General methods that leverage computation are ultimately the most effective.

"The Bitter Lesson"

Handcrafted (e.g. SIFT) | CNNs | Transformers

Many! | Fewer | Fewer still

Inductive biases

Rich Sutton

(Reinforcement Learning Pioneer)

Motivation

Where they come from

How they work

Why Care About Neural Network Architectures?

Background

Deep learning descends from **connectionism**:

Wiring of computational networks plays key role in building intelligent machines

Structures that define the wiring:

- Architecture - connections fixed in training (e.g. operation types) ← Focus of architecture design
- Parameters - connections updated in training (e.g. kernels learned via SGD/backprop)

Goals

We inhabit a resource-limited environment. We have *limited supplies of*:

Energy | Computation | Memory | Time

Typically, we want architectures with:

Greatest task performance (e.g. accuracy) | Acceptable resource burden ← Changes over time!

References

J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis", *Cognition* (1988)

D.E. Rumelhart, G. E. Hinton and J. L. McClelland, "A general framework for parallel distributed processing", *PDP: Explorations in the microstructure of cognition* (1986)

Why did it take 3 years?

Why Google?

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

Łukasz 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 in an encoder-decoder configuration. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [31, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [34, 22, 14].

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 results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

1 Introduction

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters. With the models and datasets growing, there is still no sign of saturating performance.

Natural Language Processing

Machine Translation

Computer Vision

Vision Transformers (ViTs)

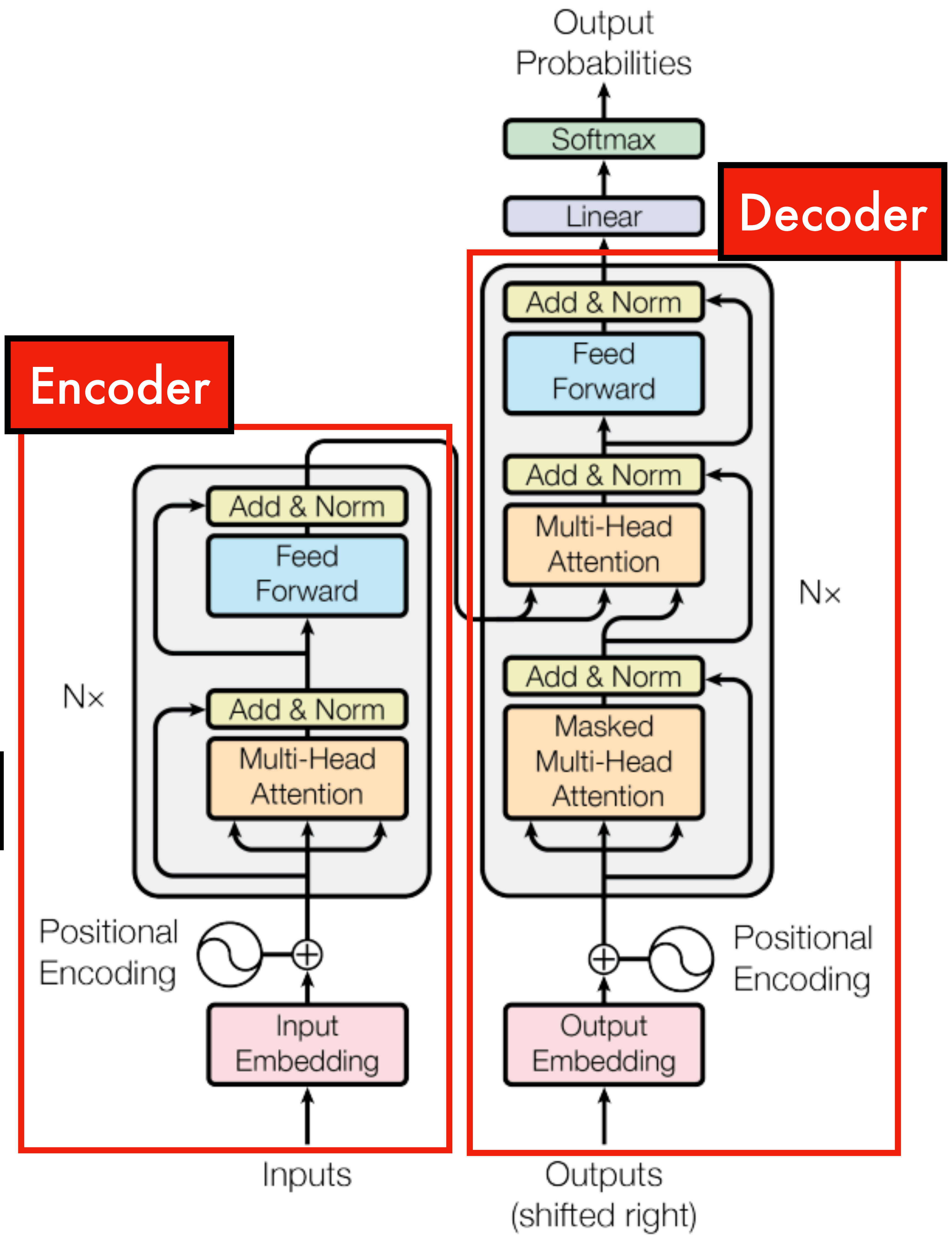
What is a Transformer?

Encoder: learns useful representation of input

Decoder: "decodes" encoded representation and combines with other input to predict output

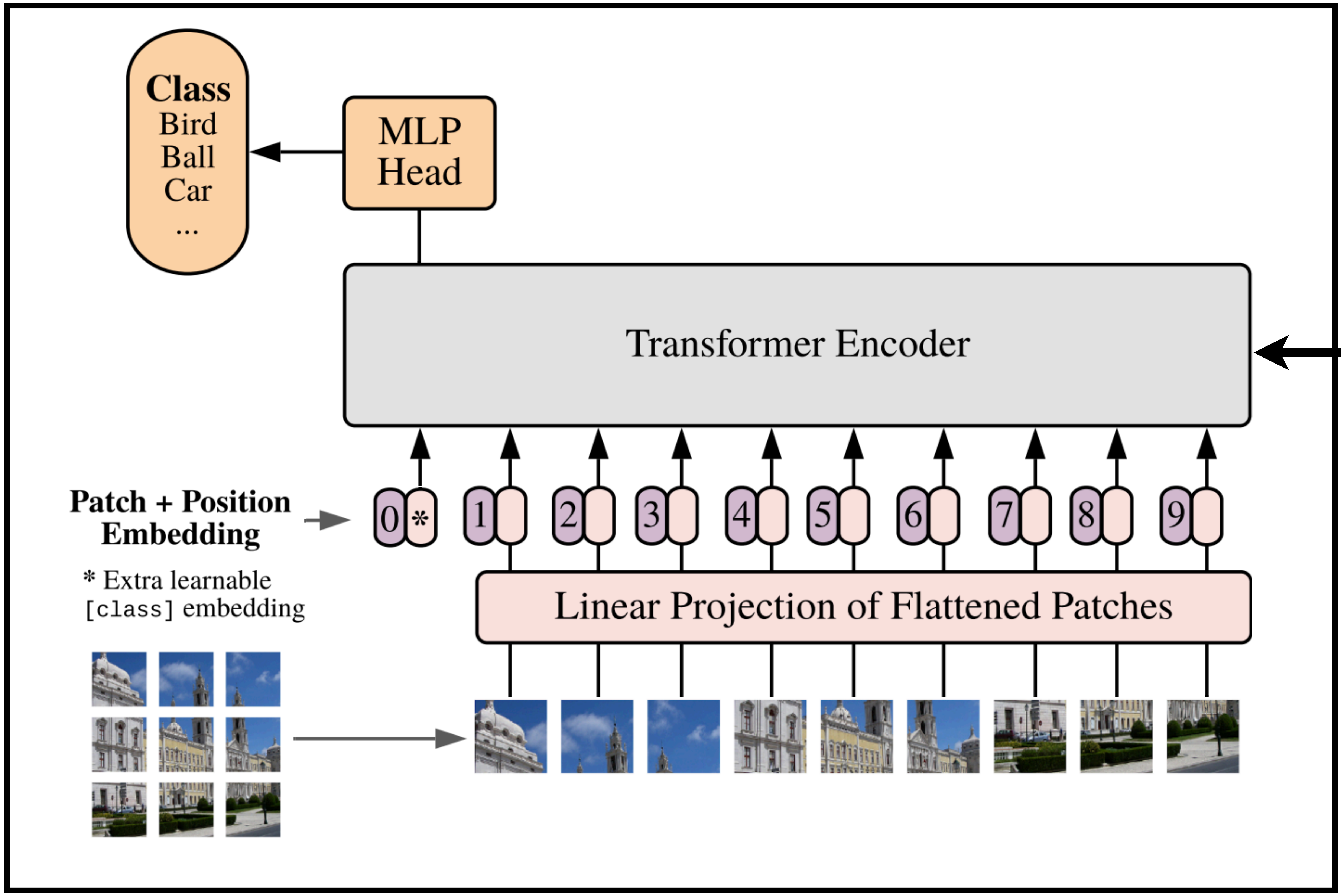
Three popular variants:

- Encoder-Only - useful for learning representations BERT
- Decoder-Only - useful for generation tasks GPT-3
- Encoder-Decoder - useful for sequence-to-sequence



References
A. Vaswani, et al. "Attention is all you need." Advances in neural information processing systems (2017)
"Transformer Models", https://huggingface.co/learn/nlp-course/chapter1

ViT: Vision Transformer (Encoder-Only)



What's going on inside here?

References/Image credits
Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR (2021)

