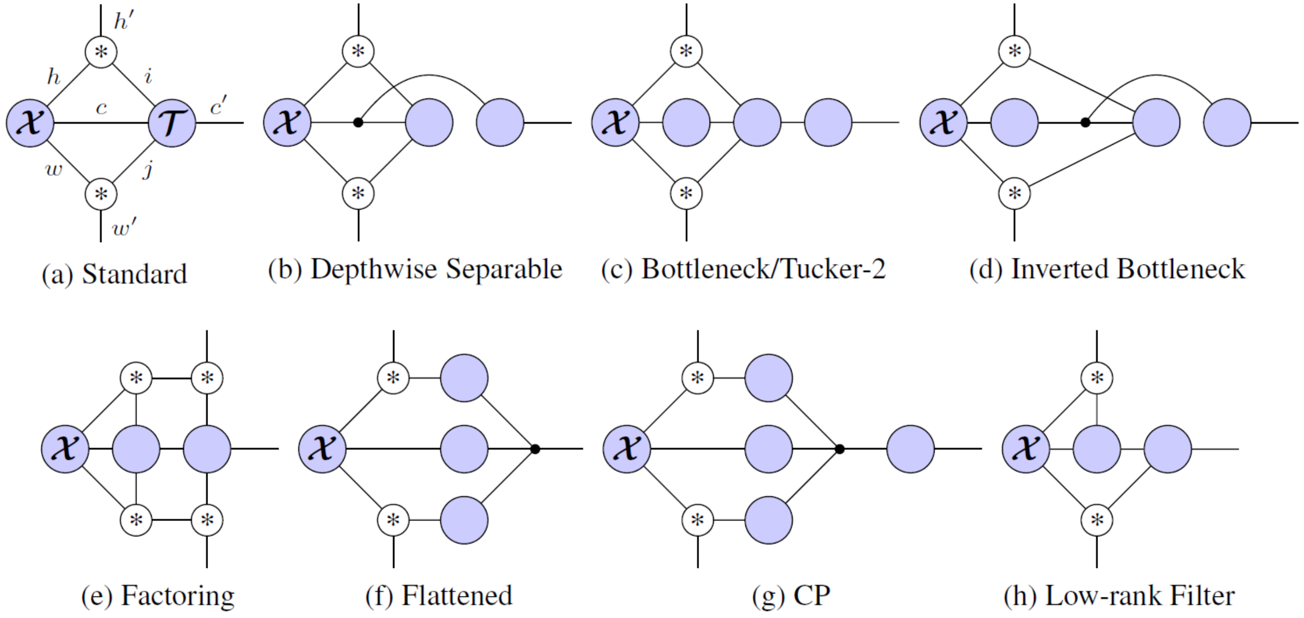
哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

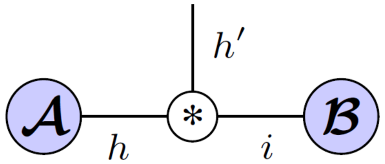
A Unified Weight Initialization Paradigm for Tensorial Convolutional Neural Networks

Yu Pan[1] , Zeyong Su[2] , Ao Liu[3] , Jingquan Wang[1] , Nannan Li[4,5] , Zenglin Xu[1,6*]

[1]Harbin Institute of Technology Shenzhen, Shenzhen, China
[2]University of Electronic Science and Technology of China, Chengdu, China
[3]Tokyo Institute of Technology, Tokyo, Japan
[4]State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[5]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[6]Pengcheng Laboratory, Shenzhen, China

Tensorial Convolutional Neural Networks (TCNNs)

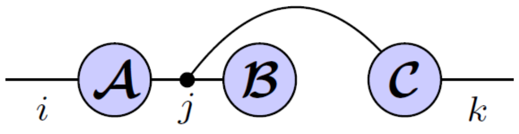




$$\Longleftrightarrow \sum_{h,i} p_{hh'i} a_h b_i$$

$$\mathcal{P} \in \{0,1\}^{H \times H' \times I} \quad p_{hh'i} = 1 \text{ if } h = h'_i$$

$$h'_i = (h' - 1)S + i - P$$



$$\Longleftrightarrow \sum_{j} a_{ij} b_j c_{jk}$$

Denoting convolutional operation with tensor diagram

Kohei Hayashi, Taiki Yamaguchi, Yohei Sugawara, Shin-ichi Maeda:
Exploring Unexplored Tensor Network Decompositions for Convolutional Neural Networks. NeurIPS 2019

