

Tokens and Transformers

Instructor - Simon Lucey
RVSS - 2026



**AUSTRALIAN
INSTITUTE FOR
MACHINE LEARNING**

Today

- **Tokenization and ConvNets**
- Skip Connections and BatchNorm
- Attention and Transformers
- Self Supervision



$$\mathbf{x} \in \mathbb{R}^{768 \times 320}$$

Reminder: Shallow vs Deep Network

- A shallow learner has only one hidden unit,

$$\theta \eta(\mathbf{W}\mathbf{x} + \boldsymbol{\delta})$$

- A deep learner has more than one hidden unit,

$$\theta \eta(\mathbf{W}^{(1)} \eta(\mathbf{W}^{(0)} \mathbf{x} + \boldsymbol{\delta}^{(0)}) + \boldsymbol{\delta}^{(1)})$$


76800 x 76800 76800 x 1

Reminder: Shallow vs Deep Network

- A shallow learner has only one hidden unit,

$$\theta \eta(\mathbf{W}\mathbf{x} + \delta)$$

- A deep learner **has more than one** hidden unit,

11.8 billion parameters!!!

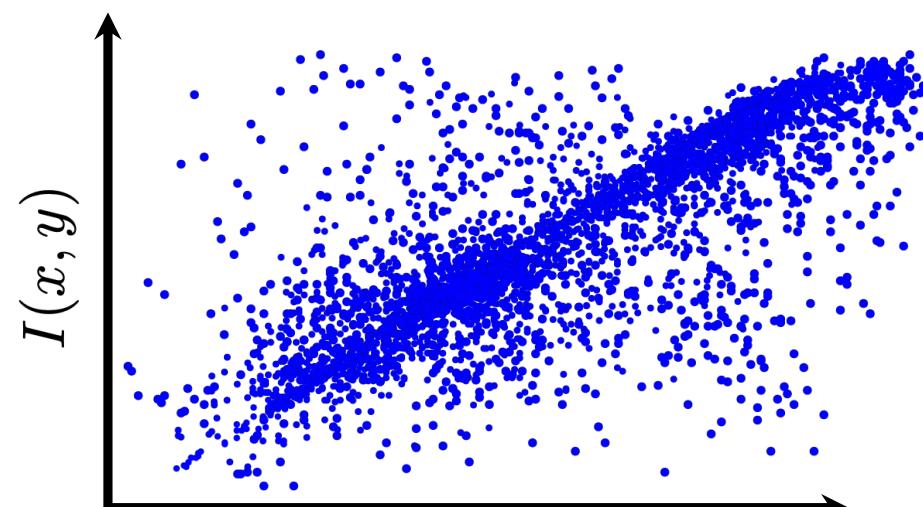
Let's have a play!!!



https://github.com/slucey-cs-cmu-edu/RVSS26/blob/main/Classification_MLP_simple.ipynb



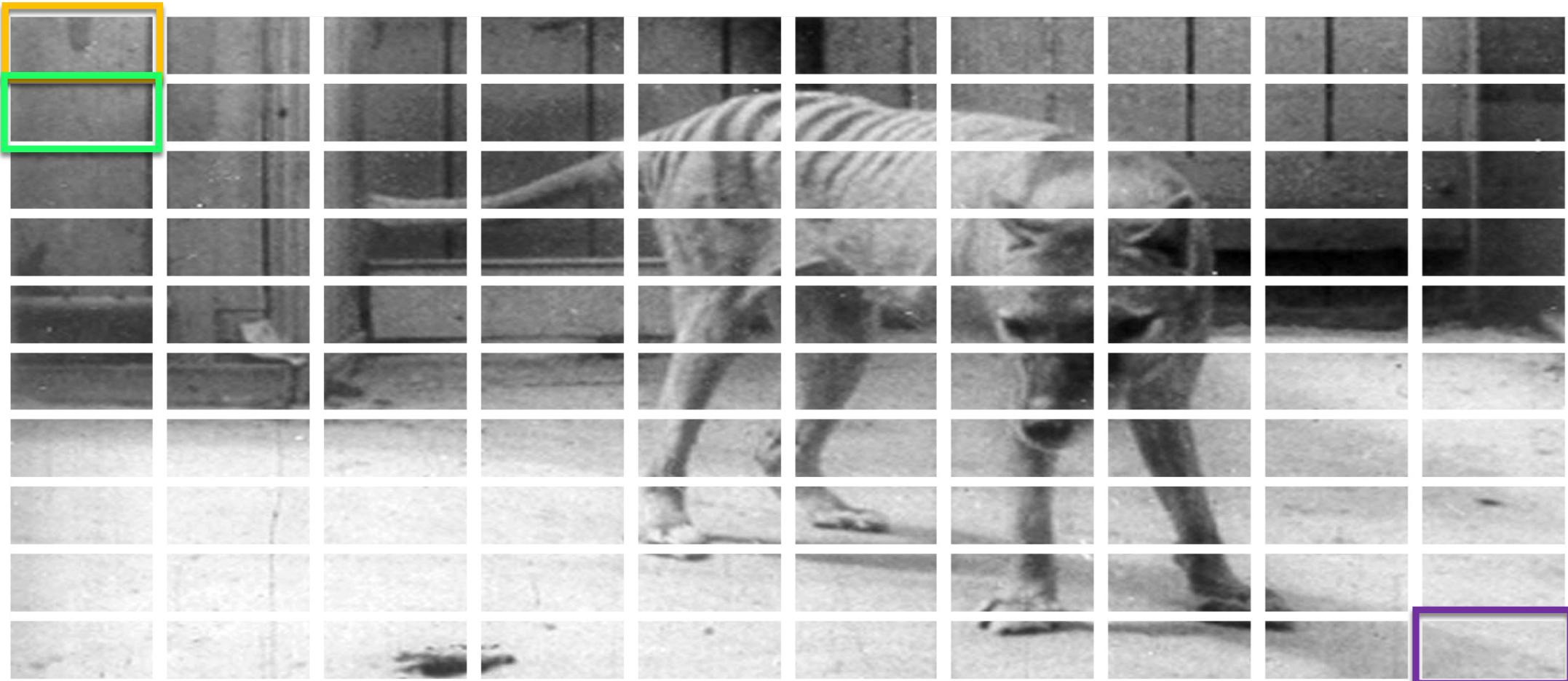
I



$I(x + \theta_y y + \theta_x x)$

Simoncelli & Olshausen 2001

“token”



$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times D}$$

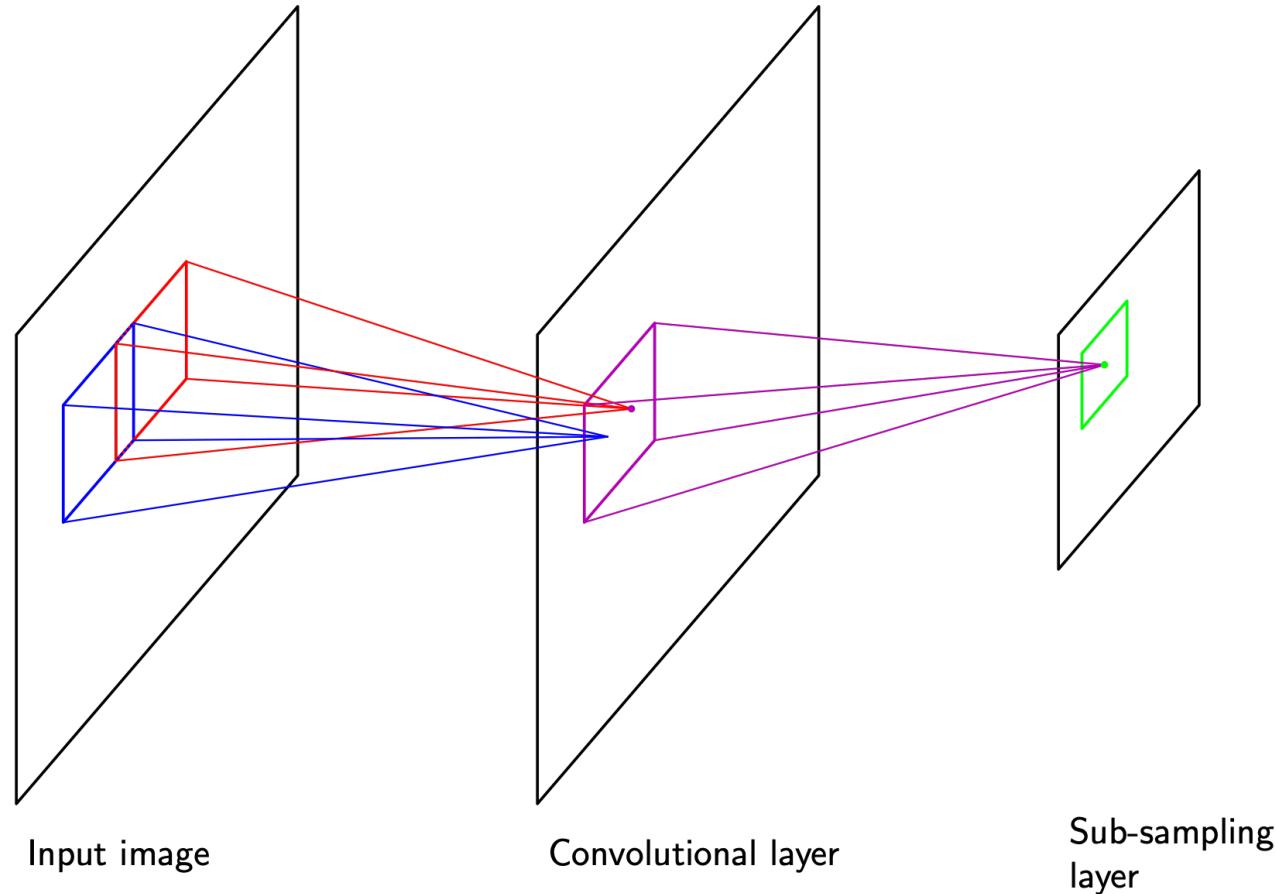
$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times D}$$

$N \rightarrow$ # of tokens

$D \rightarrow$ # of channels

$$\mathbf{x}_n = \text{vec}\left(\begin{array}{c} \text{[image]} \\ \text{"token"} \end{array}\right)$$

Convolutional Neural Network



LeCun 1980

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called “dropout” that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

Reminder: Shallow vs Deep Network

- A shallow learner has only one hidden unit,

$$\theta \eta(\mathbf{W}\mathbf{x} + \boldsymbol{\delta})$$

- A deep learner has more than one hidden unit,

$$\theta \eta(\mathbf{W}^{(1)} \eta(\mathbf{W}^{(0)} \mathbf{x} + \boldsymbol{\delta}^{(0)}) + \boldsymbol{\delta}^{(1)})$$

$$\theta \eta(\mathbf{W}^{(1)} \eta(\mathbf{W}^{(0)} \mathbf{x} + \boldsymbol{\delta}^{(0)}) + \boldsymbol{\delta}^{(1)})$$

 “tokenization”

$$\begin{aligned}\mathbf{W}^{(0)} \mathbf{x} &= \text{vec}(\mathbf{U}^{(0)} \mathbf{X} \mathbf{V}^{(0)}) \\ &= \mathbf{V}^{(0)T} \otimes^{\text{“spatial mixing”}} \mathbf{U}^{(0)} \text{vec}(\mathbf{X}) \\ &= [\mathbf{V}^{(0)T} \otimes^{\text{“channel mixing”}} \mathbf{U}^{(0)}] \mathbf{x}\end{aligned}$$

why? $N^2 + D^2 \gg N^2 D^2$ ($\mathbf{X} \in \mathbb{R}^{N \times D}$)

Let's have another play!!!

