims to investigate how a robust NN model might be obtained.

One successful method in this field is based on adversarial training (AT) (Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017), which improves the robustness of NN by augmenting the training set with adversarial samples (Szegedy et al., 2014). Schmidt et al. (2018) show that AT requires a large amount of data, while the labels of this data are expensive to obtain. Thus, several existing works have attempted to improve the robustness of adversarially trained models through the use of additional unlabeled data (Alayrac et al., 2019; Carmon et al., 2019; Zhai et al., 2019). Recently, attempts have been made to combine AT with contrastive learning (CL) (Kim et al., 2020; Fan et al., 2021). CL (Chen et al., 2020; He et al., 2020) has demonstrated superior abilities in unsupervised learning: specifically, it can surpass the standard accuracy of supervised learning on downstream image classification tasks by utilizing a large-scale unlabeled dataset. It is therefore of primary interest to study the robust performance achieved by contrastive AT (Gowal et al., 2021).

However, existing contrastive AT methods use the *empirical robustness metric* (e.g. robust accuracy) to evaluate the robustness of encoders, an approach that relies on attack algorithms, image labels and downstream tasks. Three key questions naturally arise from this kind of measurement:

1. Attack algorithm: Robust accuracy is related to a specific attack algorithm (e.g. PGD attack (Madry et al., 2018)); thus, the results may not be consistent with powerful adversaries.
2. Image label: It is farfetched to train with unlabeled images while evaluating the encoder with labeled images.
3. Downstream task: We do not know whether the robustness benefits from the encoder or the linear classifier.

In fact, the first question has already been considered in supervised learning. To explore the first question in more detail, the robustness analysis of NN is entangled with the specific attack algorithms used for evaluation (Weng et al., 2018). Most defense heuristics have subsequently been shown to fail against suitably powerful attack algorithms (Carlini & Wagner, 2017; Uesato et al., 2018). Even if the model is made robust to the attack algorithm used by evaluation, there is no guarantee that will remain robust to other unseen attacks. This has encouraged researchers to develop *robustness verification* (Katz et al., 2017; Wong et al., 2018): i.e., classifiers whose prediction at point x is verified to be constant within a neighborhood of x, regardless of what attack algorithm is applied. The key concept is to find the largest radius of this neighborhood, referred to as the *robust radius*, which is an important metric for use in studying the robustness of NN.

Unfortunately, previous works on robustness verification have used only supervised learning; i.e., true labels of data are required for both the supervised classifier and verifier.

[1]School of Computer Science, Wuhan University, China. Correspondence to: Weiwei Liu <liuweiwei863@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

A naive approach would be to apply a current verification framework to supervised downstream tasks, although this method would also suffer from the issues highlighted in Questions 2 & 3, which prompts us to ask the following questions:

- *Can we design a robustness verification framework for contrastive learning that does not require class labels and downstream tasks?*
- *Is there any relationship between the robust radius of the CL encoder and that of the downstream task?*

Our work attempts to conduct a rigorous and comprehensive study that addresses the above questions. Our main contributions are summarized below.

1. We propose RVCL, a novel Robustness Verification framework for Contrastive Learning. We then define the robust radius for CL such that no points inside this radius can be recognized as negative samples; this is a robustness metric for encoders that does not require the involvement of an attack algorithm, image label and downstream task.
2. We use extreme value theory to prove that the robust radius of the CL encoder is the upper bound of the robust radius of the downstream task. This implies that robust data representations rendered by the CL encoder (with a large robust radius) can benefit from the model's robust performance on downstream tasks.
3. To verify the efficacy of the proposed RVCL, we validate our proposed framework on two verification benchmark datasets (MNIST and CIFAR-10). Our experimental results illustrate that the proposed RVCL is a suitable robustness metric for models without labels, and accordingly validate our theoretical analysis. Moreover, RVCL is also able to evaluate the anti-disturbance ability for distinct images.

2. Related work

Self-supervised Contrastive Learning Self-supervised learning (Jing & Tian, 2021), which involves training models using unlabeled data and various pretext tasks, has become popular as a means of extracting feature representation for deep NNs. Early advances have been used to solve image jigsaw puzzles (Noroozi & Favaro, 2016), predict rotation angles (Gidaris et al., 2018), fill image patches (Doersch et al., 2015), etc. Recently, contrastive learning (CL) (Wu et al., 2018; Chen et al., 2020; He et al., 2020; Tian et al., 2020) has been proposed by maximizing the agreement between positive samples while contrasting with negative samples, and has further been shown to work well in learning effective representations. Some theoretical works have also been proposed; for example, Saunshi et al. (2019) provide the first generalization bound for CL, while Nozawa et al. (2020) extend it by means of a PAC-Bayesian approach.

Contrastive Adversarial Training Due to the brittleness of NNs when faced with tiny input perturbations, AT (Madry et al., 2018) is one of the most powerful robust training methods used to enhance model robustness. Several recent works (Kim et al., 2020; Ho & Vasconcelos, 2020; Jiang et al., 2020; Fan et al., 2021) have explored how to improve robustness using contrastive AT. To obtain more robust data representations, AT is used on contrastive pretraining tasks following the "contrastive adversarial pretraining + supervised finetuning" paradigm. However, existing methods use an empirical robustness metric; the systematic study of the verified robustness of CL have been less explored.

Robustness Verification In this paper, we focus on deterministic verification, following the taxonomy of Li et al. (2020). When the given input is non-robust against the attack, deterministic verification is guaranteed to identify this nonrobustness. The literature for this setting can be broadly divided into several categories: complete verifiers using satisfiability modulo theory (SMT) (Katz et al., 2017; Ehlers, 2017), mixed integer programming (MIP) (Tjeng et al., 2019; Anderson et al., 2020) and branch and bound (BaB) (Bunel et al., 2018; Wang et al., 2021); incomplete verifiers using bound propagation (Zhang et al., 2018; Xu et al., 2020), and convex relaxation by linear programming (Wong et al., 2018; Wong & Kolter, 2018). However, these works focus on supervised settings; verification with CL settings remain unknown.

Extreme Value Theory Extreme value theory (EVT) has been recognized as a powerful tool, since it enables the limit distribution of properly normalized maxima to be effectively modeled (Scheirer et al., 2011). This success has produced strong empirical results for describable visual attributes (Scheirer et al., 2012), visual inspection tasks (Gibert et al., 2015) and open set recognition problems (Rudd et al., 2018), etc. Recently, CLEVER (Weng et al., 2018) estimates the robust radius of supervised verification using EVT. The difference is discussed in more detail in Appendix B. In this paper, we creatively use EVT to theoretically analyze the relationship between the robust radius of the CL encoder and that of the downstream task.

3. Preliminaries

We first present notations and describe the frameworks for contrastive learning and supervised verification problem that will be essential for our analysis.

Notions Let $\|\cdot\|_2$ and $\|\cdot\|_\infty$ denote the Euclidean norm ℓ_2 and infinity norm ℓ_∞, respectively. $\mathbb{1}_{[Boolean\ expression]}$ is the indicator function (equal to 1 if the expression is True and 0 otherwise). $\mathcal{B}_\infty(x_0, \epsilon) := \{x \mid \|x - x_0\|_\infty \leq \epsilon\}$ denotes that the input x is constrained into the ℓ_∞ ball. We

let $\text{sign}(x) = 1$ for $x \geq 0$ and $\text{sign}(x) = -1$ for $x < 0$. If $\mathcal{S}$ is a set, $|\mathcal{S}|$ denotes its cardinality. The transpose of the vector/matrix is represented by the superscript $\top$. Given point x, the ℓ_2 normalization is defined as $\widetilde{\rho}(u) = u/\|u\|_2$. Given points u and v, the instance similarity is defined as $\rho(u, v) = u^\top v/\|u\|_2\|v\|_2$, which is the dot product between the ℓ_2 normalized u and v (i.e., cosine similarity). We use $[n]$ to represent the set $\{1, 2, \ldots, n\}$.

Neural network Consider an input vector $x \in \mathbb{R}^{d_0}$ for a neural network with L layers. Let the number of neurons in the k-th layer be d_k, while $\mathbf{W}_k \in \mathbb{R}^{d_k \times d_{k-1}}$ and $\boldsymbol{b}_k \in \mathbb{R}^{d_k}$ ($k \in [L]$) represent the weights and biases of NN. Let $\phi_k : \mathbb{R}^{d_0} \to \mathbb{R}^{d_k}$ be the operator mapping the input layer to layer k. $\sigma(v)$ is the activation function, while the ReLU activation is $\sigma(v) = \max(v, 0)$. For each $k \in [L]$, $\phi_k(x) = \mathbf{W}_k\widehat{\phi}_{k-1}(x) + \boldsymbol{b}_k$, $\widehat{\phi}_k(x) = \sigma(\phi_k(x))$, $\widehat{\phi}_0(x) = x$. We simply use $\phi_k^{(j)}$ and $\widehat{\phi}_k^{(j)}$ to represent the pre-activation and post-activation values of the k-th layer and j-th neuron. The output of the neural network is $\phi_L(x) \in \mathbb{R}^{d_L}$. d_L further denotes the number of input classes.

3.1. Contrastive Learning

Let $\mathcal{X}$ denote the set of all possible data points. Let $\mathcal{Y}$ denote the label set of the downstream task, which is a discrete and finite set. $\mathcal{F}$ is a class of *feature encoder* $f : \mathcal{X} \to \mathbb{R}^d$. To highlight the key ideas, we present the CL framework proposed by Saunshi et al. (2019) in a simplified binary scenario, i.e., $\mathcal{Y} = \{-1, 1\}$. CL assumes that we obtain the similar data in the form of pairs (x, x^+) and K independent and identically distributed (i.i.d.) negative samples $x_1^-, x_2^-, \ldots, x_K^-$. Given an unlabeled dataset $U = \{z_i\}_{i=1}^m$, where $z_i = (x_i, x_i^+, x_{i1}^-, \ldots, x_{iK}^-)$, we aim to learn an encoder f that makes $f(x)$ similar to $f(x^+)$, while keeping away from $f(x_1^-), \ldots, f(x_K^-)$ at the same time.

Linear evaluation One standard method for evaluating the performance of the CL model is *linear evaluation* (Chen et al., 2020; Kim et al., 2020), which learns a downstream linear layer after the base encoder, then uses a modified model for class-level classification. The test accuracy on the downstream task is used as a proxy for representation quality. The model with downstream layer is fine-tuned from a labeled dataset $S = \{(x_i, y_i)\}_{i=1}^n$. Both U and S are assumed to be i.i.d. collections.

Data distributions Let $\mathcal{C}$ denote the set of *latent classes* (Saunshi et al., 2019) that are all possible classes for points in $\mathcal{X}$. For each class $c \in \mathcal{C}$, there is a probability $\mathcal{D}_c$ over $\mathcal{X}$ that captures the probability that a point belongs to class c. The distribution on $\mathcal{C}$ is denoted by η. Let c^+, c^- denote the positive and negative latent class drawn from η; thus, $\mathcal{D}_{c^+}$ and $\mathcal{D}_{c^-}$ are the distributions to sample positive and negative samples, respectively. The process for generating an unlabeled sample $z = (x, x^+, \{x_i^-\}_{i=1}^K) \in U$ as follows: 1. Draw two latent classes $(c^+, c^-) \sim \eta^2$; 2. Draw two positive samples $(x, x^+) \sim \mathcal{D}_{c^+}^2$ and K negative samples $\{x_i^- \sim \mathcal{D}_{c^-} \mid i \in [K]\}$.

To set up the labeled dataset S for binary scenario, we build the binomial distribution η_{sup} by fixing two classes c^+, c^-: $\eta_{sup}(c^+) = \frac{\eta(c^+)}{\eta(c^-)+\eta(c^+)}$, $\eta_{sup}(c^-) = \frac{\eta(c^-)}{\eta(c^-)+\eta(c^+)}$. We fix $y_{c^+} = +1$ and $y_{c^-} = -1$, then generate a labeled sample $(x, y) \in S$ as follows: 1. Draw a class $c \sim \eta_{sup}$ and set the label $y = y_c$; 2. Draw a sample $x \sim \mathcal{D}_c$.

