



Transformers

"The best idea in A.I." – Andrej Karpathy

Everything is a set of tokens

$$\{x_i\}_{i=1}^T \subset \mathbb{R}^D, \quad X \in \mathbb{R}^{D \times T}$$

The "feature maps" of transformers are sets of tokens that are "transformed" as they pass through the network.

tokenize $\downarrow$ Binary Pair Encoding dictionary (P. Gage 1994)

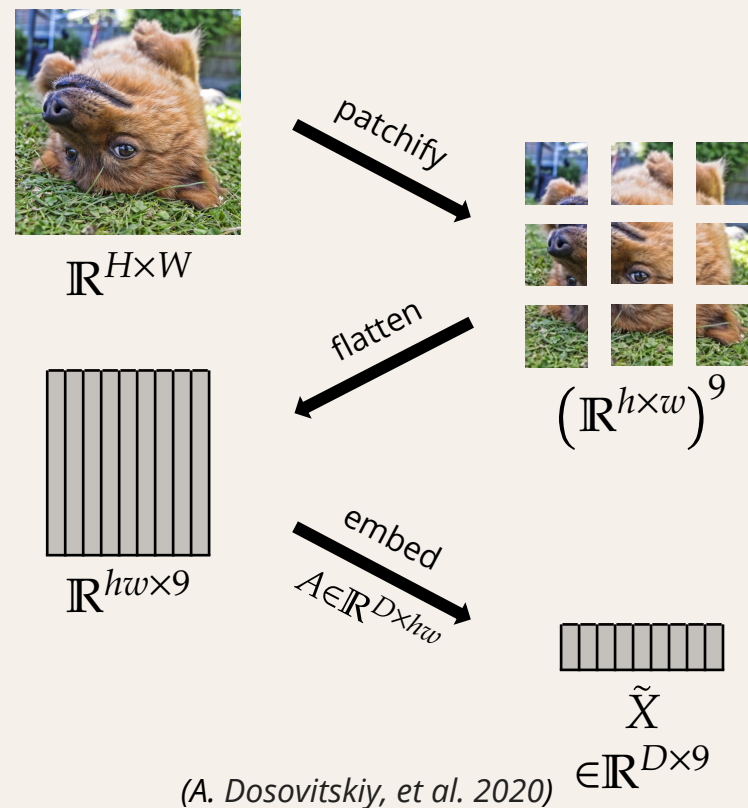
[791, 1595, 13043, 14370, 63, 315, 87970, 527, 7437, 315, 11460, 430, 527, 1595, **4806**, **291**, 63, 439, 814, 1522, 1555, **279**, 4009, 13]

embed $\downarrow$

- Skip-Gram with Negative Sampling (T. Mikolov, et al. 2013)
- Trained together with the transformer (J. Devlin, et al. 2018)
- Matroyshka Representation Learning (A. Kusupati, et al. 2022)

$$\{\tilde{x}_1, \dots, \tilde{x}_{23}\} \subset \mathbb{R}^D$$

$$\tilde{X} \in \mathbb{R}^{D \times 23}$$



Positional encoding

$$x_i = \tilde{x}_i + p_i$$

what?

where?

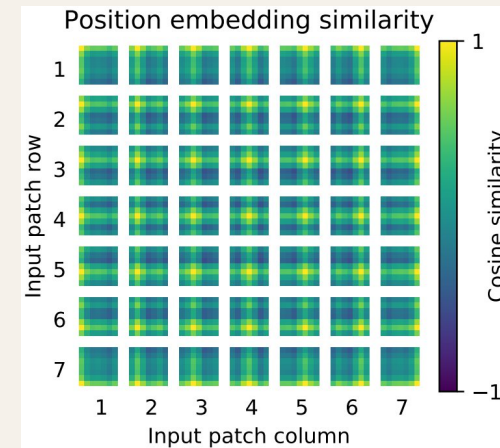
$p_i^T p_j$ is small if tokens i and j are far away and big if they are near

Trigonometric 1D for text

$$(p_i)_j = \begin{cases} \sin\left(\frac{i}{10.000^{j/D}}\right) & \text{if } j \text{ even,} \\ \cos\left(\frac{i}{10.000^{(j-1)/D}}\right) & \text{if } j \text{ odd.} \end{cases}$$

Vaswani, Ashish, et al. → **Attention is all you need.** Advances in neural information processing systems 30 (2017).

Trained 2D for images

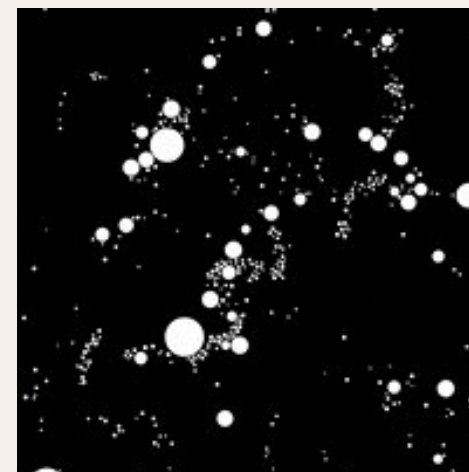
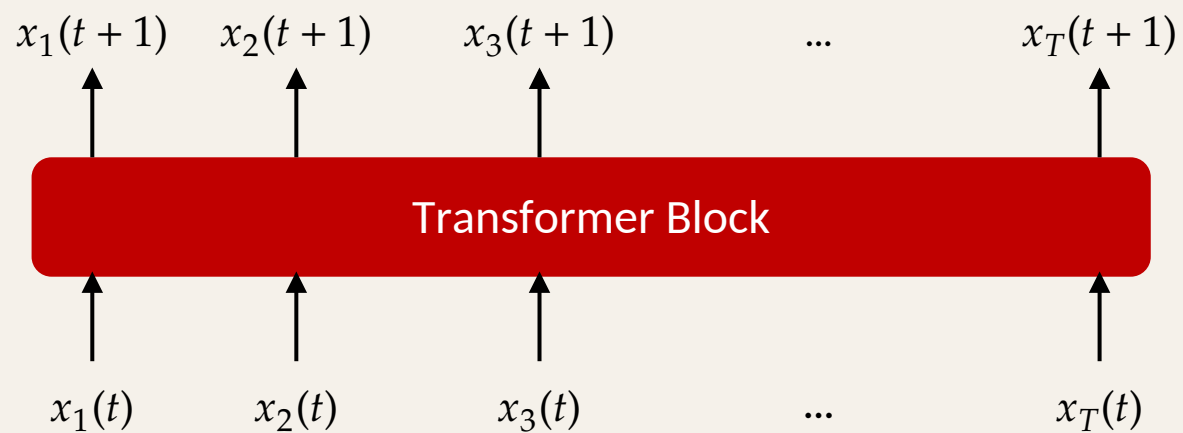


Dosovitskiy, Alexey, et al. → **An image is worth 16x16 words: Transformers for image recognition at scale.** arXiv:2010.11929 (2020).