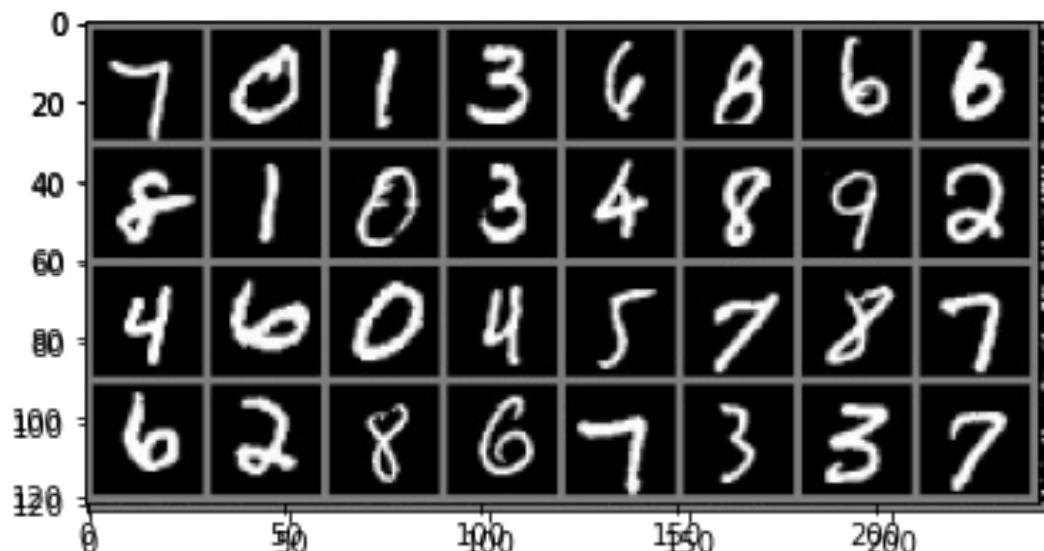


https://colab.research.google.com/github/slucey-cs-cmu-edu/RVSS26/blob/main/Classification_TokenMix_simple.ipynb

Some things to try!!!

- What happens to performance if you use a linear function?
- What happens when you permute the pixels?

```
from numpy.random import permutation
idx_permute = torch.from_numpy(permutation(784))
transform = transforms.Compose([transforms.ToTensor(),
                               transforms.Lambda(lambda x: x.view(-1)[idx_permute].view(1, 28, 28) ),
                               transforms.Normalize((0.5,), (0.5,)),
                               ])
```

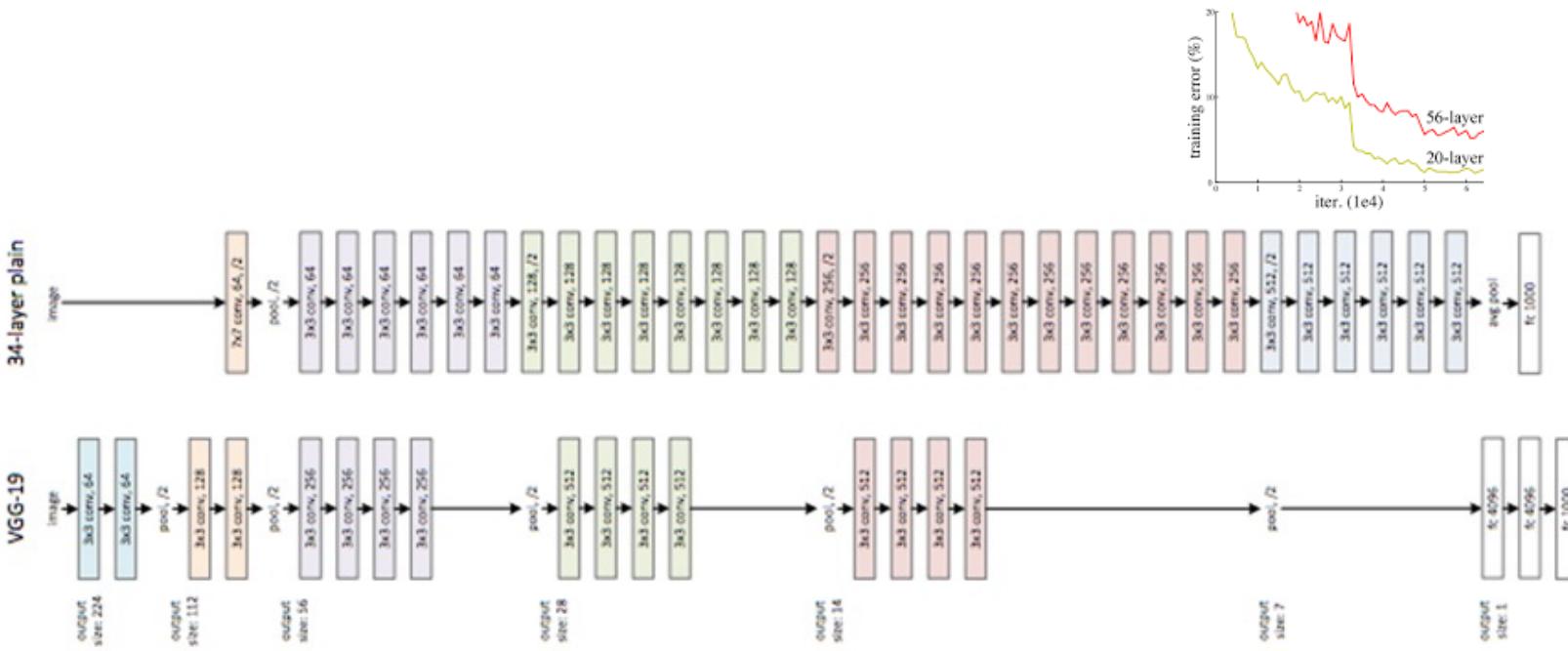


Today

- Tokenization and ConvNets
- **Skip Connections and BatchNorm**
- Attention and Transformers

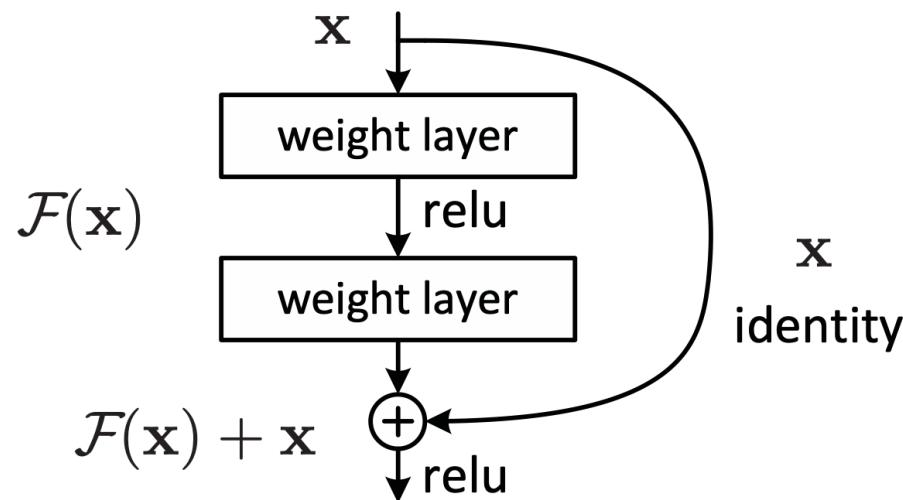
Neural Net : Deeper is Better?

- Very deep network -- training error increases.



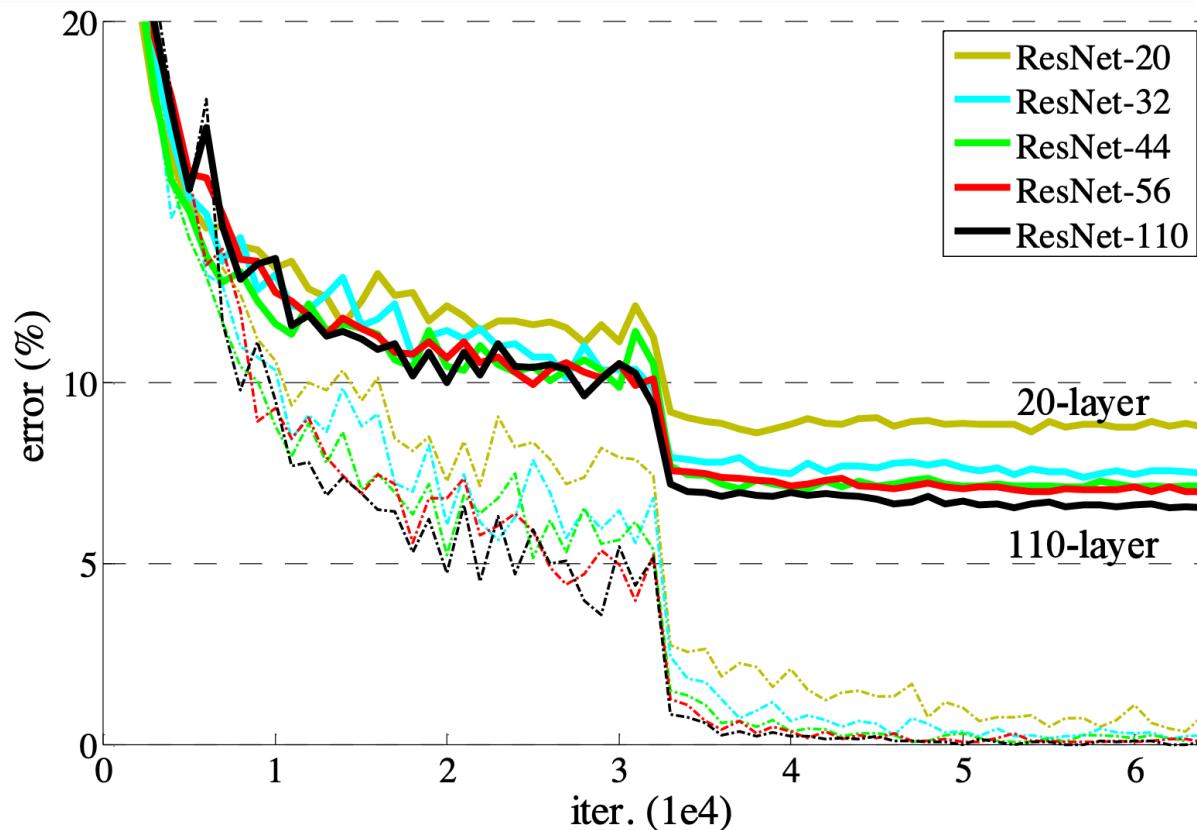
How Deep?

- Network depth is crucial for good performance (e.g. ImageNet).
- Counter intuitively, naively trained deeper networks tend to have higher train error than shallow networks.
- Innovation of residual learning has greatly helped with this.



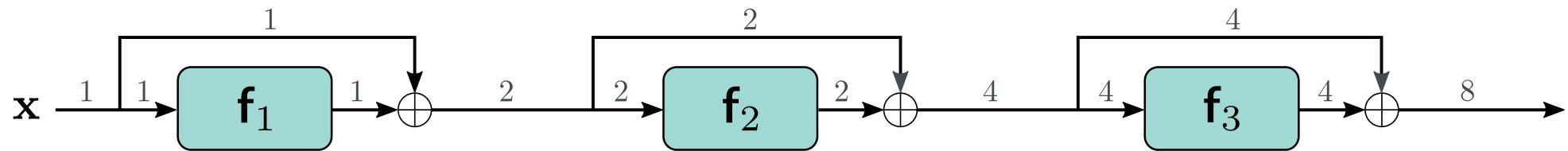
K. He, et al. "Deep residual learning for image recognition", CVPR 2016.

How Deep?



