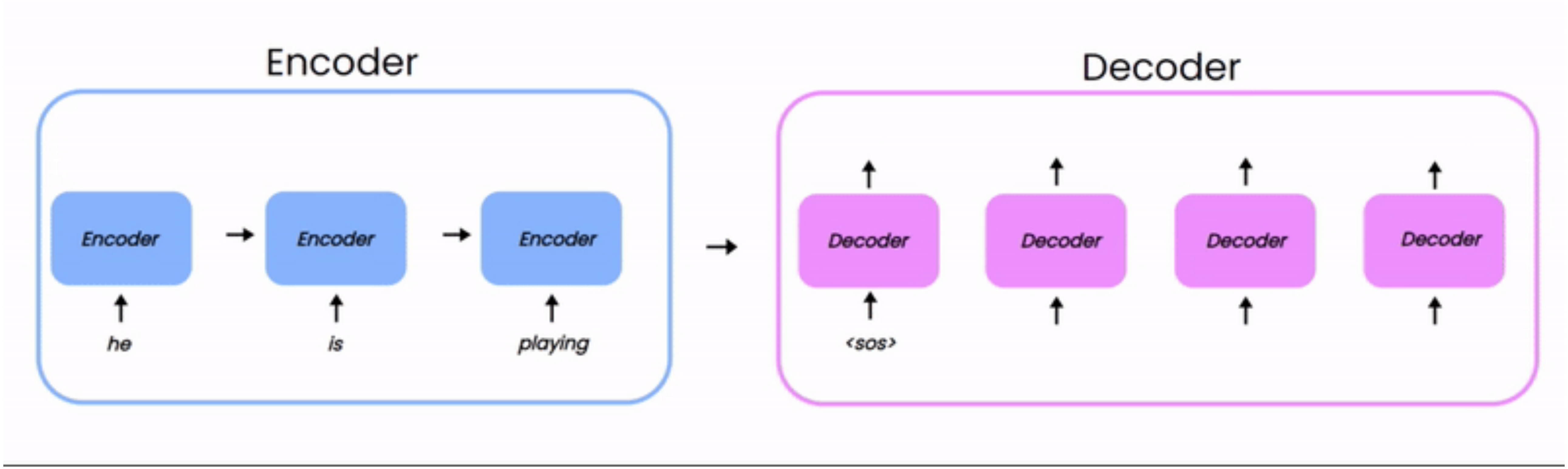
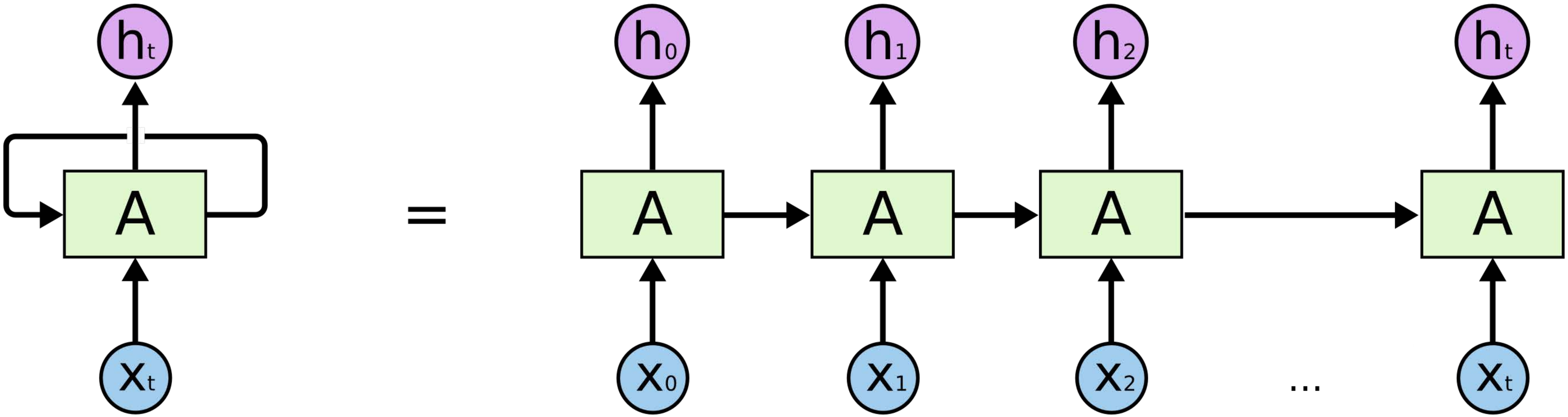
ENM 5310: Data-driven Modeling and Probabilistic Scientific Computing

Lecture #15: Transformers

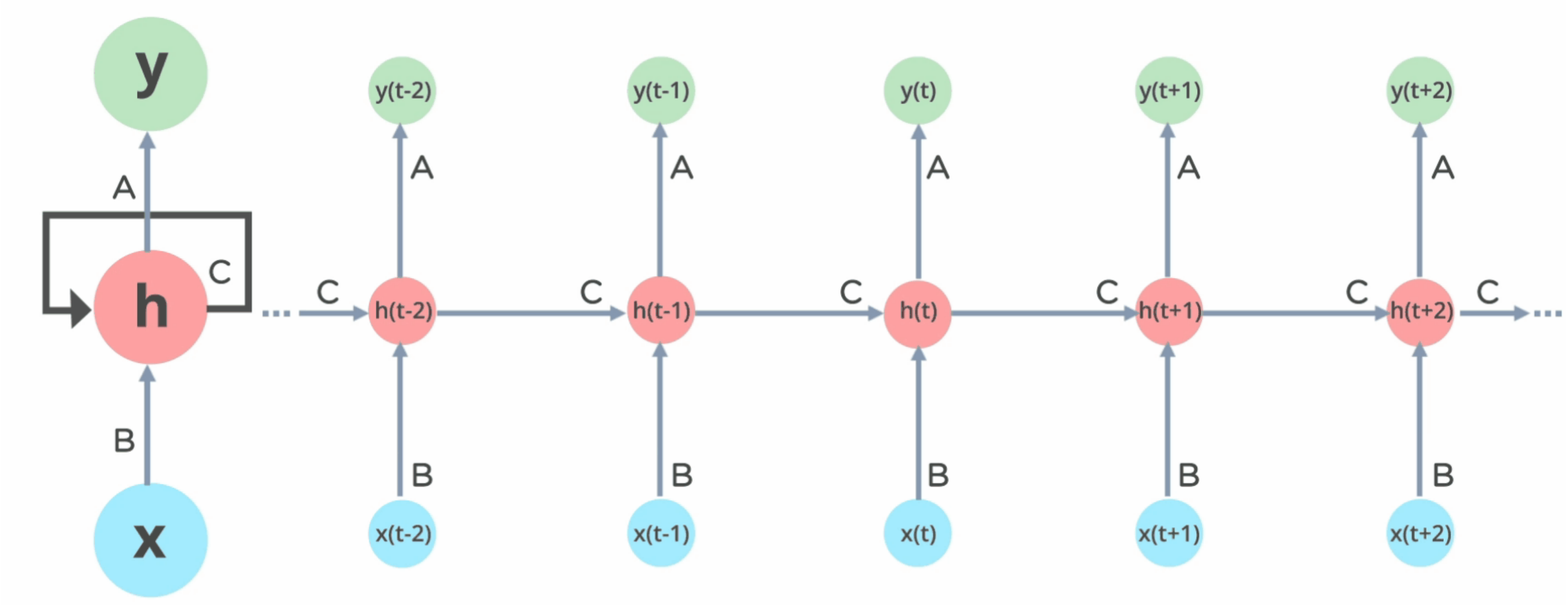
Modeling of sequence data



RNNs



An unrolled recurrent neural network.



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN limitations

- Sequential processing is slow (can't parallelize)
- Long-range dependency issues (vanishing gradients)
- Limited context window in practice

“The cat is on the table”

“The cat, who belongs to my mother, is on the table”.

E.g.: when translating we need to remember "cat" is the subject even after processing many words.

- A need for better models:
  - Language translation requiring full sentence context
  - Document summarization
  - Question answering

  - Protein Structure Prediction (long amino acid sequences)
  - Molecular Property Prediction (multi-atom interactions)
  - Climate Science (long timescales)
  - Astronomical Data Analysis (vast spatial and time scales)
  -

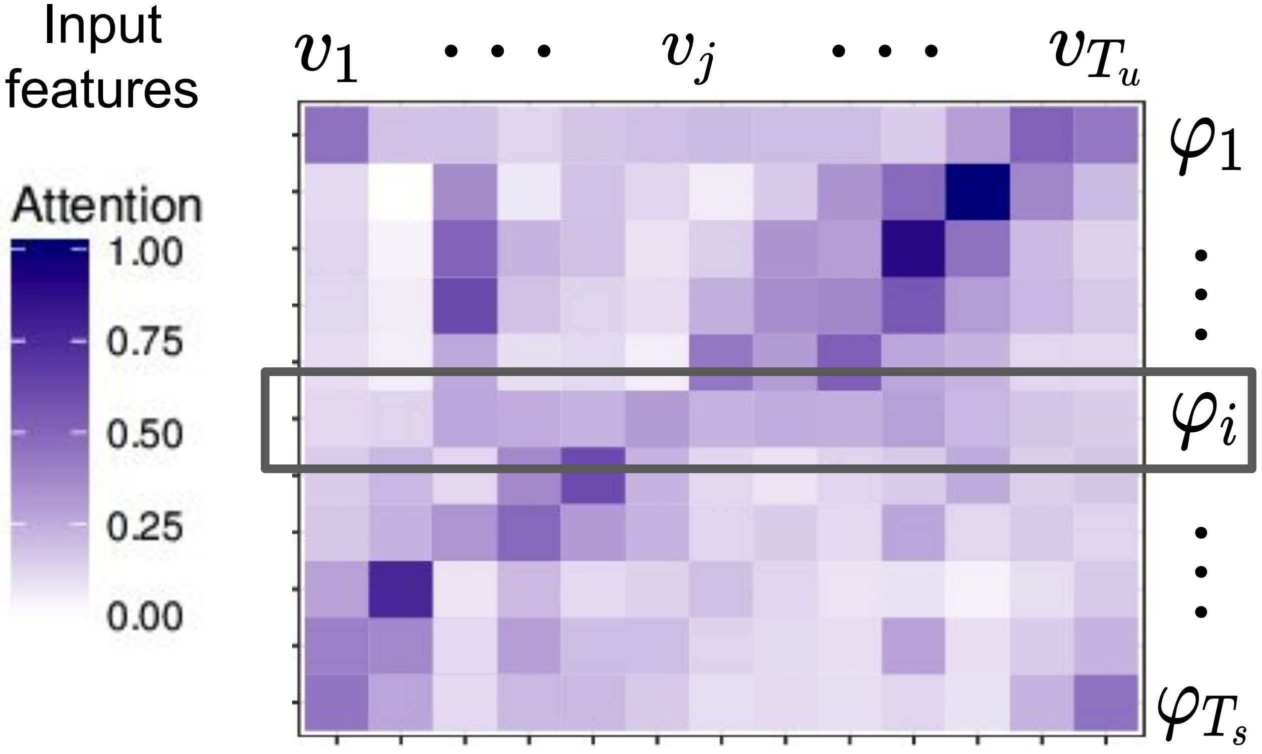
Key Innovation: Attention

Bahdanau Attention

Input sentence $[u_1, \ldots, u_i, \ldots, u_{T_u}]$

↓

Input features



Attention weights φ_{ij}

$$\begin{bmatrix} c_1 \\ \vdots \\ c_i \\ \vdots \\ c_{T_s} \end{bmatrix} \rightarrow c_i = \sum_{j=1}^{T_u} \varphi_{ij} v_j$$

Context vector

↓

$$p(s_i \mid c_i, u)$$

Probability of word s_i at output position i

↓

$$[s_1, \ldots, s_i, \ldots, s_{T_s}]$$

Output sentence

Bahdanau, Dzmitry, Kyung Hyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *3rd International Conference on Learning Representations, ICLR 2015*. 2015.

