

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

CTBench: A Library and Benchmark for Certified Training

Yuhao Mao, Stefan Balauca, Martin Vechev

Department of Computer Science

ETH zürich SRILAB INSAIT

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.

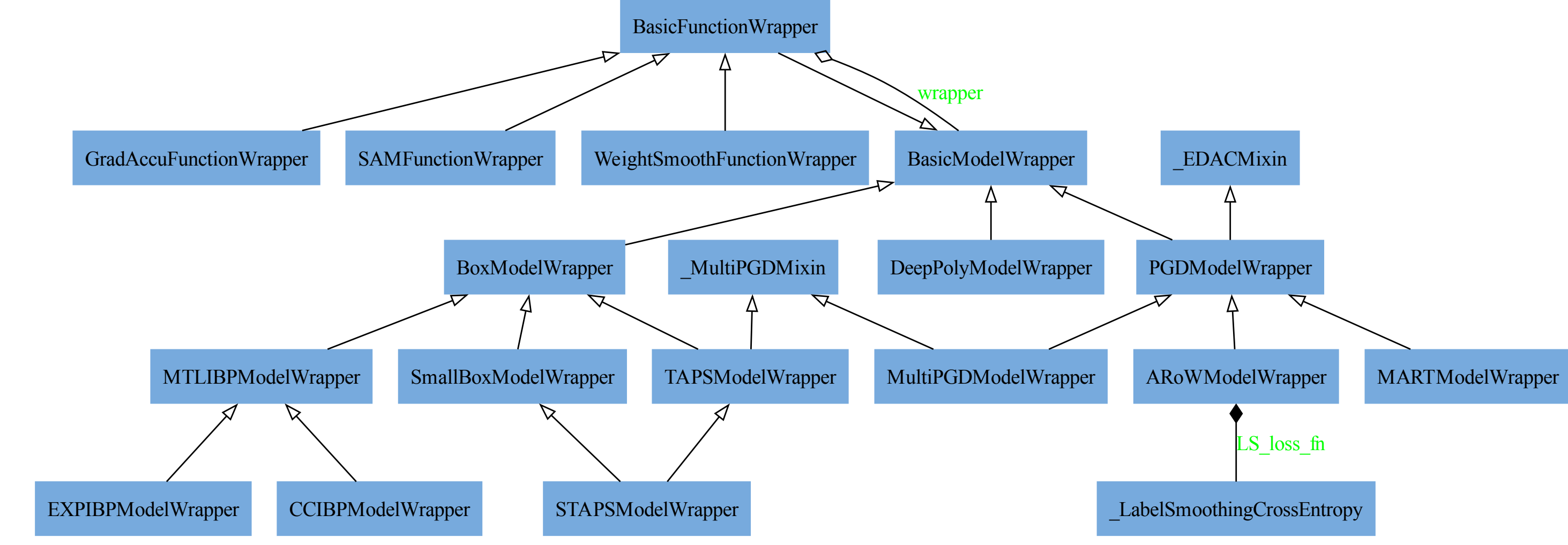
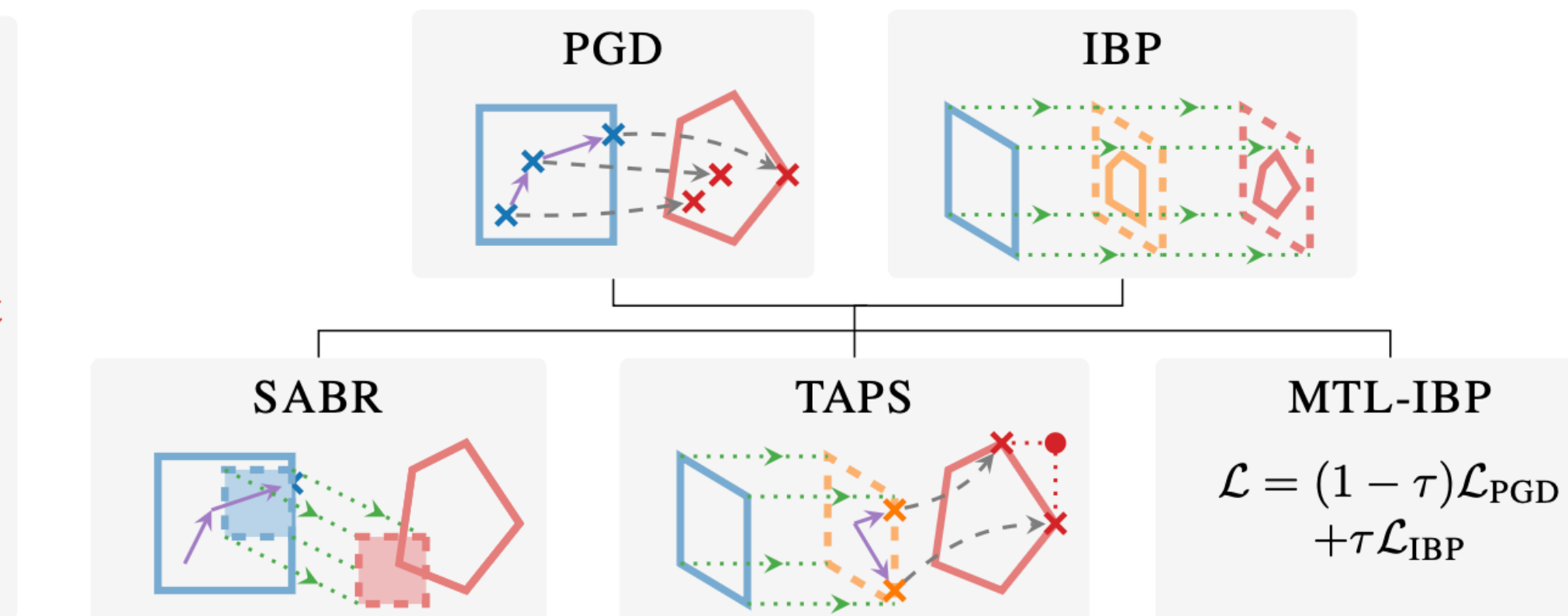
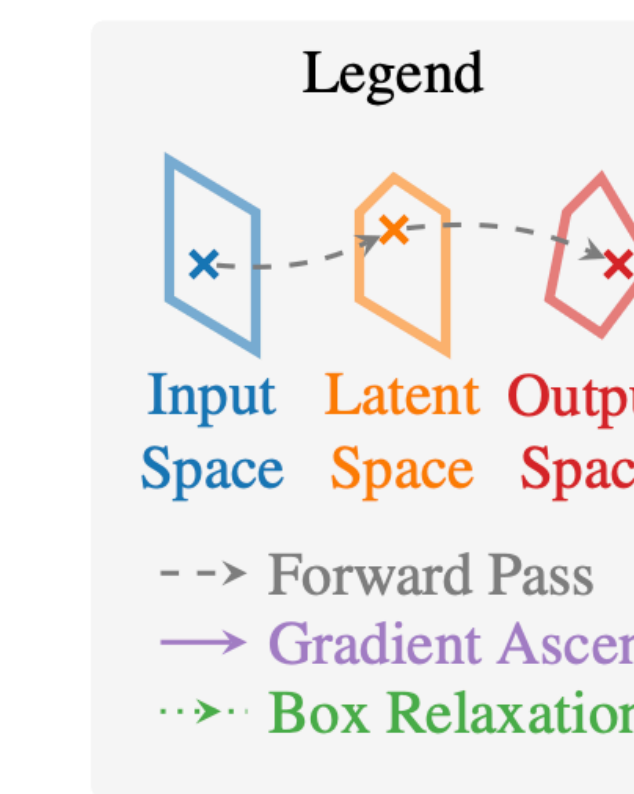
Certified training: train neural networks that have both high performance and high certified robustness!

