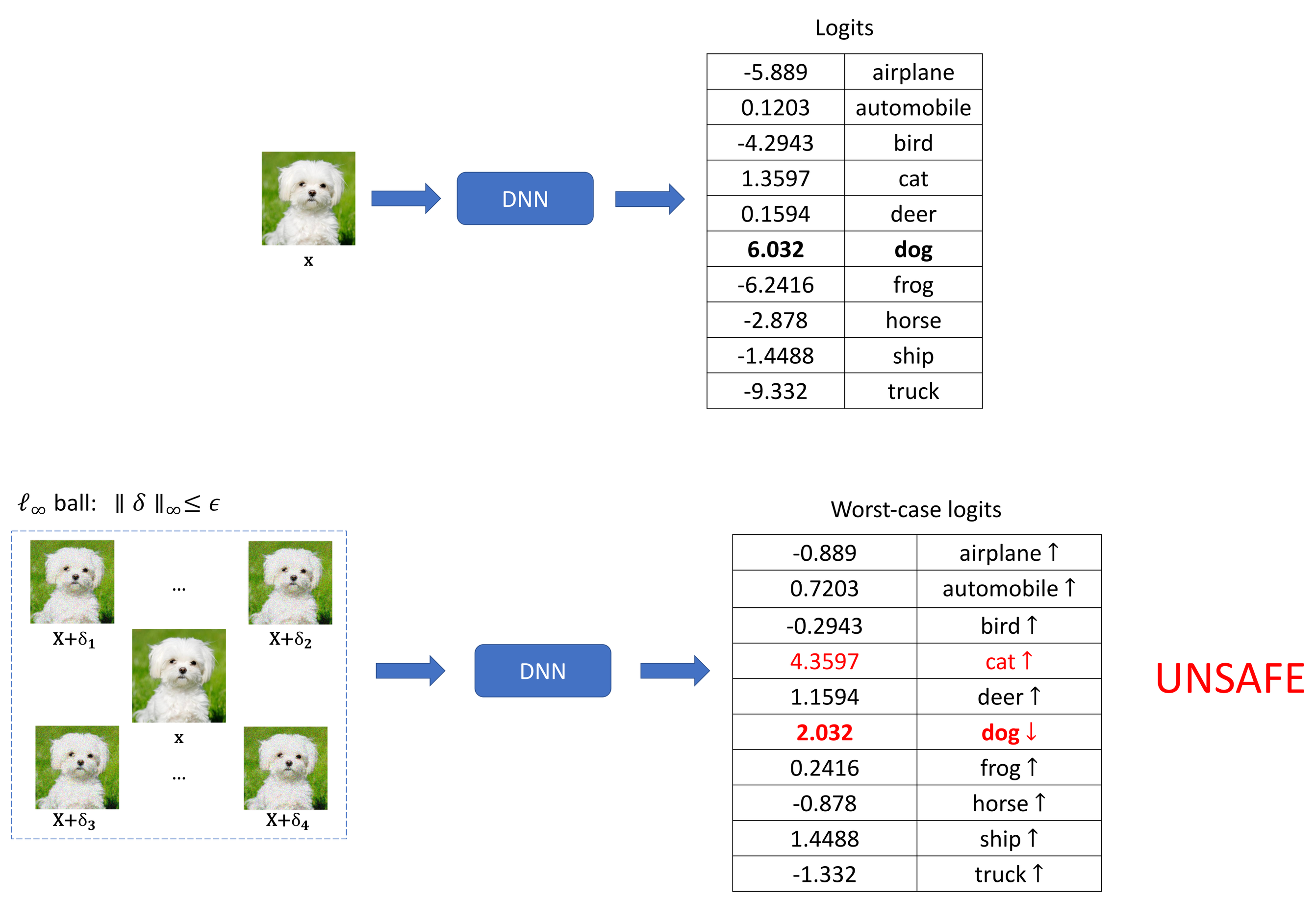
Fast Certified Robust Training with Short Warmup