From Sequential to Parallel Processing

- Compare how humans read vs how RNNs process text
  - Humans: Quick glances at relevant parts
  - RNNs: Must process word by word
- Attention is "looking" at all words simultaneously

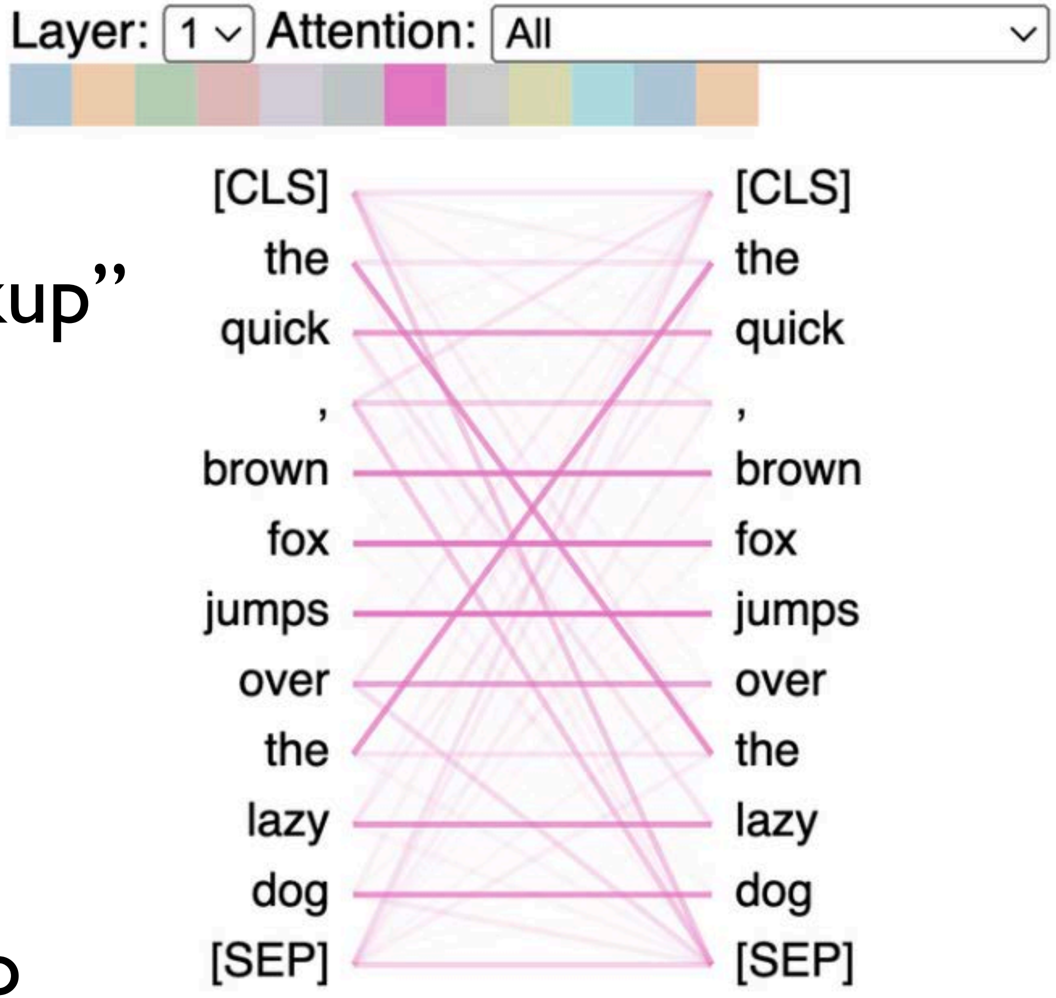
Key innovation: Attention

Think of attention as a "smart dictionary lookup"

- Keys: What you're looking for
- Values: What you get back
- Queries: What you're asking for

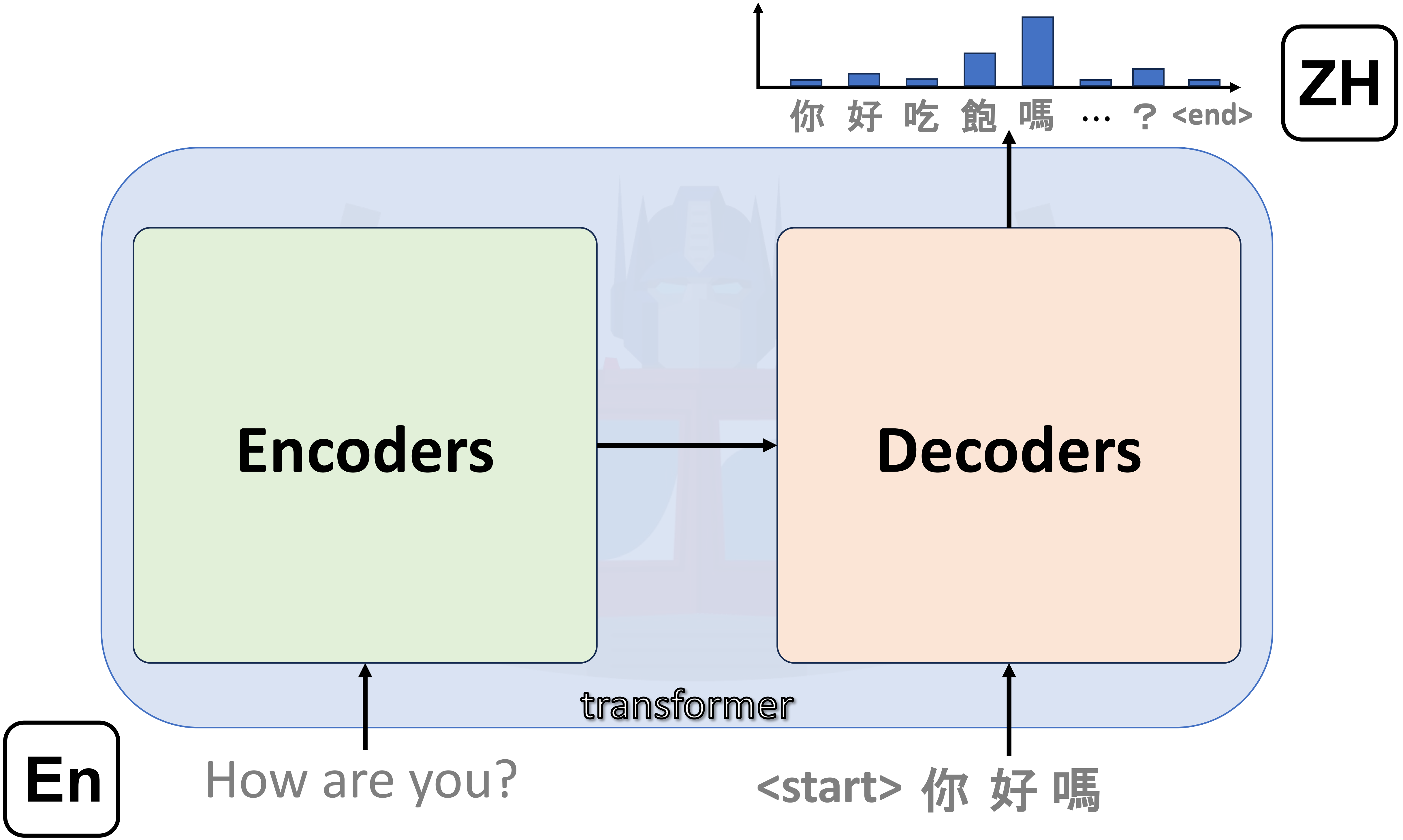
Simple example: Translation attention

- Each word in target language might need to "look at" multiple source words



Layer: 1 Attention: All

Transformers



Encoder-Decoder Auto-regressive Models

Slide by Jia-Bin Huang, University of Maryland College Park

Key Building Blocks

- Tokenization
- Token embeddings
- Position embeddings
- Self-attention & Multi-head self-attention
- MLPs
- Cross-attention

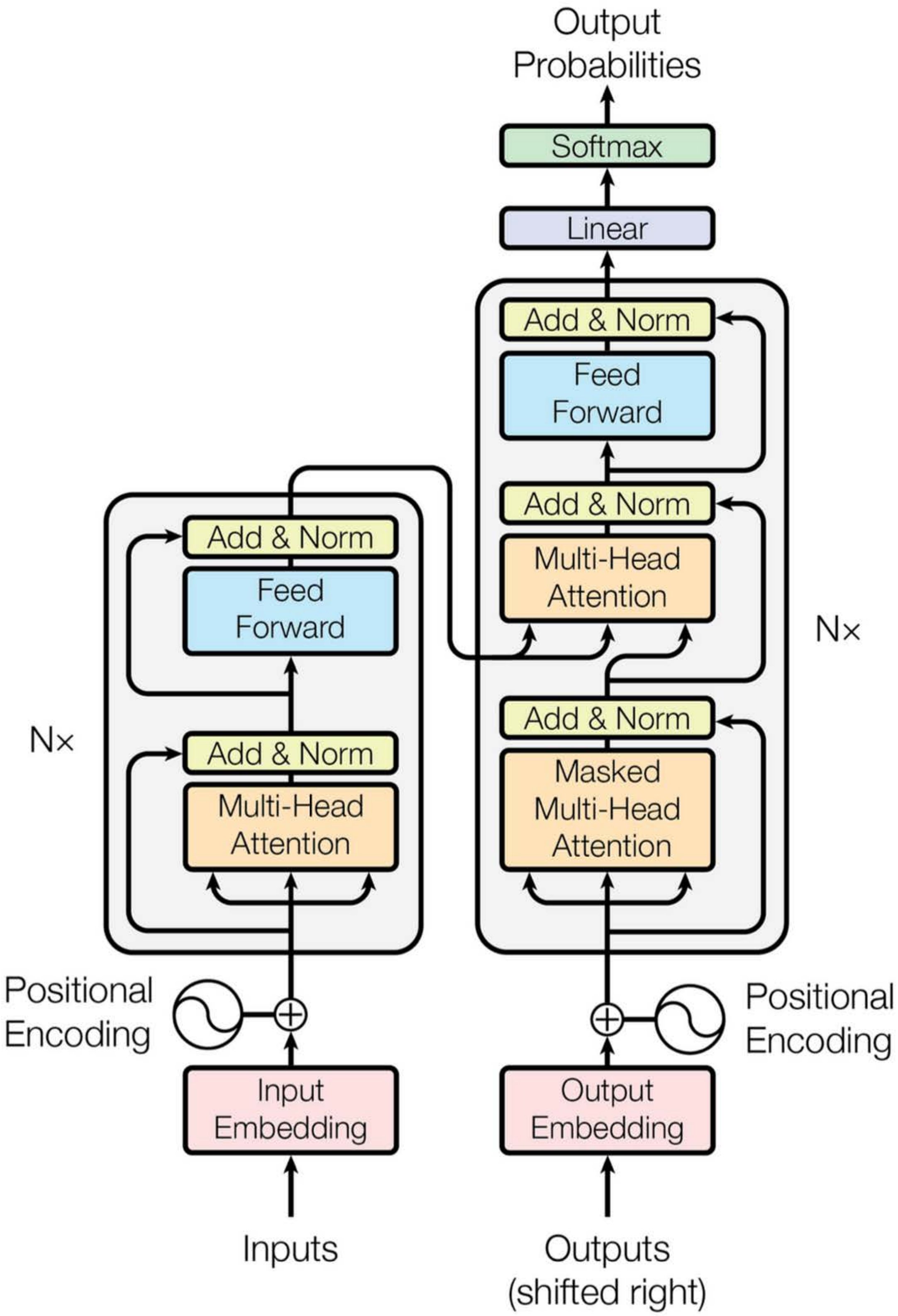
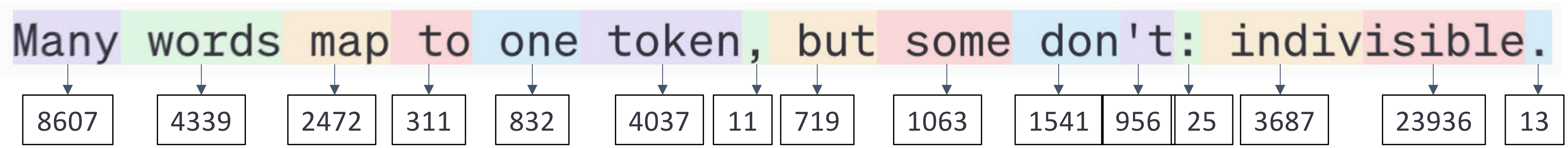


Figure 1: The Transformer - model architecture.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Tokenization in NLP



One-hot encoding

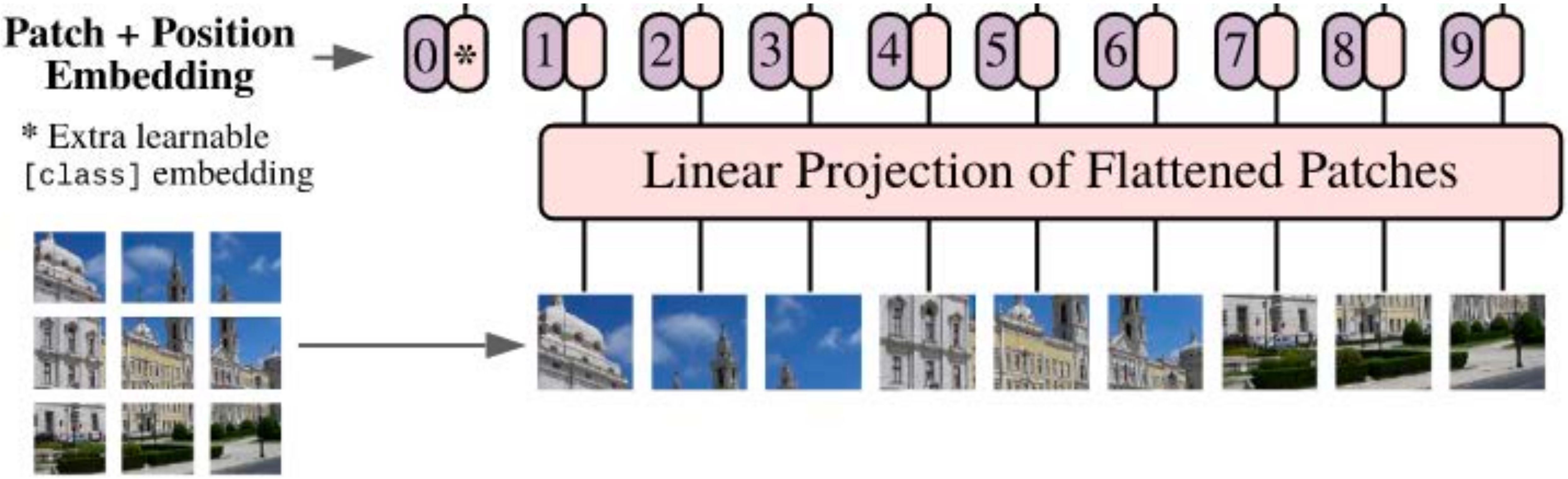
# tokens

$$
\begin{array}{ccccc}
\text{cat} & \text{dog} & \text{bear} & \text{cow} & \text{indiv} \\
\begin{bmatrix}1\\0\\0\\0\\0\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\1\\0\\0\\0\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\1\\0\\0\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\1\\0\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\0\end{bmatrix}
\end{array}
$$

