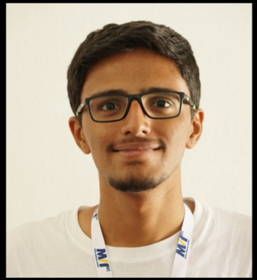
Deep Expander Networks

Efficient Deep Networks from Graph Theory



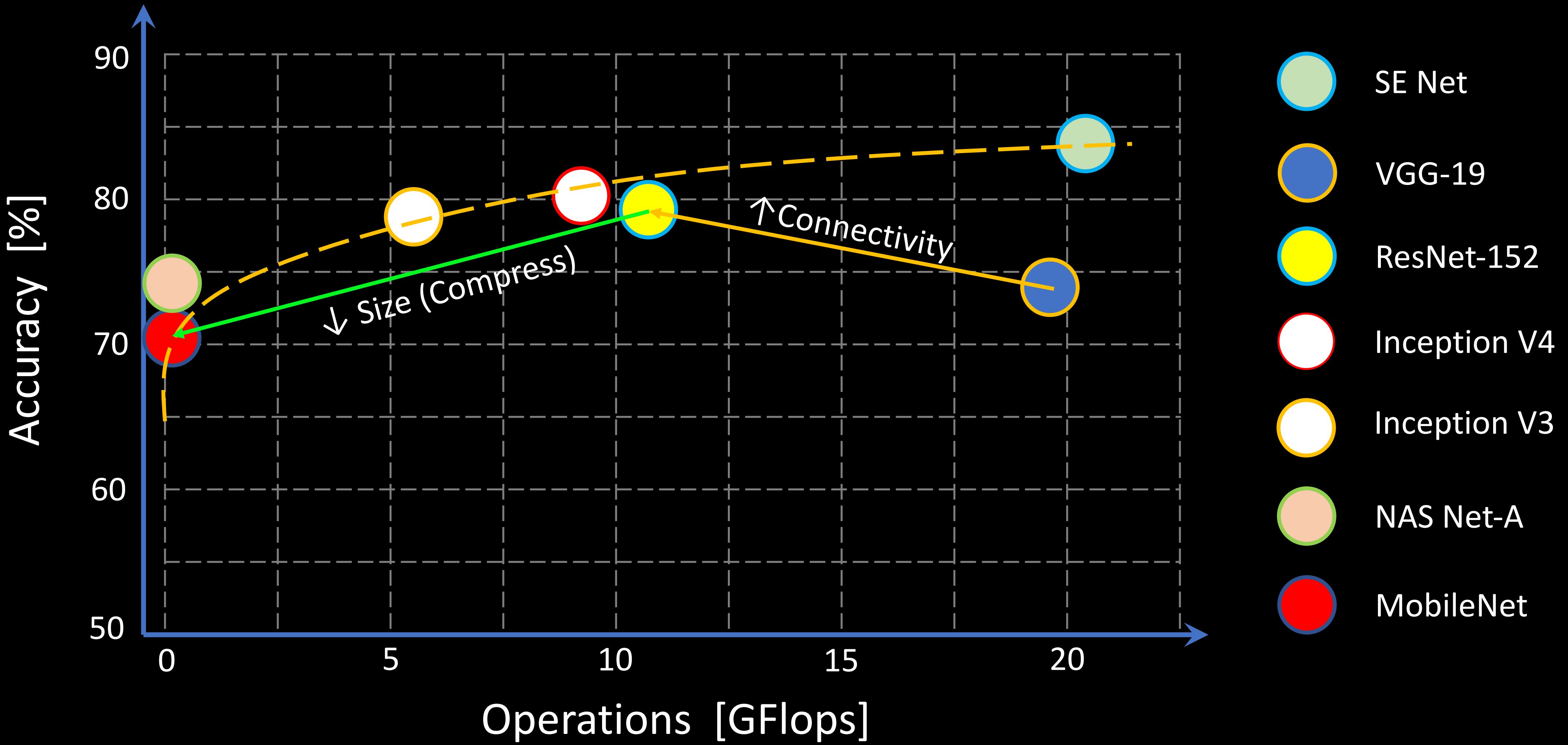
Ameya Prabhu*, Girish Varma*, Anoop Namboodiri

