CerDEQ: Certifiable Deep Equilibrium Model

Anonymous Authors[1]

Abstract

Recently, certifiable robust training methods via bound propagation have been proposed for training neural networks with certifiable robustness guarantees. However, no neural architectures with regular convolution and linear layers perform better in the certifiable training than the plain CNNs, since the output bounds for the deep explicit models increase quickly as their depth increases. And such a phenomenon significantly hinders certifiable training. Meanwhile, the Deep Equilibrium Models (DEQs) are more representative and robust due to their equivalent infinite depth and controllable global Lipschitz. But no work has been proposed to explore whether DEQ can show advantages in certified training. In this work, we aim to tackle the problem of DEQ's certified training. To obtain the output bound based on the bound propagation scheme in the implicit model, we first involve the adjoint DEQ for bound approximation. Furthermore, we also use the weight orthogonalization method and other tricks specified for DEQ to stabilize the certifiable training. With our approach, we can obtain the certifiable DEQ called CerDEQ. Our CerDEQ can achieve state-of-the-art performance compared with models using regular convolution and linear layers on ℓ_∞ tasks with $\epsilon = 8/255$: 64.72% certified error for CIFAR-10 and 94.45% certified error for Tiny ImageNet.

1. Introduction

Although deep neural networks (DNNs) have achieved great success in various areas, the discovery of the adversarial examples (Szegedy et al., 2014; Goodfellow et al., 2015; Carlini & Wagner, 2017; Madry et al., 2019; Kurakin et al., 2017) has raised concerns about the security of DNNs and hinders their development in safety-critical areas like autonomous driving and medical diagnosis. Many training algorithms (Madry et al., 2019; Wang et al., 2020; Wu et al., 2020) have been proposed to solve such a problem, which use adversarial examples for training to enhance the robustness. However, the robustness obtained by such methods doesn't have theoretical guarantees.

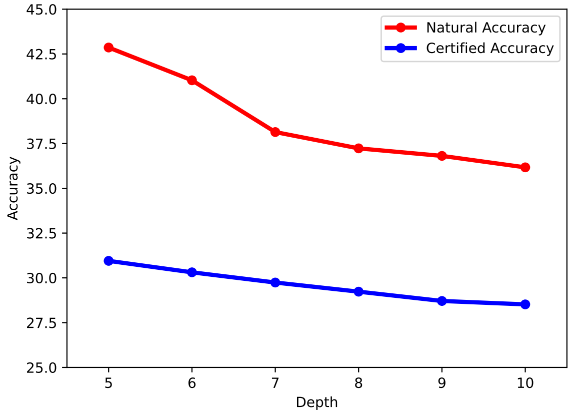


Figure 1. The certified error of CNNs with BN of different depth, the models are trained on CIFAR-10 and evaluated under $\epsilon = 8/255$.

Except for these researches, works (Wong & Kolter, 2018; Chen et al., 2021; Katz et al., 2017) have been proposed to theoretically evaluate their certified robustness by calculating the worst output for all possible input perturbations within the given region. Besides evaluation tasks, certified robust training methods have been proposed for CNNs by minimizing the certified robust loss obtained by calculating the upper bound for loss of the worst-case for given input perturbations. In order to find a tighter upper bound for the certified loss, works (Dvijotham et al., 2018; Gowal et al., 2018; Mirman et al., 2018b; Zhang et al., 2018; Shi et al., 2021) have been proposed for CNNs to obtain the output bounds of DNNs for efficient training. However, as the approximated output bounds overgrow with the increment of the layers, the result for certifiably training a deep model is unsatisfactory, as shown in Figure 1.

For this account, the well-designed CNN structures like WideResNet (Zagoruyko & Komodakis, 2016) and

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ResNeXt (Xie et al., 2017) cannot show their advantages against plain CNN models on these tasks as shown in Shi et al. (2021). Such a phenomenon implies that explicit DNNs models may not suit the certified tasks since these models need deep layers to be representative, while the depth will make the certified training harder.

Recently, DEQ (Bai et al., 2019) has been proposed as a potential alternative to classical DNNs. For a given sample $\mathbf{x}$, a DEQ layer uses the equilibrium state as output via the following fixed point equation:

$$\mathbf{z} = \sigma(\mathbf{W}\mathbf{z} + \mathbf{U}\mathbf{x} + \mathbf{b}), \tag{1}$$

where $\mathbf{z}$ is the equilibrium state of DEQ for input $\mathbf{x}$, we call $\mathbf{W}$ as fusion weights in the following and $\mathbf{U}, \mathbf{b}$ are learnable parameters with respect to the input $\mathbf{x}$ and σ is ReLU function. We call $\mathbf{U}\mathbf{x}$ the DEQ's extractor part in the following. After stacking (Gu et al., 2020a; Winston & Kolter, 2020a; Bai et al., 2020) a few DEQ layers, DEQ model can provide competitive results on machine learning tasks with constant memory usage. Instead of explicitly forwarding the input features layer by layer, DEQ's forward process reaches the output state by solving the fixed point iterations via the root-finding methods only if Eqn (1) can converge to an equilibrium point. Since DEQ's forward process is equivalent to forwarding the weight-tired neural layer for infinite times, we can also regard a DEQ layer as an weight-tired explicit module with infinite layers.

Since DEQ's generalization ability is not attributed to its explicit depth, we wonder whether it can perform better in certifiable training cases. However, no researches have been done to implement the certifiable training methods on DEQ and explore whether DEQ is more suitable for the certified robustness problem or not. In this paper, we manage to propose a way to certifiably train a DEQ model. The main contributions of our paper are listed below:

1. Firstly, we propose an approach to efficiently calculate the output bound for the implicit DNN models via an adjoint DEQ. Then we can implement certified training on DEQ.
2. Secondly, we propose an orthogonalization method for DEQ in certified training, ensuring the convergence of DEQ and enhancing the DEQ's performance in certified training.
3. Thirdly, we convert some essential tricks of the state-of-the-art Fast-IBP (Shi et al., 2021) training methods for explicit models to their DEQ version, like new initialization methods.

With our proposed certified training method, we can obtain a certifiable robust DEQ called CerDEQ. CerDEQ can achieve state-of-the-art performance on the certified tasks compared with other explicit models. Significantly, CerDEQ achieves 32.8% certified accuracy quickly with only 70 training epochs and 35.3% certified accuracy for 200 epochs on CIFAR-10 with $\epsilon = 8/255$, demonstrating that CerDEQ is more suitable for certified training and be more stable under the certifiable tasks. Furthermore, our CerDEQ can achieve 5.55% certified accuracy on Tiny ImageNet with $\epsilon = 8/255$, nearly 20% improvement compared with the explicit models.

2. Related Works

2.1. Methods for Certified Training

The widely used way to train a robust neural network can be viewed as solving the following min-max optimization problem:

$$\min_{\theta} \mathbb{E}\left[\max_{\delta\in\Delta(\epsilon)} L(f_\theta(\mathbf{x}+\delta, y))\right], \tag{2}$$

where f_θ stands for the neural architecture parameterized by θ, $\mathbf{x}$ denotes the data, y denotes the label. δ is the perturbation that constrained by ϵ. In this paper, we set the perturbation constraint as a ℓ_∞ ball with radius ϵ. The empirical adversarial training algorithms use optimization methods like projected gradient methods to obtain δ for the inner optimization problem and then use it to do the outer minimizing problem for training. However, such an empirical way can not guarantee that δ will converge to the inner problem's solution. In contrast, the certified training tries to compute upper bounds for the inner maximization problem, which can provably cover the worst-case perturbation.

Since the upper bound in the certified training needs to be calculated for each training iteration, many works (Raghunathan et al., 2020; Wong & Kolter, 2018; Mirman et al., 2018a; Dvijotham et al., 2018; Wang et al., 1811) are not suitable due to their high computation cost for large models. To obtain cheap and tight output bounds, Gowal et al. (2018) proposed a more efficient method called the interval bound propagation (IBP), which is widely used. In order to make the IBP bound tighter and quicker, CROWN-IBP and its variants (Zhang et al., 2018; Xu et al., 2020) are proposed with tighter relaxation bounds to improve the performance. Based on IBP and CROWN-IBP, methods (Balunovic & Vechev, 2019; Lyu et al., 2021; Shi et al., 2021) are proposed to further improve the performance by adding adversarial perturbations, proposing different warming up schedules or regularizers. These methods significantly improve the benchmark for certified training. However, as our following illustration, these methods are designed for explicit models and will encounter problems for implicit models. In this paper, we propose a certified training scheme for DEQ based on Shi et al. (2021)'s work (we call it FastIBP in the following), which can efficiently achieve the state-of-the-art certifiable robustness for explicit models.

