ICML International Conference On Machine Learning

EFFICIENT AND INTELLIGENT COMPUTING

DepthShrinker: A New Compression Paradigm Towards Boosting Real-Hardware Efficiency of Compact Neural Networks

ICML 2022

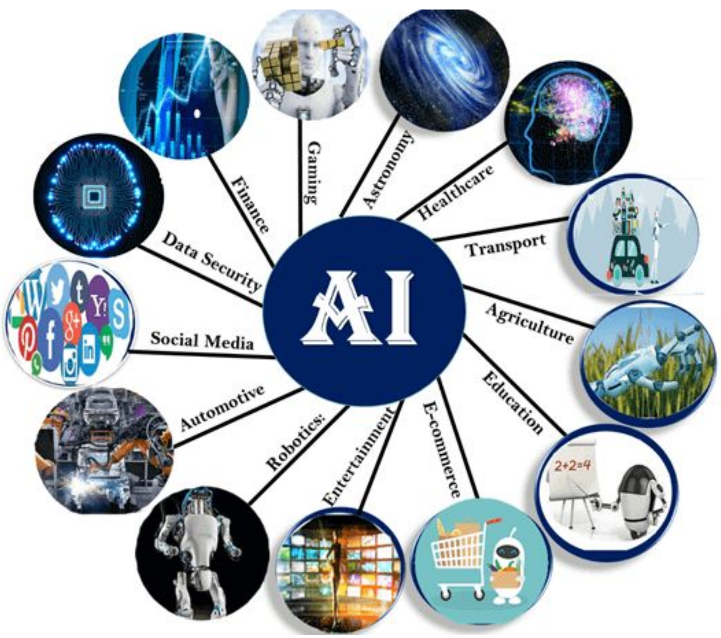
Yonggan Fu, Haichuan Yang, Jiayi Yuan, Meng Li, Cheng Wan, Raghuraman Krishnamoorthi, Vikas Chandra, Yingyan Lin

Meta

RICE Electrical and Computer Engineering

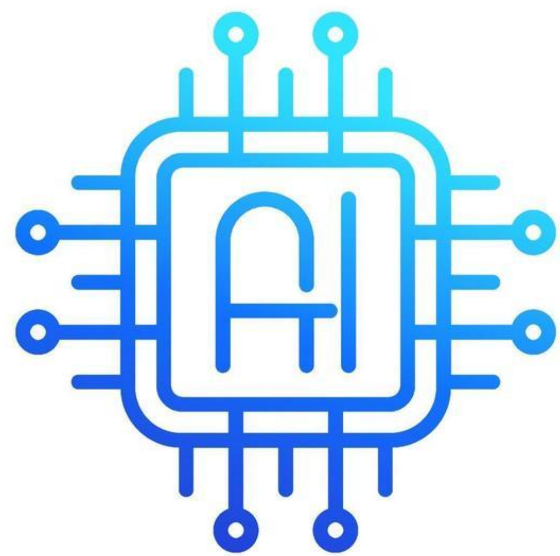
Background: Demanding Efficient ML

- **A growing demand**: Accelerate deep neural networks (DNNs) on real-world devices