K. He, et al. "Deep residual learning for image recognition", CVPR 2016.

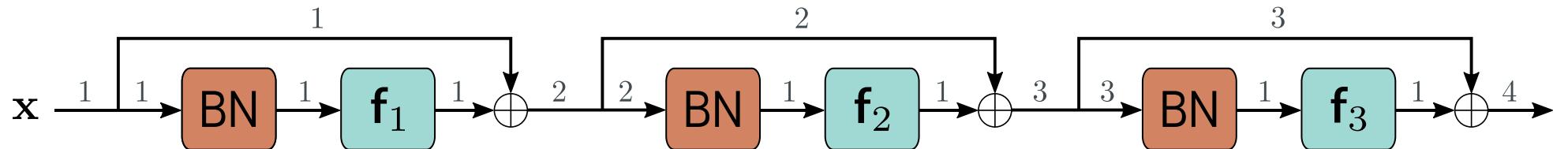
Initialization Problem



Try example:-

[https://colab.research.google.com/github/udlbook/udlbook/blob/main/Notebooks/Chap11/11_3_Batch Normalization.ipynb](https://colab.research.google.com/github/udlbook/udlbook/blob/main/Notebooks/Chap11/11_3_Batch%20Normalization.ipynb)

Batch Norm



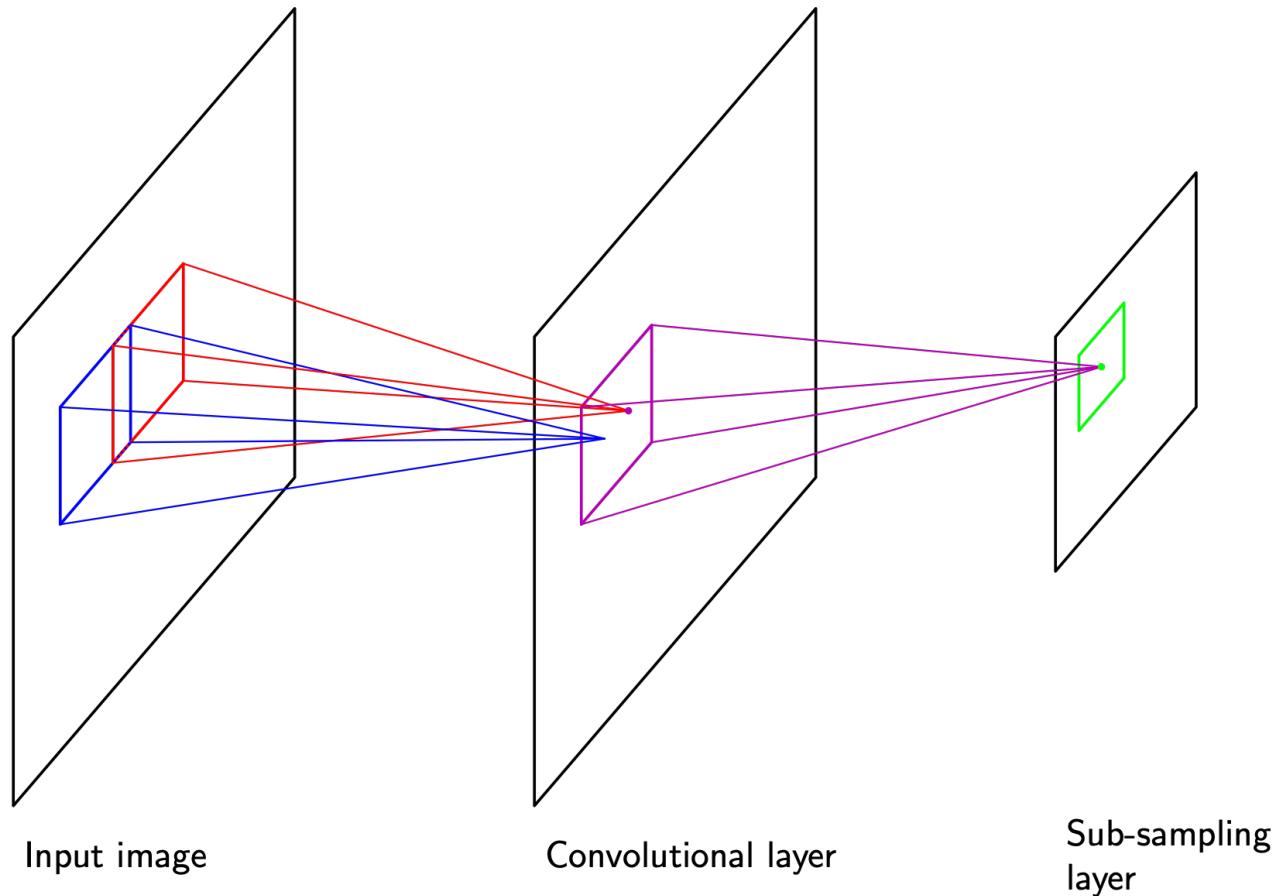
$$m_h = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i \quad h_i \leftarrow \frac{h_i - m_h}{s_h + \epsilon}$$

$$s_h = \sqrt{\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (h_i - m_h)^2}, \quad h_i \leftarrow \gamma h_i + \delta$$

Today

- Tokenization and ConvNets
- Skip Connections and BatchNorm
- **Attention and Transformers**
- Self Supervision

Convolutional = A Type of Attention??



LeCun 1980

Attention is all you need!!

- Convolution is hard-coded attention.
 - Could there be any way to learn it from data?
-

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

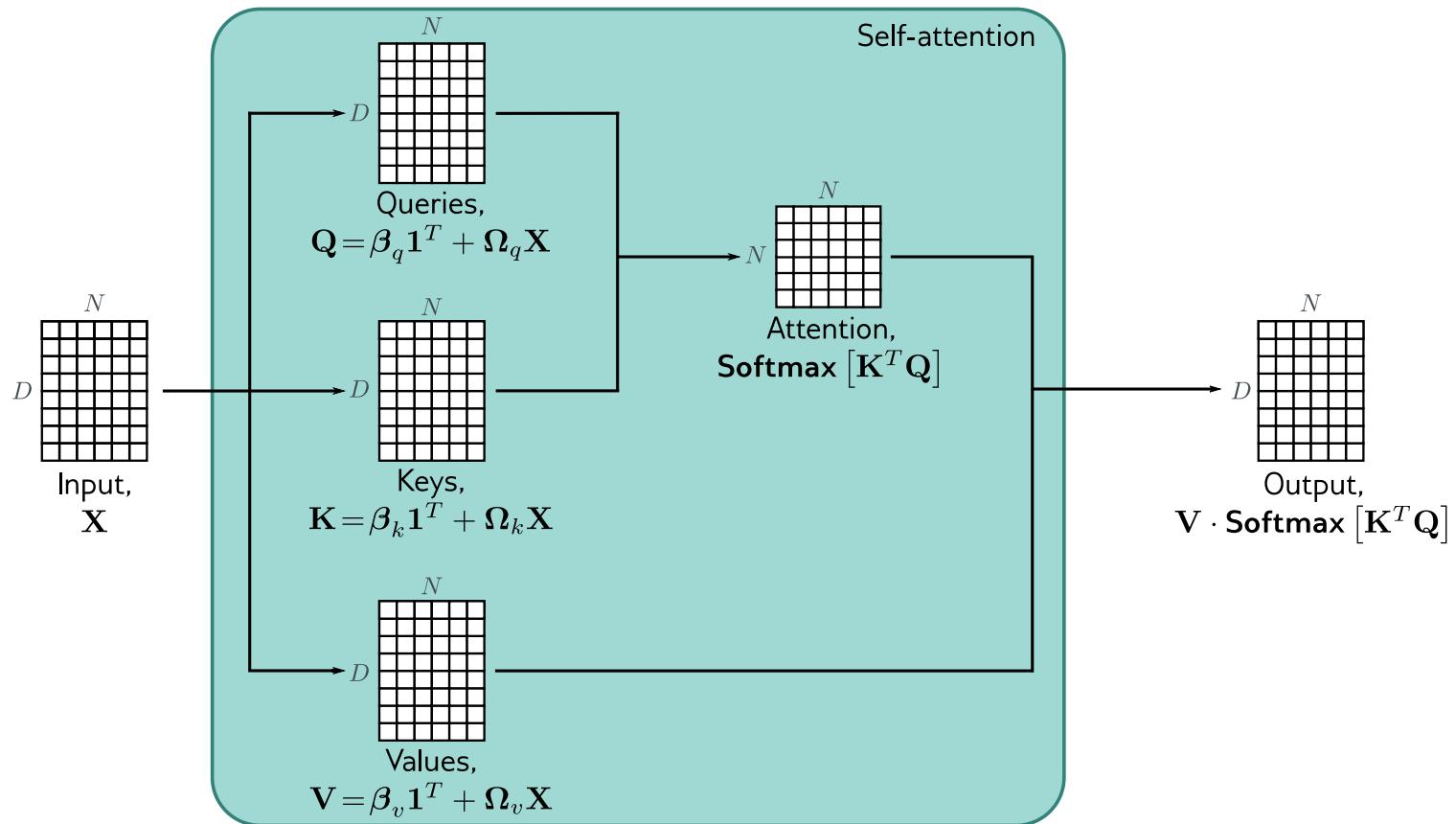
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention

A. Vaswani et al. “Attention is all you need”, NeurIPS 2017.

Soft-Max Attention

$$\begin{aligned} a[\mathbf{x}_m, \mathbf{x}_n] &= \text{softmax}_m [\mathbf{k}_\bullet^T \mathbf{q}_n] \\ &= \frac{\exp [\mathbf{k}_m^T \mathbf{q}_n]}{\sum_{m'=1}^N \exp [\mathbf{k}_{m'}^T \mathbf{q}_n]}, \end{aligned}$$

What is Attention?



What is Attention?

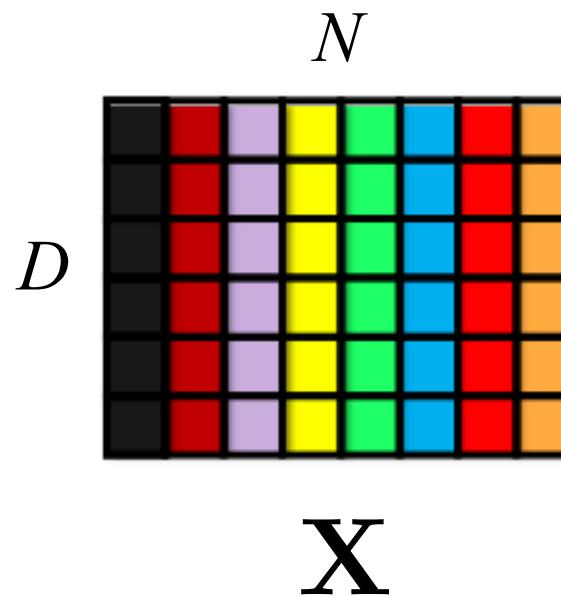
$$\mathbf{V}[\mathbf{X}] = \beta_v \mathbf{1}^T + \Omega_v \mathbf{X}$$

$$\mathbf{Q}[\mathbf{X}] = \beta_q \mathbf{1}^T + \Omega_q \mathbf{X}$$

$$\mathbf{K}[\mathbf{X}] = \beta_k \mathbf{1}^T + \Omega_k \mathbf{X},$$

$$\mathbf{Sa}[\mathbf{X}] = \mathbf{W}_t[\mathbf{X}] \cdot \text{Softmax}\left[\mathbf{K}[\mathbf{X}]^T \mathbf{Q}[\mathbf{X}] \right],$$

