Towards Efficient AI with Tensor Decomposition and Optimization

March 6, 2023

miao.yin@rutgers.edu

Yin, Miao, et al. "Towards efficient tensor decomposition-based DNN model compression with optimization framework." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021.
Yin, Miao, et al. "HODEC: Towards Efficient High-Order DEcomposed Convolutional Neural Networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022.

Agenda

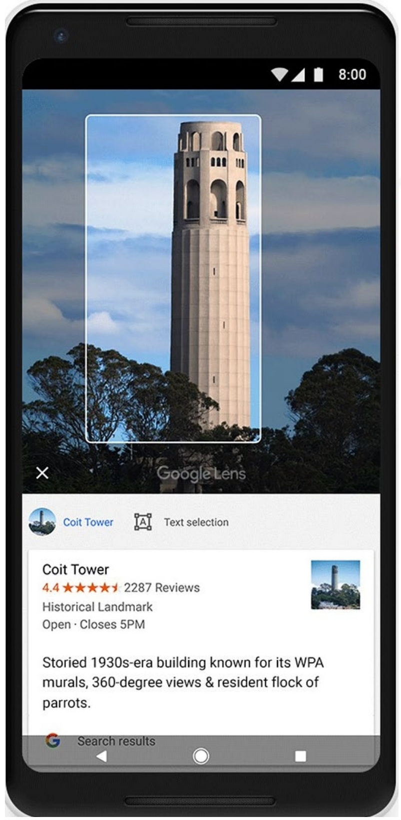
- Background
- Optimization-based Compression
- Efficient Tensor Train-based Convolution
- Experimental Results & Summary

Agenda

- **Background**
- Optimization-based Compression
- Efficient Tensor Train-based Convolution
- Experimental Results & Summary