Zhouxing Shi [1*], Yihan Wang[1*], Huan Zhang[1,2], Jinfeng Yi[3], Cho-Jui Hsieh[1]

[1]University of California, Los Angeles [2]Carnegie Mellon University [3]JD AI Research *Equal contribution

Certified Robustness and Certified Robust Training



Certified Robustness:

- It checks whether the model predicts correctly under the worst-case perturbation.
- Find the tractable bounds of the output logits.
- Safe if the lower bound of ground truth is larger than the upper bound of the others.
- Or the lower bound of the margin is larger than zero.

Certified Robust Training:

- Minimize an upper bound of the worst-case loss:

$$\min_\theta \overline{L}(f_\theta, \mathbf{x}, y, \epsilon), \quad \text{where } \overline{L}(f_\theta, \mathbf{x}, y, \epsilon) \ge \max_{\|\theta\|_\infty \le \epsilon} L(f_\theta, \mathbf{x}+\delta, y).$$

- It generally requires a **long** warmup/ramp-up schedule for ϵ.

Interval Bound Propagation (IBP) (Mirman et al., 2018; Gowal et al., 2018):

- Method to compute the output bounds.
- It computes and propagates an interval lower and upper bound for each neuron.

Motivations:

- Existing works using long training schedules are costly.
- We significantly reduce training schedules while maintain or even improve the robustness.

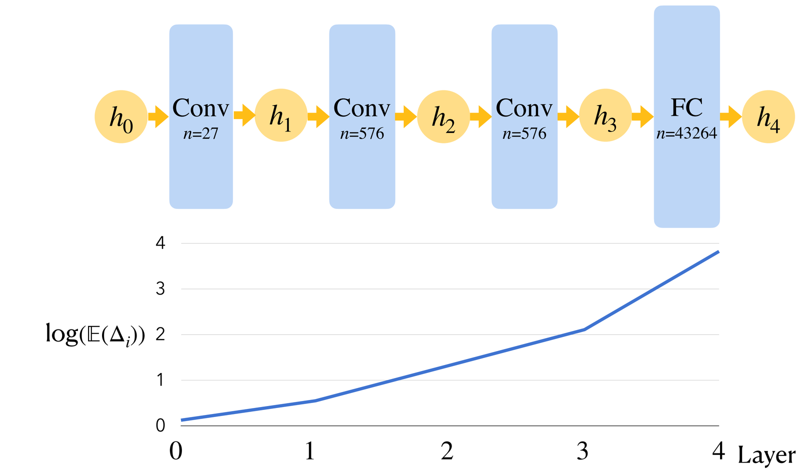
Issue in Existing IBP Training

Exploded Bounds:

- For affine layer $\mathbf{h}_i = \mathbf{W}_i \mathbf{z}_{i-1} + \mathbf{b}_i$, IBP computes:

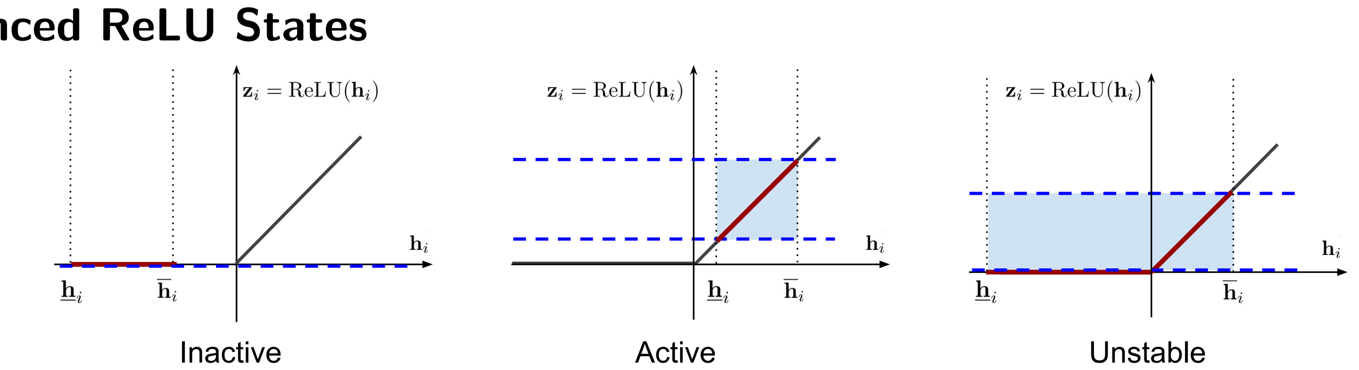
$$\underline{\mathbf{h}}_i = \mathbf{W}_{i,+}\underline{\mathbf{z}}_{i-1} + \mathbf{W}_{i,-}\overline{\mathbf{z}}_{i-1} + \mathbf{b}_i, \quad \overline{\mathbf{h}}_i = \mathbf{W}_{i,+}\overline{\mathbf{z}}_{i-1} + \mathbf{W}_{i,-}\underline{\mathbf{z}}_{i-1} + \mathbf{b}_i.$$

Method	Difference Gain				
	Closed form	$n_i = 27$	$n_i = 576$	$n_i = 1152$	$n_i = 32768$
Xavier (uniform)	$\frac{1}{4}\sqrt{n_i}$	1.30	6.00	8.48	45.25
Orthogonal	-	2.09	9.58	13.54	72.22
Kaiming (uniform)	$\frac{\sqrt{3}}{4}\sqrt{n_i}$	3.20	14.70	20.77	110.85
IBP Initialization (ours)	1	1.01	1.00	1.00	1.00



- Tightness of bounds, $\Delta_i = \overline{\mathbf{h}}_i - \underline{\mathbf{h}}_i = |\mathbf{W}_i|(\overline{\mathbf{z}}_{i-1} - \underline{\mathbf{z}}_{i-1})$, grows as $\mathbb{E}(\Delta_i) = \frac{n_i}{2}\mathbb{E}(|\mathbf{W}_i|)\mathbb{E}(\Delta_{i-1})$, for fan-in number n_i.
- *Difference gain* as $\mathbb{E}(\Delta_i)/\mathbb{E}(\Delta_{i-1}) = \frac{n_i}{2}\mathbb{E}(|\mathbf{W}_i|)$ is large for existing weight initialization.

Imbalanced ReLU States



- IBP tends to prefer inactive (dead) neurons for tighter bounds, but it can harm training.
- Shorter ramp-up leads to harder optimization and more severe imbalance.

The Proposed Method

IBP initialization:

- Initialize weights with a normal distribution, such that the *difference gain* is 1:

$$\frac{n_i}{2}\mathbb{E}(|\mathbf{W}_i|) = \frac{n_i}{2}\sqrt{2/\pi}\sigma_i = 1, \quad \Rightarrow \sigma_i = \frac{\sqrt{2\pi}}{n_i}$$

Fully Adding Batch Normalization (BN):

- BN can balance ReLU states and normalize the variance of bounds.
- But BN was partly or fully missed in the models used by prior works.
- We fully add BN after every convolution or fully-connected layer in IBP training.

