Standardizing Evaluation of Neural Network Pruning

Jose Javier Gonzalez

Davis Blalock

John V. Guttag

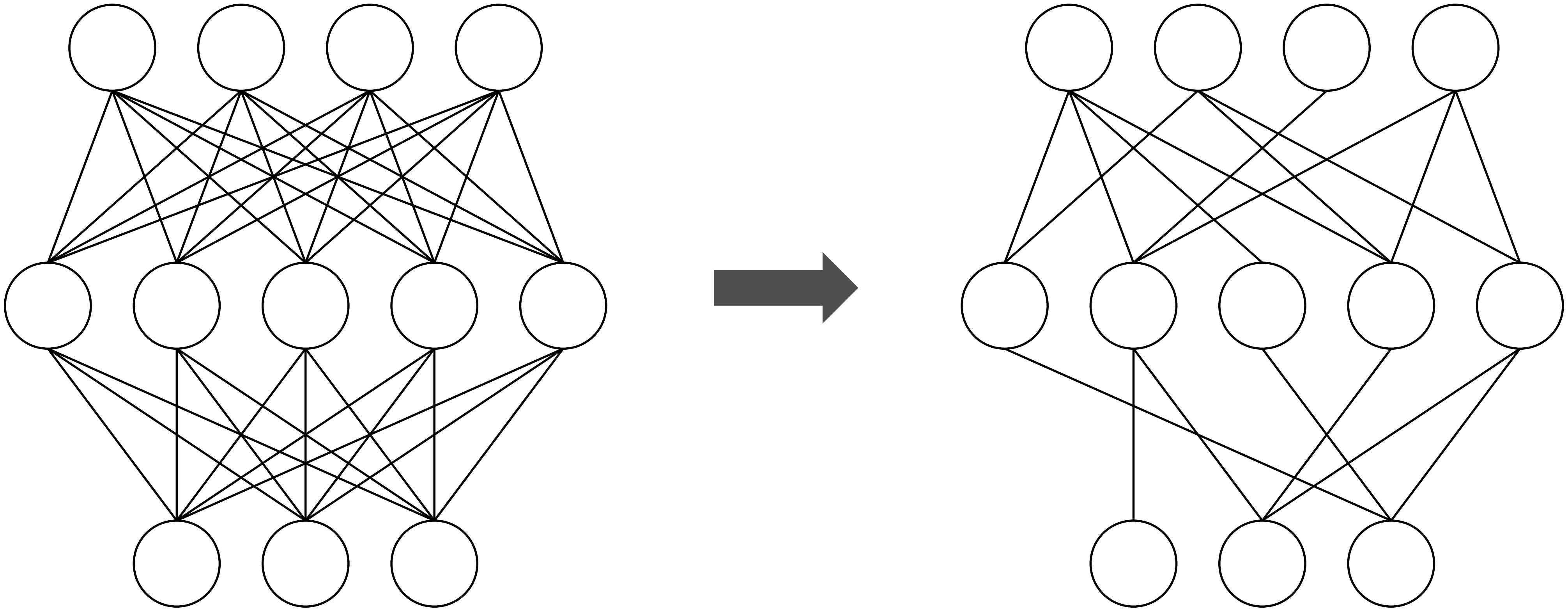
ShrinkBench

ShrinkBench:
Open source PyTorch library to facilitate development and standardized evaluation of neural network pruning methods

- Rapid prototyping of NN pruning methods
- Makes it easy to use standardized datasets, pretrained models and finetuning setups
- Controls for potential confounding factors