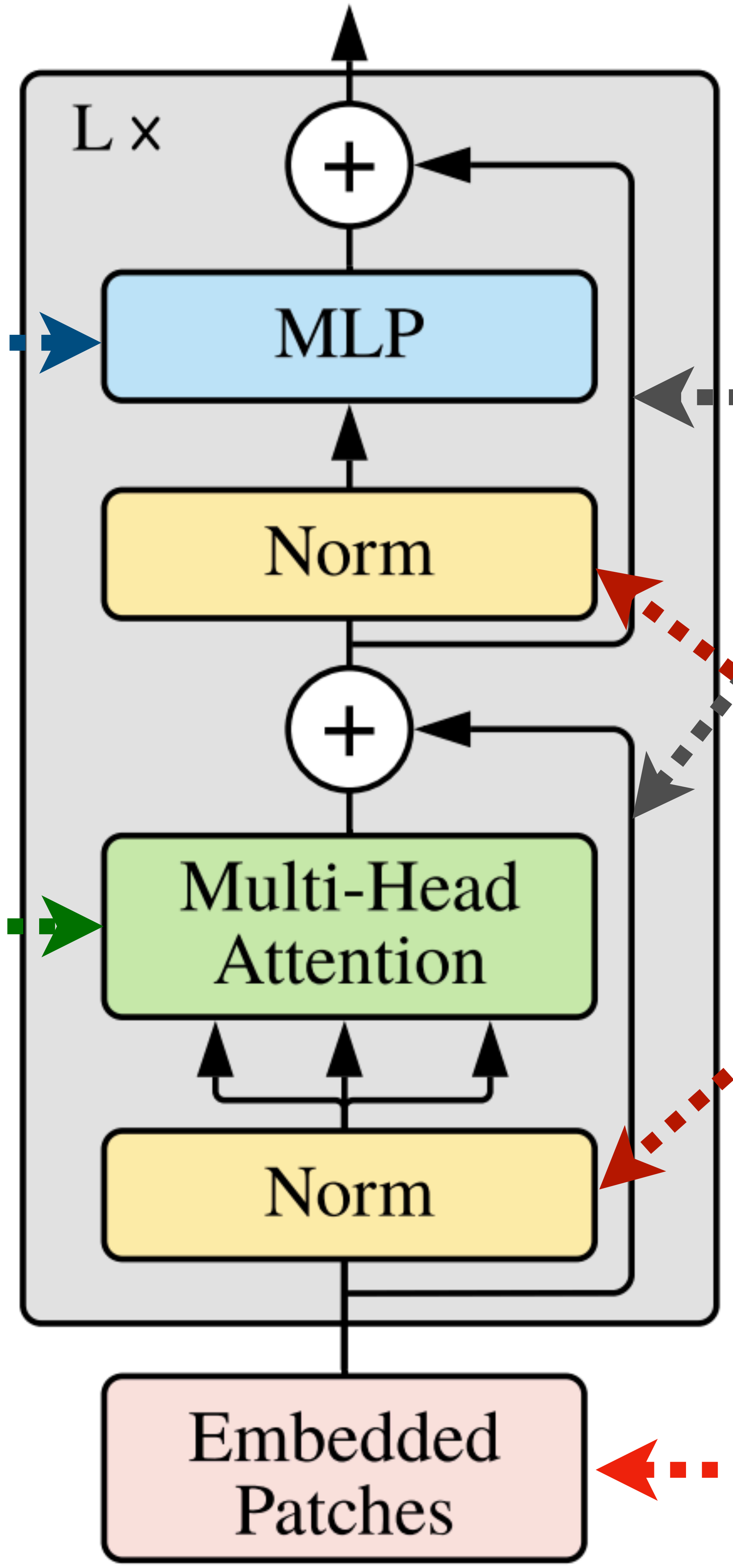
Transformer Encoder

Five key ideas

Compute-heavy blocks

Multi-Layer Perceptron
Allows embeddings to "think" independently

Multi-Head Attention
Allows embeddings to communicate



Optimisation blocks

Residual Connections
Help gradients flow

Layer Normalisation
Stabilise learning

Positional Embeddings
Allows transformer to identify embeddings

References/Image credits
Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR (2021)
Credit for "communicate/think" metaphor - Andrej Karpathy

Single-Head Attention

Input to the attention block

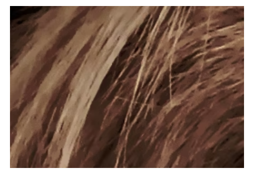
N embeddings with dimension D

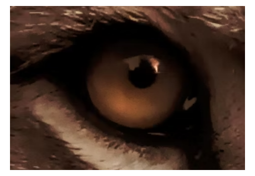
N is the num. patches $+1$

$N = 3,\ D = 2$

Stack embeddings into matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$

Problem: How can we allow the N embeddings to communicate with each other?





We project each embedding: Queries Keys Values

Queries: "Here's what I'm looking for" $\mathbf{W}^Q \in \mathbb{R}^{D \times d_k}$

Keys: "Here's what I have" $\mathbf{W}^K \in \mathbb{R}^{D \times d_k}$

Values: "What gets communicated" $\mathbf{W}^V \in \mathbb{R}^{D \times d_v}$

d_k is dimension of queries & keys, d_v is dimension of values

$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q \in \mathbb{R}^{N \times d_k}$ $\mathbf{K} = \mathbf{X}\mathbf{W}^K \in \mathbb{R}^{N \times d_k}$ $\mathbf{V} = \mathbf{X}\mathbf{W}^V \in \mathbb{R}^{N \times d_v}$

Scaled dot-product attention

$$\mathbf{Y} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \in \mathbb{R}^{N \times d_v}$$

$N \times N$ matrix

Normalise rows to prob. vectors

Weighted sum of values

Avoids "peaky" affinities

If q and k are independent random variables with mean 0 and variance 1, then $q \cdot k = \sum_{i=1}^{d_k} q_i k_i$ has variance d_k

References

Credit for attention metaphor - Andrej Karpathy

https://unsplash.com/photos/lion-in-black-background-in-grayscale-photography-8a7ZTFKax_I

Multi-Head Attention

What if the patches want to send multiple messages?

Solution: perform multiple attention operations in parallel

We use H attention "heads":

for $h = 1, \ldots, H$: Executed in parallel

$$\mathbf{Q}_h = \mathbf{X}\mathbf{W}_h^Q$$
$$\mathbf{K}_h = \mathbf{X}\mathbf{W}_h^K$$
$$\mathbf{V}_h = \mathbf{X}\mathbf{W}_h^V$$

Can be achieved efficiently with batched matrix multiplication

$$\text{head}_h = \text{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^T}{\sqrt{d_k}}\right)\mathbf{V}_h$$

$$\text{MultiHead}(\mathbf{X}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O$$

Project the results

Typically, for multi-head attention (MHA) we make the head dimensions smaller:

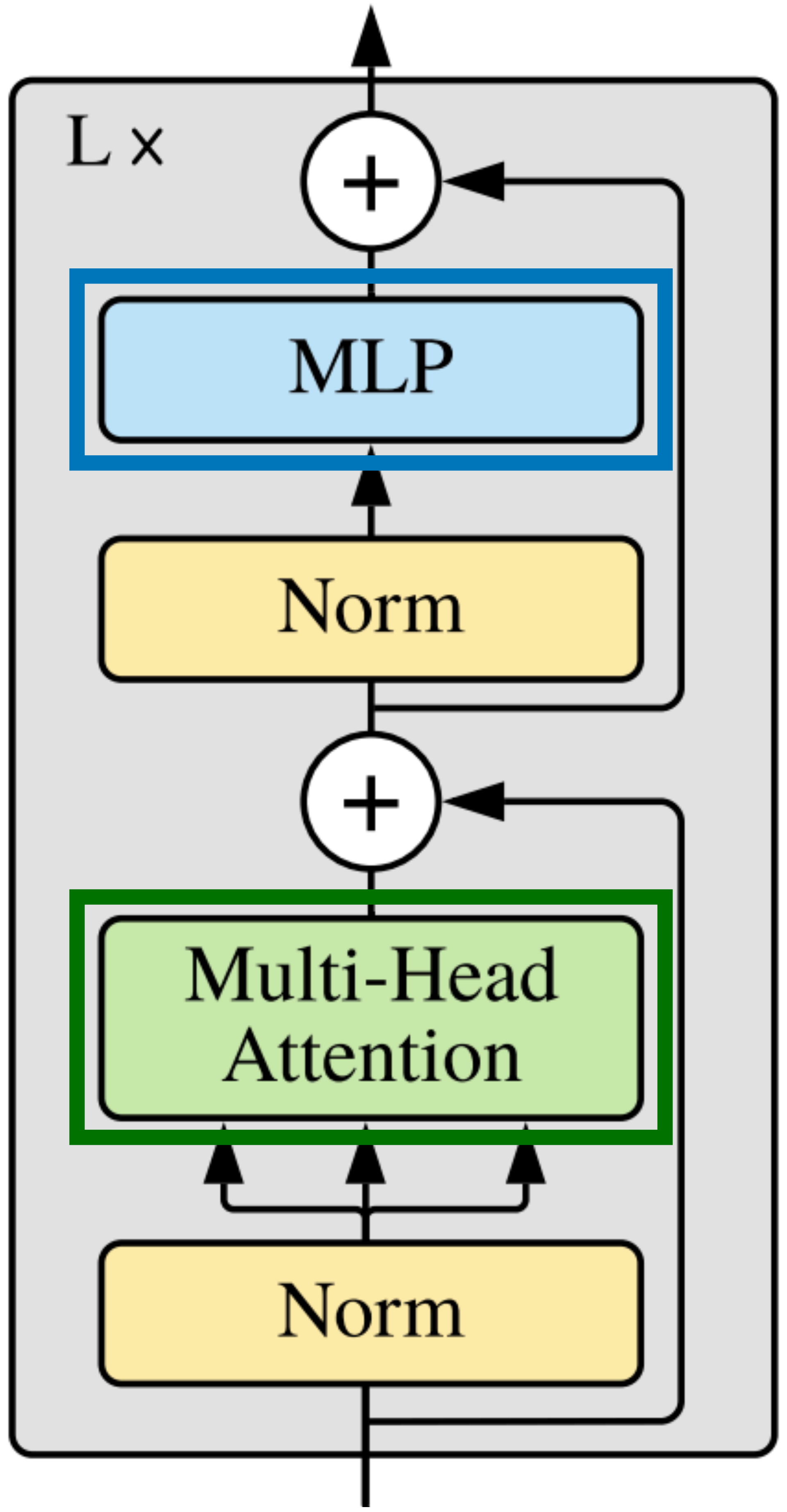
$$d_k = d_v = D/H$$

Total computational cost is similar to single-head attention

Complexity of MHA (ignoring projections): $O(N^2 \cdot D)$

Quadratic in sequence length!

Multi-Layer Perceptron (MLP)

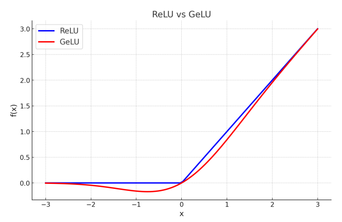


After the embeddings have communicated, we'd like them to do some "thinking alone" about what they've learned

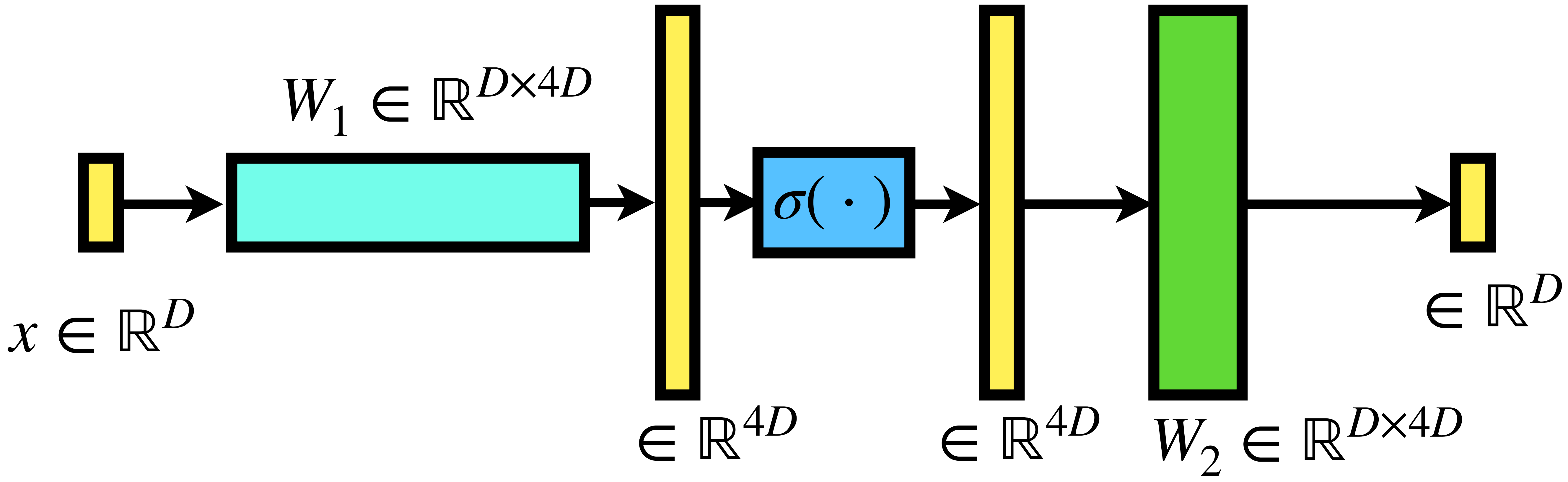
This is implemented with a 2-layer MLP that is applied independently on each embedding:

$$\mathrm{MLP}(x) = W_2\, \sigma(W_1 x + b_1) + b_2$$

where $\sigma(\cdot)$ is a non-linearity **ReLU** **GeLU**



Typically, we use an expansion factor of 4:



