TOWARDS GENERALIZED CERTIFIED ROBUSTNESS WITH MULTI-NORM TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

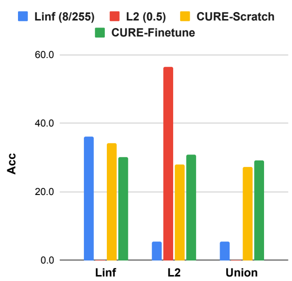
Existing certified training methods can only train models to be robust against a certain perturbation type (e.g. l_∞ or l_2). However, an l_∞ certifiably robust model may not be certifiably robust against l_2 perturbation (and vice versa) and also has low robustness against other perturbations (e.g. geometric and patch transformation). By constructing a theoretical framework to analyze and mitigate the tradeoff, we propose the first multi-norm certified training framework **CURE**, consisting of several multi-norm certified training methods, to attain better *union robustness* when training from scratch or fine-tuning a pre-trained certified model. Inspired by our theoretical findings, we devise bound alignment and connect natural training with certified training for better union robustness. Compared with SOTA-certified training, **CURE** improves union robustness to 32.0% on MNIST, 25.8% on CIFAR-10, and 10.6% on TinyImagenet across different epsilon values. It leads to better generalization on a diverse set of challenging unseen geometric and patch perturbations to 6.8% and 16.0% on CIFAR-10. Overall, our contributions pave a path towards *generalized certified robustness*.

1 INTRODUCTION

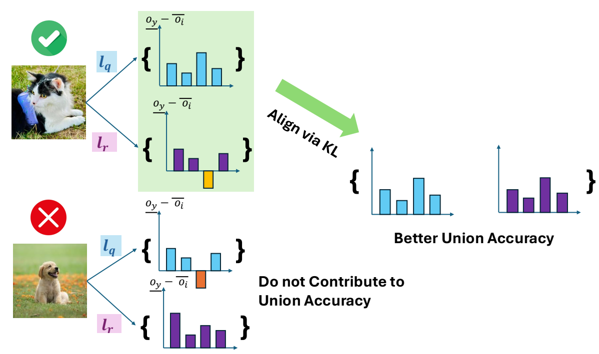
While deep neural networks (DNNs) are widely deployed in various vision applications, they remain vulnerable to adversarial attacks (Goodfellow et al., 2014; Kurakin et al., 2018). Many empirical defenses (Madry et al., 2017; Zhang et al., 2019a; Wang et al., 2023) against adversarial attacks have been proposed, however, they do not provide provable guarantees and remain vulnerable to stronger attacks. Hence, it is important to train DNNs to be *formally* robust against adversarial perturbations. Various deterministic certified training methods for specific perturbations (Mirman et al., 2018; Gowal et al., 2018; Zhang et al., 2019b; Balunović & Vechev, 2020; Shi et al., 2021; Müller et al., 2022; Yang et al., 2022; Hu et al., 2023; 2024; Mao et al., 2024)(e.g., l_∞, l_2, and geometric transformations) have been proposed. However, those defenses are mostly limited to a specific perturbation and cannot easily be generalized to other perturbation types (Yang et al., 2022; Chiang et al., 2020). Multi-norm attacks that examine models' robustness against l_p norms simultaneously have arisen in real-world settings such as cybersecurity Zhang et al. (2024), video recognition Lo & Patel (2021), and social media filtering Dai et al. (2024): it is essential for building models that are robust across diverse l_p norms, to generalize better against other non-l_p perturbations (Jiang & Singh, 2024).

In this work, we propose the first multi-norm **C**ertified training for **U**nion **R**obustn**E**ss (**CURE**) framework, consisting of several multi-norm certified training methods. Inspired by SABR (Müller et al., 2022), we use a deterministic l_2 defense that first finds the l_2 adversarial examples in a slightly truncated l_2 region and then propagates the smaller l_∞ box using the IBP loss (Gowal et al., 2018). In Figure 1a, we show that an l_∞ certified robust model may lack l_2 certified robustness and vice versa: l_∞ model only has 6.0% l_2 robustness and l_2 model has 0% l_∞ robustness, which reveals the **robustness tradeoff** among different l_p perturbations. Therefore, we first construct a theoretical framework for binary classification to analyze the tradeoff, from which we propose several methods based on multi-norm empirical defenses with different loss formulations (Tramer & Boneh, 2019; Madaan et al., 2021; Croce & Hein, 2022; Jiang & Singh, 2024). Our proposed methods successfully improve union and generalized certified robustness, shown in Table 1, Figure 4, and Table 2a.

However, the aforementioned methods achieve sub-optimal union robustness since they do not exploit the in-depth connections between certified training for different l_p perturbations as well as natural training. Thus, we propose the following improvements. **(1) Bound alignment:** Inspired by the upper bound of theoretical analysis (Theorem 3.2), we propose a new *bound alignment* method to mitigate the $l_q - l_r$ tradeoff better. We regularize the distributions of output bound differences, computed with IBP, for l_q, l_r perturbations on the correctly certified subset γ, as shown in Figure 1b. In this way, we encourage the model to *emphasize optimizing* the samples that can potentially become certifiably robust against multi-norm perturbations. To achieve this, we use a KL loss to encourage the distributions of the l_q, l_r output bound differences on subset γ to be close to each other for better union accuracy. **(2) Gradient Projection:** We find that there exist some useful components in natural training that can be extracted and leveraged to improve certified robustness (Jiang & Singh, 2024). To achieve this, we find and incorporate the layer-wise useful natural training components by comparing the similarity of the certified and natural training model updates. **(3) Quick fine-tuning:** Fine-tuning an l_p-robust model using bound alignment quickly achieves superior multi-norm certified robustness. By addressing the $l_q - l_r$ tradeoff, bound alignment preserves more l_q robustness when fine-tuning with l_r perturbations, focusing on correctly certified samples. This technique enables efficient multi-norm robustness using pre-trained models with single l_p robustness. Figure 1a shows that both scratch training (CURE-Scratch) and fine-tuning (CURE-Finetune) significantly enhance union robustness over single-norm training. **(4) Generalized robustness:** As a perhaps surprising side effect, improving union-certified robustness leads to stronger *generalized certified robustness* by generalizing better to other geometric and patch transformations (Section 4.1), confirming that l_p robustness is the bedrock for non-l_p robustness (that non-ℓ_p perturbations may be modeled through ℓ_p-bounded formulations) (Mangal et al., 2023).



(a) $l_\infty - l_2$ tradeoff on CIFAR10 with $\epsilon_\infty = \frac{8}{255}, \epsilon_2 = 0.5$. The X-axis represents l_∞, l_2, and union robustness; different colors refer to different training methods.



(b) Bound alignment during training.

Figure 1: (a) $l_\infty - l_2$ tradeoff: an l_∞ certified robust model may lack l_2 certified robustness and vice versa. **CURE-Scratch** (yellow) and **CURE-Finetune** (green) improve union robustness significantly. (b) We align the output bound differences for l_q, l_r perturbations on the correctly certified l_q subset γ to mitigate $l_q - l_r$ tradeoff for better union robustness.

Main Contributions:

- We design a theoretical framework to analyze the multi-norm certified robustness tradeoff. Based on this, we propose three training methods, CURE-Joint, CURE-Max, and CURE-Random with different loss formulations for better union and generalized certified robustness.
- Inspired by our theoretical findings, we introduce techniques including bound alignment, connecting natural training with certified training, and certified fine-tuning for better union robustness. CURE-Scratch and CURE-Finetune further facilitate our multi-norm certified training procedure and advance multi-norm robustness.
- Compared with a SOTA certified training method (Müller et al., 2022), **CURE** improves union robustness up to 32.0% on MNIST, 25.8% on CIFAR-10, and 10.6% on TinyImagenet. It improves robustness against unseen geometric and patch perturbations up to $0.6\%, 8.5\%$ on MNIST and $6.8\%, 16\%$ on CIFAR-10.

2 BACKGROUND

In this section, we provide the necessary background of neural network verification and certified training, with related work discussed in Appendix B. Given samples $\{(x_i, y_i)\}_{i=0}^N$ from a data distribution $\mathcal{D}$, the input comprises images $x \in \mathbb{R}^d$ with labels $y \in \mathbb{R}^k$. The goal is to train a classifier f, parameterized by θ, to minimize a loss function $\mathcal{L} : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ over $\mathcal{D}$.

2.1 Neural network verification

Neural network verification formally proves a network's robustness, with the provably robust samples defining the *certified accuracy*. Interval Bound Propagation (IBP) (Gowal et al., 2018; Mirman et al., 2018) is a simple yet effective method for verification. It over-approximates the input region $B_p(x, \epsilon_p), p \in \{2, \infty\}$, propagates it layer by layer through the network $f = L_j \circ \sigma \circ L_{j-2} \circ \ldots \circ L_1$ (with linear layers L_i and ReLU activations σ), and verifies whether the reachable outputs classify correctly. Robustness is certified if the lower bound of the correct class exceeds the upper bounds of all others ($\forall i \neq y, \overline{o}_i - \underline{o}_y < 0$) (for more details, see Gowal et al. (2018)).

2.2 Training for robustness

A classifier is adversarially robust on an l_p-norm ball $B_p(x, \epsilon_p) = \{x' \in \mathbb{R}^d : \|x' - x\|_p \leq \epsilon_p\}$ if it correctly classifies all points within this region, i.e., $\arg\max f(x') = y$ for all $x' \in B_p(x, \epsilon_p)$. Training for robustness is framed as a min-max optimization problem, defined for an l_p attack as:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{x' \in B_p(x,\epsilon_p)} \mathcal{L}(f(x'), y) \right] \tag{1}$$

The inner maximization problem is often approximated through adversarial training (Madry et al., 2017) or certified training (Gowal et al., 2018; Müller et al., 2022). However, such methods are typically tailored to specific p values, leaving networks vulnerable to other perturbations. To address this, prior work has only trained networks to be *adversarially* robust against multiple perturbations (l_1, l_2, l_∞). Our focus is on training networks to be *certifiably* robust to multiple l_p perturbations.

2.3 Certified training

There are two main categories of methods to train certifiably robust models: unsound and sound methods. Sound methods optimize a rigorously defined upper bound of the inner maximization problem, ensuring provable robustness guarantees. In contrast, unsound methods give up this guarantee to have a more precise approximation. IBP, a sound method, optimizes the following loss function based on logit differences:

$$\mathcal{L}_{\text{IBP}}(x, y, \epsilon_\infty) = \ln(1 + \sum_{i \neq y} e^{\overline{o}_i - \underline{o}_y}) \tag{2}$$

Also, state-of-the-art certified training methods SABR (Müller et al., 2022), TAPs (Mao et al., 2024), and CC/EXP/MTL-IBP (Palma et al., 2024) relax the robustness guarantee within the specification loss, but in practice, result in better standard and certified accuracy. Given a small box size τ_∞, SABR finds an adversarial example $x' \in B_\infty(x, \epsilon_\infty - \tau_\infty)$ and propagates a small box region $B_\infty(x', \tau_\infty)$ across all layers using IBP loss, expressed as:

$$\mathcal{L}_{l_\infty}(x, y, \epsilon_\infty, \tau_\infty) = \max_{x' \in B_\infty(x, \epsilon_\infty - \tau_\infty)} \mathcal{L}_{\text{IBP}}(x', y, \tau_\infty) \tag{3}$$

2.4 Evaluation metrics

Union certified accuracy (UCA). We focus on the union threat model $\Delta = B_1(x, \epsilon_1) \cup B_2(x, \epsilon_2) \cup B_\infty(x, \epsilon_\infty)$ which requires the DNN to be *certifiably* robust within the l_1, l_2 and l_∞ adversarial regions simultaneously. Union accuracy is then defined as the robustness against $\Delta_{(i)}$ for each x_i sampled from $\mathcal{D}$. In this paper, similar to the prior works (Croce & Hein, 2022), we use union accuracy as the main metric to evaluate the multi-norm *certified* robustness.

$$\textbf{UCA} = \mathbb{E}_{x_i \sim \mathcal{D}} \left[\mathbf{1}\{\forall x' \in \Delta \text{ with bounds } \overline{o}_i, \underline{o}_i, \forall i \neq y_i, \overline{o}_i - \underline{o}_{y_i} < 0\} \right],$$

where y_i is the true label for sample x_i, and $\mathbf{1}\{\cdot\}$ is the indicator function.

