A Heuristic Exploration of Retraining-free Weight-Sharing for CNN Compression

Etienne Dupuis, Ian O'Connor, David Novo, Alberto Bosio

ASP-DAC -2022

ADEQUATE DL

Institut des Nanotechnologies de Lyon UMR 5270

LIRMM

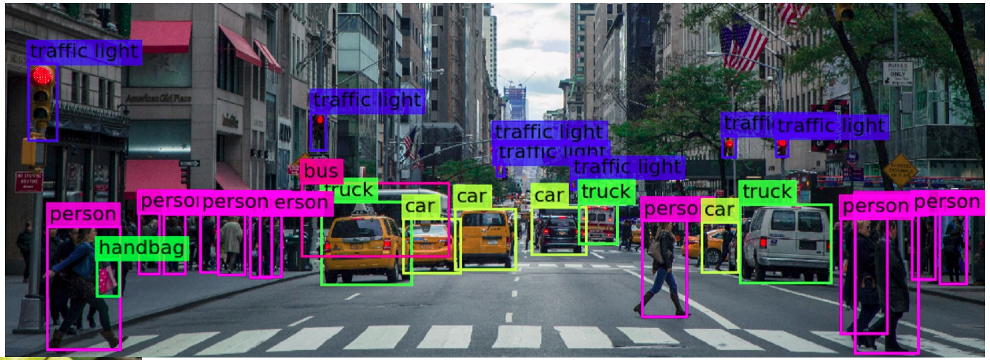
Outline

I. Computer vision in embedded devices
II. Approximate computing in CNNs
III. Weight sharing principles and challenges
IV. Divide & conquer strategy
V. ImageNet results
VI. Conclusions