AI is Changing Our Lives

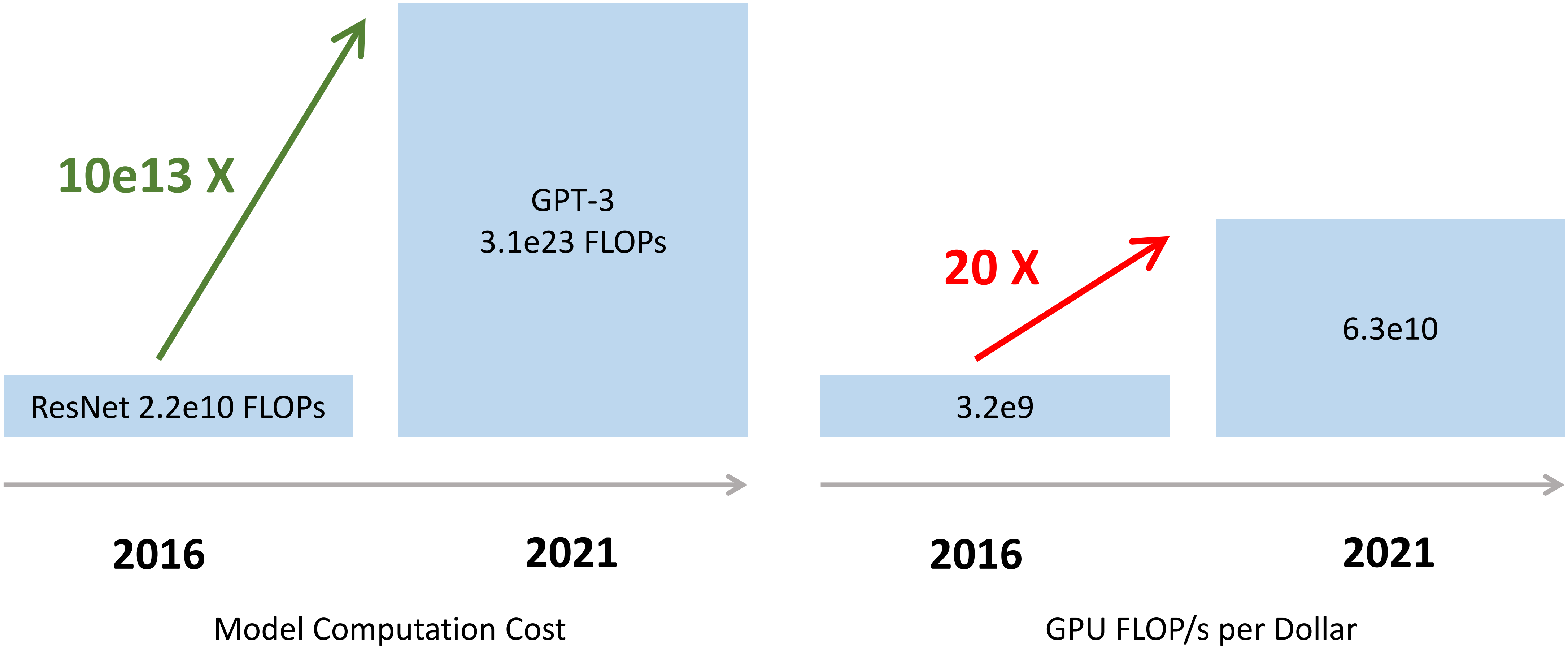
Self-driving Cars



Smart Lens

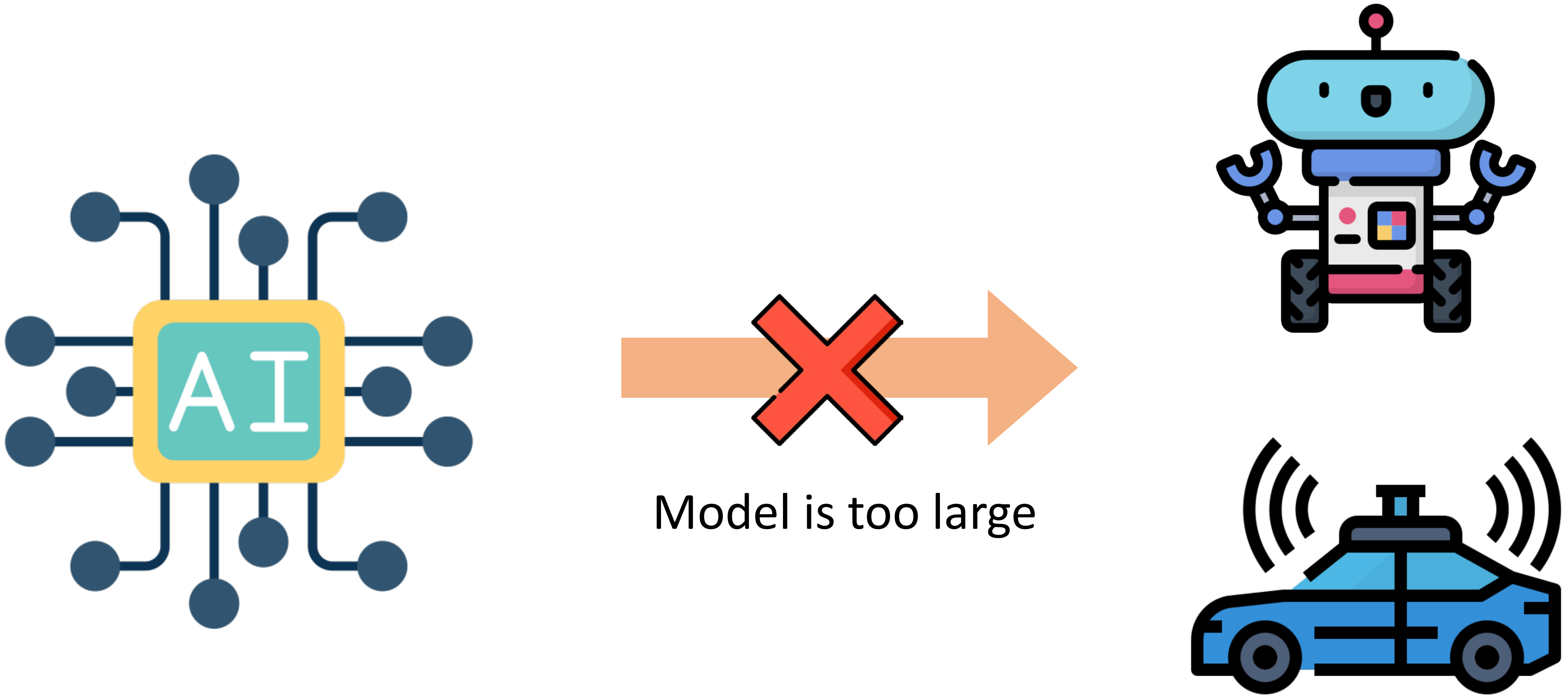
Robots

Computing Power is Lagging behind Models

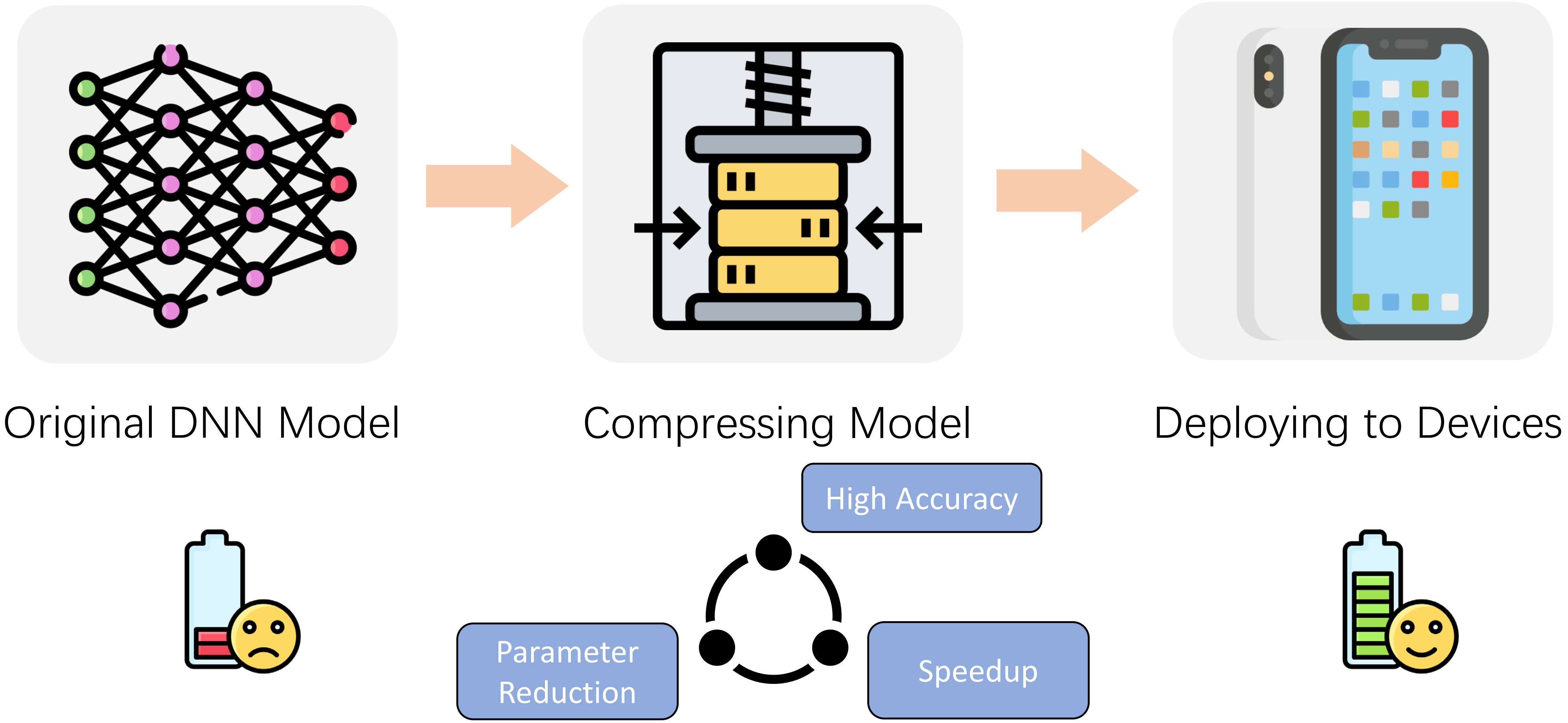


Data source: https://epochai.org/blog/trends-in-gpu-price-performance

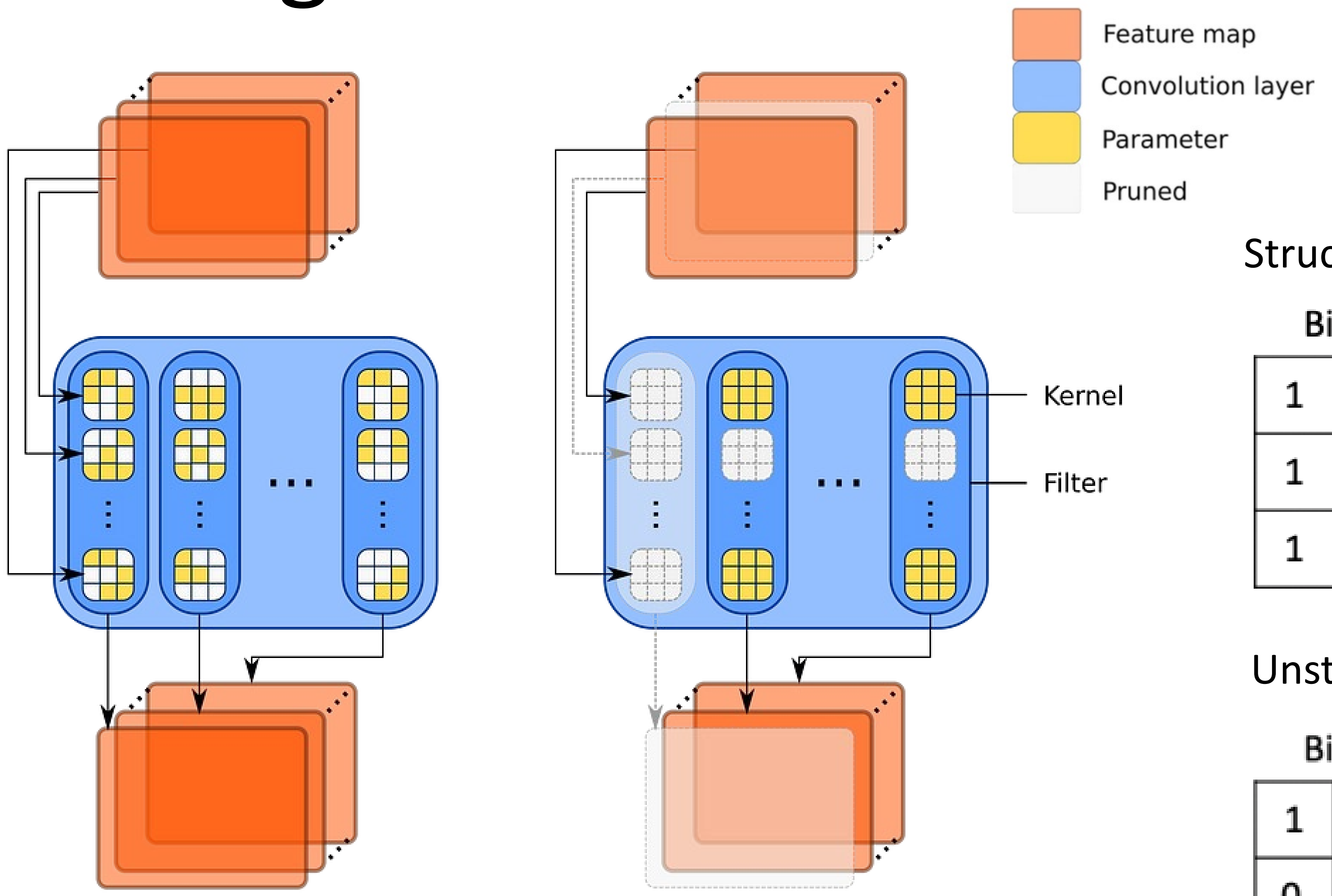
Deployment Challenge



Model Compression



Pruning



"Unstructured" : weight pruning "Structured" : filter pruning

Structured Pruning:

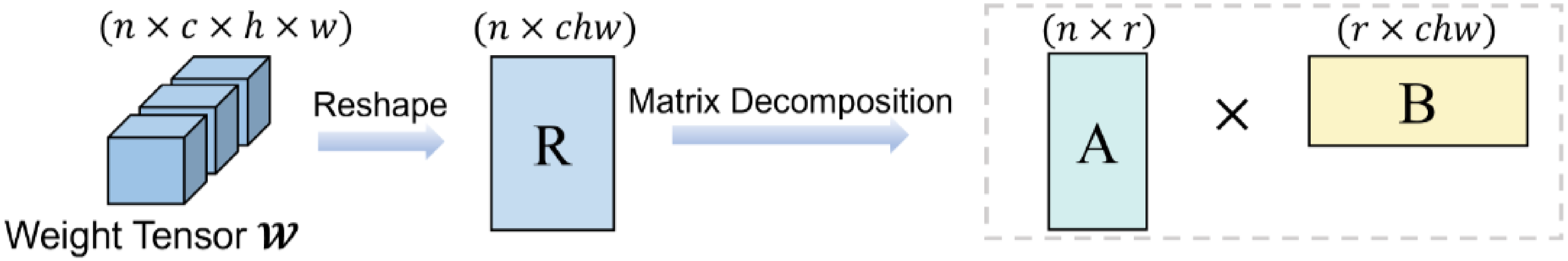
Bit Mask				Weight				Pruned		
1	0	1		.7	.2	.1		.7	0	.1
1	0	1	⊗	-.2	.8	.9	=	-.2	0	.9
1	0	1		.2	.1	.3		.2	0	.3

Unstructured Pruning:

Bit Mask				Weight				Pruned		
1	0	1		.7	.2	.1		.7	0	1
0	1	1	⊗	-.2	.8	.9	=	0	.8	1
1	1	0		.2	.1	.3		.2	1	0

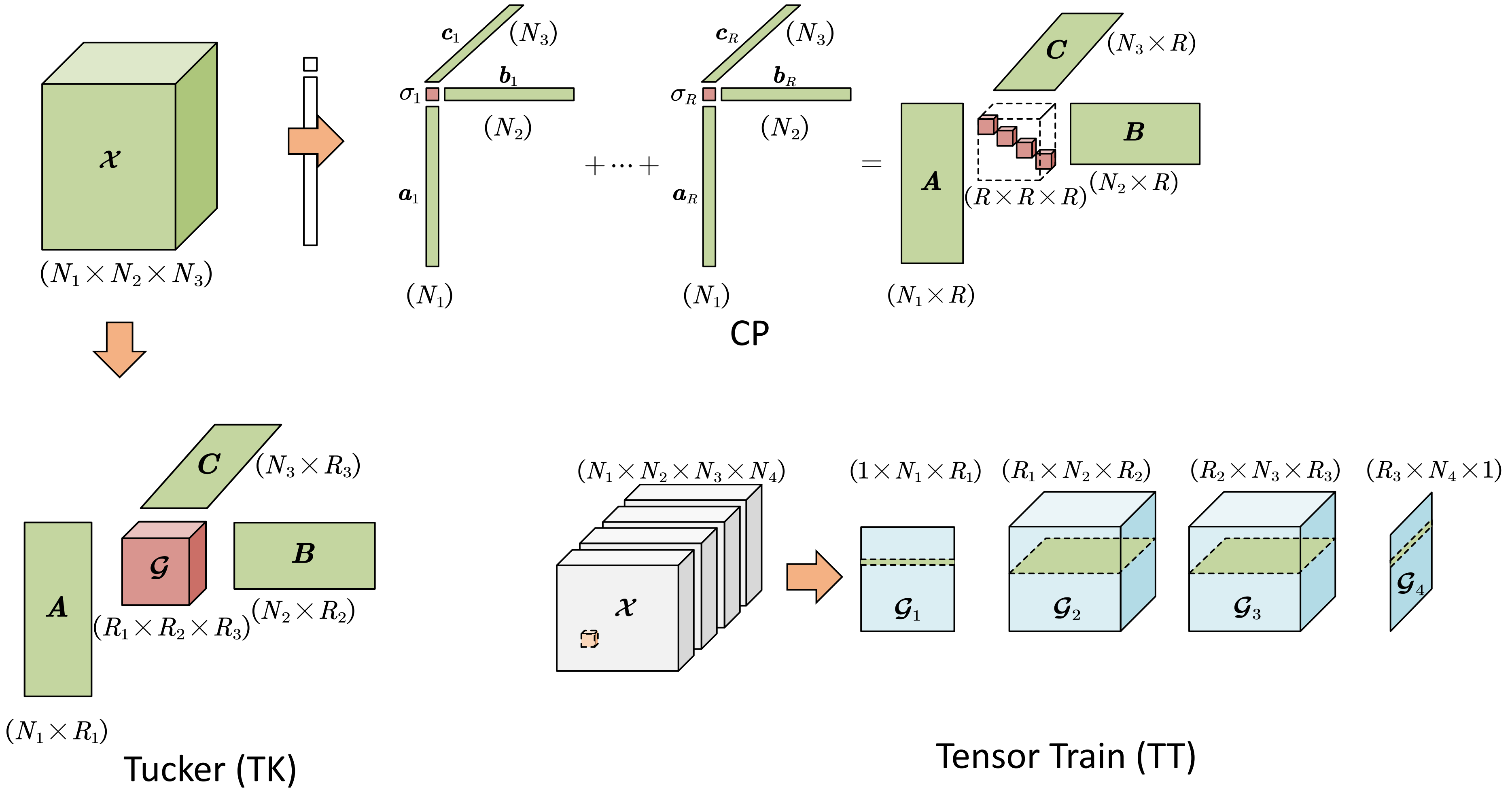
https://towardsdatascience.com/neural-network-pruning-101-af816aaea61

Matrix Decomposition



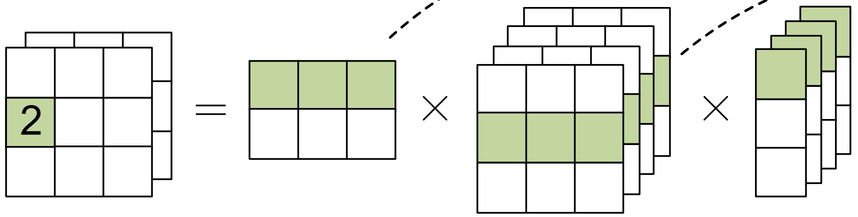
- For CNNs, the original 4-D kernel tensor needs to be flattened to a matrix
- Only one-dimensional linear correlation can be leveraged
- The reshape may lead to unbalanced matrix shape, e.g., 64*32*3*3 -> 64*9216, the number of columns is much larger than rows
- Hardware friendly

Tensor Decomposition



Tensor Decomposition

A numeric example for TT decomposition:



$$\mathcal{X}(1,2,1) = \mathcal{G}_1(:,1,:)\cdot\mathcal{G}_2(:,2,:)\cdot\mathcal{G}_3(:,1,:) = \begin{bmatrix}1 & 3 & 2\end{bmatrix} \times \begin{bmatrix}-2 & -1 & 4 & 3\\ 2 & 2 & -1 & -2\\ -1 & -1 & 2 & 3\end{bmatrix} \times \begin{bmatrix}3\\ -2\\ 1\\ -1\end{bmatrix}$$

- Directly decompose weight tensors into a series of small tensor cores without reshaping
- Ultra-high compression ratio, e.g., >1000X for RNNs
- Hardware friendly, parallel memory accessible
- Multi-dimensional correlation can be leveraged

Comparison among Compression Methods

01
High Accuracy

02
Speedup

03
Parameter Reduction

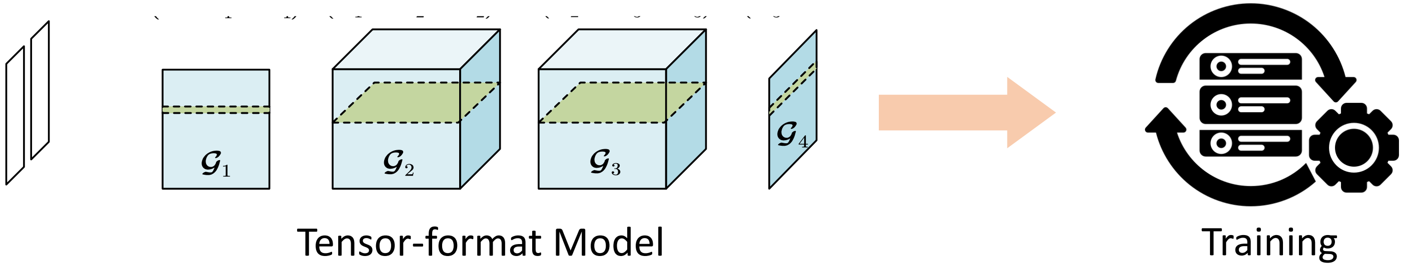
Method	High Accuracy	Hardware Friendly	Ultra-high Compression
Structured Pruning	✗	✓	✗
Unstructured Pruning	✓	✗	✗
Matrix Decomposition	✗	✓	✗
Tensor Decomposition	?	✓	✓

Agenda

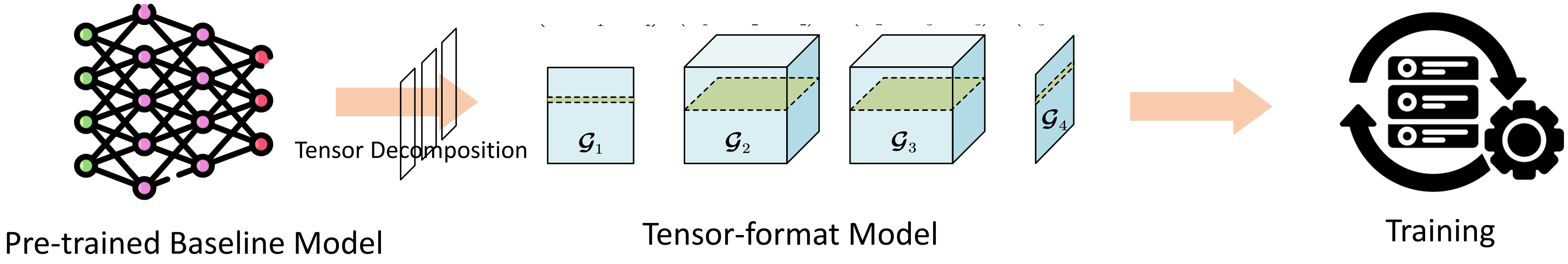
- Background
- **Optimization-based Compression**
- Efficient Tensor Train-based Convolution
- Experimental Results & Summary