Besides the above methods for the deterministic certified robustness, there are works using the randomization based methods like random smoothing for probabilistic certified defenses (Cohen et al., 2019; Li et al., 2018; Lecuyer et al., 2019; Salman et al., 2019). However, these methods need a lot of time on sampling during testing, and it is usually for ℓ_2 perturbations and can hardly be implemented on ℓ_∞ perturbations (Yang et al., 2020; Blum et al., 2020; Kumar et al., 2020).

2.2. Robustness for Deep Equilibrium Models

Deep Equilibrium Models (Bai et al., 2019; 2020; Winston & Kolter, 2020b) are new types of implicit models that perform like a neural network with infinite depth. It can achieve comparable performance with efficient memory cost. Furthermore, as illustrated in (Xie et al., 2021), DEQ also enjoys some advantages on models' interpretability.

Moreover, Pabbaraju et al. (2020) have shown that the Lipschitz constant for DEQ is controllable and more robust against some easy adversarial examples. Chen et al. (2021) and Müller et al. (2021) propose methods for evaluating DEQ's verifiable robustness and also demonstrate that DEQ enjoys advantages on the certifiable robustness when they are naturally trained compared with DNNs. However, these methods are not computation-friendly and cannot be used for certified training.

3. The Proposed Certified Training Method for DEQ Models

3.1. Bound Approximation with Adjoint DEQ

Like explicit models, we need to obtain the output bound for our DEQ layer by layer to get the final bounds $\overline{f_\theta(\mathbf{x}+\delta)}, \underline{f_\theta(\mathbf{x}+\delta)}$ for our model. Then we can obtain the upper bound for Eqn (2)'s inner part and use it for training. For this account, we need to find the output bound of a DEQ layer's output $\mathbf{z}$ with respect to changes of the input $\mathbf{x}$.

In order to obtain the output bound for DEQ, we need solve the following equations:

$$\begin{aligned} \overline{\mathbf{z}}_i^* &:= \max_{\|\delta\|_\infty \le \epsilon} \{\mathbf{e}_i^\top \mathbf{z}^* : \mathbf{z}^* = \sigma(\mathbf{W}\mathbf{z}^* + \mathbf{U}(\mathbf{x}+\delta) + \mathbf{b})\}, \\ \underline{\mathbf{z}}_i^* &:= \min_{\|\delta\|_\infty \le \epsilon} \{\mathbf{e}_i^\top \mathbf{z}^* : \mathbf{z}^* = \sigma(\mathbf{W}\mathbf{z}^* + \mathbf{U}(\mathbf{x}+\delta) + \mathbf{b})\}, \end{aligned}$$

where δ is the perturbation bounded by ϵ, $\mathbf{z}_i^*$ denotes the i-th element for the equilibrium state $\mathbf{z}^*$ and $\mathbf{e}_i$ is a unit vector. As illustrated in Chen et al. (2021), the above problems are non-convex and solving the problem using their proposed method will cost a lot of time.

However, we can use an adjoint DEQ network to obtain the outputs' upper bound and lower bound since DEQ can be unrolled into the following type:

$$\begin{aligned} \mathbf{z}^{(1)} &= \sigma(\mathbf{W}\mathbf{z}^{(0)} + \mathbf{U}\mathbf{x} + \mathbf{b}), \\ &\cdots\cdots \\ \mathbf{z}^{(k+1)} &= \sigma(\mathbf{W}\mathbf{z}^{(k)} + \mathbf{U}\mathbf{x} + \mathbf{b}). \end{aligned} \tag{3}$$

If we treat the whole structure as a weight-tired deep neural network, then we can obtain the bound for the model using the interval bound propagation method:

$$\begin{aligned} \overline{\mathbf{z}}^{(1)} &= \sigma(\mathbf{U}_+\overline{\mathbf{x}} + \mathbf{U}_-\underline{\mathbf{x}} + \mathbf{b}), \\ \underline{\mathbf{z}}^{(1)} &= \sigma(\mathbf{U}_-\overline{\mathbf{x}} + \mathbf{U}_+\underline{\mathbf{x}} + \mathbf{b}), \\ &\cdots\cdots \\ \overline{\mathbf{z}}^{(k+1)} &= \sigma(\mathbf{W}_+\overline{\mathbf{z}}^{(k+1)} + \mathbf{W}_-\underline{\mathbf{z}}^{(k+1)} + \mathbf{U}_+\overline{\mathbf{x}} + \mathbf{U}_-\underline{\mathbf{x}} + \mathbf{b}), \\ \underline{\mathbf{z}}^{(k+1)} &= \sigma(\mathbf{W}_-\overline{\mathbf{z}}^{(k+1)} + \mathbf{W}_+\underline{\mathbf{z}}^{(k+1)} + \mathbf{U}_+\overline{\mathbf{x}} + \mathbf{U}_-\underline{\mathbf{x}} + \mathbf{b}), \end{aligned} \tag{4}$$

where $(\overline{\mathbf{z}}_0, \underline{\mathbf{z}}_0)$ set to be $(0,0)$, $\mathbf{W}_+ = \max\{\mathbf{W}, 0\}$, $\mathbf{W}_- = \min\{\mathbf{W}, 0\}$, $\mathbf{U}_+, \mathbf{U}_-$ denotes the matrix $\mathbf{W}, \mathbf{U}$ with their negative and positive values truncated to 0.

If the sequences Eqn (4) can converge, we can rewrite the above fixed-point iterations as the following DEQ model,

$$\begin{pmatrix} \overline{\mathbf{z}}^* \\ \underline{\mathbf{z}}^* \end{pmatrix} = \sigma\left(\begin{pmatrix} \mathbf{W}_+ & \mathbf{W}_- \\ \mathbf{W}_- & \mathbf{W}_+ \end{pmatrix} \begin{pmatrix} \overline{\mathbf{z}}^* \\ \underline{\mathbf{z}}^* \end{pmatrix} + \begin{pmatrix} \mathbf{U}_+\overline{\mathbf{x}} + \mathbf{U}_-\underline{\mathbf{x}} + \mathbf{b} \\ \mathbf{U}_+\overline{\mathbf{x}} + \mathbf{U}_-\underline{\mathbf{x}} + \mathbf{b} \end{pmatrix} \right) \tag{5}$$

And the output of the above DEQ model (we called adjoint DEQ) can be used as the bound for the original DEQ's upper bound and lower bound with respect to the input's perturbation as demonstrated in the following proposition:

Proposition 3.1. *If the fusion weight matrix for the adjoint DEQ (5) satisfies the following condition:*

$$\left\| \begin{pmatrix} \mathbf{W}_+ & \mathbf{W}_- \\ \mathbf{W}_- & \mathbf{W}_+ \end{pmatrix} \right\|_2 < 1,$$

then its output $\overline{\mathbf{z}}^*, \underline{\mathbf{z}}^*$ are the upper bound and the lower bound of the original DEQ $(\mathbf{z}^* = \sigma(\mathbf{W}\mathbf{z}^* + \mathbf{U}\mathbf{x} + \mathbf{b}))$ with respect to the perturbations on input $\mathbf{x}$:

$$\begin{aligned} \underline{\mathbf{z}}^* &\le \min_{\underline{\mathbf{x}} \le \mathbf{x} \le \overline{\mathbf{x}}} \mathbf{z}(\mathbf{x})^*, \\ \overline{\mathbf{z}}^* &\ge \max_{\underline{\mathbf{x}} \le \mathbf{x} \le \overline{\mathbf{x}}} \mathbf{z}(\mathbf{x})^*, \end{aligned}$$

where we use $\mathbf{z}^*(\mathbf{x})$ to denote the output of DEQ with respect to the input $\mathbf{x}$.

For this account, we can do certified training for DEQ to enhance its robustness only if we can ensure the convergence of both DEQ and its adjoint DEQ model. Futhermore, since the equilibrium points for the contractive mappings are unique, we can use the root-finding algorithms for the adjoint DEQ instead of unrolling for the forward procedure.

3.2. Strict Weight Normalization is Required in Certified Training

The convergence of a DEQ layer can be ensured by constraining its weight matrix's spectral norm since a DEQ layer like Eqn (1) is contractive if $\|\mathbf{W}\|_2 < 1$. However, ensuring the convergence of a DEQ and its adjoint one simultaneously needs stricter constraints.

Proposition 3.2. *For any matrix* $\mathbf{W}$*, following property always holds:*

$$\left\| \begin{pmatrix} \mathbf{W}_+ & \mathbf{W}_- \\ \mathbf{W}_- & \mathbf{W}_+ \end{pmatrix} \right\|_2 = \max(\|\mathbf{W}\|_2, \||\mathbf{W}|\|_2).$$

where $|\mathbf{W}|$ is the matrix whose entries are the absolute values of $\mathbf{W}$.