Generalized certified robustness (GCR). We measure the generalization ability of multi-norm certified training to other perturbation types, including rotation, translation, scaling, shearing, contrast, and brightness change of geometric transformations (Balunovic et al., 2019; Yang et al., 2022), as well as patch attacks (Chiang et al., 2020). If we have perturbation sets $T_j(x)$ representing each transformation or attack type j, we define:

$$\textbf{GCR} = \mathbb{E}_{x_i \sim \mathcal{D}} \left[\frac{1}{J} \sum j = 1^J \mathbf{1}\{\forall x' \in T_j(x_i) \text{ with bounds } \overline{o}_i, \underline{o}_i, \forall i \neq y_i, \overline{o}_i - \underline{o}_{y_i} < 0\} \right],$$

(a) $\mathcal{S}_q$ and $\mathcal{S}_r$ are disjoint and cover X

(b) $\mathcal{S}_r$ includes $\mathcal{S}_q$

Figure 2: $l_q - l_r$ trade-off visualization. Blue and purple points belong to $\mathcal{S}_q \subseteq X$ and $\mathcal{S}_r \subseteq X$.

Figure 3: Comparisons of union errors of two extreme cases. Note that $\mathcal{R}_r \leq \mathcal{R}_{\text{union}} \leq 1$. A larger union error has a more severe $l_q - l_r$ trade-off.

	$\mathcal{S}_q \cap \mathcal{S}_r = \emptyset \wedge \mathcal{S}_q \cup \mathcal{S}_r = X$	$\mathcal{S}_q \subseteq \mathcal{S}_r$
$\mathcal{R}_{\text{align}}$	1 - $\mathcal{R}_r$	0
$\mathcal{R}_{\text{union}}$	1	$\mathcal{R}_r$ (optimal)

where J is the total number of considered perturbation types.

3 CURE: MULTI-NORM CERTIFIED TRAINING FOR UNION ROBUSTNESS

This section presents our multi-norm certified training (CT) framework **CURE**. We introduce our framework with binary classification to analyze the tradeoff between certified l_p, l_q perturbations. However, we note our algorithms presented in this work are all multi-class, and the binary classification framework can be easily extended to the multi-class case Zhang et al. (2019a). Based on the theoretical analysis (Eq. 4), we propose three methods for multi-norm CT against l_2, l_∞ perturbations using different loss formulations, which serve as the base instantiations of our framework. Then, we design new techniques to improve union-certified accuracy inspired by our theoretical findings.

Notations. For binary classification, we denote the sample instance as $x \in \mathcal{X}$, with the label $y \in \{-1, +1\}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the instance space. The dataset is denoted as $D = \{(x_i, y_i)\}_{i=1}^n$, where $X = \{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ is the set of instances and $Y = \{y_1, \ldots, y_n\} \subseteq \{-1, +1\}$ is the set of corresponding labels. Let $f : \mathcal{X} \to \mathbb{R}$ map instances to output values $\in \{-1, +1\}$, which can be parameterized (e.g., by deep neural networks). We use $\mathbf{1}\{\text{event}\}$, the 0-1 loss, as an indicator function that is 1 if an event happens and 0 otherwise. For any function $\psi(\boldsymbol{u})$, we use ψ^{-1} to denote the inverse function. $\phi(\cdot)$ is used to denote the surrogate for the 0-1 loss function.

Robust, alignment and union error. To characterize the robustness of a score function $f : \mathcal{X} \to \mathbb{R}$, similar to Schmidt et al. (2018); Cullina et al. (2018); Bubeck et al. (2019), we define *robust error* under the threat model of ϵ_q perturbation: $\mathcal{R}_q(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbf{1}\{\exists x'_q{\in}B_q(x, \epsilon_q) \text{ s.t. } f(x'_q)y \leq 0\}$. We define $\mathcal{R}_r(f)$ similarly to $\mathcal{R}_q(f)$ for ϵ_r perturbation, and without loss of generality, assume $\mathcal{R}_r(f) \geq \mathcal{R}_q(f)$. Then, we introduce *alignment error* as the risk calculated by $x \in X$ that are robust against l_r attack but not robust against l_q attack: $\mathcal{R}_{\text{align}}(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbf{1}\{\exists x'_r \in B_r(x, \epsilon_r), x'_q \in B_q(x, \epsilon_q), \text{ s.t. } f(x'_r)y > 0 \text{ and } f(x'_q)y \leq 0\}$. The *union error* is the risk calculated by $x \in X$ that are either not robust against l_q or l_r attack. We have the following relationship of $\mathcal{R}_{\text{union}}(f)$:

$$\mathcal{R}_{\text{union}}(f) = \mathcal{R}_r(f) + \mathcal{R}_{\text{align}}(f). \tag{4}$$

Trade-off between l_q, l_r perturbations. Our study is motivated by the trade-off between l_q and l_r robust errors, as shown empirically in Figure 1a. To illustrate, we provide two extreme cases in Figure 2. We define $S_r = \{x | \exists x'_r \in B_r(x, \epsilon_r) \text{ s.t. } f(x'_r)y \leq 0, (x, y) \in D\}$ (define S_q similarly). As shown in Table 3, we have (a) the *lowest* union accuracy of 0: when all instances in X can be successfully attacked by l_q or l_r norms yet no single instance can be attacked in both l_q and l_r, we have a union error of 1; (b) the *highest* union accuracy of $1 - \mathcal{R}_r$: when the instances that are not robust against l_r attack includes all instances that are not robust against l_q attack, we have a union error $\mathcal{R}_r$. A larger union error indicates a bigger l_q, l_r trade-off. R_{union} is lower bounded by $\mathcal{R}_r$.

3.1 CERTIFIED TRAINING FOR MULTIPLE NORMS

Eq. 4 reveals that we need to minimize $\mathcal{R}_{\text{align}}$ by not only training with one kind of adversarial examples x'_r since it will lead to a large $\mathcal{R}_{\text{align}}$ with more instances not robust against l_q attack. To effectively combine the optimization of l_q and l_r ($q = 2, r = \infty$) certified training, based on the work (Tramer & Boneh, 2019; Madaan et al., 2021; Croce & Hein, 2022) on adversarial training for multiple norms, we propose the following methods:

1. **CURE-Joint**: optimizes $\mathcal{L}_{l_\infty}$ and $\mathcal{L}_{l_2}$ together: it takes the sum of two worst-case IBP losses with l_∞ and l_2 examples using a convex combination of weights with hyperparameter $\alpha \in [0, 1]$.

$$\mathcal{L}_{Joint} = (1-\alpha)\cdot\mathcal{L}_{l_\infty}(x, y, \epsilon_\infty, \tau_\infty) + \alpha\cdot\mathcal{L}_{l_2}(x, y, \epsilon_2, \tau_2)$$

2. **CURE-Max**: compares $\mathcal{L}_{l_2}$ and $\mathcal{L}_{l_\infty}$, selecting the higher IBP loss as the worse-case outcome. This approach acts as a *worst-case* defense, accounting for adversarial examples with the highest IBP loss across multiple perturbation types. The max loss $\mathcal{L}_{Max}$ is defined as:

$$\mathcal{L}_{Max} = \max_{p\in\{2,\infty\}} \max_{x'\in B_p(x,\epsilon_p-\tau_p)} \mathcal{L}_{\text{IBP}}(x, y, \epsilon_p, \tau_p)$$

3. **CURE-Random**: randomly partitions a batch of data $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$ into equal sized blocks $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{x}_2, \mathbf{y}_2)$. For $(\mathbf{x}_1, \mathbf{y}_1)$, we calculate the l_∞ worst-case IBP loss $\mathcal{L}_{l_\infty}$ with l_∞ perturbations. For the other half $(\mathbf{x}_2, \mathbf{y}_2)$, similarly, we get the l_2 worst-case IBP loss by applying l_2 perturbations. After that, we optimize the **Joint** loss of these two with equal weights, as shown below. In this way, we reduce the time cost of propagating the bounds and generating the adversarial examples by $\frac{1}{2}$.

$$\mathcal{L}_{Random} = \mathcal{L}_{l_\infty}(\mathbf{x}_1, \mathbf{y}_2, \epsilon_\infty, \tau_\infty) + \mathcal{L}_{l_2}(\mathbf{x}_2, \mathbf{y}_2, \epsilon_2, \tau_2), \text{where } \mathbf{x} = \mathbf{x}_1 \cup \mathbf{x}_2, \mathbf{y} = \mathbf{y}_1 \cup \mathbf{y}_2$$

3.2 Unified and effective multi-norm certified training

The methods proposed above are still suboptimal as they fail to fully explore the relationship between worst-case IBP losses across different perturbations, certified training (CT), and natural training (NT). To address this, we introduce the following techniques to enhance the union robustness of **CURE**: (1) We derive an upper bound on the terms, which informs us to propose a bound alignment technique to mitigate the trade-off better, improving multi-norm robustness. (2) We analyze and connect certified and natural training to attain better union accuracy. (3) the first certified fine-tuning method to quickly improve union accuracy with pre-trained single-norm models (Table 1).

Bound alignment (BA). First, we aim to design *tight* upper bounds for different risk terms, leveraging the theory of classification-calibrated loss, which informs how to design methods to mitigate the $l_r - l_q$ tradeoff more efficiently. First, classification-calibrated surrogate loss is a surrogate loss $\mathcal{R}_\phi(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\phi(f(x)y)$ designed to approximate the 0-1 loss, making it computationally efficient for optimization while maintaining a meaningful relationship with the true error (Zhang et al., 2019a). A loss is classification-calibrated if it ensures that any decision rule inconsistent with the Bayes optimal classifier has a strictly larger ϕ-risk of the loss function ϕ. This property is crucial for achieving optimal classification performance, and examples include hinge loss, logistic loss, and exponential loss. Here, we show the binary IBP loss falls into this loss category.

Lemma 3.1. *Binary IBP loss is a logistic loss in the classification-calibrated surrogate loss family.*

Proof. We have binary $\mathcal{L}_{\text{IBP}}(x, y, \epsilon_p) = \ln(1 + e^{\overline{o}_i - \underline{o}_y}), i \neq y$, which is a logistic loss.

Upper bound. Our following analysis provides a performance guarantee for minimizing the surrogate loss. We introduce a transformation ψ of classification-calibrated losses. $\psi : [0, 1] \to [0, \infty)$ is defined as the convex conjugate of a function that lower bounds the gap between a modified entropy function (e.g., a surrogate loss like cross-entropy) and the standard Shannon entropy (Zhang et al., 2019a). This gap quantifies how well the surrogate loss approximates the true 0-1 classification error. The function ψ is used to bound the difference between the union risk $\mathcal{R}_{\text{union}}$ and the optimal risk under individual ℓ_r perturbations $\mathcal{R}_r^* := \min_f \mathcal{R}_r(f)$. It has desirable properties: ψ is non-decreasing, convex, continuous on $[0, 1]$, and satisfies $\psi(0) = 0$. By Eq.4, we have $\mathcal{R}_{\text{union}}(f) - \mathcal{R}_r^* = \mathcal{R}_r(f) - \mathcal{R}_r^* + \mathcal{R}_{\text{align}}(f) \leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*) + \mathcal{R}_{\text{align}}(f)$, where the inequality holds because ϕ is constructed from a classification-calibrated loss (Bartlett et al., 2006).

Theorem 3.2. *Let $\mathcal{R}_\phi(f) := \mathbb{E}\phi(f(x)y)$ and $\mathcal{R}_\phi^* := \min_f \mathcal{R}_\phi(f)$. Under Assumption 1 in Zhang et al. (2019a), with $\mathbb{E}$ taken over the data distribution, for any non-negative loss function ϕ such that $\phi(0) \geq 1$, any measurable $f : \mathcal{X} \to \mathbb{R}$, any probability distribution on $\mathcal{X} \times \{\pm 1\}$, IBP output bound differences from f as $d(x) = \overline{o}_i - \underline{o}_y (i \neq y)$, and any $\lambda > 0$, we have*

$$\mathcal{R}_{union}(f) - \mathcal{R}_r^* \leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*) + \mathbb{E} \max_{\substack{x'_r\in\\ B_r(x,\epsilon_r)}} \max_{\substack{x'_q\in\\ B_q(x,\epsilon_q)}} (\phi(d(x'_r)d(x'_q)/\lambda), \overline{o}_i \leq \underline{o}_y \text{ for } d(x'_r)).$$

The proof is in Appendix A.1, which sheds light on how we can further improve union-certified robustness. Algorithmically, we can extend the framework to the case of multi-class classifications by replacing ϕ with a multi-class calibrated loss $L(\cdot, \cdot)$ (Zhang et al., 2019a), such as cross-entropy, which ensures that minimizers of the surrogate risk align with those of the 0-1 loss. $\phi(d(x'_r)d(x'_p)/\lambda)$ indicates that we need to align the distributions between output bound differences of two perturbations, so Theorem 3.2 has a tighter upper bound. $\forall i \neq y, \overline{o}_i \leq \underline{o}_y$ means we need to regularize those bounds only on the *correctly predicted l_r subsets* (Definition 3.3), meaning the subset γ for which the lower bound computed with IBP of the correct class is higher than the upper bounds of other classes.

