UNIVERSITY OF AMSTERDAM

Universiteit Leiden
The Netherlands

ASIA SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE

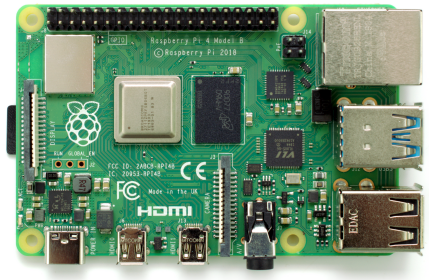
RobustDiCE: Robust and Distributed CNN Inference at the Edge

Xiaotian Guo

I. Motivation

Improve Robustness of Distributed CNN inference at the edge

Resource constrained



Better Quality of Service
Privacy
Security
…

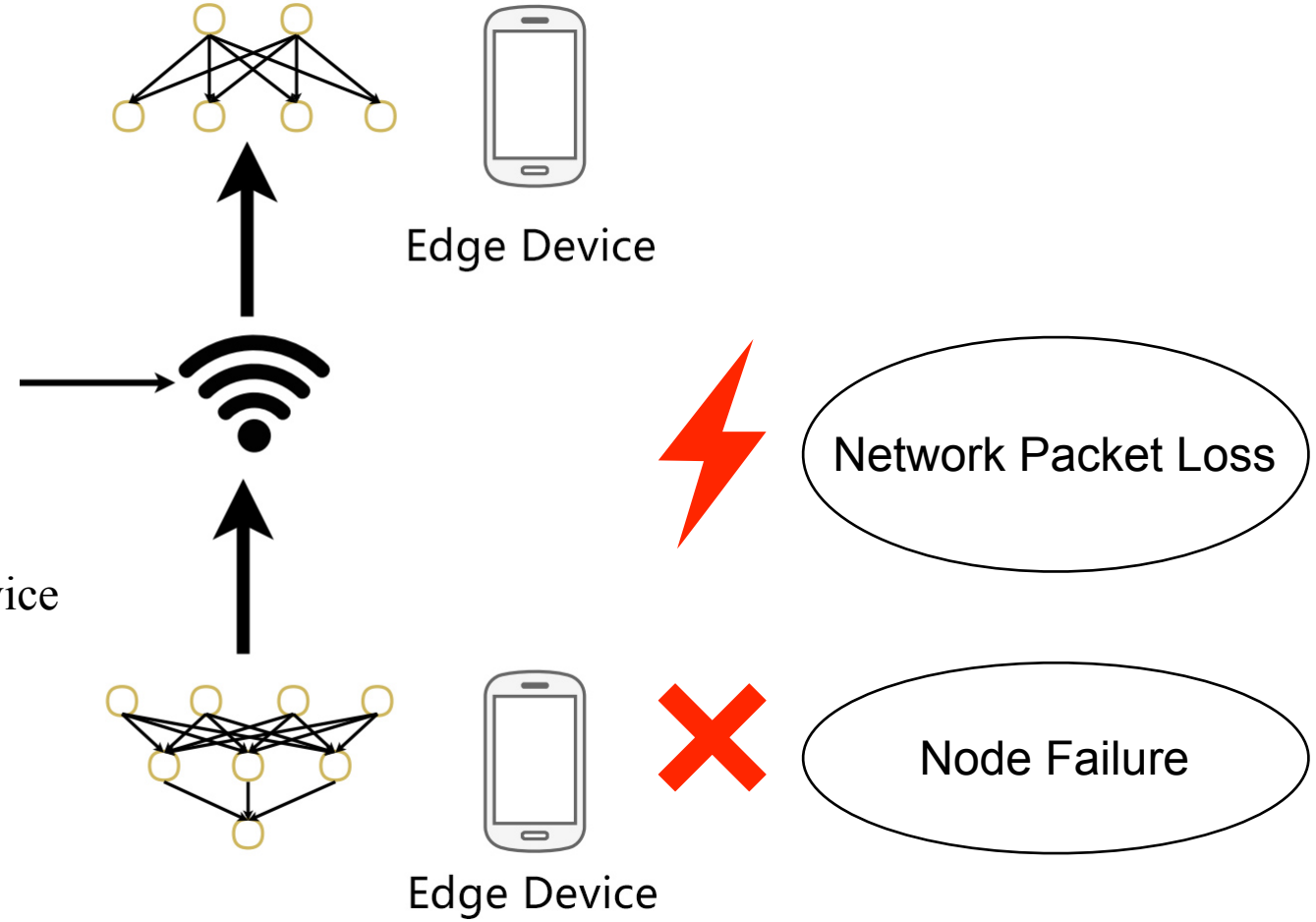
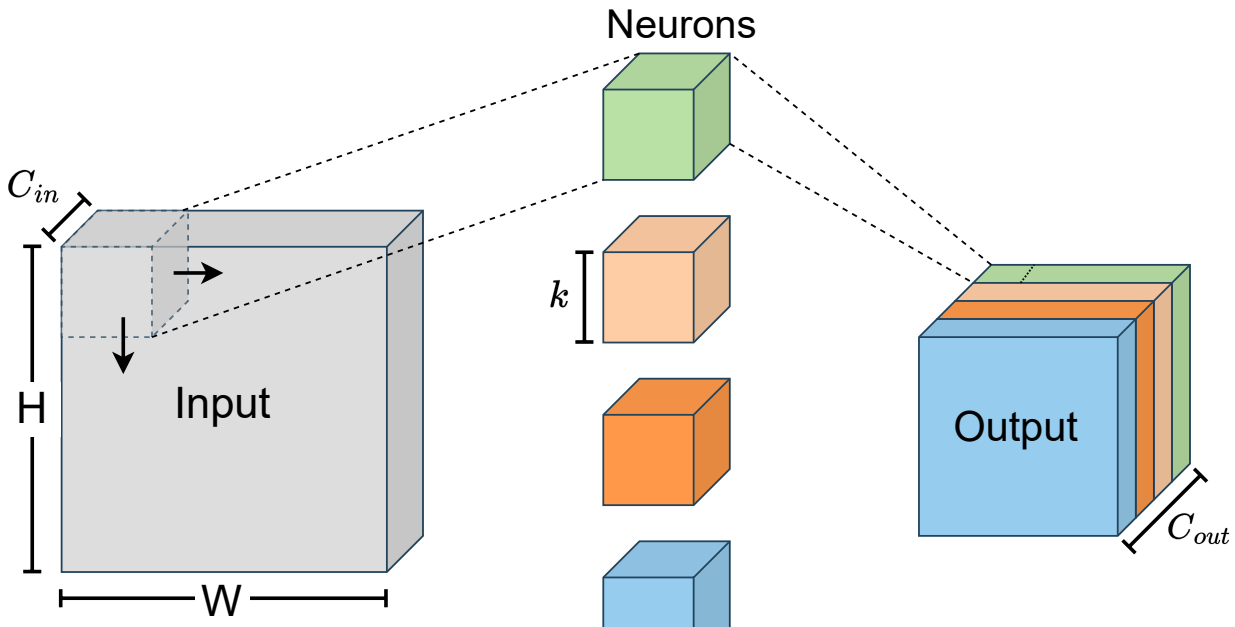