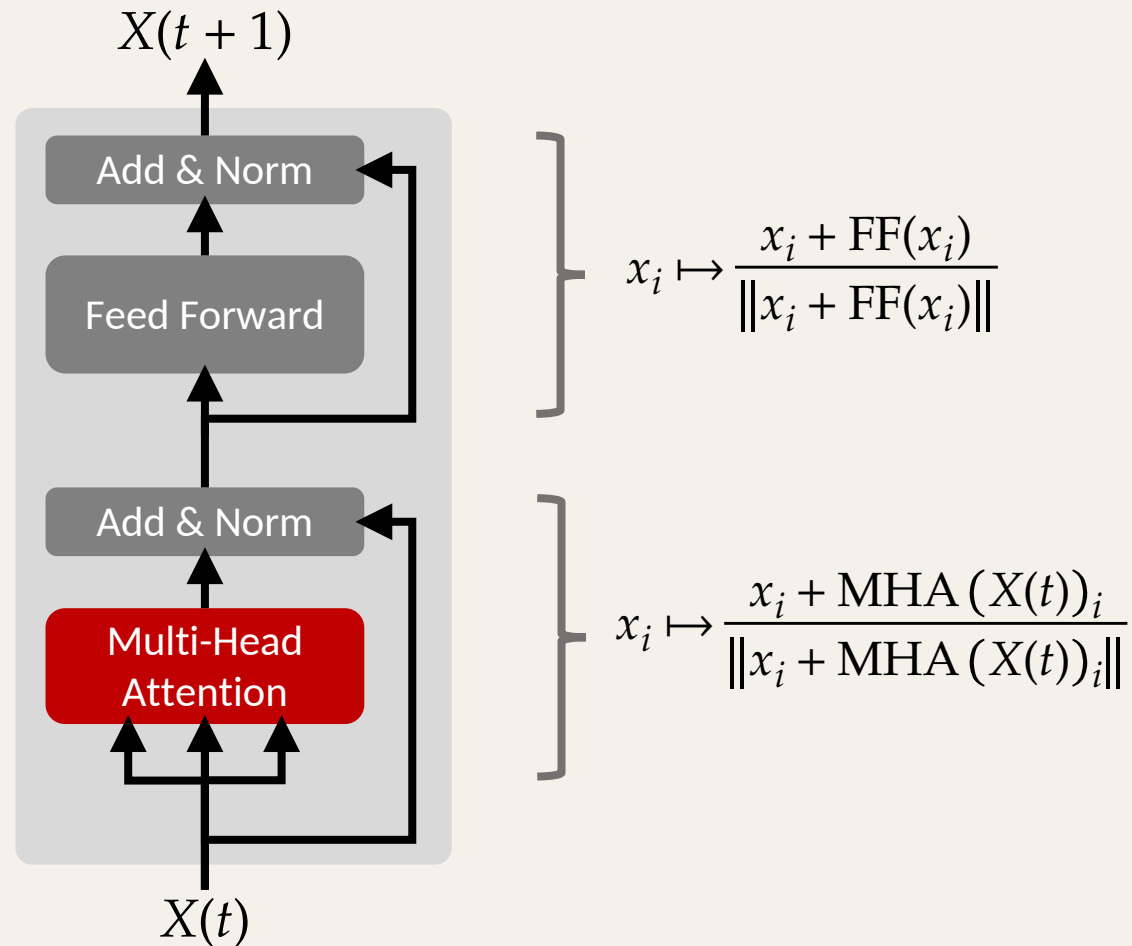
Conjecture

For any metric space (M, d_M) there exists a continuous $\phi: M \rightarrow L^2(\Omega)$ for some compact $\Omega \subset \mathbb{R}^n$ so that for all $p, q \in M$: $d_M(p, q)$ is a monotone decreasing function of $(\phi(p), \phi(q))_{L^2(\Omega)}$.

Transformer block



Transformer block



Attention

Vaswani, Ashish, et al. → Attention is all you need." *Advances in neural information processing systems* 30 (2017).

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
lukaszkaizer@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 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.

- What conventional NNs train:

$$y = Ax$$

- What transformers train in addition:

$$s_{ij} = x_i^T B x_j \in \mathbb{R} \quad \text{for all } i, j = 1, \dots, T$$

$$a_{ij} = \frac{e^{\tau s_{ij}}}{\sum_{k=1}^T e^{\tau s_{ik}}}$$

$$S = \begin{bmatrix} -0.5 & 3.1 \\ 2.0 & 2.4 \end{bmatrix} \mapsto A = \begin{bmatrix} 0.027 & 0.973 \\ 0.401 & 0.599 \end{bmatrix}$$

Query, key & value

D^2 is a lot of parameters: pick $d_{qk} < D$ and $W_q, W_k \in \mathbb{R}^{d_{qk} \times D}$

$$S = X^T B X = X^T \left((W_q)^T W_k \right) X = (W_q X)^T (W_k X)$$

$$A = \text{row-wise softmax of } \frac{1}{\sqrt{d_{qk}}} S \in \mathbb{R}^{T \times T}$$

Generate a *value* for each token, $d_v < D$ and $W_v \in \mathbb{R}^{d_v \times D}$

$$Y = W_v X A \in \mathbb{R}^{d_v \times T}$$

The Y constitutes the output of a single *attention head* and W_q, W_k and W_v are its parameters.

Multi-head attention

Use a set of $h > 0$ *attention heads* to generate a set of outputs $Y^{(1)}, Y^{(2)}, \dots, Y^{(h)} \in \mathbb{R}^{d_v \times T}$

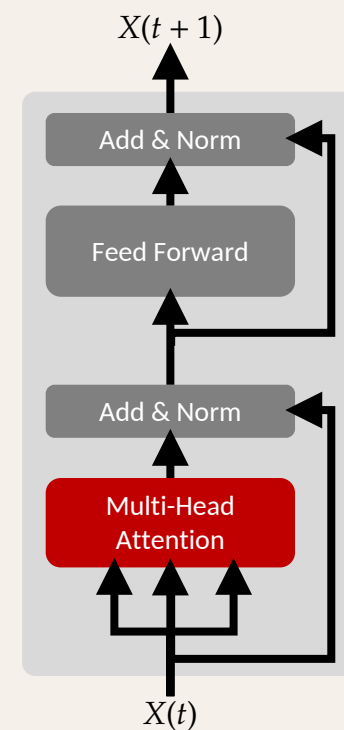
$$Y = \begin{bmatrix} Y^{(1)} \\ \vdots \\ Y^{(h)} \end{bmatrix} \in \mathbb{R}^{hd_v \times T}.$$

Take a matrix $W_p \in \mathbb{R}^{D \times hd_v}$ to project the concatenated outputs of the heads

