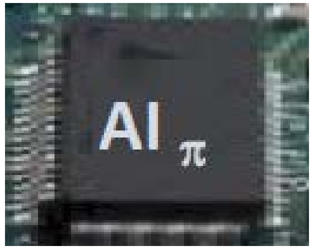
Enhancing External learning with Internal Training Paradigm

Methodical Design and Trimming of Deep Learning Networks: Enhancing External BP learning with Internal Omnipresent-Supervision Training Paradigm

S. Y. Kung Zejiang Hou Yuchen Liu

Princeton

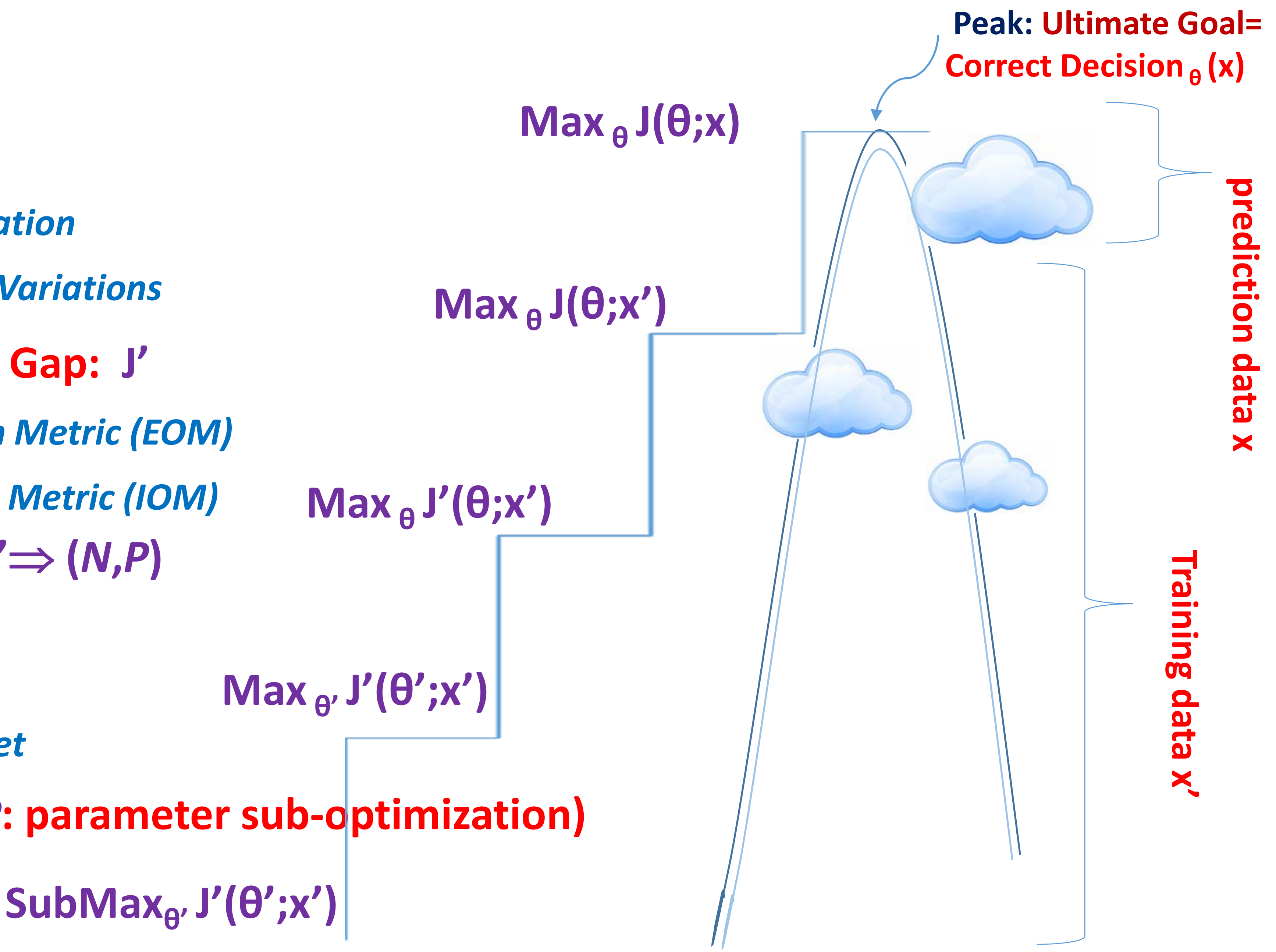
Automatic Bigdata Deep Compiler (ABDC)



Dedication: In fond memory of Jan Larsen (1965-2018).

Four (DONA) Gaps Differentiating Generalization from Optimization:

- **D: Data/Model Gap: x’**
    - *Data Augmentation*
    - *Image/Speech Variations*
- **O: Optimization Metric Gap: J’**
    - *External Optimization Metric (EOM)*
    - *Internal Optimization Metric (IOM)*
- **N: *Net* Capacity Gap: θ’⇒ (*N*,*P*)**
    - *Growing*
    - *Cherry Picking*
    - *Pruning Oversized Net*
- **A: : Algorithmic Gap (*P*: parameter sub-optimization)**
- *Regularization: Explicit and our Implicit methods*



- Current: Parameter Learning Only (**External BP**)
- Ours: Both Parameter/Structure Learning

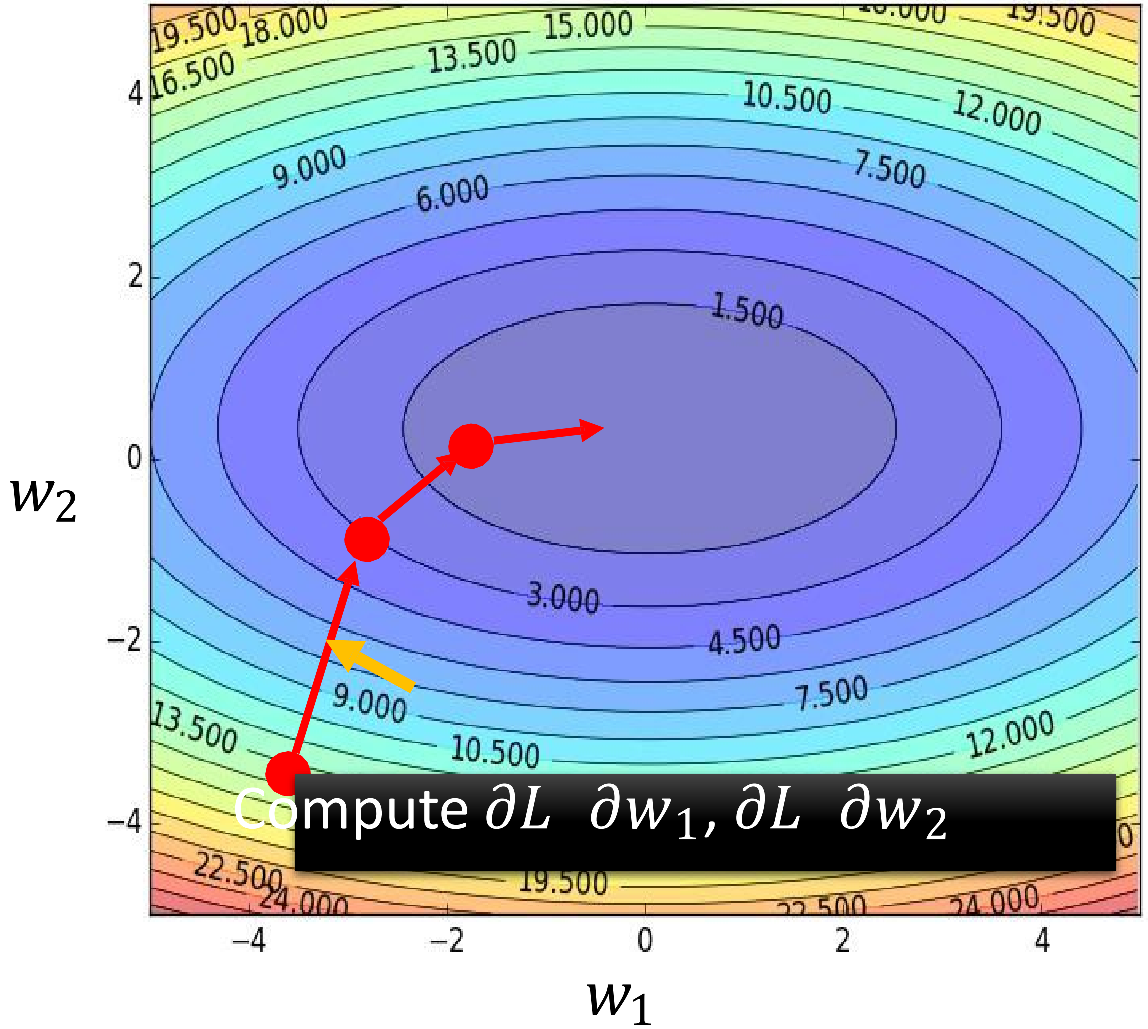
Parameter Learning:

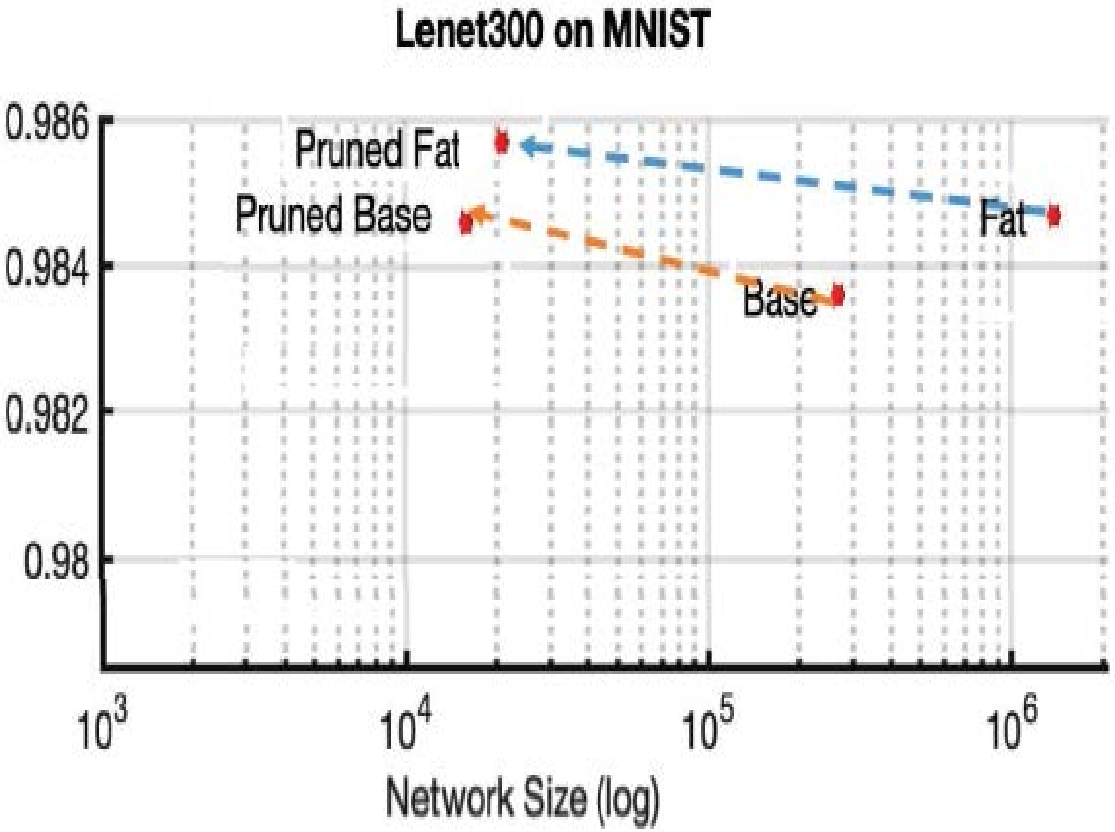
EOM Gradient Descent

\+

Structural Learning :

