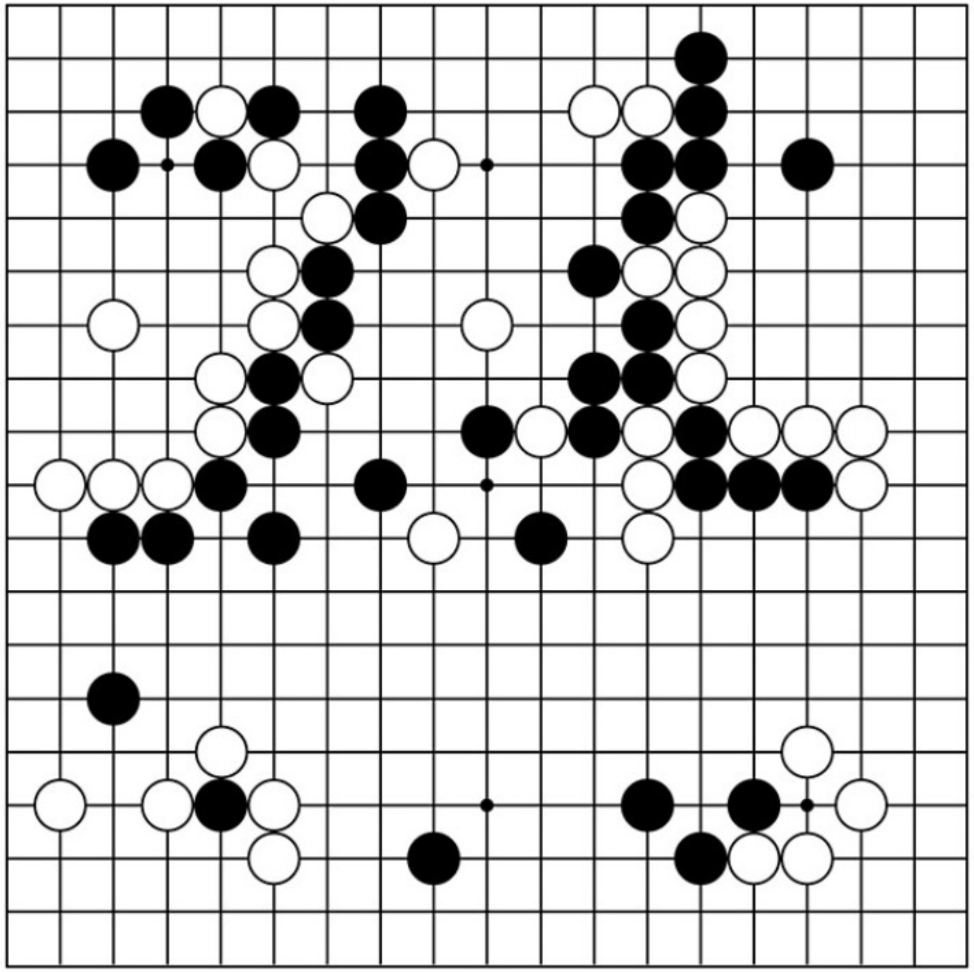
# Scalable Visual Pre-training: Past, Present, and Paths Forward

Saining Xie
Courant CS, NYU
saining.xie@nyu.edu

04/26/2023 @ UT Dallas



$3^{361}$ possible states?🤔

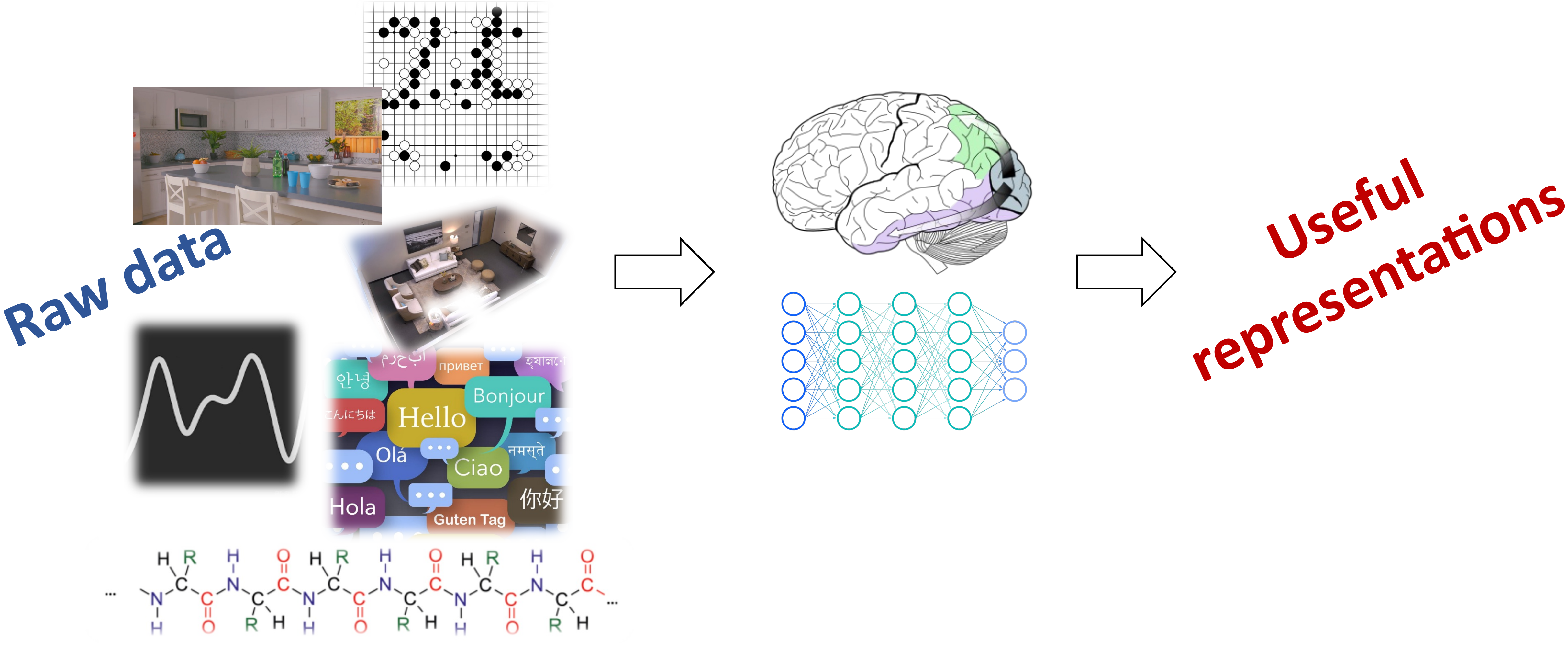
(>> number of atoms in the universe)

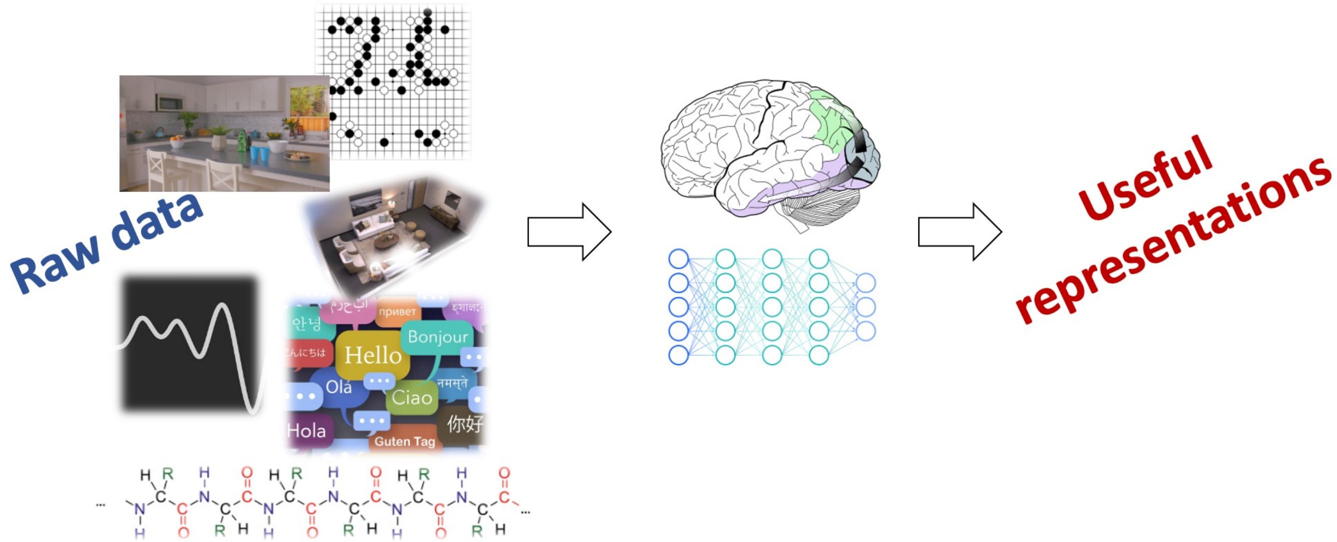
Figure: AlphaGo vs Lee Sedol G4 move 78


$256^{3\times600\times800}$

possible states?

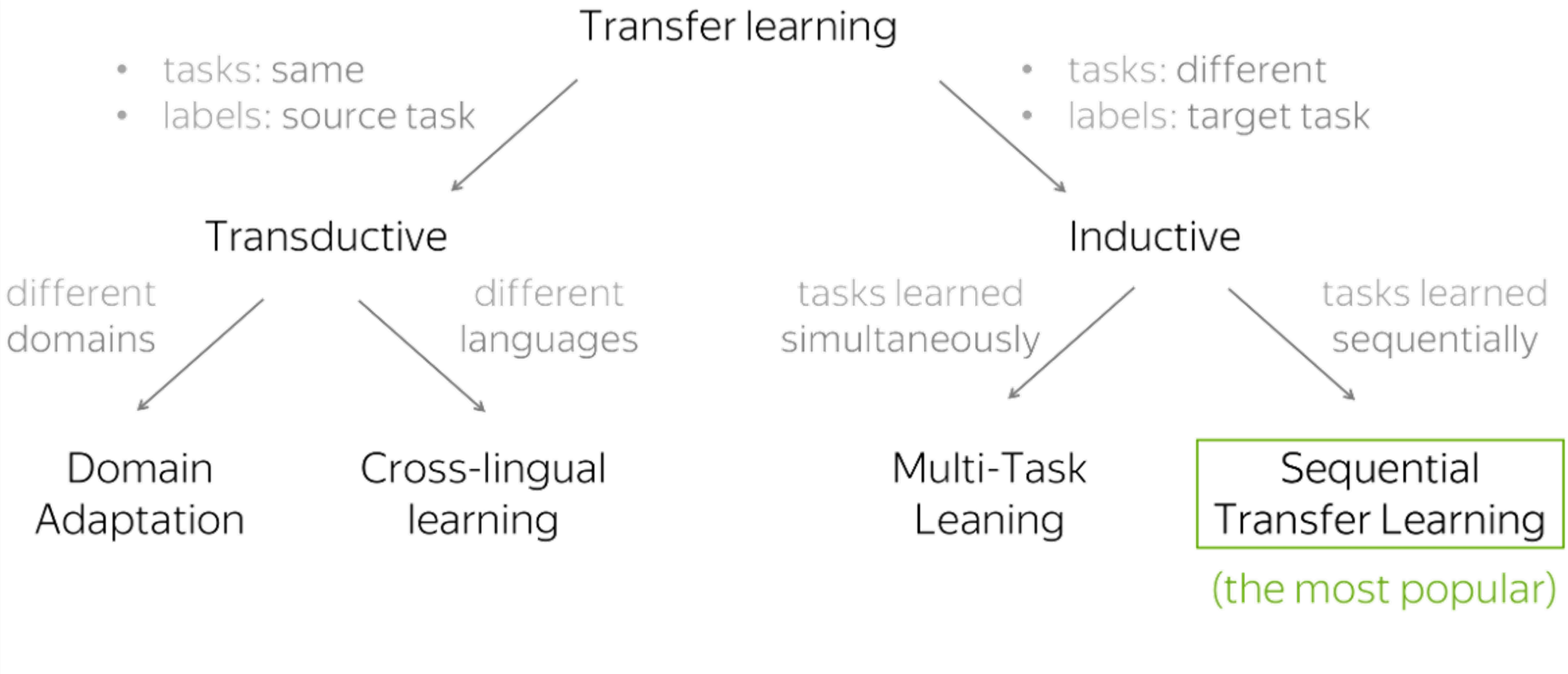
Figure: detectron2


Raw data
привет
Bonjour
Hello
Olá
Ciao
नमस्ते
Hola
你好
Guten Tag
Useful representations

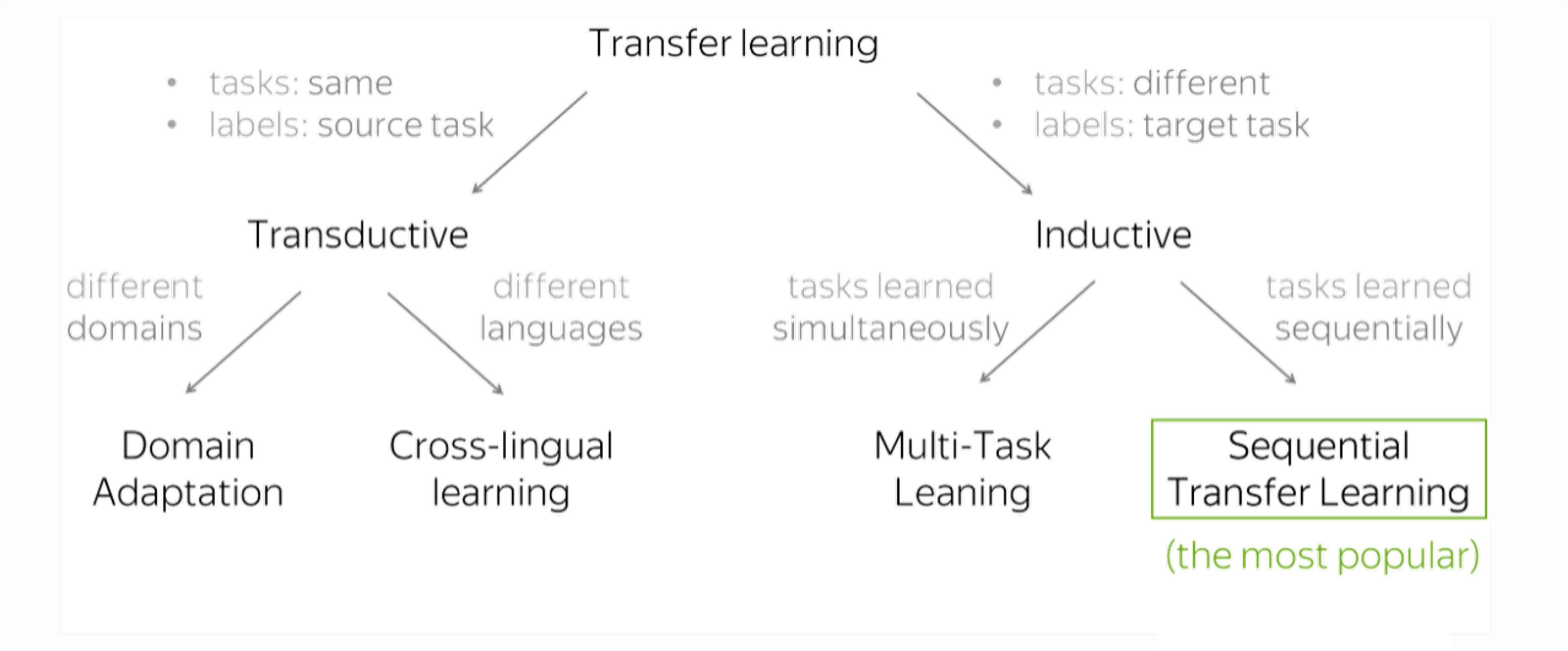



- In some cases, we can directly "***use***" the useful representations.

- In many applications, we need to "***transfer***" the useful representations to some downstream tasks.




This taxonomy is from Sebastian Ruder's blog post.




Pre-training → Fine-tuning*

# Architecture: what to train?

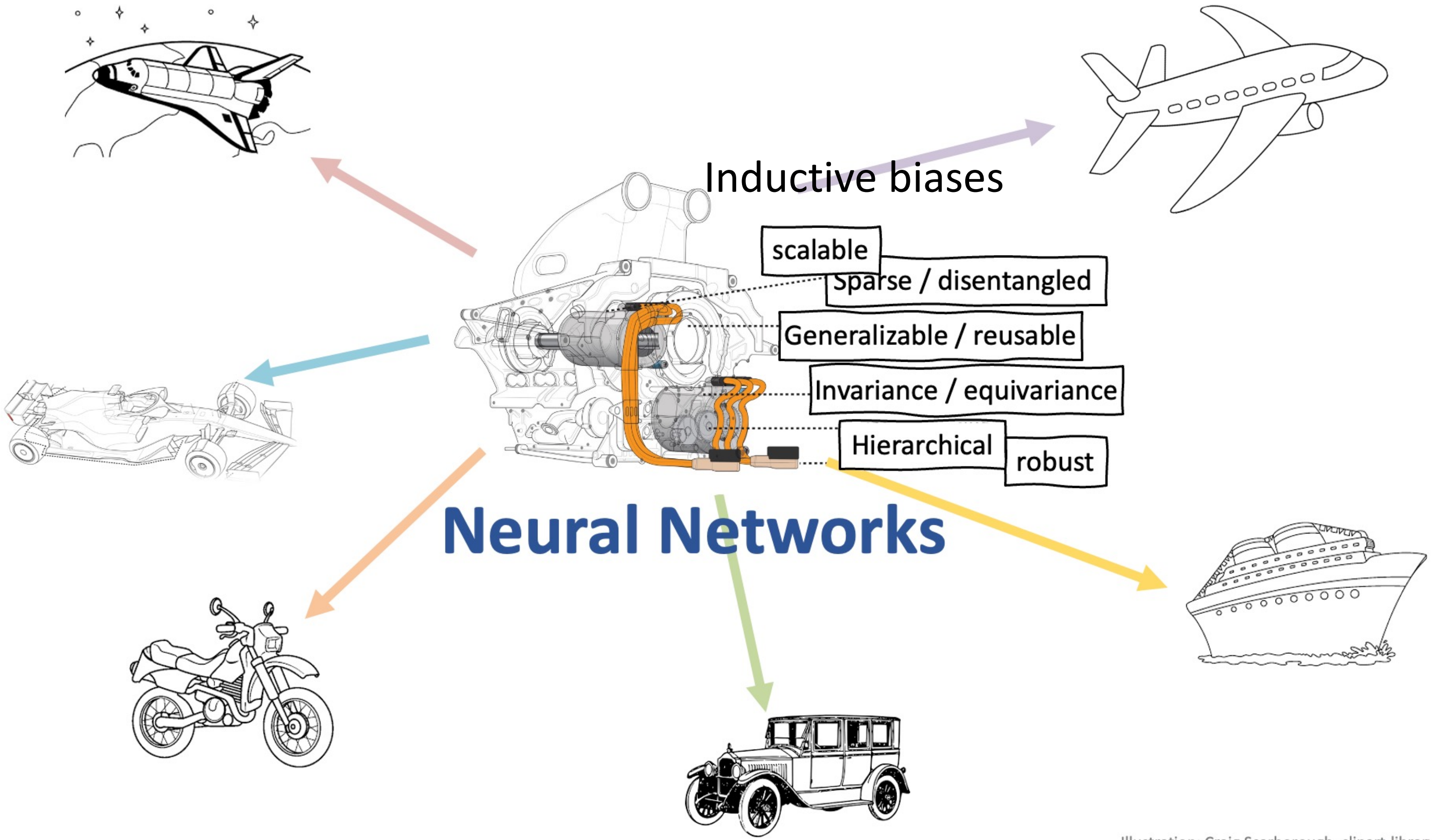
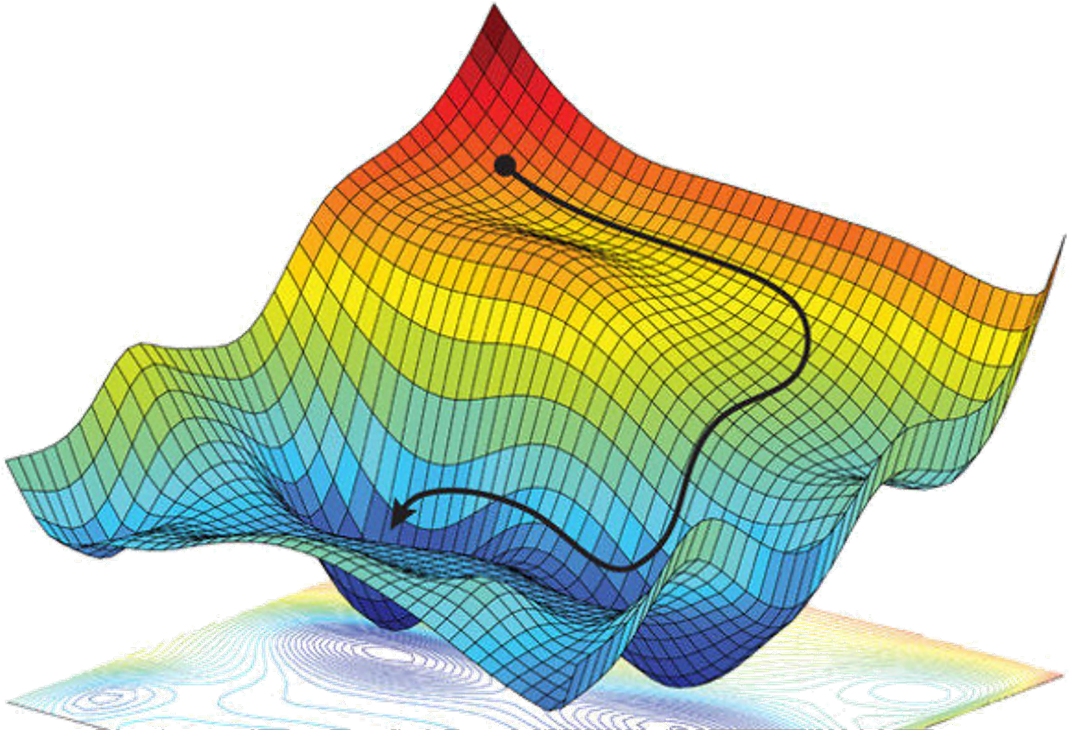


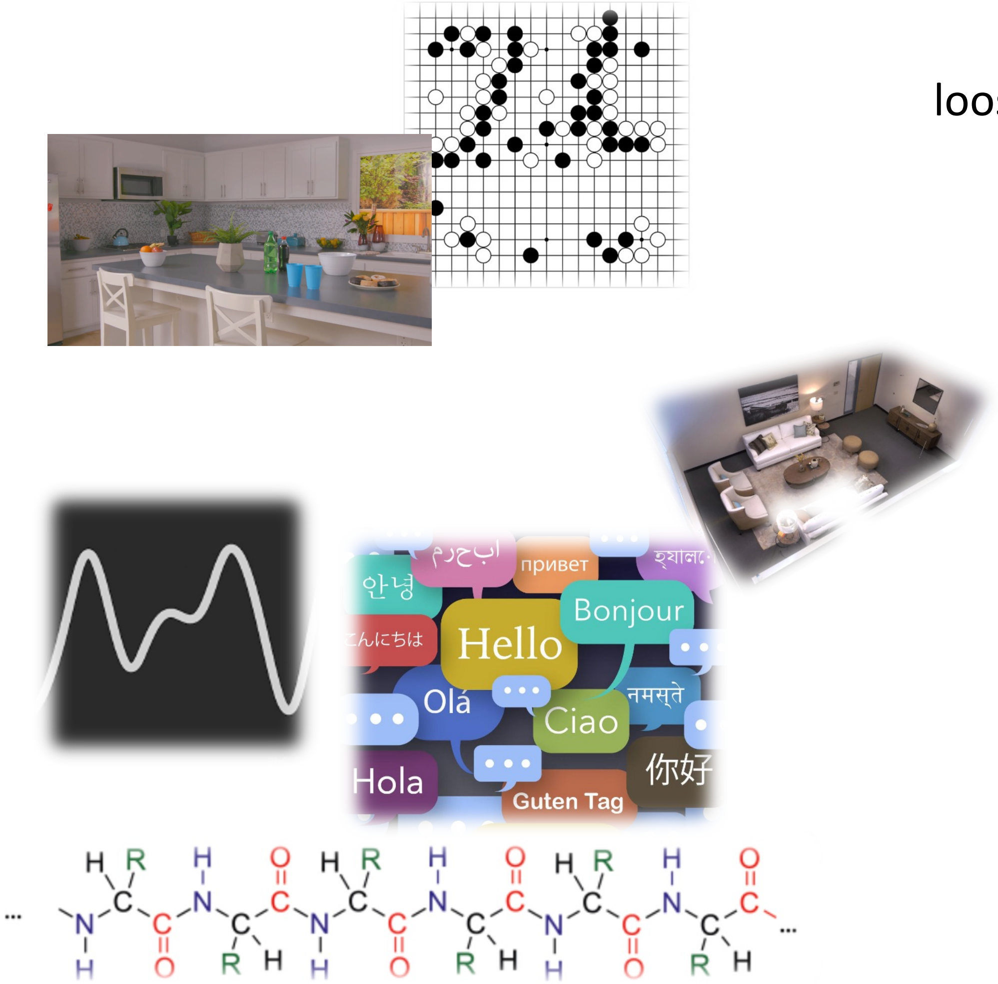

Illustration: Craig Scarborough, clipart-library.com

# Objective: how to train?



Amini, Alexander & Amini, Ava & Karaman, Sertac & Rus, Daniela. (2018). Spatial Uncertainty Sampling for End-to-End Control.

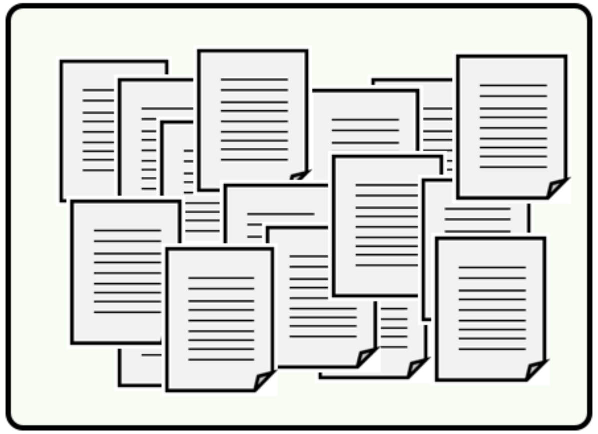
# Data: where to train?




huge diverse corpus

general domain

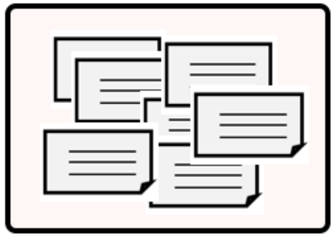
loosely organized



not labeled

not huge, or not diverse, or both

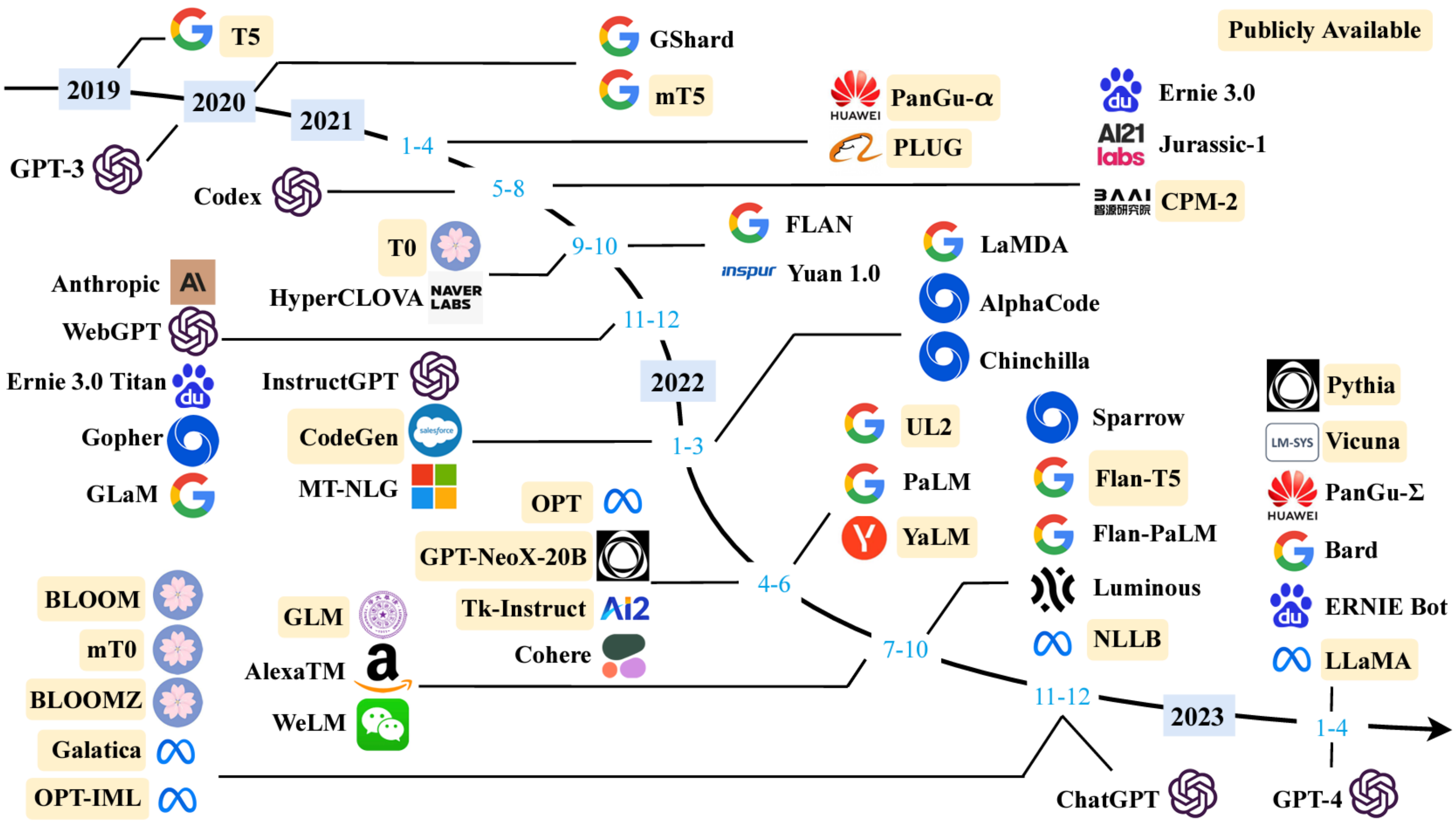
carefully curated



task-specific domain

task-specific annotations

# Large pre-trained language models



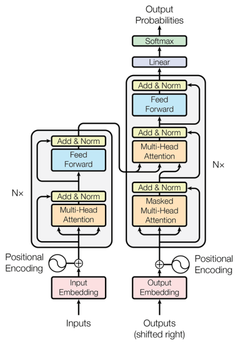

Zhao, Wayne Xin, et al. "A Survey of Large Language Models." *arXiv preprint arXiv:2303.18223* (2023).

# Large pre-trained language models

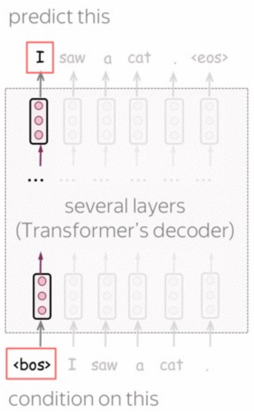
**Data:**

Common Crawl
arXiv
GitHub
Wikipedia
The Free Encyclopedia
StackExchange

**Architecture:**




**Objective:**

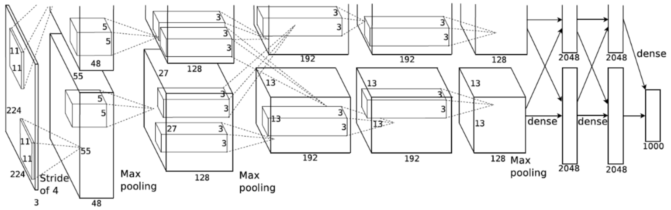


# Visual Pre-training has a longer history…

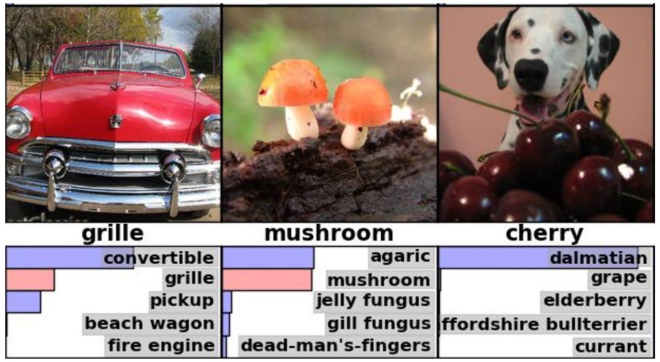
**D**ata:

[Deng et al., 2009]

**A**rchitecture:

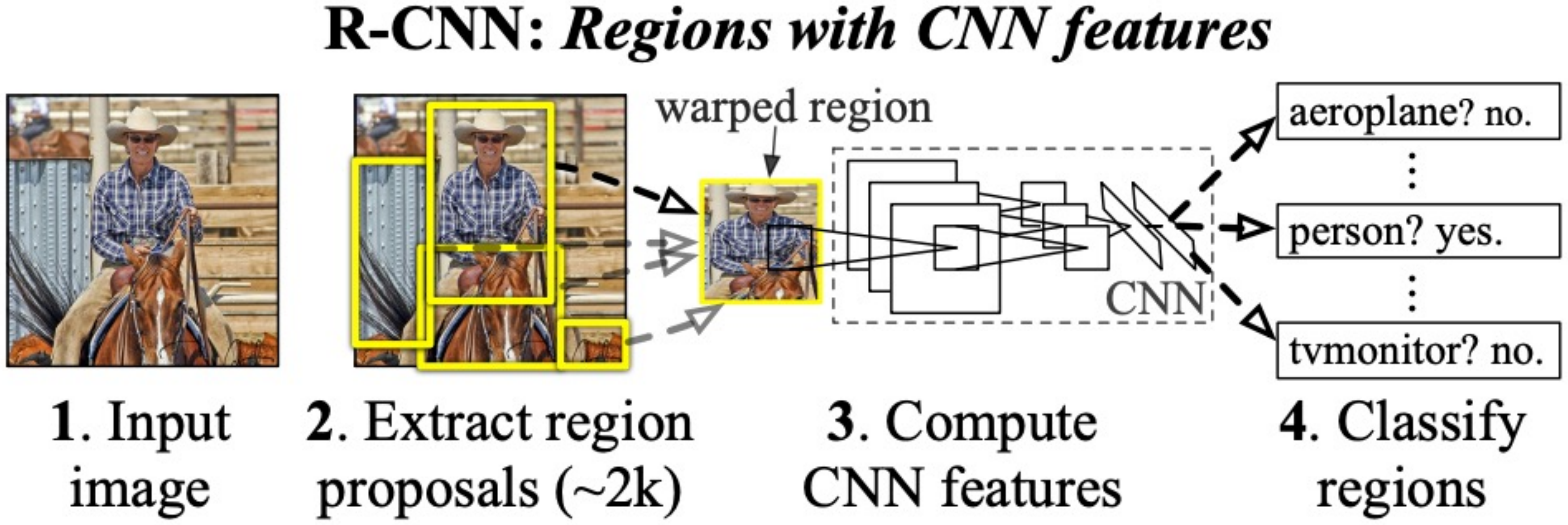



**O**bjective:



# Visual Pre-training has a longer history…

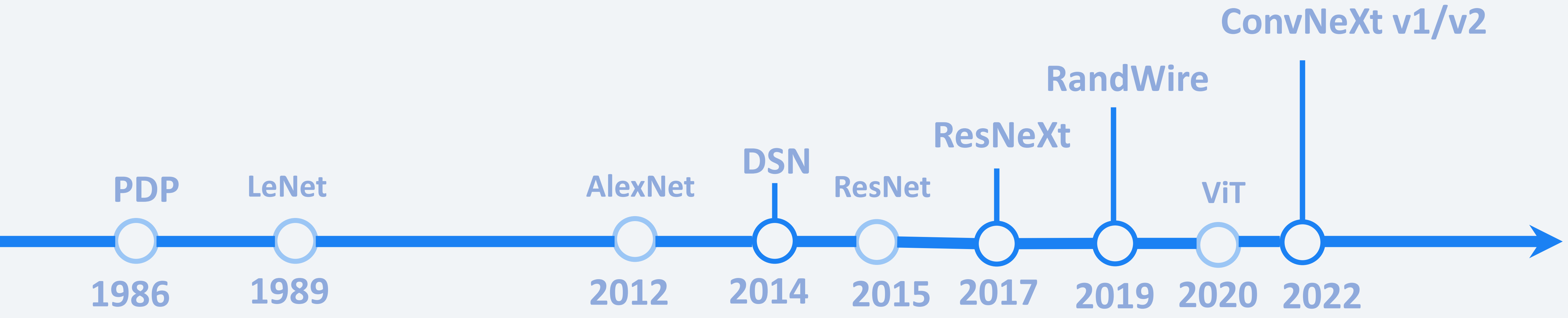
Arguably the first major success of “pre-training"…




Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

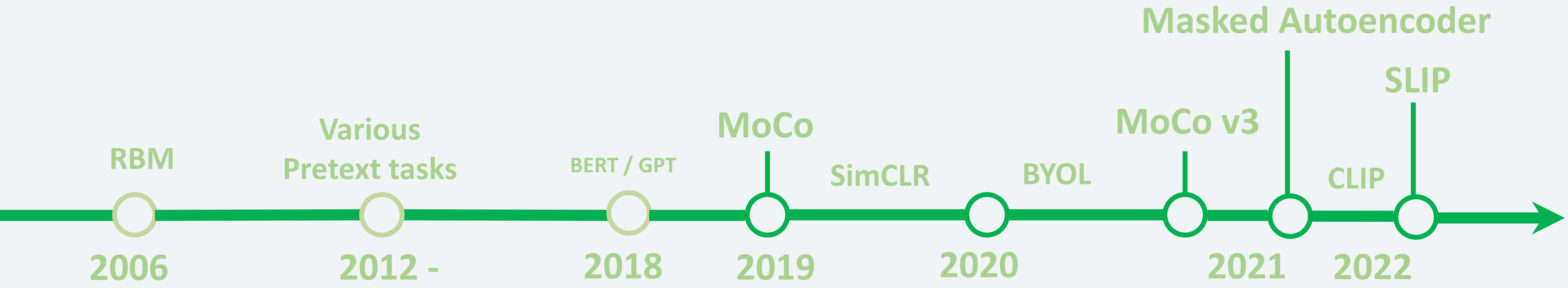
Architecture / Objective / Data

## 1. How to design neural network architectures



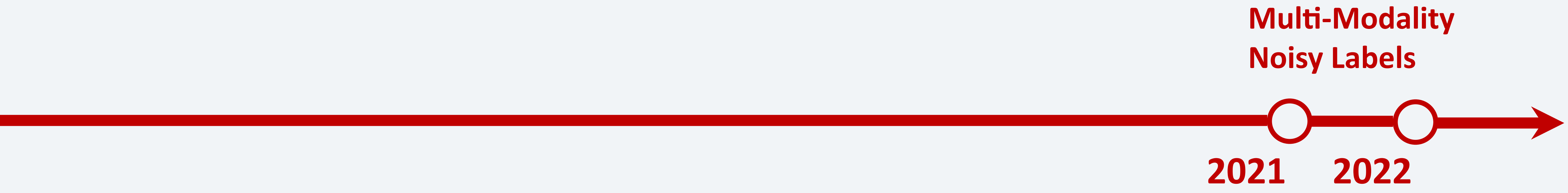
Architecture / Objective / Data

## 2. Training objectives beyond supervised classification: Are labels necessary?



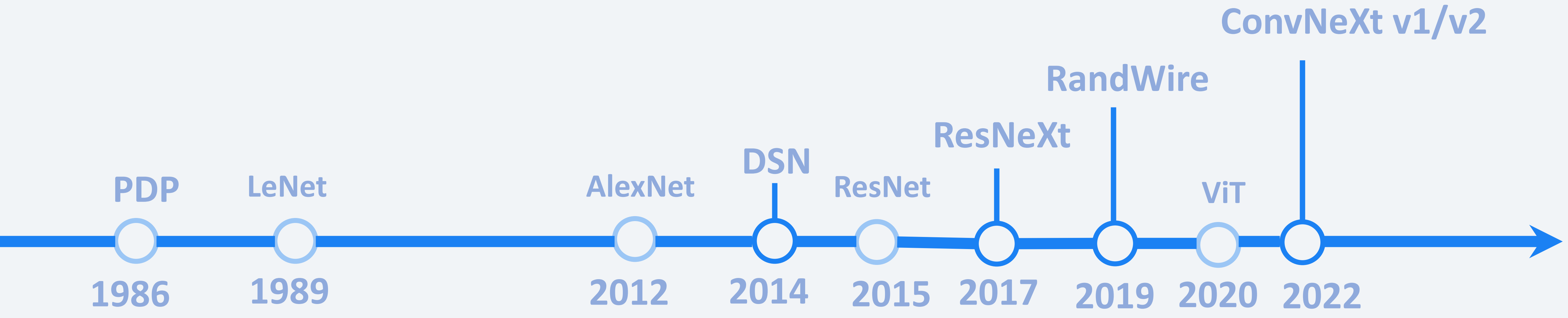
Architecture / Objective / Data

# 3. What data to use for visual pre-training



Architecture / Objective / Data

# 1. How to design neural network architectures



# Connectionism

[A general framework for parallel distributed processing. Rumelhart et al., 1986]

**PARALLEL DISTRIBUTED PROCESSING**

**Explorations in the Microstructure of Cognition**

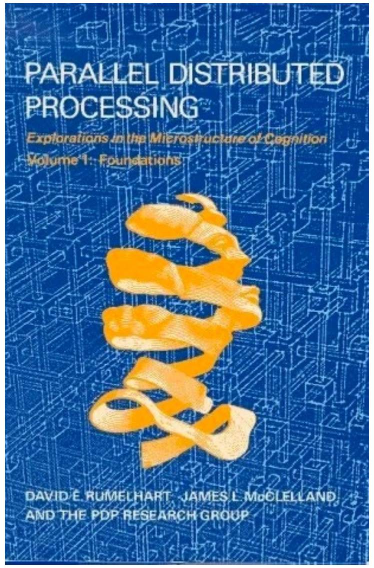
**Volume 1: Foundations**

David E. Rumelhart James L. McClelland
and the PDP Research Group

Chisato Asanuma
Francis H. C. Crick
Jeffrey L. Elman
Geoffrey E. Hinton
Michael I. Jordan
Alan H. Kawamoto
Paul W. Munro
Donald A. Norman
Daniel E. Rabin
Terrence J. Sejnowski
Paul Smolensky
Gregory O. Stone
Ronald J. Williams
David Zipser

Institute for Cognitive Science
University of California, San Diego

**(PDP group is now at** Stanford**)**




- A *set of processing units*
- A *state of activation*
- An *output function* for each unit
- A *pattern of connectivity* among units
- A *propagation rule* for propagating patterns of activities through the network of connectivities
- An *activation rule* for combining the inputs impinging on a unit with the current state of that unit to produce a new level of activation for the unit.
- A *learning rule* whereby patterns of connectivity are modified by experience
- An *environment* within which the system must operate

1986

# Convolutional Neural Networks

[Learning Internal Representations by Error Propagation. Rumelhart et al., 1986]

ConvNet using BP

- Receptive field
- Translation equivariance
- Trained by error propagation

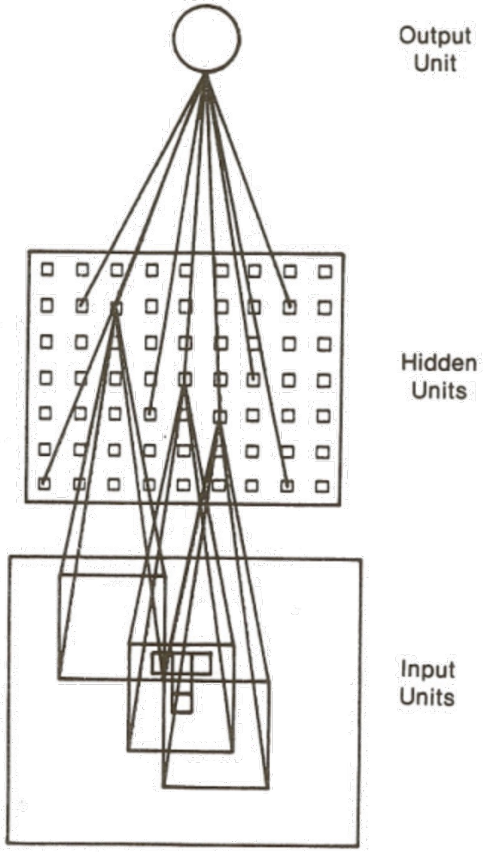
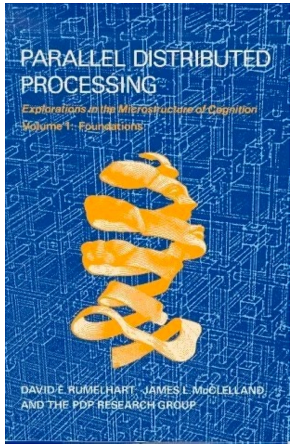



FIGURE 14. The network for solving the T-C problem. See text for explanation.




**PARALLEL DISTRIBUTED PROCESSING**

**Explorations in the Microstructure of Cognition**

**Volume 1: Foundations**

David E. Rumelhart James L. McClelland
and the PDP Research Group

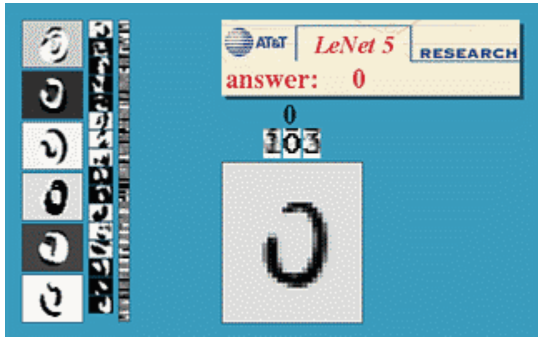
Chisato Asanuma, Francis H. C. Crick, Jeffrey L. Elman, Geoffrey E. Hinton, Michael I. Jordan, Alan H. Kawamoto, Paul W. Munro, Donald A. Norman, Daniel E. Rabin, Terrence J. Sejnowski, Paul Smolensky, Gregory O. Stone, Ronald J. Williams, David Zipser

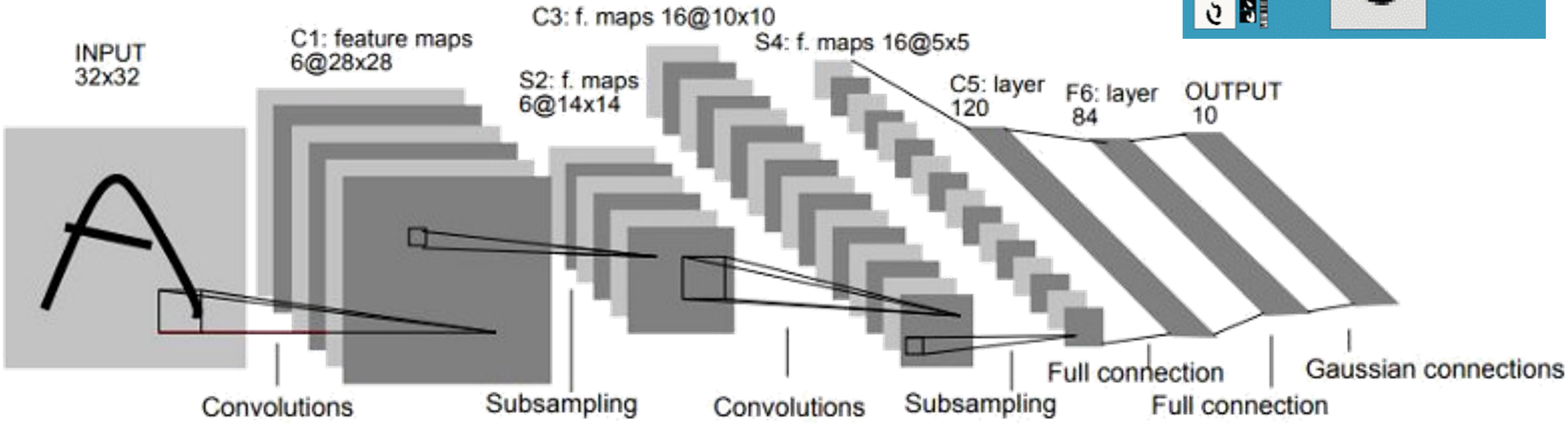
Institute for Cognitive Science
University of California, San Diego

**1986**

# LeNet

[Backpropagation Applied to Handwritten Zip Code Recognition, LeCun et al., 1989]



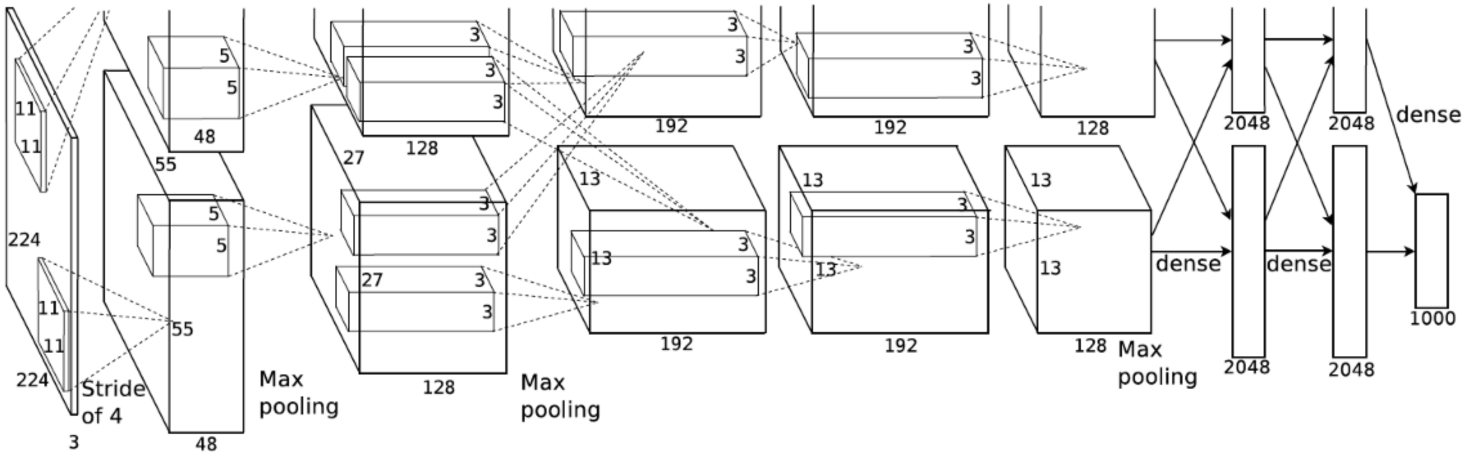




**LeNet**

**1989**

# AlexNet

[Krizhevsky, Sutskever and Hinton, 2012]



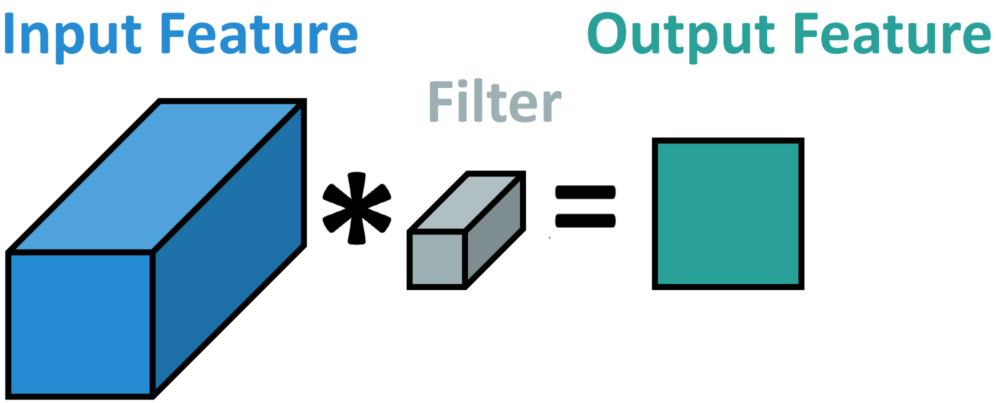


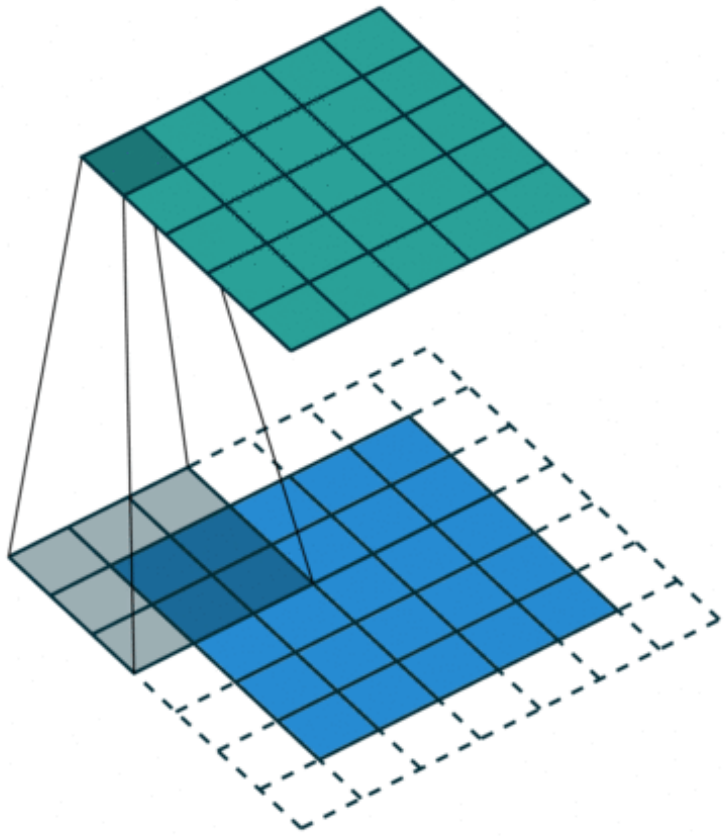
[Deng et al., 2009]
[Russakovsky et al., 2015]