Motivation:

Many certified training algorithms,

NO unified library & high-quality evaluation.

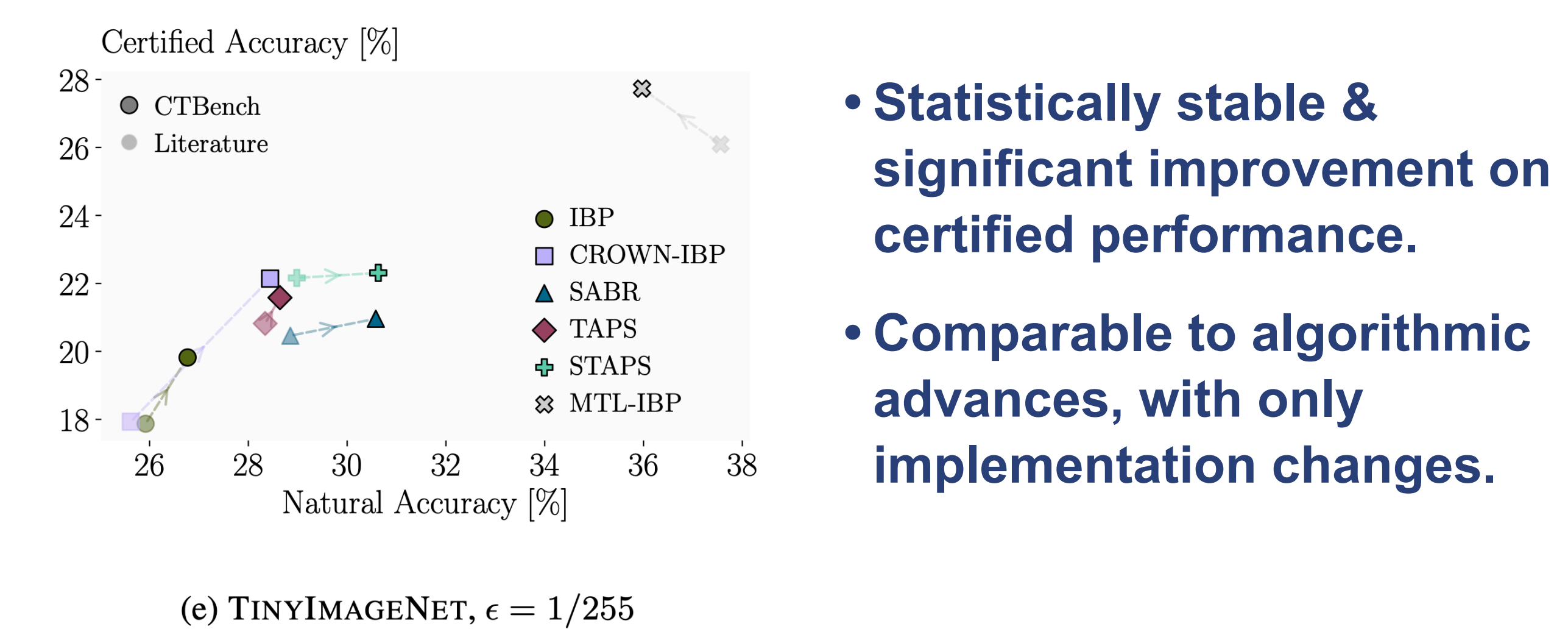