Computer vision in embedded devices

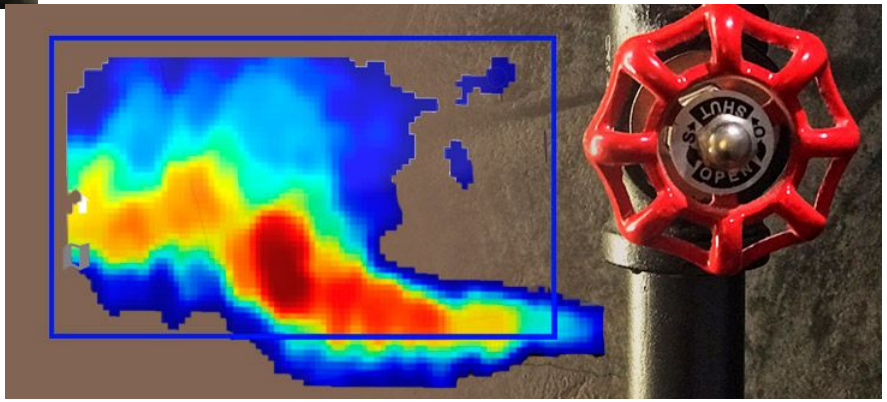
Computer Vision

Autonomous Driving

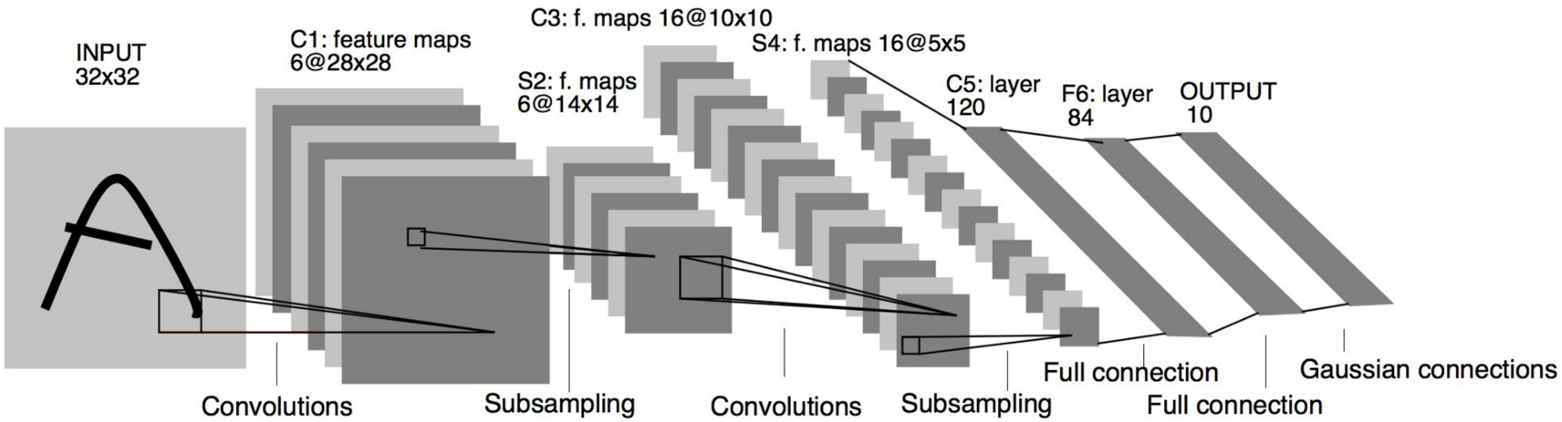


Quality Monitoring

Predictive Maintenance



Convolutional Neural Networks



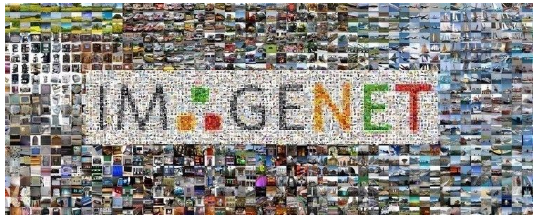
Lecun & al., 1998

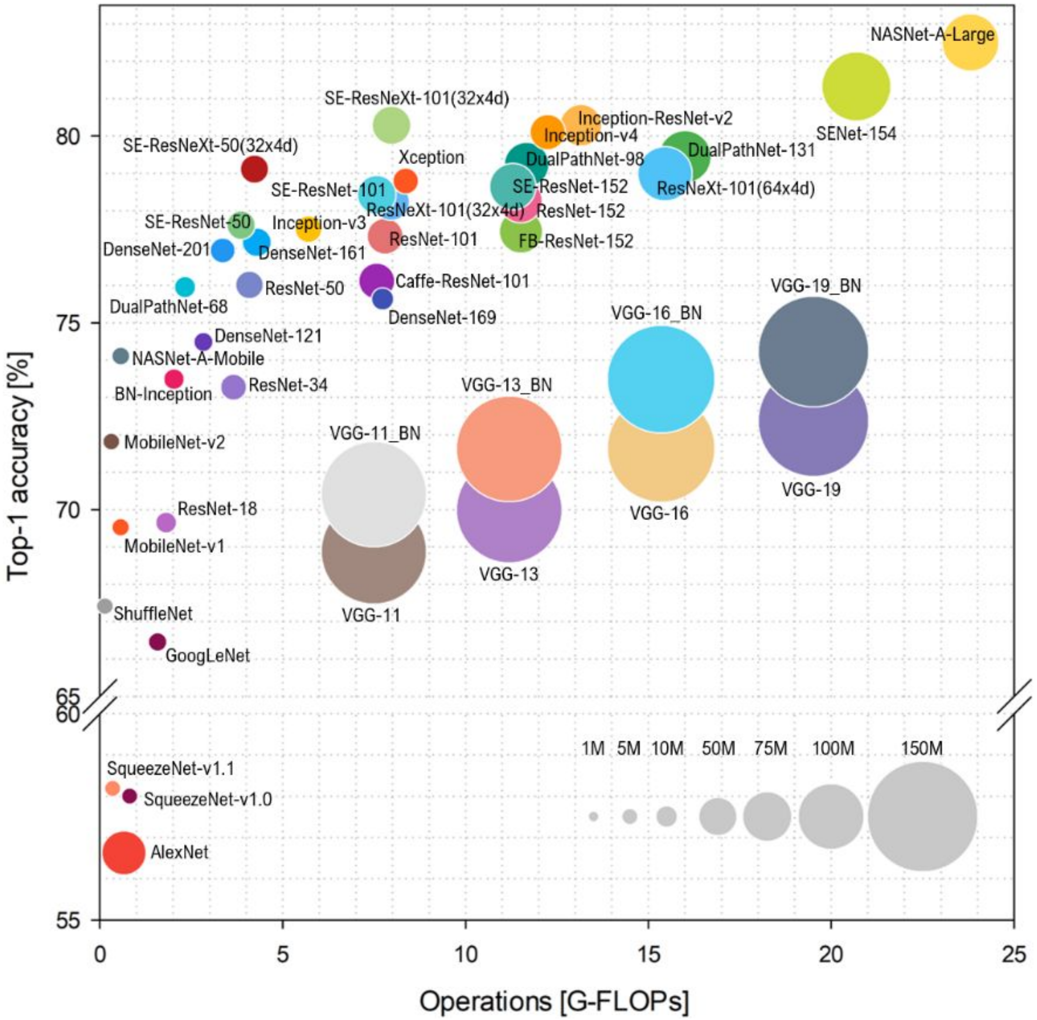
Kernel = 2D matrix of weights

Execution cost

High Resolution Image

Low Resolution Image



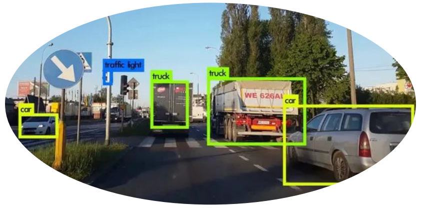


Bianco & Al, IEEE Access, 2018

CNN in Embedded Devices

Specific Constraints

- Computation Resources
- Memory
- Power

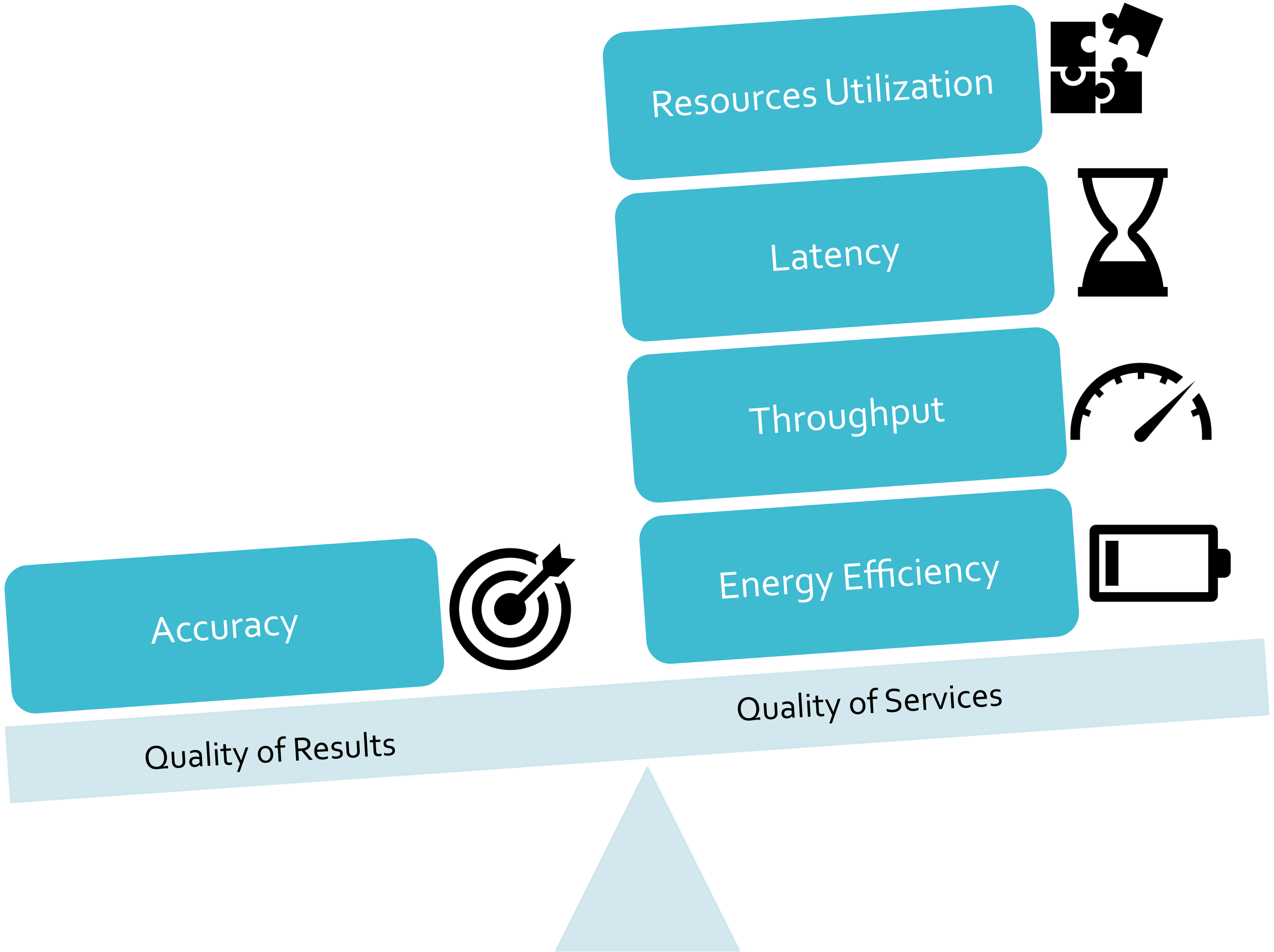


"For example, smartphones nowadays cannot even run object classification with AlexNet in real-time for more than an hour" [1]

[1] Yang & al. CVPR, 2017

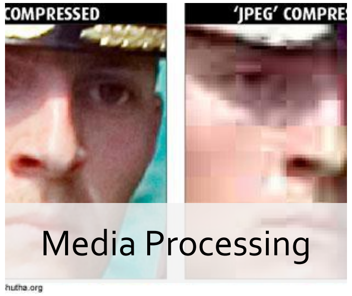
Approximate computing in CNNs

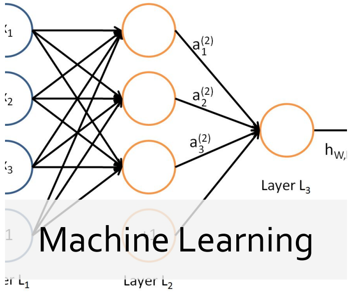
Approximate computing



Error resilient applications

Many applications are **error resilient** [1]





CNN resilience factors

Empirical Training

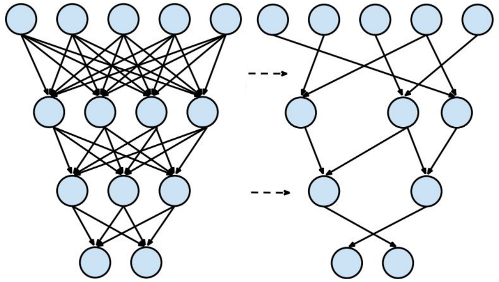
- Algorithm level noise tolerance

High Level of Redundancy

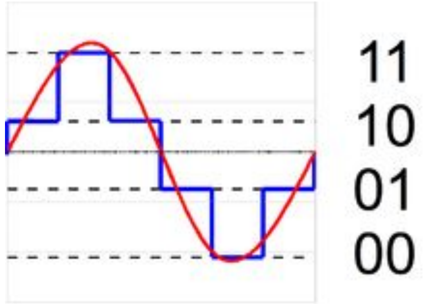
- Layer Connections
- Weights Values
- Values Encoding

[1] Chippa & al. DAC, 2013

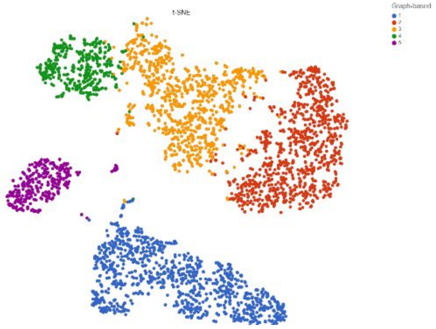