IIIT Hyderabad

INDIA

ECCV 2018
European Conference
on Computer Vision

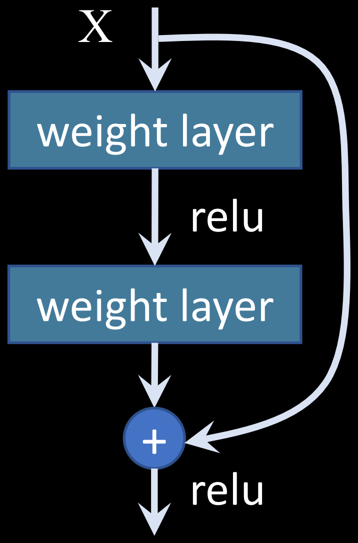
EFFICIENT CNNs

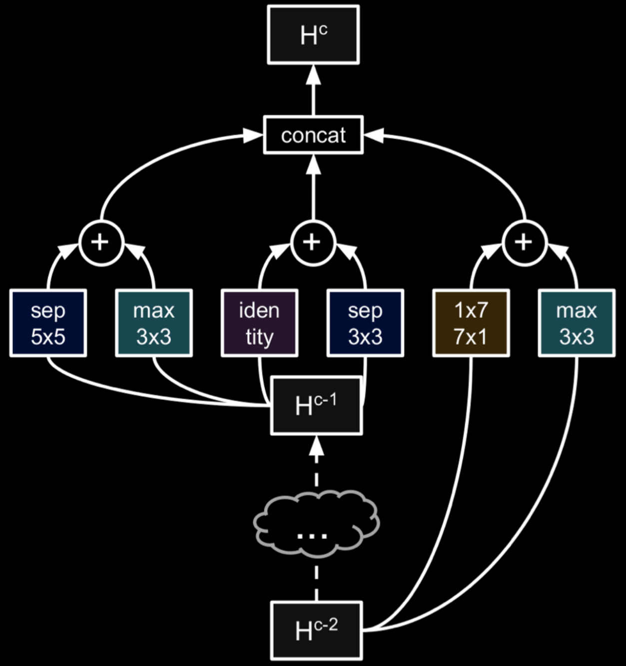


Approaches towards Efficiency

Better Network Design

- Inception Net
- ResNet
- NAS Net
- P-NAS Net

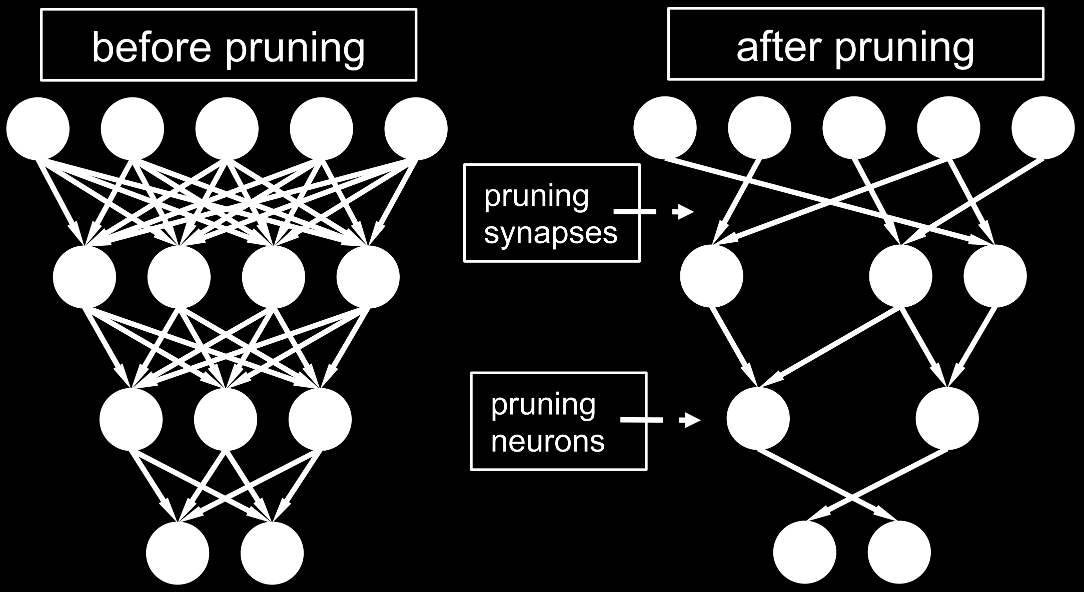




↑ Connectivity

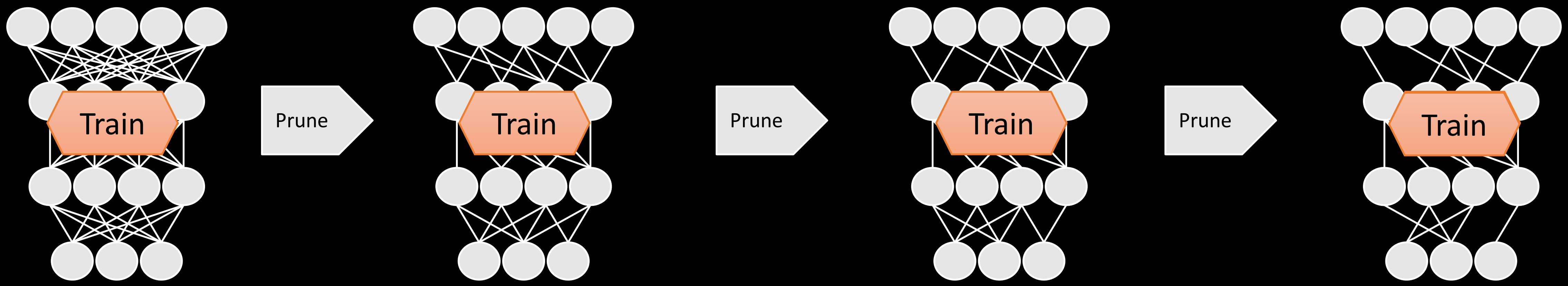
Efficient Layer Modification

- Group Convolutions
- Pruning



↓ Size

Better Layer Connections: Train ⟶ Prune



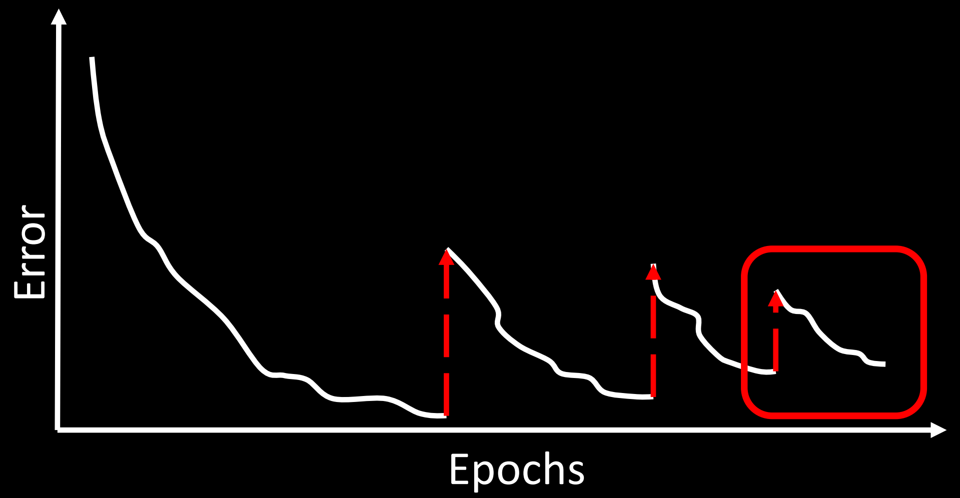
Train → Prune

✗ Need to train full network

✗ Need Multiple trainings

✗ Layer structure specific to given data

Not Transferrable



Can We: Prune → Train

Train → Prune

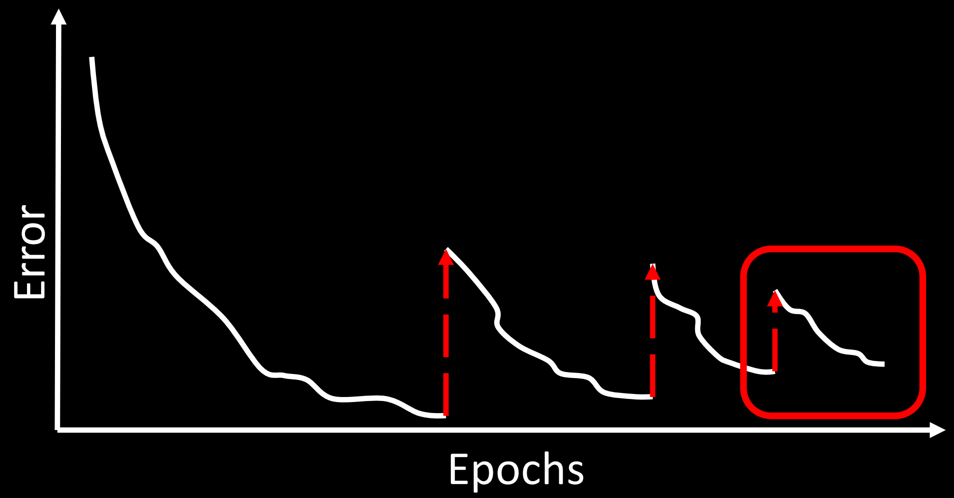
- ✗ Need to train full network
- ✗ Need Multiple trainings
- ✗ Layer structure specific to given data

Not Transferrable

Prune → Train

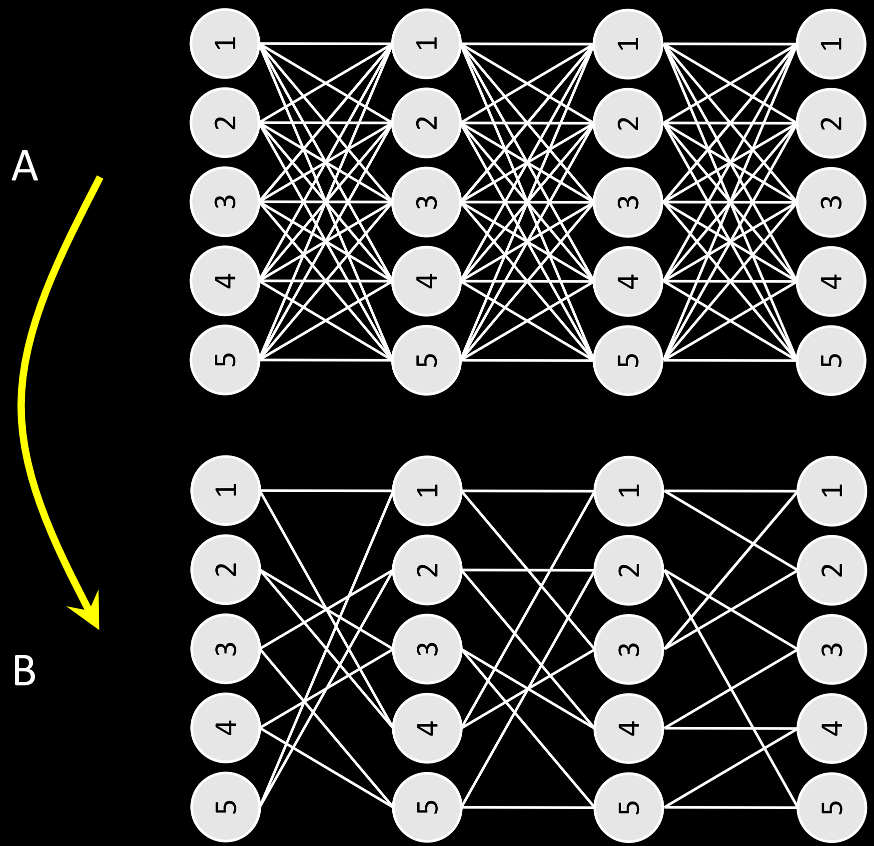
- ✓ Train a compact network
- ✓ Single training
- ✓ Generic layer structure, independent of data

Transferrable



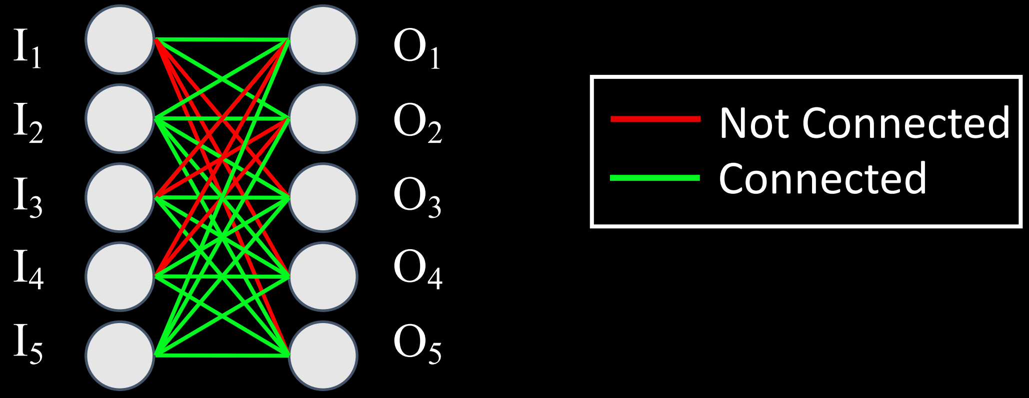
Pruning Without Data

- Need to sparsify connections
- Need to ensure multi-layer connectivity



Regular Pruning

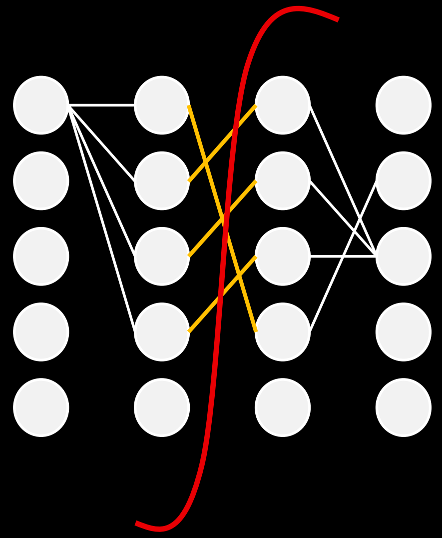
- Not well connected
- Retraining does not help



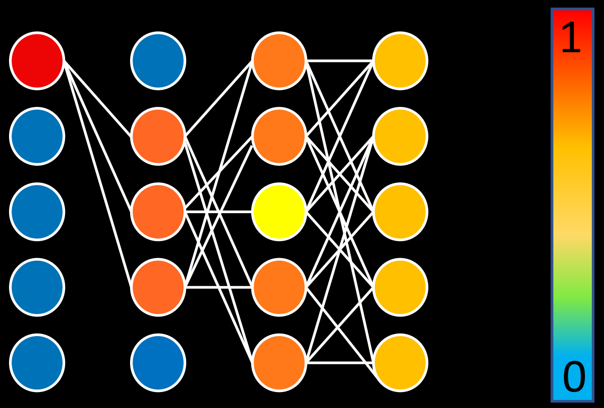
All this to be done without data!!