IOM Guided Adaptation



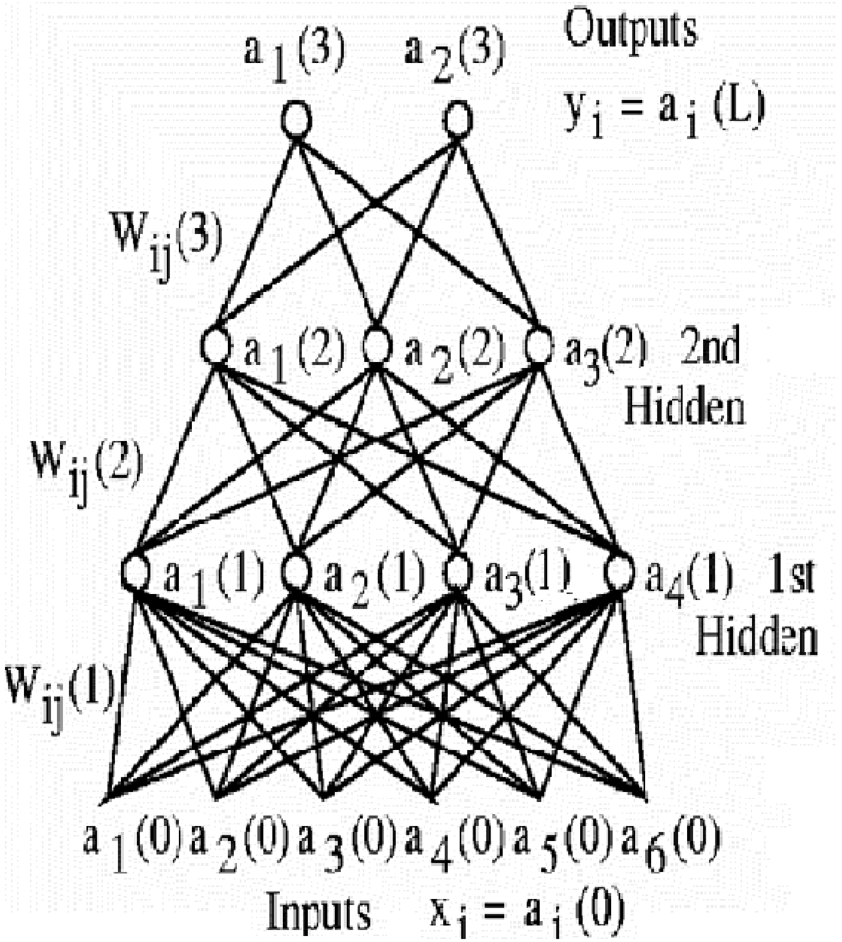


Structural characterization:

- **The number of layers in the model.**
- **The number of nodes in each layer.**

Internal Node Evaluation/Ranking (INER)

Nodes/Layers Treated Equally!

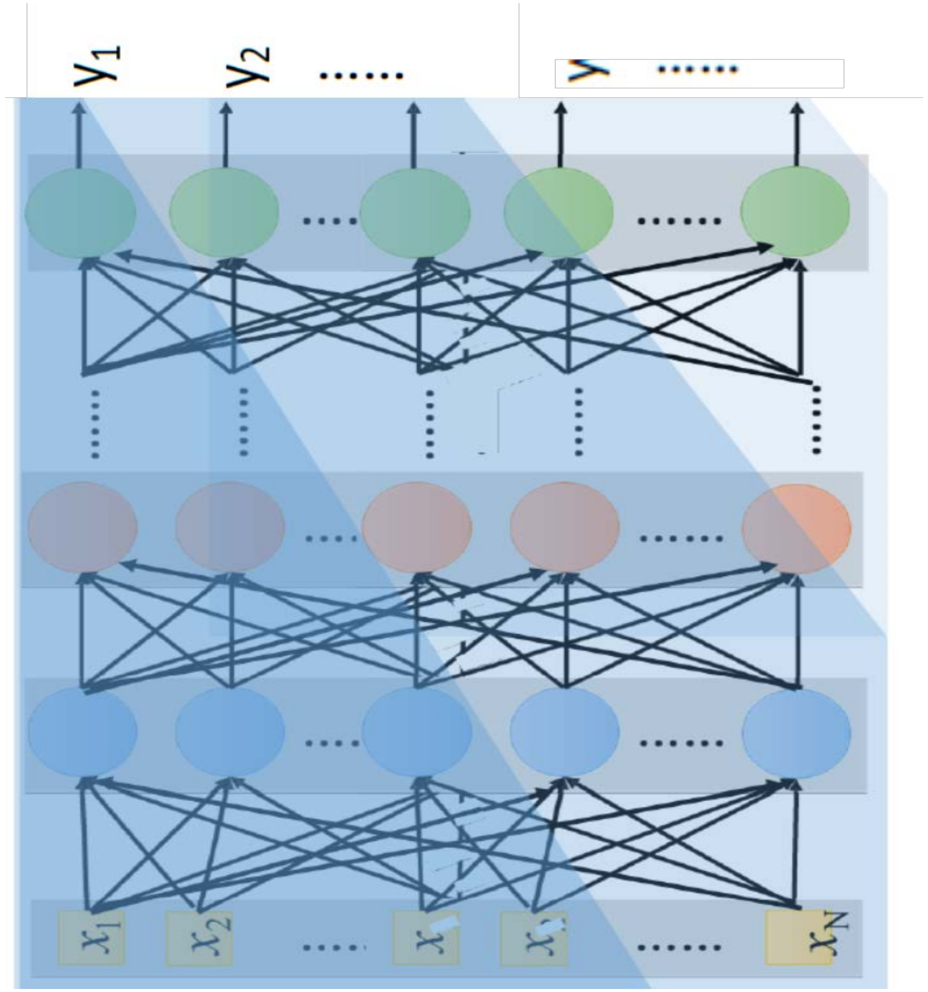


Ranking of Nodes:

Horizontal Structural Training

Ranking of Layers:

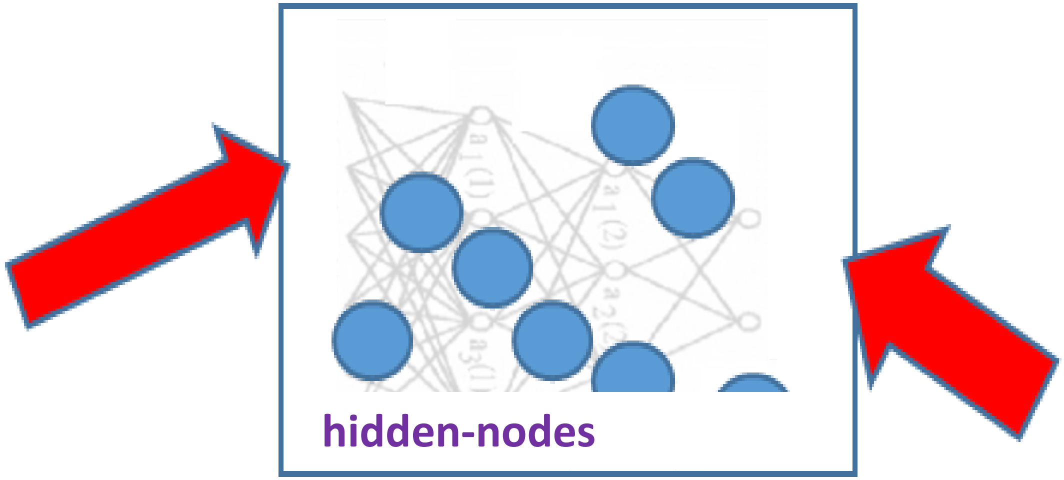
Vertical Structural Training



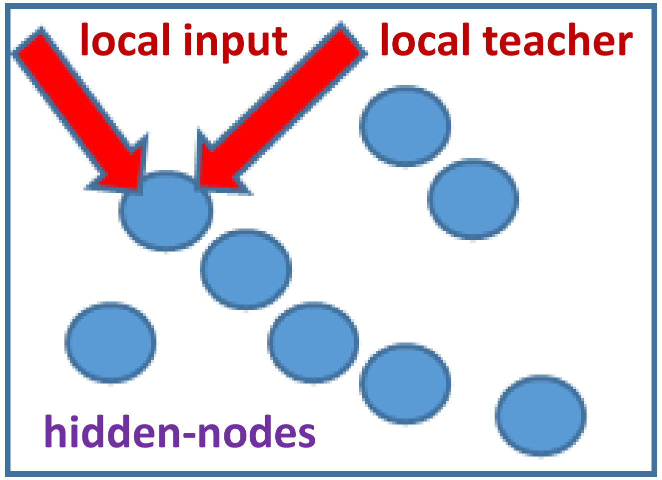
The internal learning paradigm allows us to evaluate/train hidden layers/nodes for directly.

(1) Internal Teacher Labels (ITL)

EXTERNAL



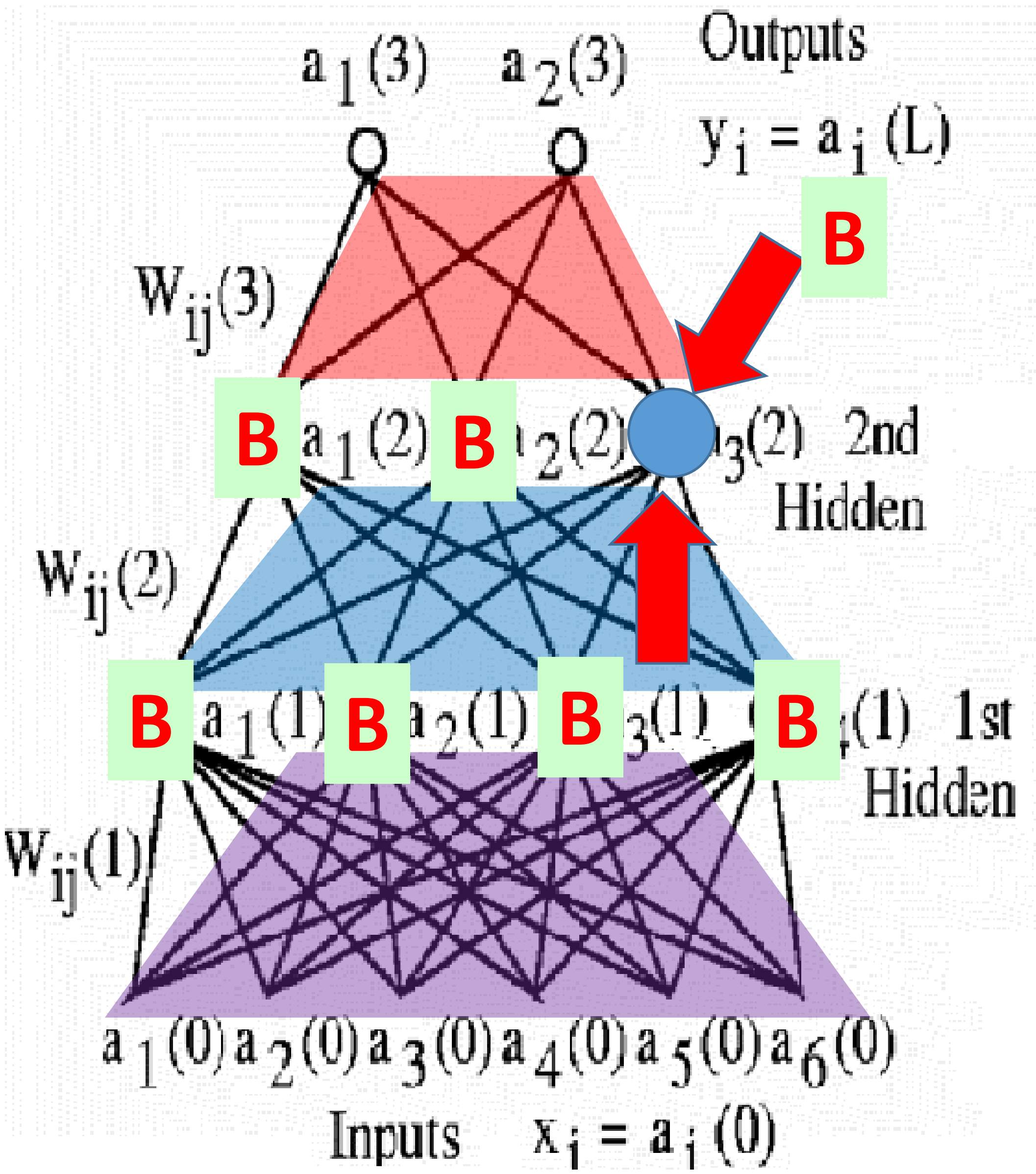
INTERNAL



(2) Internal Optimization Metrics (IOM)

Local Metrics for Internal Training must be of classification-type because one-hot encoding won’t work!!

ITL may be adaptive w.r.t. layer, hierarchical, or end-user.