Approximate Computing Techniques for CNNs



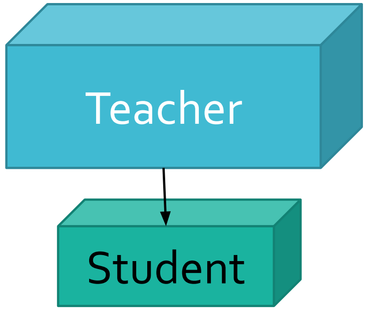
Pruning [1]



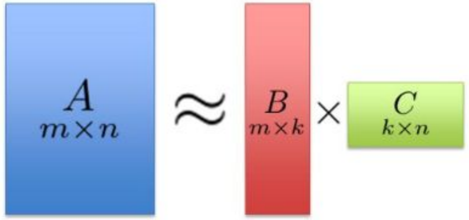
Quantization [2]



Weight-sharing [4]



Knowledge distillation [5]



Low-rank factorization [3]

[1] Han & al. NIPS, 2015 [2] Zhu & al. ICLR, 2017 [3] Sainath & al. IEEE ICASS, 2013
[4] Takahashi & al. CoRR, 2017 [5] Hinton & al. NIPS, 2014

Weight sharing principles and challenges

Weight Sharing

- Group weights values together
- Store a single **shared value** per group
- Use smaller **index** in weight matrix

17	3	18	11	19
5	14	9	13	7
19	5	7	0	12
10	12	20	8	10
14	1	16	10	14

e	a	e	c	e
b	d	b	d	b
e	b	b	a	c
c	c	e	c	c
d	a	d	c	d

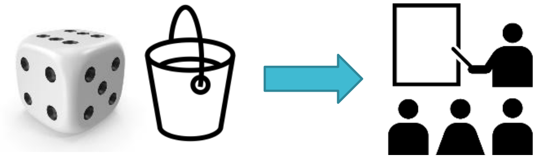
a	0
b	5
c	10
d	15
e	20

$$Compression\ Rate = \frac{Baseline\ Model\ Memory}{Compressed\ Model\ Memory} = \frac{W * B}{W * ceil(\log_2 K) + K * B}$$

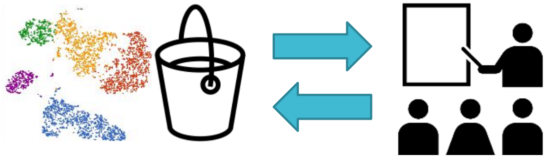
Number of weights (W), number of clusters (K), bit used to represent a weight (B)

$$\text{Compression rate (CR)} = \frac{25 * 32}{25 * ceil(\log_2 5) + 5 * 32} = \frac{800}{235} = 3.4$$

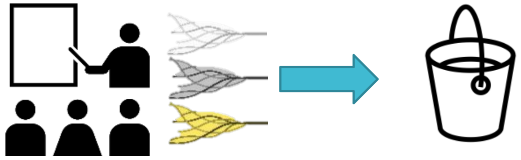
Where we Have Been in Weight Sharing



Random grouping
Hashed Net[1]



Iterative grouping/training
Deep Compression[2]



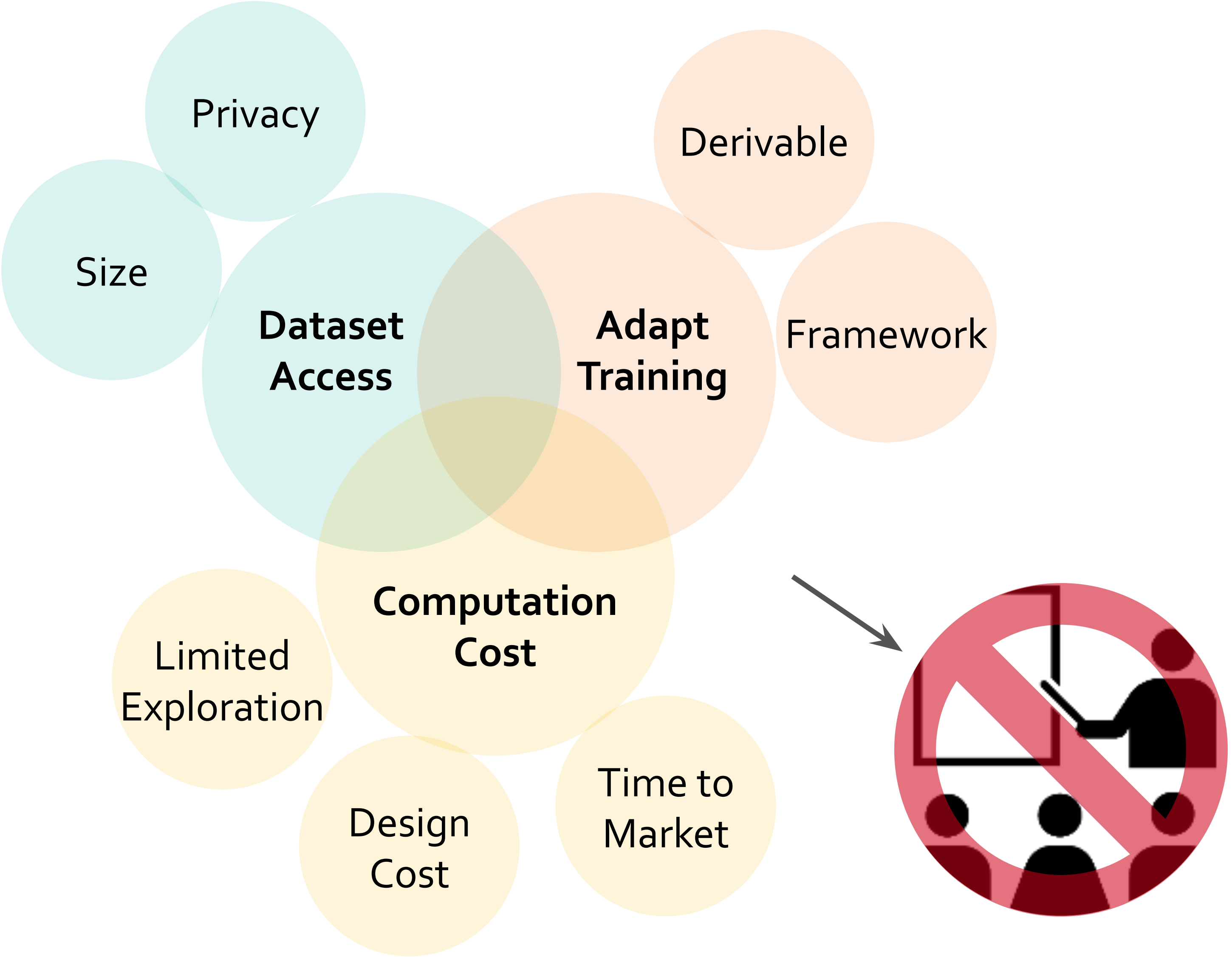
Regularized training
Deep K-means[3]
Soft Weight Sharing [4]