Expander Graphs

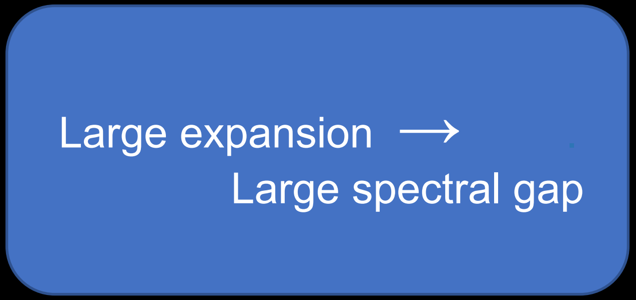
Combinatorics: Highly connected; Sever many edges to disconnect any large part of the graph



Probability: Random walk on these converges to its limiting distribution as rapidly as possible.

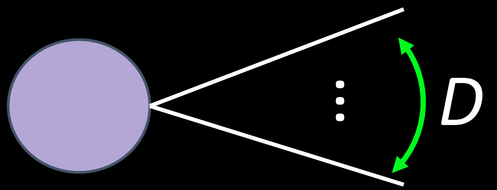


Algebra: First positive eigenvalue of their laplace operator is bounded away from zero.

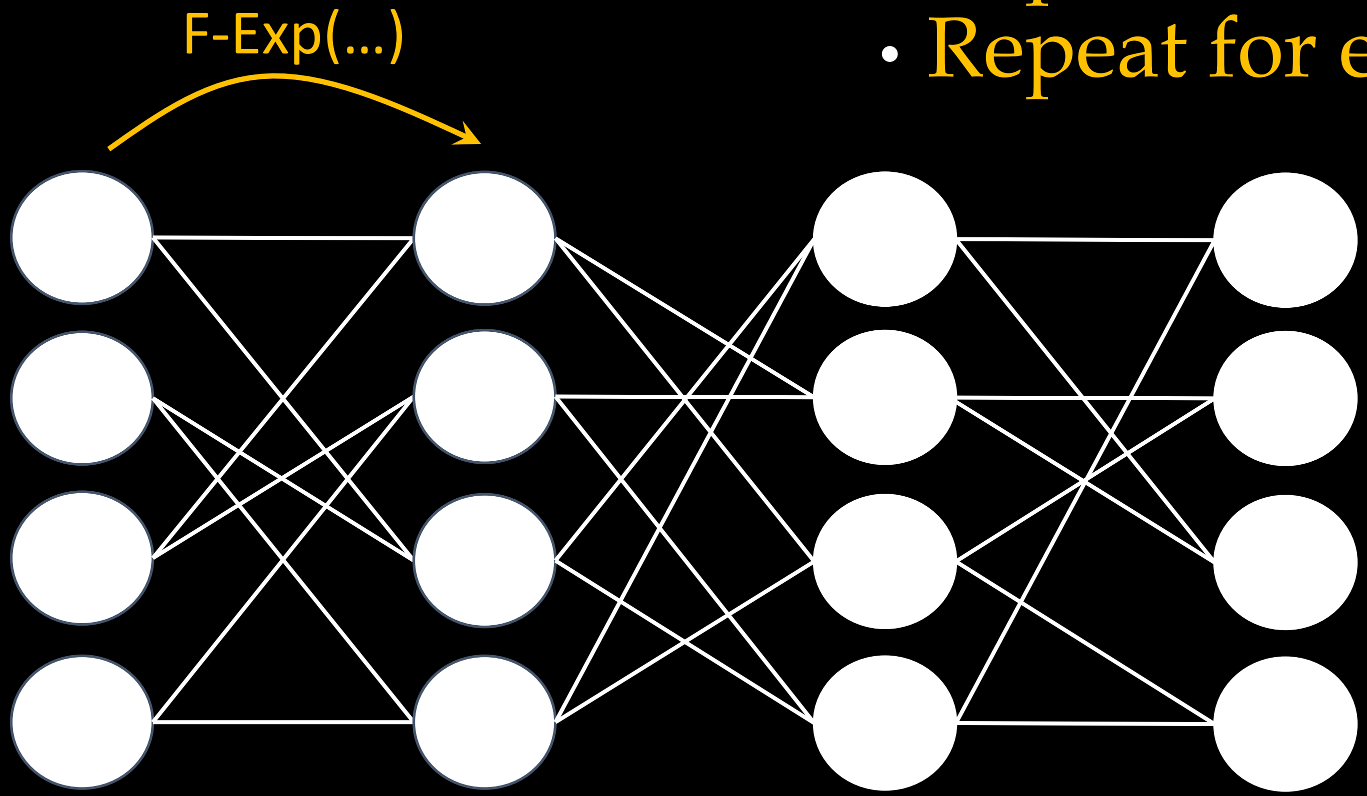


Expander Graph are are simultaneously sparse and highly connected.

Constructing Expander Layers

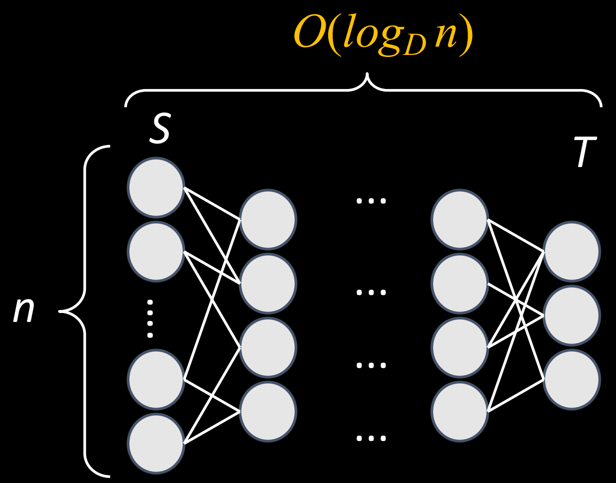


- Pick an input node
- Connect it to *D* random outputs
- Repeat for every input node
- Repeat for every layer

