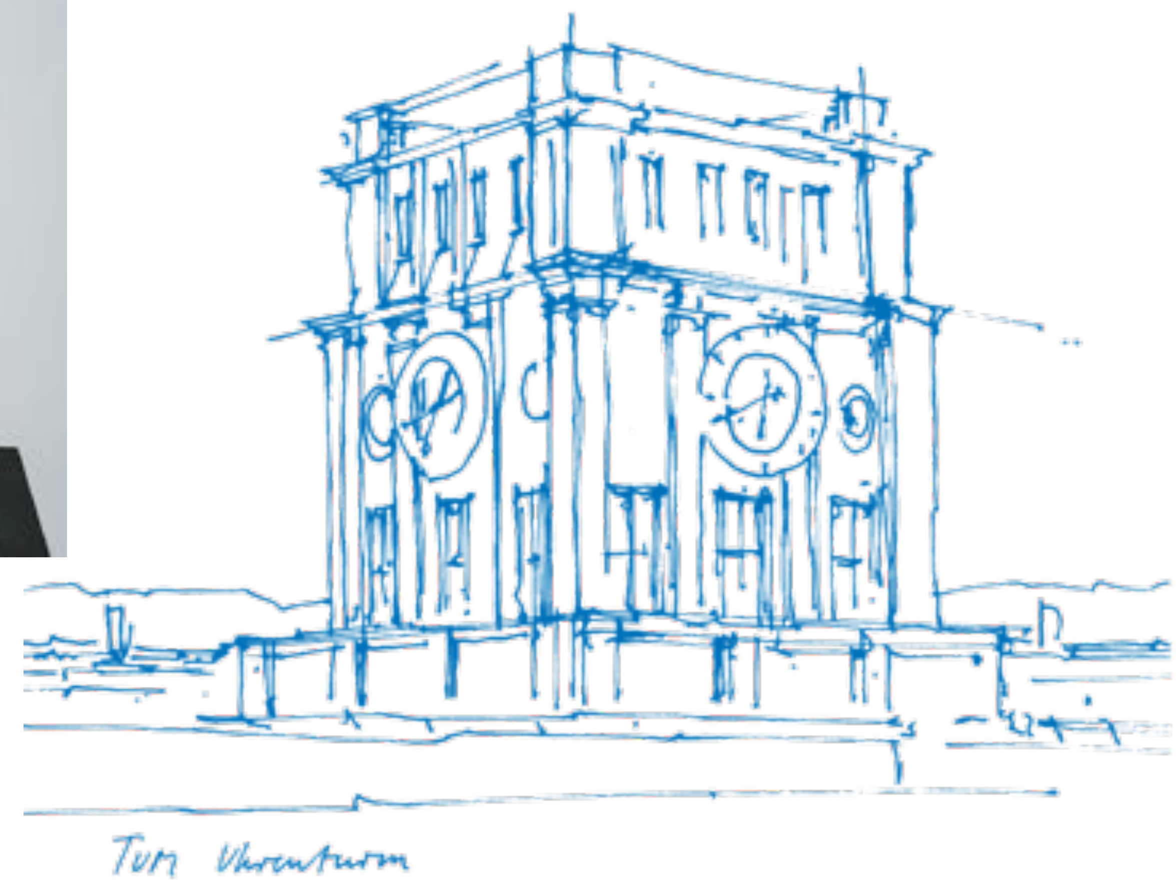
Transformers for Medical Imaging

Chengzhi Shen
MS.c. Biomedical Computing



Outline

- Self-attention & Transformers[1]
- VisionTransformers[2]
- Applying Transformers in Medical Imaging[3][4]

[1] Attention is all you need, A. Vaswani, et al., Advances in Neural Information Processing Systems pp. 5998-6008, 2017
[2] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, A. Dosovitskiy, et al. arXiv:2010.11929, 2020
[3] UNETR: Transformers for 3D Medical Image Segmentation, A. Hatamizadeh et al., arXiv:2103.10504v1, 2021
[4] Shamshad, Fahad, et al. "Transformers in medical imaging: A survey." arXiv preprint arXiv:2201.09873 (2022).

Attention is All You Need

Attention is All You Need

- First Transformer model, designed for text sequences in Natural Language Processing (NLP)
- Based on **self-attention**
- Considered as **fundamental** AI architectures like CNN, RNN
- A large number of following up works:
  - Language models: BERT, GPT etc.
  - Other modalities: VisionTransformers, Graphormers…

Source: https://jalammar.github.io/illustrated-transformer/

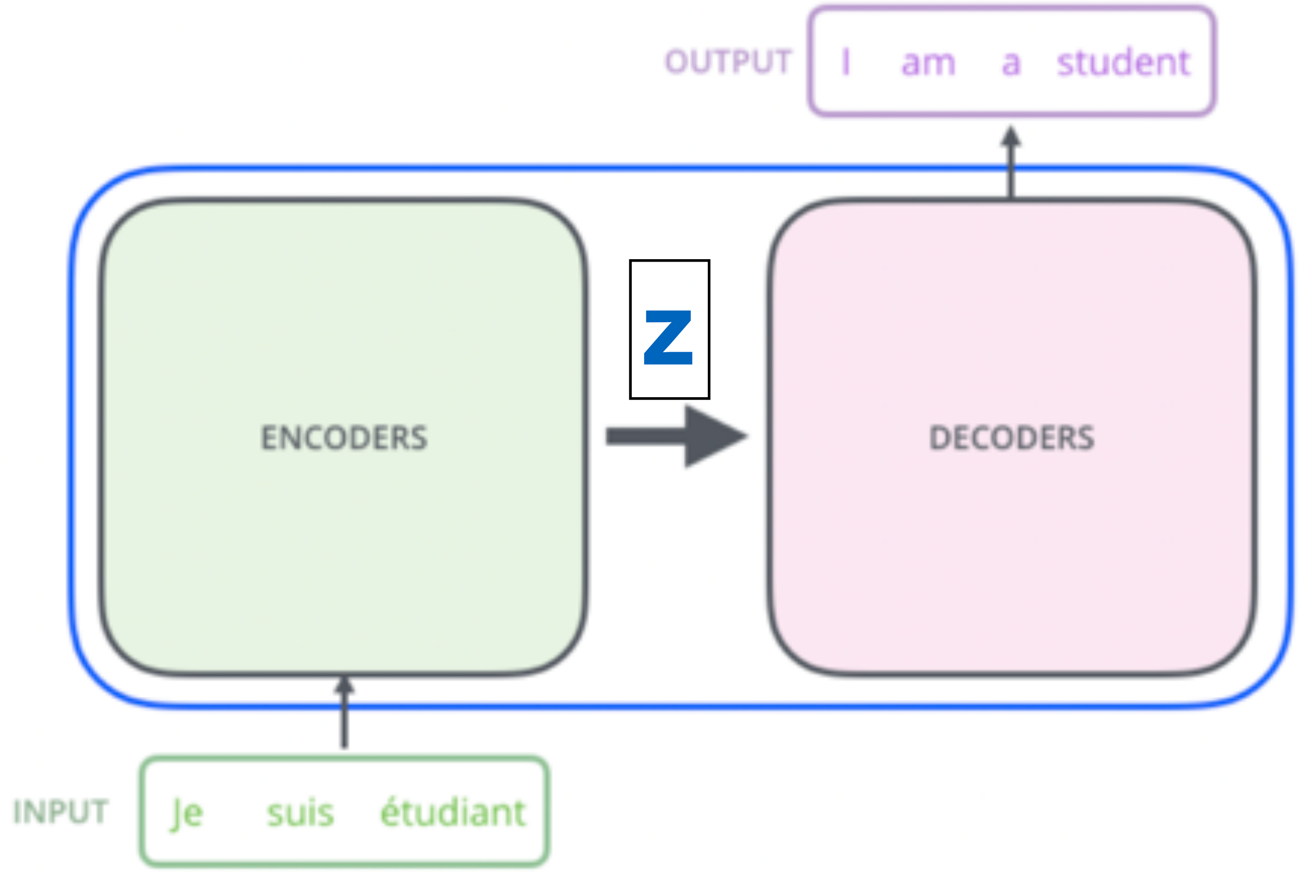
Motivation

- *The **cat** drank the milk because **it** was hungry.*
- *The cat drank the **milk** because **it** was sweet.*

Source: https://jalammar.github.io/illustrated-transformer/

Background: seq2seq models

- The encoder-decoder architecture
- **Encoder**: input sequence $X = (x_1, \ldots, x_n)$, get representation z
- **Decoder**: given z , generate output sequence $Y = (y_1, \ldots, y_m)$



Source: https://jalammar.github.io/illustrated-transformer/

Transformer Architecture

- **Encoder - Decoder**

N attention blocks

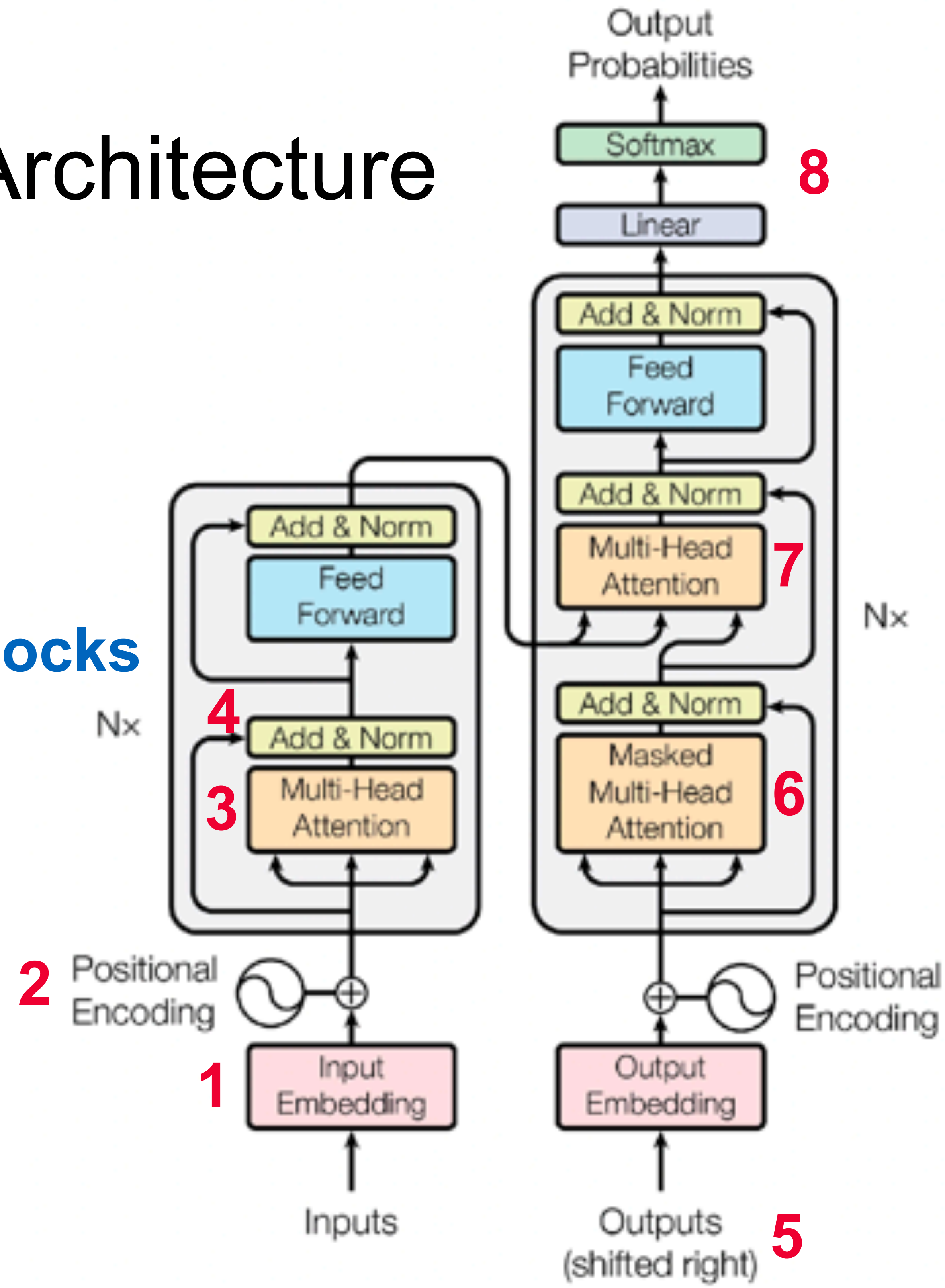


Figure 1: The Transformer - model architecture.

Transformer Architecture

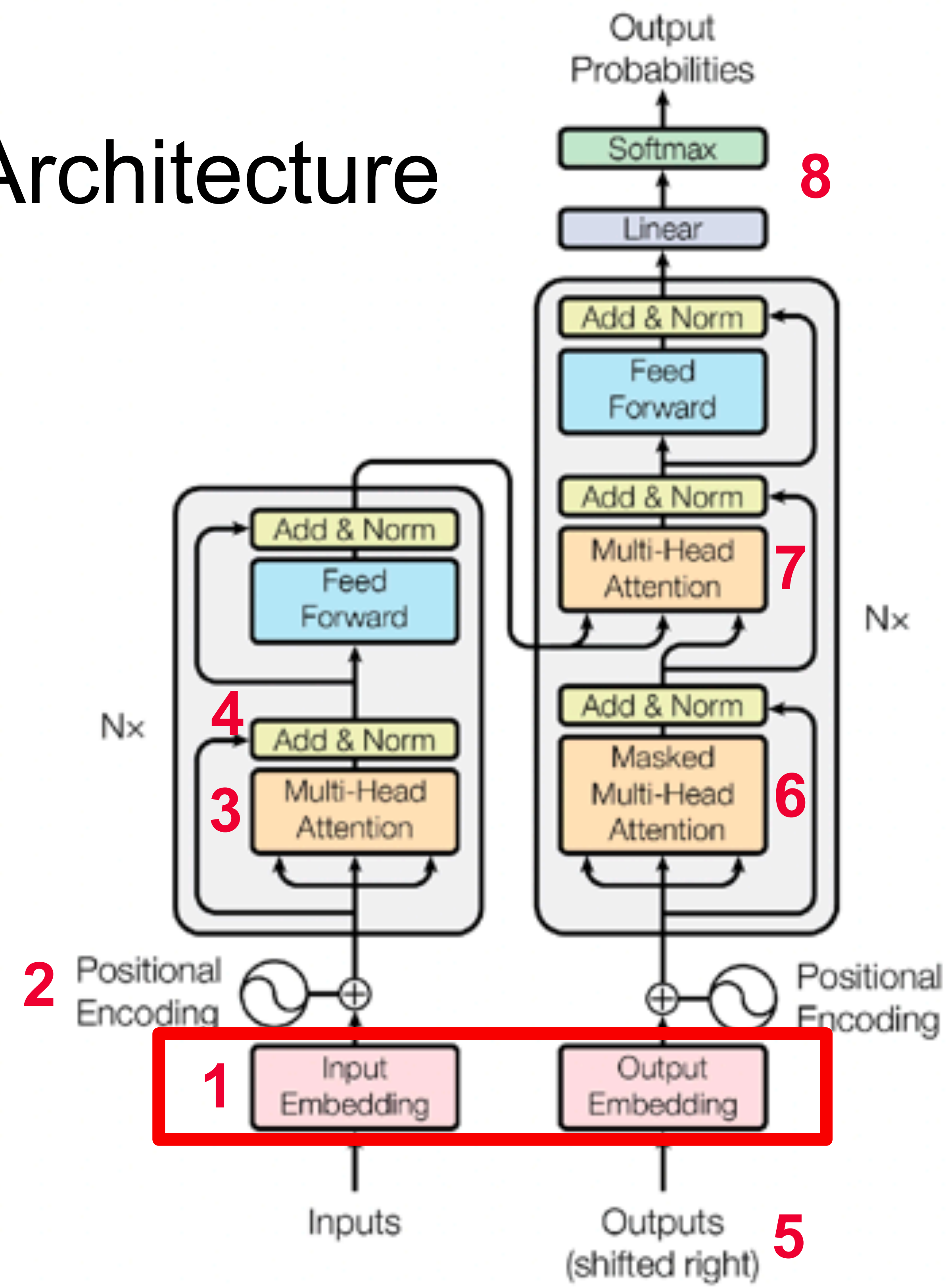
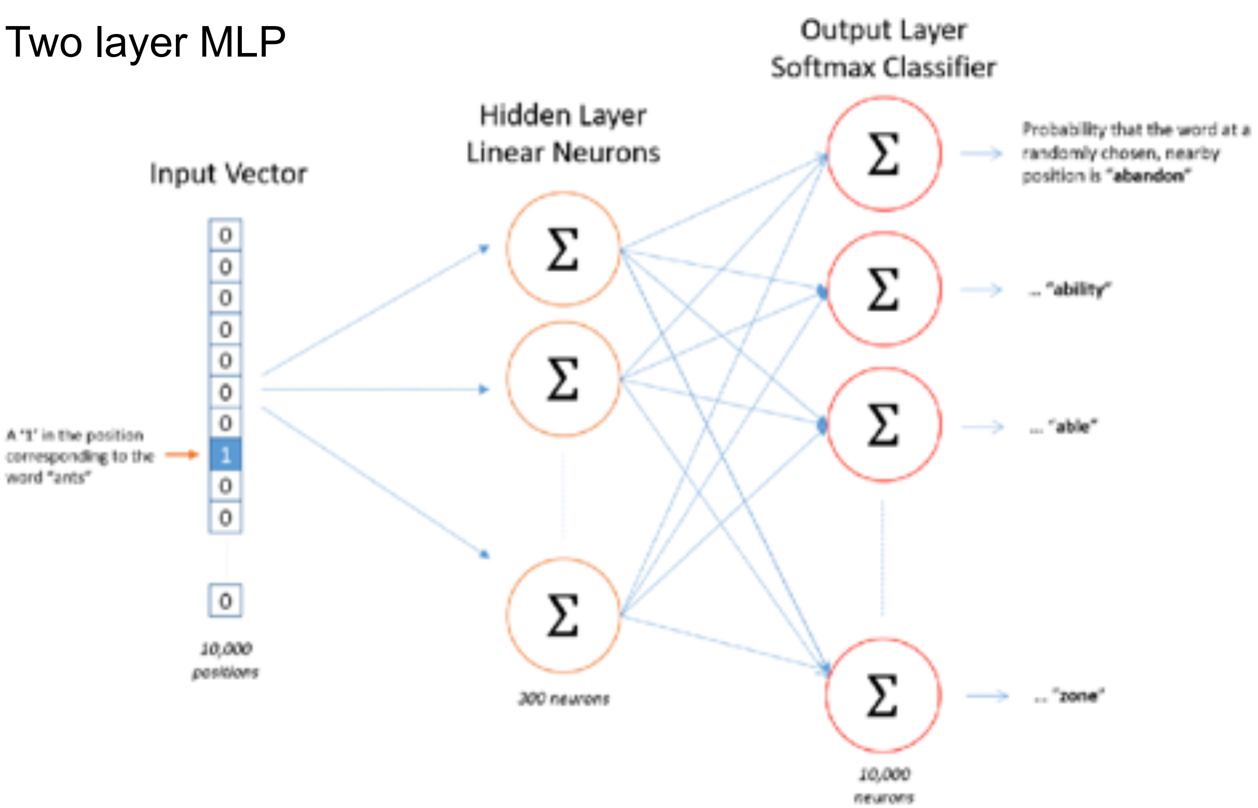


Figure 1: The Transformer - model architecture.