Loss function The learning process is divided into two steps: minimizing the contrastive loss on the encoder and fine-tuning on the downstream layer using supervised loss. We focus on *logistic loss*: $\ell(v) = \log_2(1 + \sum_j \exp(-v_j))$ for $v \in \mathbb{R}^K$. Thus, the *contrastive loss* (Chen et al., 2020; He et al., 2020) associated with the encoder f in this framework is defined as follows:

$$\mathcal{L}_{un}(f) = \mathop{\mathbb{E}}_{\substack{c^+,c^- \\ \sim\eta^2}} \mathop{\mathbb{E}}_{\substack{x,x^+\sim\mathcal{D}_{c^+}^2 \\ x_i^-\sim\mathcal{D}_{c^-}}} \ell\left(\left\{f(x)^T\left(f(x^+) - f(x_j^-)\right)\right\}\right). \tag{1}$$

For linear evaluation, the supervised learning algorithm is given the *mapped* dataset $\widehat{S} := \{(f(x_i), y_i)\}_{i=1}^n$ and returns a predictor $g : \mathbb{R}^d \to \mathbb{R}$. The label of $\widehat{x} \in \widehat{S}$ is obtained from $\widehat{y} = \text{sign}(g(\widehat{x}))$, $\widehat{y} \in \{-1, 1\}$. The logistic loss in (2) is $\ell(v) = \log_2(1 + \exp(-v))$ for $v \in \mathbb{R}$. We aim to minimize the supervised loss as follows:

$$\mathcal{L}_{sup}(g \circ f) = \mathop{\mathbb{E}}_{c\sim\eta_{sup}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}_c} \ell\left(y_c \cdot g(f(x))\right). \tag{2}$$

3.2. Supervised Verification

In this section, we set up the verification problem for supervised learning. For simlicity, we here outline the simplified binary scenario. The supervised algorithm is given the labeled dataset $S = \{(x_i, y_i)\}_{i=1}^n$. Let the dimension of NN output $d_L = 1$. $\mathcal{H}$ contains a class of *supervised predictor* $h : \mathcal{X} \to \mathbb{R}$ with $h := \phi_L(x)$. Thus, the label of x is predicted by $\widehat{y} = \text{sign}(h(x))$, $\widehat{y} \in \{-1, 1\}$.

Verification problem We refer to $x' = x + \delta$ as an *adversarial sample* of x for classifier h if h correctly classifies x but assigns a different label to x'. Because many powerful attack methods (Goodfellow et al., 2015; Carlini & Wagner, 2017) and adversarial training frameworks (Madry et al., 2018; Zhang et al., 2019) use ℓ_∞ norm, this paper also focuses on the setting in which δ satisfies the ℓ_∞ norm constraint $\|\delta\|_\infty \leq \epsilon$. We say that model h is l_∞^ϵ-*verified* at (x, y) if it correctly classifies both x and x' as y for any $x' \in \mathcal{B}_\infty(x, \epsilon)$, i.e., there are no adversarial samples around

x. The supervised verification at $(x, y), y \in \{-1, 1\}$ seeks the solution of the following optimization problem:

$$\begin{aligned} &\widetilde{h}(x, y, \epsilon) \coloneqq \min_{x'} \ y \cdot h(x') \\ \text{s.t. } &\phi_k(x') = \mathbf{W}_k \widehat{\phi}_{k-1}(x') + \boldsymbol{b}_k, k \in [L], \\ &\widehat{\phi}_k(x') = \sigma(\phi_k(x')), k \in [L-1], \\ &h(x') = \phi_L(x'), \\ &x' \in \mathcal{B}_\infty(x, \epsilon). \end{aligned} \tag{3}$$

If $\widetilde{h} \leq 0$, $\exists x' \in \mathcal{B}_\infty(x, \epsilon)$ fools the model into producing an incorrect label. h is l_∞^ϵ-*verified* if $\widetilde{h}(x, y, \epsilon) > 0$. The complete verifier aims to solve (3) and calculates $\widetilde{h}$ exactly. Unfortunately, the complete verification is proven to be an NP-complete problem (Katz et al., 2017; Sinha et al., 2018). Therefore, many previous works (Wong & Kolter, 2018; Zhang et al., 2018; Xu et al., 2020) propose incomplete verifiers that relax the non-convexity part of NN to derive a lower bound $\widetilde{h} \geq \underline{h}$. If $\underline{h}(x, y, \epsilon) > 0$ is given by the incomplete verifier, model h is also l_∞^ϵ-*verified* at (x, y).

Robust radius The l_∞^ϵ-*verified* of h at (x, y) depends on the radius of the largest ℓ_∞ ball centered at x in which h does not change its prediction. This radius is called the *robust radius*, which is formally defined as follows:

$$\begin{aligned} \mathrm{R}(h; x, y) &\coloneqq \inf_{\mathrm{sign}(h(x')) \neq y} \|x' - x\|_\infty \\ &= \sup_\epsilon \ \epsilon \text{ s.t. } \widetilde{h}(x, y, \epsilon) > 0. \end{aligned} \tag{4}$$

If $h(x) \neq y$, then $\mathrm{R}(h; x, y) \coloneqq 0$. It is natural to regard the robust radius as a robustness metric. Recall that computing the robust radius is an NP-hard problem due to the need for complete verification. We can thus derive a tight lower bound of R given by $\underline{h}$, referred to as the *certified radius*, which is formally defined as

$$\underline{\mathrm{R}}(h; x, y) \coloneqq \sup_\epsilon \ \epsilon \text{ s.t. } \underline{h}(x, y, \epsilon) > 0. \tag{5}$$

The certified radius satisfies $0 \leq \underline{\mathrm{R}}(h; x, y) \leq \mathrm{R}(h; x, y)$.

4. RVCL: Robustness Verification Framework for Contrastive Learning

In this section, we first introduce the verification problem on supervised downstream tasks by simply modifying supervised verification (3), and further present several weaknesses of adopting (3) in CL. We go on to propose a novel RVCL framework to solve these issues.

4.1. Verification Problem for Linear Evaluation

By defining the supervised verification problem on linear evaluation, we can regard the robust radius R as a proxy robustness metric for CL.

We denote the encoder $f : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ with $f \coloneqq \phi_L(x)$, where d_0 and d_L are the dimensions of the encoder's input and output, while $\mathbf{W}_{\mathrm{LE}} \in \mathbb{R}^{1 \times d_L}$ and $\boldsymbol{b}_{\mathrm{LE}} \in \mathbb{R}$ are the weight and bias of the downstream linear predictor $g(x)$. The optimization problem for linear evaluation is defined by simply modifying the constraints of (3), as follows:

$$\begin{aligned} &\widetilde{g}(x, y, \epsilon) \coloneqq \min_{x'} \ y \cdot g(x') \\ \text{s.t. } &\phi_k(x') = \mathbf{W}_k \widehat{\phi}_{k-1}(x') + \boldsymbol{b}_k, k \in [L], \\ &\widehat{\phi}_k(x') = \sigma(\phi_k(x')), k \in [L-1], \\ &g(x') = \mathbf{W}_{\mathrm{LE}} \phi_L(x') + \boldsymbol{b}_{\mathrm{LE}}, \\ &x' \in \mathcal{B}_\infty(x, \epsilon). \end{aligned} \tag{6}$$

The difference between (3) and (6) is that there is no active function $\sigma(\cdot)$ between the encoder and downstream layer. There is no barrier to applying incomplete verifiers on (6). Thus, the definition of robust radius $\mathrm{R}_{\mathrm{LE}}(g; x, y)$ and certified radius $\underline{\mathrm{R}}_{\mathrm{LE}}(g; x, y)$ on the downstream task are similar to (4) and (5), respectively.

We can therefore regard $\underline{\mathrm{R}}_{\mathrm{LE}}$ as a proxy robust radius at data point x. However, this approach has serious problems:

1. $\underline{\mathrm{R}}_{\mathrm{LE}}$ cannot be computed directly without a label.
2. Even if we have the label to compute $\underline{\mathrm{R}}_{\mathrm{LE}}$, and use $\underline{\mathrm{R}}_{\mathrm{LE}}$ to evaluate the model robustness, we do not know whether the robustness benefits from the encoder or downstream layer.

These problems motivate us to propose RVCL, a novel framework for verifying the robustness of encoders without the need for labels and downstream tasks.

4.2. RVCL Framework

Many existing works have studied the supervised verification problem stated in § 3.2. However, the performing of robustness verification for CL has received less research attention. In this section, we present the formal definition of the robustness verification problem on the encoder f, after which we provide two robustness metrics to study the performance of the CL encoder and incomplete verifier.

4.2.1. Verification problem for CL

The core concept behind supervised verification is that the points in the small $\mathcal{B}_\infty(x, \epsilon)$ ball should have the same label as x. Inspired by this idea, we define *the conditions under which the disturbance successfully attacks the encoder*.

Given a positive sample x^+, let the negative sample x^- be the attack target of x^+. We hope that the points $x' \in \mathcal{B}_\infty(x^+, \epsilon)$ will be more similar to x^+ than any other negative samples x^-, while the instance-wise attack algorithm generates an adversarial sample $x' \in \mathcal{B}_\infty(x^+, \epsilon)$ with the attack strength ϵ, in order to fool the model by judging x' as similar to x^-.

If the instance similarity $\rho(f(x^+), f(x')) > \rho(f(x^-), f(x'))$, then x' is similar to x^+ (i.e., $\theta_1 < \theta_2$ in Figure 1), which means that the encoder f is not successfully attacked by x'. We say that the encoder f is l_∞^ϵ-*verified* at (x^+, x^-) if x' is more similar to x^+ than to x^- for any $x' \in \mathcal{B}_\infty(x^+, \epsilon)$, i.e., there are no adversarial samples similar to x^- around x^+. Note that the comparison of instance similarity has an equivalent form:

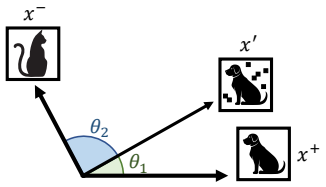


Figure 1. θ is the angle between features given by f. If $\theta_1 < \theta_2$, f is not attacked successfully by the adversarial sample x'.

$$\rho(f(x^+), f(x')) > \rho(f(x^-), f(x')) \iff \tag{7}$$

$$\left(\widetilde{\rho}(f(x^+)) - \widetilde{\rho}(f(x^-))\right)^\top f(x') > 0 \tag{8}$$

Judging whether or not (8) is True can be regarded as a part of forward propagation; thus, we can define the optimization problem for CL by adding a linear layer after f with weight $\mathbf{W}_{\mathrm{CL}} = \left(\widetilde{\rho}(f(x^+)) - \widetilde{\rho}(f(x^-))\right)^\top$.