For this account, not all DEQ with $\|\mathbf{W}\|_2 < 1$ can obtain the output bounds through the above method unless its adjoint DEQ is contractive. We need to make some constraints on $\begin{pmatrix} \mathbf{W}_+ & \mathbf{W}_- \\ \mathbf{W}_- & \mathbf{W}_+ \end{pmatrix}$ to make the whole structure contractive, so that we can use the adjoint DEQ model to obtain the output bound for the certified training. A straight-forward way to ensure the convergence of DEQ and its adjoint one at the same time is directly scaling the weight matrix with $\max(\|\mathbf{W}\|_2, \||\mathbf{W}|\|_2)$ as its scaling factor.

However, as illustrated in former works (Anil et al., 2019), strictly constraints model's Lipschitz will induce the degeneration of weights. As our latter experiments show, such though naive scaling will reduce the rank of DEQ's learnable weights $\mathbf{W}$ and hinder model's representation ability.

3.3. Weight Orthogonalization in DEQ

In order to alleviate the harmful effects stated above, we project our weights to their nearest orthogonal matrix in DEQ after each update. Then the singular values for our weights are encouraged to be the same to avoid too small ones, which may lead to the degeneration of the weight matrix. In this paper, we use Björck orthogonormalization (Björck & Bowie, 1971) after each training iteration for our DEQ.

Björck orthogonalization is a widely used iterative method to approximate the nearest orthogonal approximation of a given matrix. We do an order-2 Björck orthogonalization in experiments iteratively to obtain the orthogonal approximation for the fusion weight $\mathbf{W}$ of our CerDEQ. The iteration can be formulated as follows:

$$\mathbf{A}_{k+1} = \frac{15}{8}\mathbf{A}_k - \frac{5}{4}\mathbf{A}_k(\mathbf{A}_k^\top \mathbf{A}_k) \quad (6)$$

$$+ \frac{3}{8}\mathbf{A}_k(\mathbf{A}_k^\top \mathbf{A}_k)(\mathbf{A}_k^\top \mathbf{A}_k) \quad (7)$$

where $\mathbf{A}_0 = \mathbf{W}^\top$ and $\mathbf{A}_K^\top$ is the orthogonal approximation for our weight $\mathbf{W}$ with K iterations. The sufficient conditions for this method's convergence are listed in the following Lemma:

Lemma 3.3. *(Björck & Bowie, 1971) If* $\mathbf{A}_{k+1}$ *is calculated by Eqn (7) and the following condition is satisfied,*

$$\|\mathbf{I} - \mathbf{W}\mathbf{W}^\top\|_2 < 1,$$

then $lim_{k\to\infty}\mathbf{A}_k = \mathbf{P}$ *and* $\mathbf{P}$ *is the solution for the following problem:*

$$\min_{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}_m} \|\mathbf{Q} - \mathbf{W}^\top\|_F,$$

where $\mathbf{Q}, \mathbf{W}^\top \in \mathbf{R}^{n\times m}$ *with* $m \le n$*. In other words,* $\mathbf{P}$ *is the nearest orthogonal matrix for given matrix* $\mathbf{W}^\top$*.*

For this account, we need to scale the weight to ensure that all its singular values are less than 1 before the orthogonalization. As for the scaling factor, we use the following matrix norm inequalities to approximate:

$$\sigma_{\max} \le \sqrt{mn}\|\mathbf{W}\|_{\max},$$

where $\sigma_{\max}$ denotes the largest singular value of matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$ while $\|\mathbf{W}\|_{\max}$ denotes its largest absolute value. However, only orthogonalizing the weight matrix $\mathbf{W}$ cannot ensure the convergence of the adjoint DEQ. Therefore we scale the orthogonal weights for the second time to ensure the convergence for the certified training due to the following proposition.

Proposition 3.4. *For an matrix* $\mathbf{O} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{O}\mathbf{O}^\top = \mathbf{I}_m$ *with* $m \le n$*, following property always holds:*

$$1 \le \left\| \begin{pmatrix} \mathbf{O}_+ & \mathbf{O}_- \\ \mathbf{O}_- & \mathbf{O}_+ \end{pmatrix} \right\|_2 \le \sqrt{m} \le \sqrt{n}.$$

In practice, m, n denote output channel number and input channel number multiplying the convolution kernel size for weight $\mathbf{W}$. Thereby, we have $m \le n$. For convenience, we scale the weight matrix by $\sqrt{n}$ to ensure adjoint DEQ's convergence and Eqn (8) established in the latter section. Because of the weight is orthogonalized, our CerDEQ can still perform well after the strict scaling. In the following, we experimentally indicate the superiority of our approach over other weight normalization methods in DEQs for natural tasks (Bai et al., 2020; Gu et al., 2020b).

3.4. Initialization for CerDEQ's Extractor Part

Inspired by the IBP initialization method proposed by Shi et al. (2021) for certified training, we are going to propose a new initialization to stabilize DEQ's certified training. Since the fusion weight matrix $\mathbf{W}$ for DEQ is orthogonalized, we only need to initialize weight $\mathbf{U}$ for DEQ's extractor part.

Since DEQ needs to reach the equilibrium for Eqn (1), the IBP initialization of $\mathbf{U}$ for explicit models cannot ensure the expectation of DEQ's output and input bound to be almost the same. Therefore, we need to calculate the new initialization schemes for DEQ. With $\Delta_{in}, \Delta_{out}$ denoting the distribution of DEQ's input and output elements, we have the following proposition with the same setting as the IBP initialization.

Proposition 3.5. *If we independently initialize each element of $\mathbf{U}_i$ following the normal distribution $\mathcal{N}(0, \sigma^2)$ and $\mathbf{W}$ is row-orthogonal with $\|\mathbf{W}\|_2 \leq \frac{1}{\sqrt{n}}$, then we bound the difference gain of the DEQ layers' output and input bound by the following equation:*

$$\frac{\mathbb{E}\left[\Delta_{out}\right]}{\mathbb{E}\left[\Delta_{in}\right]} \leq \frac{n_u \mathbb{E}(|\mathbf{U}|)}{2 - \sqrt{n}\|\mathbf{W}\|_2} \leq n_u \mathbb{E}(|\mathbf{U}|), \qquad (8)$$

with $\mathbf{U} \in \mathbb{R}^{m \times n_u}$ and $\mathbf{W} \in \mathbb{R}^{m \times n}$.

When $\mathbf{U}$'s elements are initialized following the normal distribution, we can get $\mathbb{E}(|\mathbf{U}|) = \sqrt{\frac{2}{\pi}}\sigma$. For this account, we use normal distribution with zero mean and $\sigma = \frac{\sqrt{\pi}}{\sqrt{2}n_u}$ for $\mathbf{U}$'s initialization to make the difference of gain $\frac{\mathbb{E}[\overline{\mathbf{z}}^* - \underline{\mathbf{z}}^*]}{\mathbb{E}[\overline{\mathbf{x}} \ - \underline{\mathbf{x}} \ ]}$ less than 1.

3.5. Batch Normalization Layer for DEQ's Training

As illustrated in Fast-IBP, adding Batch Normalization layers can reduce the inactive neurons of DEQ during the certified training and is essential for performance. Thereby, we also use Batch Normalization for DEQ's extractor parts for this account. Nevertheless, for the equilibrium state $\mathbf{z}^*$, we cannot use Batch Normalization because we cannot obtain the inner running mean and variance during the bound propagation. Because BN layers can be regarded as a linear layer with bias during the evaluation period, we use biased convolution or linear layer with learnable channel scaling factors γ for DEQ to imitate the original BN layer as the replacement for BN layers. During training, we bounded the learnable scaling factors γ in $(0, 1)$ to ensure convergence.

3.6. Training Objectives for CerDEQ

Like other certified training, the base training objective for our CerDEQ is:

$$\mathcal{L}_c = \overline{L}_\epsilon(f_\theta, \mathbf{x}, y), \text{ where } \overline{L}_\epsilon \geq \max_{\|\delta\| \leq \epsilon} L(f_\theta(\mathbf{x} + \delta), y, \epsilon). \quad (9)$$

Like Gowal et al. (2018), we also use the loss for natural samples $\mathcal{L}_n$ in training. And we add the bound tightness regularizer $\mathcal{L}_t$ and relu regularizers $\mathcal{L}_r$ proposed by Shi et al. (2021) with their settings. These regularizers aim to reduce the inactive neurons and diminish the layer's output bound growth. Thereby, they will stabilize the certified training process for both explicit models and DEQs.

The final training objective for our DEQ's certified training can be formulated as:

$$\mathcal{L} = \frac{1}{1 + \lambda_n}(\mathcal{L}_c + \lambda_n \mathcal{L}_n) + \lambda(\mathcal{L}_t + \mathcal{L}_r),$$

where λ_n is a balancing parameter for the certified loss and clean loss. $\lambda = \lambda_0(1 - \epsilon_{now}/\epsilon)$ is a balancing parameter only exists during the warmup period, during which the target budget ϵ_{now} gradually increases from 0 to ϵ for the bound calculation. Following the above strategies, we can efficiently train a DEQ with satisfactory certifiable robustness, which we called **CerDEQ** in the following.