**AlexNet**

**2012**

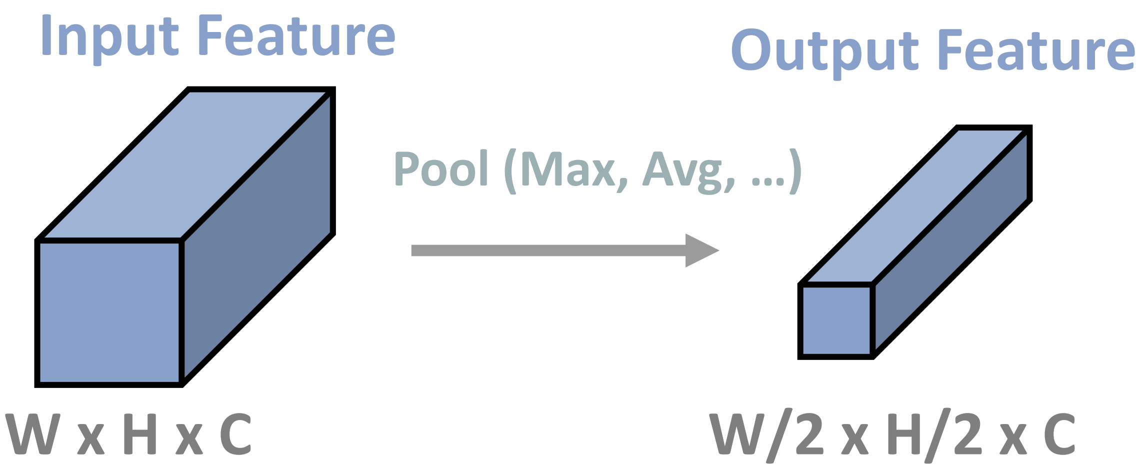
# ConvNet Basics: Convolution

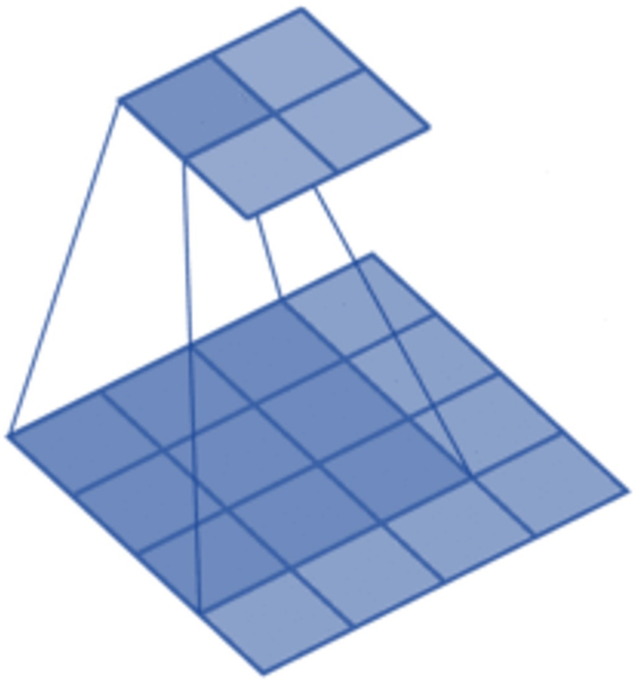


- Locally-connected
- Spatial weight-sharing
- Translation equivariance

# ConvNet Basics: Pooling

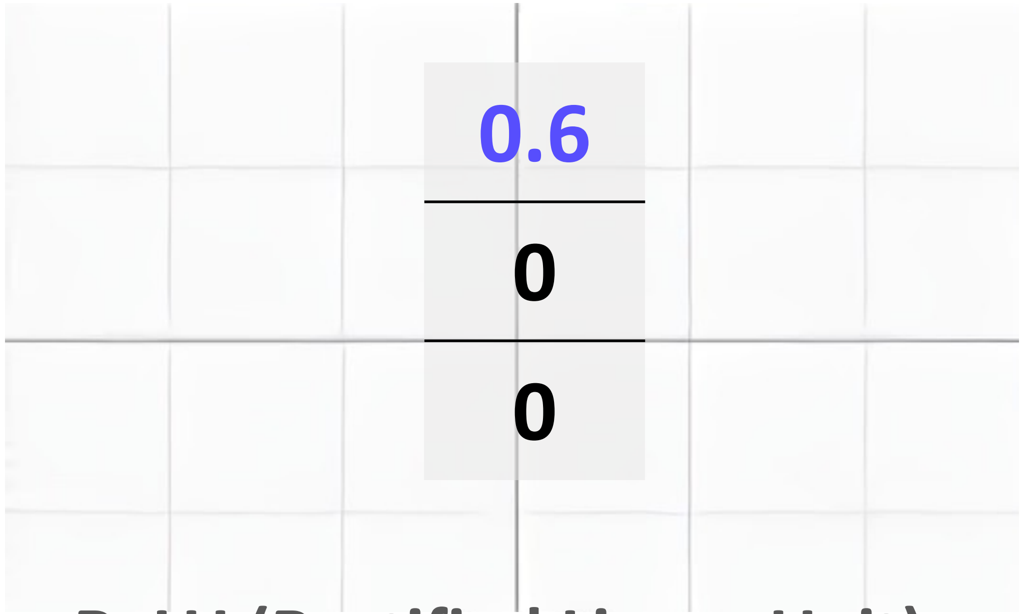


- Invariant to small translations
- Reduce spatial resolutions

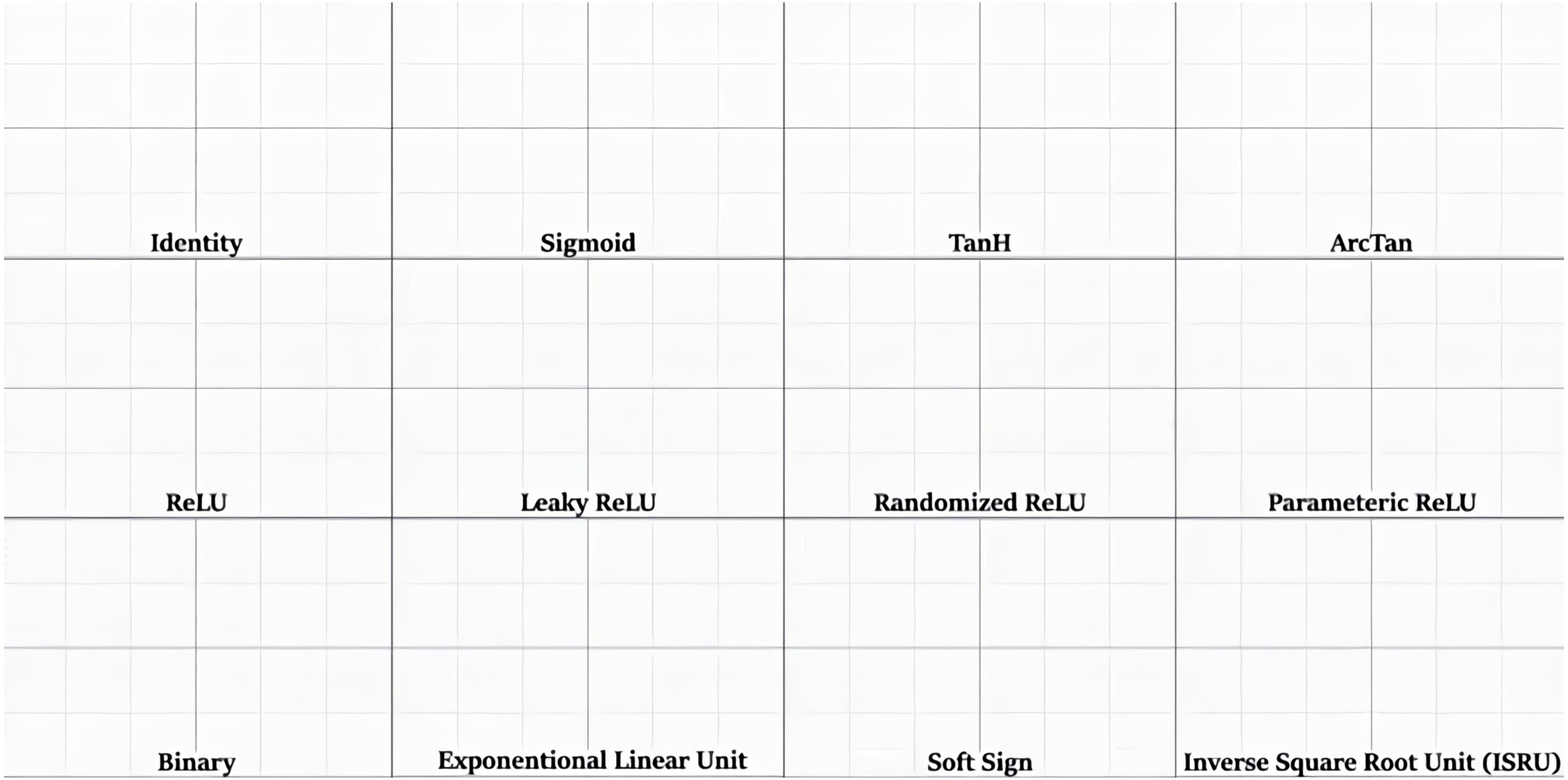
# ConvNet Basics: Activation Function

| | |
|---|---|
| 0.6 | 0.6 |
| -0.2 | 0 |
| -0.7 | 0 |

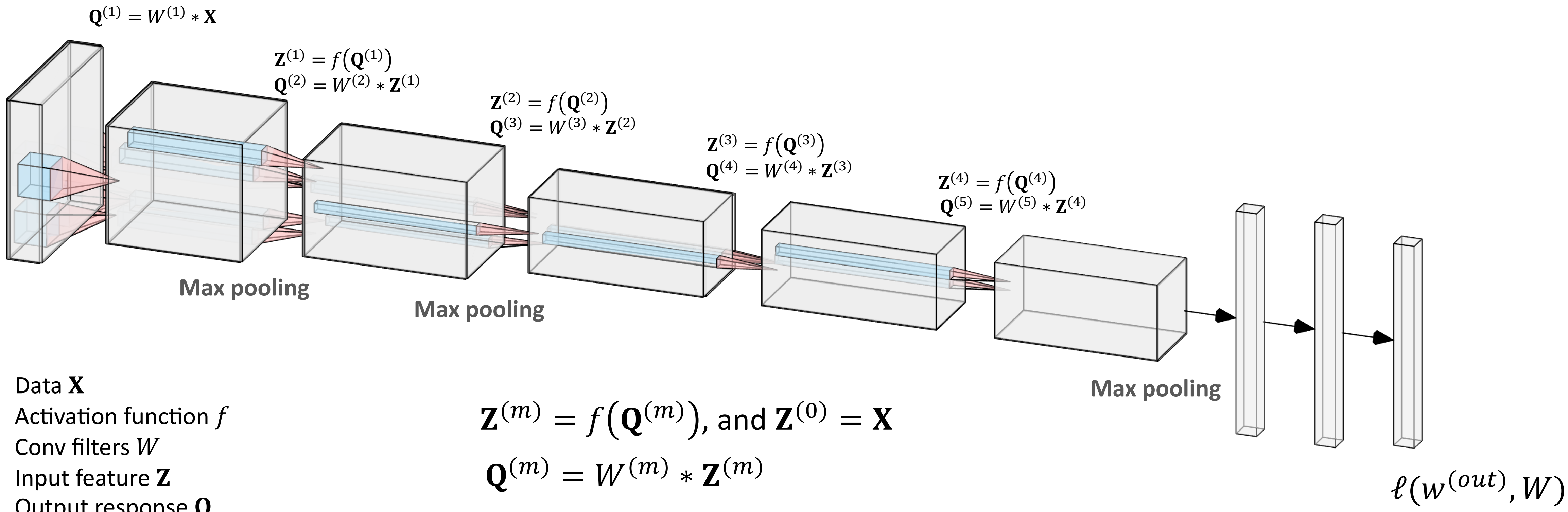


ReLU (Rectified Linear Unit)

# ConvNet Basics: Activation Function



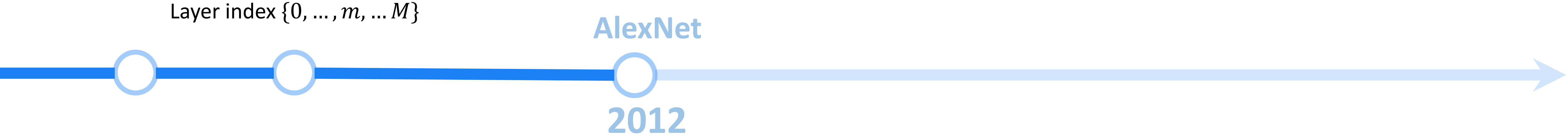
# ConvNet Basics: Stacking layers




Data $\mathbf{X}$
Activation function $f$
Conv filters $W$
Input feature $\mathbf{Z}$
Output response $\mathbf{Q}$
Layer index $\{0, \ldots, m, \ldots M\}$

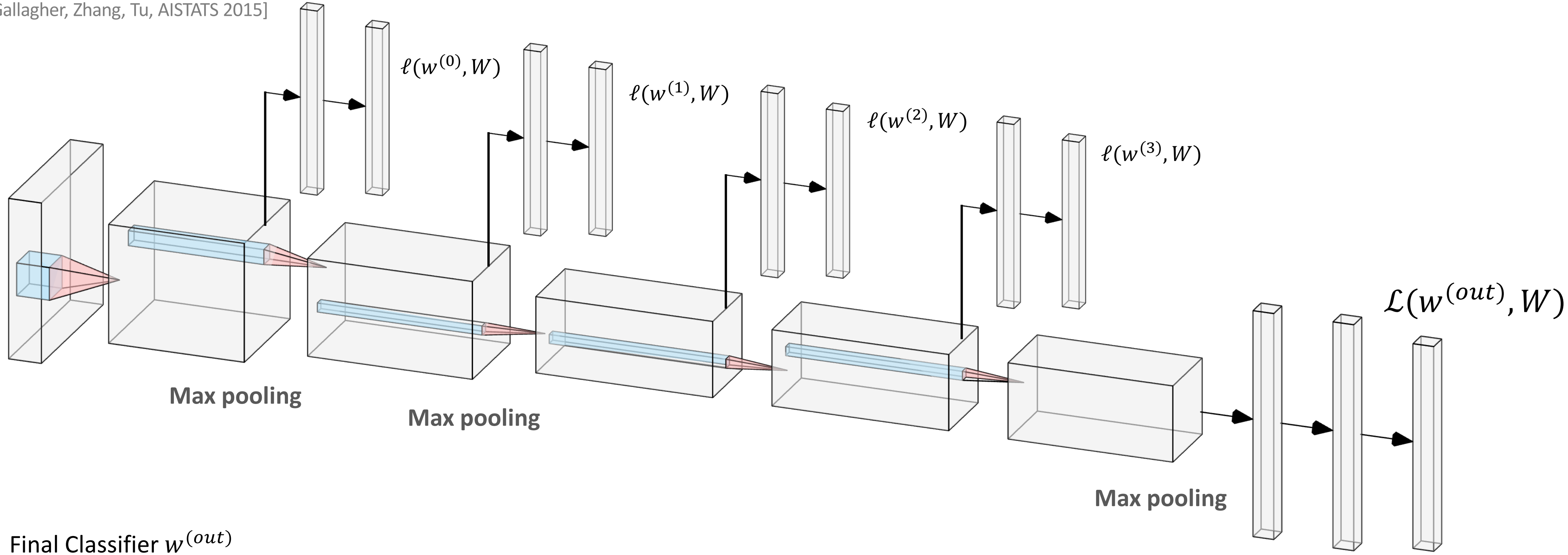
$$\mathbf{Z}^{(m)} = f(\mathbf{Q}^{(m)}), \text{ and } \mathbf{Z}^{(0)} = \mathbf{X}$$

$$\mathbf{Q}^{(m)} = W^{(m)} * \mathbf{Z}^{(m)}$$



# Deeply-supervised Nets

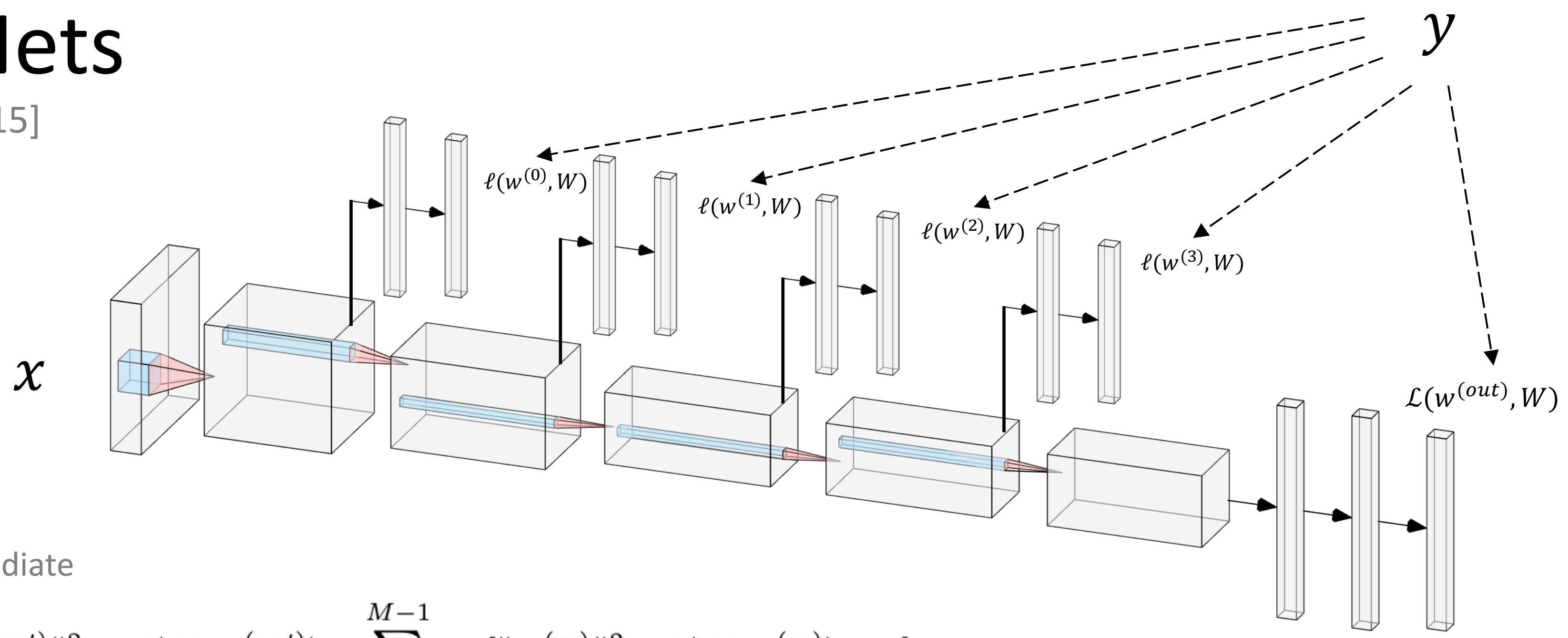
[Lee*, **Xie***, Gallagher, Zhang, Tu, AISTATS 2015]




Final Classifier $w^{(out)}$
Intermediate Classifier $w^{(m)}$

# Deeply-supervised Nets

[Lee*, **Xie***, Gallagher, Zhang, Tu, AISTATS 2015]




Using hinge loss (SVM) for both output and intermediate classifiers

*Final training objective* $$\|\mathbf{w}^{(out)}\|^2 + \mathcal{L}(\mathsf{W}, \mathbf{w}^{(out)}) + \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(\mathsf{W}, \mathbf{w}^{(m)}) - \gamma]_+,$$

where

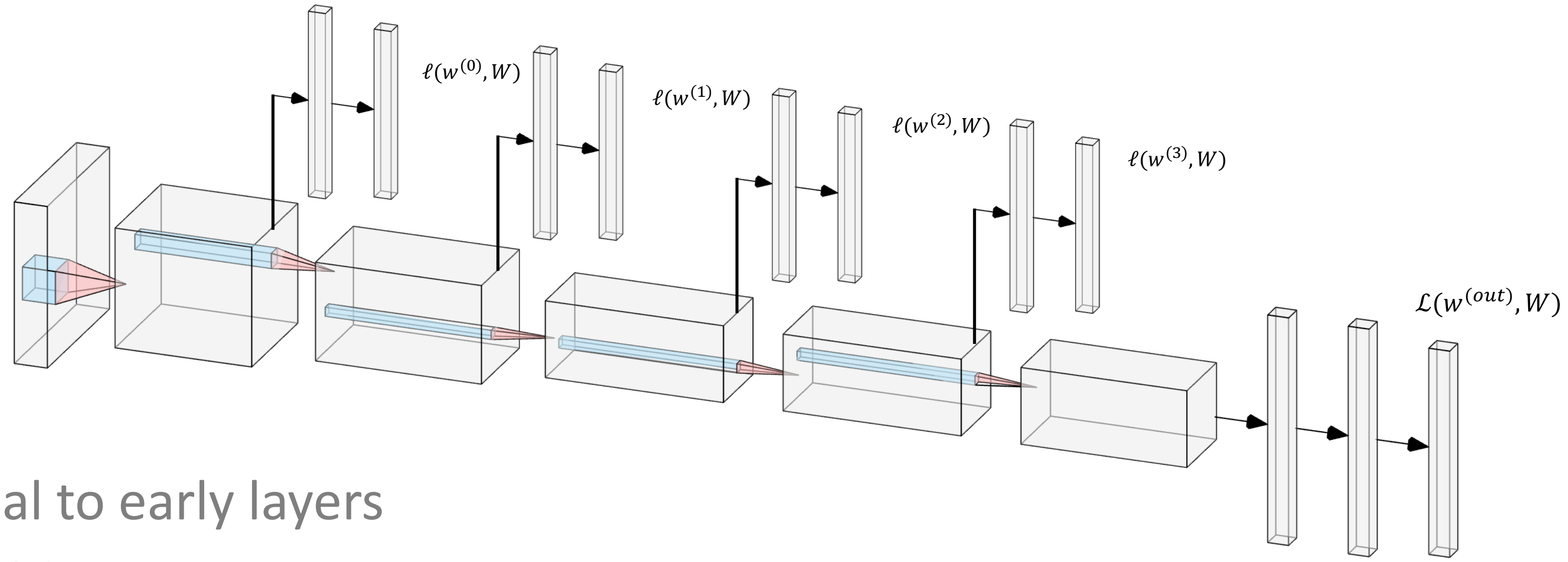
$$\mathcal{L}(\mathsf{W}, \mathbf{w}^{(out)}) = \sum_{y_k \neq y} [1 - < \mathbf{w}^{(out)}, \phi(\mathbf{Z}^{(M)}, y) - \phi(\mathbf{Z}^{(M)}, y_k) >]_+^2$$ *Overall loss*

and

$$\ell(\mathsf{W}, \mathbf{w}^{(m)}) = \sum_{y_k \neq y} [1 - < \mathbf{w}^{(m)}, \phi(\mathbf{Z}^{(m)}, y) - \phi(\mathbf{Z}^{(m)}, y_k) >]_+^2$$ *Companion losses*

# Deeply-supervised Nets

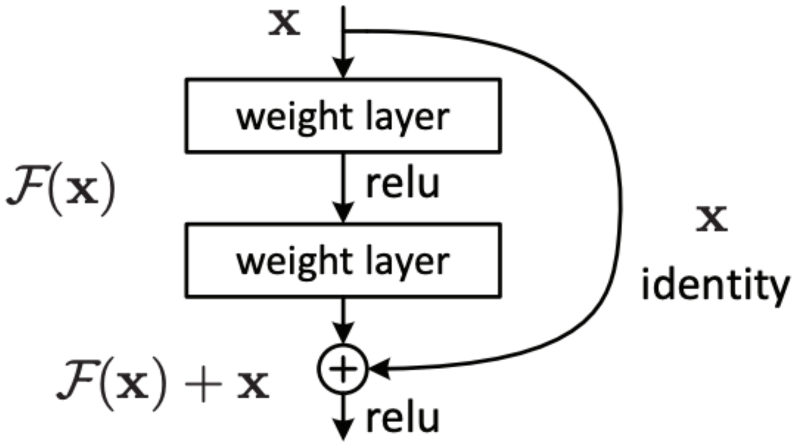
[Lee*, **Xie***, Gallagher, Zhang, Tu, AISTATS 2015]




- Exposing training signal to early layers
  - Better regularization
  - Better convergence
- Popularized the idea of deep supervision (2000+ citations)

# ResNet

[He et al., 2015]




repeating motif: a residual block

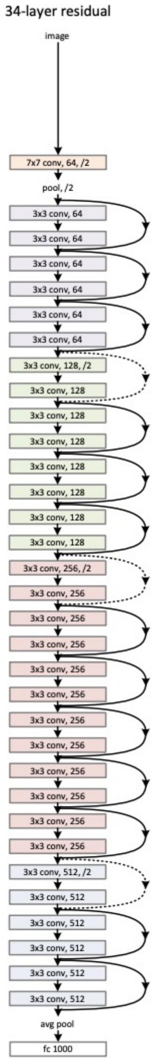
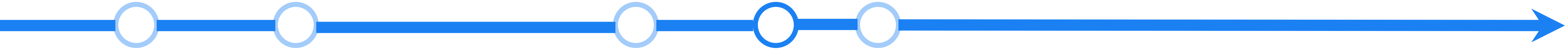




figure: reddit?

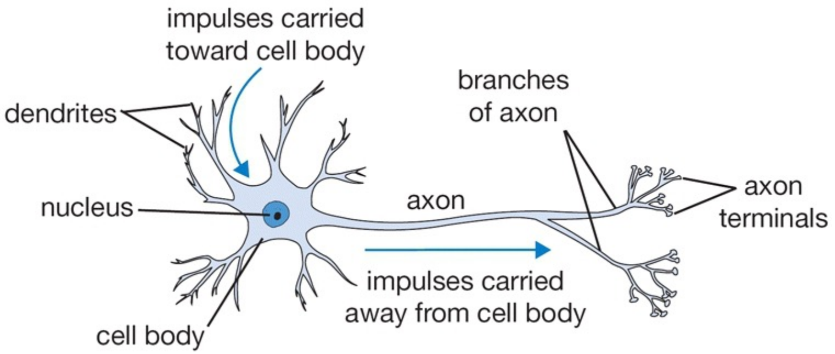


ResNet

2015

# ResNeXt

[**Xie**, Girshick, Dollár, Tu, He, CVPR 2017]



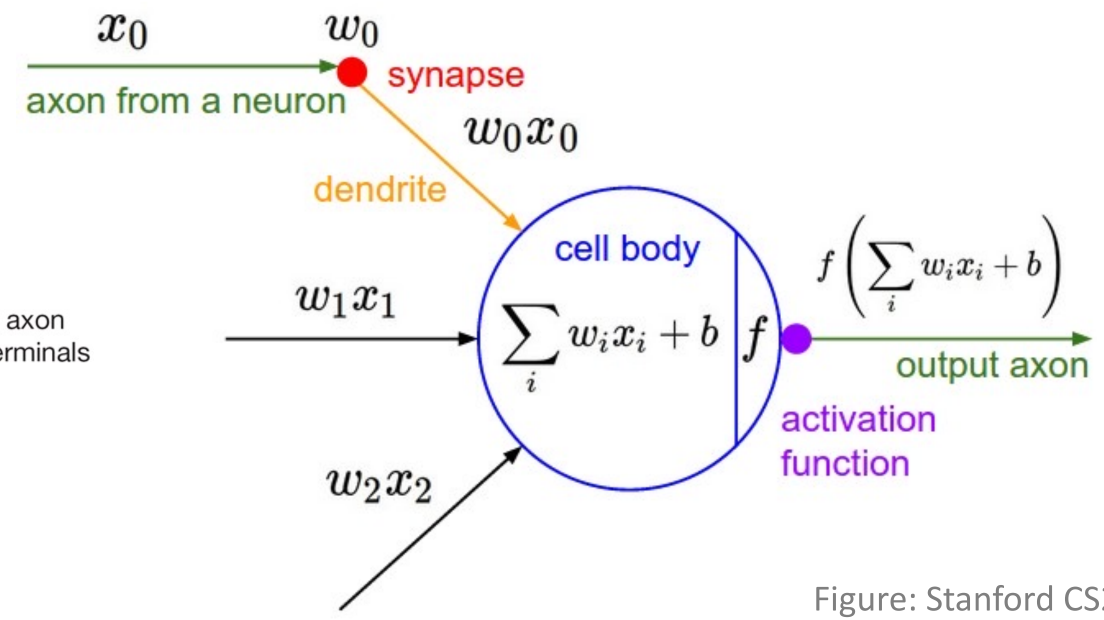
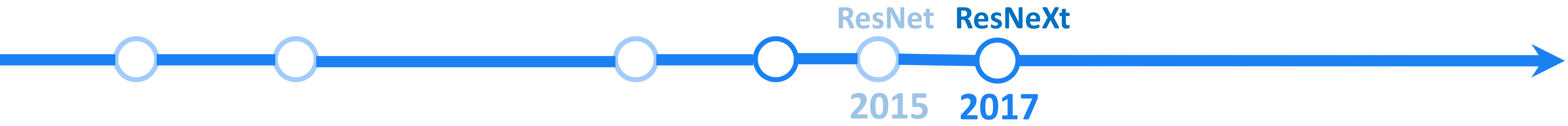




Figure: Stanford CS231n

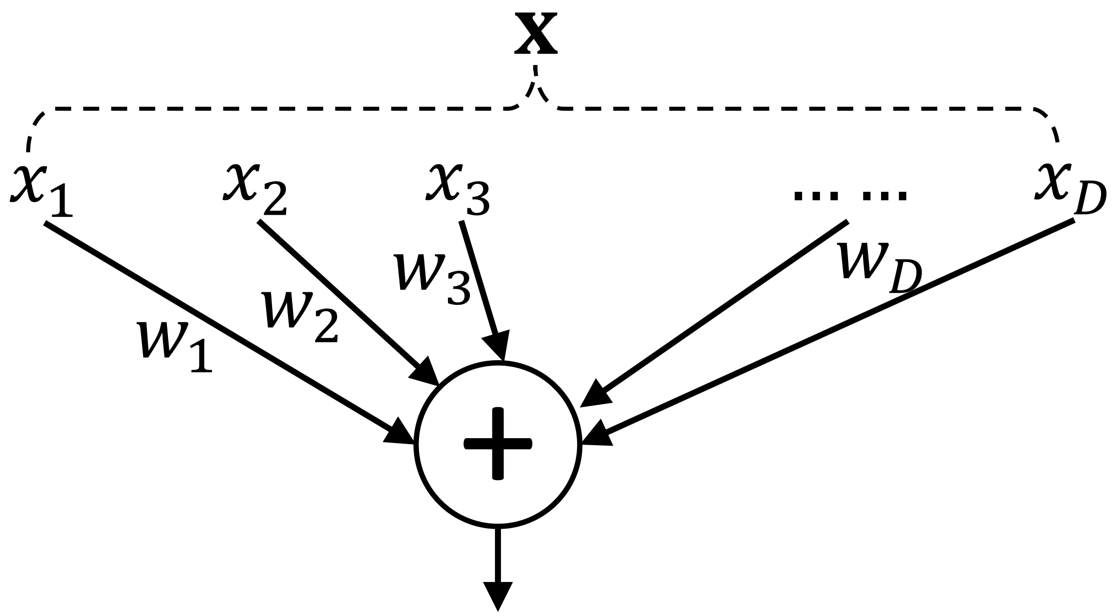


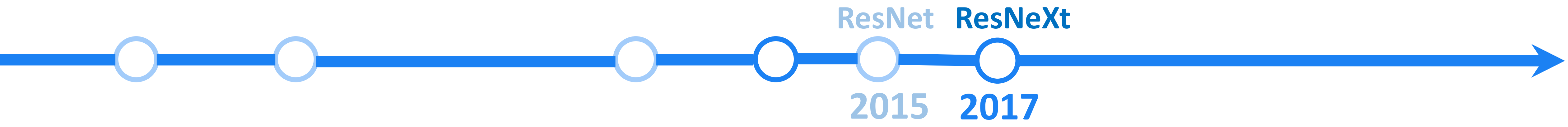
# ResNeXt

[**Xie**, Girshick, Dollár, Tu, He, CVPR 2017]

Analysis of a simple neuron:

- Splitting: $\mathbf{x} \rightarrow x_i$
- Transforming: $w_i x_i$
- Aggregating: $\sum_{i=1}^{D} w_i x_i$



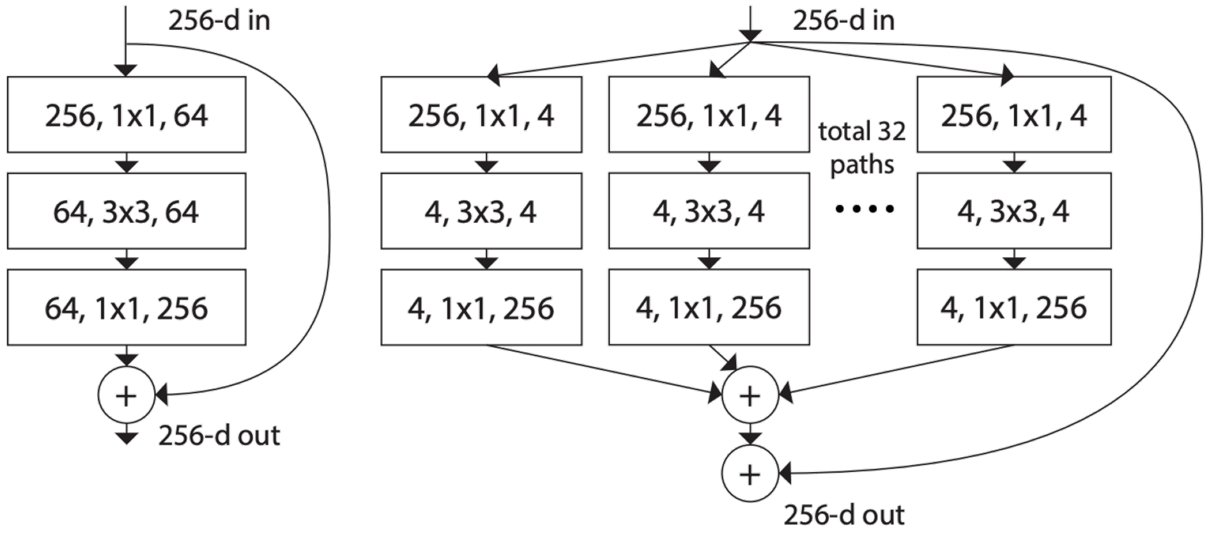



# ResNeXt

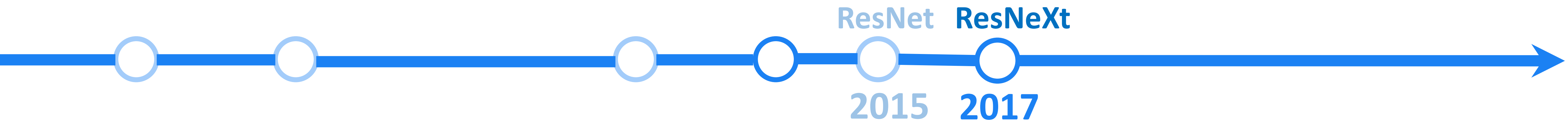
[**Xie**, Girshick, Dollár, Tu, He, CVPR 2017]

"Network-in-Neuron":

- Splitting: Parallel pathways
- Transforming: Conv block
- Aggregating: $\mathcal{F}(\mathbf{x}) = \sum_{i=1}^{C} \mathcal{T}_i(\mathbf{x})$




**ResNet**: Single Stream **ResNeXt**: Multiple pathways

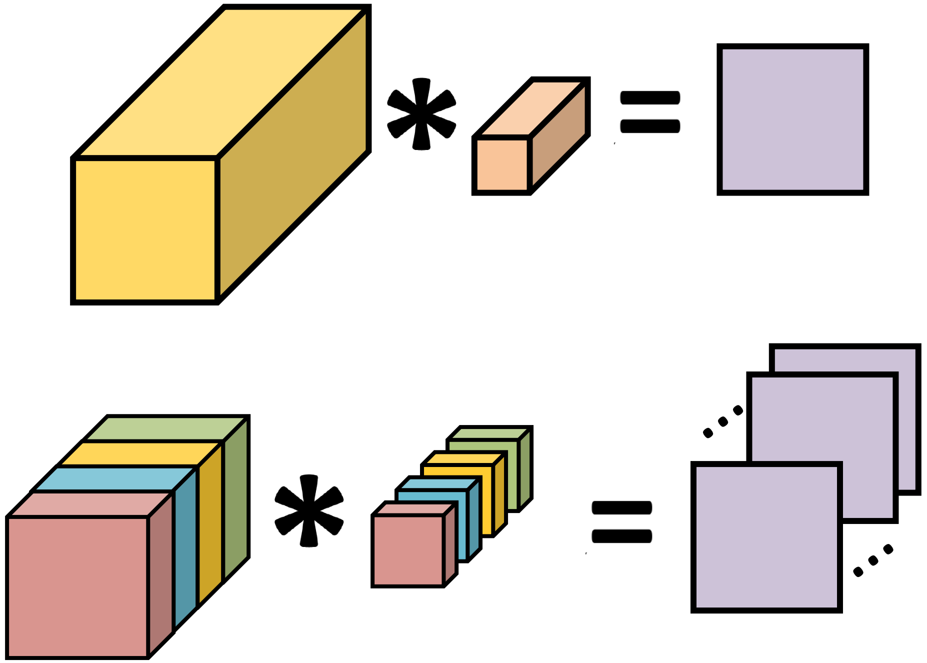


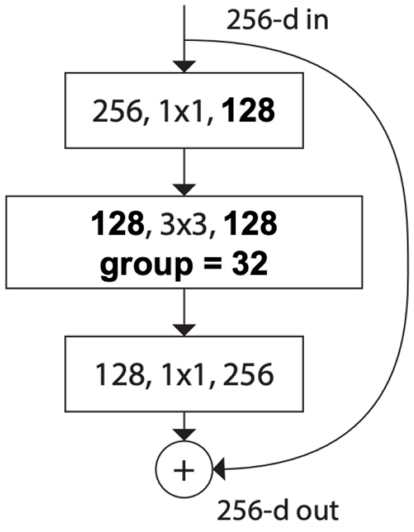
# ResNeXt

[**Xie**, Girshick, Dollár, Tu, He, CVPR 2017]

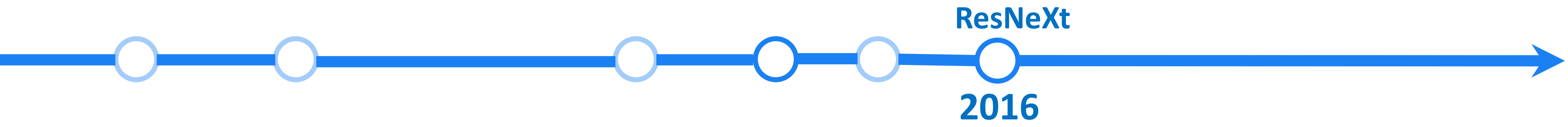
Dense Conv

Grouped Conv





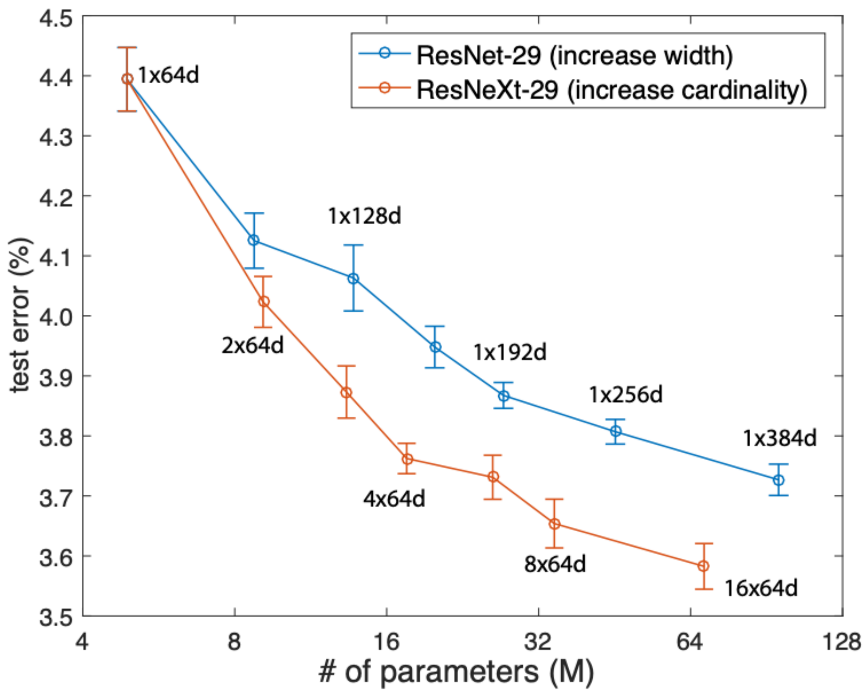

a ResNeXt block

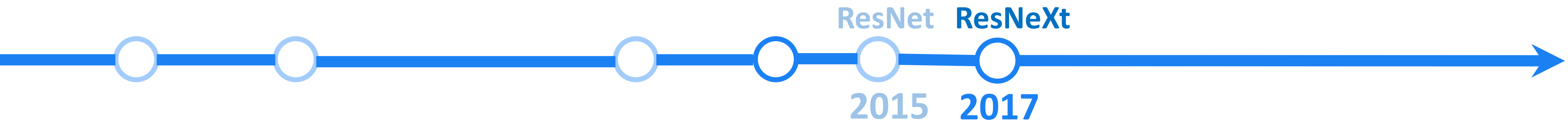


# ResNeXt

[**Xie**, Girshick, Dollár, Tu, He, CVPR 2017]

- Scaling neural networks:
  - Going deeper / wider
  - We identify the **NeXt** dimension: Cardinality (# number of groups)
- Increasing cardinality:
  - Better capacity-accuracy trade-off
    - Wider and sparser
    - Implicit ensemble
    - Disentangled features

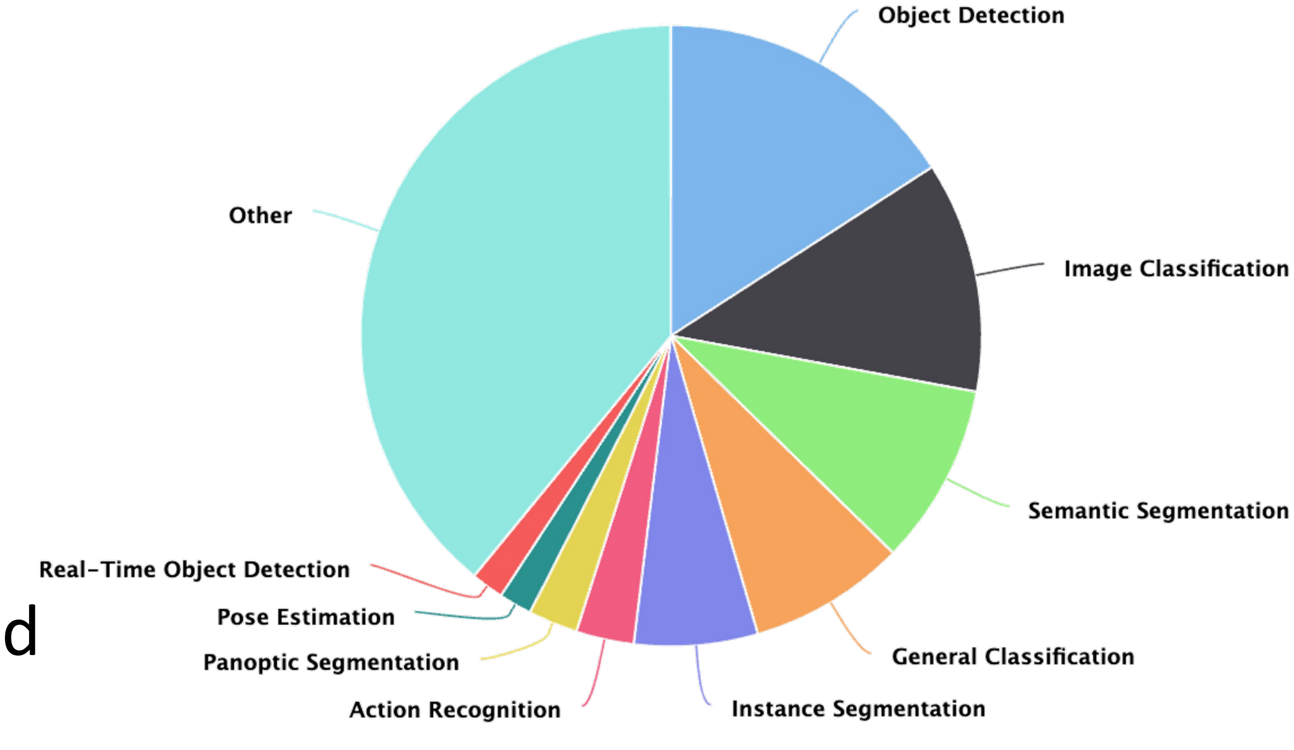


# ResNeXt

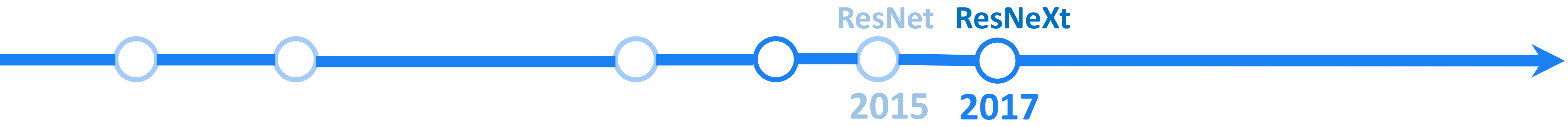
[**Xie**, Girshick, Dollár, Tu, He, CVPR 2017]

ResNeXt empowers many visual recognition tasks




Source: paperswithcode.com/method/resnext

- Better feature learning
- One of the most popular visual backbones
- Widely used in various academic and industrial applications



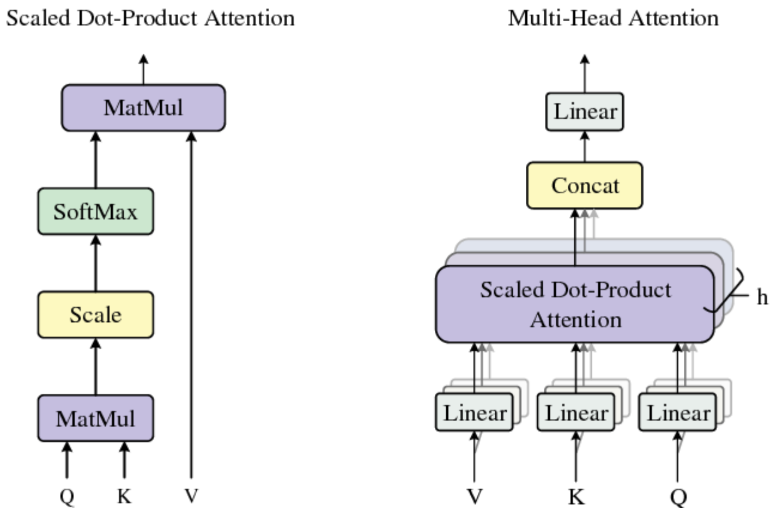
# Vision Transformer (ViT)

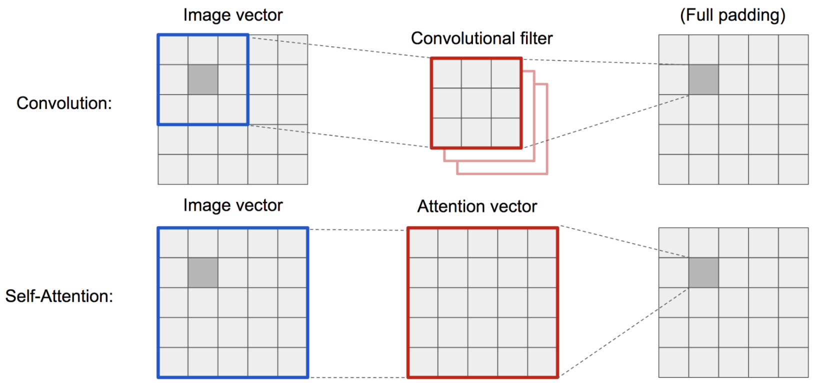
[Dosovitskiy et al., ICLR 2021]

- Self-attention:
  - Less inductive bias

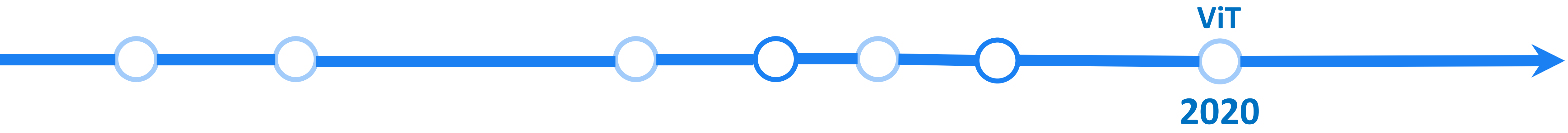
⇩

- Scalable
  - w/ bigger model
  - w/ larger data




courtesy: Lilian Weng



# Vision is NOT just about classification

- Expanding vision capabilities
  - Large resolution
  - Multi-scale

- Vision Transformer
  - Quadratic complexity
  - No hierarchy

# Advanced Vision Transformers

- Self-attention:
  - Less inductive bias

- Scalable
  - w/ bigger model
  - w/ larger data

Self-attention (ViT)

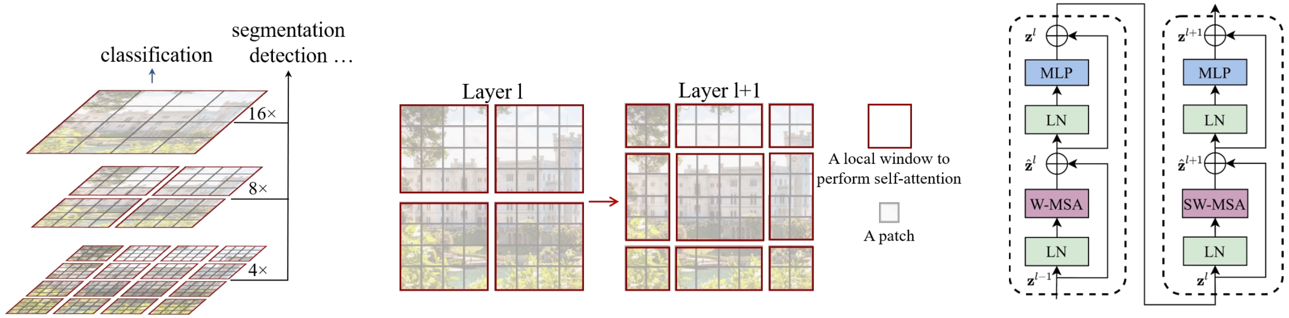
**+ Conv Priors**

**Translation equivariance**

**Locality**

**Hierarchical**

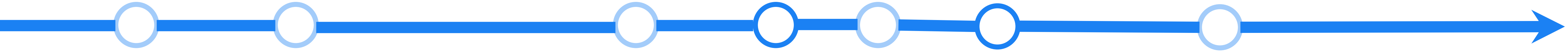
**More complicated designs + Specialized modules**




(a) Swin Transformer (b) Shifted Window (c) Two Successive Swin Transformer Blocks

[Liu et al., 2021]

**Swin Transformer**



**2021**

# ConvNet losing steam?

| Venue | Convolution, CNN, ConvNet | Attention, "-Former" |
|---|---|---|
| ECCV 2020 | 56 | 54 |
| CVPR 2021 | 49 | 78 |
| ICCV 2021 | 44 | 176 |
| CVPR 2022 | 44 | 263 |

Swin Transformer

State of the Art | Object Detection on COCO test-dev (using additional training data)

State of the Art | Instance Segmentation on COCO test-dev

State of the Art | Semantic Segmentation on ADE20K (using additional training data)

Ranked #4 | Action Classification on Kinetics-400 (using additional training data)

# Swin Transformers **is a *hybrid* architecture**

- Similarity:
  - Convolution inductive bias
- Difference:
  - “Core” component (attention vs. convolution)
  - Training procedures
  - Macro and micro architecture design decisions

- Common **assumption** in the 2020s:
  - **Self-attention** is the key for superior performance and scalability.
  - ConvNet is NOT a scalable architecture.

# A ConvNet for the 2020s

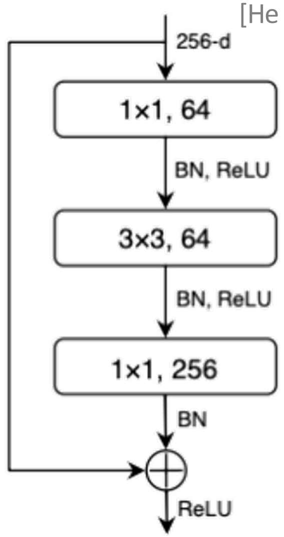
[Liu, Mao, Wu, Feichtenhofer, Darrell, **Xie.** CVPR 2022]

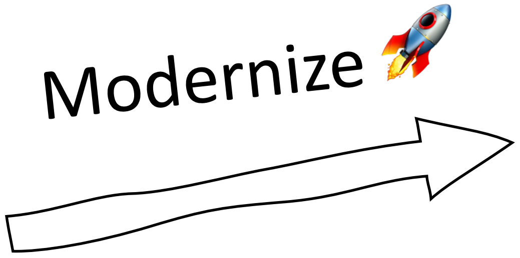
**Central question:**

- How do the design choices in Transformers impact a ConvNet's performance?