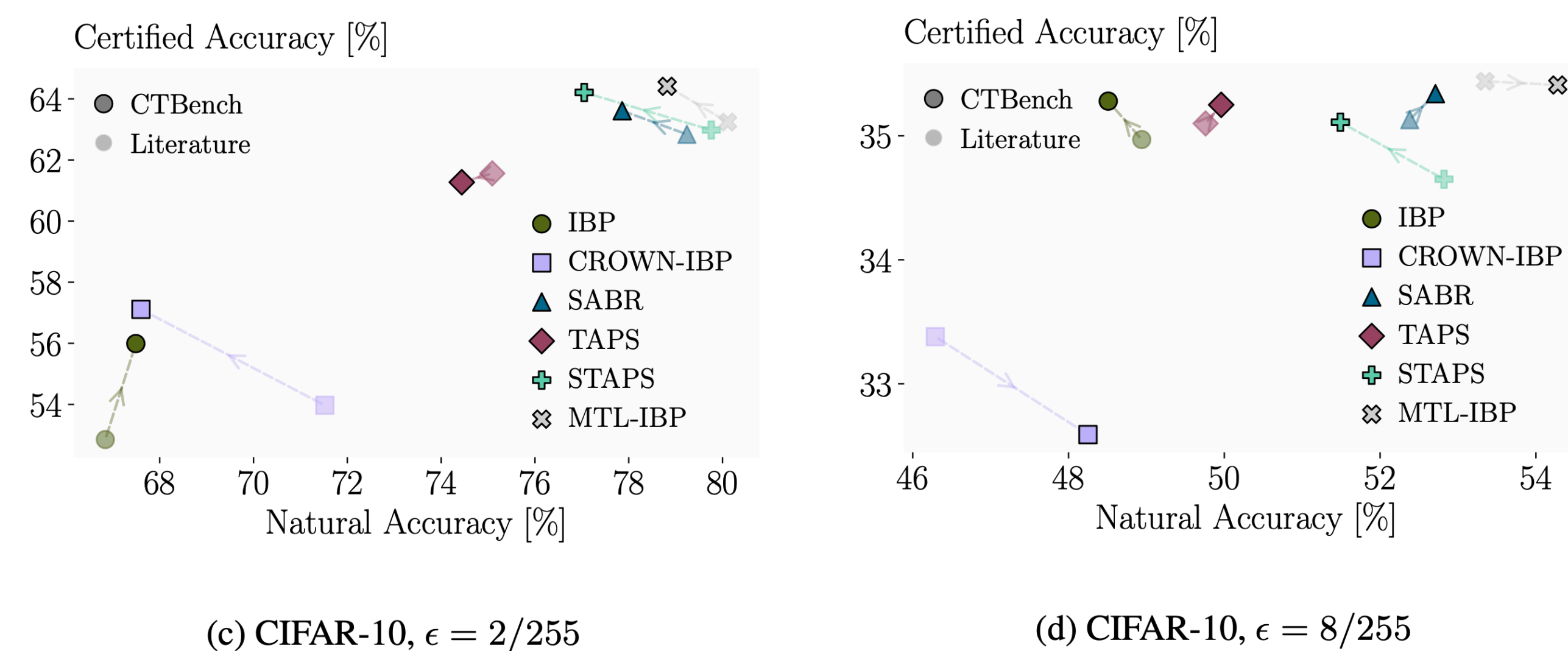
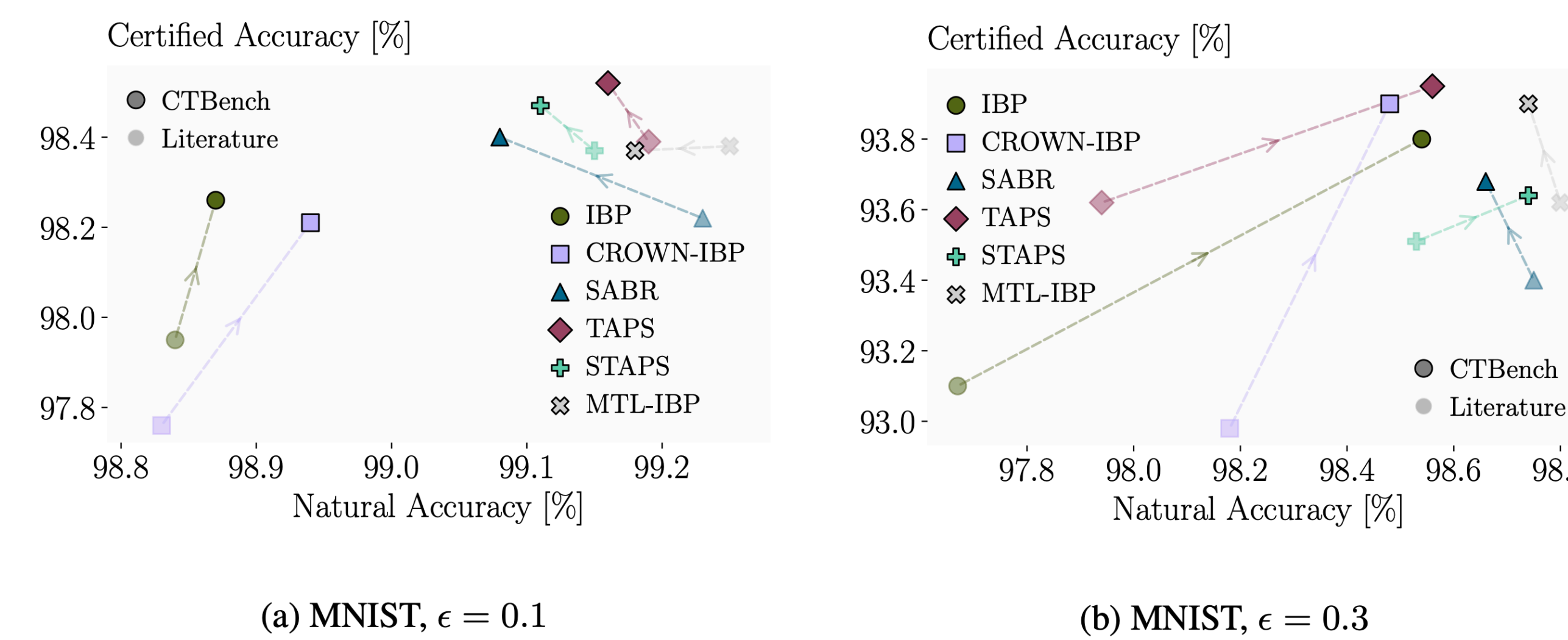
The CTBench Library