References

Credit for "communicate/think" metaphor - Andrej Karpathy

Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR (2021)

Residual Connections

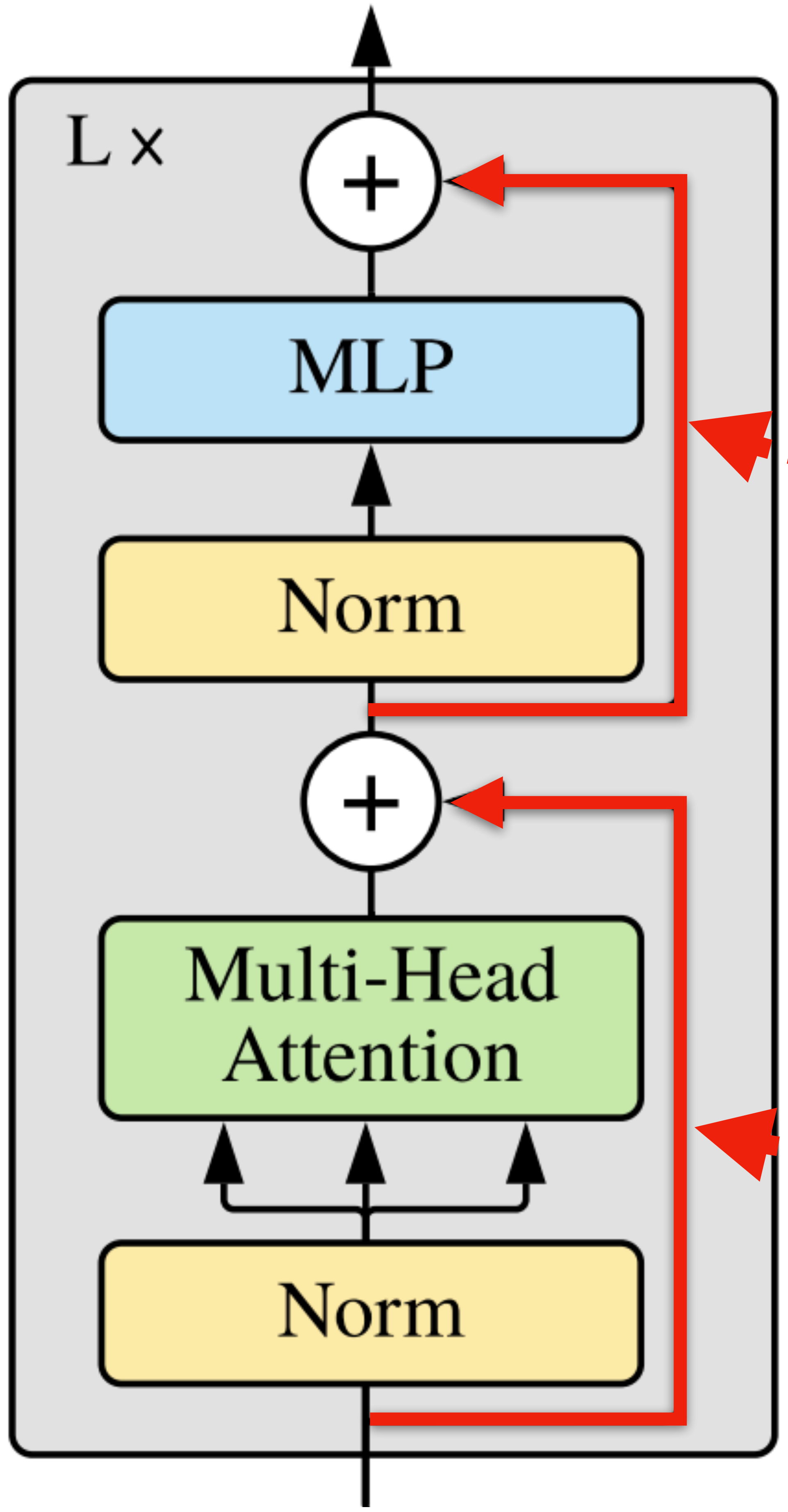
Deep residual learning for **image recognition**
K He, X Zhang, S Ren, J Sun - … and **pattern recognition**, 2016 - open
… **Deeper** neural **networks** are more difficult to train. We present a **resi**
to ease the training of **networks** that are substantially **deeper** than those
Save Cite Cited by 188754 Related articles All 76 versions

Residual connections help with optimisation

Why? 🤷‍♀️ Deep learning…

"We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping."

Learning deep networks without residual connections is difficult



$$\text{Out} = \mathbf{Y} + \text{MLP}(\text{Norm}(\mathbf{Y}))$$

$$\mathbf{Y} = \mathbf{X} + \text{MHA}(\text{Norm}(\mathbf{X}))$$

<u>Intuitions:</u>

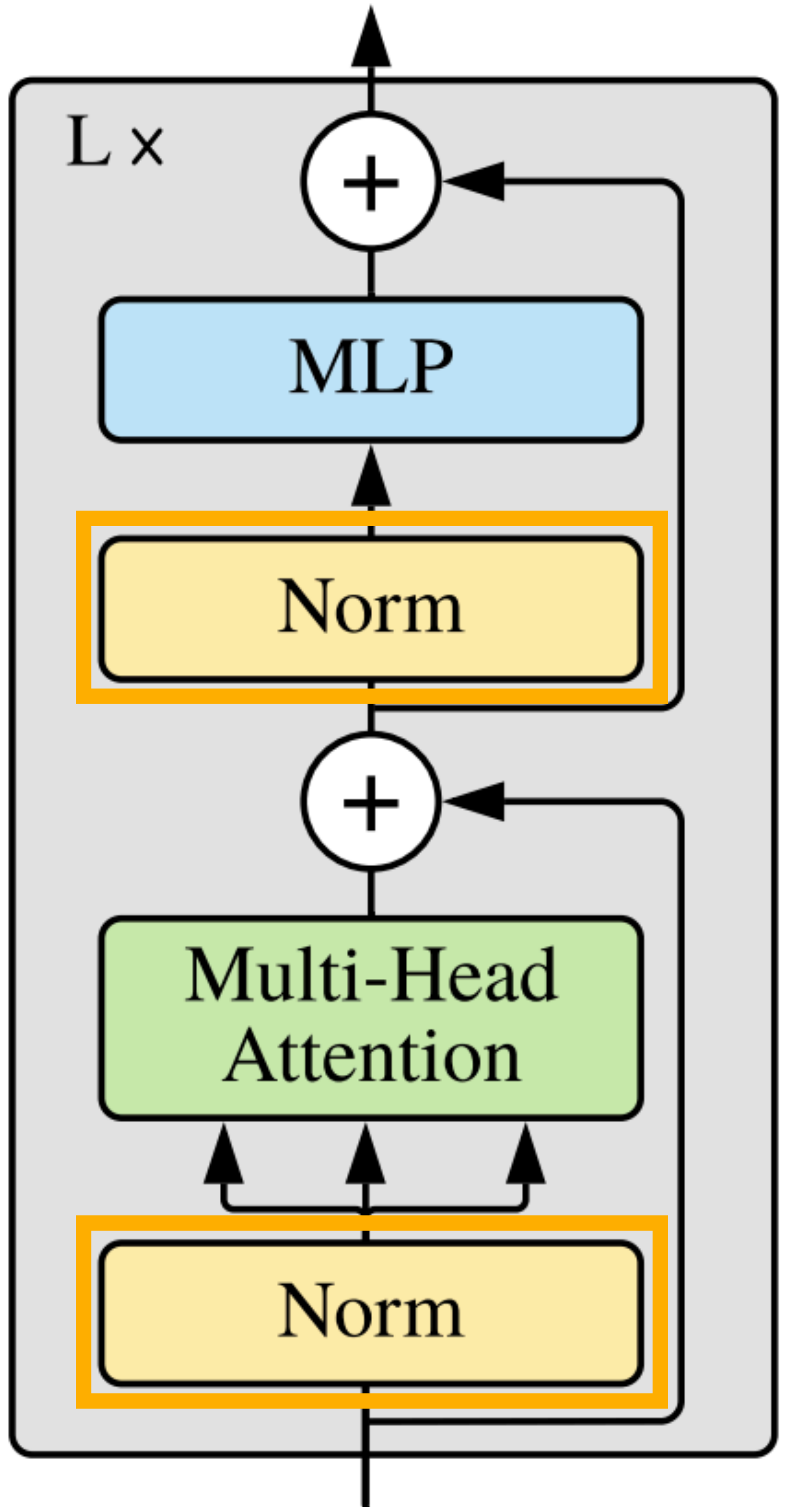
Help with gradient flow

(avoids vanishing gradients)

Help with preconditioning

References
Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR (2021)
Quote from K. He et al., "Deep residual learning for image recognition", CVPR (2016)

LayerNorm



LayerNorm is very similar to BatchNorm:

$$y = \frac{x - \mathbb{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} \cdot \gamma + \beta$$

