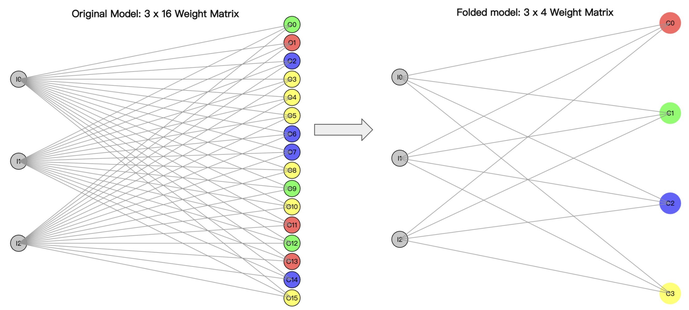
Forget the Data and Fine-Tuning! Just Fold the Network to Compress

Dong Wang[1,*] Haris Šikić [1,*] Lothar Thiele [3] Olga Saukh [1,2]

[1]Graz University of Technology [2]Complexity Science Hub Vienna [3]ETH Zurich [*]Equal contribution

Q: Can you compress a model without data or fine-tuning?



A: Yes, we can. Just fold it!

Motivation

- Resource constraints limit real-world deployment of large models.
- Existing methods often require data and fine-tuning to recover performance.
- SGD-trained models show correlated patterns in weight space.

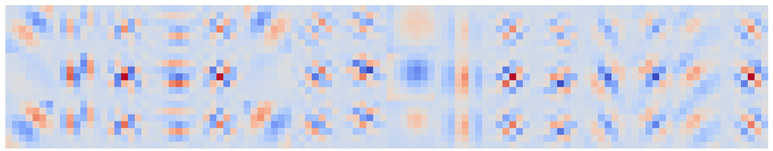


Figure 1. Similar patterns in weight map of *conv1* layer in ResNet18 pre-trained on ImageNet.

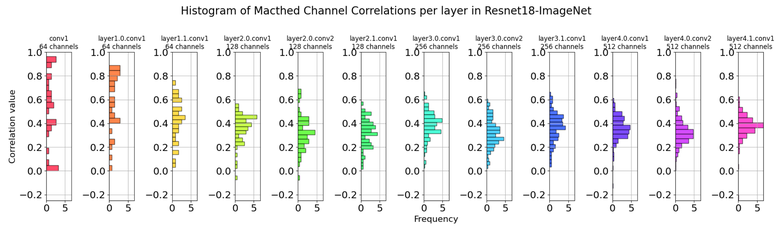


Figure 2. Layer-wise correlation between matched channels in ResNet18 trained on CIFAR10.

Keypoint → fold similar structures instead of zeroing them out.

Model Folding

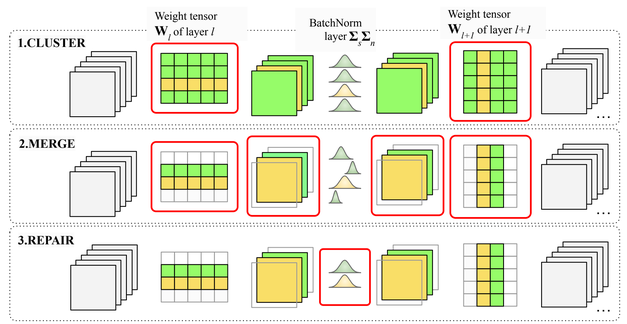


Figure 3. Model Folding Pipeline

1. **Cluster similar channels**. Match similar channels via k-means clustering.
2. **Merge channels**. Merge a channel cluster by averaging the aligned weights
3. **REPAIR BN statistics**. Correct data statistics in a compressed model.
4. **No fine-tuning**.

Data-free repair

However, variance collapse or explosion leads to suboptimal performance.