Definition 3.3 (Correctly Certified l_r Subset)**.** At epoch e, given the perturbation size $\epsilon_r \in \mathbb{R}$ and model f, for a batch of data $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$ of size n, we have the output upper and lower bounds computed by IBP for l_r perturbations. We define a function h for this procedure as $h(\mathbf{x}) = \{\overline{\mathbf{o}}_j, \underline{\mathbf{o}}_j\}_{j=0}^{j<n}$, where $\mathbf{o} = \{o_i\}_{i=0}^{i<k}$ is a vector of bounds for all classes. Then, the correctly certified subset γ at the current step is defined as:

$$\forall j \in \gamma \text{ with } (\mathbf{x}_j, \mathbf{y}_j) \text{ and bounds } \{\overline{\mathbf{o}}_j = \{\overline{o}_i\}_{i=0}^{i<k}, \underline{\mathbf{o}}_j = \{\underline{o}_i\}_{i=0}^{i<k}\}, \text{ we have } \forall i \neq y_j, \overline{o}_i \leq \underline{o}_{y_j}.$$

For certified training, Gowal et al. (2018); Müller et al. (2022) optimize the model using bound differences $\{\overline{o}_i - \underline{o}_y\}_{i=0}^{i<k}$ (y is the correct class). Inspired by Theorem 3.2, we align the bound differences $\{\{\overline{o}_i - \underline{o}_y\}_{i=0}^{i<k}\}_n$ of l_r and l_q CT outputs with a batch of n samples, specifically on the correctly certified l_q subset γ. Specifically, for each batch of data $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$, we denote the bounds differences after softmax normalization for two perturbations as d_q and d_r. Then, we select indices γ, according to Definition 3.3. We denote the size of the indices as $n_c \leq n$. We compute a KL-divergence loss over this set of samples using $KL(d_q[\gamma] \| d_r[\gamma])$ (Eq. 5). Intuitively, we aim to encourage $d_r[\gamma]$ and $d_q[\gamma]$ distributions to become close to each other, such that we gain more union robustness.

$$\mathcal{L}_{KL} = \frac{1}{n_c} \cdot \sum_{i=1}^{n_c} \sum_{j=0}^{k} d_q[\gamma[i]][j] \cdot \log\left(\frac{d_q[\gamma[i]][j]}{d_r[\gamma[i]][j]}\right) \tag{5}$$

Apart from the KL loss, we add another loss term using a Max-style approach in Eq. 6, since Max performs relatively well, as shown in Table 1. We also consider combining with Random/Joint losses if they lead to a better performance. Our final loss $\mathcal{L}_{\text{Scratch}}$ combines $\mathcal{L}_{KL}$ and $\mathcal{L}_{Max}$, via a hyper-parameter η, as shown in Eq. 7.

$$\mathcal{L}_{Max} = \max_{p \in \{2,\infty\}} \max_{x' \in B_p(x, \epsilon_p - \tau_p)} \mathcal{L}_{\text{IBP}}(x, y, \epsilon_p, \tau_p) \tag{6}$$

$$\mathcal{L}_{\text{Scratch}} = \mathcal{L}_{Max} + \eta \cdot \mathcal{L}_{KL} \tag{7}$$

Integrate NT into CT. In the context of adversarial robustness, Jiang & Singh (2024) shows that there exist a useful portion of model updates in natural training, which can be extracted and integrated into adversarial training to improve adversarial robustness. Based on this, we propose a technique to integrate NT into CT, to enhance union-certified robustness. Specifically, for model $p^{(r)}$ at any epoch r, we examine the model updates of NT and CT over all samples from $\mathcal{D}$. The models $p_n^{(r)}$ and $p_c^{(r)}$ represent the results after one epoch of NT and CT, from the same initial model $p^{(r)}$. Then we compare the updates of the two $g_n = p_n^{(r)} - p^{(r)}$ and $g_c = p_c^{(r)} - p^{(r)}$. For a specific layer l, by comparing g_n^l and g_c^l, we retain a portion of g_n^l according to their cosine similarity score (Eq.8). Negative scores indicate that g_n^l does not contribute to certified robustness, so we discard components with similarity scores ≤ 0. The **GP** (Gradient Projection) operation, defined in Eq.9, projects g_c^l towards g_n^l.

$$\cos(g_n^l, g_c^l) = \frac{g_n^l \cdot g_c^l}{\|g_n^l\| \|g_c^l\|} \tag{8}$$

$$\mathbf{GP}(g_n^l, g_c^l) = \begin{cases} \cos(g_n^l, g_c^l) \cdot g_n^l, & \cos(g_n^l, g_c^l) > 0 \\ 0, & \cos(g_n^l, g_c^l) \leq 0 \end{cases} \tag{9}$$

Therefore, the total projected (useful) model updates g_p coming from g_n could be computed as Eq. 10. We use $\mathcal{M}$ to represent all layers of the current model update. The expression $\bigcup_{l \in \mathcal{M}}$ concatenates the useful natural model update components from all layers. A hyper-parameter β is introduced to balance the contributions of g_{GP} and g_c, as outlined in Eq.11. It is important to note that this projection procedure is applied only after the eps-annealing phase of certified training. The theoretical analysis of why connecting NT with CT works is discussed in Appendix A.2.

$$g_p = \bigcup_{l \in \mathcal{M}} \mathbf{GP}(g_n^l, g_c^l) \tag{10}$$

$$p^{(r+1)} = p^{(r)} + \beta \cdot g_p + (1 - \beta) \cdot g_c \tag{11}$$

Quick certified fine-tuning. In adversarial robustness, Croce & Hein (2022) shows that public models can be made more robust with only the application of fine-tuning, which reduces the computational cost significantly compared with training from scratch. In this work, we propose the first fine-tuning certified multi-norm robustness scheme **CURE-Finetune**. Starting from a single norm pre-trained model, we perform the bound alignment technique by optimizing $\mathcal{L}_{\text{Scratch}}$ for a few epochs. Because of the $l_q - l_r$ tradeoff, certifiably finetuning a l_q pre-trained model on l_r perturbations reduces l_q robustness. Thus, we want to preserve more l_q robustness when doing certified fine-tuning, which makes bound alignment useful here. By regularizing on the correctly certified l_q subset with $\mathcal{L}_{\text{Scratch}}$, we can prevent losing more l_q robustness when boosting l_r robustness, which leads to better union accuracy. We note that **CURE-Finetune** can be adapted to any single-norm certifiably pre-trained models. As shown in Table 1, we can obtain a superior multi-norm certified robustness by performing quick fine-tuning on pre-trained l_∞ models.

4 EXPERIMENT

In this section, we present and discuss the results of union, geometric, and patch robustness, as well as ablation studies on hyper-parameters for MNIST, CIFAR-10, and TinyImagenet experiments. Other ablation studies, visualizations, and algorithms of **CURE** can be found in Appendix D and F.

Experimental Setup. For datasets, we use MNIST (LeCun et al., 2010) and CIFAR-10 (Krizhevsky et al., 2009) which both include 60K images with 50K and 10K images for training and testing, as well as TinyImageNet (Le & Yang, 2015) which consists of 200 object classes with 500 training images, 50 validation images, and 50 test images per class. We compare the following methods: 1. l_∞: l_∞ certified defense SABR (Müller et al., 2022), 2. l_2: l_2 certified defense based on SABR, 3. **CURE-Joint**: take a weighted sum of l_2, l_∞ IBP losses. 4. **CURE-Max**: take the worst of l_2, l_∞ IBP losses. 5. **CURE-Random**: randomly partitions the samples into two blocks, then applies the Joint loss with equal weights. 6. **CURE-Scratch**: training from scratch with bound alignment and gradient projection techniques. 7. **CURE-Finetune**: robust fine-tuning with the bound alignment technique using l_∞ pre-trained models. We use a 7-layer convolutional architecture CNN7, a standard architecture (Müller et al., 2022) for certified training. In Table 12, we compare our proposed l_2 defense with Hu et al. (2023), where we show our method outperforms the SOTA l_2 deterministic certified defense on CIFAR-10. We choose similar hyperparameters and training setup as Müller et al. (2022) for l_∞ certified training. We select $\alpha = 0.5$, l_2 subselection ratio $\lambda_2 = 1e^{-5}$, $\beta = 0.5$, and $\eta = 2.0$ according to our ablation study results in Section 4.2 and Appendix D. For certified fine-tuning, we finetune 20% of the epochs of CURE-Scratch and are only performed on l_∞ models as they generally have higher robust errors. Full implementation details are in Appendix C.

4.1 MAIN RESULTS

Evaluation. We choose the common $\epsilon_\infty, \epsilon_2, \epsilon_1$ values used in the literature (Müller et al., 2022; Hu et al., 2023) to construct multi-norm regions. These include $(\epsilon_1 = 1.0, \epsilon_2 = 0.5, \epsilon_\infty = 0.1)$, $(\epsilon_1 = 2.0, \epsilon_2 = 1.0, \epsilon_\infty = 0.3)$ for MNIST, $(\epsilon_1 = 0.5, \epsilon_2 = 0.25, \epsilon_\infty = \frac{2}{255})$, $(\epsilon_1 = 1.0, \epsilon_2 = 0.5, \epsilon_\infty = \frac{8}{255})$ for CIFAR-10 and $(\epsilon_1 = \frac{72}{255}, \epsilon_2 = \frac{36}{255}, \epsilon_\infty = \frac{1}{255})$ for TinyImageNet. We make sure the adversarial regions with sizes ϵ_∞, ϵ_1 and ϵ_2 do not include each other. We report the clean accuracy, certified accuracy against l_1, l_2, l_∞ perturbations, union accuracy, and individual/average certified robustness against geometric transformations as well as patch attacks. Further, we use alpha-beta crown (Zhang et al., 2018) for certification on l_2, l_∞ perturbations, FGV (Yang et al., 2022) for efficient certification of geometric transformations, and Chiang et al. (2020) for 2×2 patch attacks. Additional experiment results on CIFAR-100, varying epsilons for l_p norms where we show our methods generalize to a wide choice of epsilons and ablation studies can be found in Appendix D.

Union accuracy on MNIST, CIFAR-10, and TinyImagenet with CURE framework. In Table 1, we show the results of clean accuracy and certified robustness against single and multi-norm with **CURE** on MNIST, CIFAR-10, and TinyImagenet. CURE-Joint, CURE-Max, and CURE-Random usually yield better union robustness than l_2 and l_∞ certified training. Further, **CURE-Scratch** and **CURE-Finetune** consistently improve the union accuracy compared with other multi-norm methods with significant margins in most cases (20% for MNIST and 8% for CIFAR-10 experiments), showing the effectiveness of bound alignment and gradient projection techniques. Also, for quick fine-tuning, we show it is possible to quickly fine-tune a l_∞ robust model with good union robustness using bound alignment, achieving SOTA union accuracy on MNIST and CIFAR-10 experiments. More results on MNIST, CIFAR-10, and CIFAR-100 are available in Appendix D.

Robustness against unseen geometric and patch transformations. Table 4 and Table 6 (in Appendix) compare **CURE** with single norm training against various geometric perturbations on MNIST and CIFAR-10 datasets. **CURE** outperforms single norm training on diverse geometric transformations (e.g., 6% for CIFAR-10 on average), leading to better *generalized certified robustness*. Also, **CURE-Scratch** has better geometric robustness than **CURE-Max** on both datasets, which reveals that bound alignment and gradient projection lead to better generalized certified robustness.

Dataset	$(\epsilon_\infty, \epsilon_2, \epsilon_1)$	Methods	Clean	l_∞	l_2	l_1	Union
MNIST	(0.3, 1.0, 2.0)	l_∞	98.7	92.1	69.6	38.9	38.5
		l_2	99.4	0.0	94.5	94.7	0.0
		CURE-Joint	98.7	90.5	76.3	50.8	50.3
		CURE-Max	98.7	91.1	76.2	47.2	46.5
		CURE-Random	98.7	90.5	76.3	50.8	50.3
		CURE-Finetune	98.5	90.1	83.5	64.0	63.2
		CURE-Scratch	98.0	89.4	85.9	71.5	**70.5**
CIFAR-10	$(\frac{8}{255}, 0.5, 1.0)$	l_∞	51.8	36.3	6.0	3.8	3.5
		l_2	78.6	0.0	56.5	75.8	0.0
		CURE-Joint	51.3	23.9	34.0	38.6	21.4
		CURE-Max	51.5	33.9	19.5	21.6	16.8
		CURE-Random	53.0	28.9	28.0	34.6	24.0
		CURE-Finetune	40.2	30.2	30.8	34.8	**29.3**
		CURE-Scratch	49.5	34.2	28.1	32.0	26.3
TinyImagnet	$(\frac{1}{255}, \frac{36}{255}, \frac{72}{255})$	l_∞	28.3	19.4	19.4	12.9	12.9
		l_2	36.2	2.9	30.6	23.5	2.9
		CURE-Joint	30.2	20.0	25.9	18.8	18.8
		CURE-Max	29.6	21.8	23.5	18.2	18.2
		CURE-Random	30.5	25.9	28.2	23.5	**23.5**
		CURE-Fintune	28.1	21.2	21.8	18.2	16.6
		CURE-Scratch	29.7	23.5	26.5	22.4	22.4

Table 1: Comparison of the clean accuracy, individual, and union certified accuracy (%). **CURE** consistently improves union accuracy compared with single-norm training with significant margins on all datasets. **CURE-Scratch** and **CURE-Finetune** outperform other methods in most cases.