Learned gain

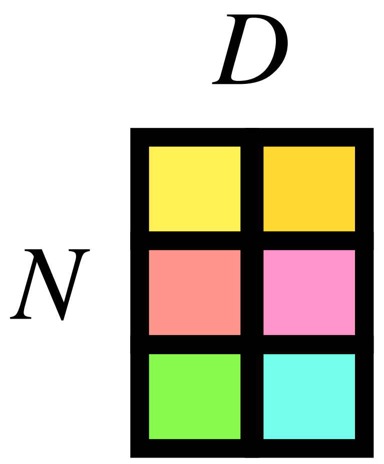
Learned bias

for numerical stability

Difference vs BatchNorm: how we estimate $\mathbb{E}[x]$ and $\mathrm{Var}[x]$

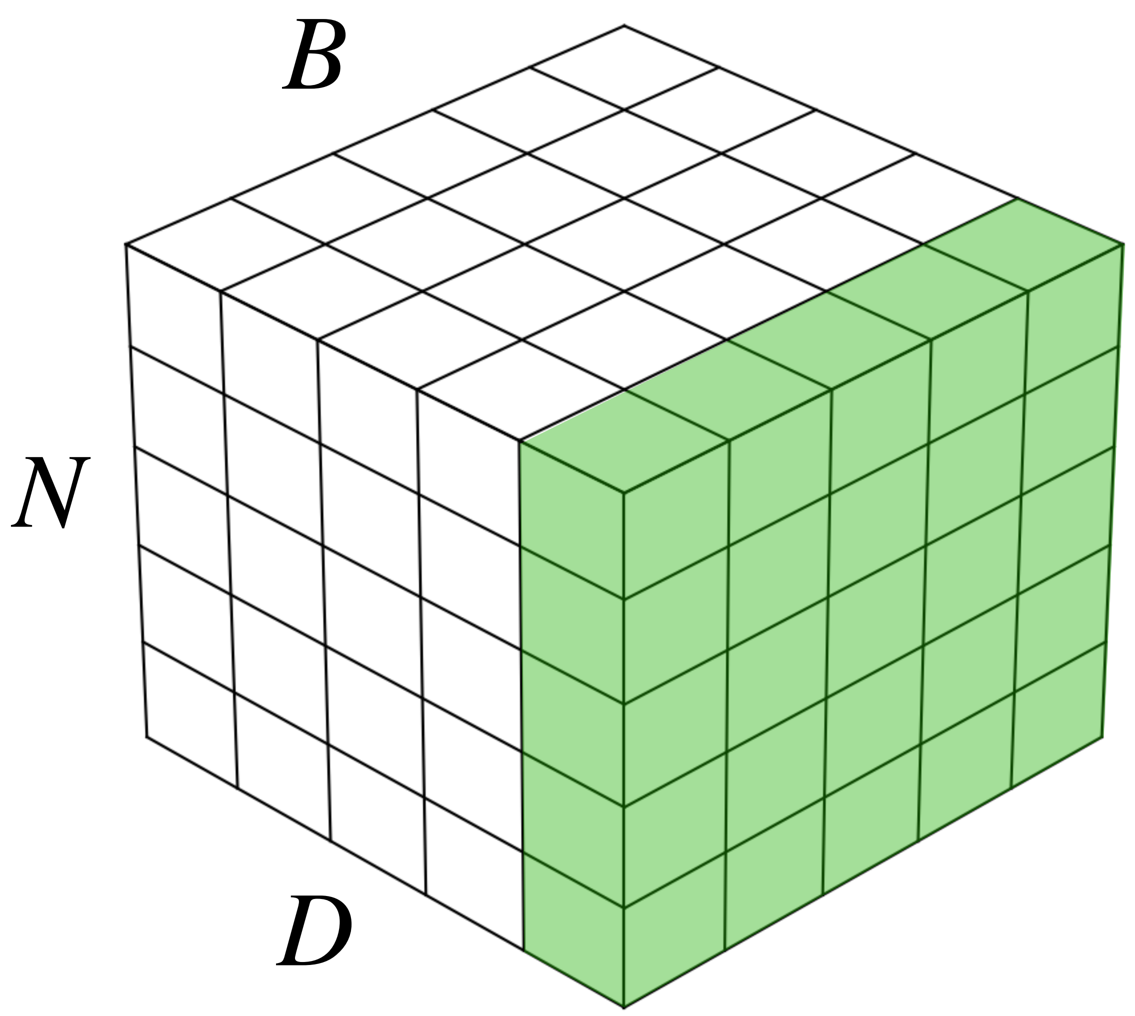
So far, we've had $N \times D$ input matrices:

N is the sequence length, D is the embedding dim.

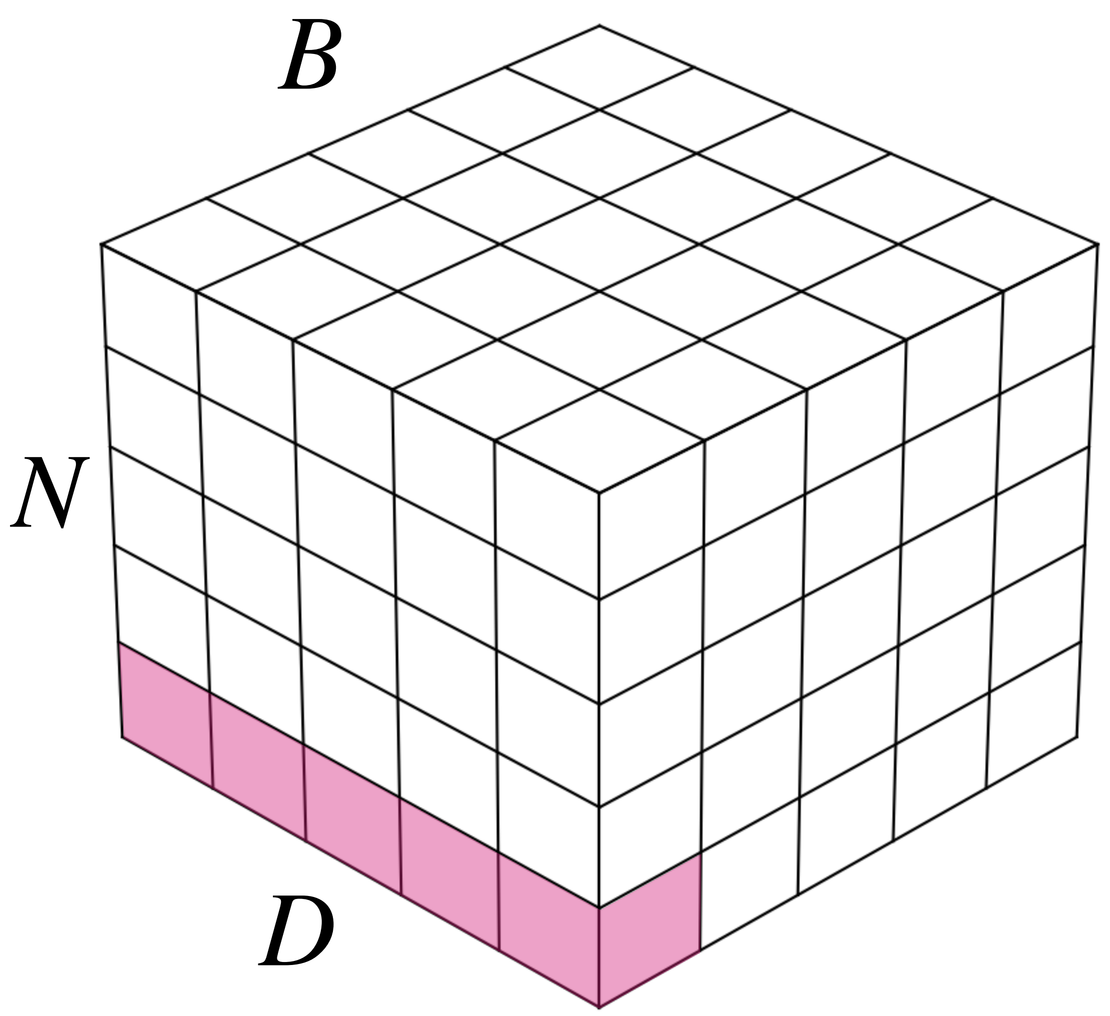


In practice, we process $B \times N \times D$ (where B is the minibatch size)

BatchNorm



LayerNorm



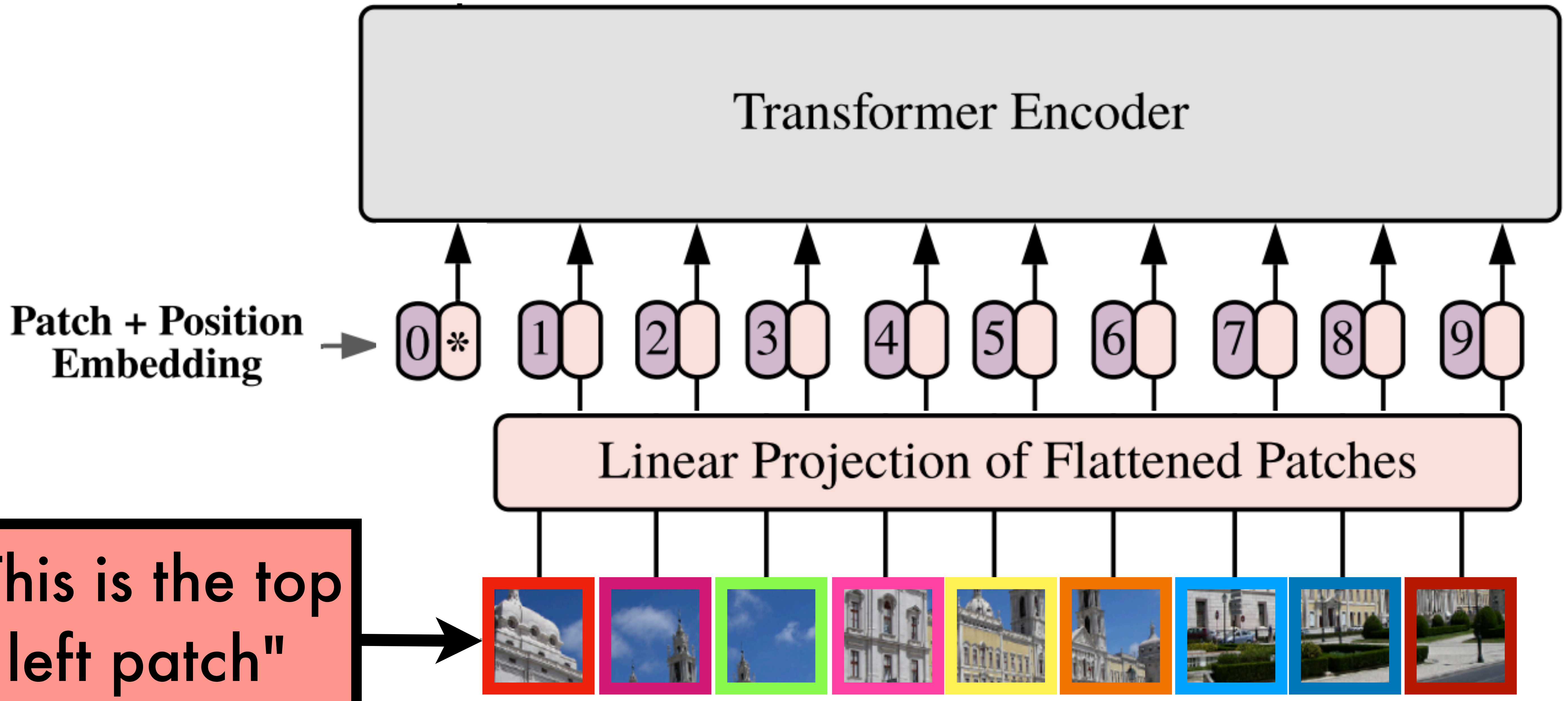
LayerNorm has
- No dependence on batch dim.
- Same procedure at train/test time

References

Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ICLR (2021)

S. Shen et al., "Powernorm: Rethinking batch normalization in transformers", ICML (2020)

Position Embeddings



"This is the top left patch"

Motivation: transformer encoder treats input as set

Once we've split the images into patches, we've thrown away their relative positions!