4. Experiments

In this section, we try to demonstrate the effectiveness of our proposed method for training a certified deep equilibrium model and the advantages of CerDEQ via experiments on CIFAR-10 and Tiny ImageNet.

4.1. Settings

We adopt two datasets, CIFAR-10 and Tiny ImageNet, to demonstrate the effectiveness of our method. We mainly consider three deep models with regular convolution and linear layers for comparison: a 7-layer feedforward convolution network with BN (CNN-7-BN), Wide-ResNet and ResNeXt, which are the widely used models for the certifiable tasks. We stack three DEQ layers for our CerDEQ on CIFAR-10, consisting of two convolutional DEQ layers for downsampling and one linear DEQ layer. As for Tiny ImageNet, whose input size is larger than CIFAR-10, we add a downsampling convolution DEQ layer for our CerDEQ model. We change the channel number of our CerDEQ in order to ensure that it contains the same learnable parameters as CNN-7 for fairness. As for the perturbation radii, we set $\epsilon = 8/255$ for CIFAR-10 and $\epsilon = 8/255, 1/255$ for Tiny ImageNet.

Like other certified training schemes, we gradually increase ϵ_{now} from 0 to ϵ with the same smoothed schedules as widely used in other works (Xu et al., 2020) as a warmup for epochs. And then, we use the target ϵ for the rest of the training. As for the iteration number for the orthogonalization, we set the iteration number for Björck orthogonalization to be 5 in all experiments.

We use the Anderson Acceleration Algorithm (Walker & Ni, 2011) and Phantom Gradient Method (Geng et al., 2022) for DEQ's forward and backward propagations. As for training, we adopt Adam optimizer (Kingma & Ba, 2017) with a learning rate starting from 0.0005. All the experiments are run on the PyTorch platform with GTX1080Ti. Other hyper-parameters for the experiments can be found in Appendix E.

4.2. Certified Robustness Compared with Other Models

Firstly, we are going to show that the DEQ can quickly achieve better certified robustness via experiments. We train our CerDEQ for 70 epochs in total on CIFAR-10 with 2 epochs natural training. The results are listed in Table 1.

Model	Standard Error	Certified Error
CNN-7	$56.64 \pm 0.48\%$	$68.81 \pm 0.24\%$
WideResNet	$56.74 \pm 0.40\%$	$68.79 \pm 0.29\%$
ResNeXt	$59.33 \pm 0.40\%$	$70.62 \pm 0.59\%$
CerDEQ (ours)	$\mathbf{53.43 \pm 0.33\%}$	$\mathbf{67.21 \pm 0.12\%}$

Table 1. The comparison of different models' certified robustness under $\epsilon = 8/255$ on CIFAR-10. The results are obtained for 5 trials. The explicit models are trained by the Fast-IBP methods for the same epochs.

From the results above, one can see that our training method and construction for CerDEQ is effective since we can achieve around 3% higher natural accuracy and almost 2% higher certified accuracy on CIFAR-10. The superiorities are attributed to DEQ's better generalization ability. And CerDEQ's controllable bounds compared with other explicit networks shown in Figure 3 also stabilize the certified training and make it enable to achieve satisfactory performance quickly. These characteristics make its certified training much easier than the state-of-the-art neural networks.

Apart from CIFAR-10, we also evaluate the robustness of CerDEQ on Tiny ImageNet for 80 epochs training in total with 2 epochs natural training in the beginning. The results are listed in Table 2.

Model	Standard Error	Certified Error
CNN-7	74.29%	82.36%
WideResNet	74.59%	82.75%
ResNeXt	78.91%	85.78%
ℓ_∞-dist Net	78.18%	83.69%
CerDEQ	$\mathbf{73.51\%}$	$\mathbf{82.16\%}$

Table 2. The comparison of different model's standard error and certified error under $\epsilon = 1/255$ on Tiny ImageNet. The explicit models are trained by the Fast-IBP method and the result for ℓ_∞-dist net is copied directly from their paper.

One can see that our CerDEQ consistently achieves better performance compared with other models. ℓ_∞-dist Net is a network with ℓ_∞ neurons" proposed by Zhang et al. (2021). The results above also imply that the explicit models are not suitable for the certified training since WideResNet and ResNeXt can not achieve better results than CNN, which is the opposite of their natural and adversarial training results.

4.3. Training with Longer Epochs or Larger Radius

In addition to the short training schedule, we also finish the experiments on CIFAR-10 with 200 epochs to find out whether the certified DEQ can obtain state-of-the-art performance. The results are listed in Table 3.

Model	Standard Error	Certified Error
CNN-7-BN	$51.72 \pm 0.40\%$	$65.58 \pm 0.24\%$
WideResNet	$51.95 \pm 0.32\%$	$65.91 \pm 0.14\%$
ResNeXt	$53.68 \pm 0.33\%$	$66.91 \pm 0.40\%$
CerDEQ (ours)	$\mathbf{50.34 \pm 0.33\%}$	$\mathbf{64.98 \pm 0.26\%}$
CerDEQ (best)	$\mathbf{49.97\%}$	$\mathbf{64.72\%}$

Table 3. The comparison of different model's certified robustness under $\epsilon = 8/255$ on CIFAR-10. The results are obtained for 5 trivials. The explicit models are trained by the Fast-IBP methods.

From the table, one can see that our CerDEQ can consistently show state-of-the-art performance. Trained by our approach, our CerDEQ offers over 1% higher standard accuracy with 0.5% higher certified accuracy for 5 trials with longer training epochs. The best trial for our CerDEQ even achieves 64.72% certified accuracy with only 200 epochs training. Furthermore, we also notice that WideResNet and ResNeXt still perform worse in the certified training scenario compared with plain CNNs. The results also demonstrate the superiority of implicit models as they can consistently perform better in different scenarios.

Furthermore, we also finish the experiments on Tiny ImageNet with $\epsilon = 8/255$. It is a challenging task even for the empirical adversarial training methods. We train our certified DEQ with $\epsilon = 8/255$ for 80 epochs and list the results in Table 4.

Model	Standard Error	Certified Error
CNN-7 (Crown-IBP)	90.76%	95.98%
CNN-7 (Fast-IBP)	89.69%	95.44%
ℓ_∞-dist Net	88.99%	$\mathbf{94.22\%}$
CerDEQ (ours)	$\mathbf{87.98\%}$	94.45%

Table 4. The comparison of the best results for different model's certified robustness under $\epsilon = 8/255$ on Tiny ImageNet.

The results for other models in Table 4 are directly copied from Anonymous (2022). They finish the experiments for their ℓ_∞-dist Net and CNN-7 following Crown-IBP and Fast-IBP's repo with larger ϵ. From the figures, one can see that our CerDEQ can achieve almost 20% improvement no matter under the natural or certified evaluation scenario or not compared with the explicit CNN model. Compared with the ℓ_∞-dist Net, which uses "ℓ_∞ distance neurons" instead of traditional convolution or linear layers, our CerDEQ can achieve the comparable certified robustness with a signicantly better natural performance. But we note that ℓ_∞-dist Net needs thousands of epochs to converge due to its neurons, while our CerDEQ only needs 80 epochs for the Tiny ImageNet task. From the above experiments, we can conclude that our CerDEQ enjoys advantages on the certifiable robustness tasks.

4.4. Ablation Studies on Our Weigth Orthogonalization

Firstly, we conduct experiments to evaluate whether our proposed weight orthogonalization method for DEQ can genuinely improve its performance on certified training. We finish the certified training for DEQs with weight normalization and spectral normalization to ensure their convergence. These methods are widely used for DEQ (Bai et al., 2020; Gu et al., 2020b) to ensure the convergence in natural tasks. For fairness, we use the same approach and hyperparameters as our CerDEQ for their certified training. The results are listed in Table 5.

Model	Standard Error	Certified Error
DEQ+WN	$56.34 \pm 0.32\%$	$69.84 \pm 0.13\%$
DEQ+SN	$57.43 \pm 0.41\%$	$68.66 \pm 0.15\%$
CerDEQ	$\mathbf{53.43} \pm \mathbf{0.33\%}$	$\mathbf{67.21} \pm \mathbf{0.12\%}$

Table 5. The comparison of DEQ's certified robustness trained by different methods under $\epsilon = 8/255$ on CIFAR-10. "WN" and "SN" here denote the weight normalization and spectral normalization methods to ensure the convergence of DEQ and its adjoint DEQ.

From Table 5, one can see that DEQs with weight normalization or spectral normalization perform worse in the certified training than explicit models listed in Table 1. However, our CerDEQ enjoys over 3% higher clean accuracy and over 2% higher certified accuracy compared with DEQs with other normalization methods under the certified evaluations by utilizing our weight orthogonalization method. Such results demonstrate that our weight orthogonalization method can effectively boost the performance of DEQ under natural and certified robustness cases.

