Linearity Grafting: Relaxed Neuron Pruning Helps Certifiable Robustness

[ICML 2022] Tianlong Chen*[1], Huan Zhang *[2], Zhenyu Zhang[1],
Shiyu Chang[3], Sijia Liu[4,5], Pin-Yu Chen[5,6], Zhangyang Wang[1]
[1] University of Texas at Austin, [2] Carnegie Mellon University, [3] University of California, Santa Barbara,
[4] Michigan State University, [5] MIT-IBM Watson AI Lab, [6] IBM Research

The University of Texas at Austin
Electrical and Computer Engineering

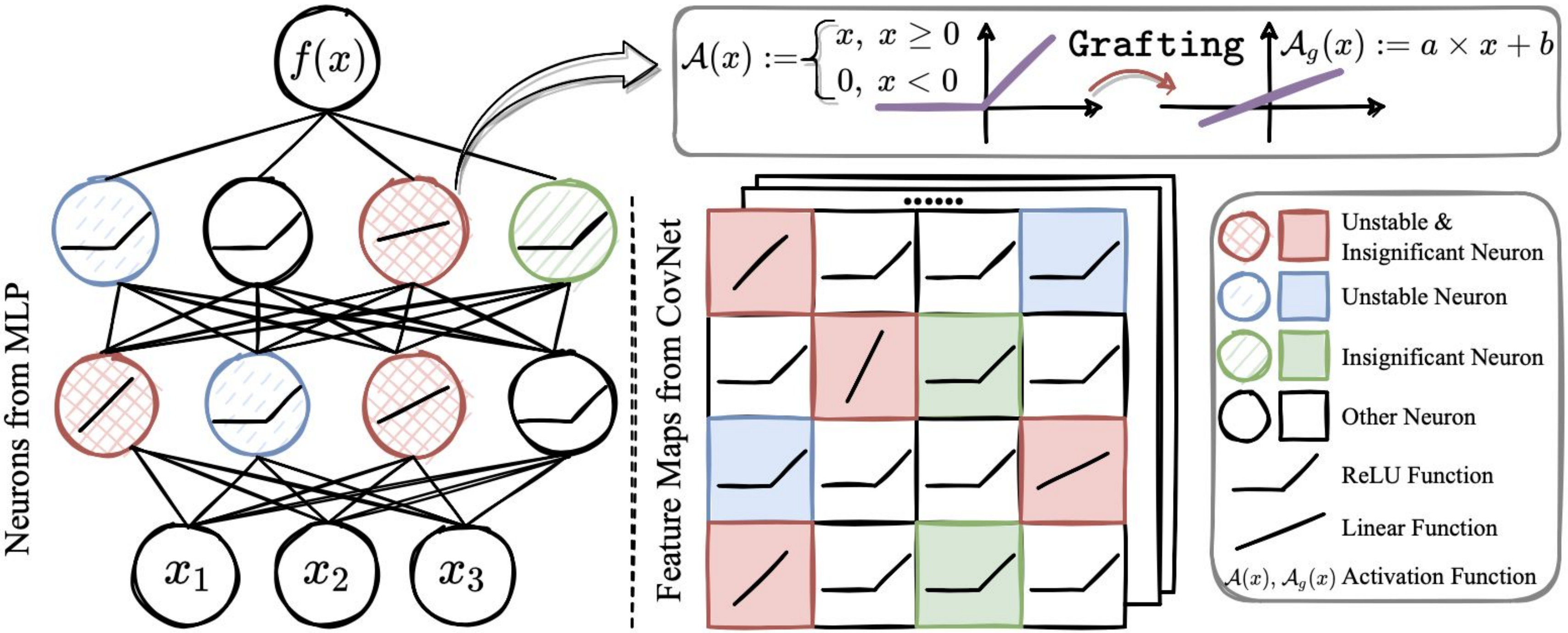
Agenda

- ➢ Motivations
- ➢ Methodology
- ➢ The Superiority of Grafting for Verification
- ➢ More Experiment results

Motivations

- ➔ Certifiable robustness is a highly desirable property for adopting deep neural networks (DNNs) in safety-critical scenarios, but often demands tedious computations to establish.
- ➔ The main hurdle lies in the massive amount of non-linearities in large DNNs. For instance, the “unstable” neurons in ReLU networks.
- ➔ To trade off the DNN expressiveness (which calls for more non-linearities) and robustness certification scalability (which prefers more linearities), we propose a novel solution to strategically manipulate neurons, by “grafting” appropriate levels of linearity.