Neural Network Pruning

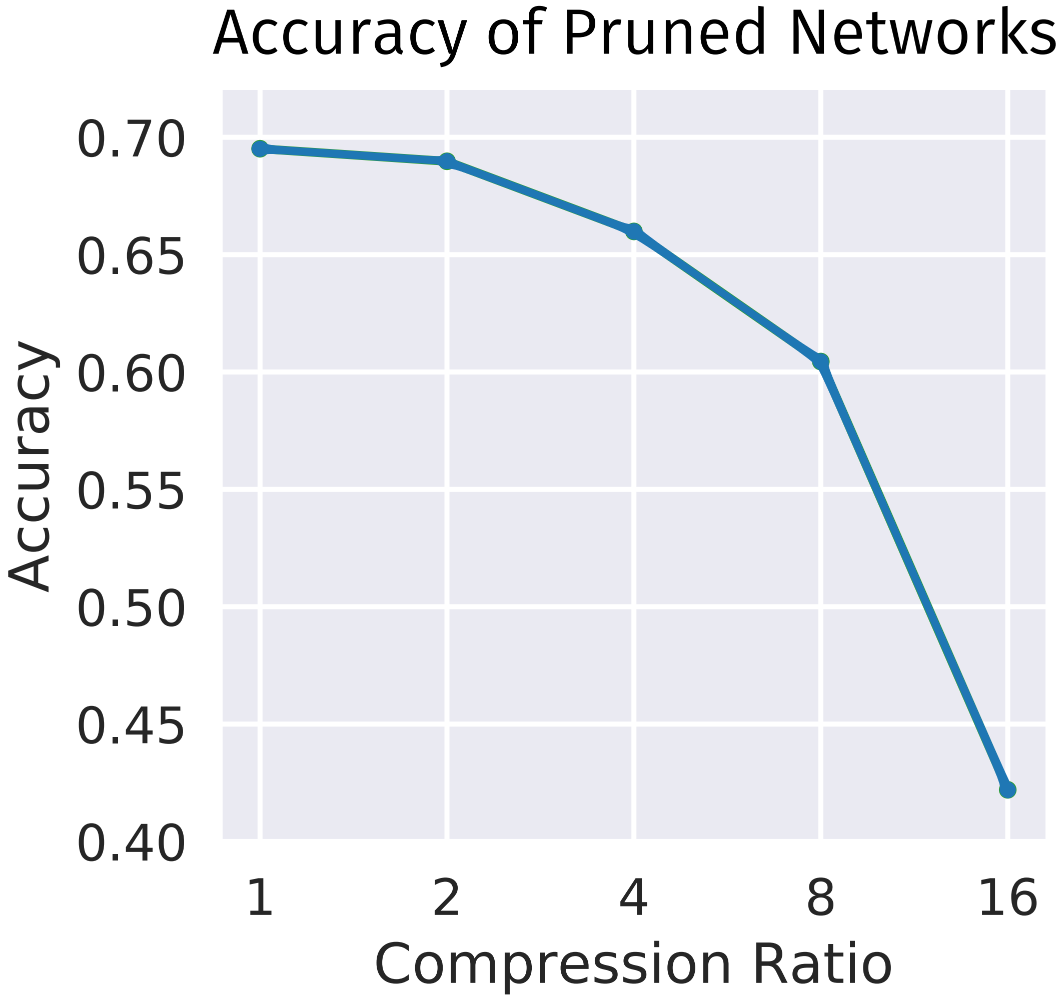
- Pretrained networks are often quite accurate but large
- *Pruning*: Systematically remove parameters from a network



Neural Network Pruning

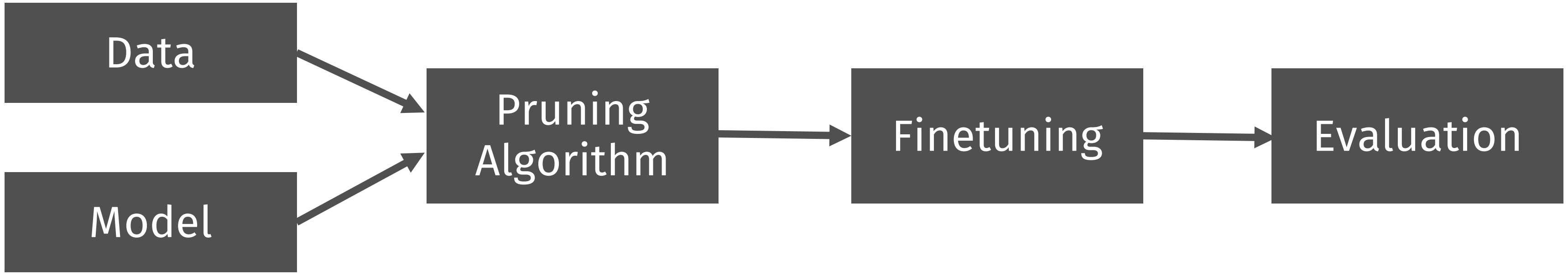
- Goal: Reduce size of network as much as possible with minimal drop in accuracy