Embeddings

- Intuition: convert texts into vectors(embeddings) that can be handled by computer
- One-hot encoding fails: too sparse when having large vocabularies
- Learnable embeddings: Word2vec
  - CBOW, skip-gram

Two layer MLP

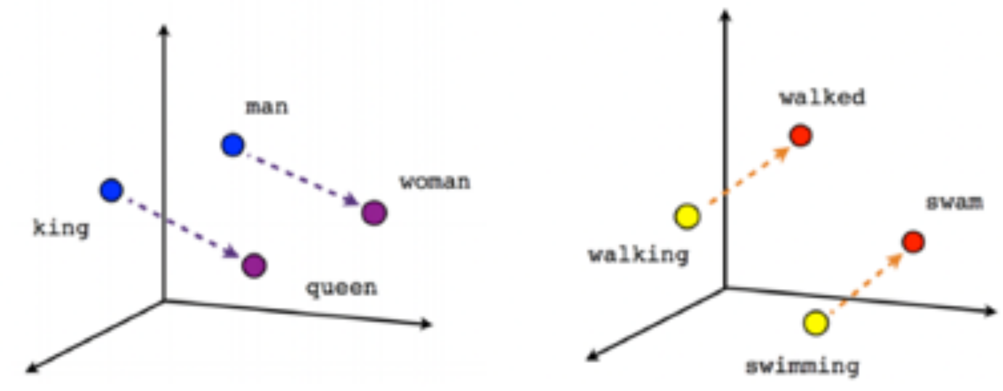


The embedding matrix

Word position / Word embedding

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

The more similar, the closer



Male-Female Verb tense

Source: *https://medium.com/deeper-learning/glossary-of-deep-learning-word-embedding-f90c3cec34ca*

Transformer Architecture

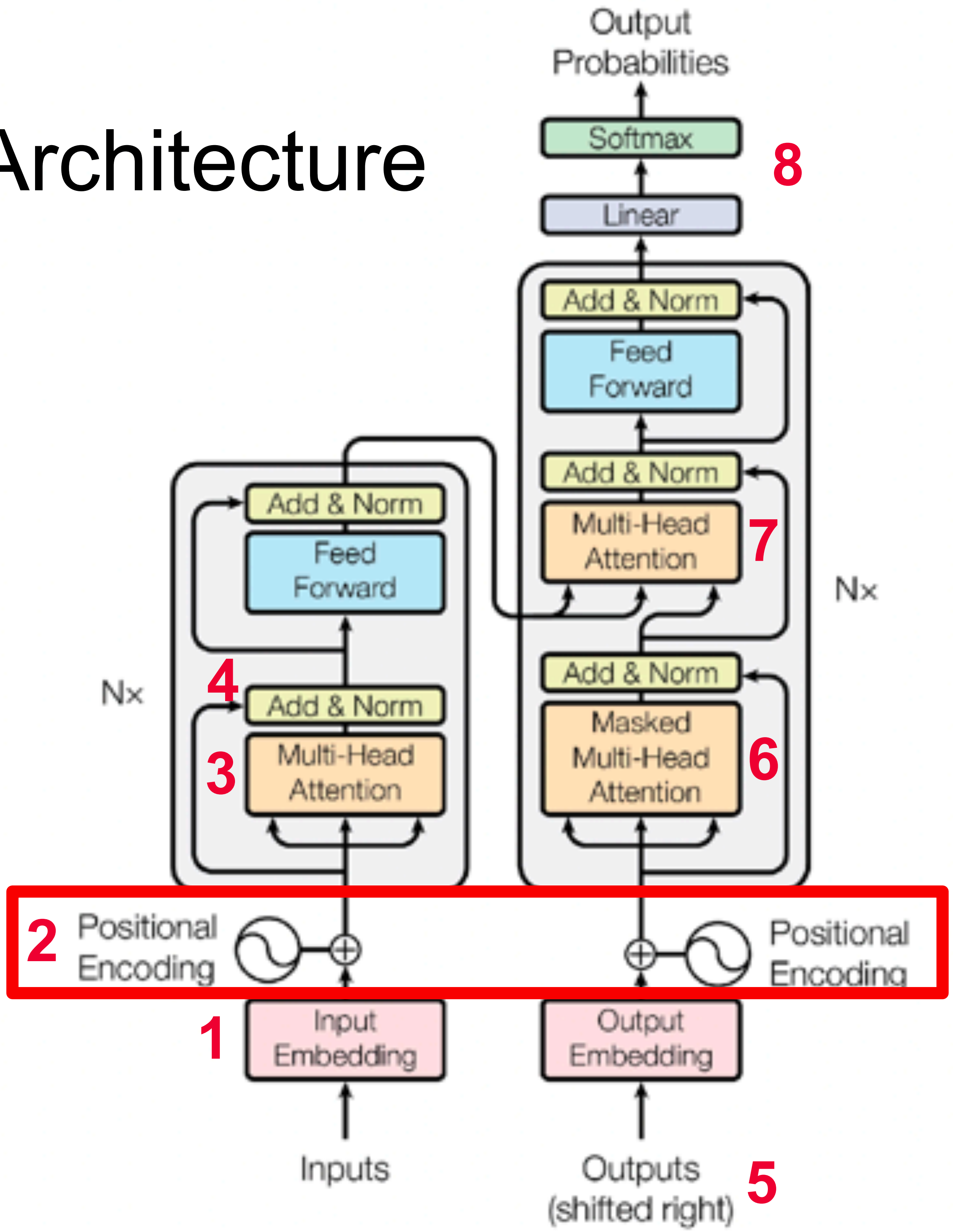


Figure 1: The Transformer - model architecture.

Positional Encodings

- **Order of words** matter in the sentence!!
- Same dimension as input embeddings, directly sum up with them

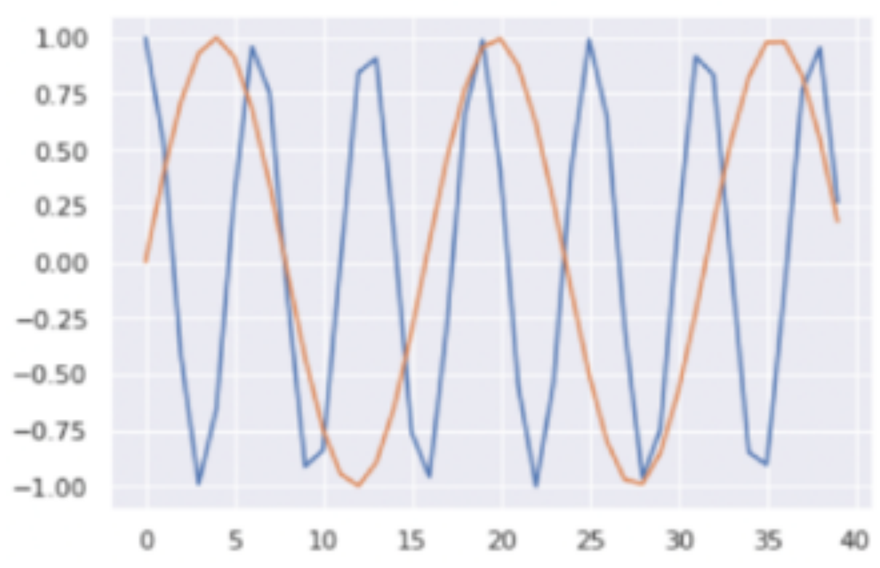
*“Since our model contains no recurrence and no convolution, in order for the model to make use of **the order of the sequence**, we must inject some information about the **relative or absolute position** of the tokens in the sequence.”*

Positional Encodings

- Sin and Cos functions:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

where *pos* is the position and *i* is the dimension. **Fixed values that depend only on the max length of the sequence.**



Orange: 0th word's encoding for a 40-word sequence.
Blue: 1st word's encoding for a 40-word sequence.

- **Also can be learnable (later in VisionTransformer…)**

Transformer Architecture

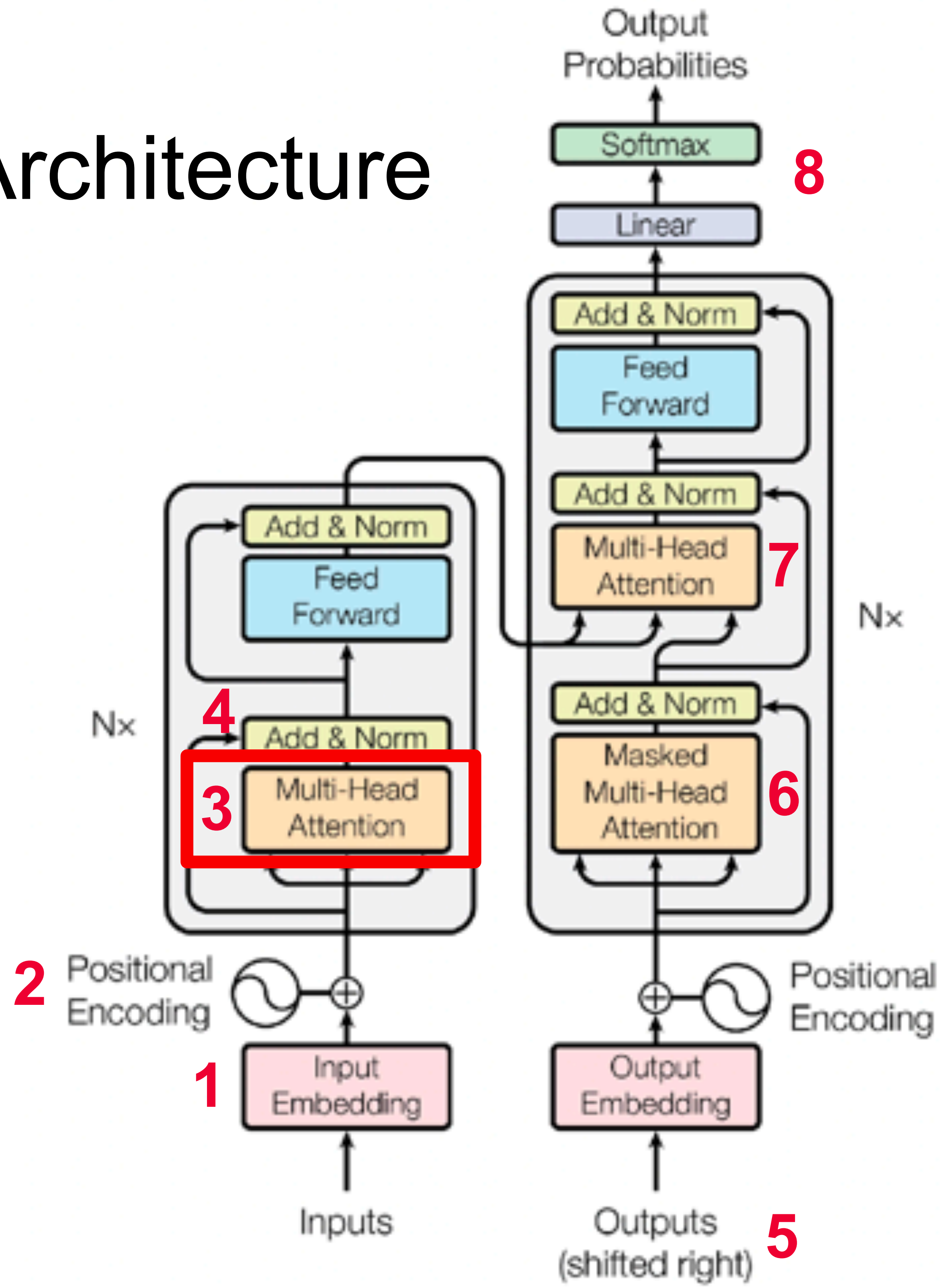


Figure 1: The Transformer - model architecture.

Self-Attention

- Intuition: to understand the meaning of the current word, we should look at other words in the sentence
  - *The **cat** drank the milk because **it** was hungry.*
  - *The cat drank the **milk** because **it** was sweet.*

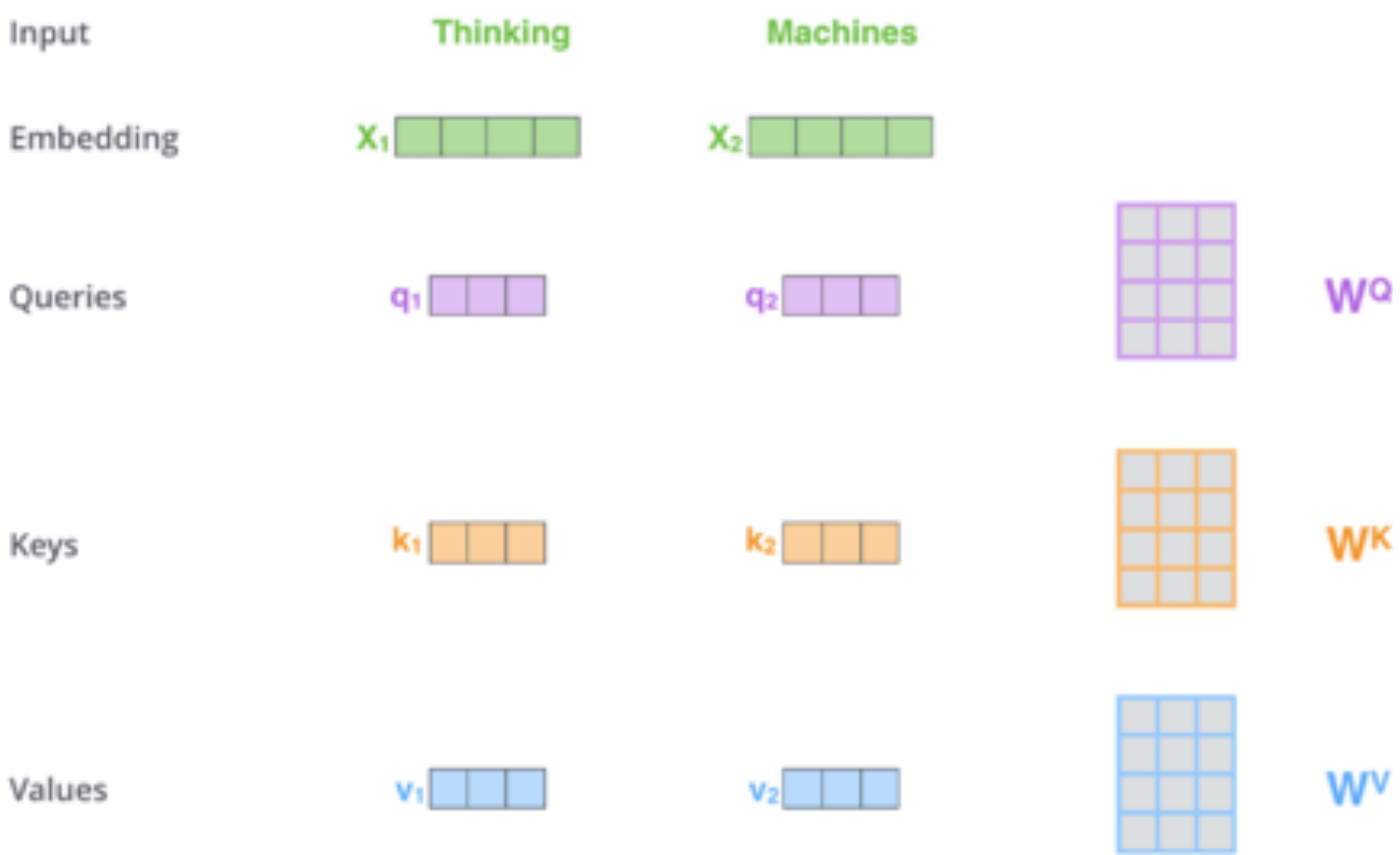
Source: https://jalammar.github.io/illustrated-transformer/

Self-Attention

- Initialize a **Query** W^Q**, a Key** W^K, and **a Value** W^V
- Matrix multiplication with word embeddings -> reduce dimensionality

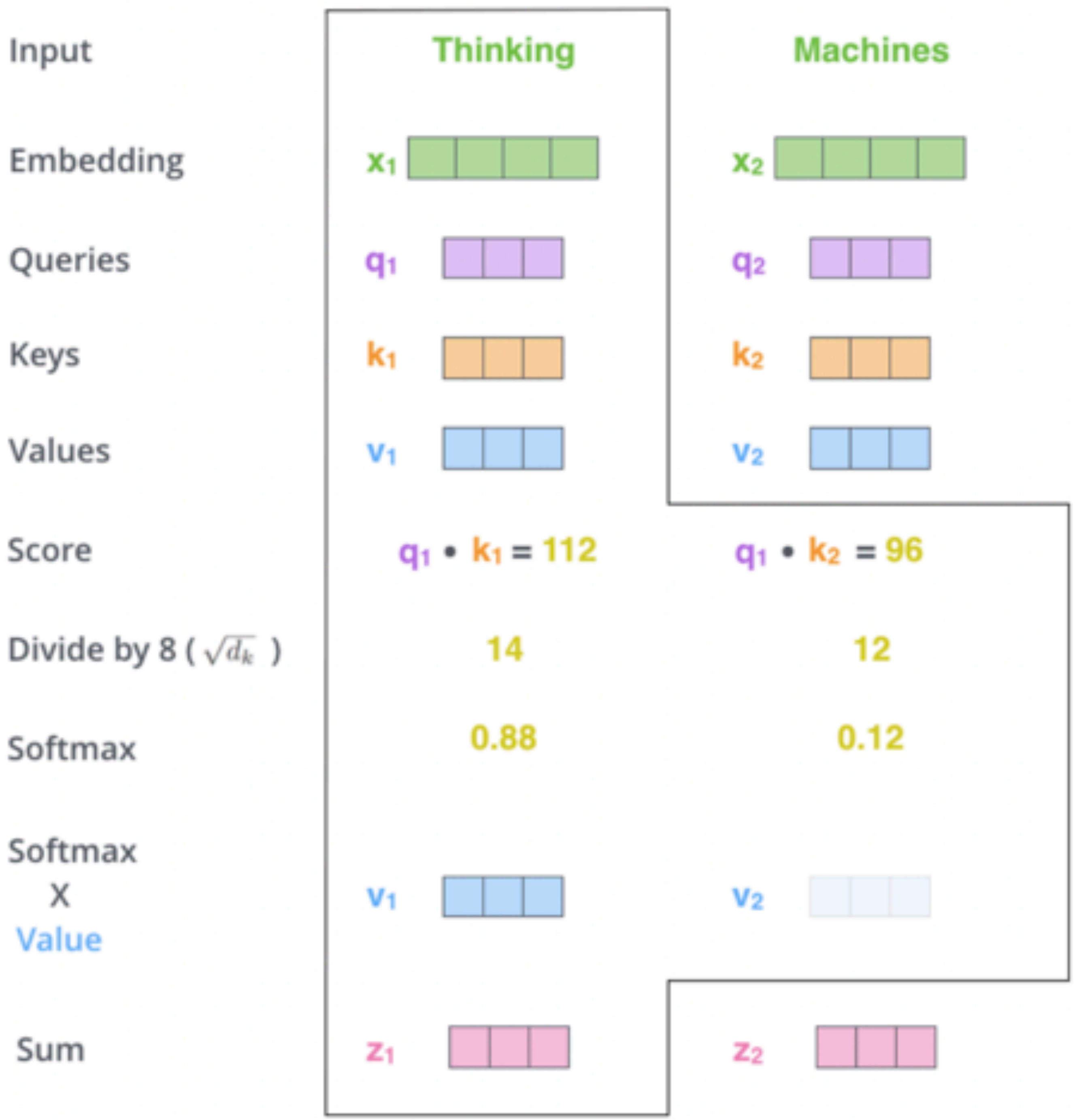
$\boxed{d_E = 512}$

$\boxed{d_k = 64}$



Source: https://jalammar.github.io/illustrated-transformer/

Self-Attention

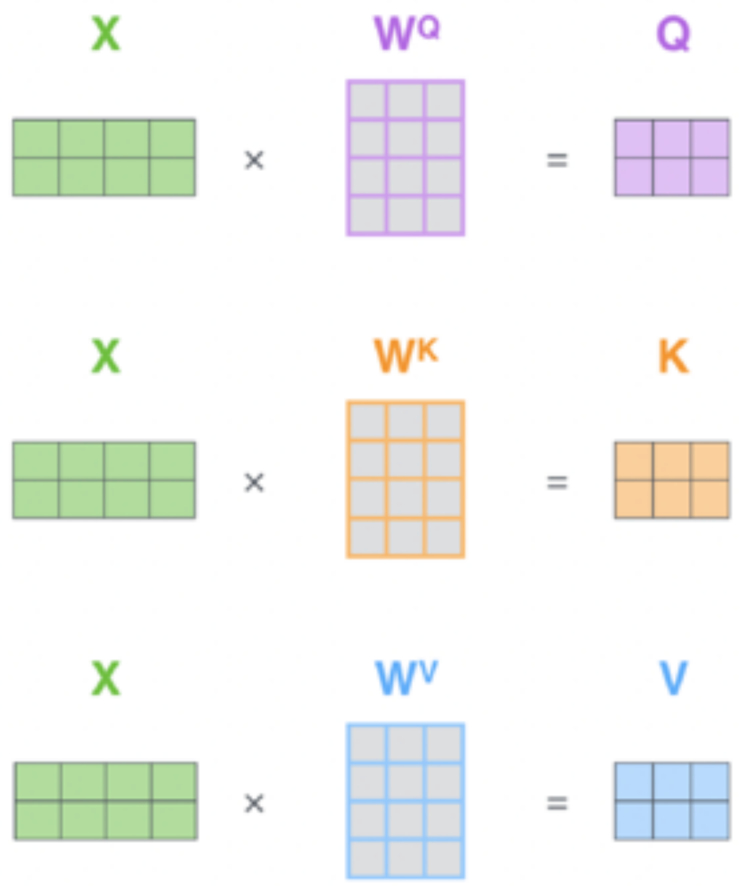


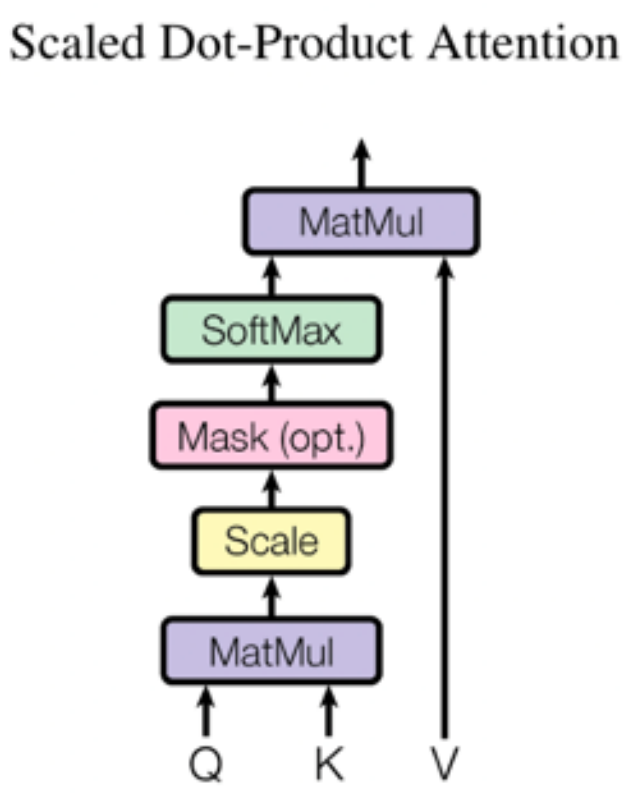
Source: *https://jalammar.github.io/illustrated-transformer/*