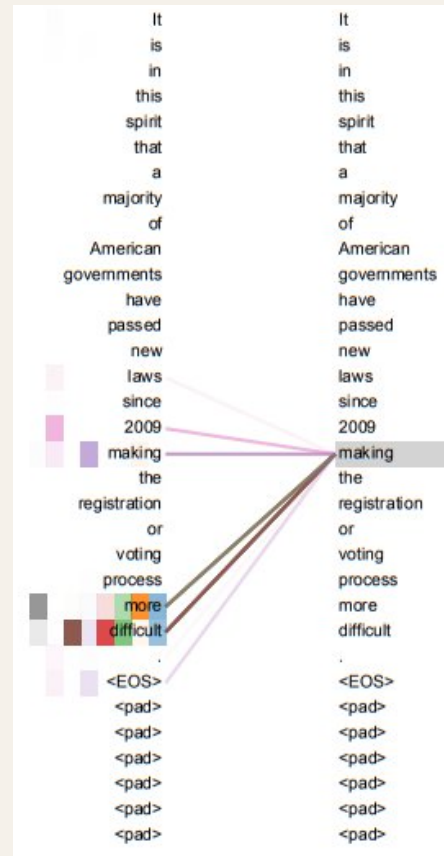
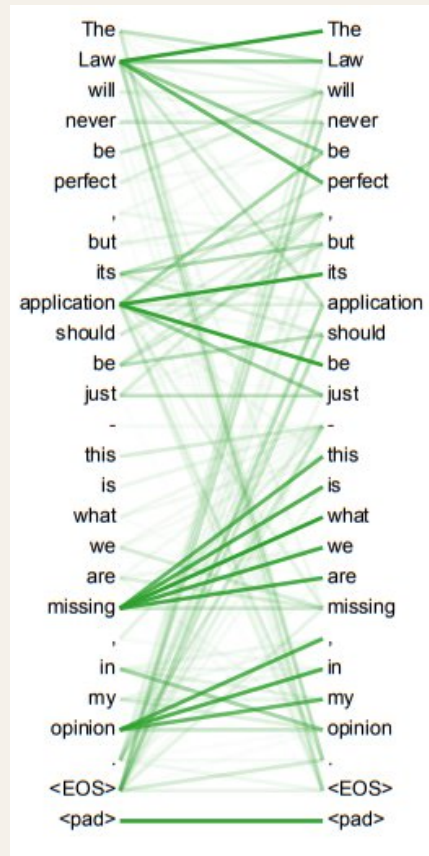
$$\text{MHA}(X) = W_p Y \in \mathbb{R}^{D \times T}.$$

Add and normalize token-wise (LayerNorm, RMSNorm):

$$X \mapsto \text{norm}(X + \text{MHA}(X)).$$



Attention can be intuitive



Vaswani, Ashish, et al. → **Attention is all you need.** Advances in neural information processing systems 30 (2017).

jacobgil.github.io/deeplearning/vision-transformer-explainability

Attention computation is the bottleneck

Quadratic complexity in number of tokens T (a.k.a. *context window* in text transformers)
⇒ lots of work on efficient implementation and simplifying

arXiv:2205.14135v2 [cs.LG] 23 Jun 2022

FLASHATTENTION: Fast and Memory Efficient Attention with IO-Aware

Tri Dao[†], Daniel Y. Fu[‡], Stefano Ermon[‡], Atri

[†]Department of Computer Science,
Department of Computer Science and Engineering
{trid,danfu}@cs.stanford.edu, ermon@stanford.edu
chrismre@cs.stanford.edu

June 24, 2022

Abstract

Transformers are slow and memory-hungry on long sequence of self-attention is quadratic in sequence length. Approximate to address this problem by trading off model quality to reduce not achieve wall-clock speedup. We argue that a missing prior is aware—accounting for reads and writes between levels of GPU an IO-aware exact attention algorithm that uses tiling to reduce between GPU high bandwidth memory (HBM) and GPU on-chip of FLASHATTENTION, showing that it requires fewer HBM optimal for a range of SRAM sizes. We also extend FLASHATT to approximate attention algorithm that is faster than any FLASHATTENTION trains Transformers faster than existing bas on BERT-large (seq. length 512) compared to the MLPerf 1 GPT-2 (seq. length 1K), and 2.4x speedup on long-range attention and block-sparse FLASHATTENTION enable longer context in T (0.7 better perplexity on GPT-2 and 6.4 points of lift on long capabilities: the first Transformers to achieve better-than-chal (seq. length 16K, 61.4% accuracy) and Path-256 (seq. length 6

1 Introduction

Transformer models [82] have emerged as the most widely used language processing and image classification. Transformers equipping them with longer context remains difficult [80], since has time and memory complexity quadratic in sequence length attention faster and more memory-efficient can help Transformer challenges for long sequences.

Many approximate attention methods have aimed to reduce attention. These methods range from sparse-approximation [51] and their combinations [3, 9, 92]. Although these methods reduce near-linear in sequence length, many of them do not display wall and have not gained wide adoption. One main reason is that they correlate with wall-clock speed) and tend to ignore overheads of

FLASHATTENTION: Faster Attention with Better Parallelism

Tri Dao

¹Department of Computer Science
²Department of Computer Science
trid@cs.stanford.edu

July 18, 2022

Abstract

Scaling Transformers to longer sequence lengths has promising to improve performance in language models as well as to unlock new applications in code, audio, and a main bottleneck in scaling to longer sequences, as is the sequence length. FLASHATTENTION [5] exploits this significant memory saving (linear instead of quadratic) as baselines), with no approximation. However, FLASHATT matrix-multiply (GEMM) operations, reaching only 25% observe that the inefficiency is due to suboptimal work warps on the GPU, causing either low-occupancy or unne FLASHATTENTION-2, with better work partitioning to the algorithm to reduce the number of non-matmul FLOPs for a single head, across different thread blocks to more distribute the work between warps to reduce communication 2x speedup compared to FLASHATTENTION, reaching 5 A100 and getting close to the efficiency of GEMM open end-to-end to train GPT-style models, FLASHATTENTION per A100 GPU (72% model FLOPs utilization).¹

1 Introduction

Scaling up the context length of Transformers [18] is a has runtime and memory requirements quadratic in the beyond the standard 2k sequence length limit to train medium and long-form videos. Just within the last year, there has context than before: GPT-4 [12] with context length 32k Anthropic’s Claude with context length 100k. Emerging writing have demonstrated a need for models with such large

To reduce the computational requirement of attention methods proposed to approximate attention [2, 3, 4, 8, some use cases, as far as we know, most large-scale training this, Dao et al. [5] proposed to reorder the attention computation (recomputation) to significantly speed it up and reduce its

Linear Attention Mechanism for Semantic Segmentation

Rui Li^{*}, Jianlin Su^{*}, Chenxi Duan, and Si

Abstract—The attention mechanism can meticulously refine the feature maps and eminently boost the performance of the network, and has become an essential technology applied in computer vision and natural language processing. However, the memory and computational costs of the dot-product attention mechanism widely used in vision and language tasks increase quadratically with the spatio-temporal size of the input. Such growth hinders the usage of attention mechanism in application scenarios with large inputs, e.g., large-scale videos, long sequences, or high-resolution images. In this paper, to remedy this deficiency, we propose a Linear Attention Mechanism which is approximate to dot-product attention with much less memory and computational costs. The efficient design makes the incorporation between attention mechanisms and neural networks more flexible and versatile. Experiments conducted on semantic segmentation demonstrated the effectiveness of linear attention mechanism. Code is available at <https://github.com/jianlinsu/Linear-Attention-Mechanism>.

Index Terms—semantic segmentation, high-resolution remote sensing images, linear attention mechanism

I. INTRODUCTION

Attention mechanisms, bolstered by their powerful capability of exploiting long-range dependencies of feature maps and facilitating neural networks to attain global contextual information, are at the forefront of convolution and recurrence researches. Dot-product attention mechanism, which generates the response at every pixel by weighting features at all pixels in the previous layer, expands the receptive field to the whole input feature maps in one pass. Benefiting from their specialty on capturing long-range dependencies, dot-product attention mechanisms have been widely used in vision and language tasks. Dot-product-attention-based Transformer has manifested the state-of-the-art performances in nearly all tasks in natural language processing. The non-local module [1], a dot-product-based attention modified for computer vision, has demonstrated its superiority on image classification [2], object detection [3], semantic segmentation [4], and panoptic segmentation [5].