	000	001	010
000	000000	000000	000000
001	000000	000001	000010
010	000000	000010	000100

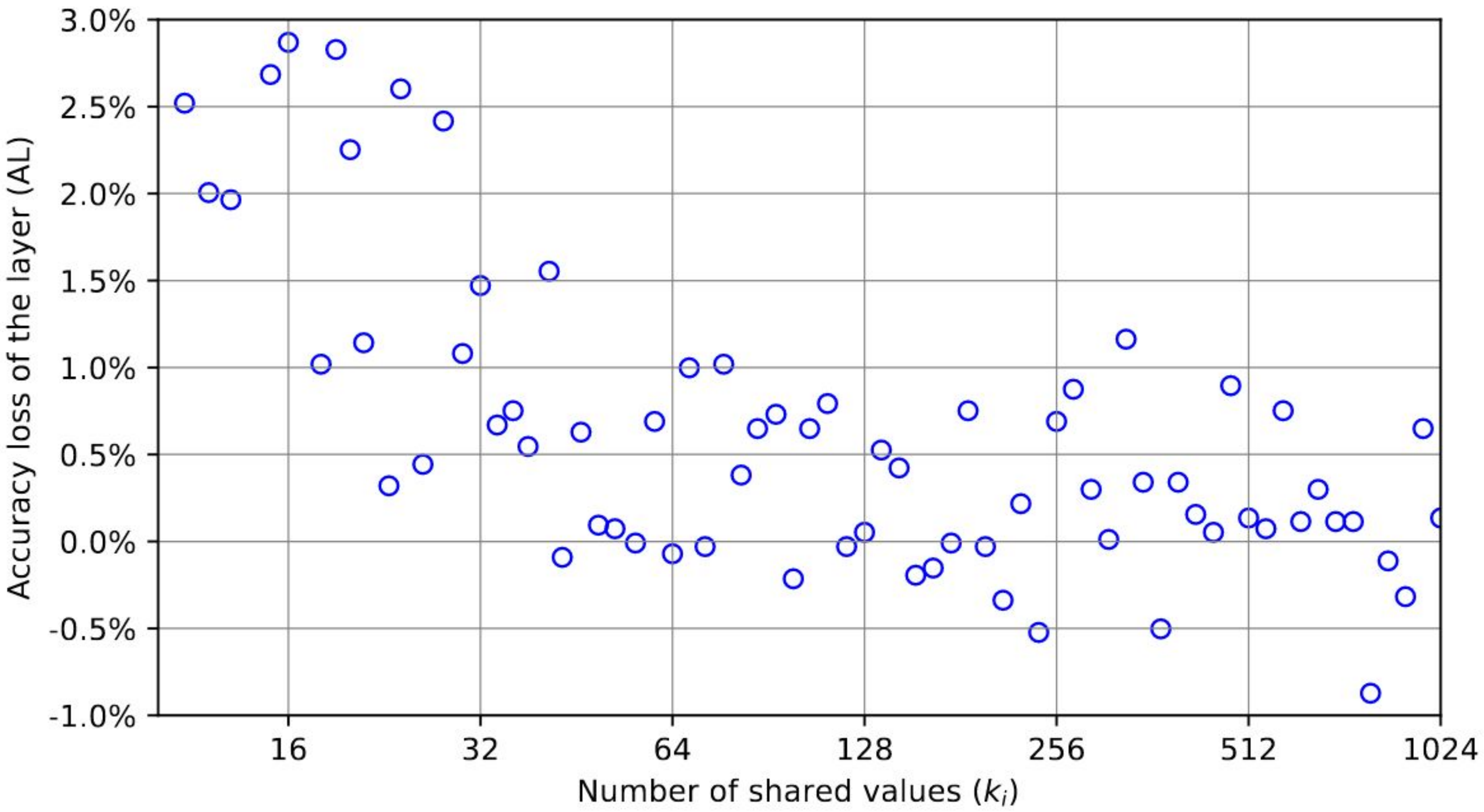
LUT multiplication
LookNN[5]
QuantizedNN[6]

[1] Chen & al. CoRR, 2015
[2] Han & al. ICLR, 2016
[3]Wu & al. ICML, 2018
[4]Ullrich & al. ICLR, 2017
[5] Razlighi & al. DATE, 2017
[6] Hubara & al. CoRR, 2016

Retraining Issues



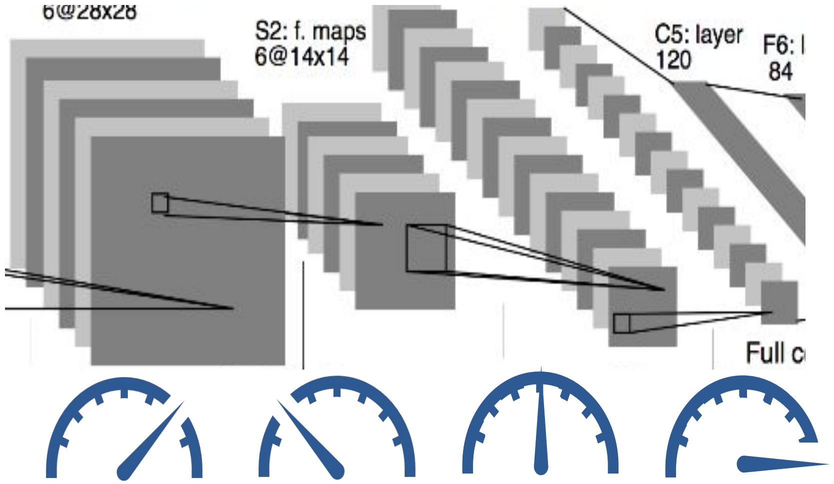
Layer sensitivity to approximation



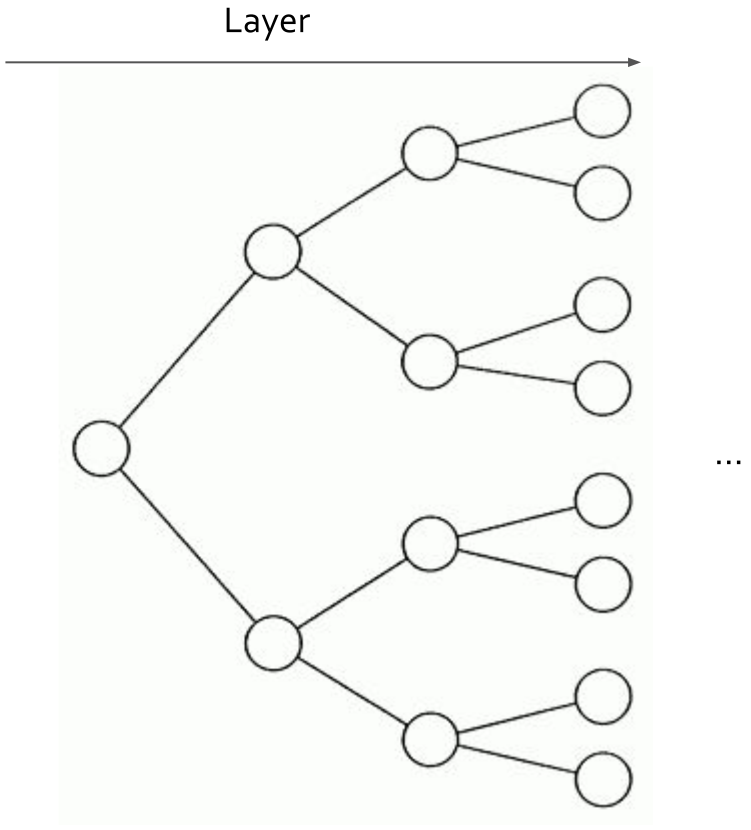
ResNet50V2, first layer

Research Question

Find the optimal number of shared values for each layer of a CNN



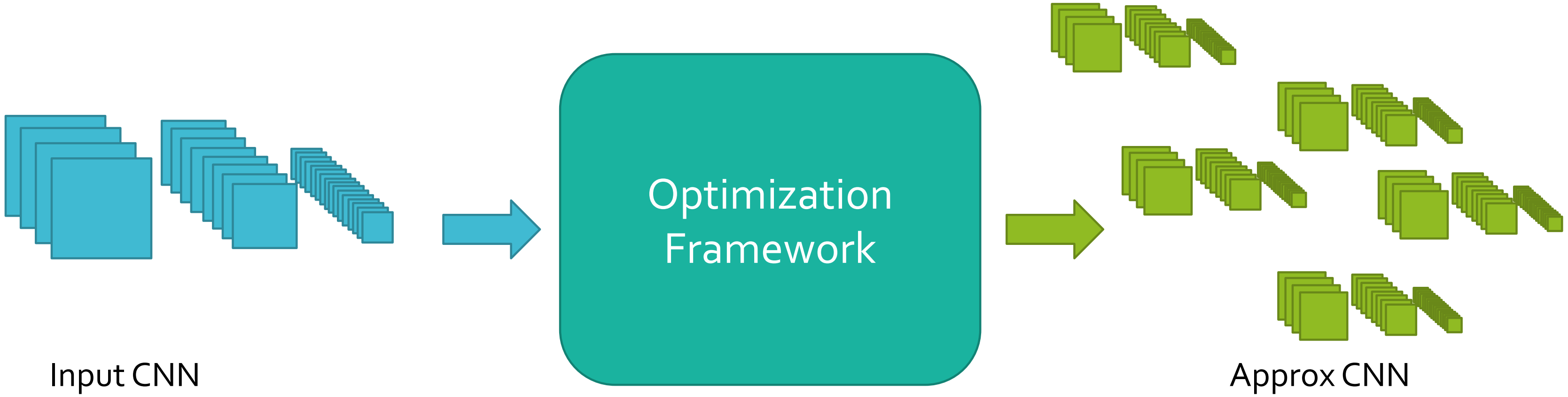
Search space issue



100 possible number of shared values
5 layers (Lenet)
=> 10 billions possible combinations