Warmup Regularization:

- Two regularizers for the warmup stage of IBP training to explicitly tighten certified bounds and balance ReLU activation states:
  - *Bound tightness regularizer*:

$$\mathcal{L}_{\text{tightness}} = \frac{1}{\tau m}\sum_{i=1}^{m} \mathrm{ReLU}(\tau - \frac{\hat{\mathbb{E}}(\Delta_0)}{\hat{\mathbb{E}}(\Delta_i)}).$$

  - *ReLU activation state balancing regularizer*:

$$\alpha_i = \frac{\sum_j \mathbb{I}(\underline{\mathbf{h}}_{i,j} > 0)\mathbf{c}_{i,j}}{-\sum_j \mathbb{I}(\overline{\mathbf{h}}_{i,j} < 0)\mathbf{c}_{i,j}}, \quad \beta_i = \frac{\sum_j \mathbb{I}(\underline{\mathbf{h}}_{i,j} > 0)(\mathbf{c}_{i,j} - \hat{\mathbb{E}}(\mathbf{c}_i))^2}{\sum_j \mathbb{I}(\overline{\mathbf{h}}_{i,j} < 0)(\mathbf{c}_{i,j} - \hat{\mathbb{E}}(\mathbf{c}_i))^2},$$

$$\mathcal{L}_{\text{relu}} = \frac{1}{\tau m}\sum_{i=1}^{m}(\mathrm{ReLU}(\tau - \min(\alpha_i, \frac{1}{\alpha_i})) + \mathrm{ReLU}(\tau - \min(\beta_i, \frac{1}{\beta_i}))).$$

Experiments

Table 1. Main results on CIFAR-10 ($\epsilon_{\text{target}} = 8/255$). "†" indicates concurrent works.

Schedule (epochs)	Method	CNN-7 Standard	CNN-7 Verified	Wide-ResNet Standard	Wide-ResNet Verified	ResNeXt Standard	ResNeXt Verified
160 (1+80+79)	Vanilla IBP	53.80 ± 0.71	67.01 ± 0.29	54.31 ± 0.46	67.45 ± 0.21	55.23 ± 0.12	68.28 ± 0.15
	CROWN-IBP	58.76 ± 0.76	69.67 ± 0.38	60.39 ± 0.33	70.07 ± 0.42	61.08 ± 0.35	71.26 ± 0.11
	Ours	**51.72 ± 0.40**	**65.58 ± 0.32**	**51.95 ± 0.27**	**65.91 ± 0.14**	**53.68 ± 0.33**	**66.91 ± 0.40**
	Ours (best)	**51.06**	**65.03**	51.63	65.72	53.38	66.41

Literature results	Warmup	Total (epochs)	Standard	Verified
Gowal et al., 2018	(5K+50K) steps	3,200	50.51	68.44
Zhang et al., 2019	(320 + 1600) epochs	3,200	54.02	66.94
Balunovic & Vechev, 2020	N/A	800	48.3	72.5
Xu et al., 2020	(100 + 800) epochs	2,000	53.71	66.62
†IBP+ParamRamp (Lyu et al., 2021)	(320 + 1600) epochs	3,200	55.28	67.09
†CROWN-IBP+ParamRamp (Lyu et al., 2021)	(320 + 1600) epochs	3,200	51.94	65.08
†ℓ_∞-dist net (other architecture) (Zhang et al., 2021)	N/A	800	48.32	64.90

Table 2. Comparison of estimated time cost (seconds), for CNN-7 on CIFAR-10.

Method	Epochs	Total
IBP	3200	40496 × 4
CROWN-IBP (w/o loss fusion)	3200	91288 × 4
CROWN-IBP	2000	52362 × 4
†IBP+ParamRamp	3200	40496 × 4 × 1.09
†CROWN-IBP+ParamRamp	3200	91288 × 4 × 1.51
Vanilla IBP (verified error 67.01±0.29)	160	8747.9
CROWN-IBP (verified error 69.67±0.38)	160	10641.3
Ours (verified error 65.58±0.32)	160	9512.3