However, as the memory and computational costs of the dot-product attention mechanism increase quadratically with the spatio-temporal size of the input, how to model global dependency on large inputs, e.g. large-scale videos, long sequences, or high-resolution images, remains an intractable problem. To alleviate the enormous computational

consumptions, attention matrix $O(N^2)$. Using the complexity to a linear dot-product modified the paper

In this paper, we propose a Linear Attention Mechanism which is approximate to dot-product attention with much less memory and computational costs. The efficient design makes the incorporation between attention mechanisms and neural networks more flexible and versatile. Experiments conducted on semantic segmentation demonstrated the effectiveness of linear attention mechanism. Code is available at <https://github.com/jianlinsu/Linear-Attention-Mechanism>.

- 1) We propose a Linear Attention Mechanism which is approximate to dot-product attention with much less memory and computational costs.
- 2) The linear attention mechanism is more flexible and versatile.
- 3) Performance is improved by the proposed mechanism.

A. Dot-Product Attention
To enhance the performance of the network, the dot-product attention mechanism is widely used in vision and language tasks. Dot-product attention mechanism, which generates the response at every pixel by weighting features at all pixels in the previous layer, expands the receptive field to the whole input feature maps in one pass. Benefiting from their specialty on capturing long-range dependencies, dot-product attention mechanisms have been widely used in vision and language tasks. Dot-product-attention-based Transformer has manifested the state-of-the-art performances in nearly all tasks in natural language processing. The non-local module [1], a dot-product-based attention modified for computer vision, has demonstrated its superiority on image classification [2], object detection [3], semantic segmentation [4], and panoptic segmentation [5].

Efficient Attention: Attention with Linear Complexities

Shen Zhuoran^{1*}

Independent Researcher
4244 University Way NE #85406, Seattle, WA 98105, United States
cmsflash99@gmail.com

Zhang Mingyuan², Zhao Haiyu, Yi Shuai
SenseTime International

182 Cecil Street, #36-02 Frasers Tower, Singapore 069547
zhangmingyuan, zhaohaiyu, yishuai@sensetime.com

Li Hongsheng
The Chinese University of Hong Kong
Sha Tin, Hong Kong
hsl1@ee.cuhk.edu.hk

Abstract

Dot-product attention has wide applications in computer vision and natural language processing. However, its memory and computational costs grow quadratically with the input size. Such growth prohibits its application on high-resolution inputs. To remedy this drawback, this paper proposes a novel efficient attention mechanism equivalent to dot-product attention but with substantially less memory and computational costs. Its resource efficiency allows more widespread and flexible integration of attention modules into a network, which leads to better accuracies. Empirical evaluations demonstrated the effectiveness of its advantages. Efficient attention modules brought significant performance boosts to object detectors and instance segmenters on MS-COCO 2017. Further, the resource efficiency democratizes attention to complex models, where high costs prohibit the use of dot-product attention. As an exemplar, a model with efficient attention achieved state-of-the-art accuracies for stereo depth estimation on the Scene Flow dataset. Code is available at <https://github.com/cmsflash/efficient-attention>.

1. Introduction

Dot-product attention [1, 22, 23] is a prevalent mechanism in neural networks for long-range dependency model-

ing, a key challenge to deep learning that convolution and recurrence struggle to solve. The mechanism computes the response at every position as a weighted sum of features at all positions in the previous layer. In contrast to the limited receptive fields of convolution or the recurrent layer, dot-product attention expands the receptive field to the entire input in one pass. Using dot-product attention to efficiently model long-range dependencies allows convolution and recurrence to focus on local dependency modeling, in which they specialize. The non-local module [23], an adaptation of dot-product attention for computer vision, achieved state-of-the-art performance on video classification [23] and generative adversarial image modeling [28, 2] and demonstrated significant improvements on object detection [23], instance segmentation [23], person re-identification [14], image de-raining [13], etc.

However, global dependency modeling on large inputs (e.g. long sequences, high-resolution images, large videos) remains an open problem. The quadratic¹ memory and computational complexities with respect to the input size of dot-product attention inhibits its application on large inputs. For instance, a non-local module uses over 1 GB of GPU memory and over 25 TMACC² of computation for a 64-channel 128 × 128 feature map or over 68 GB and over 1.6 TMACC for a 64-channel 64 × 64 × 32 3D feature vol-

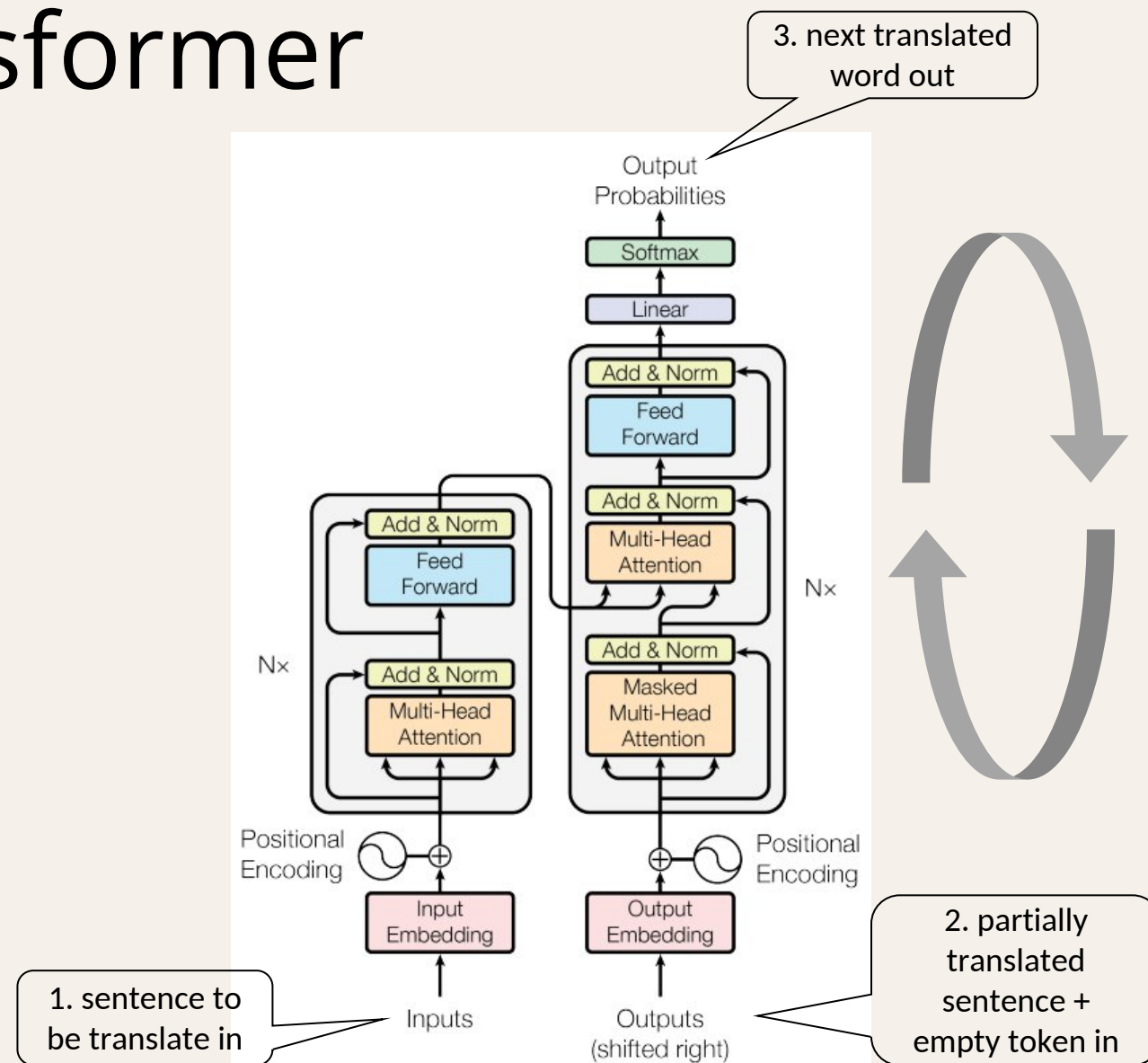
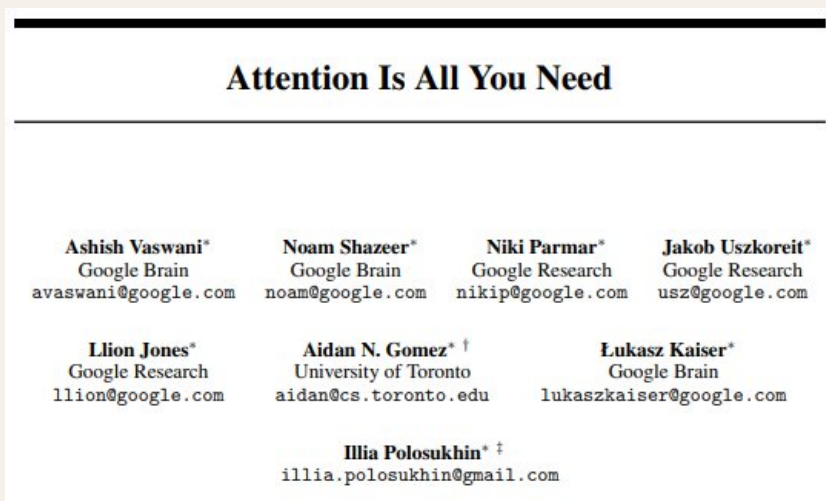
¹The complexities are quadratic with respect to the spatiotemporal size of the input, which is quadratically w.r.t. the side length of a 2D feature map, or cubically w.r.t. the dimension of a 3D feature volume.