Definition 4.1 (Verification problem for CL)**.** Given two positive and negative samples $x^+, x^- \in \mathbb{R}^{d_0}$, respectively, the feature encoder $f : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ with $f := \phi_L(x)$, ℓ_2 normalization $\widetilde{\rho}(u) = u/\|u\|_2$, for any fixed ϵ, the robustness verification problem for CL is defined as follows:

$$\begin{aligned}
&\widetilde{f}(x^+, x^-, \epsilon) := \min_{x'} \ \mathbf{W}_{\mathrm{CL}} f(x') \\
\text{s.t. } &\phi_k(x') = \mathbf{W}_k \widehat{\phi}_{k-1}(x') + \boldsymbol{b}_k, k \in [L], \\
&\widehat{\phi}_k(x') = \sigma(\phi_k(x')), k \in [L-1], \\
&\mathbf{W}_{\mathrm{CL}} = \left(\widetilde{\rho}(f(x^+)) - \widetilde{\rho}(f(x^-))\right)^\top \in \mathbb{R}^{1 \times d_L}, \\
&f(x') = \phi_L(x'),\ x' \in \mathcal{B}_\infty(x^+, \epsilon).
\end{aligned} \tag{9}$$

Moreover, the *robust radius* R_{CL} and *certified radius* $\underline{\mathrm{R}}_{\mathrm{CL}}$ for CL are defined as follows:

$$\begin{aligned}
\mathrm{R}_{\mathrm{CL}}(f; x^+, x^-) &:= \inf_{\substack{\rho(f(x'), f(x^+)) \\ < \rho(f(x'), f(x^-))}} \left\|x' - x^+\right\|_\infty \\
&= \sup_\epsilon \ \epsilon \text{ s.t. } \widetilde{f}(x^+, x^-, \epsilon) > 0, \\
\underline{\mathrm{R}}_{\mathrm{CL}}(f; x^+, x^-) &:= \sup_\epsilon \ \epsilon \text{ s.t. } \underline{f}(x^+, x^-, \epsilon) > 0,
\end{aligned} \tag{10}$$

where $\underline{f}$ is the verified lower bound of $\widetilde{f}$ given by the verifier; thus, $0 \le \underline{\mathrm{R}}_{\mathrm{CL}}(f; x^+, x^-) \le \mathrm{R}_{\mathrm{CL}}(f; x^+, x^-)$. For verified prediction, $\underline{f}(x^+, x^-, \epsilon) > 0$ for a given strength ϵ implies that f is l_∞^ϵ-*verified* at (x^+, x^-). The pseudocode is presented as PREDICT in Appendix C.2.

To compute $\underline{\mathrm{R}}_{\mathrm{CL}}$, we can apply *binary search*, because $\underline{f}(x^+, x^-, \epsilon)$ is non-increasing with increasing ϵ. The pseudocode is presented as CERTIFY in Appendix C.2.

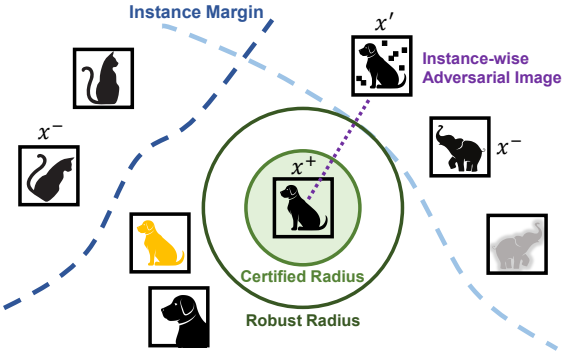


Figure 2. Illustration for RVCL. $\|x' - x^+\|_\infty$ must be larger than the robust radius R_{CL} if x' is an instance-wise adversarial sample. In this case, the latent class of x' is "dog", while the feature of x' is similar to that of x^-, which is an "elephant". The certified radius $\underline{\mathrm{R}}_{\mathrm{CL}}$ is provided by the incomplete verifier, which is the lower bound of robust radius R_{CL}. It is verified that no instance-wise adversarial sample exists in $\mathcal{B}_\infty(x^+, \underline{\mathrm{R}}_{\mathrm{CL}})$.

Note that (9) and (10) are both defined directly on the CL encoder without reference to any labels or downstream tasks, which resolves the issue articulated in § 4.1.

4.2.2. ROBUSTNESS METRICS FOR CL

This subsection provides two robustness metrics used to study the robustness of the CL encoder and the performance of incomplete verifiers.

Average certified radius (ACR) For supervised verification, ACR (Zhai et al., 2020) is an important metric used in evaluating robustness. Specifically, it is the average of the certified radius on the test dataset. We can define it directly on the supervised downstream task: $\mathrm{ACR}_{\mathrm{LE}} := \frac{1}{|S_{\mathtt{test}}|} \sum_{(x,y) \in S_{\mathtt{test}}} \underline{\mathrm{R}}_{\mathrm{LE}}(g; x, y)$, where $S_{\mathtt{test}}$ is a labeled test dataset satisfying $g(x) = y$ for all $(x, y) \in S_{\mathtt{test}}$. However, $\mathrm{ACR}_{\mathrm{LE}}$ still suffers from the problems discussed in § 4.1. We therefore define ACR for CL based on $\underline{\mathrm{R}}_{\mathrm{CL}}$, which directly reflects the robustness of CL encoder without the label:

Definition 4.2 (Average certified radius for CL)**.** Given an unlabeled test dataset $U_{\mathtt{test}}$ generated following § 3.1, $z = (x^+, \{x_i^-\}_{i=1}^K) \in U_{\mathtt{test}}$, $\underline{\mathrm{R}}_{\mathrm{CL}}$ is defined in (10). The average certified radius for CL is defined as follows:

$$\mathrm{ACR}_{\mathrm{CL}} := \frac{1}{K|U_{\mathtt{test}}|} \sum_{z \in U_{\mathtt{test}}} \sum_{i=1}^{K} \underline{\mathrm{R}}_{\mathrm{CL}}(f; x^+, x_i^-). \tag{11}$$

Certified instance accuracy For supervised verification, certified accuracy is a metric used to evaluate the performance of incomplete verifiers. Wang et al. (2021) state that the verifier will be stronger if the certified accuracy is the tighter lower bound of supervised robust accuracy. Since there is no definition of "robust accuracy" provided for the CL encoder, we propose a novel robust accuracy without

label and downstream task for CL — called *robust instance accuracy* — based on (7):

Definition 4.3 (Robust instance accuracy)**.** Given an unlabeled test dataset $U_{\texttt{test}}$, $z = (x^+, x^-) \in U_{\texttt{test}}$, we use *instance-wise PGD attack* (Kim et al., 2020) to generate the adversarial point $x' \in \mathcal{B}(x^+, \epsilon)$ by maximizing the contrastive loss (1). The robust instance accuracy with strength ϵ is defined as follows:

$$\mathcal{A}^{\epsilon}_{\mathrm{CL}} = \frac{1}{|U_{\texttt{test}}|} \sum_{z \in U_{\texttt{test}}} \mathbb{1}_{[\rho(f(x'), f(x^+)) - \rho(f(x'), f(x^-)) > 0]}. \tag{12}$$

We then define the certified instance accuracy with strength ϵ, which is the fraction of the test dataset for which f is l^{ϵ}_{∞}-*verified* at (x^+, x^-), i.e., $\underline{f} > 0$.

Definition 4.4 (Certified instance accuracy)**.** Given an unlabeled test dataset $U_{\texttt{test}}$, $z = (x^+, x^-) \in U_{\texttt{test}}$. The certified instance accuracy with strength ϵ is defined as

$$\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}} = \frac{1}{|U_{\texttt{test}}|} \sum_{z \in U_{\texttt{test}}} \mathbb{1}_{[\underline{f}(x^+, x^-, \epsilon) > 0]}. \tag{13}$$

f being l^{ϵ}_{∞}-*verified* at (x^+, x^-) is the sufficient but not necessary condition of correctly classifying x' generated by a specific attack algorithm with attack strength ϵ. This means that the hold of the judgement condition in (13) implies the hold of that in (12), but not vice versa. Thus, (13) is the lower bound of (12).

Remark 4.5. The certified instance accuracy $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ is used to compare the verifiers on the CL encoder without labels. The smaller gap between robust instance accuracy $\mathcal{A}^{\epsilon}_{\mathrm{CL}}$ and $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ implies a stronger incomplete verifier (see experiments in § 6.3). However, as $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ is a function of the fixed attack strength ϵ, it is difficult to compare the robustness of two models unless one is uniformly better than the other for all strength ϵ. Thus, $\mathrm{ACR}_{\mathrm{CL}}$ is a more suitable choice than $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ for evaluating the robustness of CL encoders.

5. Theoretical Analysis for Robust Radius

What is the relationship between the robust radius R_{CL} and R_{LE}? This section demonstrates that R_{CL} is the upper bound of R_{LE}, which is further verified by the experimental results in § 6.1.

To provide the main insights, we first consider the situation in which only one positive sample is used. Formally, given an unlabeled sample $z = (x^+, \{x_i^-\}_{i=1}^K)$, we introduce the *margin distance* of x^+ as half of the minimum distance between $f(x^+)$ and $f(x_i^-)$, defined as $M \coloneqq \min_{i \in [K]} D_i$, where $D_i \coloneqq (1 - \rho(f(x^+), f(x_i^-)))/2$.

The idea is to estimate the lower tail of the distribution of M by fitting the λ smallest D_i of the negative samples x_i^-. We can then use this estimated distribution to produce the probability of a new point x falling into the margin of x^+, which can be interpreted as the probability of x being a positive sample of x^+. x is classified as a positive sample of x^+ if it is inside the margin of x^+ with high probability.

To estimate the distribution of the margin distance, we turn to the Fisher-Tippett-Gnedenko Theorem in extreme value theory (see the complete statement in Appendix A.1).

Lemma 5.1 (Fisher-Tippett-Gnedenko theorem (Coles et al., 2001))**.** *Let $X_1, X_2, \ldots$ be a sequence of independent random variables with common distribution function F. Let $M_n = \sup(X_1, \ldots, X_n)$. If there exists a sequence $a_n > 0, b_n \in \mathbb{R}$ such that*

$$\lim_{n \to \infty} \mathbb{P}\left(\frac{M_n - b_n}{a_n} \leq z\right) = G(z),$$

where G is a non-degenerate distribution function, then G belongs to either the Gumbel family, the Fréchet family or the Reverse Weibull family.

The theorem states that the maximum of a sequence of i.i.d. random variables after proper normalization can only converge to one of three possible distributions.

Theorem 5.2 (Margin distribution)**.** *Assume a continuous non-degenerate margin distribution exists. The distribution for margin distance M is then given by the Reverse Weibull distribution. The probability of x being a positive sample of x^+ is given by the following:*

$$\Psi(x; x^+, \alpha, \sigma) = \exp\left\{-\left(\frac{1 - \rho(f(x), f(x^+))}{\sigma}\right)^{\alpha}\right\},$$

where $\rho(f(x), f(x^+))$ is the instance similarity between x and x^+. $\alpha, \sigma > 0$ are Weibull shape and scale parameters, obtained from fitting to the λ smallest margin distances D_i.

See proof in Appendix A.2. Theorem 5.2 demonstrates that the probability of x being a positive sample can be given by the Reverse Weibull distribution fitting on finite samples. This enables us to compare the robust radius of the encoder and that of the downstream task. Intuitively, if the classifier can predict correctly with higher confidence, this implies that the classifier provides better certified robustness.

Theorem 5.3 (Robust radius bound)**.** *Given an encoder $f : \mathcal{X} \to \mathbb{R}^d$ and an unlabeled sample $z = (x^+, \{x_i^-\}_{i=1}^K)$, the downstream predictor $g : \mathbb{R}^d \to \mathbb{R}$ is trained on $\widehat{S} = \{(f(x^+), y_{c^+}), (f(x_i^-), y_{c^-})_{i=1}^K\}$. Then, for different negative samples x_i^-, we have*

$$\mathrm{R}_{\mathrm{CL}}(f; x^+, x_i^-) \geq \mathrm{R}_{\mathrm{LE}}(g; x^+, y_{c^+}).$$

Proof sketch. The possibility of downstream layer predicting x as positive can be given by Ψ_{LE}, which is fitted from margin distances $\{D_i\}_{i=1}^K$ computed by $\widehat{S}$. Ψ_{CL} is fitted from specific D_i, since $\mathrm{R}_{\mathrm{CL}}(f; x^+, x_i^-)$ is the robust radius between specific pair of positive and negative sample.

Figure 3 plots the cumulative distribution function (CDF) of Ψ_{CL} and Ψ_{LE}. $\Psi \to 1$ when $x \to x^+$, which means x is very likely to be the positive sample of x^+. If there exists negative samples between x_i^- and x^+ ("$-$" in Figure 3), then Ψ_{LE} will fit to these negative samples and make CDF grows slower than Ψ_{CL}, i.e., $\Psi_{\mathrm{CL}}(x) \geq \Psi_{\mathrm{LE}}(x)$. Lemma A.4 offers the correspondence that a higher probability to be a positive sample implies a larger robust radius, which recovers the theorem statement. See complete proof in Appendix A.3. □

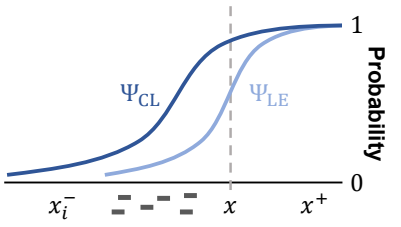


Figure 3. Probability of x as a positive sample. "$-$" means the negative sample other than x_i^-.