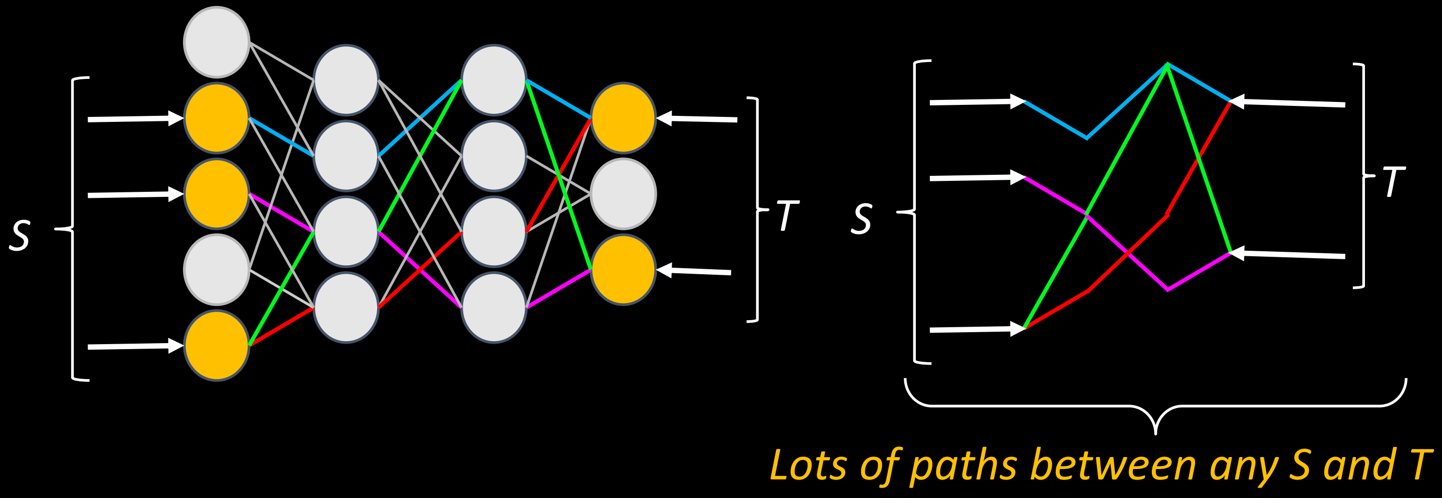


Guarantees About X-Nets

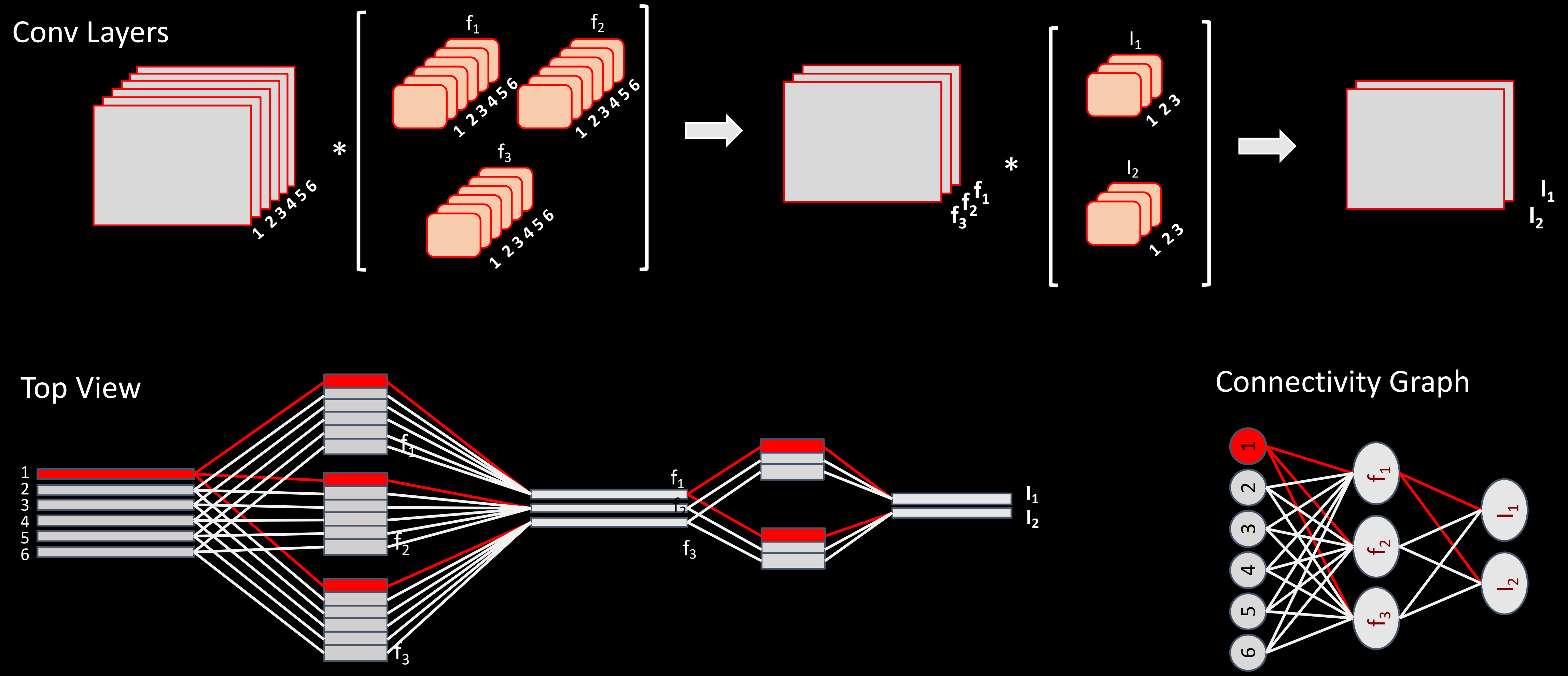
Theorem 1 (Sensitivity):
Let n be the number of input as well as output nodes in the network and G1 , G2 ,⋯ , Gt be D-regular bipartite expander graphs with n nodes on both sides. Then every output neuron is sensitive to every input in a Deep X-Net defined by G i 's with depth $t = O(\log_D n)$.



Theorem 2 (Rich Connectivity):
Let n be the number of input as well as output nodes in the network and G be D regular bipartite expander graph with n nodes on both sides. Let S,T be subsets of input and output nodes in the X-Net layer defined by G. The number of edges between S and T
$|E| \approx \frac{D|S||T|}{n}$