ResNet

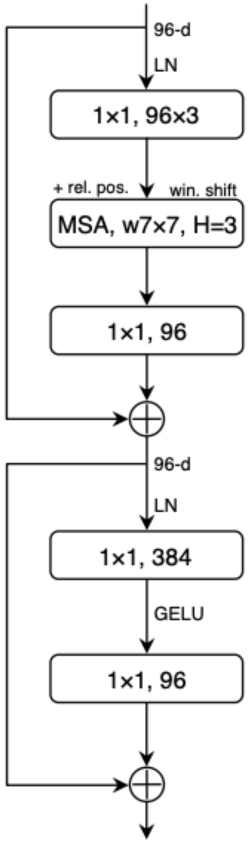
[He et al., 2015]

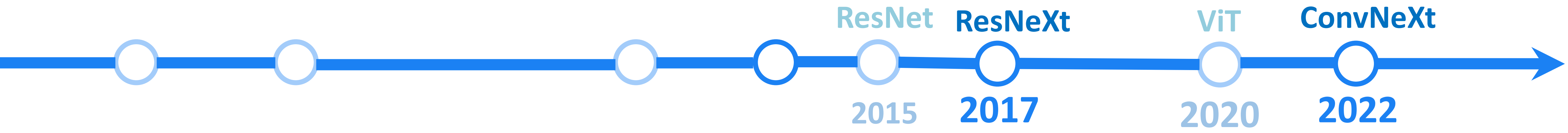



Hierarchical Vision Transformer

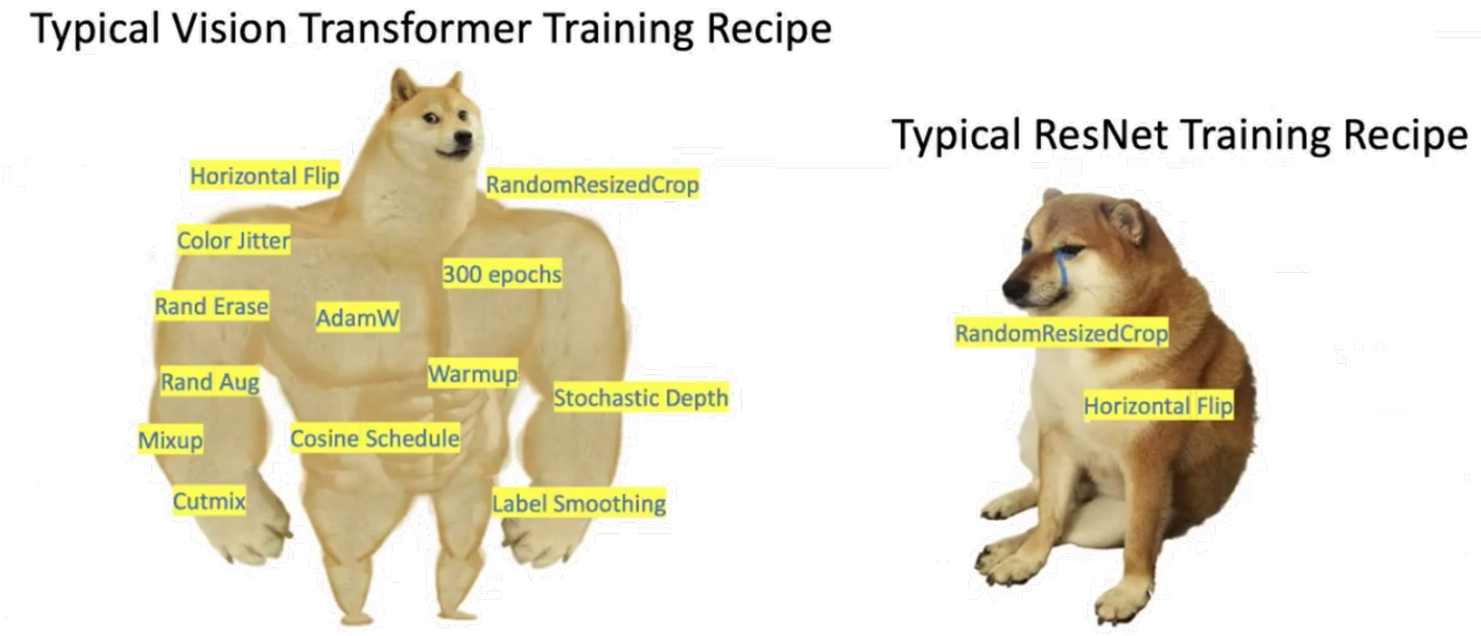
[Liu et al., 2021]



# Improved training recipe

("DeiT Recipe") [Touvron et al., 2021]



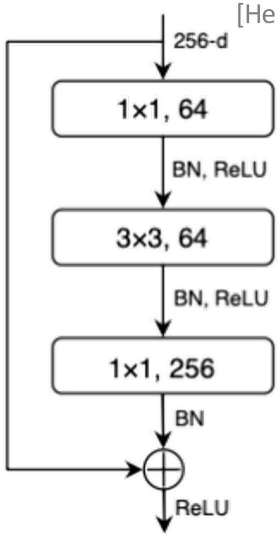

ResNet-50 ImageNet top-1: 76.7% -> 78.8% 🚀

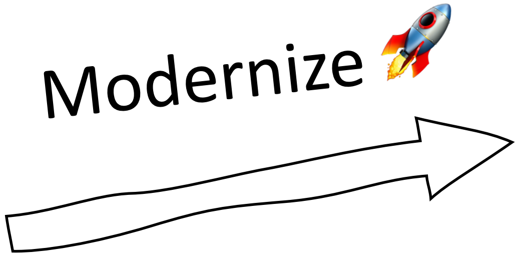
[Revisiting ResNets: Improved Training and Scaling Strategies, Bello et al, 2021]

[ResNet strikes back: An improved training procedure in timm Wightman, et al, 2021]

ResNet

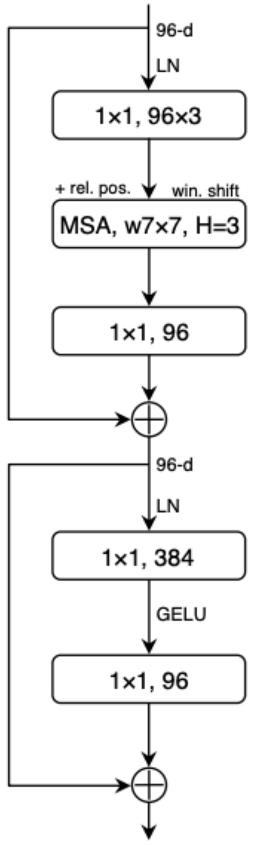
[He et al., 2015]

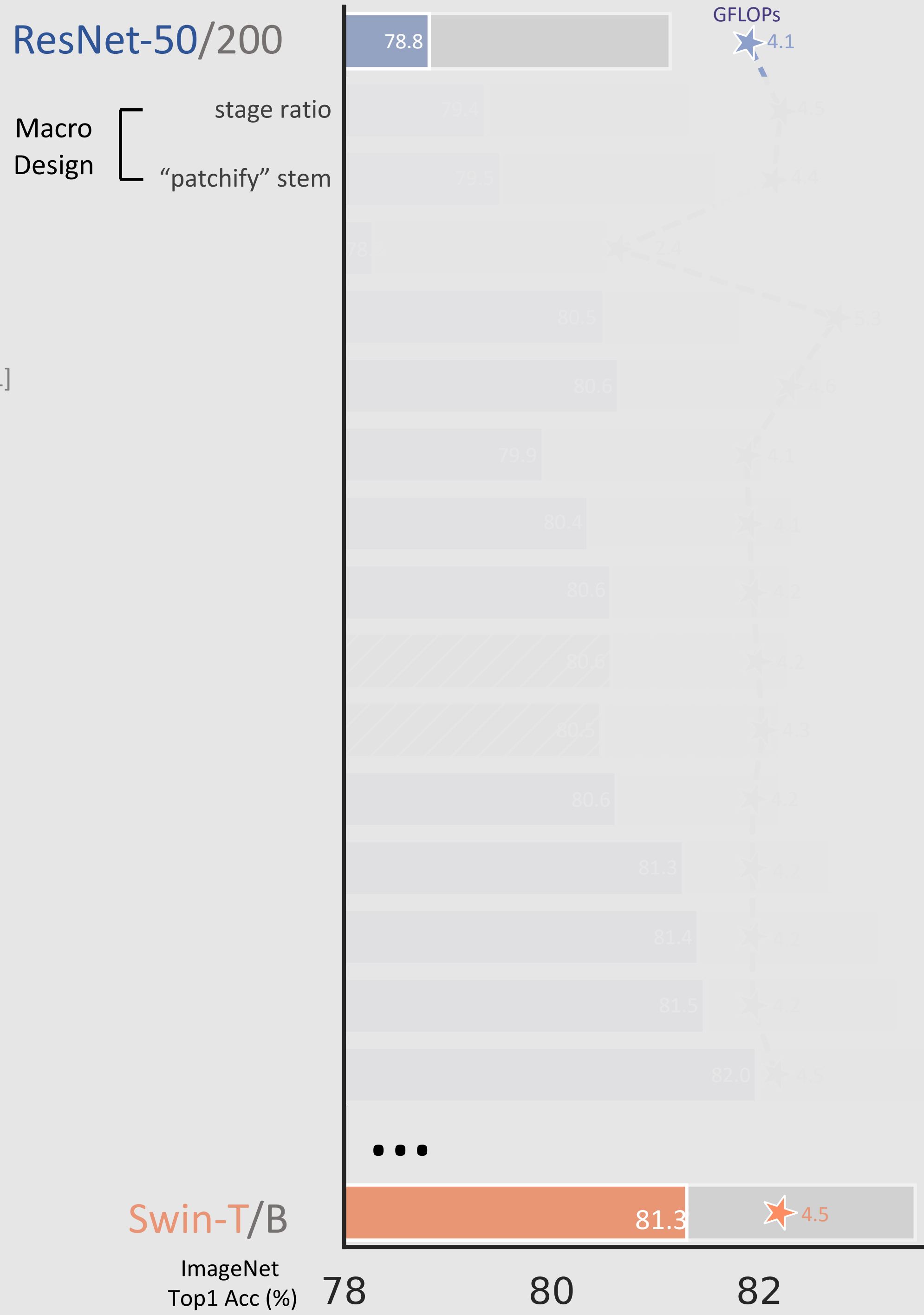



SwinTransformer

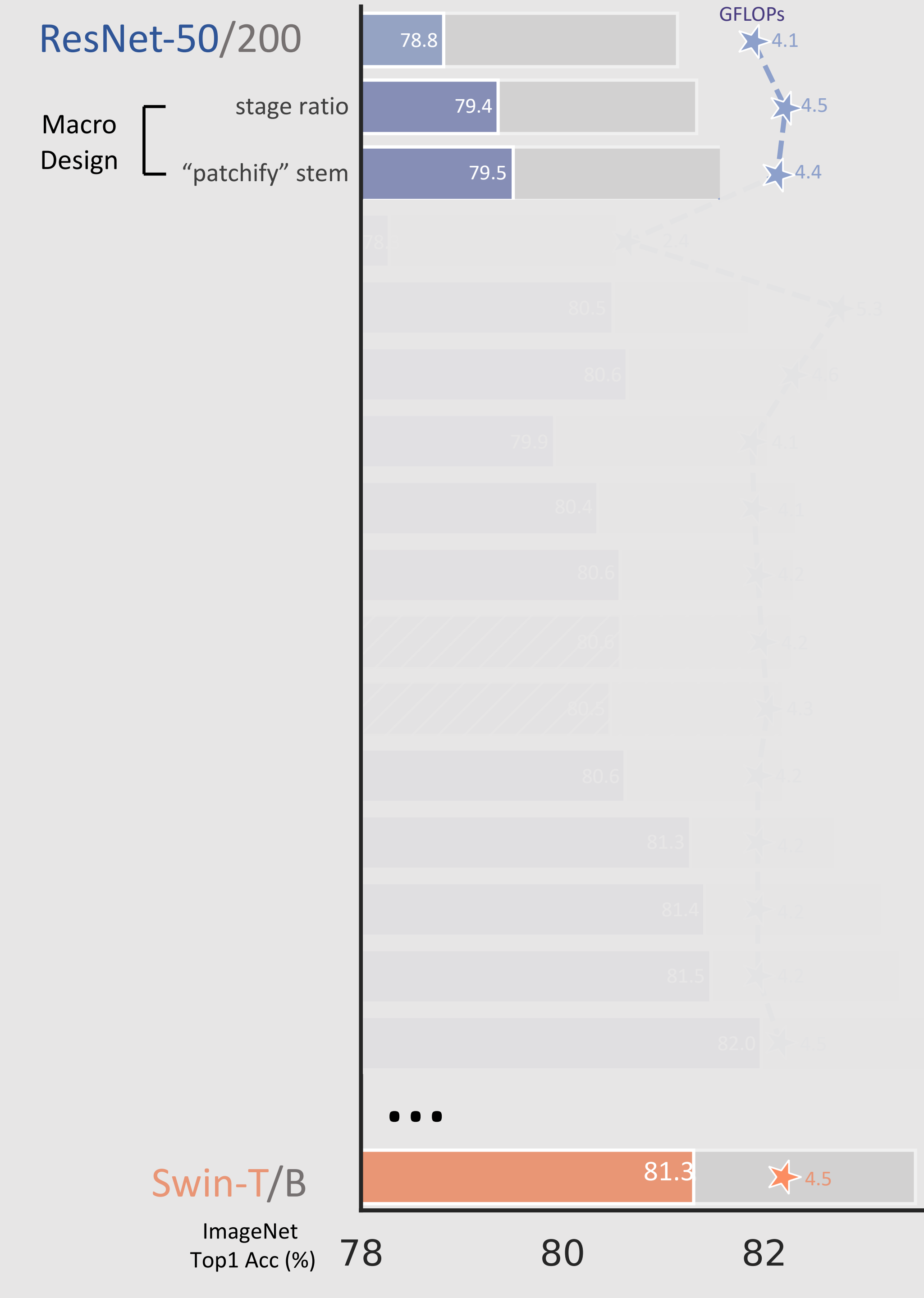
[Liu et al., 2021]



# Input Stem (ResNet)

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| | | 3×3 max pool, stride 2 | | | | |
| conv2_x | 56×56 | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

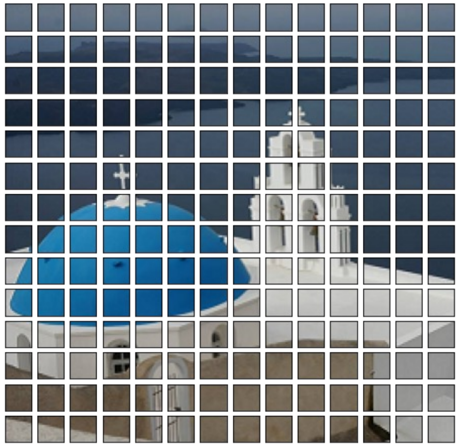


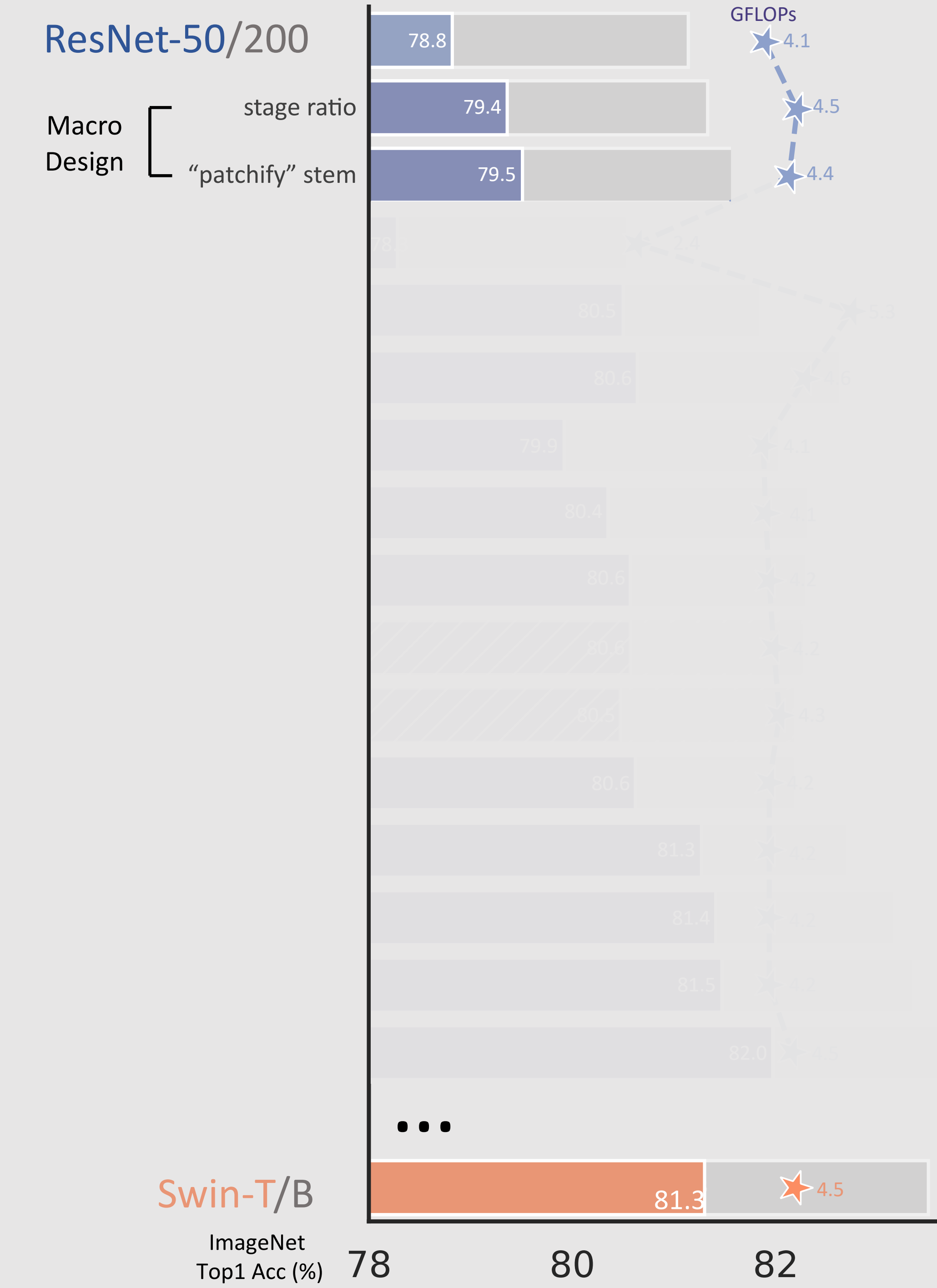
# Input Stem

Overlapping Conv + Max Pooling

Non-overlapping Conv (4x4, stride 4)

(a.k.a Patchify)





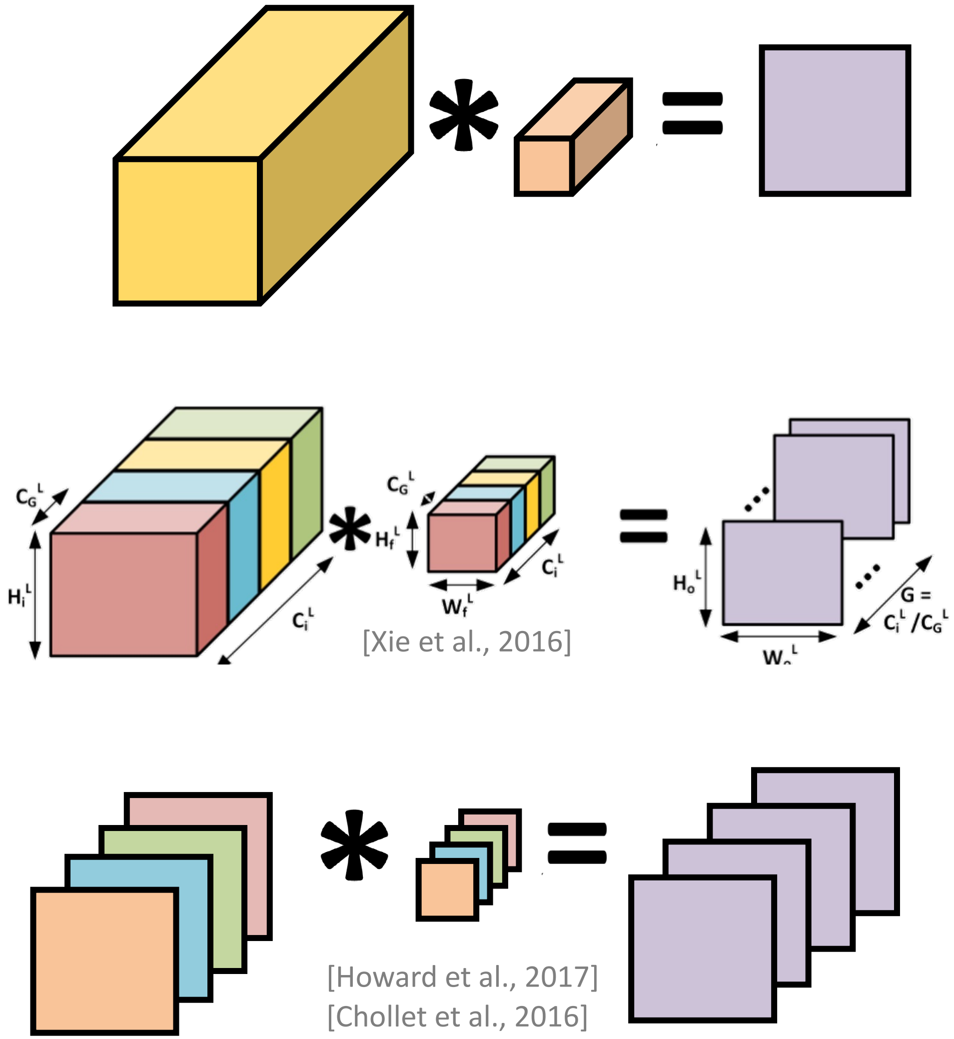
# ResNeXt-ify

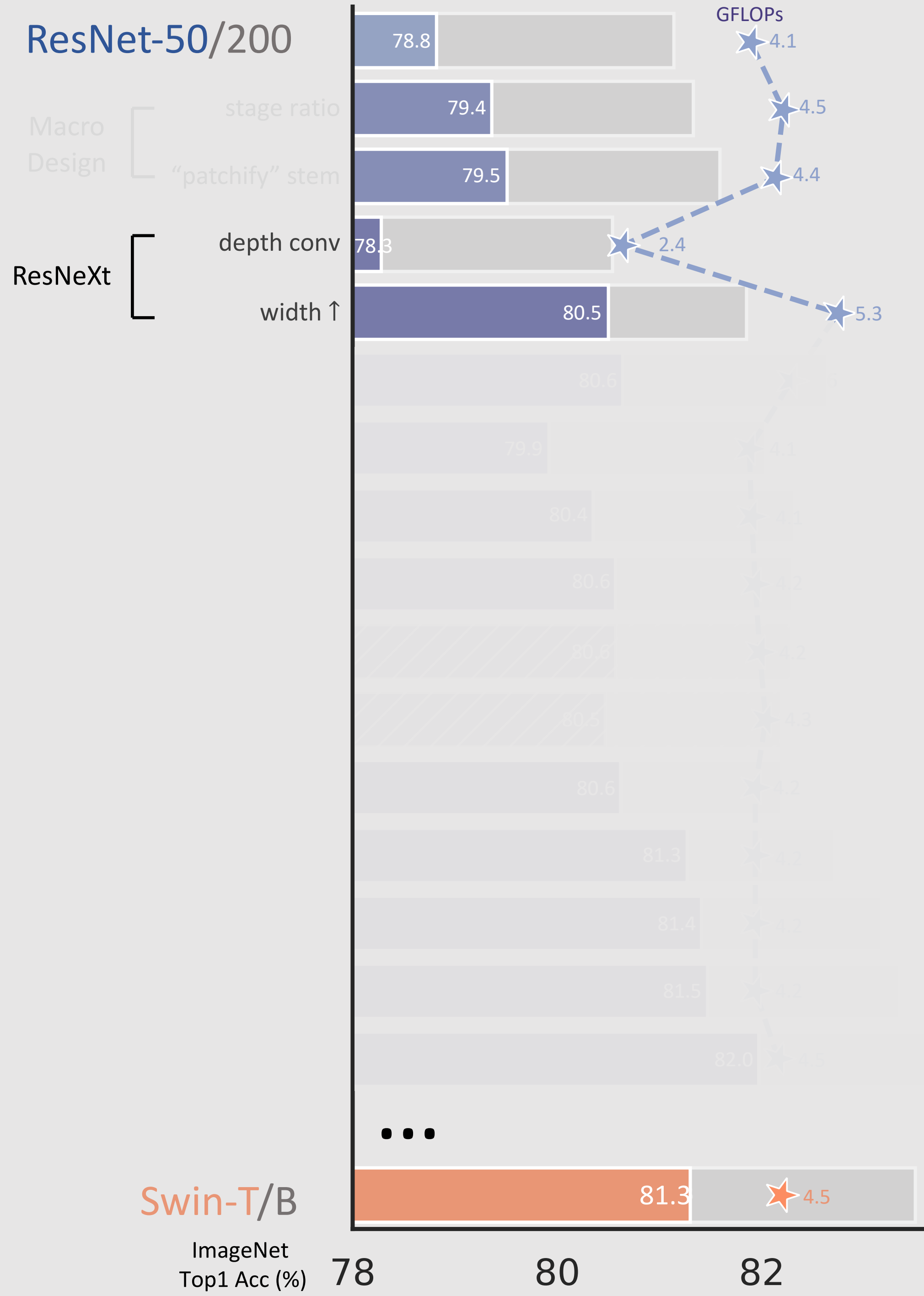
**Dense Conv**

**Grouped Conv**

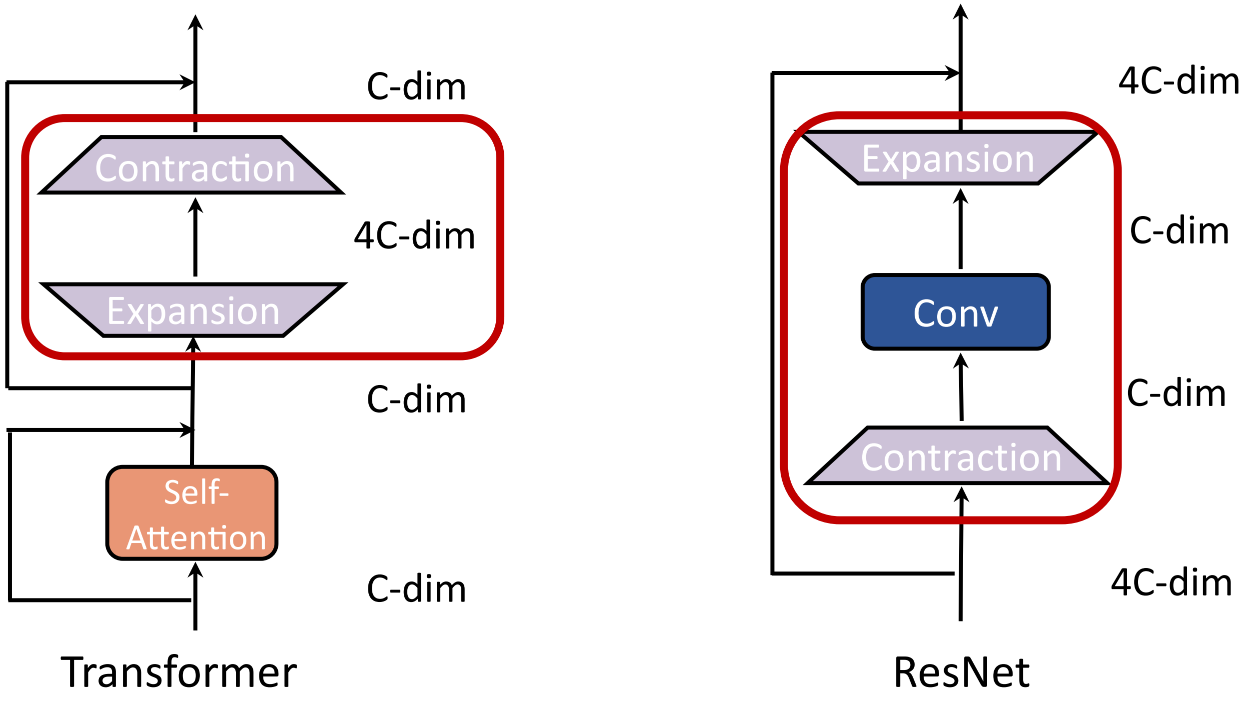
**Depthwise Conv**

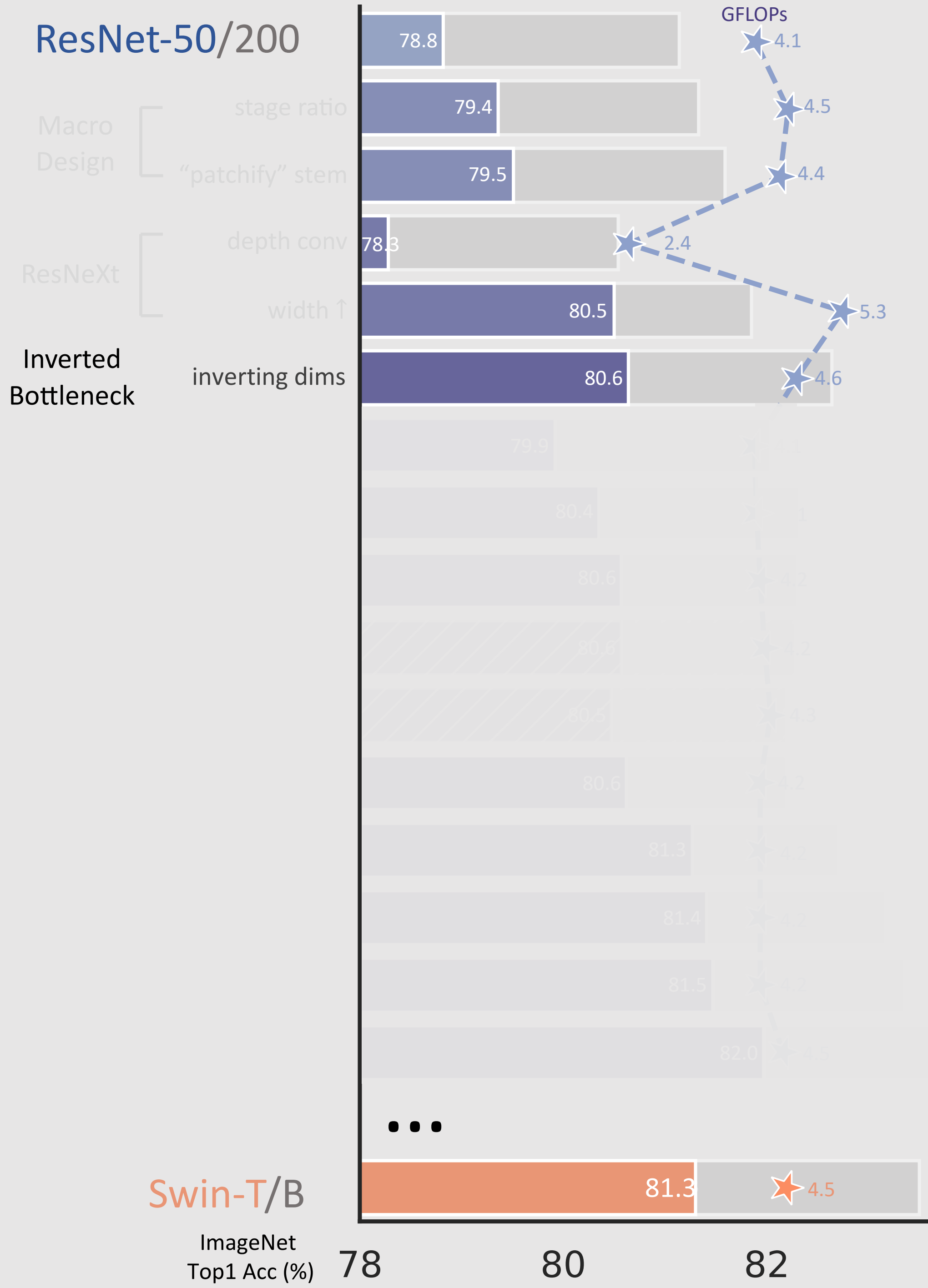
**# groups = # channels**



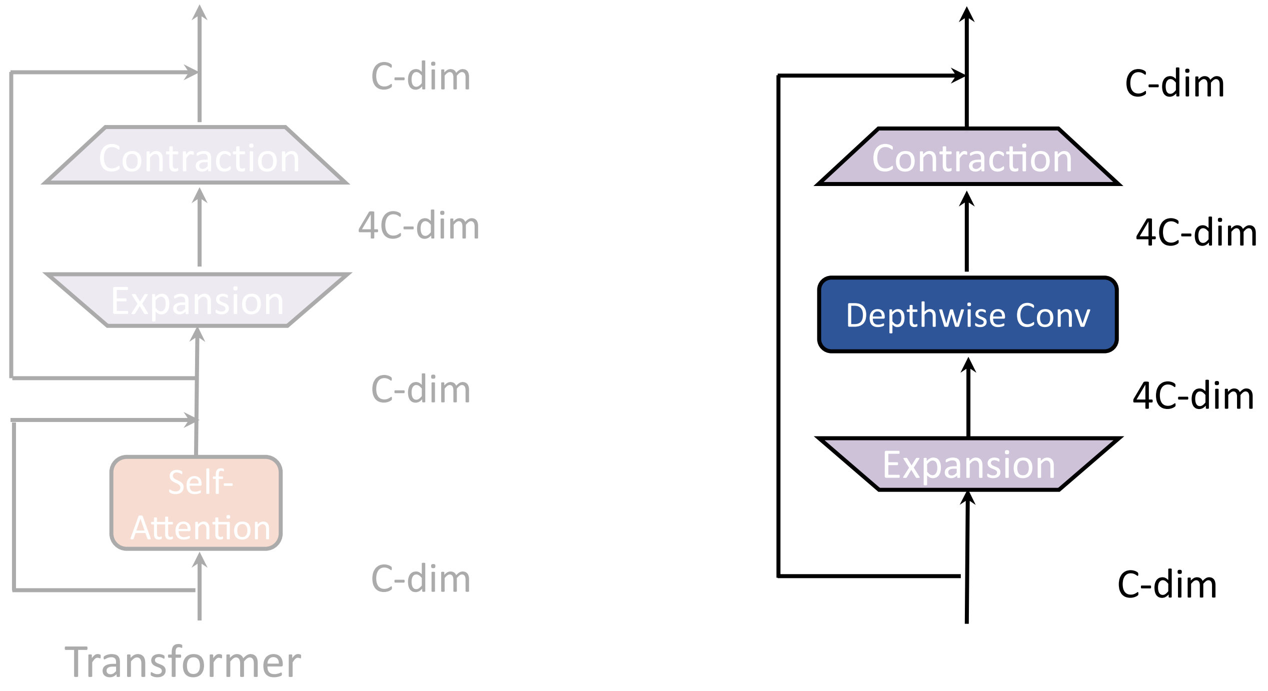



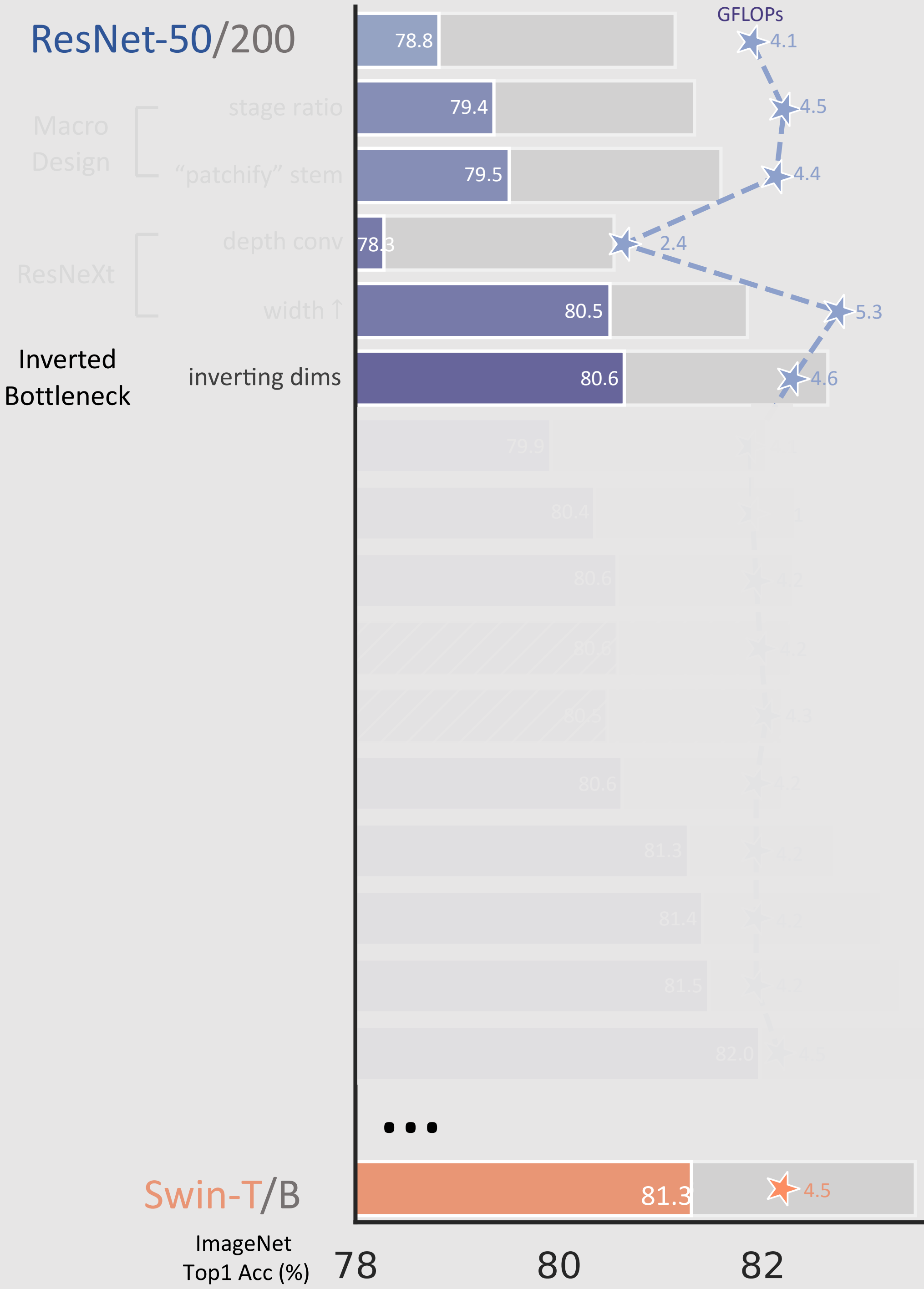
# Inverted Bottleneck



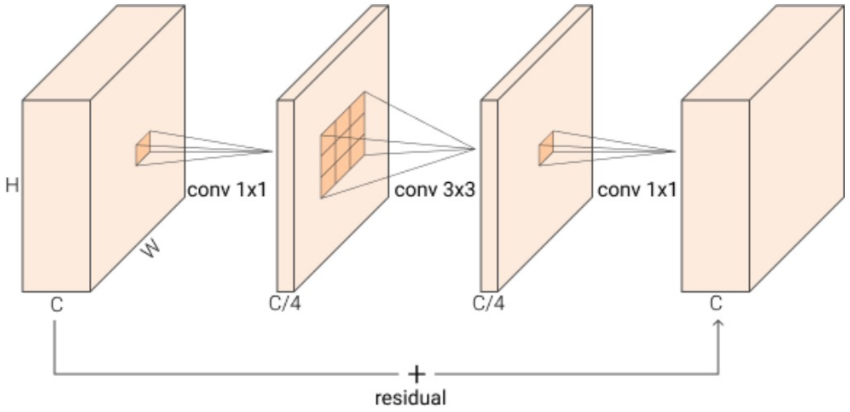



# Inverted Bottleneck



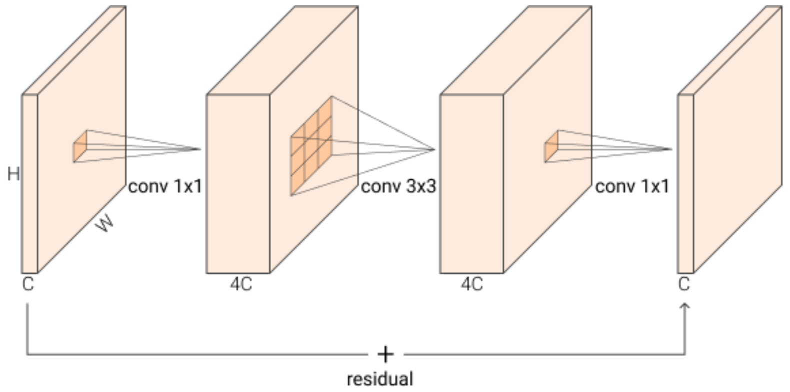
**2016** **ResNet**

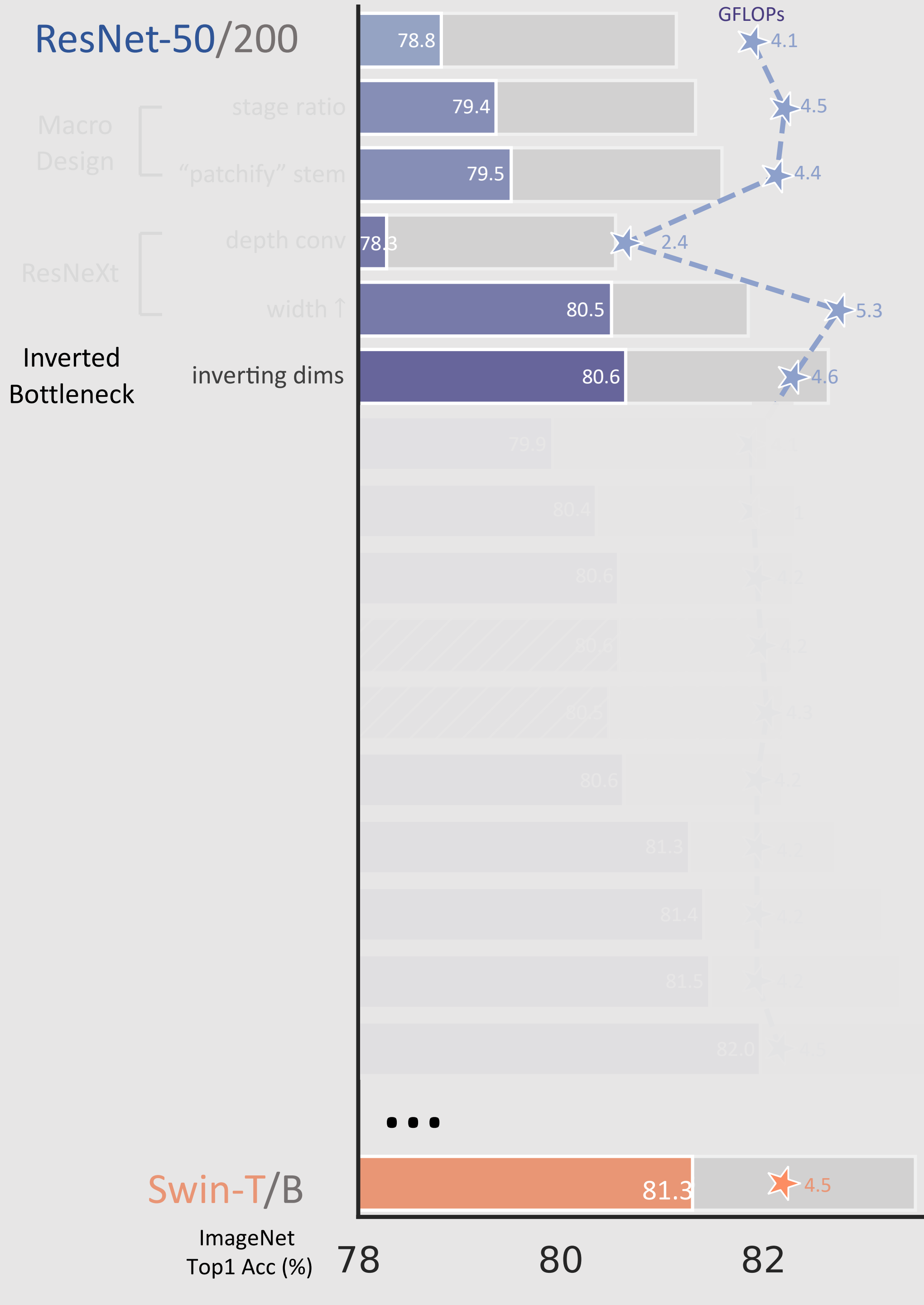



**2018** **Inverted Bottleneck**

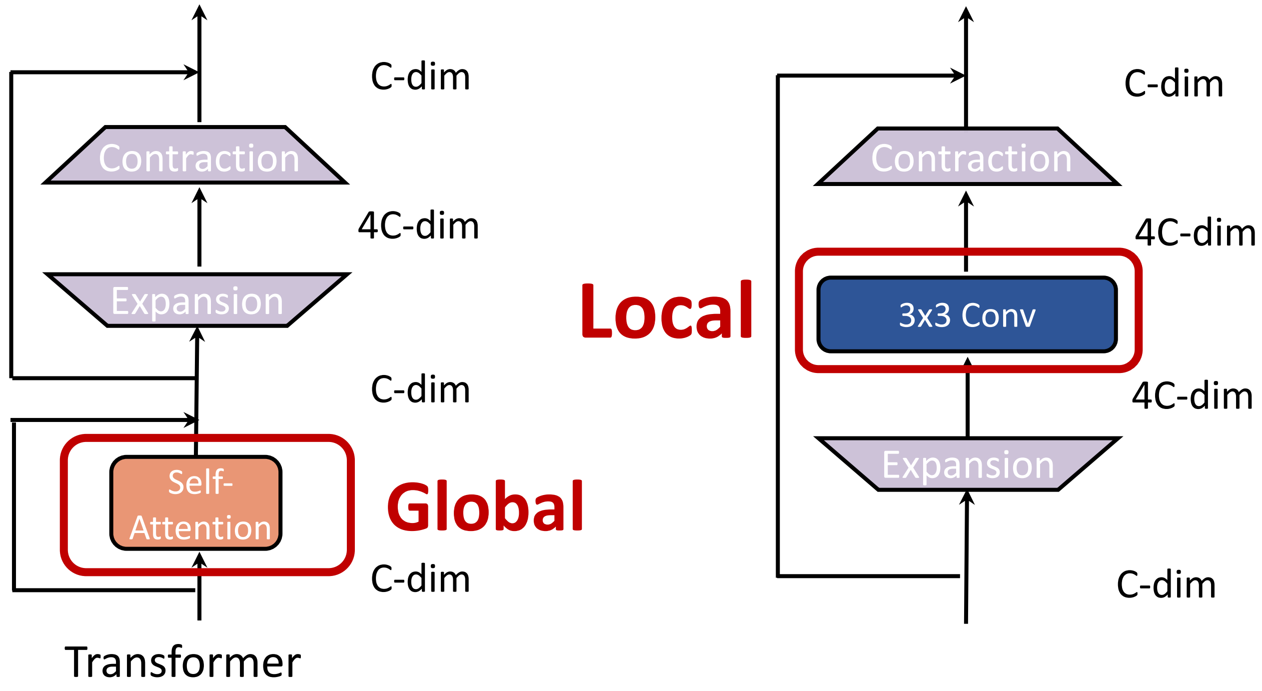
**MobileNet v2**

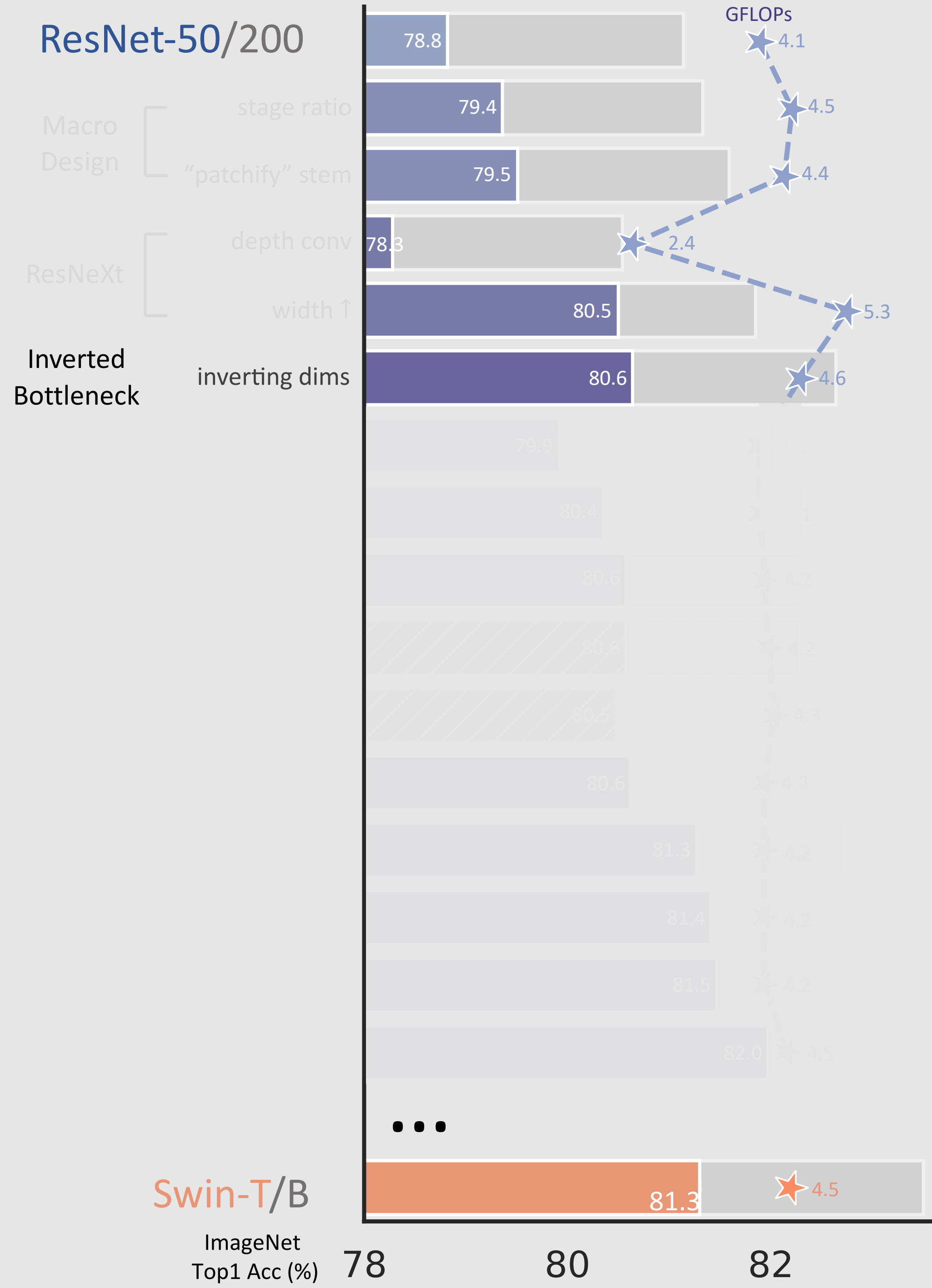
[Sandler et al, 2018]



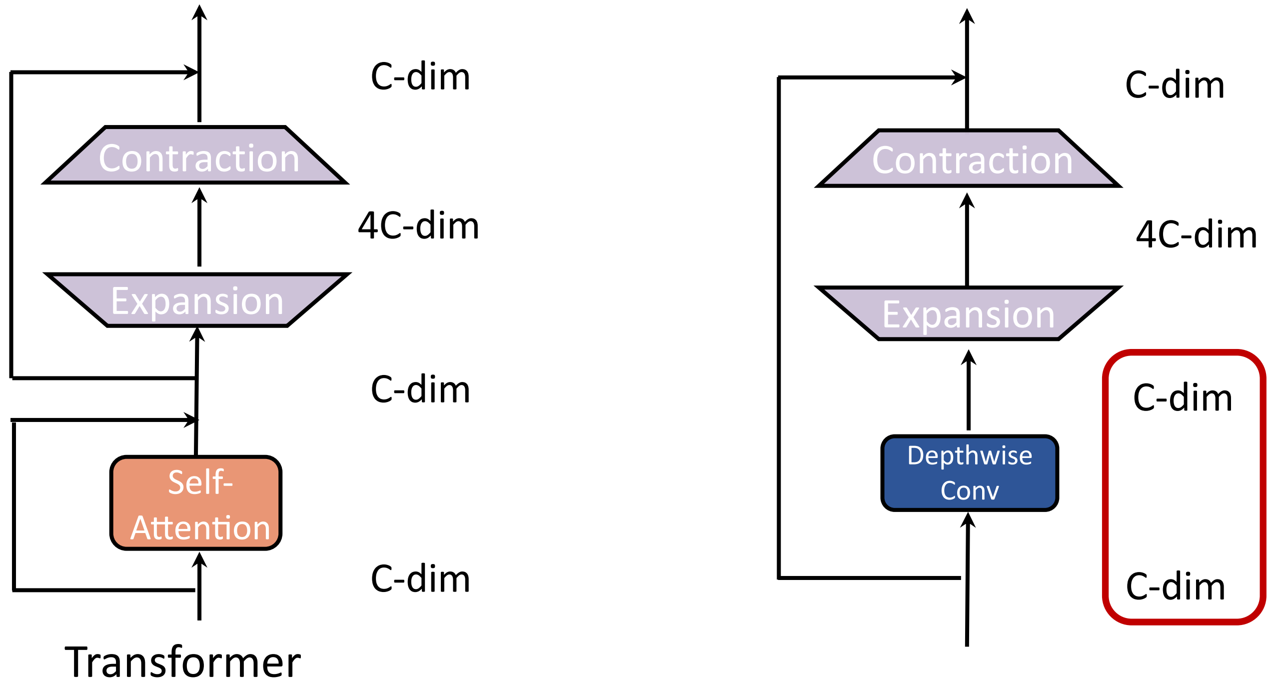



# Large kernel size

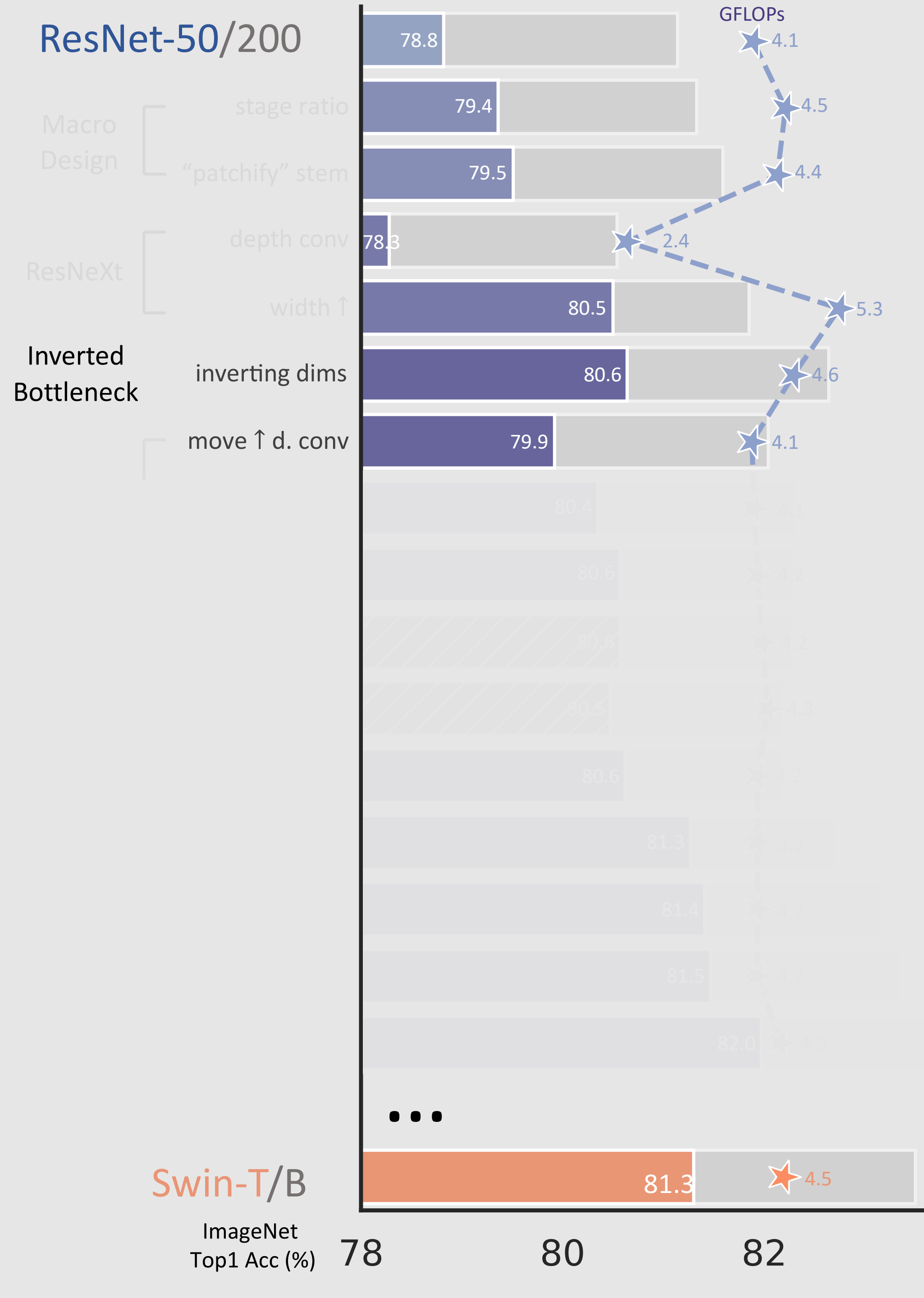


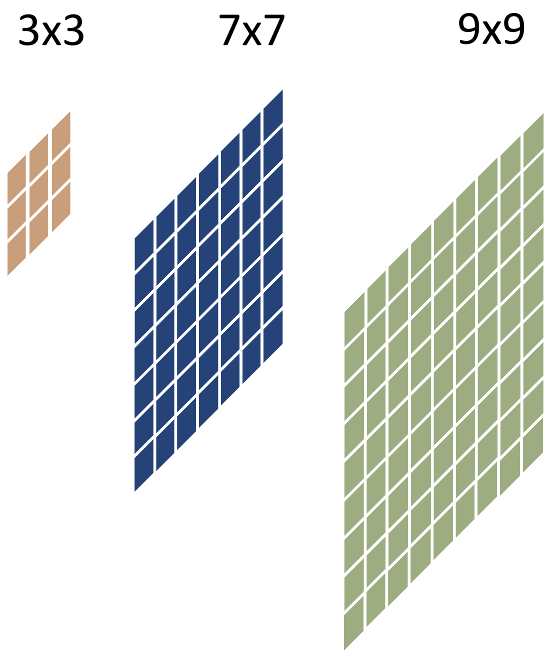
# Large kernel size

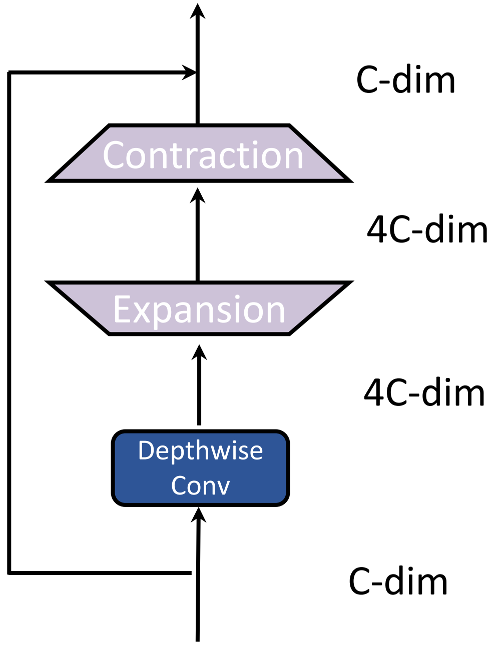


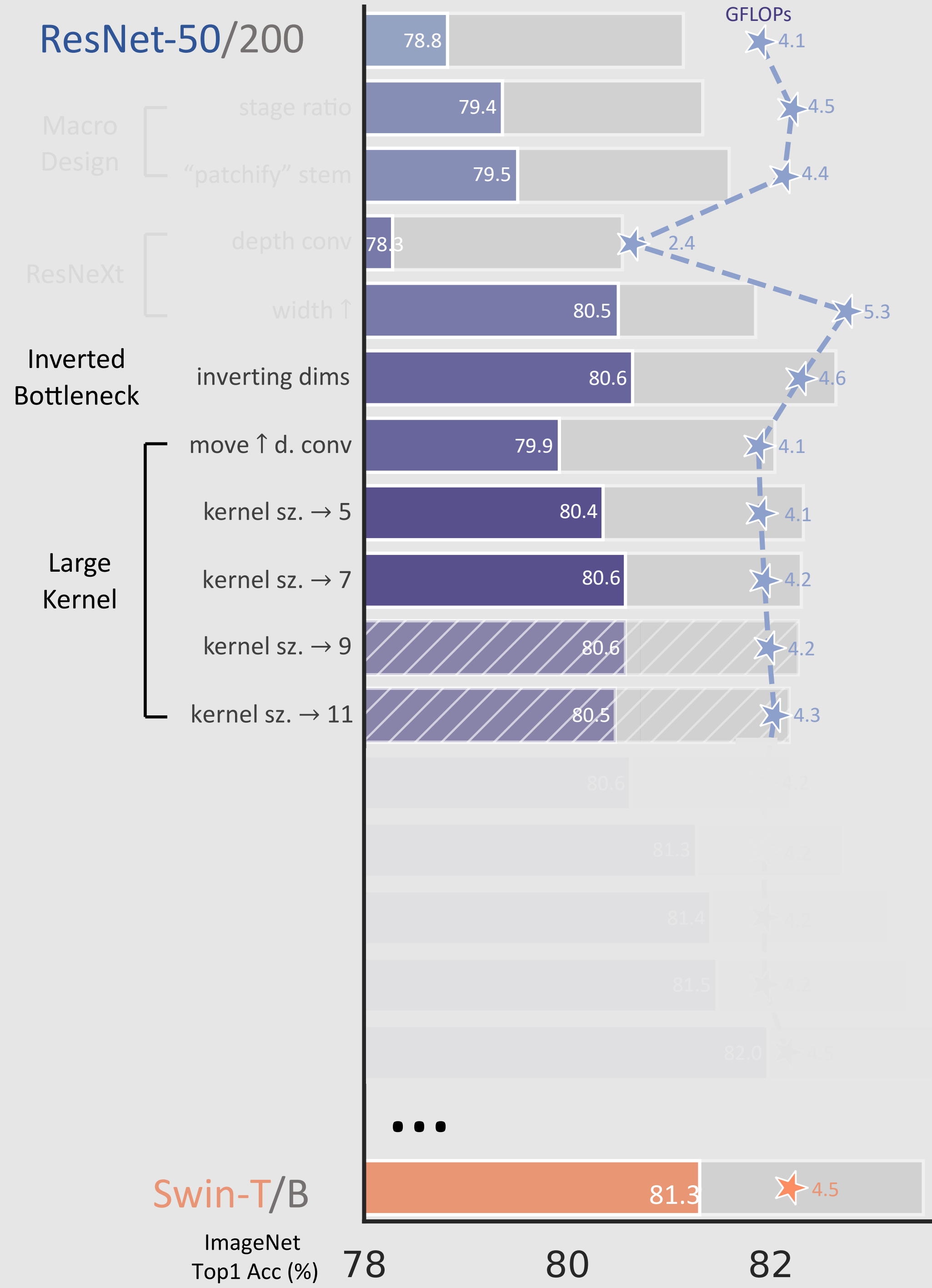

Prerequisite: move depth-wise before “expansion”.
Reduce flops w/ larger kernel size.