In order to further explore the reason for the improvements, we draw the singular values of the weights for the final linear DEQ layer with spectral normalization or with our orthogonalization. The curves are shown in Figure 2. From the figure, one can notice that DEQs with spectral normalization will lead the weights to be "inactive" (the relative singular value larger smaller than 10^{-4}) and the rank of weights become lower after training than DEQ with our orthogonalization. Thereby, DEQs with spectral normalization will perform as a model with fewer channels. For this reason, vanilla DEQs with spectral or weight normalization perform worse in the certified training scenario. And the curve for DEQ with our orthogonalization also indicates that our method can effectively alleviate such a phenomenon.

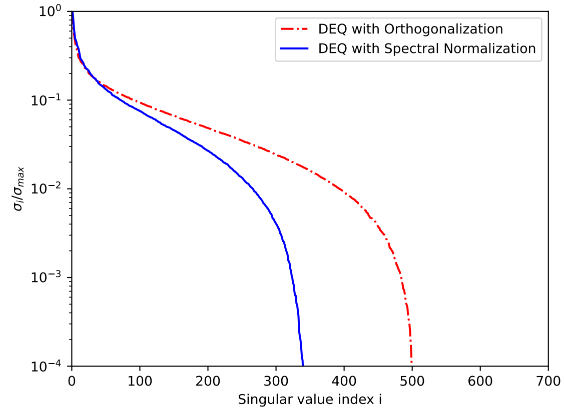


Figure 2. The relative singular value $\left(\frac{\sigma_i}{\sigma_{\max}}\right)$ distribution of the certified trained DEQ with Orthogonalization or Spectral Normalization to ensure their convergence.

4.5. The Relationship between Bound and Model Depth

In this section, we will experimentally explore why implicit models can consistently preserve their generalization abilities in certified training tasks while other state-of-the-art robust explicit models cannot. In order to explore that, we draw the increment curve for final layer's output bounds' norm divided by the norm of input bounds before classification with respect to the increments of the models' depth (or iteration). The results are in Figure 3.

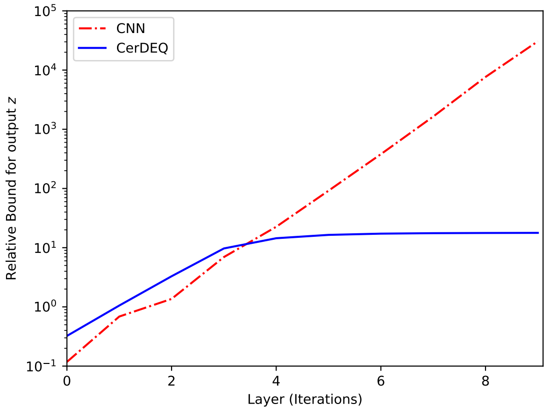


Figure 3. The relative output bound increment for plain CNN network and DEQ with respect to its depth (iteration).

From the above figures, one can see that the output bounds of the explicit models explode with the increment of the model's depth, but the output bounds for CerDEQ will converge as their iteration increases. Explicit models need more layers to ensure their generalization ability, while larger output bounds caused by depth will increase the difficulty for the certified training. Therefore, explicit models' generalization ability and hardness for their certified training are trade-offs. And this reason leads ResNeXt and WideResNet to perform worse than CNN-7 in the above experiments. Fortunately, CerDEQ does not encounter such a trade-off and can show better performance under different evaluations trained by our methods. This phenomenon also indicates the importance of studies for DEQs or other implicit models.

4.6. Ablation Studies on Explicit Models with Orthogonal Normalization

In addition to the explorations on the orthogonalization effect for the DEQ's training, we also finish experiments for CNN-7 with and without our orthogonalization for 200 epochs training to further explore the influence of weight orthogonalization, listed in Table 6. The experiment settings for CNN is the same as our CerDEQ.

Model	Stand-Err	Cert-Err	PGD-Err
CNN	52.73%	66.75%	63.7%
Orth-CNN	69.59%	71.66%	66.2%
CerDEQ	$\mathbf{49.97\%}$	$\mathbf{64.72\%}$	$\mathbf{62.1\%}$

Table 6. The comparison of CNN, Orth-CNN (trained with orthogonalization) and CerDEQ evaluated in natural cases (Stand-Err), certifiable cases (Cert-Err) and adversarial cases (PGD-Err) based on PGD-20 attack with $\epsilon = 8/255$ on CIFAR-10.

The table shows that using orthogonalization to project weights in explicit models will not benefit the performance or hinder its certified training. The phenomenon is reasonable since explicit models do not need a strict Lipshictz constant to ensure convergence, and constraints on the weights will influence the network's generalization ability. From the experiment, we can get two conclusions. First, the CerDEQ structure is superior to the explicit models on the certified tasks because it can perform better even with strict constraints on weights. Secondly, the orthogonalization only alleviates the negative impact of the strict constraints on the weights instead of completely solving the problem since they are harmful in CNNs. We leave the further modifications on CerDEQ for certified tasks as our future work.

4.7. Ablation Studies on Our Initialization Methods

In this section, we are going to validate whether our initialization problem can help the output bound be tighter at the start of training through experiments. Therefore, we compare the first five epochs of CerDEQ's training on CIFAR-10 with our initialization and IBP initialization proposed by Fast-IBP. We draw figures for the relative bound increment $\frac{\Delta_{final}}{\Delta_{input}}$ in the end of each epoch shown in Figure 4.

The figure indicates that our initialization can make CerDEQ's final output bound tighter during the early training process, which will help the CerDEQ's early training stage.

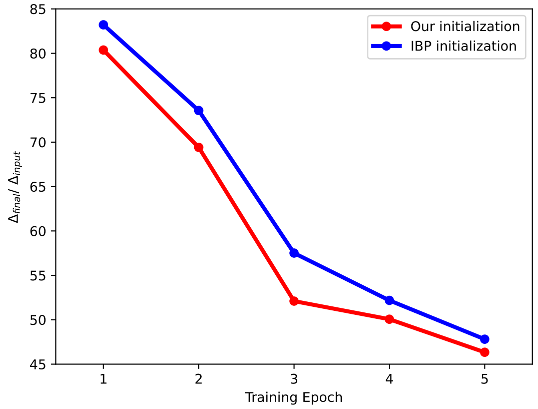


Figure 4. The output bound after the first five training epochs on CIFAR-10 for CerDEQ initialized by different methods. Δ_{input} denotes the bound on the input image and Δ_{final} stands for final output bound for our CerDEQ before the classification layer.

4.8. The Computational Cost for CerDEQ's Training

In this section, we list the longest training time of one epoch during the certified training for WideResNet, ResNeXt and CerDEQ in Table 7.

Model	Training Method	Time
WideResNet	Fast-IBP	$450s$
	Crown-IBP	$600s$
ResNeXt	Fast-IBP	$650s$
	Crown-IBP	$900s$
CerDEQ	Ours	$877s$

Table 7. The longest time of one epoch during certified training on Tiny ImageNet for each model on GTX-1080Ti.

The table shows that the computational complexities are comparable against the widely used certified training methods for explicit models. The additional computation time is due to the following. Firstly, the implicit models need to implement the root-finding algorithms to obtain the equilibrium. Secondly, we use the iterative Björck orthogonalization to make the weights orthogonal, which will consume a lot of time. We leave the acceleration as future work.

5. Conclusion

In this paper, we propose a complete approach for DEQ's certified training for the first time. Trained by our approach, CerDEQ can achieve state-of-the-art performance against other explicit models in the certified tasks. Our work also demonstrates that DEQ is more suitable for the certified training than explicit models, since its output bound will not explode like other explicit models. The superiority of our model also implies that analyzing implicit models for certified training is a promising way for the certifiable tasks.

References

Anil, C., Lucas, J., and Grosse, R. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pp. 291–301. PMLR, 2019.

Anonymous. Boosting the certified robustness of l-infinity distance nets. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Q76Y7wkiji. under review.

Bai, S., Kolter, J. Z., and Koltun, V. Deep equilibrium models. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 688–699, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/01386bd6d8e091c2ab4c7c7de644d37b-Abstract.html.

Bai, S., Koltun, V., and Kolter, J. Z. Multiscale deep equilibrium models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3812f9a59b634c2a9c574610eaba5bed-Abstract.html.

Balunovic, M. and Vechev, M. Adversarial training and provable defenses: Bridging the gap. In *International Conference on Learning Representations*, 2019.

Björck, Å. and Bowie, C. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

Blum, A., Dick, T., Manoj, N., and Zhang, H. Random smoothing might be unable to certify linf robustness for high-dimensional images. *J. Mach. Learn. Res.*, 21:211–1, 2020.