- Often requires finetuning afterwards



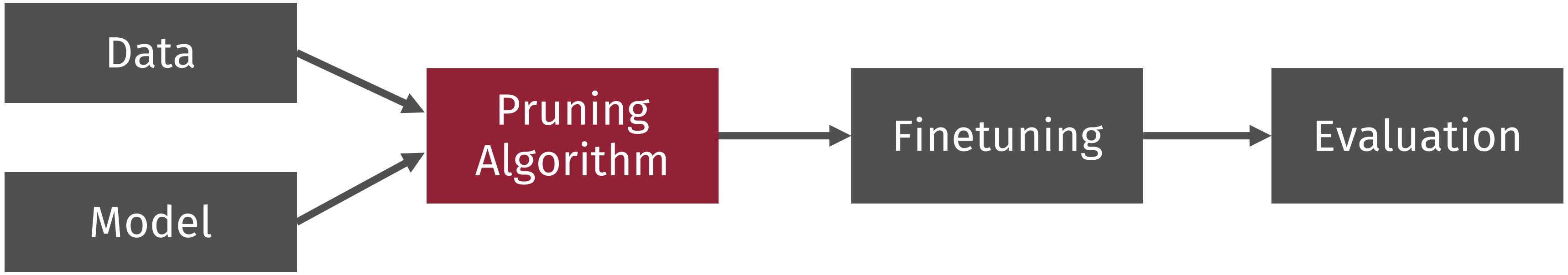
Traditional Pipeline

Need a whole pipeline for performing experiments



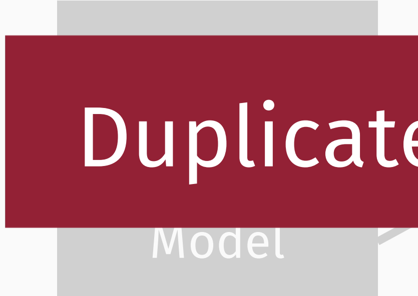
Traditional Pipeline

But only the pruning algorithm usually changes



Traditional Pipeline

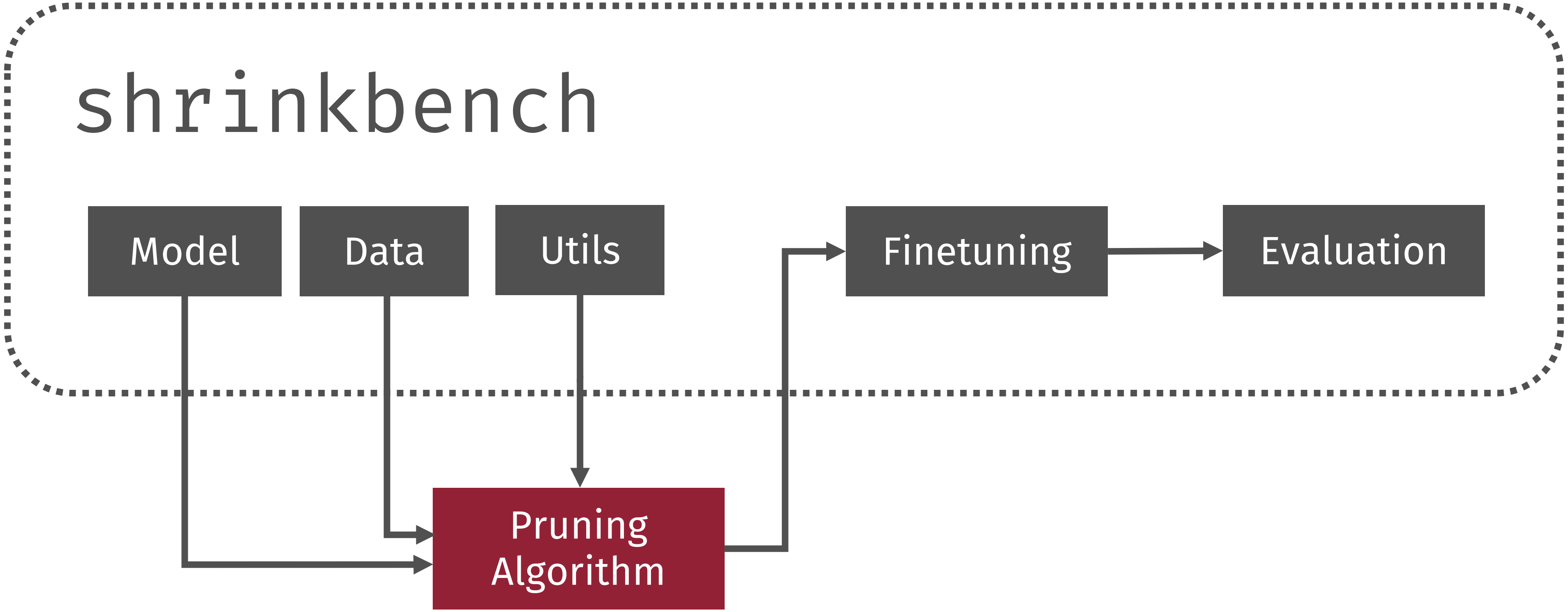
But only the pruning algorithm usually changes