Remark 5.4. Theorem 5.3 implies that improving the robustness of the CL encoder enlarges the robust radius $\mathrm{R_{CL}}$, which can benefit the robustness of the downstream layer by providing a large upper bound of $\mathrm{R_{LE}}$. A larger $\mathrm{R_{CL}}$ implies that the model will achieve a higher robust performance on the downstream task; thus, it is reasonable to regard $\mathrm{R_{CL}}$ as a robustness metric.

In the above, we discuss the decision margin by analyzing the case with single x^+ and multiple x^-. Here, we discuss the robust radius of different x^+ by making the following theroem:

Theorem 5.5. *Given an encoder $f : \mathcal{X} \to \mathbb{R}^d$, two positive samples x_1^+, x_2^+ and one negative sample x^-, if $\rho(f(x_1^+), f(x^-)) \geq \rho(f(x_2^+), f(x^-))$, then*

$$\mathrm{R_{CL}}(f; x_1^+, x^-) \leq \mathrm{R_{CL}}(f; x_2^+, x^-).$$

See proof in Appendix A.4. Theorem 5.5 proves that if $f(x^+)$ is similar to $f(x^-)$, it is easy to recognize the adversarial sample x' of x^+ as a negative sample. In § 6.2, we empirically show that a vague image for which it is difficult to identify the class characteristic is easy to attack.

6. Experiments

In this section, we verify the effectiveness of our proposed RVCL by means of numerical experiments.

More specifically, § 6.1 demonstrates the effectiveness of the average certified radius for CL. § 6.2 shows that $\mathrm{ACR_{CL}}$ can evaluate the anti-disturbance ability of individual images. § 6.3 compares the strength of incomplete verifiers by means of certified instance accuracy. § 6.4 illustrates the verified lower bound $\underline{f}$ given by the verifiers. § 6.5 provides the sensitivity analysis for parameters in RVCL. Due to space limitations, we compare the efficiency of different verifiers in Appendix E.2.

Set-up. Our main experiments utilize four architectures: **Base**, **Deep** from Wang et al. (2021), **CNN-A**, **CNN-B** from Dathathri et al. (2020), from which the last layer is removed to form the CL encoders. SimCLR (Chen et al., 2020) and RoCL (Kim et al., 2020) are used in this paper for CL training and contrastive AT, respectively. Contrastive AT is trained with instance-wise adversarial samples with different attack strengths ϵ_{train}; $\epsilon_{train} = 0$ indicates that the encoder is trained with benign images. Using the same dataset as in previous deterministic verification works, all CL encoders are trained on MNIST (LeCun & Cortes, 2010) and CIFAR-10 (Krizhevsky & Hinton, 2009).

We utilize two incomplete verifiers with different verified tightness in the RVCL framework: CROWN (Zhang et al., 2018) and CBC (β-CROWN (Wang et al., 2021) for CL). A more detailed introduction to incomplete verifiers is presented in Appendix C.1.

Further details of the models and experimental settings are presented in Appendix D. The code can be found in the supplementary materials and GitHub.

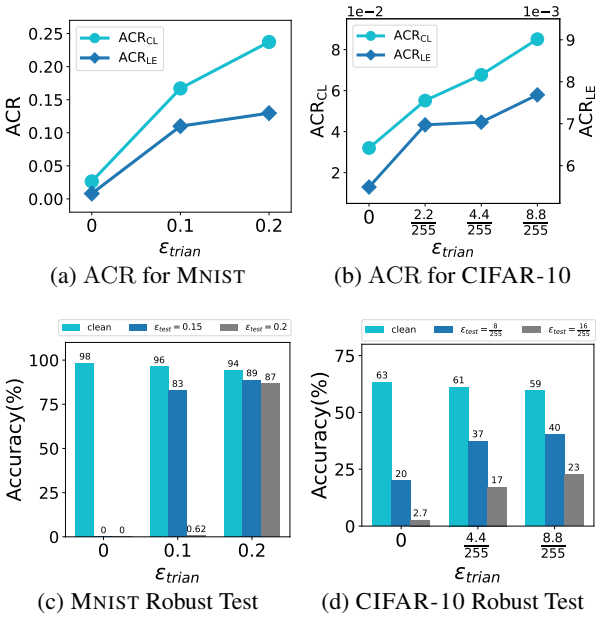
6.1. Average certified radius

In this subsection, we compute the average certified radius (ACR in § 4.2.2) over 100 test samples (i.e., $|S_{\mathtt{test}}| = |U_{\mathtt{test}}| = 100$) following previous verification works. For $\mathrm{ACR_{CL}}$, we set the number of negative samples $K = 10$. In the interests of efficiency, we use CROWN to compute the $\mathrm{ACR_{CL}}$, which is discussed further in Appendix E.2.

To determine whether the model robustness benefits from the encoder or linear classifier, we fix the encoder and fine-tune the downstream layer with benign images without perturbations. For the empirical robust test, we utilize supervised robust accuracy on the whole test dataset; this is the downstream classification accuracy over adversarial samples via label-wise PGD attack (Madry et al., 2018) with attack strength ϵ_{test}.

Note that a larger value of ϵ_{train} implies that a more robust encoder is obtained by contrastive AT. The results in Figure 4(a,b) show that $\mathrm{ACR_{CL}}$ and $\mathrm{ACR_{LE}}$ increase with increasing ϵ_{train} on the two datasets. Meanwhile, those in Figure 4(c,d) show that with different values of ϵ_{test}, the robust accuracy of the downstream classifier increases as ϵ_{train} increases. We therefore conclude that:

1. It is effective to measure the robustness using $\mathrm{ACR_{CL}}$ without labels and downstream tasks, because both the $\mathrm{ACR_{CL}}$ and supervised metric ($\mathrm{ACR_{LE}}$ and robust accuracy) grow consistently with increasing ϵ_{train}.
2. Figure 4(a,b) show that $\mathrm{ACR_{CL}}$ is larger than $\mathrm{ACR_{LE}}$ with the same ϵ_{train}, which supports our Theorem 5.3. Theorem 5.3 demonstrates that $\mathrm{R_{CL}}$ is the upper bound of $\mathrm{R_{LE}}$, while $\mathrm{ACR_{CL}}$ and $\mathrm{ACR_{LE}}$ are related to $\mathrm{R_{CL}}$ and $\mathrm{R_{LE}}$, respectively.



(a) ACR for MNIST (b) ACR for CIFAR-10

(c) MNIST Robust Test (d) CIFAR-10 Robust Test

Figure 4. (a,b) ACR_{CL} and ACR_{LE} on MNIST and CIFAR-10 with different values of ϵ_{train}. (c,d) Supervised robust accuracy on the downstream classifier. On MNIST, attack strengths ϵ_{test} are **0.15**, **0.2**; on CIFAR-10, attack strengths ϵ_{test} are $^{8}/_{255}$, $^{16}/_{255}$. **Clean** represents testing with benign images. (a,c) run on CNN-A, (b,d) run on CNN-B.

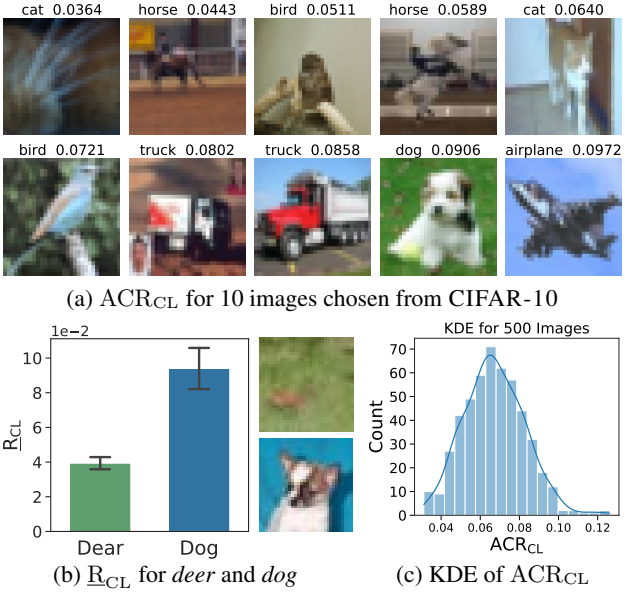
3. Figure 4(a,b) show that a robust encoder can significantly improve the model's robust performance on downstream tasks, since ACR_{LE} grows with increasing ϵ_{train} even though the downstream layer is learned on benign images.

6.2. Anti-disturbance ability of images

To study the robustness property of each image, in this subsection, we compute ACR_{CL} for a single test sample; i.e., $|U_{\texttt{test}}| = 1$, then $\text{ACR}_{\text{CL}} \coloneqq \frac{1}{K}\sum_{i=1}^{K} \underline{\text{R}}_{\text{CL}}(f; x^+, x_i^-)$.

We sample two images from CIFAR-10, as shown in Figure 5(b). The above image is labeled as *deer*, which is vague and makes it difficult to identify the latent class. The below image is labeled as *dog*, which is much clearer than *deer*. We calculate $\underline{\text{R}}_{\text{CL}}$ with fifty negative samples ($K = 50$). Our findings suggest that the $\underline{\text{R}}_{\text{CL}}$ of *deer* is significantly smaller than that of *dog*; this means that the distance between the feature of *deer* and its negative samples is smaller than that between the feature of *dog* and its negative samples, which supports our Theorem 5.5. We can therefore conclude that the anti-disturbance ability of *dog* is stronger than that of *deer*, which means that *dog* is l_∞^ϵ-*verified* with a larger ϵ than *deer*. These results verify that ACR_{CL} is able to quantify the anti-disturbance ability of images.

We sample 10 more images from CIFAR-10, and plot the images and their ACR_{CL} in Figure 5(a). It comes to the same conclusion: the vague image which is difficult to identify the latent class has a low ACR_{CL}. We further visualize the distribution of ACR_{CL} for 500 images from CIFAR-10 by calculating ACR_{CL} with $K = 20$, and additionally provide the kernel density plot to show the distribution (see Figure 5(c)). About 90% of images' ACR_{CL} are distributed within the range $[0.044, 0.093]$, concentrated around 0.07.



(a) ACR_{CL} for 10 images chosen from CIFAR-10

(b) $\underline{\text{R}}_{\text{CL}}$ for *deer* and *dog* (c) KDE of ACR_{CL}

Figure 5. (a,b,c) run on CNN-B with $\epsilon_{train} = {}^{4.4}/_{255}$. (a) ACR_{CL} for images, which are randomly chosen from CIFAR-10; $K = 50$. (b) $\underline{\text{R}}_{\text{CL}}$ for two images from CIFAR-10; $K = 50$. (c) Frequency distribution histogram and kernel density estimation (KDE) of ACR_{CL} for 500 images from CIFAR-10; $K = 20$.

6.3. Certified instance accuracy

Table 1. Comparison of certified instance accuracy across various networks and attack strength ϵ_{test} on CIFAR-10. The number of test samples $|U_{\texttt{test}}| = 100$.