Consider some questions:

- For a unknown tensorial CNN,

  whether we need to analyze its data-flow to derive its initialization

  - time and resource consuming

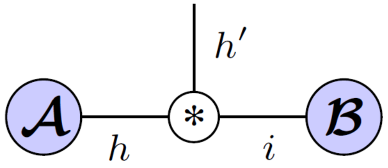
- For a NAS, if it found a pretty special structure,

  how to train it for a stable performance

  - a fatal shortage for NAS

Is there any initializing Principle for arbitrary tensorial CNNs?

Two Correction on Dummy Tensor

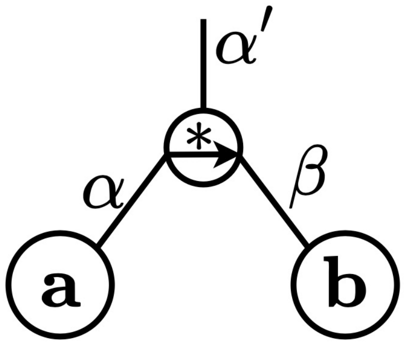


$$\Longleftrightarrow \quad \sum_{h,i} p_{hh'i} a_h b_i$$

$$\mathcal{P} \in \{0,1\}^{H \times H' \times I} \quad p_{hh'i} = 1 \text{ if } h = h'_i$$