Divide & conquer strategy

Proposed Framework

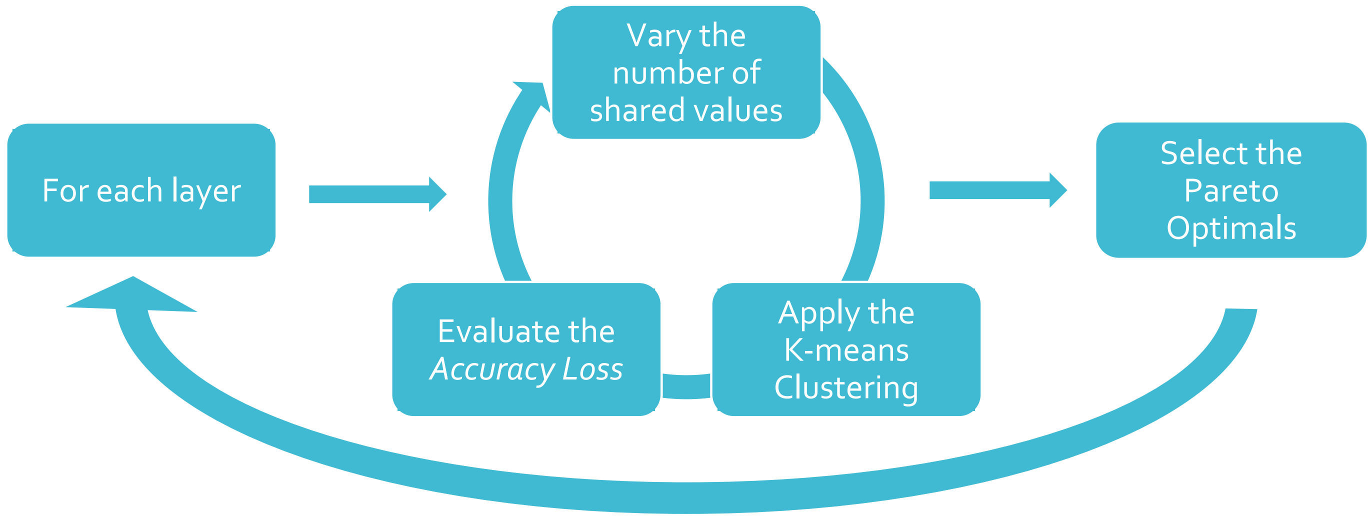


Two-Steps Approach

Solution space reduction

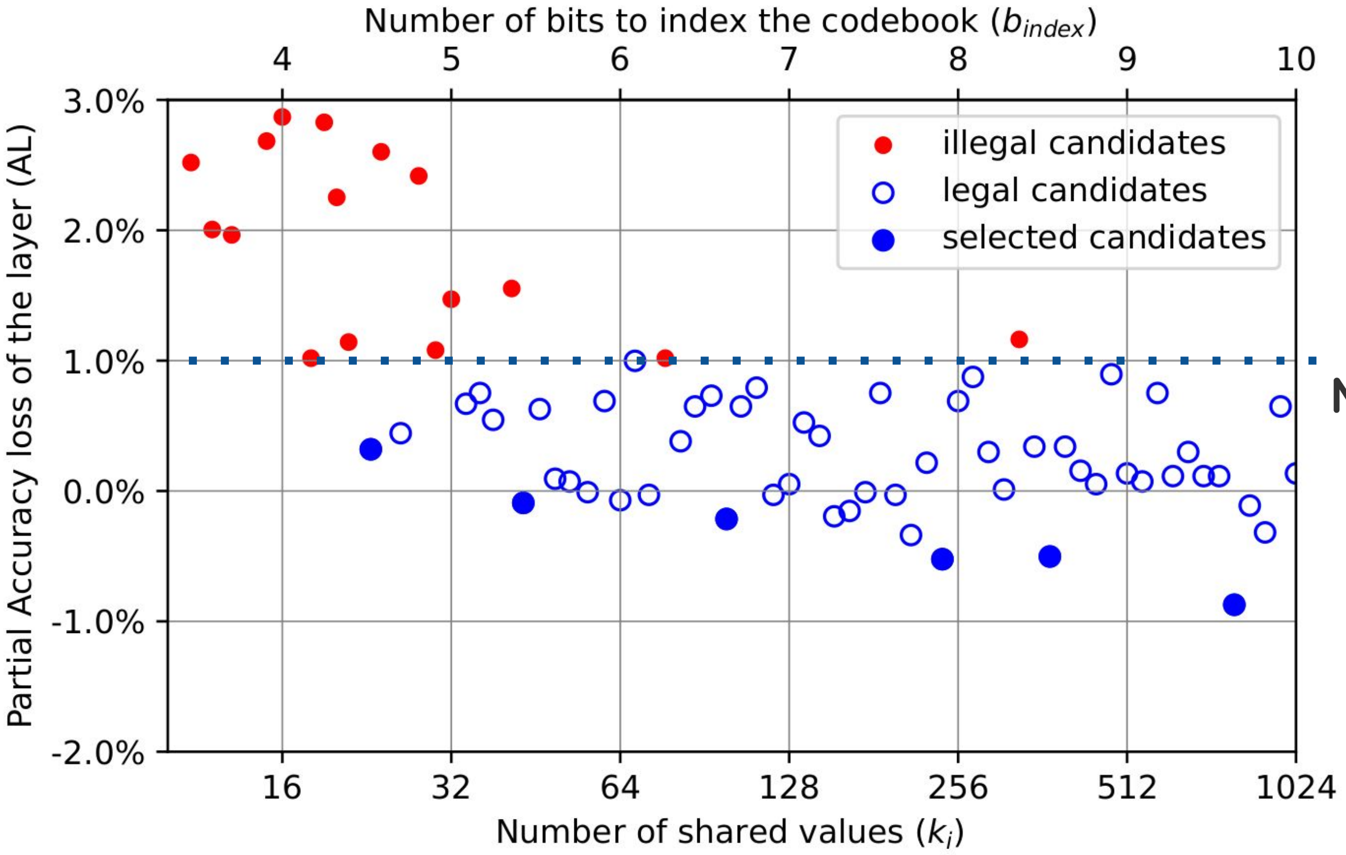
Heuristic optimization

Layer Optimization Method



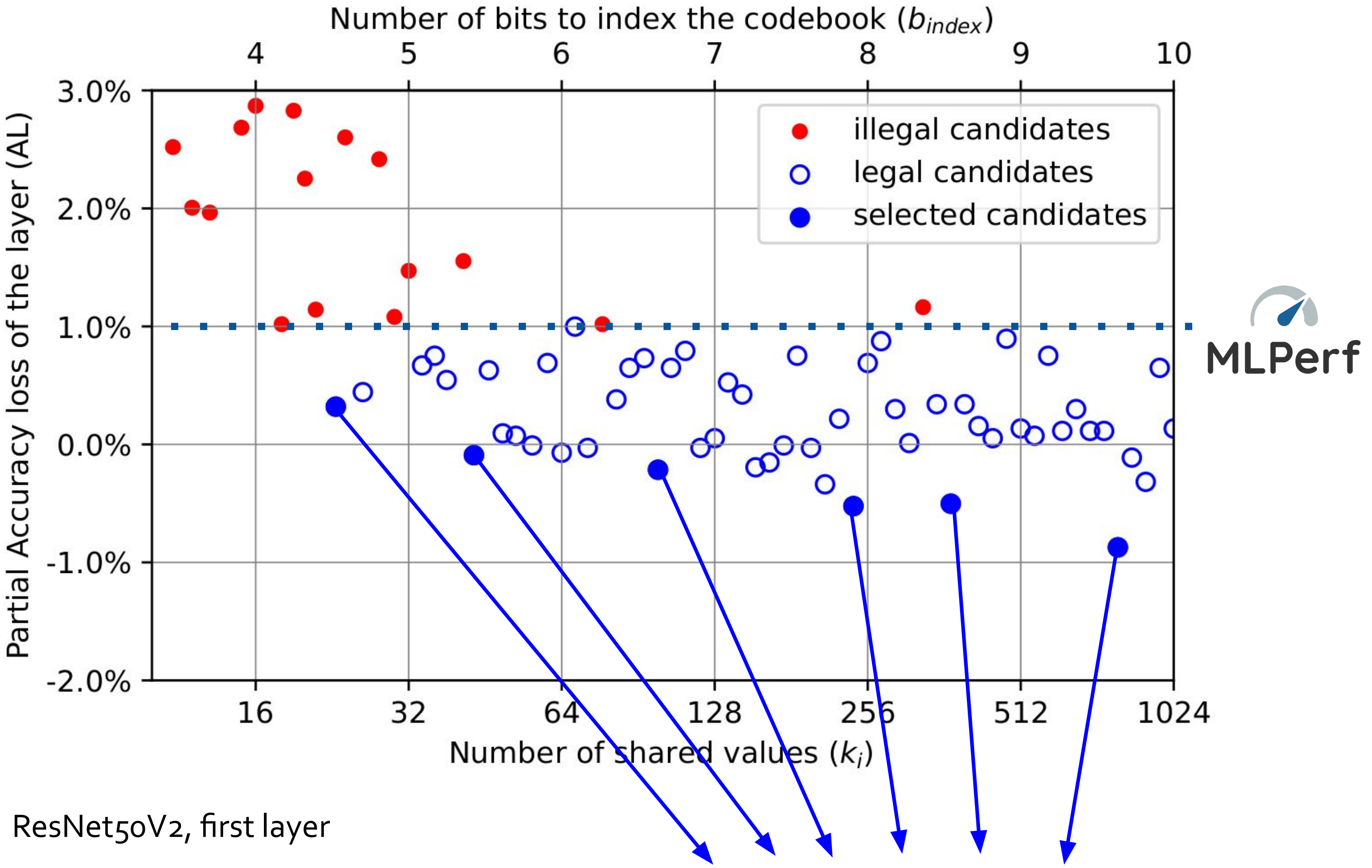
a	0
b	5
c	10
d	15
e	20

Layer Optimization Example



ResNet50V2, first layer

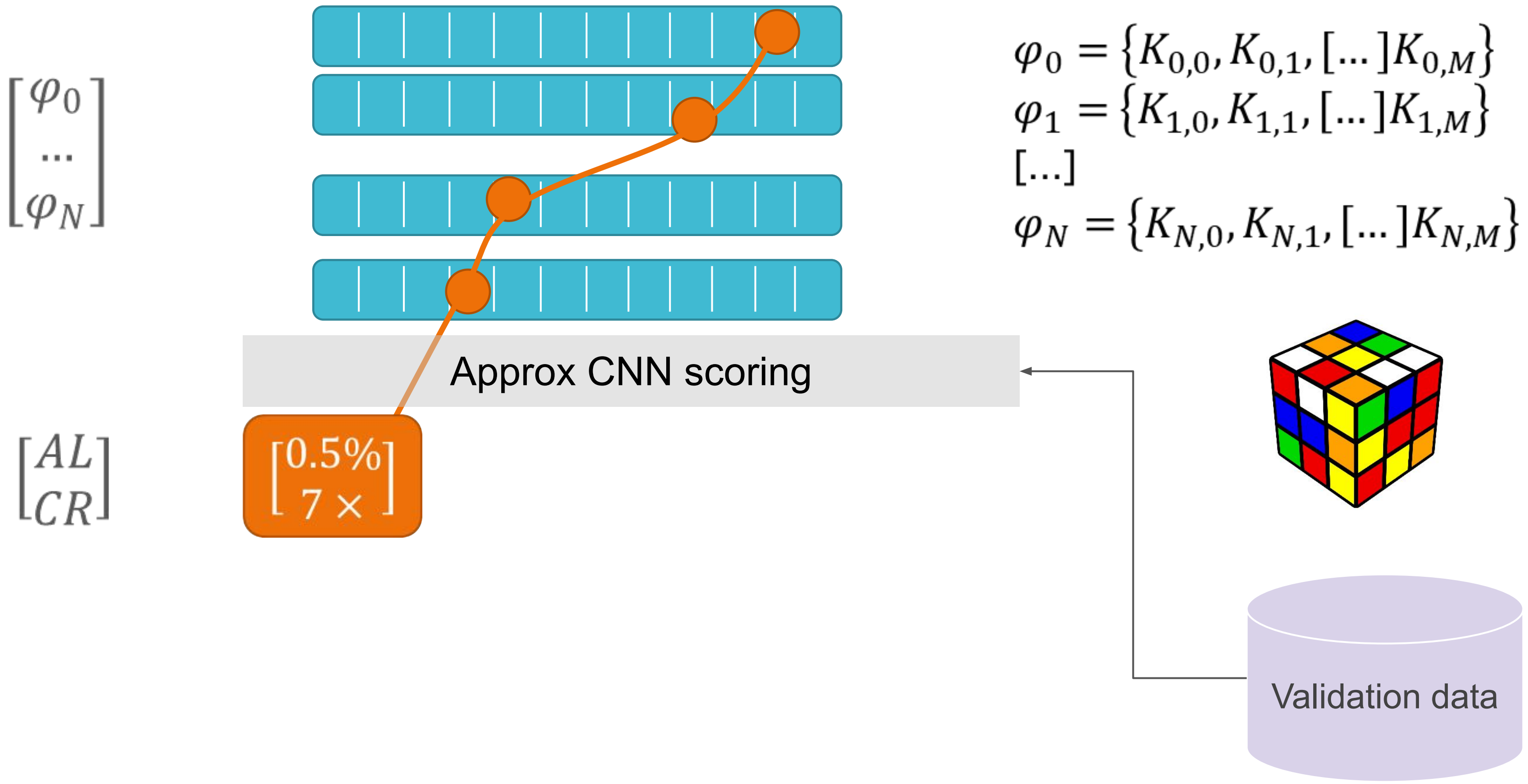
Layer Optimization Example



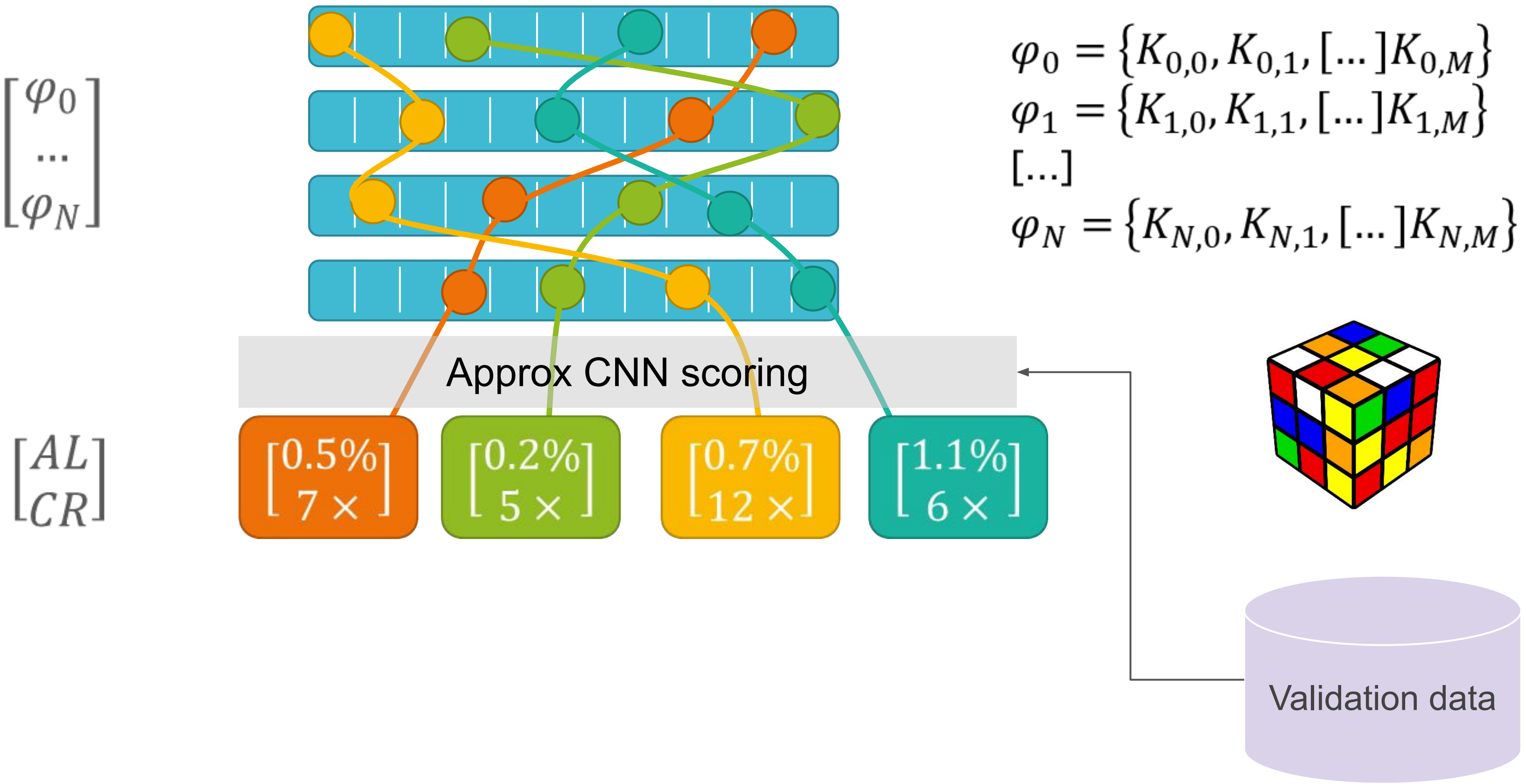
ResNet50V2, first layer

φ_1 = {23, 49, 86, 248, 302, 860}

Network optimization

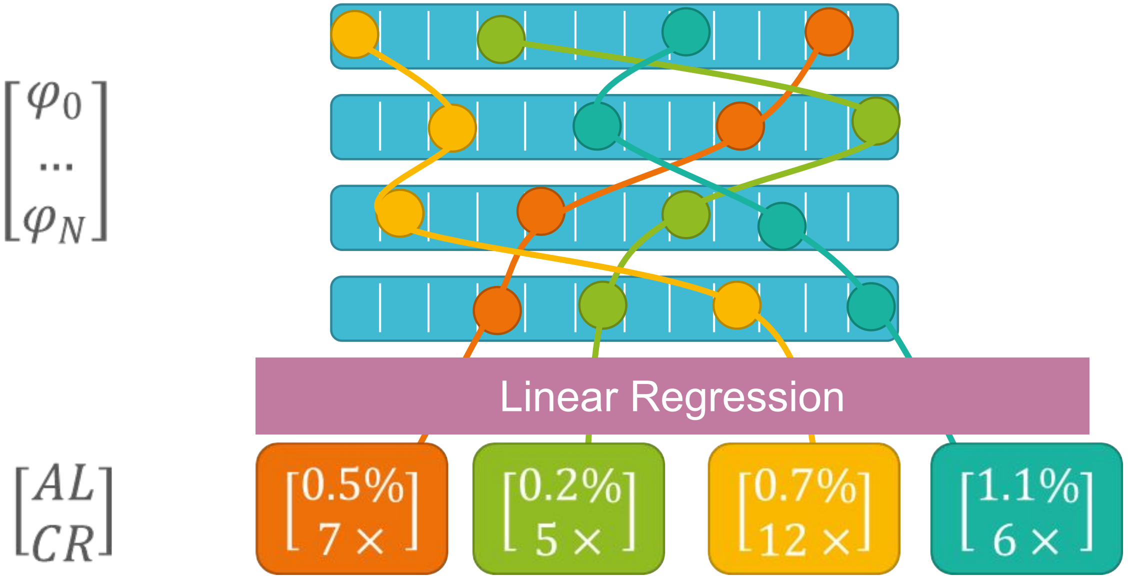


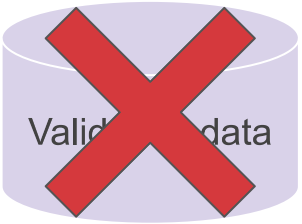
Network optimization



Linear Regression

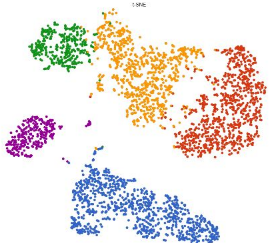
- Avoid the cost of combination evaluation
- Accuracy loss(Approx CNN) = F(Approx Layers Accuracy Loss)