Carlini, N. and Wagner, D. A. Adversarial examples are not easily detected: Bypassing ten detection methods. *CoRR*, abs/1705.07263, 2017. URL http://arxiv.org/abs/1705.07263.

Chen, T., Lasserre, J.-B., Magron, V., and Pauwels, E. Semialgebraic representation of monotone deep equilibrium models and applications to certification. *arXiv preprint arXiv:2106.01453*, 2021.

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

Dvijotham, K., Gowal, S., Stanforth, R., Arandjelovic, R., O'Donoghue, B., Uesato, J., and Kohli, P. Training verified learners with learned verifiers, 2018.

Geng, Z., Zhang, X.-Y., Bai, S., Wang, Y., and Lin, Z. On training implicit models, 2022.

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

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T. A., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018. URL http://arxiv.org/abs/1810.12715.

Gu, F., Chang, H., Zhu, W., Sojoudi, S., and Ghaoui, L. E. Implicit graph neural networks. *arXiv preprint arXiv:2009.06211*, 2020a.

Gu, F., Chang, H., Zhu, W., Sojoudi, S., and Ghaoui, L. E. Implicit graph neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/8b5c8441a8ff8e151b191c53c1842a38-Abstract.html.

Katz, G., Barrett, C., Dill, D., Julian, K., and Kochenderfer, M. Reluplex: An efficient smt solver for verifying deep neural networks, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017.

Kumar, A., Levine, A., Goldstein, T., and Feizi, S. Curse of dimensionality on randomized smoothing for certifiable robustness. In *International Conference on Machine Learning*, pp. 5458–5467. PMLR, 2020.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HJGU3Rodl.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE, 2019.

Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. *arXiv preprint arXiv:1809.03113*, 2018.

Lyu, Z., Guo, M., Wu, T., Xu, G., Zhang, K., and Lin, D. Towards evaluating and training verifiably robust neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4308–4317, 2021.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks, 2019.

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

Mirman, M., Gehr, T., and Vechev, M. T. Differentiable abstract interpretation for provably robust neural networks. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3575–3583. PMLR, 2018b. URL http://proceedings.mlr.press/v80/mirman18b.html.

Müller, M. N., Staab, R., Fischer, M., and Vechev, M. Effective certification of monotone deep equilibrium models, 2021.

Pabbaraju, C., Winston, E., and Kolter, J. Z. Estimating lipschitz constants of monotone deep equilibrium models. In *International Conference on Learning Representations*, 2020.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples, 2020.

Salman, H., Yang, G., Li, J., Zhang, P., Zhang, H., Razenshteyn, I., and Bubeck, S. Provably robust deep learning via adversarially trained smoothed classifiers. *arXiv preprint arXiv:1906.04584*, 2019.

Shi, Z., Wang, Y., Zhang, H., Yi, J., and Hsieh, C.-J. Fast certified robust training with short warmup. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Walker, H. F. and Ni, P. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

Wang, S., Chen, Y., Abdou, A., and Jana, S. Mixtrain: scalable training of formally robust neural networks. corr abs/1811.02625 (2018). *arxiv. org/abs*, 1811.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rklOg6EFwS.

Winston, E. and Kolter, J. Z. Monotone operator equilibrium networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a. URL https://proceedings.neurips.cc/paper/2020/hash/798d1c2813cbdf8bcdb388db0e32d496-Abstract.html.

Winston, E. and Kolter, J. Z. Monotone operator equilibrium networks. *arXiv preprint arXiv:2006.08591*, 2020b.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope, 2018.

Wu, D., Xia, S., and Wang, Y. Adversarial weight perturbation helps robust generalization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1ef91c212e30e14bf125e9374262401f-Abstract.html.

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5987–5995. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.634. URL https://doi.org/10.1109/CVPR.2017.634.

Xie, X., Wang, Q., Ling, Z., Li, X., Wang, Y., Liu, G., and Lin, Z. Optimization induced equilibrium networks. *arXiv preprint arXiv:2105.13228*, 2021.

Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K.-W., Huang, M., Kailkhura, B., Lin, X., and Hsieh, C.-J. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33, 2020.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I., and Li, J. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pp. 10693–10705. PMLR, 2020.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL http://arxiv.org/abs/1605.07146.

Zhang, B., Cai, T., Lu, Z., He, D., and Wang, L. Towards certifying l infinity) robustness using neural networks with l infinity-dist neurons. *CoRR*, abs/2102.05363, 2021. URL https://arxiv.org/abs/2102.05363.

Zhang, H., Weng, T., Chen, P., Hsieh, C., and Daniel, L. Efficient neural network robustness certification with general activation functions. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 4944–4953, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/d04863f100d59b3eb688a11f95b0ae60-Abstract.html.

A. Proof for Proposition 3.1

We list the proof for Proposition 3.1 as follows:

Proof. When $\left\| \begin{pmatrix} \mathbf{W}_+ & \mathbf{W}_- \\ \mathbf{W}_- & \mathbf{W}_+ \end{pmatrix} \right\|_2 < 1$, DEQ layer $f(\overline{\mathbf{z}}, \underline{\mathbf{z}})$ with formulation:

$$f(\overline{\mathbf{z}}, \underline{\mathbf{z}}) = \sigma\left(\begin{pmatrix} \mathbf{W}_+ & \mathbf{W}_- \\ \mathbf{W}_- & \mathbf{W}_+ \end{pmatrix} \begin{pmatrix} \overline{\mathbf{z}} \\ \underline{\mathbf{z}} \end{pmatrix} + \begin{pmatrix} \overline{\mathbf{h}} \\ \underline{\mathbf{h}} \end{pmatrix}\right) \tag{10}$$

with

$$\begin{aligned} \overline{\mathbf{h}} &= \mathbf{U}^{+}\overline{\mathbf{x}} + \mathbf{U}^{-}\underline{\mathbf{x}} + \mathbf{b}, \\ \underline{\mathbf{h}} &= \mathbf{U}^{-}\overline{\mathbf{x}} + \mathbf{U}^{+}\underline{\mathbf{x}} + \mathbf{b}, \end{aligned}$$

is a contractive mapping. Therefore, the fixed point $(\overline{\mathbf{z}}, \underline{\mathbf{z}}) = f(\overline{\mathbf{z}}, \underline{\mathbf{z}})$ is unique. Then we discuss the propagation from the view of the fixed point iteration:

$$\begin{pmatrix} \overline{\mathbf{z}}^{(k+1)} \\ \underline{\mathbf{z}}^{(k+1)} \end{pmatrix} = \sigma\left(\begin{pmatrix} \mathbf{W}_+ & \mathbf{W}_- \\ \mathbf{W}_- & \mathbf{W}_+ \end{pmatrix} \begin{pmatrix} \overline{\mathbf{z}}^{(k+1)} \\ \underline{\mathbf{z}}^{(k+1)} \end{pmatrix} + \begin{pmatrix} \overline{\mathbf{h}} \\ \underline{\mathbf{h}} \end{pmatrix}\right) \tag{11}$$

with $\lim_{k\to\infty} \overline{\mathbf{z}}^{(k)} = \overline{\mathbf{z}}^*$ and $\lim_{k\to\infty} \underline{\mathbf{z}}^{(k)} = \underline{\mathbf{z}}^*$. Then we need to prove that for every $\mathbf{h}$ chosen from $[\underline{\mathbf{h}}, \overline{\mathbf{h}}]$, since from IBP's proof $\underline{\mathbf{h}} \leq \mathbf{h}(\mathbf{x}) \leq \overline{\mathbf{h}}$ holds for every $\mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ The DEQ's output $\mathbf{z}^* = \sigma(\mathbf{W}\mathbf{z}^* + \mathbf{h})$ (we denote $\mathbf{z}^*(\mathbf{h})$ at $\mathbf{h}$ as $\mathbf{h}$) satisfies the following inequality:

$$\underline{\mathbf{z}}^{(k)} \leq \mathbf{z}^*(\mathbf{h}) \leq \overline{\mathbf{z}}^{(k)} \tag{12}$$

Start from $k = 1$ and $\underline{\mathbf{z}}^{(0)} = \overline{\mathbf{z}}^{(0)} = \mathbf{z}^{(0)} = 0$, then we can get:

$$\underline{\mathbf{z}}^{(1)} = \sigma(\underline{\mathbf{h}}) \leq \mathbf{z}^{(1)}(\mathbf{h}) = \sigma(\mathbf{h}) \leq \sigma(\overline{\mathbf{h}}) = \overline{\mathbf{z}}^{(1)} \tag{13}$$

since $\underline{\mathbf{h}} \leq \mathbf{h} \leq \overline{\mathbf{h}}$. And Since $\underline{\mathbf{z}}^{(1)} \leq \mathbf{z}^{(1)} \leq \overline{\mathbf{z}}^{(1)}$, we can get the following equations:

$$\begin{aligned} \mathbf{W}_-\overline{\mathbf{z}}^{(1)} \leq \mathbf{W}_-\mathbf{z}^{(1)}(\mathbf{h}) \leq \mathbf{W}_-\underline{\mathbf{z}}^{(1)} \quad &(14) \\ \mathbf{W}_+\underline{\mathbf{z}}^{(1)} \leq \mathbf{W}_+\mathbf{z}^{(1)}(\mathbf{h}) \leq \mathbf{W}_+\overline{\mathbf{z}}^{(1)} \quad &(15) \\ &(16) \end{aligned}$$