Self-Attention

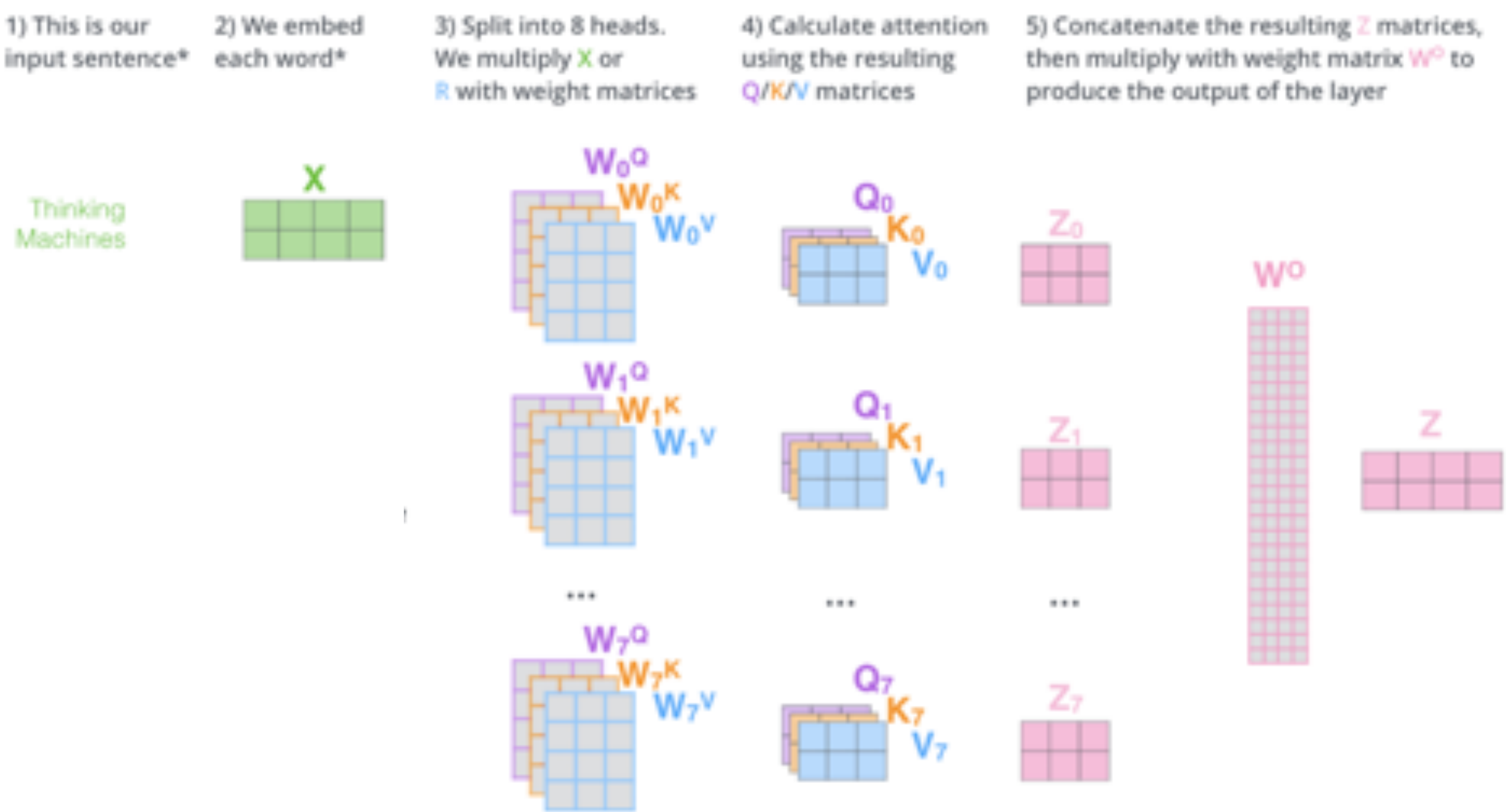
- General formula and matrix calculations:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$





Multi-Head Self-Attention



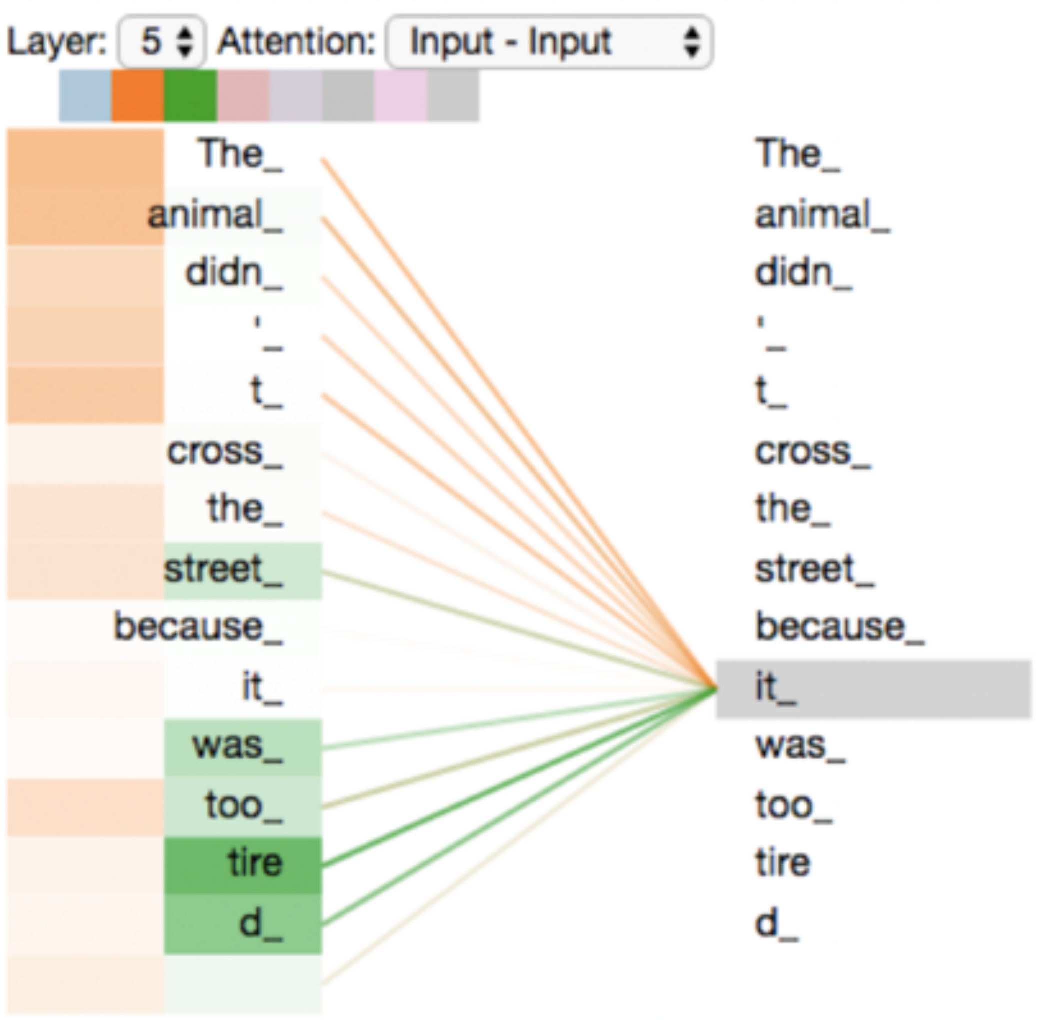
Source: https://jalammar.github.io/illustrated-transformer/

Multi-Head Self-Attention

- Intuition: different heads care for different aspects of the sentence.

*"The animal didn't cross the street because **it** was too tired."*

- Head0: the animal
- Head1: tired



Source: *https://jalammar.github.io/illustrated-transformer/*

Transformer architecture

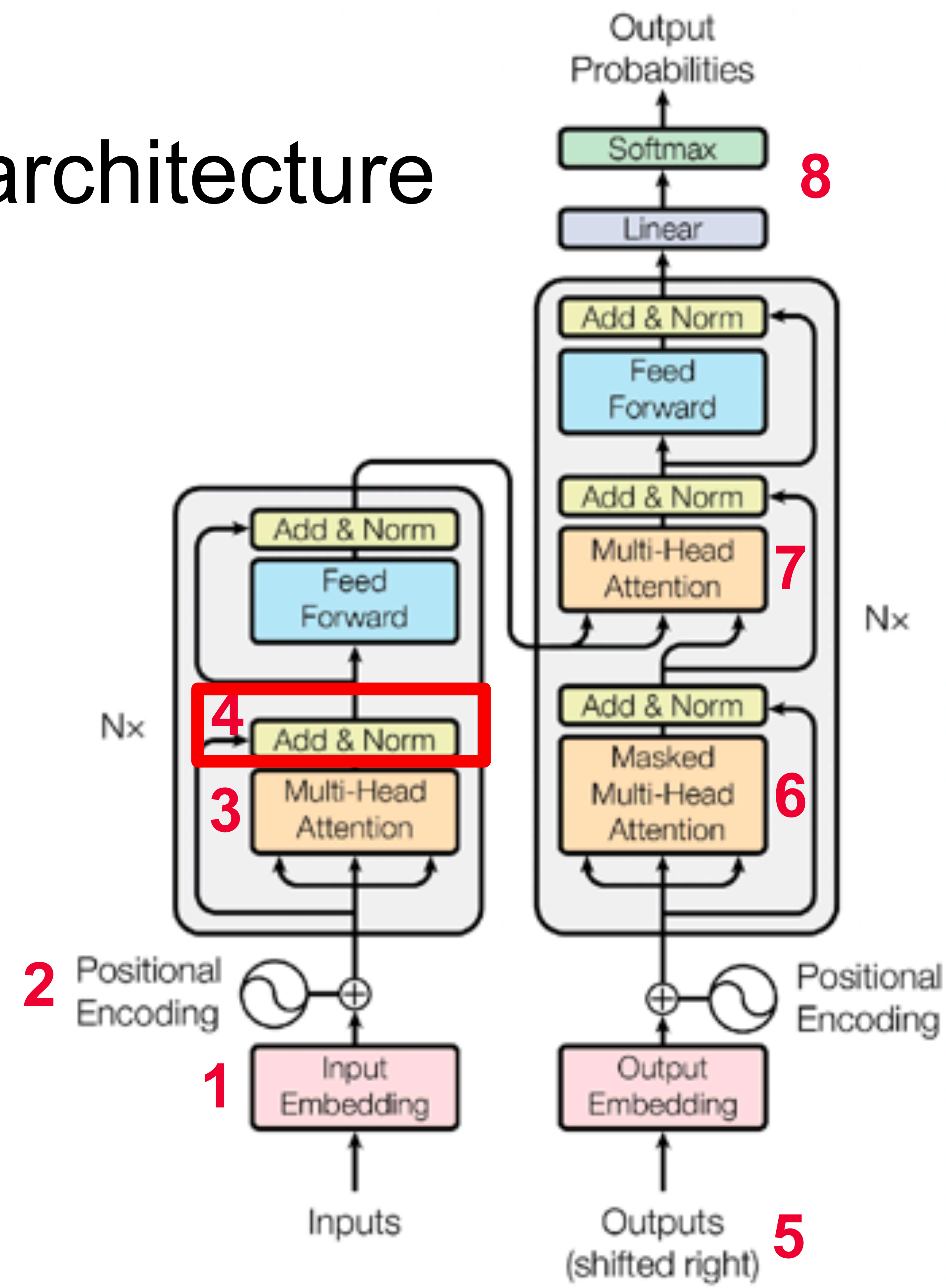
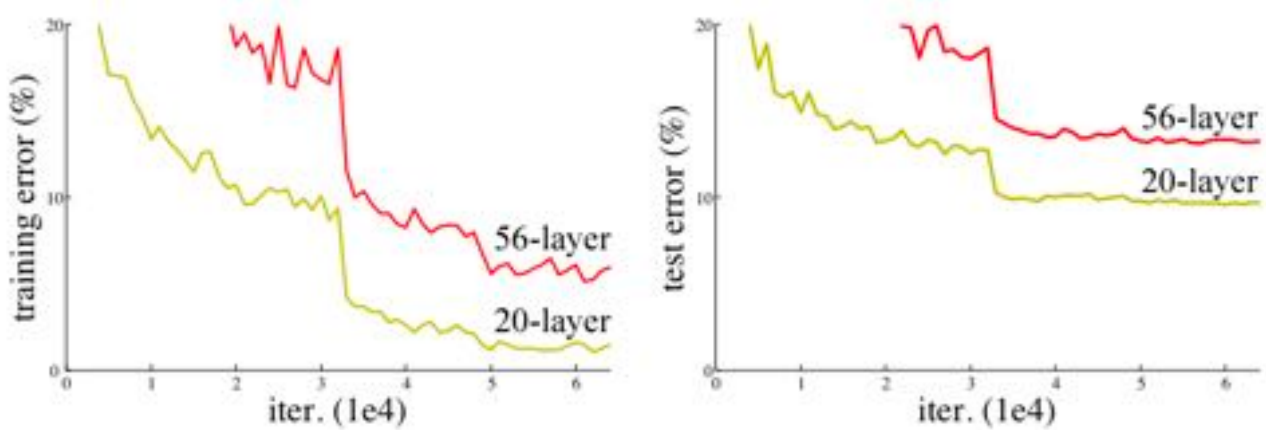
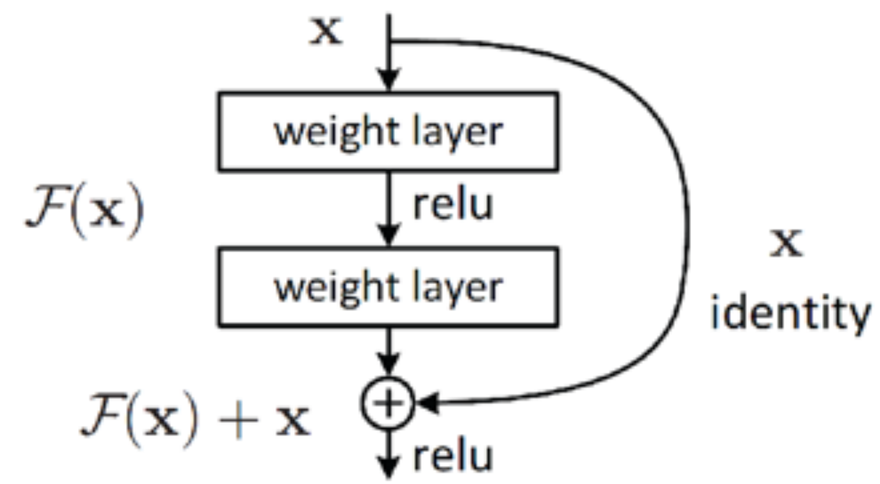


Figure 1: The Transformer - model architecture.

Residual Connections

- Network degradation problem

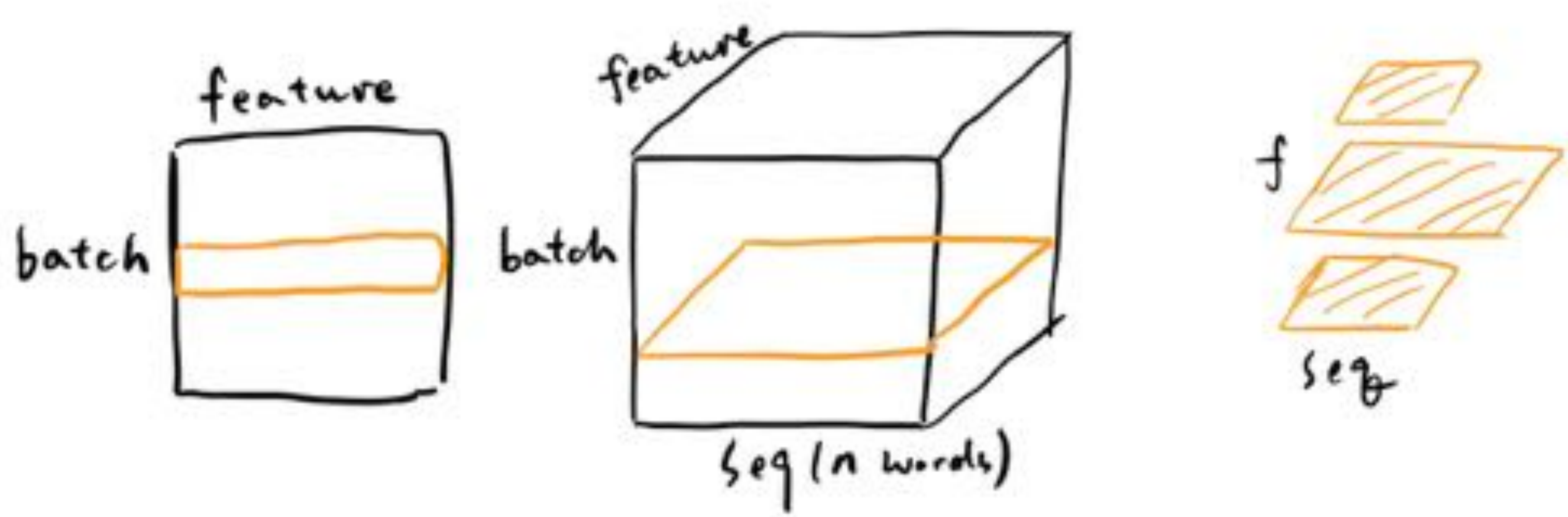




Source: He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Layer Normalization

- To get stable gradients and faster convergence
- Normalization(with learnable params) **over the layer(sequence)**
- Work well on **arbitrary** length of sequence



Transformer architecture

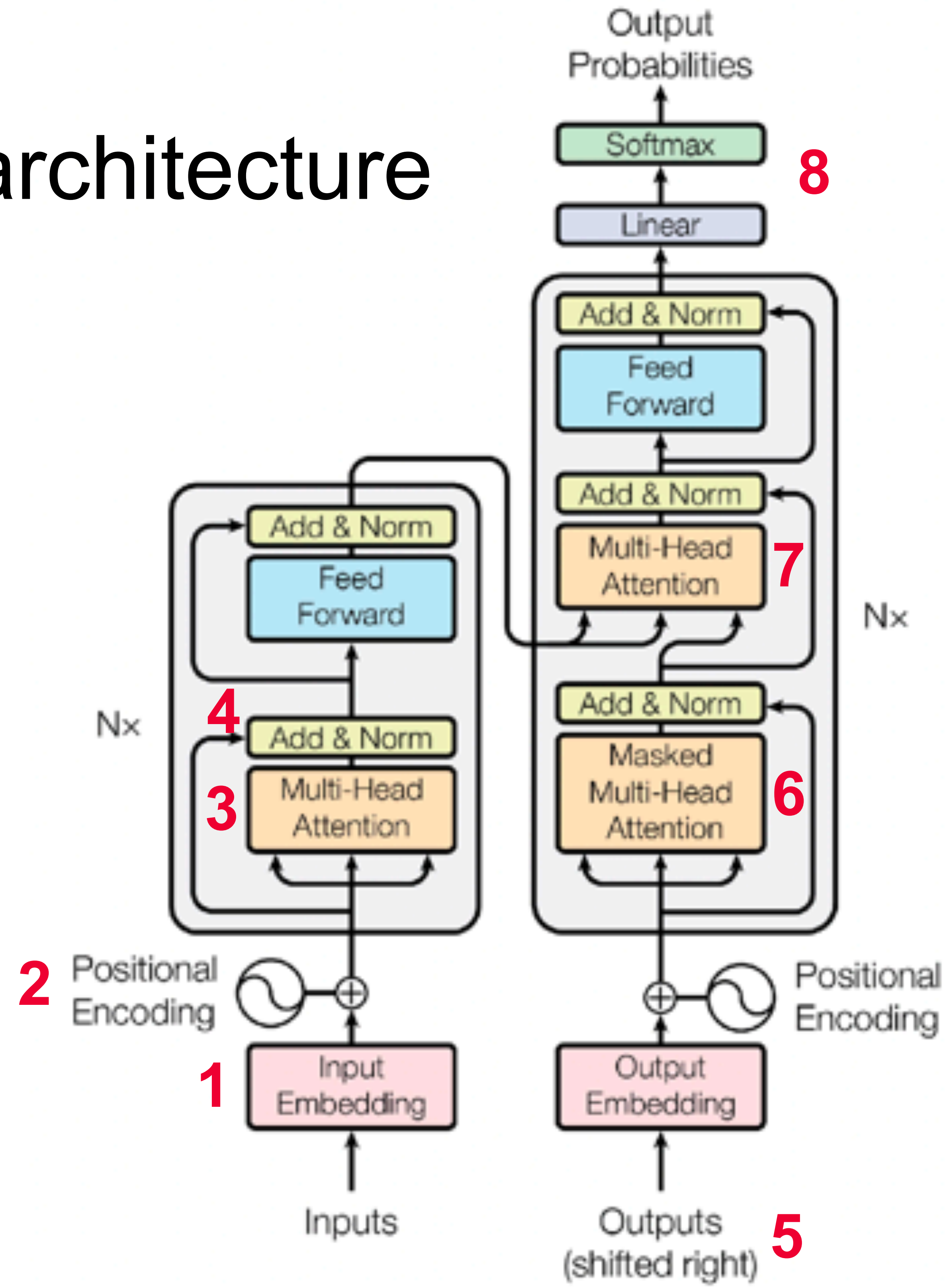


Figure 1: The Transformer - model architecture.