Notable transformers and their use

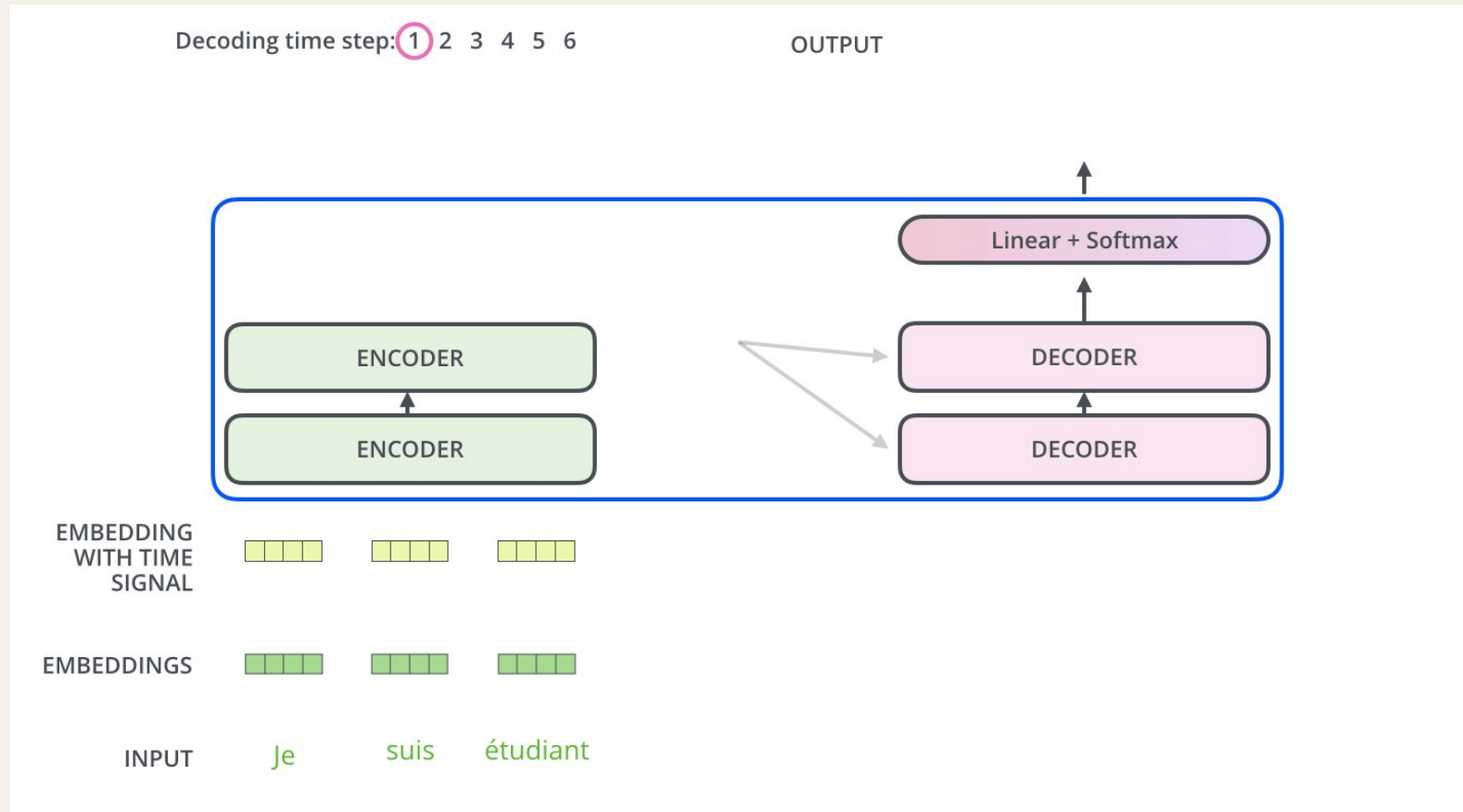
Model	Use	Type
The original transformer <i>A. Vaswani, et al. 2017</i>	language translation	encoder/decoder
BERT <i>J. Devlin, et al. 2018</i>	text embedding • sentiment analysis • search • spam detector	encoder-only
GPT <i>OpenAI 2017-...</i>	chatbot, etc.	auto-regressive decoder-only
Llama <i>Meta 2023</i>	chatbot, etc.	open source, auto-regressive decoder-only
ViT <i>A. Dosovitsky, et al. 2020</i>	image classification	decoder-only or as an encoder in image-to-text models
DiT <i>W. Peebles, et al. 2023</i>	image generation	transformer and diffusion model have a baby