# Large kernel size

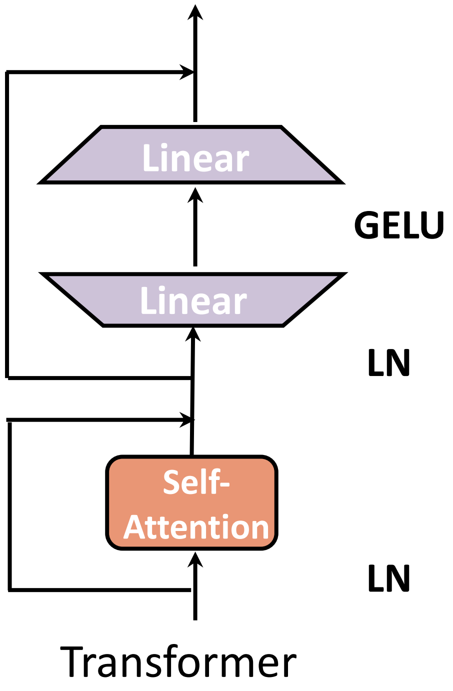


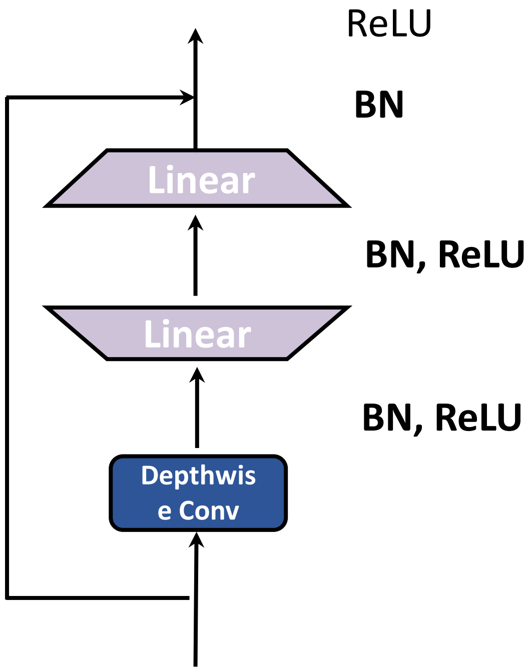



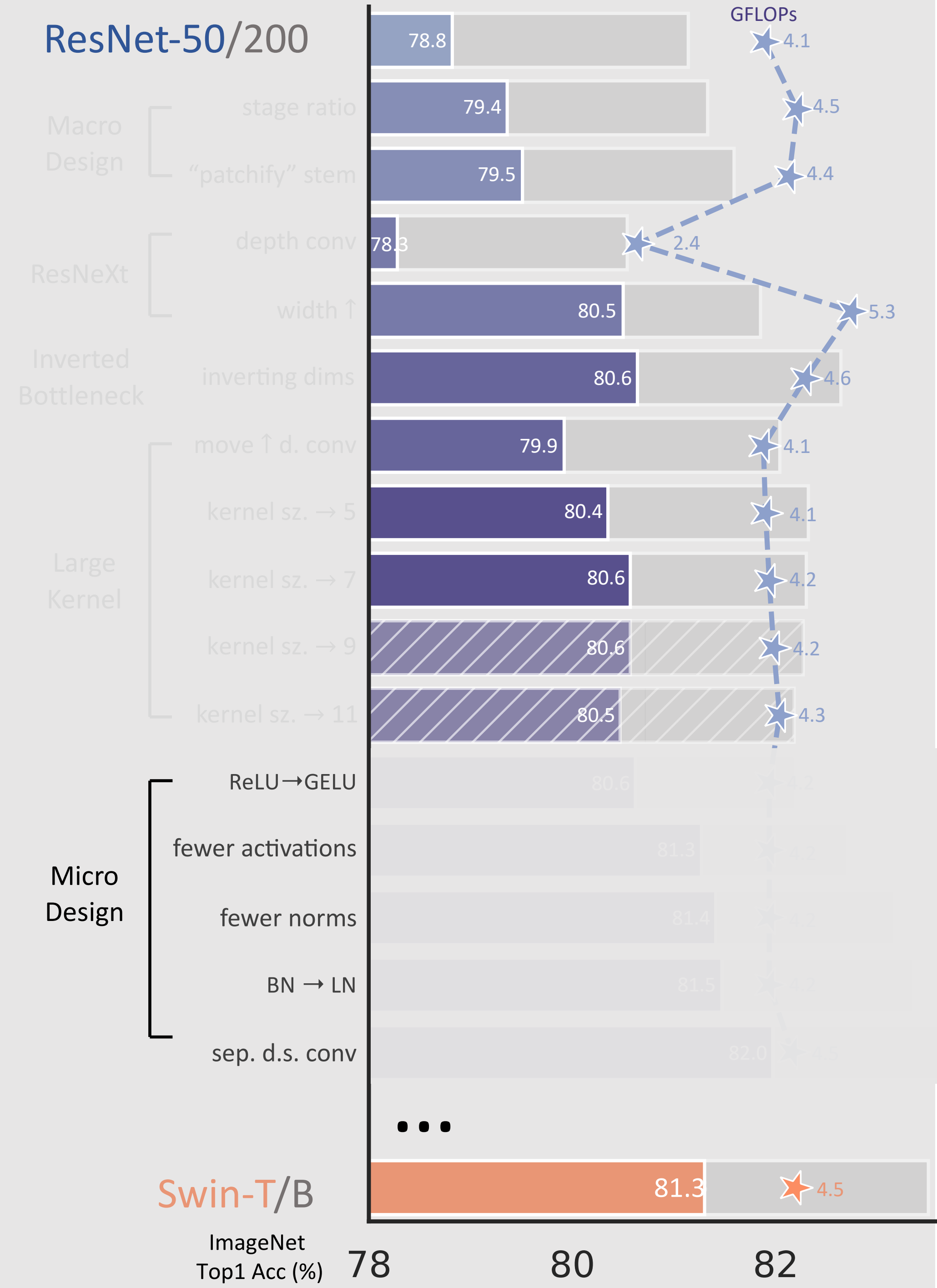




Larger kernel size helps; performance saturates at 7x7
Swin's choice of local window size is also 7 🤔

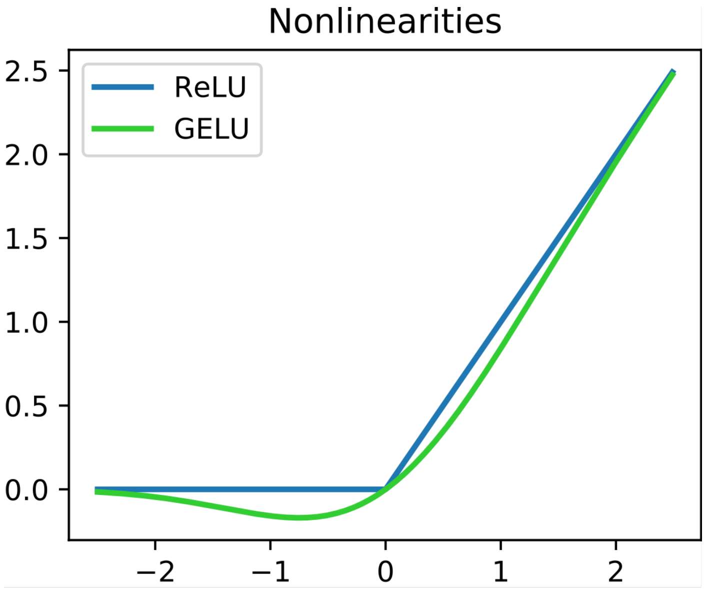
# Micro Design

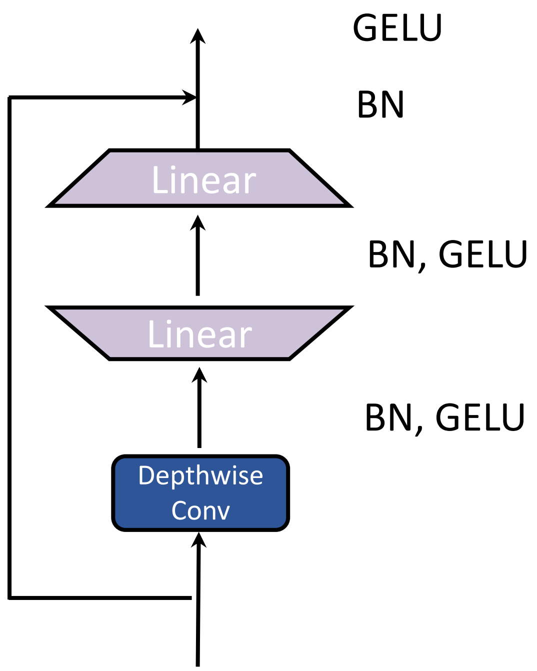


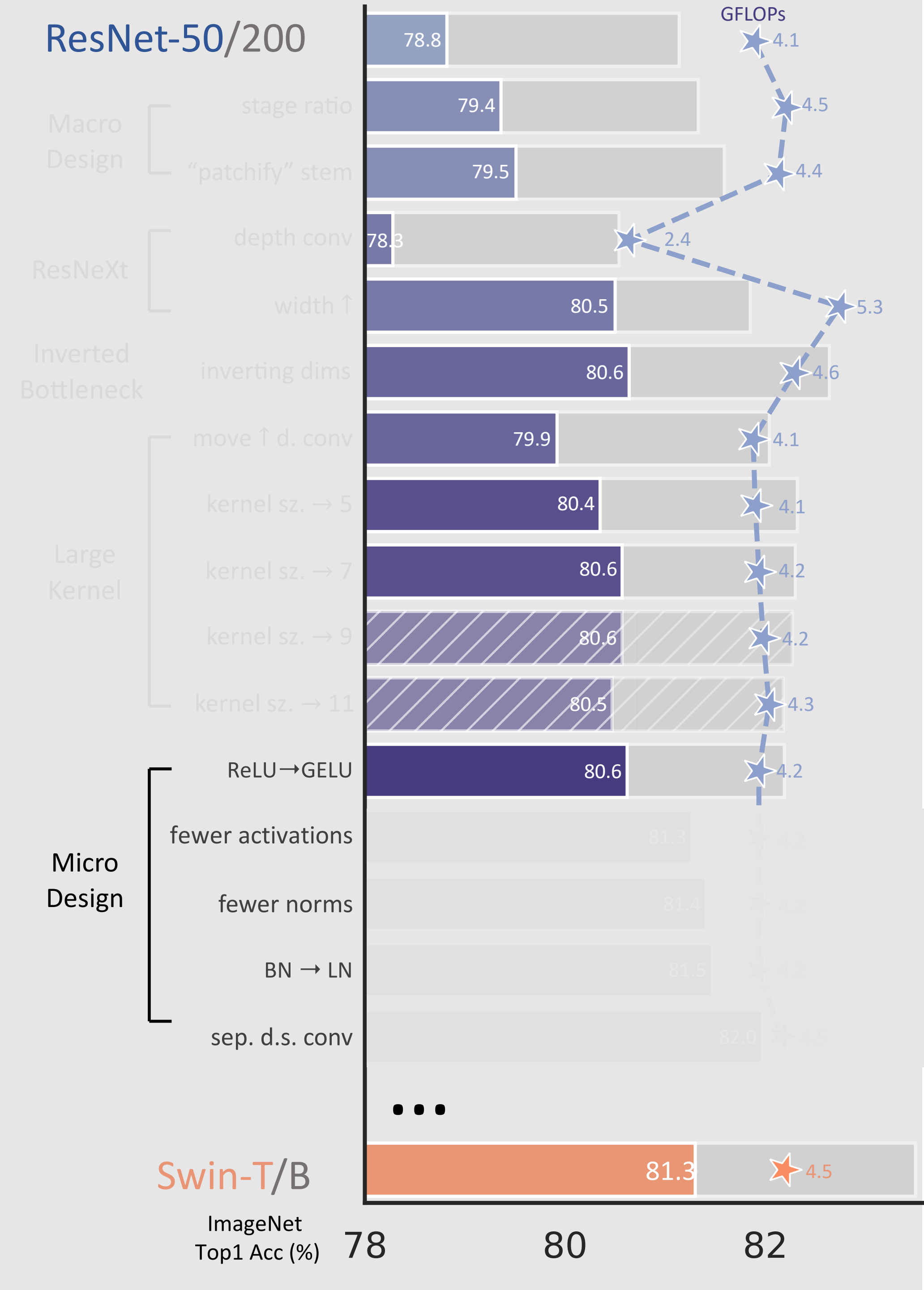



# Activations

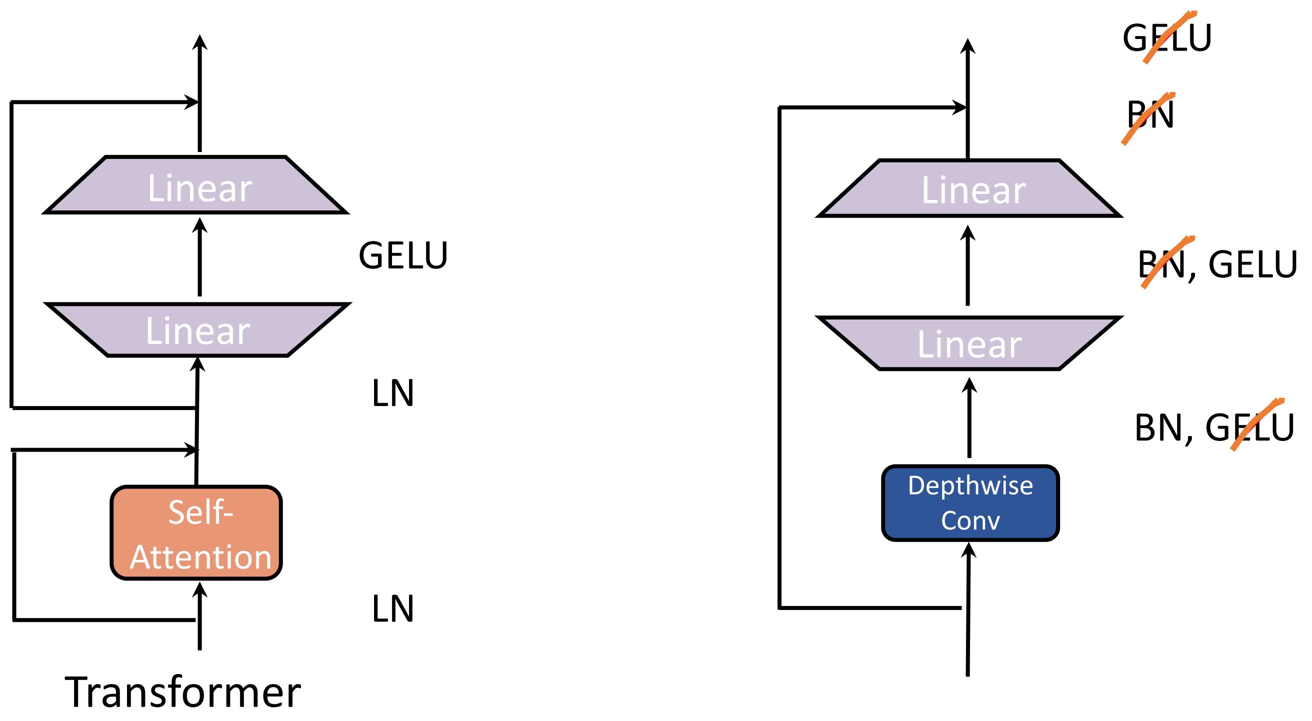



ReLU → GELU

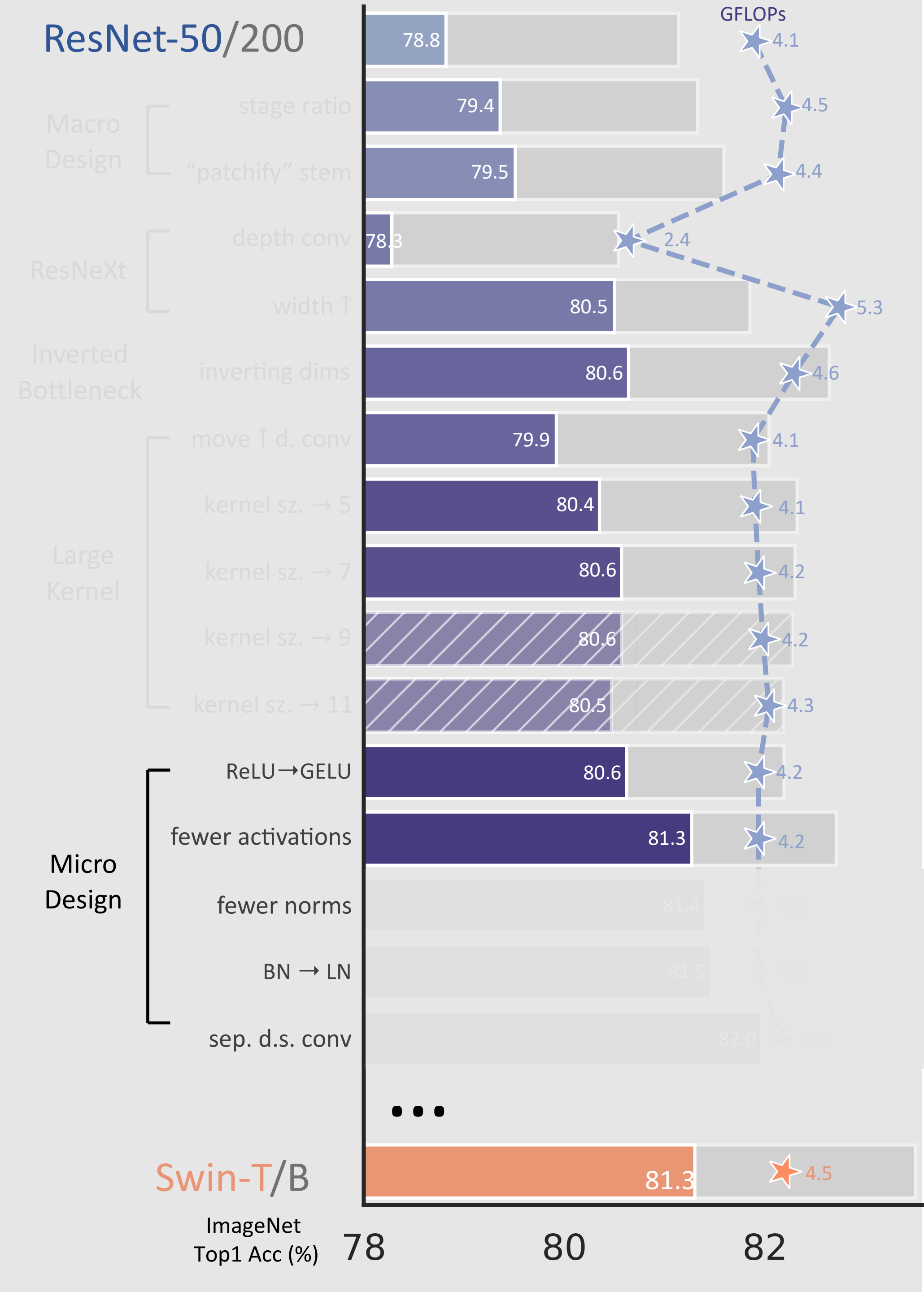


# Fewer Activations / Norms

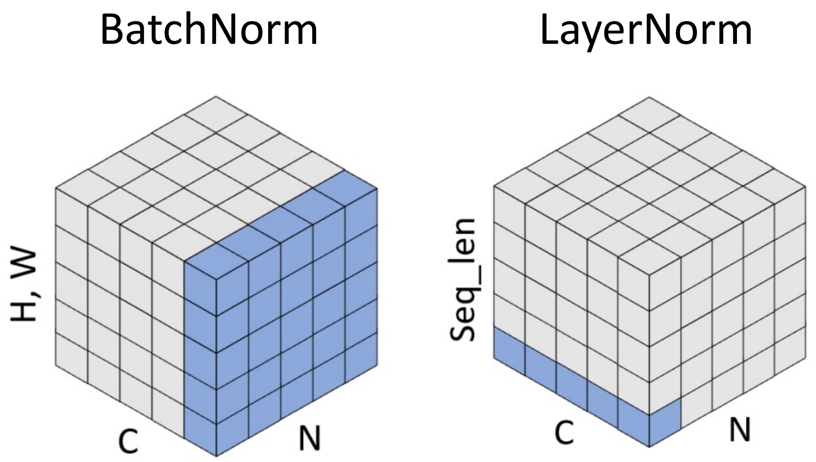


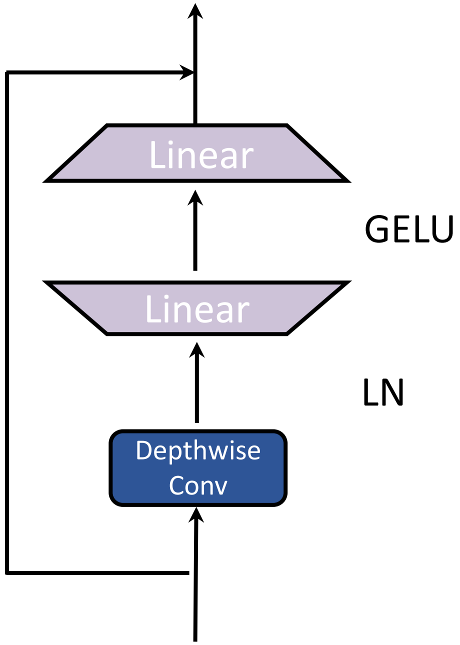

One norm, one activation is good enough per block

+0.8% without changing FLOPs

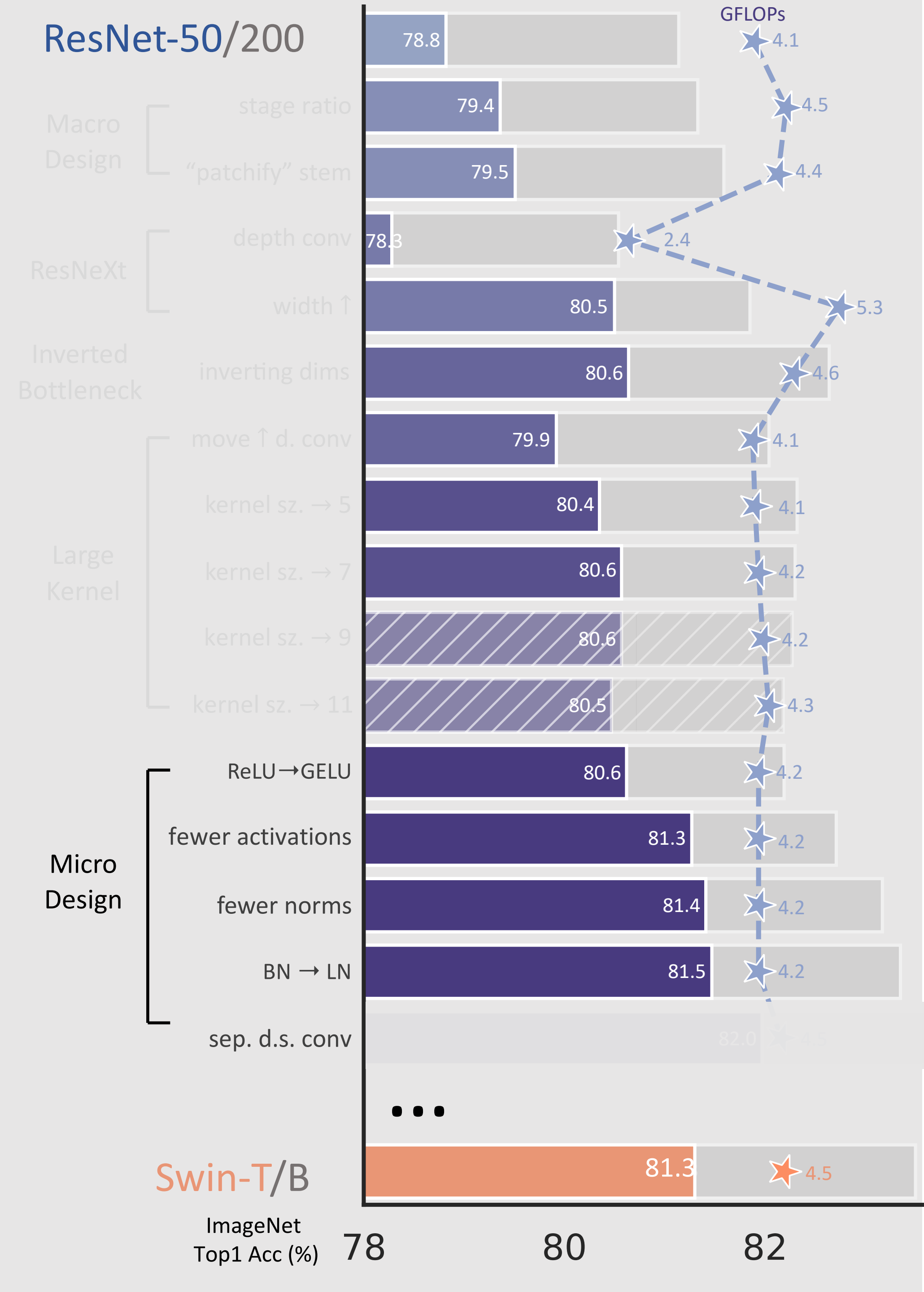


# Normalization

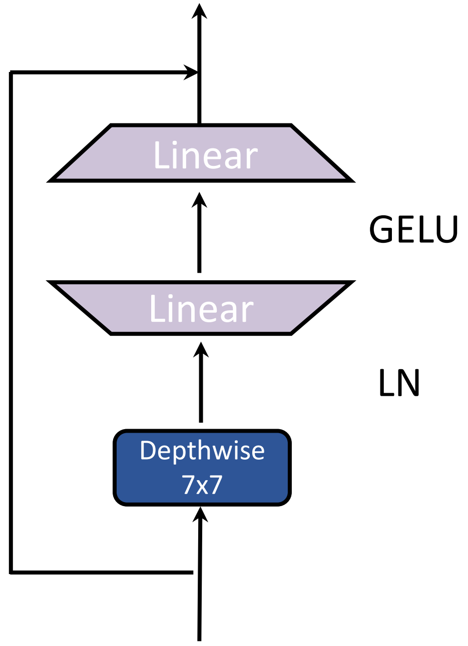



One **LayerNorm** is good enough.
Say 👋 to **BatchNorm** pitfalls!



# ConvNeXt Block

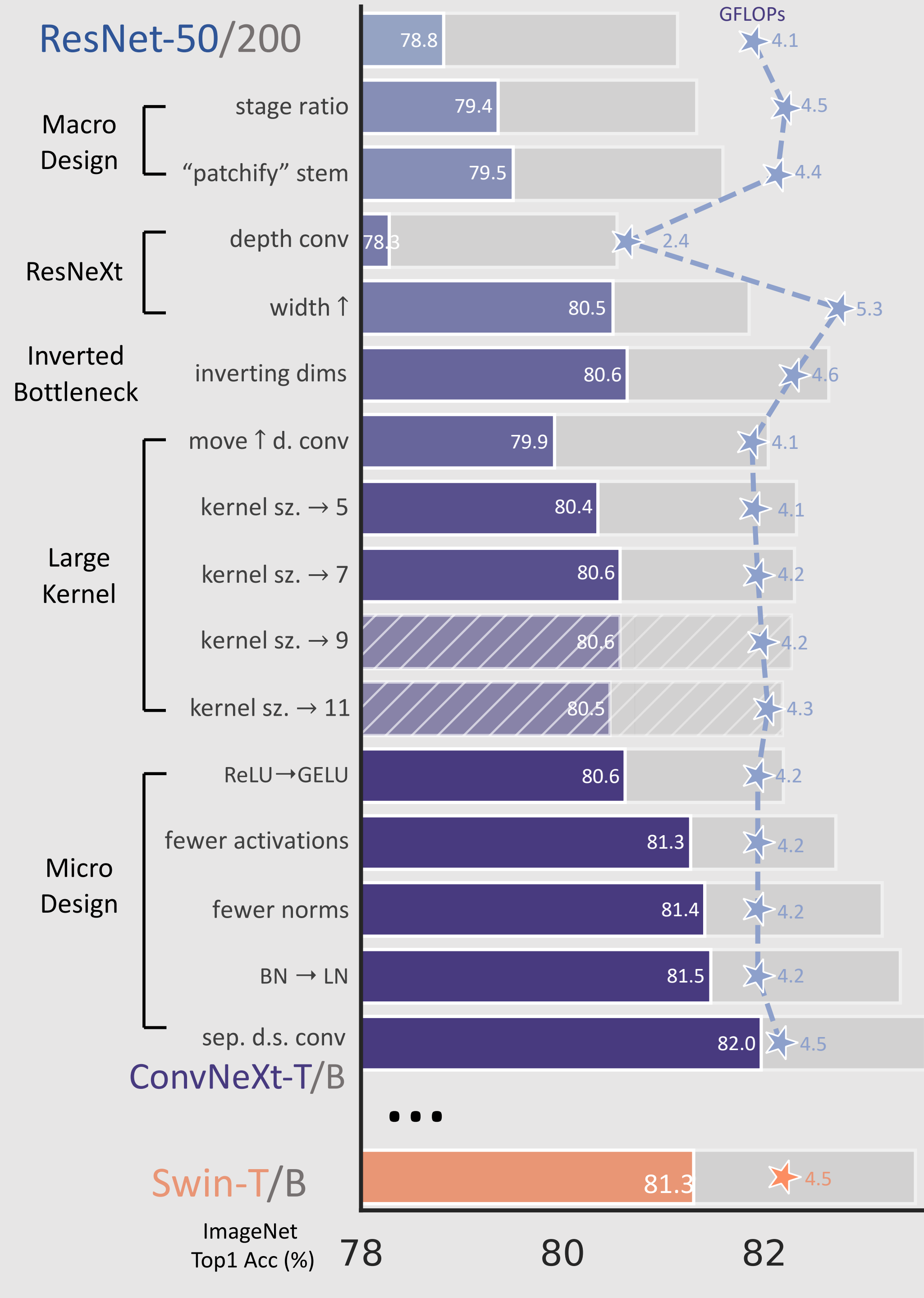



**Minimal design:**

⭐ as simple as possible (but not simpler).

⭐ only 100 lines of code

(vs. 500+ for advanced vision transformers)



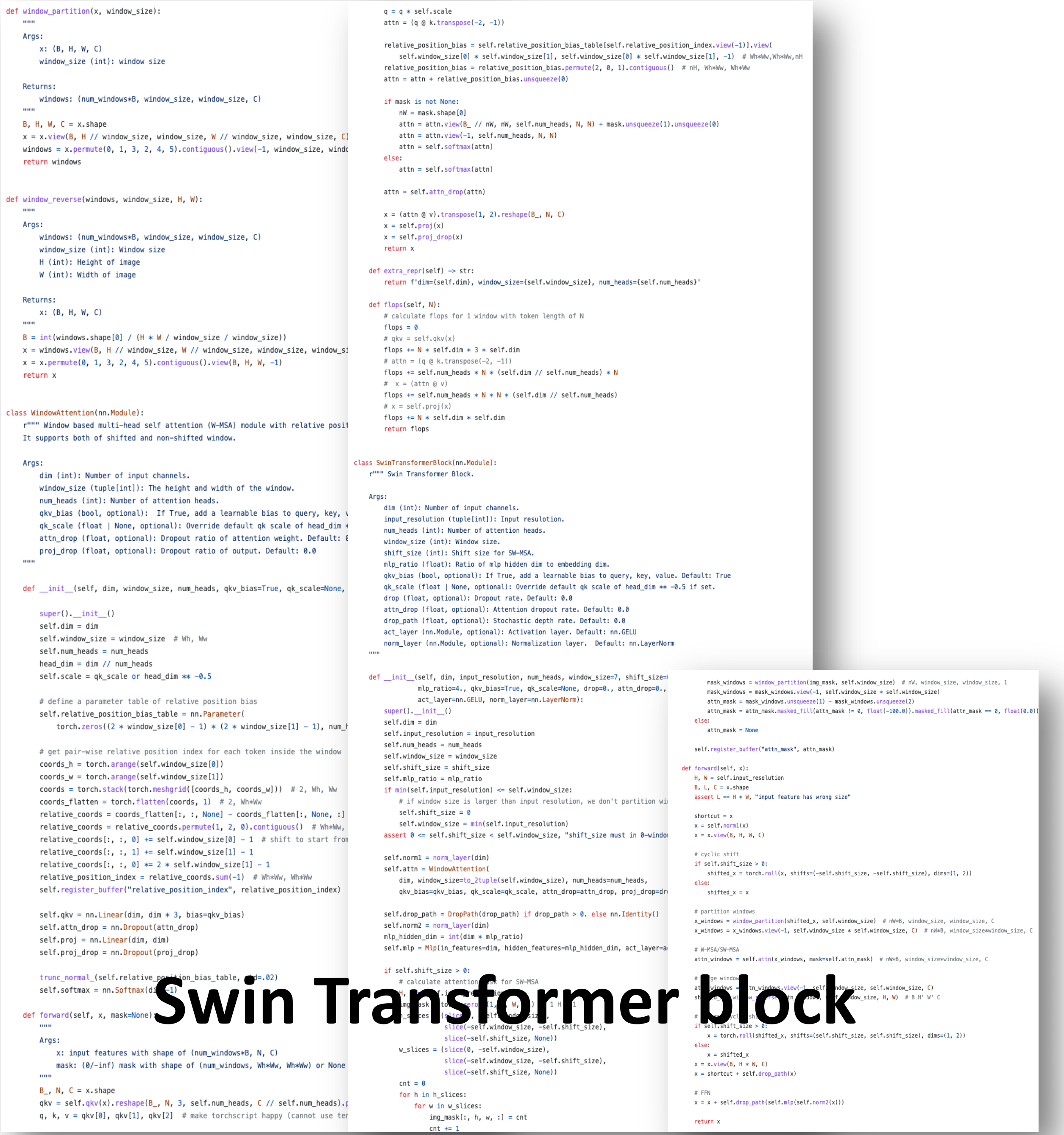
```python
class Block(nn.Module):
    r""" ConvNeXt Block. There are two equivalent implementations:
    (1) DwConv -> LayerNorm (channels_first) -> 1x1 Conv -> GELU -> 1x1 Conv; all in (N, C, H, W)
    (2) DwConv -> Permute to (N, H, W, C); LayerNorm (channels_last) -> Linear -> GELU -> Linear; Permute back
    We use (2) as we find it slightly faster in PyTorch

    Args:
        dim (int): Number of input channels.
        drop_path (float): Stochastic depth rate. Default: 0.0
        layer_scale_init_value (float): Init value for Layer Scale. Default: 1e-6.
    """
    def __init__(self, dim, drop_path=0., layer_scale_init_value=1e-6):
        super().__init__()
        self.dwconv = nn.Conv2d(dim, dim, kernel_size=7, padding=3, groups=dim) # depthwise conv
        self.norm = LayerNorm(dim, eps=1e-6)
        self.pwconv1 = nn.Linear(dim, 4 * dim) # pointwise/1x1 convs, implemented with linear layers
        self.act = nn.GELU()
        self.pwconv2 = nn.Linear(4 * dim, dim)
        self.gamma = nn.Parameter(layer_scale_init_value * torch.ones((dim)),
                                  requires_grad=True) if layer_scale_init_value > 0 else None
        self.drop_path = DropPath(drop_path) if drop_path > 0. else nn.Identity()

    def forward(self, x):
        input = x
        x = self.dwconv(x)
        x = x.permute(0, 2, 3, 1) # (N, C, H, W) -> (N, H, W, C)
        x = self.norm(x)
        x = self.pwconv1(x)
        x = self.act(x)
        x = self.pwconv2(x)
        if self.gamma is not None:
            x = self.gamma * x
        x = x.permute(0, 3, 1, 2) # (N, H, W, C) -> (N, C, H, W)

        x = input + self.drop_path(x)
        return x
```

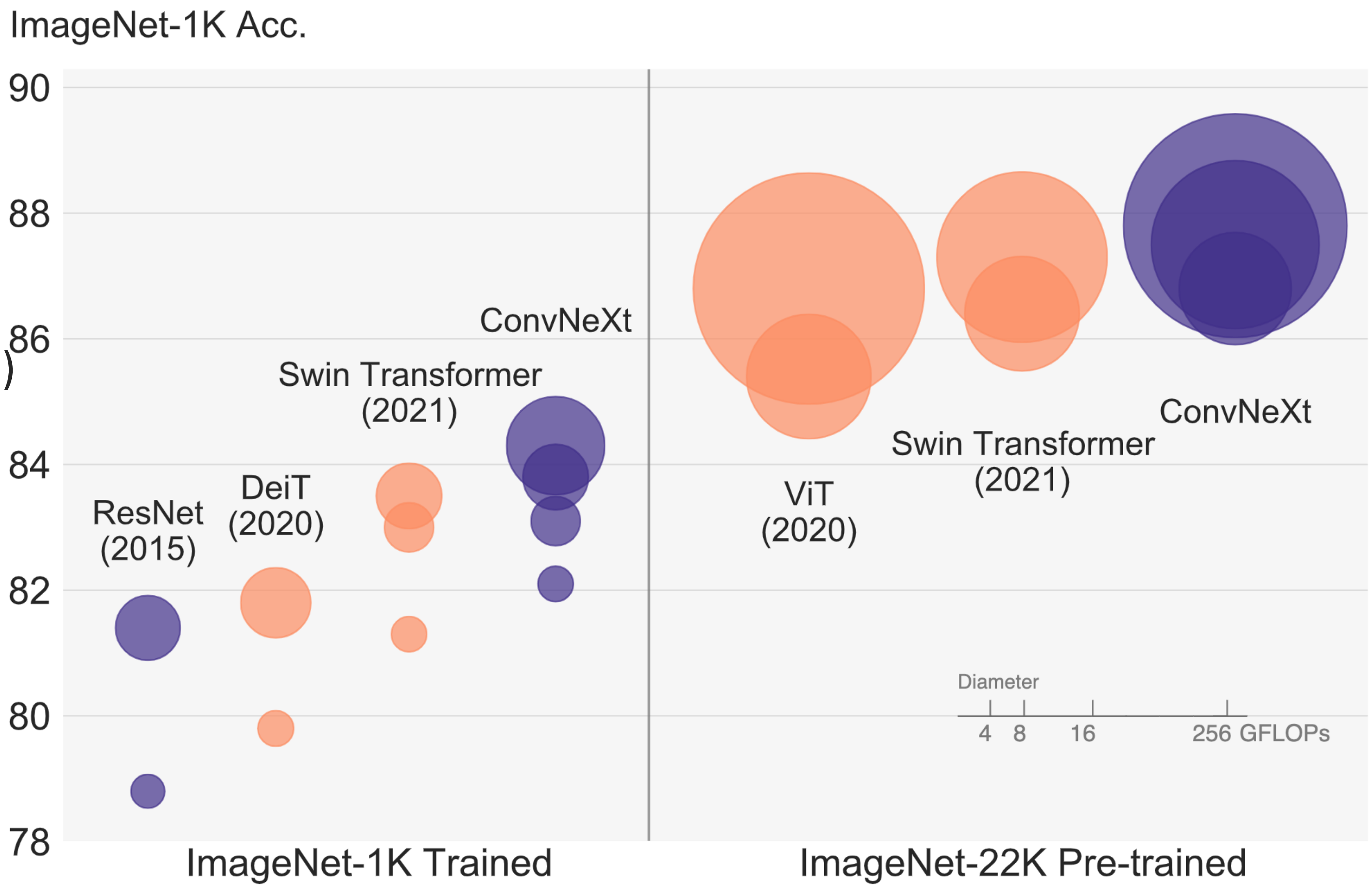
**ConvNeXt block**



**Swin Transformer block**

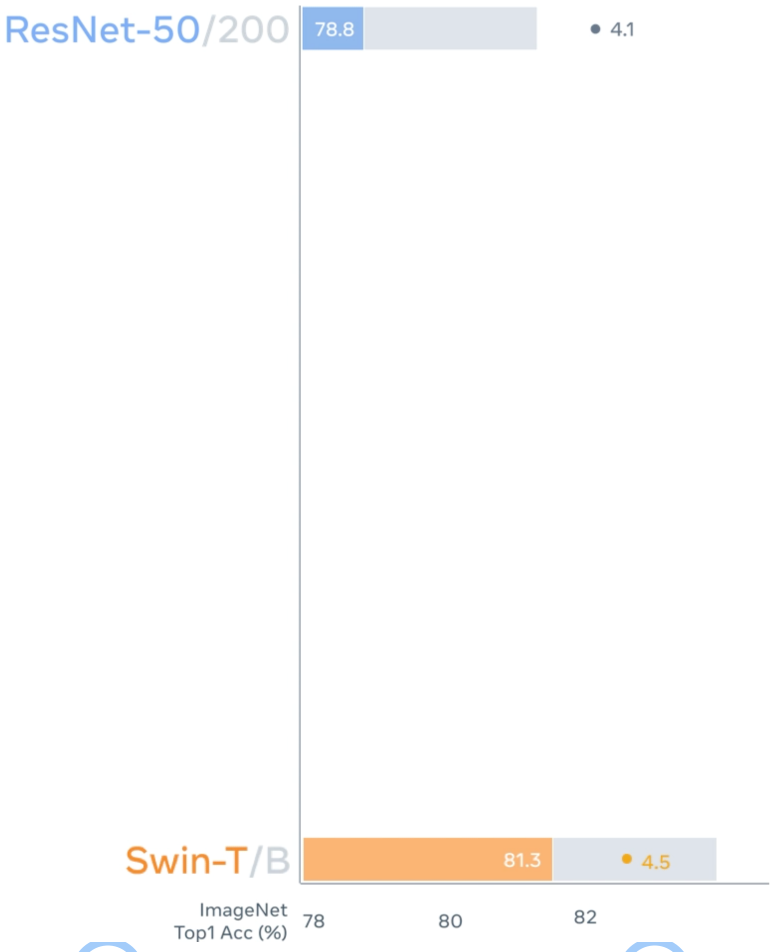
# ConvNeXt: Results

- Attention is *NOT* essential
- ConvNets can be scalable
  (while being much **simpler** in design)

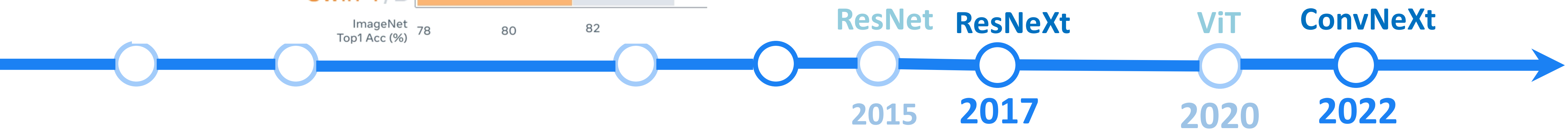


# ConvNeXt

[Liu, Mao, Wu, Feichtenhofer, Darrell, **Xie.** CVPR 2022]




- Not to push for SOTA
  - Focus on enabling fair comparisons
- Key design decisions are:
  - Mimicking Vision Transformers
  - Using standard ConvNet modules



# ConvNeXt: Downstream Transfer

## Consistently ***outperforms*** SOTA vision transformers

| backbone | FLOPs | FPS | $AP^{box}$ | $AP^{box}_{50}$ | $AP^{box}_{75}$ | $AP^{mask}$ | $AP^{mask}_{50}$ | $AP^{mask}_{75}$ |
|---|---|---|---|---|---|---|---|---|
| Mask-RCNN 3× schedule | | | | | | | | |
| ○ Swin-T | 267G | 23.1 | 46.0 | 68.1 | 50.3 | 41.6 | 65.1 | 44.9 |
| ● ConvNeXt-T | 262G | 25.6 | **46.2** | 67.9 | 50.8 | **41.7** | 65.0 | 44.9 |
| Cascade Mask-RCNN 3× schedule | | | | | | | | |
| ● ResNet-50 | 739G | 11.4 | 46.3 | 64.3 | 50.5 | 40.1 | 61.7 | 43.4 |
| ● X101-32 | 819G | 9.2 | 48.1 | 66.5 | 52.4 | 41.6 | 63.9 | 45.2 |
| ● X101-64 | 972G | 7.1 | 48.3 | 66.4 | 52.3 | 41.7 | 64.0 | 45.1 |
| ○ Swin-T | 745G | 12.2 | 50.4 | 69.2 | 54.7 | 43.7 | 66.6 | 47.3 |
| ● ConvNeXt-T | 741G | 13.5 | **50.4** | 69.1 | 54.8 | **43.7** | 66.5 | 47.3 |
| ○ Swin-S | 838G | 11.4 | 51.9 | 70.7 | 56.3 | 45.0 | 68.2 | 48.8 |
| ● ConvNeXt-S | 827G | 12.0 | **51.9** | 70.8 | 56.5 | **45.0** | 68.4 | 49.1 |
| ○ Swin-B | 982G | 10.7 | 51.9 | 70.5 | 56.4 | 45.0 | 68.1 | 48.9 |
| ● ConvNeXt-B | 964G | 11.4 | **52.7** | 71.3 | 57.2 | **45.6** | 68.9 | 49.5 |
| ○ Swin-B‡ | 982G | 10.7 | 53.0 | 71.8 | 57.5 | 45.8 | 69.4 | 49.7 |
| ● ConvNeXt-B‡ | 964G | 11.5 | **54.0** | 73.1 | 58.8 | **46.9** | 70.6 | 51.3 |
| ○ Swin-L‡ | 1382G | 9.2 | 53.9 | 72.4 | 58.8 | 46.7 | 70.1 | 50.8 |
| ● ConvNeXt-L‡ | 1354G | 10.0 | **54.8** | 73.8 | 59.8 | **47.6** | 71.3 | 51.7 |
| ● ConvNeXt-XL‡ | 1898G | 8.6 | **55.2** | 74.2 | 59.9 | **47.7** | 71.6 | 52.2 |

COCO Detection and Instance Segmentation

| backbone | input crop. | mIoU | #param. | FLOPs |
|---|---|---|---|---|
| ImageNet-1K pre-trained | | | | |
| ○ Swin-T | $512^2$ | 45.8 | 60M | 945G |
| ● ConvNeXt-T | $512^2$ | **46.7** | 60M | 939G |
| ○ Swin-S | $512^2$ | 49.5 | 81M | 1038G |
| ● ConvNeXt-S | $512^2$ | **49.6** | 82M | 1027G |
| ○ Swin-B | $512^2$ | 49.7 | 121M | 1188G |
| ● ConvNeXt-B | $512^2$ | **49.9** | 122M | 1170G |
| ImageNet-22K pre-trained | | | | |
| ○ Swin-B‡ | $640^2$ | 51.7 | 121M | 1841G |
| ● ConvNeXt-B‡ | $640^2$ | **53.1** | 122M | 1828G |
| ○ Swin-L‡ | $640^2$ | 53.5 | 234M | 2468G |
| ● ConvNeXt-L‡ | $640^2$ | **53.7** | 235M | 2458G |
| ● ConvNeXt-XL‡ | $640^2$ | **54.0** | 391M | 3335G |

ADE20K Semantic Segmentation

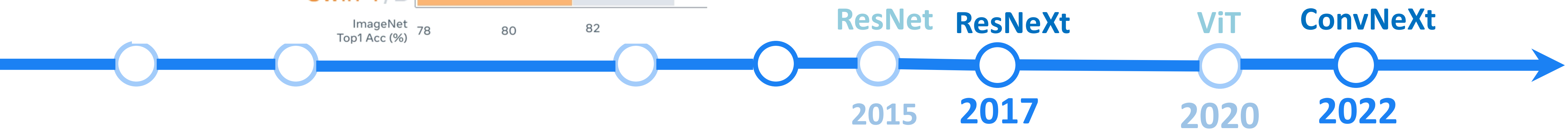
# ConvNeXt

[Liu, Mao, Wu, Feichtenhofer, Darrell, **Xie.** CVPR 2022]


- Not to push for SOTA
  - Focus on enabling fair comparisons

⭐ ConvNeXt represents the **community effort**!

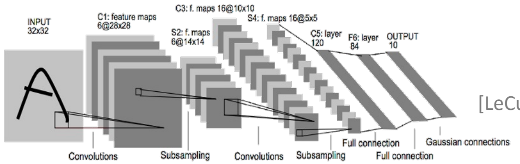
Many design choices have been examined **separately** over the last decade, but *not* **collectively**.



# Connectivity pattern matters

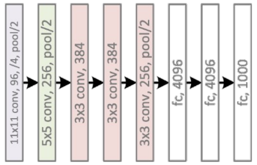
## chain-like wiring patterns

LeNet



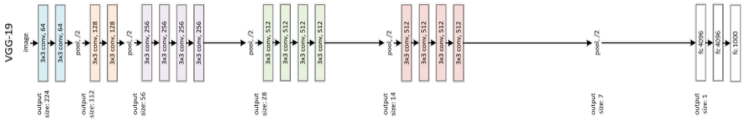

[LeCun et al, 1989]

AlexNet




[Krizhevsky et al, 1989]

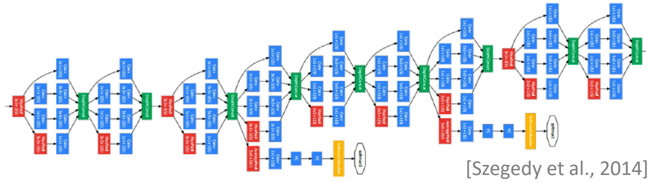
VGGNet



[Simonyan and Zisserman, 2014]

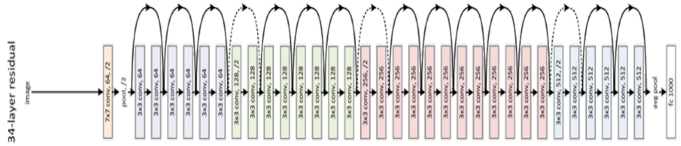
## multiple wiring paths

Inception



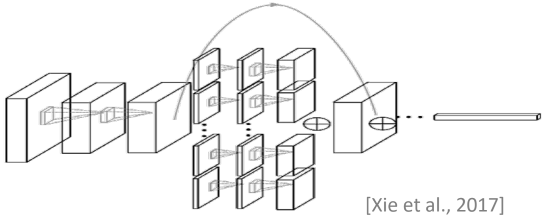
[Szegedy et al., 2014]

ResNet



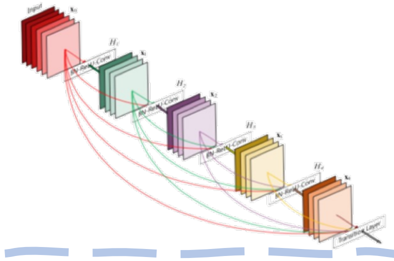
[He et al., 2015]

ResNeXt



[Xie et al., 2017]

DenseNet

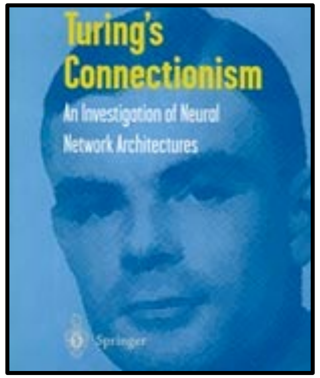


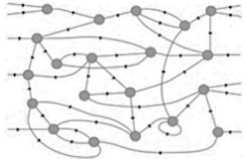
[Huang et al., 2017]

connectivity pattern = graph structure

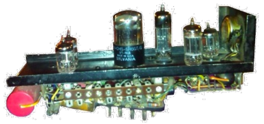
# How to wire your computational networks?