Compression with Tensor Decomposition

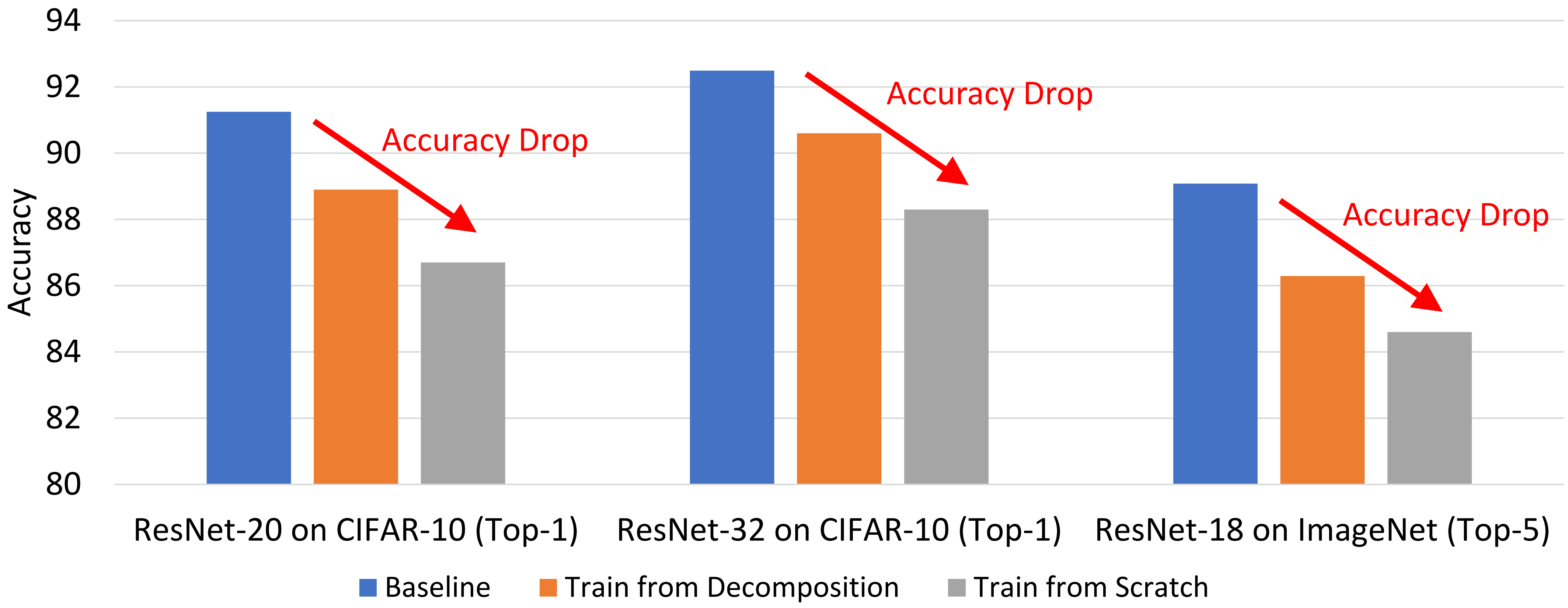
- Two ways to train a tensor-format DNN model:
  - Train from scratch (randomly initialize)



  - Train from decomposing a pre-trained DNN model to tensor format



Unsatisfied Accuracy



Wenqi Wang, et al. Wide compression: Tensor ring nets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9329–9338, 2018
Timur Garipov, et al. Ultimate tensorization: compressing convolutional and fc layers alike. arXiv preprint arXiv:1611.03214, 2016.
Nannan Li, et al. Heuristic rank selection with progressively searching tensor ring network. arXiv preprint arXiv:2009.10580, 2020.

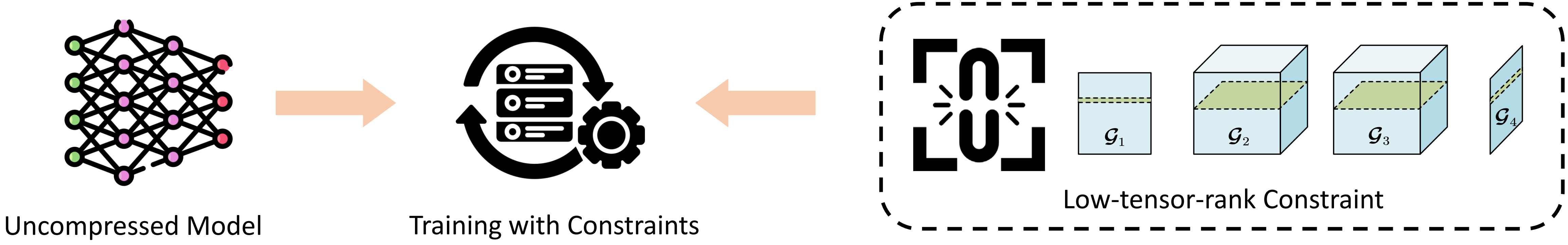
Why Accuracy Degradation?

- Train from scratch
  - Without any information from pre-trained model
  - Tucker-format model is of limited capacity

- Train from decomposition
  - The pre-trained model lacks low-tensor-rank property
  - Direct decomposition may lead to significant approximation error

Proposed Training Framework

Key idea: Impose low-tensor-rank property onto uncompressed model during training



- Utilize full capacity of uncompressed model
- Reduce the approximation error after decomposition

ADMM-based Compression Framework

Training objective:

Uncompressed weights

$$\min_{\mathcal{W}} \ \ell(\mathcal{W}),$$
$$\boxed{\text{s.t.} \ \ \operatorname{rank}(\mathcal{W}) \leqslant \boldsymbol{r}^*}$$

Low-tensor-rank constraint

Non-differentiable

Optimization with ADMM:

$$\min_{\mathcal{W},\mathcal{Z}} \ \ell\left(\mathcal{W}\right) + g(\mathcal{Z}),$$
$$\text{s.t.} \ \ \mathcal{W} = \mathcal{Z}.$$

$$g(\mathcal{W}) = \begin{cases} 0 & \mathcal{W} \in \mathcal{S}, \\ +\infty & \text{otherwise.} \end{cases}$$

Augmented Lagrangian form:

$$\mathcal{L}_\rho(\mathcal{W}, \mathcal{Z}, \mathcal{U}) = \ell(\mathcal{W}) + g(\mathcal{Z}) + \frac{\rho}{2}\|\mathcal{W} - \mathcal{Z} + \mathcal{U}\|_F^2 + \frac{\rho}{2}\|\mathcal{U}\|_F^2,$$

ADMM-based Compression Framework

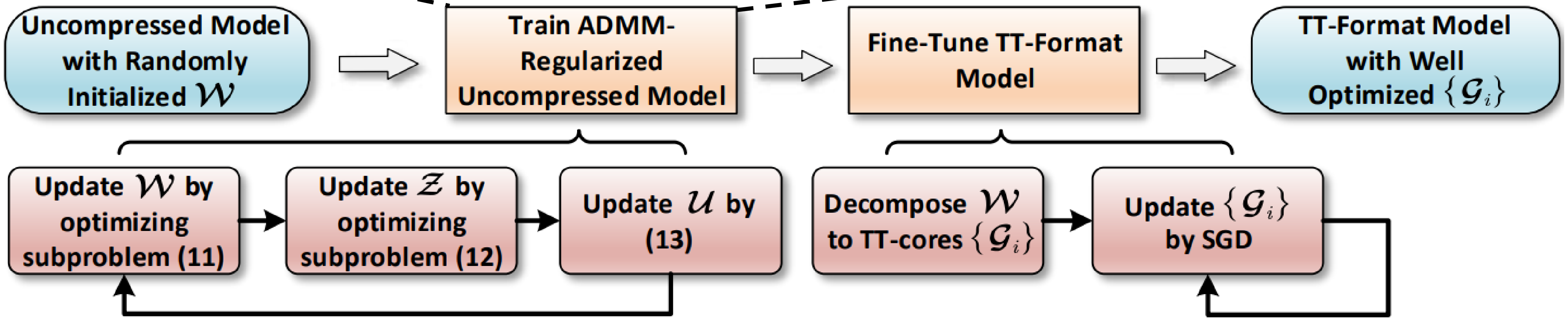
- Updating steps:

$$\mathcal{W}^{t+1} = \underset{\mathcal{W}}{\operatorname{argmin}} \ \mathcal{L}_\rho\left(\mathcal{W}, \mathcal{Z}^t, \mathcal{U}^t\right), \qquad (11) \longrightarrow \mathcal{W}^{t+1} = \mathcal{W}^t - \eta \frac{\partial \mathcal{L}_\rho(\mathcal{W}, \mathcal{Z}^t, \mathcal{U}^t)}{\partial \mathcal{W}},$$