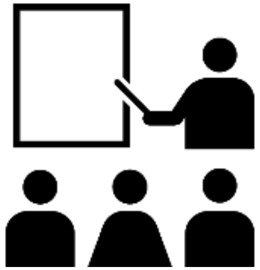




Regression Training Data

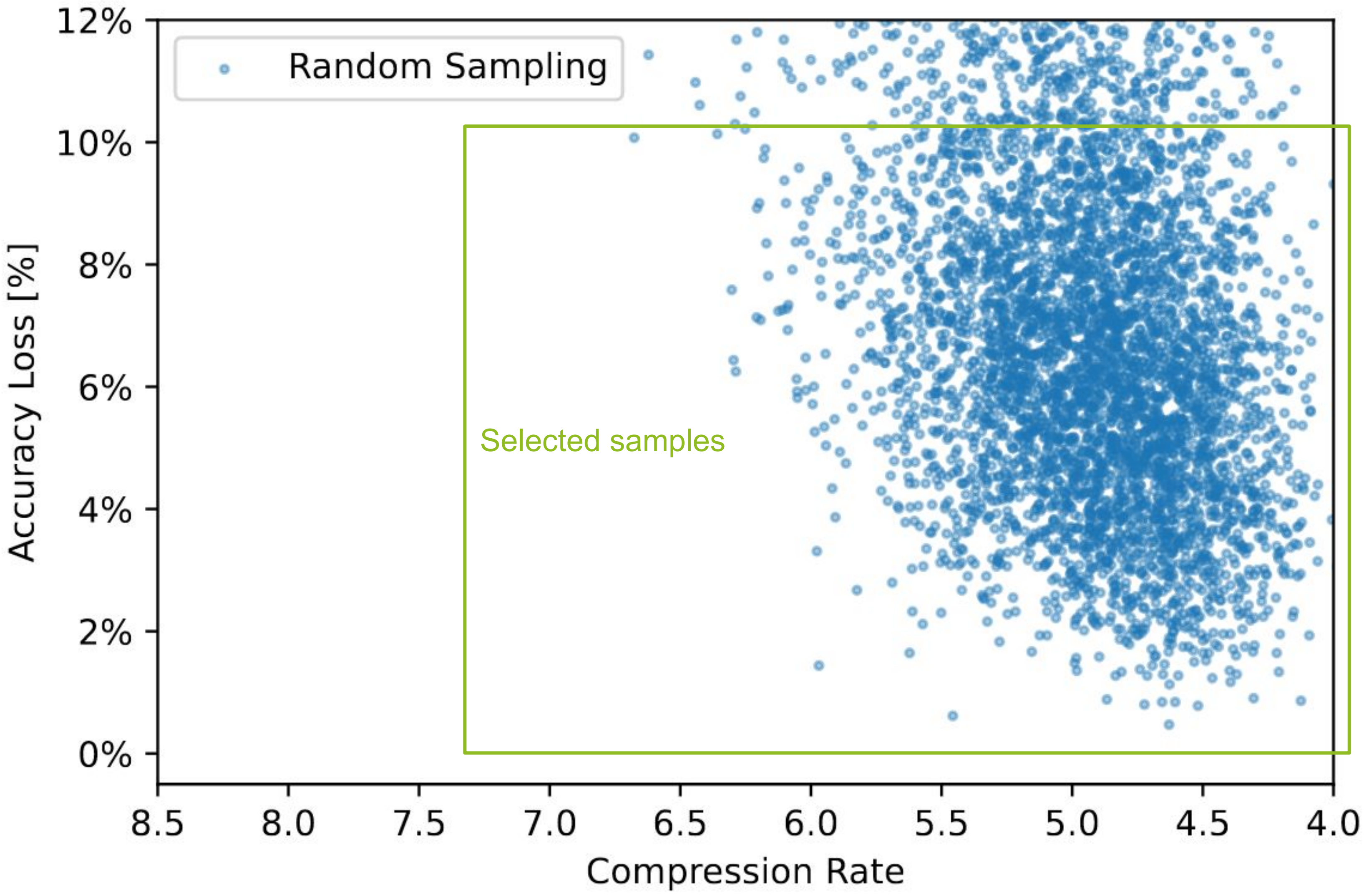
1. Random sub-sampling the search space
2. Apply each approximations
3. Evaluate each candidates
4. Train the regression model





Regression Training Data

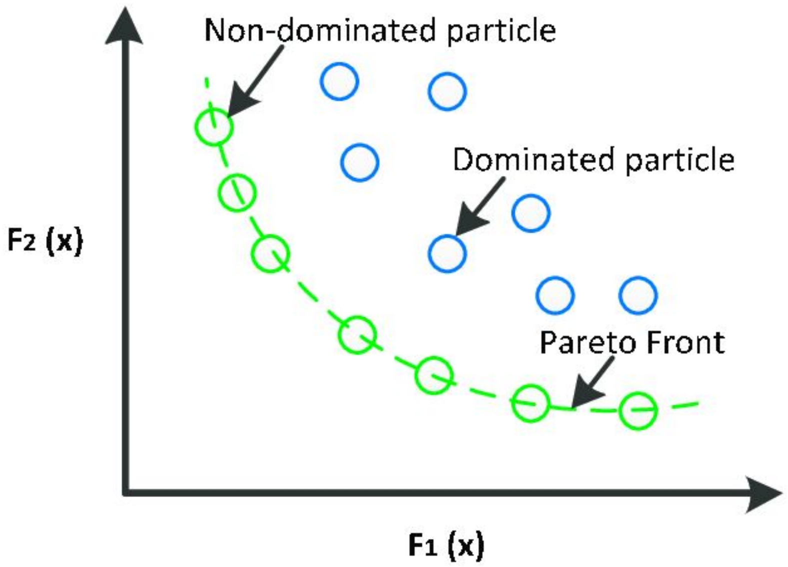
5K samples over the 10^{54} possible combinations



Resnet50V2, Linear regression model R^2 = 76%

Pareto Improvement

- NSGA-II Genetic Algorithm [1]
- Using the regression model

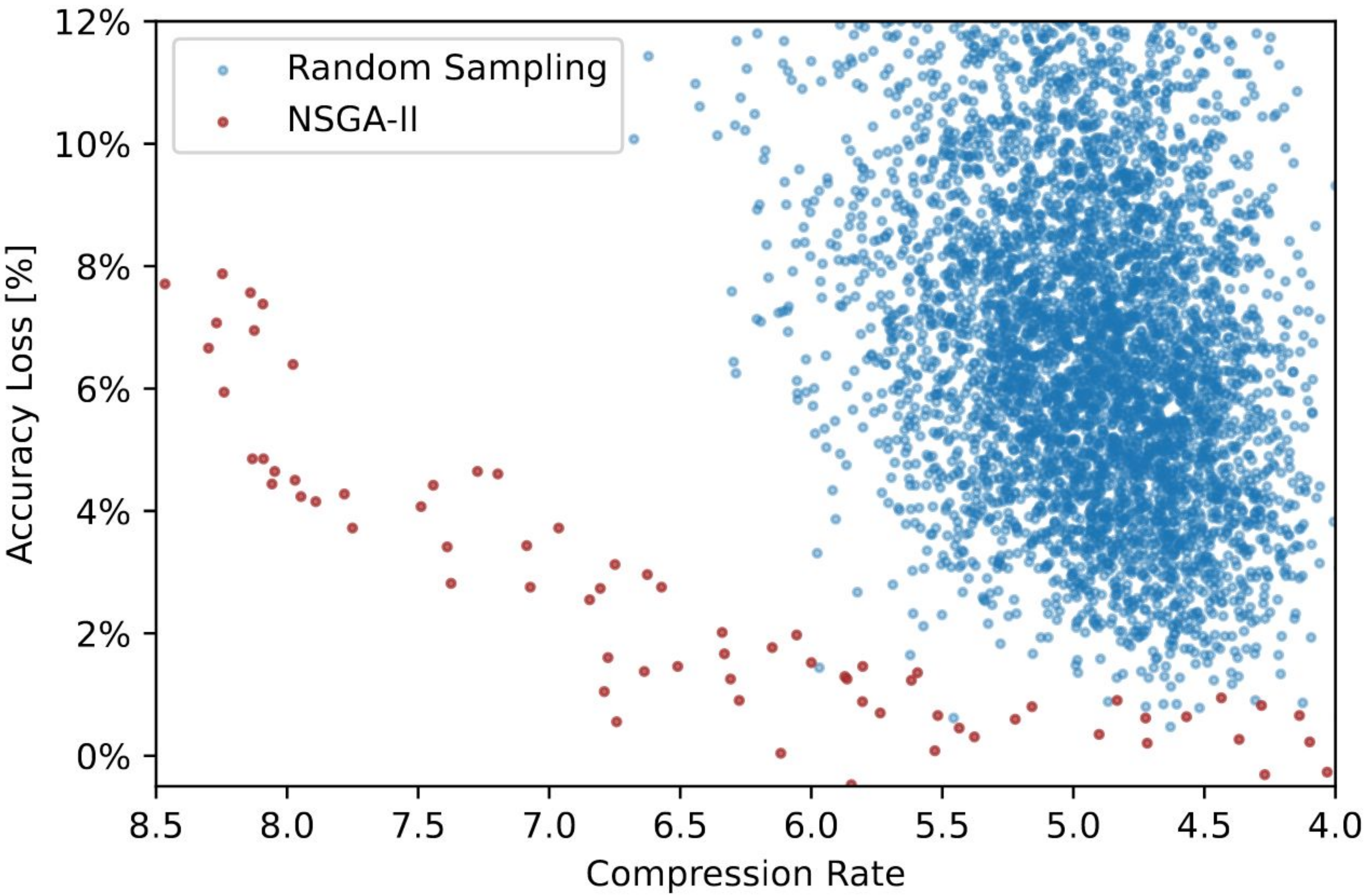


1. Population Evaluation
2. Non-Dominated Selection
3. Breeding + Mutation

(Exploitation/Exploration)