Task specific decoder design
See Backup for details

Demo

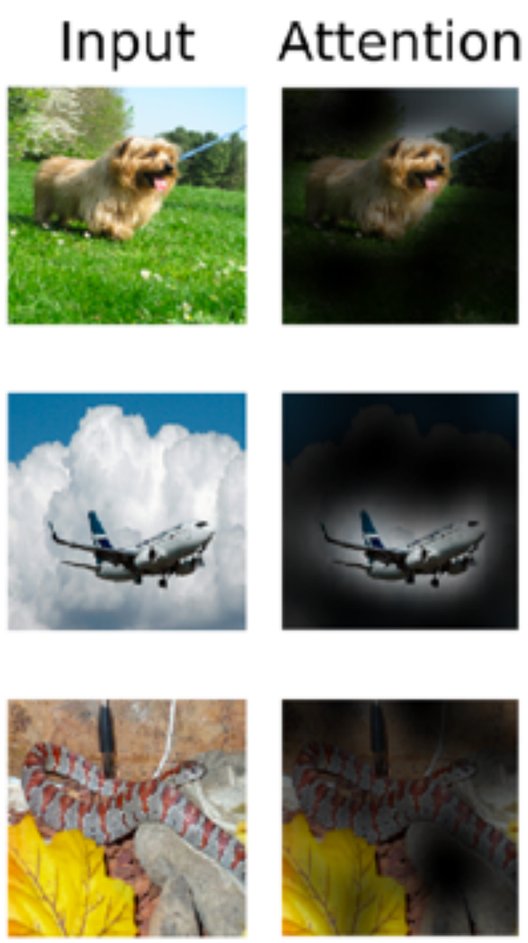
https://colab.research.google.com/drive/1hXIQ77A4TYS4y3UthWF-Ci7V7vVUoxmQ?usp=sharing

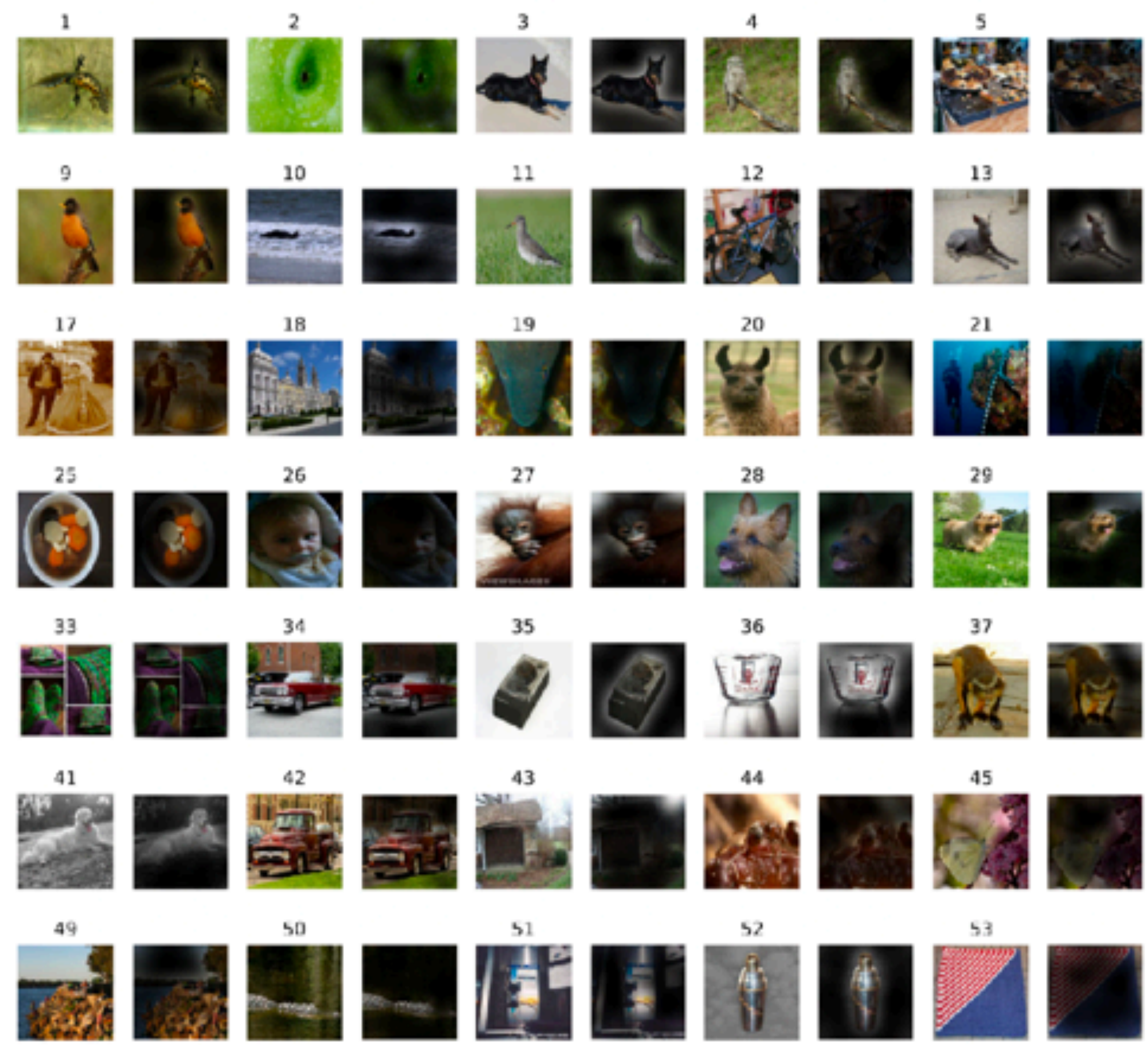
VisionTransformer

An Image is Worth 16x16 Words:
Transformers for Image Recognition at Scale

Motivation

- Can we directly apply Transformers for image classification?
- What's the network's focus when classifying images?

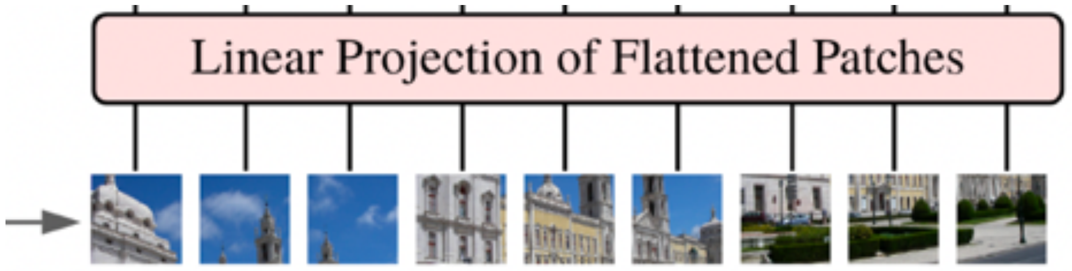




Divide Images into Patches, then Linear Project

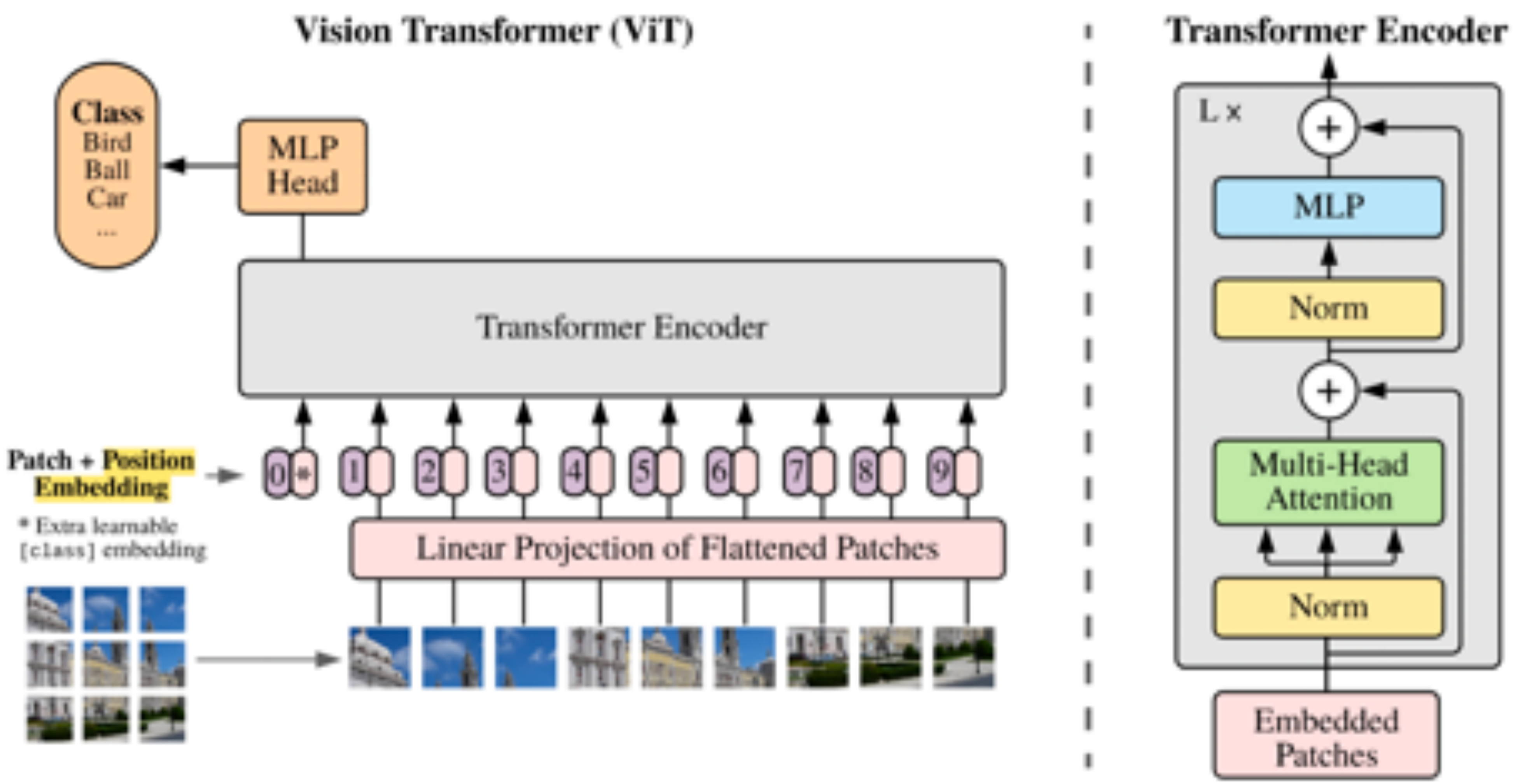
- Number of pixels(e.g. 224x224, 600x800…) are much larger than the length of text sequences

- *Given an image* $x \in \mathbb{R}^{H \times W \times C}$*, divide it by 16x16 patches to get N patches:* $x_p \in \mathbb{R}^{N \times (P^2 \times C)}$
- *Flatten each patch, map to D dimension with trainable linear projection*
  - *patch embeddings* $x_E \in \mathbb{R}^{N \times (1 \times D)}$



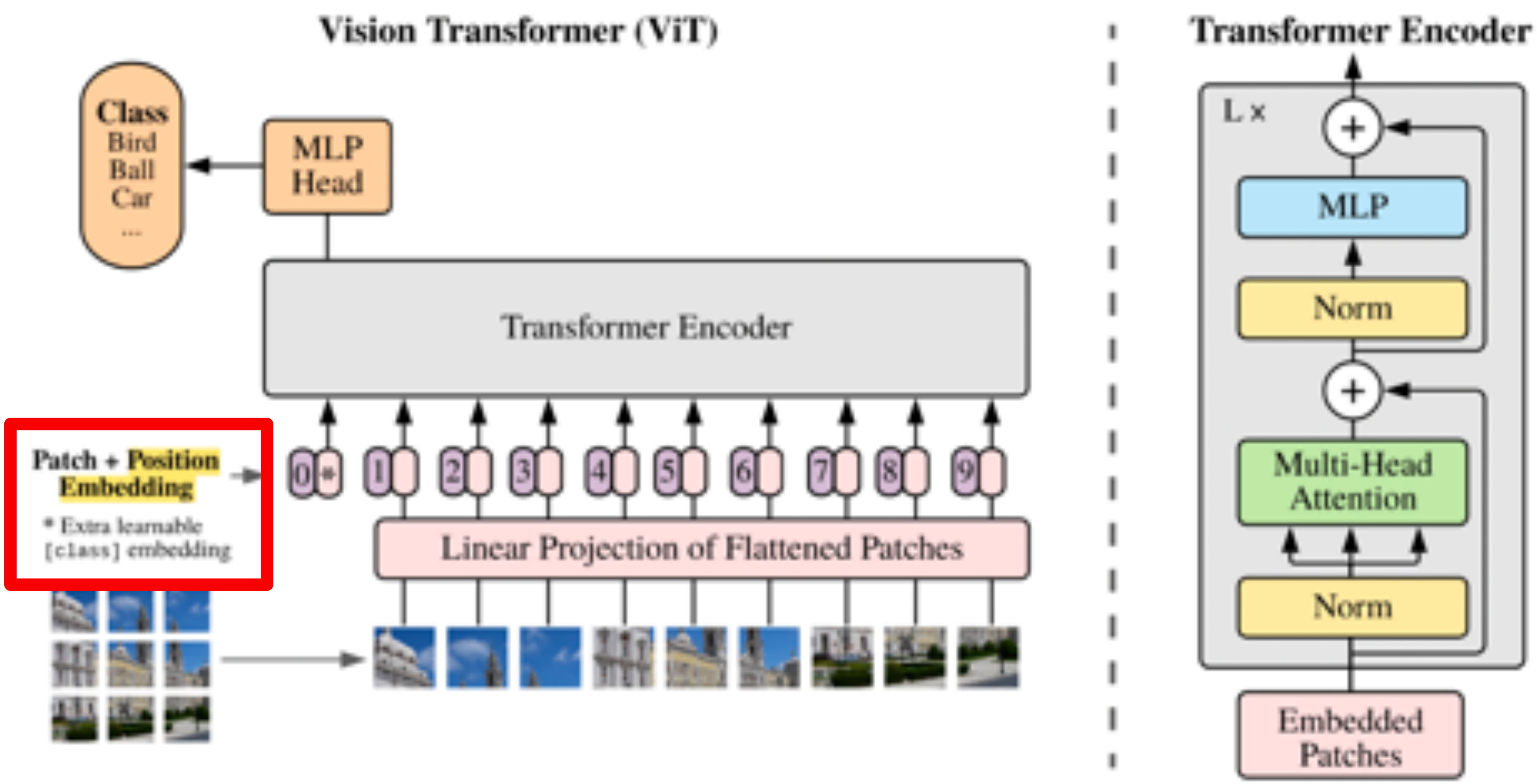
Model Architecture

- Only the Transformer Encoder are used, same design as the original Transformer
- Use patch to divide input images, then project into sequences. Use position embedding
- Task specific MLP head.



Model Architecture

- Only the Transformer Encoder are used, same design as the original Transformer
- Use patch to divide input images, then project into sequences. Use position embedding
- Task specific MLP head.



Position Embeddings (but learnable!)

Different strategies, same goal: to encode the position of each patch!

- **1-dimensional positional embedding(used):** inputs as a sequence of patches
- **2-dimensional positional embedding:** inputs as a grid of patches in two dimensions
- **Relative positional embeddings:** self-attention between patches for the relative distance

*"In patch-level inputs, the spatial dimensions are **much smaller** than the original pixel-level inputs, e.g., 14 × 14 as opposed to 224 × 224, and learning to represent the spatial relations in this resolution is **equally easy** for these different positional encoding strategies."*

The [CLS] token

- Self-attention is patch-wise. **BUT how to represent the whole image?**
  - Average all the embeddings
  - Attach an extra learnable [CLS] token, involved in all self-attention calculations

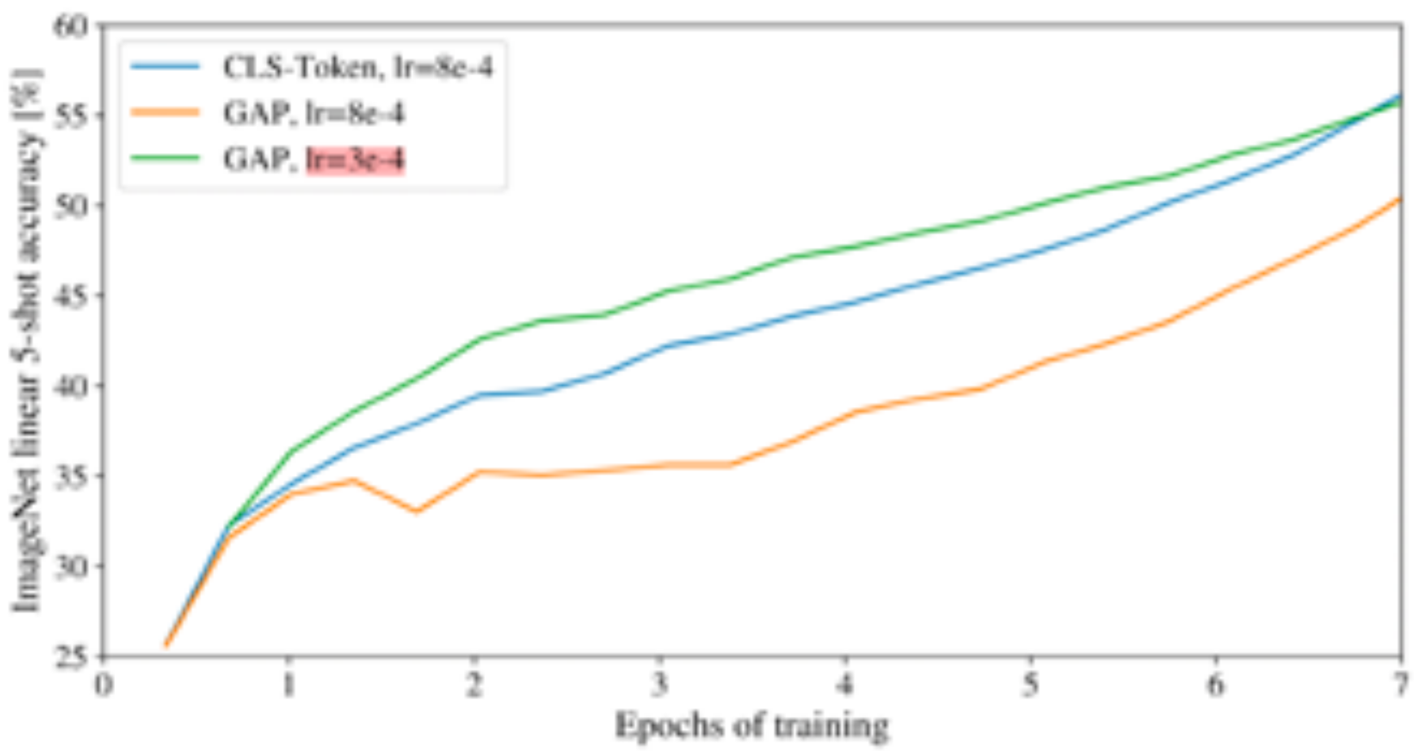


Figure 9: Comparison of class-token and global average pooling classifiers. Both work similarly well, but require different learning-rates.