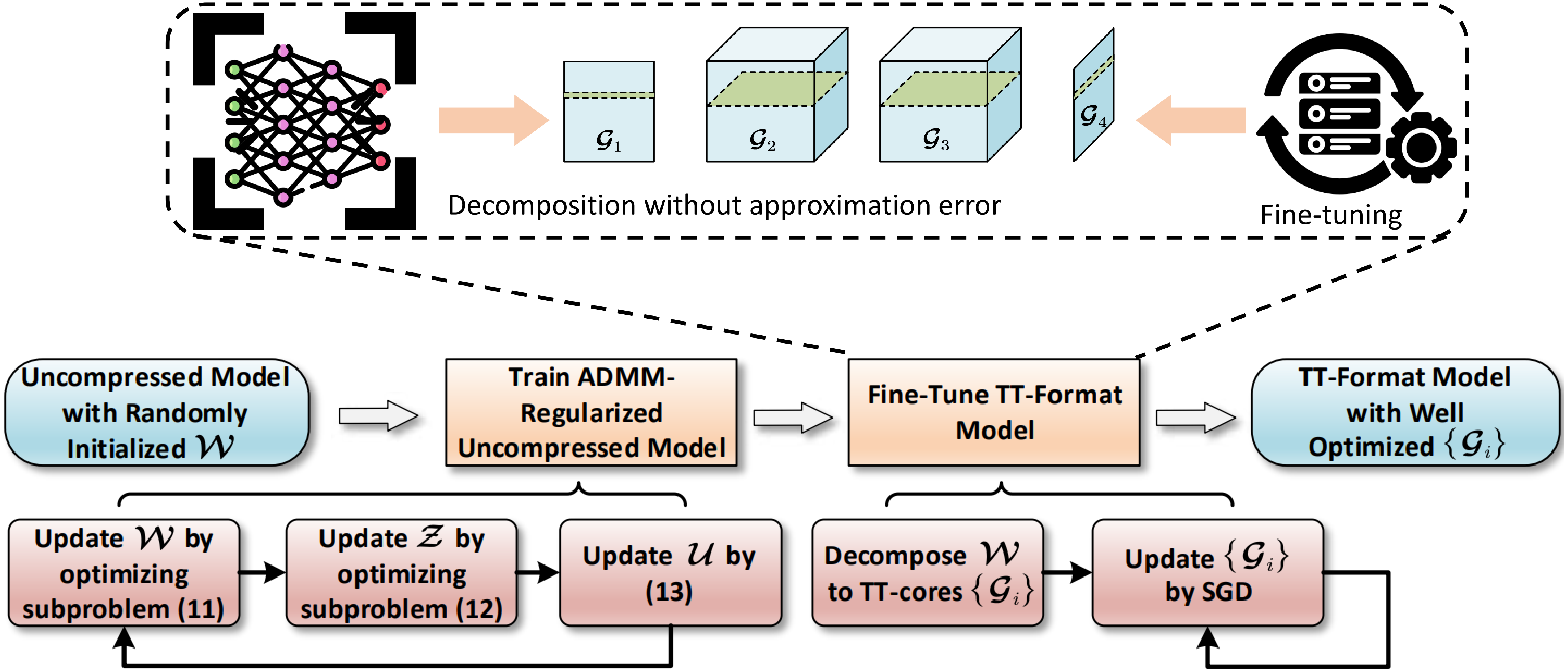
$$\mathcal{Z}^{t+1} = \underset{\mathcal{Z}}{\operatorname{argmin}} \ \mathcal{L}_\rho\left(\mathcal{W}^{t+1}, \mathcal{Z}, \mathcal{U}^t\right), \qquad (12) \longrightarrow \mathcal{Z}^{t+1} = \Pi_{\mathcal{S}}(\mathcal{W}^{t+1} + \mathcal{U}^t)$$

$$\mathcal{U}^{t+1} = \mathcal{U}^t + \mathcal{W}^{t+1} - \mathcal{Z}^{t+1}, \qquad (13)$$

Truncate tensor ranks to target ***r****



ADMM-based Compression Framework

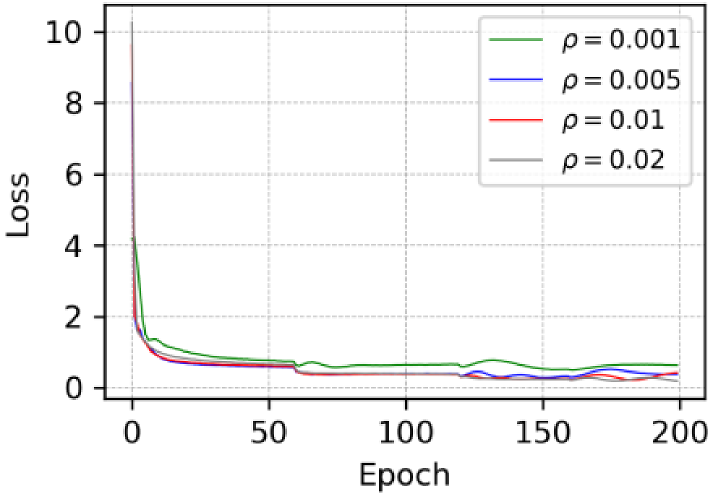


Sensitivity Analysis

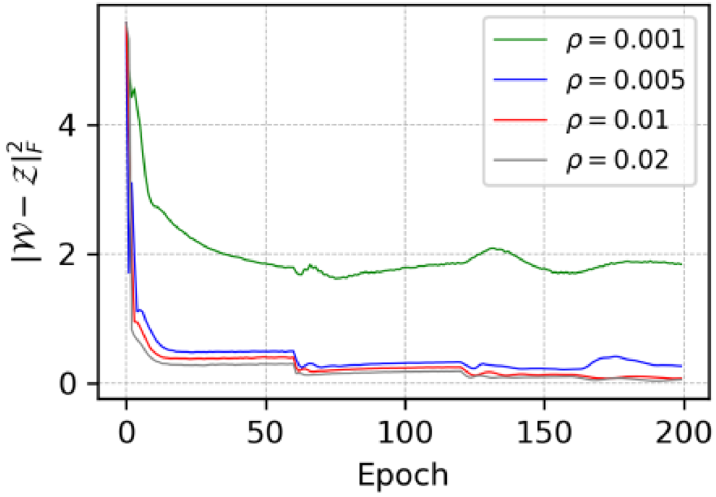
Lagrangian form:

$$\mathcal{L}_\rho(\mathcal{W},\mathcal{Z},\mathcal{U}) = \ell(\mathcal{W}) + g(\mathcal{Z}) + \boxed{\frac{\rho}{2}}\|\mathcal{W}-\mathcal{Z}+\mathcal{U}\|_F^2 + \boxed{\frac{\rho}{2}}\|\mathcal{U}\|_F^2,$$

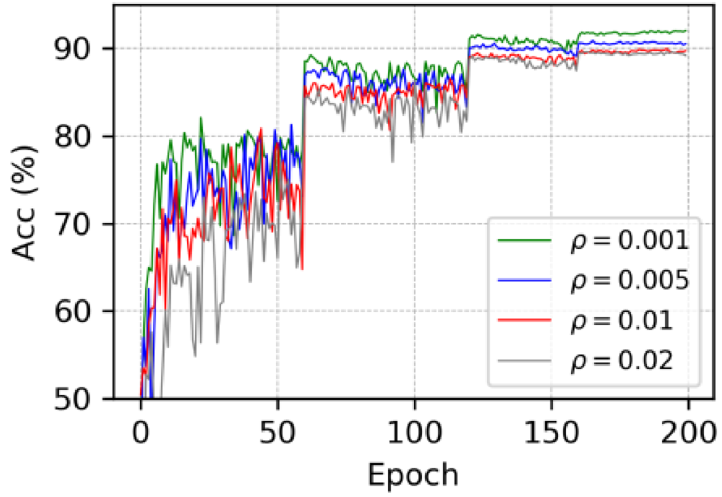
Hyper-parameter that controls low-tensor-rank property