Configs	R(10)	R(2),Sh(2)	Sc(1),R(1), C(1),B(0.001)	Avg
l_∞	27.8	33.2	23.3	28.1
l_2	36.6	0.0	0.0	12.2
CURE-Joint	**35.0**	**41.4**	**28.2**	**34.9**
CURE-Max	33.7	39.0	23.3	32.0
CURE-Random	35.1	40.9	26.2	34.1
CURE-Scratch	34.2	39.6	24.9	32.9

Figure 4: Comparison on **CURE** against geometric transformations for CIFAR-10 ($\epsilon_1 = 1.0, \epsilon_2 = 0.5, \epsilon_\infty = \frac{8}{255}$) experiment. We denote $\mathrm{R}(\varphi)$ a rotation of $\pm\varphi$ degrees; $\mathrm{T_u}(\Delta u)$ and $\mathrm{T_v}(\Delta v)$ a translation of $\pm\Delta u$ pixels horizontally and $\pm\Delta v$ pixels vertically, respectively; $\mathrm{Sc}(\lambda)$ a scaling of $\pm\lambda\%$; $\mathrm{Sh}(\gamma)$ a shearing of $\pm\gamma\%$; $\mathrm{C}(\alpha)$ a contrast change of $\pm\alpha\%$; and $\mathrm{B}(\beta)$ a brightness change of $\pm\beta$. **CURE** improves the geometric certified robustness compared with single norm training. **CURE-Scratch** achieves the best average geometric transformation robustness.

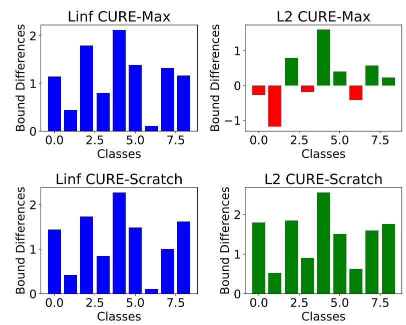
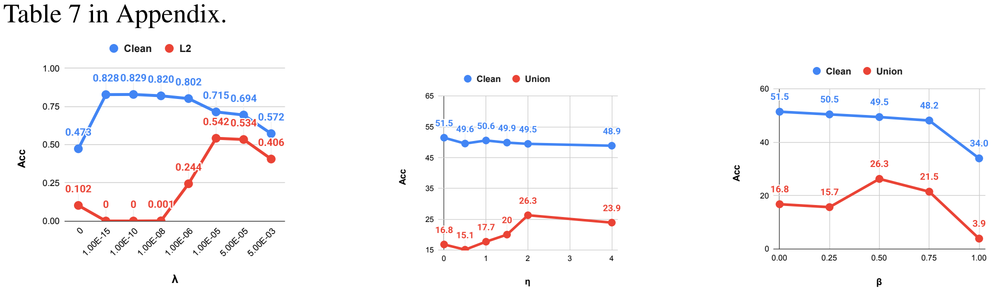


Figure 5: CURE-Max and CURE-Scratch bound difference visualization.

In addition, in Table 2a, we display the certified robustness of **CURE** compared with single-norm baselines against patch 2×2 attacks. Our framework outperforms related baselines with 8.5% for MNIST and 16.0% for CIFAR-10, showing better *generalized certified robustness*. We hypothesize that many non-l_p perturbations can be approximated or parameterized using l_p-bounded formulations, and improving l_p robustness enhances robustness to such transformations - we find that CURE training achieves significantly higher bound overlap compared to single-norm models (Table 8). However, we also observe that a geometrically robust model lacks multi-norm robustness, as shown in Table 7 in Appendix.



(a) λ_2: subselection ratio for l_2. (b) η: weight for bound alignment. (c) β: hyper-parameter for GP.

Figure 6: Alabtion studies on λ_2, η and β hyper-parameters.

4.2 Ablation Study and Discussions

Subselection ratio λ. For l_∞ certified training, we use the same λ_∞ as in Müller et al. (2022). For λ_2, in Figure 6a, we show the l_2 certified robustness using varying $\lambda_2 \in [0, ..., 1e^{-2}]$ with $\epsilon_2 = 0.5$. According to Figure 6a, we choose $\tau_2 = 1e^{-5}$.

Bound alignment (BA) hyper-parameter η. We perform CIFAR-10 ($\epsilon_\infty = \frac{8}{255}, \epsilon_2 = 0.5, \epsilon_1 = 1.0$) experiments with η values in $[0.5, 1.0, 1.5, 2.0, 4.0]$. In Figure 6b, the clean accuracy generally drops as we have larger η values, with union accuracy improving then dropping. We pick $\eta = 2.0$ with the best union accuracy for most experiments.

Gradient projection (GP) hyper-parameter β. Figure 6c displays the change of clean and union accuracy with choices of varying β values on CIFAR-10 ($\epsilon_\infty = \frac{8}{255}, \epsilon_2 = 0.5, \epsilon_1 = 1.0$). CURE-Scratch is generally insensitive to β values. Thus, we choose $\beta = 0.5$ for the experiments.

Ablation study on BA and GP. In Table 2b, we show the ablation study of BA and GP techniques on CIFAR-10 ($\epsilon_\infty = \frac{8}{255}, \epsilon_2 = 0.5, \epsilon_1 = 1.0$) experiment. BA and GP improve union accuracy by 6.8% and 2.7%, demonstrating the individual effectiveness of our proposed techniques.

Methods	MNIST	CIFAR-10
l_∞	68.9	0.0
l_2	0.0	0.0
CURE-Joint	68.5	0.2
CURE-Max	65.8	0.1
CURE-Random	72.8	**16.0**
CURE-Scratch	**77.4**	10.1

(a) Robust accuracy against 2×2 patch attacks on MNIST ($\epsilon_1 = 2.0, \epsilon_2 = 1.0, \epsilon_\infty = 0.3$) and CIFAR-10 ($\epsilon_1 = \frac{72}{255}, \epsilon_2 = \frac{36}{255}, \epsilon_\infty = \frac{1}{255}$) datasets. Results show CURE significantly outperforms single-norm training.

	Clean	l_∞	l_2	l_1	Union
CURE-Max	51.5	33.9	19.5	21.6	16.8
+BA	50.2	33.8	25.4	27.9	23.6
+BA + GP	49.5	34.2	28.1	32.0	**26.3**

(b) Ablations on BA and GP.

Visualization of bound differences. Figure 5 displays the bound differences $\{\underline{o}_y - \overline{o}_i\}_{i=0, i\neq y}^{i<k}$ of one example that is improved by **CURE-Scratch** (second row), compared with the **CURE-Max** (first row), from the CIFAR-10 ($\epsilon_\infty = \frac{8}{255}, \epsilon_2 = 0.5, \epsilon_1 = 1.0$) experiment. We use outputs from α, β-CROWN. For l_2 perturbations (blue diagrams), **CURE-Scratch** consistently shows positive bound differences enabling robust union prediction, while **CURE-Max** has several negative ones (highlighted in red). The second-row distributions are more aligned than the first, showing that **CURE-Scratch** effectively aligns bound differences. This highlights the effectiveness of the bound alignment method. Additional visualizations are in Appendix D.

Time cost of CURE. The extra training costs of GP are small, taking $6, 24, 82$ seconds using a single NVIDIA A40 GPU on MNIST, CIFAR-10, and TinyImageNet datasets (Table 14), respectively. Compared with the total training cost of **CURE-Scratch**, it only accounts for $\sim 6\%$ of the total cost. For runtime comparison of different methods with the same number of training epochs, we have a complete runtime analysis (Table 13) in Appendix E for the MNIST experiment. **CURE-Joint** has the largest cost among all methods. **CURE-Scratch** has a small extra time cost than **CURE-Max**, showing our proposed techniques have little additional cost.

Limitations. For l_2 certified training, we use a l_∞ box instead of l_2 ball for bound propagation, which leads to more over-approximation and the potential loss of precision. Also, we notice drops in clean accuracy when training with **CURE** methods. BA and GP techniques lead to a slight decrease in clean accuracy in experiments. Further, our work does not claim to achieve universal certified robustness, but takes a step toward it by showing that multi-norm training offers broader certified robustness than single-norm or geometric-certified models (Table 7).

5 Conclusion

We propose a framework **CURE** with multi-norm certified training methods for better union robustness. We establish a theoretical framework to analyze the tradeoff between perturbations, which inspires us to devise bound alignment, gradient projection, and robust certified fine-tuning techniques to enhance and facilitate the union-certified robustness. Extensive experiments on MNIST, CIFAR-10, and TinyImagenet show that **CURE** significantly improves union accuracy and robustness against geometric and patch transformations, paving the path to generalized certified robustness.

6 Ethics Statement

This work adheres to the ICLR Code of Ethics. Our research focuses on developing methods for improving the certified robustness of machine learning models. We do not involve human subjects, personal data, or sensitive demographic attributes in our experiments, as all evaluations are conducted on widely used benchmark datasets such as MNIST, CIFAR-10, and TinyImageNet that are publicly available and ethically sourced. While robustness research has the potential to be misused for creating stronger adversarial attacks, we emphasize that our contributions are specifically designed to advance defense techniques, improve safety guarantees, and guide the development of trustworthy AI systems. We release our code and results in alignment with principles of transparency, reproducibility, and research integrity, while carefully avoiding the release of harmful attack-specific artifacts beyond what is necessary for scientific validation. Our work complies with relevant privacy, security, and fairness considerations, and we believe it contributes positively toward the broader goal of building safer and more reliable AI systems.

7 Reproducibility Statement

We provide the source code of **CURE** as part of the supplementary material that can be used to reproduce our results. We provide the details of our hyper-parameters, training scheme, and model architecture in Section 4. We also provide additional details including other training details, further evaluation, and pseudocode not covered in the main text in the appendix.

References

Mislav Balunović and Martin Vechev. Adversarial training and provable defenses: Bridging the gap. In *8th International Conference on Learning Representations (ICLR 2020)(virtual)*. International Conference on Learning Representations, 2020.

Mislav Balunovic, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin Vechev. Certifying geometric robustness of neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pp. 831–840. PMLR, 2019.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in neural information processing systems*, 32, 2019.

Ping-yeh Chiang, Renkun Ni, Ahmed Abdelkader, Chen Zhu, Christoph Studor, and Tom Goldstein. Certified defenses for adversarial patches. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HyeaSkrYPH`.

Francesco Croce and Matthias Hein. Adversarial robustness against multiple and single l_p-threat models via quick fine-tuning of robust classifiers. In *International Conference on Machine Learning*, pp. 4436–4454. PMLR, 2022.

Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of adversaries. *Advances in Neural Information Processing Systems*, 31, 2018.

Sihui Dai, Chong Xiang, Tong Wu, and Prateek Mittal. Position paper: Beyond robustness against single attack types. *CoRR*, abs/2405.01349, 2024. URL `https://doi.org/10.48550/arXiv.2405.01349`.

Edoardo Debenedetti and Carmela Troncoso—EPFL. Adversarially robust vision transformers, 2022.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

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

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

Kai Hu, Klas Leino, Zifan Wang, and Matt Fredrikson. A recipe for improved certifiable robustness: Capacity and data. *arXiv preprint arXiv:2310.02513*, 2023.

Kai Hu, Andy Zou, Zifan Wang, Klas Leino, and Matt Fredrikson. Unlocking deterministic robustness certification on imagenet. *Advances in Neural Information Processing Systems*, 36, 2024.

Enyi Jiang and Gagandeep Singh. Ramp: Boosting adversarial robustness against multiple l_p perturbations. *arXiv preprint arXiv:2402.06827*, 2024.

Daniel Kang, Yi Sun, Tom Brown, Dan Hendrycks, and Jacob Steinhardt. Transfer of adversarial robustness between perturbation types. *arXiv preprint arXiv:1905.01034*, 2019.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pp. 97–117. Springer, 2017.

Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pp. 99–112. Chapman and Hall/CRC, 2018.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. URL `https://api.semanticscholar.org/CorpusID:16664790`.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks. In *International Conference on Machine Learning*, pp. 6212–6222. PMLR, 2021.

Aishan Liu, Shiyu Tang, Xianglong Liu, Xinyun Chen, Lei Huang, Zhuozhuo Tu, Dawn Song, and Dacheng Tao. Towards defending multiple adversarial perturbations via gated batch normalization. *arXiv preprint arXiv:2012.01654*, 2020.

Shao-Yuan Lo and Vishal M Patel. Defending against multiple and unforeseen adversarial videos. *IEEE Transactions on Image Processing*, 31:962–973, 2021.

Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Learning to generate noise for multi-attack robustness. In *International Conference on Machine Learning*, pp. 7279–7289. PMLR, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Pratyush Maini, Eric Wong, and Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pp. 6640–6650. PMLR, 2020.

Pratyush Maini, Xinyun Chen, Bo Li, and Dawn Song. Perturbation type categorization for multiple adversarial perturbation robustness. In *Uncertainty in Artificial Intelligence*, pp. 1317–1327. PMLR, 2022.

Ravi Mangal, Klas Leino, Zifan Wang, Kai Hu, Weicheng Yu, Corina Pasareanu, Anupam Datta, and Matt Fredrikson. Is certifying l_p robustness still worthwhile? *arXiv preprint arXiv:2310.09361*, 2023.

Yuhao Mao, Mark Müller, Marc Fischer, and Martin Vechev. Connecting certified and adversarial training. *Advances in Neural Information Processing Systems*, 36, 2024.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pp. 3578–3586. PMLR, 2018.

Mark Niklas Müller, Franziska Eckert, Marc Fischer, and Martin Vechev. Certified training: Small boxes are all you need. *arXiv preprint arXiv:2210.04871*, 2022.

Soumalya Nandi, Sravanti Addepalli, Harsh Rangwani, and R Venkatesh Babu. Certified adversarial robustness within multiple perturbation bounds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2298–2305, 2023.

Jay Nandy, Wynne Hsu, and Mong Li Lee. Approximate manifold defense against multiple adversarial perturbations. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

Alessandro De Palma, Rudy R Bunel, Krishnamurthy Dj Dvijotham, M. Pawan Kumar, Robert Stanforth, and Alessio Lomuscio. Expressive losses for verified robustness via convex combinations. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=mzyZ4wzKlM`.

ShengYun Peng, Weilin Xu, Cory Cornelius, Matthew Hull, Kevin Li, Rahul Duggal, Mansi Phute, Jason Martin, and Duen Horng Chau. Robust principles: Architectural design principles for adversarially robust cnns. *arXiv preprint arXiv:2308.16258*, 2023.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31, 2018.

Zhouxing Shi, Yihan Wang, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Fast certified robust training with short warmup. *Advances in Neural Information Processing Systems*, 34:18335–18349, 2021.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Advances in neural information processing systems*, 31, 2018.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–30, 2019.

Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. *Advances in neural information processing systems*, 32, 2019.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. *Advances in Neural Information Processing Systems*, 34:29909–29921, 2021.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36246–36263. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/wang23ad.html`.

Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *Advances in Neural Information Processing Systems*, 31, 2018.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

Jiancong Xiao, Zeyu Qin, Yanbo Fan, Baoyuan Wu, Jue Wang, and Zhi-Quan Luo. Adaptive smoothness-weighted adversarial training for multiple perturbations with its stability analysis. *arXiv preprint arXiv:2210.00557*, 2022.

Kaidi Xu, Chenan Wang, Hao Cheng, Bhavya Kailkhura, Xue Lin, and Ryan Goldhahn. Mixture of robust experts (more): A robust denoising method towards multiple perturbations. *arXiv preprint arXiv:2104.10586*, 2021.

Xiaojun Xu, Linyi Li, and Bo Li. Lot: Layer-wise orthogonal training on improving l2 certified robustness. *Advances in Neural Information Processing Systems*, 35:18904–18915, 2022.

Rem Yang, Jacob Laurel, Sasa Misailovic, and Gagandeep Singh. Provable defense against geometric transformations. *arXiv preprint arXiv:2207.11177*, 2022.

Hangsheng Zhang, Jiqiang Liu, and Jinsong Dong. Care: Ensemble adversarial robustness evaluation against adaptive attackers for security applications. *arXiv preprint arXiv:2401.11126*, 2024.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019a.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in Neural Information Processing Systems*, 31:4939–4948, 2018. URL `https://arxiv.org/pdf/1811.00866.pdf`.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019b.

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=iAX0l6Cz8ub`.

A Proofs of the Theorems

In this section, we provide the proofs of the Theorems.

A.1 Proof of Theorem 3.2

Theorem 3.2 (restated). *Let $\mathcal{R}_\phi(f) := \mathbb{E}\phi(f(x)y)$ and $\mathcal{R}^*_\phi := \min_f \mathcal{R}_\phi(f)$. Under Assumption 1 in Zhang et al. (2019a), for any non-negative loss function ϕ such that $\phi(0) \geq 1$, any measurable $f : \mathcal{X} \to \mathbb{R}$, any probability distribution on $\mathcal{X} \times \{\pm 1\}$, IBP output bound differences from f as $d(x) = \overline{o}_i - \underline{o}_y (i \neq y)$, and any $\lambda > 0$, we have*

$$\begin{aligned}
\mathcal{R}_{union}(f) - \mathcal{R}^*_r &\leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \Pr[x'_r{\in}B_r(x, \epsilon_r), x'_q{\in}B_q(x, \epsilon_q), f(x'_r)y > 0 \textit{ and } f(x'_q)y \leq 0] \\
&\leq \psi^{-1}(\mathcal{R}_\phi(f){-}\mathcal{R}^*_\phi) + \mathbb{E} \max_{\substack{x'_r \in B_r(x,\epsilon_r),\\ x'_q \in B_q(x,\epsilon_q)}} (\phi(f(x'_r)f(x'_p)/\lambda), f(x'_r)y > 0) \\
&\leq \psi^{-1}(\mathcal{R}_\phi(f){-}\mathcal{R}^*_\phi) + \mathbb{E} \max_{\substack{x'_r \in B_r(x,\epsilon_r),\\ x'_q \in B_q(x,\epsilon_q)}} (\phi(d(x'_r)d(x'_p)/\lambda), \overline{o}_i \leq \underline{o}_y \textit{ for } d(x'_r)).
\end{aligned}$$

Proof. By Eqn. equation 4, $\mathcal{R}_{\text{union}}(f) - \mathcal{R}^*_r = \mathcal{R}_r(f) - \mathcal{R}^*_r + \mathcal{R}_{\text{align}}(f) \leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathcal{R}_{\text{align}}(f)$, where the last inequality holds because we choose ϕ as a classification-calibrated loss Bartlett et al. (2006). This leads to the first inequality.

Also, notice that

$$\begin{aligned}
&\Pr[x'_r{\in}B_r(x, \epsilon_r), x'_q{\in}B_q(x, \epsilon_q), f(x'_r)y > 0 \text{ and } f(x'_q)y \leq 0] \\
&\leq \Pr[x'_r{\in}B_r(x, \epsilon_r), x'_q{\in}B_q(x, \epsilon_q), f(x'_r)f(x'_q) \leq 0, f(x'_r)y > 0] \\
&= \mathbb{E} \max_{x'_r \in B_r(x,\epsilon_r)} \max_{x'_q \in B_q(x,\epsilon_q)} (\mathbf{1}\{f(x'_r) \neq f(x'_q)\}, f(x'_r)y > 0) \\
&= \mathbb{E} \max_{x'_r \in B_r(x,\epsilon_r)} \max_{x'_q \in B_q(x,\epsilon_q)} (\mathbf{1}\{f(x'_r)f(x'_q)/\lambda < 0\}, f(x'_r)y > 0) \\
&\leq \mathbb{E} \max_{x'_r \in B_r(x,\epsilon_r)} \max_{x'_q \in B_q(x,\epsilon_q)} (\phi(f(x'_r)f(x'_q)/\lambda), f(x'_r)y > 0) \\
&\leq \mathbb{E} \max_{x'_r \in B_r(x,\epsilon_r)} \max_{x'_q \in B_q(x,\epsilon_q)} (\phi(d(x'_r)d(x'_p)/\lambda), \overline{o}_i \leq \underline{o}_y \text{ for } d(x'_r)).
\end{aligned}$$

The last inequality holds because the adversarial loss is always upper-bounded by the IBP loss. Therefore, we get the second and third inequality in Theorem A.1. □

A.2 Theory of connecting NT with CT

The proof for connecting NT with CT via gradient projection (GP) is very similar to what has been done in Jiang & Singh (2024), where authors analyze and compare the delta errors of two aggregation rules (standard training and training with GP). Delta errors are the indicators of convergences of different aggregation rules based on a mild assumption on the Lipschitz continuity of loss function gradients. GP leads to a smaller Delta error, which means GP results in a better convergence. The only difference in connecting NT with CT is that we use a different loss function compared with adversarial training, which makes the proof almost the same. One can refer to Jiang & Singh (2024) for the more detailed proof of GP.

B Related Work

Neural network verification. We rely on deterministic verification techniques to evaluate robustness under multiple norms. Although exact verification is NP-complete and infeasible for large models (Katz et al., 2017), scalable relaxations such as abstract interpretation (Singh et al., 2019) and convex optimization approaches (Wang et al., 2021) make it possible to obtain sound, though sometimes conservative, certificates. These methods are widely used in certified training because they strike a balance between tractability and rigor, enabling provable guarantees at scale. Our analysis of multi-norm certified training builds on this foundation, leveraging deterministic verification to provide stronger and more general robustness guarantees.

Certified training. For l_∞ certified training, a widely-used method IBP (Mirman et al., 2018; Gowal et al., 2018) minimizes a sound over-approximation of the worst-case loss, calculated using the Box relaxation method. Wong et al. (2018) applies DeepZ (Singh et al., 2018) relaxations, estimating using Cauchy random projections. CROWN-IBP (Zhang et al., 2019b) integrates efficient Box propagation with precise linear relaxation-based bounds during the backward pass to estimate the worst-case loss. Balunović & Vechev (2020) consists of a verifier that aims to certify the network using convex relaxation and an adversary that tries to find inputs causing verification to fail. Shi et al. (2021) proposes a new weight initialization method for IBP, adds Batch Normalization (BN) to each layer and designs regularization with a short warmup schedule. Besides this, SABR (Müller et al., 2022) and TAPS (Mao et al., 2024) are unsound improvements over IBP by connecting IBP to adversarial attacks and adversarial training. For l_2 deterministic certified training, recent works (Leino et al., 2021; Xu et al., 2022; Hu et al., 2023; 2024) are based on Lipschitz-based certification methods. They design specialized architectures under a particular l_p norm, which do not naturally extend to robustness under the diverse settings considered in our work. To the best of our knowledge, **CURE** is the first deterministic framework for multi-norm certified robustness, compatible with diverse model architectures. In comparison to previous works, **CURE** is a more general deterministic framework for multi-norm certified robustness.

Robustness against multiple perturbations. Adversarial Training (AT) usually employs gradient descent to discover adversarial examples and incorporates them into training for enhanced adversarial robustness (Tramèr et al., 2017; Madry et al., 2017). Numerous works focus on improving robustness (Zhang et al., 2019a; Carmon et al., 2019; Raghunathan et al., 2020; Wang et al., 2020; Wu et al., 2020; Gowal et al., 2020; Zhang et al., 2021; Debenedetti & Troncoso—EPFL, 2022; Peng et al., 2023; Wang et al., 2023) against a *single* perturbation type while remaining vulnerable to other types. Tramer & Boneh (2019); Kang et al. (2019) observe that robustness against l_p attacks does not necessarily transfer to other l_q attacks ($q \neq p$). Previous studies (Tramer & Boneh, 2019; Maini et al., 2020; Madaan et al., 2021; Croce & Hein, 2022; Jiang & Singh, 2024) modified Adversarial Training (AT) to enhance robustness against multiple l_p attacks, employing average-case (Tramer & Boneh, 2019), worst-case (Tramer & Boneh, 2019; Maini et al., 2020; Jiang & Singh, 2024), and random-sampled (Madaan et al., 2021; Croce & Hein, 2022) defenses. There are also works (Nandy et al., 2020; Liu et al., 2020; Xu et al., 2021; Xiao et al., 2022; Maini et al., 2022) that use preprocessing, ensemble methods, mixture of experts, and stability analysis to solve this problem. For multi-norm certified robustness, Nandi et al. (2023) study the certified multi-norm robustness with probabilistic guarantees. They apply randomized smoothing, which is expensive to compute in nature, making it impractical for real-world applications. Our work in contrast to these works, proposes the first *deterministic* certified multi-norm training for better multi-norm and generalized certified robustness.