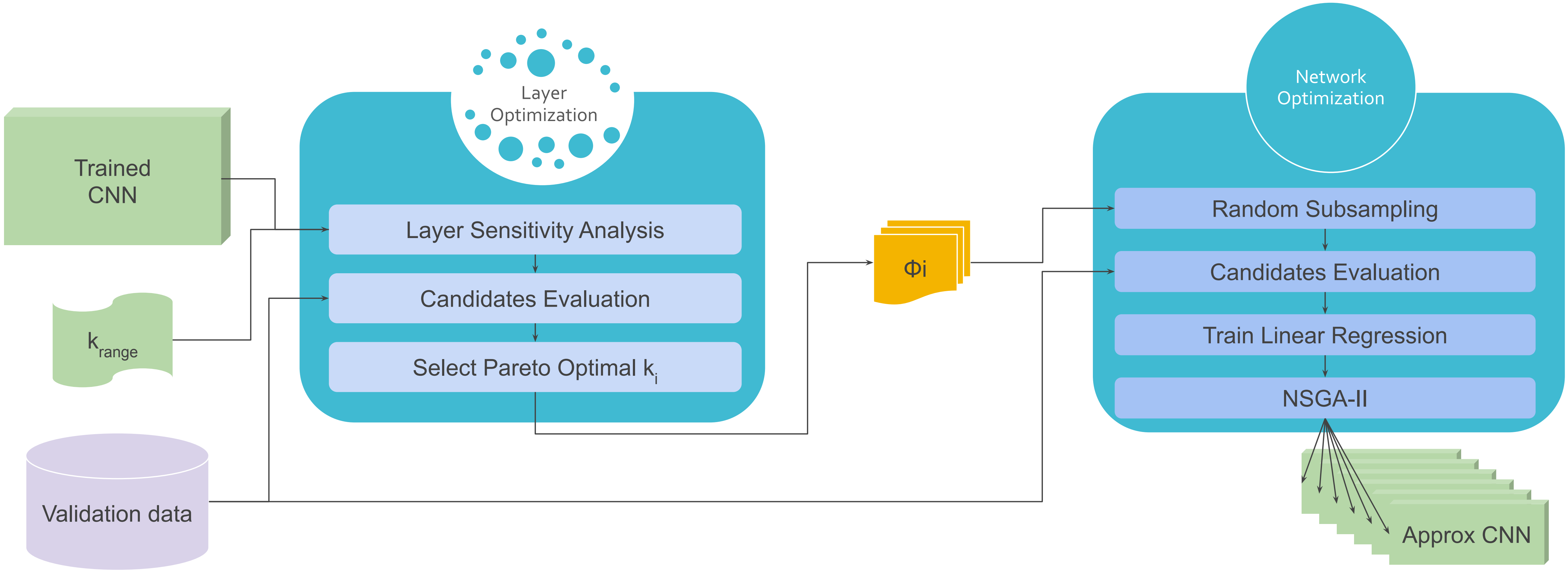
[1] IEEE Trans. Evol. Comput., 2002

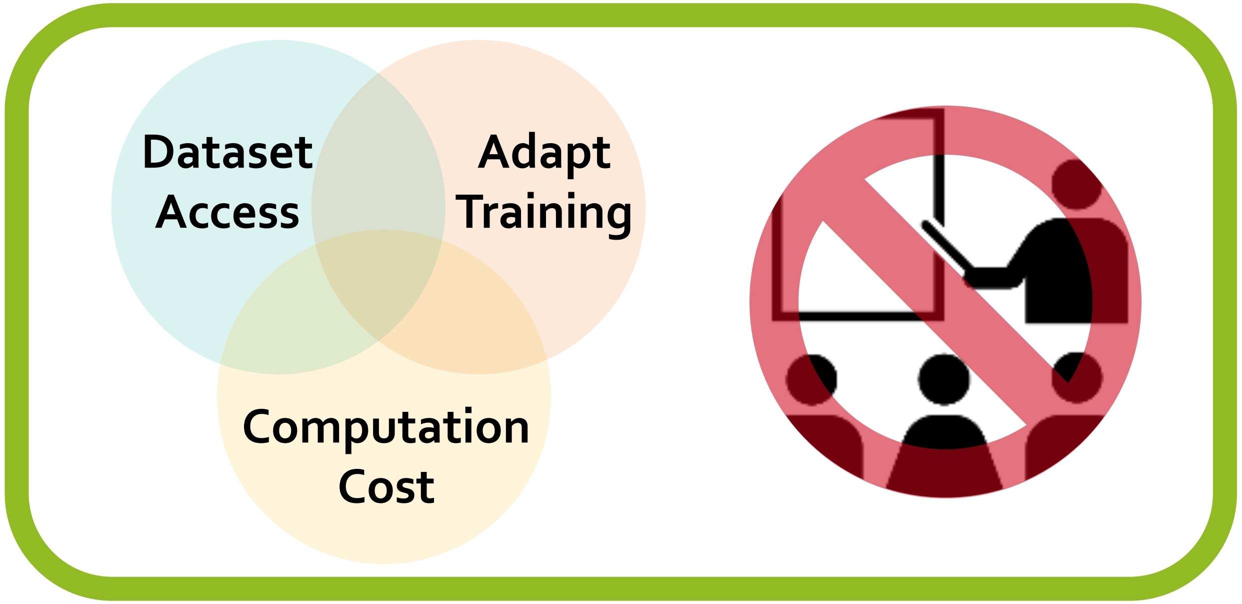
Pareto Improvement

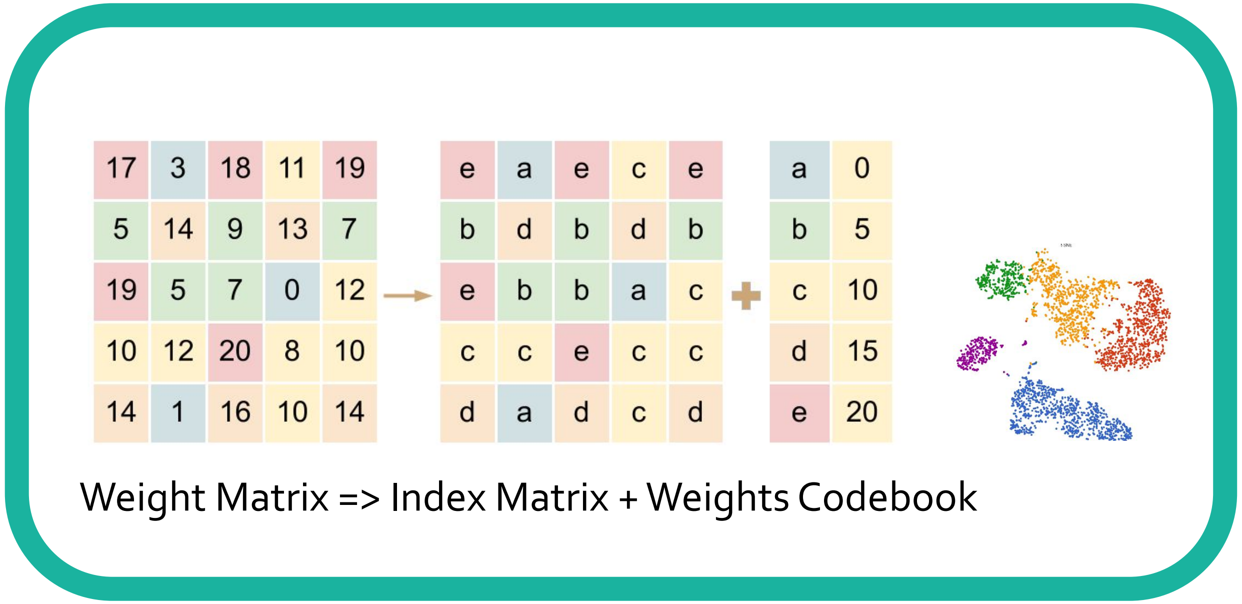


Resnet50V2, Linear regression model R^2 = 76%, NSGA-II iteration = 500

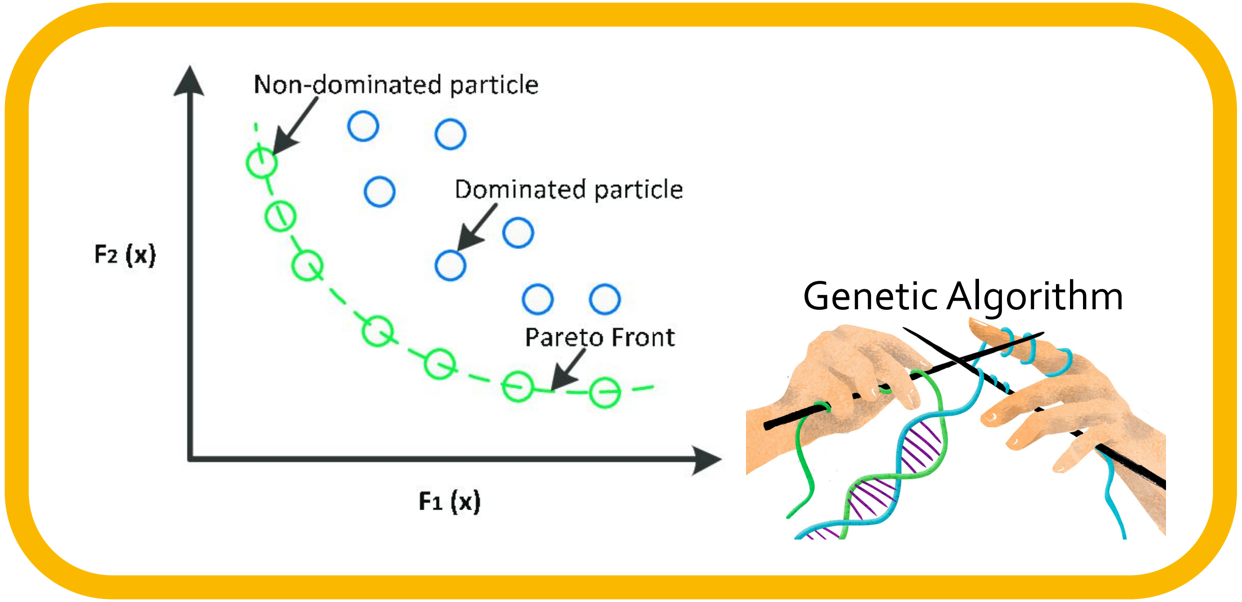
Conceptual View







A Heuristic Exploration of Retraining-free Weight-Sharing for CNN Compression

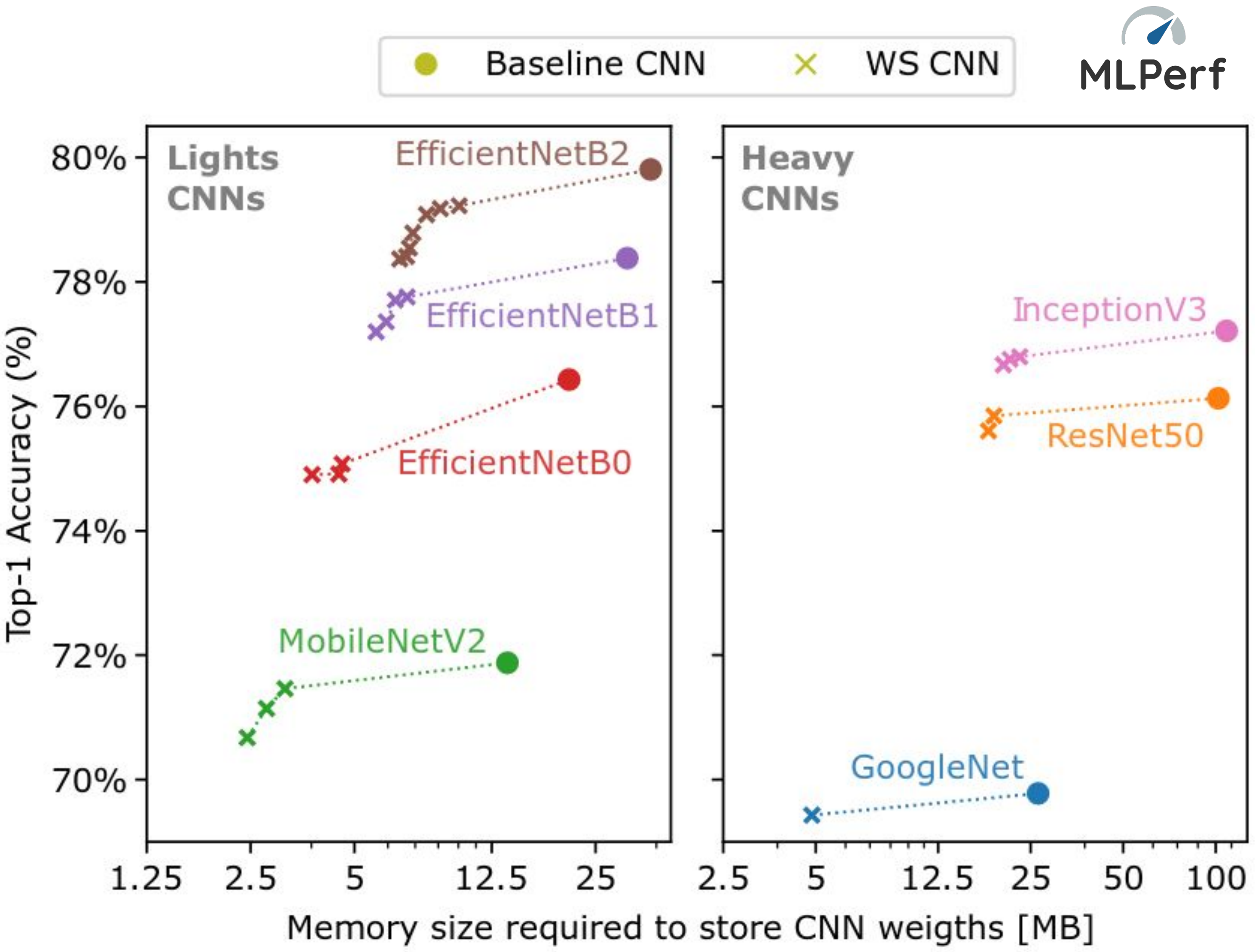


ImageNet results

Experimental Setup

- Imagenet Dataset (1m high-res images / 50k validation set)
- Tensorflow / Pytorch
- On-premise GPU Server (single NVIDIA Tesla V100 - 32GB)

Various Imagenet CNNs



MobileNetV2: IEEE/CVF, 2018
GoogleNet: CVPR, 2015
EfficientNet: Arxiv, 2019
InceptionV3: CVPR, 2016
ResnetV2: Arxiv, 2016

ImageNet Results

Network	MLPerf category	#Layer	Mem. [MB]	top-1 Acc. [%]	CR min, max
GoogleNet	heavy	58	50	69.7	5.4
ResNet50V2	heavy	54	97	76.0	5.3, 5.6
InceptionV3	heavy	2.8	104	77.2	4.7, 5.3
MobileNetV2	light	53	13	71.9	4.4, 5.7
EfficientNetB0	light	82	20	76.4	4.5, 5.6
EfficientNetB1	light	116	30	78.4	4.3, 5.3
EfficientNetB2	light	116	35	79.8	3.5, 5.3

- Up to 5x compression
- Both Heavy & light MLPERF categories
- 4h-16h exploration time (depends on #layer)

MLPerf

Comparison with others WS techniques (GoogleNet/Imagenet)

Method	Retraining-Free	CR	Top-1 AL (%)
Deep K-means(2018) [18]	Yes	1.5	1.22
	Yes	2	3.7
	Yes	3	13.72
	Yes	4	48.95
	No	1.5	0.26
	No	2	0.17
	No	3	0.36
	No	4	1.95
DP-Net (2020) [20]	No	7	-0.3
	No	10	1.56
FastWS [25]	Yes	4.55	0.83
This Work	**Yes**	$\boxed{5.44}$	$\boxed{0.35}$

- Both Deep K-means and DP-Net involves complex retraining

[18] Wu & al. ICML, 2018 [20] Yang & al. ASPDAC, 2021

Comparison with others WS techniques (GoogleNet/Imagenet)

Method	Retraining-Free	CR	Top-1 AL (%)
Deep K-means(2018) [18]	**Yes**	1.5	1.22
	Yes	2	3.7
	Yes	3	13.72
	Yes	4	48.95
	No	1.5	0.26
	No	2	0.17
	No	3	0.36
	No	4	1.95
DP-Net (2020) [20]	No	7	-0.3
	No	10	1.56
FastWS [25]	**Yes**	4.55	0.83
This Work	**Yes**	$\boxed{5.44}$	$\boxed{0.35}$

- Both Deep K-means and DP-NET involves complex retraining
- This work Pareto dominate the retraining free Deep K-means

[18] Wu & al. ICML, 2018 [20] Yang & al. ASPDAC, 2021

