CTBench: A Library and Benchmark for Certified Training

Yuhao Mao, Stefan Balauca, Martin Vechev

Department of Computer Science
ETH zürich SRILAB INSAIT

We introduce CTBench, a unified library and high-quality benchmark for certified training

- Simple interface, easy to use&extend
- Incorporates all state-of-the-art certified training methods
- Establishes a fair comparison across certified training algorithms
- Provides a deep insight into certified models

Training Certifiably Robust Neural Networks

Robustness: $f(x')_{i^*} - f(x')_i \geq 0, \forall i, x' \text{ s.t. } \|x' - x\|_\infty \leq \epsilon.$

Certified robustness: mathematically prove the robustness property via autonomous algorithms. Powerful but not scalable to large networks!

Practice: some networks are easier to certify.

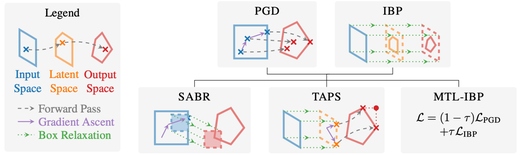
Certified training: train neural networks that have both high performance and high certified robustness!

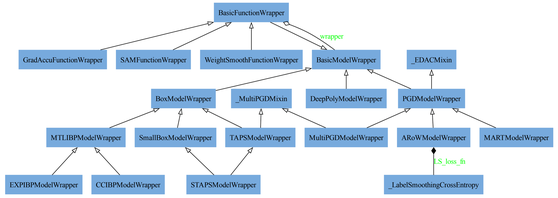
Motivation:

Many certified training algorithms,

NO unified library & high-quality evaluation.

The CTBench Library

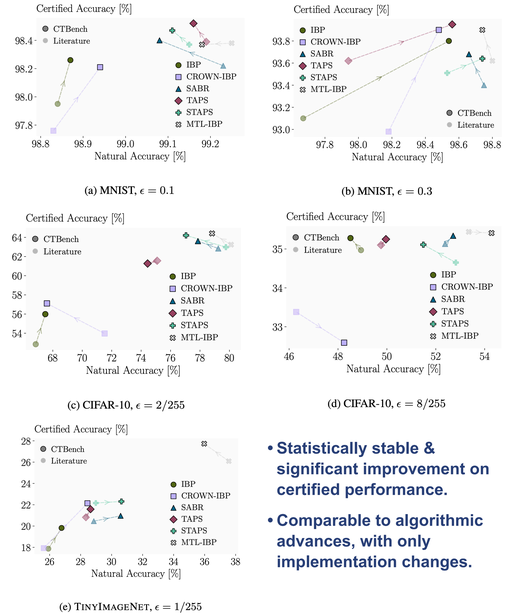




- `model_wrapper = BasicModelWrapper(net, nn.CrossEntropyLoss(), input_dim, device, args)`
- `model_wrapper = BoxModelWrapper(net, nn.CrossEntropyLoss(), input_dim, device, args)`
- `model_wrapper = TAPSModelWrapper(net, nn.CrossEntropyLoss(), input_dim, device, args, block_sizes=args.block_sizes, relu_shrinkage=args.relu_shrinkage)`
- `model_wrapper = DeepPolyModelWrapper(net, nn.CrossEntropyLoss(), input_dim, device, args, use_dp_box=True, loss_fusion=args.use_loss_fusion, keep_fusion_when_test=args.keep_fusion_when_test)`

- **Easy-to-use interface.**
- **Clean architecture design.**
- **Comprehensive & unified.**
- **Easy to extend.**

The CTBench Benchmark



(a) MNIST, $\epsilon = 0.1$ (b) MNIST, $\epsilon = 0.3$

(c) CIFAR-10, $\epsilon = 2/255$ (d) CIFAR-10, $\epsilon = 8/255$

(e) TinyImageNet, $\epsilon = 1/255$

- **Statistically stable & significant improvement on certified performance.**
- **Comparable to algorithmic advances, with only implementation changes.**

Understanding Certified Models

Mistakes are not independent

For models trained with different algorithms

		# models succeeded						
		0	1	2	3	4	5	6
$\epsilon = 0.1$	obs.	93	25	21	30	32	56	9743
	exp.	0	0	0	1	37	900	9062
$\epsilon = 0.3$	obs.	452	73	53	51	80	111	9180
	exp.	0	0	2	39	445	2698	6816

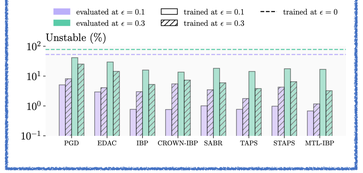
For different certification algorithms

		neither certify	one certifies	both certify
$\epsilon = 2/255$	obs.	3549	15	6436
	exp.	1264	4585	4151
$\epsilon = 8/255$	obs.	6454	9	3537
	exp.	4171	4575	1254

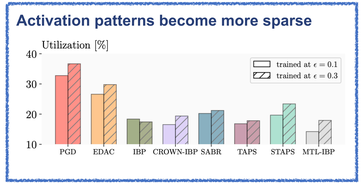
For different model architectures

Models	Number of not certified samples	
	Observed	Expected if independent
CNN5 IBP	771	/
CNN5 SABR	793	/
CNN5 MTL-IBP	746	/
CNN7 IBP	620	/
CNN7 SABR	632	/
CNN7 MTL-IBP	610	/
CNN5, CNN7 IBP	526	48
CNN5, CNN7 SABR	541	50
CNN5, CNN7 MTL-IBP	516	46
All 3 CNN5 networks	593	5

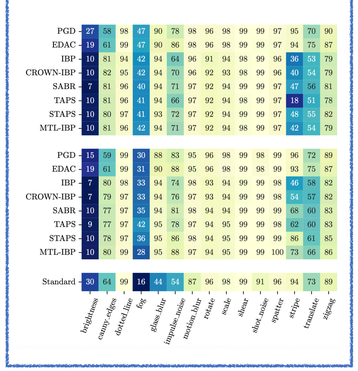
Loss fragmentation is further reduced



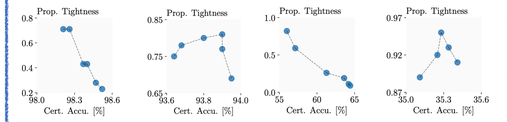
Activation patterns become more sparse



Out-of-distribution performance



Optimal regularization strength



(a) MNIST $\epsilon = 0.1$ (b) MNIST $\epsilon = 0.3$ (c) CIFAR-10 $\epsilon = \frac{2}{255}$ (d) CIFAR-10 $\epsilon = \frac{8}{255}$