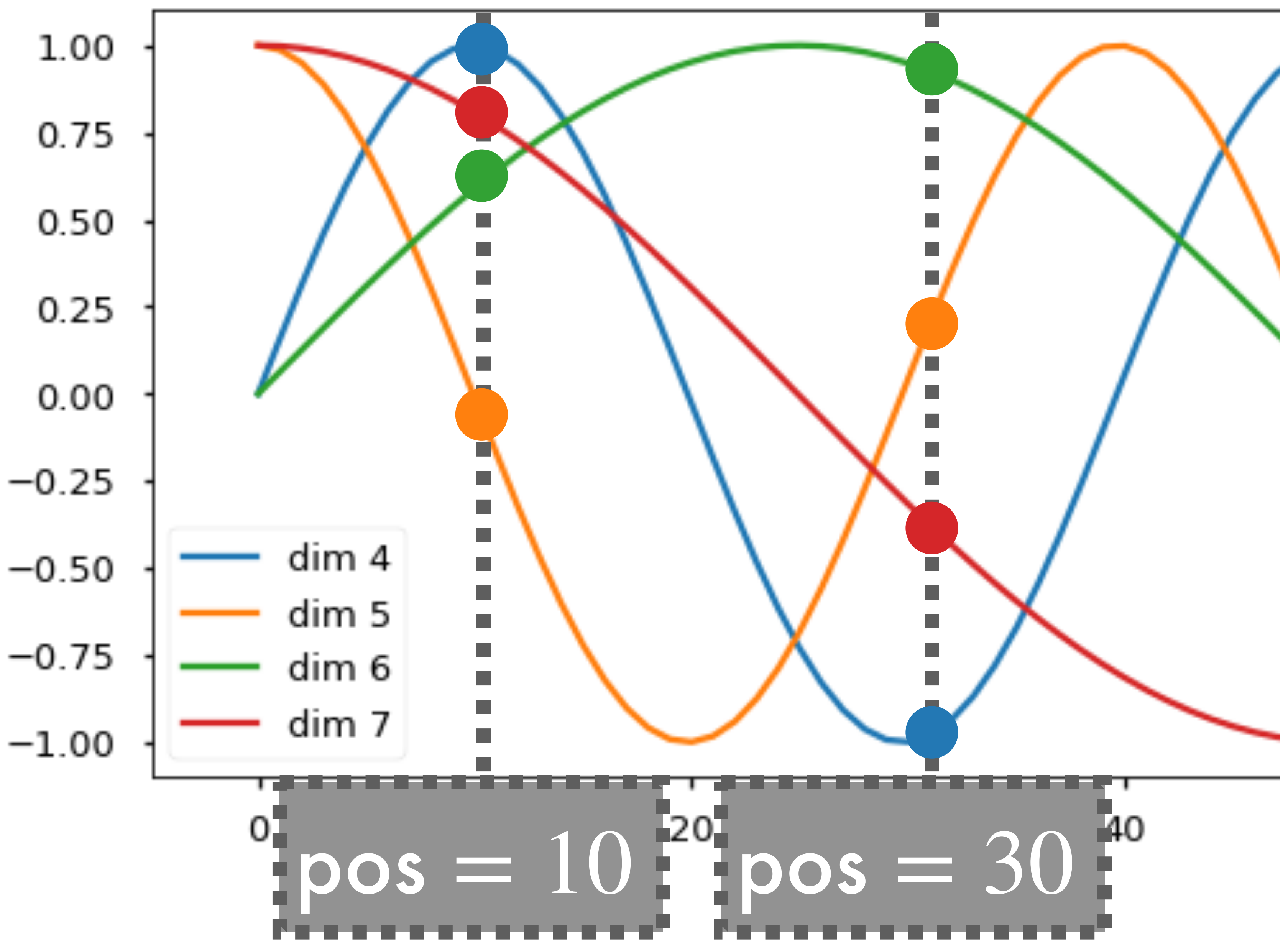
Solution: position embeddings "label" patch with position

How do we "label" positions?

Hand-crafted position embeddings:

$$\mathrm{PE}(\mathrm{pos}, 2i) = \sin\left(\frac{\mathrm{pos}}{10{,}000^{2i/D}}\right)$$

$$\mathrm{PE}(\mathrm{pos}, 2i+1) = \cos\left(\frac{\mathrm{pos}}{10{,}000^{2i/D}}\right)$$



Alternative (used in ViT): learn the embeddings from scratch

PEs are an active area of research

References
Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR (2021)
Positional Embeddings Visualisation https://nlp.seas.harvard.edu/2018/04/03/attention.html

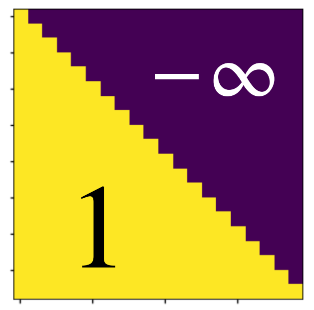
Cross/Causal Attention

So far: queries, keys and values have been produced from the same sequence

This is called "self attention"

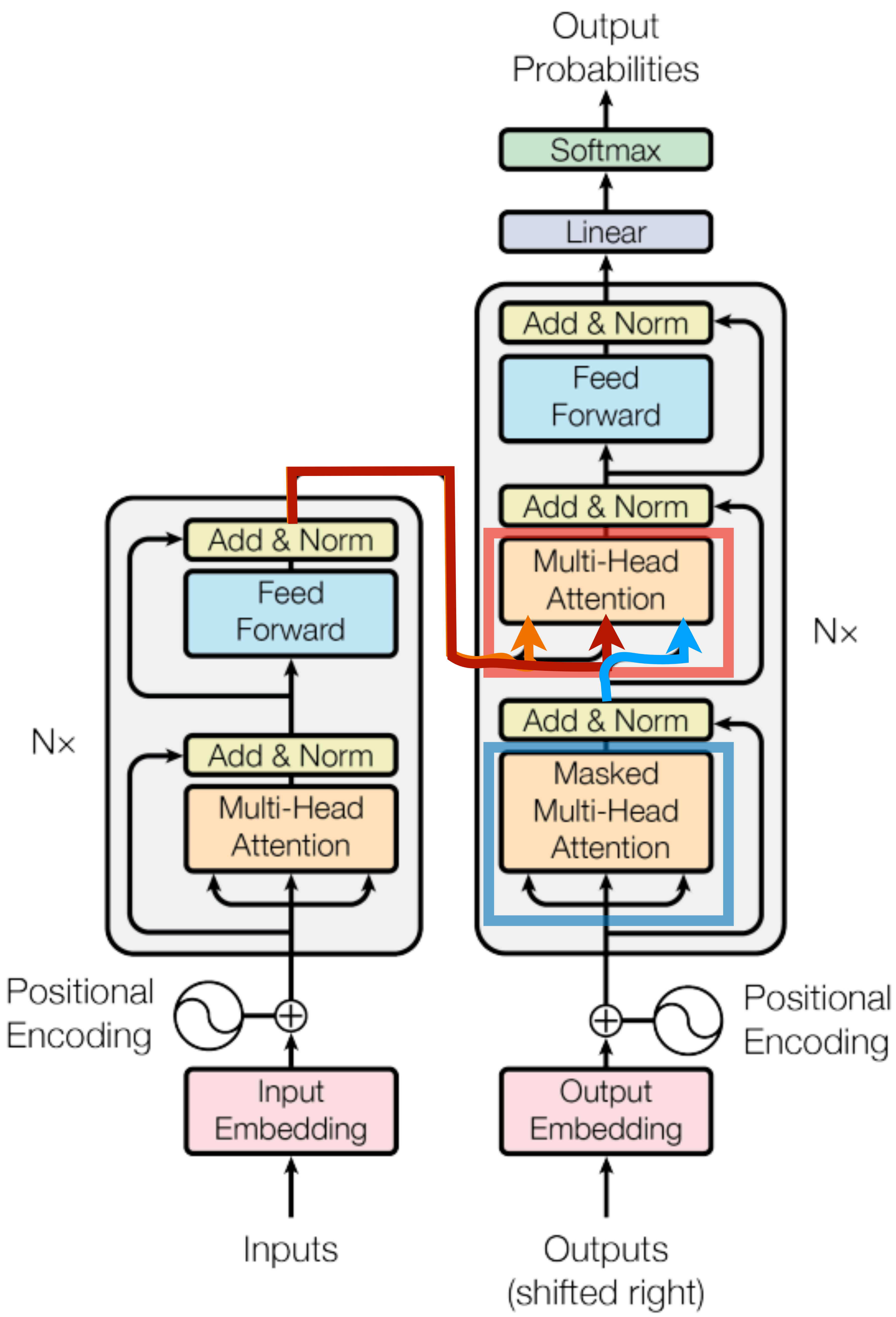
Alternative: "cross attention" - queries from one sequence, keys and values from a different sequence 🦩 **Flamingo**

When generating sequences, we don't want all embeddings to communicate

Only allow "causal" attention: N 

N

(softmax turns each $-\infty$ into 0)



References

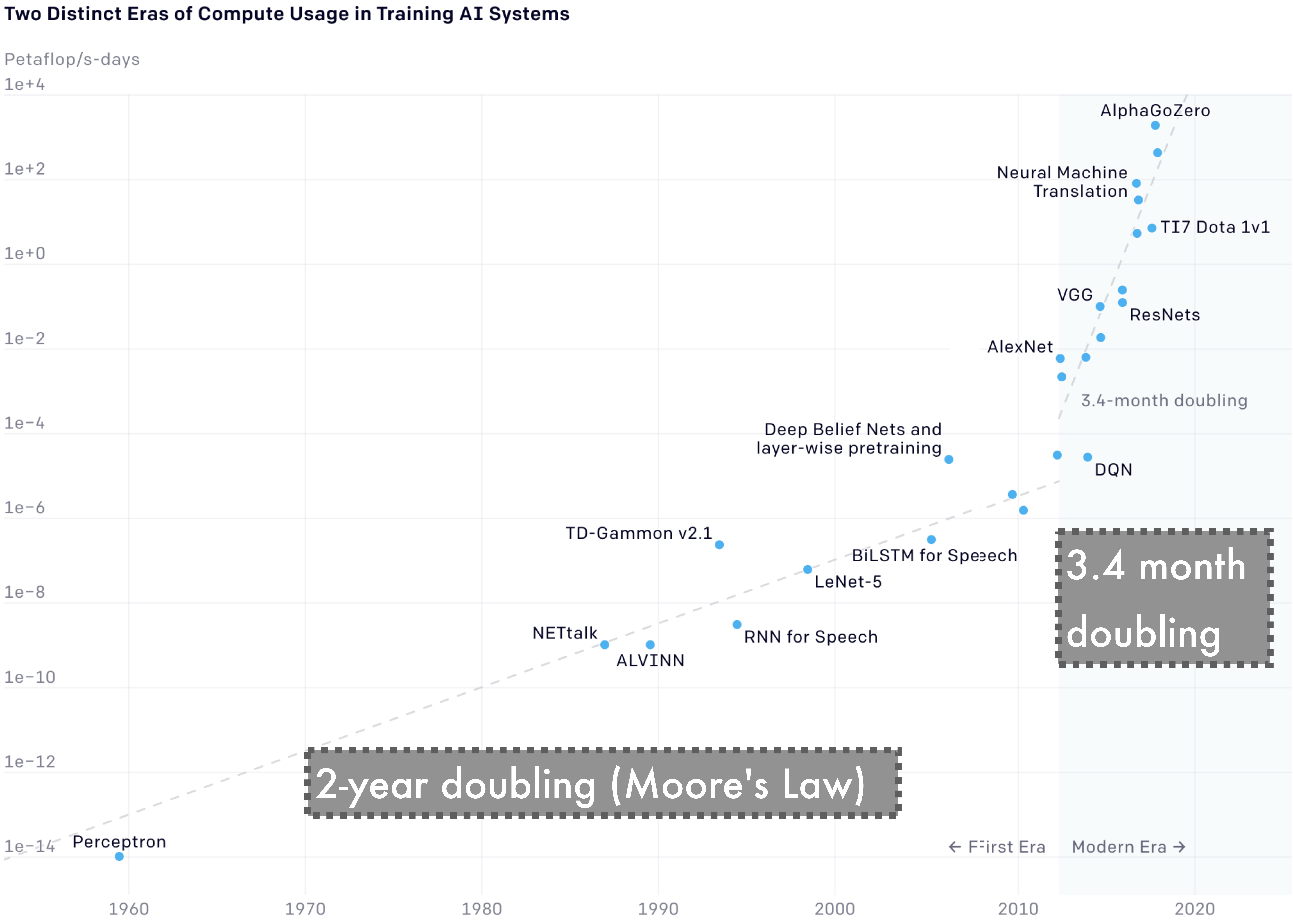
A. Vaswani, et al. "Attention is all you need." Advances in neural information processing systems (2017)