Complex AI Models

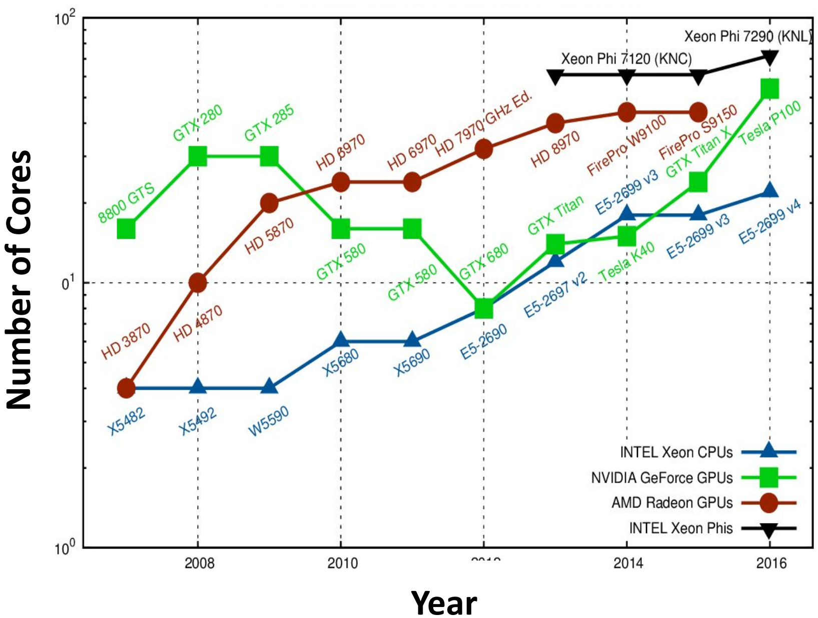
Bridge the Gap

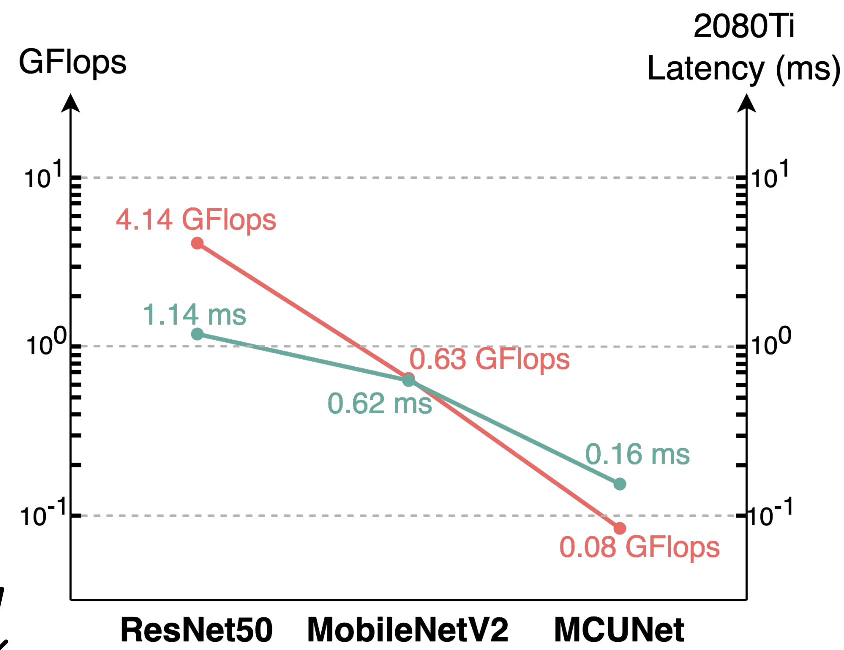


Computing Platforms with Limited Resources

Challenge: A Dilemma

- **A dilemma** between the trends of efficient DNN design vs. modern computing platform advances





Modern computing platforms: A higher degree of parallel computing [*K. Rupp*]

Efficient DNNs: Adopt Lightweight operators featuring low utilization

Our Driven Question and Proposed Solution

How can we build efficient/compact DNNs with boosted hardware utilization to harvest more parallelized capability of modern hardware?

Our Driven Question and Proposed Solution

How can we build efficient/compact DNNs with boosted hardware utilization to harvest more parallelized capability of modern hardware?

We propose ***DepthShrinker***:

- ➢ Shrinking consecutive operations into one single *dense* operation

Overview

- Background and Challenge
- **Motivating Profiling**
- The DepthShrinker Framework
- Experimental Results

Motivating Real-device Profiling

- **Goal**: Validate the hw benefits of dense operators

- **Profiling setup**
  - Replace each building block by one dense conv layer
  - Scale channel numbers to ensure the same FLOPs

- **Considered devices**: Desktop + Edge GPUs
  - NVIDIA Tesla V100 GPU
  - NVIDIA RTX 2080Ti GPU
  - Jetson TX2 Edge GPU

Motivating Real-device Profiling

- **Compact models vs. their dense counterparts**
  - Dense convs lead to higher throughputs
    - 3.45x ~ 4.38x on top of the MobileNetV2 family
    - 1.18x ~ 2.43x on top of the ResNet family
    - More notable on Desktop GPUs than Edge ones