Duplicate effort & confounding variables

ShrinkBench

Library to facilitate standardized evaluation of pruning methods



ShrinkBench

- Provides standardized datasets, pretrained models, and evaluation metrics
- Simple and generic parameter masking API
- Measures nonzero parameters, activations, and FLOPs
- Controlled experiments show the need for standardized evaluation

Towards Standardization

But how do we standardize?

Towards Standardization

But how do we standardize?

- Standardized datasets.
  Widely adopted datasets, representative of real-world tasks
- Standardized architectures
  With reproducibility record, matched in complexity to the chosen dataset

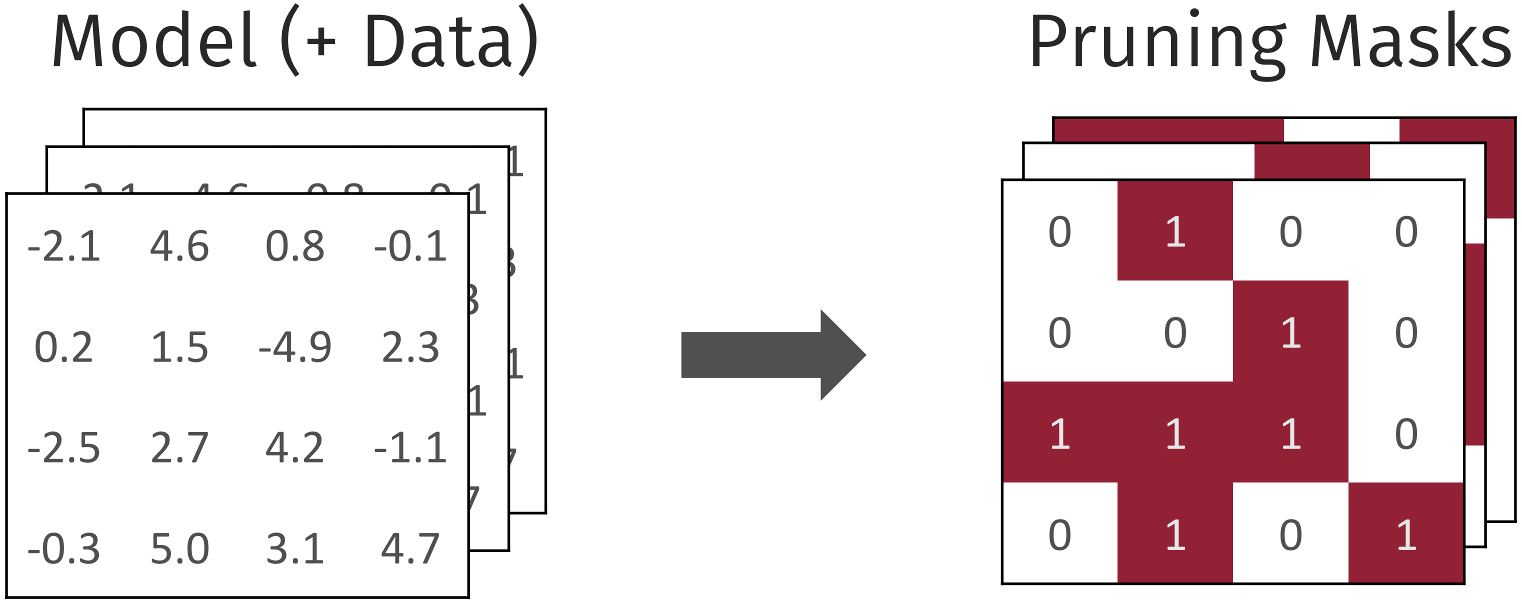
Towards Standardization

But how do we standardize?