$$h'_i = (h' - 1)S + i - P$$

(1) Correcting Diagram



Adding an arrow

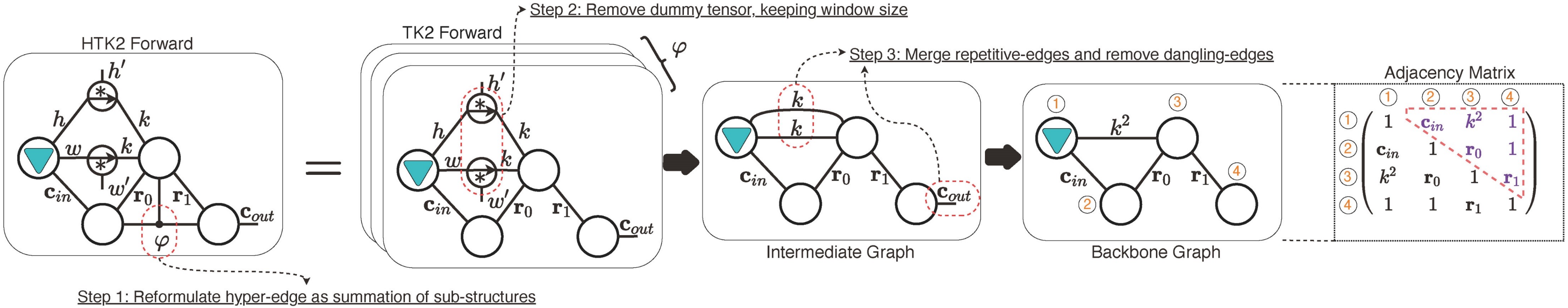
(2) Correcting Formulation

$$\mathcal{P}_{j,j',k} = 1 \text{ if } j = sj' + k - p$$

Removing -1

Unified Initialization

Graph-in



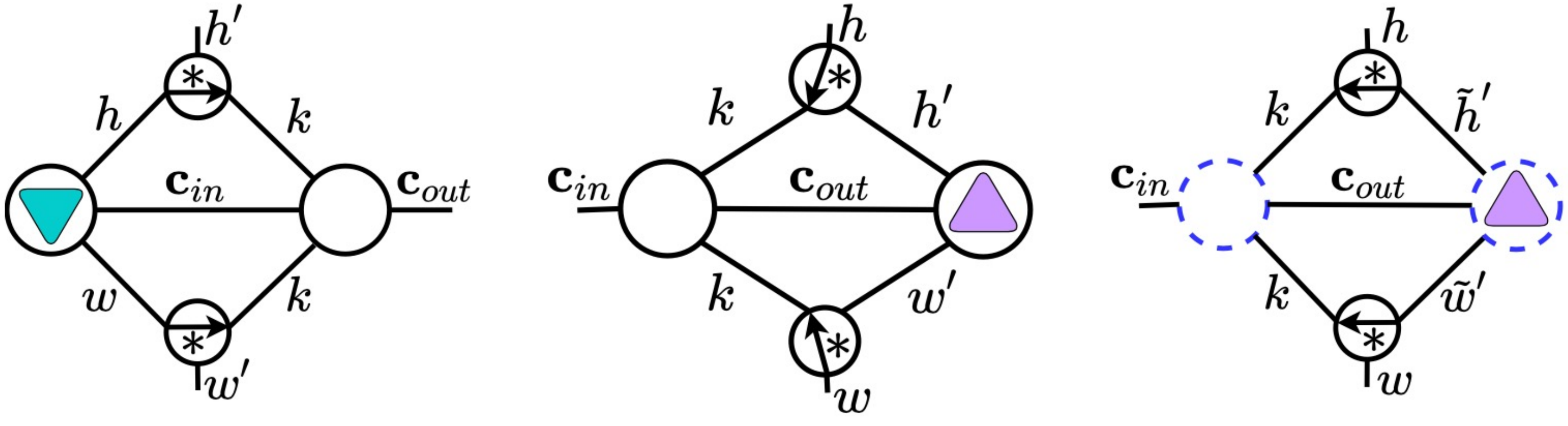
Unified Paradigm: $$\prod_{k=0}^{n-1} \sigma^2(\mathcal{W}^{(k)}) = \frac{1}{\boldsymbol{p_a}\varphi \prod_{i=0}^{n-1} \prod_{j=i+1}^{\tau-1} e_{ij}}$$

Maintaining data-flow variance in transition !

Unified Initialization

How to construct the fan-out mode?

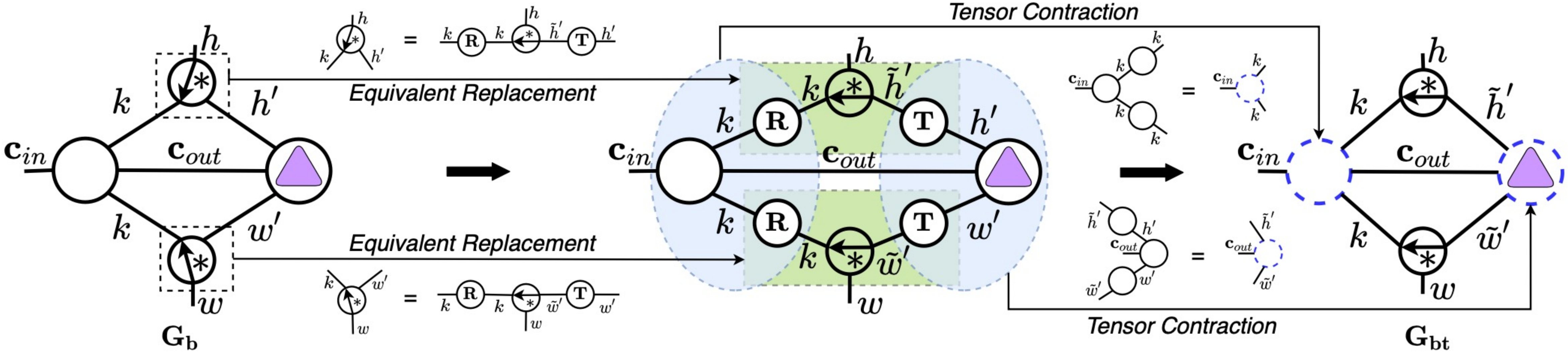
Convolution　　　　Not Convolution　　　　Convolution



(a) $\mathbf{G_f}$　　(b) $\mathbf{G_b}$　　(c) $\mathbf{G_{bt}}$ (ours)

How to transfer G_b to G_{bt} ?