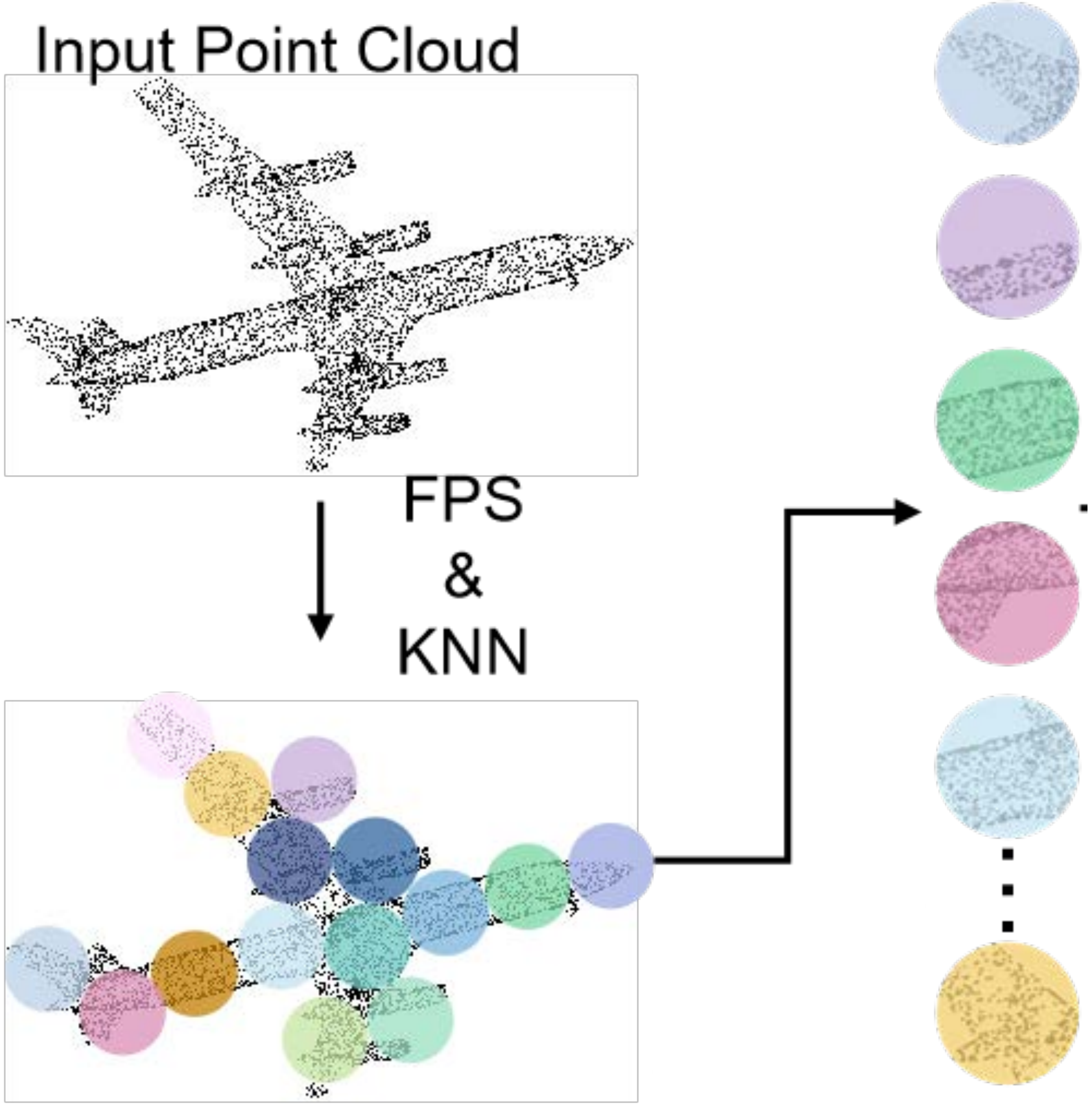
Value 1 at 3687th entry

Slide by Jia-Bin Huang, University of Maryland College Park

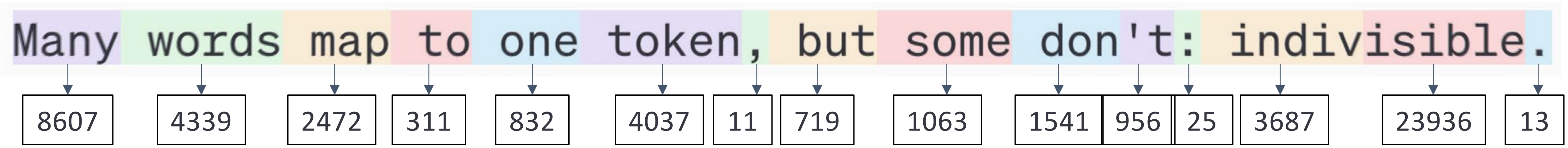
Tokenization in Vision


Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Tokenization in Point Clouds



Tokenization in NLP



Many words map to one token, but some don't: indivisible.

8607 4339 2472 311 832 4037 11 719 1063 1541 956 25 3687 23936 13

One-hot encoding

# tokens

$$\begin{array}{ccccc} \text{cat} & \text{dog} & \text{bear} & \text{cow} & \text{indiv} \\ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{array}$$

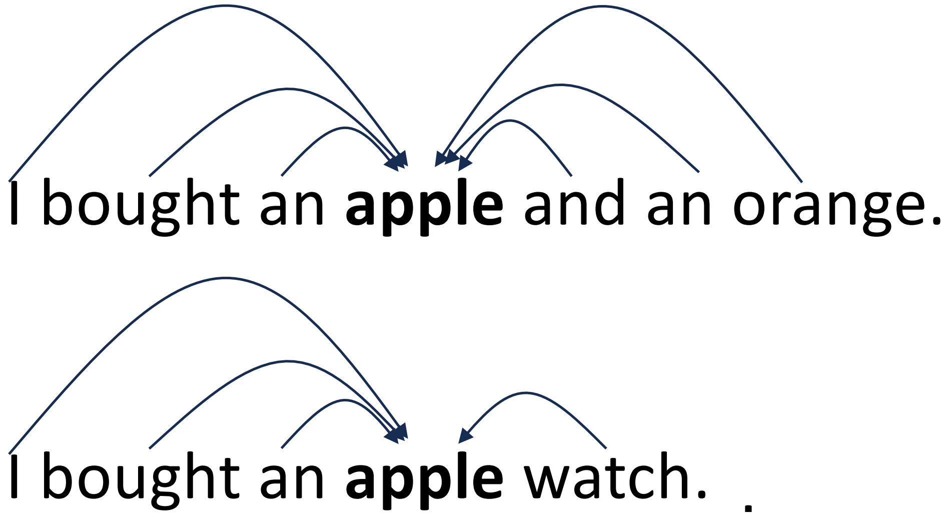
Value 1 at 3687th entry

Slide by Jia-Bin Huang, University of Maryland College Park

Token Embeddings

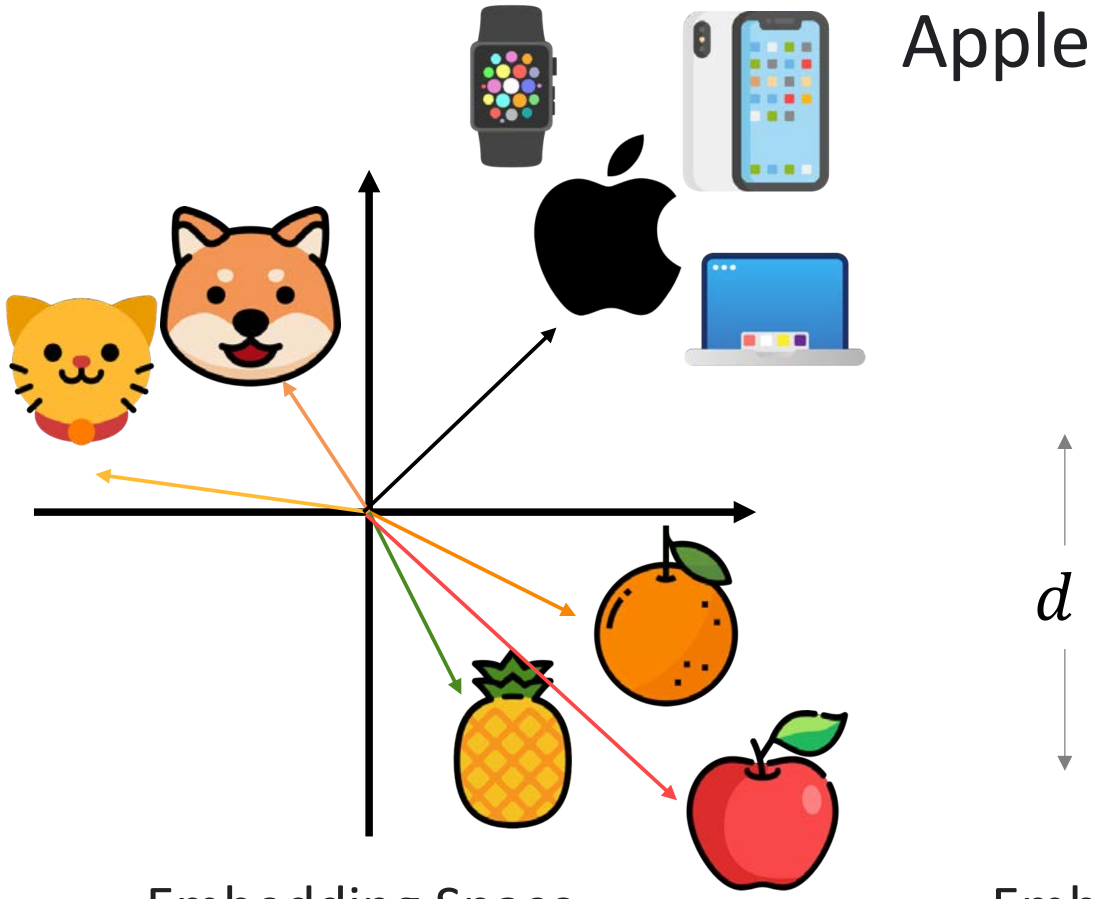
$$\begin{array}{ccccc} \text{cat} & \text{dog} & \text{bear} & \text{cow} & \text{indiv} \\ \begin{bmatrix}1\\0\\0\\0\\0\\\vdots\\0\end{bmatrix} & \begin{bmatrix}0\\1\\0\\0\\0\\\vdots\\0\end{bmatrix} & \begin{bmatrix}0\\0\\1\\0\\0\\\vdots\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\1\\0\\\vdots\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\0\end{bmatrix} \end{array}$$

Value 1 at 3687th entry

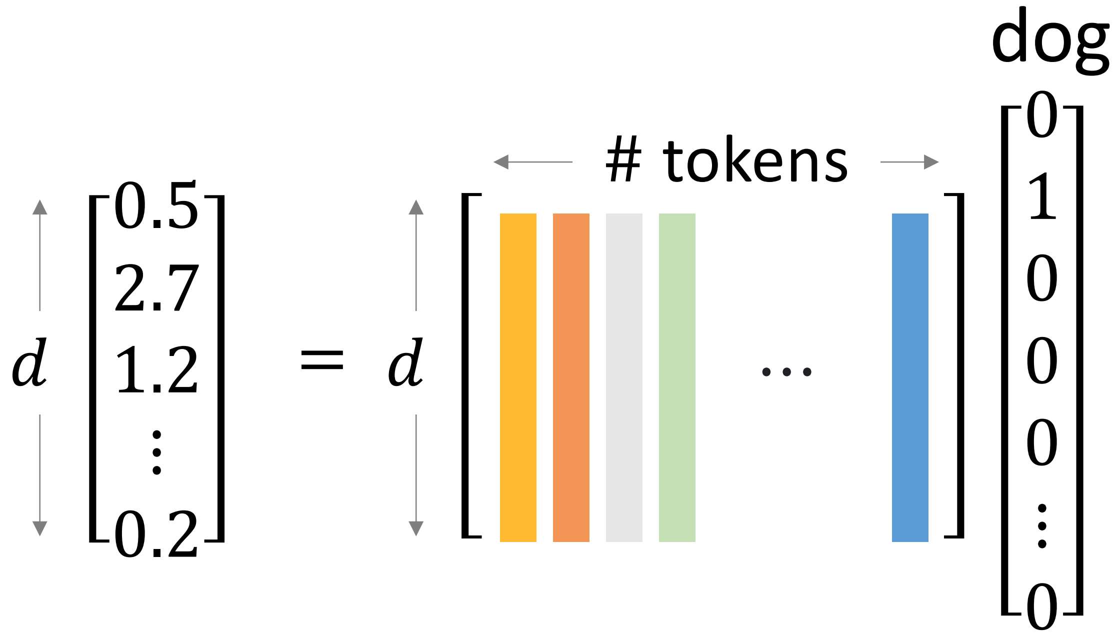


I bought an **apple** and an orange.