C More training details

Certified training for l_2 robustness. We propose a new l_2 deterministic certified training method, inspired by SABR Müller et al. (2022). For the specified ϵ_2 and τ_2 values, we first generate adversarial examples by computing the gradient in the l_2 direction (Kim, 2020), then truncating the perturbation to lie within a slightly reduced l_∞ ball $B_\infty(x, \epsilon_2 - \tau_2)$. After that, we propagate a smaller box region $B_\infty(x', \tau_2)$ using the IBP loss. The loss we optimize can be formulated as follows:

$$\mathcal{L}_{l_2}(x, y, \epsilon_2, \tau_2) = \max_{x' \in B_\infty(x, \epsilon_2 - \tau_2)} \mathcal{L}_{\text{IBP}}(x', y, \tau_2)$$

Training details. We mostly follow the hyper-parameter choices from Müller et al. (2022) for **CURE**. We include weight initialization and warm-up regularization from Shi et al. (2021). Further, we use ADAM (Kingma, 2014) with an initial learning rate of $1e^{-4}$, decayed twice with a factor of 0.2. For CIFAR-10, we train 160 and 180 epochs for ($\epsilon_\infty = \frac{2}{255}, \epsilon_2 = 0.25$) and ($\epsilon_\infty = \frac{8}{255}, \epsilon_2 = 0.5$), respectively. We decay the learning rate after 120 and 140, 140 and 160 epochs, respectively. For the TinyImagenet experiment, we use the same setting as the CIFAR-10 ($\epsilon_\infty = \frac{8}{255}, \epsilon_2 = 0.5$) experiment. For the MNIST dataset, we train 70 epochs, decaying the learning rate after 50 and 60 epochs. For batch size, we set 128 for CIFAR-10 and TinyImagenet and 256 for MNIST. For all experiments, we first perform one epoch of standard training. Also, we anneal $\epsilon_\infty, \epsilon_2$ from 0 to their final values with 80 epochs for CIFAR-10 and TinyImagenet and 20 epochs for MNIST. We only apply GP after training with the final epsilon values. For certification, we verify 1000 examples on MNIST and CIFAR-10, as well as 170 examples on TinyImagenet. The values of all hyperparameters can be found in Table 3.

	MNIST-small	MNIST-large	CIFAR-small	CIFAR-large	TinyImagenet
λ_∞	0.4	0.6	0.1	0.7	0.4
λ_2	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05
Learning rate	1.00E-04	1.00E-04	1.00E-04	1.00E-04	1.00E-04
LR decay ratio	0.2	0.2	0.2	0.2	0.2
Training epochs	70	70	160	180	180
Decay epochs	50, 60	50, 60	120, 140	140, 160	140, 160
Batch size	256	256	128	128	128
α (CURE)	0.5	0.5	0.5	0.5	0.5
η (CURE)	2.0	0.5	2.0	2.0	2.0
β (CURE)	0.5	0.5	0.5	0.5	0.5

Table 3: Training specifications of our main experiments on MNIST, CIFAR-10, and TinyImagenet.

Certifications for evaluations on l_1, l_2, l_∞ norms. We evaluated our trained models using α, β-CROWN (Zhang et al., 2018). Specifically, α, β-CROWN employs an efficient linear bound propagation framework coupled with a branch-and-bound algorithm to certify the robustness of neural networks against adversarial attacks. It propagates bounds on network outputs layer-by-layer. These bounds are linear functions representing the range of potential values the network's output can take under a given set of input constraints. In addition, the branch-and-bound algorithm systematically divides the input space into smaller regions (branching) and computes tighter bounds on each subregion. α, β-CROWN is versatile and supports various activation functions, including ReLU, sigmoid, and tanh, making it applicable to a wide range of neural network architectures. Also, it supports the certification on different $l_p (p = 1, 2, \infty)$ norms, which fits the goal of our CURE framework for multi-norm certified robustness.

D Other experiment results and ablation studies

Additional experiment on MNIST ($\epsilon_\infty = 0.1$, $\epsilon_2 = 0.5$, $\epsilon_1 = 1.0$) and CIFAR-10 ($\epsilon_\infty = \frac{2}{255}$, $\epsilon_2 = 0.25$, $\epsilon_1 = 0.5$). As shown in Table 4, our CURE-Scratch method achieves higher union-certified accuracy on both MNIST and CIFAR-10 compared to all baseline methods. This demonstrates that training from scratch with our proposed multi-norm certified training framework not only consistently outperforms single-norm approaches.

Additional experiment on CIFAR-100 ($\epsilon_\infty = 2/255$, $\epsilon_2 = 0.25$, $\epsilon_1 = 0.5$). As shown in Table 5, our CURE-Scratch method significantly improves union-certified accuracy on the CIFAR-100 dataset compared to all baseline methods. Specifically, CURE-Scratch reaches a union accuracy of 30.4%, outperforming CURE-Joint, CURE-MAX, and CURE-Random by substantial margins.

Robustness against geometric transformations on CIFAR-10. Table 6 displays the certified robustness against geometric transformations on CIFAR-10. **CURE** outperforms the single-norm baselines with significant margins. Also, we notice that CURE-Scratch improves CURE-Max, which indicates the effectiveness of bound alignment and gradient projection.

CURE compares to models trained to be robust against geometric perturbations. To evaluate whether geometric robustness generalizes to multi-norm robustness, we conducted additional experiments on CIFAR-10 using ($\epsilon_\infty = 2/255, \epsilon_2 = 0.25, \epsilon_1 = 0.5$). We tested models trained with various geometric transformations, including rotation $R(\varphi)$, translation $T_u(\Delta u), T_v(\Delta v)$, scaling $Sc(\lambda)$, shearing $Sh(\gamma)$, contrast $C(\alpha)$, and brightness $B(\beta)$, where the values denote the perturbation magnitudes (e.g., $R(10)$ applies up to $\pm 10^\circ$ rotation). As shown in the table below, models trained with geometric perturbations Yang et al. (2022) achieve substantially lower union certified accuracy (e.g., 21.5%) compared to our CURE model (61.2%). This indicates that geometric robustness alone

Dataset	$(\epsilon_\infty, \epsilon_2, \epsilon_1)$	Methods	Clean	l_∞	l_2	l_1	Union
MNIST	(0.1, 0.5, 1.0)	l_∞	99.2	97.0	96.5	95.0	94.9
		l_2	99.5	2.6	98.6	98.0	2.6
		CURE-Joint	99.2	97.6	97.9	97.5	97.2
		CURE-Max	99.3	97.6	97.3	96.8	96.8
		CURE-Random	99.2	97.3	97.2	97.1	96.9
		CURE-Finetune	99.1	97.0	97.3	96.8	96.5
		CURE-Scratch	99.2	97.5	98.0	97.9	**97.5**
CIFAR-10	$(\frac{2}{255}, 0.25, 0.5)$	l_∞	79.2	60.3	67.3	75.9	60.3
		l_2	82.1	5.8	71.3	81.7	5.8
		CURE-Joint	79.4	56.2	68.1	77.1	56.2
		CURE-Max	77.6	60.0	69.3	75.2	60.0
		CURE-Random	78.4	59.0	68.5	76.9	58.9
		CURE-Finetune	78.0	59.7	68.2	75.9	59.7
		CURE-Scratch	76.0	61.2	67.7	74.6	**61.2**

Table 4: Comparison of the clean accuracy, individual, and union certified accuracy (%). **CURE** consistently improves union accuracy compared with single-norm training with significant margins on all datasets. **CURE-Scratch** and **CURE-Finetune** outperform other methods in most cases.

	Clean	**Linf (2/255)**	**L2 (0.25)**	**L1 (0.5)**	**Union**
Linf (2/255)	39.7	26.4	16.0	18.6	14.8
L2 (0.25)	54.3	2.4	37.4	47.4	2.4
CURE-Joint	42.5	28.0	26.8	32.4	26.0
CURE-MAX	40.7	26.8	22.8	29.4	22.6
CURE-Random	41.3	28.4	27.2	34.0	27.0
CURE-Scratch	40.4	30.6	31.4	36.2	**30.4**

Table 5: Multi-norm certified accuracy (%) on CIFAR100 dataset.

does not transfer well to multi-norm robustness, while our approach offers strong generalization across diverse norm-bounded threats.

Comparing bound overlap across models. In Table 8, we compare single-norm and multi-norm trained models in terms of their bound overlap with CGT models. For fairness, we compute the maximum overlap across each batch and normalize the bound outputs. The results show that CURE-Scratch achieves substantially higher overlap than the ℓ_∞ certified baseline, highlighting its stronger alignment and generalization across perturbation types.

l_∞, l_2 and CURE-scratch trained on CIFAR-10 ($\epsilon_\infty = 8/255$, $\epsilon_2 = 0.5$, $\epsilon_1 = 1.0$) union certified robustness analysis with varying l_∞, l_2, and l_1 epsilons. We evaluate the certified robustness of our trained l_∞, l_2, and CURE-Scratch models across a range of perturbation sizes under l_1, l_2, and l_∞ norms. This comprehensive evaluation reveals that CURE-Scratch consistently outperforms the single-norm trained models across all tested settings. The results highlight the effectiveness of our approach in achieving strong multi-norm certified robustness, demonstrating that CURE-Scratch not

Configs	R(30)	$T_u(2),T_v(2)$	Sc(5),R(5), C(5),B(0.01)	Sh(2),R(2),Sc(2), C(2),B(0.001)	Avg
l_∞	54.6	20.9	82.5	95.6	63.4
l_2	0.0	0.0	0.0	0.0	0.0
CURE-Joint	**55.9**	21.3	82.3	**95.7**	63.8
CURE-Max	50.1	20.7	80.2	94.8	61.5
CURE-Random	54.8	18.8	83.5	95.6	63.2
CURE-Scratch	51.0	**24.3**	**85.5**	95.1	**64.0**

Table 6: Comparison on **CURE** against geometric transformations for MNIST ($\epsilon_1 = 2.0, \epsilon_2 = 1.0, \epsilon_\infty = 0.3$) experiment. **CURE** improves the generalized certified robustness significantly compared with single norm training.

Method	ℓ_∞	ℓ_2	ℓ_1	Union
CGT: R(10)	1.6	22.8	38.0	1.6
CGT: R(2), Sh(2)	32.2	22.8	38.0	18.1
CGT: Sc(1), R(1), C(1), B(0.001)	33.2	22.8	38.0	21.5
Ours	61.2	67.7	74.6	**61.2**

Table 7: Comparison of CGT (Yang et al., 2022) versus our CURE model on CIFAR-10.

	R(10)	R(2), Sh(2)	Sc(1), R(1), C(1), B(0.001)
Linf	0.141	0.723	0.791
CURE-Scratch	**0.277**	**0.789**	**0.892**

Table 8: Comparison of multi-norm (CURE-Scratch) versus single-norm certified trained models on the bound overlap with the CGT model.

only generalizes better across different norms but also maintains superior certification performance under varying attack strengths.

The overlapping of l_∞ and l_2 balls. A reasonable misconception is that because l_∞ and l_2 balls contain some overlap, training for robustness in one norm will sufficiently account for the weakness in the other. Besides choices of l_∞ and l_2 that completely overlap each other, the true regions of successful attacks have a significant mismatch across different norms.

To illustrate the mismatch between l_∞ and l_2 regions, it suffices to show the existence of successful attacks that lie further enough from the original data input such that they are not covered by the other norm ball. We ran PGD 19968 times through a full testing run of a naturally trained model in the MNIST setting ($\epsilon_\infty = 0.1, \epsilon_2 = 0.5$) with the following results:

% of ℓ_∞ attacks not in ℓ_2 ball	% of ℓ_2 attacks not in ℓ_∞ ball
100.00% (9984/9984)	98.95% (9879/9984)

Table 9: Comparison of ℓ_∞ and ℓ_2 attack coverage.

Of course, PGD may not find the absolute strongest adversarial examples in each ball. That only makes the mismatch claim stronger, because if PGD can find adversarial examples outside the other norm's ball, more attacks almost certainly exist in those regions as well.

Comparison of l_2 certified training and PGD training. Table 10 shows the l_2 certified robustness comparison between certified training and PGD training. The results demonstrate that determinist-certified training greatly improves the certified robustness.

Hyper-parameter α for Joint certified training. As shown in Table 11, we test the changing of l_∞, l_2, and union accuracy with different α values in $[0, 0.25, 0.5, 0.75, 1.0]$ on MNIST ($\epsilon_\infty = 0.1, \epsilon_2 = 0.5$) experiments. We observe that $\alpha = 0.5$ has the best union accuracy and is generally a good choice for our experiments by balancing the two losses.

Comparison of l_2 certified robustness on l_2 deterministic certified training methods. In Table 12, we compare our proposed l_2 certified defense with SOTA l_2 certified defense Hu et al. (2023) on CIFAR-10 with $\epsilon_2 = 0.25$ and 0.5. The results show that our proposed l_2 deterministic certified training method improves over l_2 robustness by $2 \sim 4\%$ compared with the SOTA method.