Then adding Eqn (14), Eqn (15) and $\underline{\mathbf{h}} \leq \mathbf{h} \leq \overline{\mathbf{h}}$, and use the ReLU's monotonicity we can get the inequality for $k = 2$:

$$\sigma(\mathbf{W}_-\overline{\mathbf{z}}^{(1)} + \mathbf{W}_+\underline{\mathbf{z}}^{(1)} + \underline{\mathbf{h}}) \leq \sigma(\mathbf{W}\mathbf{z} + \mathbf{h}) \leq \sigma(\mathbf{W}_-\underline{\mathbf{z}}^{(1)} + \mathbf{W}_+\overline{\mathbf{z}}^{(1)} + \overline{\mathbf{h}}), \tag{17}$$

then we get $\underline{\mathbf{z}}^{(2)} \leq \mathbf{z}^{(2)} \leq \overline{\mathbf{z}}^{(2)}$ for $k = 2$. With the same procedure, we can extend the results to every $k > 0$:

$$\underline{\mathbf{z}}^{(k)} \leq \mathbf{z}^{(k)}(\mathbf{h}) \leq \overline{\mathbf{z}}^{(k)} \tag{18}$$

Since $\lim_{k\to\infty} \left[\overline{\mathbf{z}}^{(k)}, \mathbf{z}^{(k)}(\mathbf{h}), \underline{\mathbf{z}}^{k}\right] = [\overline{\mathbf{z}}^*, \mathbf{z}^*(\mathbf{h}), \underline{\mathbf{z}}^*]$, we can finish our proposition:

$$\overline{\mathbf{z}}^* \geq \mathbf{z}^*(\mathbf{h}) \geq \underline{\mathbf{z}}^*, \tag{19}$$

for $\mathbf{h} \in [\underline{\mathbf{h}}, \underline{\mathbf{h}}]$ which means the inequality holds for every $\mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$. □

B. Proof for Proposition 3.2

The proof for Proposition 3.2 is as follows:

Proof. Multiplying the orthogonal matrix $\frac{1}{\sqrt{2}}\begin{pmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & -\mathbf{I}\end{pmatrix}$ from $\begin{pmatrix}\mathbf{W}_+ & \mathbf{W}_-\\ \mathbf{W}_- & \mathbf{W}_+\end{pmatrix}$'s left and right, we can get:

$$\begin{aligned}\frac{1}{2}\begin{pmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & -\mathbf{I}\end{pmatrix}\begin{pmatrix}\mathbf{W}_+ & \mathbf{W}_-\\ \mathbf{W}_- & \mathbf{W}_+\end{pmatrix}\begin{pmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & -\mathbf{I}\end{pmatrix} &= \frac{1}{2}\begin{pmatrix}\mathbf{W} & \mathbf{W}\\ |\mathbf{W}| & |\mathbf{W}|\end{pmatrix}\begin{pmatrix}\mathbf{I} & \mathbf{I}\\ \mathbf{I} & -\mathbf{I}\end{pmatrix}\\ &= \begin{pmatrix}\mathbf{W} & \mathbf{0}\\ \mathbf{0} & |\mathbf{W}|\end{pmatrix}\end{aligned}$$

Since orthogonal matrix won't change the spectral norm, we can prove our proposition:

$$\left\|\begin{pmatrix}\mathbf{W}_+ & \mathbf{W}_-\\ \mathbf{W}_- & \mathbf{W}_+\end{pmatrix}\right\|_2 = \left\|\begin{pmatrix}\mathbf{W} & \mathbf{0}\\ \mathbf{0} & |\mathbf{W}|\end{pmatrix}\right\|_2 = \max\left(\|\mathbf{W}\|_2, \||\mathbf{W}|\|_2\right) \tag{20}$$

then the proof is finished □

C. Proof for Proposition 3.4

The proof for Proposition 3.4 is as follows:

Proof. From Proposition 3.4, we can get that:

$$\left\|\begin{pmatrix}\mathbf{O}_+ & \mathbf{O}_-\\ \mathbf{O}_- & \mathbf{O}_+\end{pmatrix}\right\|_2 = \max(\||\mathbf{O}|\|_2, \|\mathbf{O}\|_2) \leq \|\mathbf{O}\|_F, \tag{21}$$

Since $\mathbf{O}\mathbf{O}^\top = \mathbf{I}$, then $\|\mathbf{O}\|_2 = 1$. Thereby,

$$1 \leq \left\|\begin{pmatrix}\mathbf{O}_+ & \mathbf{O}_-\\ \mathbf{O}_- & \mathbf{O}_+\end{pmatrix}\right\|_2, \tag{22}$$

then since $\mathbf{W}_{i,:}\mathbf{W}_{i,:}^\top = 1$ for $i = 0, ..., m-1$ because of the orthogonality, we can get

$$\|\mathbf{O}\|_F = \sqrt{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} W_{ij}^2} = \sqrt{\sum_{i=0}^{m-1}\mathbf{W}_{i,:}\mathbf{W}_{i,:}^\top} = \sqrt{m}, \tag{23}$$

and since $m < n$, we finally proved our proposition:

$$1 \leq \left\|\begin{pmatrix}\mathbf{O}_+ & \mathbf{O}_-\\ \mathbf{O}_- & \mathbf{O}_+\end{pmatrix}\right\|_2 \leq \sqrt{m} \leq \sqrt{n}, \tag{24}$$

□

D. Proof for Proposition 3.5

The proof for Porposition 3.5 is as follows:

Proof. We would like to finish the proof from the fixed point iteration view in the following. And we discuss the scenario for DEQ linear fusion layer or 1×1 convolution in the following, which implies $m = n$ for weight matrix $\mathbf{W}$. And the conclusion can easily extend to widely used 3×3 convolution because we can reformulate the 3×3 convolution as linear layers by vectorizing the input and obtain the same results.

Since $\mathbf{W}$ is obtained following our orthogonalization and scaling, its adjoint DEQ is contractive and can be regarded as iteratively solving the following functions.

$$\begin{pmatrix}\overline{\mathbf{z}}^{(k)}\\ \underline{\mathbf{z}}^{(k)}\end{pmatrix} = \sigma\left(\begin{pmatrix}\mathbf{W}_+ & \mathbf{W}_-\\ \mathbf{W}_- & \mathbf{W}_+\end{pmatrix}\begin{pmatrix}\overline{\mathbf{z}}^{(k-1)}\\ \underline{\mathbf{z}}^{(k-1)}\end{pmatrix} + \begin{pmatrix}\overline{\mathbf{h}}\\ \underline{\mathbf{h}}\end{pmatrix}\right), \tag{25}$$

with $(\overline{\mathbf{z}}^{(0)}, \underline{\mathbf{z}}^{(0)}) = (0, 0)$.

Firstly, from (Shi et al., 2021), one can get the following results for the explicit models for the bounds of a linear layer $\overline{\mathbf{g}}, \underline{\mathbf{g}}$, we have the following results:

$$\mathbb{E}\left[\sigma(\overline{\mathbf{g}}) - \sigma(\underline{\mathbf{g}})\right] = \frac{1}{2}\mathbb{E}\left[\overline{\mathbf{g}} - \underline{\mathbf{g}}\right], \tag{26}$$

where σ is the ReLU function.