Methodology



Methodology

1. Robustify a DNN as the starting point.
2. Identify insignificant and unstable neurons.
   a. Rank all neurons according to unstable scores. $r_u^{(i)} \in [0, 1]$
   b. Compute the importance of each neuron via certain heuristics or optimized scores. $r_s^{(i)} \in [0, 1]$
   c. Identify insignificant and unstable neurons by $\mathrm{argmax}_i \gamma \times r_u^{(i)} - r_s^{(i)}$
3. Linearize and tune the grafted activation functions.
4. Robustness verification with a complete verifier.

The Superiority of Grafting for Verification

[Finding 1] Achieving competitive certifiable robustness *without certified robust training.*
[Finding 2] Scaling up complete verification to large models.

Table 1. **Unstable neuron ratio (UNR %), verified accuracy (VA %), standard accuracy (SA %), PGD-100 robust accuracy (RA %), and average time (s)** of FAT trained models w./w.o. grafting on MNIST, SVHN, and CIFAR-10. α,β-CROWN, a SOTA complete verifier is used to compute VA. The target ℓ_∞ norm perturbation is $\epsilon = \frac{2}{255}$ except for MNIST. "OOM" indicates that DNNs have too many unstable neurons and the verifier is unable to load it with 48 GB GPU memory, leading to "∞" verification time and a null VA ("-").

FAT ($\epsilon = \frac{2}{255}$)	(ConvBig, MNIST w. $\epsilon = 0.1$)					(ConvBig, SVHN)					(CNN-B, CIFAR-10)				
	UNR	VA	SA	RA	Time	UNR	VA	SA	RA	Time	UNR	VA	SA	RA	Time
Baseline	31.27	0.10	99.29	97.14	262.11	10.78	16.70	89.71	75.74	218.49	15.85	37.40	79.95	62.23	127.50
SAP (Dhillon et al., 2018) (50%)	7.38	4.20	99.22	96.34	292.94	5.65	25.90	89.85	76.03	195.87	6.27	47.30	75.10	58.01	58.98
GAP† (Ye et al., 2020) (50%)	17.29	3.50	99.19	96.46	295.21	6.14	26.20	90.09	77.28	195.78	10.22	42.50	79.05	61.81	103.03
Hydra‡ (Sehwag et al., 2020) (50%)	15.39	12.70	98.90	95.22	269.71	5.04	26.60	81.28	62.92	172.98	6.28	44.40	72.99	55.55	59.99
Random Grafting (50%)	17.16	12.00	98.93	95.38	273.94	6.13	37.40	87.37	73.27	150.23	9.07	42.50	75.02	57.19	83.25
Grafting (50%)	5.85	82.30	98.68	92.73	40.21	3.11	57.80	78.75	63.90	16.68	5.36	50.40	74.08	58.76	39.32
Grafting (30%)	10.43	59.40	99.13	95.24	137.40	5.45	56.80	80.71	66.05	31.76	7.15	49.00	77.10	60.87	64.80
Grafting (80%)	4.04	82.40	98.63	92.71	39.64	1.63	58.70	78.56	63.91	12.93	1.87	44.40	61.20	48.34	15.25

FAT ($\epsilon = \frac{2}{255}$)	(ResNet-4B, CIFAR-10)					(ConvBig, CIFAR-10)					(ConvHuge, CIFAR-10)				
	UNR	VA	SA	RA	Time	UNR	VA	SA	RA	Time	UNR	VA	SA	RA	Time
Baseline	19.94	0.80	76.69	60.14	45.56	17.75	1.30	84.90	68.10	121.61	OOM	-	90.68	73.57	∞
SAP (Dhillon et al., 2018) (50%)	6.18	21.70	49.03	38.30	137.77	5.54	25.80	65.08	50.45	156.28	8.52	2.00	80.29	60.29	181.06
GAP† (Ye et al., 2020) (50%)	13.67	5.10	68.42	53.43	239.14	10.97	1.10	81.91	64.50	190.42	7.43	1.00	86.38	67.91	111.77
Hydra‡ (Sehwag et al., 2020) (50%)	9.52	15.10	42.01	31.27	162.34	11.10	1.10	67.97	47.77	297.19	9.88	1.00	70.68	48.81	291.00
Random Grafting (50%)	13.59	7.40	69.56	52.53	267.74	12.23	3.90	79.33	60.92	285.71	11.34	1.00	84.47	64.76	206.97
Grafting (50%)	6.03	38.10	60.13	46.12	42.83	4.32	39.12	62.23	47.73	42.80	4.41	28.30	62.62	49.37	155.78
Grafting (30%)	12.89	24.50	63.71	49.16	153.69	10.30	27.30	71.97	54.97	159.74	OOM	-	90.19	72.34	∞
Grafting (80%)	2.91	39.70	57.64	44.61	25.16	1.89	41.00	55.20	44.27	10.87	0.17	32.30	40.80	33.43	4.06