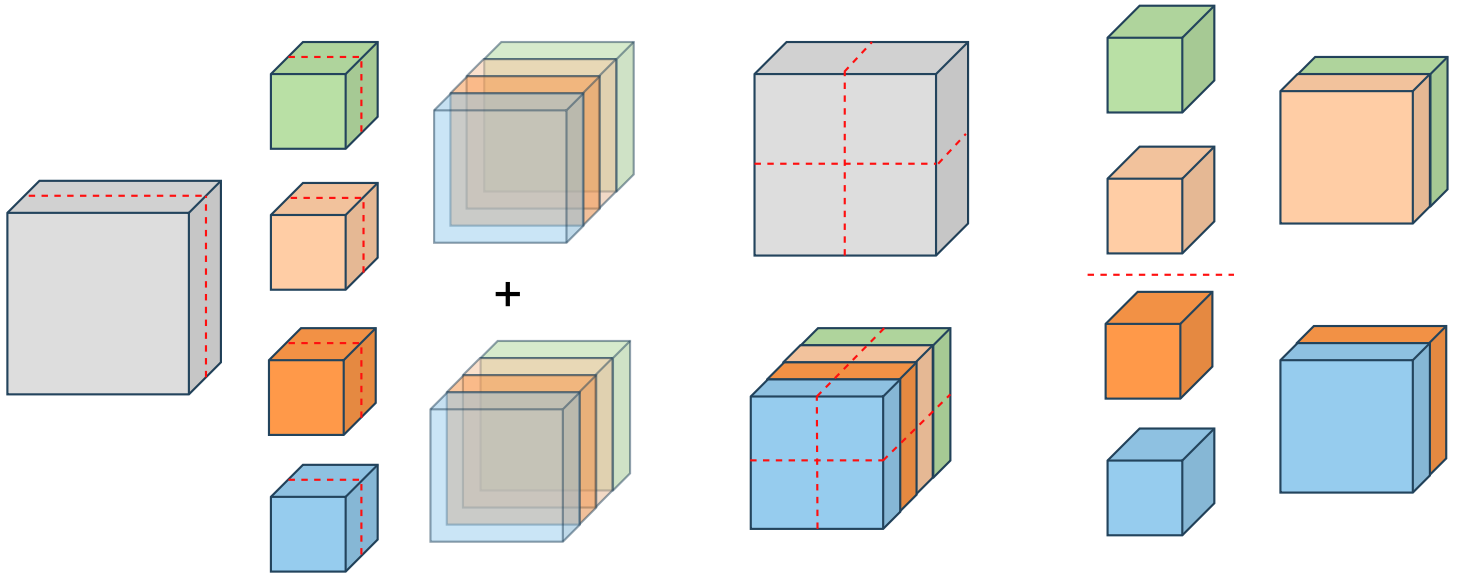
Fig. BranchNet, Von Zitzewitz, et.al. "Survey of neural networks in autonomous driving." 2017