The original transformer

Language translation

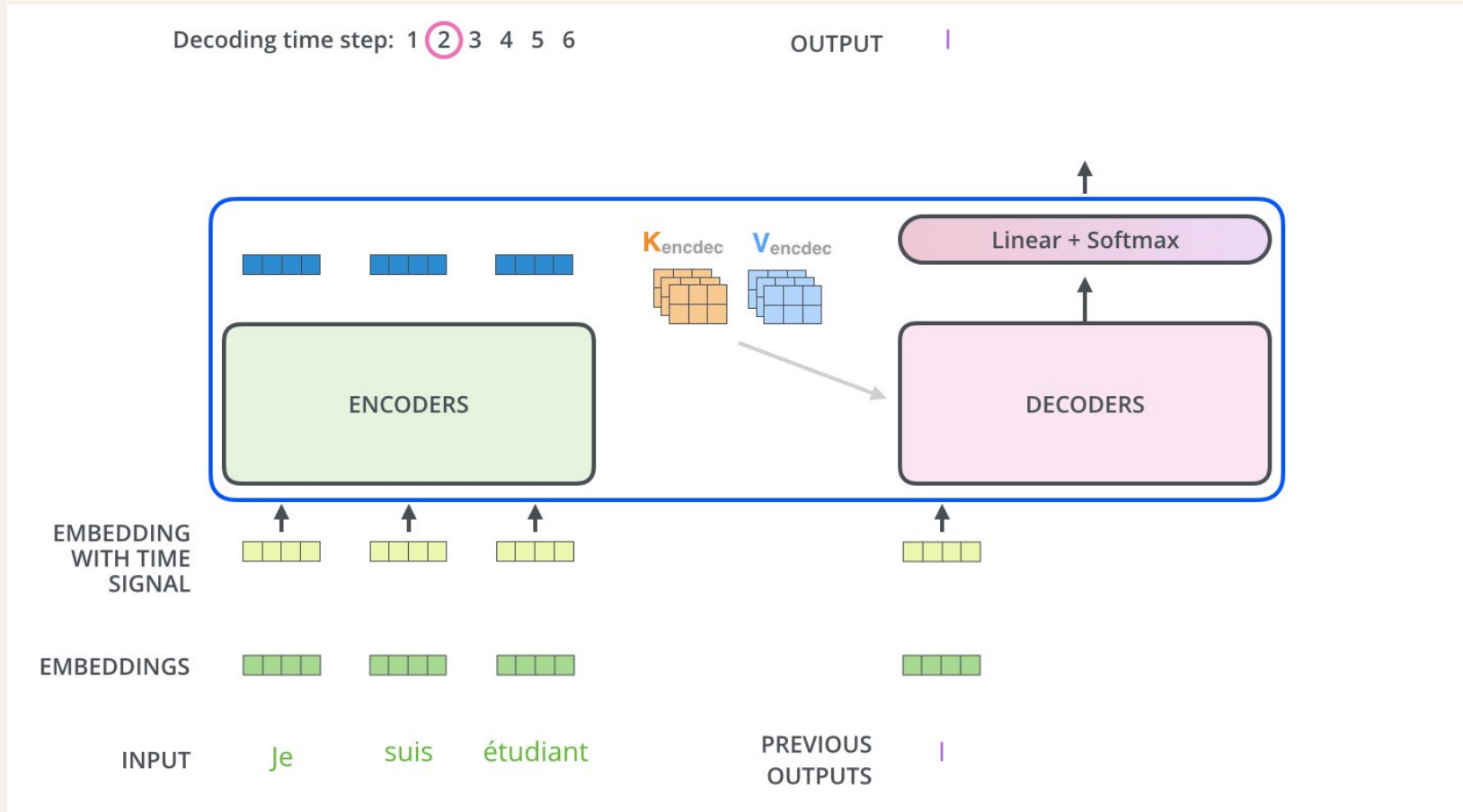


The original transformer



J. Alammr (2018) <https://jalammar.github.io/illustrated-transformer/>

The original transformer



J. Alammr (2018) <https://jalammar.github.io/illustrated-transformer/>

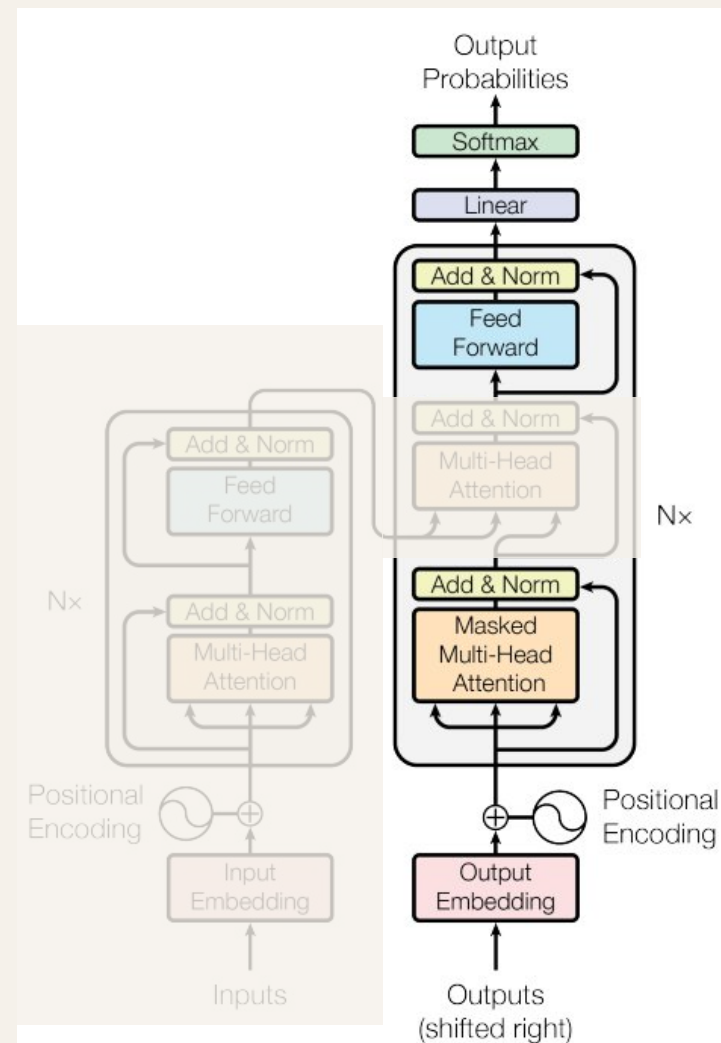
Llama (and relatives)

- Decoder-only
- 70B parameters (Llama 3)
- Autoregressive
- Training:
 - Self-supervised pre-training (trillions of tokens)
 - Fine-tuning (instruction following, etc.)
- SwiGLU: $x_i \mapsto A_3(\text{swish}_1(A_1x_i) \odot A_2x_i)$
- 8000 context window (Llama 3)