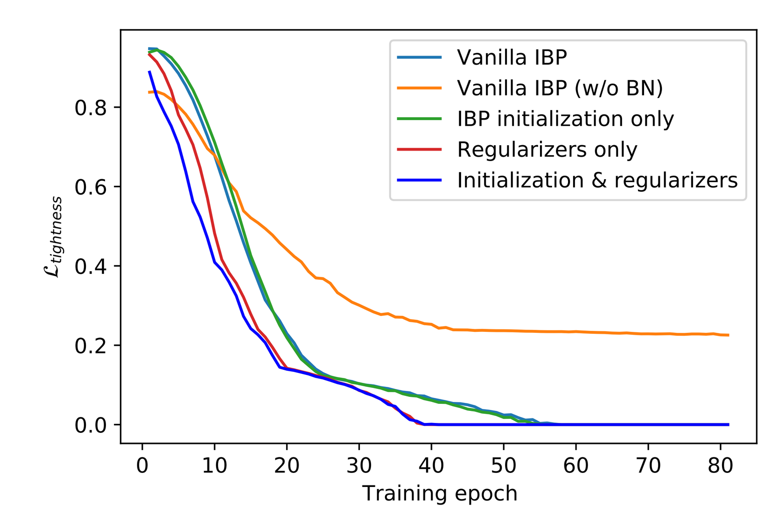


Figure 1. Curve of $\mathcal{L}_{\text{tightness}}$.

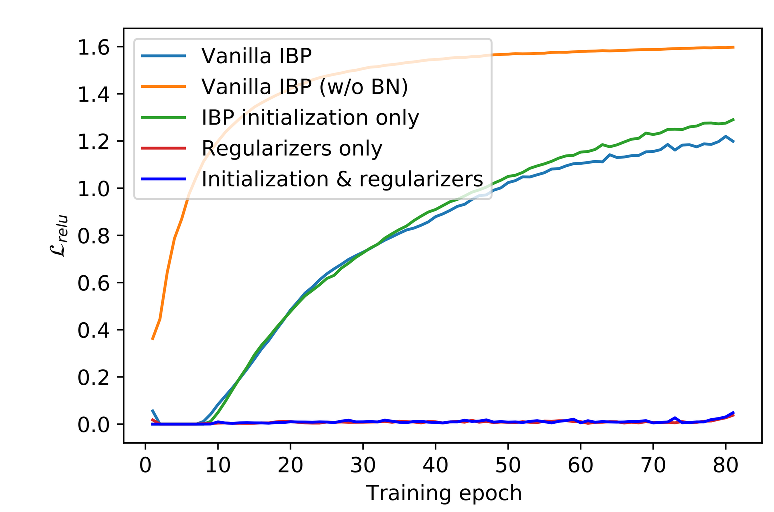


Figure 2. Curve of $\mathcal{L}_{\text{relu}}$.

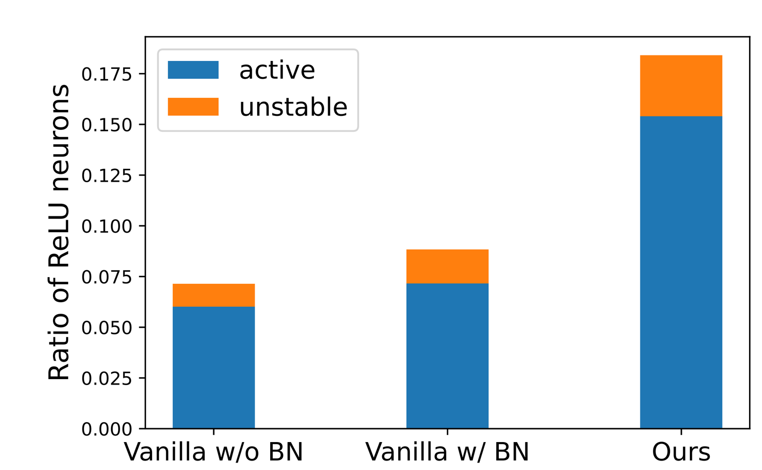


Figure 3. Ratios of active and unstable ReLU neurons a CNN on CIFAR-10.

- **Fast certified robust training with short training time (17 times speed-up) while achieving the state-of-the-art verified errors with CNN.**