$\{\ a_i(3),\ \text{B}\ \}$ i=1,…N

$\Uparrow$

$\{\ a_i(2),\ \text{B}\ \}$ i=1,…N

$\Uparrow$

$\{\ a_i(1),\ \text{B}\ \}$ i=1,…N

$\Uparrow$

$\{\ a_i(0),\ \text{B}\ \}$ i=1,…N

$\{\ x_i\ ,\ \text{B}\ \}$ i=1,…N

IOM must be of Classification-type:

DI (Discriminant Information)

$$\mathrm{DI} = \mathrm{DI}(\mathbf{I}) = \; \mathtt{tr}\left([\bar{\mathbf{S}} + \rho\mathbf{I}]^{-1}\mathbf{S}_B\right)$$

ρ: variance of additive noise

Three Scatter Matrices

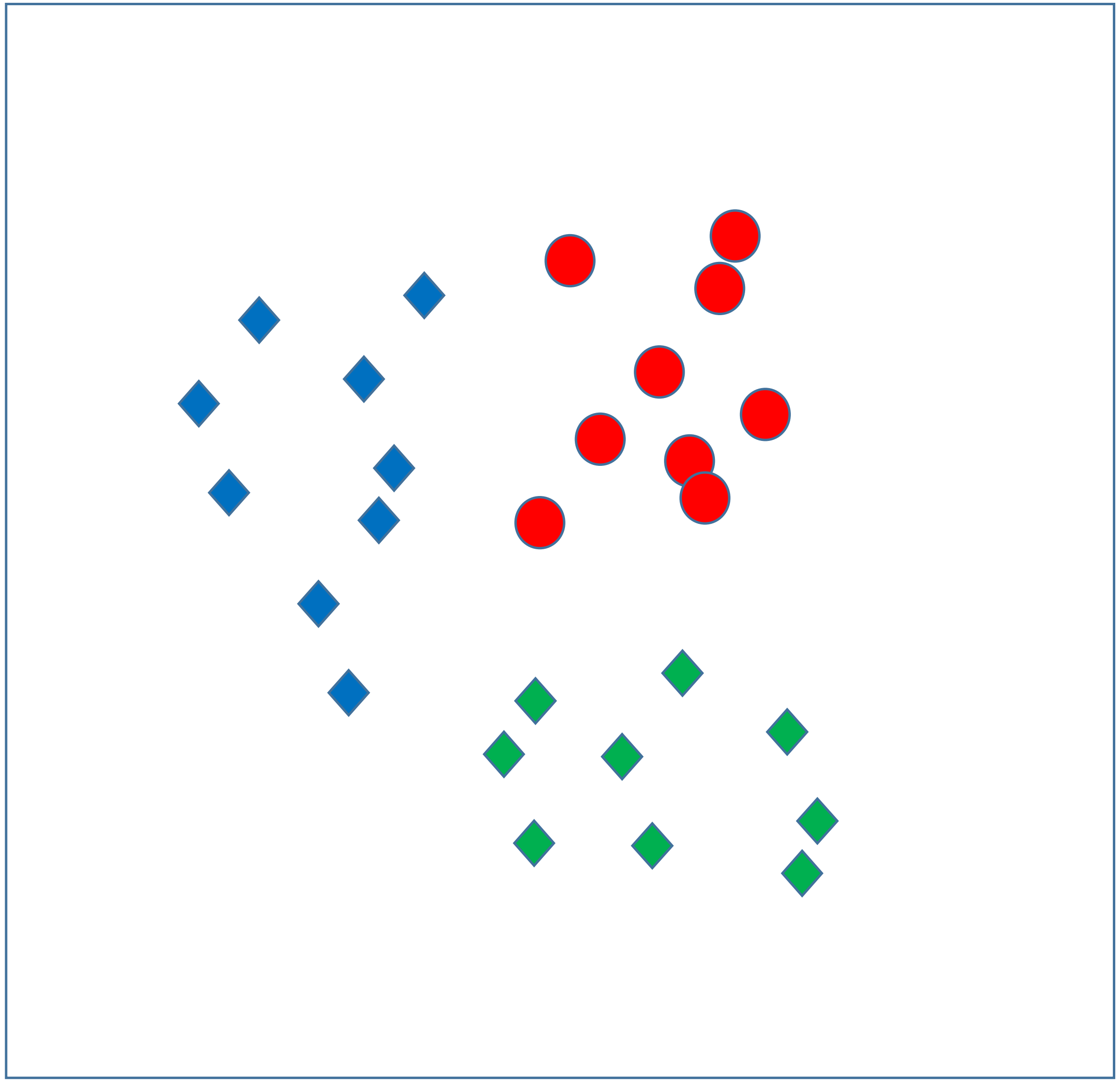
Scatter Matrix
$$\bar{\mathbf{S}} \equiv \bar{\mathbf{X}}\bar{\mathbf{X}}^T = \sum^{N}[\mathbf{x}_l - \vec{\mu}\,][\mathbf{x}_l - \vec{\mu}\,]^T$$

Between Class Scatter Matrix
$$\mathbf{S}_B = \sum_{\ell=1}^{L} N_\ell\,[\vec{\mu}_\ell - \vec{\mu}][\vec{\mu}_\ell - \vec{\mu}]^T = \Delta\Xi\Delta^T$$

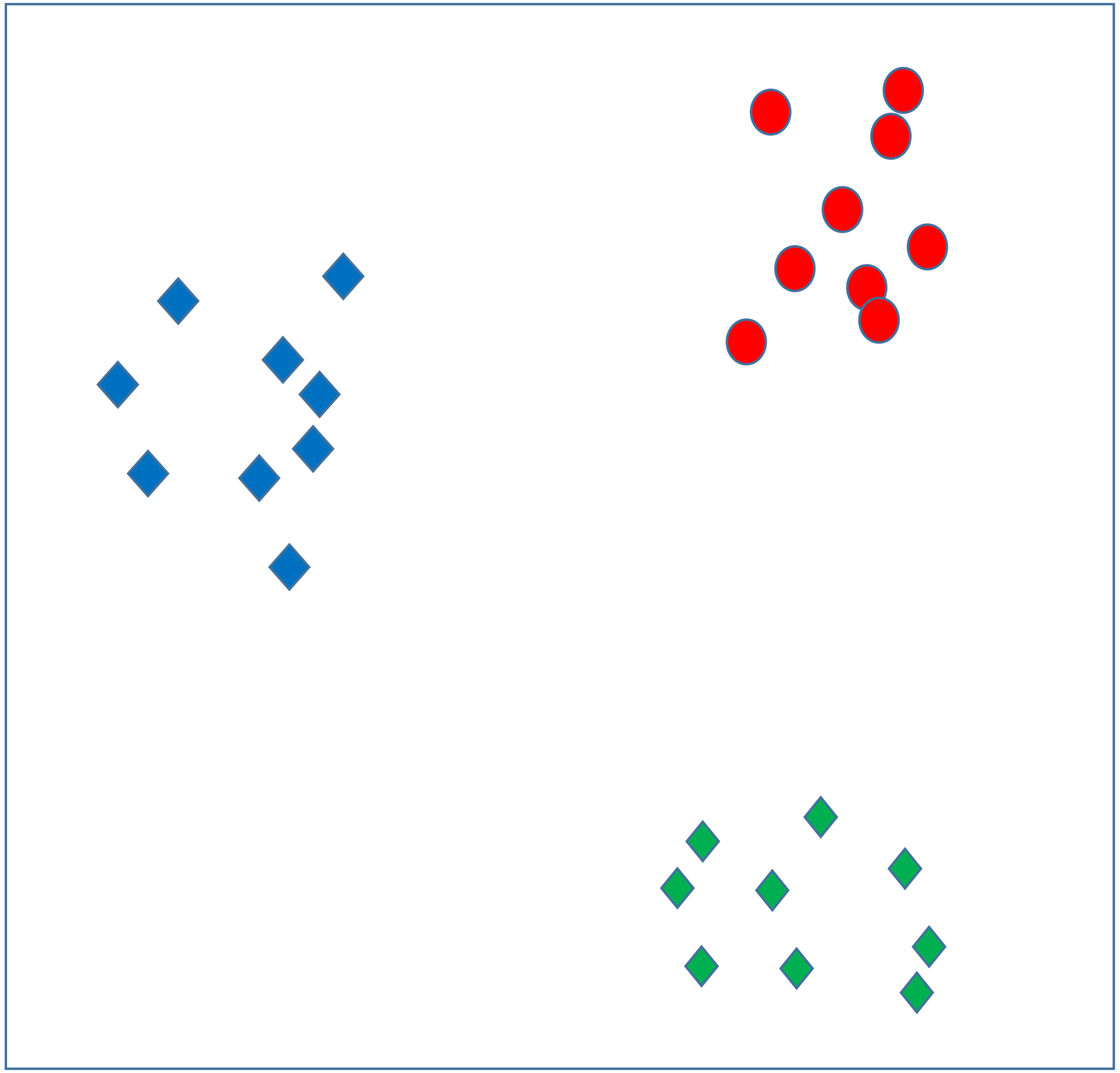
Within Class Scatter Matrix
$$\mathbf{S}_W = \sum_{\ell=1}^{L}\sum_{j=1}^{N_\ell}\left[\mathbf{x}_j^{(\ell)} - \vec{\mu}_\ell\right]\left[(\mathbf{x}_j^{(\ell)} - \vec{\mu}_\ell\right]^T$$

Suppose a hidden layer has 2 nodes (or more):

Low-DI Space (2 nodes)