ϵ_{test}	Model	ϵ_{train}	Instance Accuracy	Certified Instance Accuracy	
			PGD	**CBC**	**CROWN**
$\frac{2}{255}$	CNN-B	0	100%	97%	96%
		$\frac{2.2}{255}$	100%	100%	100%
		$\frac{4.4}{255}$	91%	26%	11%
		$\frac{8.8}{255}$	100%	55%	34%
$\frac{4}{255}$	CNN-B	$\frac{4.4}{255}$	100%	68%	52%
	Based		100%	99%	95%
	Deep		100%	96%	84%
	CNN-A		99%	91%	81%
$\frac{8}{255}$	CNN-B	$\frac{8.8}{255}$	1%	0%	0%

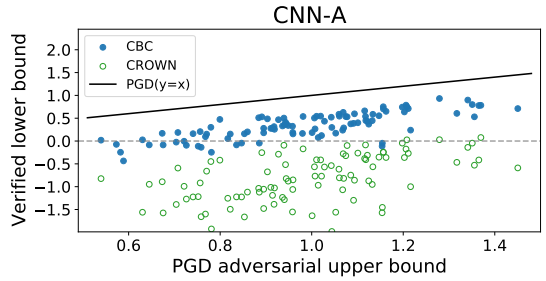
In this subsection, we study the performance of incomplete verifers on the CL encoder via certified instance accuracy ($\underline{\mathcal{A}}_{\text{CL}}^{\epsilon}$ in Definition 4.4). Table 1 summarizes some of the results on CIFAR-10. Due to space limitations, we provide detailed settings and explanations for more experimental results (on MNIST) in Appendix E.1. The two incomplete

verifiers we utilize herein are CROWN and CBC; CBC is the state-of-the-art verifier, which is more powerful than CROWN for supervised verification (Wang et al., 2021).

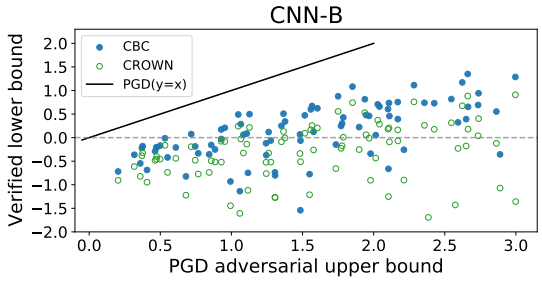
$\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ is the lower bound of the robust instance accuracy $\mathcal{A}^{\epsilon}_{\mathrm{CL}}$ (Definition 4.3, obtained by instance-wise PGD attack (Kim et al., 2020)). The tighter lower bound given by $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ indicates a stronger incomplete verifier. Table 1 shows that the gap between $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ given by CBC and $\mathcal{A}^{\epsilon}_{\mathrm{CL}}$ given by PGD is smaller than that between CROWN and PGD, which shows that CBC is a stronger verifier than CROWN.

We further illustrate the verified lower bound $\underline{f}$ given by these two incomplete verifiers in § 6.4, from which the same conclusions can be drawn. All these results demonstrate the efficacy of our proposed RVCL: a stronger supervised verifier can still achieve a tighter certified radius in the RVCL framework.

6.4. Tightness of verification



(a) MNIST, $\epsilon_{train} = 0.3$, $\epsilon_{test} = 0.3$



(b) CIFAR-10, $\epsilon_{train} = \frac{4.4}{255}$, $\epsilon_{test} = \frac{4}{255}$

Figure 6. **Comparing the tightness of verifiers.** For 100 test samples on MNIST and CIFAR-10. (a) runs on CNN-A, (b) runs on CNN-B. We plot the verified lower bound $\underline{f}(x^+, x^-, \epsilon)$ against PGD upper bound $\overline{f}$. Some points exceed the plotted axes limits.

Instance-wise PGD attack (Kim et al., 2020) provides the upper bound of minimum distortion, $\overline{f} \geq \widetilde{f}$, while RVCL provides the lower bound, $\widetilde{f} \geq \underline{f}(x^+, x^-, \epsilon)$. It should be noted that, unlike with supervised verification (Dathathri et al., 2020; Wang et al., 2021), all distortion here is instance-wise. (x^+, x^-) is *verified* if $\underline{f}(x^+, x^-, \epsilon) > 0$, meaning that x^+ cannot be disturbed to x^- with attack strength ϵ_{test}. The closer the verified lower bound $\underline{f}(x^+, x^-, \epsilon)$ is to the PGD upper bound $\overline{f}$ ($y = x$ in Figure 6), the stronger the verifier would be.

One hundred test samples ($|U_{\mathtt{test}}| = 100$) are used to illustrate the tightness of the verification. As Figure 6 shows, the points above the dotted line are successfully verified. CBC achieves tight verification across all samples, and furthermore consistently outperforms CROWN on MNIST and CIFAR-10. This result is consistant with supervised verification in Wang et al. (2021), which demonstrates the effectiveness of RVCL.

6.5. Sensitivity analysis

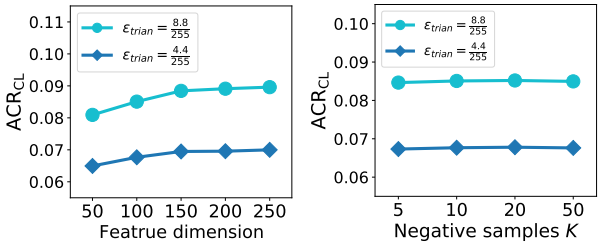


Figure 7. **Left:** Sensitivity analysis of feature dimension influencing $\mathrm{ACR}_{\mathrm{CL}}$. **Right:** Sensitivity analysis of the number of negative samples K influencing $\mathrm{ACR}_{\mathrm{CL}}$. Experiments run on CNN-B with different ϵ_{train} on CIFAR-10.

Feature dimension We investigate the influence of the feature dimension on $\mathrm{ACR}_{\mathrm{CL}}$ (see **Left** of Figure 7). From the result, we can determine that $\mathrm{ACR}_{\mathrm{CL}}$ increases slightly with the growing feature dimension, then remains stable on dimensions larger than 150. The results illustrate that $\mathrm{ACR}_{\mathrm{CL}}$ is not sensitive to feature dimension.

Number of negative samples We validate the influence of the number of negative samples K on $\mathrm{ACR}_{\mathrm{CL}}$ (see **Right** of Figure 7). The result shows that $\mathrm{ACR}_{\mathrm{CL}}$ is not sensitive to K with different values of ϵ_{train}, which means that we can use small values of K to efficiently evaluate the model robustness or the anti-disturbance ability of an image.

7. Conclusion

In this paper, we tackle the robustness verification problem for CL without any labels, and accordingly propose a novel RVCL framework that does not depend on any class labels, downstream tasks or specific attack algorithms. We then use extreme value theory to reveal the quantitative relationship between the robust radius of the CL encoder and that of the downstream task. All our experiments show that RVCL is an efficient robustness framework for CL encoders, and can also be used to evaluate the anti-disturbance ability of images. Moreover, our experimental results verify our theory. We believe that RVCL is a novel perspective from which to understand robustness on contrastive learning.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant 61976161.

References

Alayrac, J., Uesato, J., Huang, P., Fawzi, A., Stanforth, R., and Kohli, P. Are labels required for improving adversarial robustness? In *NeurIPS*, pp. 12192–12202, 2019.

Anderson, R., Huchette, J., Ma, W., Tjandraatmadja, C., and Vielma, J. P. Strong mixed-integer programming formulations for trained neural networks. *Math. Program.*, 183(1):3–39, 2020.

Bunel, R., Turkaslan, I., Torr, P. H. S., Kohli, P., and Mudigonda, P. K. A unified view of piecewise linear neural network verification. In *NeurIPS*, pp. 4795–4804, 2018.

Carlini, N. and Wagner, D. A. Towards evaluating the robustness of neural networks. In *SP*, pp. 39–57, 2017.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. Unlabeled data improves adversarial robustness. In *NeurIPS*, pp. 11190–11201, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119, pp. 1597–1607, 2020.

Coles, S., Bawa, J., Trenner, L., and Dorazio, P. *An introduction to statistical modeling of extreme values*, volume 208. 2001.

Dathathri, S., Dvijotham, K., Kurakin, A., Raghunathan, A., Uesato, J., Bunel, R., Shankar, S., Steinhardt, J., Goodfellow, I. J., Liang, P., and Kohli, P. Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. In *NeurIPS*, 2020.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *ICCV*, pp. 1422–1430, 2015.

Ehlers, R. Formal verification of piece-wise linear feed-forward neural networks. In *Automated Technology for Verification and Analysis*, volume 10482, pp. 269–286, 2017.

Fan, L., Liu, S., Chen, P.-Y., Zhang, G., and Gan, C. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? In *NeurIPS*, 2021.

Gibert, X., Patel, V. M., and Chellappa, R. Sequential score adaptation with extreme value theory for robust railway track inspection. In *ICCV*, pp. 131–138, 2015.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Gowal, S., Huang, P., van den Oord, A., Mann, T., and Kohli, P. Self-supervised adversarial robustness for the low-label, high-data regime. In *ICLR*, 2021.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9726–9735, 2020.

Ho, C. and Vasconcelos, N. Contrastive learning with adversarial examples. In *NeurIPS*, 2020.

Jiang, Z., Chen, T., Chen, T., and Wang, Z. Robust pre-training by adversarial contrastive learning. In *NeurIPS*, 2020.

Jing, L. and Tian, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(11):4037–4058, 2021.

Katz, G., Barrett, C. W., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient SMT solver for verifying deep neural networks. In *CAV*, volume 10426, pp. 97–117, 2017.

Kim, M., Tack, J., and Hwang, S. J. Adversarial self-supervised contrastive learning. In *NeurIPS*, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

LeCun, Y. and Cortes, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2010.

Li, L., Qi, X., Xie, T., and Li, B. Sok: Certified robustness for deep neural networks. *CoRR*, abs/2009.04131, 2020.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Moosavi-Dezfooli, S., Fawzi, A., and Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, pp. 2574–2582, 2016.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, volume 9910, pp. 69–84, 2016.

Nozawa, K., Germain, P., and Guedj, B. Pac-bayesian contrastive unsupervised representation learning. In *UAI*, volume 124, pp. 21–30, 2020.

Rudd, E. M., Jain, L. P., Scheirer, W. J., and Boult, T. E. The extreme value machine. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):762–768, 2018.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, volume 97, pp. 5628–5637, 2019.

Scheirer, W. J., Rocha, A., Micheals, R. J., and Boult, T. E. Meta-recognition: The theory and practice of recognition score analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1689–1695, 2011.

Scheirer, W. J., Kumar, N., Belhumeur, P. N., and Boult, T. E. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, pp. 2933–2940, 2012.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In *NeurIPS*, pp. 5019–5031, 2018.

Sinha, A., Namkoong, H., and Duchi, J. C. Certifying some distributional robustness with principled adversarial training. In *ICLR*, 2018.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *ECCV*, volume 12356, pp. 776–794, 2020.

Tjeng, V., Xiao, K. Y., and Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. In *ICLR*, 2019.

Uesato, J., O'Donoghue, B., Kohli, P., and van den Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In *ICML*, volume 80, pp. 5032–5041, 2018.

Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C., and Kolter, J. Z. Beta-crown: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. In *NeurIPS*, 2021.

Weng, T., Zhang, H., Chen, P., Yi, J., Su, D., Gao, Y., Hsieh, C., and Daniel, L. Evaluating the robustness of neural networks: An extreme value theory approach. In *ICLR*, 2018.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, volume 80, pp. 5283–5292, 2018.

Wong, E., Schmidt, F. R., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *NeurIPS*, pp. 8410–8419, 2018.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pp. 3733–3742, 2018.

Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K., Huang, M., Kailkhura, B., Lin, X., and Hsieh, C. Automatic perturbation analysis for scalable certified robustness and beyond. In *NeurIPS*, 2020.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I. P., and Li, J. Randomized smoothing of all shapes and sizes. In *ICML*, volume 119, pp. 10693–10705, 2020.

Zhai, R., Cai, T., He, D., Dan, C., He, K., Hopcroft, J. E., and Wang, L. Adversarially robust generalization just requires more unlabeled data. *CoRR*, abs/1906.00555, 2019.

Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C., and Wang, L. MACER: attack-free and scalable robust training via maximizing certified radius. In *ICLR*, 2020.

Zhang, H., Weng, T., Chen, P., Hsieh, C., and Daniel, L. Efficient neural network robustness certification with general activation functions. In *NeurIPS*, pp. 4944–4953, 2018.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, volume 97, pp. 7472–7482, 2019.

A. Proofs

A.1. Extreme Value Theory

Before provding the proofs of main results, we first provide two important lemmas in extreme value theory (EVT).