I bought an **apple** watch.



Embedding Space



Embedded token

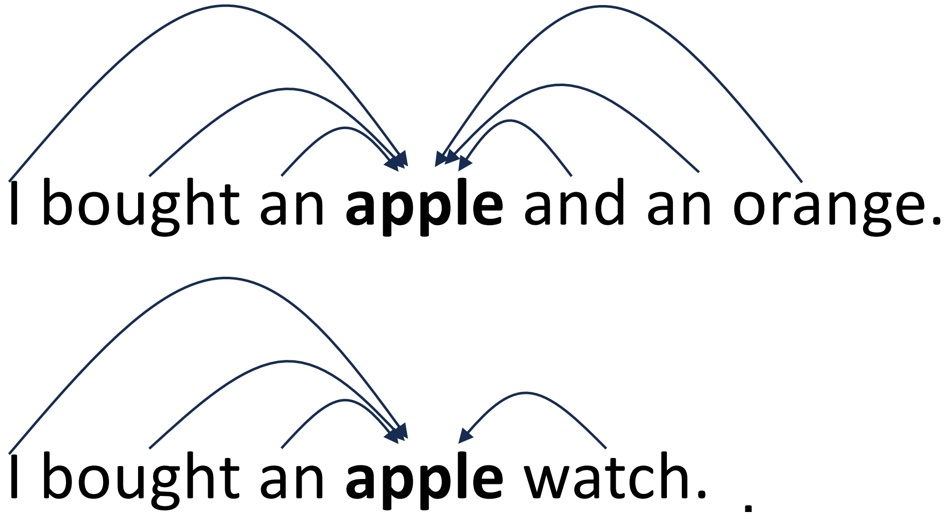
Embedding Matrix

Slide by Jia-Bin Huang, University of Maryland College Park

Token Embeddings

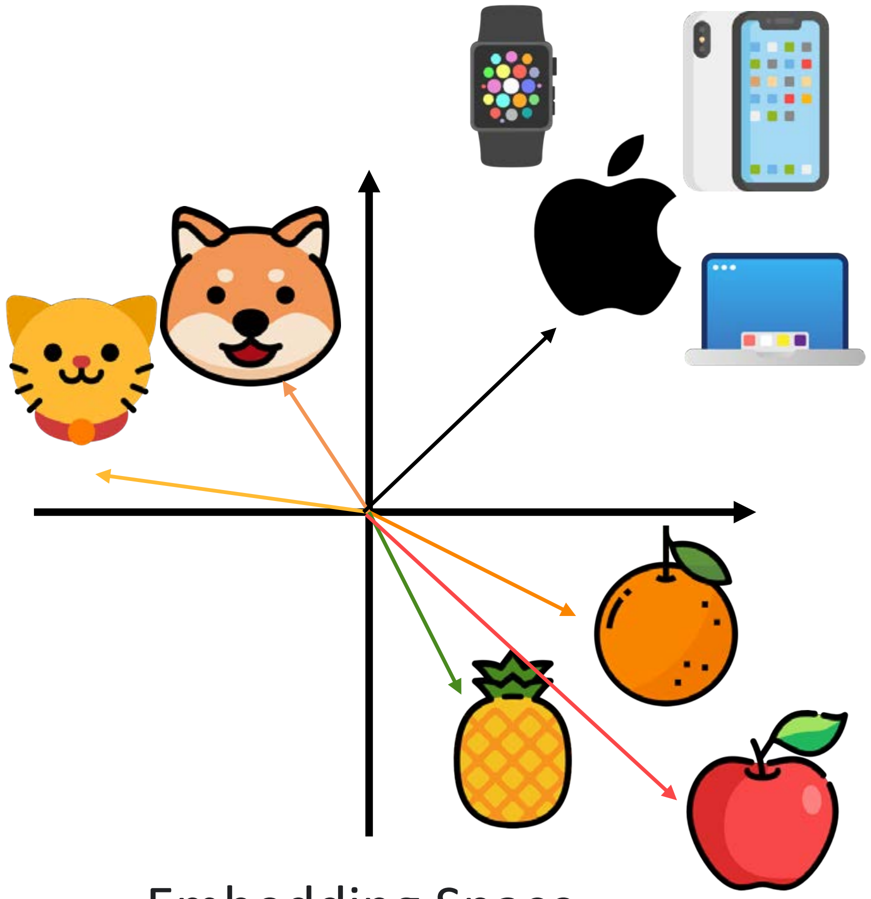
$$\begin{array}{ccccc} \text{cat} & \text{dog} & \text{bear} & \text{cow} & \text{indiv} \\ \begin{bmatrix}1\\0\\0\\0\\0\\\vdots\\0\end{bmatrix} & \begin{bmatrix}0\\1\\0\\0\\0\\\vdots\\0\end{bmatrix} & \begin{bmatrix}0\\0\\1\\0\\0\\\vdots\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\1\\0\\\vdots\\0\end{bmatrix} & \begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\0\end{bmatrix} \end{array}$$

Value 1 at 3687th entry

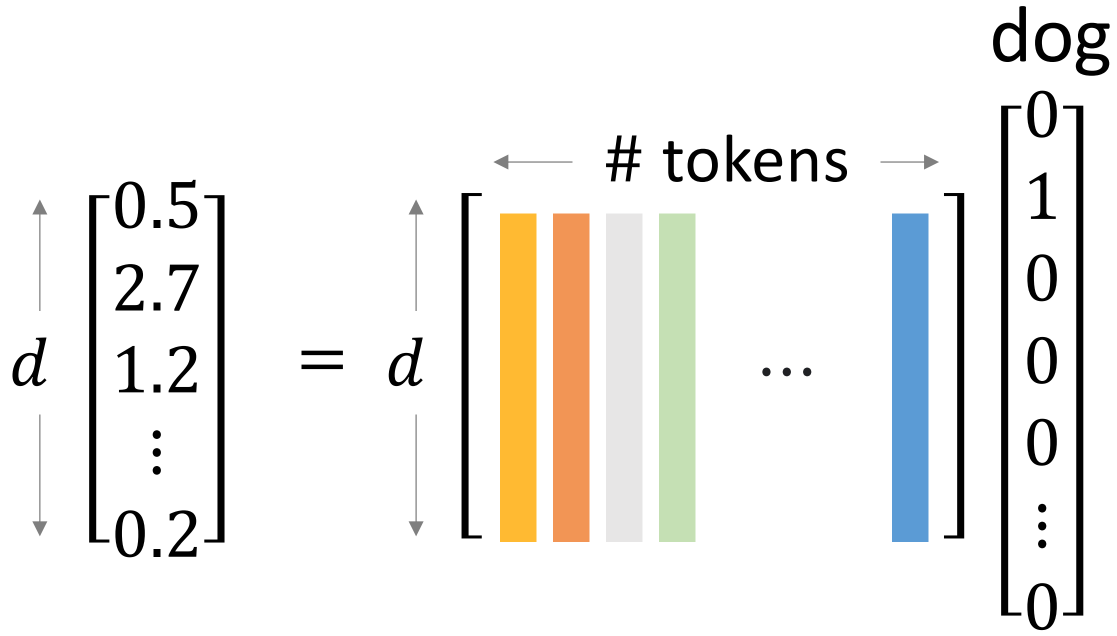


I bought an **apple** and an orange.

I bought an **apple** watch.



Embedding Space

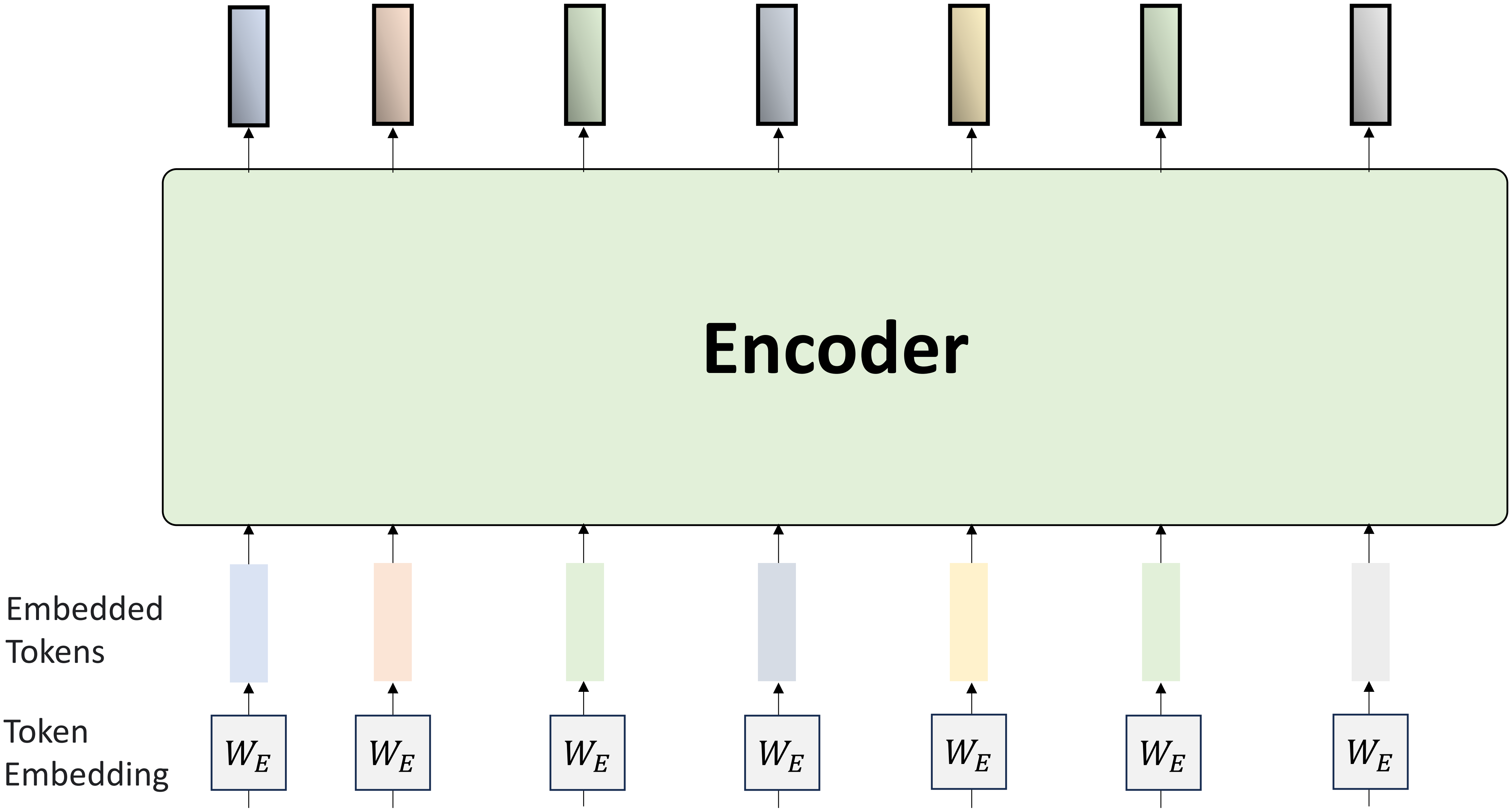


Embedded token

Embedding Matrix

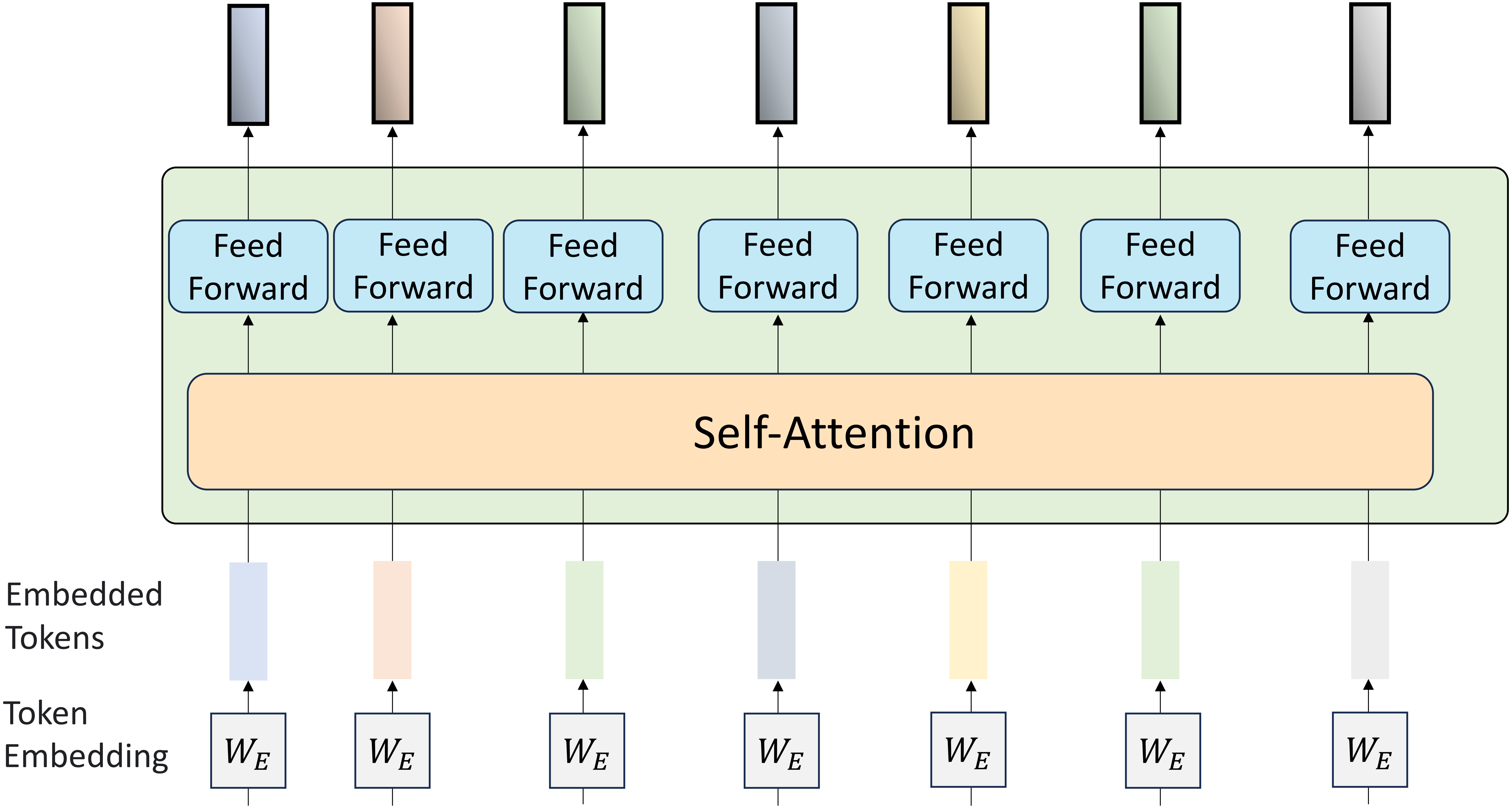
Slide by Jia-Bin Huang, University of Maryland College Park