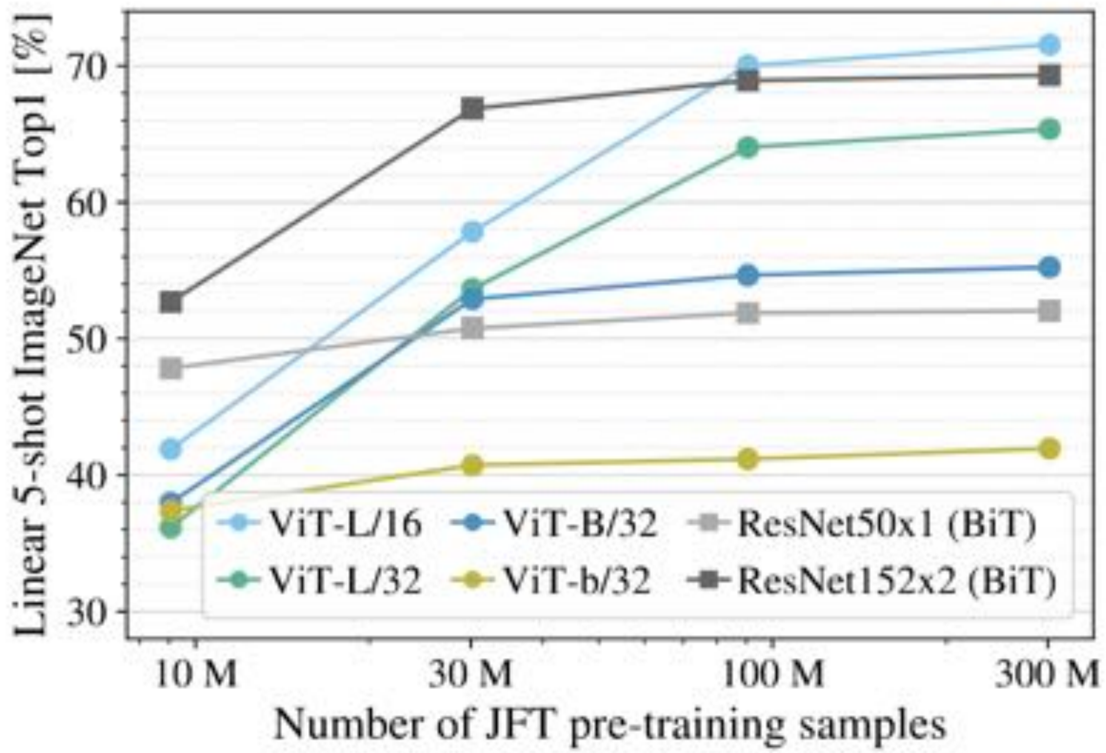
Demo

https://colab.research.google.com/drive/1rTz-uwb-nYXAQb6fJq1nWD43ahllwnFU?usp=sharing

Properties of VisionTransformers

VisionTransformers are data-hungry

- On mid-sized datasets (e.g. ImageNet), poorer performances than ResNets family
- On large datasets (e.g. ImageNet 21K, JFT-300M), state-of-the-art



ResNets perform better with smaller pre-training datasets,
but plateau sooner than ViT for larger datasets

Properties of VisionTransformers

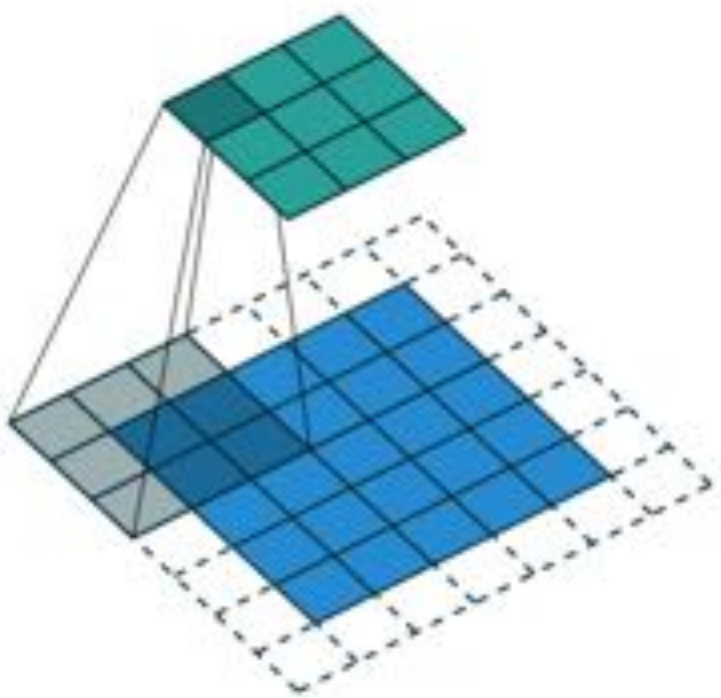
VisionTransformers lack inductive biases in CNNs

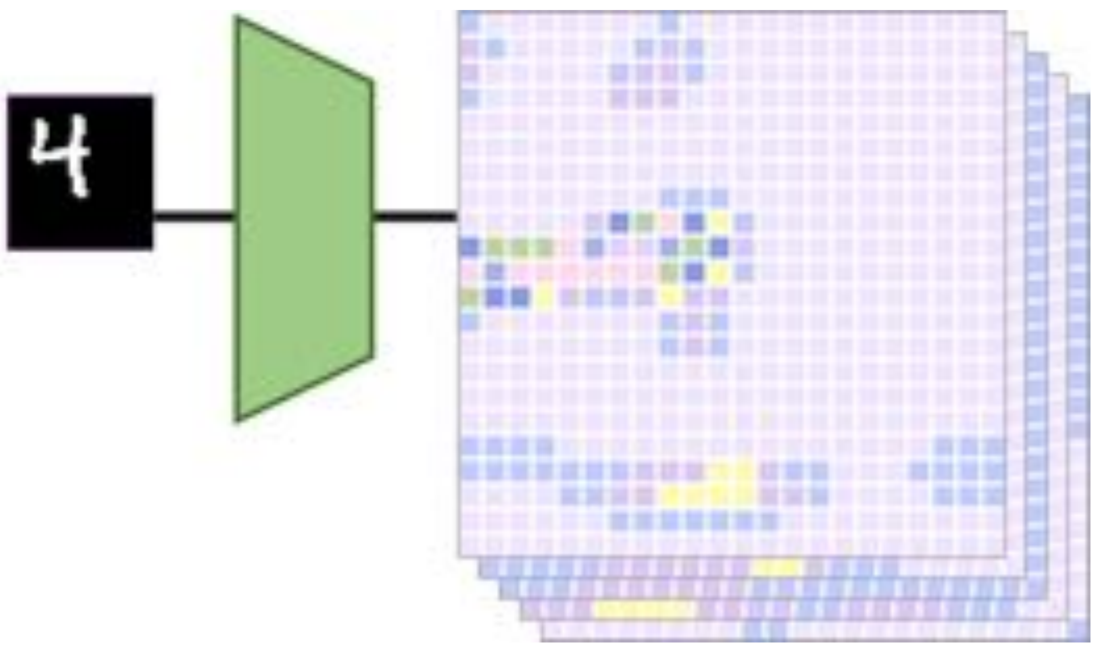
Inductive bias: maybe this is a cow

Source: *https://chriswolfvision.medium.com/what-is-translation-equivariance-and-why-do-we-use-convolutions-to-get-it-6f18139d4c59*

Properties of VisionTransformers

VisionTransformers lack inductive biases in CNNs





Locality

Assume neighbor contents have similar information

Translational equivalence

Translations in input also change the output

Source: *https://chriswolfvision.medium.com/what-is-translation-equivariance-and-why-do-we-use-convolutions-to-get-it-6f18139d4c59*

Properties of VisionTransformers

VisionTransformers can be easily scaled up

- Stack multiple attention layers
- Use the [CLS] token with task-specific heads

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

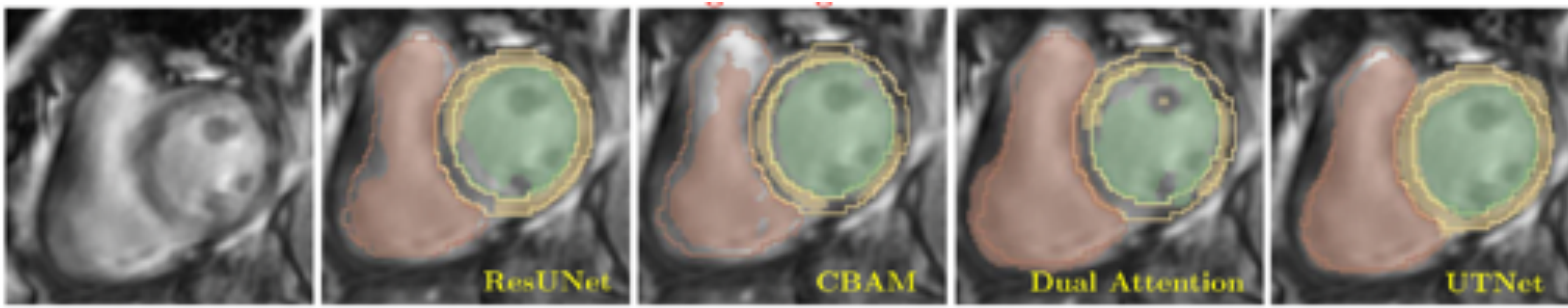
Table 1: Details of Vision Transformer model variants.

Applications

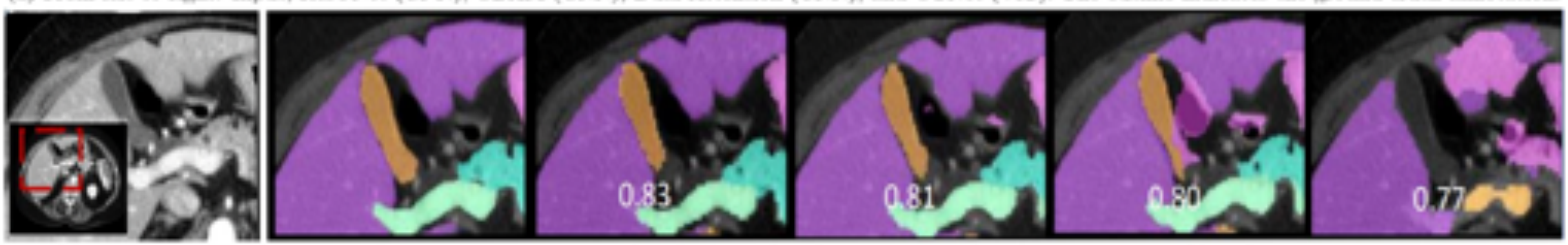
What can Transformers do for medical images?

[1] Shamshad, Fahad, et al. "Transformers in medical imaging: A survey." arXiv preprint arXiv:2201.09873 (2022).
[2] https://github.com/fahadshamshad/awesome-transformers-in-medical-imaging

Multi-organ Segmentation



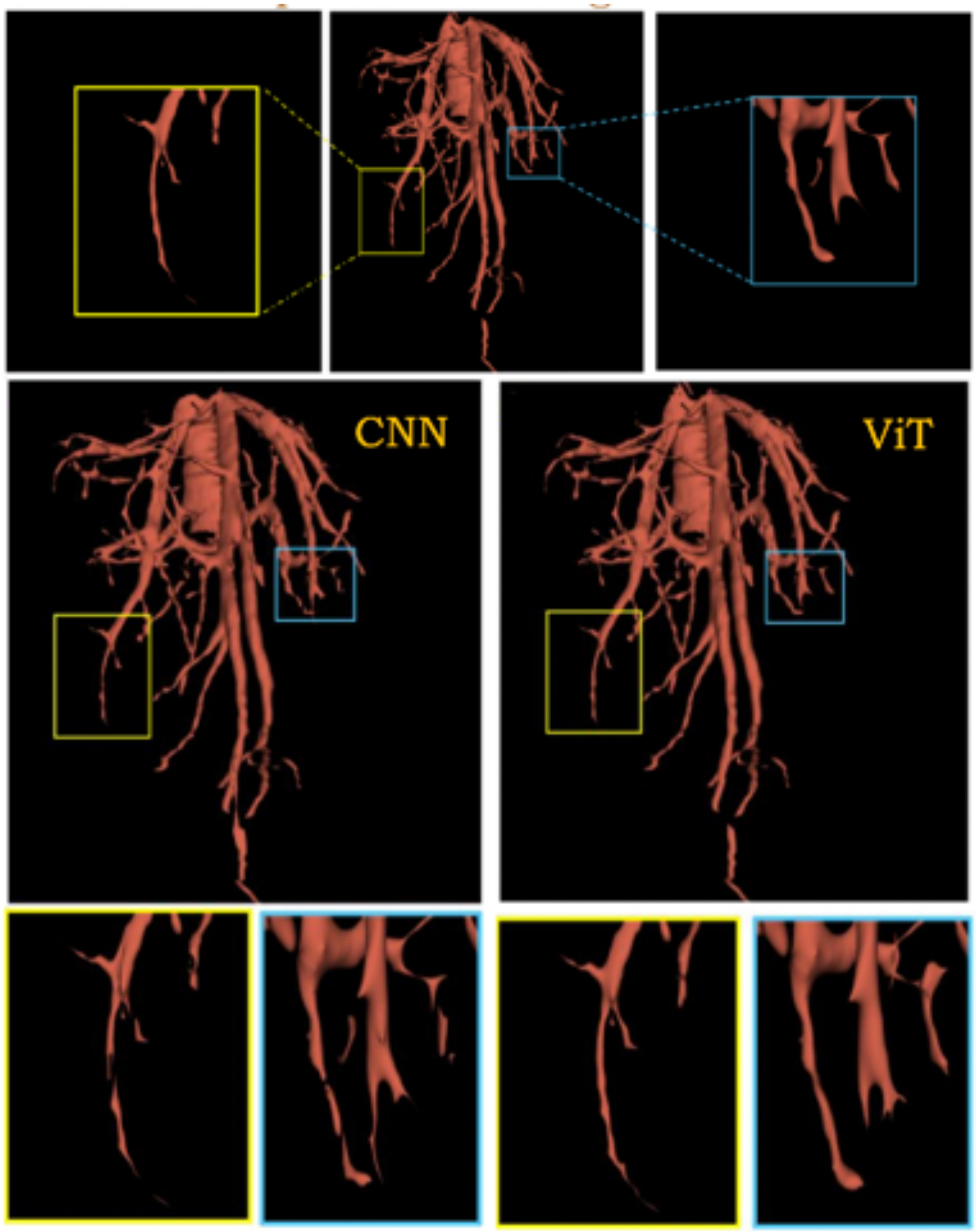
(a) From left to right: Input, ResUNet (CNN), CBAM (CNN), Dual Attention (CNN), and UTNet (ViT). The outline indicates the ground-truth annotation.



(b) From left to right: Input, Ground truth, UNETR (ViT), CoTr (ViT), TransUNet (ViT), and nnUNet (CNN). Numbers are Dice scores.

[a] Yunhe Gao, Mu Zhou, and Dimitris N Metaxas. Utnet: a hybrid transformer architecture for medical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 61–71. Springer, 2021.
[b] Ali Hatamizadeh, Dong Yang, Holger Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. arXiv preprint arXiv:2103.10504, 2021.

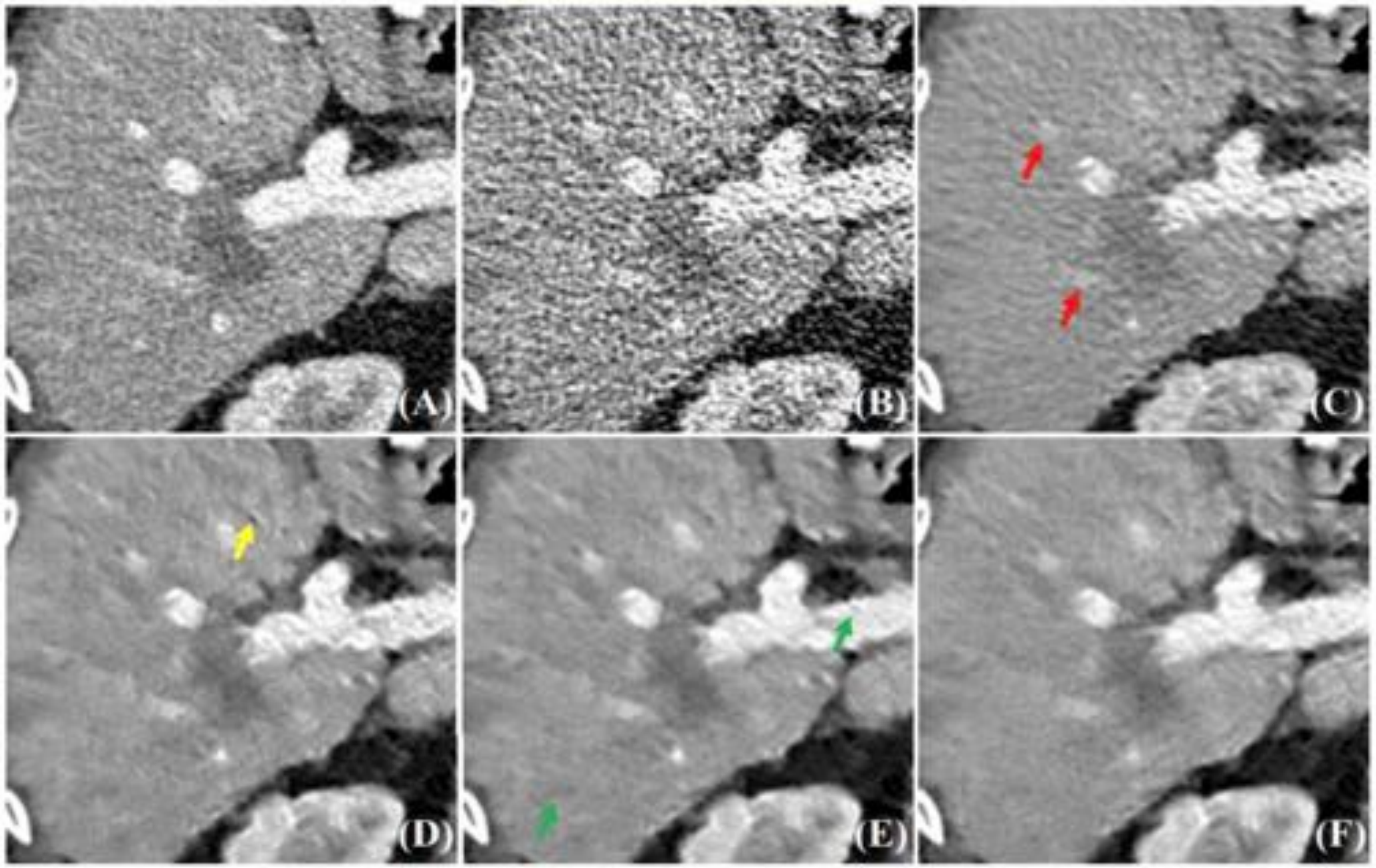
Vessel Segmentation



(c) First row: Ground truth. Second row (left) CNN based approach. Second row (right) ViT based approach.

[c] Mian Wu, Yinling Qian, Xiangyun Liao, Qiong Wang, and Pheng- Ann Heng. Hepatic vessel segmentation based on 3dswin- transformer with inductive biased multi-head self-attention. arXiv preprint arXiv:2111.03368, 2021.

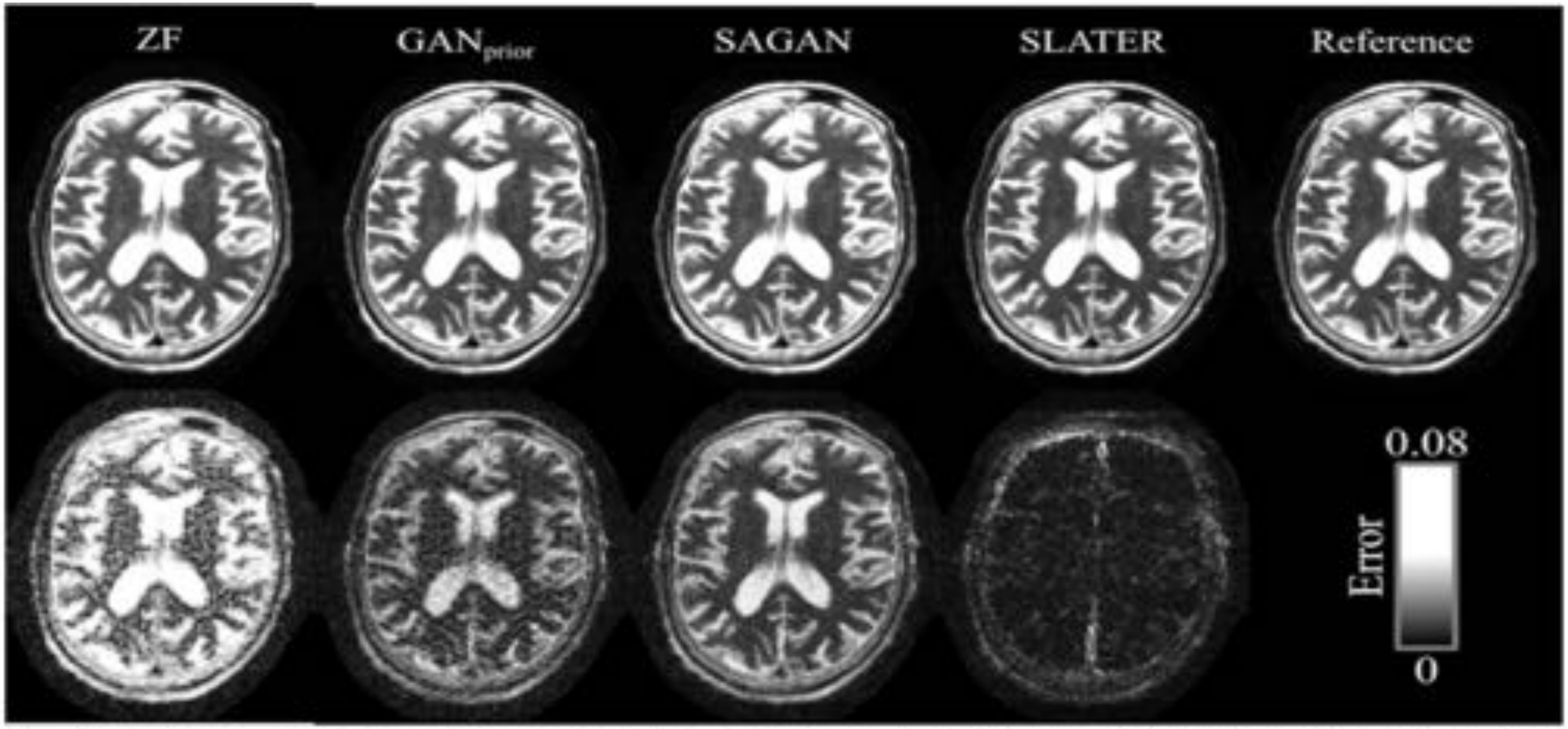
Low-Dose Image Enhancement



d) Top to bottom and left to right: Normal Dose CT, Low Dose CT, Non-Local Mean (hand-crafted), RED-CNN (CNN), MAP-NN (CNN), and TransCT (ViT)

[d] Zhicheng Zhang, Lequan Yu, Xiaokun Liang, Wei Zhao, and Lei Xing. Transct: Dual-path transformer for low dose computed tomography. arXiv preprint arXiv:2103.00634, 2021.