Model	GFLOPs	Tesla V100 GPU		RTX 2080Ti GPU		TX2 Edge GPU	
		Original	Dense	Original	Dense	Original	Dense
MobileNetV2	0.33	3088	**12090** (↑**3.91**×)	2364	**9351** (↑**3.96**×)	115	**397** (↑**3.45**×)
MobileNetV2-1.4	0.63	2127	**8846** (↑**4.16**×)	1617	**6869** (↑**4.25**×)	73	**267** (↑**3.66**×)
Efficientnet-Lite0	0.41	2731	**11174** (↑**4.09**×)	2185	**9577** (↑**4.38**×)	98	**360** (↑**3.67**×)
ResNet-50	4.14	1079	**2182** (↑**2.02**×)	874	**1862** (↑**2.13**×)	45	**53** (↑**1.18**×)
ResNet-101	7.88	642	**1509** (↑**2.35**×)	538	**1279** (↑**2.38**×)	28	**44** (↑**1.57**×)
ResNet-152	11.62	449	**1082** (↑**2.41**×)	378	**917** (↑**2.43**×)	19	**30** (↑**1.58**×)

Overview

- Background and Challenge
- Motivating Profiling
- **The DepthShrinker Framework**
- Experimental Results

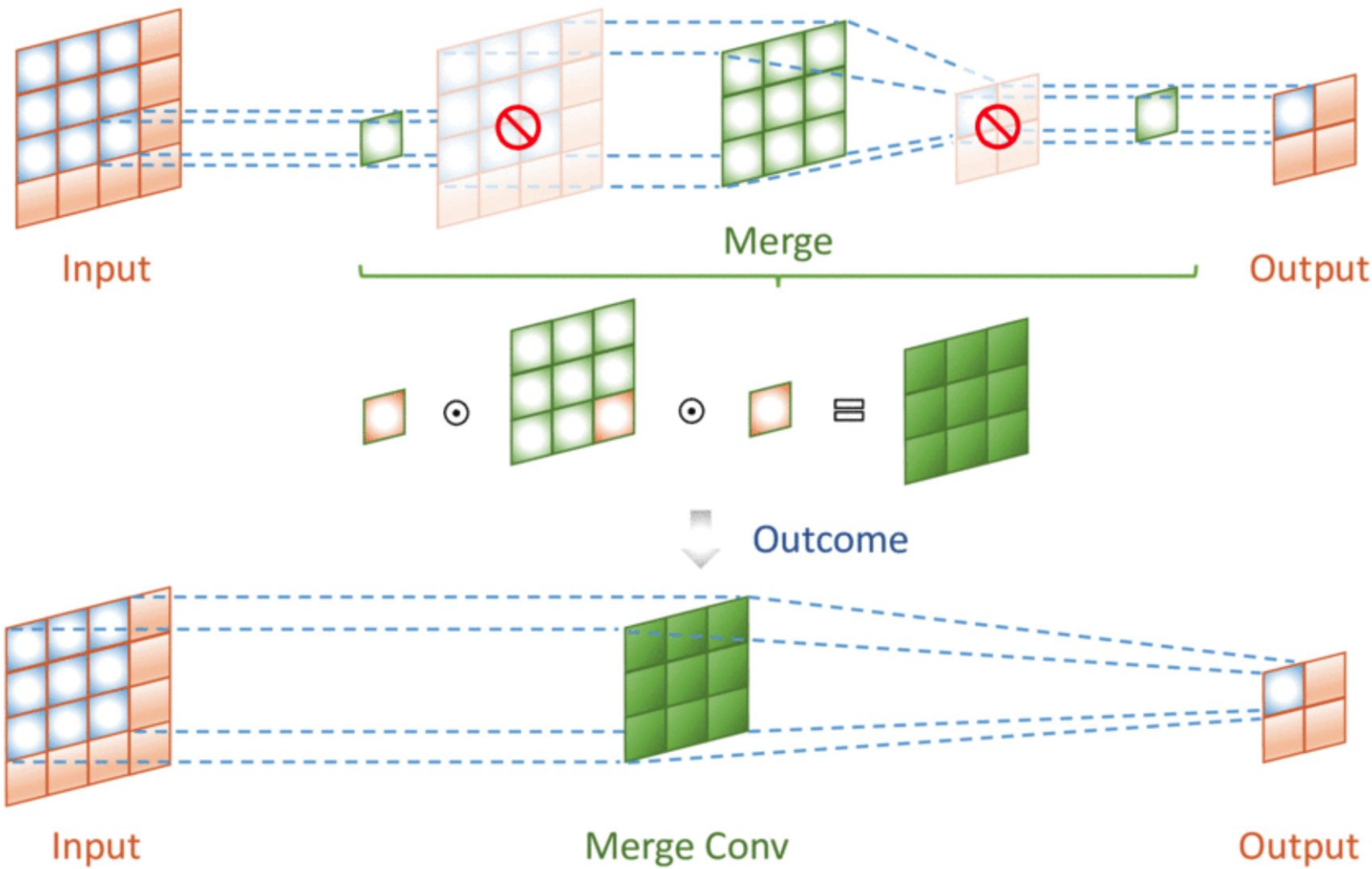
DepthShrinker: The Key Idea

- **Key idea**
  - Remove all the activation functions in one inverted residual block [*CVPR2018, M. Sandler*] and shrink it to one dense conv layer

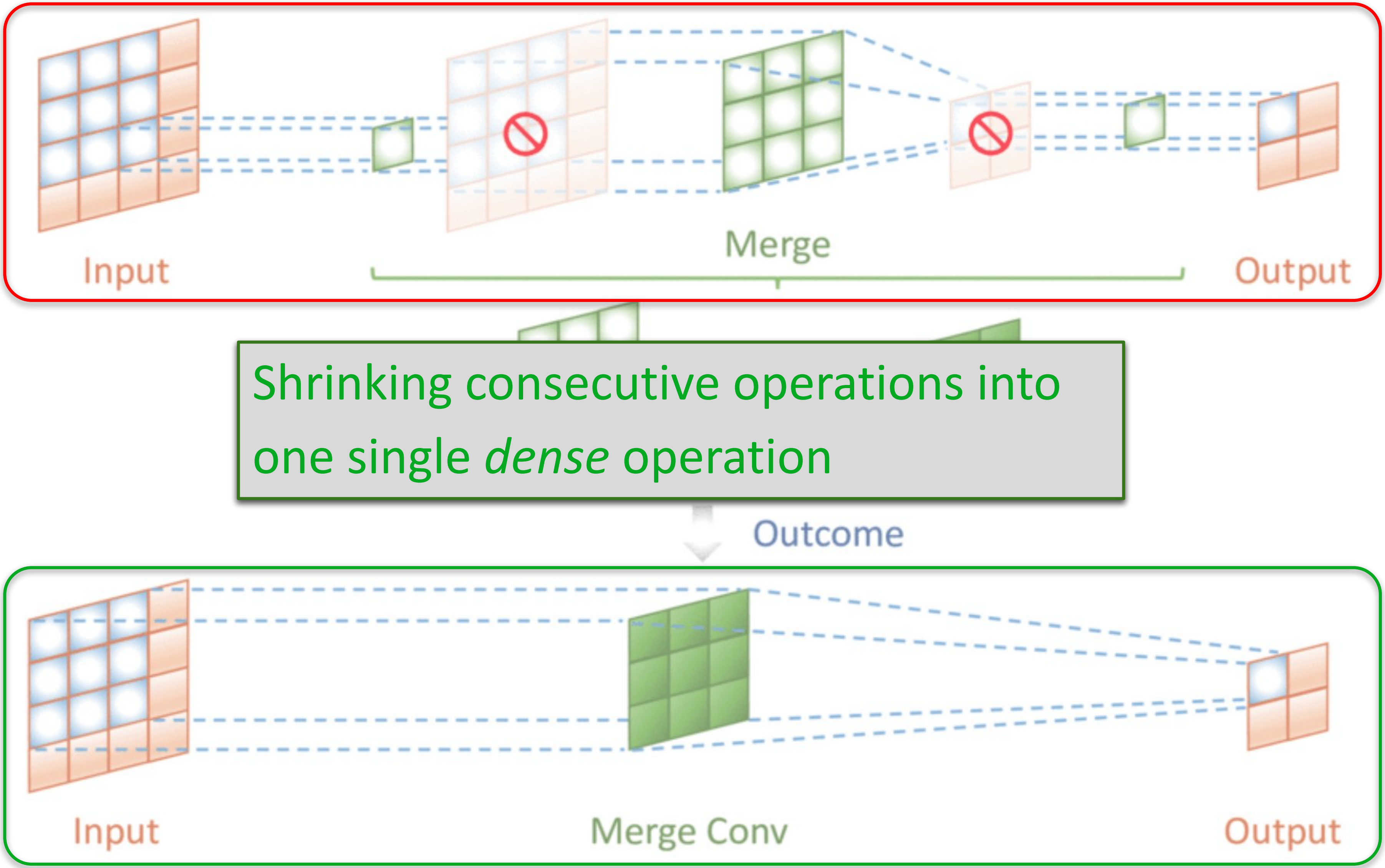
We propose ***DepthShrinker***:

➢ Shrinking consecutive operations into one single *dense* operation

DepthShrinker: The Key Idea



DepthShrinker: The Key Idea



Shrinking consecutive operations into one single *dense* operation

DepthShrinker: Research Questions

We propose ***DepthShrinker***:

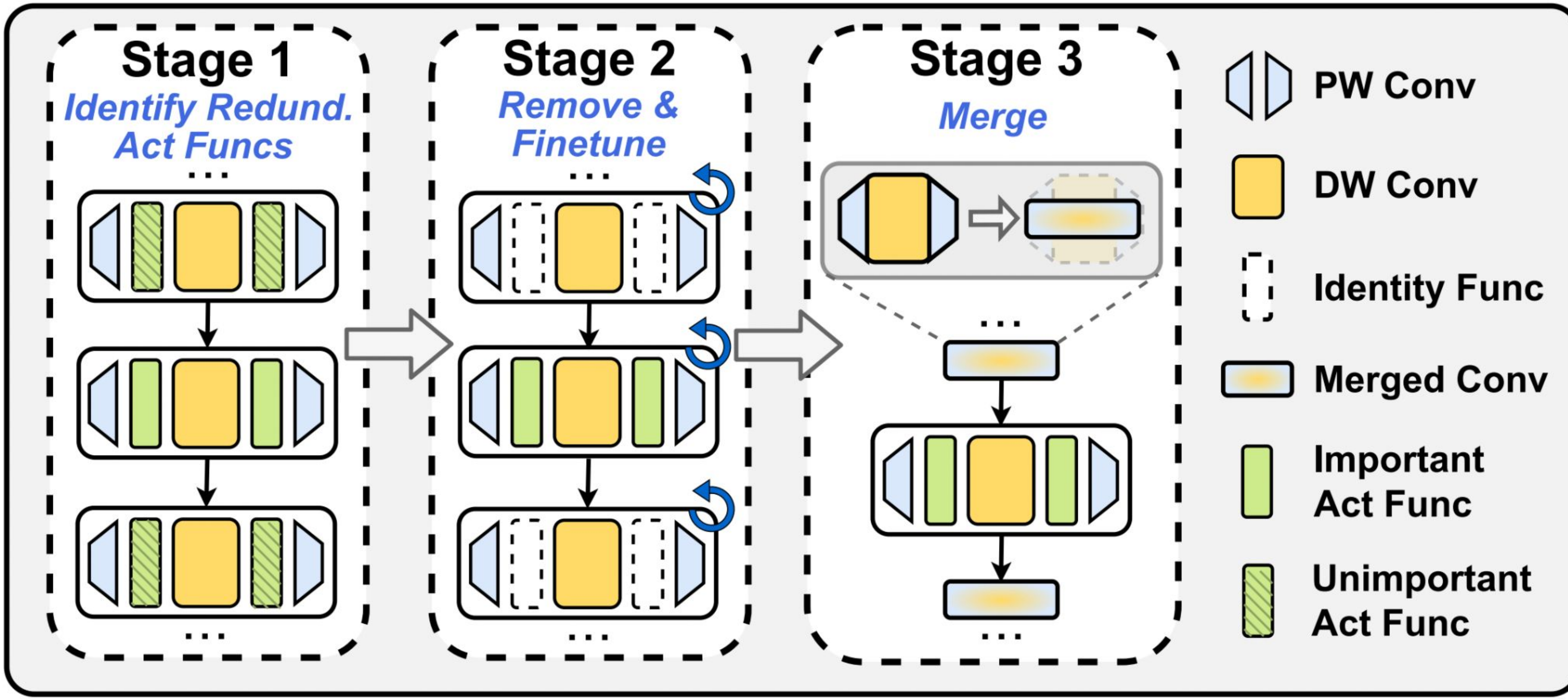
- ➢ Shrinking consecutive operations into one single *dense* operation