More visualizations on bound differences. We plot the bound difference examples from alpha-beta-crown on MNIST, CIFAR-10, and TinyImagenet datasets, where the negative bound differences are colored in red. As shown in Figure 10, 11, 12, we compare CURE-Scratch (second row) with CURE-Max (first row), with bound differences against l_∞ and l_2 perturbations colored in blue and green, respectively. CURE-Scratch produces all positive bound differences, leading to unionly robust predictions; CURE-Max is not unionly robust due to some negative bound differences. Also, we observe that CURE-Scratch successfully brings l_q, l_r bound difference distributions close to each

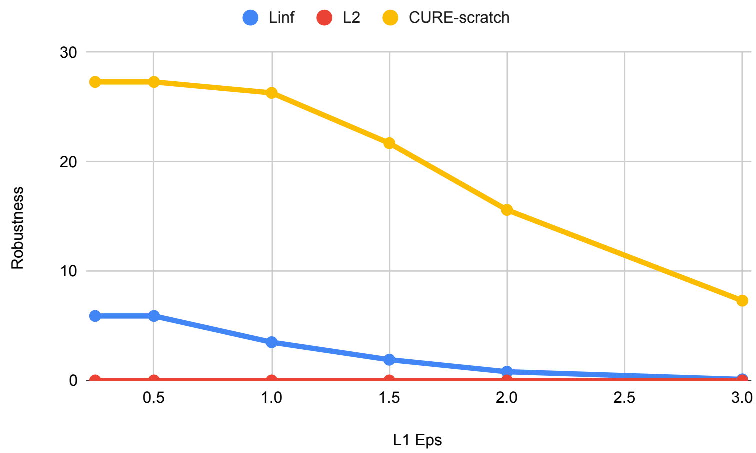


Figure 7: l_∞, l_2 and CURE-scratch trained on CIFAR-10 union certified robustness analysis with varying l_1 epsilons.

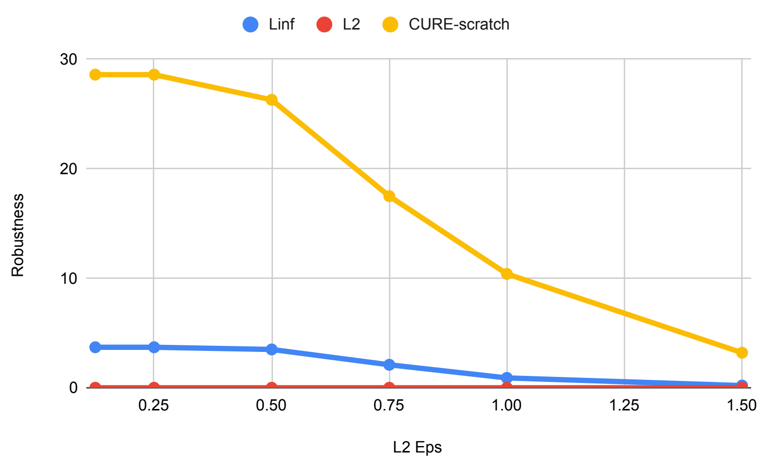


Figure 8: l_∞, l_2 and CURE-scratch trained on CIFAR-10 union certified robustness analysis with varying l_2 epsilons.

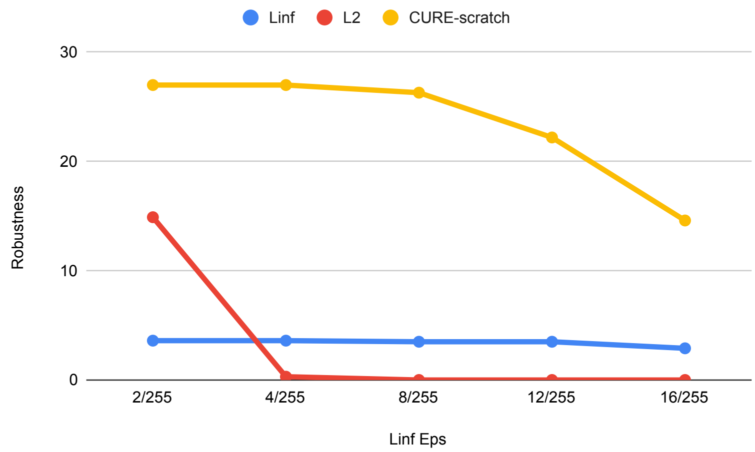


Figure 9: l_∞, l_2 and CURE-scratch trained on CIFAR-10 union certified robustness analysis with varying l_∞ epsilons.

l_2 certified robustness	MNIST-large	CIFAR-small	CIFAR-large
Certified training	**94.5**	**71.2**	**56.6**
PGD training	74.3	23.3	10.2

Table 10: Comparison on l_2 certified robustness between certified and PGD training.

α	0.0	0.25	0.5	0.75	1.0
Clean	99.2	99.2	99.3	99.2	99.5
l_∞	97.7	97.7	97.5	97.2	2.0
l_2	96.9	95.6	97.4	95.9	98.7
Union	96.9	95.6	**97.1**	95.8	2.0

Table 11: Ablation study on Joint training hyper-parameter α.

ϵ_2	0.25	0.5
Hu et al. (2023)	69.5	52.2
Ours	**71.2**	**56.6**

Table 12: Comparison of l_2 certified accuracy: our proposed l_2 certified training consistently outperforms Hu et al. (2023) by $2 \sim 4\%$.

other compared with CURE-Max in many cases, which confirms the effectiveness of our bound alignment technique.

E RUNTIME ANALYSIS

This section provides the runtime per training epoch for all methods on MNIST ($\epsilon_\infty = 0.1, \epsilon_2 = 0.75$) experiments and runtime per training epoch of CURE-Scratch with ablation studies on GP for MNIST, CIFAR10, and TinyImagenet experiments. We evaluate all the methods on a single A40 Nvidia GPU with 40GB memory and the runtime is reported in seconds (s).

Runtime for different methods on MNIST experiments. In Table 13, we show the time in seconds (s) per training epoch for single norm training (l_∞ and l_2), CURE-Joint, CURE-Max, CURE-Random, CURE-Scratch, and CURE-Finetune methods. CURE-Finetune has a relatively small training cost compared with other methods and CURE-Joint has the highest time cost (around two times of other methods) per epoch. The results indicate the efficiency of training with CURE-Scratch/Finetune.

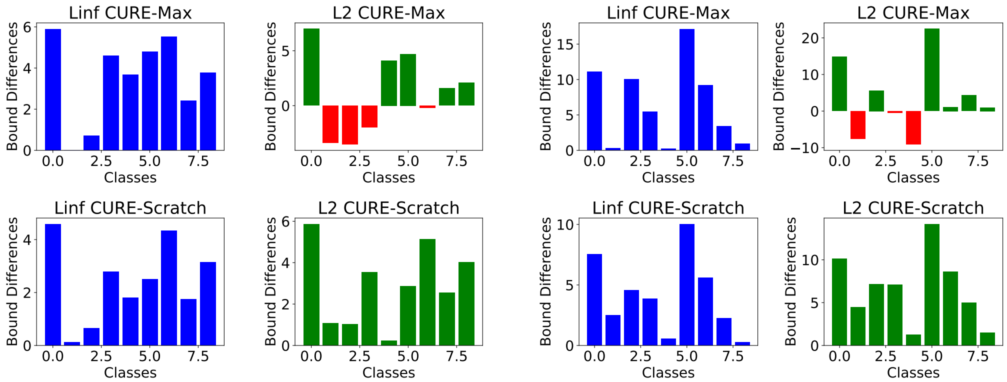


Figure 10: Bound difference visualizations on MNIST ($\epsilon_\infty = 0.3, \epsilon_2 = 1.0$) experiments.

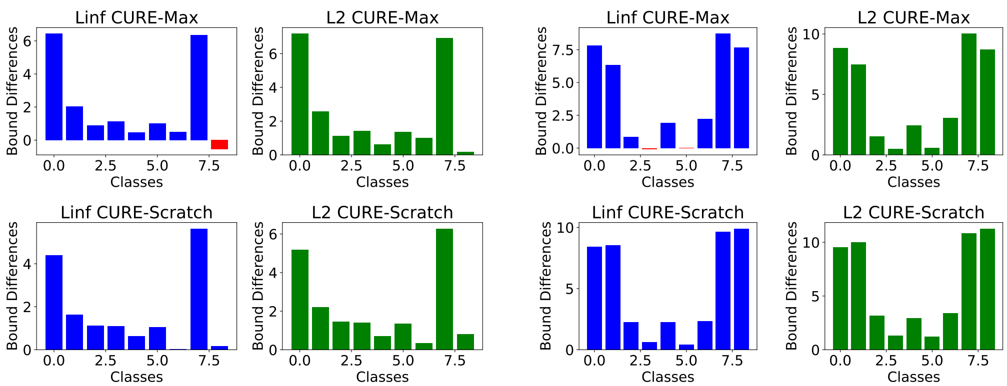


Figure 11: Bound difference visualizations on CIFAR-10 ($\epsilon_\infty = \frac{2}{255}, \epsilon_2 = 0.25$) experiments.

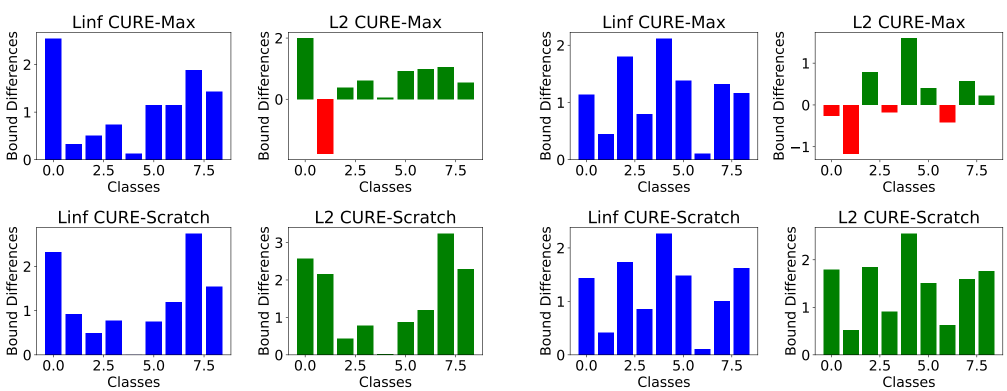


Figure 12: Bound difference visualizations on CIFAR-10 ($\epsilon_\infty = \frac{8}{255}, \epsilon_2 = 0.5$) experiments.

Runtime for CURE-Scratch on MNIST, CIFAR10, and TinyImagenet datasets. In Table 14, we show the runtime per training epoch using **CURE-Scratch** on MNIST, CIFAR10, and TinyImagenet datasets with and without GP operations. We see that the GP operation's cost is small compared with the whole training procedure, accounting for around 6% of the whole training time.

Methods	Runtime (s)
l_∞	89
l_2	82
CURE-Joint	155
CURE-Max	147
CURE-Random	101
CURE-Finetune	148
CURE-Scratch	153

Table 13: Runtime for all methods on MNIST ($\epsilon_\infty = 0.1, \epsilon_2 = 0.5$) experiment per epoch in seconds.

	MNIST	CIFAR-10	TinyImagenet
w/o GP	148	390	952
with GP	154	414	1036

Table 14: Runtime for CURE-Scratch on MNIST, CIFAR10, and TinyImagenet datasets.

F ALGORITHMS

In this section, we present the algorithms of **CURE** framework. Algorithm 1 illustrates how to get propagation region for both l_2 and l_∞ perturbations. Algorithm 2, 3, 4, 5 refer to algorithms of CURE-Joint, CURE-Max, CURE-Random, and CURE-Scratch/Finetune, respectively. Algorithm 6 is the procedure of performing GP after one epoch of natural and certified training (could be any of Algorithm 2, 3, 4, 5).