Encoder



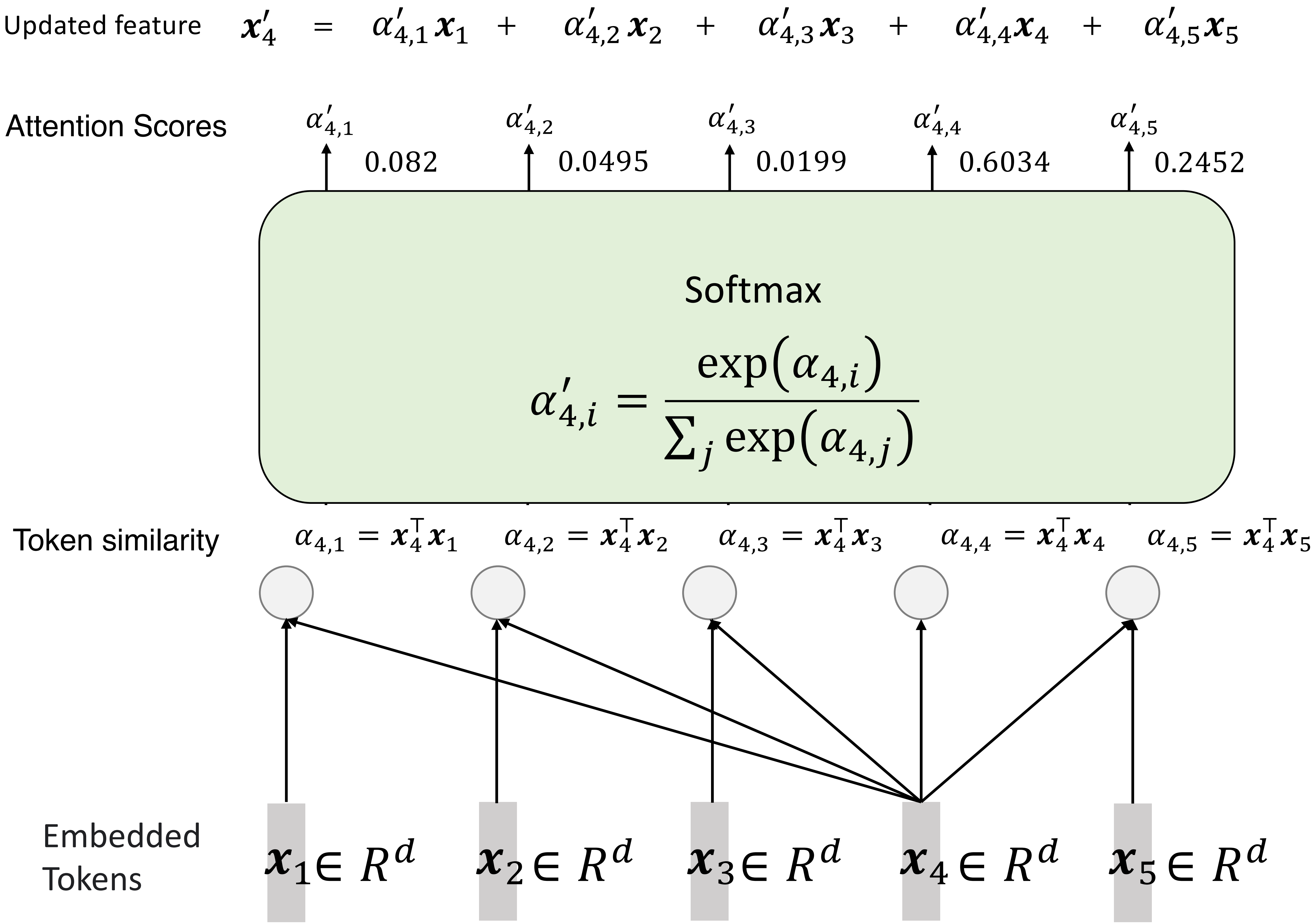
Slide by Jia-Bin Huang, University of Maryland College Park

Encoder



Slide by Jia-Bin Huang, University of Maryland College Park

Self-Attention



Updated feature $\boldsymbol{x}'_4 = \alpha'_{4,1}\,\boldsymbol{x}_1 + \alpha'_{4,2}\,\boldsymbol{x}_2 + \alpha'_{4,3}\,\boldsymbol{x}_3 + \alpha'_{4,4}\boldsymbol{x}_4 + \alpha'_{4,5}\boldsymbol{x}_5$

Attention Scores $\alpha'_{4,1}$ 0.082 $\alpha'_{4,2}$ 0.0495 $\alpha'_{4,3}$ 0.0199 $\alpha'_{4,4}$ 0.6034 $\alpha'_{4,5}$ 0.2452

Softmax

$$\alpha'_{4,i} = \frac{\exp(\alpha_{4,i})}{\sum_j \exp(\alpha_{4,j})}$$

Token similarity $\alpha_{4,1} = \boldsymbol{x}_4^\top \boldsymbol{x}_1$ $\alpha_{4,2} = \boldsymbol{x}_4^\top \boldsymbol{x}_2$ $\alpha_{4,3} = \boldsymbol{x}_4^\top \boldsymbol{x}_3$ $\alpha_{4,4} = \boldsymbol{x}_4^\top \boldsymbol{x}_4$ $\alpha_{4,5} = \boldsymbol{x}_4^\top \boldsymbol{x}_5$

Embedded Tokens $\boldsymbol{x}_1 \in R^d$ $\boldsymbol{x}_2 \in R^d$ $\boldsymbol{x}_3 \in R^d$ $\boldsymbol{x}_4 \in R^d$ $\boldsymbol{x}_5 \in R^d$

Self-Attention

Updated feature $\boldsymbol{x}'_4 = \boxed{W^O}\left(\alpha'_{4,1}\, v_1 + \alpha'_{4,2}\, v_2 + \alpha'_{4,3}\, v_3 + \alpha'_{4,4}\, v_4 + \alpha'_{4,5}\, v_5 \right)$

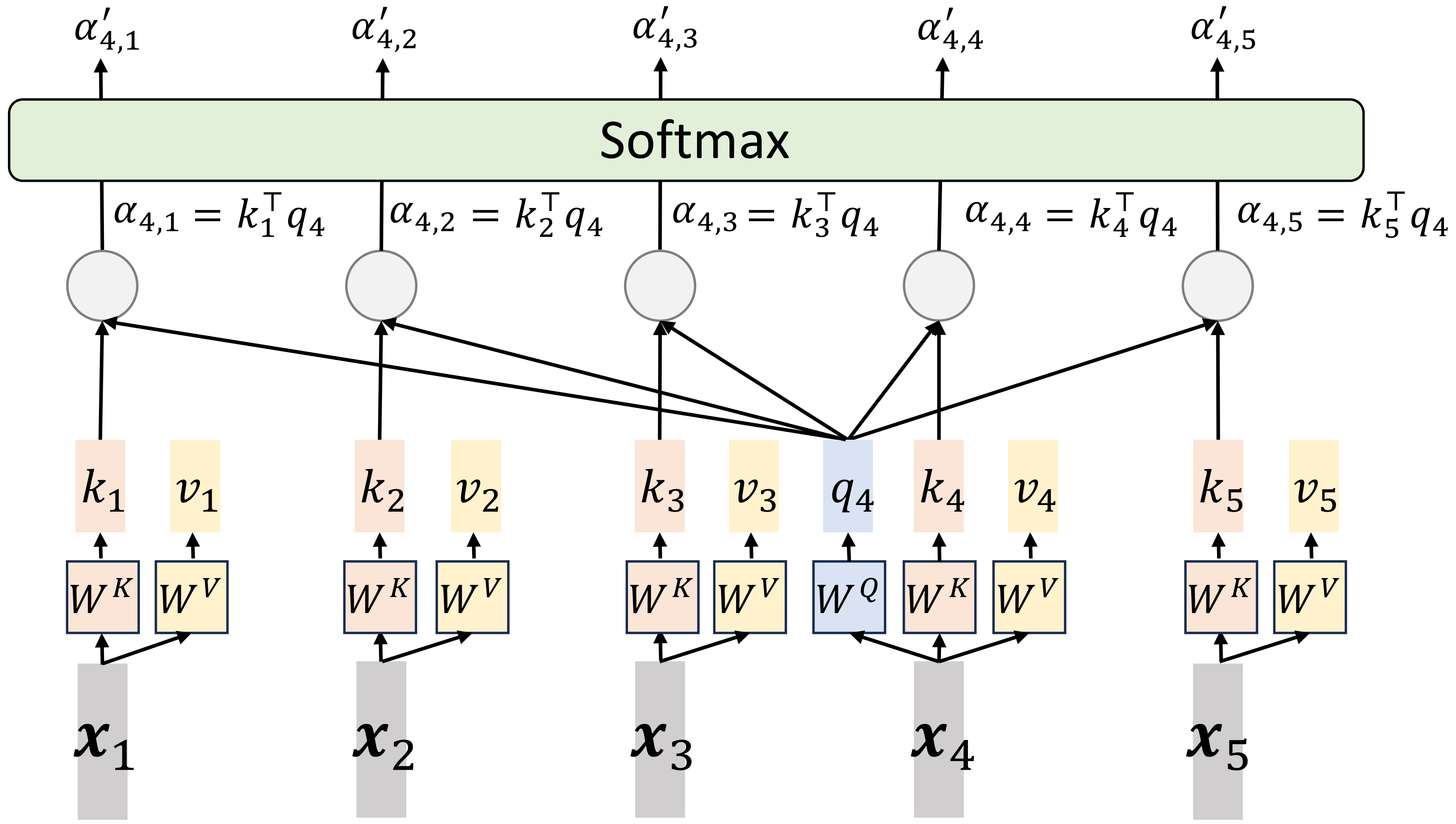
$$= \sum_i \alpha'_{4,1} \left(\boxed{W^O} \boxed{W^V} \right) \boldsymbol{x}_i$$