J-B Alayrac et al., "Flamingo: a visual language model for few-shot learning", NeurIPS (2022)

https://nlp.seas.harvard.edu/2018/04/03/attention.html

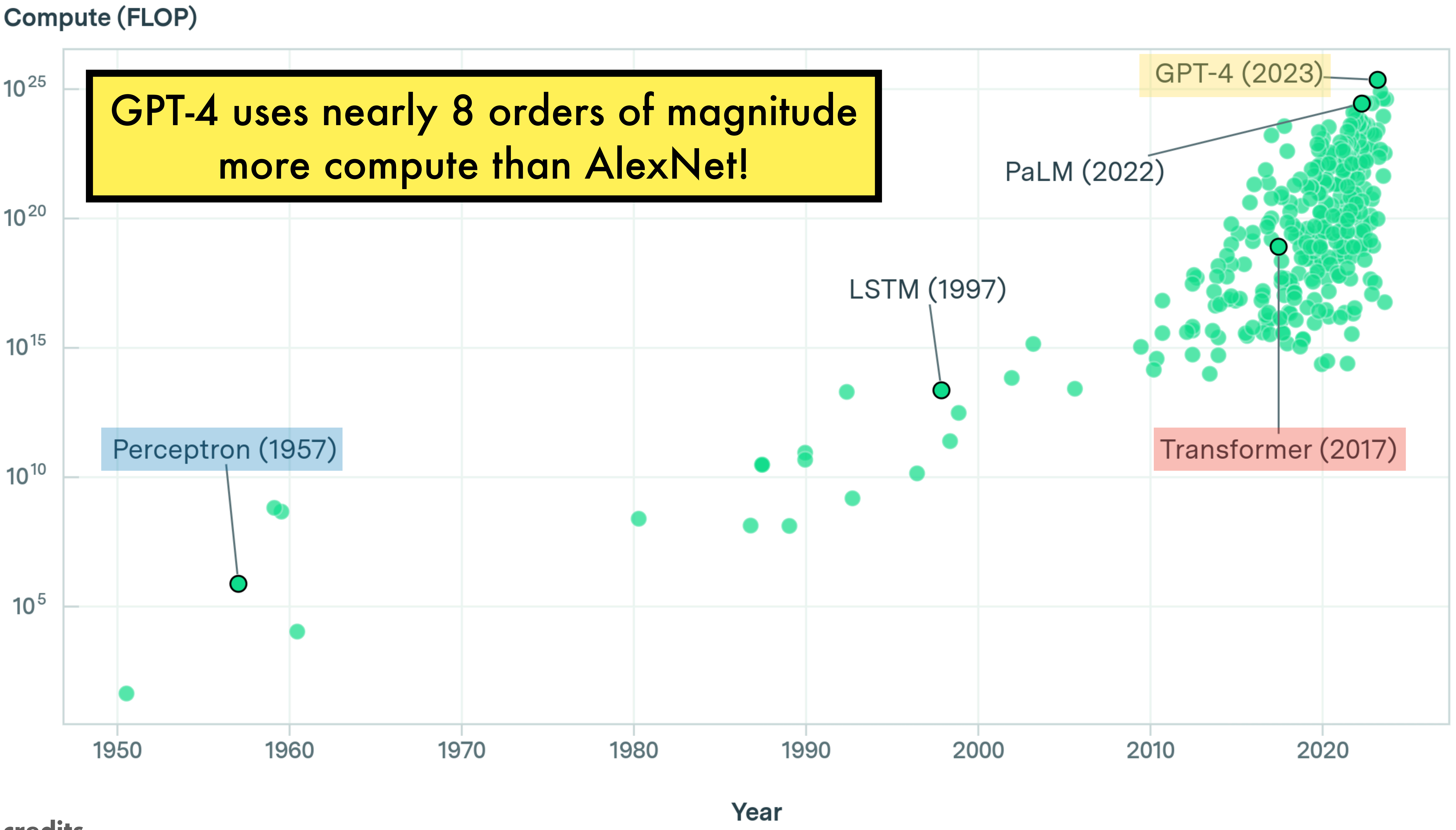
Scaling Up

1 petaflop/s-day is 8 V100 GPUs running for 1 day



References/Image credits
D. Amodei and D. Hernandez, "AI and Compute", 2018

Scaling up further



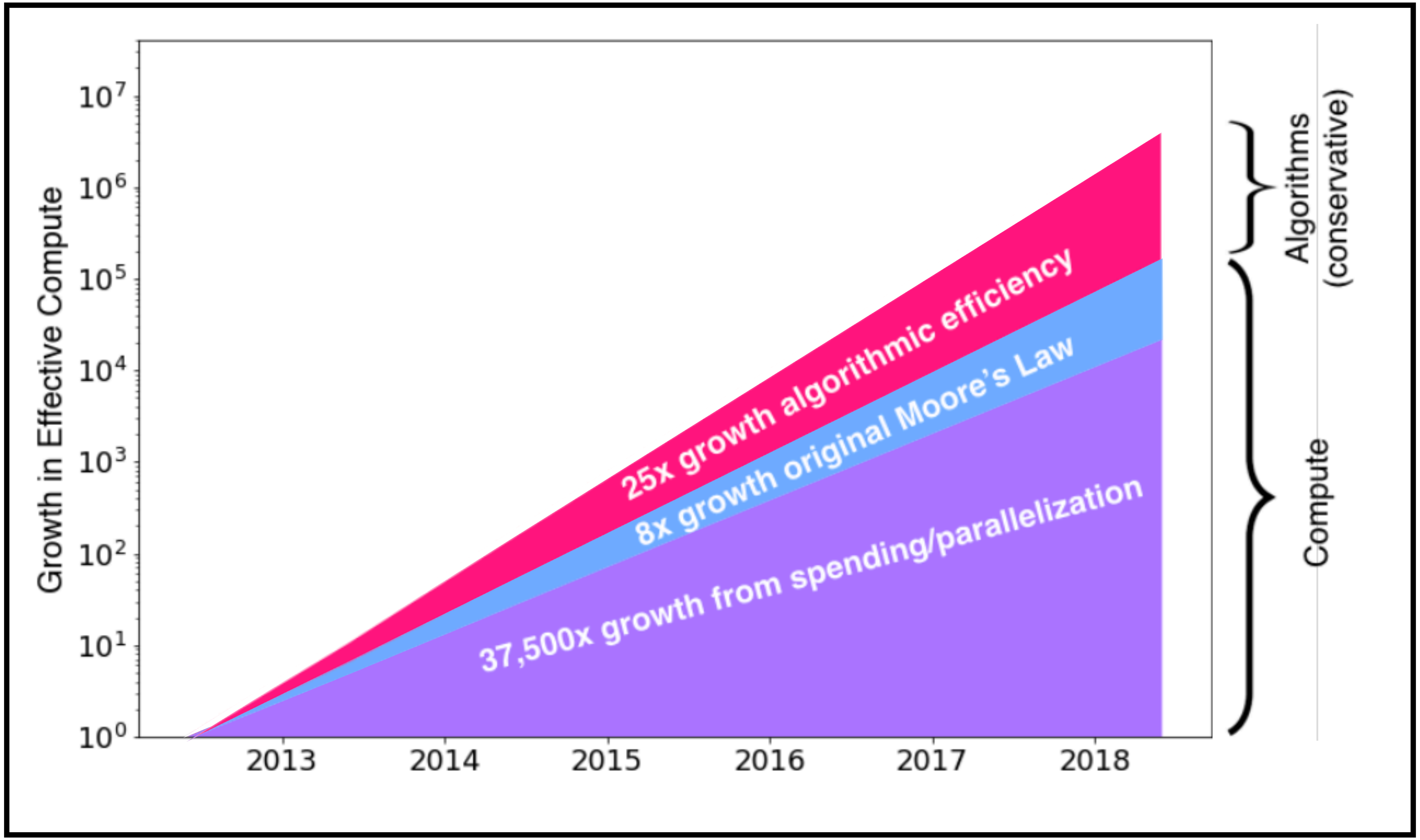
References/Image credits

https://epochai.org/blog/announcing-updated-pcd-database

What factors are enabling effective compute scaling?

Effective compute ≈ FLOPs required to reach AlexNet-level ImageNet performance

Estimated cost of training GPT-4: O(100 Million) USD



References/Image credits

D. Hernandez and T. Brown, "Measuring the Algorithmic Efficiency of Neural Networks", arXiv (2020)

https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over/

The Importance of Scale

How important is scale for Deep Neural Networks?

Is it "just engineering", or something more fundamental?

Note: It is challenging to analyse shifts from quantitative to qualitative differentiation

Hierarchy of sciences

Is cell biology "just" applied molecular biology?
Is molecular biology "just" applied chemistry?
Is chemistry "just" applied many-body physics?
....
One science obeys the laws of the other
At each stage, new laws and concepts are necessary

Qualitative vs Quantitative

FITZGERALD: The rich are different from us.
HEMINGWAY: Yes, they have more money.

"In almost all fields, a factor of ten means fundamentally new effects. If you increase magnification by a factor of 10 in Biology, you will see new things."

Hamming, Art of doing science and engineering, 1997

References/Footnotes:

P. Anderson, "More is different", Science (1972)

The "wisecrack" of Hemingway appears as a comment made by a character in one of his novels (http://www.quotecounterquote.com/2009/11/rich-are-different-famous-quote.html)

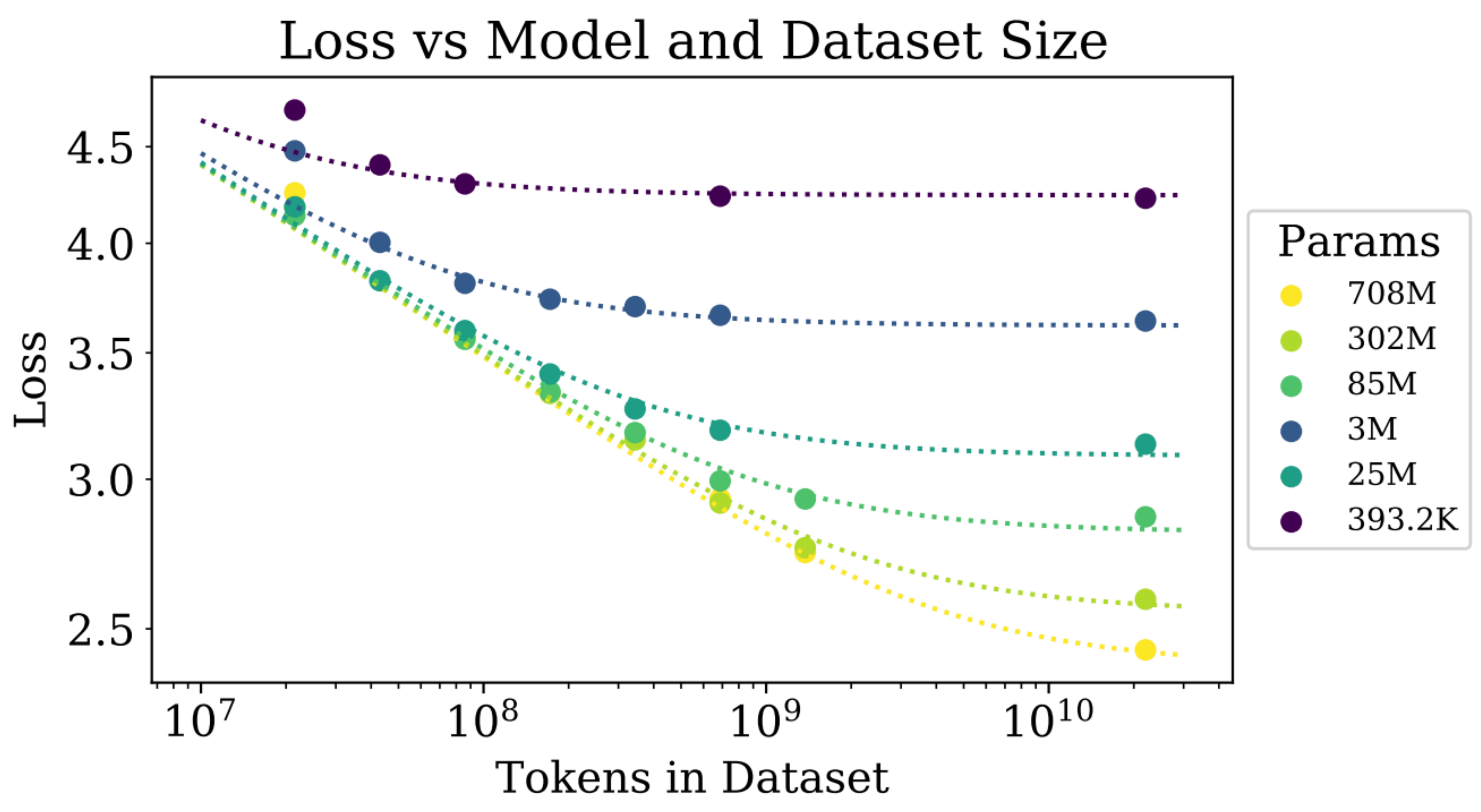
R. Hamming, "The Art of Doing Science and Engineering: Learning to Learn" (1997)