† The heuristic of activation gradient magnitude (Ye et al., 2020) is utilized to guide activation pruning.
‡ Based on the official implementation of Sehwag et al. (2020), we extend the original sparse mask learning to activation sparsification.

The Superiority of Grafting for Verification

[Finding 3] Substantially reduced unstable neurons

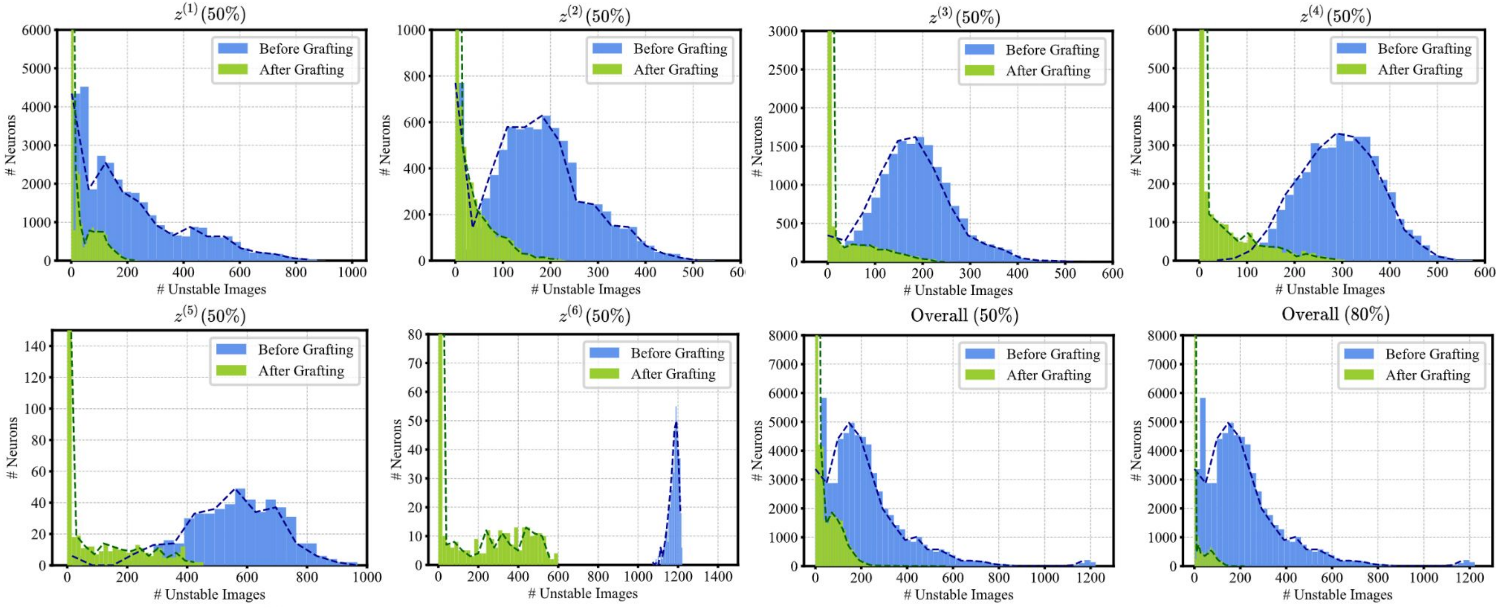


Figure 3. Layer-wise ($z^{(i)}$) and overall unstable neuron distribution of the 7-layer ConvBig on CIFAR-10, before and after performing grafting on 50% or 80% neurons. In specific, the point (m unstable images, n neurons) means that n neurons are unstable for m images.