Algorithm 1 get_propagation_region for l_∞ and l_2 perturbations

Require: Neural network f, input $\boldsymbol{x}$, label t, perturbation radius ϵ, subselection ratio λ, step size α, step number n, attack types $\in \{l_\infty, l_2\}$

Ensure: Center $\boldsymbol{x}'$ and radius τ of propagation region $\mathcal{B}^\tau(\boldsymbol{x}')$

$(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}) \leftarrow \text{clamp}((\boldsymbol{x} - \epsilon, \boldsymbol{x} + \epsilon), 0, 1)$ // Get bounds of input region

$\boldsymbol{\tau} \leftarrow \lambda/2 \cdot (\overline{\boldsymbol{x}} - \underline{\boldsymbol{x}})$ // Compute propagation region size τ

$\boldsymbol{x}_0^* \leftarrow \text{Uniform}(\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}})$ // Sample PGD initialization

for $i = 0 \ldots n-1$ **do** // Do n PGD steps

  if attack = l_∞ **then** // Find examples with l_∞ gradient direction

    $\boldsymbol{x}_{i+1}^* \leftarrow \boldsymbol{x}_i^* + \alpha \cdot \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}_i^*} \mathcal{L}_{\text{CE}}(f(\boldsymbol{x}_i^*), t))$

    $\boldsymbol{x}_{i+1}^* \leftarrow \text{clamp}(\boldsymbol{x}_{i+1}^*, \underline{\boldsymbol{x}}, \overline{\boldsymbol{x}})$

  end if

  if attack = l_2 **then** // Find examples with l_2 gradient direction

    $\boldsymbol{x}_{i+1}^* \leftarrow \boldsymbol{x}_i^* + \alpha \cdot \frac{\nabla_{\boldsymbol{x}_i^*} \mathcal{L}_{\text{CE}}(f(\boldsymbol{x}_i^*), \boldsymbol{y})}{\|\nabla_{\boldsymbol{x}_i^*} \mathcal{L}_{\text{CE}}(f(\boldsymbol{x}_i^*), \boldsymbol{y})\|_2}$

    $\delta \leftarrow \frac{\epsilon}{\|\boldsymbol{x}_{i+1}^* - \boldsymbol{x}\|_2} \cdot (\boldsymbol{x}_{i+1}^* - \boldsymbol{x})$

    $\boldsymbol{x}_{i+1}^* \leftarrow \text{clamp}(\boldsymbol{x} + \delta, \underline{\boldsymbol{x}}, \overline{\boldsymbol{x}})$

  end if

end for

$\boldsymbol{x}' \leftarrow \text{clamp}(\boldsymbol{x}_n^*, \underline{\boldsymbol{x}} + \tau, \overline{\boldsymbol{x}} - \tau)$ // Ensure that $\mathcal{B}^\tau(\boldsymbol{x}')$ will lie fully in $\mathcal{B}^\epsilon(\boldsymbol{x})$

return $\boldsymbol{x}', \tau$

Algorithm 2 CURE-Joint Training Epoch

Require: Neural network f_θ, training set $(\boldsymbol{X}, \boldsymbol{T})$, perturbation radius ϵ_2 and ϵ_∞, subselection ratio λ_∞ and λ_2, learning rate η, ℓ_1 regularization weight ℓ_1, loss balance factor α

for $(\boldsymbol{x}, t) = (\boldsymbol{x}_0, t_0) \ldots (\boldsymbol{x}_b, t_b)$ **do** // Sample batches $\sim (\boldsymbol{X}, \boldsymbol{T})$
  $(\boldsymbol{x}'_\infty, \tau_\infty) \leftarrow$ get_propagation_region (attack = l_∞) // Refer to Algorithm 1
  $(\boldsymbol{x}'_2, \tau_2) \leftarrow$ get_propagation_region (attack = l_2)
  $\mathcal{B}^{\tau_\infty}(\boldsymbol{x}'_\infty) \leftarrow \textsc{Box}(\boldsymbol{x}'_\infty, \tau_\infty)$ // Get box with midpoint $\boldsymbol{x}'_\infty, \boldsymbol{x}'_2$ and radius τ_∞, τ_2
  $\mathcal{B}^{\tau_2}(\boldsymbol{x}'_2) \leftarrow \textsc{Box}(\boldsymbol{x}'_2, \tau_2)$
  $\boldsymbol{u}_{y^\Delta_\infty} \leftarrow$ get_upper_bound$(f_\theta, \mathcal{B}^{\tau_\infty}(\boldsymbol{x}'_\infty))$ // Get upper bound $\boldsymbol{u}_{y^\Delta_\infty}, \boldsymbol{u}_{y^\Delta_2}$ on logit differences
  $\boldsymbol{u}_{y^\Delta_2} \leftarrow$ get_upper_bound$(f_\theta, \mathcal{B}^{\tau_2}(\boldsymbol{x}'_2))$ // based on IBP
  $\text{loss}_{l_\infty} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{u}_{y^\Delta_\infty}, t)$
  $\text{loss}_{l_2} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{u}_{y^\Delta_2}, t)$
  $\text{loss}_{\ell_1} \leftarrow \ell_1 \cdot$ get_ℓ_1_norm(f_θ)
  $\text{loss}_{tot} \leftarrow (1-\alpha) \cdot \text{loss}_{l_\infty} + \alpha \cdot \text{loss}_{l_2} + \text{loss}_{\ell_1}$
  $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \text{loss}_{tot}$ // Update model parameters θ
end for

Algorithm 3 CURE-Max Training Epoch

Require: Neural network f_θ, training set $(\boldsymbol{X}, \boldsymbol{T})$, perturbation radius ϵ_2 and ϵ_∞, subselection ratio λ_∞ and λ_2, learning rate η, ℓ_1 regularization weight ℓ_1

for $(\boldsymbol{x}, t) = (\boldsymbol{x}_0, t_0) \ldots (\boldsymbol{x}_b, t_b)$ **do** // Sample batches $\sim (\boldsymbol{X}, \boldsymbol{T})$
  $(\boldsymbol{x}'_\infty, \tau_\infty) \leftarrow$ get_propagation_region (attack = l_∞) // Refer to Algorithm 1
  $(\boldsymbol{x}'_2, \tau_2) \leftarrow$ get_propagation_region (attack = l_2)
  $\mathcal{B}^{\tau_\infty}(\boldsymbol{x}'_\infty) \leftarrow \textsc{Box}(\boldsymbol{x}'_\infty, \tau_\infty)$ // Get box with midpoint $\boldsymbol{x}'_\infty, \boldsymbol{x}'_2$ and radius τ_∞, τ_2
  $\mathcal{B}^{\tau_2}(\boldsymbol{x}'_2) \leftarrow \textsc{Box}(\boldsymbol{x}'_2, \tau_2)$
  $\boldsymbol{u}_{y^\Delta_\infty} \leftarrow$ get_upper_bound$(f_\theta, \mathcal{B}^{\tau_\infty}(\boldsymbol{x}'_\infty))$ // Get upper bound $\boldsymbol{u}_{y^\Delta_\infty}, \boldsymbol{u}_{y^\Delta_2}$ on logit differences
  $\boldsymbol{u}_{y^\Delta_2} \leftarrow$ get_upper_bound$(f_\theta, \mathcal{B}^{\tau_2}(\boldsymbol{x}'_2))$ // based on IBP
  $\text{loss}_{l_\infty} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{u}_{y^\Delta_\infty}, t)$
  $\text{loss}_{l_2} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{u}_{y^\Delta_2}, t)$
  $\text{loss}_{Max} \leftarrow max(\text{loss}_{l_\infty}, \text{loss}_{l_2})$ // We select the largest $l_{p\in[2,\infty]}$ loss for each sample
  $\text{loss}_{\ell_1} \leftarrow \ell_1 \cdot$ get_ℓ_1_norm(f_θ)
  $\text{loss}_{tot} \leftarrow \text{loss}_{Max} + \text{loss}_{\ell_1}$
  $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \text{loss}_{tot}$ // Update model parameters θ
end for

Algorithm 4 CURE-Random Training Epoch

Require: Neural network f_θ, training set $(\boldsymbol{X}, \boldsymbol{T})$, perturbation radius ϵ_2 and ϵ_∞, subselection ratio λ_∞ and λ_2, learning rate η, ℓ_1 regularization weight ℓ_1

for $(\boldsymbol{x}, t) = (\boldsymbol{x}_0, t_0) \ldots (\boldsymbol{x}_b, t_b)$ **do** // Sample batches $\sim (\boldsymbol{X}, \boldsymbol{T})$
  $(\boldsymbol{x}_1, \boldsymbol{x}_2), (t_1, t_2) \leftarrow$ partition$(\boldsymbol{x}, t)$ // Randomly partition inputs into two blocks
  // Apply Algorithm 1
  $(\boldsymbol{x}'_\infty, \tau_\infty) \leftarrow$ get_propagation_region ($\boldsymbol{x}_1$, t_1, attack = l_∞)
  $(\boldsymbol{x}'_2, \tau_2) \leftarrow$ get_propagation_region ($\boldsymbol{x}_2$, t_2, attack = l_2)
  $\mathcal{B}^{\tau_\infty}(\boldsymbol{x}'_\infty) \leftarrow \textsc{Box}(\boldsymbol{x}'_\infty, \tau_\infty)$ // Get box with midpoint $\boldsymbol{x}'_\infty, \boldsymbol{x}'_2$ and radius τ_∞, τ_2
  $\mathcal{B}^{\tau_2}(\boldsymbol{x}'_2) \leftarrow \textsc{Box}(\boldsymbol{x}'_2, \tau_2)$
  $\boldsymbol{u}_{y^\Delta_\infty} \leftarrow$ get_upper_bound$(f_\theta, \mathcal{B}^{\tau_\infty}(\boldsymbol{x}'_\infty))$ // Get upper bound $\boldsymbol{u}_{y^\Delta_\infty}, \boldsymbol{u}_{y^\Delta_2}$ on logit differences
  $\boldsymbol{u}_{y^\Delta_2} \leftarrow$ get_upper_bound$(f_\theta, \mathcal{B}^{\tau_2}(\boldsymbol{x}'_2))$ // based on IBP
  $\text{loss}_{l_\infty} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{u}_{y^\Delta_\infty}, t)$
  $\text{loss}_{l_2} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{u}_{y^\Delta_2}, t)$
  $\text{loss}_{\ell_1} \leftarrow \ell_1 \cdot$ get_ℓ_1_norm(f_θ)
  $\text{loss}_{tot} \leftarrow \text{loss}_{l_\infty} + \text{loss}_{l_2} + \text{loss}_{\ell_1}$
  $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \text{loss}_{tot}$ // Update model parameters θ
end for

Algorithm 5 CURE-Scratch/Finetune Training Epoch

Require: Neural network f_θ, training set $(\boldsymbol{X}, \boldsymbol{T})$, perturbation radius ϵ_2 and ϵ_∞, subselection ratio λ_∞ and λ_2, learning rate η, ℓ_1 regularization weight ℓ_1, KL loss balance factor η, mode $\in$ [Scratch, Finetune]

for $(\boldsymbol{x}, t) = (\boldsymbol{x}_0, t_0) \dots (\boldsymbol{x}_b, t_b)$ **do** // Sample batches $\sim (\boldsymbol{X}, \boldsymbol{T})$
  $(\boldsymbol{x}'_\infty, \tau_\infty) \leftarrow$ get_propagation_region (attack = l_∞) // Refer to Algorithm 1
  $(\boldsymbol{x}'_2, \tau_2) \leftarrow$ get_propagation_region (attack = l_2)
  $\mathcal{B}^{\tau_\infty}(\boldsymbol{x}'_\infty) \leftarrow \text{Box}(\boldsymbol{x}'_\infty, \tau_\infty)$ // Get box with midpoint $\boldsymbol{x}'_\infty, \boldsymbol{x}'_2$ and radius τ_∞, τ_2
  $\mathcal{B}^{\tau_2}(\boldsymbol{x}'_2) \leftarrow \text{Box}(\boldsymbol{x}'_2, \tau_2)$
  $\boldsymbol{u}_{y^\Delta_\infty} \leftarrow$ get_upper_bound$(f_\theta, \mathcal{B}^{\tau_\infty}(\boldsymbol{x}'_\infty))$ // Get upper bound $\boldsymbol{u}_{y^\Delta_\infty}, \boldsymbol{u}_{y^\Delta_2}$ on logit differences
  $\boldsymbol{u}_{y^\Delta_2} \leftarrow$ get_upper_bound$(f_\theta, \mathcal{B}^{\tau_2}(\boldsymbol{x}'_2))$ // based on IBP
  $\text{loss}_{l_\infty} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{u}_{y^\Delta_\infty}, t)$
  $\text{loss}_{l_2} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{u}_{y^\Delta_2}, t)$
  $\text{loss}_{Max} \leftarrow max(\text{loss}_{l_\infty}, \text{loss}_{l_2})$ // We select the largest $l_{p\in[2,\infty]}$ loss for each sample
  $\text{loss}_{\ell_1} \leftarrow \ell_1 \cdot$ get_ℓ_1_norm(f_θ)
  find correctly certified l_q subset γ using Definition 3.3
  $\text{loss}_{KL} \leftarrow KL(d_q[\gamma] \| d_r[\gamma])$ // Eq. 5
  $\text{loss}_{tot} \leftarrow \text{loss}_{Max} + \eta \cdot \text{loss}_{KL} + \text{loss}_{\ell_1}$
  $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \text{loss}_{tot}$ // Update model parameters θ
end for

Algorithm 6 GP: Connect CT with NT

Input: model f_θ, input images with distribution $\mathcal{D}$, training rounds R, β, natural training **NT** and certified training **CT** algorithms, perturbation radius ϵ_∞ and ϵ_2, subselection ratio λ_∞ and λ_2, learning rate η, ℓ_1 regularization weight ℓ_1.

for $r = 1, 2, ..., R$ **do**
  $f_n \leftarrow \textbf{NT}(f_\theta^{(r)}, \mathcal{D})$
  $f_c \leftarrow \textbf{CT}(f_\theta^{(r)}, \epsilon_\infty, \epsilon_2, \lambda_\infty, \lambda_2, \eta, \ell_1, \mathcal{D})$ // Can be single-norm or any CURE training
  compute $g_n \leftarrow f_n - f_\theta^{(r)}$, $g_c \leftarrow f_c - f_\theta^{(r)}$
  compute g_p using Eq. 10
  update $f_\theta^{(r+1)}$ using Eq. 11 with β and g_c
end for
Output: model f_θ.