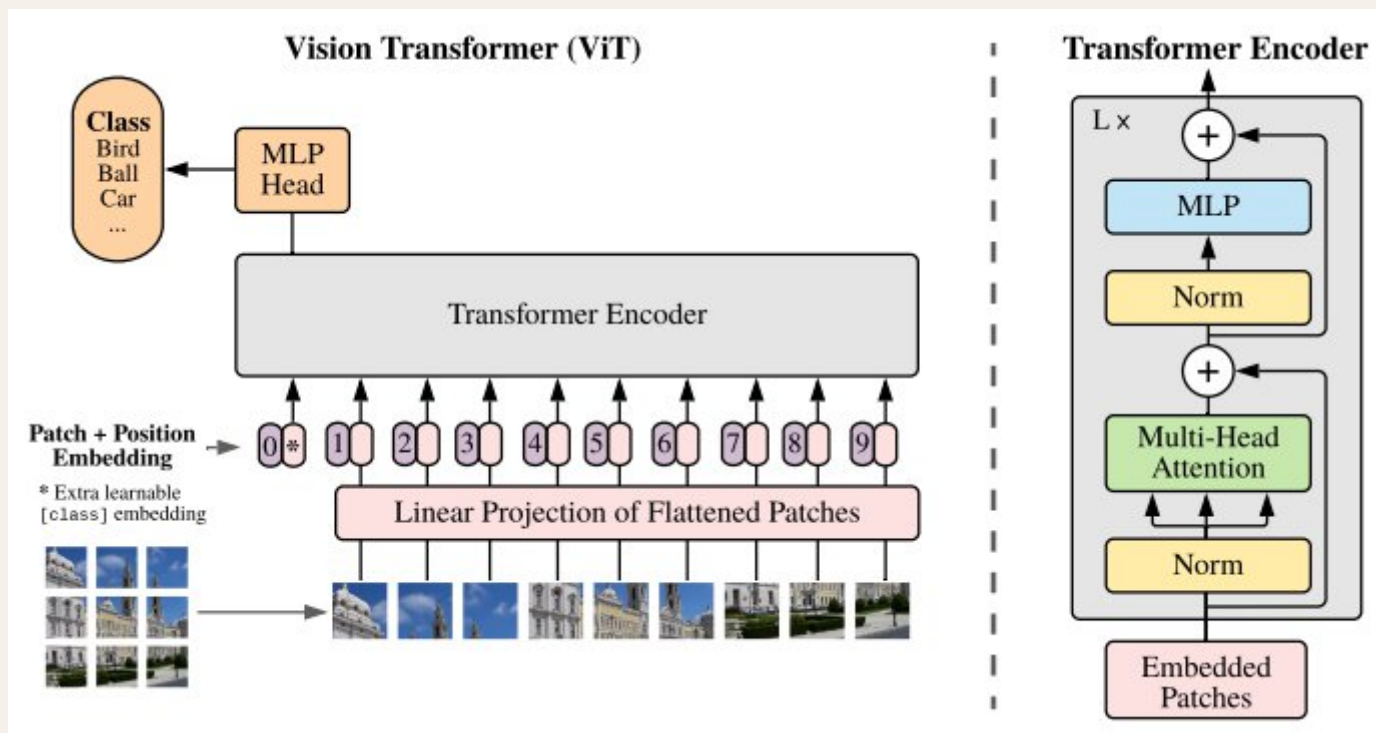
LLAMA 2: Open Foundation and Fine-Tuned Chat Models

Hugo Touvron* Louis Martin† Kevin Stone†
Peter Albert Amjad Almahairi Yasmine Babaei Nikolay Bashlykov Soumya Batra
Prajwal Bhargava Shruti Bhosale Dan Bikel Lukas Blecher Cristian Canton Ferrer Moya Chen
Guillem Cucurull David Esiobu Jude Fernandes Jeremy Fu Wenyin Fu Brian Fuller
Cynthia Gao Vedanuj Goswami Naman Goyal Anthony Hartshorn Saghar Hosseini Rui Hou
Hakan Inan Marcin Kardas Viktor Kerkez Madian Khabsa Isabel Kloumann Artem Korenev
Punit Singh Koura Marie-Anne Lachaux Thibaut Lavril Jenya Lee Diana Liskovich
Yinghai Lu Yuning Mao Xavier Martinet Todor Mihaylov Pushkar Mishra
Igor Molybog Yixin Nie Andrew Poulton Jeremy Reizenstein Rashi Rungta Kalyan Saladi
Alan Schelten Ruan Silva Eric Michael Smith Ranjan Subramanian Xiaoqing Ellen Tan Binh Tang
Ross Taylor Adina Williams Jian Xiang Kuan Puxin Xu Zheng Yan Iliyan Zarov Yuchen Zhang
Angela Fan Melanie Kambadur Sharan Narang Aurelien Rodriguez Robert Stojnic
Sergey Edunov Thomas Scialom*

GenAI, Meta



ViT



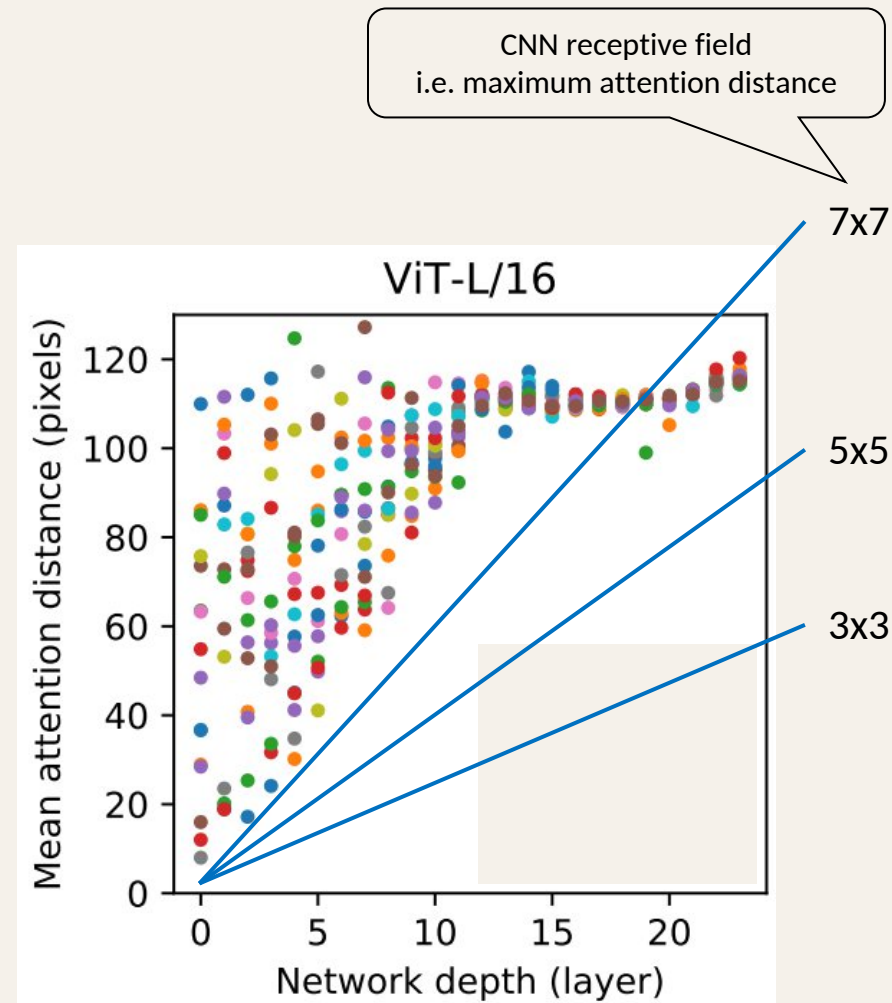
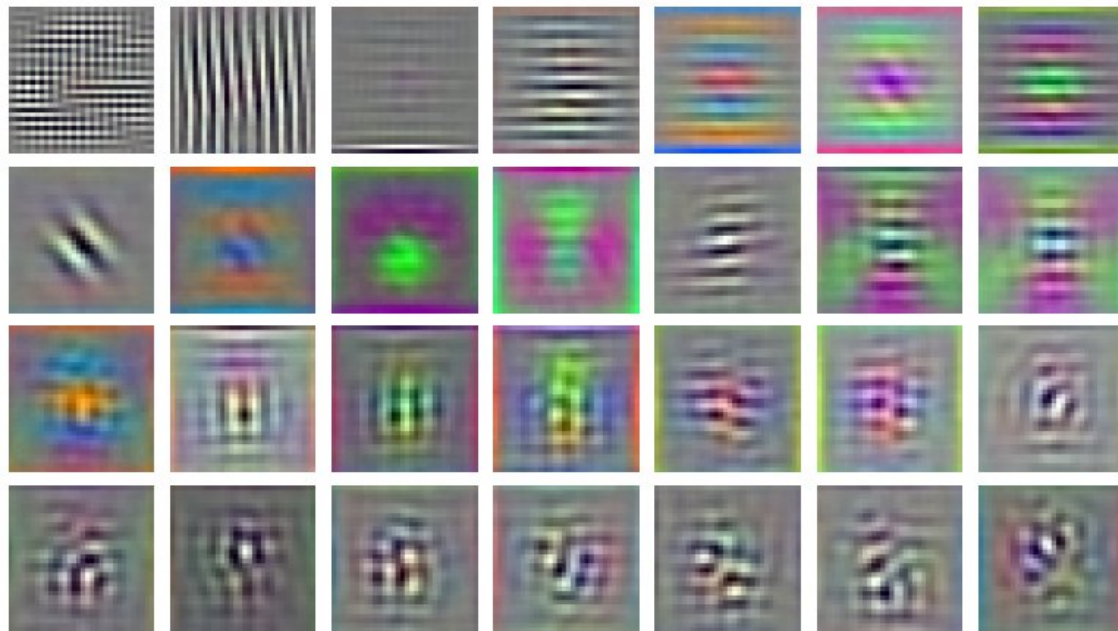
AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