Image Reconstruction



(e) From left to right (top row): Fourier method, GAN (CNN), SAGAN (CNN), and SLATTER (ViT). Bottom row shows corresponding error maps.

[e] YilmazKorkmaz,SalmanUHDar,MahmutYurt,MuzafferO¨zbey, and Tolga C¸ ukur. Unsupervised mri reconstruction via zero-shot learned adversarial transformers. arXiv preprint arXiv:2105.08059, 2021.

Thanks for your ATTENTION :)

Questions?

Backup

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

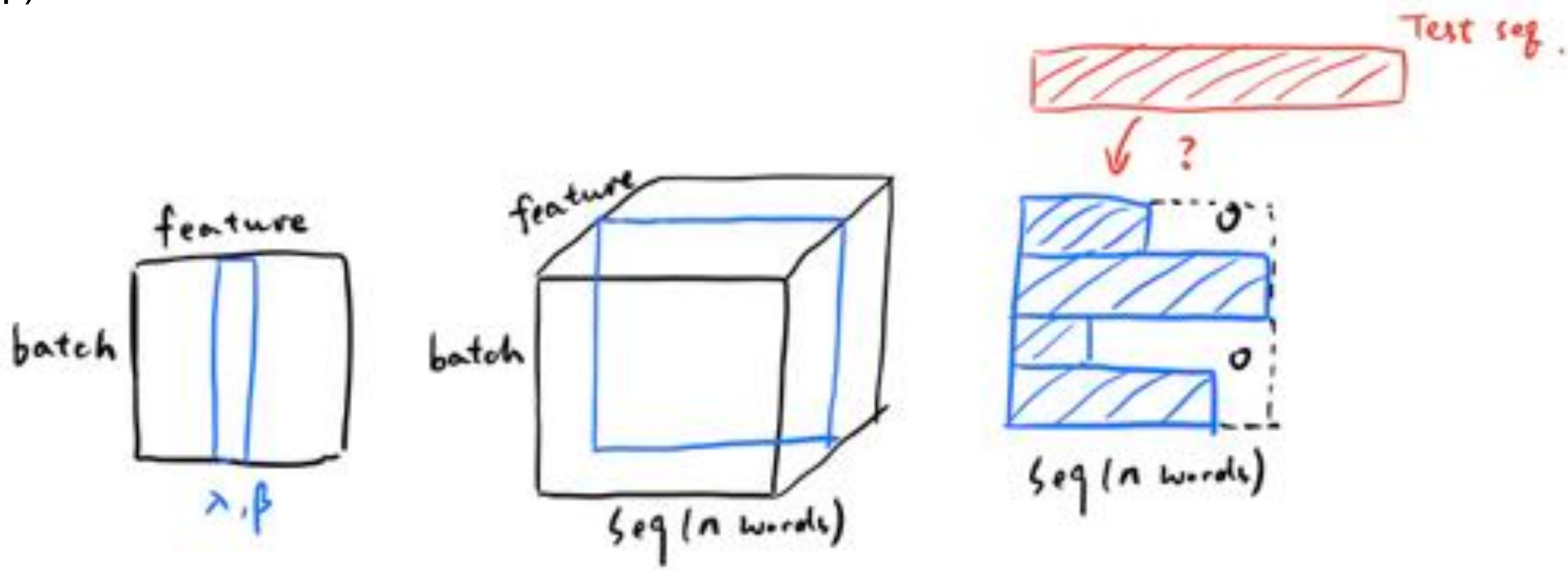
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Batch Normalization

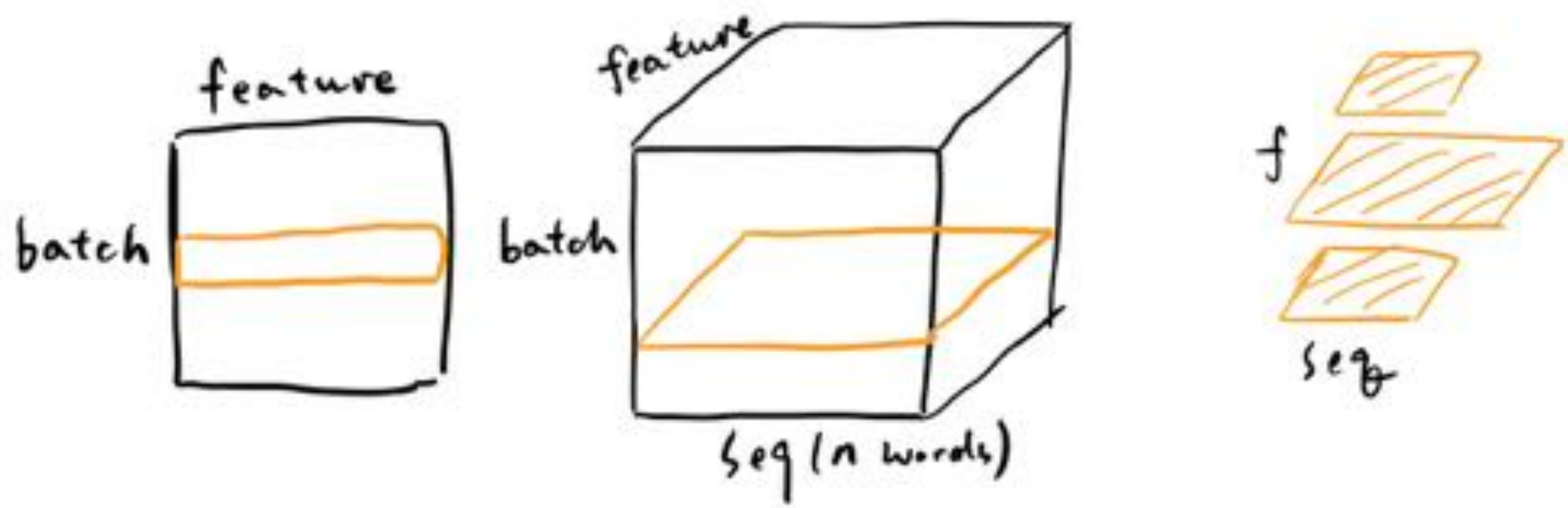
- To get stable gradients and faster convergence
- Normalization(with learnable params) **over the batch**
- **Problem:**
  - Sequence lengths vary a lot in mini-batches, causing instability
  - Fail to deal with **unseen** length of input sequence (e.g. testing seq. much longer than training seq.)



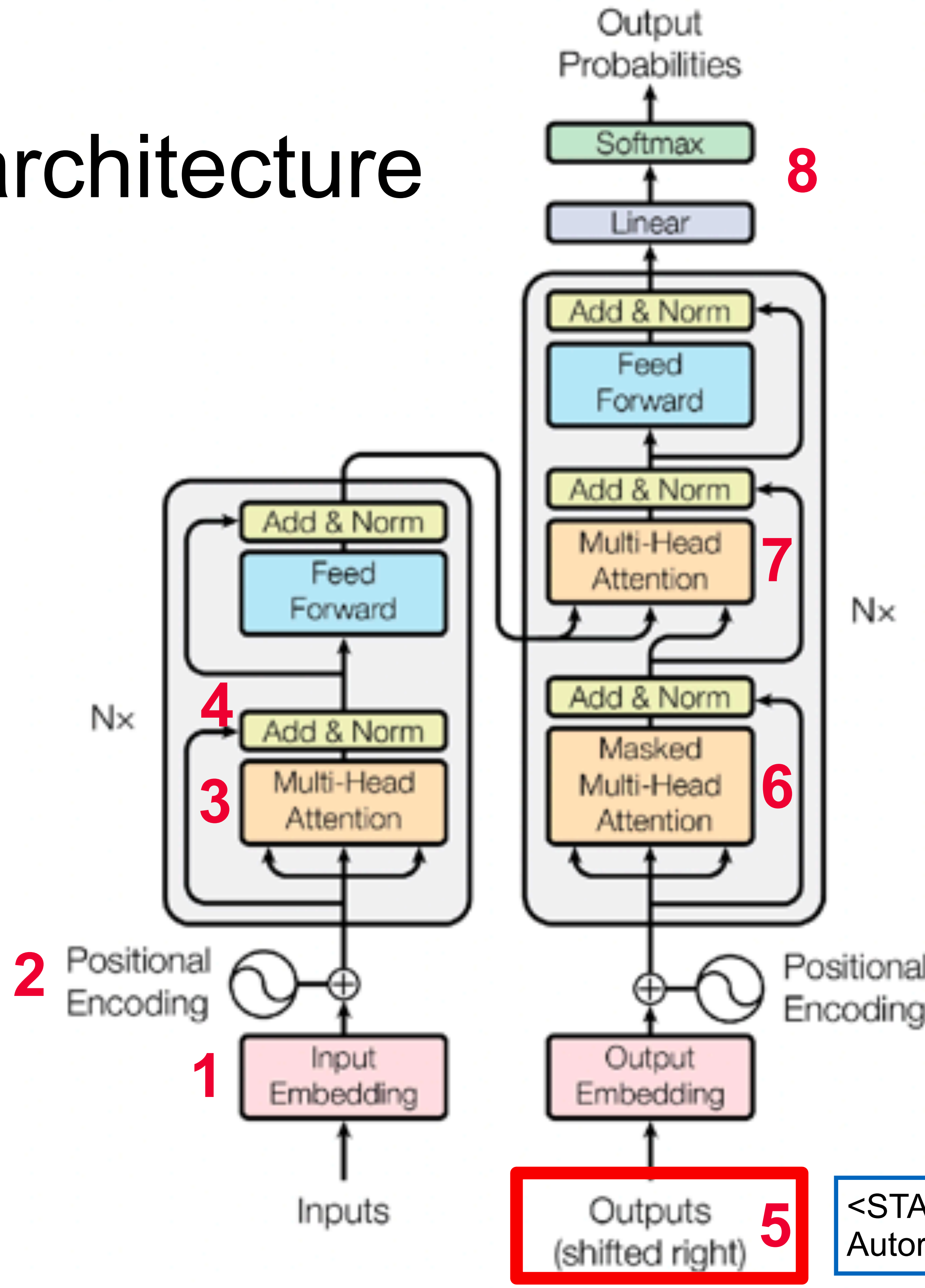
Source: Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." International conference on machine learning. PMLR, 2015.

Layer Normalization

- To get stable gradients and faster convergence
- Normalization(with learnable params) **over the layer(sequence)**
- Work well on **arbitrary** length of sequence



Transformer architecture

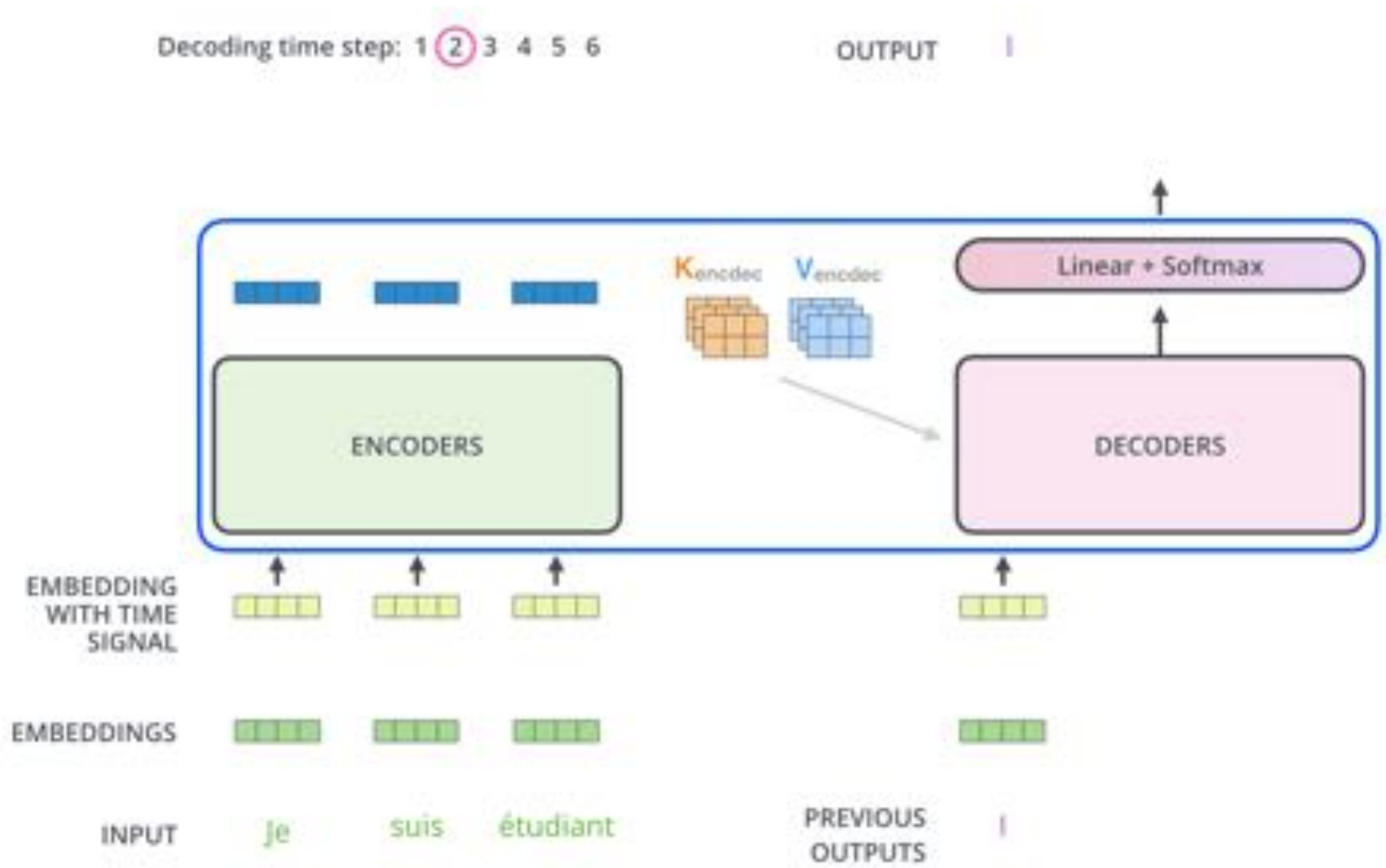


<START> as init token.
Autoregressive during inference

Figure 1: The Transformer - model architecture.

Auto-regressive models

- Auto-regressive: use previous output as current input



Source: https://jalammar.github.io/illustrated-transformer/

Transformer architecture

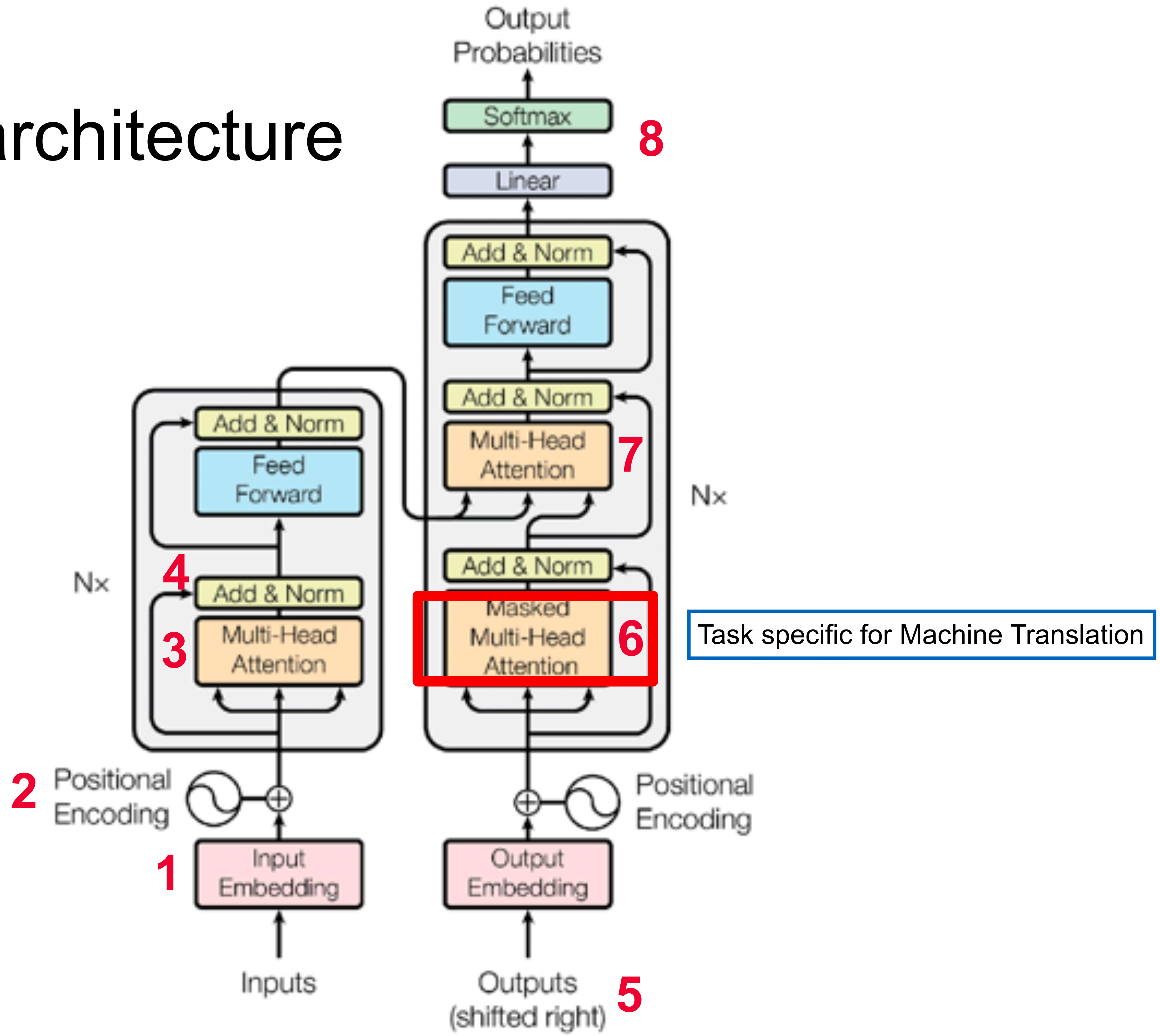


Figure 1: The Transformer - model architecture.