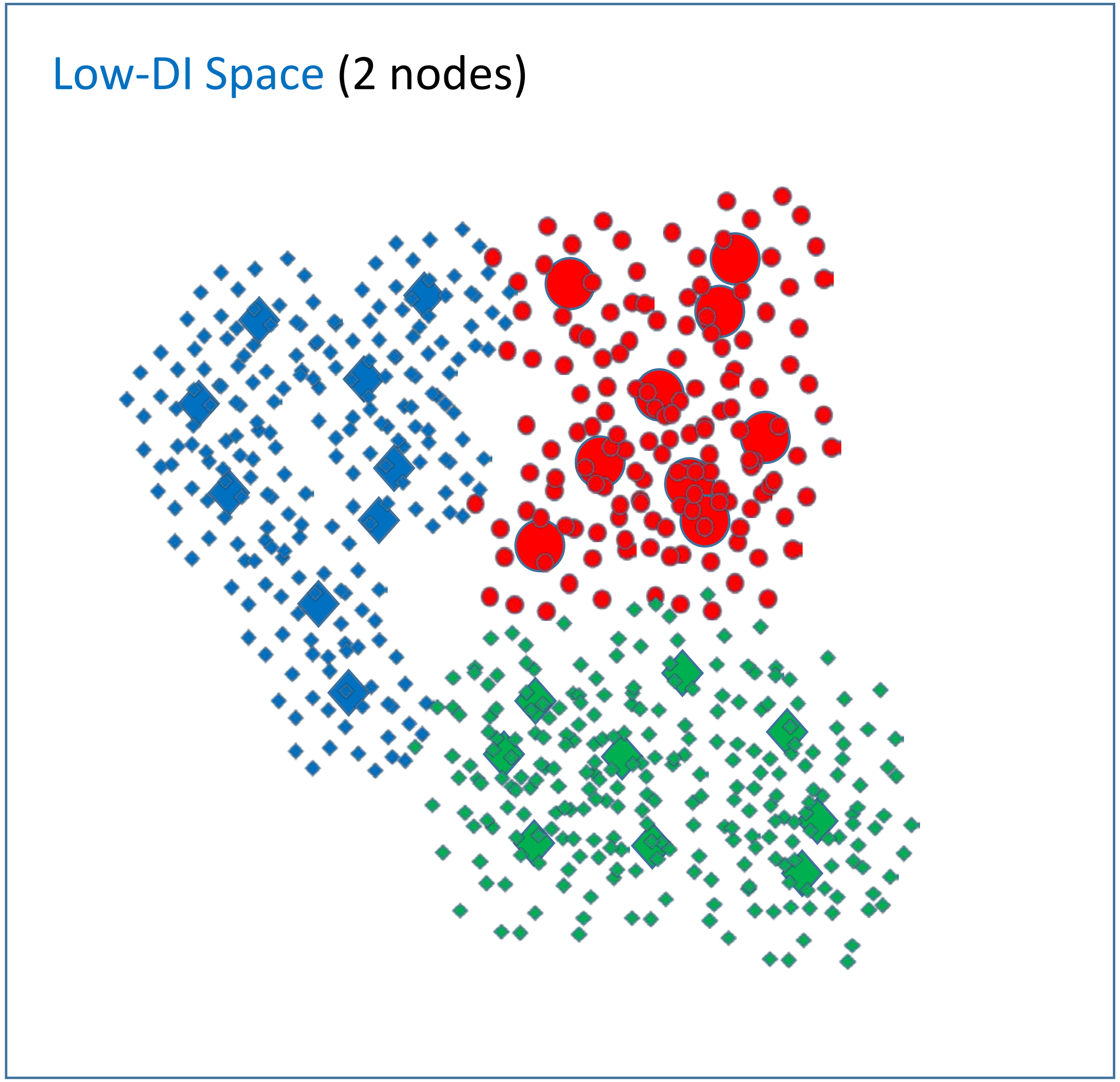


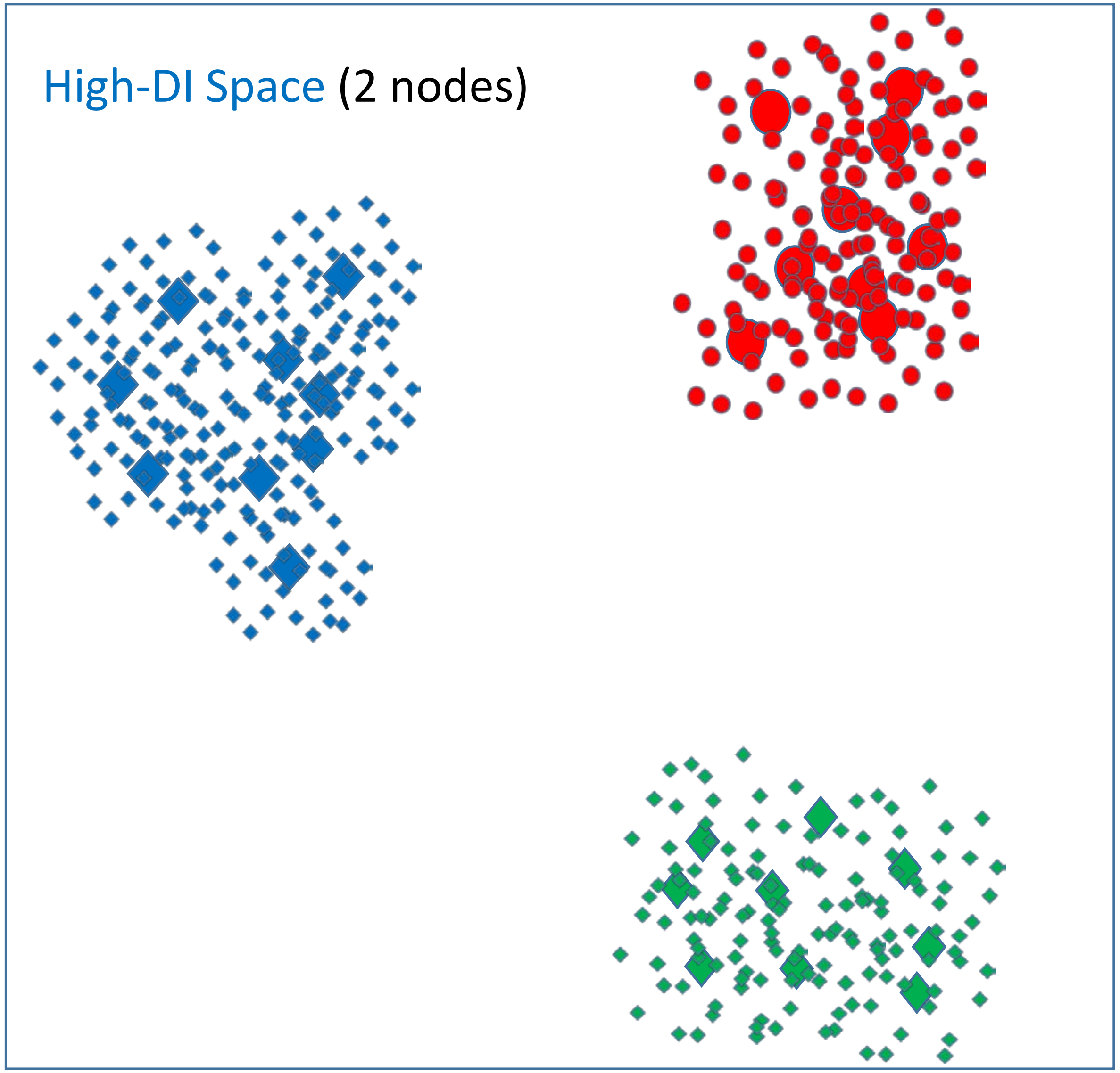
High-DI Space (2 nodes)



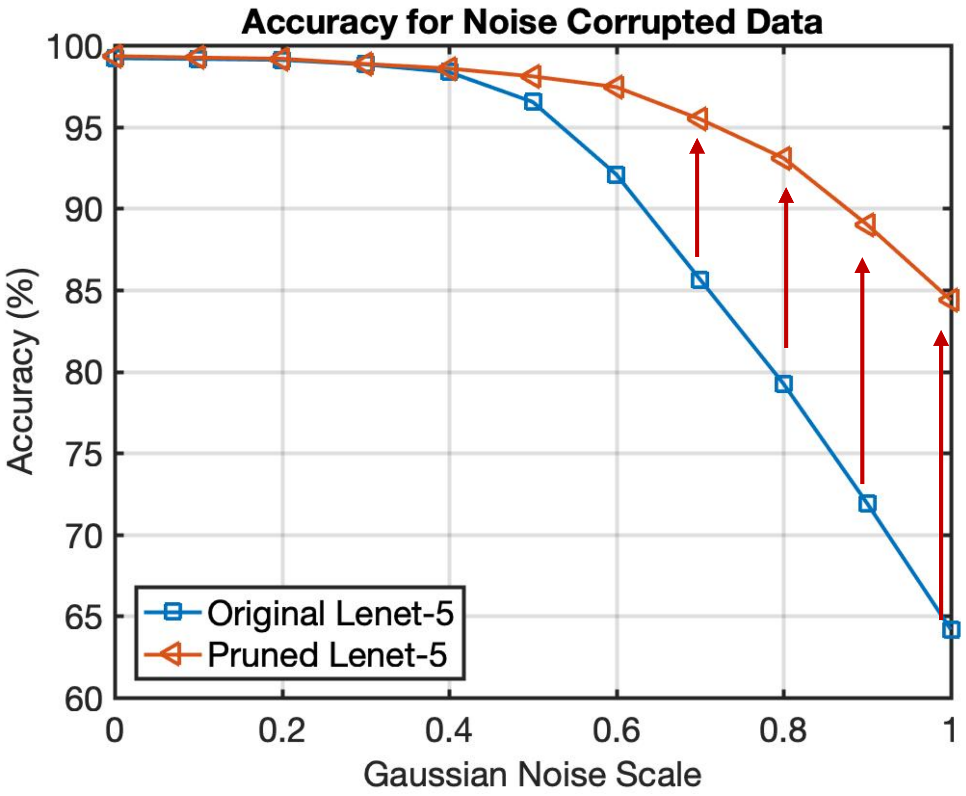
$$\mathrm{DI} = \mathrm{DI}(\mathbf{I}) = \mathtt{tr}\left([\bar{\mathbf{S}} + \rho\mathbf{I}]^{-1}\mathbf{S}_B\right)$$

ρ: variance of additive noise

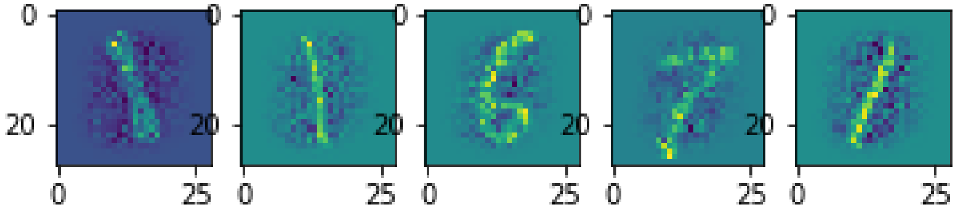




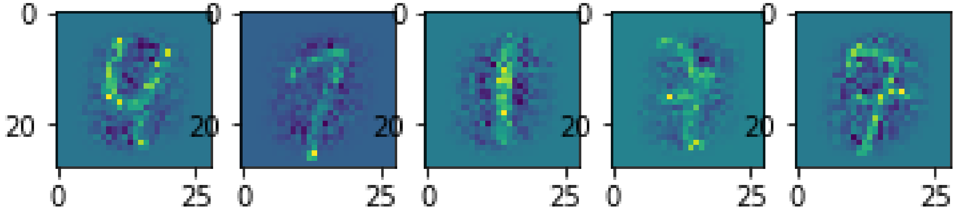
DI enhances robustness to noise: Lenet on MNIST



Noise scale: 0.8



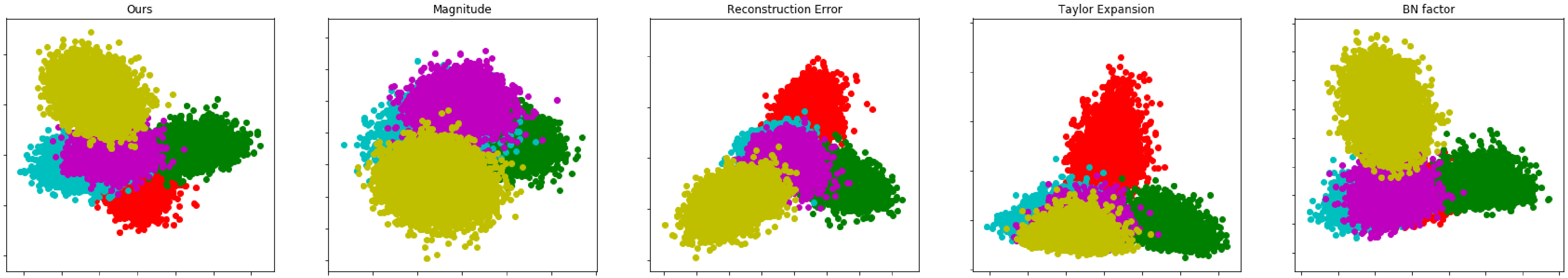
Original	8	8	8	7	8
Pruned	8	1	6	7	1



Original	4	3	7	8	3
Pruned	4	7	1	7	7

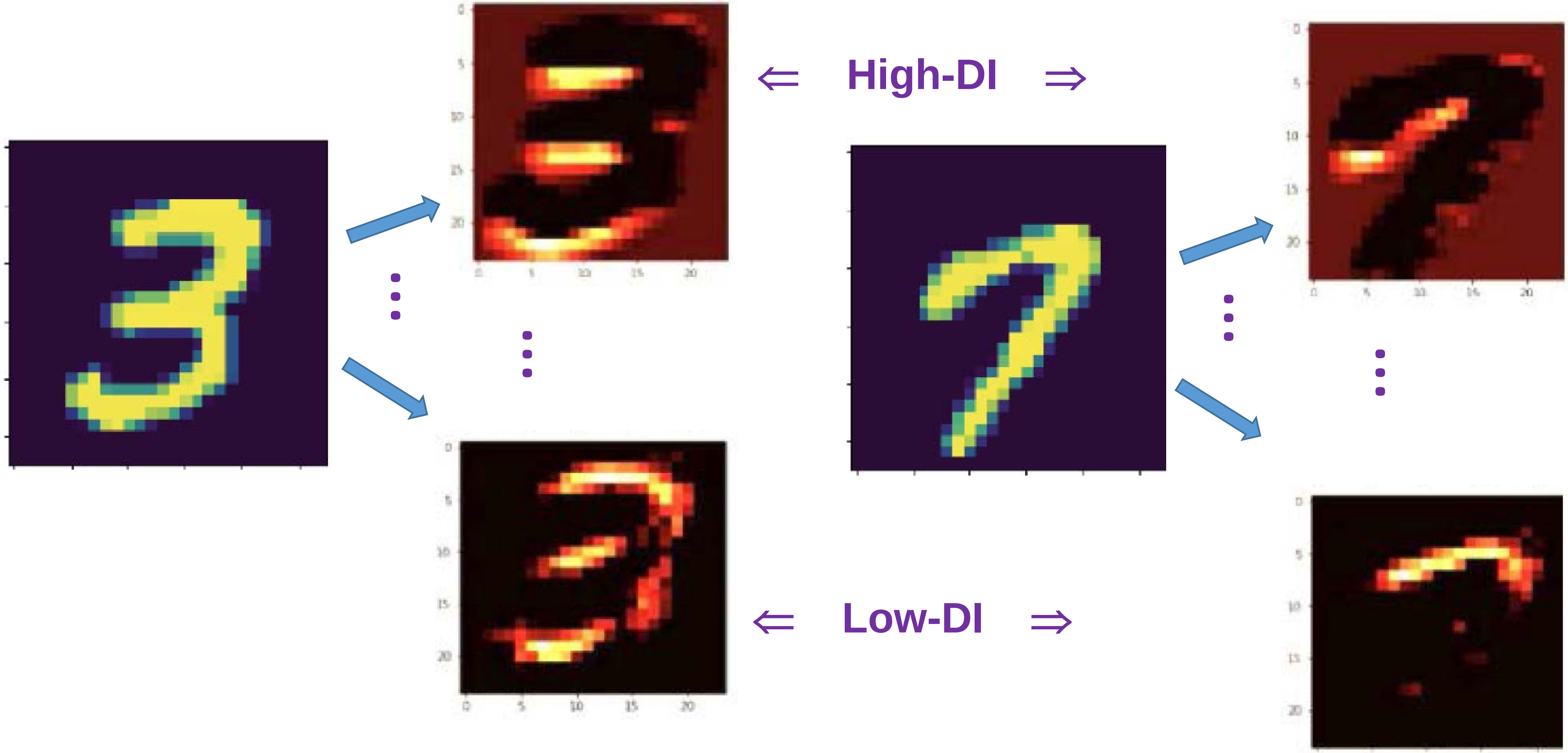
Effectiveness of DI criterion: ResNet on CIFAR