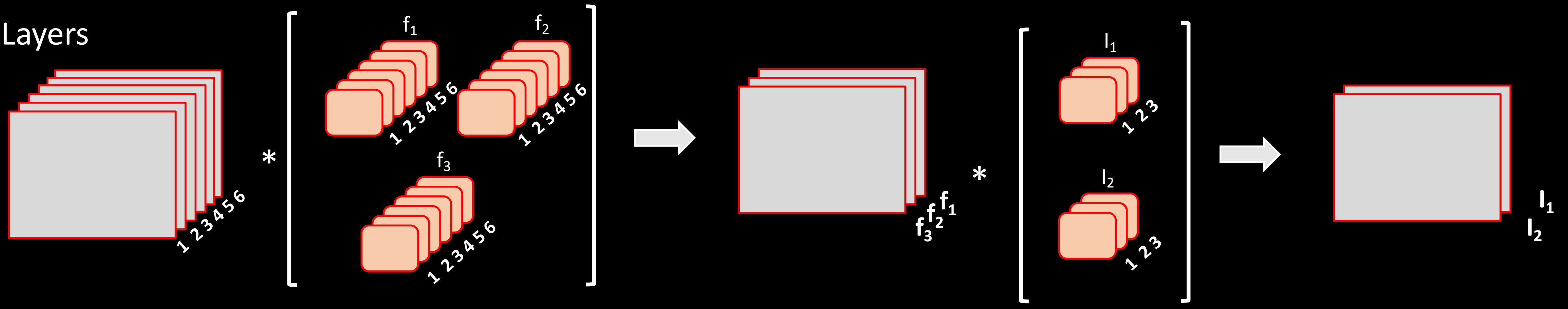


Note: Connectivity Graph of Convolutions

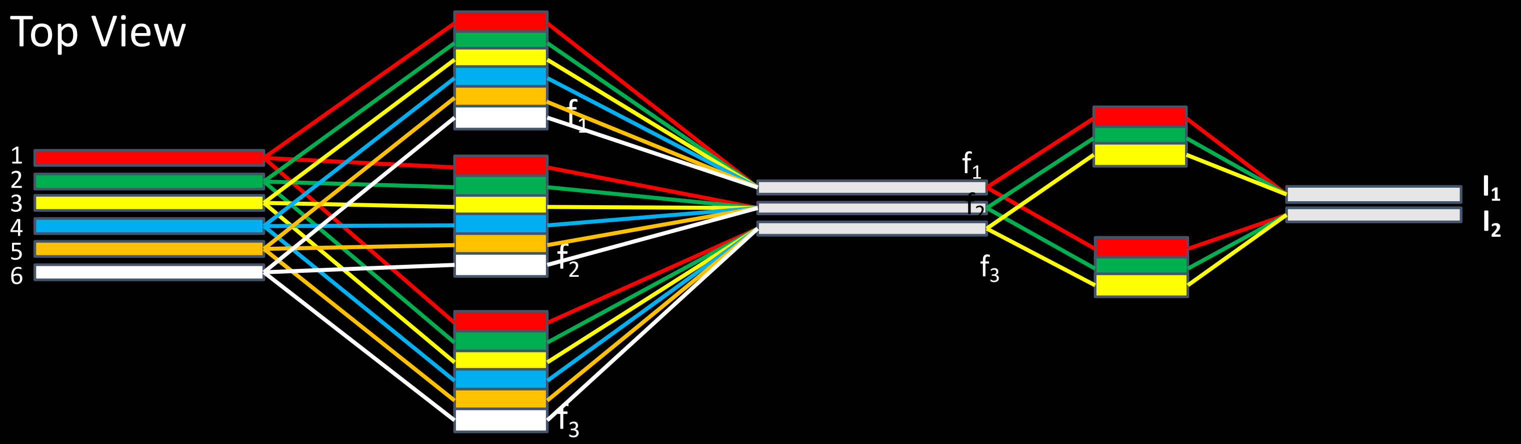


NOTE: CONNECTIVITY GRAPH OF CONVOLUTIONS

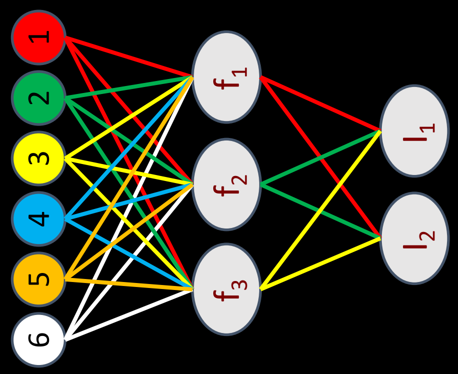
Conv Layers



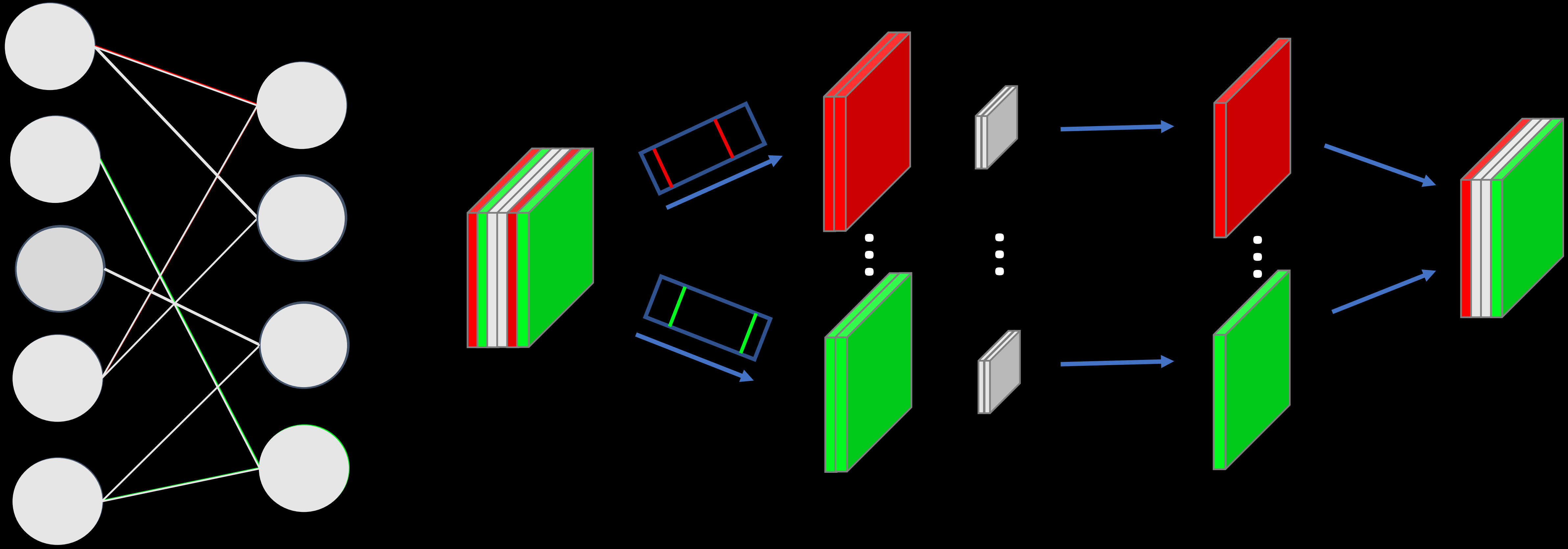
Top View



Connectivity Graph

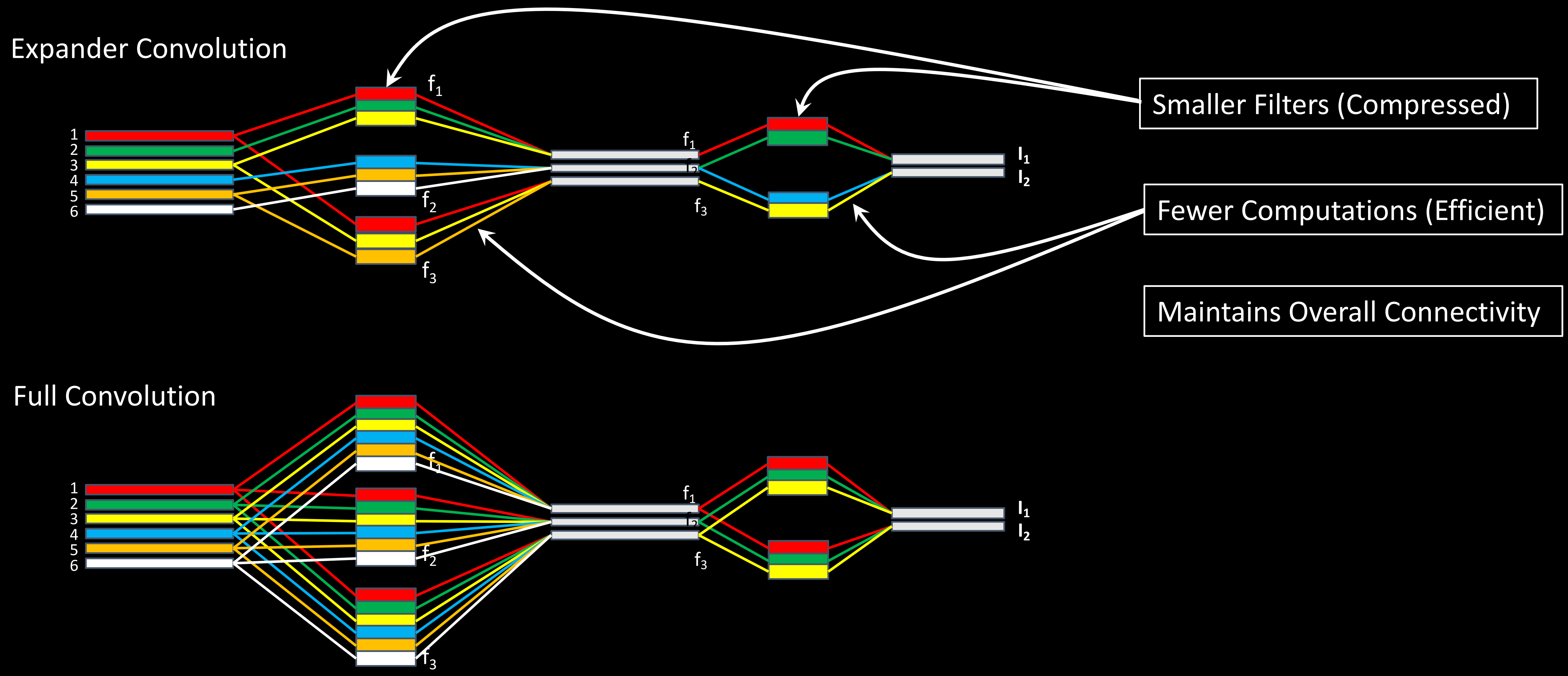


Our Convolutional Layer



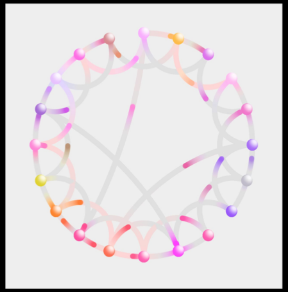
Red and green represent the subsets that are connected

Expander vs. Full Convolution



Implementing X-Nets

OpenAI



BlockSparse

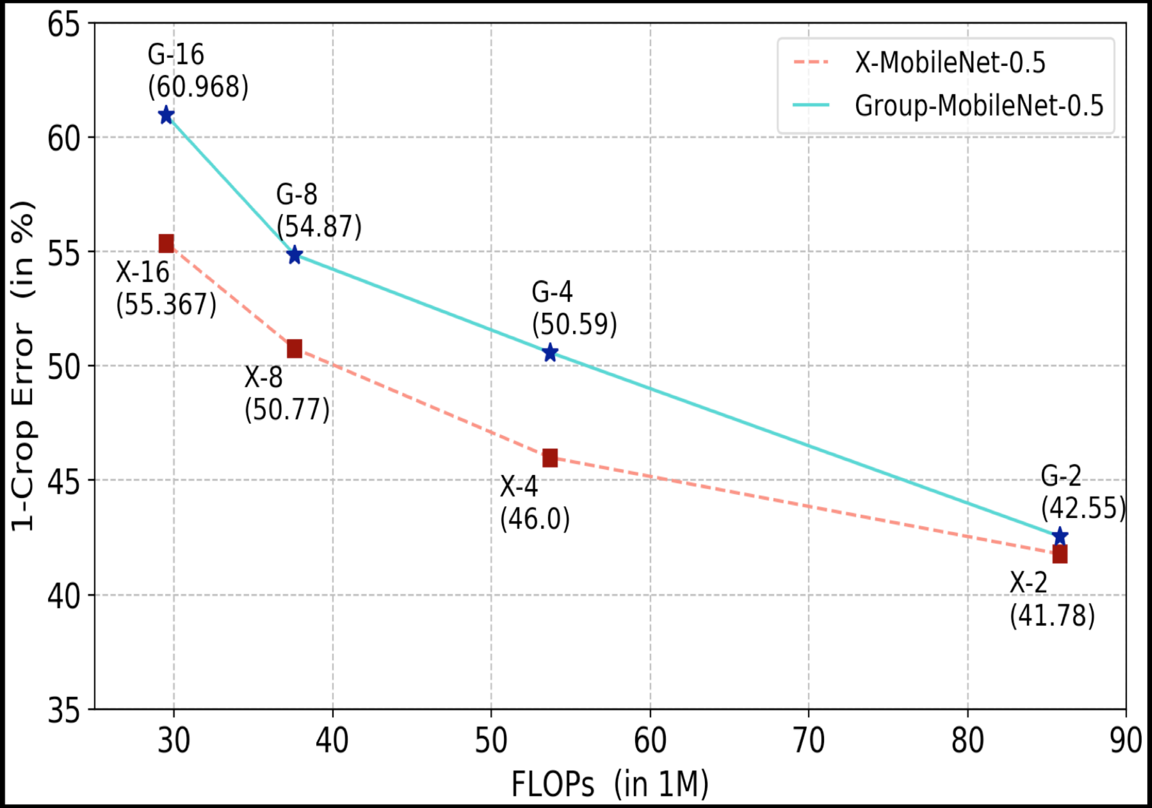
Experimental Results

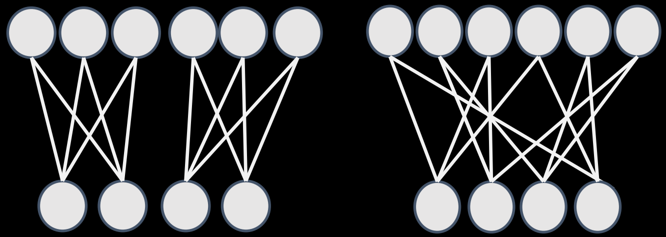
Comparisons with :

- Layer Connectivity Graphs: Group Convolution
- Network Compression: Pruning
- Efficient Architectures: ResNet and DenseNet

Benchmarking with Group Convolution

X-Conv beats G-Conv by ~ 4-5%
on a compact MobileNet-0.5 on Imagenet





Compression	G-Conv	X-Conv (Ours)	Err. Red.
2x	42.55%	41.78%	**0.8%**
4x	50.59%	46.00%	**4.6%**
8x	54.87%	50.77%	**4.1%**
16x	60.97%	55.37%	**5.6%**

Comparison with Pruning

VGG-16 on CIFAR-10

Method	Accuracy	# Params
Li et al.	93.4 %	5.4 M (2.8x)
NW Slimming	93.8 %	2.3 M (6.5x)
X-VGG 16-1	**93.4 %**	**1.65 M (9x)**
X-VGG 16-2	**93.0 %**	**1.15 M (13x)**
VGG-16 Orig	94.0 %	15.0 M (1.0x)