$\boxed{W^O} \in R^{d \times d_v}$

$\boxed{W^Q} \in R^{d_k \times d}$

$\boxed{W^K} \in R^{d_k \times d}$

$\boxed{W^V} \in R^{d_v \times d}$



Slide by Jia-Bin Huang, University of Maryland College Park

Self-Attention

Single-head attention

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

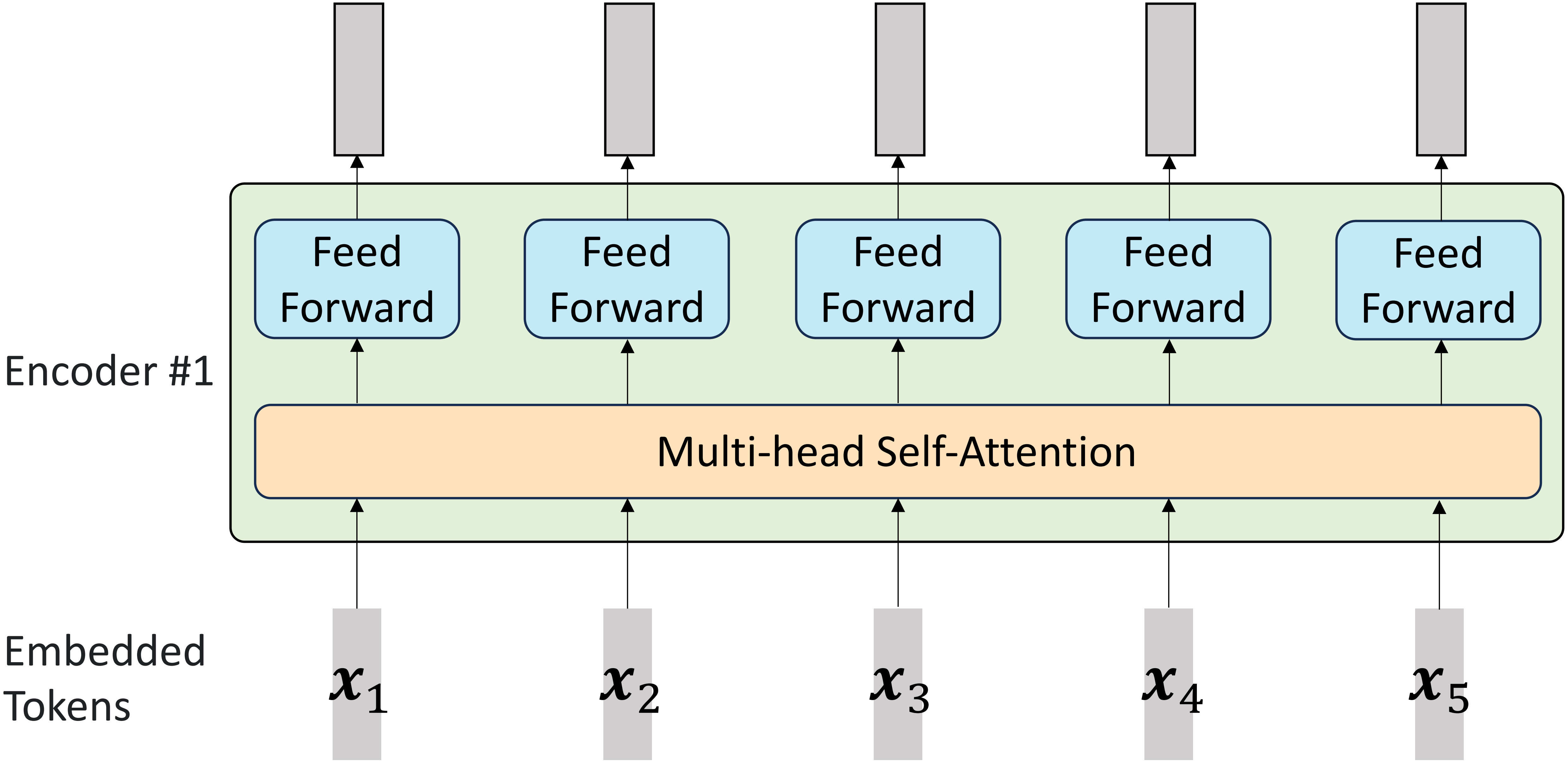
$$Q = W^Q\ \boldsymbol{x}_1\ \boldsymbol{x}_2\ \boldsymbol{x}_3\ \boldsymbol{x}_4\ \boldsymbol{x}_5$$

$$K = W^K\ \boldsymbol{x}_1\ \boldsymbol{x}_2\ \boldsymbol{x}_3\ \boldsymbol{x}_4\ \boldsymbol{x}_5$$

$$V = W^V\ \boldsymbol{x}_1\ \boldsymbol{x}_2\ \boldsymbol{x}_3\ \boldsymbol{x}_4\ \boldsymbol{x}_5$$

Slide by Jia-Bin Huang, University of Maryland College Park

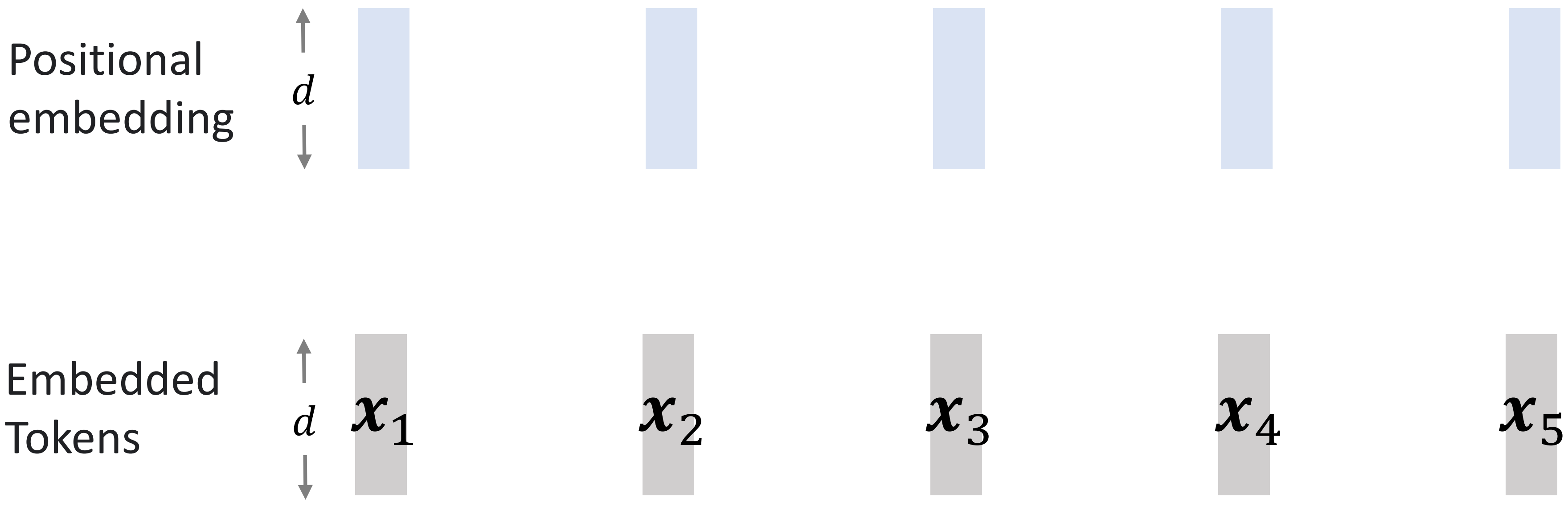
MLPs



Slide by Jia-Bin Huang, University of Maryland College Park

Positional Encoding

	Position k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Dimension	2^3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	← Slow oscillating
	2^2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	
	2^1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
	2^0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	← Fast oscillating



Slide by Jia-Bin Huang, University of Maryland College Park

Positional Encoding

$$\mathbf{S}_i = \left[\sin(\omega_1 i), \cos(\omega_1 i), \sin(\omega_2 i), \cos(\omega_2 i), \dots, \sin(\omega_{e/2} i), \cos(\omega_{e/2} i)\right]$$

Position k

Angular frequency

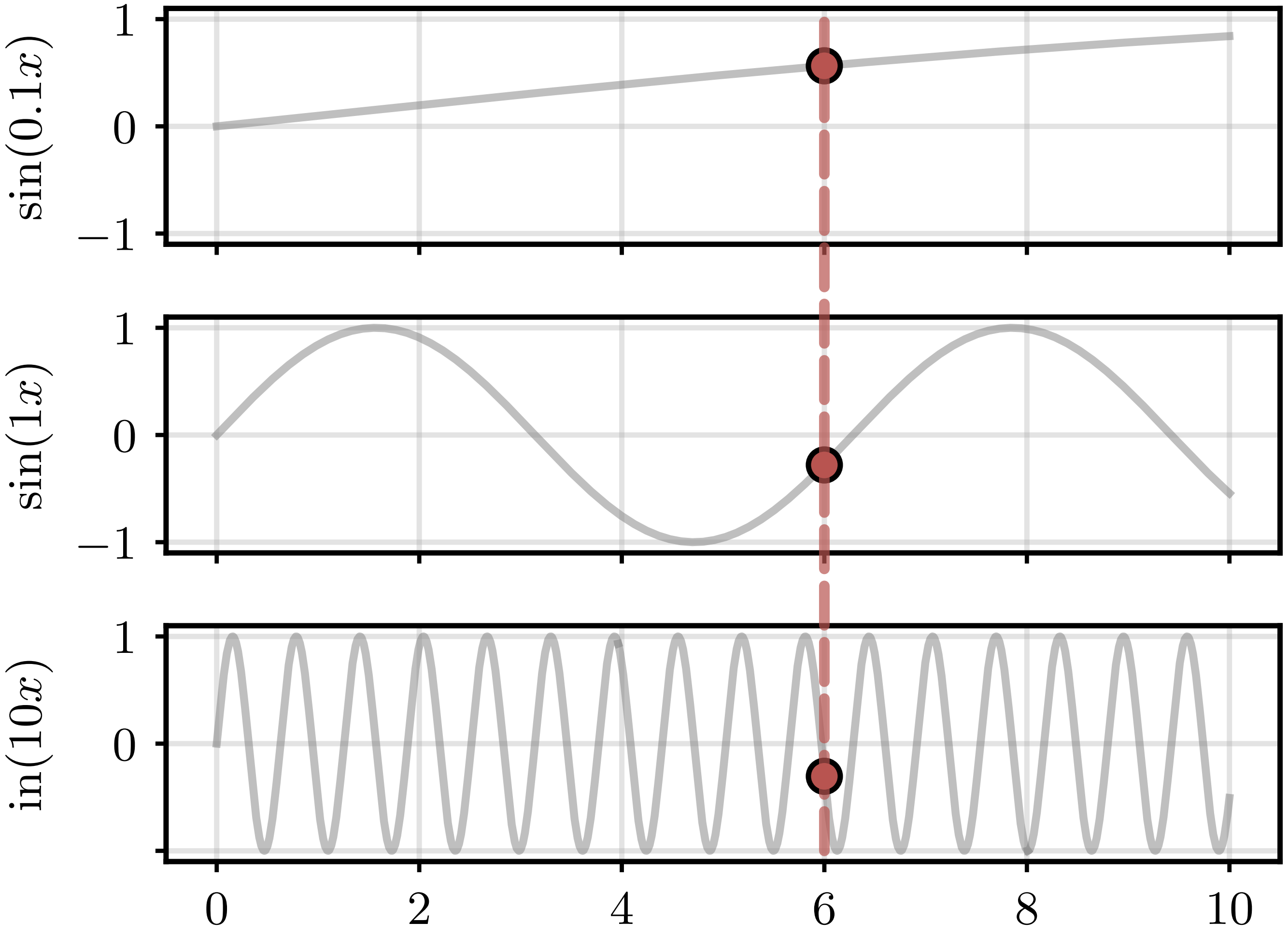
$w_i = N^{-2i/d}$

$N = 100{,}000$

$$d \quad \begin{bmatrix} \sin(w_0 k) \\ \cos(w_0 k) \\ \sin(w_1 k) \\ \cos(w_1 k) \\ \vdots \\ \vdots \\ \sin\left(w_{\frac{d}{2}-1} k\right) \\ \cos\left(w_{\frac{d}{2}-1} k\right) \end{bmatrix}$$