(a) CNN Computations

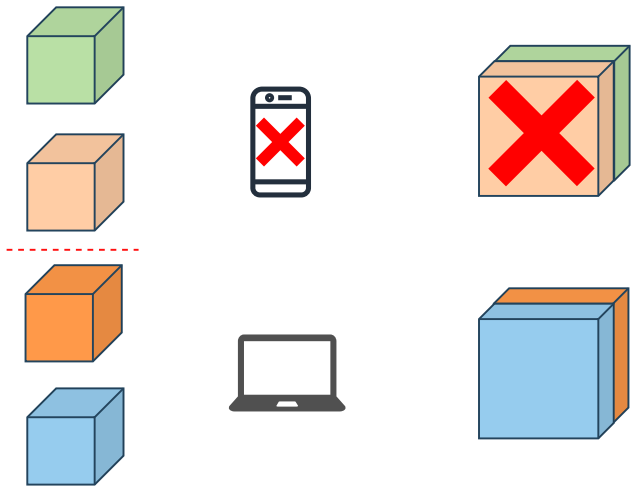
1. Unify the representation of Convolution Layer and InnerProduct Layer

2. Comparison of different partitioning methods for CNN distribution



(b) Partitioning Methods

Example: Distribute Neurons in a CNN layer into two devices: mobile and laptop



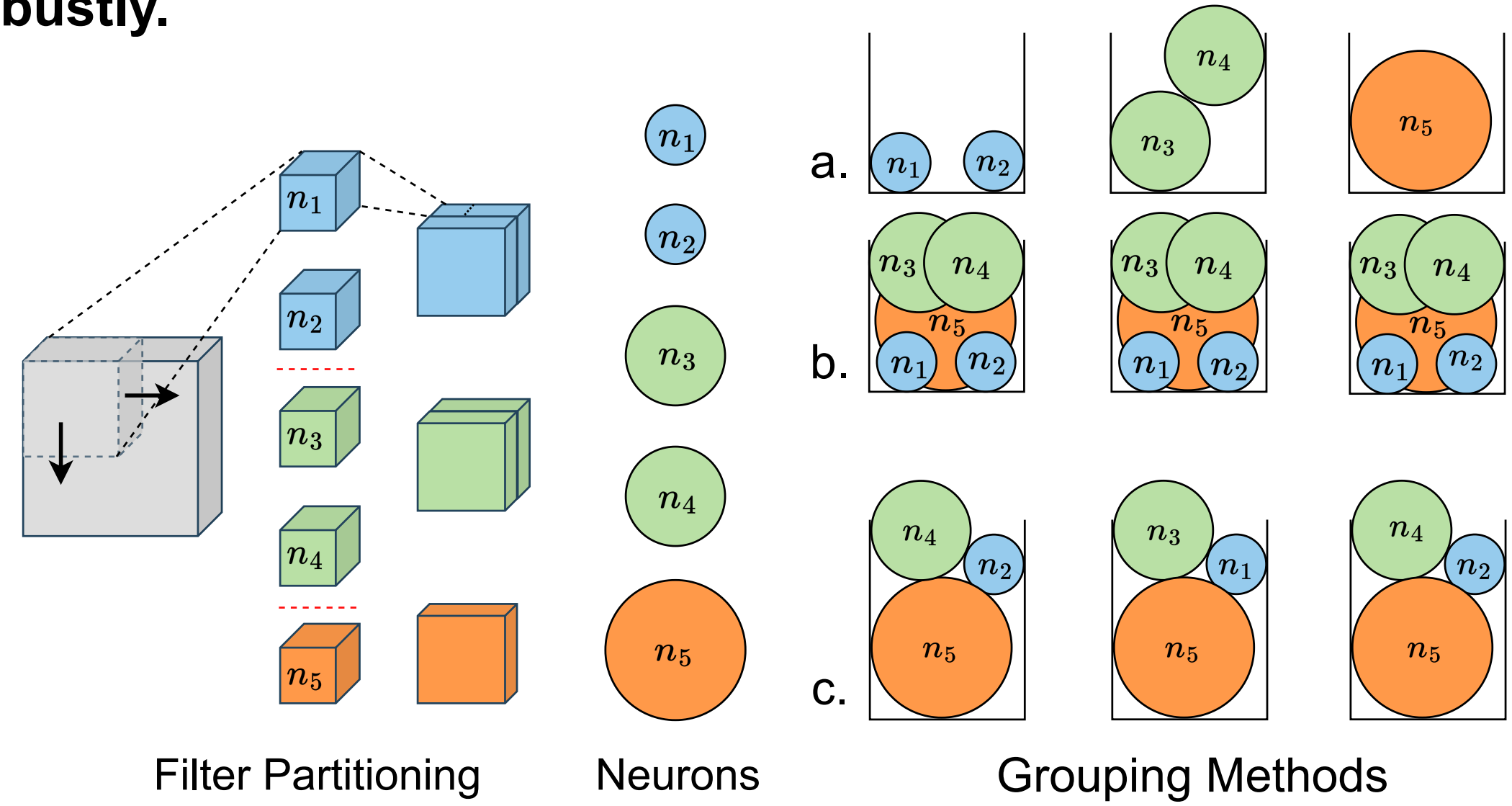
Output data loss
Due to Partition

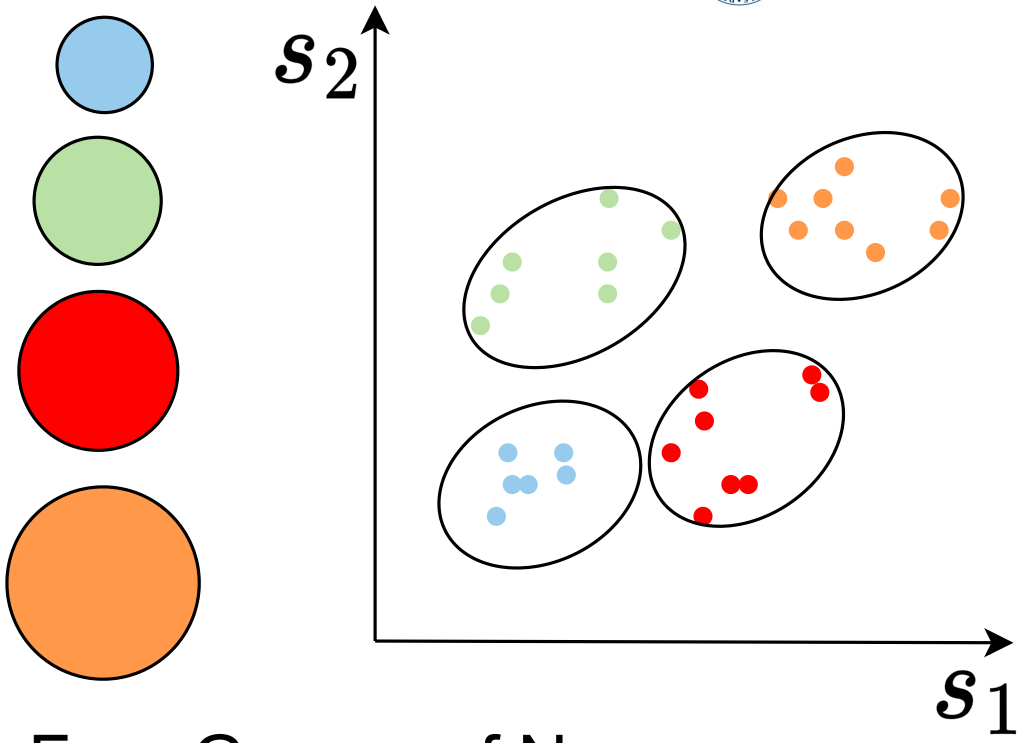
→ Don't put all your eggs in one basket

How to distribute neurons robustly over multiple edge devices?

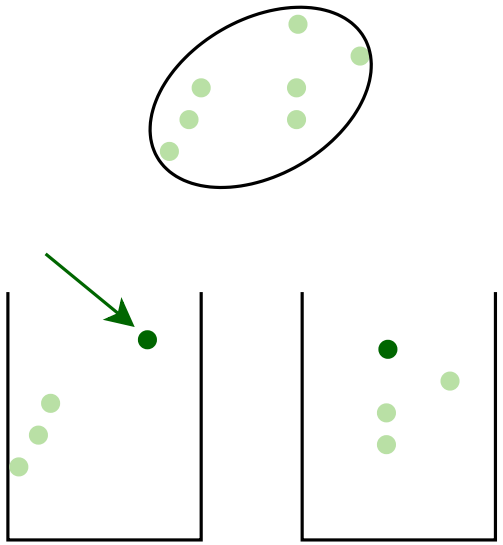
1. Ensure the execution against possible device failures
2. Largely preserve the important channels of output data
3. Balance the robustness with the resource usage per device

Our method (c) splits neurons into several groups according to neuron importance and distributes them over multiple devices robustly.





Four Groups of Neurons



2 Devices

1. **Calculate and Normalize Importance Scores**
2. **Cluster neurons into groups**
3. **Distribute neurons of each group in a round-robin manner**

II. Neuron Importance

1. Magnitude-based approach

L1-Norm, L2-Norm, LAMP, etc. measure the relative importance of each neuron in a CNN layer based on the sum or square sum of its absolute weights.