Lemma A.1 (Fisher-Tippett-Gnedenko theorem (Coles et al., 2001)). *Let $X_1, X_2, \ldots$ be a sequence of independent random variables with common distribution function F. Let $M_n = \max(X_1, \ldots, X_n)$. If there exists a sequence $a_n > 0, b_n \in \mathbb{R}$ such that*

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{M_n - b_n}{a_n} \leq z\right) = G(z), \tag{A.1}$$

where G is a non-degenerate distribution function, then G belongs to either the Gumbel family (Type I), the Fréchet family (Type II) or the Reverse Weibull family (Type III) with their CDFs as follows:

$$\text{Gumbel family (Type I):} \quad G(z) = \exp\left\{-\exp\left[-\left(\frac{z-b}{a}\right)\right]\right\}, \quad z \in \mathbb{R},$$

$$\text{Fréchet family (Type II):} \quad G(z) = \begin{cases} 0, & \text{if } z < b, \\ \exp\left\{-\left(\frac{z-b}{a}\right)^{-\alpha}\right\}, & \text{if } z \geq b, \end{cases}$$

$$\text{Reverse Weibull family (Type III):} \quad G(z) = \begin{cases} \exp\left\{-\left(\frac{b-z}{a}\right)^{\alpha}\right\}, & \text{if } z < b, \\ 1, & \text{if } z \geq b, \end{cases}$$

where $a > 0$, $b \in \mathbb{R}$ and $\alpha > 0$ are the scale, location and shape parameters, respectively.

Lemma A.1 states that the rescaled sample maxima $(M_n - b_n)/a_n$ converge in distribution to a variable that has a distribution within one of three families. Furthermore, these three families can be combined into a single family called generalized extreme value (GEV) distribution, which is a family of continuous probability distributions developed within extreme value theory. The Gumbel, Fréchet and Reverse Weibull families are special cases of GEV distribution.

Lemma A.2 (Generalized Extreme Value (GEV) distribution (Coles et al., 2001)). *Let $X_1, X_2, \ldots$ be a sequence of i.i.d. samples from the distribution function F. Let $M_n = \max(X_1, \ldots, X_n)$. If there exists a sequence $a_n > 0$, $b_n \in \mathbb{R}$ such that $\lim_{n\to\infty} \mathbb{P}(\frac{M_n - b_n}{a_n} \leq z) = G(z)$, then if G is a non-degenerate distribution function, it belongs to the class of generalized extreme value (GEV) distributions with*

$$G(z) = \exp\left[-\left\{1 + \xi\left(\frac{z-\mu}{\sigma}\right)\right\}^{-1/\xi}\right], \quad z \in \mathbb{R} : 1 + \xi\left(\frac{z-\mu}{\sigma}\right) > 0, \tag{A.2}$$

where $\xi \in \mathbb{R}$, $\mu \in \mathbb{R}$ and $\sigma > 0$ are the shape, location and scale parameters, respectively.

As the special case, the Reverse Weibull family in Lemma A.1 can be derived by Lemma A.2, which let $\xi < 0$ and the upper endpoint of F be denoted by b, then $\alpha = -1/\xi > 0$.

A.2. Proof of Theorem 5.2

Theorem 5.2 (Margin distribution). *Assume a continuous non-degenerate margin distribution exists. The distribution for margin distance M is then given by the Reverse Weibull distribution. The probability of x being a positive sample of x^+ is given by the following:*

$$\Psi(x; x^+, \alpha, \sigma) = \exp\left\{-\left(\frac{1 - \rho(f(x), f(x^+))}{\sigma}\right)^{\alpha}\right\}, \tag{A.3}$$

where $\rho(f(x), f(x^+))$ is the instance similarity between x and x^+. $\alpha, \sigma > 0$ are Weibull shape and scale parameters, obtained from fitting to the λ smallest margin distances D_i.

Proof. From the assume we know that $G(z)$ in Lemma A.1 exists. Since Lemma A.1 applies to maxima, we transform the variables via $\overline{M} = \max_{i\in[K]} -D_i$, $D_i := (1 - \rho(f(x^+), f(x_i^-))/2$. Because $-D_i$ is bounded ($-D_i < 0$), so the asymptotic distribution of $\overline{M}$ converges to the Reverse Weibull distribution:

$$W(z) = \begin{cases} \exp\left\{-\left(-\frac{z}{\sigma}\right)^{\alpha}\right\}, & \text{if } z < 0, \\ 1, & \text{if } z \geq 0, \end{cases} \tag{A.4}$$

where $\alpha > 0$, b is the upper endpoint of F, σ is the scale parameters. $b = 0$ in (A.4) since $\overline{M}$ is bounded above by 0 as a negative distance. We use margin distances D_i of the λ closest samples with x^+ to estimate the parameters α and σ, which means to estimate $\widehat{W}$ of the distribution funstion W.

The margin distance between x and x^+ defined as $1 - \rho(f(x), f(x^+))$. Then we focus on the probability of x included in the margin of x^+, which can be written as:

$$\begin{aligned}\mathbb{P}(1 - \rho(f(x), f(x^+)) < \min(D_1, \ldots, D_n)) &= \mathbb{P}(-\min(D_1, \ldots, D_n) < \rho(f(x), f(x^+)) - 1) \\ &= \mathbb{P}(\max(-D_1, \ldots, -D_n) < \rho(f(x), f(x^+)) - 1) \\ &= \mathbb{P}(\overline{M} < \rho(f(x), f(x^+)) - 1) \\ &= \widehat{W}(\rho(f(x), f(x^+)) - 1).\end{aligned} \tag{A.5}$$

Since $\rho(f(x), f(x^+)) - 1 < 0$, we can rewrite (A.5) as (A.3). Overall, we conclude our proof. □

A.3. Proof of Theorem 5.3

Theorem 5.3 (Robust radius bound). *Given an encoder $f : \mathcal{X} \to \mathbb{R}^d$ and an unlabeled sample $z = (x^+, \{x_i^-\}_{i=1}^K)$, the downstream predictor $g : \mathbb{R}^d \to \mathbb{R}$ is trained on $\widehat{S} = \{(f(x^+), y_{c^+}), (f(x_i^-), y_{c^-})_{i=1}^K\}$. Then, for different negative samples x_i^-, we have*

$$\mathrm{R}_{\mathrm{CL}}(f; x^+, x_i^-) \geq \mathrm{R}_{\mathrm{LE}}(g; x^+, y_{c^+}). \tag{A.6}$$

Before we formally prove Theorem 5.3, we first provide the following two lemmas.

Lemma A.3 focuses on a model for the k largest order statistics. It extends the result in Lemma A.1 to extreme order statistics, by defining $M_n^{(k)} = k$-th largest of $\{X_1, \ldots, X_n\}$ and further identifying the limiting behavior of this variable, for fixed k, as $n \to \infty$. Lemma A.3 implies that, if the k-th largest order statistic is normalized in exactly the same way as the maximum, then its limiting distribution is of the form given by (A.8).

Lemma A.3 (k-th largest order statistic (Coles et al., 2001))**.** *Let $X_1, X_2, \ldots$ be a sequence of i.i.d. samples from the distribution function F. Let $M_n = \max(X_1, \ldots, X_n)$. If there exists a sequence $a_n > 0$, $b_n \in \mathbb{R}$ such that $\lim_{n\to\infty} \mathbb{P}(\frac{M_n - b_n}{a_n} \leq z) = G(z)$ for a non-degenerate distribution function G, so that G is the GEV distribution function given by (A.2), then, for fixed k,*

$$\lim_{n\to\infty} \mathbb{P}((M_n^{(k)} - b_n)/a_n) = G_k(z) \tag{A.7}$$

on $\{z : 1 + \frac{\xi(z-\mu)}{\sigma} > 0\}$, where

$$G_k(z) = \exp\{-\tau(z)\} \sum_{s=0}^{k-1} \frac{\tau(z)^s}{s!} \tag{A.8}$$

with $\tau(z) = \left\{1 + \xi\left(\frac{z-\mu}{\sigma}\right)\right\}^{-1/\xi}$.

Intuitively, the classifier predicting correctly with higher confidence implies that the classifier can provide better certified robustness. Lemma A.4 provides this relationship directly.

Lemma A.4 (Robust radius (Yang et al., 2020))**.** *The robust radius in any norm $\|\cdot\|$ is at least*

$$\mathrm{R} := \int_{1-\lambda}^{1/2} \frac{1}{\Phi(p)} \mathrm{d}p, \tag{A.9}$$

where $\Phi(p) := \sup_{\|v\|=1} \sup_{U \subseteq \mathbb{R}^d : q(U) = p} \lim_{r \searrow 0} \frac{q(U - rv) - p}{r}$, λ is the probability that the binary classifier predicts the right label under perturbation, $q(U)$ is the measure of U under q, i.e. $q(U) = \Pr_{\delta \sim q}(\delta \in U)$, v is the perturbation vector.

Finally, we prove Theorem 5.3.

Proof. Ψ_{LE} is obtained from fitting to margin distances $\{D_i\}_{i=1}^K, D_i \coloneqq (1-\rho(f(x^+), f(x_i^-))/2$. From Lemma A.1 we know that (A.3) is the Reverse Weibull distribution and can be written as the form of (A.2):

$$\Psi_{\mathrm{LE}}(x; x^+, \alpha, \sigma) = \exp\left[-\left\{1+\xi\left(\frac{z-\mu}{\sigma}\right)\right\}^{-1/\xi}\right] \tag{A.10}$$

with $z = -D = \rho(f(x), f(x^+)) - 1$.

$\mathrm{R}_{\mathrm{CL}}(f; x^+, x_i^-)$ is defined on the positive and negative sample pair (x^+, x_i^-); it means that Ψ_{CL} is fitted from specific (x^+, x_i^-). We denote $-D^{(k)}$ for x_i^- is the k-th largest order statistic of $\{-D_1, \ldots, -D_K\}$. By Lemma A.3, the distribution function Ψ_{CL} for $-D^{(k)}$ can be written as:

$$\Psi_{\mathrm{CL}}(x; x^+, \alpha, \sigma, k) = \exp\{-\tau(z)\}\sum_{s=0}^{k-1}\frac{\tau(z)^s}{s!} \tag{A.11}$$

with $\tau(z) = \left\{1+\xi\left(\frac{z-\mu}{\sigma}\right)\right\}^{-1/\xi}, z = \rho(f(x), f(x^+)) - 1$. ξ, μ, σ are the same with (A.10).

Let $x \sim \mathcal{D}_{c^+}$ be the positive test sample of x^+. As we discuss in § 3.1, the latent label for unlabeled positive sample can be given by y_{c^+}. In § 4, we propose that judging positive sample correctly on CL encoder and downstream task can be transformed to judge whether or not $\mathbf{W}_{\mathrm{CL}} f(x) > 0$ and $y_{c^+} \cdot g(x) > 0$ are True, respectively. Thus, for every negative sample x_i^-, we have

$$\begin{aligned}
\mathbb{P}(\mathbf{W}_{\mathrm{CL}} f(x) > 0 \mid x) - \mathbb{P}(y_{c^+} \cdot g(x) > 0 \mid x) &= \Psi_{\mathrm{CL}}(x; x^+, \alpha, \sigma, k) - \Psi_{\mathrm{LE}}(x; x^+, \alpha, \sigma) \\
&= \exp\{-\tau(z)\}\sum_{s=0}^{k-1}\frac{\tau(z)^s}{s!} - \exp\left[-\left\{1+\xi\left(\frac{z-\mu}{\sigma}\right)\right\}^{-1/\xi}\right] \\
&= \exp\{-\tau(z)\}\sum_{s=1}^{k-1}\frac{\tau(z)^s}{s!} \\
&\geq 0
\end{aligned} \tag{A.12}$$

with $\tau(z) = \left\{1+\xi\left(\frac{z-\mu}{\sigma}\right)\right\}^{-1/\xi}$, $z = \rho(f(x), f(x^+)) - 1$. Equality holds if and only if $k = 1$. Recall that $x = x^+ + \delta, \|\delta\|_\infty \leq \epsilon$, i.e. $x \in \mathcal{B}_\infty(x^+, \epsilon)$, which is in the constraints of (6) and (9).