Two research questions to achieve ***DepthShrinker***:

- ➢ *Which activation functions to remove?*
- ➢ *How to restore the accuracy after the removal?*

DepthShrinker: Overview

- **DepthShrinker:** A three-stage framework



Identify Redundant Act Funcs

- **Search method**
  - Learnable binary masks on act funcs with *L0* sparsity

$$\underset{\theta,m}{\arg\min} \sum_i \ell(\hat{y}_{\theta,m}(x_i), y_i) \quad \text{s.t.} \quad \|m\|_0 \leqslant k$$

- **Learning scheme: Fully differentiable**
  - Forward: Activate top *k* act funcs
  - Backward: Update both model weights and masks via STE

Identify Redundant Act Funcs

- **Search method**
  - Learnable binary masks on act funcs with *L0* sparsity

$$\arg\min_{\theta,m} \sum_i \ell(\hat{y}_{\theta,m}(x_i), y_i) \quad \text{s.t.} \quad \|m\|_0 \leqslant k$$

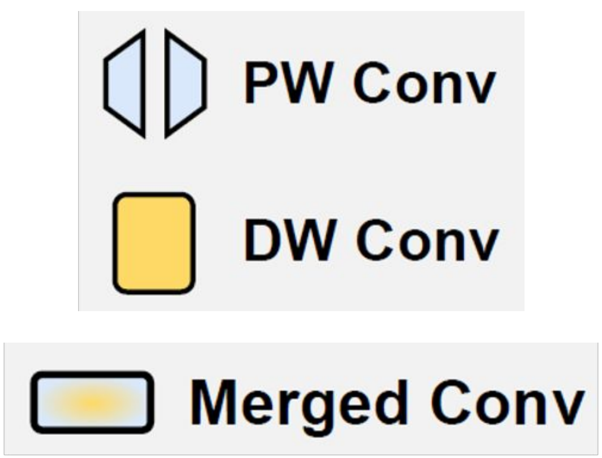
- **Other techniques in implementation**
  - Block-wise shrink: Share *m* for all act funcs in one block
  - Latency-aware decay on *m*: Penalize the importance of latency-bottleneck blocks

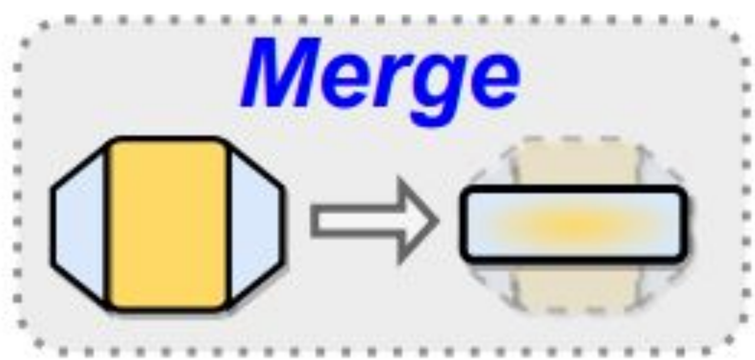
Merge Consecutive Linear OPs

One nice property during merging

- The resulting conv layer is only determined by
  - The number of the input channels in the first conv
  - The number of output channels in the last conv

➢ DepthShrinker can shrink the wide intermediate layers within inverted residual blocks





Overview

- Background and Challenge
- Motivating Profiling
- The DepthShrinker Framework
- **Experimental Results**

Experiment: Setup

- **Models and datasets**
  - MobileNetV2 families @ ImageNet
  - EfficientNet-Lite families @ ImageNet

- **Considered devices**: Desktop + Edge
  - NVIDIA Tesla V100 GPU
  - NVIDIA RTX 2080Ti GPU
  - Jetson TX2 Edge GPU
  - CPU devices: Google Pixel 3 / Raspberry Pi 4

- **Metrics:** Accuracy / Throughput (FPS)

Experiment: Benchmark with Pruning

- **Benchmark with channel-wise pruning**
  - Consistently better Acc-FPS trade-off