- Standardized datasets.
  Widely adopted datasets, representative of real-world tasks
- Standardized architectures
  With reproducibility record, matched in complexity to the chosen dataset
- Pretrained models
  Even for a fixed architecture and dataset, exact weights may affect results
- Finetuning setup
  We want improvement from pruning, not from better hyperparameters

Masking API

We can capture an arbitrary removal pattern using binary masks



Model (+ Data)

-2.1	4.6	0.8	-0.1
0.2	1.5	-4.9	2.3
-2.5	2.7	4.2	-1.1
-0.3	5.0	3.1	4.7

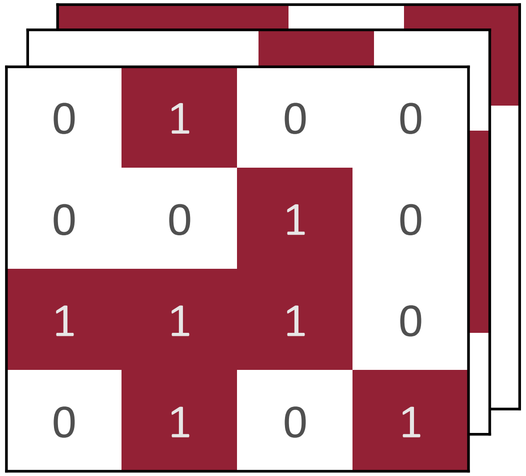
Pruning Masks

0	1	0	0
0	0	1	0
1	1	1	0
0	1	0	1

Masks → Accuracy

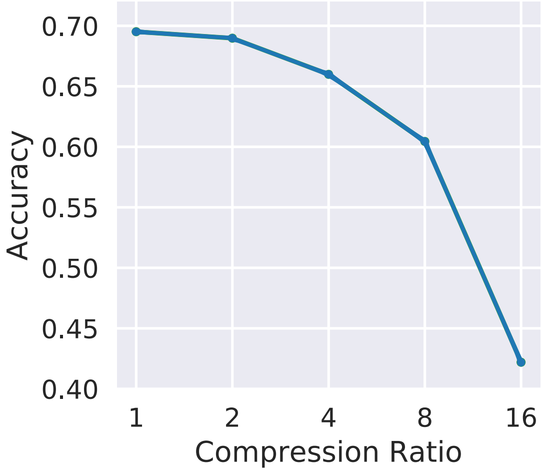
Given a pruning method in terms of masks, ShrinkBench finetunes the model and systematically evaluates it