The Superiority of Grafting for Verification

[Finding 4] Significantly tighter bound

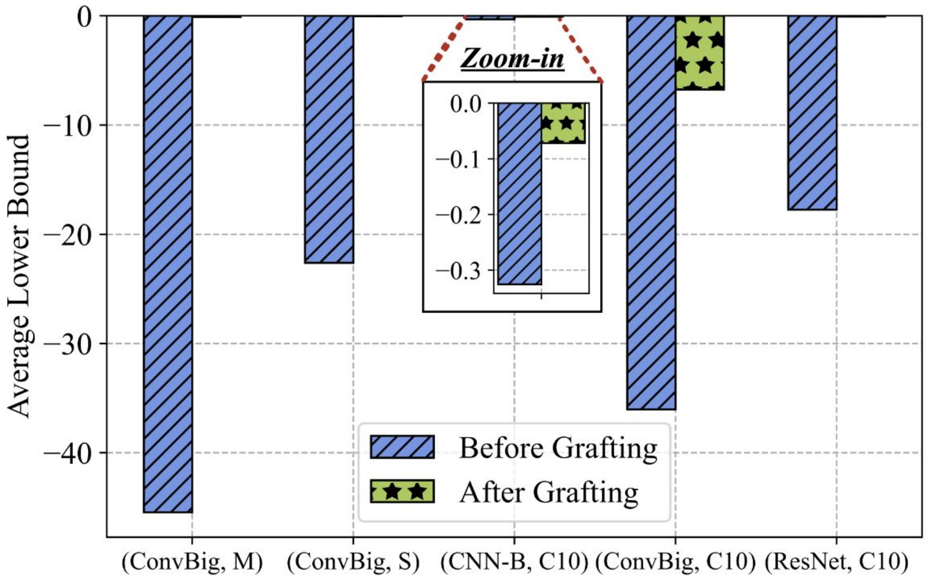


Figure 4. Average verified lower bounds of models before and after grafting 50% neurons. Bounds are produced by β-CROWN.

More Experiment Results

[Q1] How does the grafting criterion affect performance?

Table 2. Ablation on grafting criterion. Unstable neuron ratio (UNR %), VA (%), SA (%), and RA (%) of ConvBig with 50% grafted neurons on CIFAR-10 are reported.

Grafting Criterion	UNR	VA	SA	RA
$-r_s$	10.35	2.10	82.35	64.28
$r_u - r_s$	6.39	14.50	77.88	59.91
$2r_u - r_s$	4.32	38.90	62.15	47.70
r_u	4.13	38.70	59.39	45.77
$\gamma \times r_u - r_s$ (γ linearly decays $2 \to 0$)	4.32	39.12	62.23	47.73

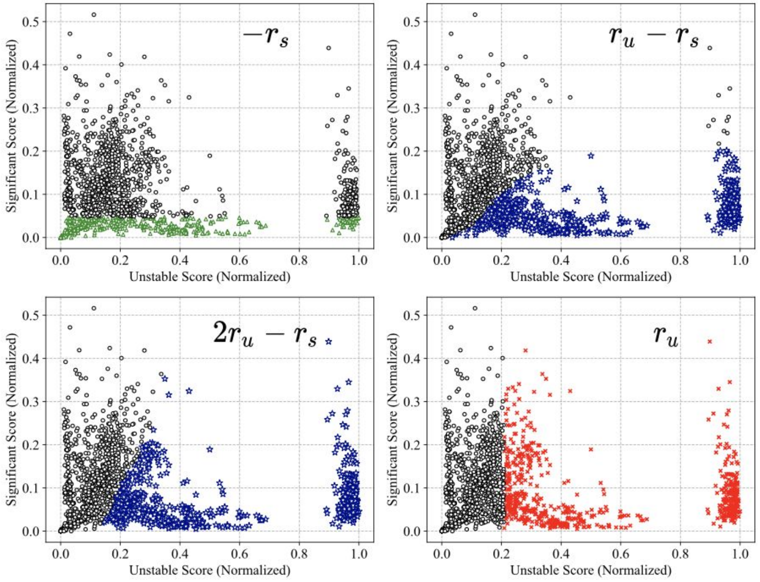


Figure 5. Neuron selections based on diverse picking criteria. △, ★, and ◊ indicate insignificant-only, insignificant-and-unstable, and unstable-only neuron selections respectively.