And the expectation on DEQ's extraction part's $(\mathbf{Ux} + \mathbf{b})$ bound difference can be formulated $\mathbb{E}\left[\Delta_h\right] = n_u\mathbb{E}\left[|U|\right]\mathbb{E}\left[\mathbf{\Delta}_{in}\right]$. We use $\Delta_{out}^{(k)}$ to denote $\overline{\mathbf{z}}^{(k)} - \underline{\mathbf{z}}^{(k)}$ at $k-$th iteration with $\Delta_{out}^{(0)} = 0$. Then from the above fixed point iteration view, we can get:

$$\mathbb{E}\left[\Delta_{out}^{(1)}\right] = \mathbb{E}\left[\sigma(\overline{\mathbf{h}}) - \sigma(\underline{\mathbf{h}})\right] = \frac{1}{2}\mathbb{E}\left[\Delta_h\right]. \tag{27}$$

And for $k > 1$, we can obtain the following results by treating each iteration as an explicit layer:

$$\mathbb{E}\left[\Delta_{out}^{(k)}\right] = \mathbb{E}\left[\sigma(\mathbf{W}_+\overline{\mathbf{z}}^{(k-1)} + \mathbf{W}_-\underline{\mathbf{z}}^{(k-1)} + \overline{\mathbf{h}}) - \sigma(\mathbf{W}_+\underline{\mathbf{z}}^{(k-1)} + \mathbf{W}_-\overline{\mathbf{z}}^{(k-1)} + \underline{\mathbf{h}})\right], \tag{28}$$

$$= \frac{1}{2}\mathbb{E}\left[\mathbf{W}_+(\overline{\mathbf{z}}^{(k-1)} - \underline{\mathbf{z}}^{(k-1)}) - \mathbf{W}_-(\overline{\mathbf{z}}^{(k-1)} - \underline{\mathbf{z}}^{(k-1)}) + (\overline{\mathbf{h}} - \underline{\mathbf{h}})\right] \tag{29}$$

$$= \frac{1}{2}\mathbb{E}\left[|\mathbf{W}|\Delta_{out}^{(k-1)}\right] + \frac{1}{2}\mathbb{E}\left[\Delta_h\right], \tag{30}$$

$$\leq \frac{1}{2}\max_{i=1,\ldots,m}\sum_{j=1}^{n}|\mathbf{W}|_{ij}\mathbb{E}\left[\Delta_{out}^{(k-1)}\right] + \frac{1}{2}\mathbb{E}\left[\Delta_h\right]. \tag{31}$$

The inequality Eqn (31) is founded since we assume the elements for the layer's output enjoys the same distribution because the inputs and weights are all random. Such assumption is also used in other work (Shi et al., 2021). Then with the following inequality:

$$\sum_{j=0}^{n}|A_{ij}| \leq \max_{i=0,\ldots,m}\sum_{j=0}^{n}|A_{ij}| = \|\mathbf{A}\|_\infty \leq \sqrt{n}\|\mathbf{A}\|_2. \tag{32}$$

Then the inequality Eqn (29) can be rewritten as:

$$\mathbb{E}\left[\Delta_{out}^{(k)}\right] \leq \frac{1}{2}\sqrt{n}\|\mathbf{W}\|_2\mathbb{E}\left[\Delta_{out}^{(k-1)}\right] + \frac{1}{2}\mathbb{E}\left[\Delta_h\right], \tag{33}$$

$$\leq \left(\frac{1}{2}\sqrt{n}\|\mathbf{W}\|_2\right)^{k-1}\mathbb{E}\left[\Delta_{out}^{(1)}\right] + (\frac{1}{2} + \frac{\sqrt{n}\|\mathbf{W}\|_2}{4} + ... + \frac{(\sqrt{n}\|\mathbf{W}\|_2)^{k-2}}{2^{k-2}})\mathbb{E}\left[\Delta_h\right]. \tag{34}$$

Since $\sqrt{n}\|\mathbf{W}\|_2 \leq 1$ by our scaling and $\lim_{k\to\infty}\Delta_{out}^{k} = \overline{\mathbf{z}}^* - \underline{\mathbf{z}}^* = \Delta_{out}$ since the adjoint DEQ is contractive and has unique fixed point. And $\mathbb{E}\left[\Delta_{out}^{(1)}\right] = \frac{1}{2}\mathbb{E}\left[\Delta_h\right] < \infty$. We can complete our prove:

$$\mathbb{E}\left[\Delta_{out}\right] \leq \frac{1}{2}\sum_{i=0}^{\infty}\left(\frac{\sqrt{n}\|\mathbf{W}\|_2}{2}\right)^{i}\mathbb{E}\left[\Delta_h\right], \tag{35}$$

$$\leq \frac{1}{2\left(1 - \frac{\sqrt{n}\|\mathbf{W}\|_2}{2}\right)}\mathbb{E}\left[\Delta_h\right], \tag{36}$$

$$= \frac{n_u\mathbb{E}\left[|U|\right]\mathbb{E}\left[\mathbf{\Delta}_{in}\right]}{2 - \sqrt{n}\|\mathbf{W}\|_2} \tag{37}$$

$$\leq n_u\mathbb{E}\left[|U|\right]\mathbb{E}\left[\mathbf{\Delta}_{in}\right] \tag{38}$$

$$\tag{39}$$

Then we get that $\frac{\mathbb{E}[\Delta_{out}]}{\mathbb{E}[\mathbf{\Delta}_{in}]} \leq n_u\mathbb{E}\left[|U|\right]$. And the proof is completed. □

E. Hyper-parameter setting for different dataset for our CerDEQ.

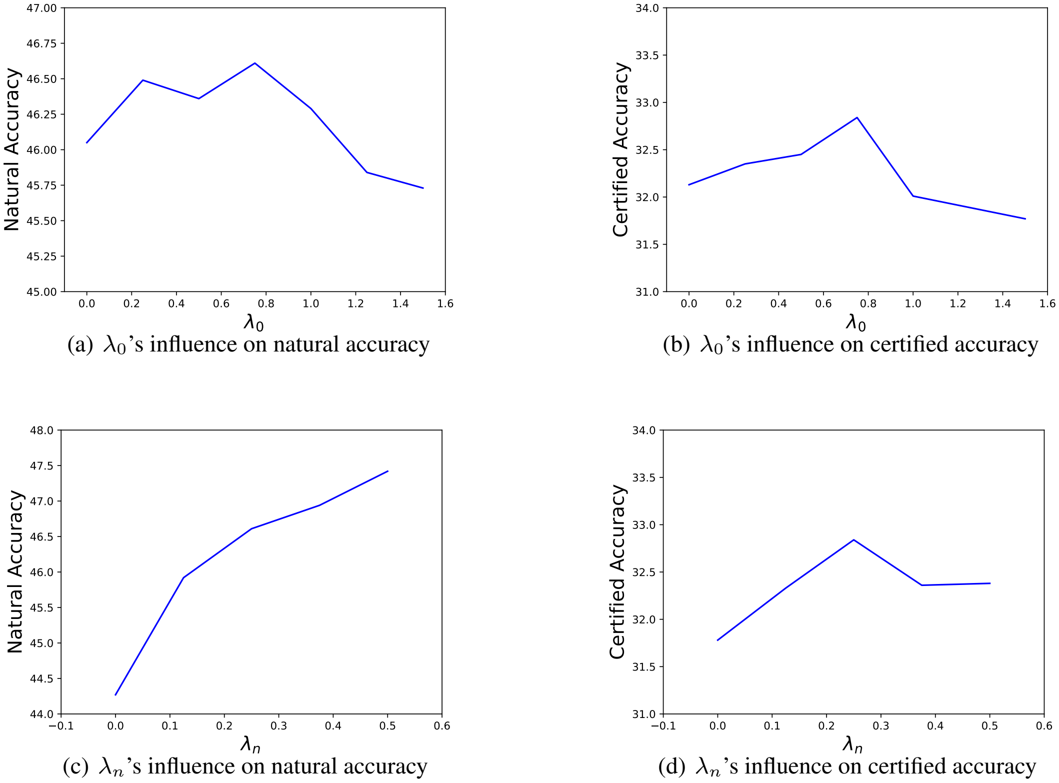
We list the hyper-parameters for our CerDEQ training as follows:

DataSet	Total Epoch	Target ϵ	ϵ-schedule Length	lr Decay	Decay Factor	Init lr	λ_n	λ_0
CIFAR-10	70	$8/255$	30	$50, 60, 67$	0.1	0.0005	0.25	0.75
	200	$8/255$	80	$140, 160, 180$	0.2	0.0005	0.25	0.75
TINY ImageNet	80	$1/255$	35	$50, 60, 67$	0.2	0.0005	0.25	1.25
	70	$8/255$	80	$140, 160, 180$	0.2	0.0005	0.25	1.25

Table 8. The hyper-paramters for CerDEQ's certified training for CIFAR-10 and Tiny ImageNet.

F. Ablation Studies on Hyperparameter λ_0 and λ_n

In this section, we are going to discuss the influence of the hyper-parameter λ_0 and λ_n as shown in Figure 5. We finish the experiments on CIFAR-10 for 70 epochs with the hyper-parameters as listed in Table 8 only changing λ_0 or λ_n. As for λ_0, one can see from (a) and (b) that the performance under natural and certified evaluation will first improve then degenerate as λ_0 increases. And although the increase of λ_n will enhance its natural performance, the certified accuracy will first increase and then decrease as λ_n increases. Thereby, a proper λ_n and λ_0 can help CerDEQ achieve a satisfactory trade-off between natural accuracy and certified accuracy.



(a) λ_0's influence on natural accuracy

(b) λ_0's influence on certified accuracy

(c) λ_n's influence on natural accuracy

(d) λ_n's influence on certified accuracy

Figure 5. The natural accuracy and certified accuracy with respect to different λ_0 or λ_n.