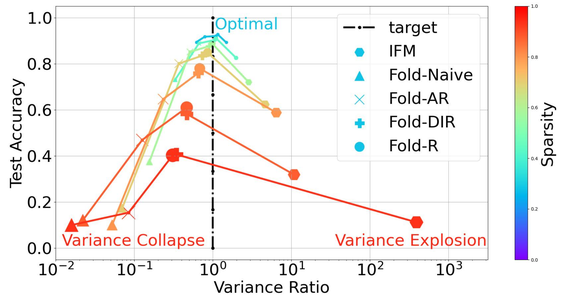


Figure 4. Accuracy VS data variance ratio

we proposed three **repair** methods for BN layers.

- **Fold-AR**: Folding with approximate REPAIR.
- **Fold-DIR**: Correcting data statistics with deep inversion.
- **Fold-R**: Folding with data-driven REPAIR.

Comparison with model pruning

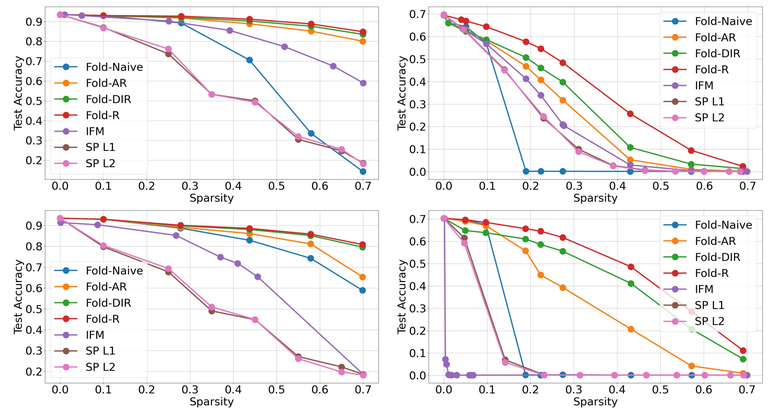


Figure 5. **Comparison with IFM and structured magnitude pruning.** Model folding, when tested on ResNet18 (**top row**) and VGG11-BN (**bottom row**) trained on CIFAR10 (**left column**) and ImageNet (**right column**), outperforms IFM with higher sparsity and increasing dataset difficulty.

More results

Model Folding on LLaMA

Prune ratio	Method	Data usage	WikiText2↓	BoolQ	WinoGrande	ARC-e	ARC-c	Average↑
0%	LLaMA-7B	/	5.68	75.05	69.93	75.34	41.89	65.55
20%	Magnitude Prune	/	36136	43.21	49.40	27.23	21.59	35.36
20%	LLM-Pruner	Gradients	10.53	59.39	61.33	59.18	37.18	54.27
20%	FLAP	Calibration	6.87	69.63	68.35	69.91	39.25	61.79
20%	Wanda_sp	Calibration	8.22	71.25	67.09	71.09	42.58	63.00
20%	SliceGPT	Calibration	7.00	57.80	67.96	62.67	36.01	56.11
20%	ShortGPT	Calibration	15.48	62.17	67.40	58.88	31.91	55.09
20%	Model Folding	/	13.33	62.29	62.19	49.83	26.37	50.17

Table 1. **Performance of structured pruning methods on LLaMA-7B without post-tuning**, showing perplexity on WikiText2 and zero-shot performance across tasks.

Model Folding on edge devices

Sparsity	10%			25%			50%			70%		
	Runtime	RAM	Flash	Runtime	RAM	Flash	Runtime	RAM	Flash	Runtime	RAM	Flash
NVIDIA Jetson Nano	2ms	59.5K	3.4M	2ms	55.7K	2.8M	1ms	48.0K	1.9M	1ms	36.5K	1.2M
ESP-EYE	2591ms	59.5K	3.4M	1868ms	55.7K	2.8M	1532ms	48.0K	1.9M	1186ms	36.5K	1.2M
Arduino Nano 33 BLE Sense	6831ms	59.5K	3.4M	3726ms	55.7K	2.8M	4218ms	48.0K	1.9M	2969ms	36.5K	1.2M

Table 2. **Performance and resource usage at various sparsity levels across devices**, with detailed breakdowns for runtime (ms), RAM usage (K), and Flash storage usage (M).