Transformer scaling laws for natural language

Predictable scaling

Transformer performance on language modelling tasks scales predictably as a *power law* with:

- Compute
- Training data size
- Model size



Some power laws were found that span more than six orders of magnitude

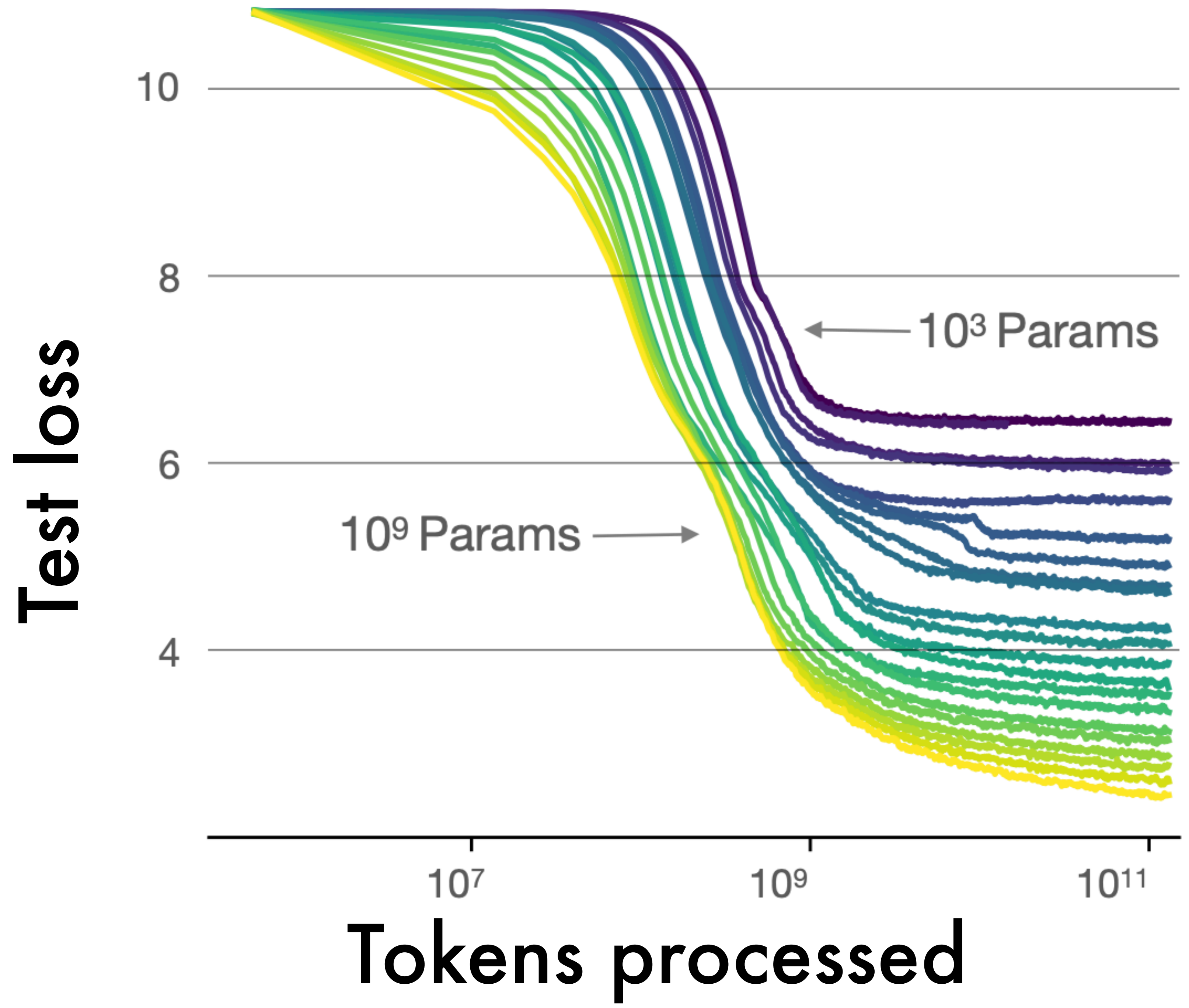
Performance also only weakly depends on model shape

References/Image credits

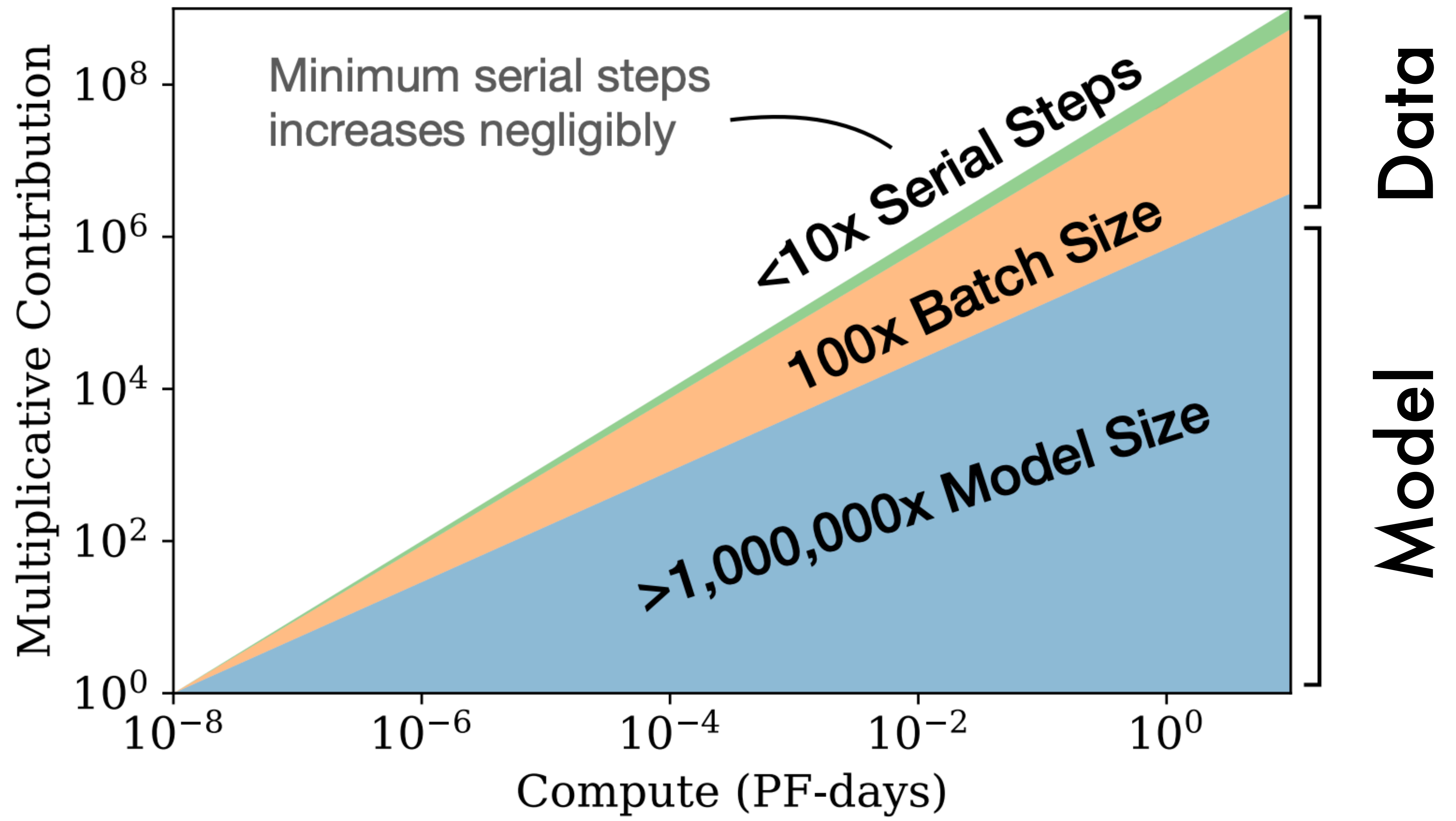
J. Kaplan et al., "Scaling Laws for Neural Language Models", arxiv (2020)

Transformer scaling laws for natural language

Intriguing characteristics



Larger models require **fewer samples** to reach the same performance



If extra compute is available, allocate most towards increasing the **model size**!

References/Image credits

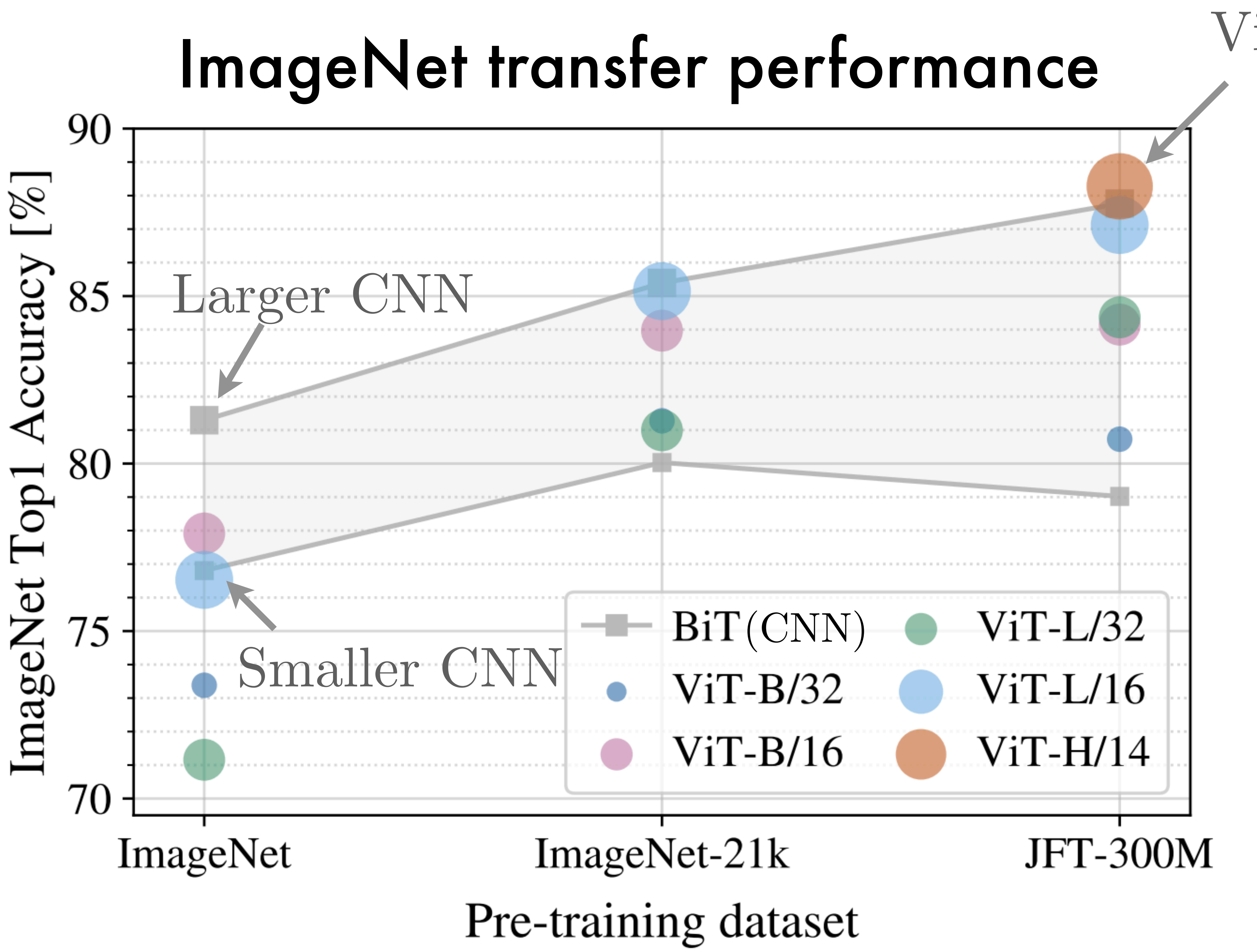
Kaplan et al., "Scaling Laws for Neural Language Models", arxiv (2020)