Pruning Masks



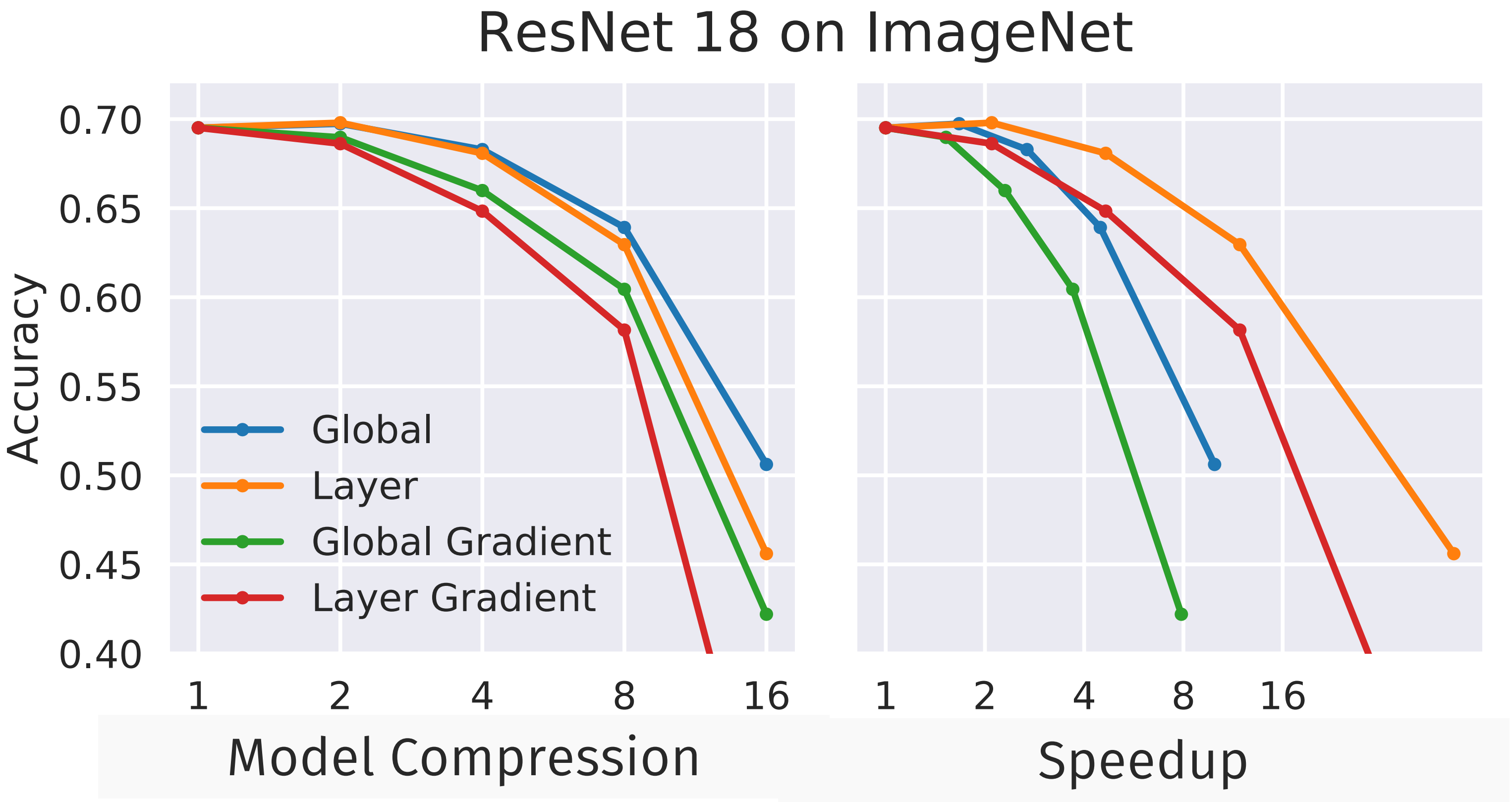
0	1	0	0
0	0	1	0
1	1	1	0
0	1	0	1

Accuracy Curve



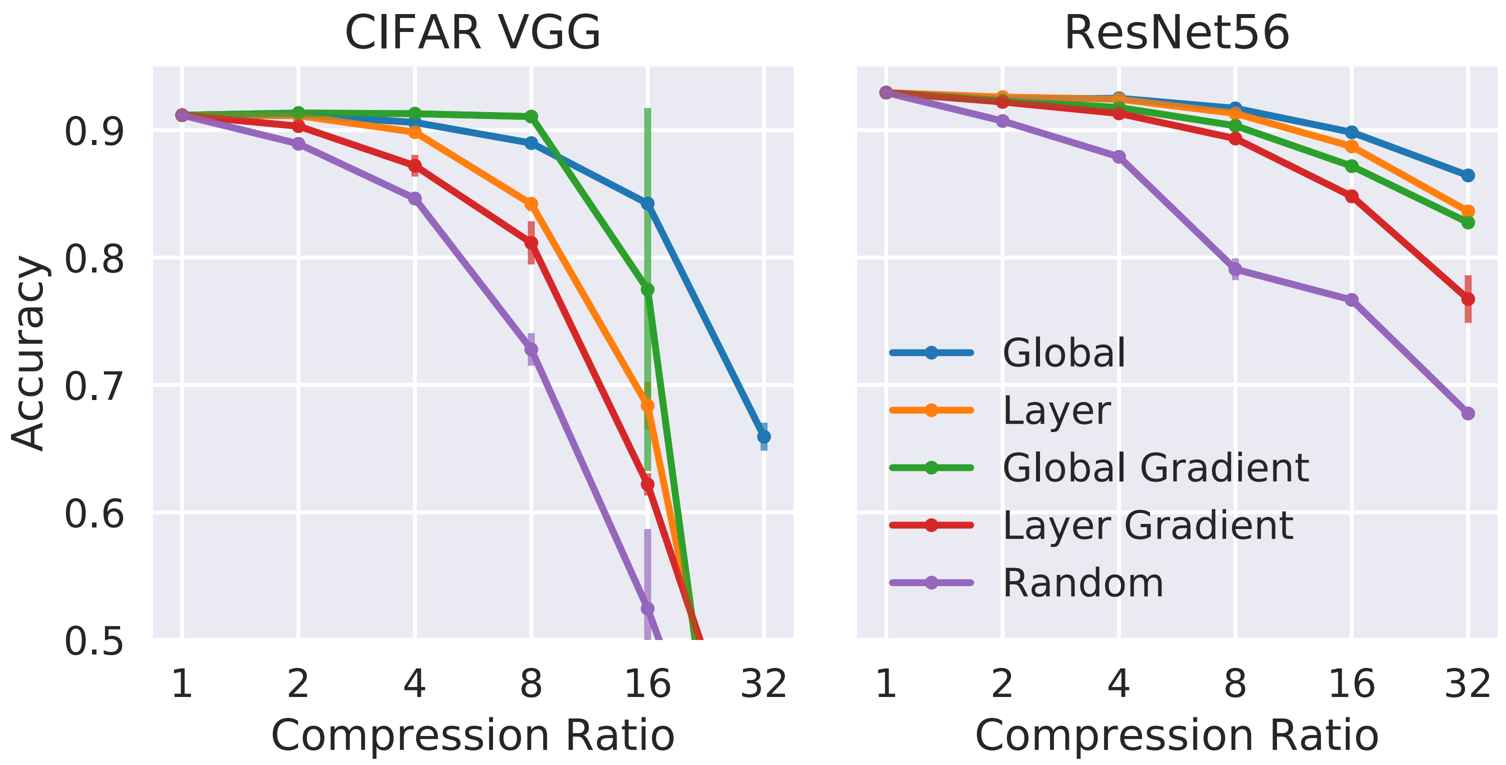
ShrinkBench Results I

- ShrinkBench returns both compression & speedup since they interact differently with pruning