Turing (1948)
*"Unorganized machines"*

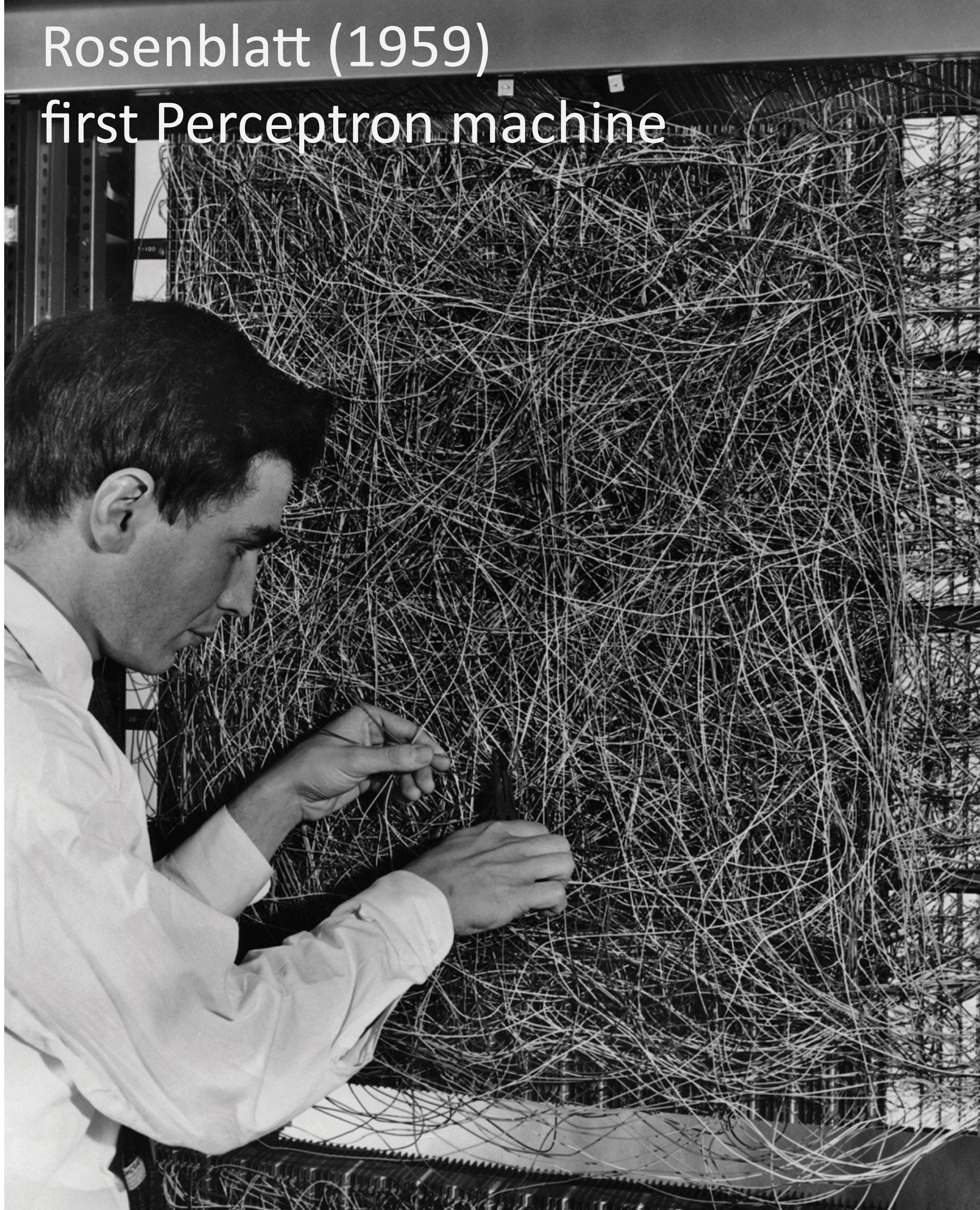


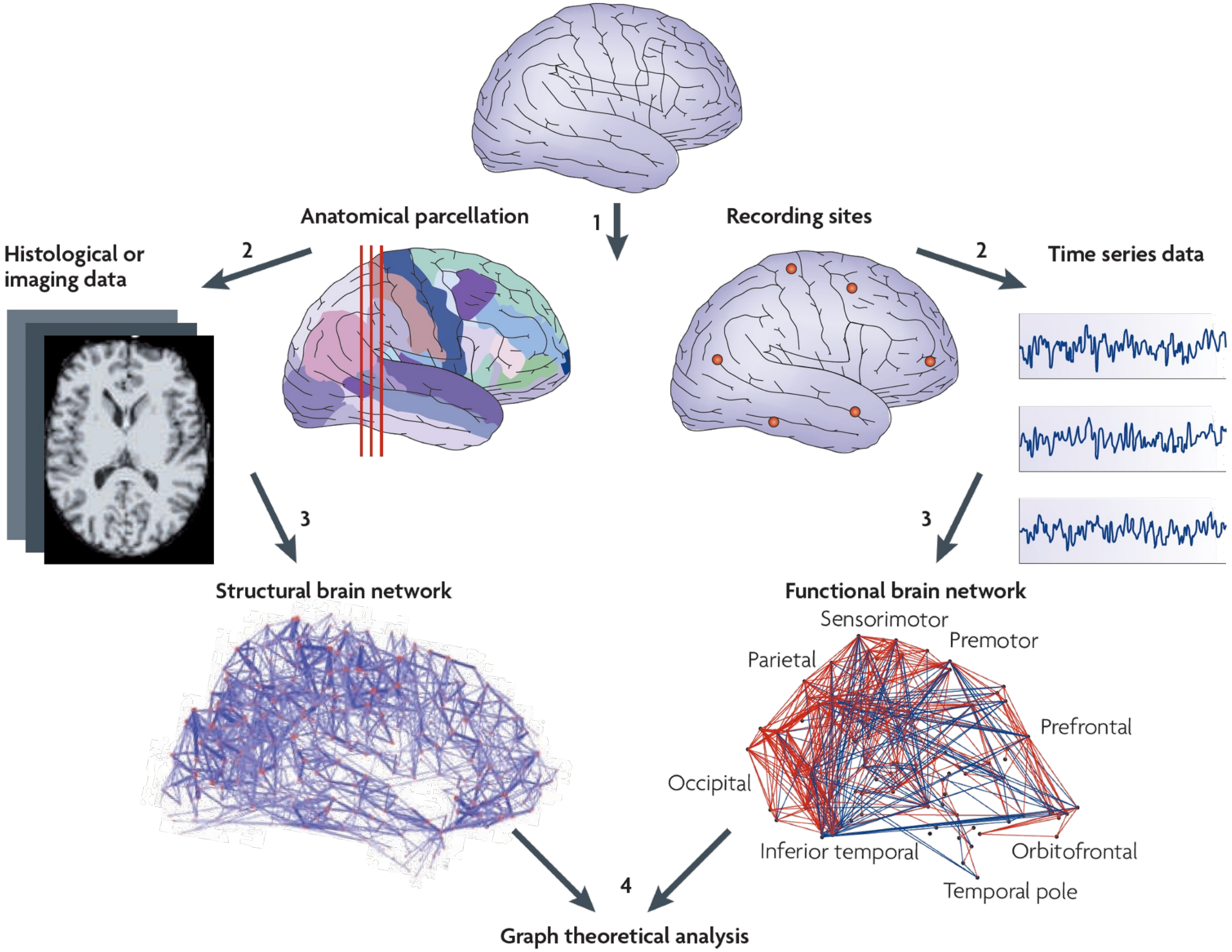
Minsky (1951)
*randomly wired "SNARC"*

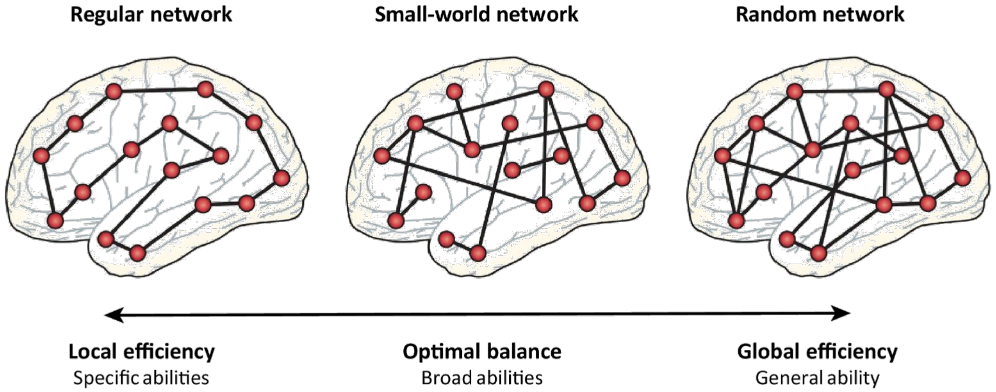


Rosenblatt (1959)
first Perceptron machine



# Neuroscience: brain networks are "small-world"?



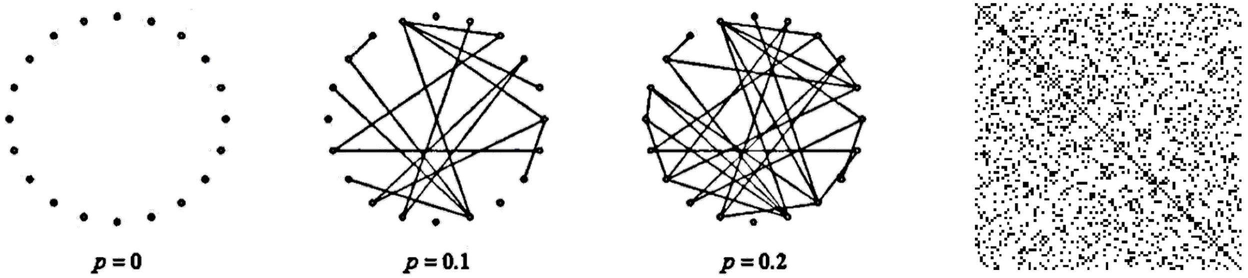


[Bassett, D.S. and Bullmore, E.D., 2006. Small-world brain networks. *The neuroscientist]*

[Bullmore, E. and Sporns, O., 2012. The economy of brain network organization. Nature reviews neuroscience]

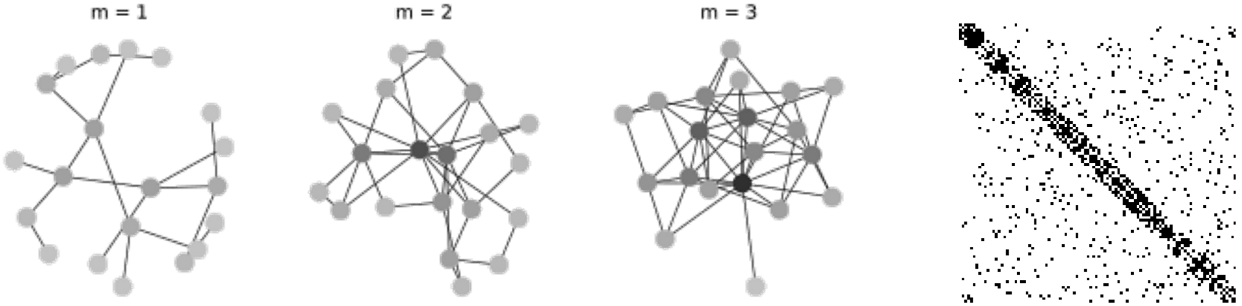
# Graph Theory

## Erdös-Rényi (ER), 1959



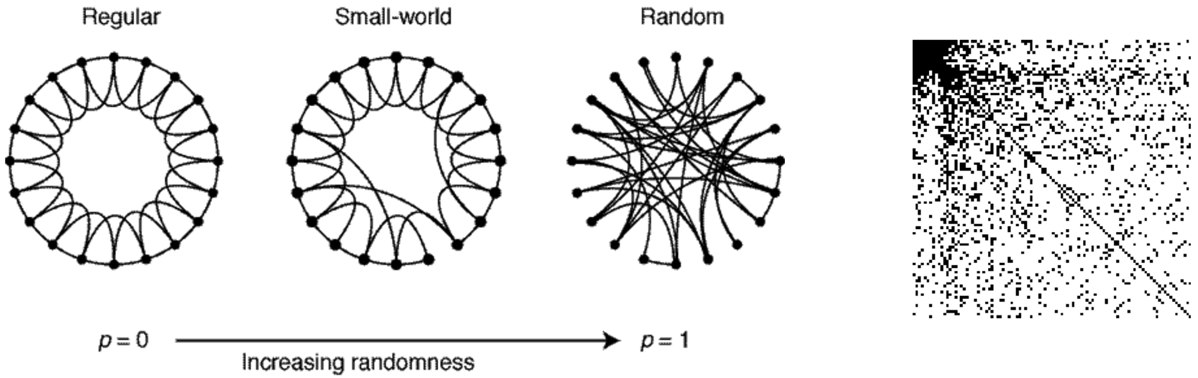

## Barabási–Albert (BA), 1998

“Scale-free” graphs




## Watts–Strogatz (WS), 1998

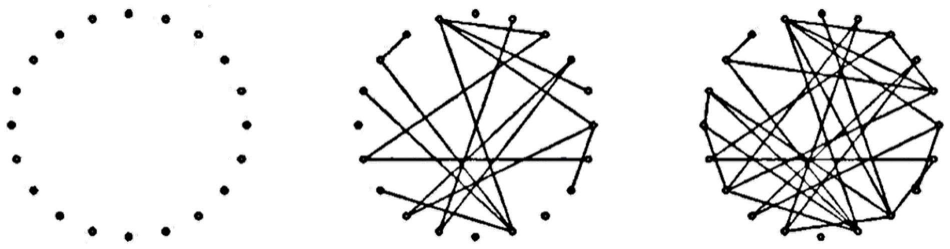
“Small-world” graphs



# Random Graph Models

## Erdös-Rényi (ER), 1959

- $G(n, p)$ model: $n$ nodes; each edge is included in the graph with probability $p$
- Degree distribution: $P(\deg(v) = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$
- Almost surely a single connected component if $p > \frac{\ln(n)}{n}$

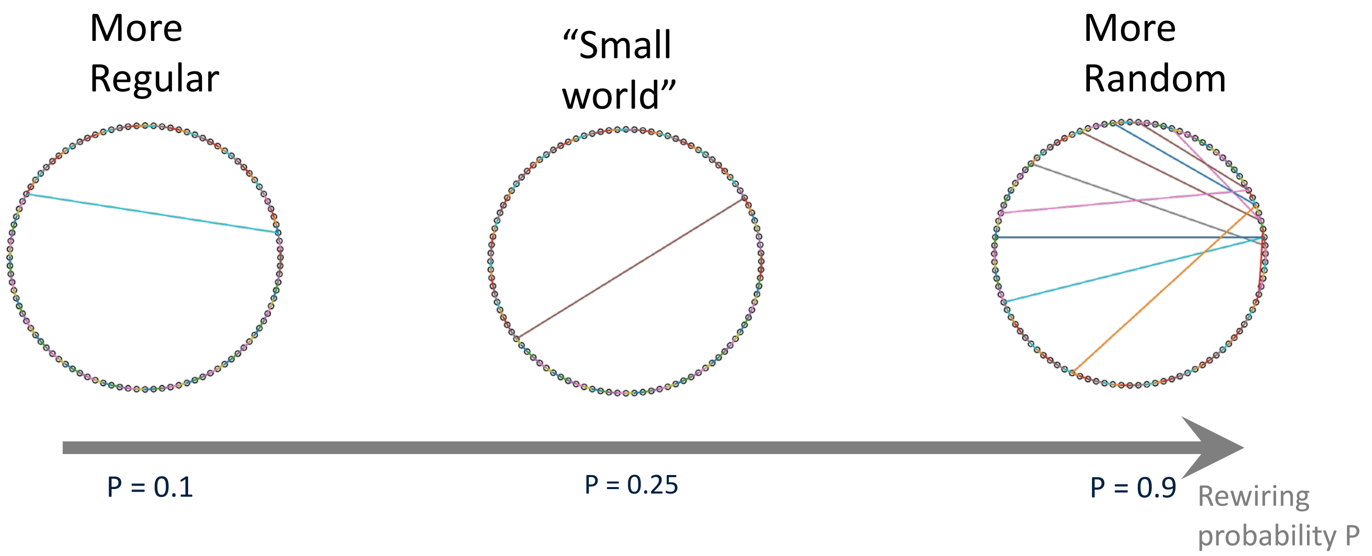


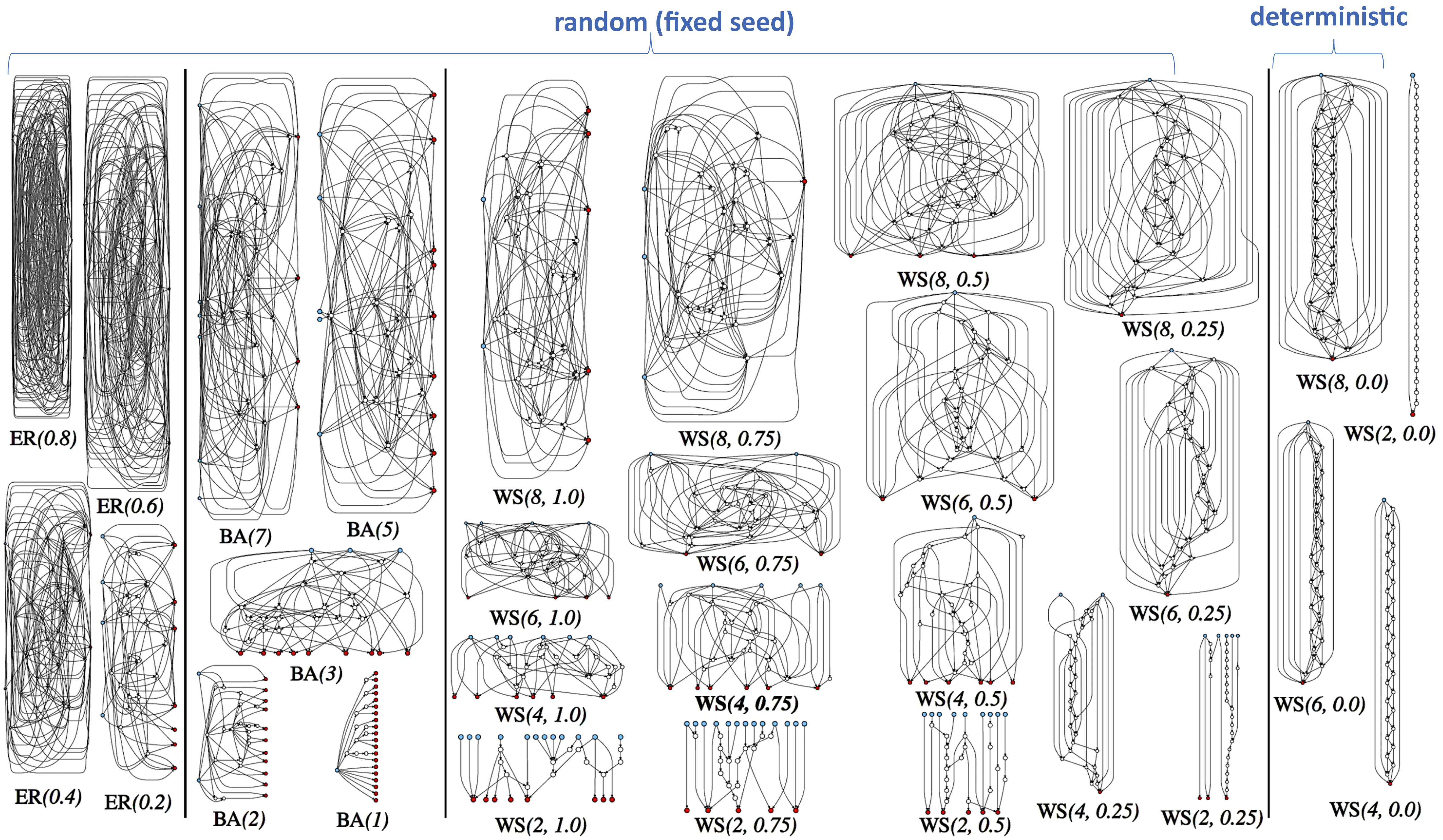
$p = 0$ $\quad$ $p = 0.1$ $\quad$ $p = 0.2$

# Random Graph Models

## Watts–Strogatz (WS), 1998

- *“Small World”* model: high clustering; formation of “Hubs”.
- N nodes regularly placed in a ring; connected to its $K/2$ neighbors on both sides.
- Then each edge is ***rewired*** with probability $p$. Randomness enables “*shortcuts*”.





random (fixed seed)
deterministic
ER(0.8)
ER(0.6)
ER(0.4)
ER(0.2)
BA(7)
BA(5)
BA(3)
BA(2)
BA(1)
WS(8, 1.0)
WS(6, 1.0)
WS(4, 1.0)
WS(2, 1.0)
WS(8, 0.75)
WS(6, 0.75)
WS(4, 0.75)
WS(2, 0.75)
WS(8, 0.5)
WS(6, 0.5)
WS(4, 0.5)
WS(2, 0.5)
WS(8, 0.25)
WS(6, 0.25)
WS(4, 0.25)
WS(2, 0.25)
WS(8, 0.0)
WS(2, 0.0)
WS(6, 0.0)
WS(4, 0.0)

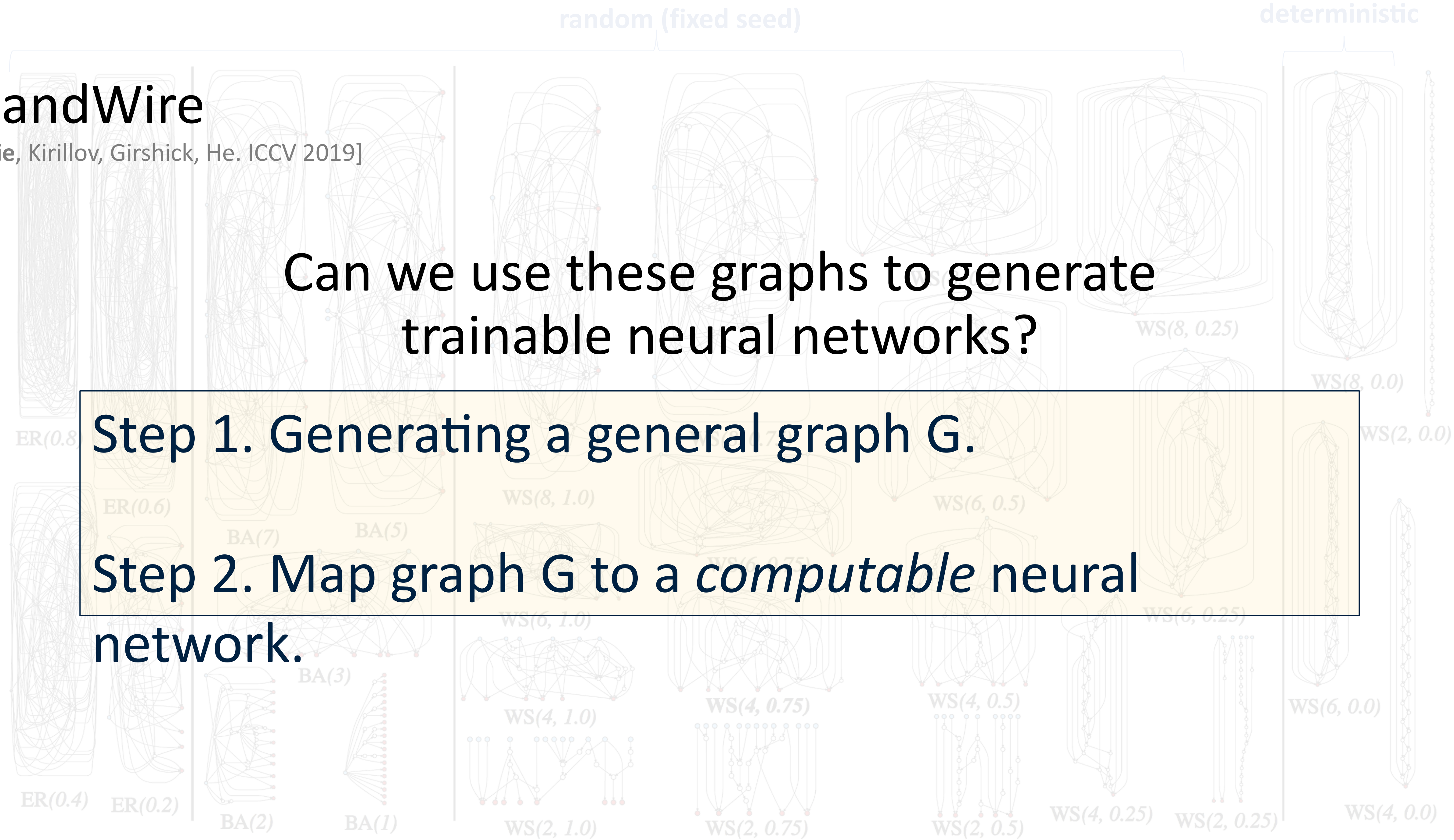
# RandWire

[**Xie**, Kirillov, Girshick, He. ICCV 2019]

Can we use these graphs to generate trainable neural networks?

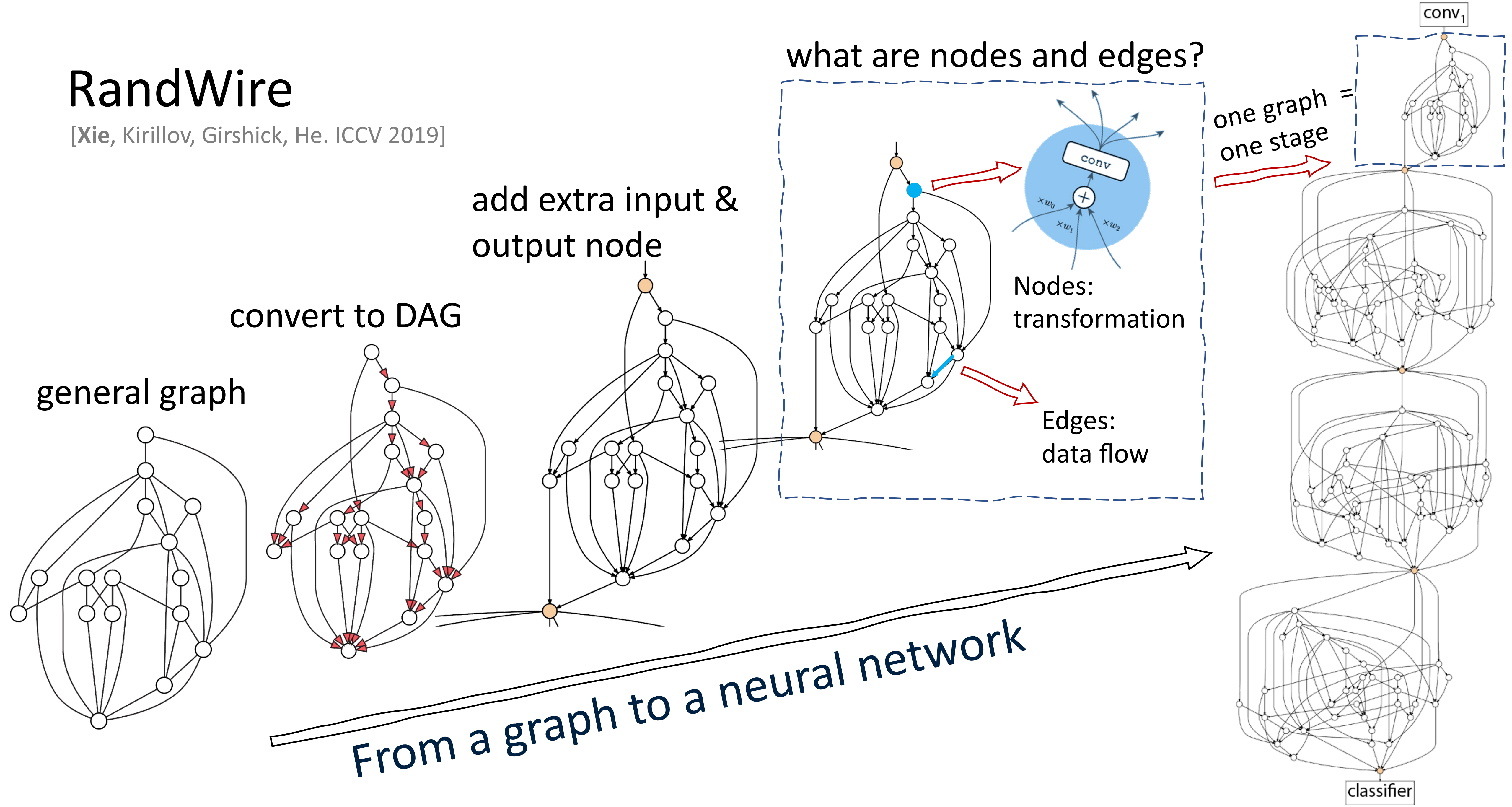
Step 1. Generating a general graph G.

Step 2. Map graph G to a *computable* neural network.



# RandWire

[**Xie**, Kirillov, Girshick, He. ICCV 2019]

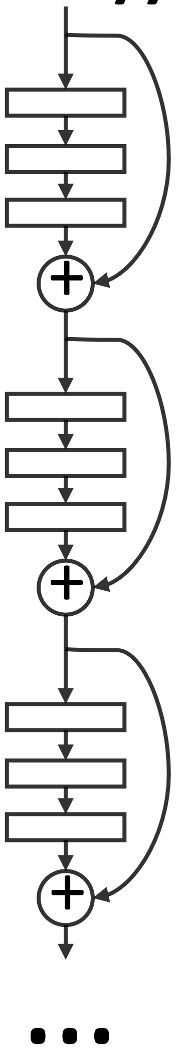


# ResNet-50

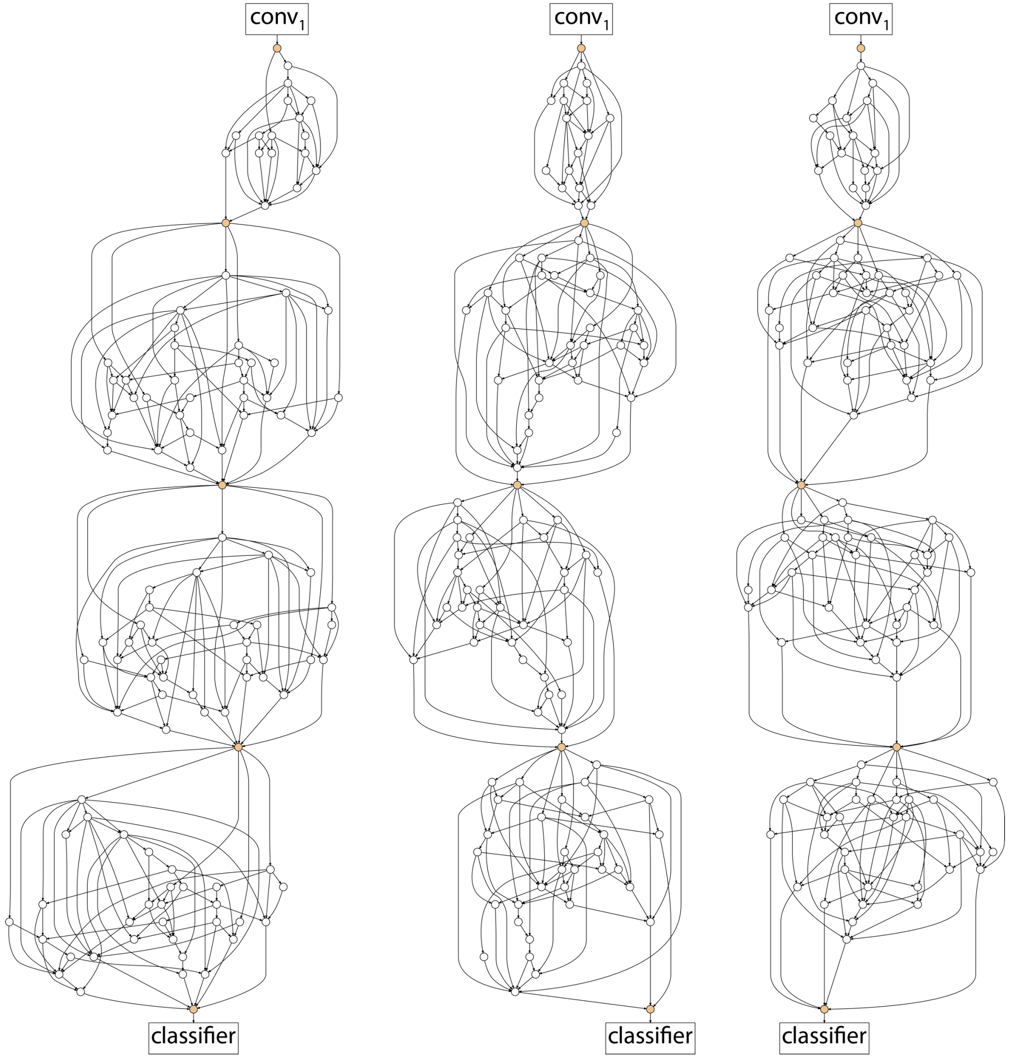
# RandWire (same FLOPs)

ImageNet Top1 Acc:

**77.1%**



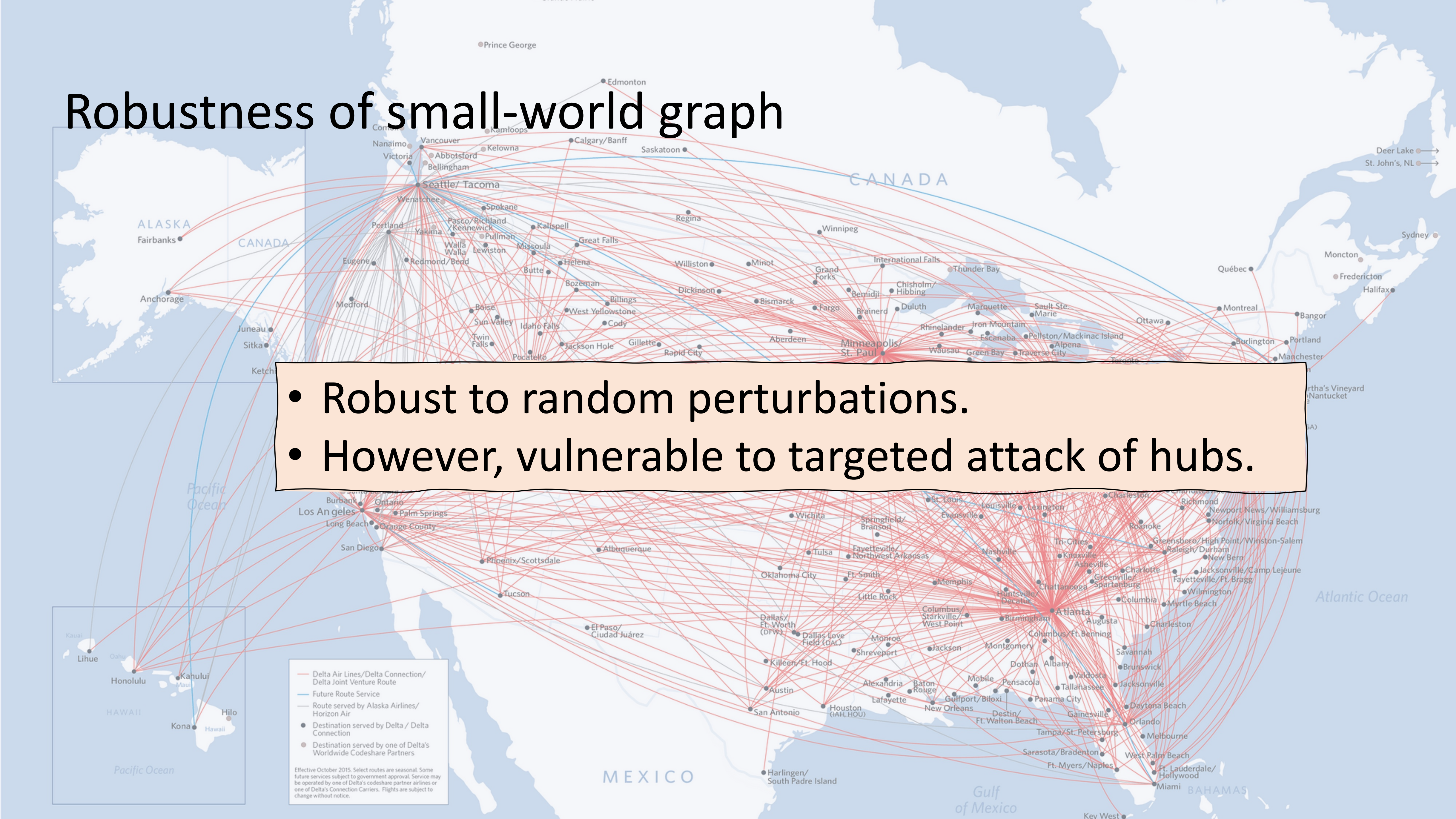
**79.1%** **79.1%** **79.0%**



# Results

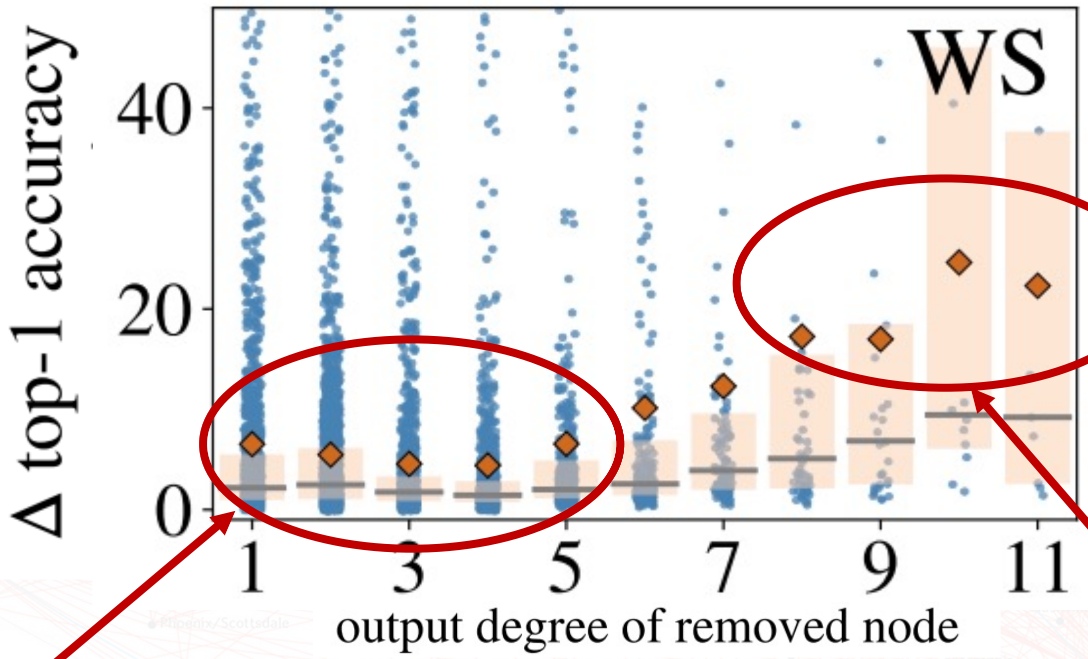
| | Network | Top-1 Acc | FLOPs |
|---|---|---|---|
| Hand designed | MobileNet | 70.6 | 569 |
| | MobileNet v2 | 74.7 | 585 |
| | ShuffleNet | 73.7 | 524 |
| | ShuffleNet v2 | 74.9 | 591 |
| AutoML searched | NASNet-A | 74.0 | 564 |
| | NASNet-C | 72.5 | 558 |
| | AmoebaNet-A | 74.5 | 555 |
| | AmoebaNet-C | 75.7 | 570 |
| | PNAS | 74.2 | 588 |
| | DARTS | 73.1 | 595 |
| | **RandWire-WS** | **74.7** $\pm0.25$ | **583** $\pm6.2$ |

# Robustness of small-world graph




- Robust to random perturbations.
- However, vulnerable to targeted attack of hubs.

# Graph Damage experiment




Delete a ***non-hub*** node:
< 10% accuracy drop

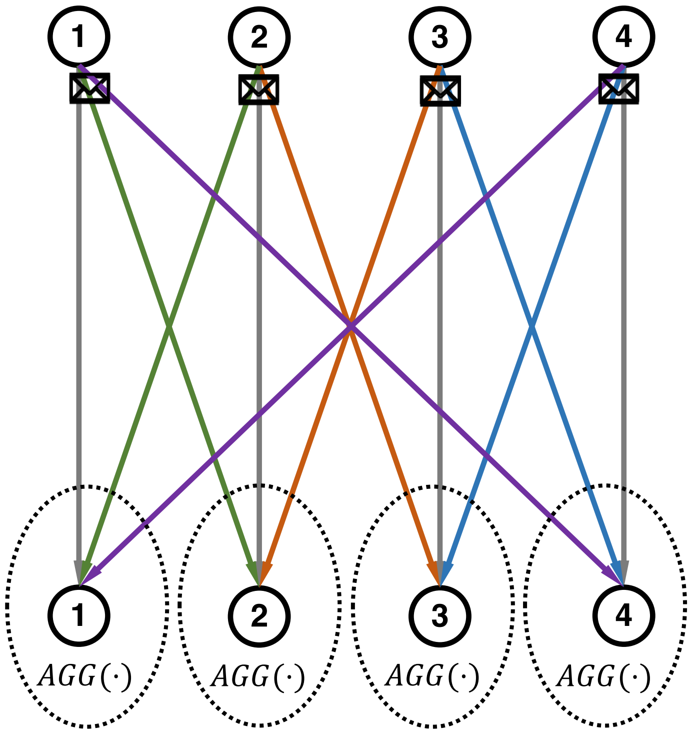
Delete a ***"hub"*** node:
20%-30% accuracy drop

# Neuron-level wiring: relational graph

[You, Leskovec, He and **Xie**. ICML 2020]

## Computational graphs

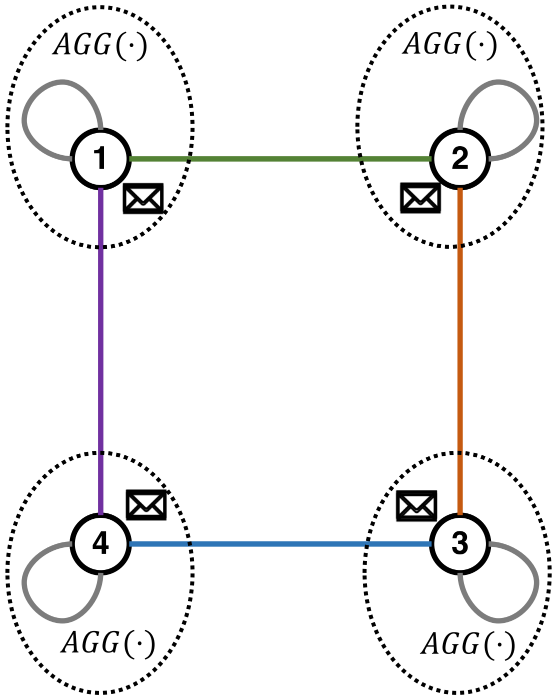
*Directed message flow*



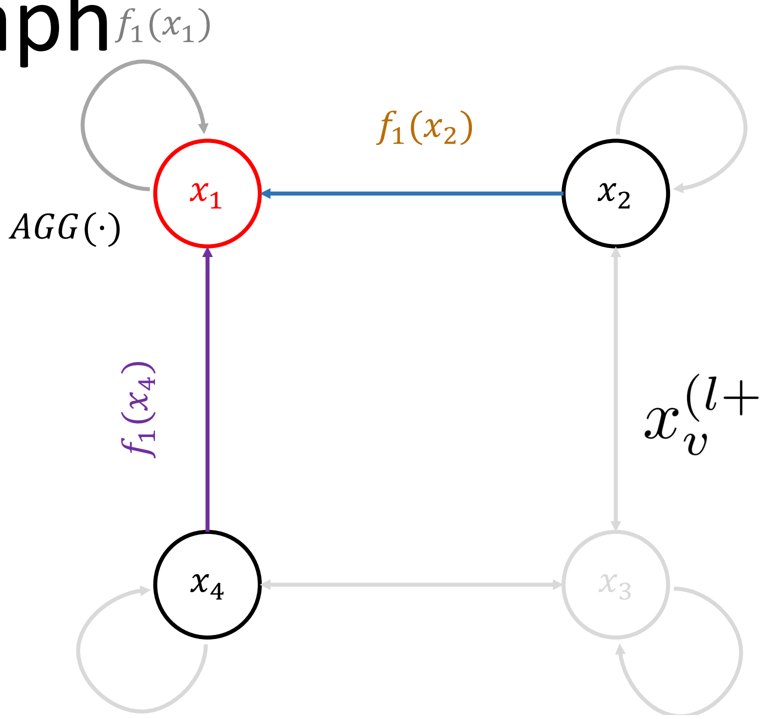

$\Longleftrightarrow$

## Relational graphs

*Bi-Directional message exchange*



# Relational graph




$$x_v^{(l+1)} = AGG^{(l)}(\{f_v^{(l)}(x_u^{(l)}), \forall u \in N(v)\})$$

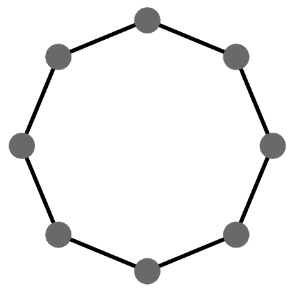
| 4 Key components | **Fixed-width MLP** | **Variable-width MLP** | **ResNet-34** |
|---|---|---|---|
| **Node feature $\mathbf{x}_i$** | Scalar: 1 dimension of data | Vector: multiple dimensions of data | Tensor: multiple channels of data |
| **Message function** $f_i(\cdot)$ | Scalar multiplication | (Non-square) matrix multiplication | 3×3 Conv |
| **Aggregation function** $AGG(\cdot)$ | $\sigma(\sum(\cdot))$ | $\sigma(\sum(\cdot))$ | $\sigma(\sum(\cdot))$ |
| **Number of rounds** $L$ | 1 round per layer | 1 round per layer | 34 rounds with residual connections |

# Graph structure measures

average path length ($L$) $$L = \sum_{s,t \in V} \frac{\text{shortest\_path}(s,t)}{n(n-1)}$$

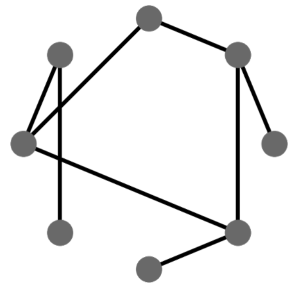
clustering coefficient ($C$) $$C_u = \frac{2T(u)}{\deg(u)(\deg(u)-1)}$$

WS graph
(n=8, k=2, p=0)



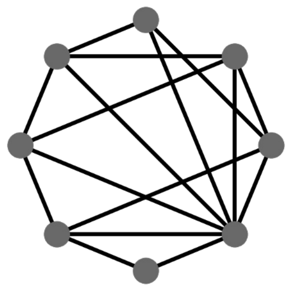
$L = 2.3$
$C = 0$

WS graph
(n=8, k=2, p=0.5)



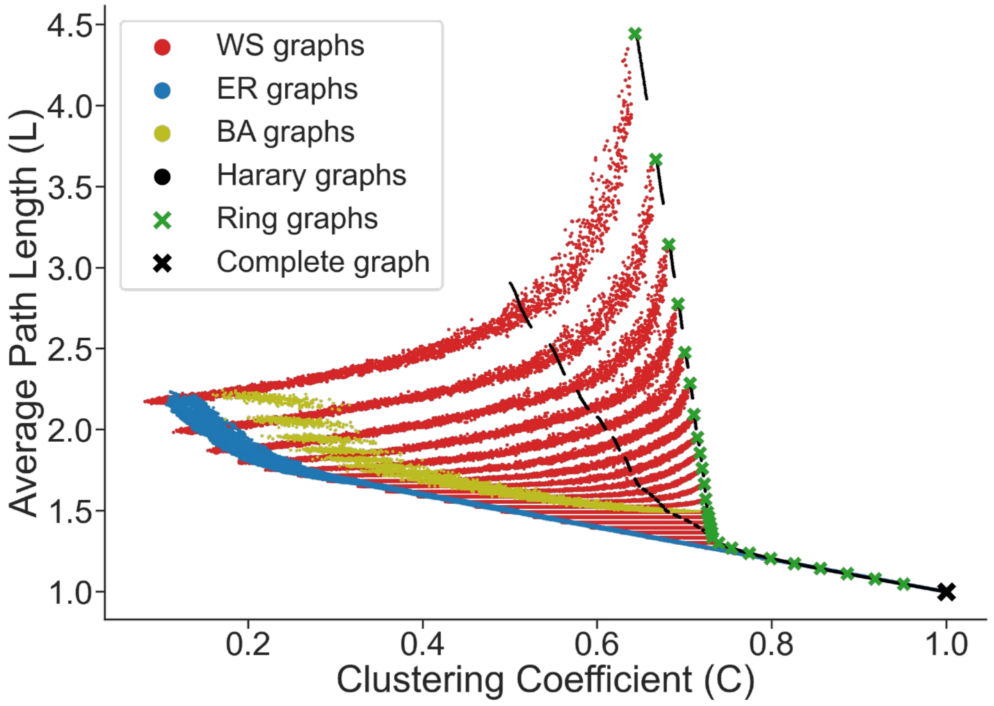
$L = 2.8$
$C = 0.3$

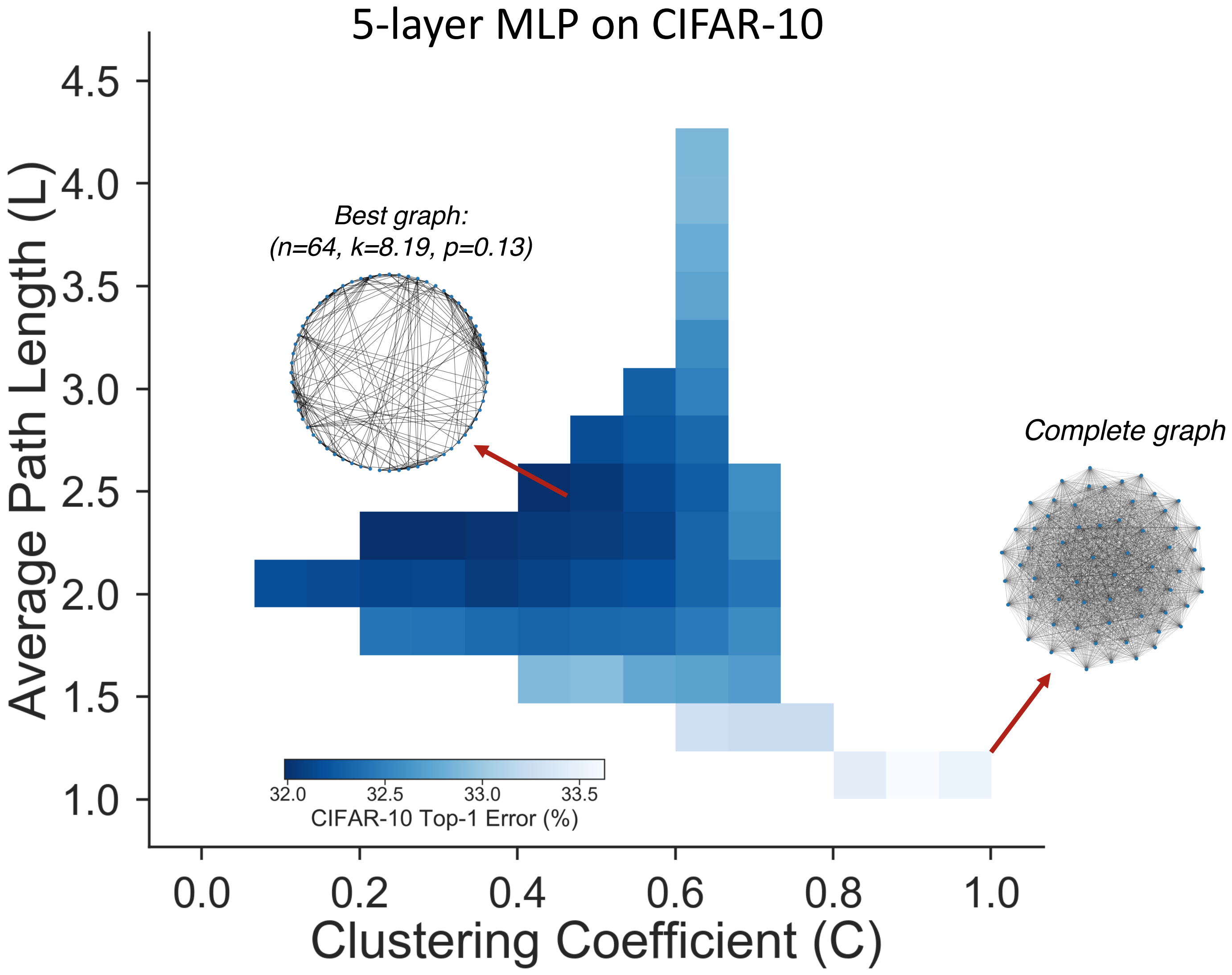
WS graph
(n=8, k=4, p=0.5)



$L = 1.4$
$C = 0.4$

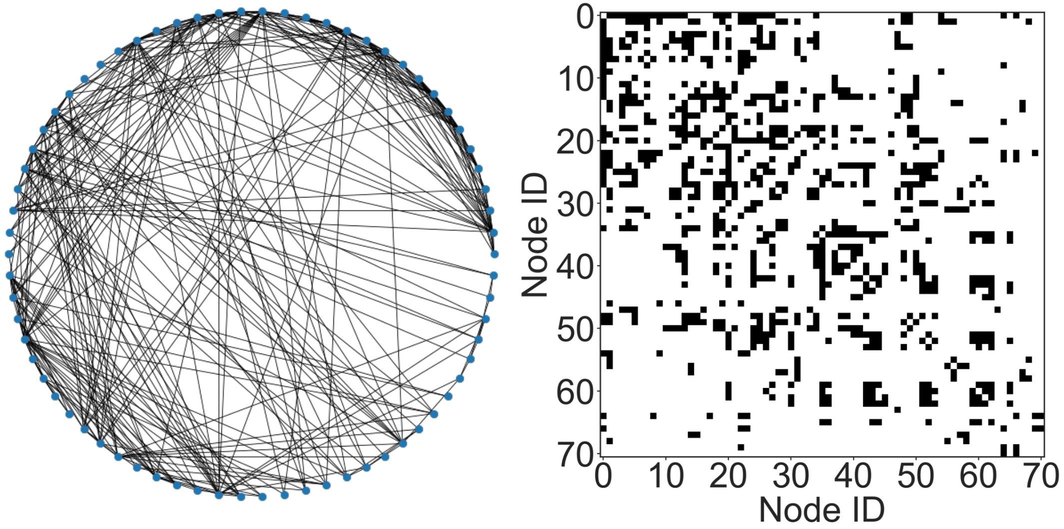
# Graph structure measures



## **Biological** neural network:
## *Macaque whole cortex*

Data from anatomical connectivity matrix [Bassett & Bullmore, 2006]

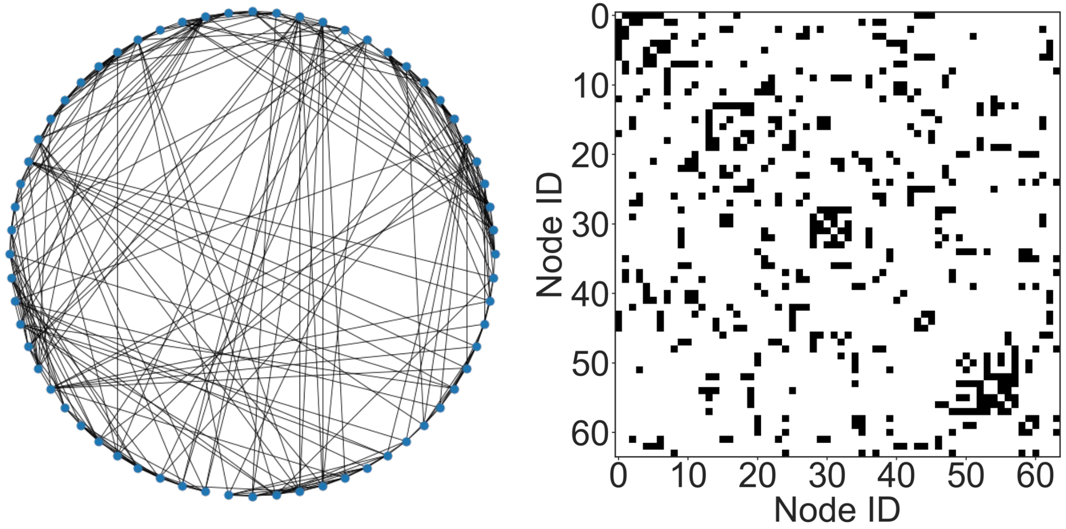



L = 2.38

C = 0.46

CIFAR-10: +0.5%

## **Artificial** neural network:
## *Best 5-layer MLP*

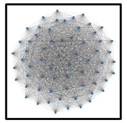



L = 2.48

C = 0.45

CIFAR-10: +1.2%

Baseline complete graph:
L = 1.0
C = 1.0
CIFAR-10: 66.7%

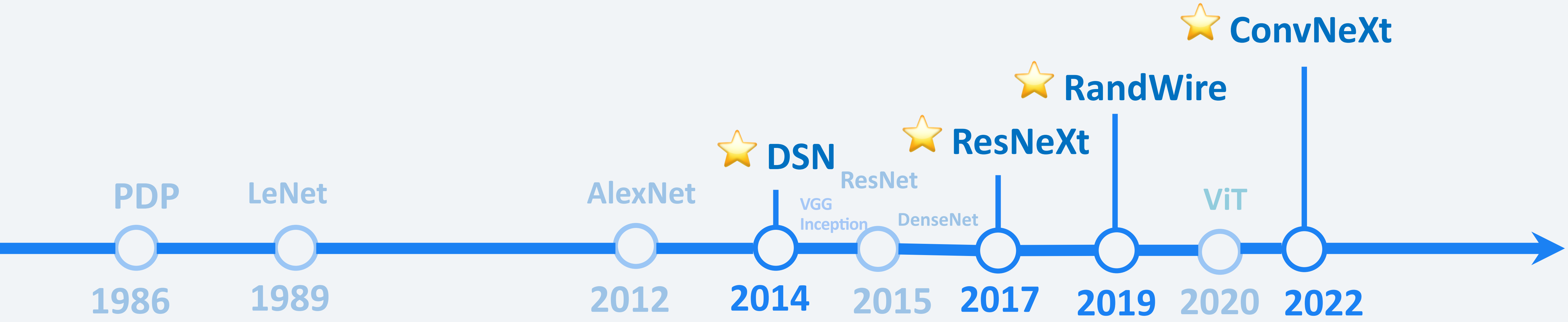


# Graph structure of NN

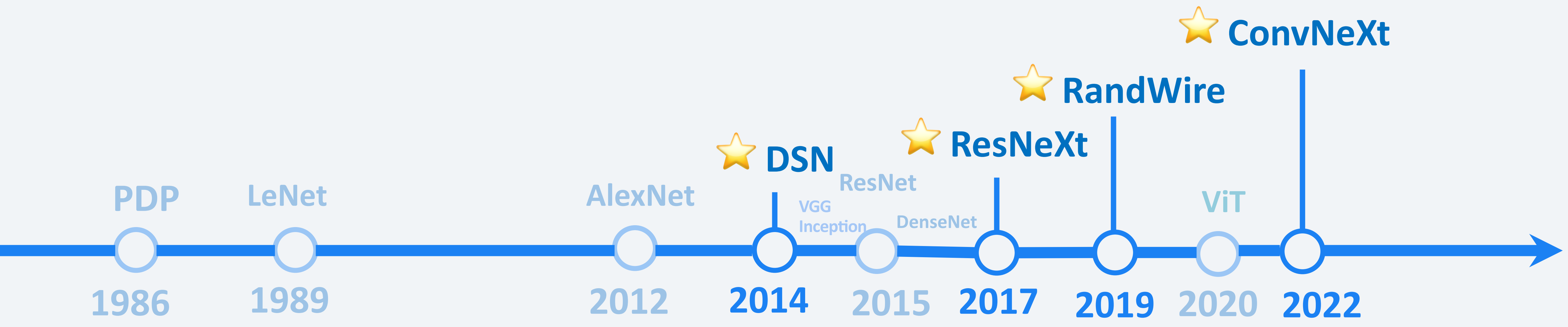
- Graph theory is a *common language*.
- Interests from neuroscience, physics, network science, complex systems, ...