By Lemma A.4, we have:

$$\begin{aligned}
\mathrm{R}_{\mathrm{CL}}(f; x^+, x_i^-) - \mathrm{R}_{\mathrm{LE}}(g; x^+, y_{c^+}) &= \int_{1-\Psi_{\mathrm{CL}}(x;x^+,\alpha,\sigma,k)}^{1/2}\frac{1}{\Phi(p)}\mathrm{d}p - \int_{1-\Psi_{\mathrm{LE}}(x;x^+,\alpha,\sigma)}^{1/2}\frac{1}{\Phi(p)}\mathrm{d}p \\
&= \int_{1-\Psi_{\mathrm{CL}}(x;x^+,\alpha,\sigma,k)}^{1-\Psi_{\mathrm{LE}}(x;x^+,\alpha,\sigma)}\frac{1}{\Phi(p)}\mathrm{d}p \\
&\geq 0,
\end{aligned} \tag{A.13}$$

which recovers the theorem statement. Equality holds if and only if $k = 1$. □

A.4. Proof of Theorem 5.5

Theorem 5.5. *Given an encoder $f : \mathcal{X} \to \mathbb{R}^d$, two positive samples x_1^+, x_2^+ and one negative sample x^-, if $\rho(f(x_1^+), f(x^-)) \geq \rho(f(x_2^+), f(x^-))$, then*

$$\mathrm{R}_{\mathrm{CL}}(f; x_1^+, x^-) \leq \mathrm{R}_{\mathrm{CL}}(f; x_2^+, x^-). \tag{A.14}$$

Proof. In this proposition, we consider the situation in which multiple positive samples x^+ and only one negative sample x^- are used. Formally, given an unlabeled sample $z = (\{x_i^+\}_{i=1}^K, x^-)$, we define the margin distance of x^- as $M^- \coloneqq \min_{i\in[K]} D_i^-$, where $D_i^- \coloneqq (1-\rho(f(x_i^+), f(x^-))/2$.

We denote the lower tail of the distribution of M^- as Ψ_{neg}. Similar to the idea of Theorem 5.2, we use Ψ_{neg} to produce the probability of x falling into the margin of x^-, which can be interpreted as the probability of x being a negative sample. From Lemma A.1, we know that Ψ_{neg} also converges to the Reverse Weibull distribution, and can be written to the form as following:

$$\Psi_{\text{neg}}(x; x^-, \alpha, \sigma) = \exp\left\{-\left(\frac{1-\rho(f(x), f(x^-))}{\sigma}\right)^{\alpha}\right\}, \tag{A.15}$$

where $\rho(f(x), f(x^-))$ is the instance similarity between x and x^-. $\alpha, \sigma > 0$ are Weibull shape and scale parameters, obtained from fitting to the λ smallest margin distances D_i^-.

Cumulative distribution function Ψ_{neg} is monotonic increasing. Given two positive samples x_1^+, x_2^+, if $\rho(f(x_1^+), f(x^-)) \geq \rho(f(x_2^+), f(x^-))$, we have:

$$\Psi_{\text{neg}}(x_1^+; x^-, \alpha, \sigma) \geq \Psi_{\text{neg}}(x_2^+; x^-, \alpha, \sigma) \tag{A.16}$$

x being the positive sample of x^+ means that x is more similar to x^+ than to x^-. From Lemma A.4 we have:

$$\begin{aligned}
\mathrm{R}_{\text{CL}}(f; x_1^+, x^-) - \mathrm{R}_{\text{CL}}(f; x_2^+, x^-) &= \int_{\Psi_{\text{neg}}(x_1^+; x^-, \alpha, \sigma)}^{1/2} \frac{1}{\Phi(p)} \mathrm{d}p - \int_{\Psi_{\text{neg}}(x_2^+; x^-, \alpha, \sigma)}^{1/2} \frac{1}{\Phi(p)} \mathrm{d}p \\
&= -\int_{\Psi_{\text{neg}}(x_2^+; x^-, \alpha, \sigma)}^{\Psi_{\text{neg}}(x_1^+; x^-, \alpha, \sigma)} \frac{1}{\Phi(p)} \mathrm{d}p \\
&\leq 0,
\end{aligned} \tag{A.17}$$

which recovers the theorem statement. Equality holds if and only if $\rho(f(x_1^+), f(x^-)) = \rho(f(x_2^+), f(x^-))$. □

B. Difference with CLEVER

CLEVER (Cross-Lipschitz Extreme Value for nEtwork Robustness) (Weng et al., 2018) estimates the robust radius R using extreme value theory (EVT). In this paper, we utilize EVT in a totally different way compared with CLEVER:

1. To produce the probability of x being a positive sample of x^+, we utilize EVT to estimate the lower tail of the **margin distance** (Theorem 5.2); thus, we can compare the probability given by the CL encoder and the downstream task. While CLEVER focuses on estimating R by proposing a sampling based approach with EVT to estimate the **local Lipschitz constant**; it is based on a theoretical analysis of formal robustness guarantee with Lipschitz continuity assumption.
2. We use EVT to reveal the quantitative relationship between the robust radius of the CL encoder and that of the downstream task. While CLEVER only focuses on the robust radius for supervised verification.

C. Complete Implementation

We first introduce two incomplete verifiers with different verified tightness used for RVCL. Then we present the complete implementation for verified prediction and certification.

C.1. Incomplete verifiers

Due to the nonlinear activations $\sigma(\cdot)$, the feasible set of (6) and (9) is nonconvex. One intuitive idea is to perform the convex relaxation of the feasible set to build incomplete verifiers. This paper discusses ReLU networks with CROWN (Zhang et al., 2018), which is a method used to relax the nonconvex equality constraints $\widehat{\phi}_k(\cdot) = \sigma(\phi_k(\cdot))$ to convex inequality constraints.

Let $\boldsymbol{l}_k^{(j)}$ and $\boldsymbol{u}_k^{(j)}$ be the lower and upper bound of $\phi_k^{(j)}$, i.e. $\boldsymbol{l}_k^{(j)} < \phi_k^{(j)} < \boldsymbol{u}_k^{(j)}$, $k \in [L]$. Given the ReLU activation function $\sigma(y) = \max(y, 0)$, CROWN uses linear constraints to relax ReLU: $\boldsymbol{\alpha}_j^{(i)} \phi_k^{(j)} \leq \widehat{\phi}_k^{(j)} \leq \frac{\boldsymbol{u}_k^{(j)}}{\boldsymbol{u}_k^{(j)} - \boldsymbol{l}_k^{(j)}} \left(\phi_k^{(j)} - \boldsymbol{l}_k^{(j)}\right)$, where $0 \leq \boldsymbol{\alpha}_j^{(i)} \leq 1$. After convex relaxation, (6) and (9) can be efficiently solved, as follows:

Lemma C.1 (CROWN bound (Zhang et al., 2018))**.** *Given an L-layer NN $\phi_L : \mathbb{R}^{d_0} \to \mathbb{R}$ with weights $\mathbf{W}_k$ and bias $\boldsymbol{b}_k$, and the pre-activation bound $\boldsymbol{l}_k^{(j)} < \phi_k^{(j)} < \boldsymbol{u}_k^{(j)}$ ($k \in [L], j \in [d_k]$), $x' \in \mathcal{B}(x, \epsilon)$, we have:*

$$\min_{x'} \mathbf{W}_L \widehat{\phi}_{L-1}(x') + \boldsymbol{b}_L \geq \min_{x'} \boldsymbol{c}^\top x' + c_0 \tag{C.18}$$

where $\boldsymbol{c}$ and c_0 can be computed by $\mathbf{W}_k$, $\boldsymbol{b}_k$, $\boldsymbol{l}_k^{(j)}$, $\boldsymbol{u}_k^{(j)}$.

Another incomplete verifier stronger than CROWN is β-CROWN (Wang et al., 2021) which is the state-of-the-art verification method. β-CROWN uses a few steps of gradient ascent to achieve bounds as tight as possible but suffer from high time cost.

Lemma C.2 (β-CROWN bound (Wang et al., 2021))**.** *Given an L-layer NN $\phi_L : \mathbb{R}^{d_0} \to \mathbb{R}$ with weights $\mathbf{W}_k$ and bias $\boldsymbol{b}_k$, the pre-activation bound $\boldsymbol{l}_k^{(j)} < \phi_k^{(j)} < \boldsymbol{u}_k^{(j)}$, $x' \in \mathcal{B}(x, \epsilon)$, and split constraints $z \in \mathcal{Z}$, we have:*

$$\min_{x', z\in\mathcal{Z}} \mathbf{W}_L \widehat{\phi}_{L-1}(x') + \boldsymbol{b}_L \geq \max_{\boldsymbol{\beta}\geq 0} \min_{x'} (\boldsymbol{a} + \boldsymbol{P}\boldsymbol{\beta})^\top x' + \boldsymbol{q}^\top \boldsymbol{\beta} + c_0 \tag{C.19}$$

where $\boldsymbol{a}, \boldsymbol{P}, \boldsymbol{q}$ and c_0 can be computed by $\mathbf{W}_k$, $\boldsymbol{b}_k$, $\boldsymbol{l}_k^{(j)}$, $\boldsymbol{u}_k^{(j)}$, $\boldsymbol{\beta}$ is the multiplier of Lagrange function.

We modify two supervised verifiers with different verification tightness in order to show that: a stronger verifier can still achieve a tighter certified radius $\underline{\mathrm{R}}_{\mathrm{CL}}$ in RVCL framework, which illustrates the efficacy of RVCL. The related experiment results are in § 6.3.

The procedure of incomplete verifier is denoted by INCOMPLETEVERIFIER in the next subsection, which aims to give the verified lower bound $\underline{f}(x^+, x^-, \epsilon)$ of the function $\widetilde{f}$ in (9).

C.2. Pseudocode

In terms of verified prediction for CL, (x^+, x^-) being verified as "correct" with strength ϵ means that f is l_∞^ϵ-*verified* at (x^+, x^-). Similar to the discussion of supervised verification in § 3.2, if $\underline{f}(x^+, x^-, \epsilon)$ given by the procedure INCOMPLETEVERIFIER is greater than 0, (x^+, x^-) is verified to be "correct". The procedure of verified prediction is presented in pseudocode as PREDICT. We utilize PREDICT to obtain the certified instance accuracy ($\underline{\mathcal{A}}_{\mathrm{CL}}^\epsilon$ in Definition 4.4) for the test dataset $U_{\texttt{test}}$, since $\underline{\mathcal{A}}_{\mathrm{CL}}^\epsilon$ is the fraction of the test dataset for which f is l_∞^ϵ-*verified* at (x^+, x^-), i.e., $\textsc{Predict}(f, x^+, x^-, \epsilon) = \texttt{True}$.

In addition to prediction, we are also interested in the certified radius $\underline{\mathrm{R}}_{\mathrm{CL}}$ for a given (x^+, x^-). Apparently, $\underline{f}(x^+, x^-, \epsilon)$ is non-increasing with ϵ because of the `inf` operator. Thus, we can apply *binary search* to obtain $\underline{\mathrm{R}}_{\mathrm{CL}}$. The procedure is presented as CERTIFY. More precisely, we determine whether f is l_∞^ϵ-*verified* at (x^+, x^-) with current ϵ. If yes, it means that (x^+, x^-) is l_∞^ϵ-*verified* with an ϵ larger than the current one, then we increase ϵ; otherwise, we decrease ϵ. The final solution of ϵ is the certified radius $\underline{\mathrm{R}}_{\mathrm{CL}}$.