← Fast oscillating

← Slow oscillating



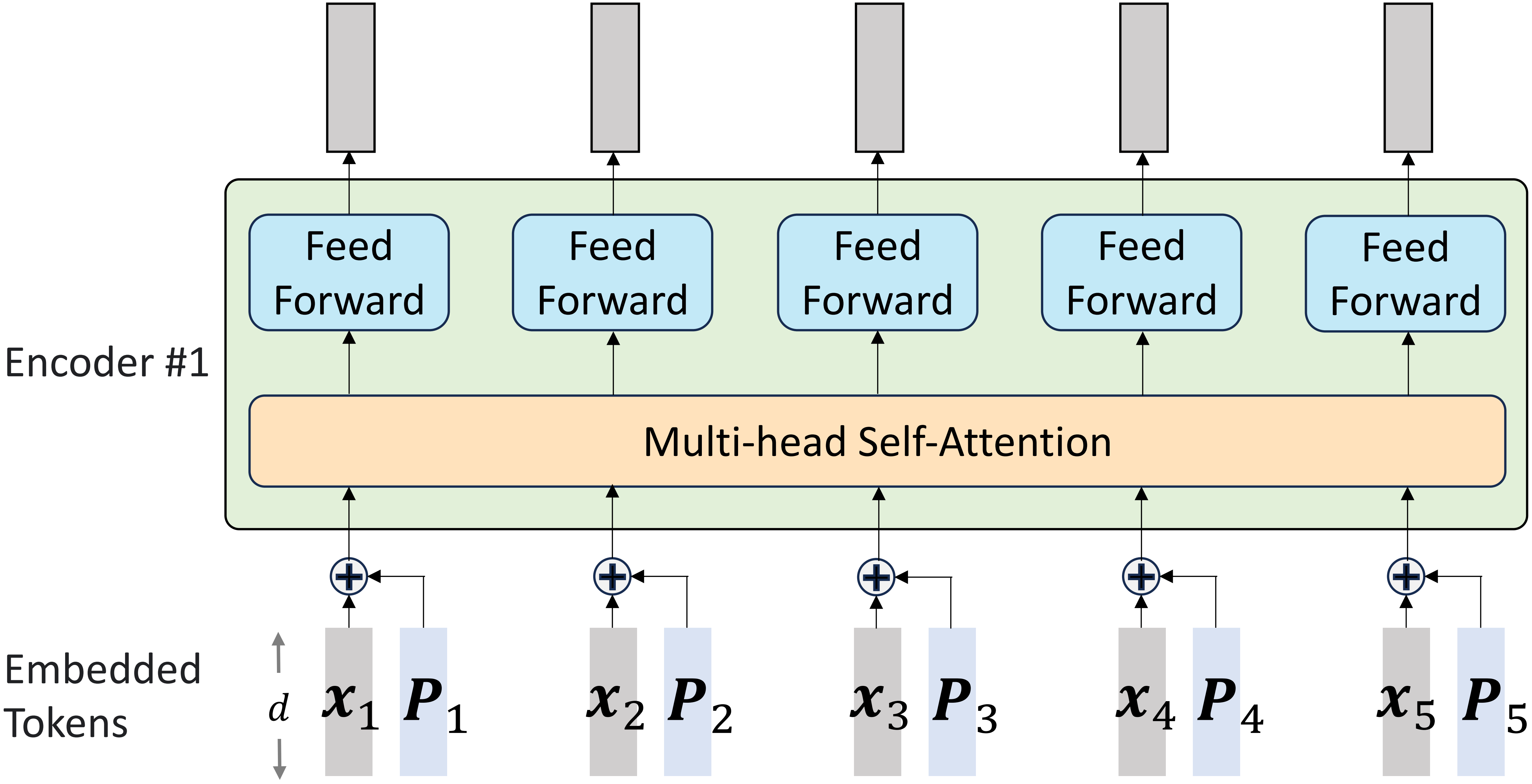
👍 **Normalized Range**

👍 **Unique identifier, unlimited length**

👍 **Relative positions as linear transform**

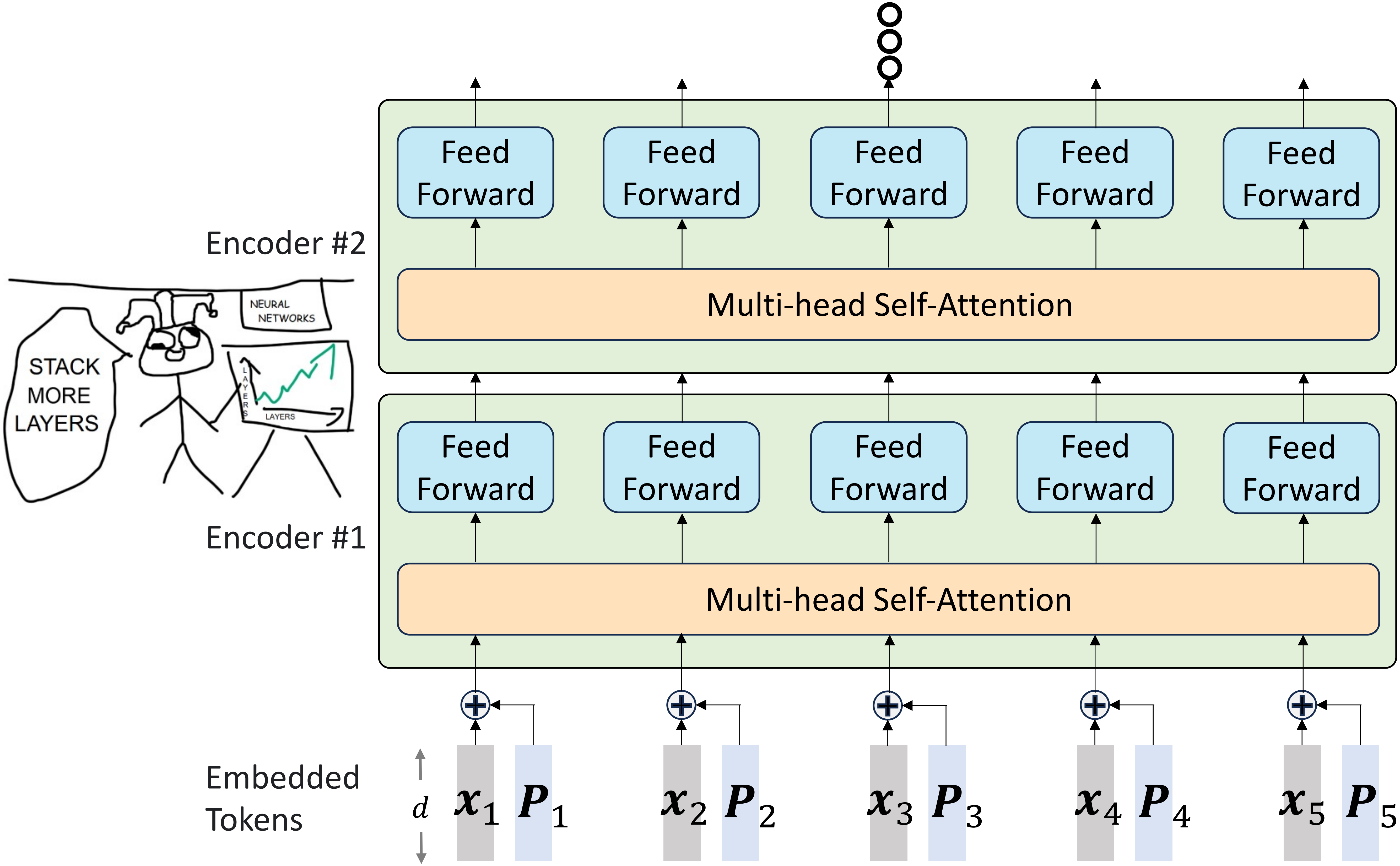
Slide by Jia-Bin Huang, University of Maryland College Park

Positional Encoding



Slide by Jia-Bin Huang, University of Maryland College Park

Transformer Encoder


Encoder #2
Feed Forward
Feed Forward
Feed Forward
Feed Forward
Feed Forward
Multi-head Self-Attention
STACK MORE LAYERS
NEURAL NETWORKS
LAYERS
LAYERS
Encoder #1
Feed Forward
Feed Forward
Feed Forward
Feed Forward
Feed Forward
Multi-head Self-Attention
Embedded Tokens
d
$\boldsymbol{x}_1$ $\boldsymbol{P}_1$
$\boldsymbol{x}_2$ $\boldsymbol{P}_2$
$\boldsymbol{x}_3$ $\boldsymbol{P}_3$
$\boldsymbol{x}_4$ $\boldsymbol{P}_4$
$\boldsymbol{x}_5$ $\boldsymbol{P}_5$

Slide by Jia-Bin Huang, University of Maryland College Park

Transformer Encoder

Residual connection

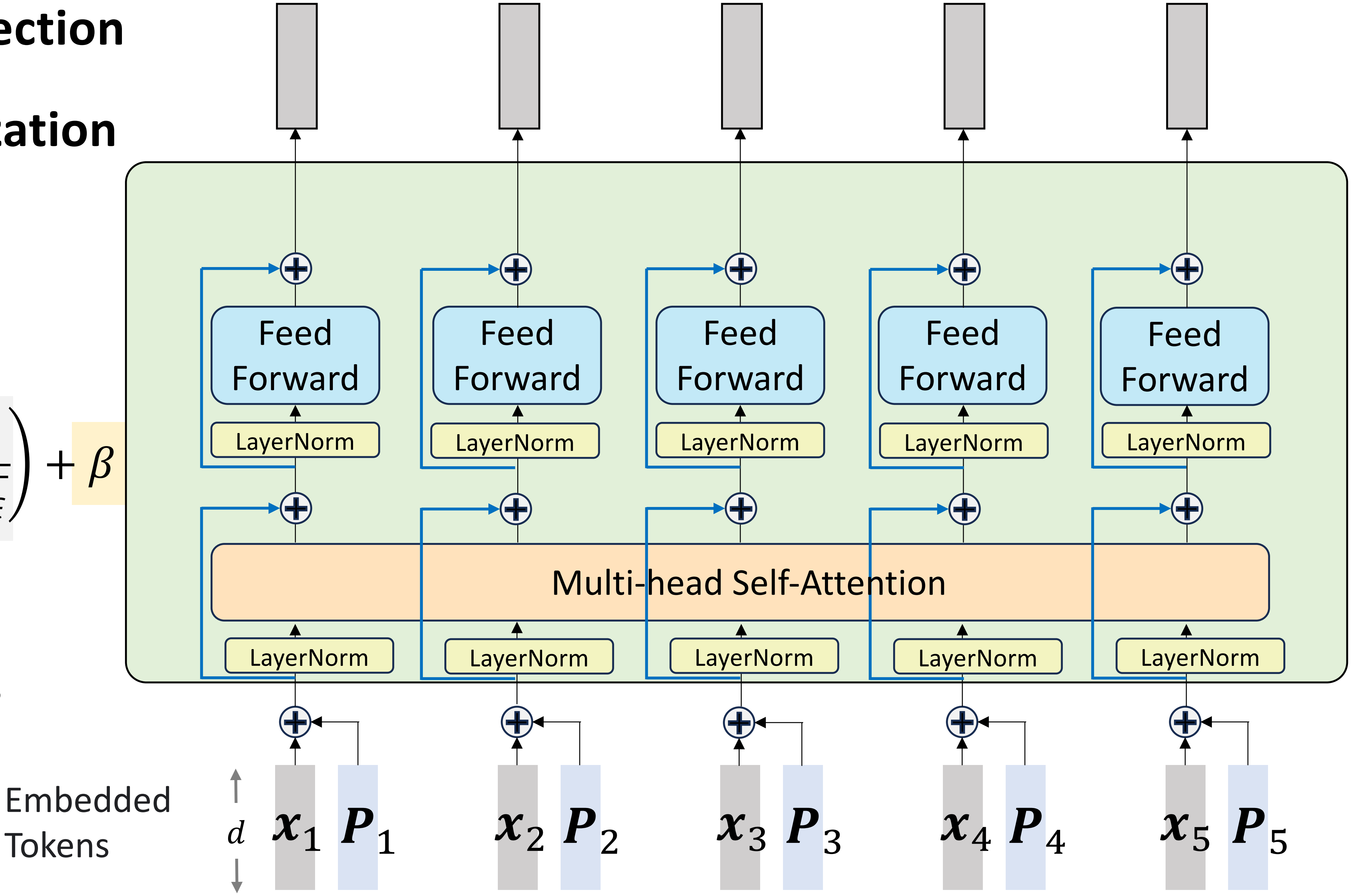
Layer normalization

$$\text{LayerNorm}(\boldsymbol{x}) =$$

$$\gamma\left(\frac{\boldsymbol{x} - \text{mean}(\boldsymbol{x})}{\sqrt{\text{Variance}(\mathbf{x}) + \epsilon}}\right) + \beta$$

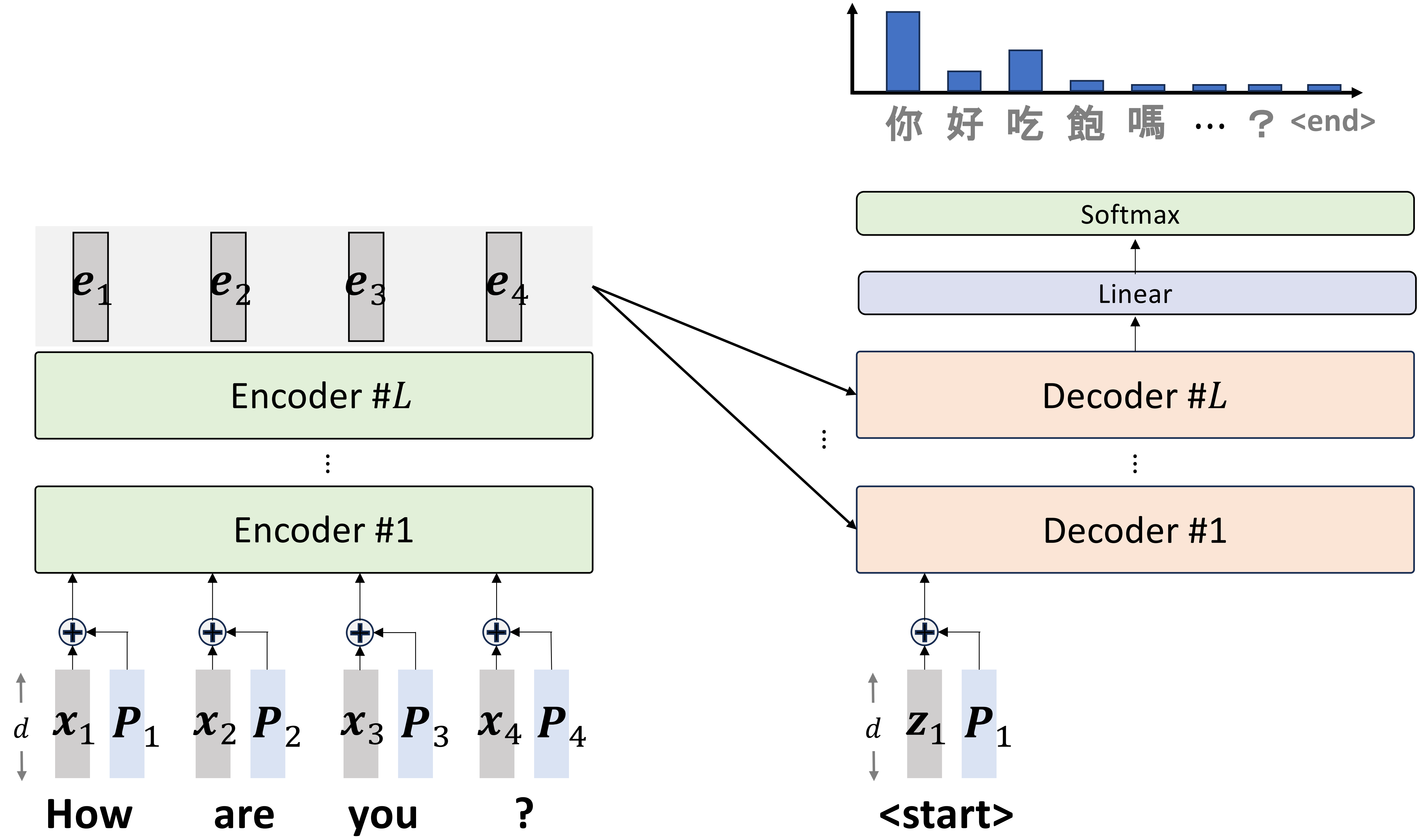
$\gamma, \beta \in R$

Learnable parameters


Feed Forward
LayerNorm
Multi-head Self-Attention
Embedded Tokens
d
$\boldsymbol{x}_1$ $\boldsymbol{P}_1$
$\boldsymbol{x}_2$ $\boldsymbol{P}_2$
$\boldsymbol{x}_3$ $\boldsymbol{P}_3$
$\boldsymbol{x}_4$ $\boldsymbol{P}_4$
$\boldsymbol{x}_5$ $\boldsymbol{P}_5$