- `model_wrapper = BasicModelWrapper(net, nn.CrossEntropyLoss(), input_dim, device, args)`
- `model_wrapper = BoxModelWrapper(net, nn.CrossEntropyLoss(), input_dim, device, args)`
- `model_wrapper = TAPSMModelWrapper(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



- 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	1	37	900	9062	
$\epsilon = 0.3$	obs.	452	73	53	51	80	111	9180
	exp.	0	0	2	39	445	2698	6816

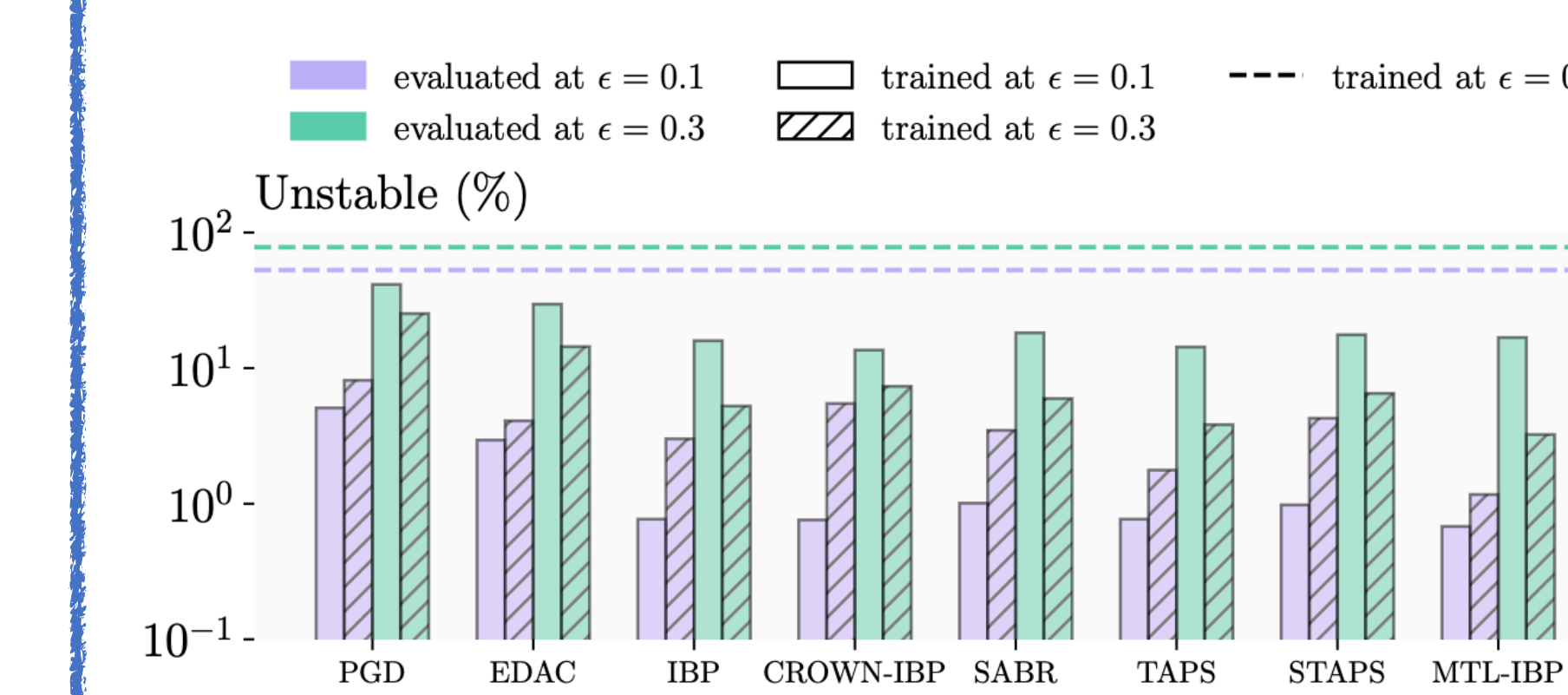
For different certification algorithms

		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

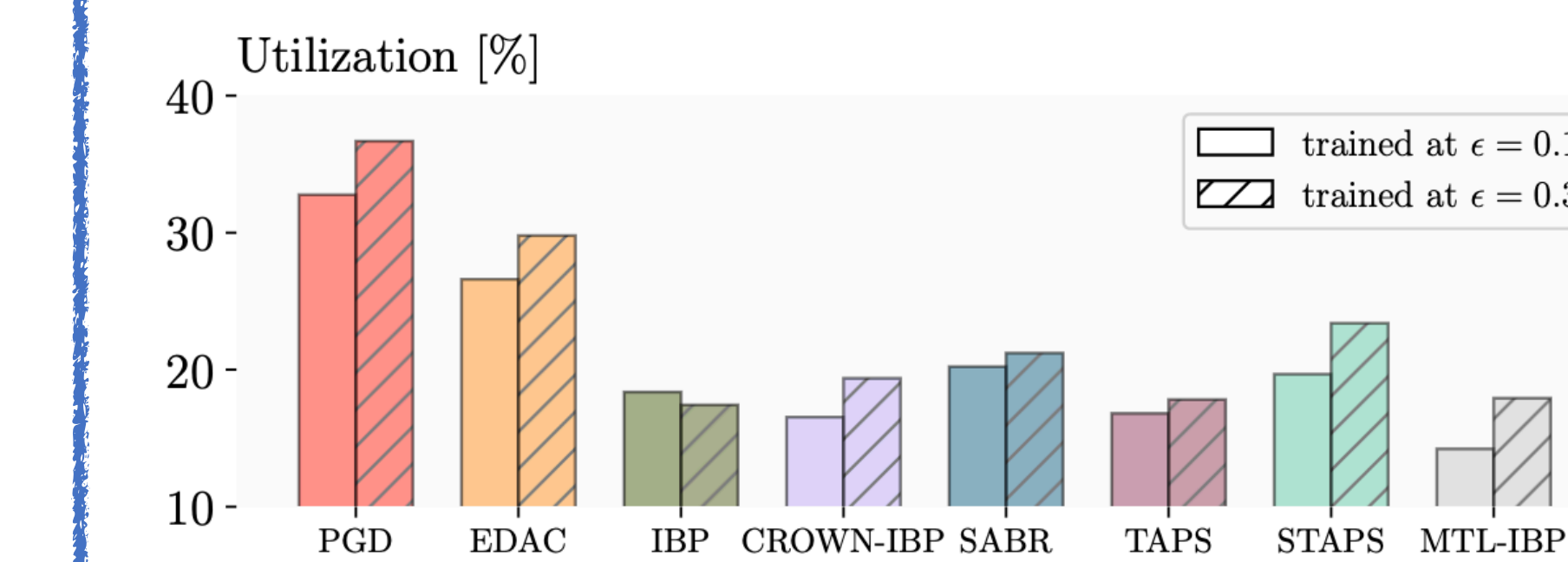
For different model architectures

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