Permutating X

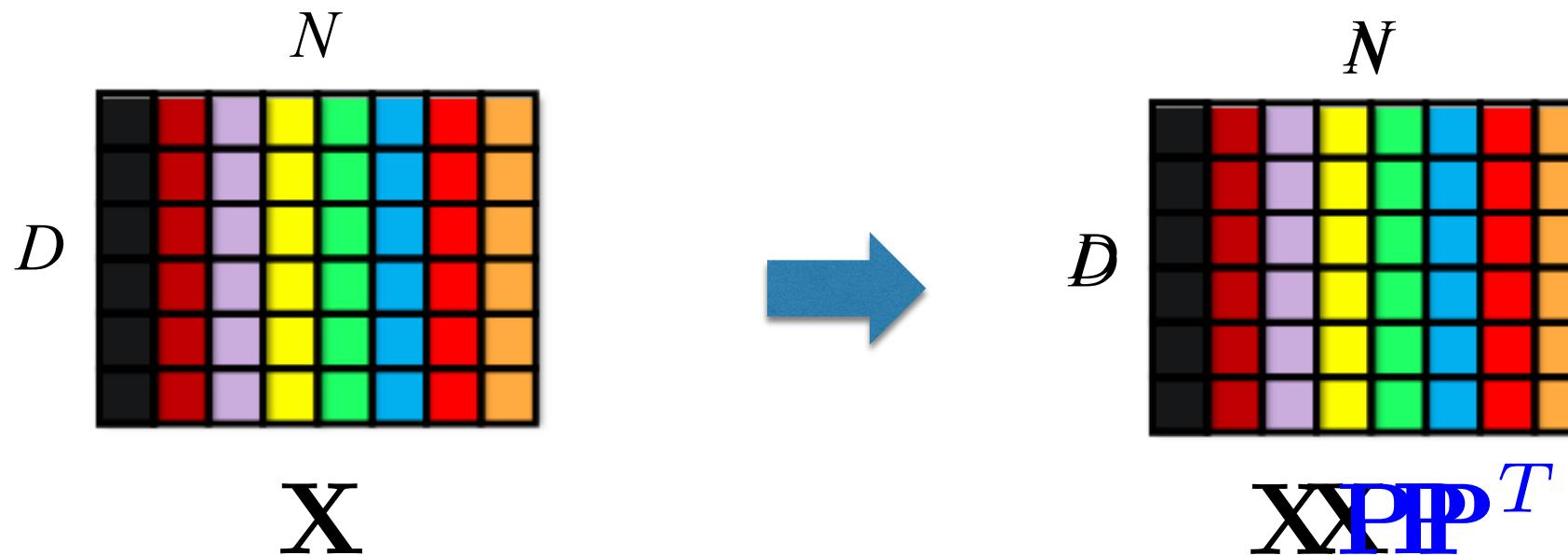


N

D

$\mathbf{X}\mathbf{P}$

Permutating X



Permutation Equivariance?

$$\text{Sa}(\mathbf{X}) = \text{Sa}(\mathbf{X}\mathbf{P}\mathbf{P}^T)$$



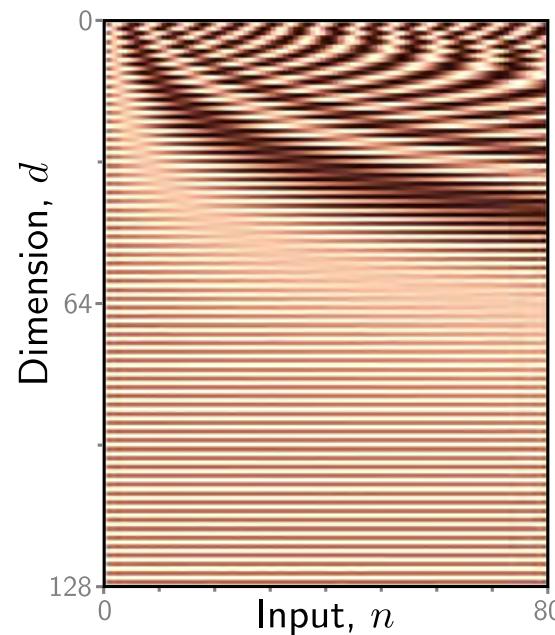
$$\text{Sa}(\mathbf{X}) = \text{Sa}(\mathbf{X}\mathbf{P})\mathbf{P}^T$$

Position information ignored!!!

Positional Encoding

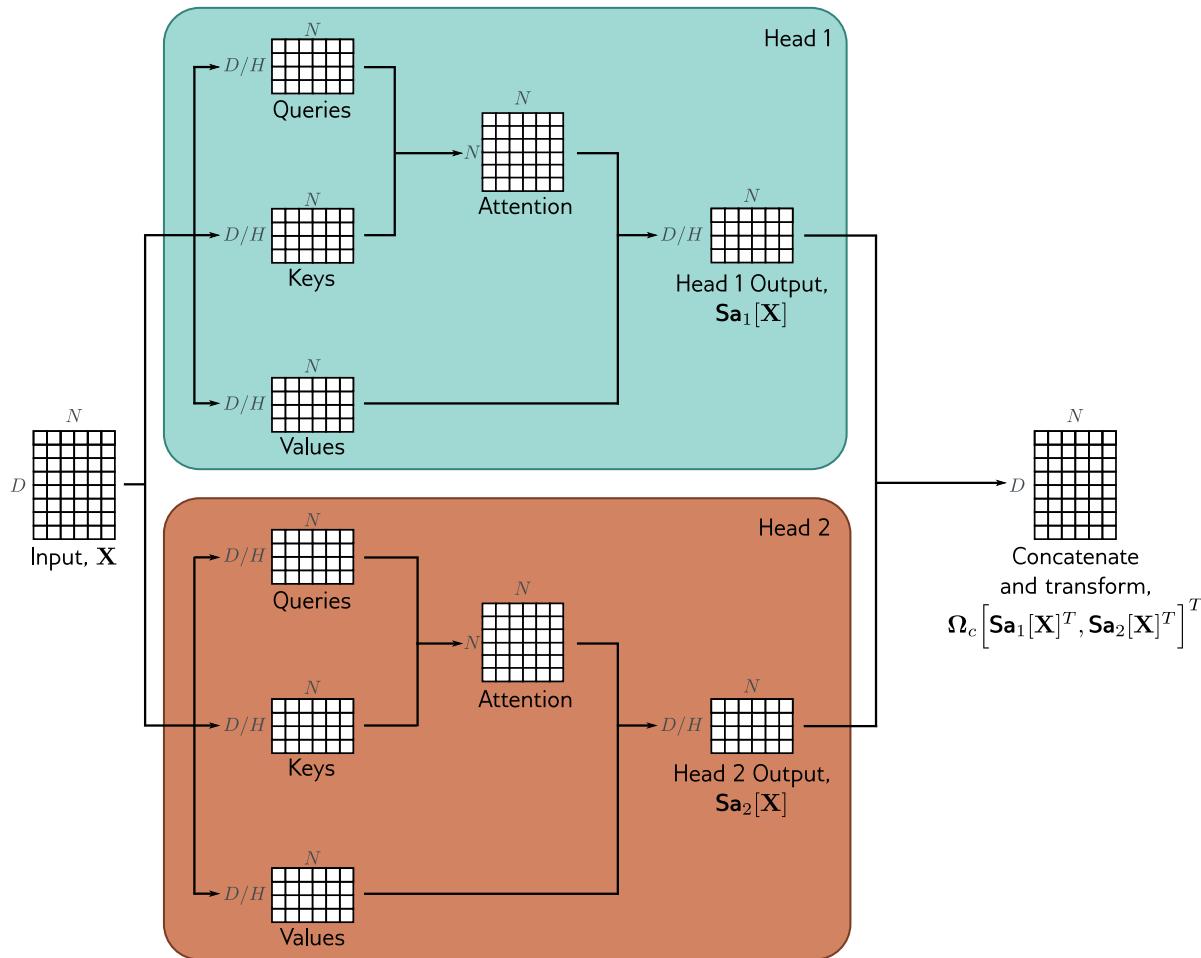
$$X \rightarrow X + P$$

$$P =$$

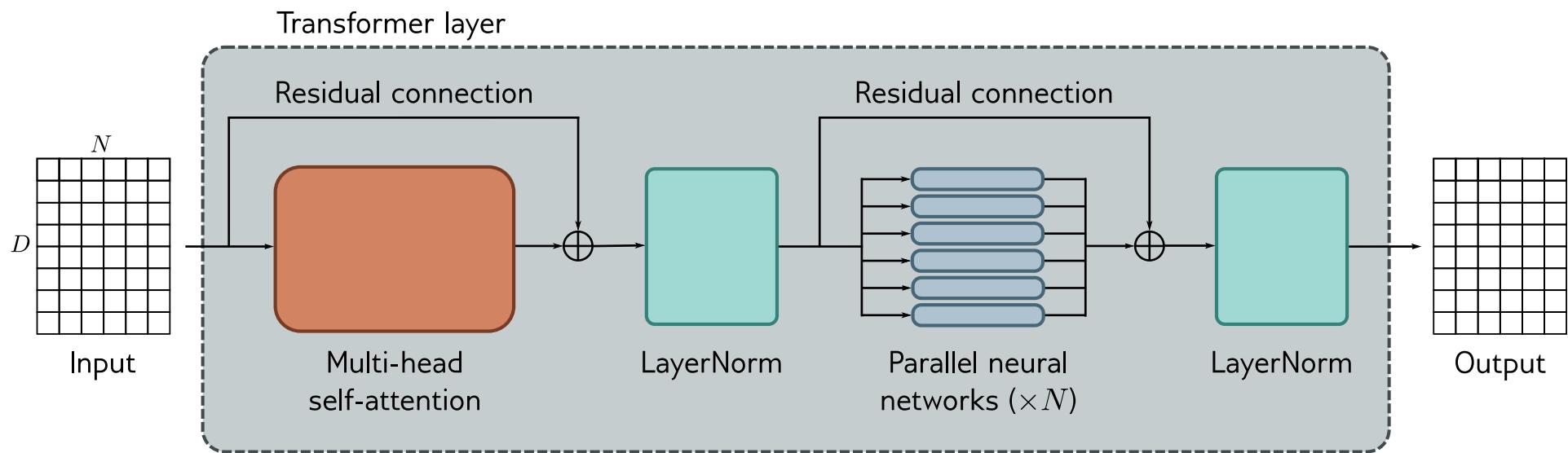


Prince “Understanding Deep Learning” 2023.