[Analyzing Neural Networks Based on Random Graphs, Janik,and Nowak, arXiv 2020
[Van Essen et al., 2013]

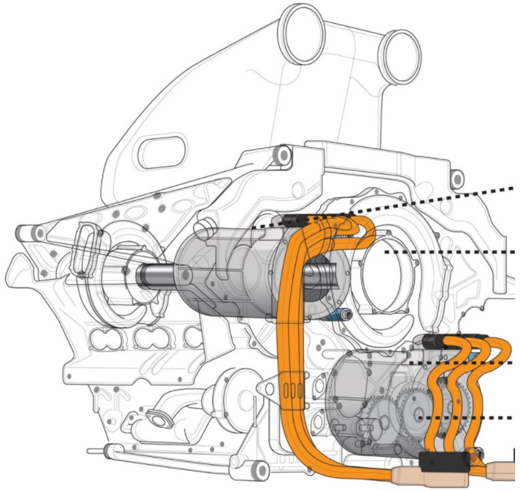
Takeaways: 1. Neural network design is important
2. Wiring pattern matters

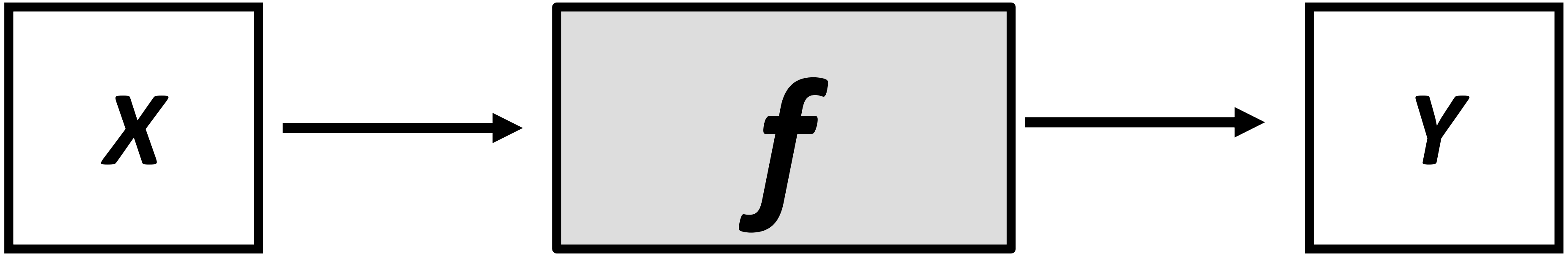


# Focused soely on supervised learning so far...



# Supervised Learning





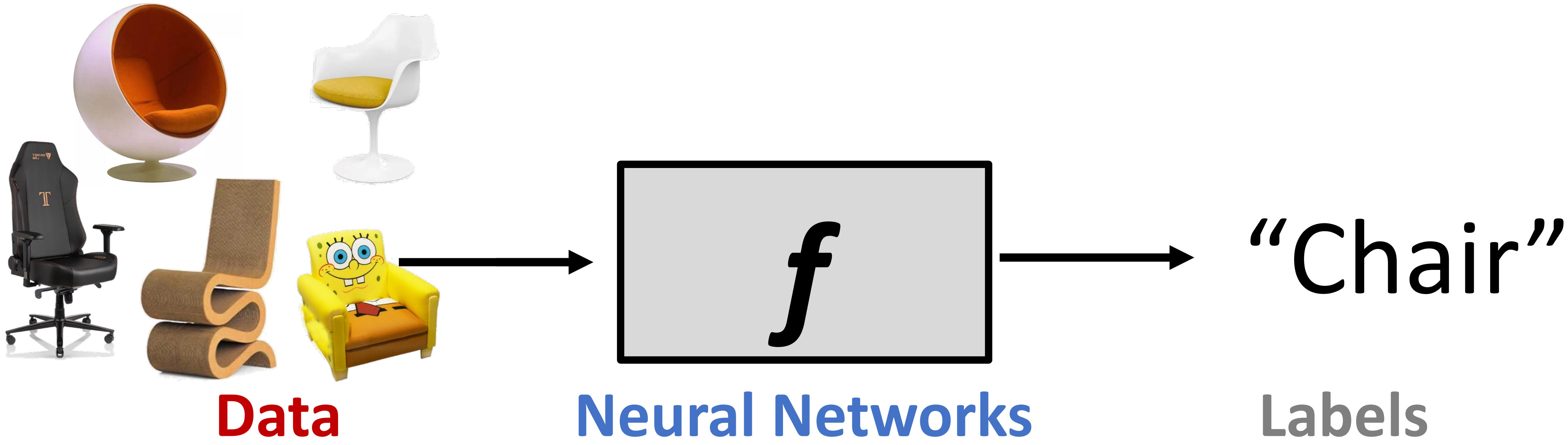

Data

Neural Networks

Labels

Illustration: Craig Scarborough, clipart-library.com

# Limitations of Supervised Learning

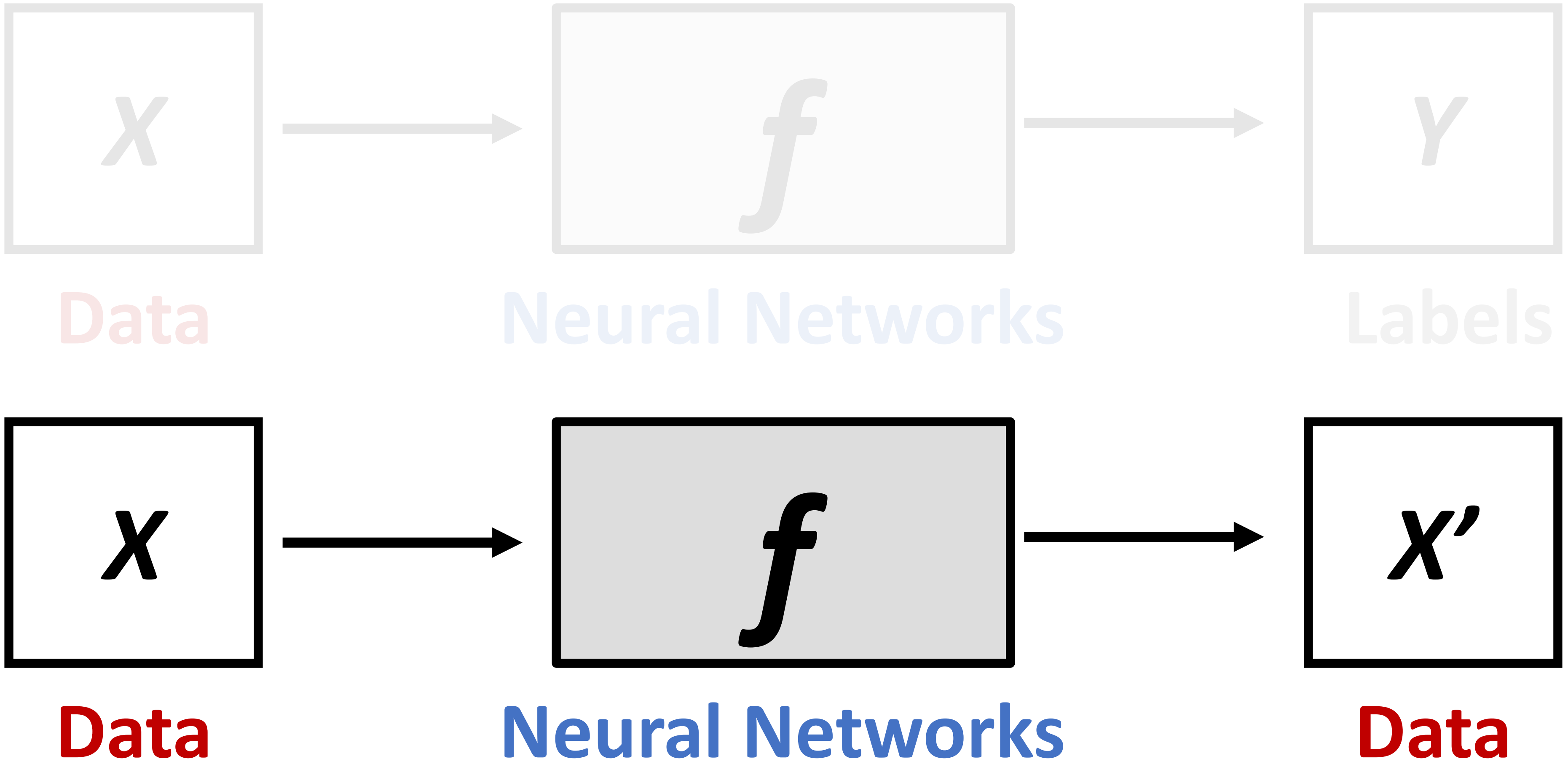



**Network might have to cheat:**

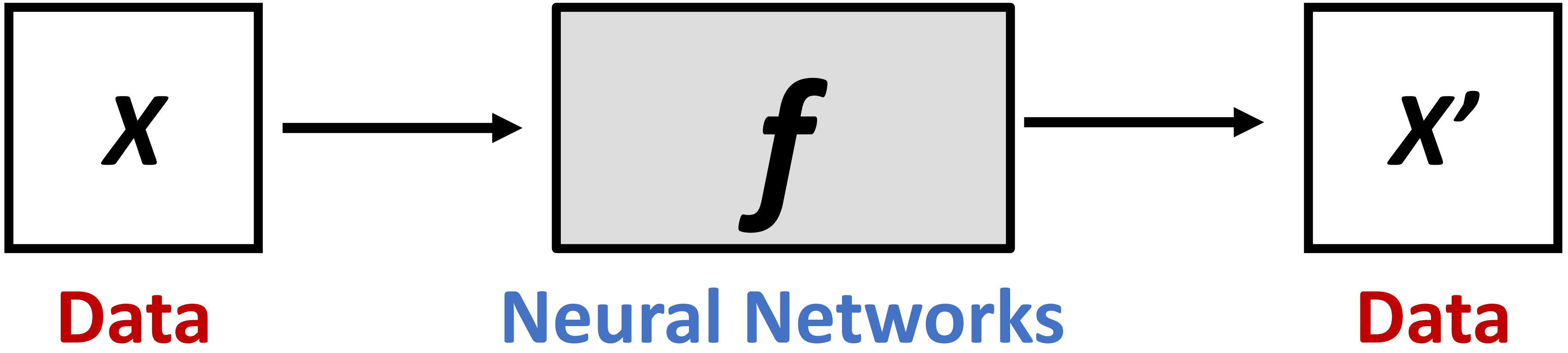
relying on spurious correlation / memorizing => poor generalization / not robust

https://www.mydomaine.com/chair-designs-51912

# Self-Supervised Learning



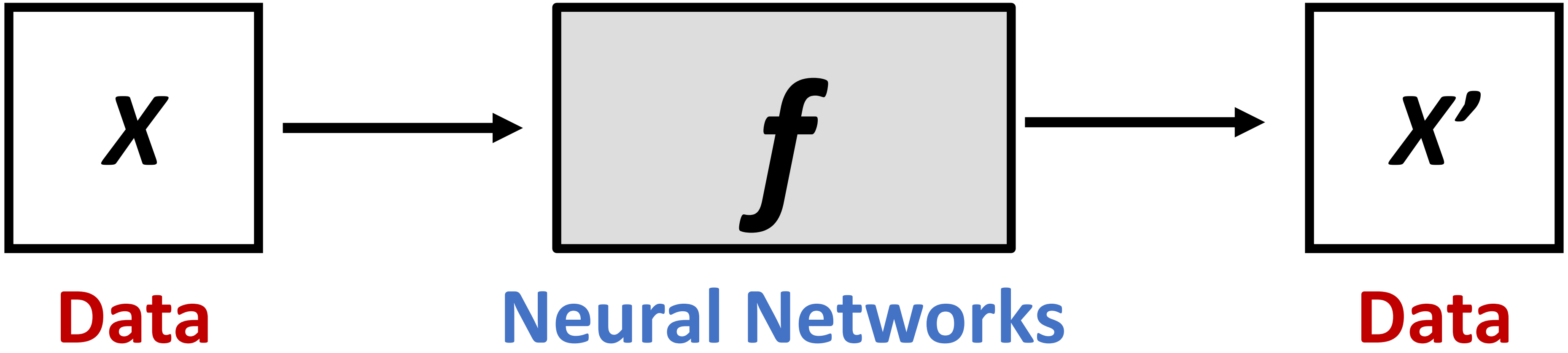
# Self-Supervised Learning




Instead of asking the machine what an object is, ask for:

- Similarity
- Correspondence
- Association
- Prediction
- Reconstruction

# Self-Supervised Learning

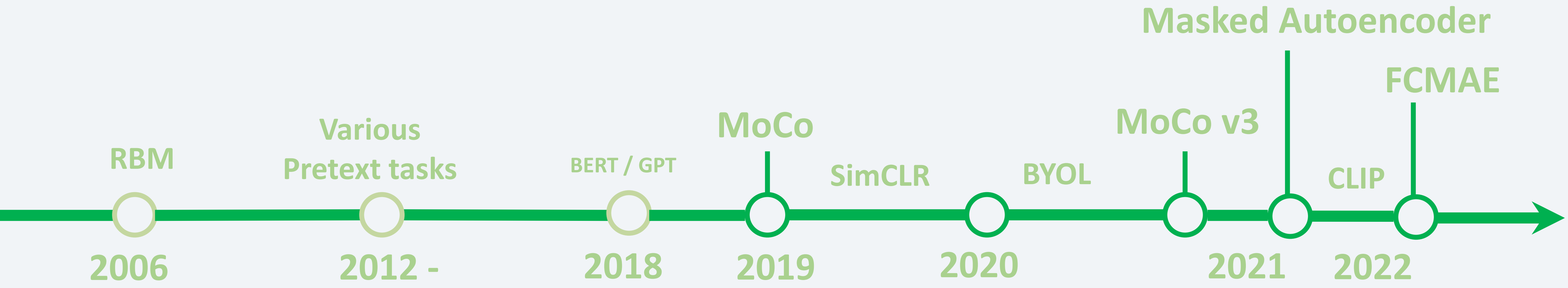



**Goal:**
To build background knowledge
and approximate a form of common sense in AI systems.

https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligen

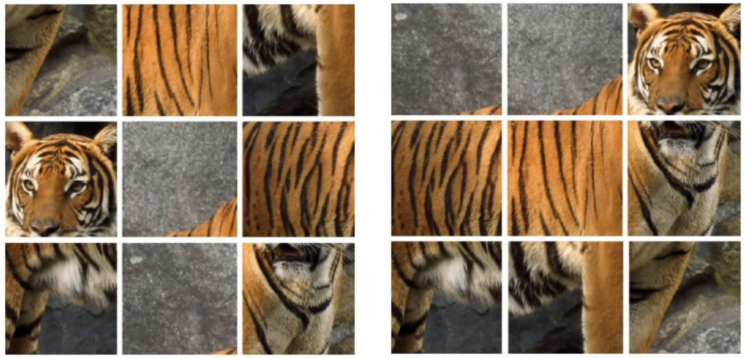
Architecture / Objective / Data

## 2. Training objectives beyond supervised classification: Are labels necessary?

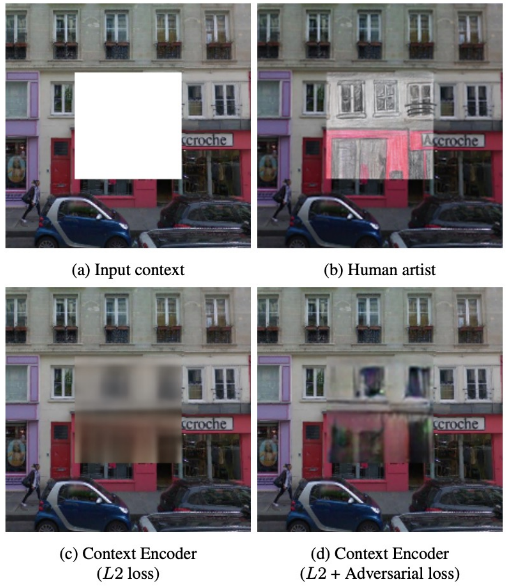


# Self-supervised Learning

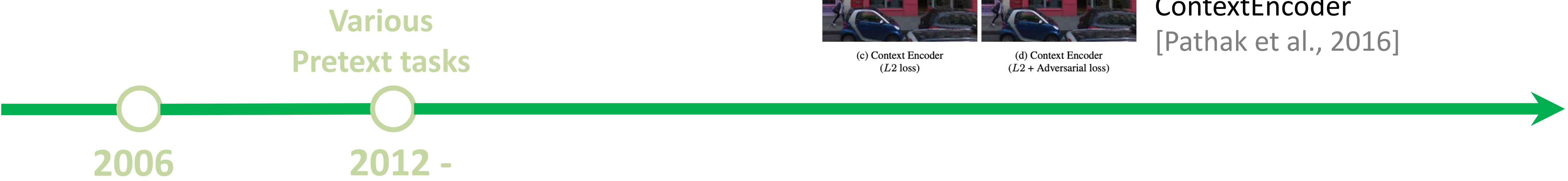
Colorization
[Zhang et al., 2016]



Jigsaw Puzzle
[Noroozi and Favaro, 2016]




ContextEncoder
[Pathak et al., 2016]



# ContextEncoder

[Pathak et al., 2016]

| Pretraining Method | Supervision | Pretraining time | Classification | Detection | Segmentation |
|---|---|---|---|---|---|
| ImageNet [26] | 1000 class labels | 3 days | 78.2% | 56.8% | 48.0% |
| Random Gaussian | initialization | < 1 minute | 53.3% | 43.4% | 19.8% |
| Autoencoder | - | 14 hours | 53.8% | 41.9% | 25.2% |
| Agrawal *et al.* [1] | egomotion | 10 hours | 52.9% | 41.8% | - |
| Wang *et al.* [39] | motion | 1 week | 58.7% | 47.4% | - |
| Doersch *et al.* [7] | relative context | 4 weeks | 55.3% | 46.6% | - |
| Ours | context | 14 hours | 56.5% | 44.5% | 30.0% |

-10% to - 20% compared to ImageNet supervised pre-training
Labels seemed necessary…

Various
Pretext tasks

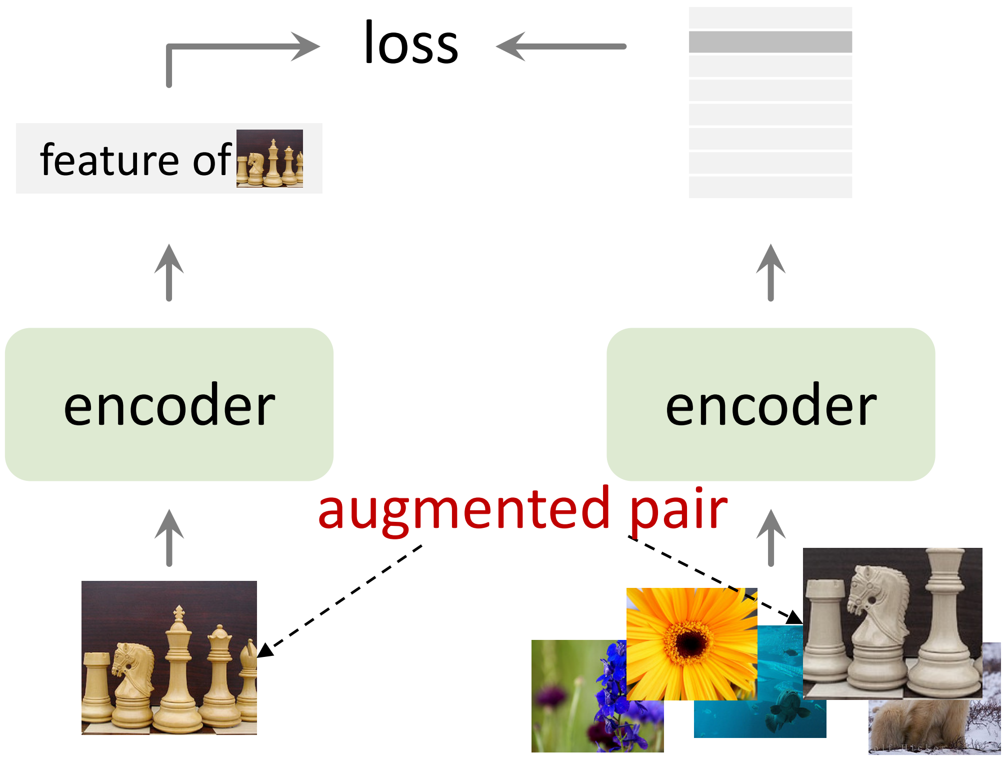
2006

2012 -

# Contrastive Learning

[Chopra *et al*. CVPR 2005]
[Hadsell *et al*. CVPR 2006]
[Wu *et al*. CVPR 2018]
...

## Goal: similarity learning

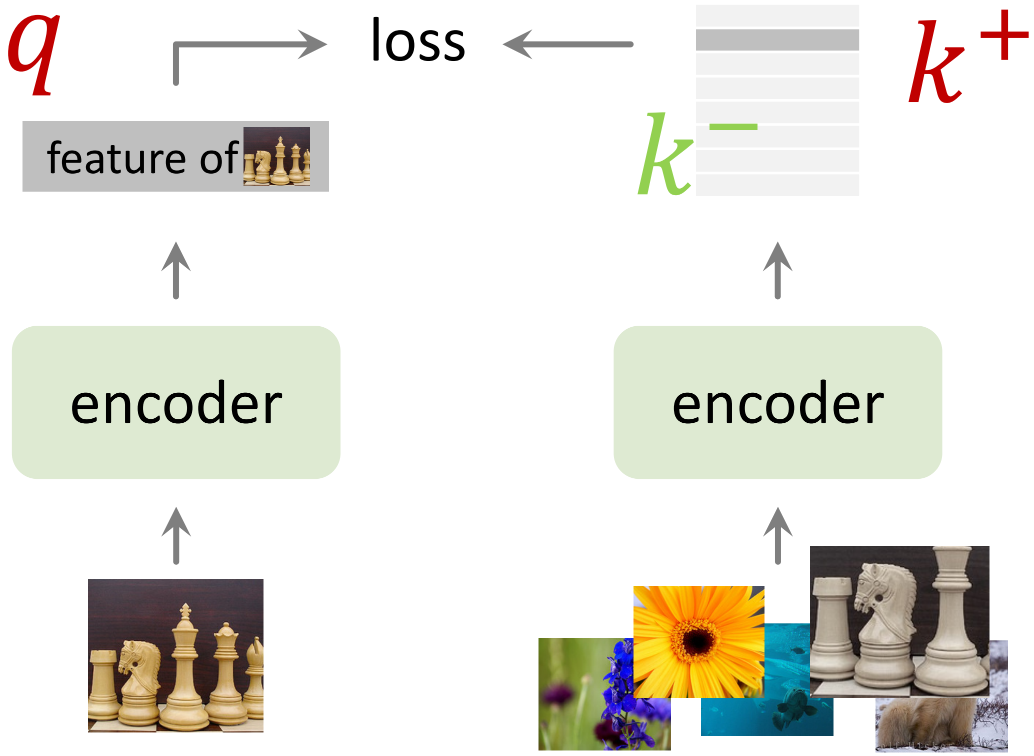



# Contrastive Learning

[Chopra *et al*. CVPR 2005]
[Hadsell *et al*. CVPR 2006]
[Wu *et al*. CVPR 2018]
…

# Contrastive Learning

[Chopra *et al*. CVPR 2005]
[Hadsell *et al*. CVPR 2006]
[Wu *et al*. CVPR 2018]
...




# Contrastive Learning

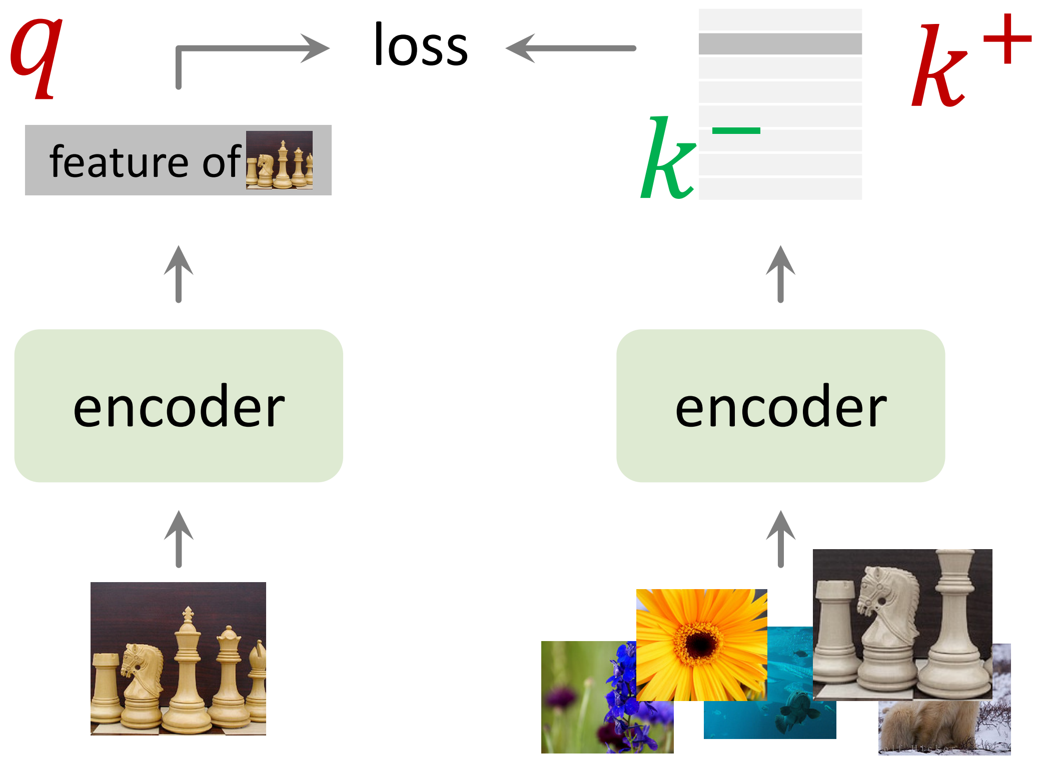
[Chopra *et al*. CVPR 2005]
[Hadsell *et al*. CVPR 2006]
[Wu *et al*. CVPR 2018]
…

InfoNCE Loss:

$$\mathcal{L}_q = -\log \frac{\exp(q{\cdot}k^+/\tau)}{\exp(q{\cdot}k^+/\tau) + \sum_{k^-} \exp(q{\cdot}k^-/\tau)}.$$

To learn features that

- attract similar samples ($q$ and $k^+$)
- dispel dissimilar samples ($q$ and many $k^-$).





$\tau$ is a hyperparameter for temperature

# Momentum Contrast

[He, Fan, Wu, **Xie**, Girshick. CVPR 2020]

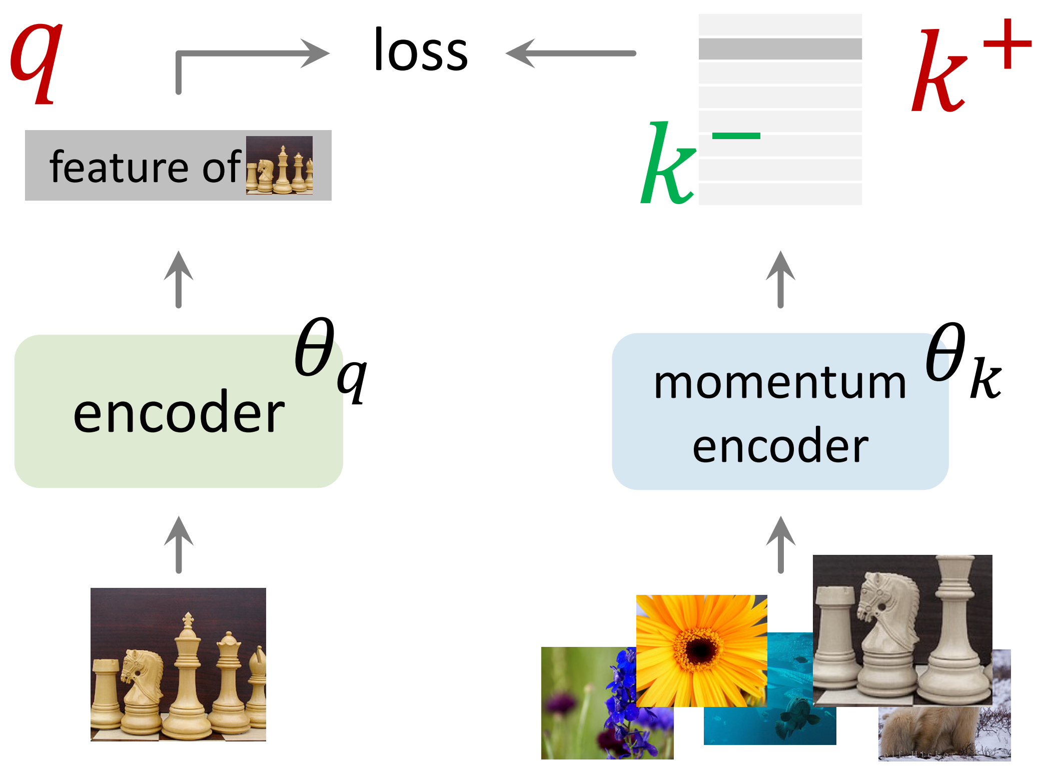
Momentum Encoder:

$$\theta_k \coloneqq m \cdot \theta_k + (1 - m) \cdot \theta_q$$

slowly update, e.g.
$m = 0.999$

Enables:

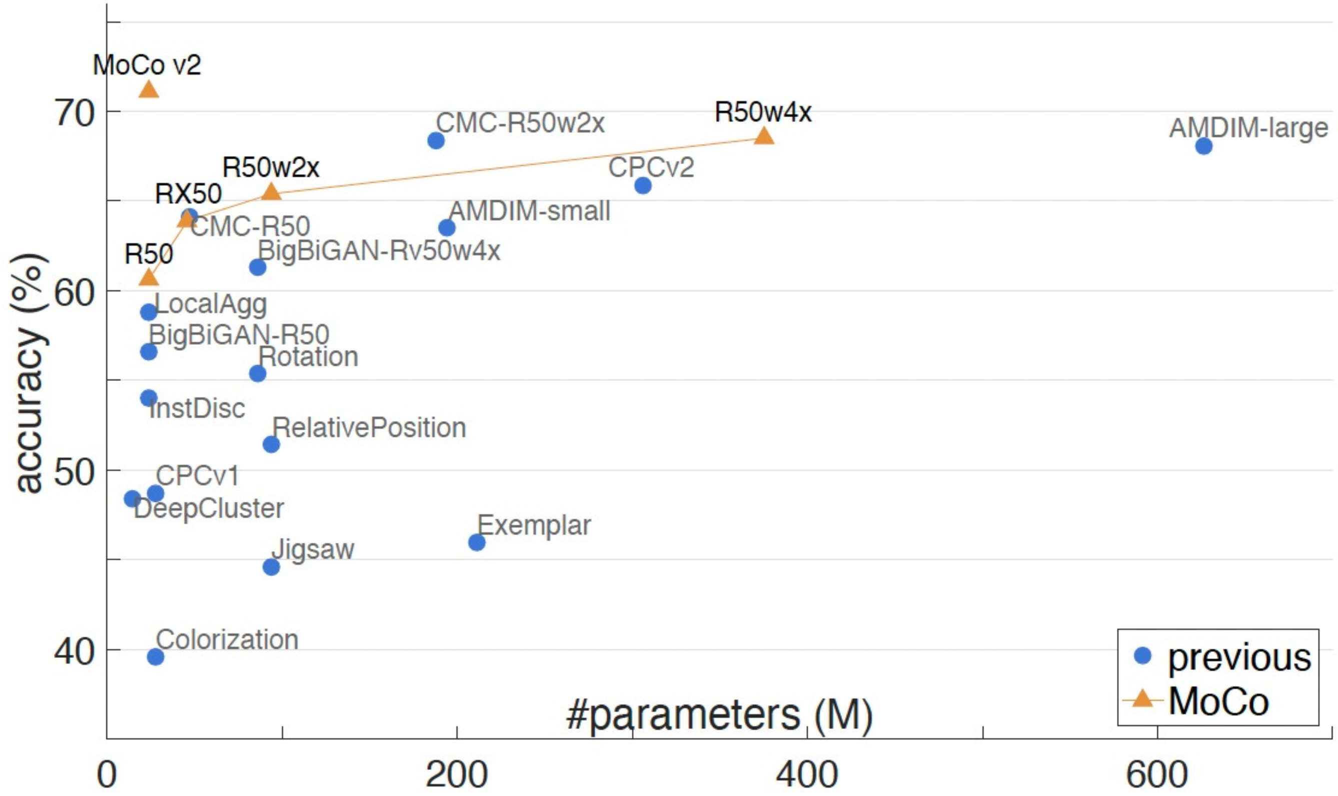
- Large feature queues
- Consistent features




# Momentum Contrast

[He, Fan, Wu, **Xie**, Girshick. CVPR 2020]

ImageNet
Linear evaluation




| pre-train | $AP_{50}$ | AP | $AP_{75}$ |
|---|---|---|---|
| random init. | 64.4 | 37.9 | 38.6 |
| super. IN-1M | 81.4 | 54.0 | 59.1 |
| **MoCo** IN-1M | 81.1 (−0.3) | 54.6 (+0.6) | 59.9 (+0.8) |
| **MoCo** IG-1B | 81.6 (+0.2) | 55.5 (+1.5) | 61.2 (+2.1) |

(a) Faster R-CNN, R50-**dilated-C5**

| pre-train | $AP_{50}$ | AP | $AP_{75}$ |
|---|---|---|---|
| random init. | 60.2 | 33.8 | 33.1 |
| super. IN-1M | 81.3 | 53.5 | 58.8 |
| **MoCo** IN-1M | 81.5 (+0.2) | 55.9 (+2.4) | 62.6 (+3.8) |
| **MoCo** IG-1B | 82.2 (+0.9) | 57.2 (+3.7) | 63.7 (+4.9) |

(b) Faster R-CNN, R50-**C4**

| $AP^{bb}$ | $AP^{bb}_{50}$ | $AP^{bb}_{75}$ | $AP^{mk}$ | $AP^{mk}_{50}$ | $AP^{mk}_{75}$ |
|---|---|---|---|---|---|
| 36.7 | 56.7 | 40.0 | 33.7 | 53.8 | 35.9 |
| 40.6 | 61.3 | 44.4 | 36.8 | 58.1 | 39.5 |
| 40.8 (+0.2) | 61.6 (+0.3) | 44.7 (+0.3) | 36.9 (+0.1) | 58.4 (+0.3) | 39.7 (+0.2) |
| 41.1 (+0.5) | 61.8 (+0.5) | 45.1 (+0.7) | 37.4 (+0.6) | 59.1 (+1.0) | 40.2 (+0.7) |

(b) Mask R-CNN, R50-**FPN**, **2**× schedule

| $AP^{bb}$ | $AP^{bb}_{50}$ | $AP^{bb}_{75}$ | $AP^{mk}$ | $AP^{mk}_{50}$ | $AP^{mk}_{75}$ |
|---|---|---|---|---|---|
| 35.6 | 54.6 | 38.2 | 31.4 | 51.5 | 33.5 |
| 40.0 | 59.9 | 43.1 | 34.7 | 56.5 | 36.9 |
| 40.7 (+0.7) | 60.5 (+0.6) | 44.1 (+1.0) | 35.4 (+0.7) | 57.3 (+0.8) | 37.6 (+0.7) |
| 41.1 (+1.1) | 60.7 (+0.8) | 44.8 (+1.7) | 35.6 (+0.9) | 57.4 (+0.9) | 38.1 (+1.2) |

(d) Mask R-CNN, R50-**C4**, **2**× schedule

VOC 07+12 Detection

surpass, +4.9 $AP_{75}$

COCO Detection

COCO Instance seg.

surpass

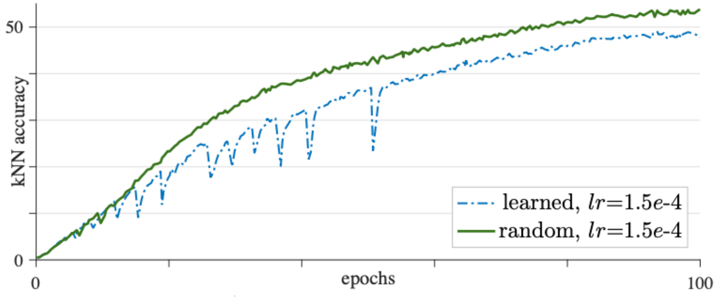
Self-supervised pre-training: surpass supervised counterparts in 7 vision tasks

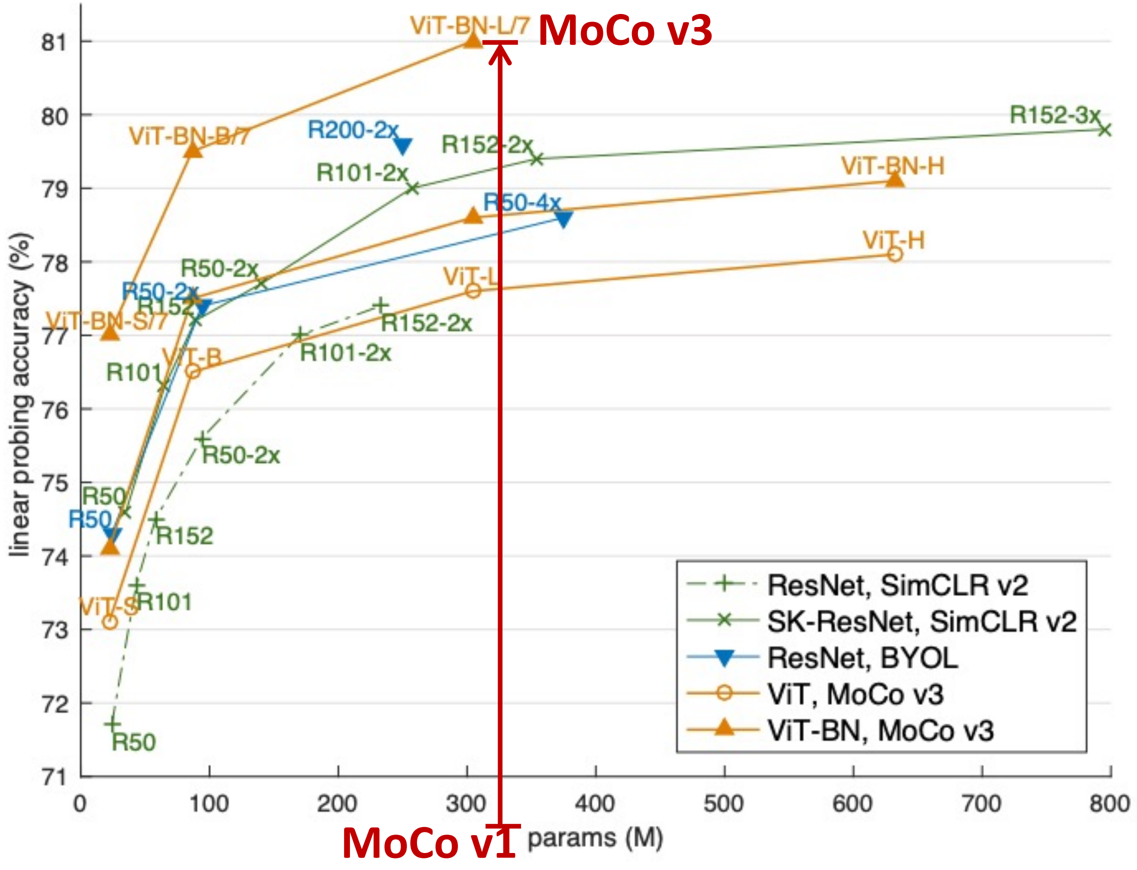
# MoCo v1 to v3

[Chen*, **Xie***, He, ICCV 2021]

- Using bigger models
- Better training techniques

- Stabilize training with *random patch projection*

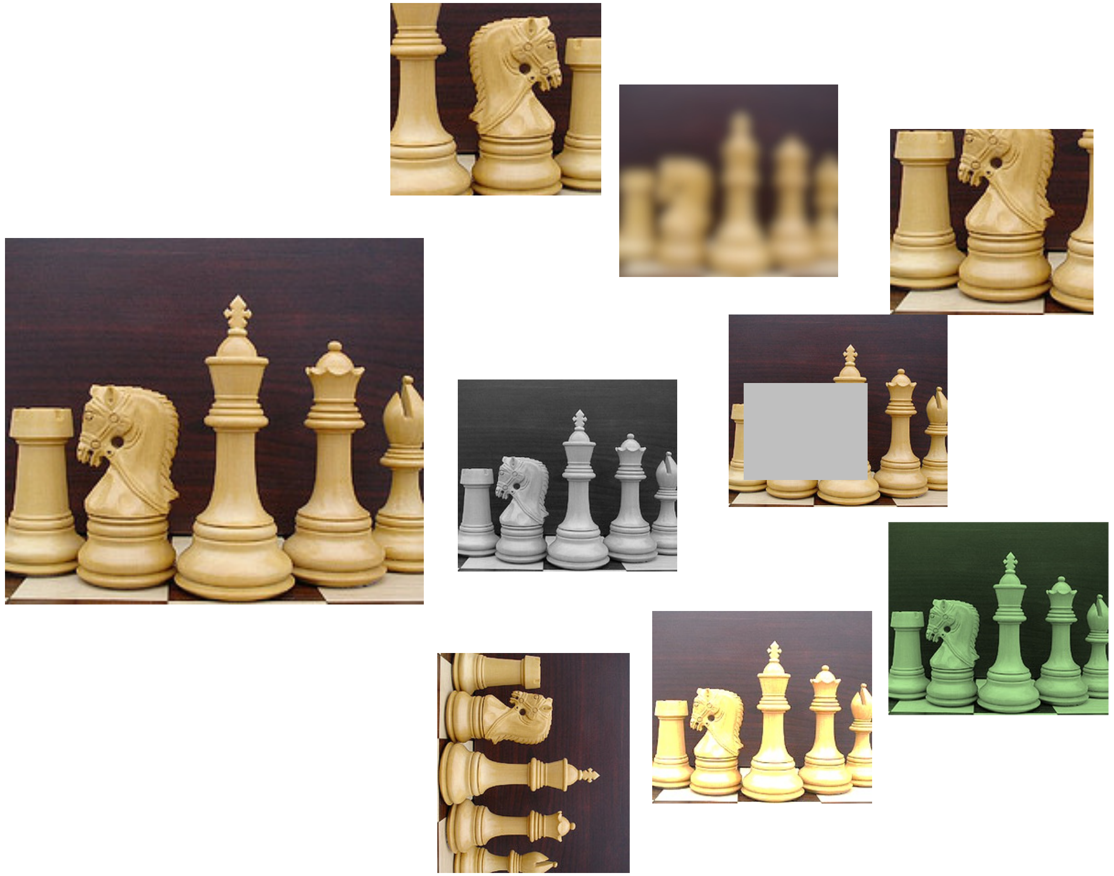



# Contrastive Learning

Rely on:

- hand-designed *data augmentations*
- “Too easy to solve” otherwise

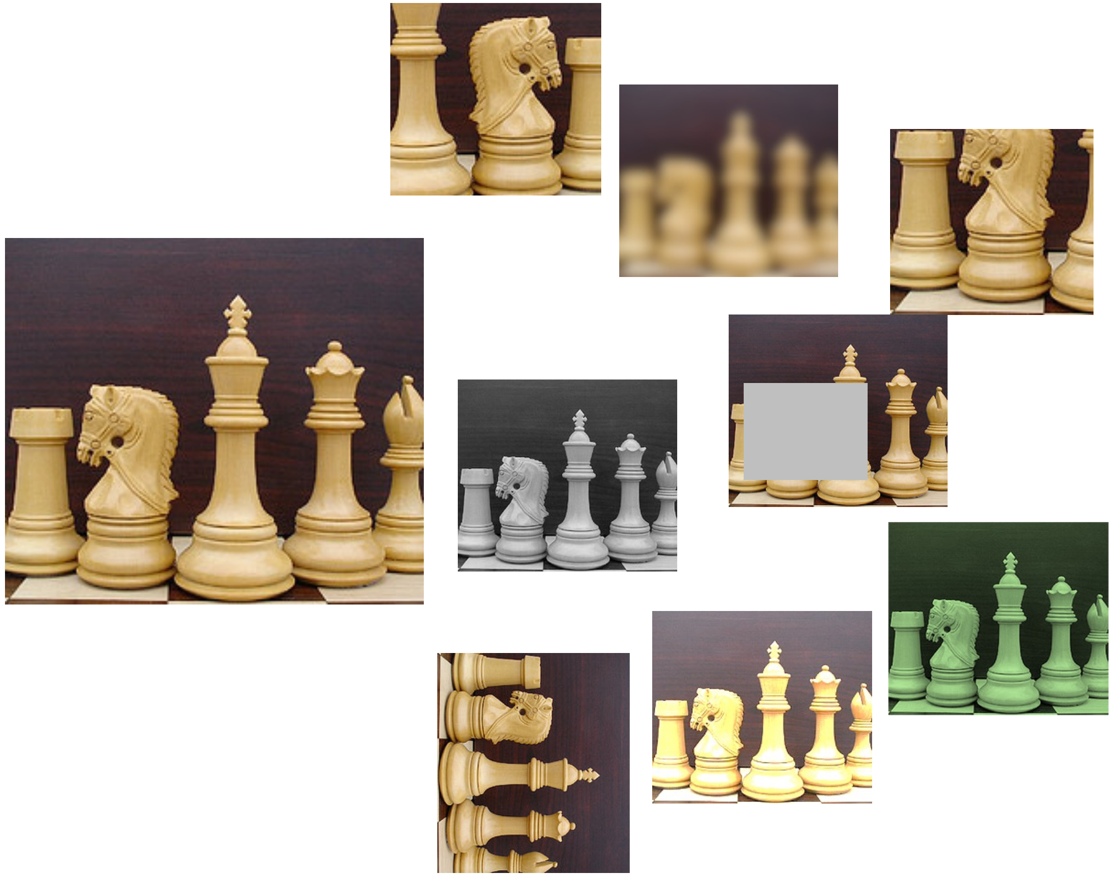


# Contrastive Learning

Rely on:

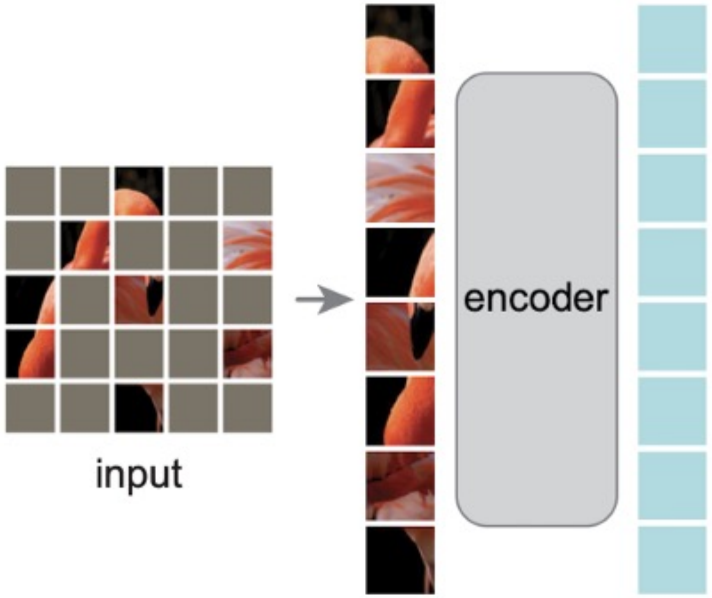
- negatives to prevent collapse
- hand-designed *data augmentations*
  "Too easy to solve" otherwise

- What are other possible tasks?
  - "Classical methods" revisited
  - Clustering? 🙁
  - Autoencoder? 🙁
  - Denoising autoencoder? 🤔

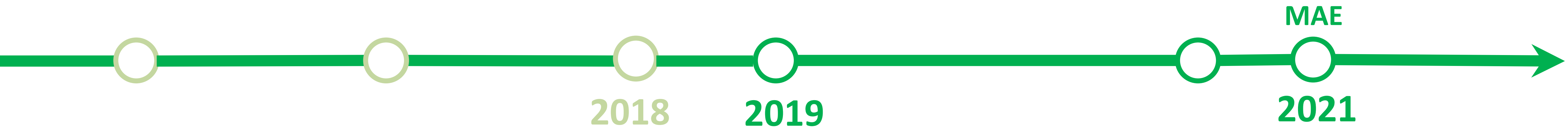


# Masked Autoencoders (MAE)

[He, Chen, **Xie**, Li, Dollár, Girshick, CVPR 2022]

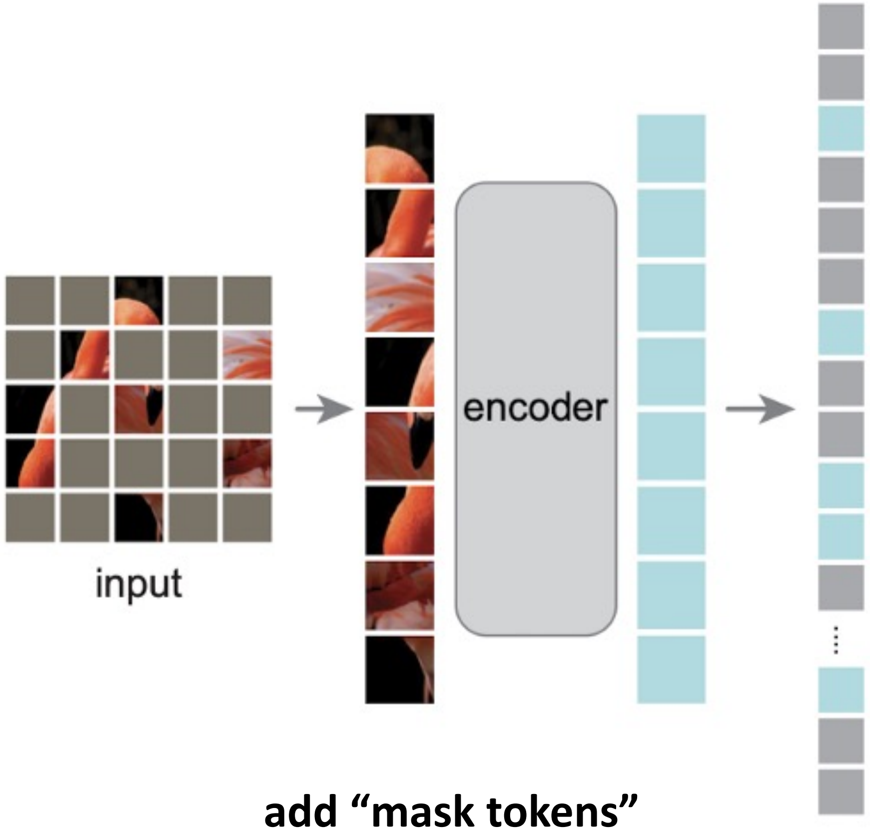



**encode visible patches**

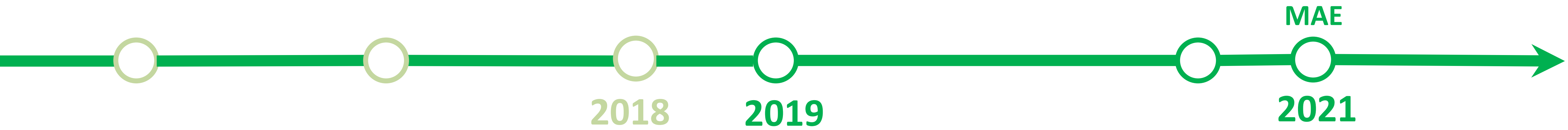


# Masked Autoencoders (MAE)

[He, Chen, **Xie**, Li, Dollár, Girshick, CVPR 2022]

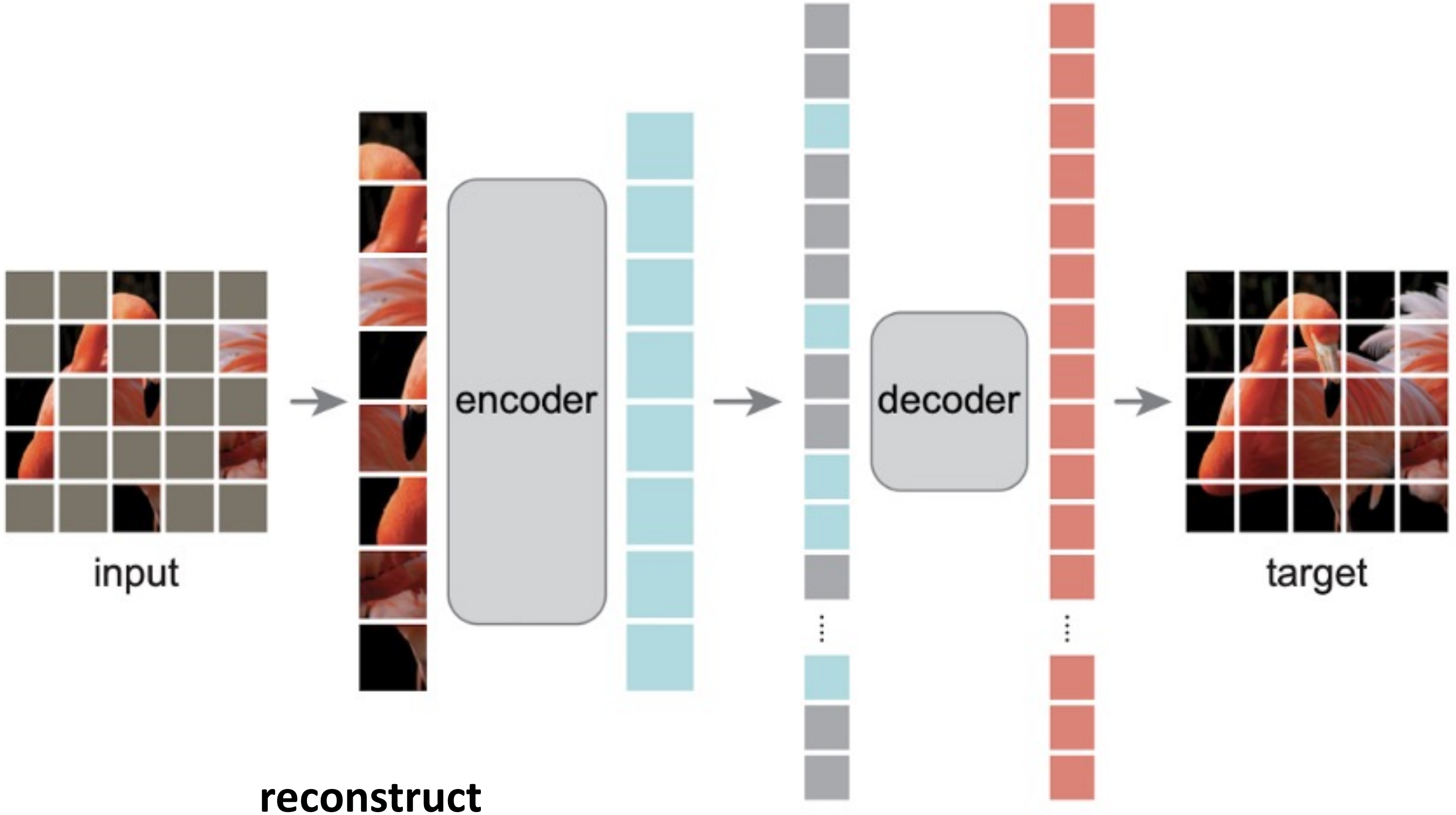



**add "mask tokens"**

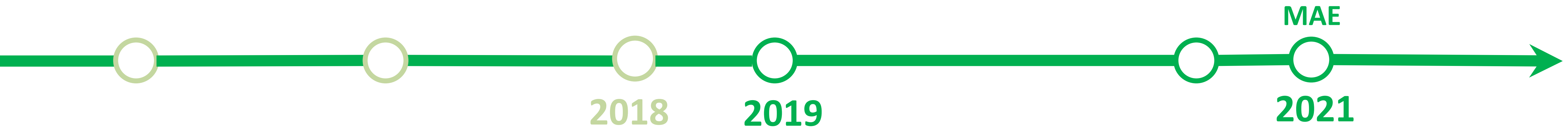


# Masked Autoencoders (MAE)

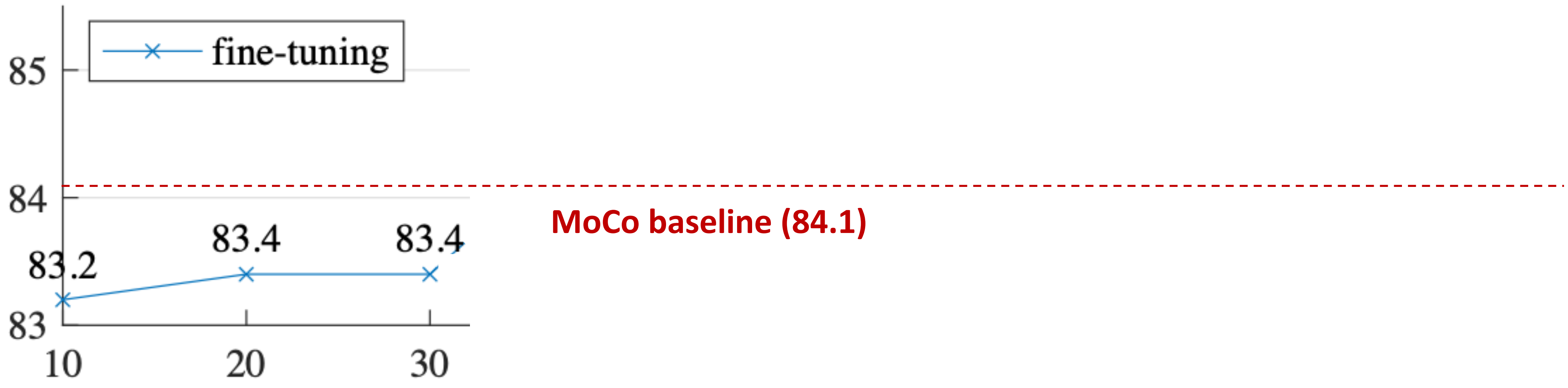
[He, Chen, **Xie**, Li, Dollár, Girshick, CVPR 2022]