More Experiment Results

[Q2] Comparison with classical certified robust training

Table 6. Comparison between a representative certified robust training using Auto-LiRPA (Xu et al., 2020a), and our grafting with FAT. UNR (%), VA (%), SA (%), RA (%), and training time (hour) of CNN-B w./w.o. 50% grafted neurons on CIFAR-10 are reported.

Settings	UNR	VA	SA	RA	Training Time
Baseline (FAT)	15.85	37.40	79.95	62.23	0.39 h
Certified Robust Training	0.96	47.55	58.00	48.62	16.26 h
FAT + Grafting (50%)	5.36	50.40	74.08	58.76	1.13 h

The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering
IBM Research

Linearity Grafting: Relaxed Neuron Pruning Helps Certifiable Robustness

Tianlong Chen*[1], Huan Zhang*[2], Zhenyu Zhang[1], Shiyu Chang[3], Sijia Liu[4,5], Pin-Yu Chen[5,6], Zhangyang Wang[1]
[1]University of Texas at Austin, [2]Carnegie Mellon University, [3]University of California, Santa Barbara,
[4]Michigan State University, [5]MIT-IBM Watson AI Lab, [6]IBM Research

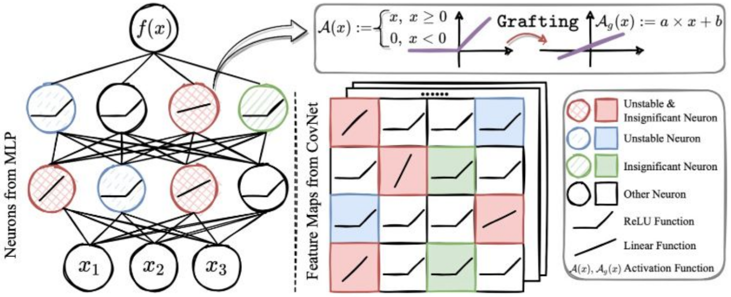
VITA GitHub

ICML International Conference On Machine Learning

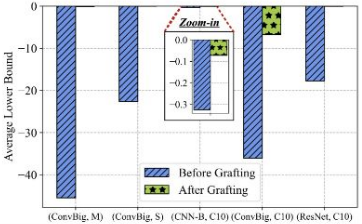
Motivations

- The main hurdle of certifying large DNNs lies in their massive amount of non-linearities, *e.g.*, the "unstable neurons" for ReLU networks.
- To trade off the DNN expressiveness (calls for more non-linearity) and robustness certification scalability (prefers more linearity), we "grafting" appropriate levels of linearity.

Methodology



(1) Robustify (*e.g.*, faster adversarial training) a DNN as the starting point;
(2) Identify insignificant and unstable neurons;
(3) Linearize and tune the grafted activation functions, $A_g(x) = a \times x + b$;
(4) Perform robustness verification with a complete verifier.



Benefits from Linearity Grafting

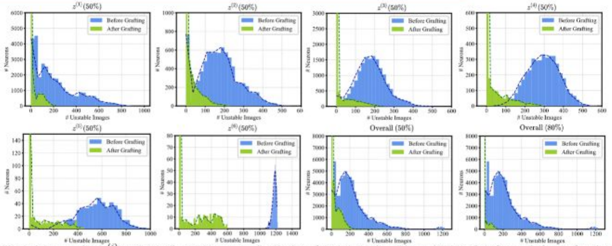


Figure 3. Layer-wise ($z^{(i)}$) and overall unstable neuron distribution of the 7-layer ConvBig on CIFAR-10, before and after performing grafting on 50% or 80% neurons. In specific, the point (m unstable images, n neurons) means that n neurons are unstable for m images.

- ✓ Substantially reduced unstable neurons and tighter bound.
- ✓ Achieving competitive certifiable robustness *without certified robust training*.
- ✓ Scaling up complete verification to large models.

Table 1. **Unstable neuron ratio (UNR %), verified accuracy (VA %), standard accuracy (SA %), PGD-100 robust accuracy (RA %), and average time (s) of FAT trained models w./w.o. grafting on MNIST, SVHN, and CIFAR-10.** α,β-CROWN, a SOTA complete verifier is used to compute VA. The target ℓ_∞ norm perturbation is $\epsilon = \frac{2}{255}$ except for MNIST. "OOM" indicates that DNNs have too many unstable neurons and the verifier is unable to load it with 48 GB GPU memory, leading to "∞" verification time and a null VA ("-").