- Compressed ResNet-56 by different criterion from the same pre-trained network
- 2D projected feature maps of the last residual layer
- CIFAR-10 samples from 5 classes for visualization
- Our method maximally preserves the feature separability, thus testing accuracy



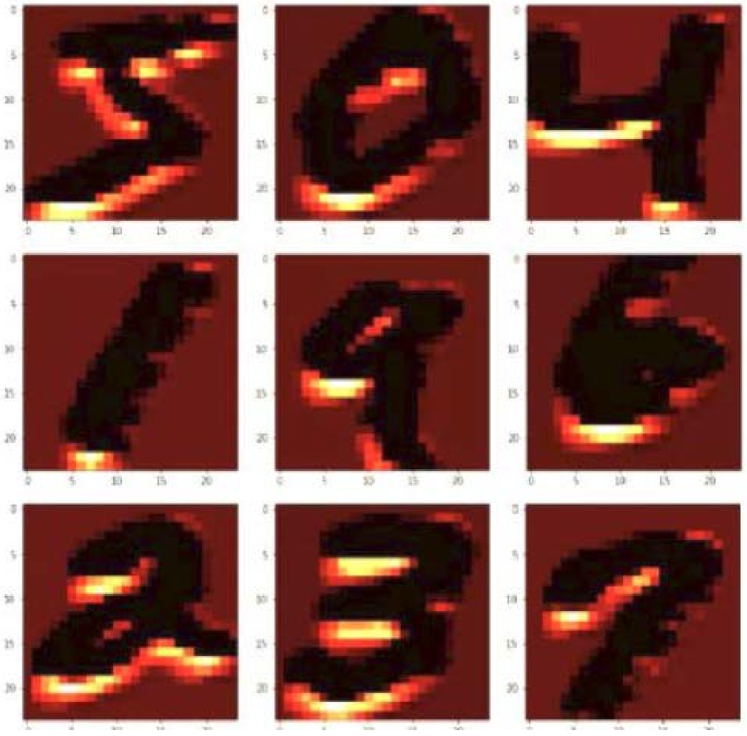
Criteria	Ours	Magnitude	Reconstruction error	Taylor Expansion	BN factor
Accuracy ↓	**1.75%**	2.17%	6.57%	2.69%	4.66%

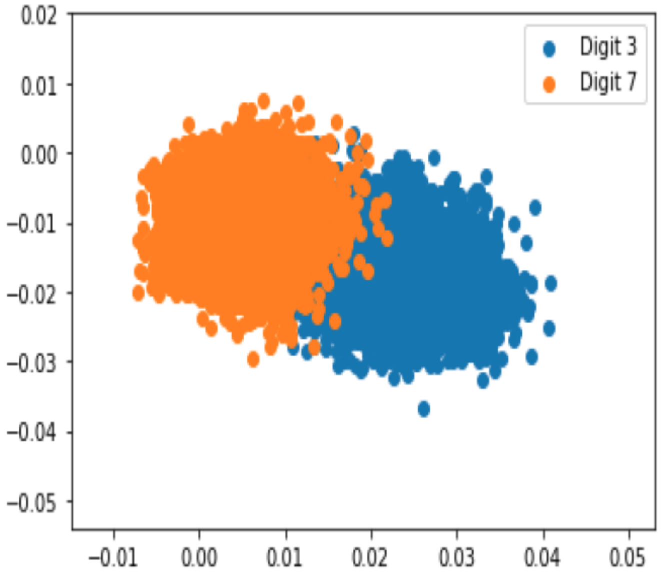
Visualization via Chanel Images [KHL19]



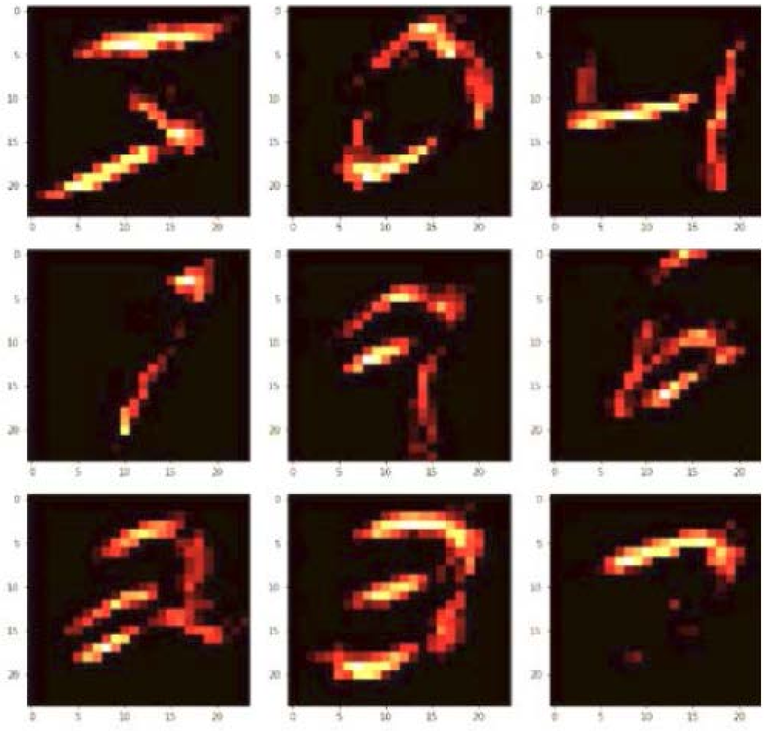
DI metric is used to determine which channels to prune

DI metric can determine which channels to prune/select

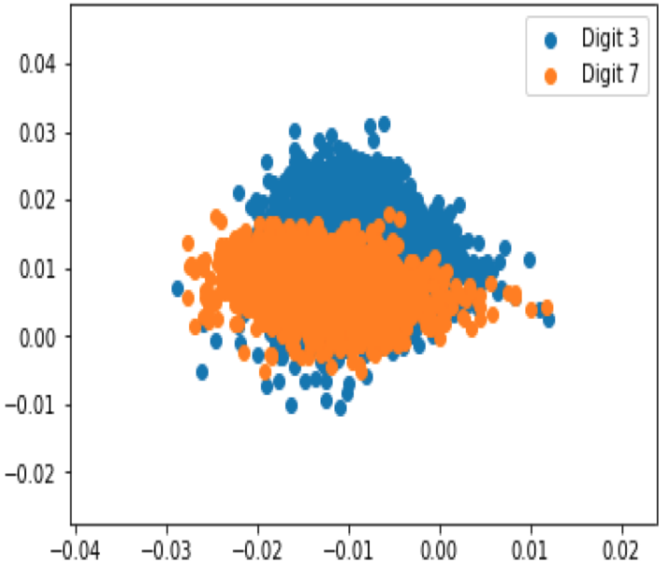




High-DI Channel



Low-DI Channel



hide

For node evaluation:

Subspace DI

$$DI = tr((\mathbf{W}^T\bar{S}\mathbf{W} + \rho I)^{-1}\mathbf{W}^T S_B \mathbf{W})$$

To assess the IOM of the full space of a layer, we set **W= I**:

For (supervised) deep compression, we adopt

$W_{i\text{-keep}}$ / $W_{i\text{-drop}}$ to keep/drop only the *i-th* node/channel:

$$\mathbf{W}_{i_{keep}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \cdots & \cdots & 0 \\ \vdots & \cdots & 1 & \cdots & \vdots \\ 0 & \cdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \qquad \mathbf{W}_{i_{drop}} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \cdots & \cdots & 0 \\ \vdots & \cdots & 0 & \cdots & \vdots \\ 0 & \cdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

For pruning nodes/channels in MLP/ConvNet, we adopt:

- Fisher Discriminant Ratio (FDR):

$$\mathrm{FDR} = DI(\mathbf{W}_{i_{keep}})$$

$$\mathbf{W}_{i_{keep}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \cdots & \cdots & 0 \\ \vdots & \cdots & 1 & \cdots & \vdots \\ 0 & \cdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

is the value of the i-th node/channel.

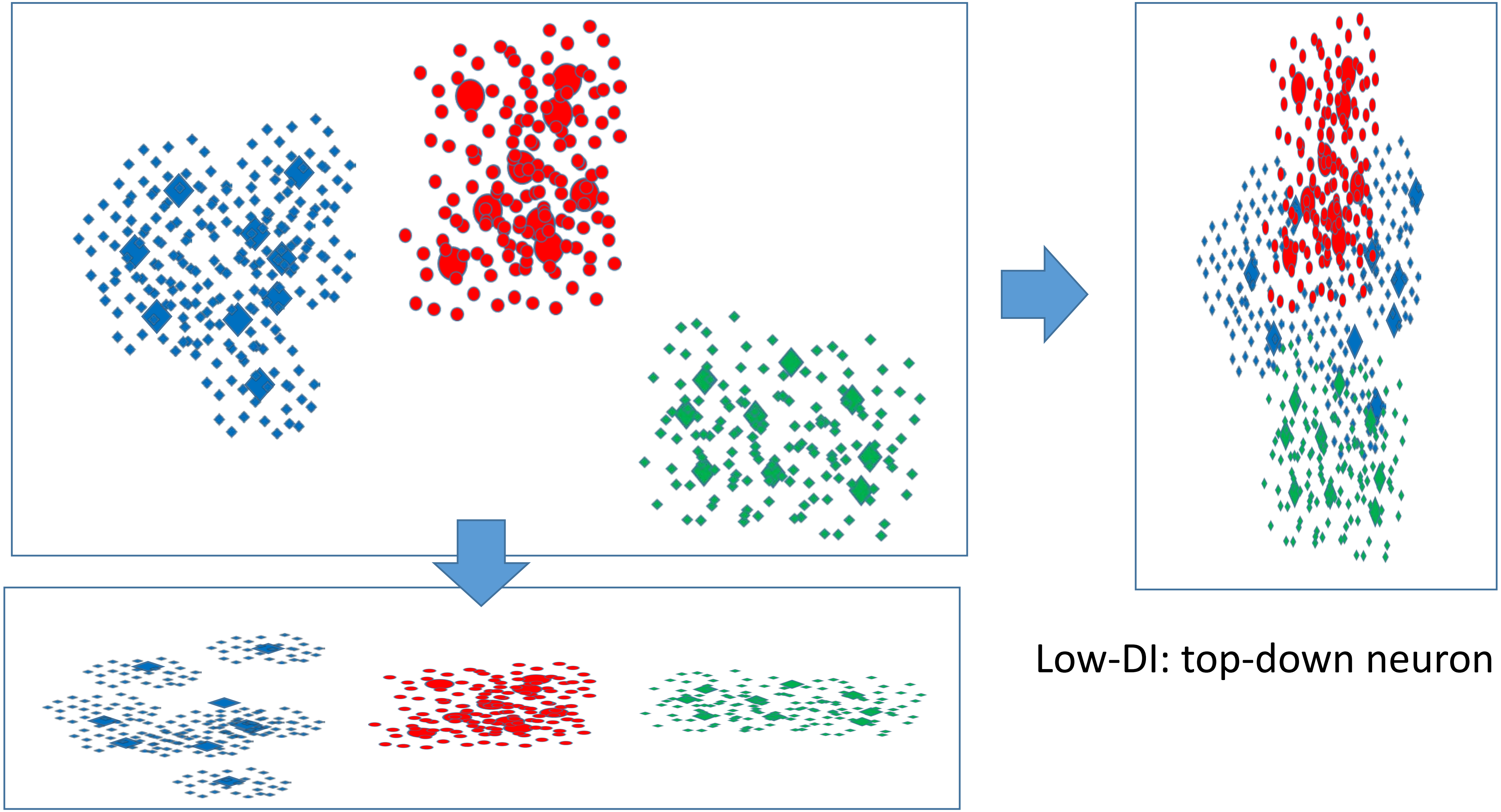
- Dispensability of a node/channel: DI-Loss:

$$\mathrm{DILoss} \equiv \mathrm{DI}(\mathbf{I}) - \mathrm{DI}(\mathbf{W}_{i_{drop}})$$

$$\mathbf{W}_{i_{drop}} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \cdots & \cdots & 0 \\ \vdots & \cdots & 0 & \cdots & \vdots \\ 0 & \cdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

is the remaining value of the layer after removing the i-th node/channel.
This reflects the dispensability of the i-th node/channel.