Pseudocode prediction and certification for CL

 # judge f is l_∞^ϵ-verified at (x^+, x^-) or not
function PREDICT(f, x^+, x^-, ϵ)
  Input: encoder f, positive sample x^+, negative sample x^-, perturbation bound ϵ
  Output: `True`: f is l_∞^ϵ-*verified* at (x^+, x^-); `False`: f is not l_∞^ϵ-*verified* at (x^+, x^-)
  $\underline{f} = \textsc{IncompleteVerifier}(f, x^+, x^-, \epsilon)$
  if $\underline{f} > 0$ **then return** `True`
  else return `False`

 # compute the certified radius of (x^+, x^-) on encoder f
function CERTIFY(f, x^+, x^-, τ, R_l, R_u)
  Input: encoder f, positive sample x^+, negative sample x^-, tolerance τ, lower bound R_l, upper bound R_u
  Output: certified radius $\underline{\mathrm{R}}_{\mathrm{CL}}$
  Initialization: $\tau = 10^{-6}$, $\mathrm{R}_l = 0$, $\mathrm{R}_u = 1$
  while $|\mathrm{R}_u - \mathrm{R}_l| > \tau$ **do**
    $\epsilon = {}^{(\mathrm{R}_l + \mathrm{R}_u)}/_2$
    # f is l_∞^ϵ-verified at (x^+, x^-), the answer should be larger than current ϵ
    if PREDICT(f, x^+, x^-, ϵ) **then** $\mathrm{R}_l = \epsilon$
    else $\mathrm{R}_u = \epsilon$
  end while
  return ϵ

D. Experimental Settings

D.1. Datasets and model architectures

Datasets For model training, we use MNIST (LeCun & Cortes, 2010) and CIFAR-10 (Krizhevsky & Hinton, 2009). MNIST is a dataset of 28×28 pixel grayscale images of handwritten single digits between 0 and 9, which contains 60,000 training images and 10,000 testing images with 10 classes. CIFAR-10 contains 50,000 training and 10,000 testing images with 10 classes. Size for color image in CIFAR-10 is 32×32.

Model architectures Table D.1 summarizes the CNN encoder architectures. Each layer (except the last linear layer) is followed by ReLU activation function. **Based** and **Deep** are used in Wang et al. (2021), **CNN-A** and **CNN-B** are used in Dathathri et al. (2020). To study the sensitivity of featrue dimension, the ouput dimension of encoder can be changed from 50 to 250 on **CNN-B**.

Table D.1. Model structures used in our experiments. For example, Conv(1, 8, 4) stands for a conventional layer with 1 input channel, 8 output channels and a 4×4 kernel. Linear(754, 100) stands for a fully connected layer with 754 input features and 100 output features.

Datasets	Model name	Encoder structure
MNIST	Base	Conv(1, 8, 4) - Conv(8, 16, 4) - Linear(784, 100)
	CNN-A	Conv(1, 16, 4) - Conv(16, 32, 4) - Linear(1568, 100)
CIFAR-10	Base	Conv(3, 8, 4) - Conv(8, 16, 4) - Linear(1024, 100)
	Deep	Conv(3, 8, 4) - Conv(8, 8, 3) - Conv(8, 8, 3) - Conv(8, 8, 4) - Linear(412, 100)
	CNN-A	Conv(3, 16, 4) - Conv(16, 32, 4) - Linear(2048, 100)
	CNN-B	Conv(3, 32, 5) - Conv(32, 128, 4) - Linear(8192, 100)

D.2. Training setup

All NNs are trained with verification-agnostic setting (Dathathri et al., 2020), which means without using any tricks to promote verifiability. We use the model mentioned in Appendix D.1 as the base encoder network and 2-layer multi-layer perceptron as the projection head (Chen et al., 2020). We set the step size of instance-wise attack $\alpha = 0.007$, the number of PGD maximize iteration as $K = 10$. For the rest, we follow the similar setup of SimCLR (Chen et al., 2020) and RoCL (Kim et al., 2020).

For optimization, we train the encoder with 500 epochs under Adam (Kingma & Ba, 2015) optimizer with the learning rate of 0.001. For the learning rate scheduling, the learning rate is dropped by a factor of 10 for every 100 epochs. The batch size in training is 256.

D.3. Evaluation setup

Linear evaluation We train the downstream linear layer on the top of the frozen encoder, and the training images are clean. We train the linear layer for 100 epochs with the learning rate of 0.001, and use the cross-entropy loss. The learning rate is dropped by a factor of 10 for every 50 epochs.

Robust test To evaluate the adversarial robustness, we use white-box project gradient descent (PGD) attack. We set ℓ_∞ attack with 20 iteration steps. $\epsilon_{test} = 0, 0.15, 0.2$ is set for MNIST, and $\epsilon_{test} = 0, {}^{8}/_{255}, {}^{16}/_{255}$ is set for CIFAR-10.

Incomplete verifiers If not special specified, CBC (β-CROWN (Wang et al., 2021) for CL) working as an incomplete verifier uses three minutes for each image.

D.4. Training efficiency

Our experiments are conducted on a Ubuntu 64-Bit Linux workstation, having 10-core Intel Xeon Silver CPU (2.20 GHz) and Nvidia GeForce RTX 2080 Ti GPUs with 11GB graphics memory. For adversarial contrastive learning on base encoder, it takes about 12 hours to train 500 epochs on MNIST **CNN-A** and CIFAR-10 **CNN-B** with a single GPU. And it takes about 150 minutes to train 100 epochs on downstream linear layer.

E. Additional Experiments

E.1. Certified instance accuracy

Table E.2. Complete comparison of certified instance accuracy across various networks and attack strength ϵ_{test}. All accuracy is computed on 100 test samples on MNIST and CIFAR-10. CBC uses 3 minutes for each sample.

Dataset	ϵ_{neg}	ϵ_{test}	Model	ϵ_{train}	Instance Accuracy	Certified Instance Accuracy	
					PGD	**CBC**	**CROWN**
MNIST	0.3	0.1	Based	0	20%	2%	0%
			CNN-A		6%	0%	0%
			Based	0.1	100%	100%	100%
			CNN-A		100%	100%	99%
	0.5	0.3	Based	0.3	100%	98%	42%
			CNN-A		100%	85%	3%
CIFAR-10	$\frac{16}{255}$	$\frac{2}{255}$	CNN-B	0	100%	97%	96%
				$\frac{2.2}{255}$	100%	100%	100%
		$\frac{4}{255}$		$\frac{4.4}{255}$	91%	26%	11%
				$\frac{8.8}{255}$	100%	55%	34%
			Based		100%	68%	52%
			Deep	$\frac{4.4}{255}$	100%	99%	95%
			CNN-A		100%	96%	84%
		$\frac{8}{255}$	CNN-B	$\frac{8.8}{255}$	99%	91%	81%
	$\frac{24}{255}$	$\frac{6}{255}$		$\frac{4.4}{255}$	1%	0%	0%
				$\frac{8.8}{255}$	100%	8%	2%
					100%	24%	11%

Appendix E.1 summarizes the complete results of *certified instance accuracy* provided by our proposed RVCL on MNIST and CIFAR-10. In order to control the similarity $\rho(f(x^-), f(x'))$ between x' and x^-, we generate the negative sample x^- via instance-wise PGD attack (Kim et al., 2020) with strength ϵ_{neg} which is much larger than ϵ_{test}. In § 6.3, we present some of the results on CIFAR-10 with $\epsilon_{neg} = {}^{16}/_{255}$.

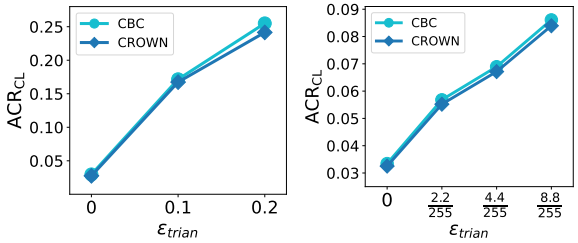
As shown by the results in Appendix E.1, the gap between CBC and PGD is smaller than that between CROWN and PGD on both MNIST and CIFAR-10 datasets. This is consistent with the experimental results in Wang et al. (2021), and further illustrates the effectiveness of our proposed RVCL. From Appendix E.1, we can also make the following observations and remarks:

Influence of ϵ_{test} If ϵ_{test} is small, the certified instance accuracy $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ of both CROWN and CBC approach the robust instance accuracy $\mathcal{A}^{\epsilon}_{\mathrm{CL}}$ given by instance-wise PGD. However, the gap between CROWN and CBC becomes large as ϵ_{test} increases. The results show that CBC is a more powerful verifer than CROWN. The reason for this is that CBC optimizes the intermediate layer bounds and then iteratively tightens the lower bound. We can further observe that instance-wise PGD successfully attacks the model on all images of CIFAR-10 under $\epsilon_{test} = {}^{8}/_{255}$.

Remark E.1. The results in Appendix E.1 often achieve a high robust instance accuracy $\mathcal{A}^{\epsilon}_{\mathrm{CL}}$. The direct reason is that the instance-wise attack is more difficult than the label-wise attack. Theoretically, Theorem 5.3 shows that the robust radius $\mathrm{R}_{\mathrm{CL}} \geq \mathrm{R}_{\mathrm{LE}}$. This implies that one may label-wise attack an image successfully with a small ϵ_{test}, but that it is nearly impossible to successfully instance-wise attack with the same small ϵ_{test}, which results in a high robust instance accuracy. Figure 4(b) experimentally certifies this conclusion, $\mathrm{ACR}_{\mathrm{CL}}$ is an order of magnitude larger than $\mathrm{ACR}_{\mathrm{LE}}$. However, without loss of effectiveness, we can still evaluate the tightness of verifiers by comparing the gap between $\mathcal{A}^{\epsilon}_{\mathrm{CL}}$ and $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$.

Influence of ϵ_{train} The certified instance accuracy $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ increases with increasing ϵ_{train} (consistent with Figure 4(c,d)), which demonstrates that $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ can also evaluate the model robustness. However, as we discuss in Remark 4.5, $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ is a function of specific attack strength ϵ_{test}, it's hard to compare the robustness of two models by comparing $\underline{\mathcal{A}}^{\epsilon}_{\mathrm{CL}}$ of various values of ϵ_{test}. Thus $\mathrm{ACR}_{\mathrm{CL}}$ is a more suitable choice to campare models with different robust performance.

E.2. Efficiency of incomplete verifiers



(a) $\mathrm{ACR}_{\mathrm{CL}}$ for MNIST (b) $\mathrm{ACR}_{\mathrm{CL}}$ for CIFAR-10

Figure E.1. $\mathrm{ACR}_{\mathrm{CL}}$ calculated by CBC and CROWN on MNIST and CIFAR-10. $|U_{\mathtt{test}}| = 100$, $K = 10$.

Table E.3. Average calculation time of $\mathrm{ACR}_{\mathrm{CL}}$, the number of negative samples $K = 5$, and timeout is set to 0.3.

Time(s)	CROWN	CBC
MNIST	1.16	464.52
CIFAR-10	3.66	921.78

§ 6.1 and § 6.2 use CROWN to compute the certified radius $\mathrm{ACR}_{\mathrm{CL}}$ in the interests of efficiency. This subsection shows that CBC achieves similar $\mathrm{ACR}_{\mathrm{CL}}$ with CROWN, but takes more time than CROWN.

Timeout is set to $0.3s$ for each step of CBC binary search. CNN-A is run on MNIST and CNN-B is run on CIFAR-10. The experimental setting of Figure E.1 is the same with that in § 6.1. Table E.3 provides the average time to caculate $\mathrm{ACR}_{\mathrm{CL}}$ defined in § 6.2 over 20 images.

The results in Figure E.1 show that $\mathrm{ACR}_{\mathrm{CL}}$ over $U_{\mathtt{test}}$ provided by CBC and CROWN is nearly the same, and both of them show the tendency of robust performance. This is because the time for each step of binary search is too short for CBC to tighten the lower bound. However, Table E.3 shows that the time cost of CBC is about **300 times** slower than CROWN, even using a small value of timeout for CBC. Therefore, we conclude that CBC is able to achieve a tight bound, but CBC is time-consuming in computing $\underline{\mathrm{R}}_{\mathrm{CL}}$, and it is reasonable to use CROWN to compute $\mathrm{ACR}_{\mathrm{CL}}$ of models and images.