Loss curves



Approximation error



Accuracy curves

Agenda

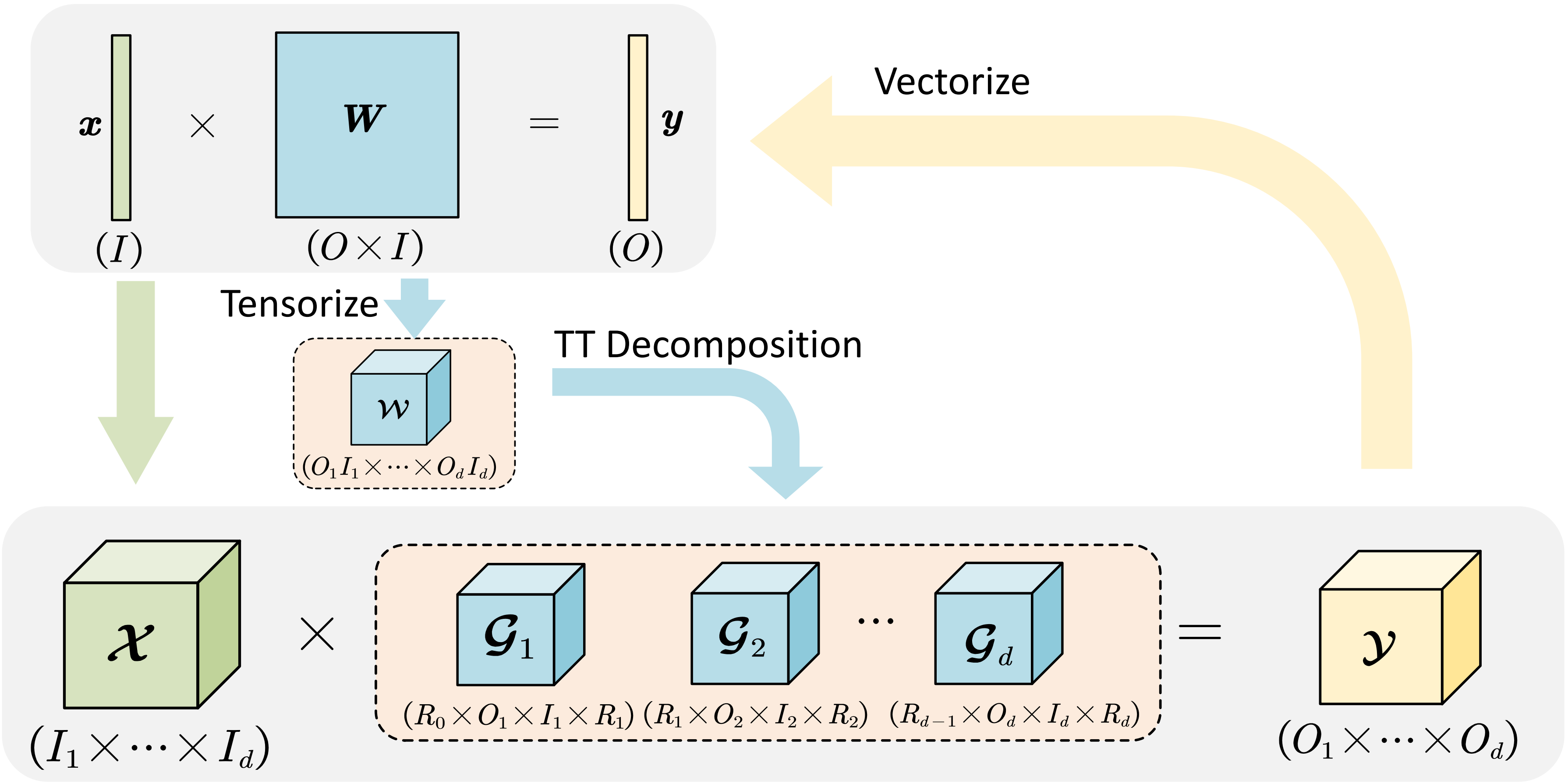
Background

Optimization-based Compression

Efficient Tensor Train-based Convolution

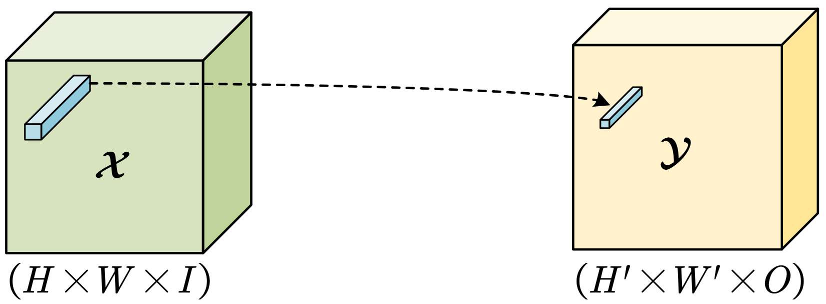
Experimental Results & Summary

TT-based Fully Connected Layer (TT-FC)

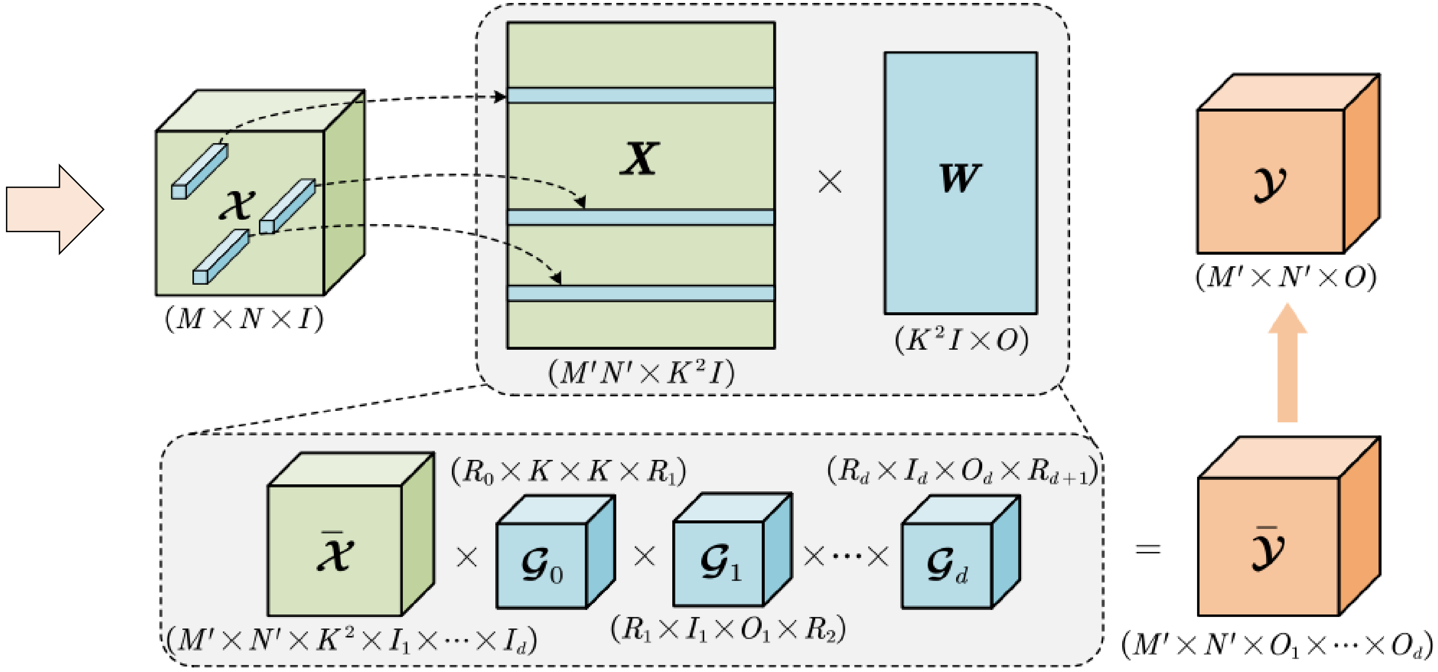


TT-based Convolutional Layer (TT-CONV)

Original convolution:



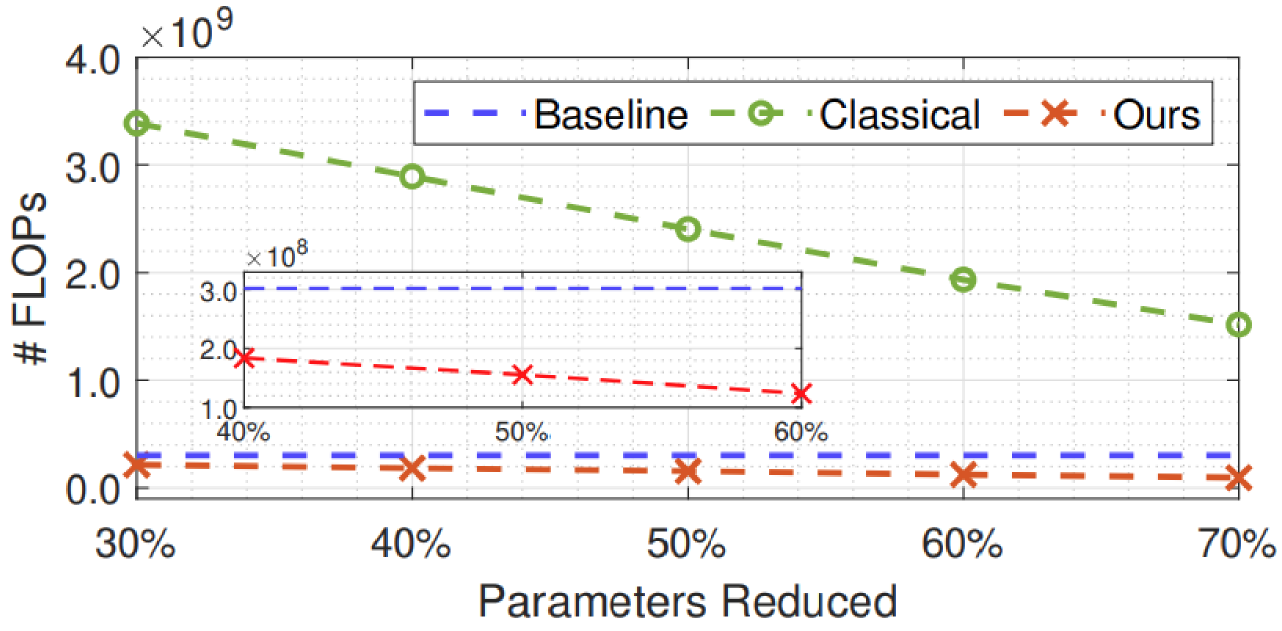
Conventional TT-CONV:



Timur Garipov, et al. Ultimate tensorization: compressing convolutional and fc layers alike. arXiv preprint arXiv:1611.03214, 2016.