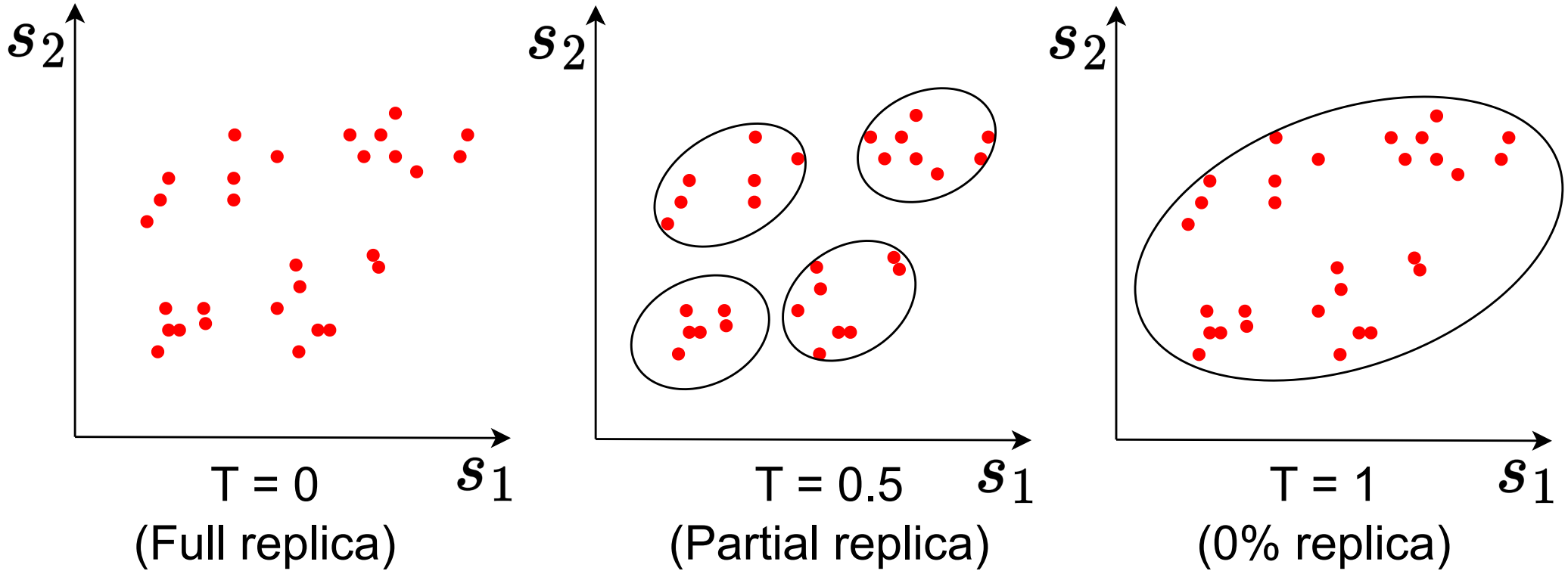
2. Data/Gradient-based approach

Taylor Expansion, SNIP, GraSP, etc. use the gradients of the training loss to effectively identify the connection sensitivity of neurons.

3. Loss-based approach

CURL, etc. approximate the change in the loss function induced by removing a neuron in CNN layers. The relative change of the loss value represents the importance of the removed neuron in the model.