Differential-DI Subspace (2 nodes: left-vs-right, top-down-neuron)



Low-DI: top-down neuron

High-DI: left-vs-right neuron

The internal teachers facilitate two structural training strategies:

(1) DI-based Cherry Picking Method

(2) DI-based Pruning Method

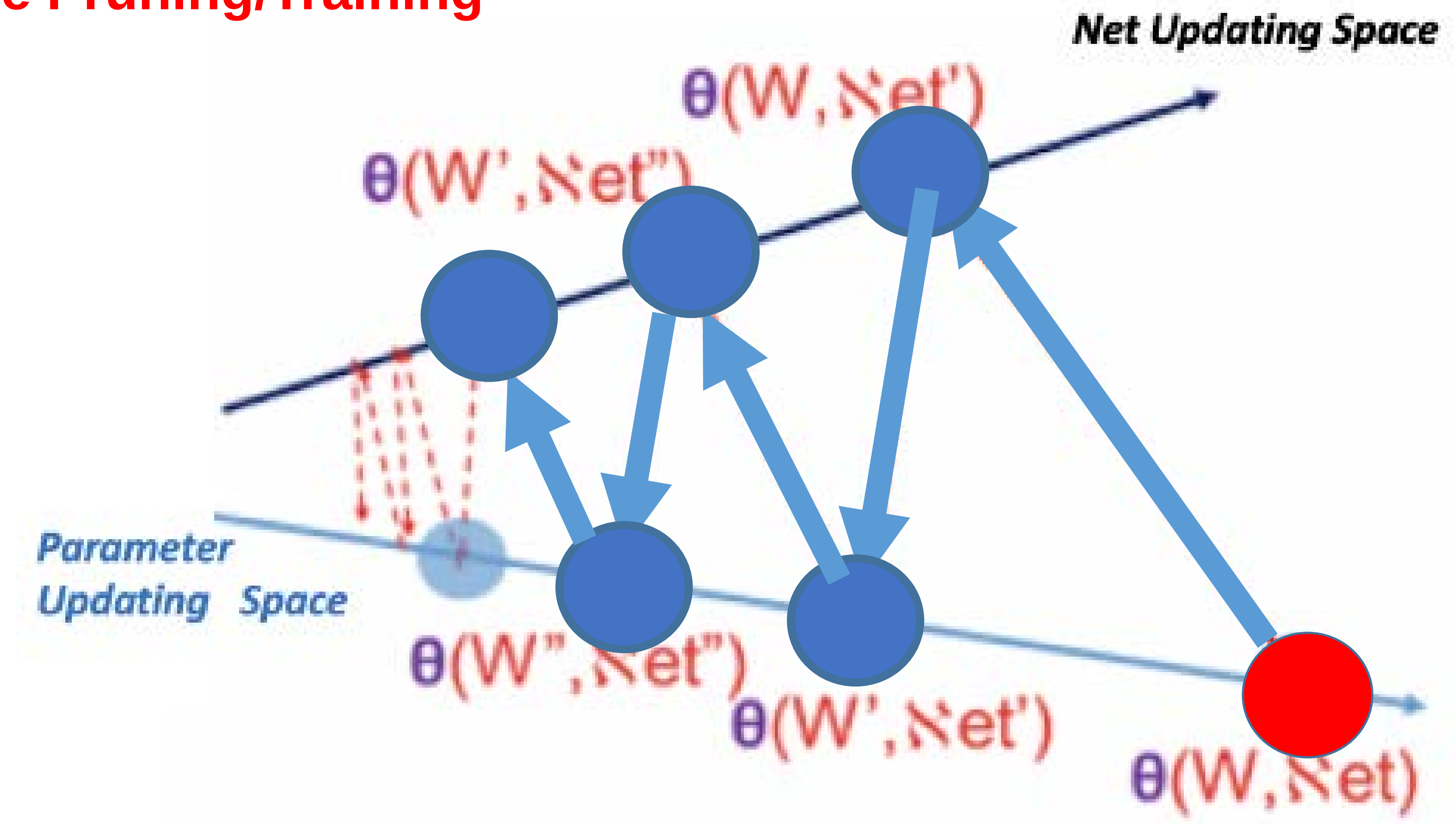
DI-based Cherry Picking Offers Rapid Deep Compression/Quantization

LeNet-300 on MNIST	Accuracy	Storage Size	Compression	Quantized
Original	98.36%	840 KB	-	16-bit
DI pruning	98.42%	71.4 KB	12x	16-bit
Cherry-Picking	94.42%	8.7 KB	97x	16-bit
Quantization*	94.22%	5.8 KB	145x	64.8% 8-bit

VGG19 on CIFAR-100	Accuracy	Storage Size	Compression	Quantized
Original	73.26%	60 MB	-	16-bit
DI pruning	73.67%	9.8 MB	6x	16-bit
Cherry-Picking	71.01%	2.4 MB	25x	16-bit
Quantization*	70.36%	1.8 MB	33x	60.9% 10-bit

DI-based Pruning Mehtod Offers Win-Win Deep Compression/Quantization

NP-Iterative Pruning/Training



1. Biological Justification of Internal Learning

2. Pruning Efficacy

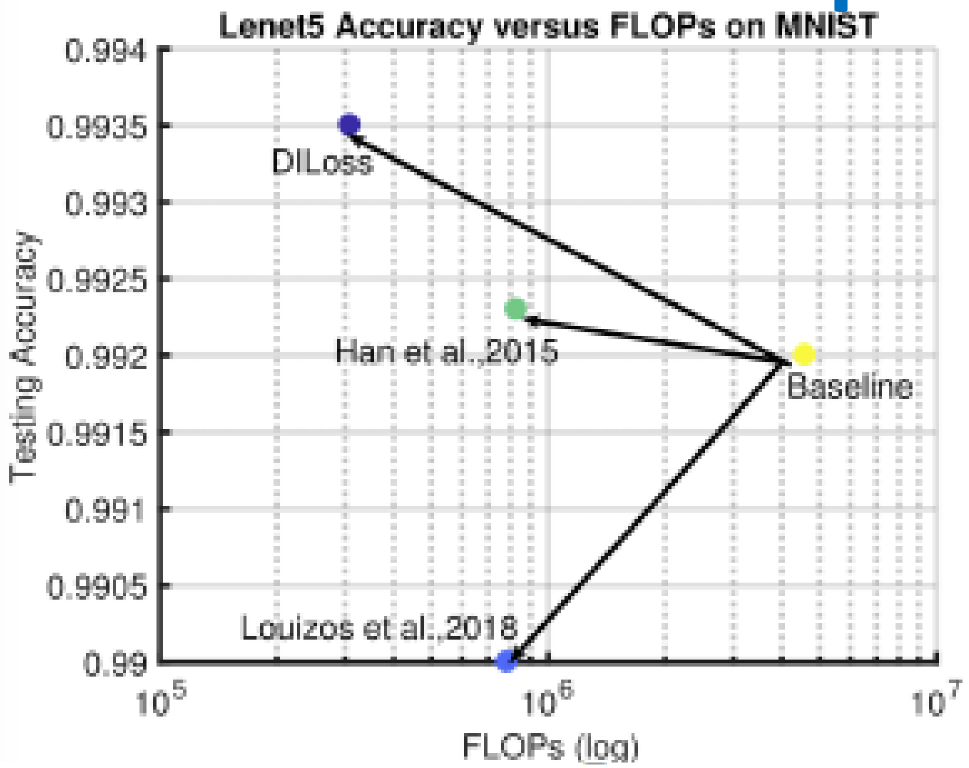
3. Performance Improvements: Experimental Results

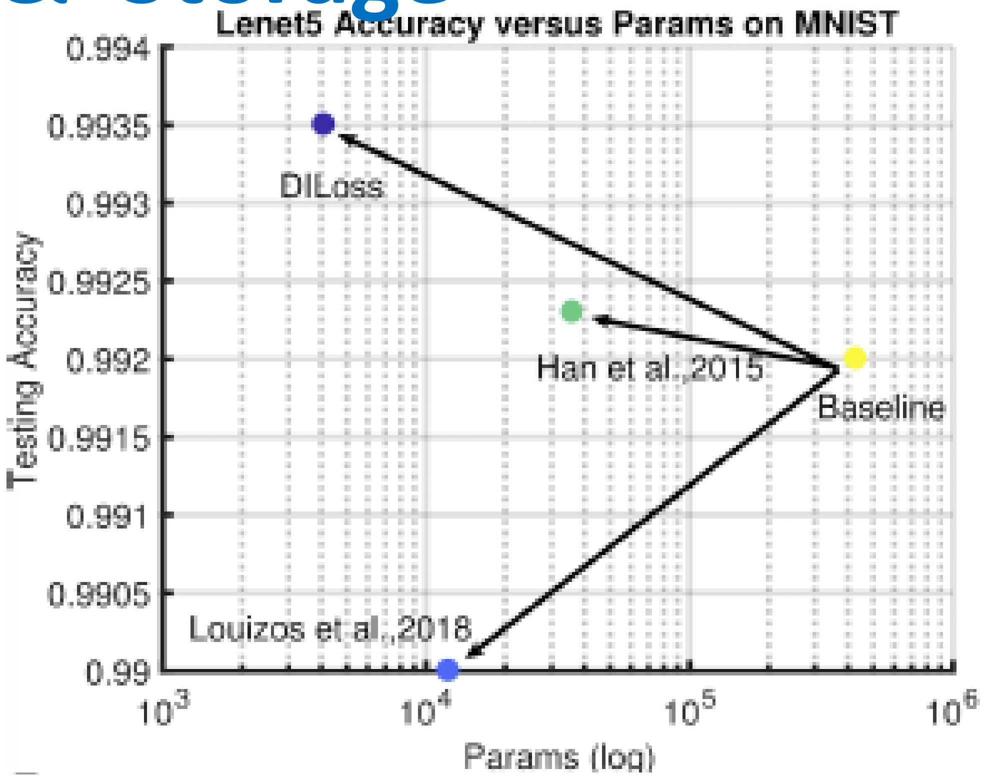
BPOS: Bridging GAP between regression-type and classification-type metrics

Definition: by balanced dataset we mean that the training samples for all the class labels are of the same size.