Unbalanced FLOPs and Parameter Reduction

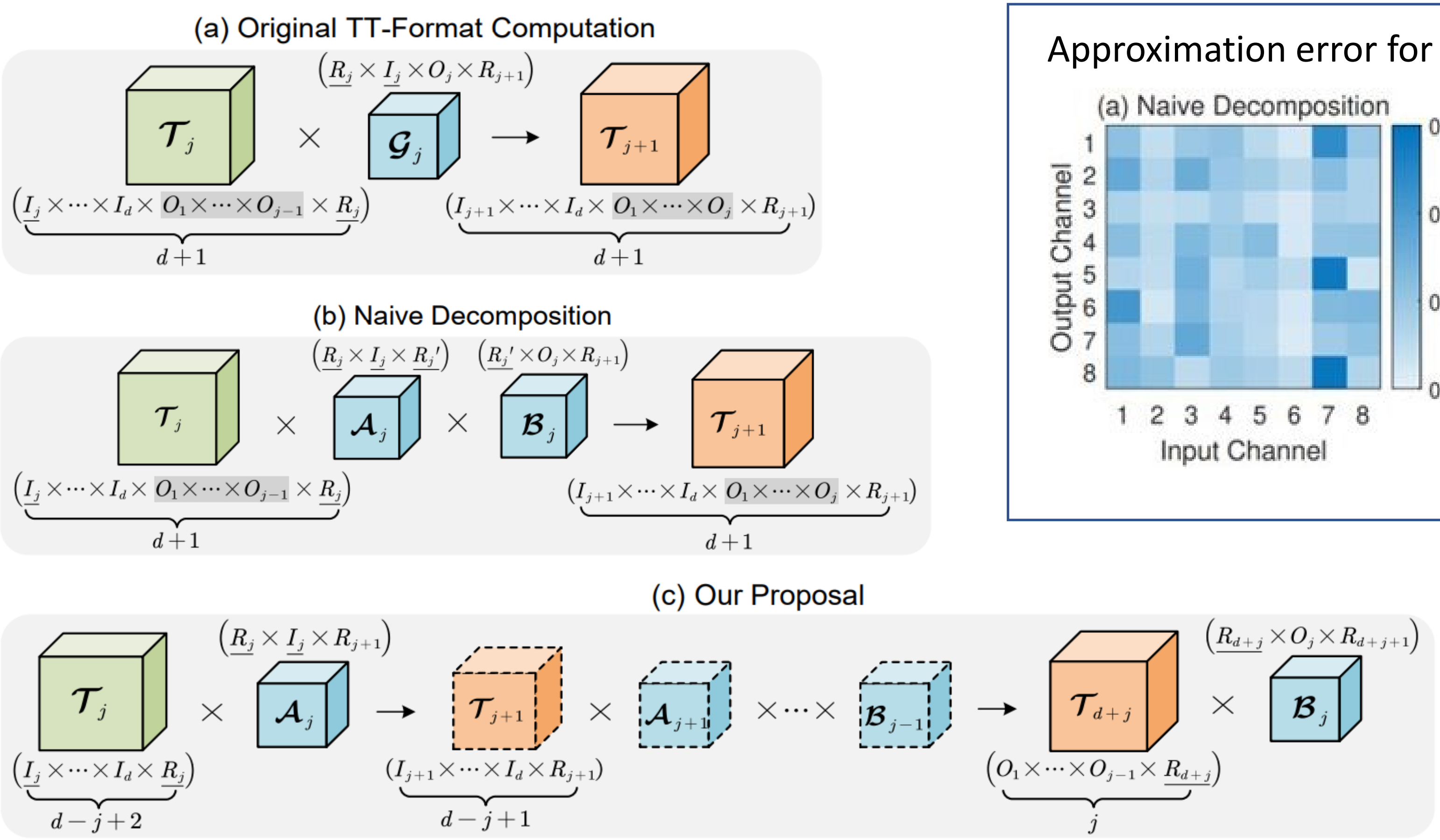
- FLOPs vs Parameter reduction:



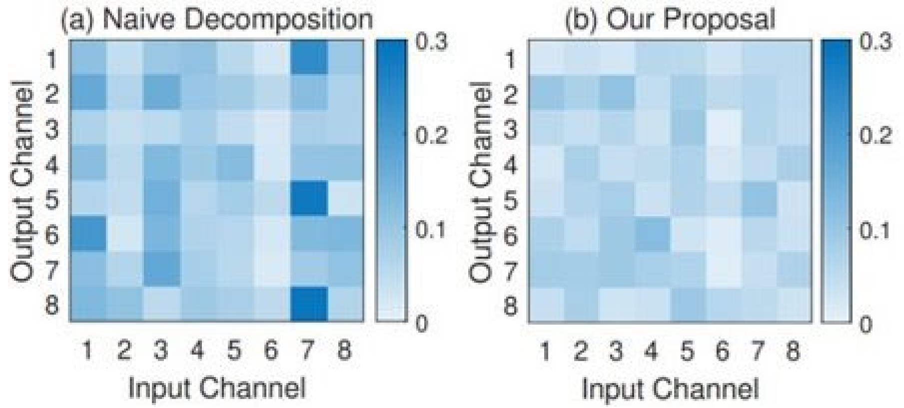
layer3.0.conv1 in ResNet-18 when using conventional TT-CONV

- ➢ Conventional TT-CONV ("Classical") causes even **higher FLOPs consumption** than the uncompressed one ("Baseline")

Analysis for Unbalanced FLOPs and Parameters



Approximation error for one layer in ResNet-32:



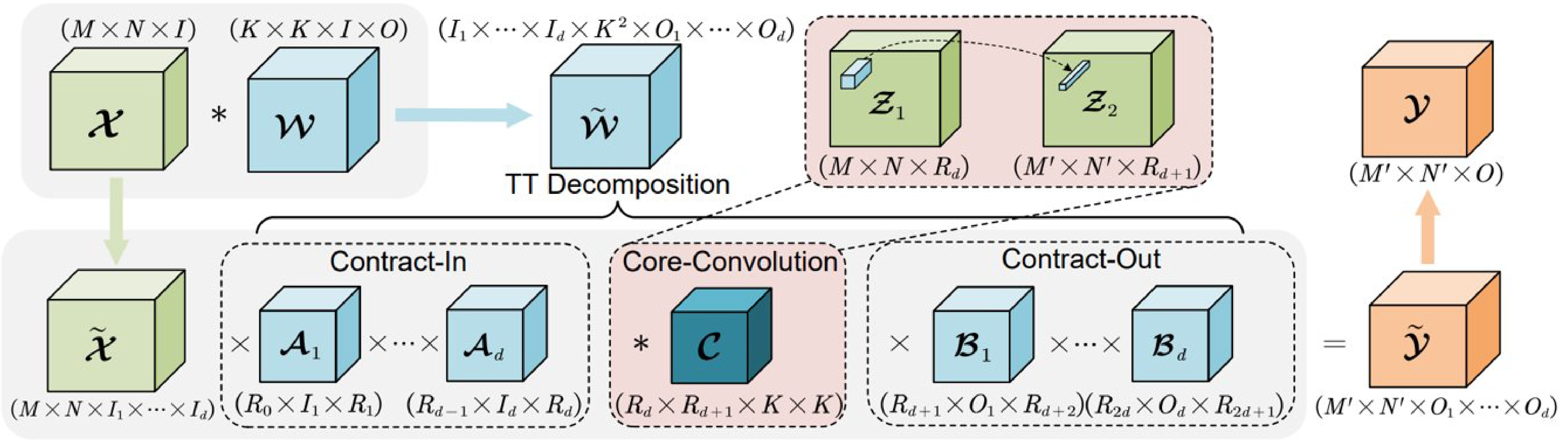
FLOPs

(a) $\mathcal{O}(I_m^{d-j+1} O_m^j R^2 M' N')$

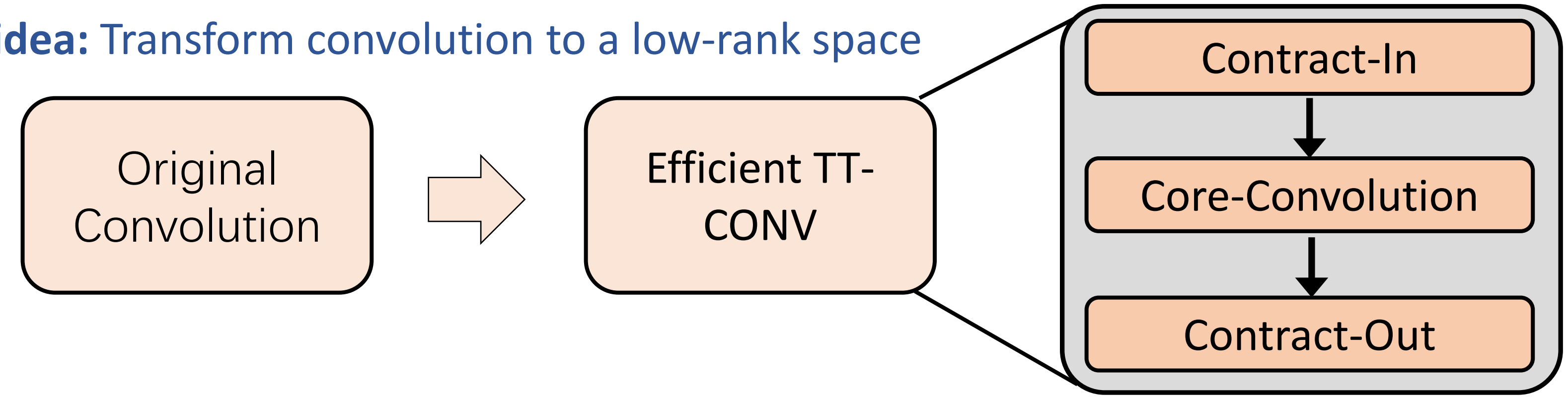
(b) $\mathcal{O}(2 I_m^{d-j} O_m^j R^2 M' N')$

(c) $\mathcal{O}((I_m^{d-j+1} R^2 + O_m^j R^2) M' N')$

Proposed Efficient TT-CONV Scheme



Key idea: Transform convolution to a low-rank space



Agenda

- Background
- Optimization-based Compression
- Efficient Tensor Train-based Convolution
- **Experimental Results & Summary**

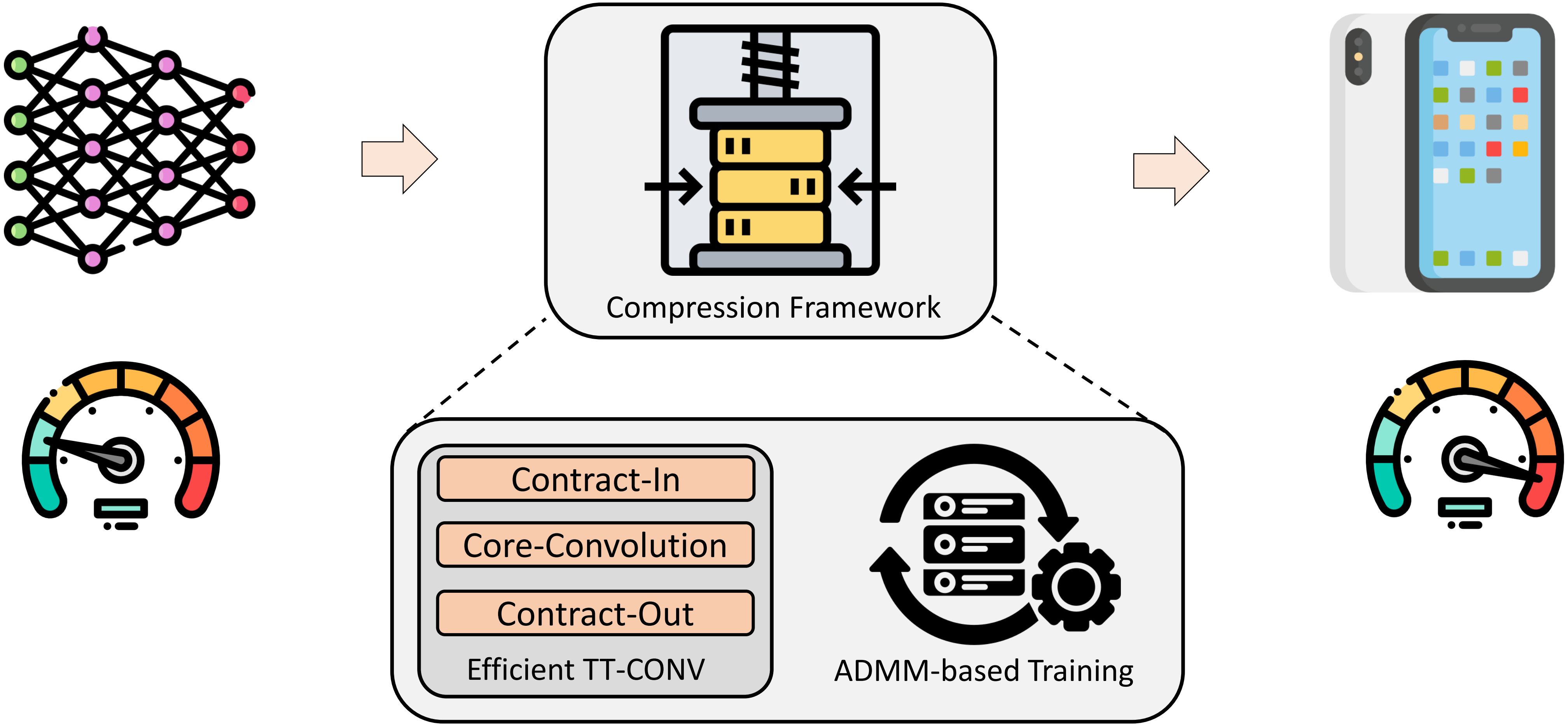
Experimental Results on CIFAR-10

Model	Compression Method	Top-1 Acc. (%)			FLOPs↓	Params.↓
		Baseline	Compressed	Δ		
ResNet-32						
Rethinking [20]	Pruning	N/A	92.56	N/A	30%	30%
FPGM [12]	Pruning	92.63	92.82	+0.19	53%	N/A
SCOP	Pruning	92.66	92.13	-0.53	56%	56%
Wide [34]	Tensor Ring	92.49	90.30	-2.19	N/A	80%
Ultimate [7, 24]	Classical TT	92.49	88.30	-4.19	×	80%
HODEC (Ours)	Proposed TT	92.49	91.28	-1.21	**72%**	**80%**
HODEC (Ours)	Proposed TT	92.49	**93.05**	**+0.56**	**60%**	65%
ResNet-56						
HRank [19]	Pruning	93.26	93.17	-0.09	50%	42%
SCOP [32]	Pruning	93.70	93.64	-0.06	56%	56%
NPPM [6]	Pruning	93.04	93.40	+0.36	50%	N/A
CHIP [31]	Pruning	93.26	94.16	+0.75	47%	43%
TRP [36]	Low-rank Matrix	93.14	92.63	-0.51	60%	N/A
CC [17]	Low-rank Matrix	93.33	93.64	+0.31	52%	48%
Ultimate [7, 24]	Classical TT	93.04	91.14	-1.90	×	50%
HODEC (Ours)	Proposed TT	93.04	**94.20**	**+1.16**	**62%**	**67%**

Experimental Results on ImageNet

Model	Compression Method	Top-1 Acc. (%)			Top-5 Acc. (%)			FLOPs↓
		Baseline	Compr.	Δ	Baseline	Compr.	Δ	
ResNet-18								
FPGM [12]	Pruning	70.28	68.41	-1.87	89.63	88.48	-1.15	42%
SCOP [32]	Pruning	69.76	68.62	-1.14	89.08	88.45	-0.63	45%
TRP [36]	Low-rank Matrix	69.10	65.51	-3.59	88.94	86.74	-2.20	60%
Stable [27]	Tucker-CP	69.76	69.07	-0.69	89.08	88.93	-0.15	67%
HODEC (Ours)	Proposed TT	69.76	**69.15**	**-0.61**	89.08	**88.99**	**-0.09**	**68%**
ResNet-50								
FPGM [12]	Pruning	76.15	75.59	-0.56	92.87	92.63	-0.24	42%
HRank [19]	Pruning	76.15	74.98	-1.17	92.87	92.33	-0.54	44%
SCOP [32]	Pruning	76.15	75.26	-0.89	92.87	92.53	-0.34	55%
NPPM [6]	Pruning	76.15	75.96	-0.19	92.87	92.75	-0.12	56%
CHIP [31]	Pruning	76.15	76.15	0.00	92.87	92.91	+0.04	49%
TRP [36]	Low-rank Matrix	75.90	74.06	-1.84	92.70	92.07	-0.63	45%
CC [17]	Low-rank Matrix	76.15	75.59	-0.56	92.87	92.64	-0.23	53%
Stable [27]	Tucker-CP	76.13	74.66	-1.47	92.87	92.16	-0.71	62%
HODEC (Ours)	Proposed TT	76.13	**76.44**	**+0.31**	92.87	**93.16**	**+0.29**	**63%**

Summary


Compression Framework
Contract-In
Core-Convolution
Contract-Out
Efficient TT-CONV
ADMM-based Training

Thanks!