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, neilhoulby}@google.com

ViT vs. CNN

RGB embedding filters
(first 28 principal components)



ViT: Geometry is optional 😞

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

Table 8: Results of the ablation study on positional embeddings with ViT-B/16 model evaluated on ImageNet 5-shot linear.

Multimodality

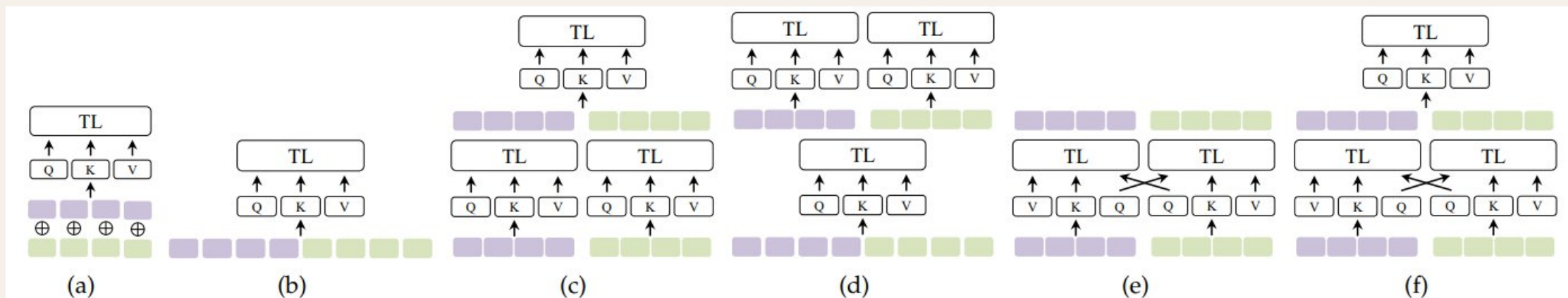


Fig. 2. Transformer-based cross-modal interactions: (a) Early Summation, (b) Early Concatenation, (c) Hierarchical Attention (multi-stream to one-stream), (d) Hierarchical Attention (one-stream to multi-stream), (e) Cross-Attention, and (f) Cross-Attention to Concatenation. “Q”: Query embedding; “K”: Key embedding; “V”: Value embedding. “TL”: Transformer Layer. Best viewed in colour.

Multimodal Learning with Transformers: A Survey

Peng Xu, Xiatian Zhu, and David A. Clifton

Abstract—Transformer is a promising neural network learner, and has achieved great success in various machine learning tasks. Thanks to the recent prevalence of multimodal applications and big data, Transformer-based multimodal learning has become a hot topic in AI research. This paper presents a comprehensive survey of Transformer techniques oriented at multimodal data. The main contents of this survey include: (1) a background of multimodal learning, Transformer ecosystem, and the multimodal big data era, (2) a systematic review of *Vanilla* Transformer, Vision Transformer, and multimodal Transformers, from a geometrically topological perspective, (3) a review of multimodal Transformer applications, via two important paradigms, *i.e.*, for multimodal pretraining and for specific multimodal tasks, (4) a summary of the common challenges and designs shared by the multimodal Transformer models and applications, and (5) a discussion of open problems and potential research directions for the community.

Index Terms—Multimodal Learning, Transformer, Introductory, Taxonomy, Deep Learning, Machine Learning.

Simplifying

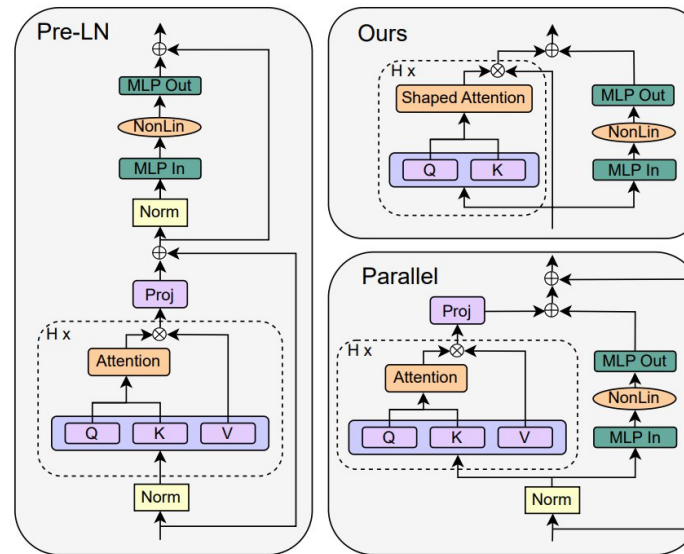


Figure 1: Comparison between different Transformer blocks. (Left) The standard Pre-LN block. (Top Right) Our most simplified block. (Bottom Right) The parallel block (Wang & Komatsuzaki, 2021). Like the parallel block, our block eschews the need for sequential sub-blocks, but we additionally remove all skip connections and normalisation layers, as well as value and projection parameters. Here, $\otimes$ denotes a matrix multiplication, and $\oplus$ denotes a (potentially weighted) sum.

SIMPLIFYING TRANSFORMER BLOCKS

Bobby He & Thomas Hofmann*
Department of Computer Science, ETH Zurich

ABSTRACT

A simple design recipe for deep Transformers is to compose identical building