For balanced case, it can be shown that all of them are mathematically **equivalent optimization metrics** if one-hot encoding is used for LSE teacher values. (Next Page)

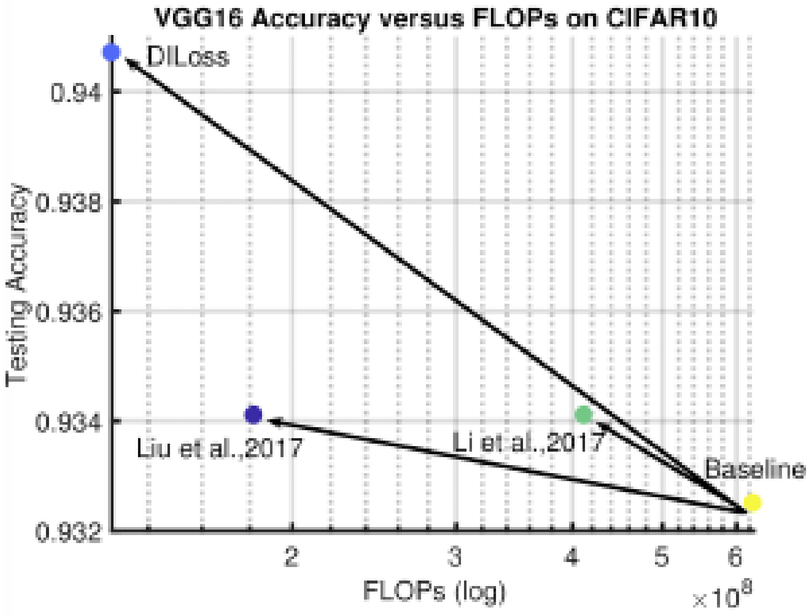
MINIST: Speedup & Storage



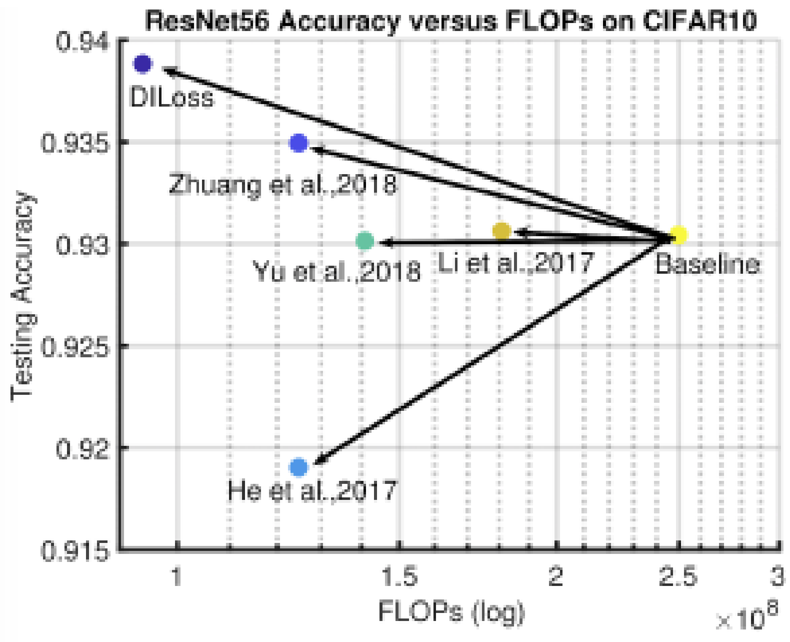


Task	Models	Accuracy %	FLOPs	Params
MNIST	*Lenet-5 (baseline)*	99.2	4.59×10^6	4.3×10^5
	Lenet-5 (Han et al., 2015)	99.23	8.3×10^5(18.1%)	3.6×10^4(8.4%)
	Lenet-5 (Louzois et al., 2018)	99	7.85×10^5(17.1%)	1.22×10^4(2.83%)
	Lenet-5 (FDR)	99.33	2.6×10^5(5.74%)	4.9×10^3(1.1%)
	Lenet-5 (DILoss)	**99.35**	**2.46×10^5(5.36%)**	**3.86×10^3(0.89%)**
	Lenet-300 (baseline)	98.36	-	2.7×10^5
	Lenet-300 (Han et al., 2015)	98.41	-	2.24×10^4(8.3%)
	Lenet-300 (Louzois et al., 2018)	98.2	-	2.7×10^4(10%)
	Lenet-300-100 (FDR)	98.42	-	2.3×10^4(8.5%)
	Lenet-300 (DILoss)	**98.46**	-	**1.63×10^4(6.04%)**

CIFAR-10: Speedup



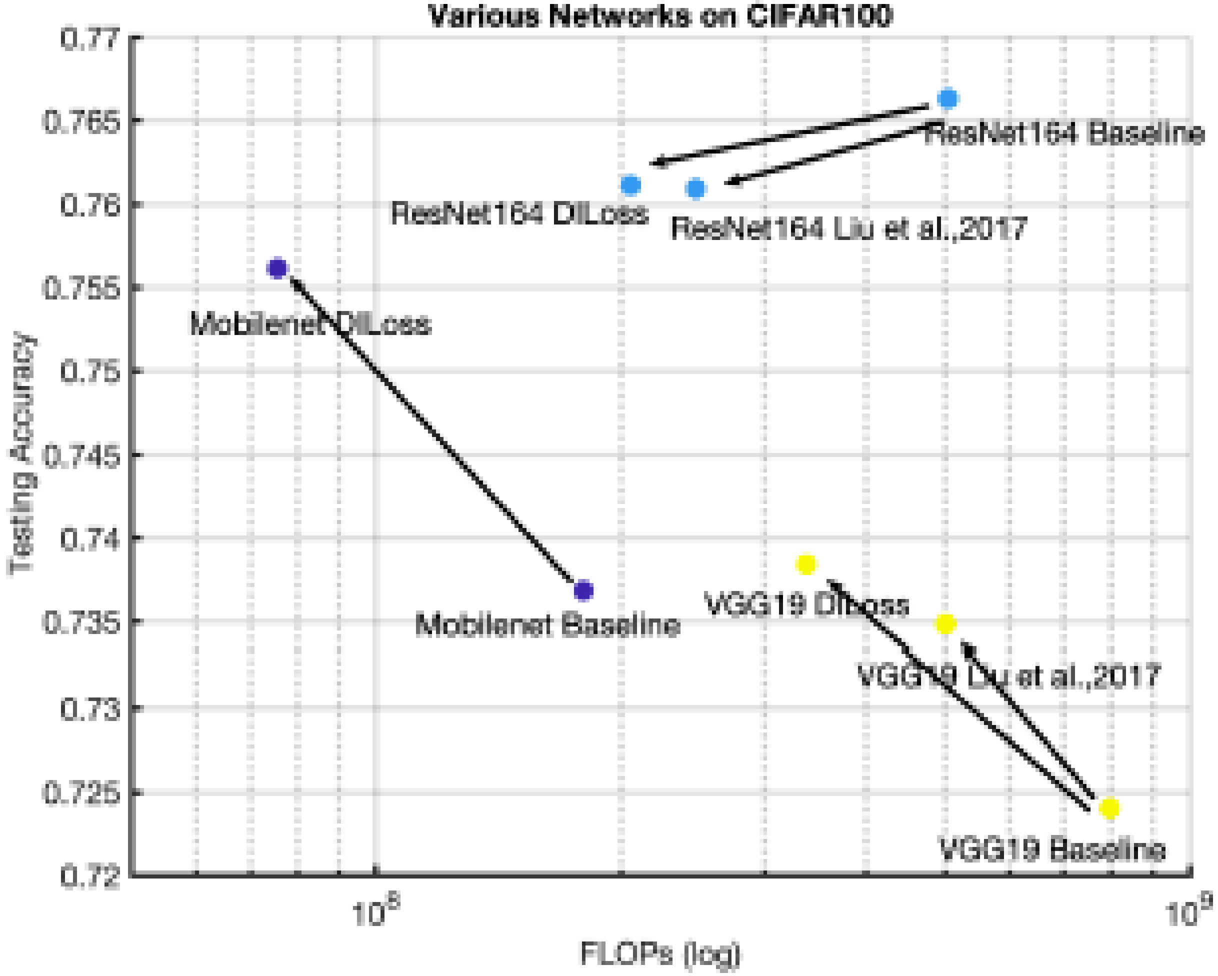
(c) CIFAR10: VGG16 speedup



(d) CIFAR10: Resnet56 speedup

Task	Models	Accuracy %	FLOPs	Params
CIFAR-10	*VGG-16 (baseline)*	93.25	6.26×10^8	1.5×10^7
	VGG-16 (Li et al., 2017)	93.41	4.12×10^8(65.81%)	5.4×10^6(36%)
	VGG-16 (FDR)	93.61	1.35×10^8(21.4%)	7.1×10^5(4.7%)
	VGG-16 (DILoss)	**94.07**	$\mathbf{1.28\times10^8}$**(20.45%)**	$\mathbf{5.32\times10^5}$**(3.55%)**
	ResNet-56 (baseline)	93.04	2.5×10^8	8.5×10^5
	ResNet-56 (Li et al., 2017)	93.06	1.81×10^8(72.4%)	7.3×10^5(85.88%)
	ResNet-56 (Yu et al., 2018)	93.01	1.41×10^8(56.4%)	4.94×10^5(58.12%)
	ResNet-56 (He et al., 2017)	91.9	1.25×10^8(50%)	-
	ResNet-56 (Zhuang et al., 2018)	93.49	1.25×10^8(50.25%)	4.3×10^5(50.76%)
	ResNet-56 (DILoss)	**93.84**	$\mathbf{8.38\times10^7}$**(33.52%)**	$\mathbf{3.12\times10^5}$**(35.52%)**
	ResNet-56 (bootstrap:DILoss+Zhuang)	**93.84**	$\mathbf{7.58\times10^7}$**(30.32%)**	$\mathbf{2.81\times10^5}$**(33.05%)**

CIFAR-100: Speedup



Task	Models	Accuracy %	FLOPs	Params
CIFAR100	*Mobilenet-v2 (baseline)*	73.68	1.8×10^8	2.4×10^6
	Mobilenet-v2 (DILoss)	**75.61**	$\mathbf{7.57\times10^7 (42.06\%)}$	$\mathbf{1.07\times10^6 (44.58\%)}$

ImageNet Classification

LPIRC 2018 Winners

Neural network architecture	Input image resolution	Data type	Accuracy (%)	Google Pixel-2 inference time (ms)	Accuracy/inference time (%/ms)
mobilenet v1	224x224	float32	70.2	81.5	0.86
mobilenet v1	224x224	uint8	65.5	68.0	0.96
mobilenet v1	**128x128**	**uint8**	**64.1**	**28.0**	**2.28**
mobilenet v2	150x150	uint8	64.4	36.6	1.75
mobilenet v2	132x132	uint8	62.7	31.8	1.97
mobilenet v2	130x130	uint8	59.9	31.2	1.91

For LPIRC 2019:

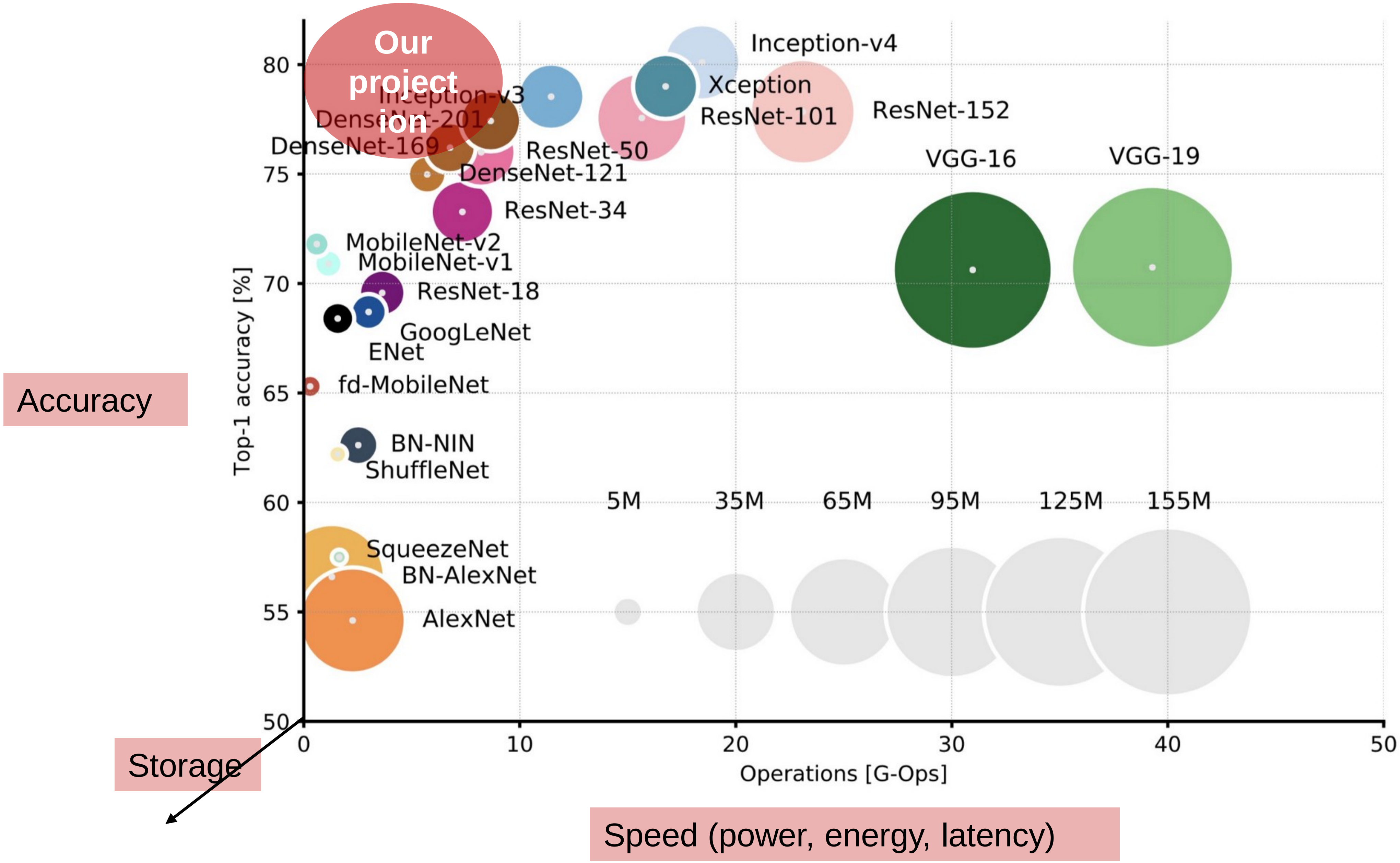
DI-Reduced Mobilenet V1:

Model	Acc %	FLOPs (10^6)	Params. (10^6)	GPU (ms)	Google Pixel-2 (ms)
MobileNet V1	70.2	574	4.24	29.1	81.5
MobileNet V1 (ours)	70.2	277	2.28	12.7	-
MobileNet V1 (8-bit)	67.26	-	-	12.2	-
MobileNet V2	71.8	300	3.41	8.3	-
MobileNet V2 (ours)	70.85	210	2.33		

DI-Reduced ResNet-50:

FLOPs	Parameters	Accuracy
4.08 B	25.50 M	76.112%

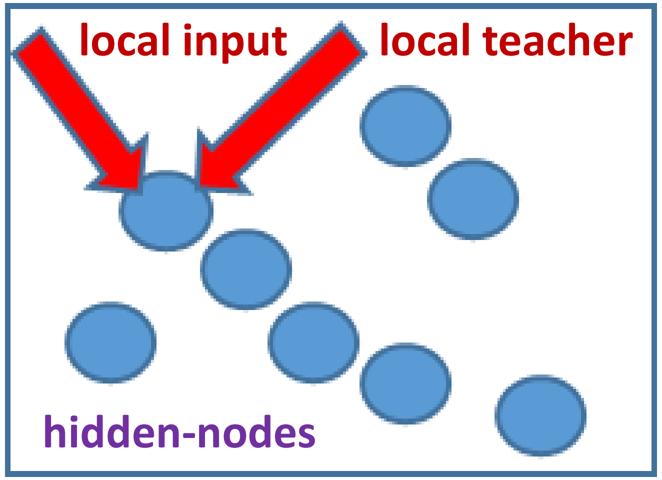
ImageNet Classification



Internal Teacher Labels (ITL)

Internal Optimization Metrics (IOM)

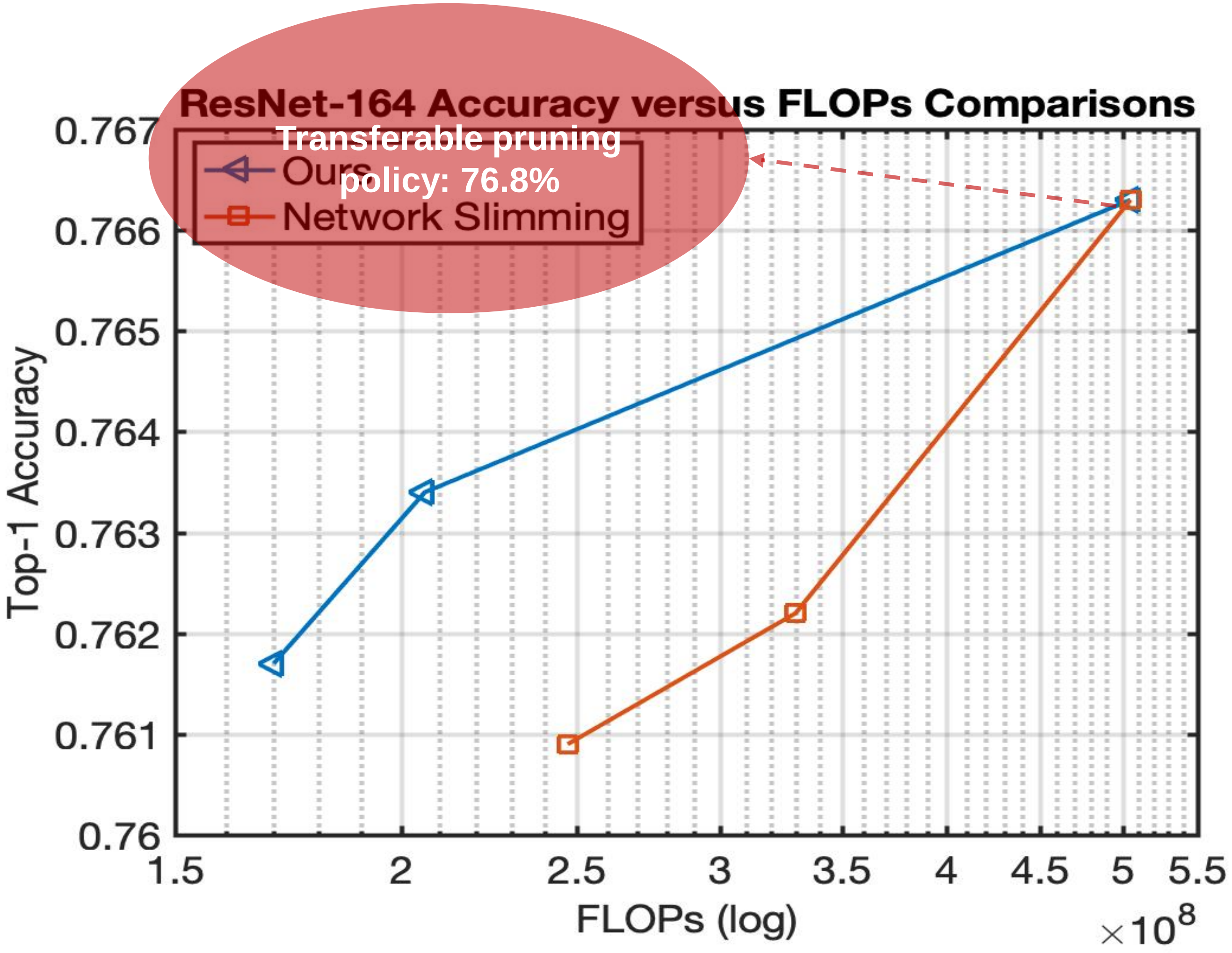
Local Metrics for Internal Training must be of classification-type because one-hot encoding won't work!!



For classification problem, the teacher labels can be metaphorically hidden in "Trojan-horses" and transported (along with the data) from the input layer to all hidden nodes.

- The original label, say B, is being sent to all hidden nodes; (discussed above)
- **Possible internal teacher labels ITLs are: granularity-adaptive (class or super-class), layer adaptive, or end-user-adaptive to facilitate INEX in XAI. (NEXT)**

Applying Transfer Learning to ResNet on CIFAR



A Major Focus of Explainable AI (XAI) is Explainable Learner (vs. Classifiable Learner)

XAI : Internal Neuron's Explainablility, championed by DARPA's XAI (or AI3.0).

Broad Agency Announcement

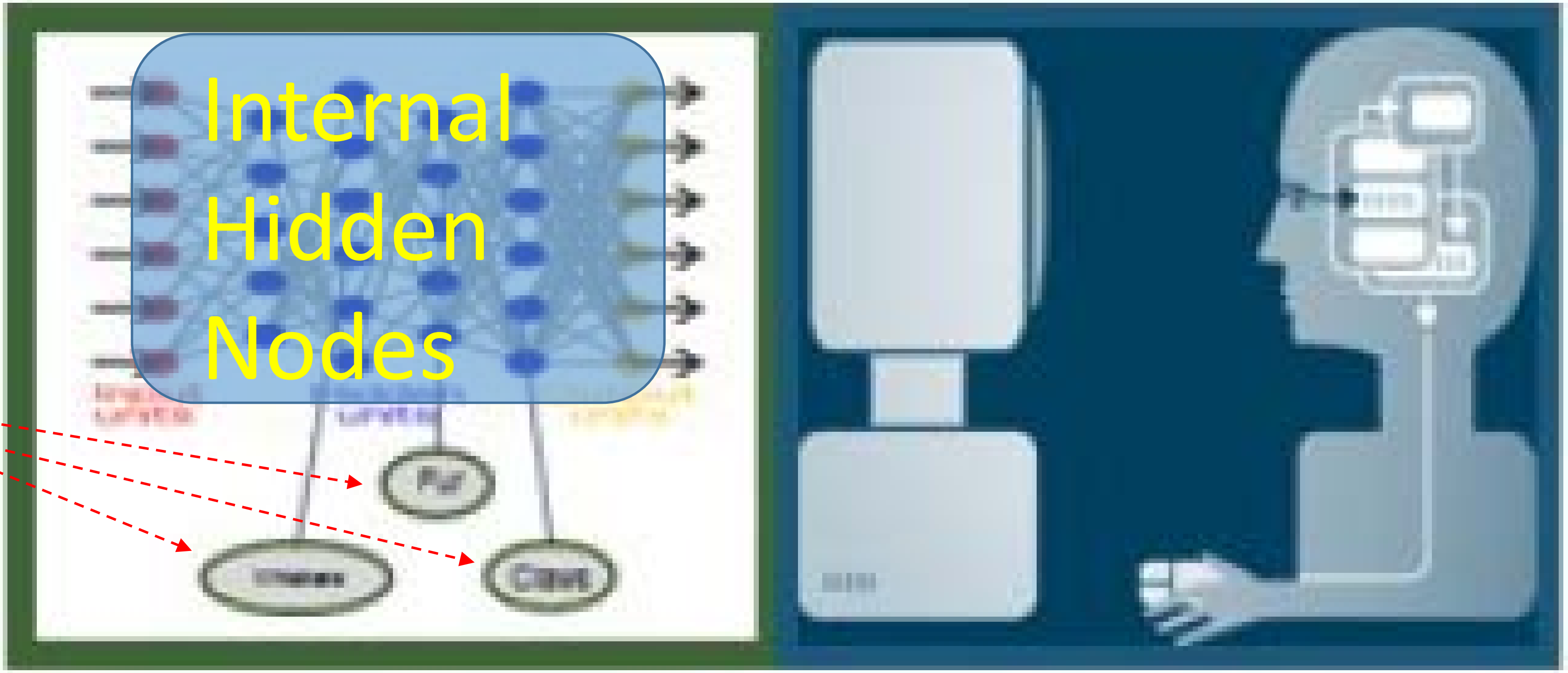
Explainable Artificial Intelligence (XAI)

DARPA-BAA-16-53

August 10, 2016

XAI TA1: Deep Leaning

Internal Node EXplainability (INEX): BAA ``XAI is most interested in explanations of higher-level decisions that would be relevant to the end user and the missions he/she needs to manage.



NN/AI: dónde quieres ir?

1950	⇒	2000	⇒	2020
NN1.0 model: MLP M. Minsky,, first neural network simulator, Princeton Ph. D. 1951, Paul Werbos, Ph. D. Harvard, 1974. Rumelhart, Hinton, and Williams, "Learning internal representations by error propagation," 1985.	⇒	**NN 2.0** ⇒ model: CNN ⇒ Deep Learning (深度学习)	⇒	new **NN** ⇒ Enhanced CNN: e.g. highways ⇒ Internal Learning (深入学习) **INER** (Internal Node Evaluation/Ranking)
AI 1.0 MIT AI Lab (1958, M. Minsky) Knowledge Systems Laboratory, (1970 Feigenbaum)	⇒	**AI 2.0** ⇒ Big-Data-Driven ⇒ Deep BP Learning	⇒	**XAI** **INEX** (Internal Node Explainability)