Unified Initialization



$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

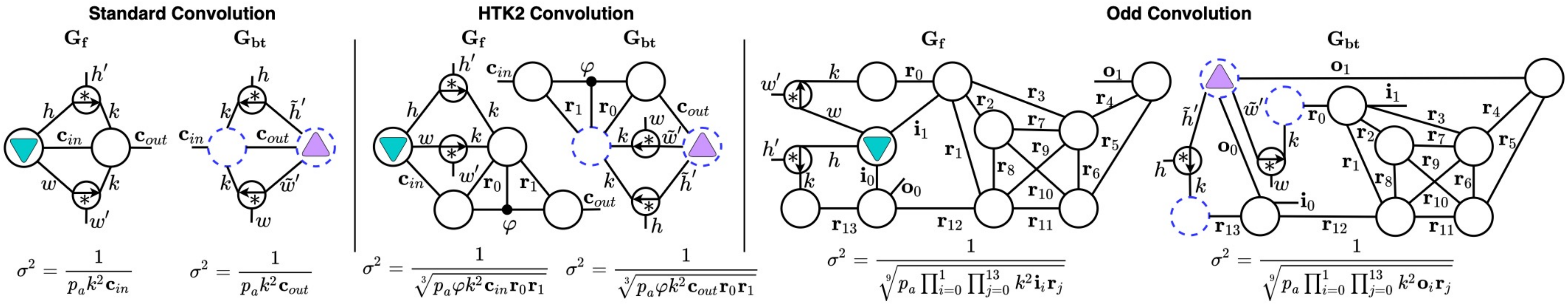
Reversal Matrix (R)

Reproducing Transformation

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

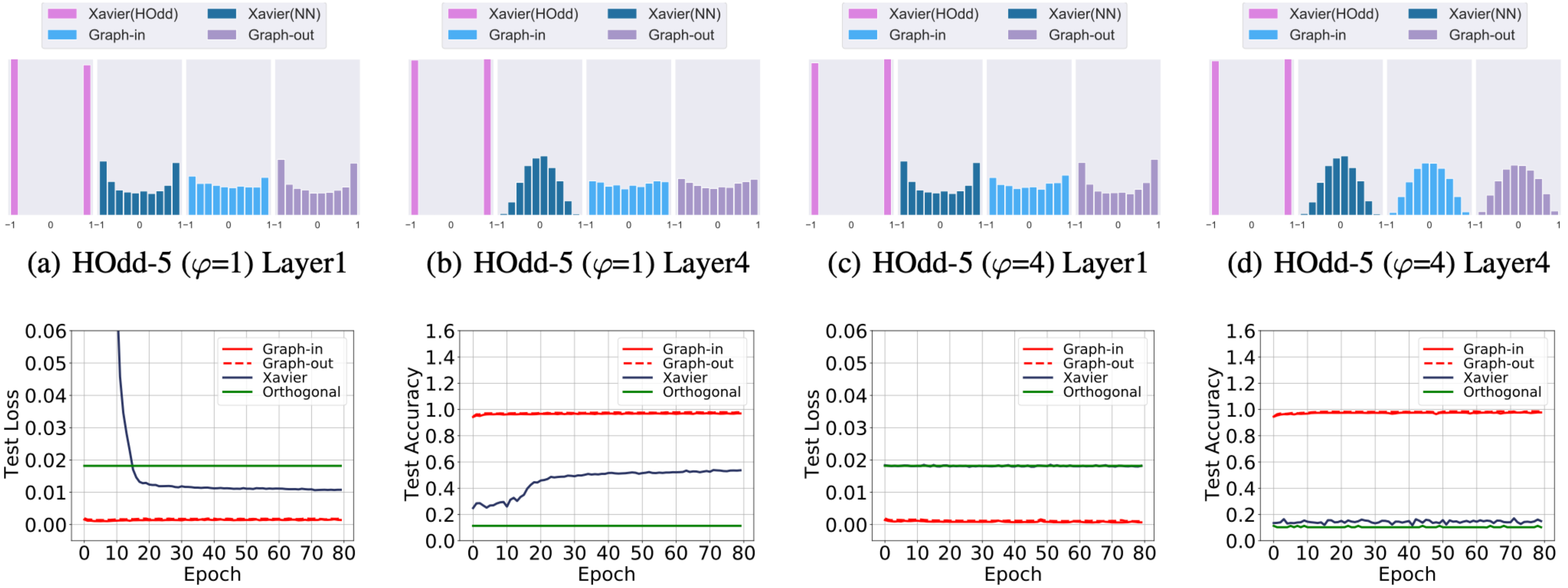
Transformation Matrix (T)