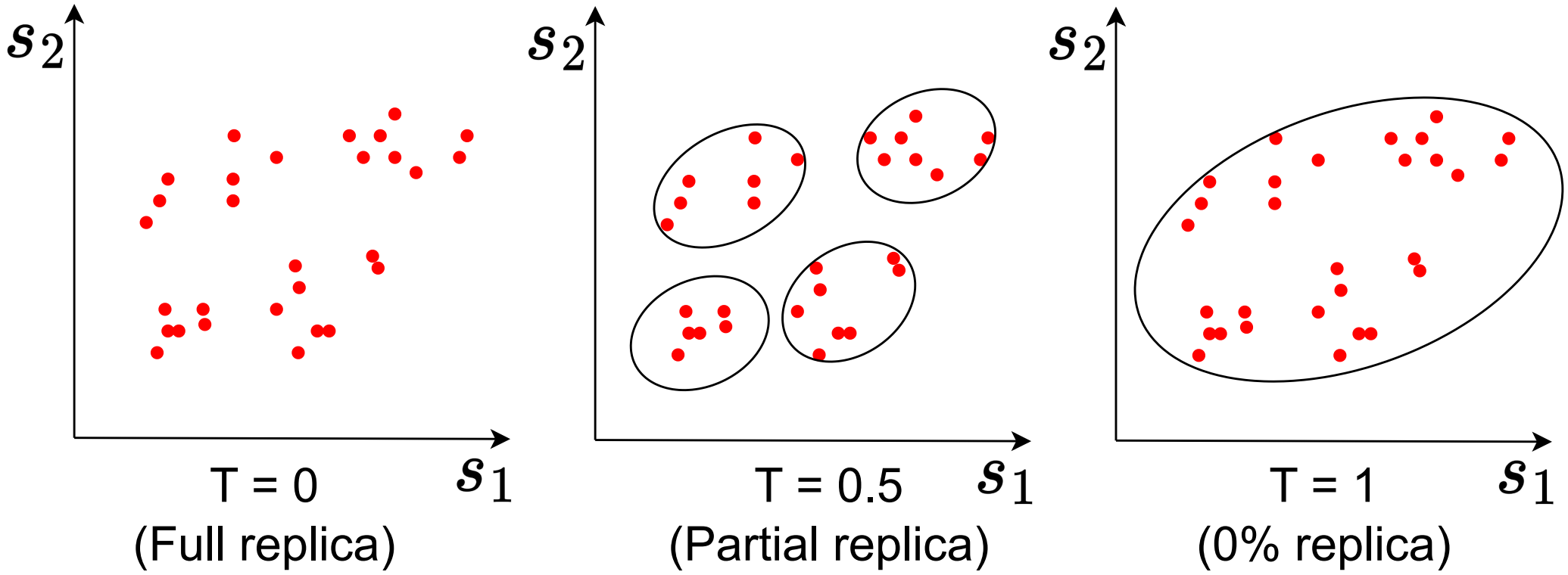
III. Neuron Grouping



Neuron Grouping/Clustering via Importance

1. Normalize Importance Scores
2. Measure neuron distance (Euclidean distance)
3. Use Distance Threshold value T control the size of group

III. Neuron Grouping



Maximum Robustness

More Memory

More Computation

Less Robustness

Less Memory

Less Computation

IV. System Setup

Eval Configurations

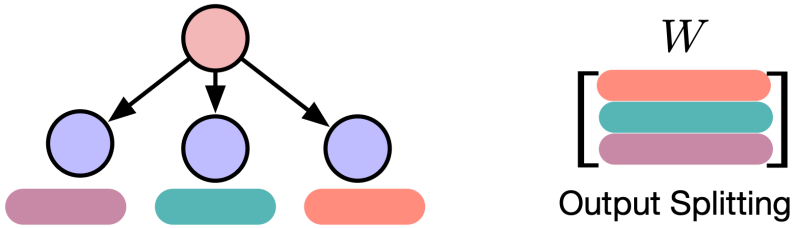
SysConf4D: system with four edge devices
SysConf3D: system with three edge devices
SysConf2D: system with two edge devices

Scenario A: SysConf4D where 1 device fails (1D-Fail), 2 devices fail (2D-Fail), or 3 devices fail (3D-Fail)
Scenario B: SysConf3D where 1D-Fail or 2D-Fail
Scenario C: SysConf2D where 1D-Fail

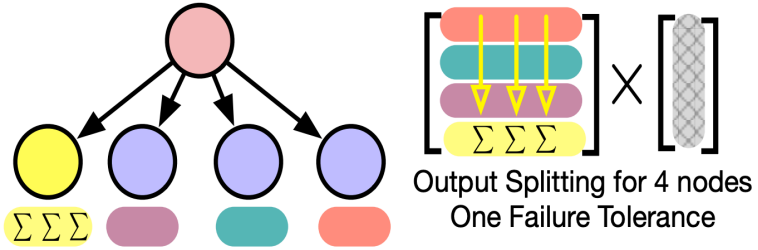
Eval Metrics

Accuracy: Top-1 accuracy on ImageNet-1k
FPS: image per second (system throughput)
Memory: maximum memory usage per device
Energy: maximum energy consumption per device

3D-Fail Example: 3 Devices fail out of 4 Devices



LOP-Method For Comparison [7]



CDC-Method For Comparison [11]

V. Experimental Results

We fix T values of each layer and test different combinations of using s1, s2, s3 importance scores