Model	Acc (%)	MFLOPs	Tesla V100	RTX 2080Ti	TX2
MBV2-1.4	75.30	630	2127 ($\uparrow$1.00$\times$)	1617 ($\uparrow$1.00$\times$)	73 ($\uparrow$1.00$\times$)
MetaPruning-1.0$\times$	73.20	332	3159 ($\uparrow$1.49$\times$)	2527 ($\uparrow$1.56$\times$)	115 ($\uparrow$1.58$\times$)
MBV2-1.4-DS-A	74.65/75.29/75.65	519	3827 ($\uparrow$1.80$\times$)	2881 ($\uparrow$1.78$\times$)	134 ($\uparrow$1.84$\times$)
MBV2-1.4-DS-B	73.67/74.8/75.13	502	4778 ($\uparrow$2.25$\times$)	3356 ($\uparrow$2.08$\times$)	163 ($\uparrow$2.23$\times$)
MBV2-1.4-DS-C	73.38/74.55/74.91	492	4597 ($\uparrow$2.16$\times$)	3537 ($\uparrow$2.19$\times$)	159 ($\uparrow$2.18$\times$)
Uniform-0.65$\times$	67.20	182	4004 ($\uparrow$1.88$\times$)	3147 ($\uparrow$1.95$\times$)	161 ($\uparrow$2.21$\times$)
MetaPruning-0.65$\times$	71.70	160	4336 ($\uparrow$2.04$\times$)	3691 ($\uparrow$2.28$\times$)	179 ($\uparrow$2.45$\times$)
MBV2-1.4-DS-D	72.51/73.93/74.50	484	5560 ($\uparrow$2.61$\times$)	3926 ($\uparrow$2.43$\times$)	184 ($\uparrow$2.52$\times$)
MBV2-1.4-DS-E	72.20/73.85/74.43	474	5317 ($\uparrow$2.50$\times$)	4175 ($\uparrow$2.58$\times$)	179 ($\uparrow$2.45$\times$)
Uniform-0.35$\times$	54.60	68	6266 ($\uparrow$2.95$\times$)	5607 ($\uparrow$3.47$\times$)	263 ($\uparrow$3.60$\times$)
MetaPruning-0.35$\times$	64.50	52	7044 ($\uparrow$3.31$\times$)	6938 ($\uparrow$4.29$\times$)	377 ($\uparrow$5.16$\times$)
MBV2-1.4-DS-F	67.56/69.04/70.13	415	10804 ($\uparrow$5.08$\times$)	7687 ($\uparrow$4.75$\times$)	344 ($\uparrow$4.71$\times$)