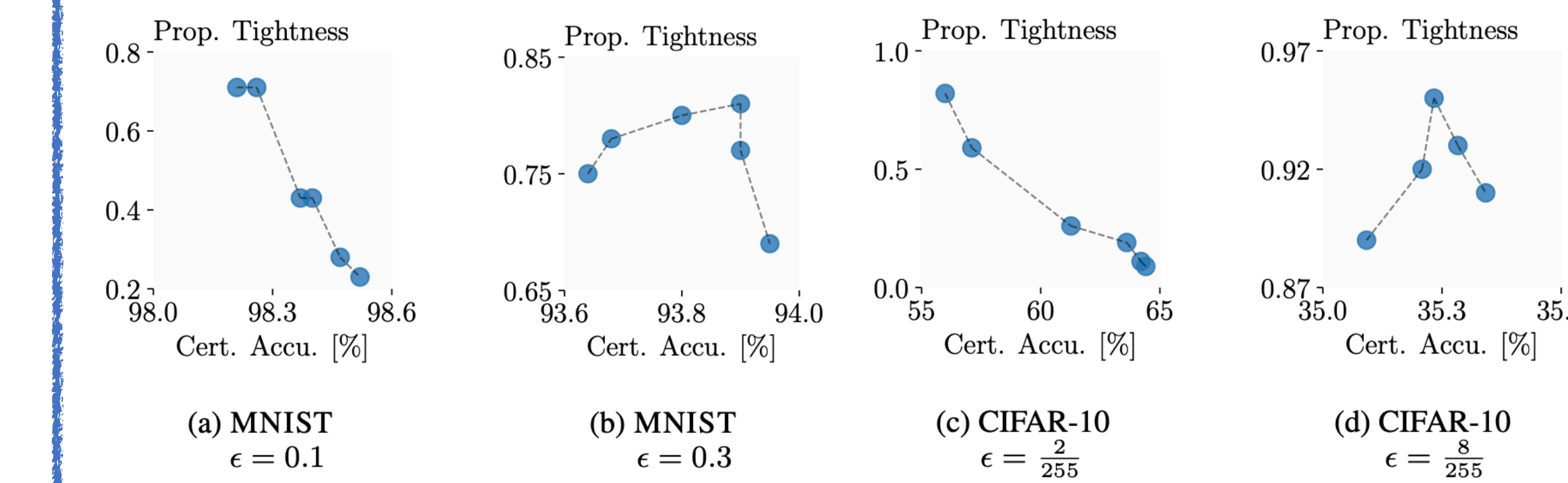
Loss fragmentation is further reduced



Activation patterns become more sparse



Optimal regularization strength



Out-of-distribution performance

PGD	27	58	98	47	90	78	98	96	98	99	97	95	70	90
EDAC	19	61	99	47	90	86	98	96	98	99	97	94	75	87
IBP	10	81	94	42	94	64	96	91	94	98	99	96	36	53
CROWN-IBP	10	82	95	42	94	70	96	92	93	98	99	96	40	54
SABR	7	81	96	40	94	71	97	92	94	99	99	97	47	56
TAPS	10	81	96	41	94	66	97	92	94	98	99	97	18	51
STAPS	10	80	97	41	93	72	97	92	94	99	99	97	48	55
MTL-IBP	10	81	96	42	94	71	97	92	94	98	99	97	42	54

PGD	15	59	99	30	88	83	95	96	98	99	98	99	96	72	89
EDAC	19	61	99	31	90	88	95	96	98	99	98	99	93	75	87
IBP	7	80	98	33	94	74	98	93	94	99	99	98	46	58	82
CROWN-IBP	7	79	97	33	94	76	97	93	94	99	99	98	54	57	82
SABR	10	77	97	35	94	81	98	94	94	99	99	99	68	60	83
TAPS	9	77	97	42	95	78	97	94	95	99	99	98	62	60	83
STAPS	10	78	97	36	95	86	98	94	95	99	99	99	86	61	85
MTL-IBP	10	80	99	28	95	88	97	94	95	99	99	100	73	66	86

Standard	30	64	99	16	44	54	87	96	98	99	91	96	94	73	89
brightness															
canary edges															
dotted line															
fog															
glass blur															
impulse noise															
motion blur															
rotate															
scale															
shear															
short noise															
stripes															
translate															
zigzag															