Transformer architecture

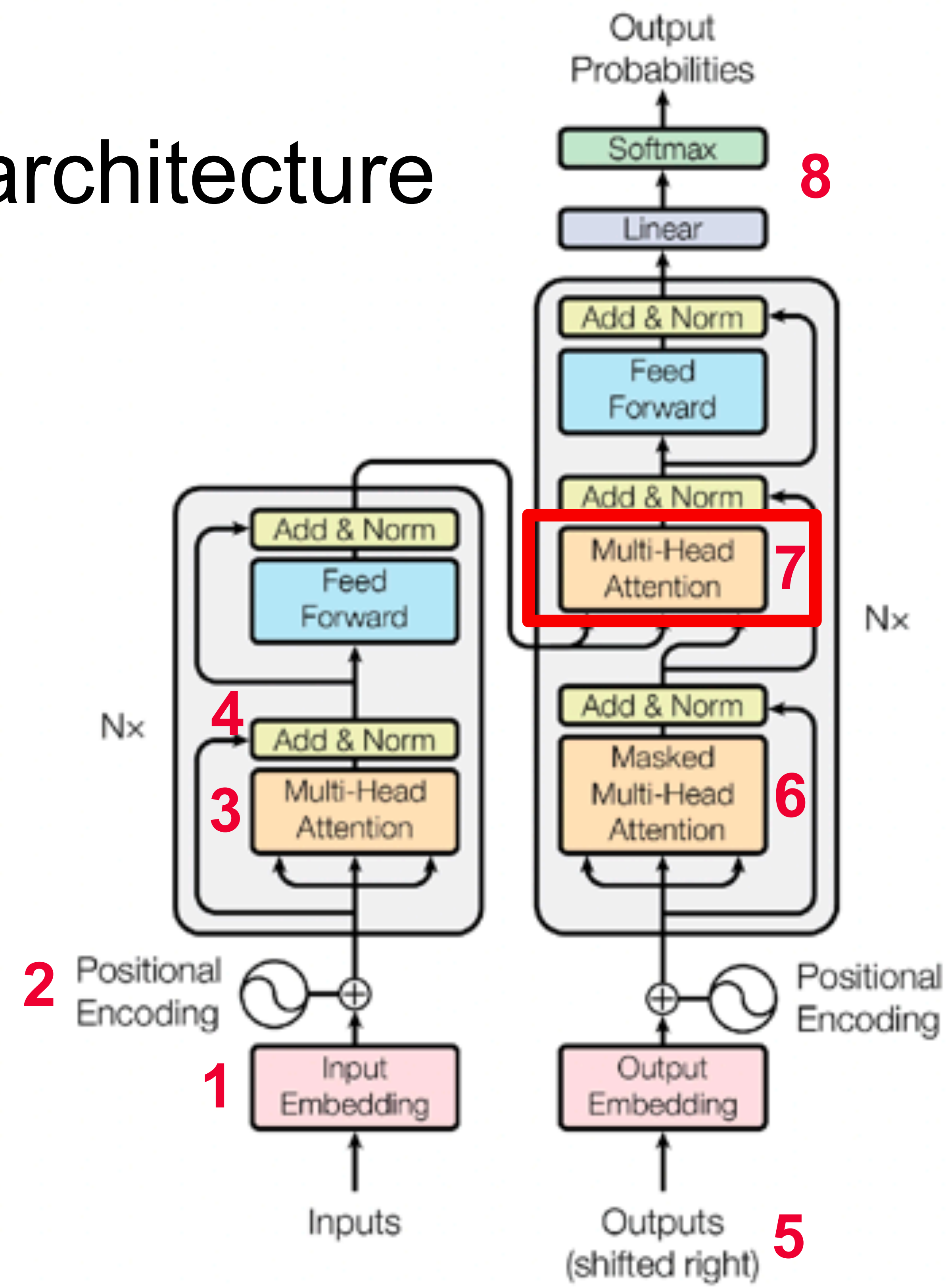
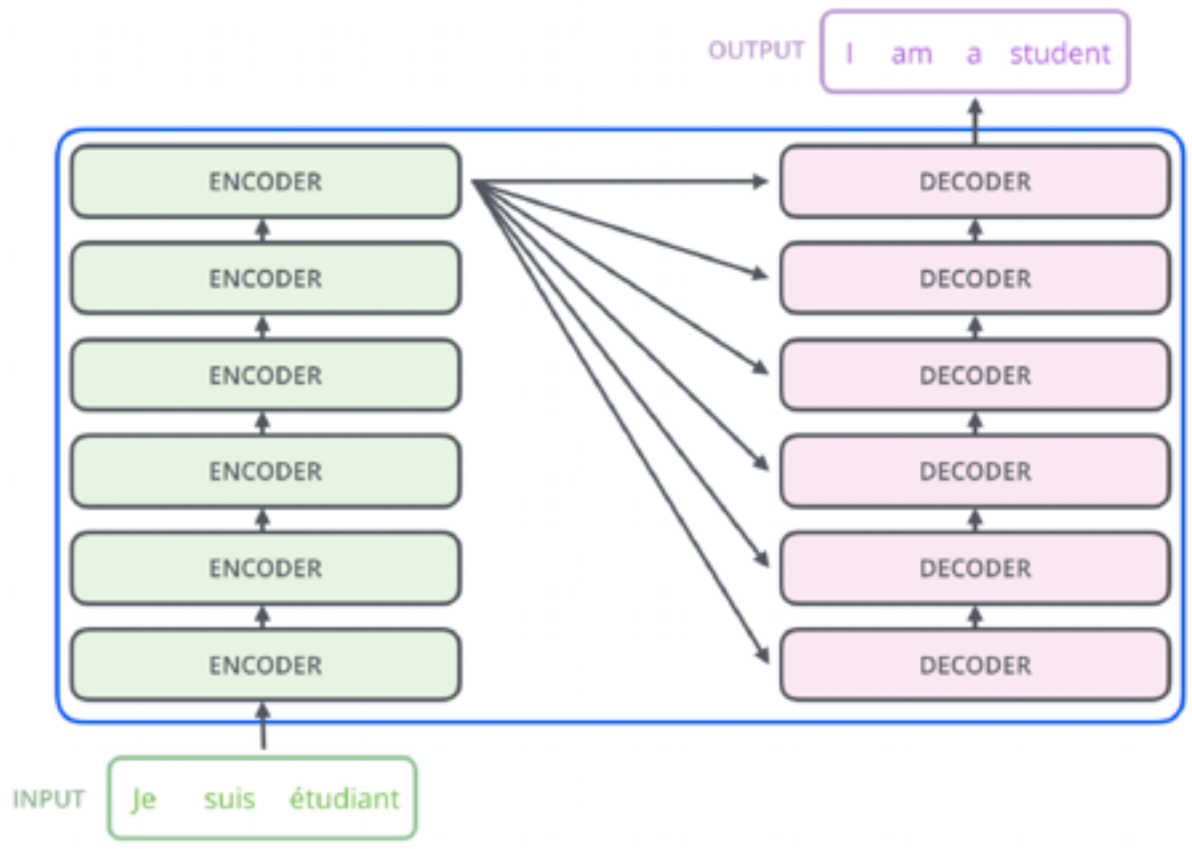


Figure 1: The Transformer - model architecture.

Attention again, but for encoders

- **Keys** and **Values**: from the the last encoder, shared
- **Queries**: from the previous decoder layer



“This allows every position in the decoder to attend over all positions in the input sequence. It mimics the typical encoder-decoder attention mechanisms in sequence-to-sequence models”

Transformer architecture

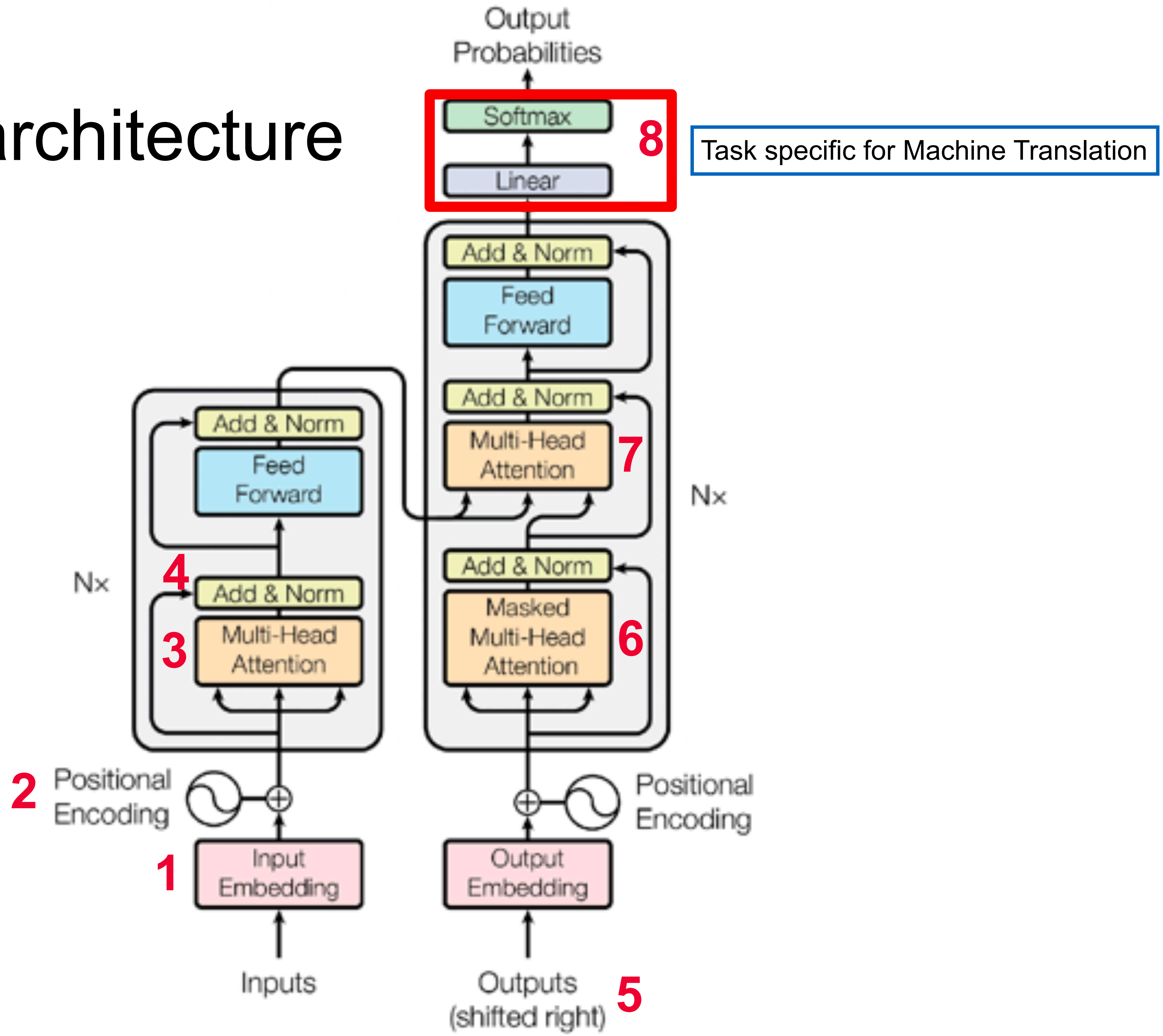


Figure 1: The Transformer - model architecture.

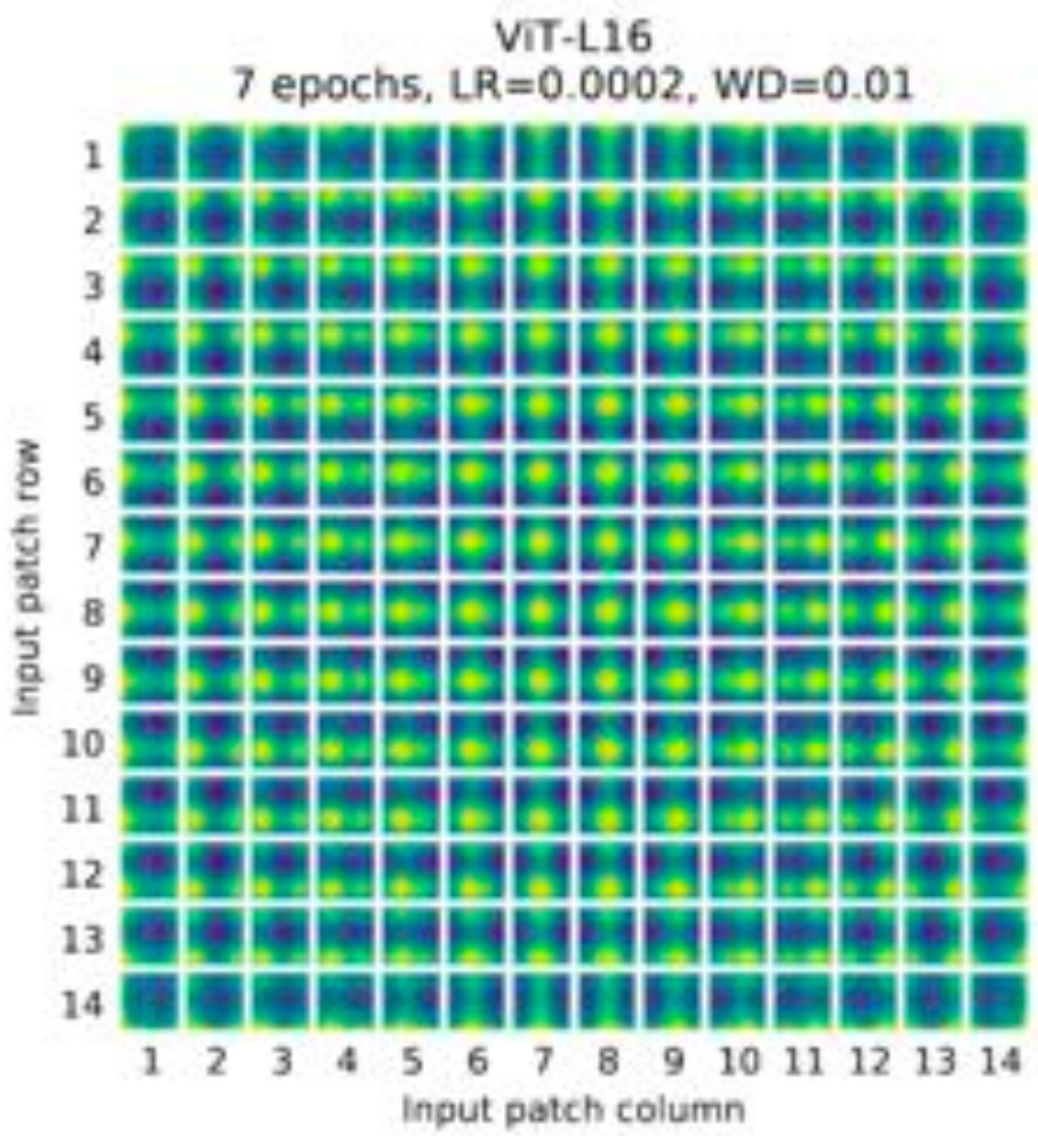
Loss function

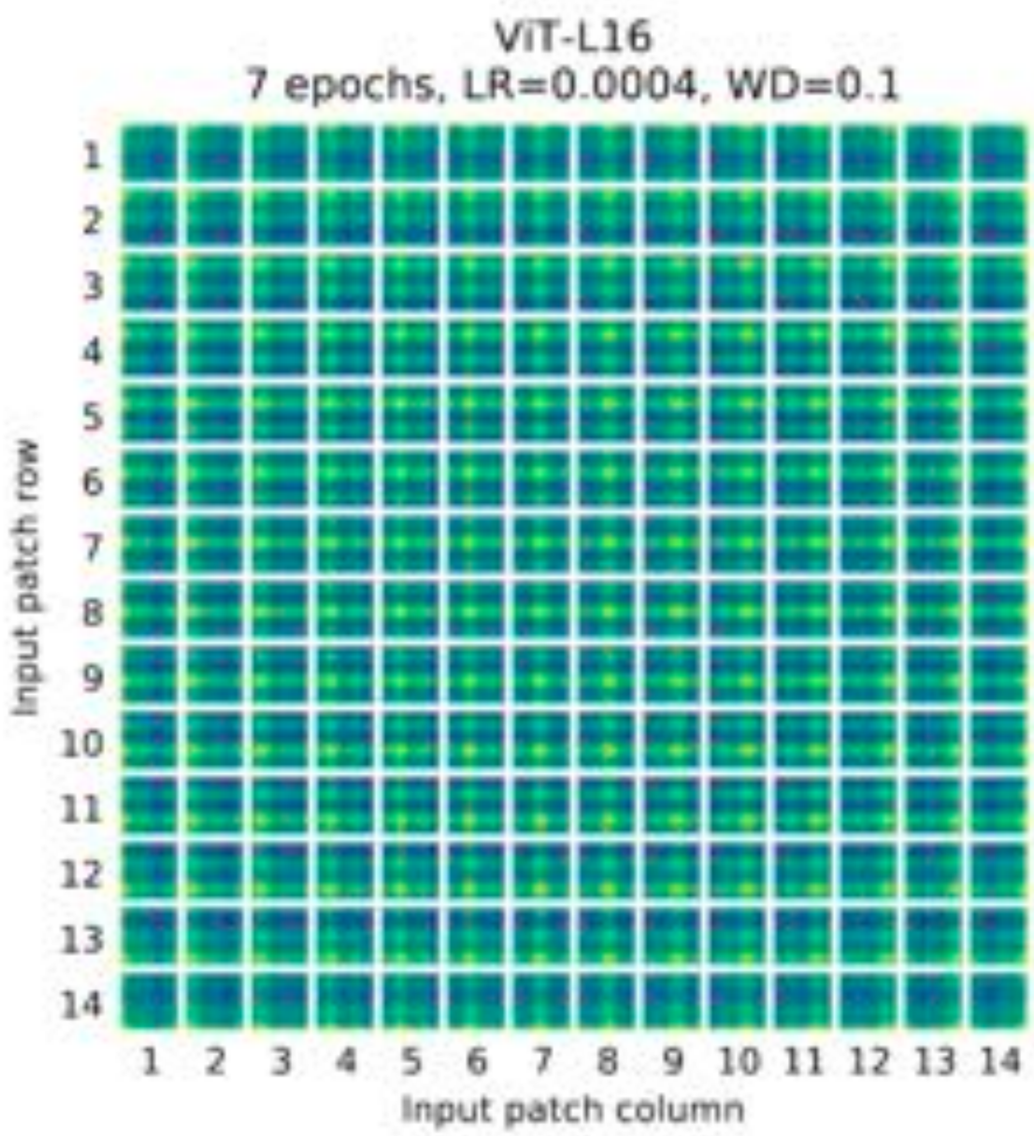
- Cross-entropy
- KL Divergence

VisionTransformer

Position Embeddings (but learnable!)

- Hyperparameters can affect learned position embeddings

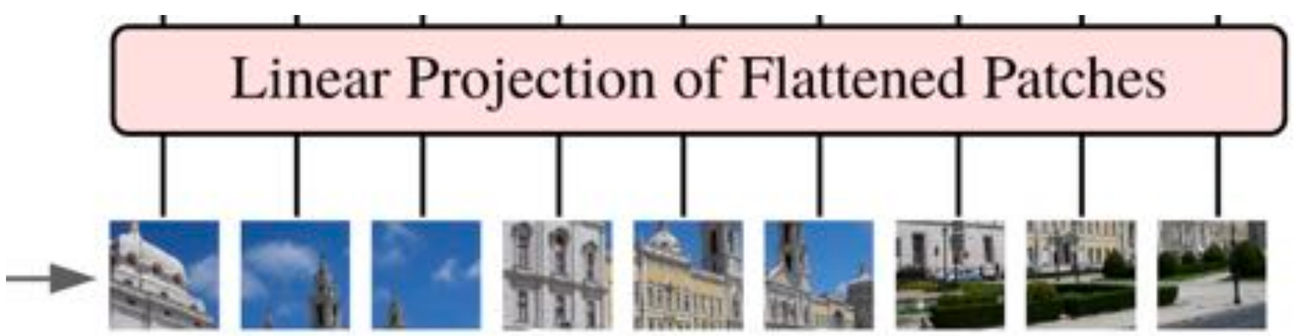




Divide Images into Patches, then Linear Project

- Given an image $x \in \mathbb{R}^{H \times W \times C}$, divide it by 16x16 patches to get N patches: $x_p \in \mathbb{R}^{N \times (P^2 \times C)}$
- Flatten each patch, map to D dimension with trainable linear projection
  - patch embeddings $x_E \in \mathbb{R}^{N \times (1 \times D)}$

- For a 224x224 image, only 14x14 patches (embeddings)!
- Reduce number of sequence length



Pre-training & Fine-tuning

- Typically, ViTs are pre-trained on large datasets, and fine-tuned to (smaller) downstream tasks.
- Initializing a new task-specific MLP head.

- **Input images have higher resolution than training images (224x224)?**
- Keep patch size same as 16x16
- Larger result sequences than 14x14

Interpolation needed for the position embeddings!

Properties of VisionTransformers

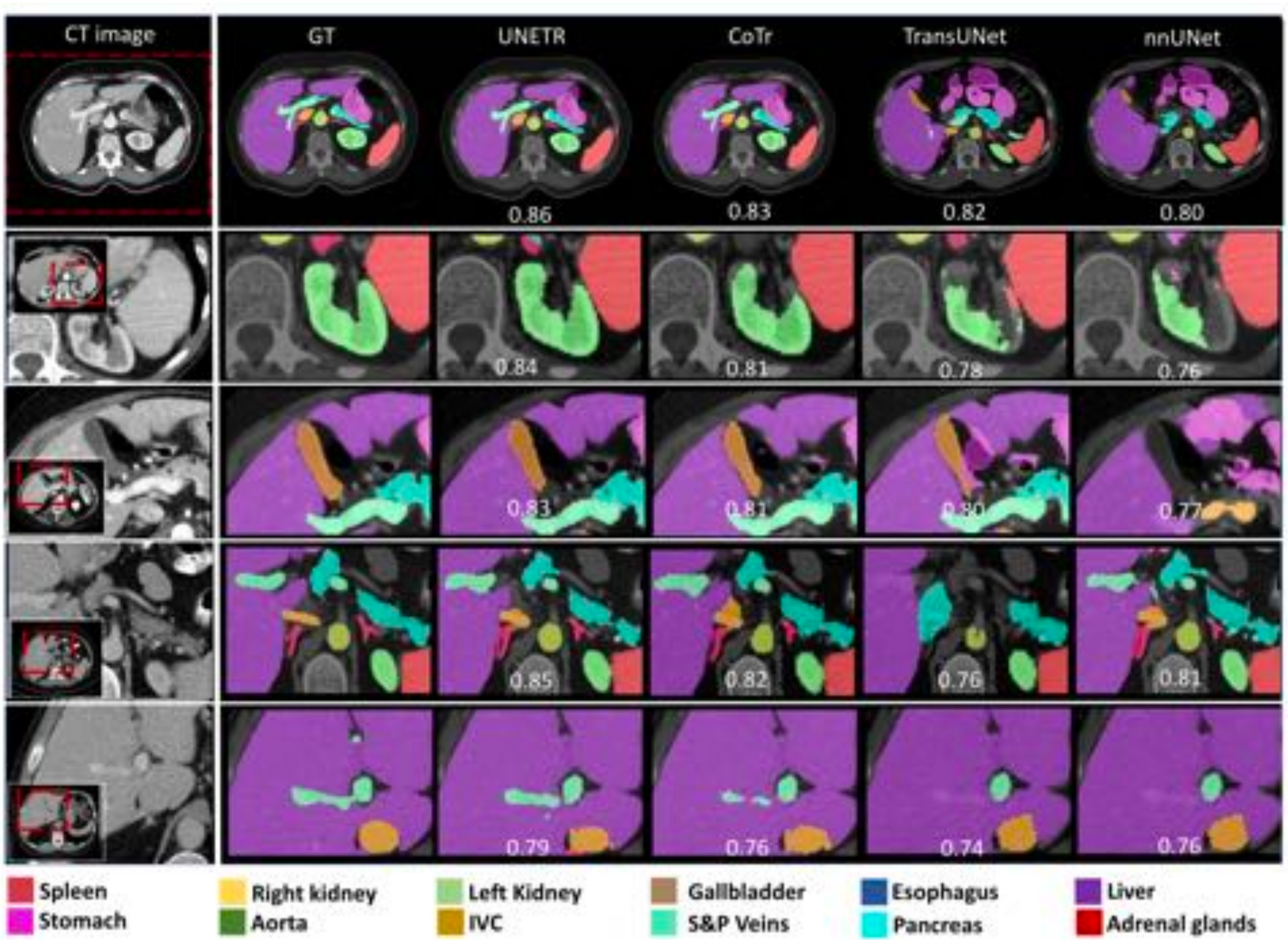
VisionTransformers has less inductive biases than CNNs

Without inductive biases, VisionTransformers have to learn everything from scratch
- Less data: worse performances than CNNs
- Big data: outperform CNNs

“Transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data… However, we find that large scale training trumps inductive bias.”