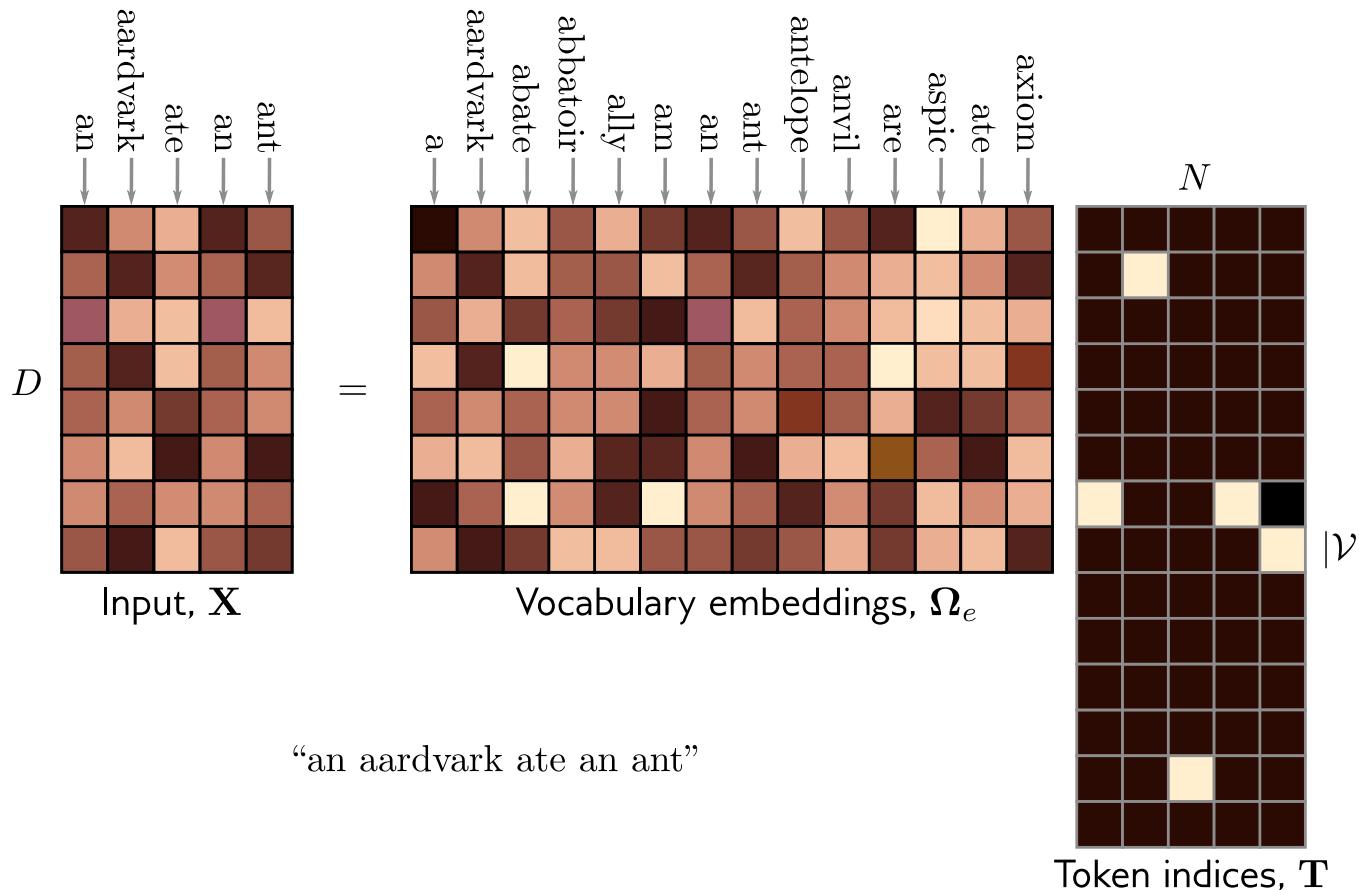
Multi-Head Attention



Transformer Layer



Language + Transformers



Prince “Understanding Deep Learning” 2023.

Vision Transformer (ViT)

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy^{*,†}, Lucas Beyer^{*}, Alexander Kolesnikov^{*}, Dirk Weissenborn^{*},
Xiaohua Zhai^{*}, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby^{*,†}

*equal technical contribution, †equal advising

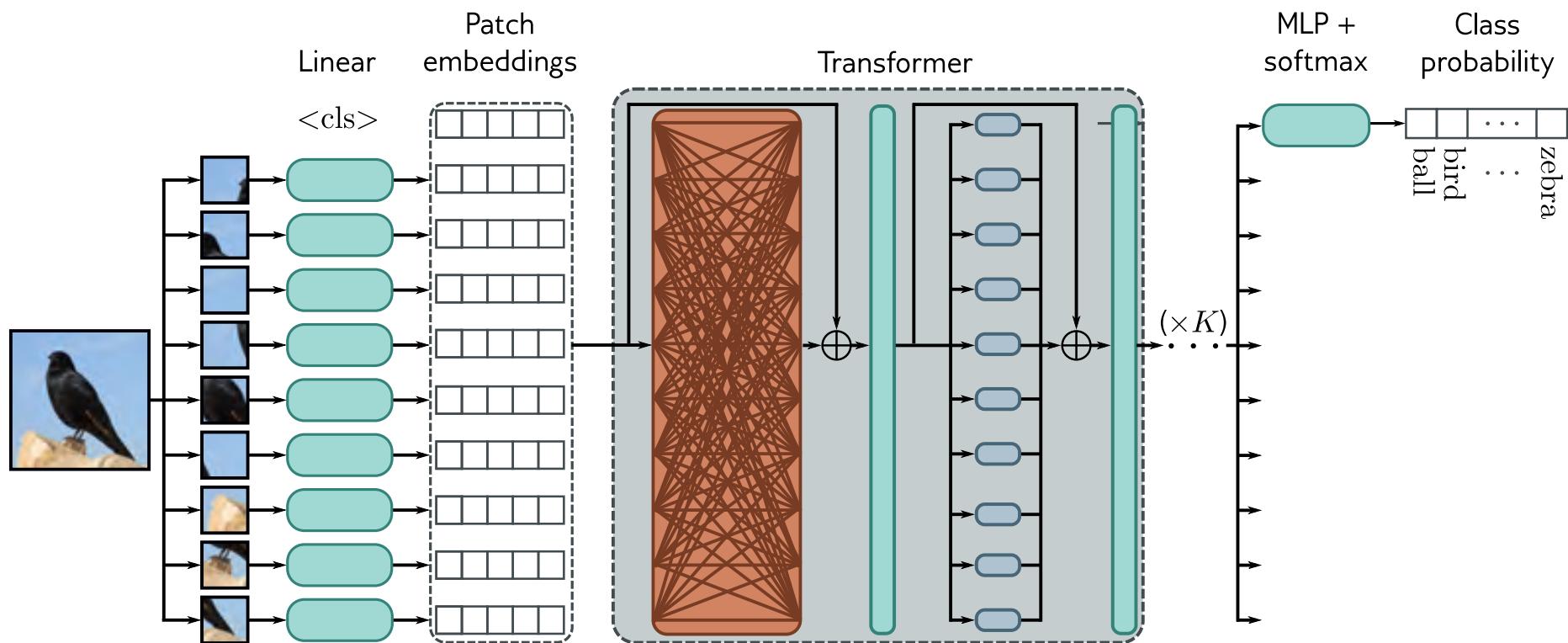
Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

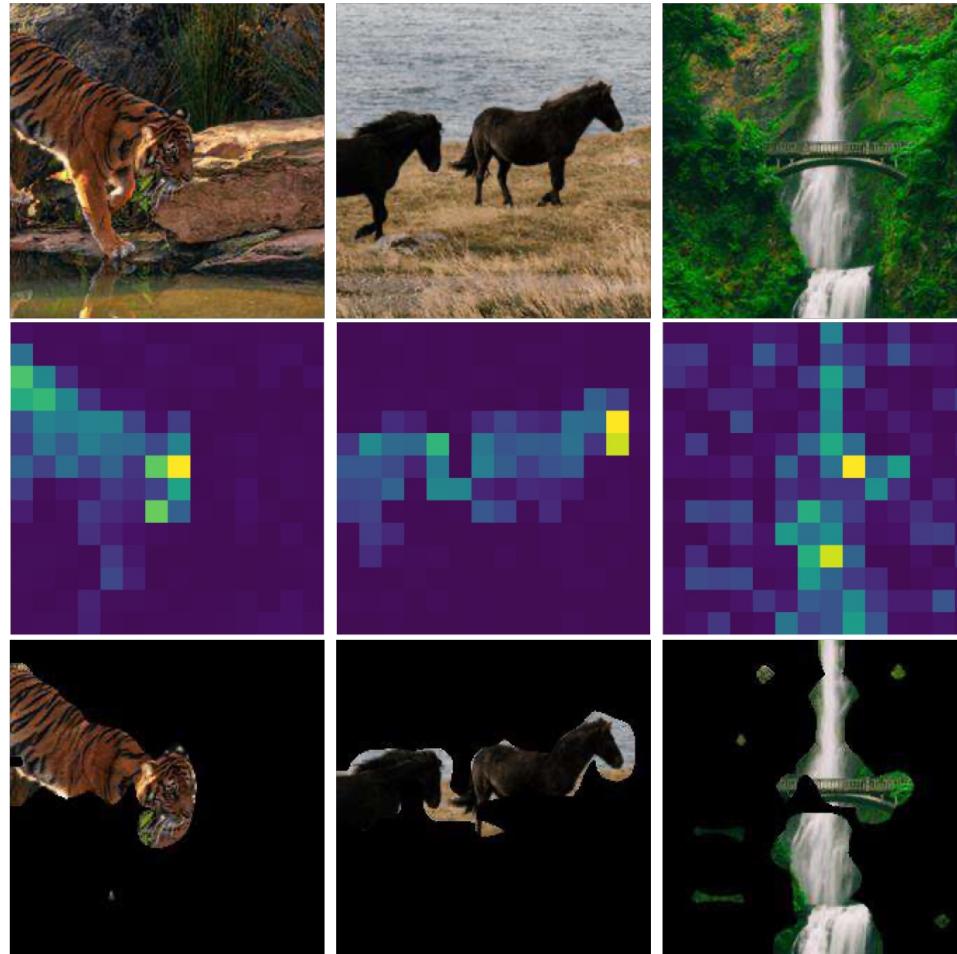
ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.). Vision Transformer (ViT) attains excellent

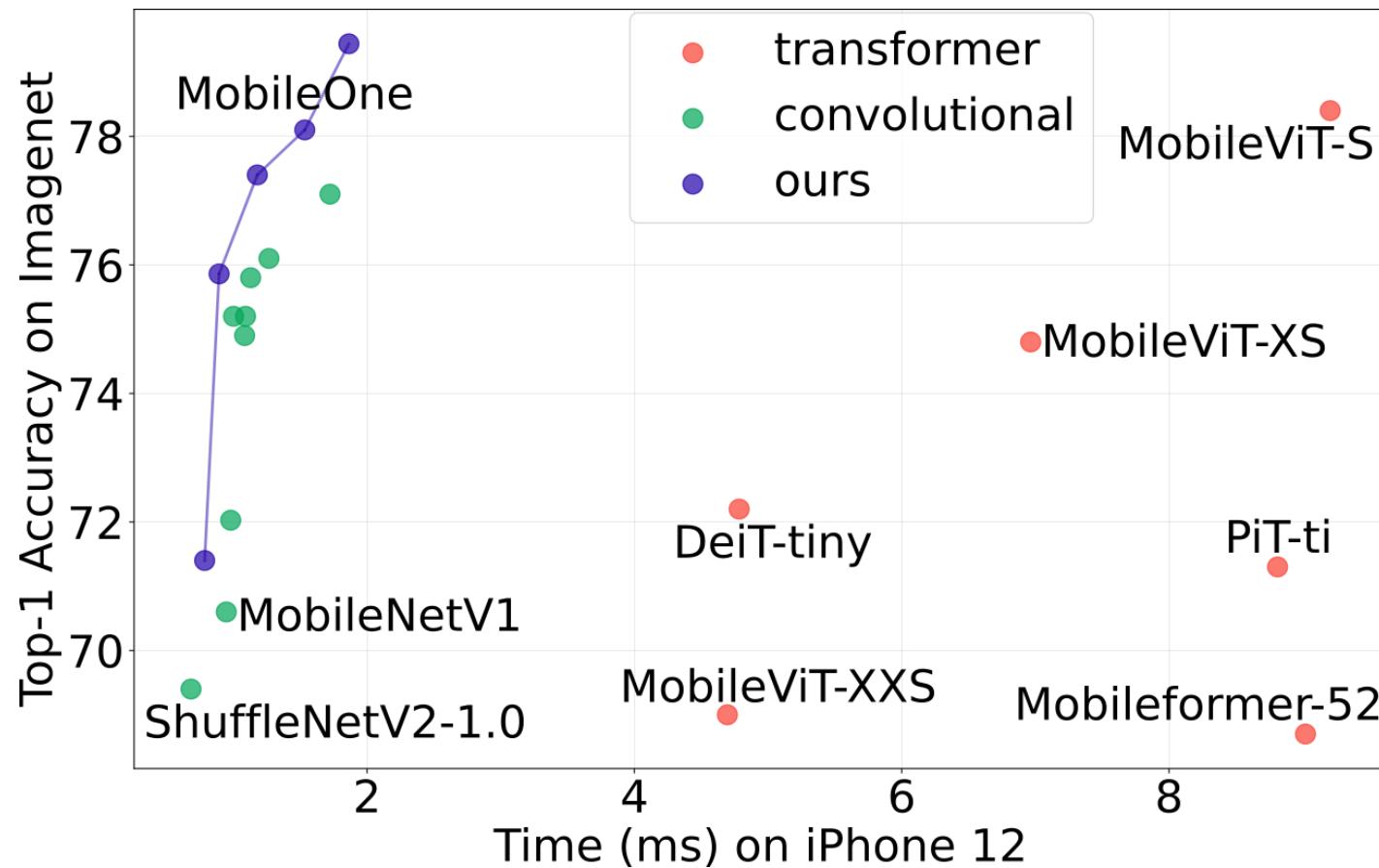
Vision Transformer (ViT)