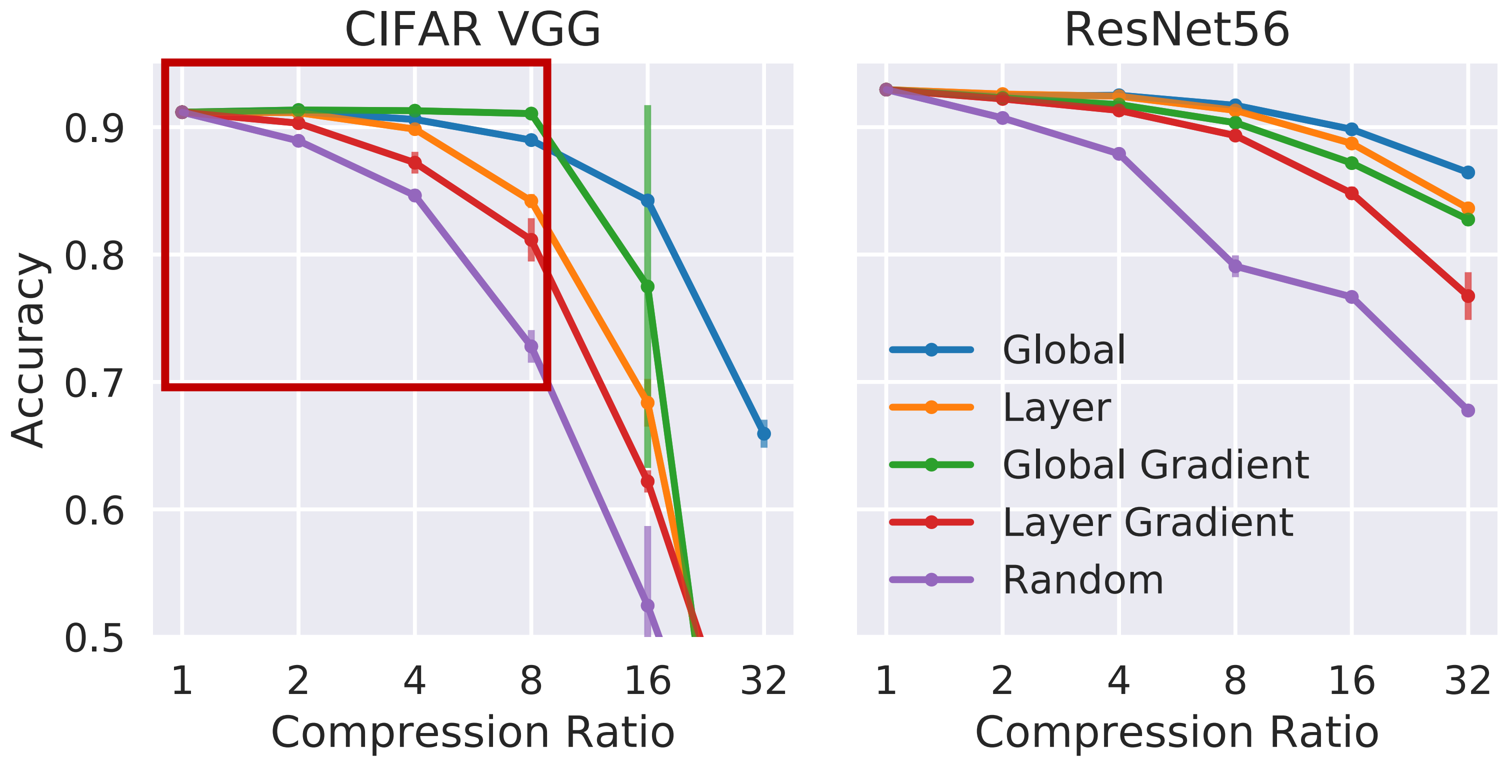
ShrinkBench Results II

- ShrinkBench evaluates with varying compression and with several (dataset, architecture) combinations



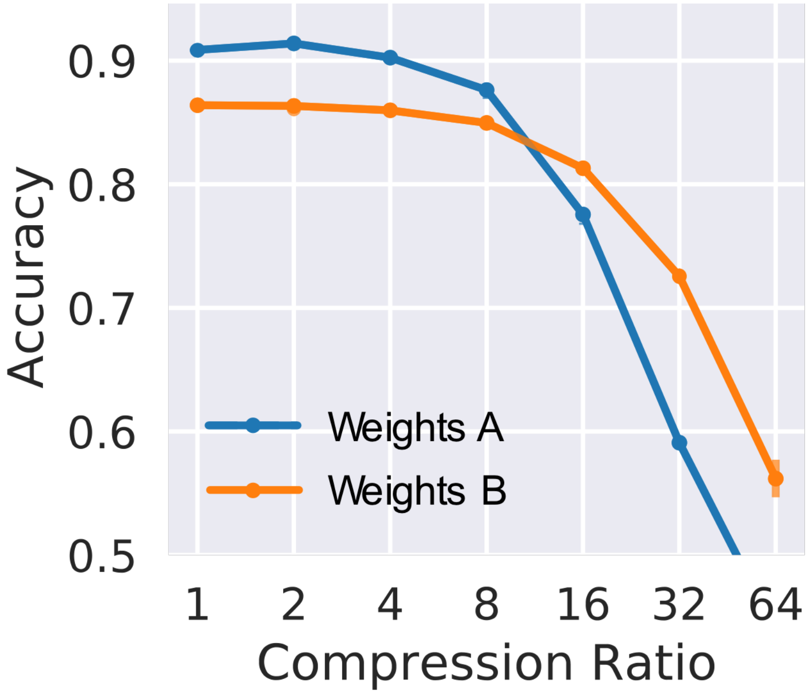
ShrinkBench Results II

- ShrinkBench evaluates with varying compression and with several (dataset, architecture) combinations



ShrinkBench Results III

- ShrinkBench controls for confounding factors such as pretrained weights or finetuning hyperparemeters



Summary

- ShrinkBench – an open source library to facilitate development and standardized evaluation of neural network pruning methods

- Our controlled experiments across hundreds of models demonstrate the need for standardized evaluation.

https://shrinkbench.github.io