J. Hoffmann et al., "Training Compute-Optimal Large Language Models", arXiv (2022)

Later studies (Chinchilla) suggest greater focus on data

Scaling Vision Transformer

ViT: The importance of pre-training scale



1.3M images | 14M images | 303M images

In lower-data regime, the *stronger inductive biases* of the CNN work better:

- locality
- translation invariance

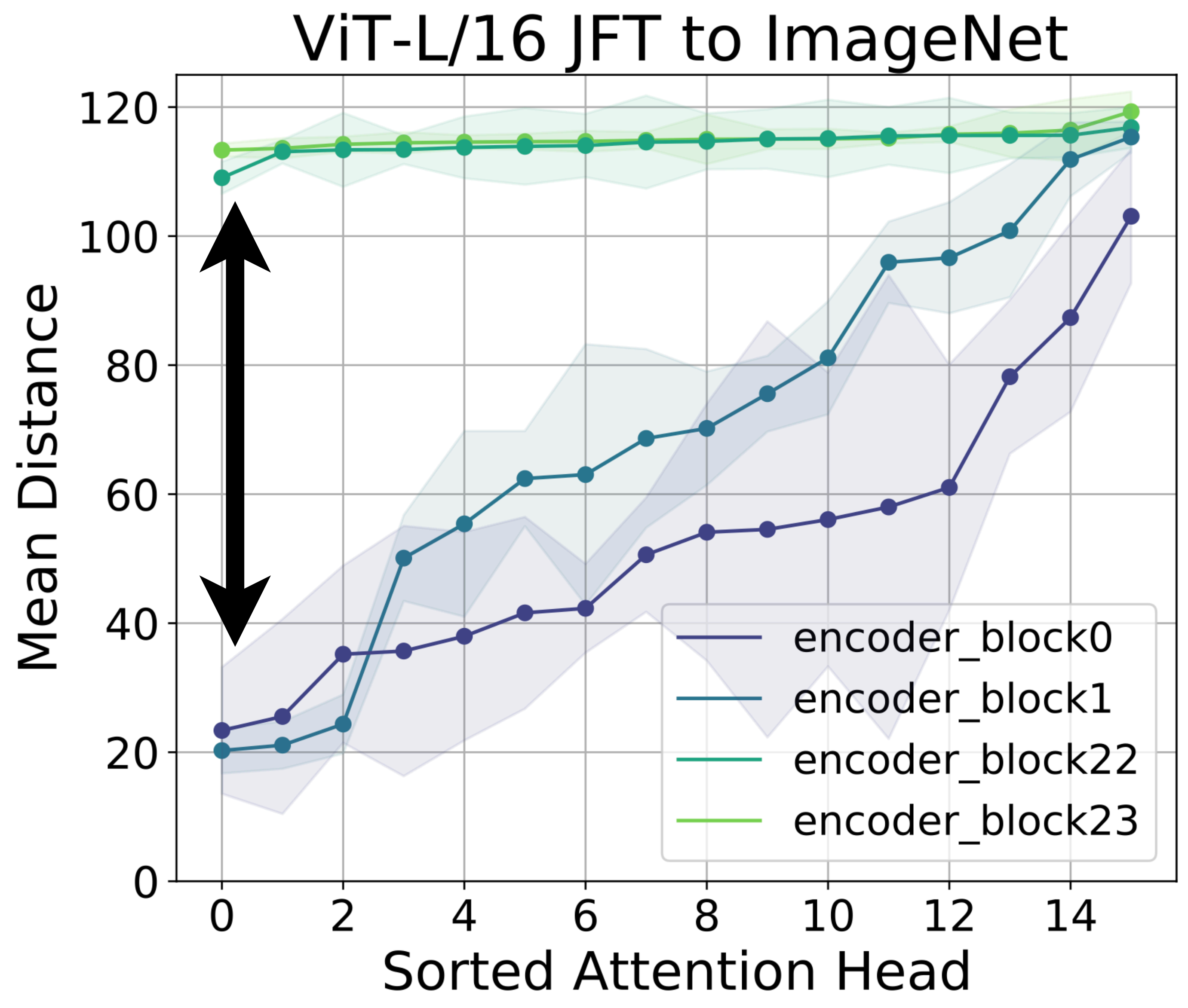
But in the higher-data regime (e.g. JFT-300M), ViT shines.

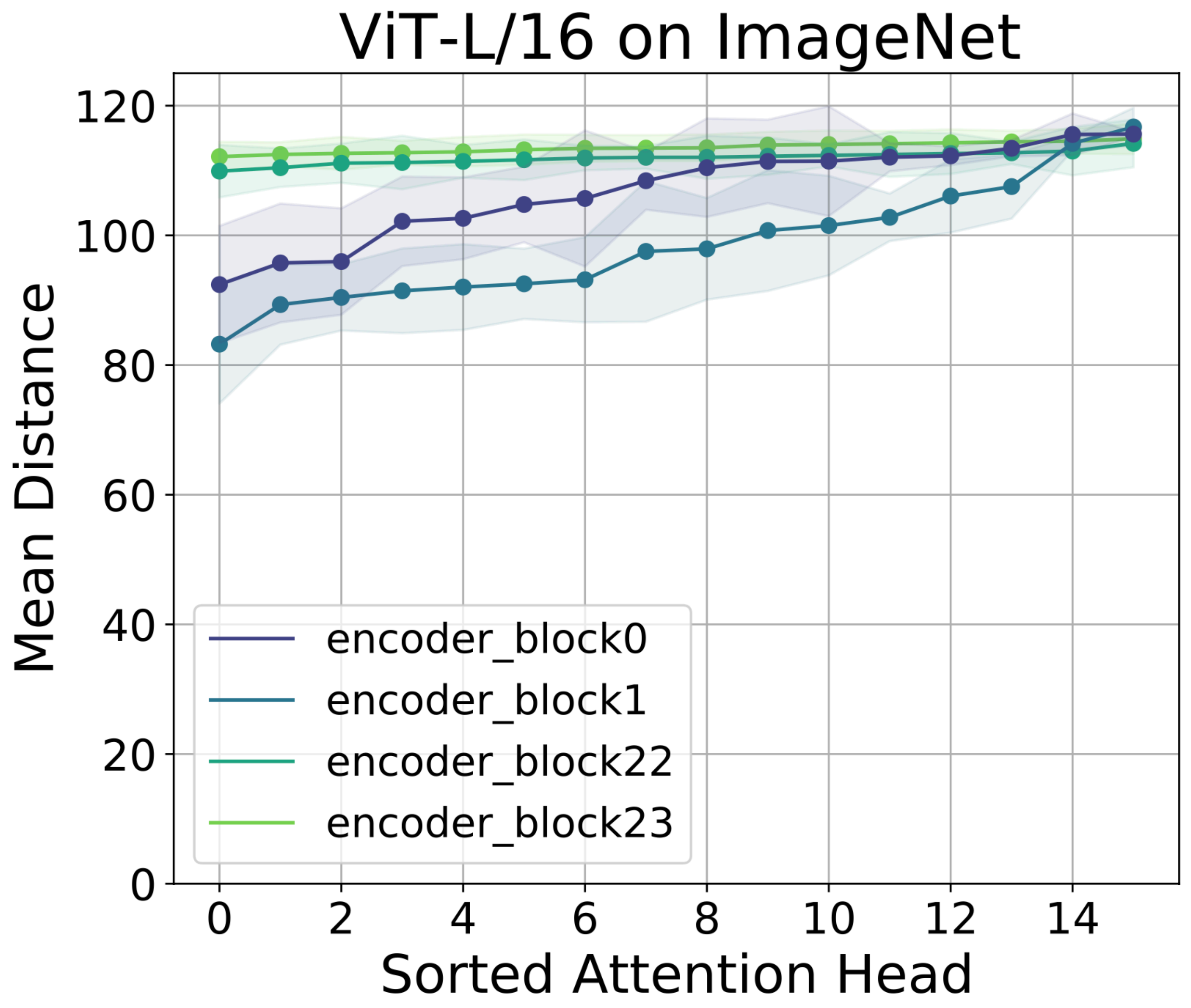
(This is "why Google")

References/Image credits

Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR (2021)

Vision Transformer and Learned Locality





Large-scale pretraining allows ViT to get "best of both": local and global

With enough data (300M images), earlier layers learn to "act locally" (like a CNN)

When pretraining on only 1M images, lower attention layers do not learn locality

References/Image credits

M. Raghu et al., "Do vision transformers see like convolutional neural networks?" NeurIPS (2021)

See video description below for links to:

Slides

References

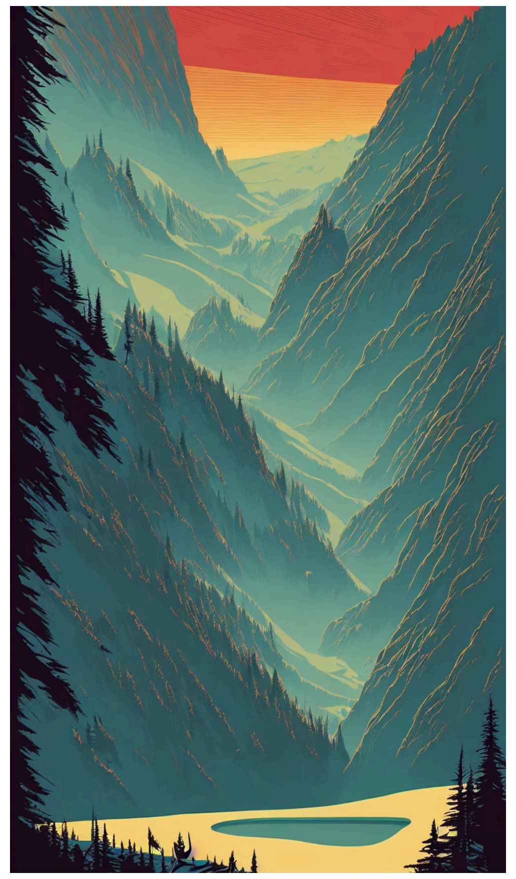


Image credit: Stable diffusion (lexica.art)
https://lexica.art/prompt/16135179-6a39-496a-9a7f-c4a06cdd8ff5