Unified Initialization



$$\prod_{k=0}^{n-1} \sigma^2(\mathcal{W}^{(k)}) = \frac{1}{\boldsymbol{p_a}\varphi \prod_{i=0}^{n-1} \prod_{j=i+1}^{\tau-1} e_{ij}}$$

Experiment result



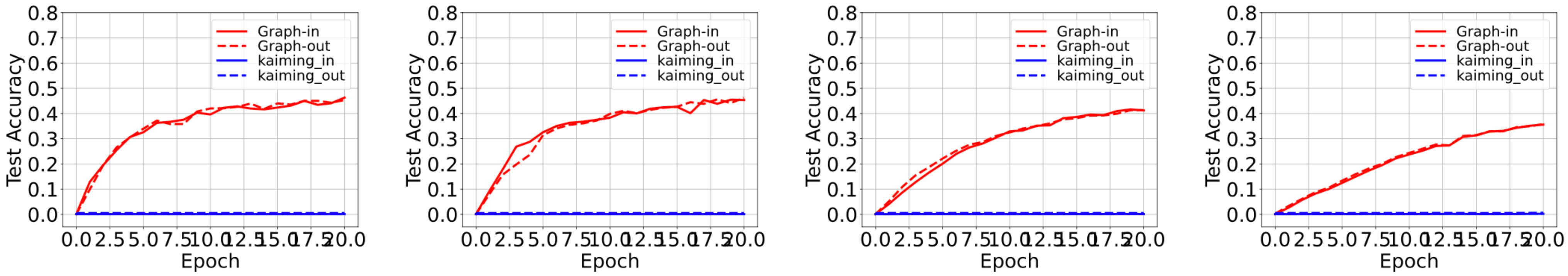
(a) HOdd-5 (φ=1) Layer1 (b) HOdd-5 (φ=1) Layer4 (c) HOdd-5 (φ=4) Layer1 (d) HOdd-5 (φ=4) Layer4

(e) HOdd-5 (φ=1) Loss (f) HOdd-5 (φ=1) Accuracy (g) HOdd-5 (φ=4) Loss (h) HOdd-5 (φ=4) Accuracy

On MNIST

Experiment result

	Rank-Edge Number	Cifar10 Kaiming (-in/-out)	Cifar10 Graph-in	Cifar10 Graph-out	Tiny-ImageNet Kaiming (-in/-out)	Tiny-ImageNet Graph-in	Tiny-ImageNet Graph-out
Low-Rank	1	0.1	0.8141	0.8163	0.307/0.2776	0.3153	0.3076
Tensor Ring	4	0.1	0.8308	0.8311	0.005	0.2494	0.249
HTK2(φ=4)	2	0.1	0.8638	0.8705	0.005	0.4014	0.4126
HOdd(φ=4)	14	0.1	0.8826	0.8806	0.005	0.5048	0.5045
Random-1	-	0.1	0.8538	0.8483	0.005	0.4965	0.5015
Random-2	-	0.1	0.8801	0.876	0.005	0.5379	0.5356
Random-3	-	0.1	0.8648	0.863	0.005	0.5475	0.5403
Random-4	-	0.1	0.8789	0.8816	0.005	0.5295	0.5306
Random-5	-	0.1	0.8622	0.8644	0.005	0.5444	0.5428



(a) HRand-RN-50 (b) HRand-RN-101 (c) HRand-gMLP-S16 (d) HRand-MLP-Mixer-B16

On ImageNet

哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

Thank you!