Applications

3D Medical Image Segmentation


CT image
GT
UNETR
CoTr
TransUNet
nnUNet
0.86
0.83
0.82
0.80
0.84
0.81
0.78
0.76
0.83
0.81
0.80
0.77
0.85
0.82
0.76
0.81
0.79
0.76
0.74
0.76
Spleen
Stomach
Right kidney
Aorta
Left Kidney
IVC
Gallbladder
S&P Veins
Esophagus
Pancreas
Liver
Adrenal glands

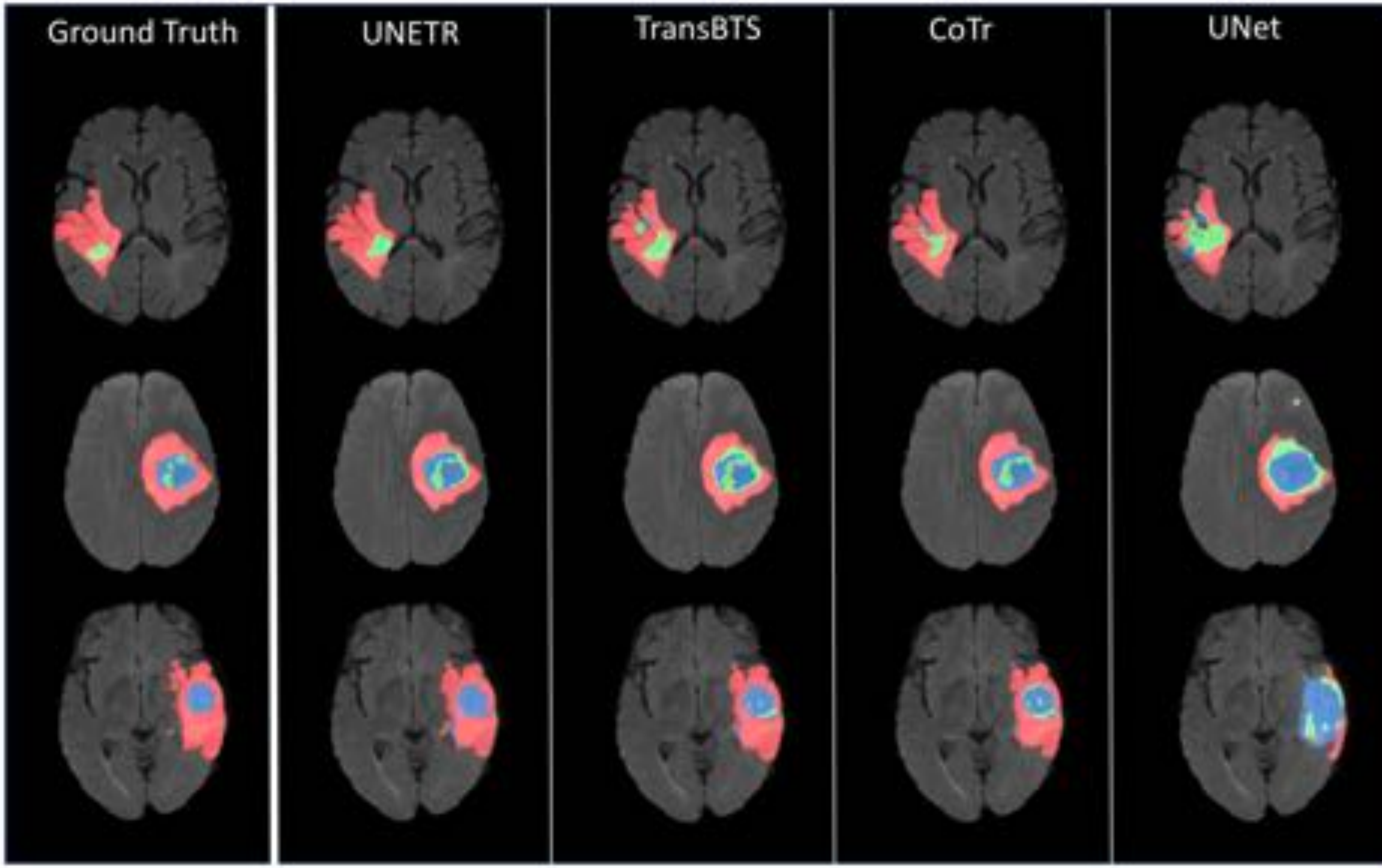


Figure 4. UNETR effectively captures the fine-grained details in segmentation outputs. The Whole Tumor (WT) encompasses a union of red, blue and green regions. The Tumor Core (TC) includes the union of red and blue regions. The Enhancing Tumor core (ET) denotes the green regions.

Model Architecture

- Input 3D medical image data
- VisionTransformer for encoder
- CNN for decoder

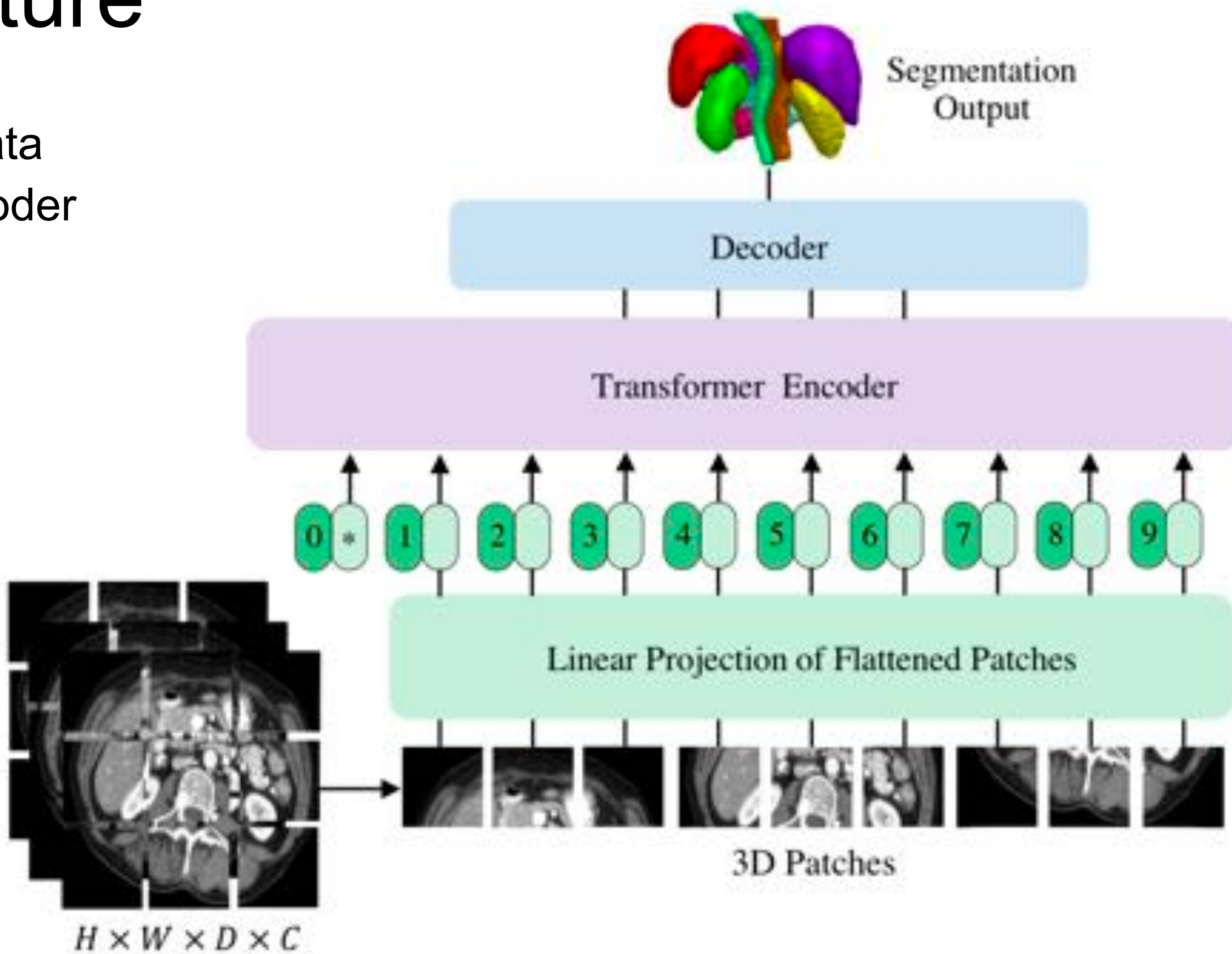
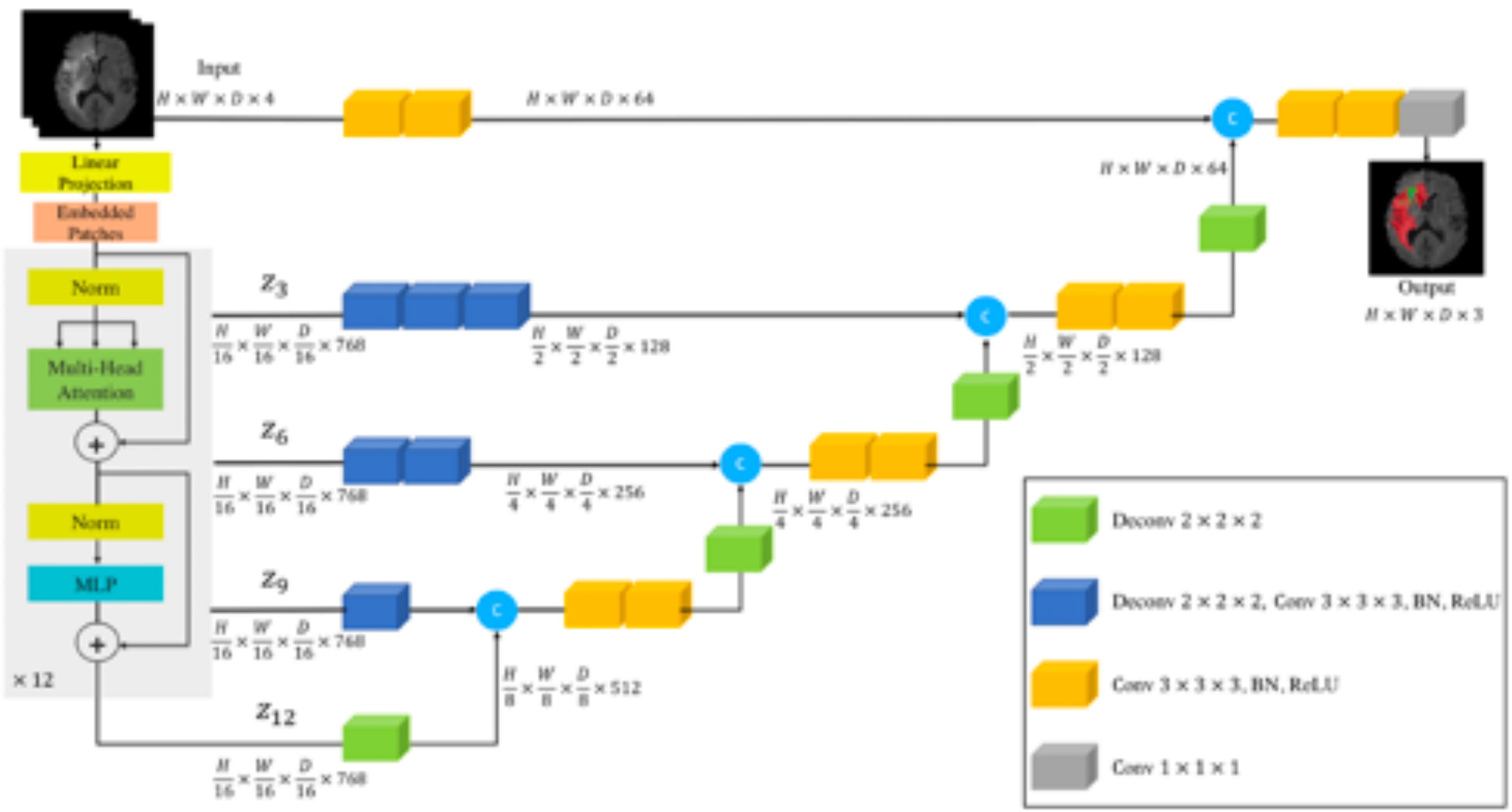


Figure 1. Overview of UNETR. Our proposed model consists of a transformer encoder that directly utilizes 3D patches and is connected to a CNN-based decoder via skip connection.

Model Architecture

- Similar to UNet, use skip connections at the {3,6,9,12}th attention layer
- CNNs as decoder for segmentation

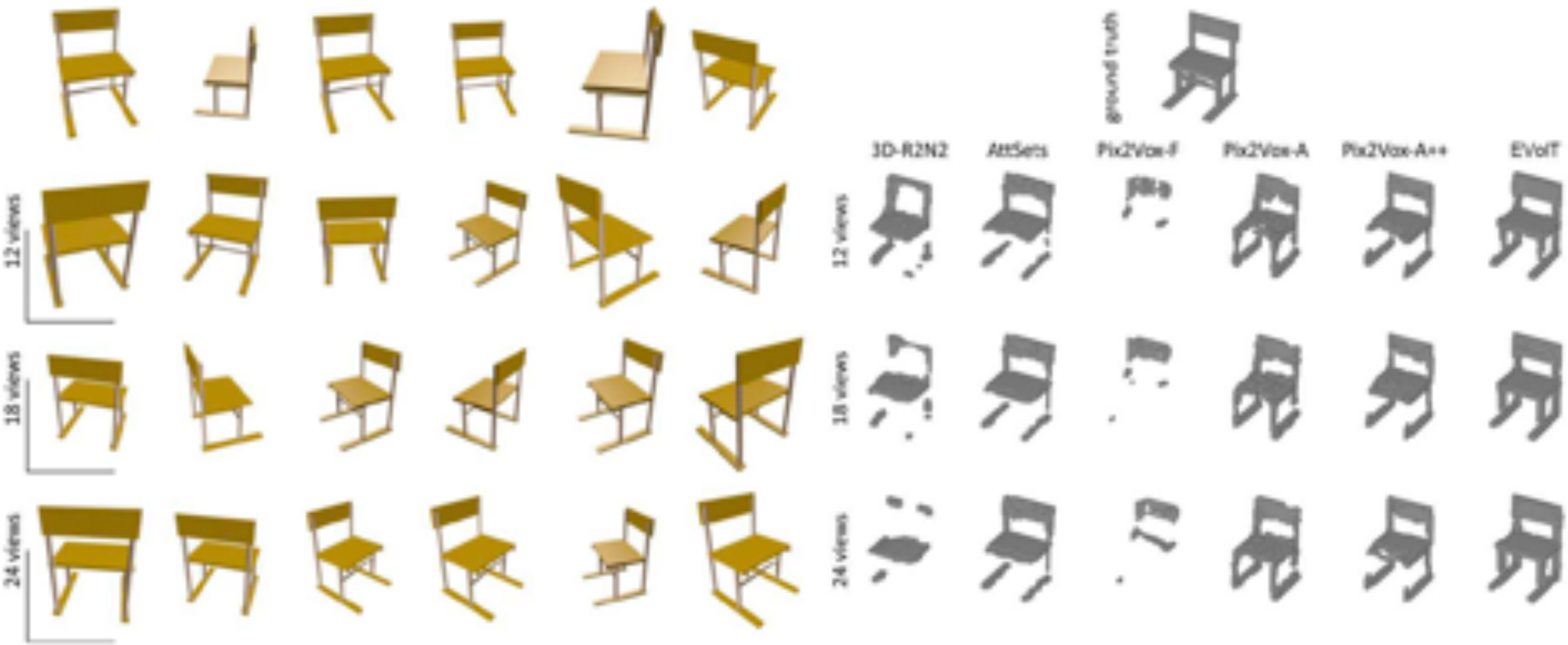


Demo

https://github.com/Project-MONAI/tutorials/blob/main/3d_segmentation/unetr_btcv_segmentation_3d.ipynb

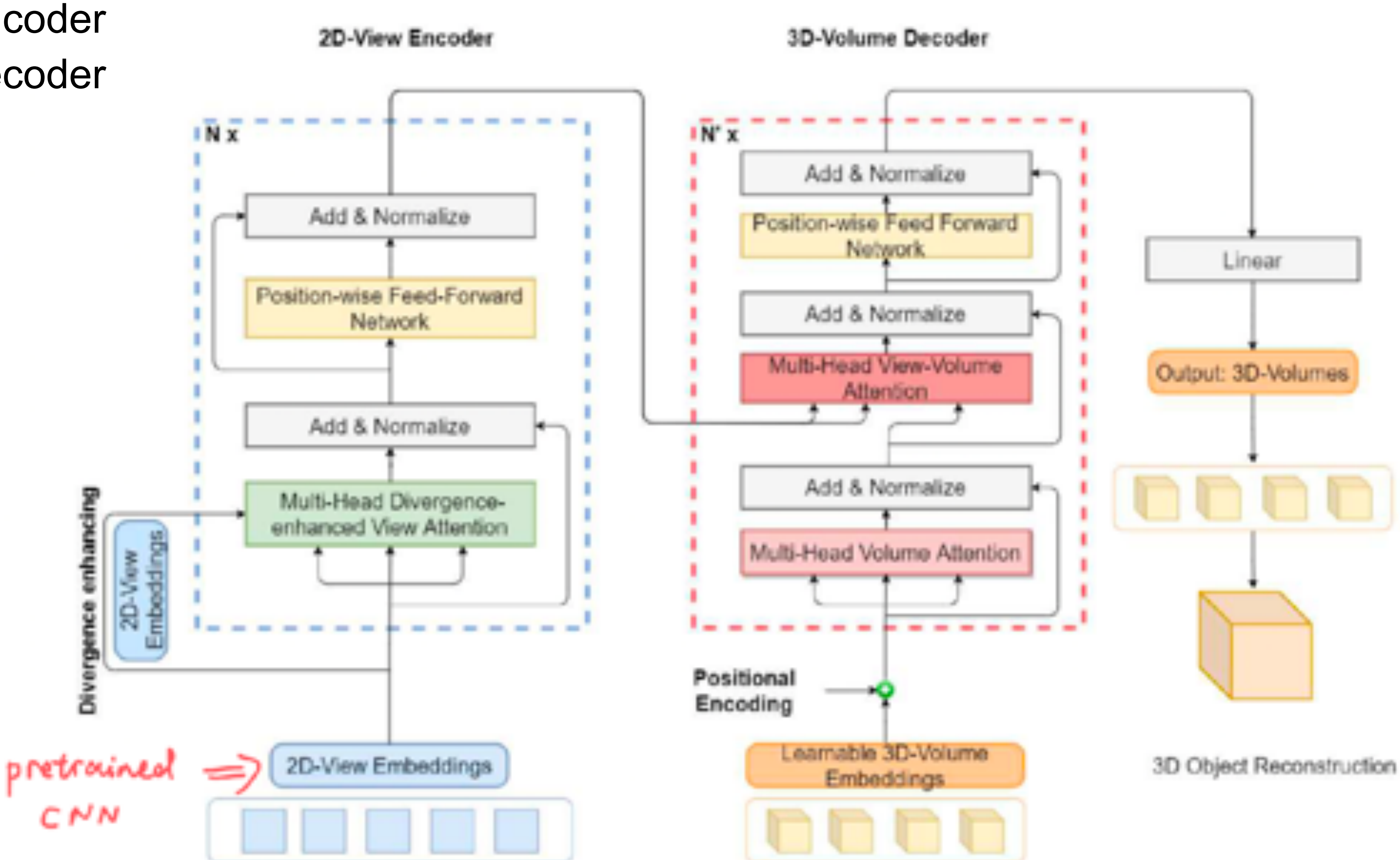
Applications

Multi-view 3D Reconstruction with Transformer


ground truth
3D-R2N2
AttSets
Pix2Vox-F
Pix2Vox-A
Pix2Vox-A++
EVolT
12 views
18 views
24 views

Model Architecture

- 2D ViT encoder
- 3D ViT decoder



Model Architecture

- 2D ViT encoder
- 3D ViT decoder