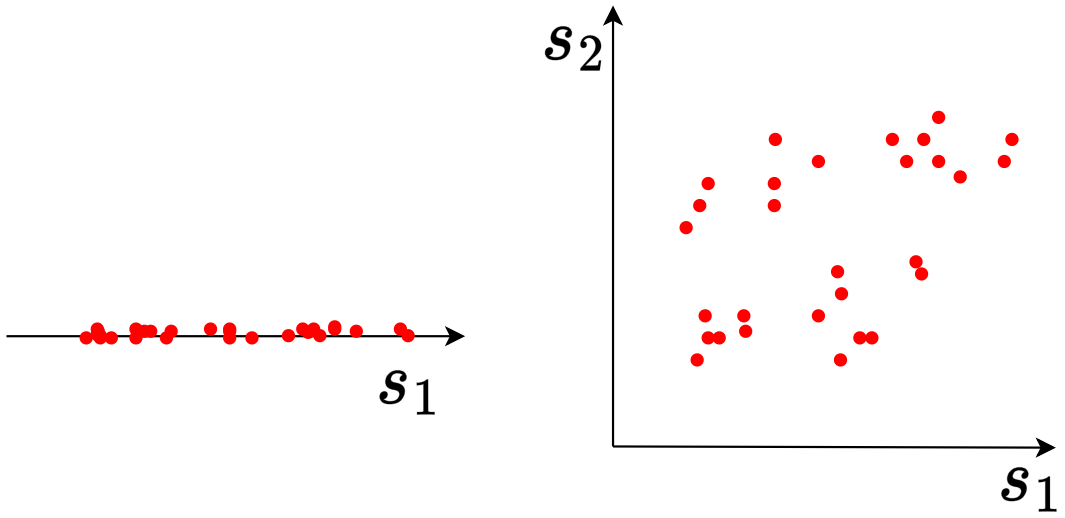
Importance Scores	AlexNet (%)	VGG16_BN (%)	ConvNext_Tiny (%)
s_1	43.718	60.426	76.618
s_2	43.642	58.920	75.904
s_3	43.432	59.942	76.134
$s_1 + s_2$	51.268	69.152	76.678
$s_1 + s_3$	51.658	71.736	76.580
$s_2 + s_3$	51.250	67.360	76.572
$s_1 + s_2 + s_3$	**52.396**	**72.500**	**76.820**