X-Nets are as compressible as the best pruning techniques

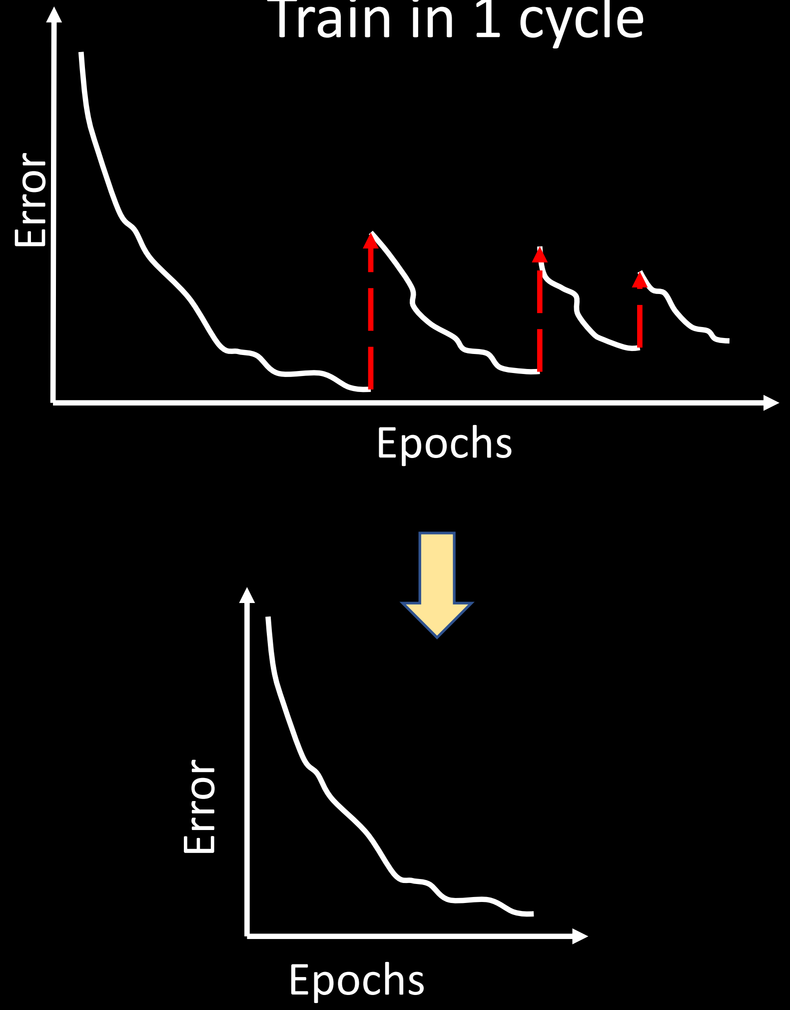
AlexNet on ImageNet

Method	Accuracy	# Params
Collins et al.	55.1 %	15.2 M (4x)
Zhou et al.	54.4 %	14.1 M (4.3x)
Han et al.	57.2 %	6.7 M (9.1x)
Srinivas et al.	56.9 %	5.9 M (10.3x)
Guo et al.	**56.9 %**	**3.4 M (18x)**
X-AlexNet-1	**55.2 %**	**7.6 M (8x)**
X-AlexNet-2	**56.2 %**	**9.7 M (6.3x)**
AlexNet-Orig	57.2 %	61 M (1.0x)

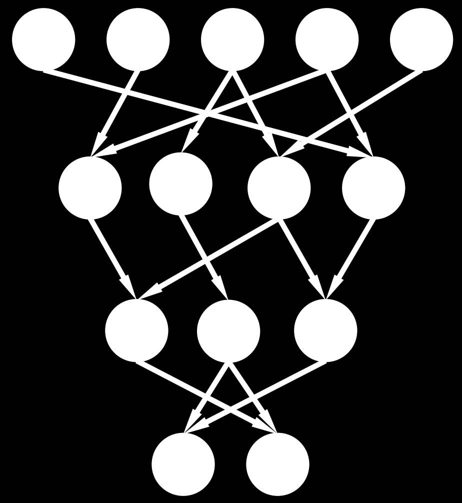
Failure Case ?