FAT ($\epsilon = \frac{2}{255}$)	(ConvBig, MNIST w. $\epsilon = 0.1$)					(ConvBig, SVHN)					(CNN-B, CIFAR-10)				
	UNR	VA	SA	RA	Time	UNR	VA	SA	RA	Time	UNR	VA	SA	RA	Time
Baseline	31.27	0.10	99.29	97.14	262.11	10.78	16.70	89.71	75.74	218.49	15.85	37.40	79.95	62.23	127.50
SAP (Dhillon et al., 2018) (50%)	7.38	4.20	99.22	96.34	292.94	5.65	25.90	89.85	76.03	195.87	6.27	47.30	75.10	58.01	58.98
GAP† (Ye et al., 2020) (50%)	17.29	3.50	99.19	96.46	295.21	6.14	26.20	90.09	77.28	195.78	10.22	42.50	79.05	61.81	103.03
Hydra‡ (Sehwag et al., 2020) (50%)	15.39	12.70	98.90	95.22	269.71	5.04	26.60	81.28	62.92	172.98	6.28	44.40	72.99	55.55	59.99
Random Grafting (50%)	17.16	12.00	98.93	95.38	273.94	6.13	37.40	87.37	73.27	150.23	9.07	42.50	75.02	57.19	83.25
Grafting (50%)	5.85	82.30	98.68	92.73	40.21	3.11	57.80	78.75	63.90	16.68	5.36	50.40	74.08	58.76	39.32
Grafting (30%)	10.43	59.40	99.13	95.24	137.40	5.45	56.80	80.71	66.05	31.76	7.15	49.00	77.10	60.87	64.80
Grafting (80%)	4.04	82.40	98.63	92.71	39.64	1.63	58.70	78.56	63.91	12.93	1.87	44.40	61.20	48.34	15.25

FAT ($\epsilon = \frac{2}{255}$)	(ResNet-4B, CIFAR-10)					(ConvBig, CIFAR-10)					(ConvHuge, CIFAR-10)				
	UNR	VA	SA	RA	Time	UNR	VA	SA	RA	Time	UNR	VA	SA	RA	Time
Baseline	19.94	0.80	76.69	60.14	45.56	17.75	1.30	84.90	68.10	121.61	OOM	-	90.68	73.57	∞
SAP (Dhillon et al., 2018) (50%)	6.18	21.70	49.03	38.30	137.77	5.54	25.80	65.08	50.45	156.28	8.52	2.00	80.29	60.29	181.06
GAP† (Ye et al., 2020) (50%)	13.67	5.10	68.42	53.43	239.14	10.97	1.10	81.91	64.50	190.42	7.43	1.00	86.38	67.91	111.77
Hydra‡ (Sehwag et al., 2020) (50%)	9.52	15.10	42.01	31.27	162.34	11.10	1.10	67.97	47.77	297.19	9.88	1.00	70.68	48.81	291.00
Random Grafting (50%)	13.59	7.40	69.56	52.53	267.74	12.23	3.90	79.33	60.92	285.71	11.34	1.00	84.47	64.76	206.97
Grafting (50%)	6.03	38.10	60.13	46.12	42.83	4.32	39.12	62.23	47.73	42.80	4.41	28.30	62.62	49.37	155.78
Grafting (30%)	12.89	24.50	63.71	49.16	153.69	10.30	27.30	71.97	54.97	159.74	OOM	-	90.19	72.34	∞
Grafting (80%)	2.91	39.70	57.64	44.61	25.16	1.89	41.00	55.20	44.27	10.87	0.17	32.30	40.80	33.43	4.06

† The heuristic of activation gradient magnitude (Ye et al., 2020) is utilized to guide activation pruning.
‡ Based on the official implementation of Sehwag et al. (2020), we extend the original sparse mask learning to activation sparsification.

Contact: {tianlong.chen, zhenyu.zhang, atlaswang}@utexas.edu, {huan}@huan-zhang.com, {chang87}@ucsb.edu, {liusiji5}@msu.edu, {pin-yu.chen}@ibm.com

Q&A