Comparison with others WS techniques (GoogleNet/Imagenet)

Method	Retraining-Free	CR	Top-1 AL (%)
Deep K-means(2018) [18]	Yes	1.5	1.22
	Yes	2	3.7
	Yes	3	13.72
	Yes	4	48.95
	No	1.5	0.26
	No	2	0.17
	No	3	0.36
	No	4	1.95
DP-Net (2020) [20]	No	7	-0.3
	No	10	1.56
FastWS [25]	Yes	4.55	0.83
This Work	**Yes**	5.44	0.35

- Both Deep K-means and DP-NET involves complex retraining
- This work Pareto dominates Deep K-means

[18] Wu & al. ICML, 2018 [20] Yang & al. ASPDAC, 2021

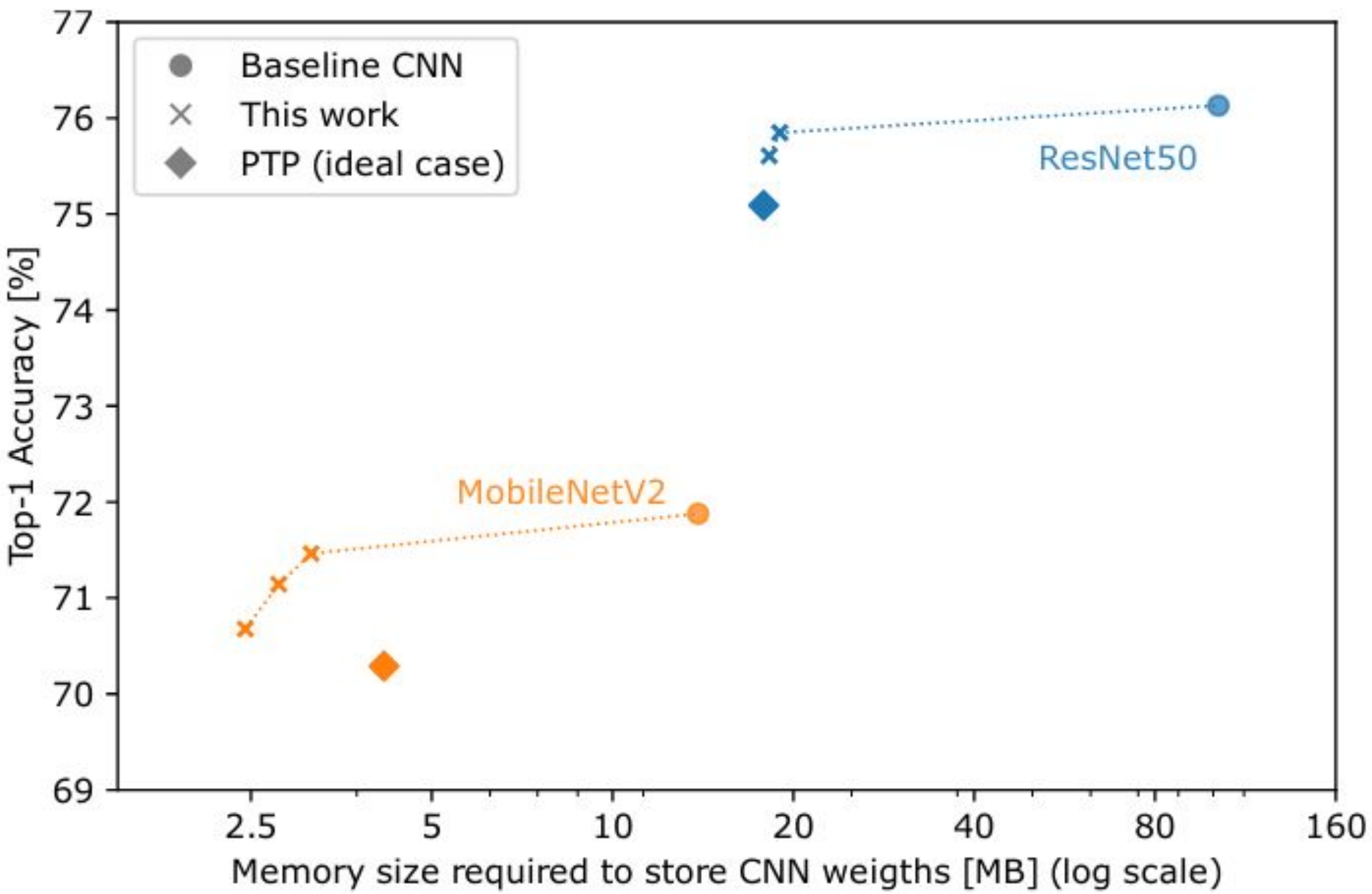
Comparison with others WS techniques
(GoogleNet/Imagenet)

Method	Retraining-Free	CR	Top-1 AL (%)
Deep K-means(2018) [18]	Yes	1.5	1.22
	Yes	2	3.7
	Yes	3	13.72
	Yes	4	48.95
	No	1.5	0.26
	No	2	0.17
	No	3	0.36
	No	4	1.95
DP-Net (2020) [20]	No	7	-0.3
	No	10	1.56
FastWS [25]	Yes	4.55	0.83
This Work	**Yes**	5.44	0.35

- Both Deep K-means and DP-NET involves complex retraining
- This work Pareto dominate Deep K-means
- Competitive results with DP-Net:
  - DP-Net best case estimated at **~30h** (30 epochs x >1h)
  - This work: **5h20 (~5.8x faster)**

[18] Wu & al. ICML, 2018 [20] Yang & al. ASPDAC, 2021

Comparison with Post-Training Pruning



- PTP: data-free pruning relying on fractal images
- Competitive if not Pareto dominating results

Lazarevich & al. ICCVW, 2021

Conclusion

Take Home

- Compression tuning allow for similar results without involving retraining
- The proposed compression method performs well on most CNNs
- Over 5x compression rate can be achieved without involving any retraining

e-dupuis/retraining-free-weight-sharing

Next Steps

- Investigate the introduction of the proposed Weight sharing optimization into **full compression pipeline**
- Analyse different **CNNs topologies resilience** to weight sharing
- Investigate the use of **calibration**
- Investigate a **channel-wise** weight sharing level

Authors & Funding

Etienne Dupuis
INL
etienne.dupuis@ec-lyon.fr

Ian O'Connor
INL

David Novo
LIRMM, CNRS

Alberto Bosio
INL

Institut des Nanotechnologies de Lyon UMR 5270

LIRMM

cnrs

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

ANR **AdequatedDL**
(ANR-18-CE23-0012)

Thank you for listening