Advantages over Pruning

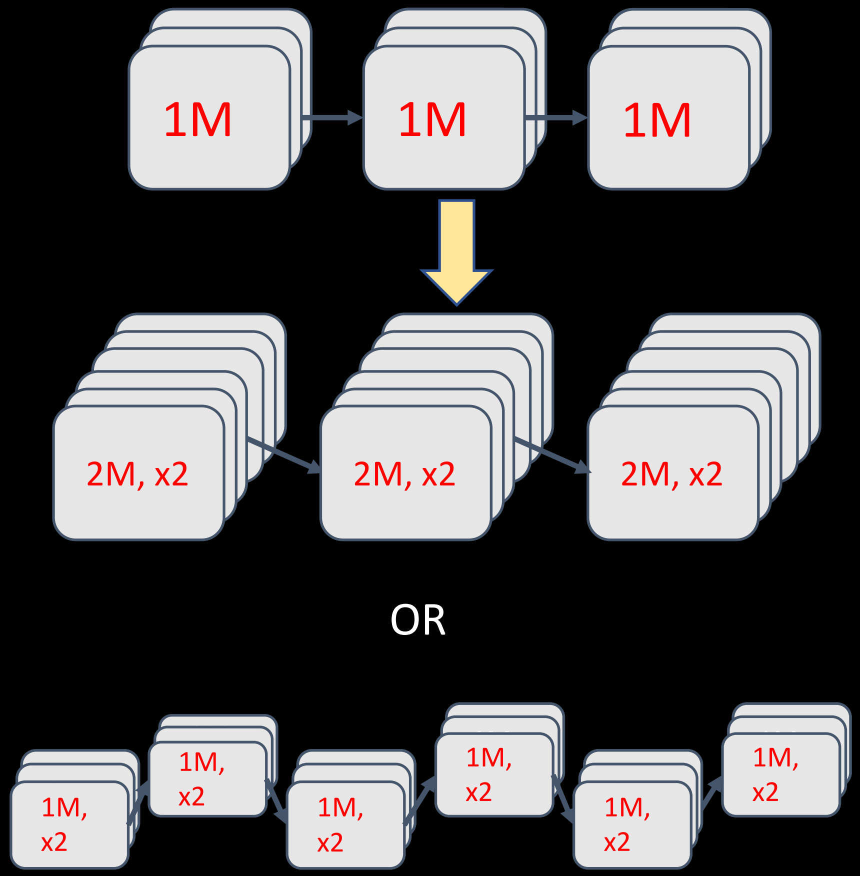
Train in 1 cycle



Transferable Architectures

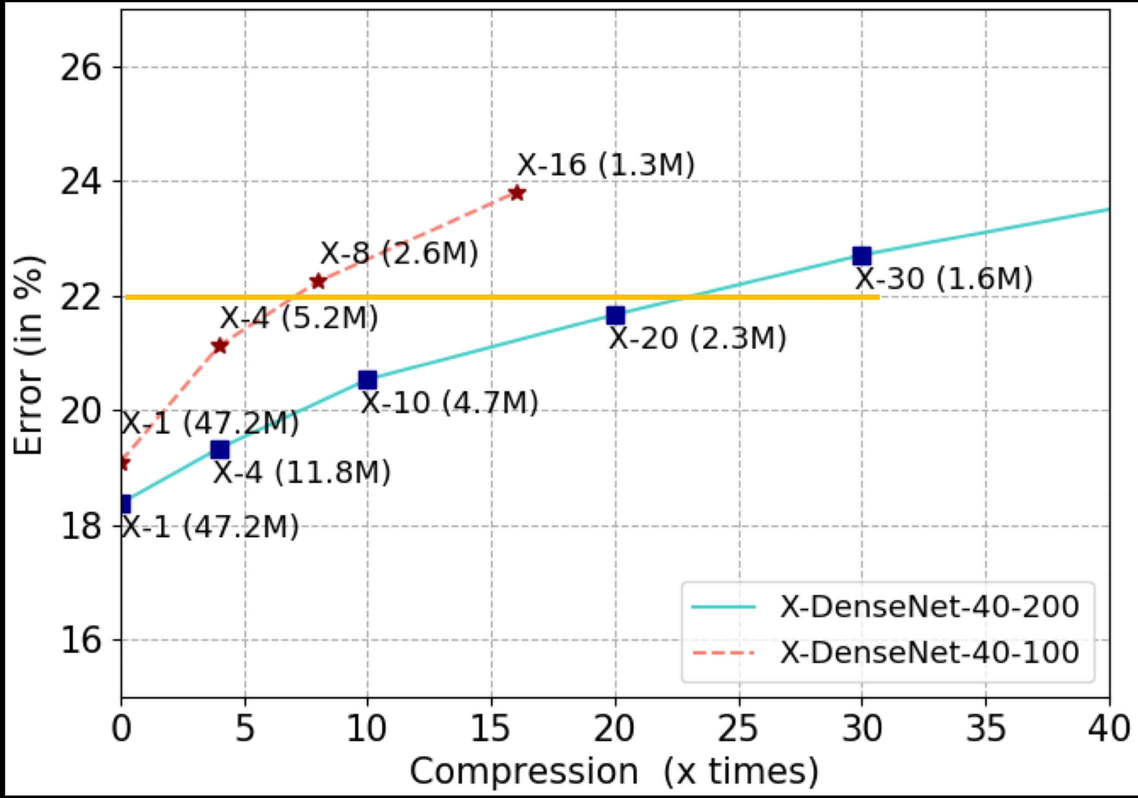


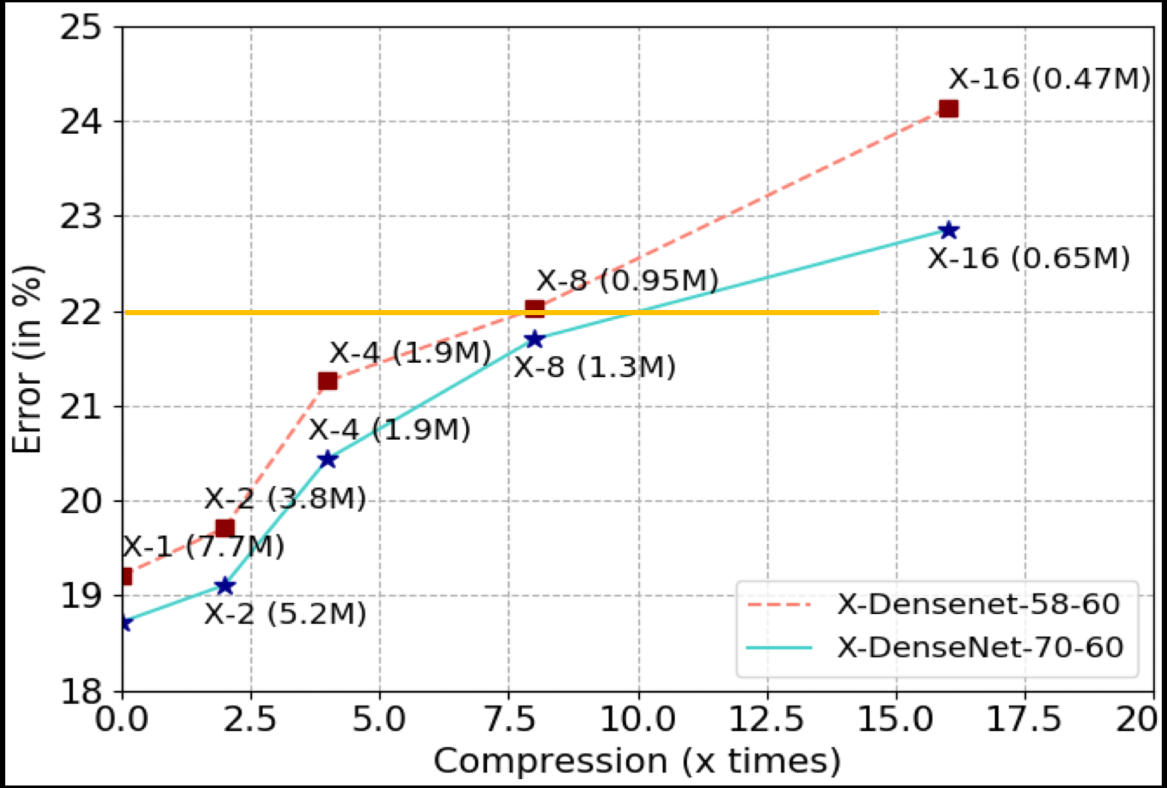
Go Wider / Deeper



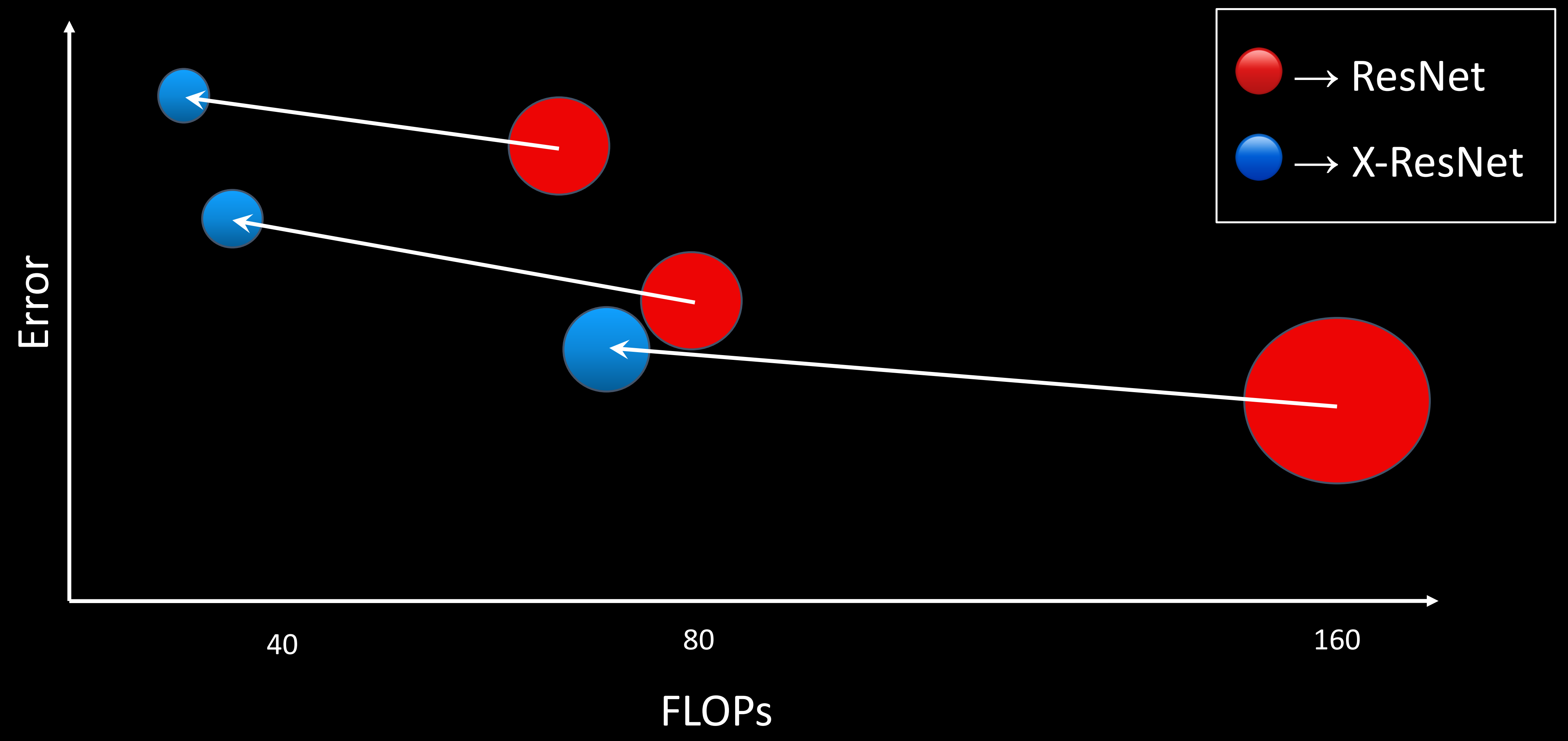
Going Wider and Deeper

Wider/Deeper networks with higher compression achieves same error rate with fewer parameters