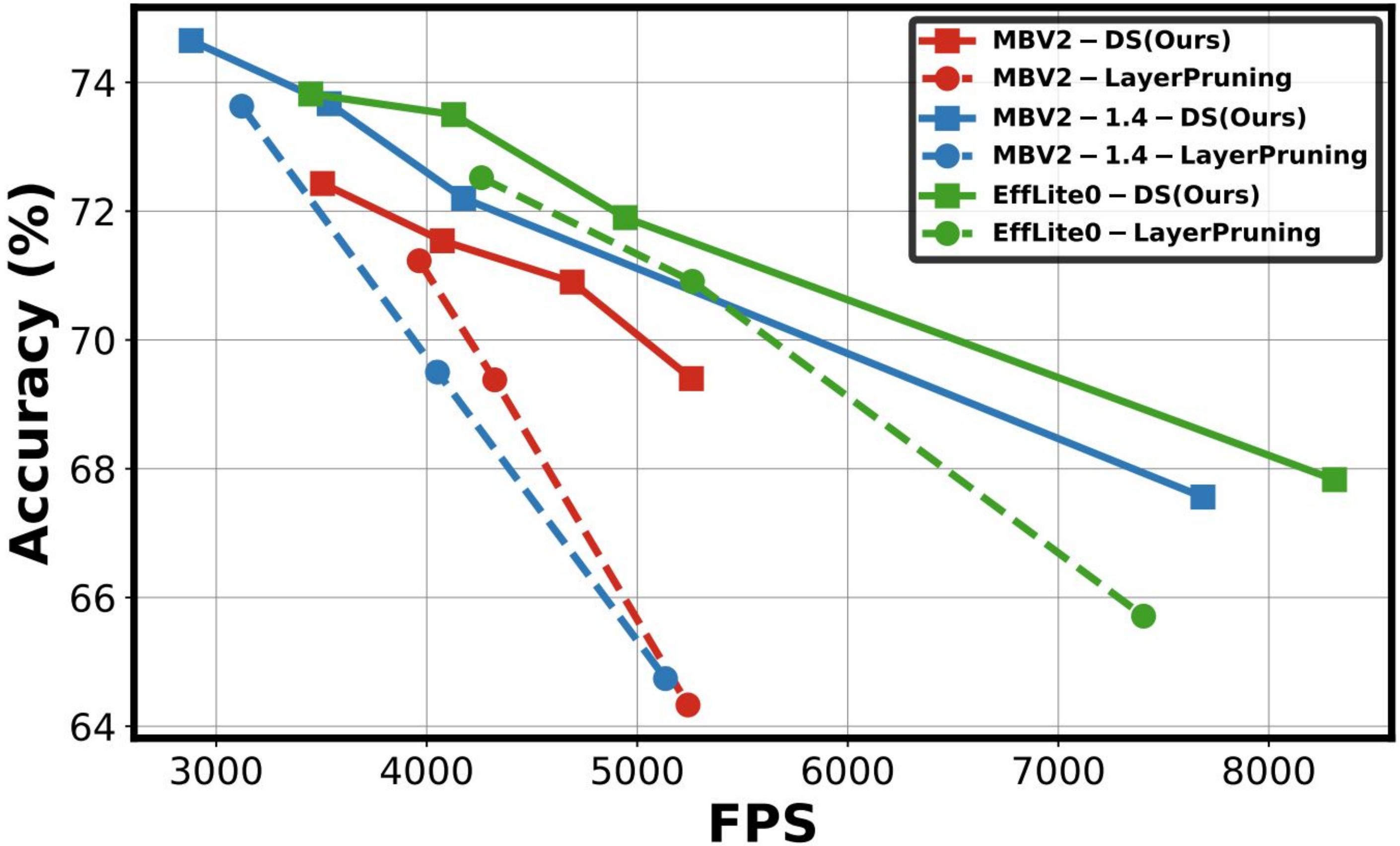
Experiment: Benchmark with Pruning

- **Benchmark with channel-wise pruning**
  - Better scalability to extremely efficient cases
    - A 3.06% higher acc with 1.53x FPS

Model	Acc (%)	MFLOPs	Tesla V100	RTX 2080Ti	TX2
MBV2-1.4	75.30	630	2127 (↑1.00×)	1617 (↑1.00×)	73 (↑1.00×)
MetaPruning-1.0×	73.20	332	3159 (↑1.49×)	2527 (↑1.56×)	115 (↑1.58×)
MBV2-1.4-DS-A	74.65/75.29/75.65	519	3827 (↑1.80×)	2881 (↑1.78×)	134 (↑1.84×)
MBV2-1.4-DS-B	73.67/74.8/75.13	502	4778 (↑2.25×)	3356 (↑2.08×)	163 (↑2.23×)
MBV2-1.4-DS-C	73.38/74.55/74.91	492	4597 (↑2.16×)	3537 (↑2.19×)	159 (↑2.18×)
Uniform-0.65×	67.20	182	4004 (↑1.88×)	3147 (↑1.95×)	161 (↑2.21×)
MetaPruning-0.65×	71.70	160	4336 (↑2.04×)	3691 (↑2.28×)	179 (↑2.45×)
MBV2-1.4-DS-D	72.51/73.93/74.50	484	5560 (↑2.61×)	3926 (↑2.43×)	184 (↑2.52×)
MBV2-1.4-DS-E	72.20/73.85/74.43	474	5317 (↑2.50×)	4175 (↑2.58×)	179 (↑2.45×)
Uniform-0.35×	54.60	68	6266 (↑2.95×)	5607 (↑3.47×)	263 (↑3.60×)
MetaPruning-0.35×	64.50	52	7044 (↑3.31×)	6938 (↑4.29×)	377 (↑5.16×)
MBV2-1.4-DS-F	67.56/69.04/70.13	415	10804 (↑5.08×)	7687 (↑4.75×)	344 (↑4.71×)

Experiment: Benchmark with Pruning

- **Benchmark with layer-wise pruning**
  - Much better acc under large compression ratios
    - Shrinking is a soft version of “hard pruning”



DepthShrinker: A New Compression Paradigm Towards Boosting Real-Hardware Efficiency of Compact Neural Networks

ICML 2022

The work is supported by the National Science Foundation (NSF) through the SCH program, the NeTS program, and the MLWiNS program.