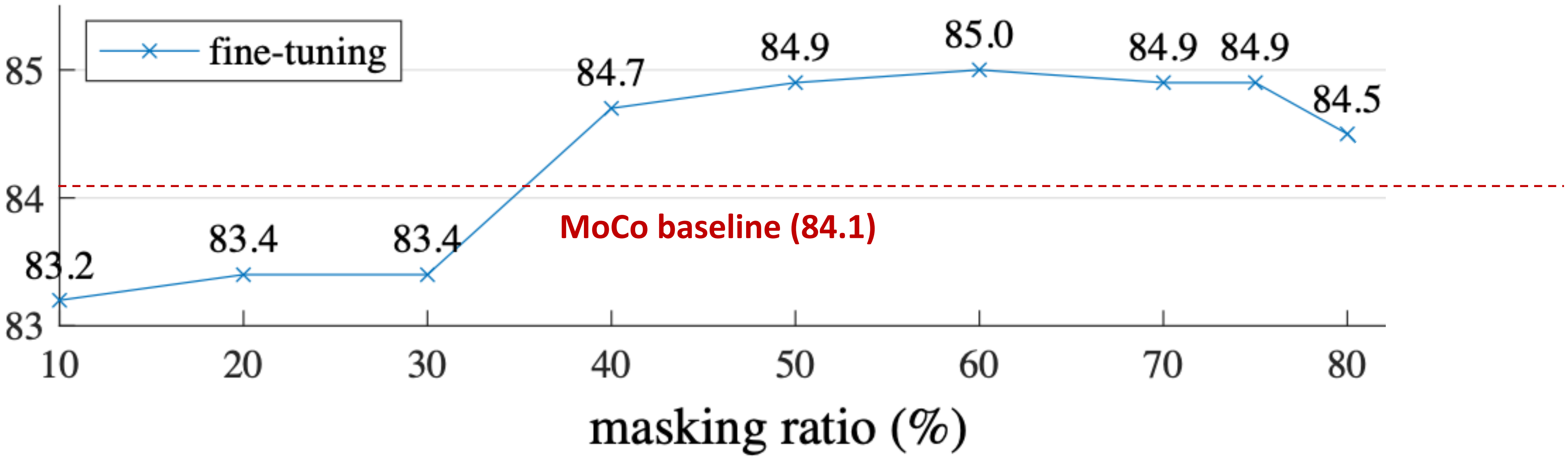

**reconstruct**



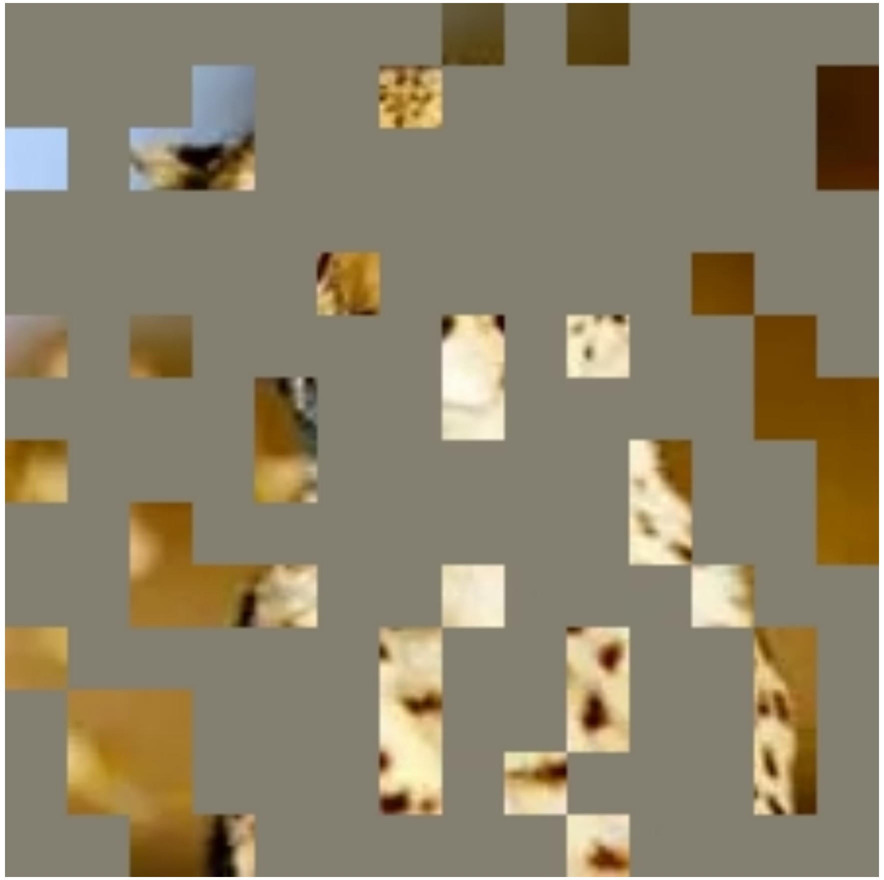
# Masking Ratio



# Masking Ratio

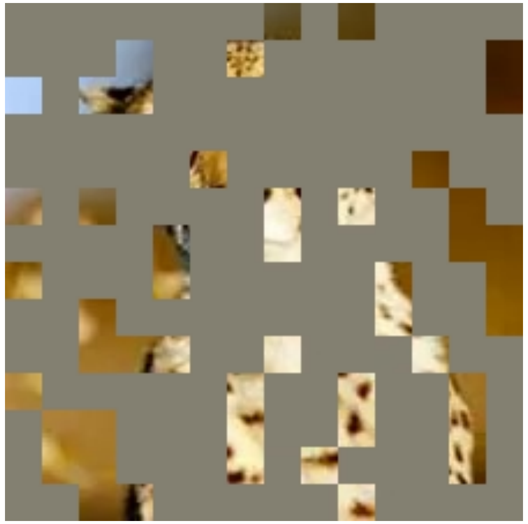


# Visualization



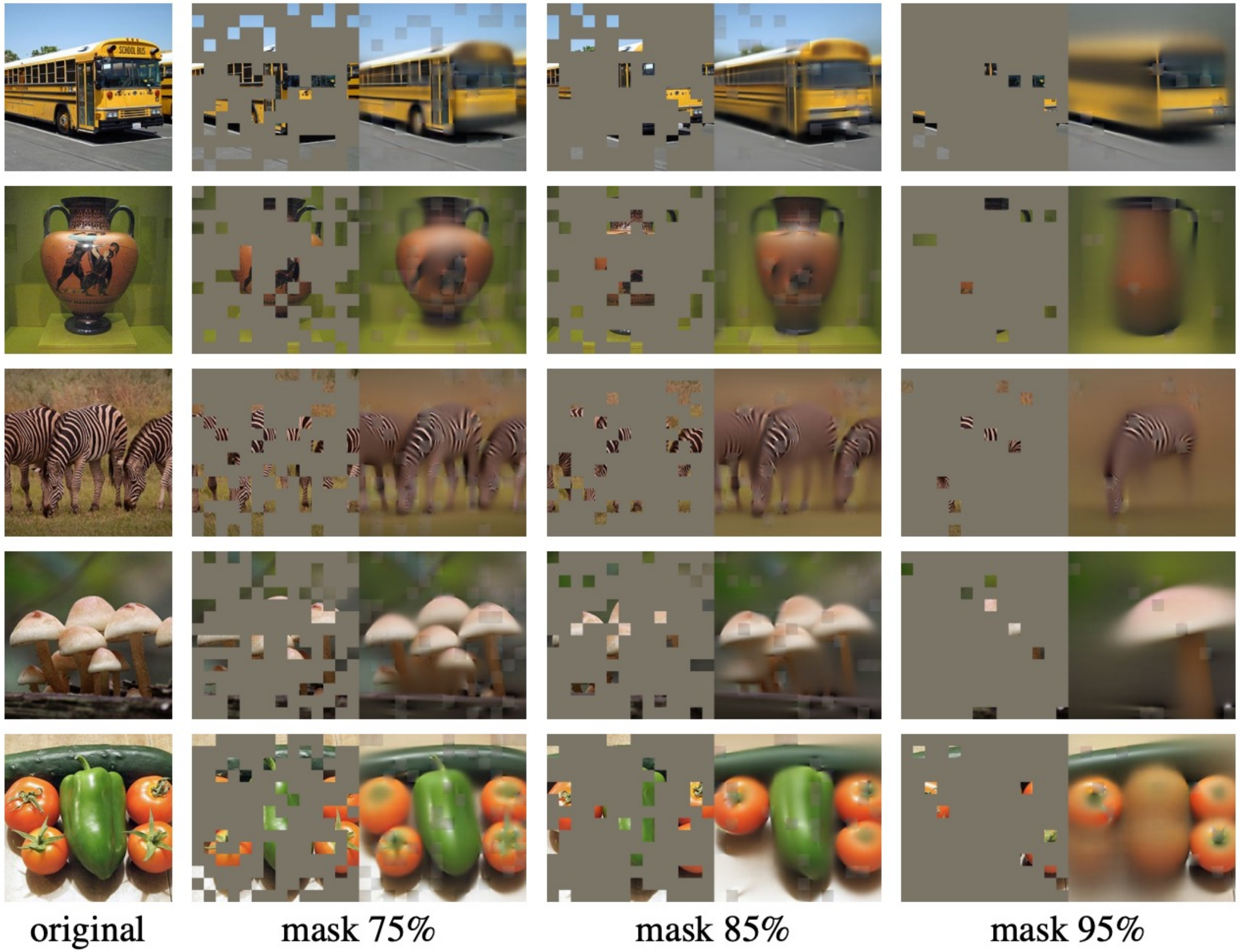
# Why high masking ratio

- Eliminates redundancy
- Cannot be easily solved by extrapolating from visible neighboring patches
- Practical benefits: significant speedup!

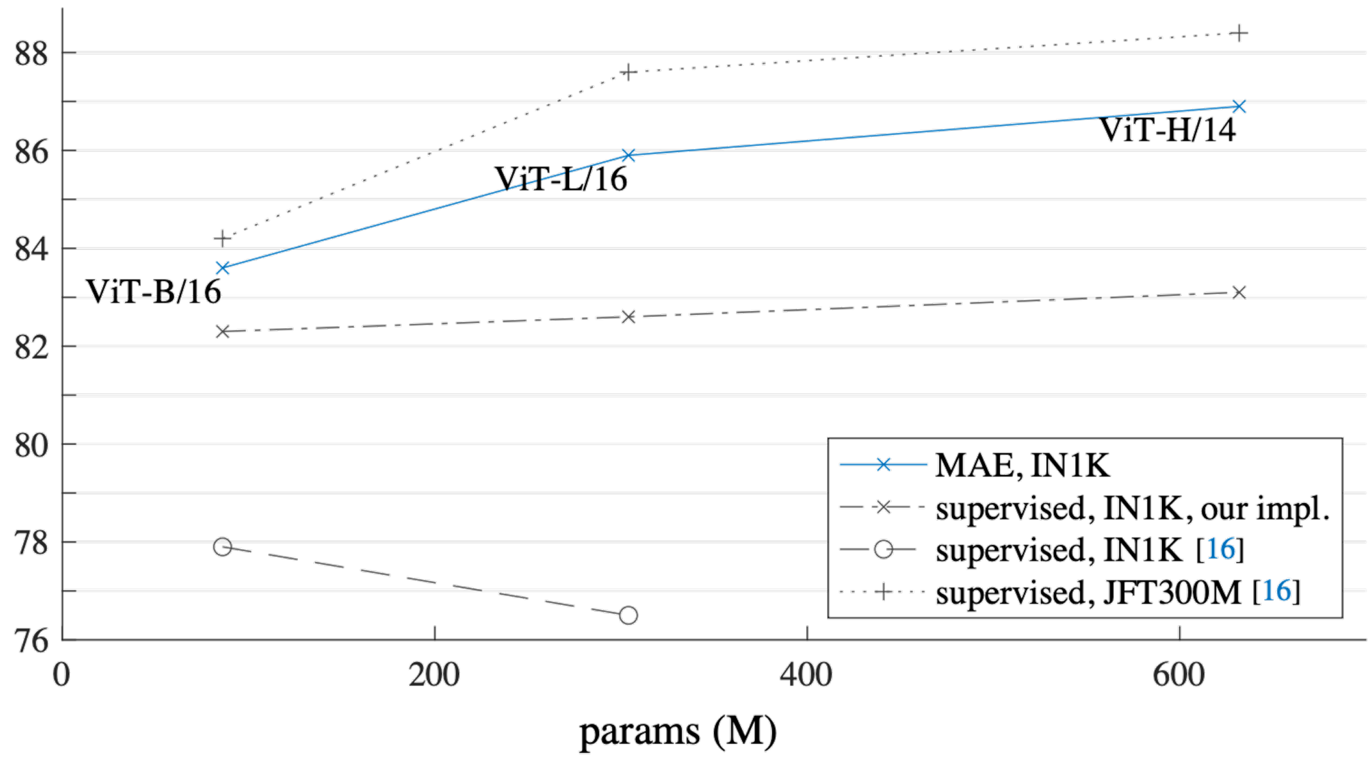


# Generalization

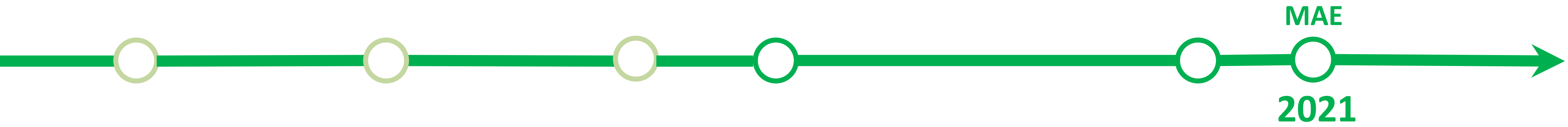
- Samples from the *validation* set
  - Pre-trained with 75% masking ratio
  - Applied on higher masking ratios
- Predictions differ plausibly
  - Representations are nontrivial
  - Model can generalize



# Results




- New SOTA in ImageNet-1K (no external data): 87.8%



# Results

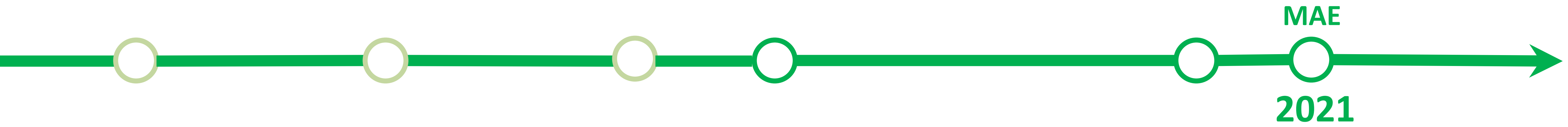
| method | pre-train data | $AP^{box}$ | | $AP^{mask}$ | |
|---|---|---|---|---|---|
| | | ViT-B | ViT-L | ViT-B | ViT-L |
| supervised | IN1K w/ labels | 47.9 | 49.3 | 42.9 | 43.9 |
| MoCo v3 | IN1K | 47.9 | 49.3 | 42.7 | 44.0 |
| BEiT | IN1K+DALLE | 49.8 | **53.3** | 44.4 | 47.1 |
| MAE | IN1K | **50.3** | **53.3** | **44.9** | **47.2** |

| method | pre-train data | ViT-B | ViT-L |
|---|---|---|---|
| supervised | IN1K w/ labels | 47.4 | 49.9 |
| MoCo v3 | IN1K | 47.3 | 49.1 |
| BEiT | IN1K+DALLE | 47.1 | 53.3 |
| MAE | IN1K | **48.1** | **53.6** |

- Transfer Learning on detection / segmentation

| dataset | ViT-B | ViT-L | ViT-H | $\text{ViT-H}_{448}$ | prev best |
|---|---|---|---|---|---|
| IN-Corruption ↓ [27] | 51.7 | 41.8 | **33.8** | 36.8 | 42.5 [32] |
| IN-Adversarial [28] | 35.9 | 57.1 | 68.2 | **76.7** | 35.8 [41] |
| IN-Rendition [26] | 48.3 | 59.9 | 64.4 | **66.5** | 48.7 [41] |
| IN-Sketch [60] | 34.5 | 45.3 | 49.6 | **50.9** | 36.0 [41] |

- Robust evaluations: +40% previous SOTA



# Rethinking model robustness

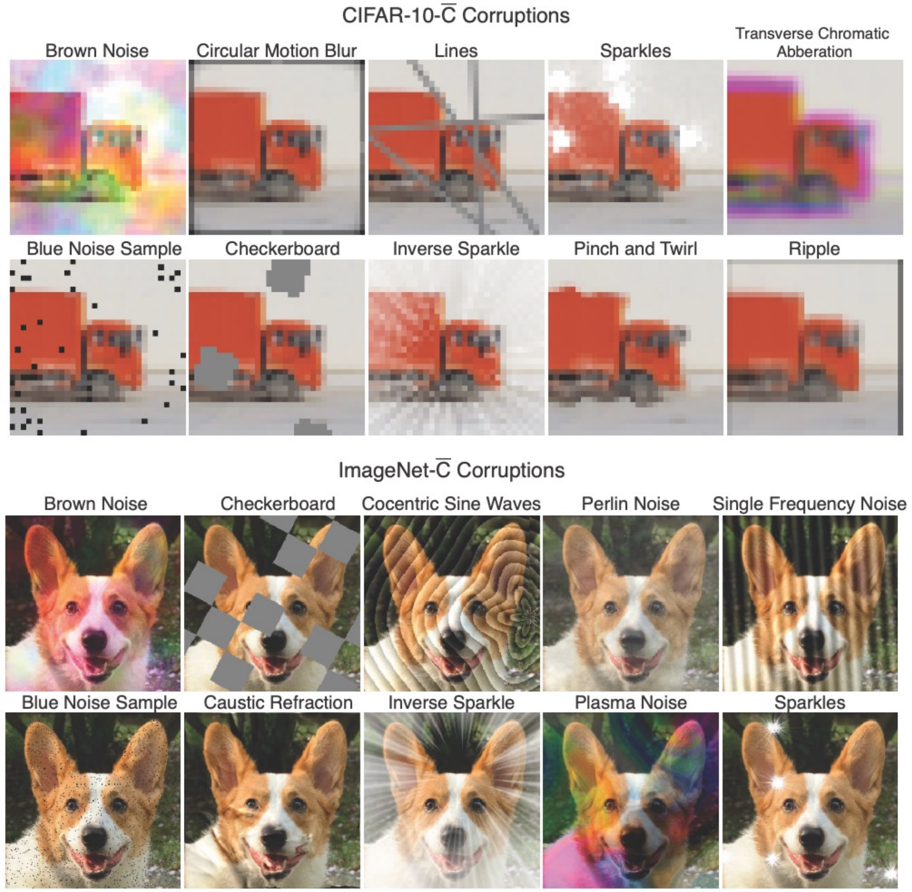
Masked Autoencoder

| dataset | ViT-B | ViT-L | ViT-H | ViT-H$_{448}$ | prev best |
|---|---|---|---|---|---|
| IN-Corruption ↓ [27] | 51.7 | 41.8 | **33.8** | 36.8 | 42.5 [32] |
| IN-Adversarial [28] | 35.9 | 57.1 | 68.2 | **76.7** | 35.8 [41] |
| IN-Rendition [26] | 48.3 | 59.9 | 64.4 | **66.5** | 48.7 [41] |
| IN-Sketch [60] | 34.5 | 45.3 | 49.6 | **50.9** | 36.0 [41] |

Scaling brings +40% in accuracy

ConvNeXt

| Model | Data/Size | FLOPs / Params | Clean | C (↓) | $\bar{C}$ (↓) | A | R | SK |
|---|---|---|---|---|---|---|---|---|
| ResNet-50 | 1K/224$^2$ | 4.1 / 25.6 | 76.1 | 76.7 | 57.7 | 0.0 | 36.1 | 24.1 |
| Swin-T [42] | 1K/224$^2$ | 4.5 / 28.3 | 81.2 | 62.0 | - | 21.6 | 41.3 | 29.1 |
| RVT-S* [44] | 1K/224$^2$ | 4.7 / 23.3 | 81.9 | 49.4 | 37.5 | 25.7 | 47.7 | 34.7 |
| ConvNeXt-T | 1K/224$^2$ | 4.5 / 28.6 | 82.1 | 53.2 | 40.0 | 24.2 | 47.2 | 33.8 |
| Swin-B [42] | 1K/224$^2$ | 15.4 / 87.8 | 83.4 | 54.4 | - | 35.8 | 46.6 | 32.4 |
| RVT-B* [44] | 1K/224$^2$ | 17.7 / 91.8 | 82.6 | 46.8 | **30.8** | 28.5 | 48.7 | 36.0 |
| ConvNeXt-B | 1K/224$^2$ | 15.4 / 88.6 | **83.8** | **46.8** | 34.4 | **36.7** | **51.3** | **38.2** |
| ConvNeXt-B | 22K/384$^2$ | 45.1 / 88.6 | 86.8 | 43.1 | 30.7 | 62.3 | 64.9 | 51.6 |
| ConvNeXt-L | 22K/384$^2$ | 101.0 / 197.8 | 87.5 | 40.2 | 29.9 | 65.5 | 66.7 | 52.8 |
| ConvNeXt-XL | 22K/384$^2$ | 179.0 / 350.2 | **87.8** | **38.8** | **27.1** | **69.3** | **68.2** | **55.0** |

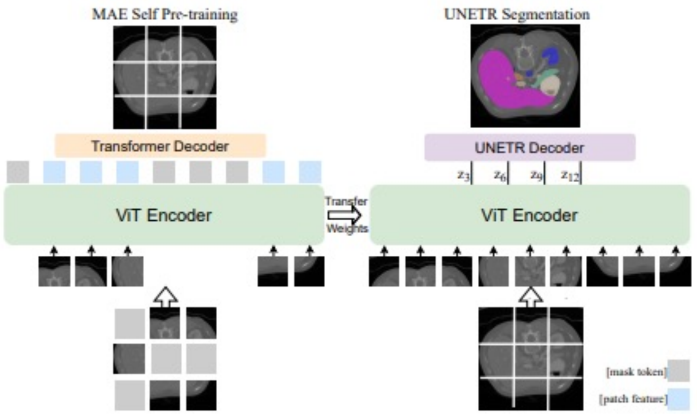



[Mintun, Kirillove, **Xie**, NeurIPS 2021]

ConvNeXt – similar story

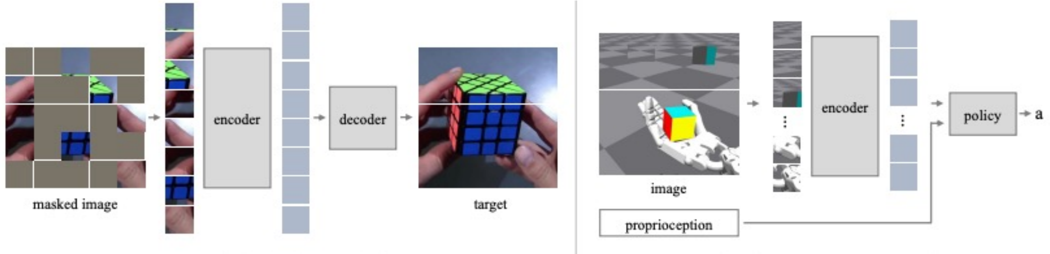
# MAE as a general methodology

## Medical Image Analysis




[Zhou et al., 2022]

## Motor Control

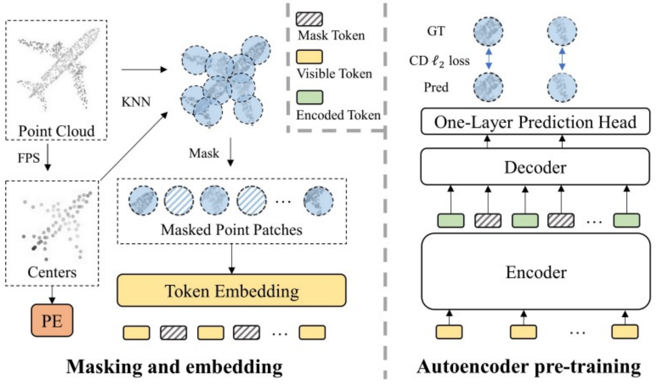



(a) masked visual pretraining

(b) learning motor control

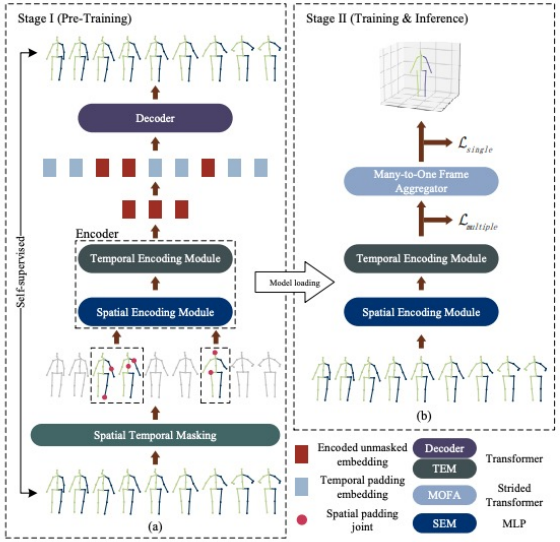
[Xiao, Radosavovic, et al., 2022]

## Point Cloud Classification

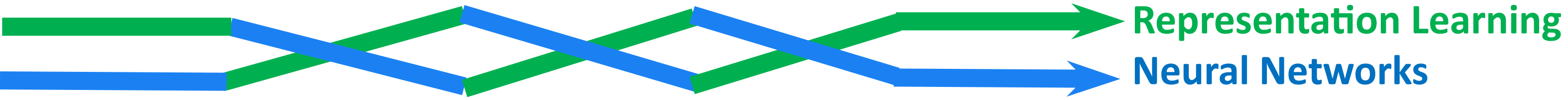



[Pang et al., 2022]

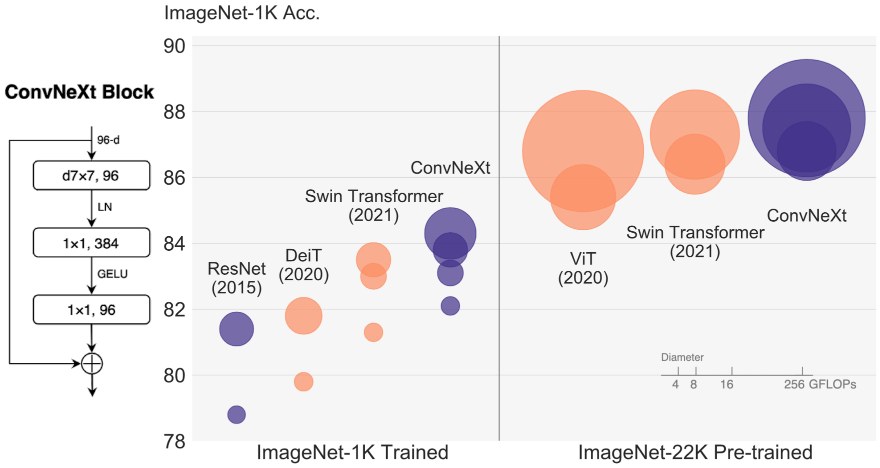
## Human Pose Estimation



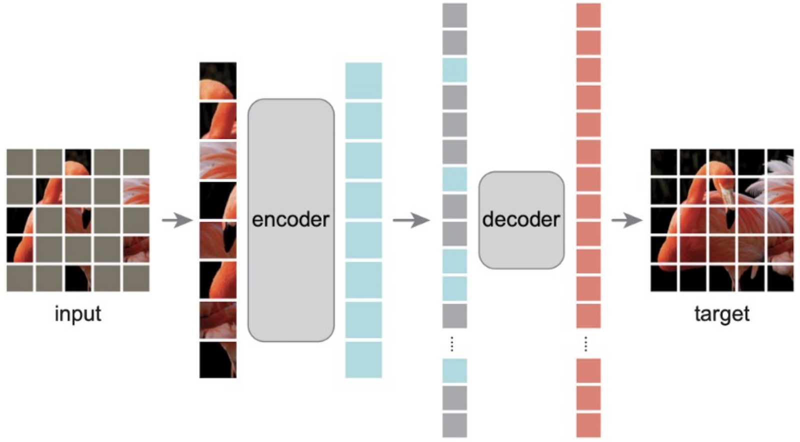

[Shan et al. 2022]




# Common theme – scalable representation learning




ConvNeXt




MAE

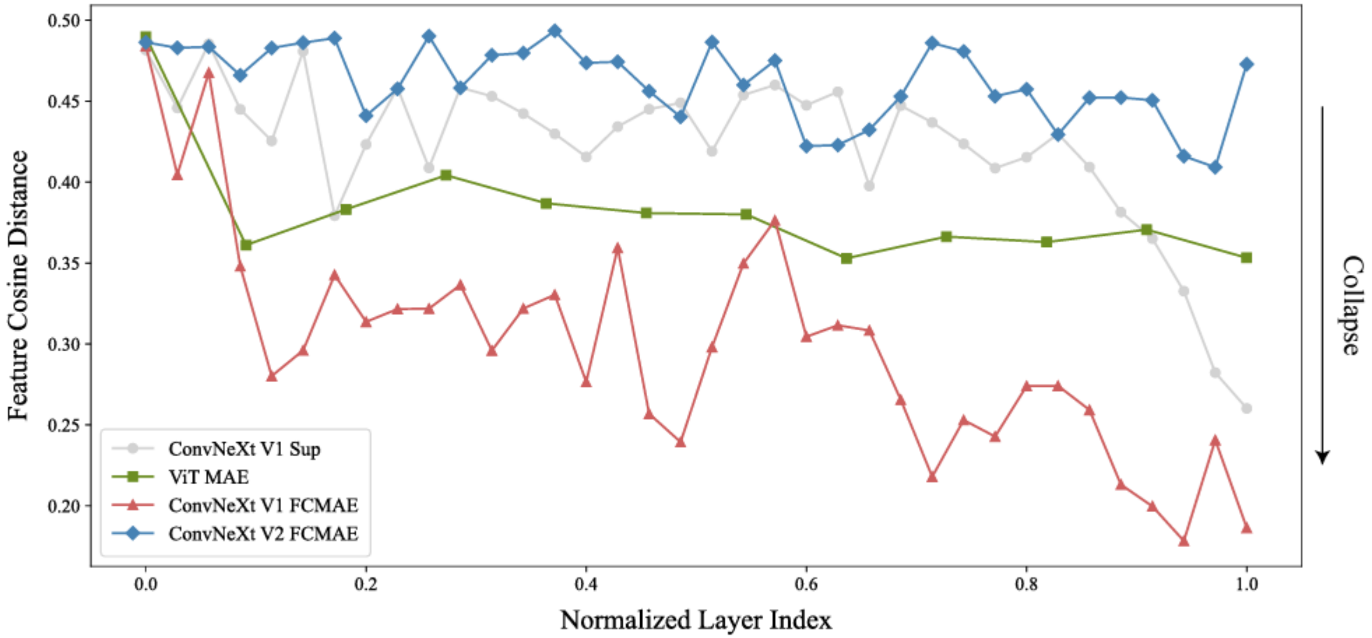
But they are not co-designed...

# Can we do MAE with ConvNeXt?

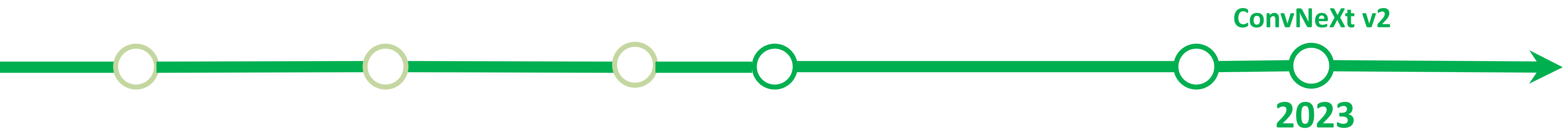
| Sup, 100ep | Sup, 300ep. [52] | FCMAE |
|---|---|---|
| 82.7 | 83.8 | 83.7 |

ConvNeXt v1 + MAE doesn't really work (!!)

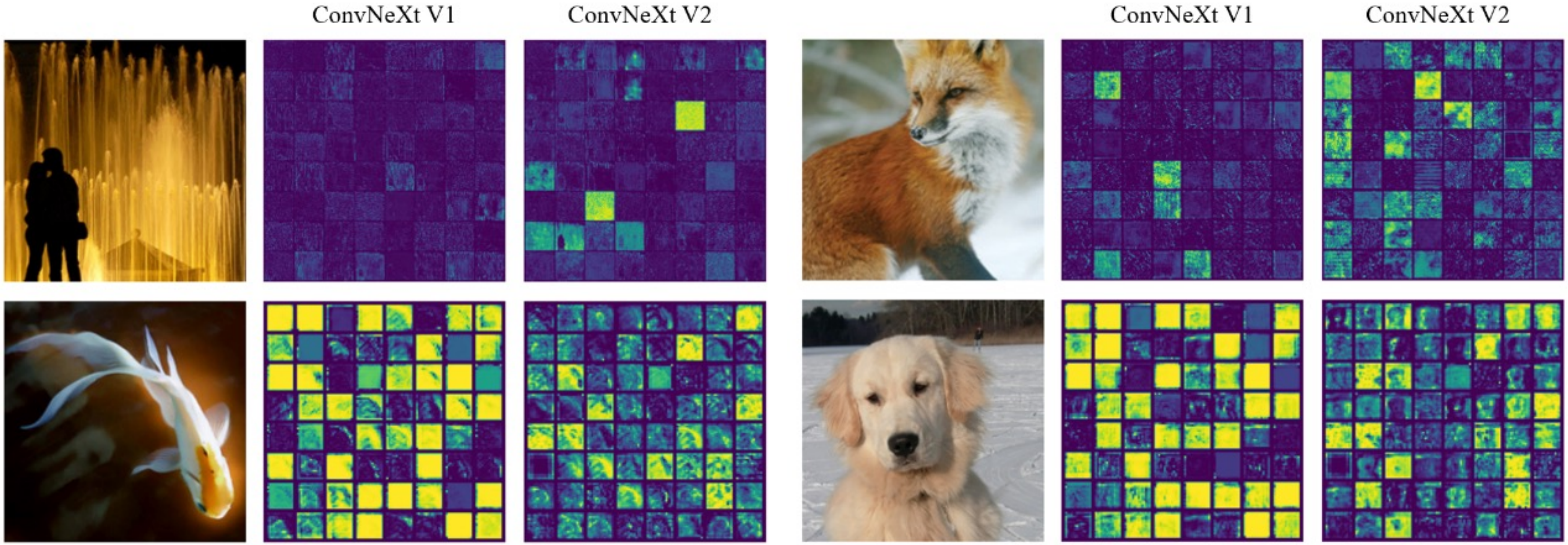
$$\frac{1}{C^2}\sum_i^C \sum_j^C \frac{1-cos(X_i,X_j)}{2}$$




Potential issue: feature collapse



# Feature Collapse



# Global Response Normalization (GRN)

Feature Aggregation:

$$\mathcal{G}(X) := X \in \mathcal{R}^{H\times W\times C} \rightarrow gx \in \mathcal{R}^{C}. \qquad (1)$$

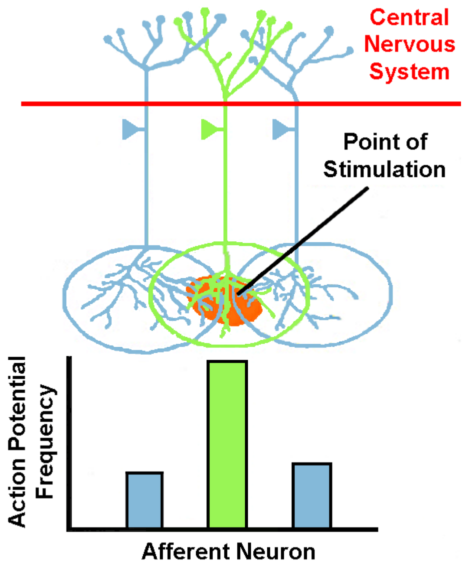
Relative Importance:

$$\mathcal{N}(||X_i||) := ||X_i|| \in \mathcal{R} \rightarrow \frac{||X_i||}{\sum_{j=1,\ldots,C} ||X_j||} \in \mathcal{R}, \qquad (2)$$

Feature Reweighting:

$$X_i = X_i * \mathcal{N}(\mathcal{G}(X)_i) \in \mathcal{R}^{H\times W} \qquad (3)$$

Related: Lateral inhibition to sharpen signals to the brain



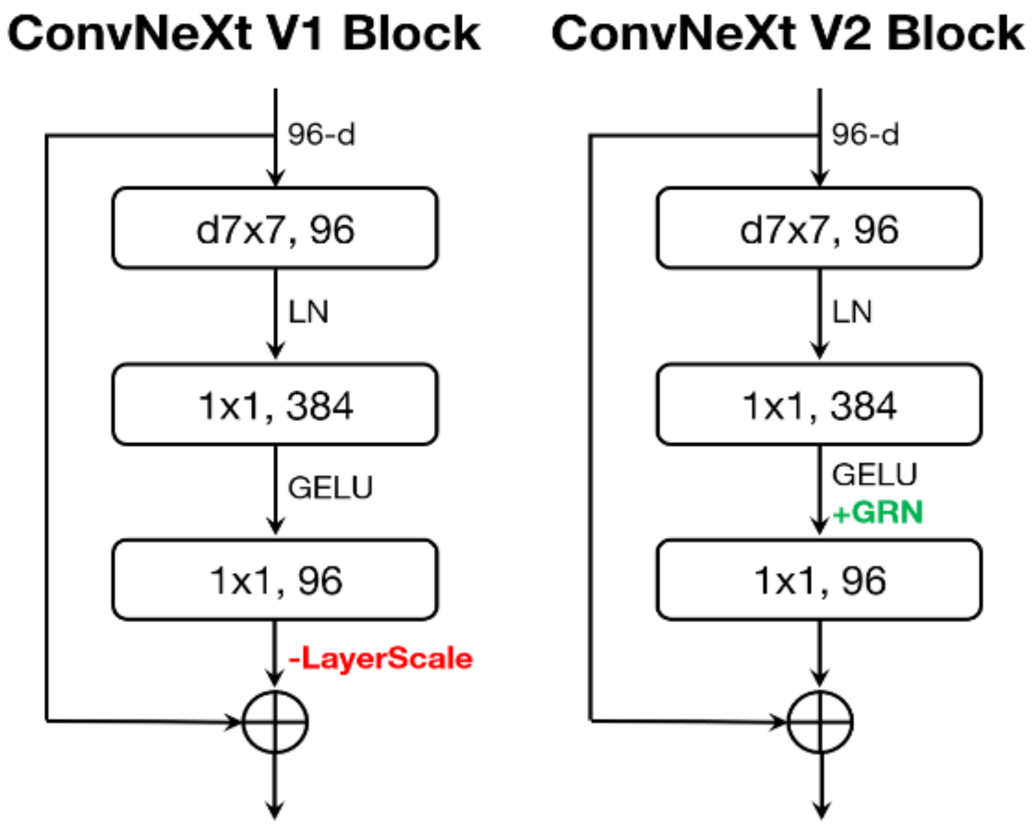
# Global Response Normalization (GRN)

**Algorithm 1** Pseudocode of GRN in a PyTorch-like style.

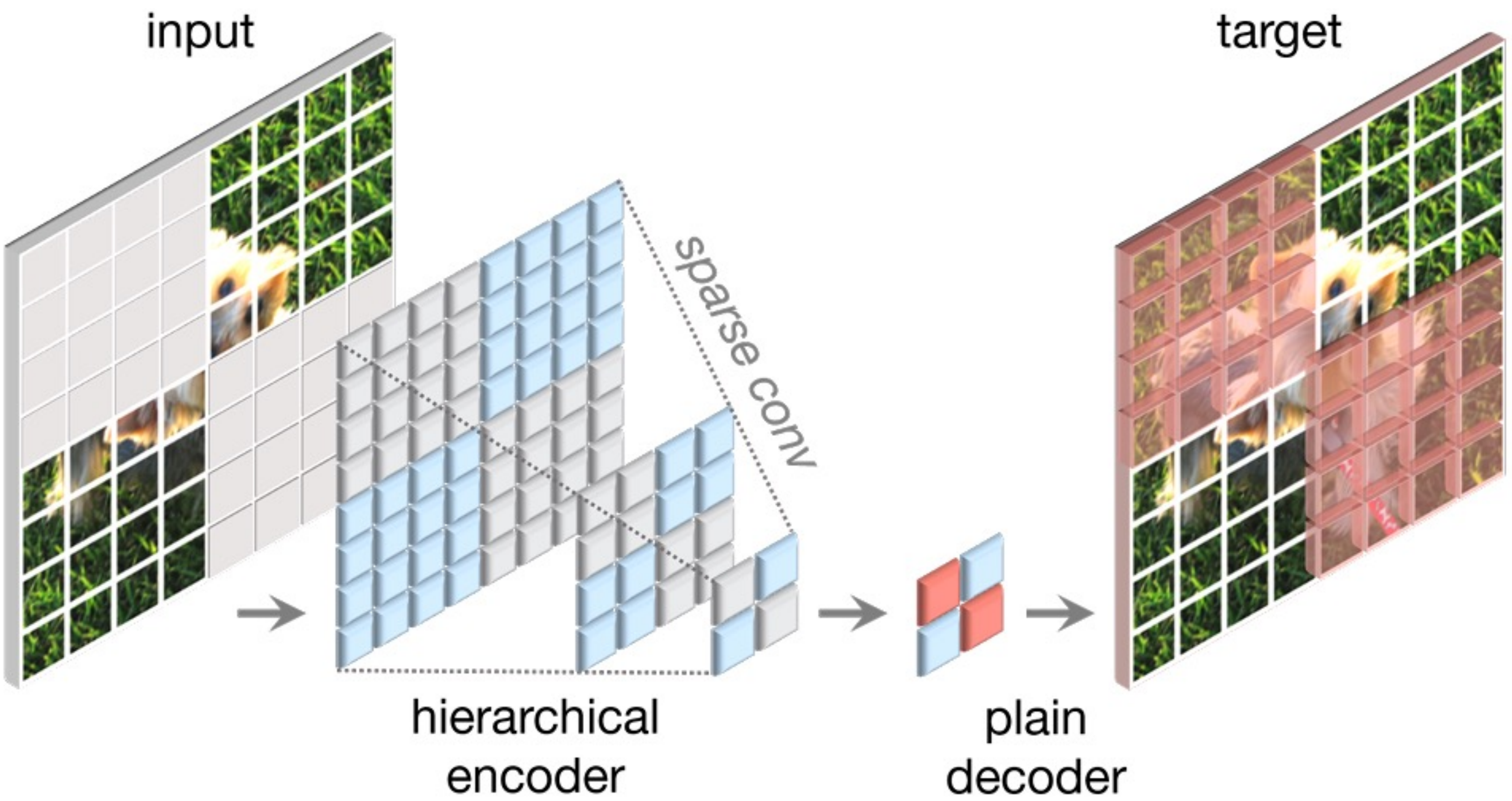
```
# gamma, beta: learnable affine transform parameters
# X: input of shape (N,H,W,C)

gx = torch.norm(X, p=2, dim=(1,2), keepdim=True)
nx = gx / (gx.mean(dim=-1, keepdim=True)+1e-6)
return gamma * (X * nx) + beta + X
```

Only three lines of code
Parameter-free!

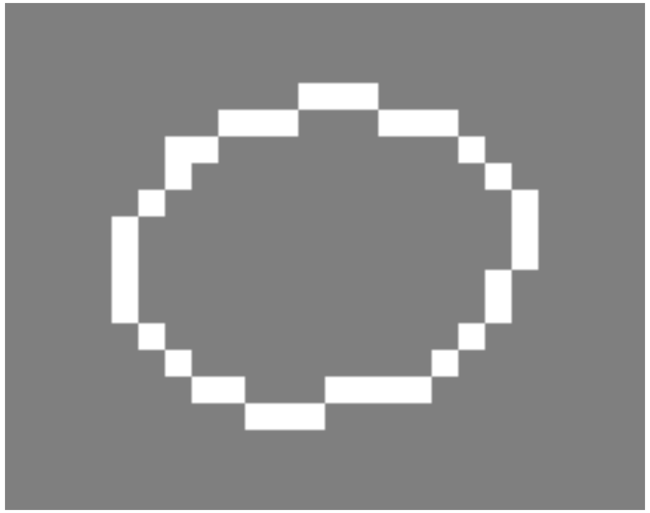


# Fully-convolutional MAE

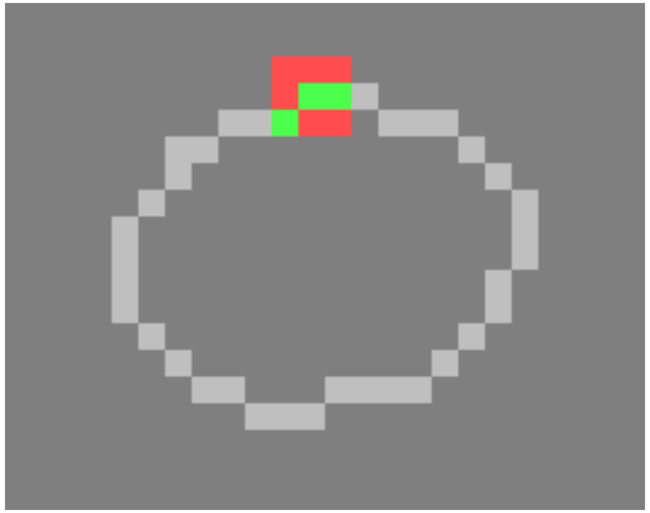


# Sparse ConvNet

Non-zero active sites grow rapidly

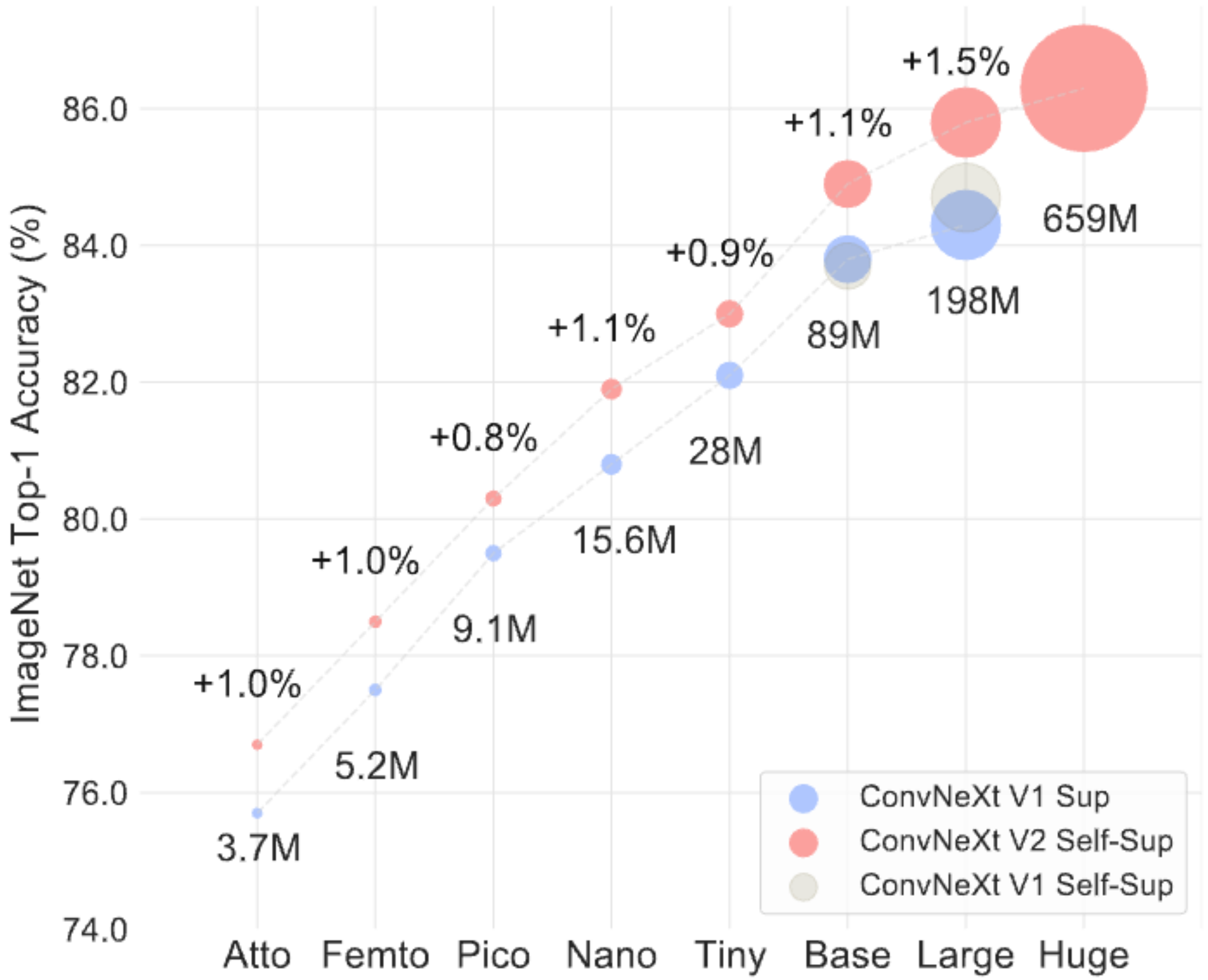


Computation only on a sparse set of “active sites”



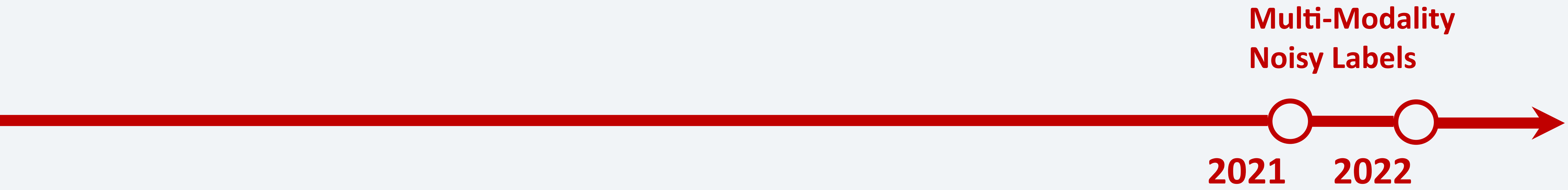
Graham, Benjamin, and Laurens van der Maaten. “Submanifold sparse convolutional networks.” *CVPR 2018*

# ConvNeXt-v2

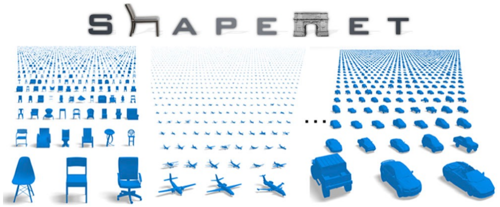

ImageNet Top-1 Accuracy (%)
86.0
84.0
82.0
80.0
78.0
76.0
74.0
+1.5%
+1.1%
+0.9%
+1.1%
+0.8%
+1.0%
+1.0%
659M
198M
89M
28M
15.6M
9.1M
5.2M
3.7M
ConvNeXt V1 Sup
ConvNeXt V2 Self-Sup
ConvNeXt V1 Self-Sup
Atto
Femto
Pico
Nano
Tiny
Base
Large
Huge

Architecture / Objective / Data

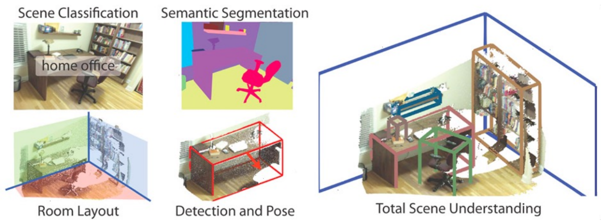
# 3. What data to use for visual pre-training



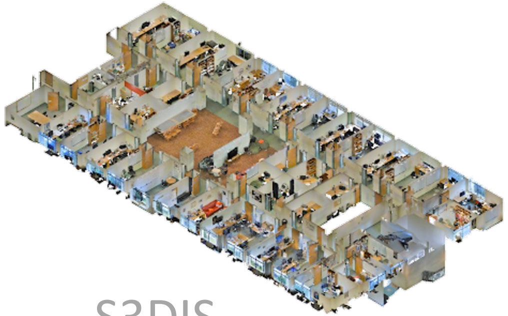
# Visual pre-training on 3D point clouds