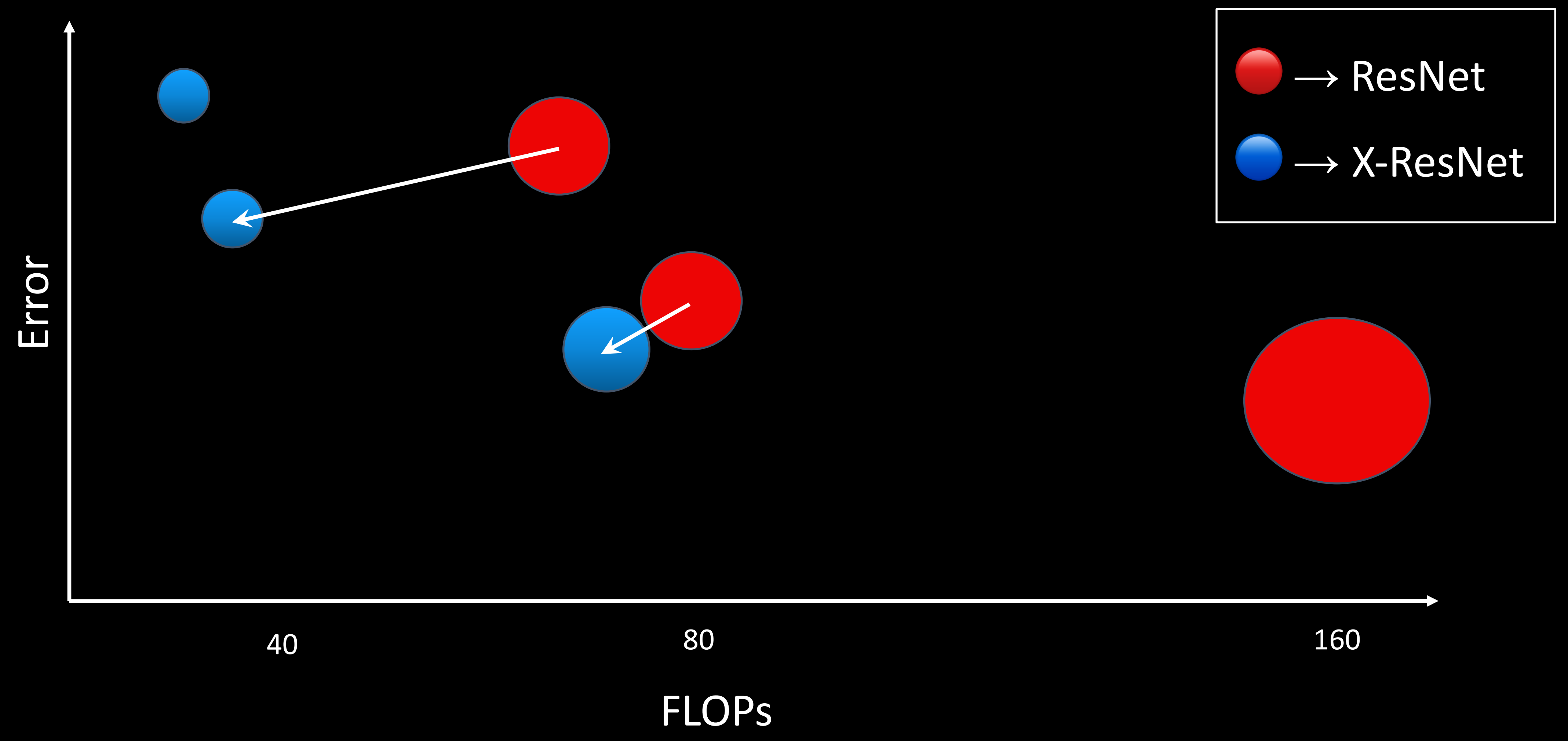




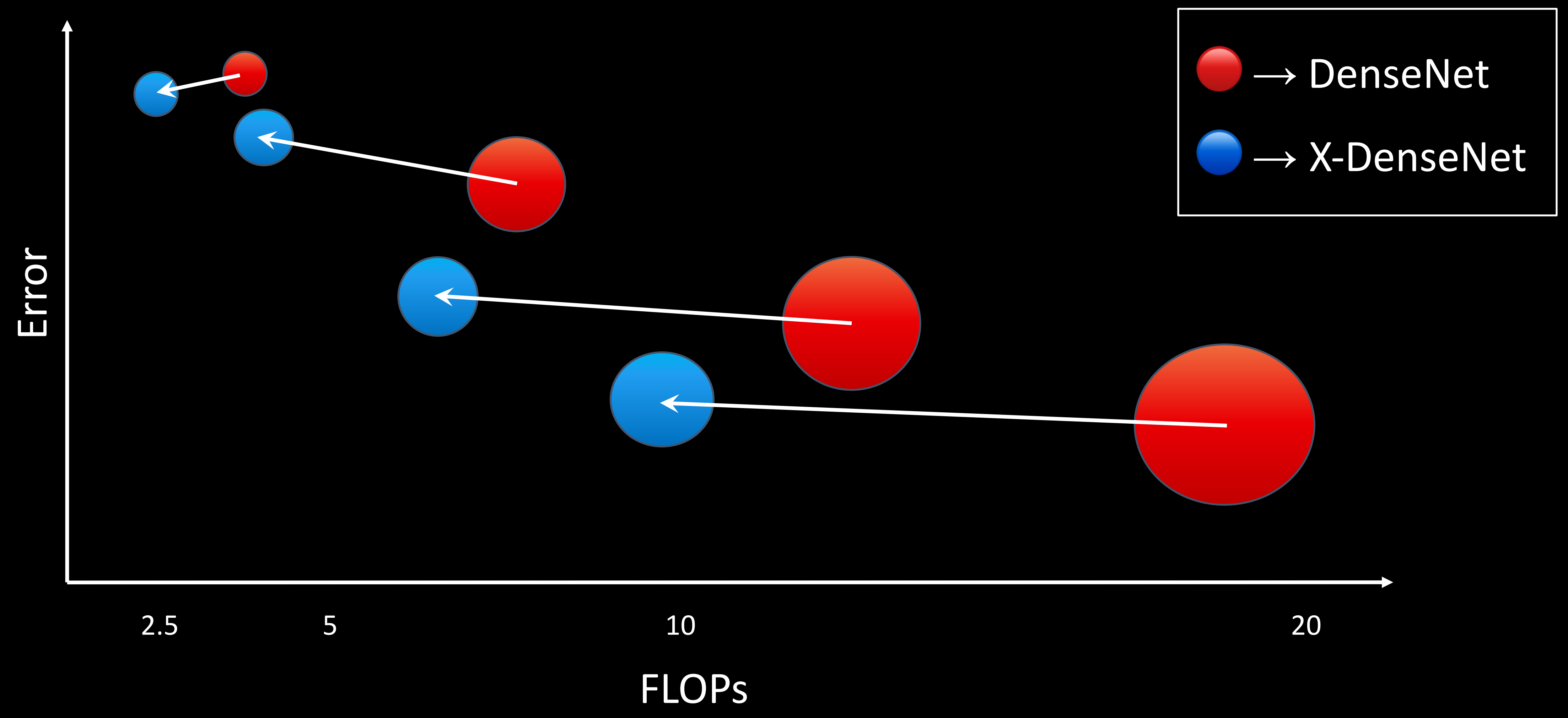
ResNet vs X-ResNet on ImageNet



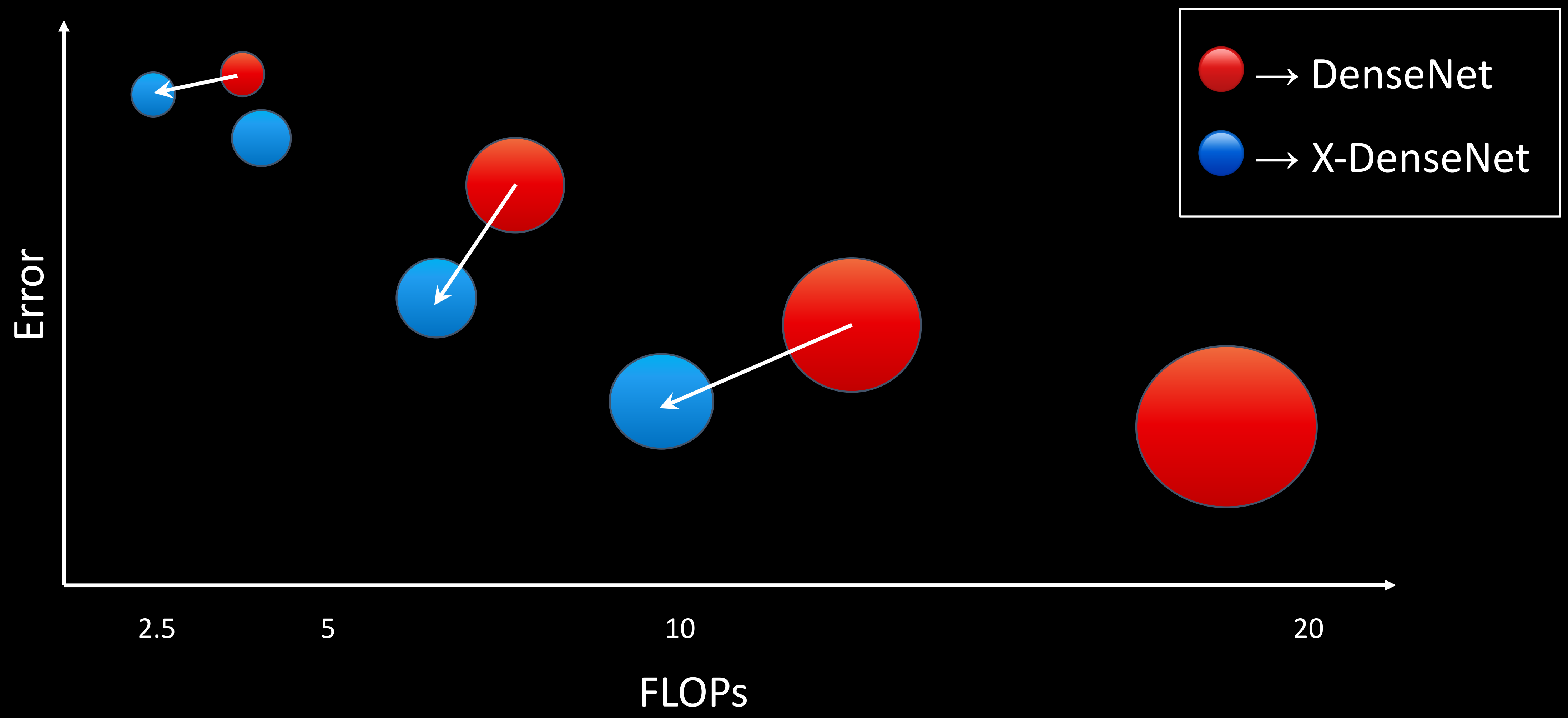
ResNet vs X-ResNet on ImageNet



DenseNet vs X-DenseNet on CIFAR-10

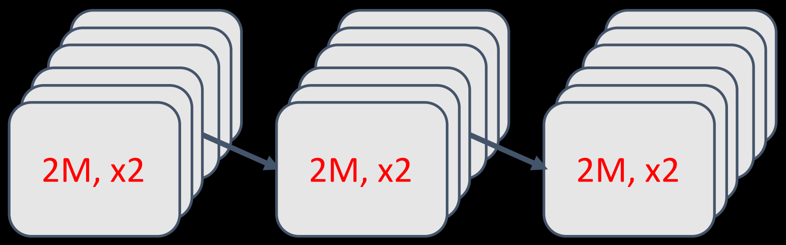


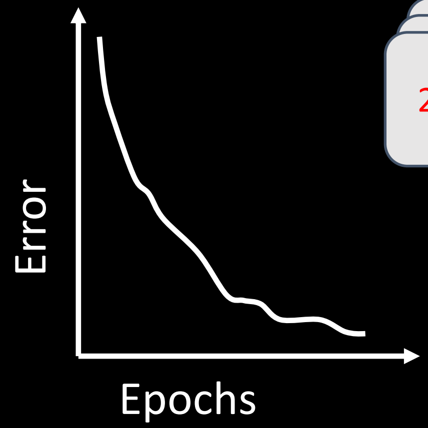
DenseNet vs X-DenseNet on CIFAR-10

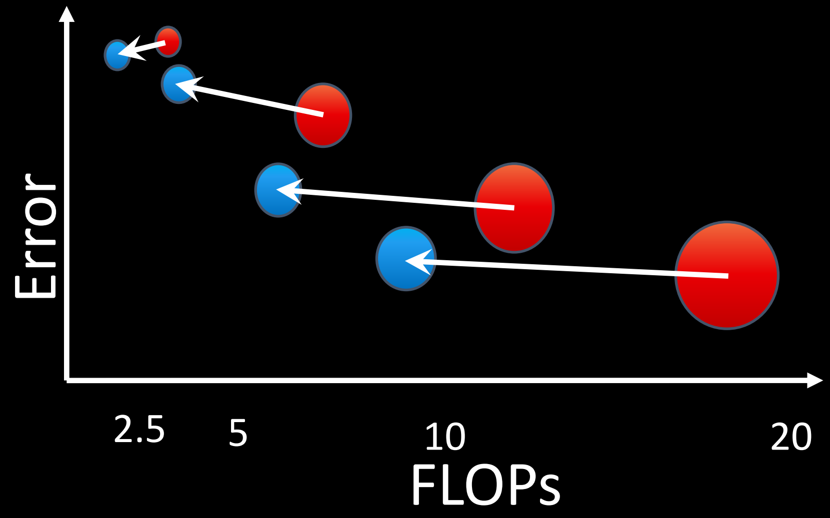


In Summary:

- X-Nets provide a principled way to compress deep networks.
- Single-cycle training of a lighter data-agnostic network.
- Allows training of wider and deeper networks.
- Achieves good error-flops trade-off.
- Highlights the use of global connectivity analysis in network architecture design.







THANK YOU !!

Using our Pytorch Code:

```
from layers import ExpanderLinear,ExpanderConv2d
nn.Conv2d(...)   →   ExpanderConv2d(..., expandSize=128)
nn.Linear(...)   →   ExpanderLinear(..., expandSize=256)
```

Visit us @
Poster ID: P-4A-04

GitHub Repo: https://github.com/DrImpossible/Deep-Expander-Networks