Visualizing Attention

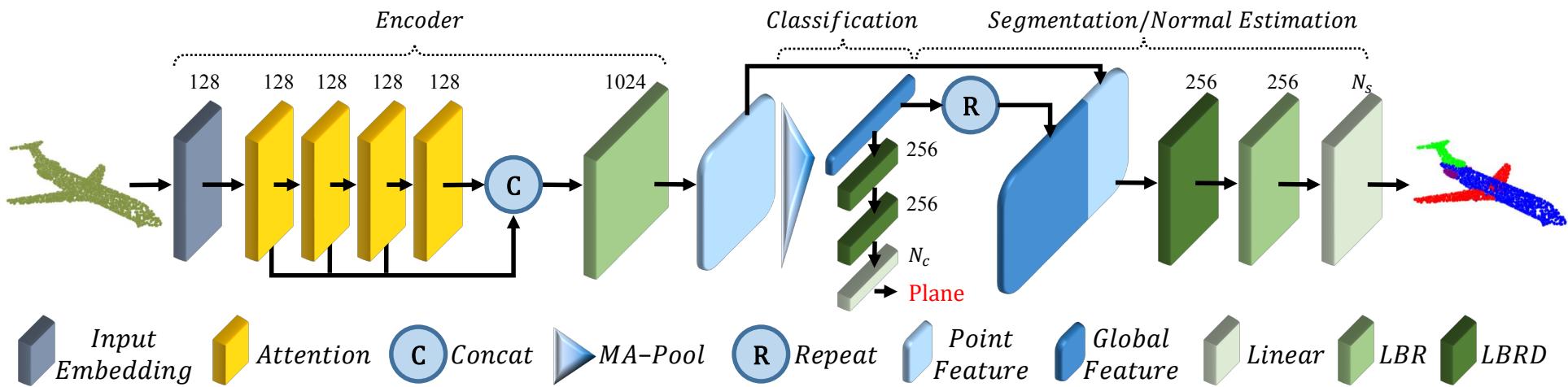


Transformers are Getting Faster!!!



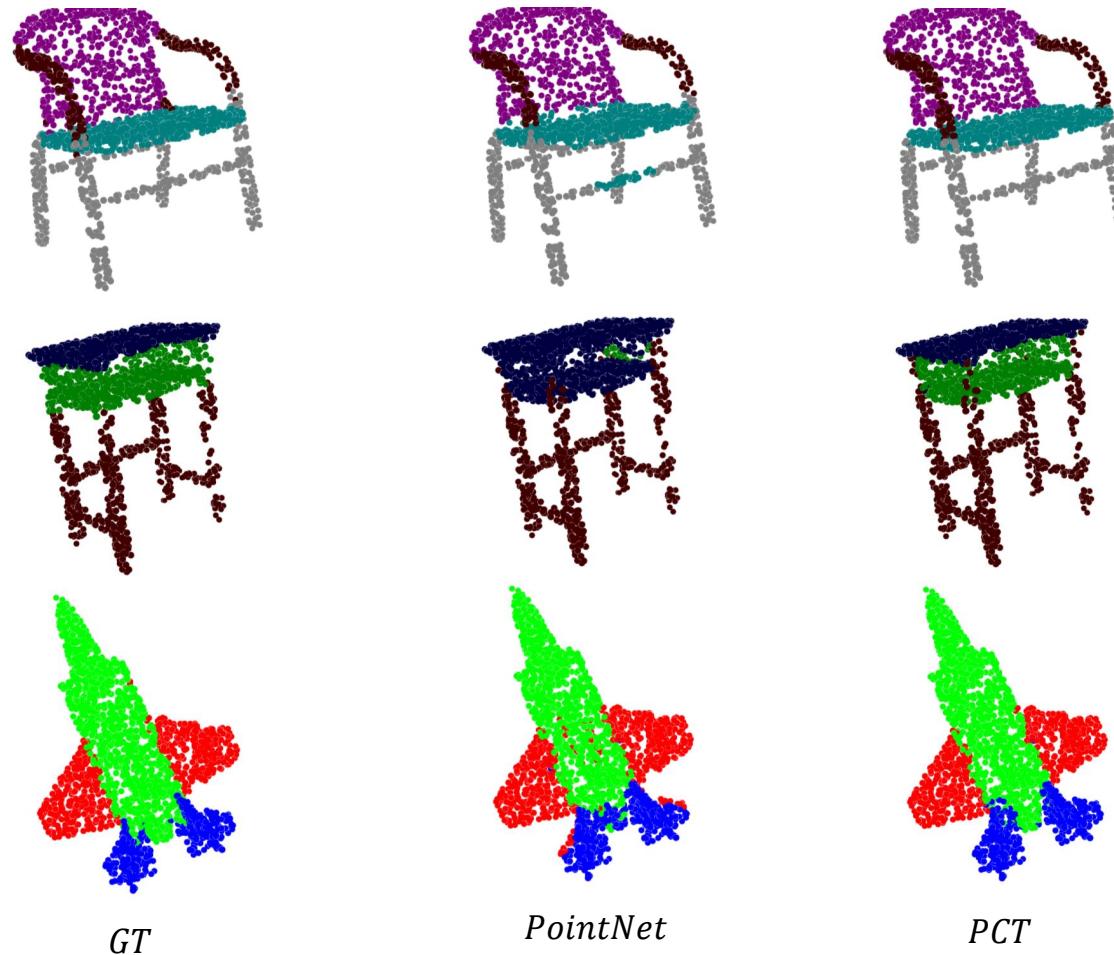
Vasu et al. "MobileOne: An Improved One millisecond Mobile Backbone" In CVPR 2023.

Point Cloud Transformers



M. Gu, “PCT: Point Cloud Transformer” in Comp. Visual Media 7, 187–199 (2021).

Point Cloud Transformers



M. Gu, “PCT: Point Cloud Transformer” in Comp. Visual Media 7, 187–199 (2021).

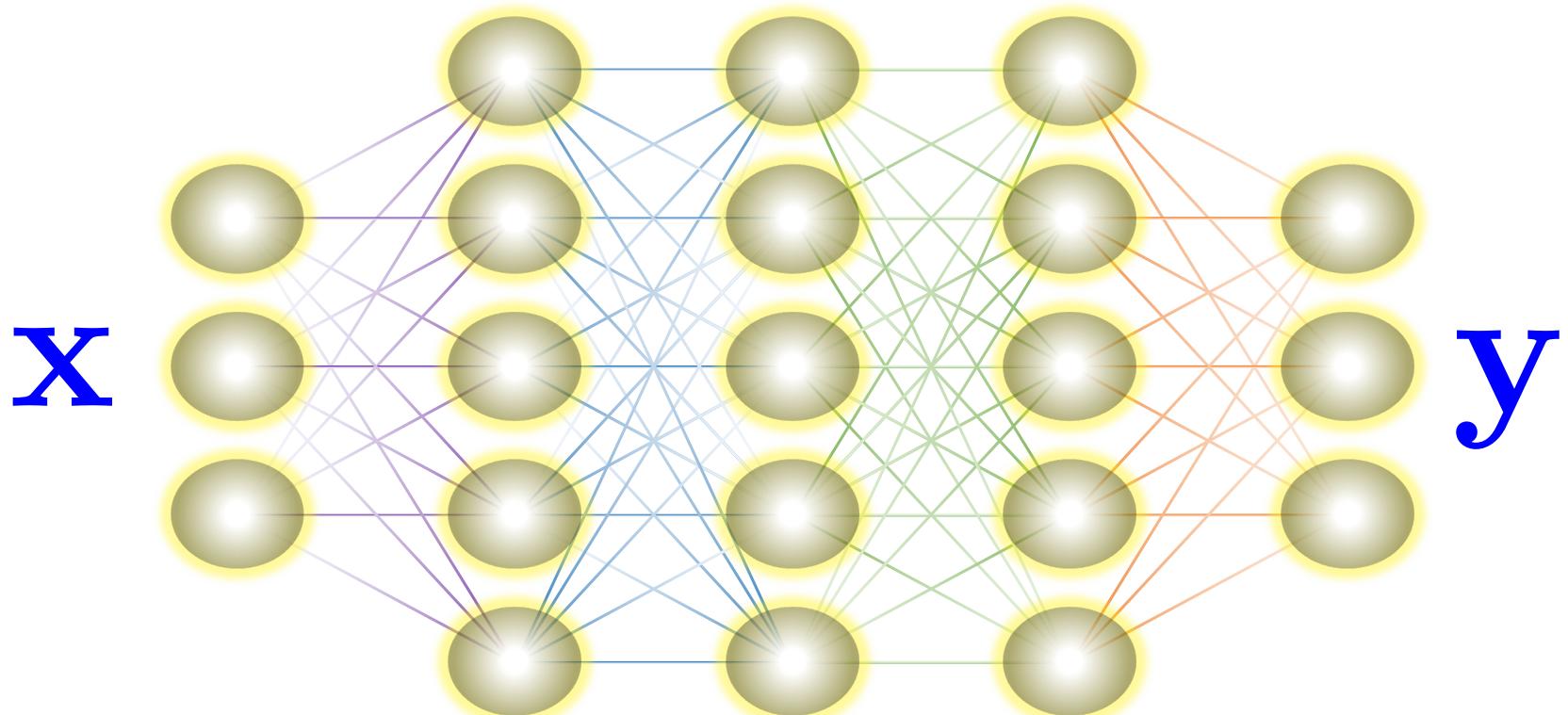
Let's have a play

Simple Vit 71.96%				Simple Vit (No value) 70.38%				Simple Vit (No PE) 57.97%			
First Layer		Following Layers		First Layer		Following Layers		First Layer		Following Layers	
Input to K&Q	Weight of K&Q	Input to K&Q	Weight of K&Q	Input to K&Q	Weight of K&Q	Input to K&Q	Weight of K&Q	Input to K&Q	Weight of K&Q	Input to K&Q	Weight of K&Q
x	MLP	x	MLP	x	MLP	x	MLP	x	MLP	x	MLP
PE		PE		PE		PE		PE		PE	
Input to V	Weight of V	Input to V	Weight of V	Input to V	Weight of V	Input to V	Weight of V	Input to V	Weight of V	Input to V	Weight of V
x	MLP	x	MLP	x	I	x	I	x	MLP	x	MLP
PE		PE		PE		PE		PE		PE	

https://colab.research.google.com/github/sluey-cs-cmu-edu/RVSS26/blob/main/Simple_ViT_in_Colab.ipynb

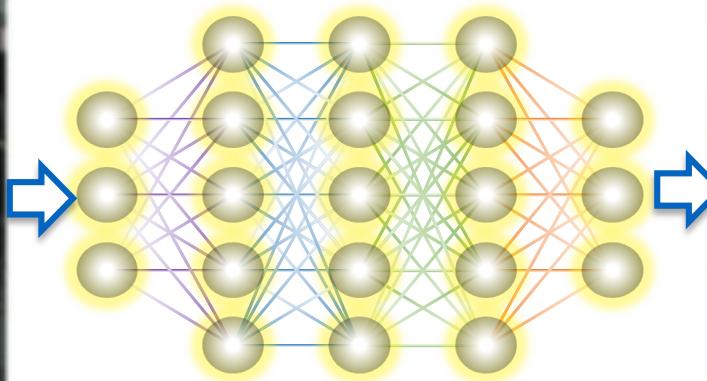
Today

- Tokenization and ConvNets
- Skip Connections and BatchNorm
- Attention and Transformers
- **Self Supervision**