ShapeNet

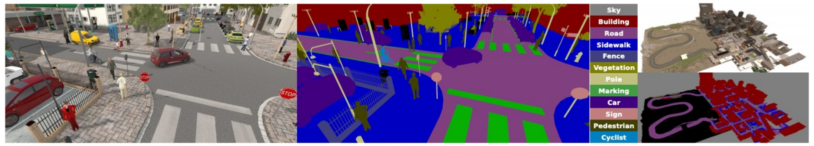



SUN RGB-D



S3DIS

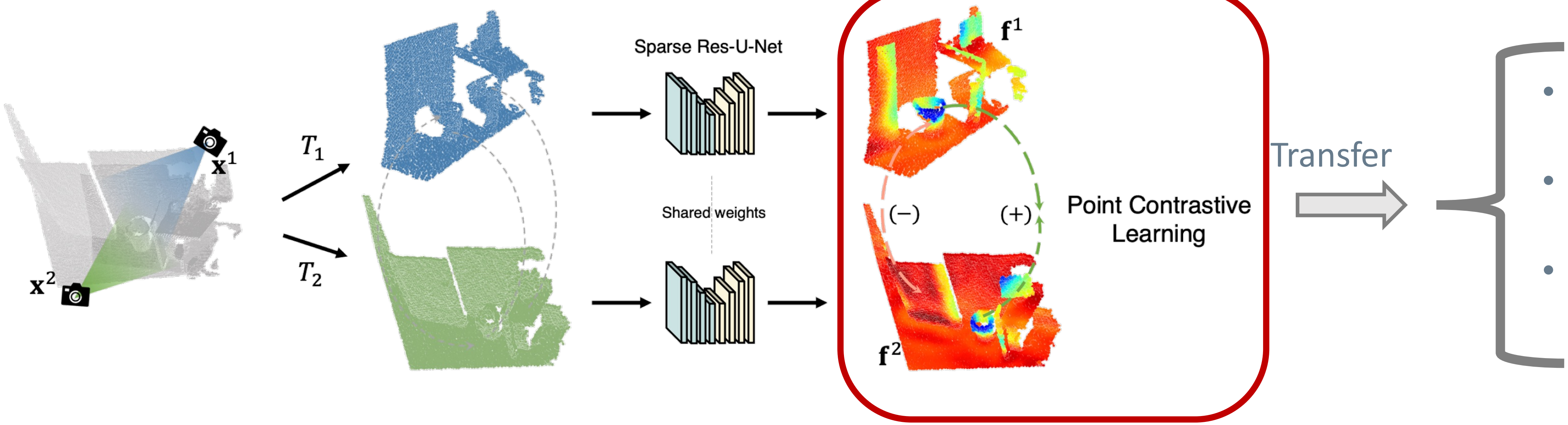
ScanNet




Synthia4D

# PointContrast

[**Xie**, Gu, Guo, Qi, Guibas, Litany, ECCV 2020]
[Hou, Graham, Nießner, **Xie**, CVPR 2021]

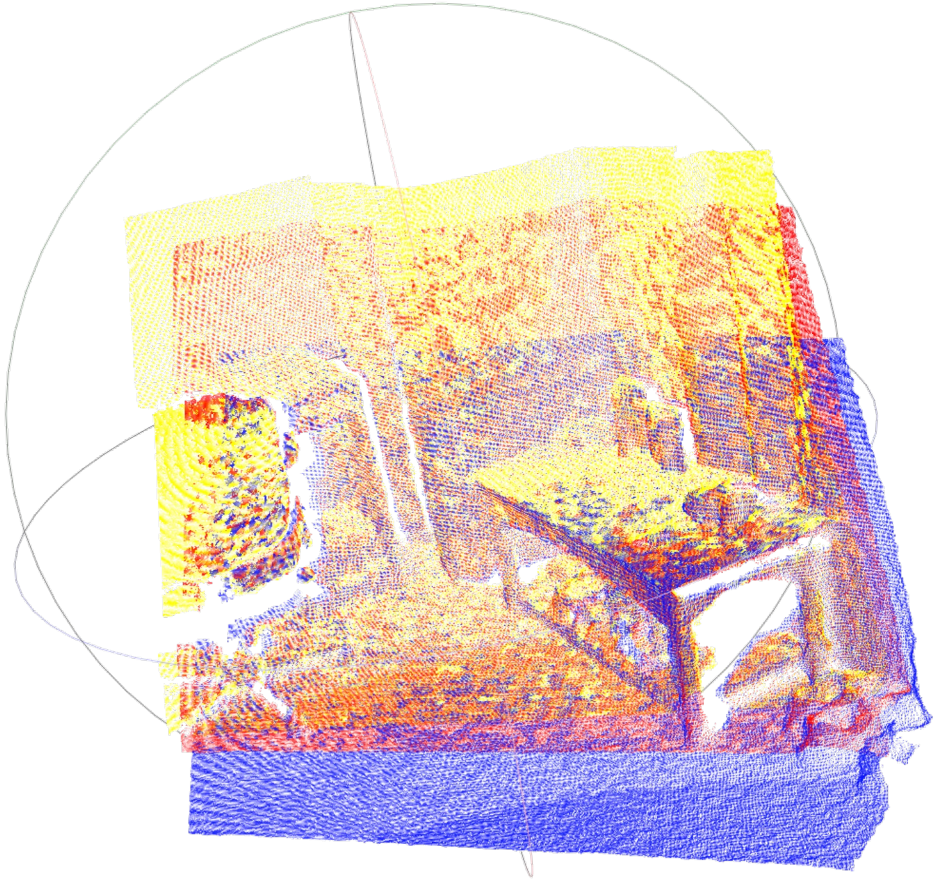



## Point Level Contrastive Learning

$$\mathcal{L}_c = -\sum_{(i,j)\in\mathcal{P}} \log \frac{\exp(\mathbf{f}_i \cdot \mathbf{f}_j/\tau)}{\sum_{(\cdot,k)\in\mathcal{P}} \exp(\mathbf{f}_i \cdot \mathbf{f}_k/\tau)}$$

## Partial frames from raw RGB-D scan

## Alignment and correspondences through reconstruction

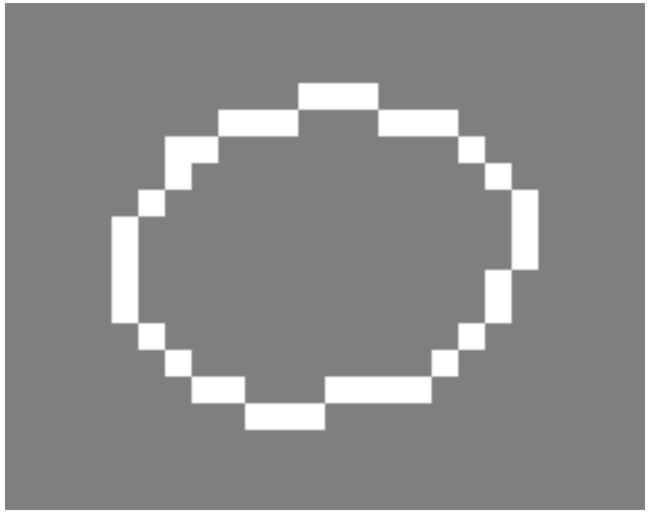



No manual annotation required

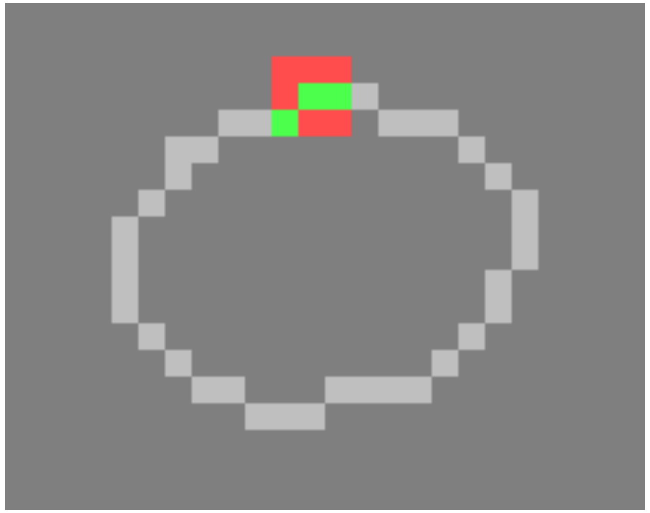
Figure: [ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. Dai et al., CVPR 2017]

# Sparse ConvNet (recall: the same thing used in ConvNeXt v2)

Non-zero active sites grow rapidly



Computation only on a sparse set of “active sites”

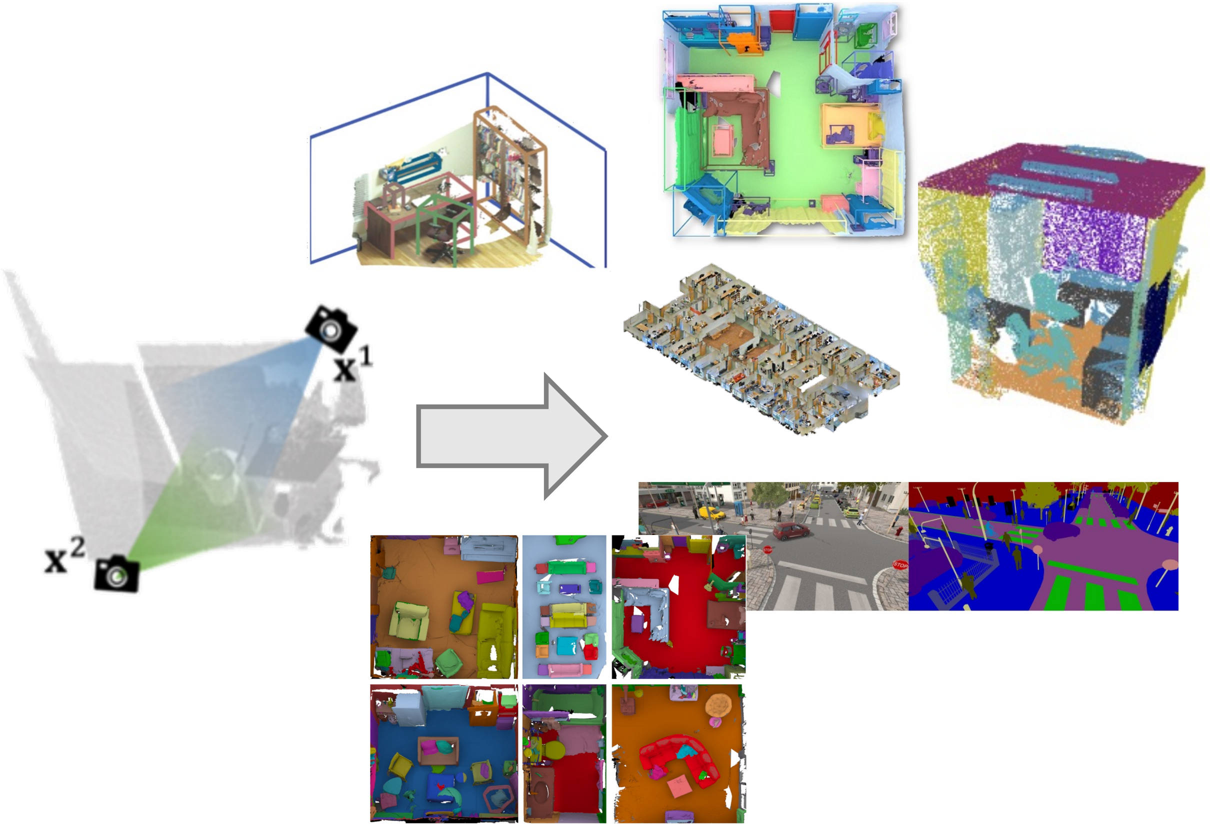


Graham, Benjamin, and Laurens van der Maaten. “Submanifold sparse convolutional networks.” *CVPR 2018*

# PointContrast

[**Xie**, Gu, Guo, Qi, Guibas, Litany, ECCV 2020]

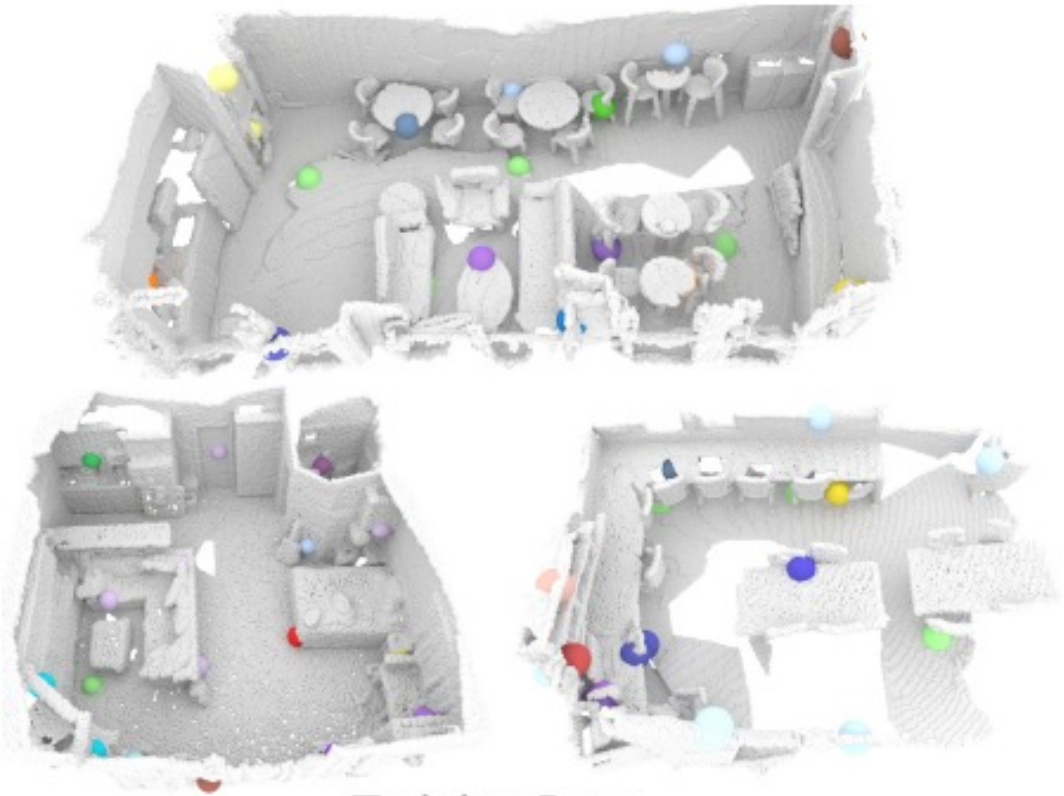
**First** work to demonstrate self-supervised learning is useful for a diverse set of 3D recognition tasks




**PointContrast: Downstream Tasks for Fine-tuning**

| Datasets | Real / Synth. | Complexity | Env. | Task | Rel. gain |
|---|---|---|---|---|---|
| S3DIS | Real | Entire floor, office | Indoor | Segmentation | (+2.7%) mIoU |
| SUN RGB-D | Real | Medium-sized cluttered rooms | Indoor | Detection | (+3.1%) mAP0.5 |
| ScanNetV2 | Real | Large rooms | Indoor | Segmentation<br>Detection | (+1.9%) mIoU<br>(+2.6%) mAP0.5 |
| ShapeNet | Synth. | Single objects | Indoor & outdoor | Classification | (+4.0%) Acc.* |
| ShapeNetPart | Synth. | Object parts | Indoor & outdoor | Segmentation | (+2.2%) mIoU* |
| Synthia 4D | Synth. | Street scenes, driving envs. | Outdoor | Segmentation | (+3.3%) mIoU |

# Representation learning enables data-efficiency

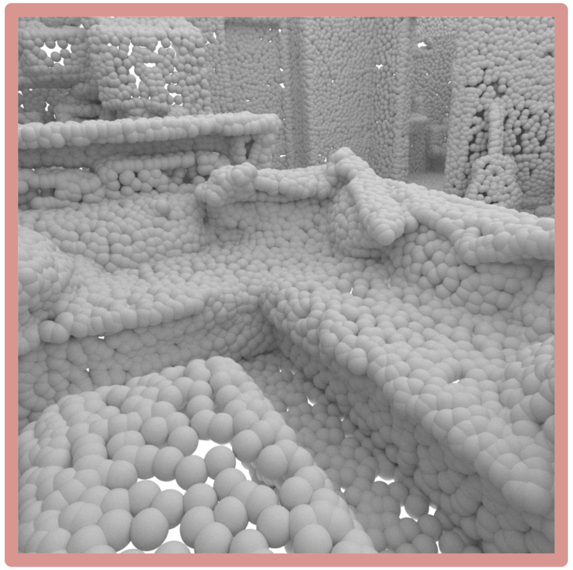


Training Data
with Limited Annotations

Instance Segmentation
Predictions

Using only 20 point labels per scene!

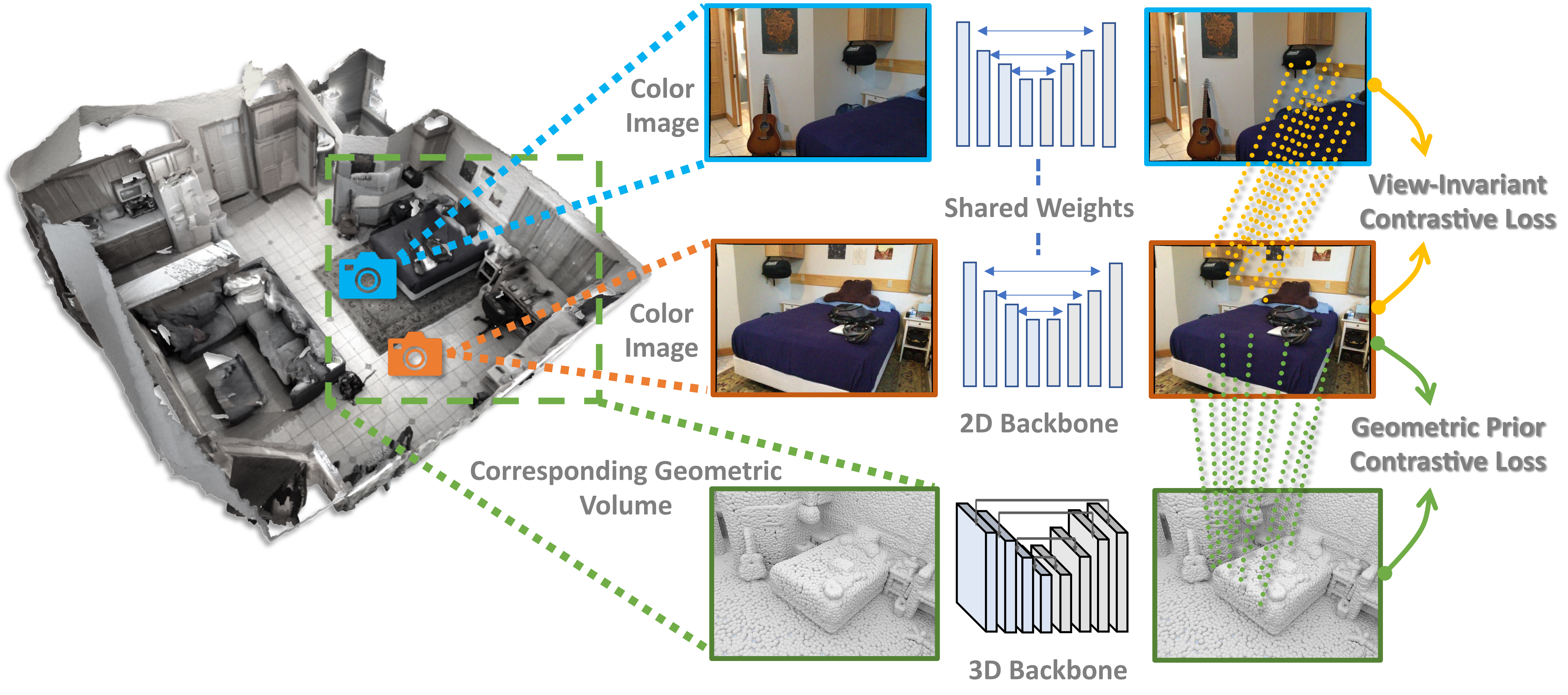
# Joint 3D-2D holistic representation learning



- Can 3D prior help 2D tasks?

[Hou, **Xie**, Graham, Dai, Nießner, ICCV 2021]

# Joint View-Invariant and Geometric Prior Contrastive Loss



# Connecting representation learning with language

**Uncurated Data**

Ttitle: Oil spill slick at NSRCC, Tanah Merah with city skyline
Desc: More about the oil spill on the wild shores of singapore blog. 300dpi photo free for download as part of the 2010 International Year of Biodiversity.

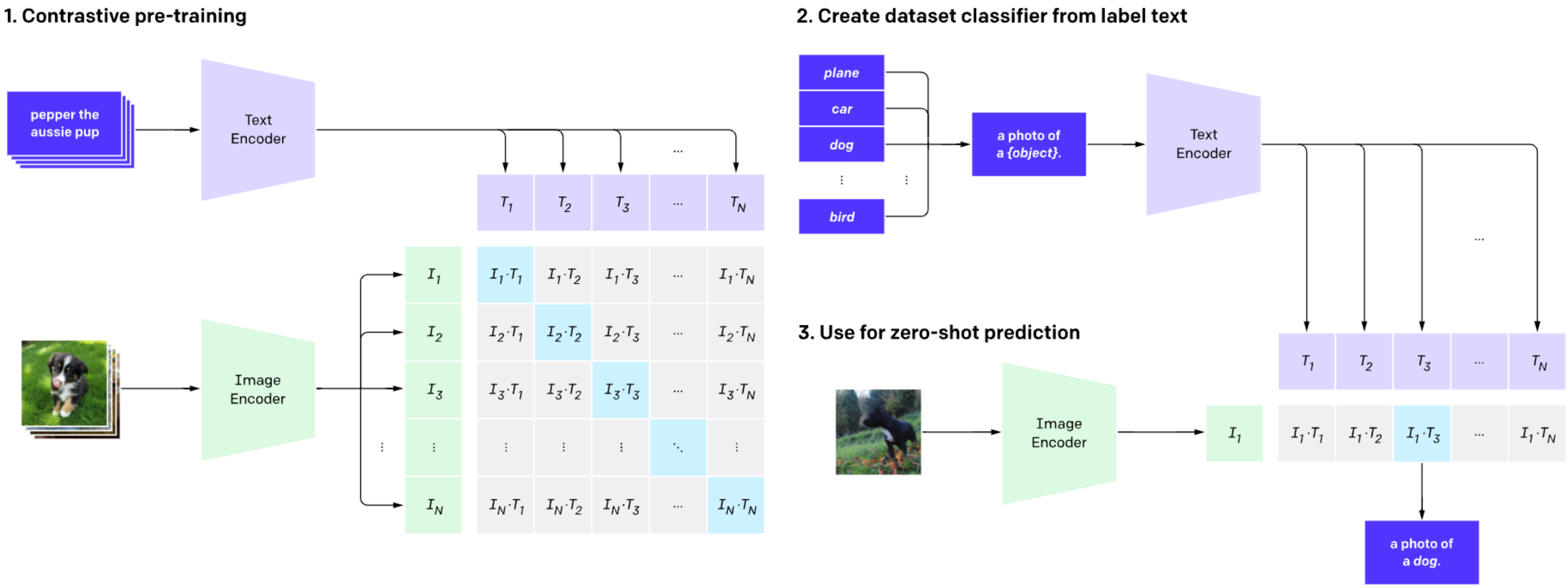
Title: Egg tastic!
Desc: we normally write the experation date on our eggs as we do not keep the container. Yesterday I found this little "Easter Egg" left by Erin!

Title: Ping pong project
Desc: Got a set of ping pong balls in for a project. Had to play with them before anything else

Samples from YFCC dataset

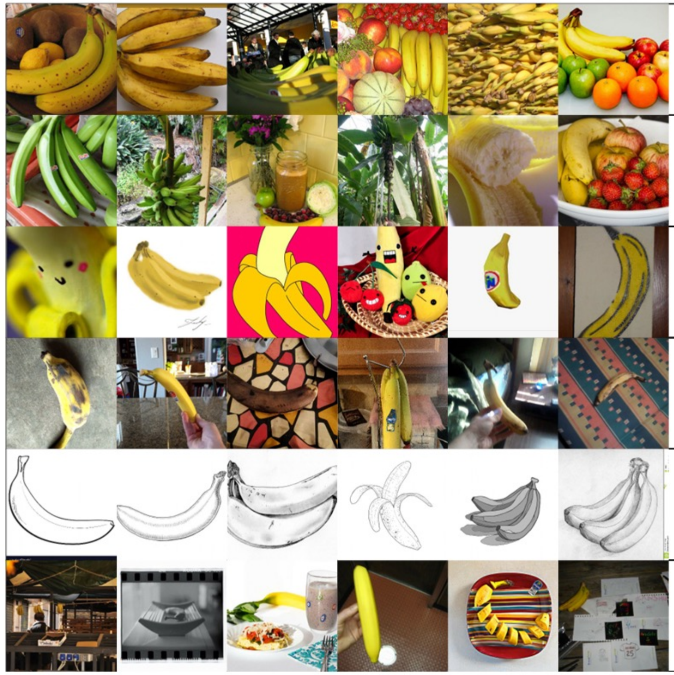
Language: a form of weak supervision?
(Well, not exactly)

# CLIP: Connecting text and images




Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *International conference on machine learning*. PMLR, 2021.

# A data-centric view of model robustness



| Dataset Examples | ImageNet ResNet101 | Zero-Shot CLIP | Δ Score |
|---|---|---|---|
| ImageNet | **76.2** | **76.2** | 0% |
| mageNetV2 | 64.3 | **70.1** | +5.8% |
| ImageNet-R | 37.7 | **88.9** | +51.2% |
| ObjectNet | 32.6 | **72.3** | +39.7% |
| ImageNet Sketch | 25.2 | **60.2** | +35.0% |
| ImageNet-A | 2.7 | **77.1** | +74.4% |

## OpenCLIP

Zero-shot accuracies at resolution 224x224 computed with CLIP Benchmark.

| Dataset | OpenCLIP H/14 | OpenCLIP G/14 |
|---|---|---|
| ImageNet | 78.0 | 80.1 |
| ImageNet-V2 | 70.8 | 73.6 |
| ImageNet-R | 89.3 | 92.1 |
| ImageNet-Sketch | 66.6 | 68.9 |
| ObjectNet | 69.7 | 73.0 |
| ImageNet-A | 59.2 | 69.3 |
| CIFAR-10 | 97.4 | 98.2 |
| CIFAR-100 | 84.7 | 87.5 |
| MNIST | 72.9 | 71.6 |
| SVHN | 56.1 | 62.5 |
| Caltech-101 | 85.0 | 86.4 |
| SUN397 | 75.2 | 74.5 |
| FGVC Aircraft | 42.8 | 49.7 |
| Country211 | 30.0 | 33.8 |
| Cars | 93.5 | 94.6 |

Radford, Alec, et al. "Learning transferable visual models from natural language supervision." *International conference on machine learning*. PMLR, 2021.

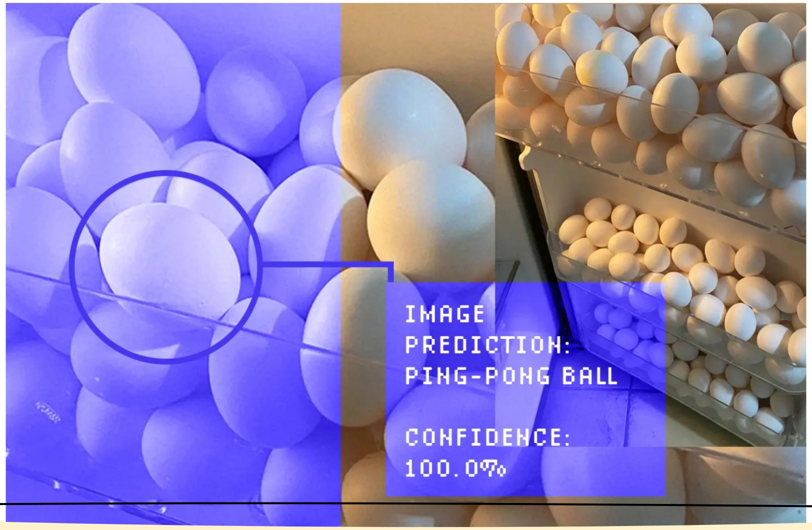
Cherti, Mehdi, et al. "Reproducible scaling laws for contrastive language-image learning." *arXiv preprint arXiv:2212.07143* (2022).

# Building accurate models

AI

## The creator of the viral "neural net guesses memes" Twitter account explains how it works

Computer engineer Glen Neumann breaks down ResNeXtGuesser, his bot that often hilariously reacts to bizarre images.




*"Just as artificial intelligence notoriously lacks common sense, it also lacks an understanding of memes."*

**neural net guesses memes**
@ResNeXtGuesser

Image prediction: ping-pong ball
Confidence: 99.99%
Submission by @Minish900

10:53 PM · Nov 9, 2021 · ImageGuesserBot

**10.5K** Retweets **1,181** Quote Tweets **115.3K** Likes

# Recall: model robustness
# from model scaling & self-supervised learning

Masked Autoencoder

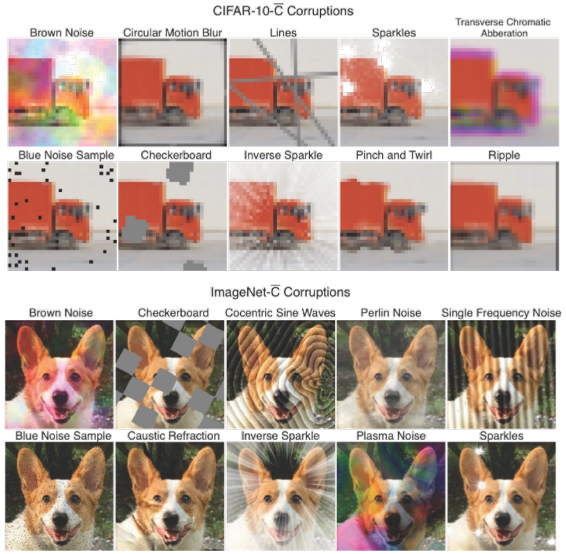
| dataset | ViT-B | ViT-L | ViT-H | ViT-$H_{448}$ | prev best |
|---|---|---|---|---|---|
| IN-Corruption ↓ [27] | 51.7 | 41.8 | **33.8** | 36.8 | 42.5 [32] |
| IN-Adversarial [28] | 35.9 | 57.1 | 68.2 | **76.7** | 35.8 [41] |
| IN-Rendition [26] | 48.3 | 59.9 | 64.4 | **66.5** | 48.7 [41] |
| IN-Sketch [60] | 34.5 | 45.3 | 49.6 | **50.9** | 36.0 [41] |

Scaling brings +40% in accuracy

ConvNeXt

| Model | Data/Size | FLOPs / Params | Clean | C (↓) | $\bar{C}$ (↓) | A | R | SK |
|---|---|---|---|---|---|---|---|---|
| ResNet-50 | 1K/$224^2$ | 4.1 / 25.6 | 76.1 | 76.7 | 57.7 | 0.0 | 36.1 | 24.1 |
| Swin-T [42] | 1K/$224^2$ | 4.5 / 28.3 | 81.2 | 62.0 | - | 21.6 | 41.3 | 29.1 |
| RVT-S* [44] | 1K/$224^2$ | 4.7 / 23.3 | 81.9 | 49.4 | 37.5 | 25.7 | 47.7 | 34.7 |
| ConvNeXt-T | 1K/$224^2$ | 4.5 / 28.6 | 82.1 | 53.2 | 40.0 | 24.2 | 47.2 | 33.8 |
| Swin-B [42] | 1K/$224^2$ | 15.4 / 87.8 | 83.4 | 54.4 | - | 35.8 | 46.6 | 32.4 |
| RVT-B* [44] | 1K/$224^2$ | 17.7 / 91.8 | 82.6 | 46.8 | **30.8** | 28.5 | 48.7 | 36.0 |
| ConvNeXt-B | 1K/$224^2$ | 15.4 / 88.6 | **83.8** | **46.8** | 34.4 | **36.7** | **51.3** | **38.2** |
| ConvNeXt-B | 22K/$384^2$ | 45.1 / 88.6 | 86.8 | 43.1 | 30.7 | 62.3 | 64.9 | 51.6 |
| ConvNeXt-L | 22K/$384^2$ | 101.0 / 197.8 | 87.5 | 40.2 | 29.9 | 65.5 | 66.7 | 52.8 |
| ConvNeXt-XL | 22K/$384^2$ | 179.0 / 350.2 | **87.8** | **38.8** | **27.1** | **69.3** | **68.2** | **55.0** |

ConvNeXt – similar story

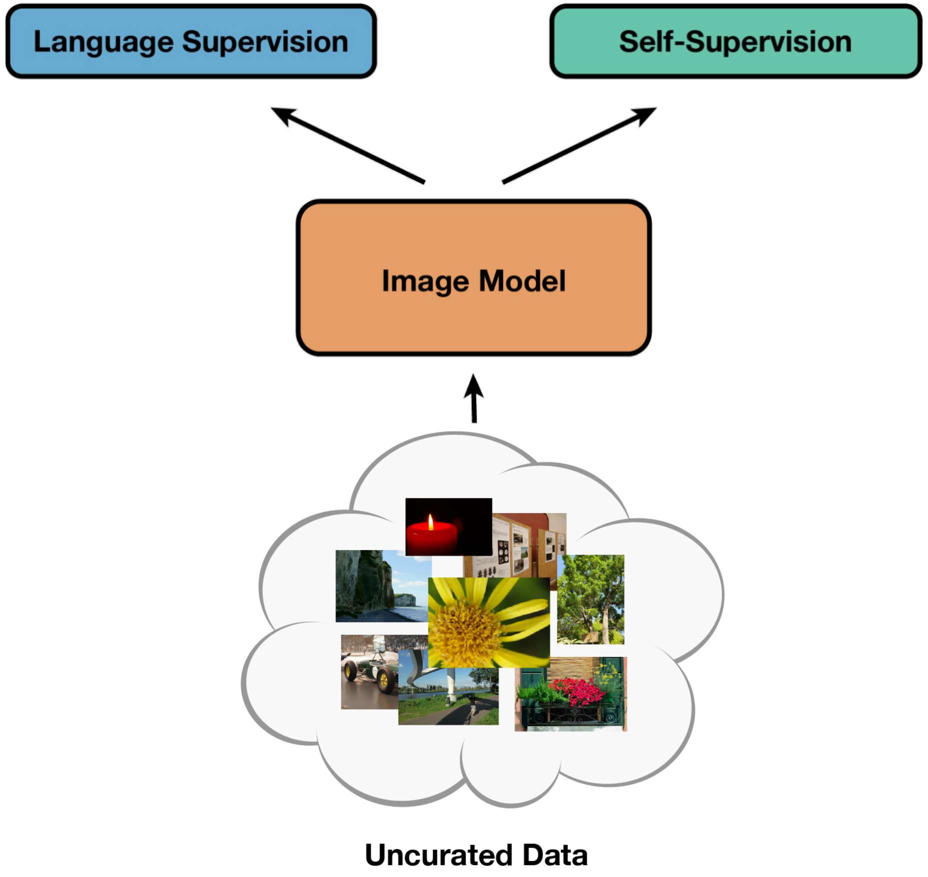


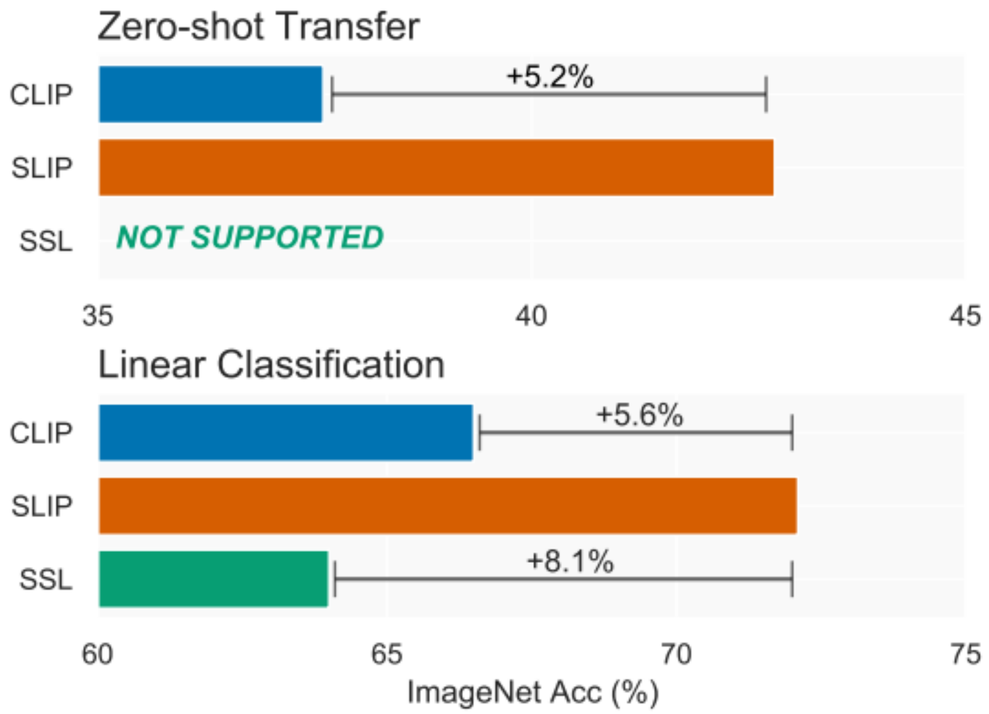

[Mintun, Kirillove, **Xie**, NeurIPS 2021]

These are different concepts and should be studied independently.

# Language-supervision meets self-supervision

[Mu, Kirillov, Wagner, **Xie**, arXiv 2021]



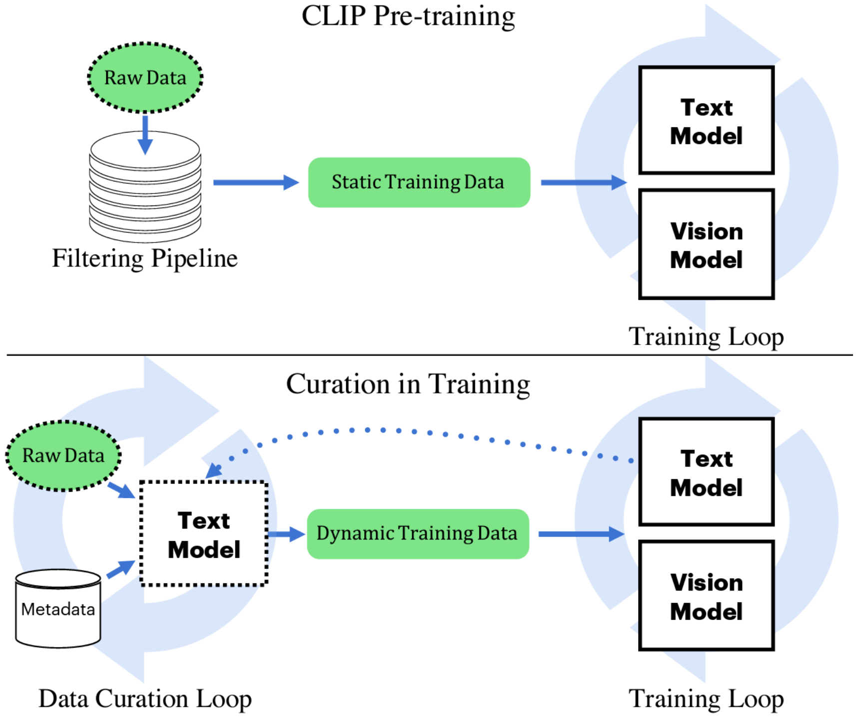




They are complementary and
we can have the best of both worlds!

# Curation-in-Training: Co-designing data and objective

[Hu, Xie, et al. ArXiv 2022]

## Data quality matters

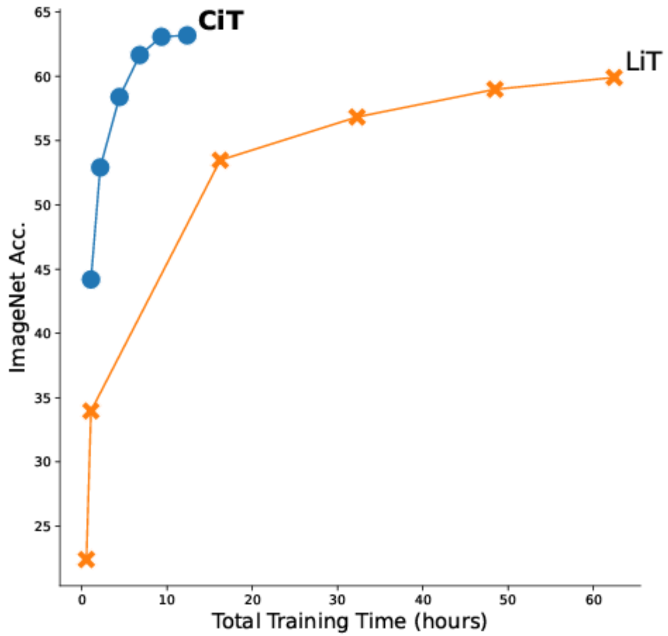
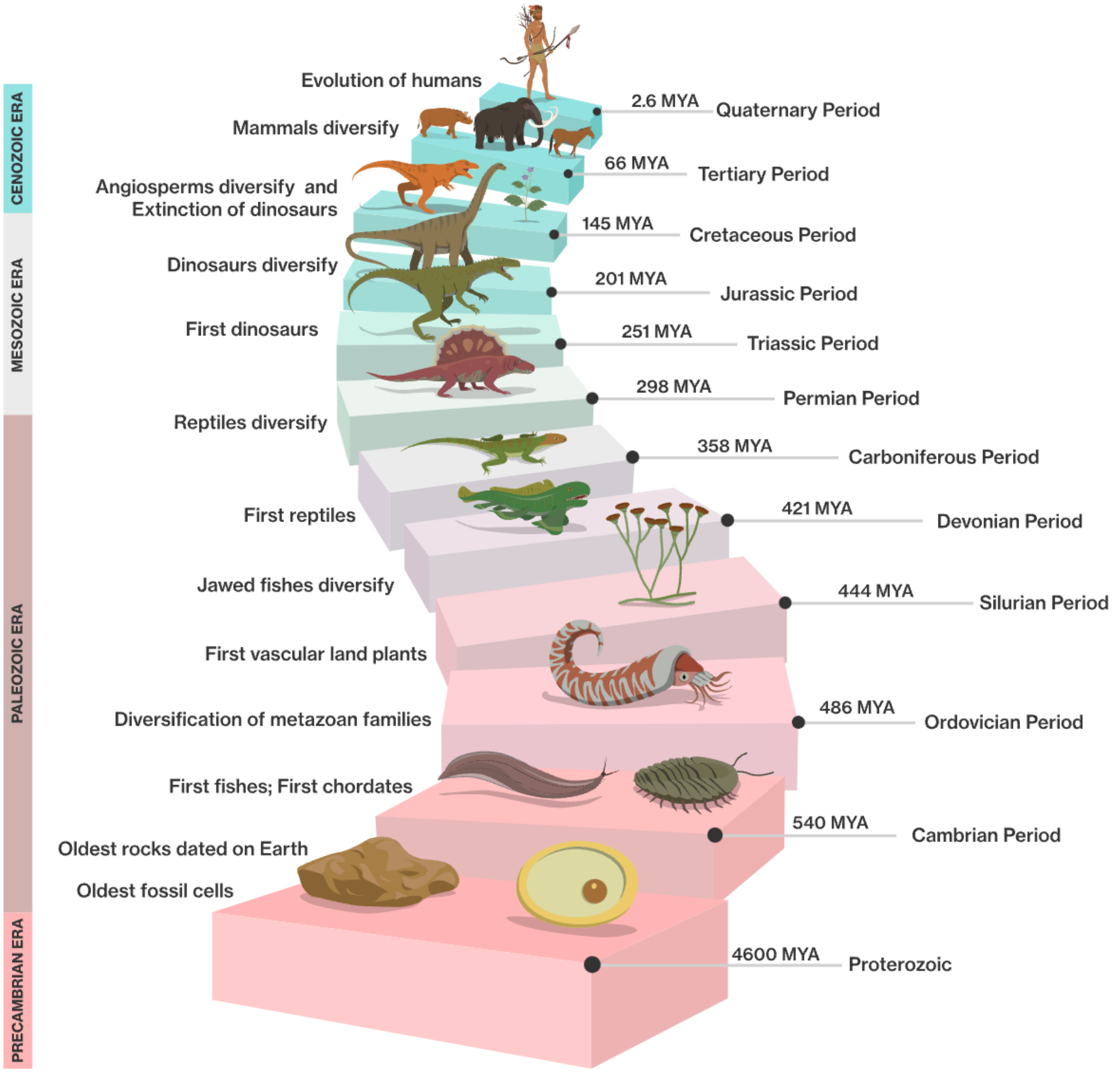




Figure 2. CiT on provides >5× speedup and +3.4% accuracy gain over LiT [38] on AugReg ViT-B/32 vision encoders. Training data is YFCC15M. Models are evaluated at 6 evenly sampled iterations.

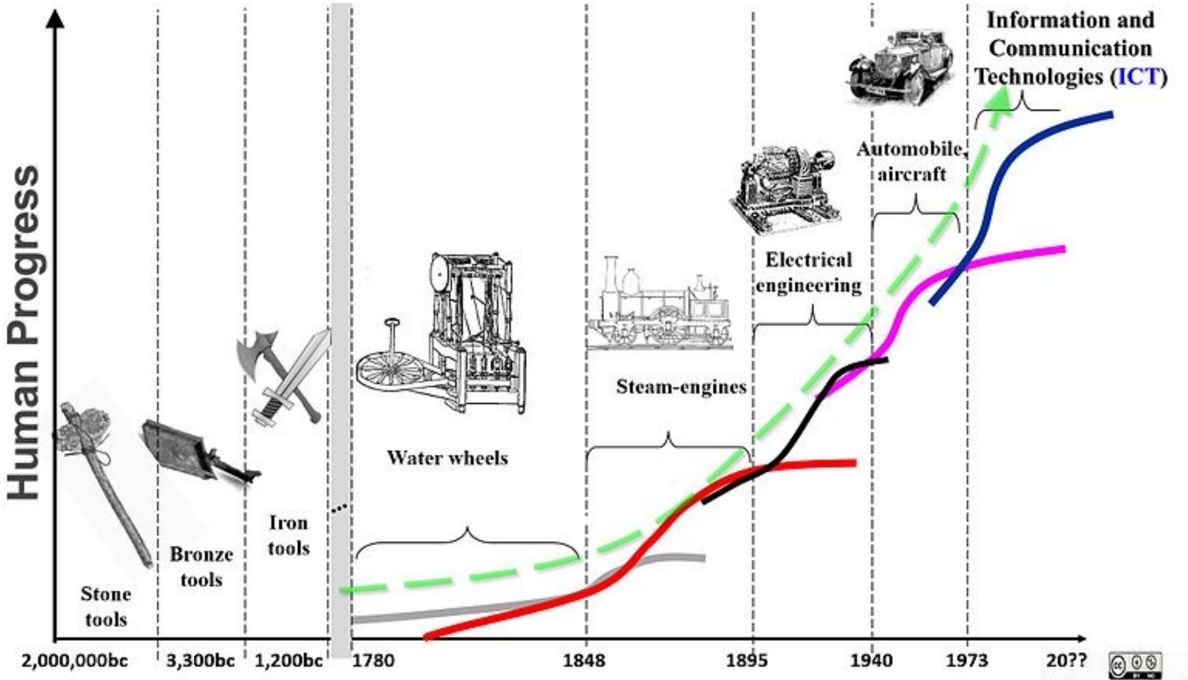
Architecture / Objective / Data

**To summarize:**

- **Co-design matters.**
- **Implication to model robustness.**
- **Diverse tasks, diverse applications.**
- **Great progress, but do we have vision foundation models?**

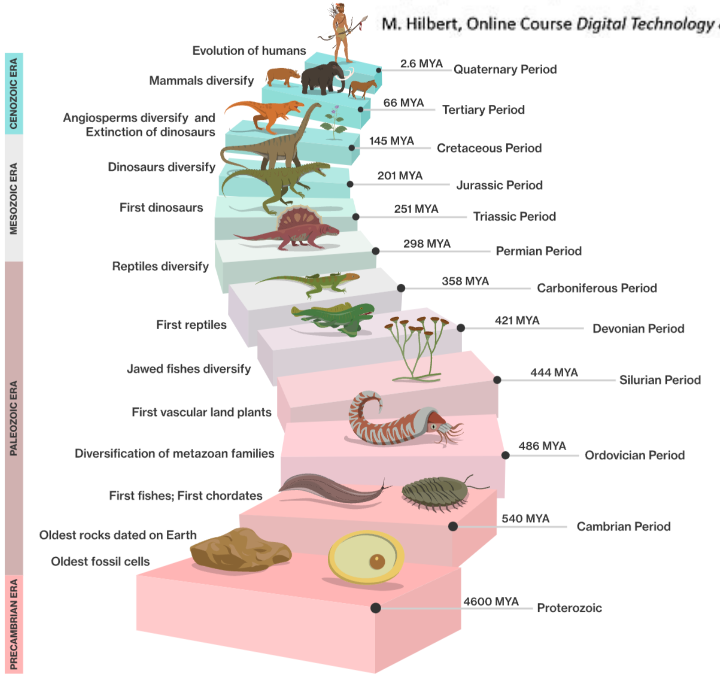



https://www.coursehero.com/sg/introduction-to-biology/understanding-macroevolution/

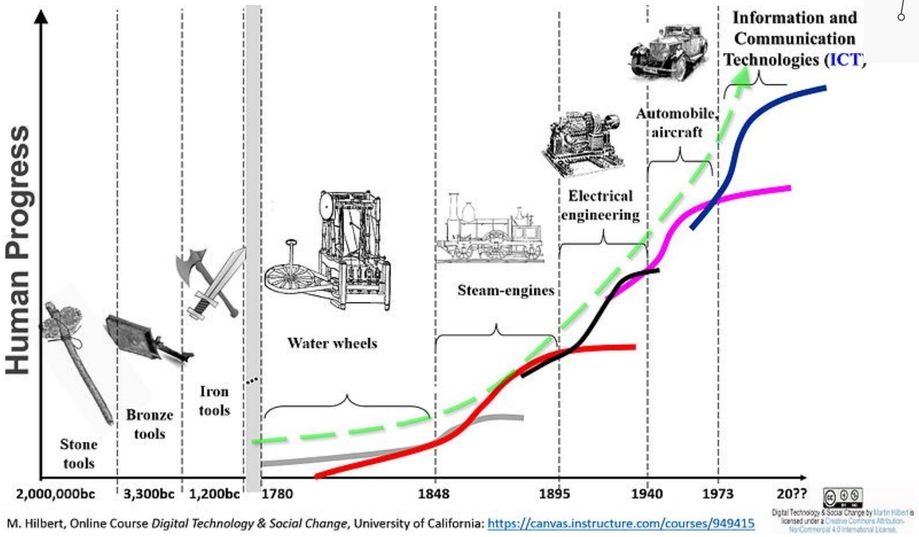


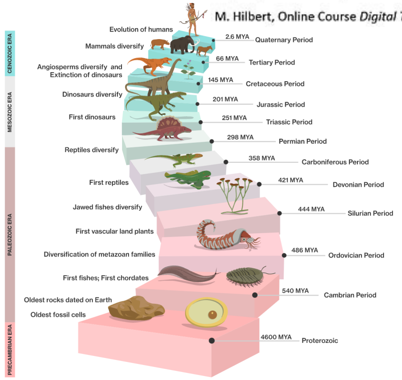

M. Hilbert, Online Course *Digital Technology & Social Change*, University of California: https://canvas.instructure.com/courses/949415





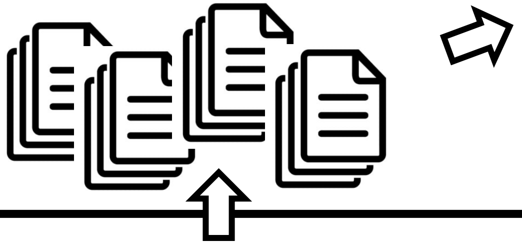
Internet

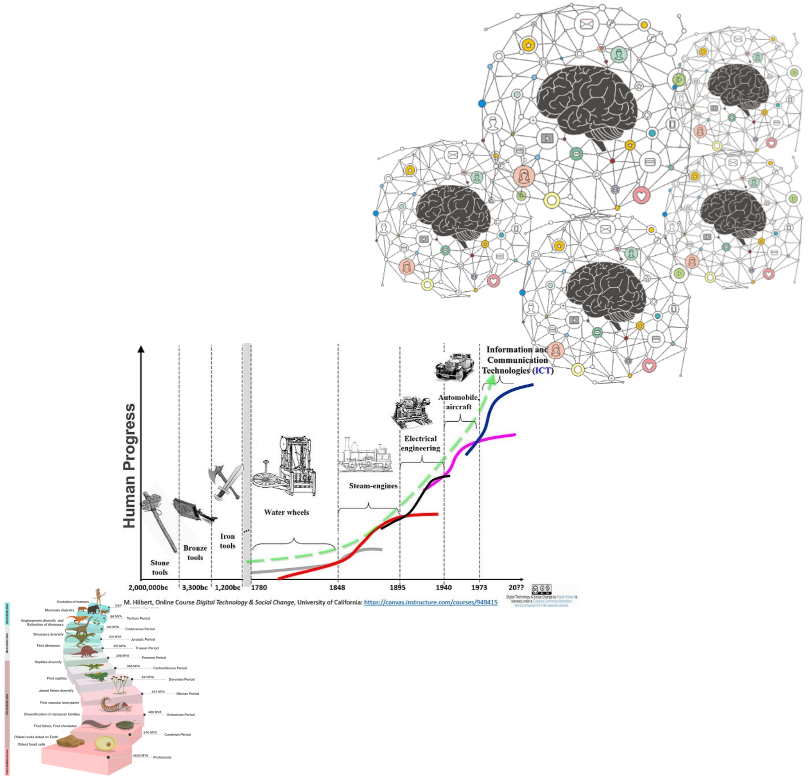

Human Progress
Stone tools
Bronze tools
Iron tools
Water wheels
Steam-engines
Electrical engineering
Automobile, aircraft
Information and Communication Technologies (ICT)
2,000,000bc
3,300bc
1,200bc
1780
1848
1895
1940
1973
20??
M. Hilbert, Online Course Digital Technology & Social Change, University of California: https://canvas.instructure.com/courses/949415

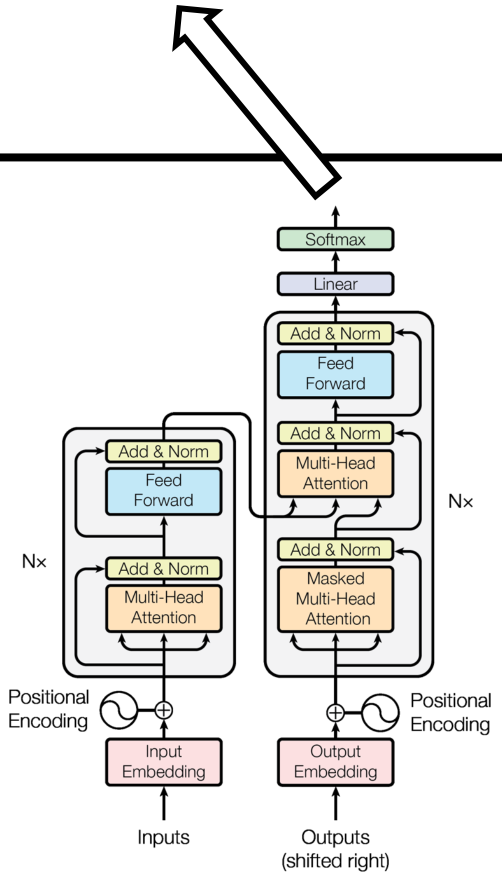

Evolution of humans
Mammals diversify
Angiosperms diversify and Extinction of dinosaurs
Dinosaurs diversify
First dinosaurs
Reptiles diversify
First reptiles
Jawed fishes diversify
First vascular land plants
Diversification of metazoan families
First fishes; First chordates
Oldest rocks dated on Earth
Oldest fossil cells
2.6 MYA Quaternary Period
66 MYA Tertiary Period
145 MYA Cretaceous Period
201 MYA Jurassic Period
251 MYA Triassic Period
298 MYA Permian Period
358 MYA Carboniferous Period
421 MYA Devonian Period
444 MYA Silurian Period
486 MYA Ordovician Period
540 MYA Cambrian Period
4600 MYA Proterozoic


## Next token prediction

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

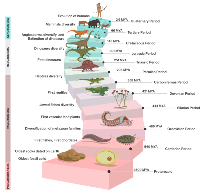


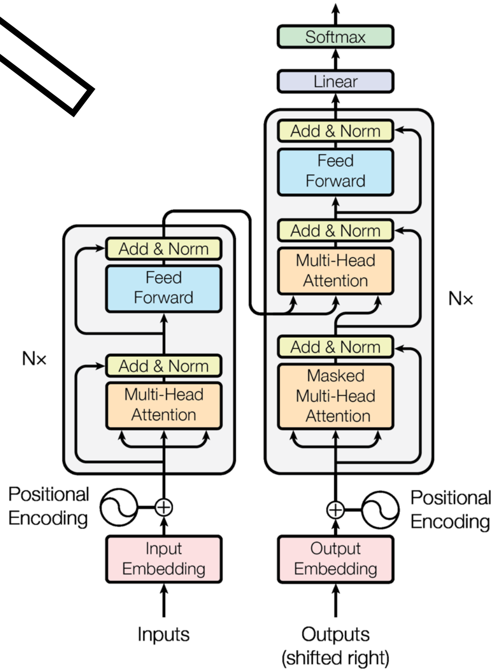


# For ***robust*** vision: shortcut no more…

?





Architecture / Objective / Data

# Thank You!