Ablation study for Importance Metrics

1 Device fail out of 4 Devices (1D-Fail in SysConf4D)

More dimensional evaluation of the neuron importance

Facilitate a more effective clustering of neurons

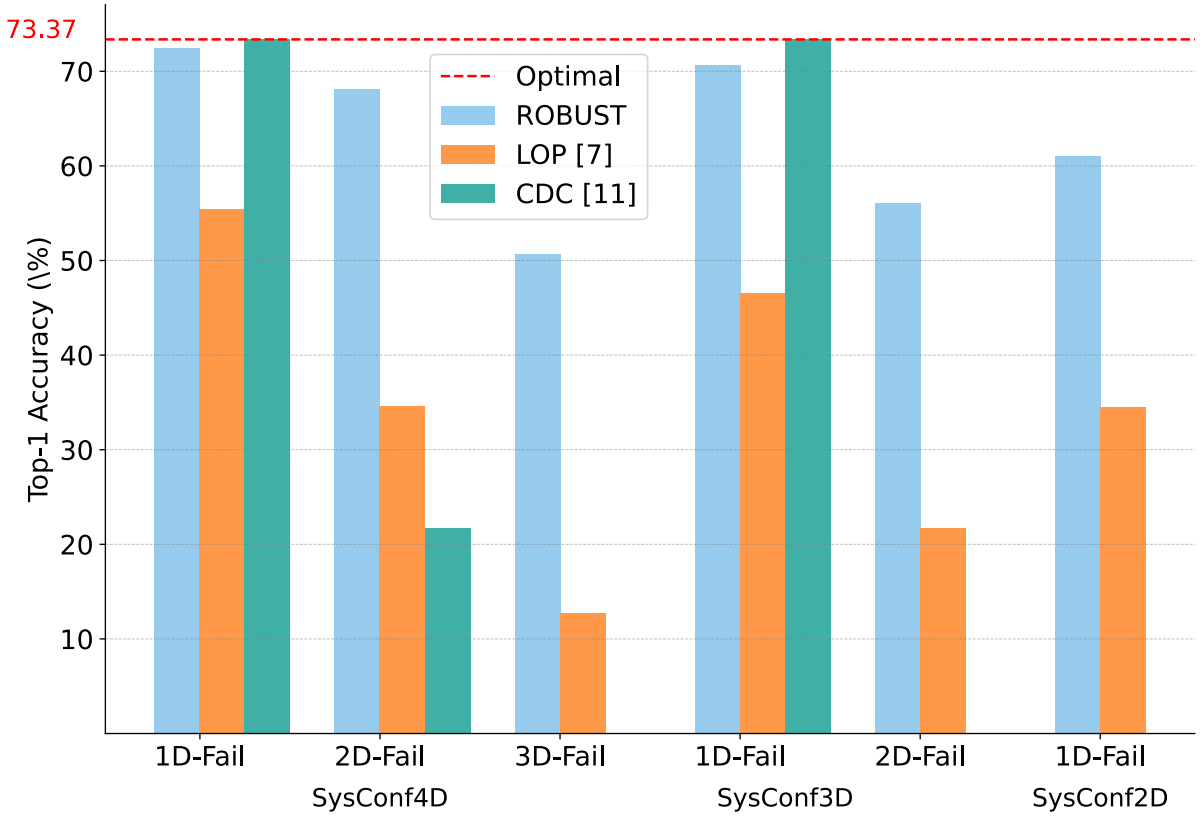


Comparison

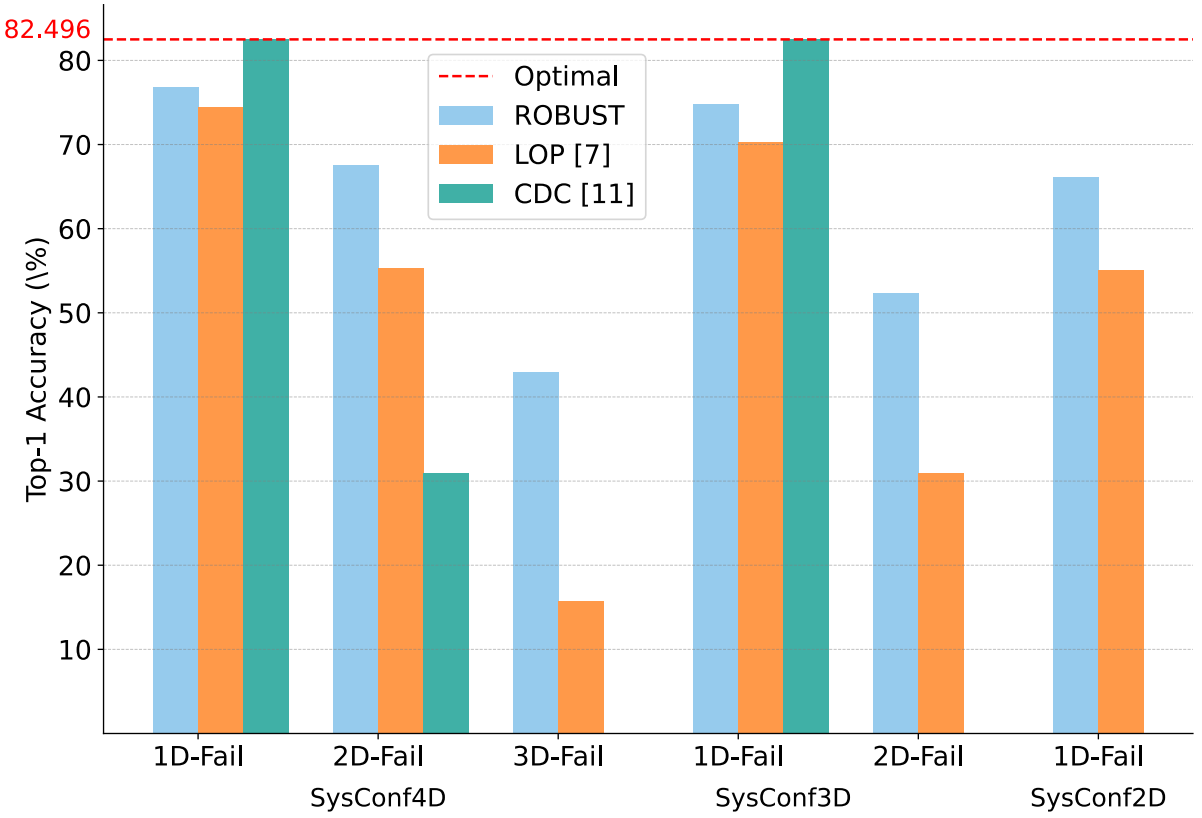
Fix T values of each layer

Use s1, s2, s3 importance scores

Network	System Configuration	Max. per-device Energy (J/img)	System Throughput (FPS)	Max. per-device Memory (MB)
AlexNet	QMR/TMR/DMR	0.179	46.255	150.914
	CDC-SysConf3D	0.165	43.670	94.117
	CDC-SysConf4D	0.157	45.587	78.852
	Robust-SysConf2D	0.159	48.214	99.254
	Robust-SysConf3D	0.148	50.045	80.777
	Robust-SysConf4D	0.142	51.219	72.801
VGG16-BN	QMR/TMR/DMR	0.850	10.744	429.215
	CDC-SysConf3D	0.809	10.634	313.688
	CDC-SysConf4D	0.799	10.485	272.293
	Robust-SysConf2D	0.826	10.761	328.426
	Robust-SysConf3D	0.799	10.993	295.086
	Robust-SysConf4D	0.779	11.078	267.395
ConvNext-Tiny	QMR/TMR/DMR	0.308	28.223	88.895
	CDC-SysConf3D	0.307	27.107	69.129
	CDC-SysConf4D	0.297	28.248	59.961
	Robust-SysConf2D	0.301	28.044	76.465
	Robust-SysConf3D	0.296	28.415	65.203
	Robust-SysConf4D	0.288	29.034	58.090



VGG-16



ConvNext-Tiny

Robustness Under Failures

Take Away

1. Neuron Importance

Various importance scores offer unique insights, e.g. gradient-based scores assess inter-layer dependencies, magnitude-based scores evaluate intra-layer dependencies, etc.

2. Neuron Clustering

Utilizing a combination of importance scores enhances overall effectiveness.

3. Importance-aware Partitioning

Importance-aware partitioning maintains CNN model accuracy on multiple edge devices more effectively against device failures than current partitioning methods.

UNIVERSITY OF AMSTERDAM

Universiteit Leiden
The Netherlands

Thanks!
Questions?