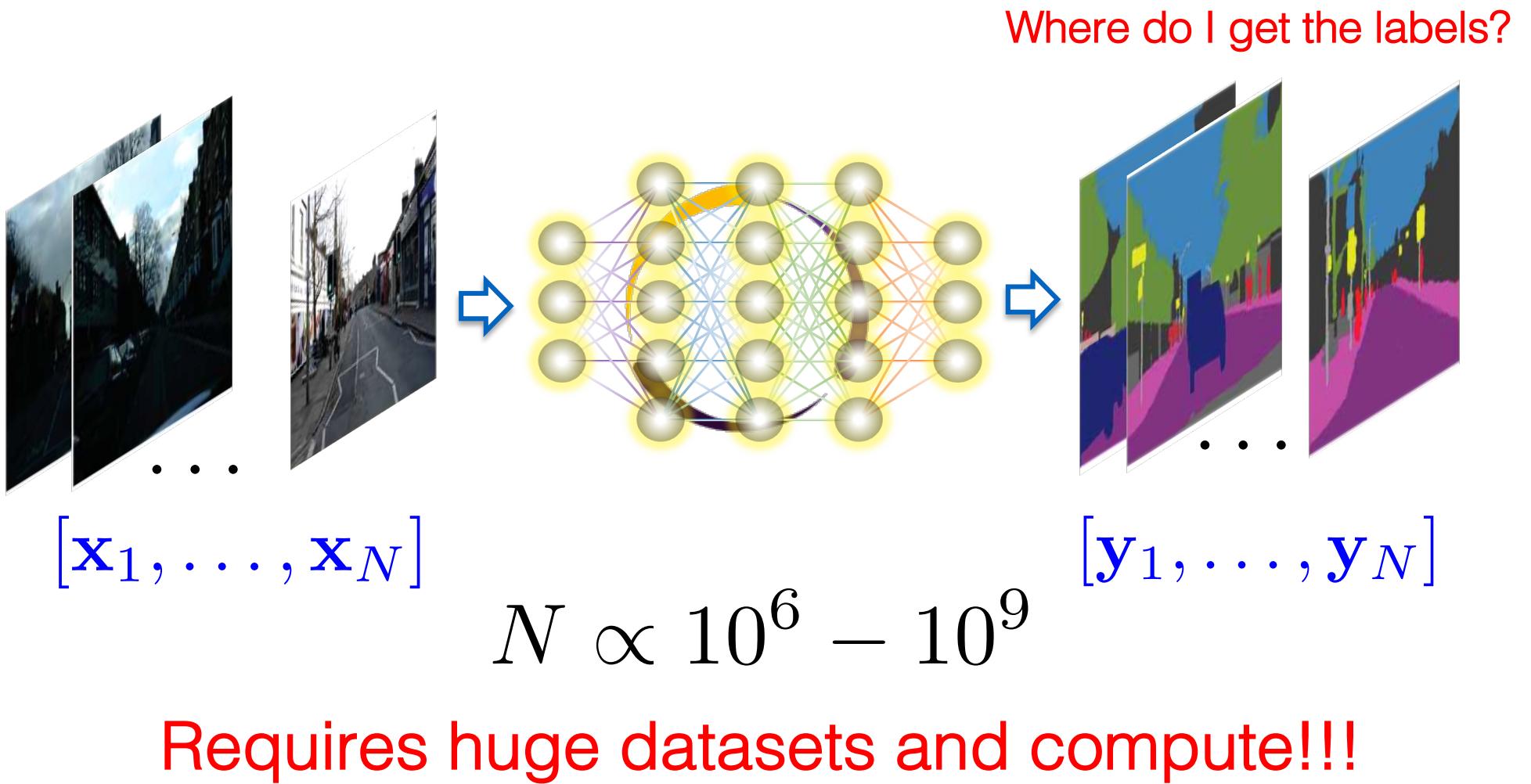


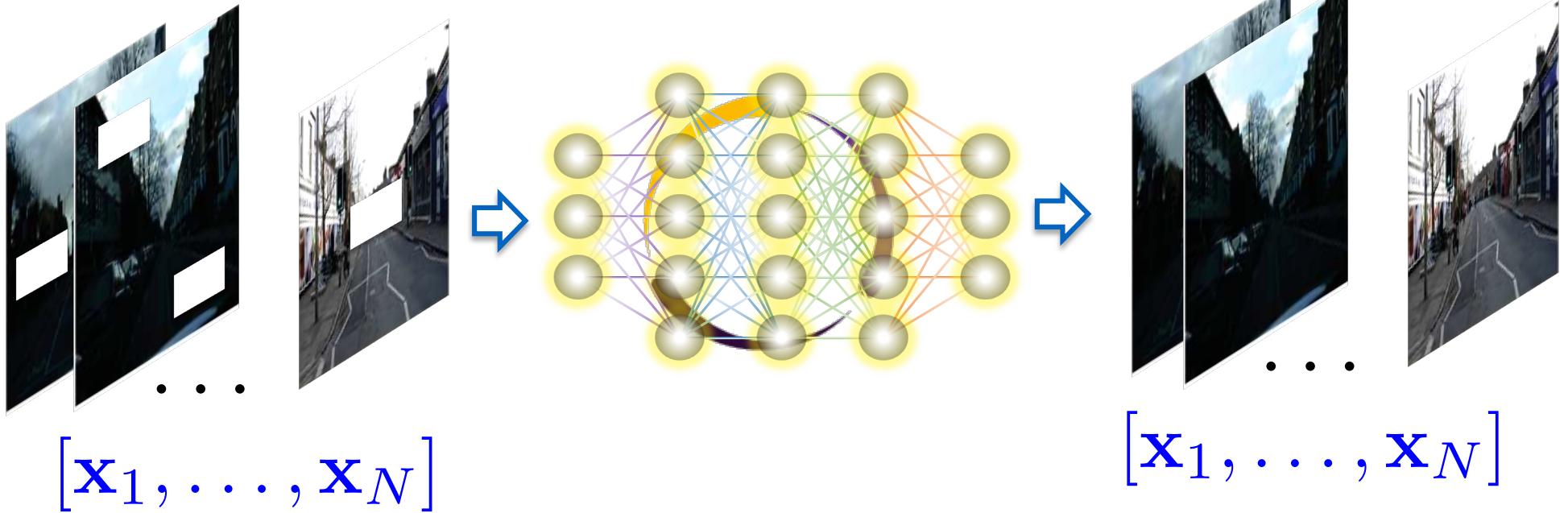


x



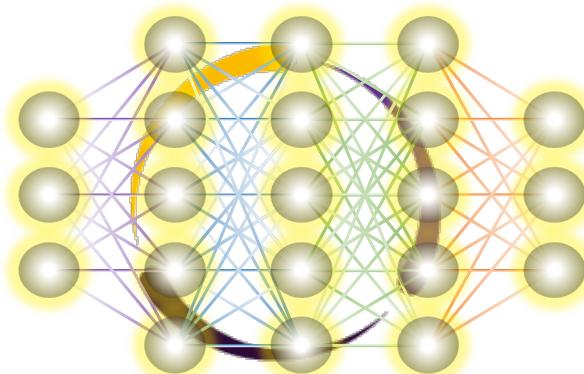
y





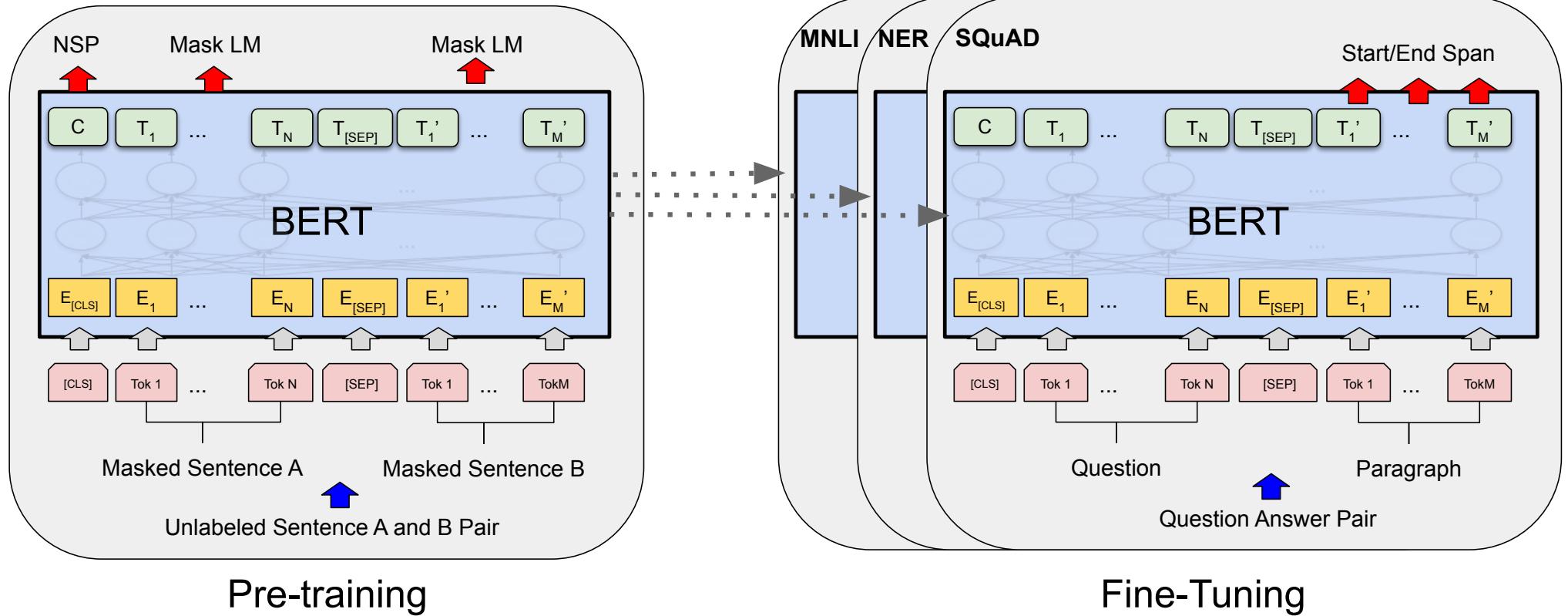
Self Supervision

"The ███ walked across the ████"



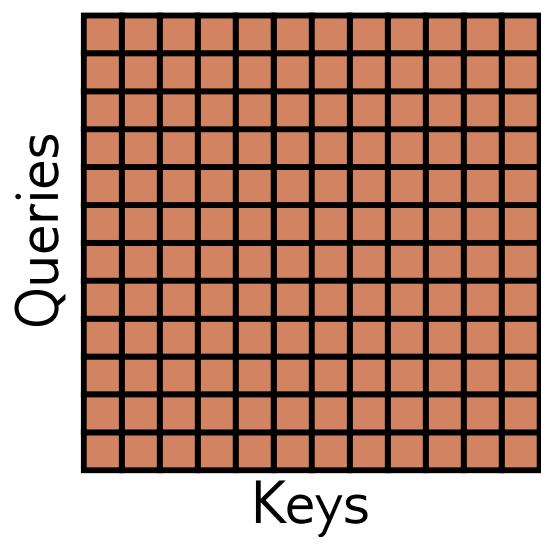
"The cat walked across the room"

Self Supervision

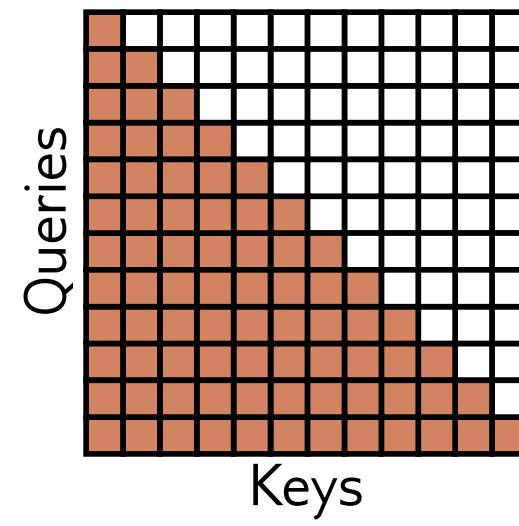


J. Devlin, et al. “BERT: Pre-training of deep bidirectional transformers for language understanding”. In NAACL, 2019..

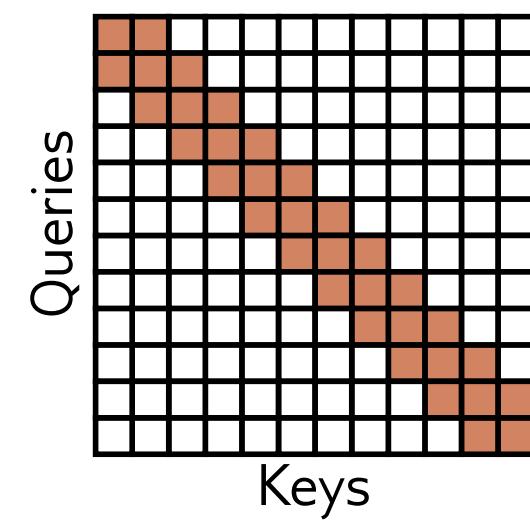
Interaction Matrices for Self Attention



Encoder

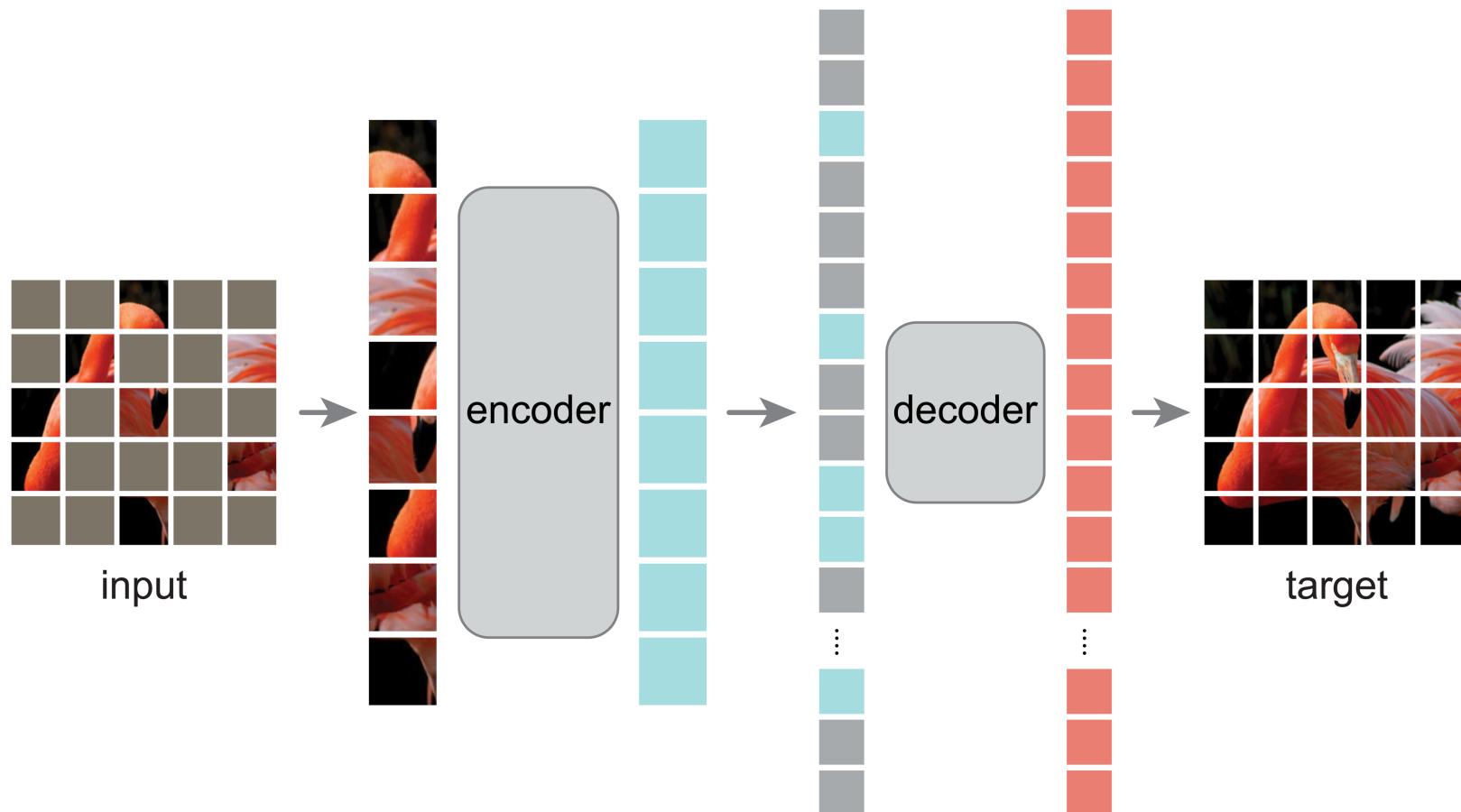


Decoder



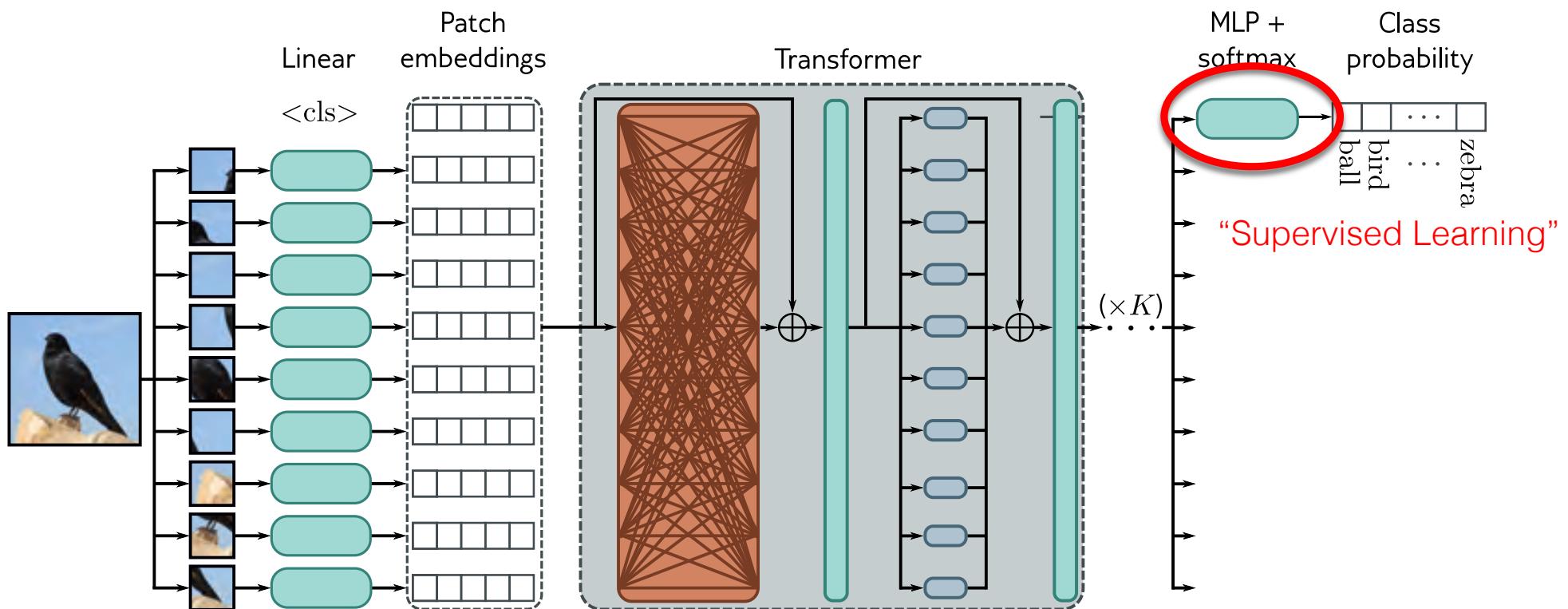
Convolutional
Encoder

Masked ViT

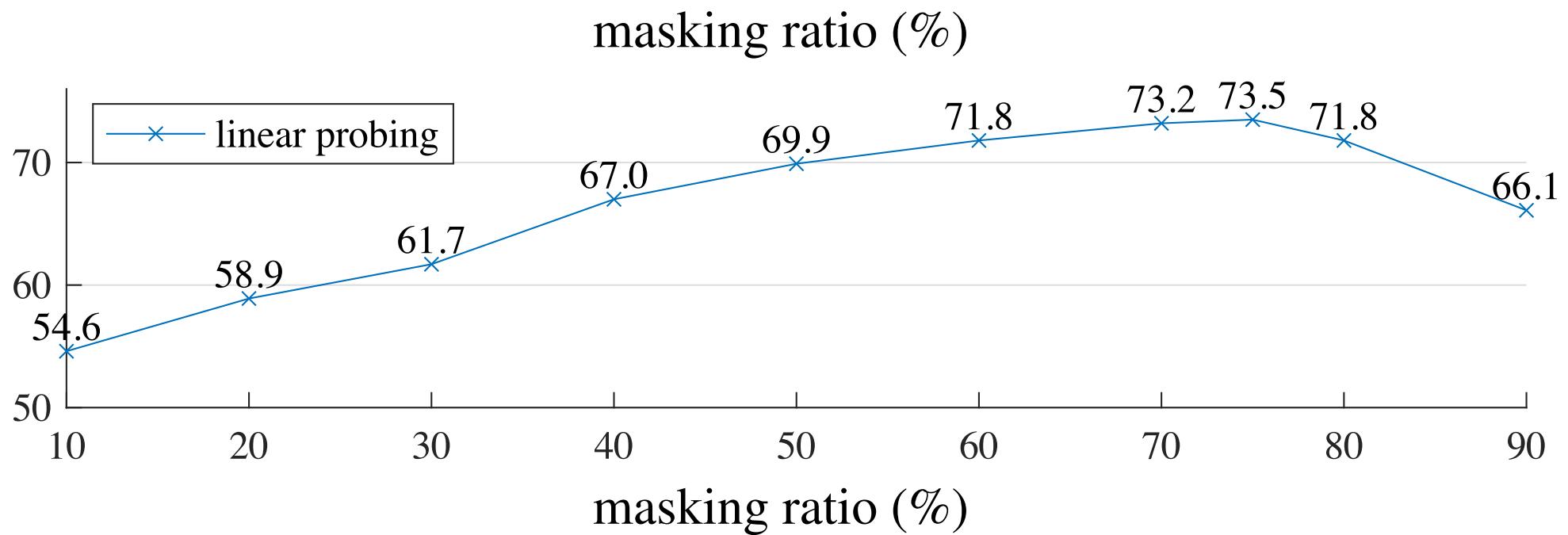


K. He, et al. "Masked Autoencoders Are Scalable Vision Learners" In CVPR 2022.

Masked ViT



Masked ViT



K. He, et al. "Masked Autoencoders Are Scalable Vision Learners" In CVPR 2022.

More to read...