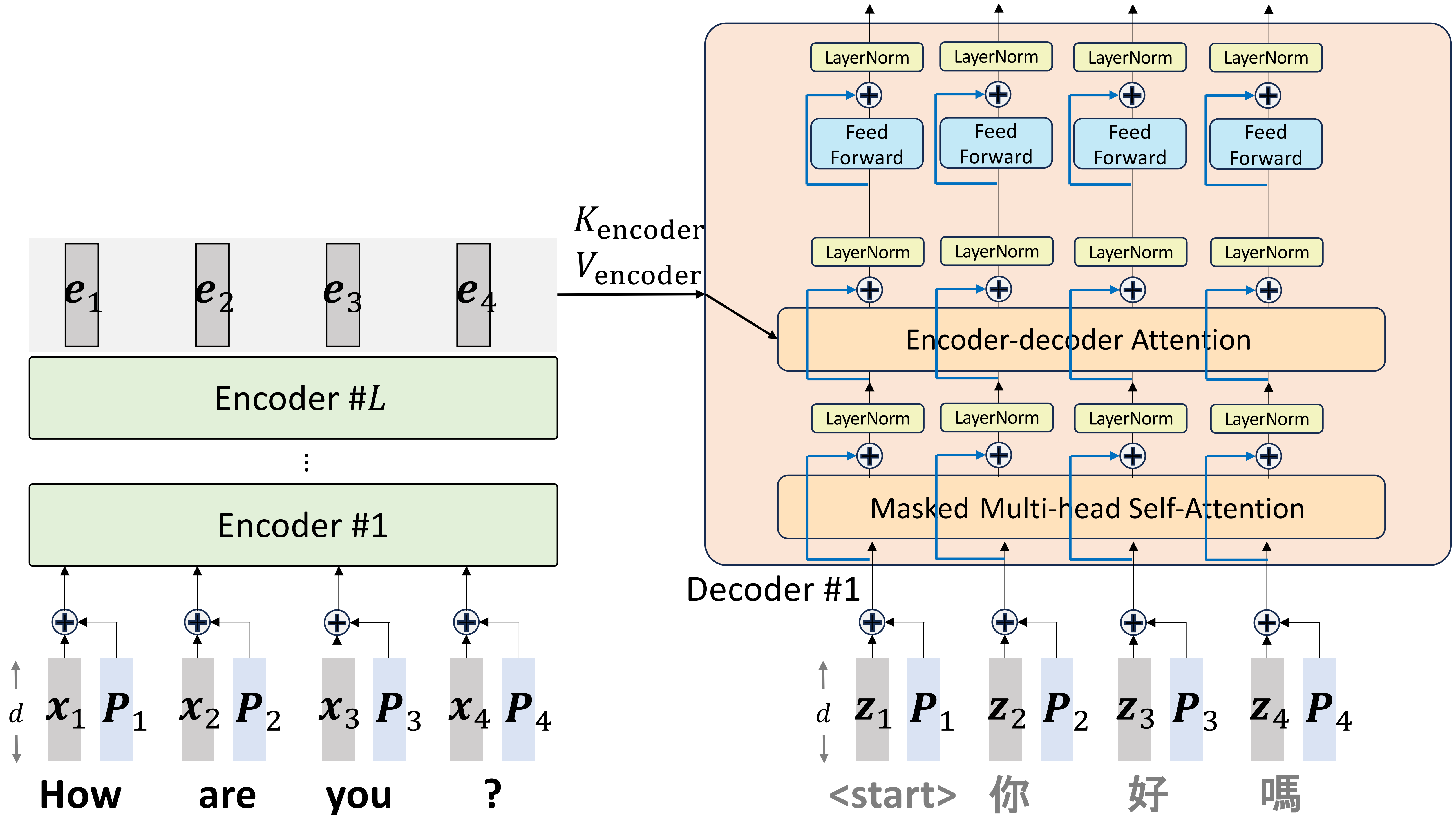
Slide by Jia-Bin Huang, University of Maryland College Park

Decoders



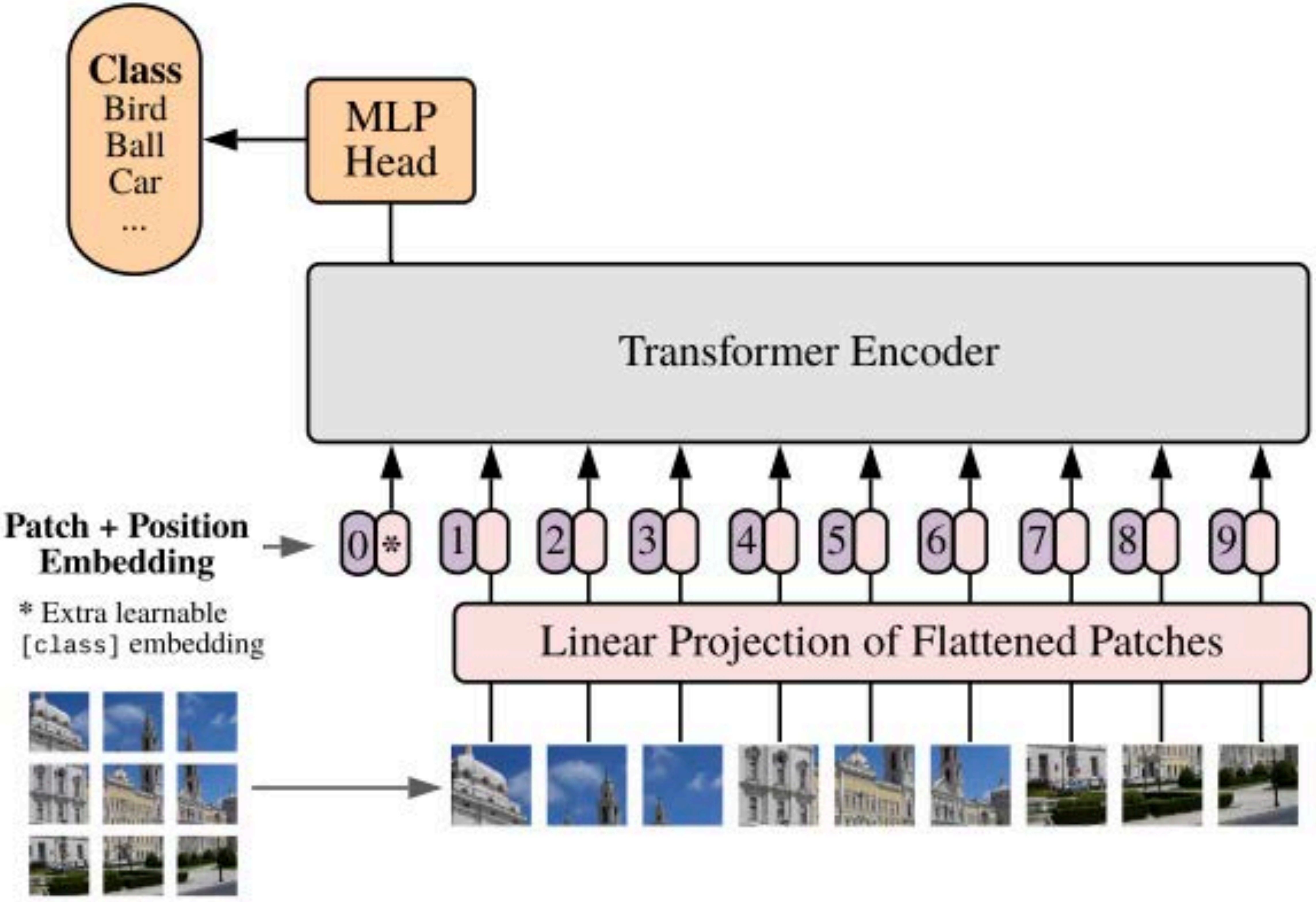
Slide by Jia-Bin Huang, University of Maryland College Park

Decoders


LayerNorm
Feed Forward
LayerNorm
Encoder-decoder Attention
LayerNorm
Masked Multi-head Self-Attention
Decoder #1
K_{encoder}
V_{encoder}
$\boldsymbol{e}_1$
$\boldsymbol{e}_2$
$\boldsymbol{e}_3$
$\boldsymbol{e}_4$
Encoder #L
⋮
Encoder #1
d
$\boldsymbol{x}_1$ $\boldsymbol{P}_1$
$\boldsymbol{x}_2$ $\boldsymbol{P}_2$
$\boldsymbol{x}_3$ $\boldsymbol{P}_3$
$\boldsymbol{x}_4$ $\boldsymbol{P}_4$
How
are
you
?
d
$\boldsymbol{z}_1$ $\boldsymbol{P}_1$
$\boldsymbol{z}_2$ $\boldsymbol{P}_2$
$\boldsymbol{z}_3$ $\boldsymbol{P}_3$
$\boldsymbol{z}_4$ $\boldsymbol{P}_4$
<start>
你
好
嗎

Slide by Jia-Bin Huang, University of Maryland College Park

Vision Transformers

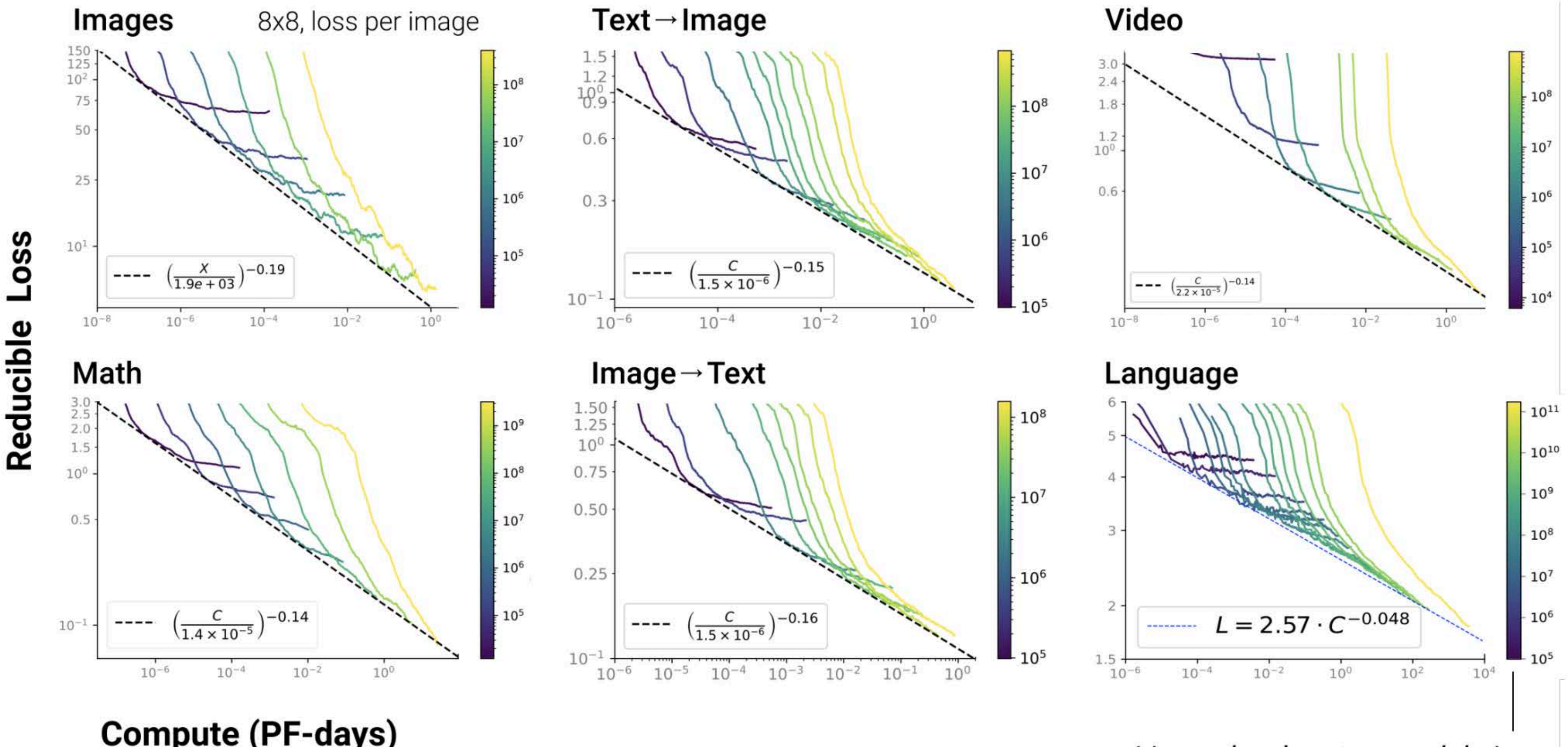


Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
0
*
1
2
3
4
5
6
7
8
9
* Extra learnable [class] embedding
Linear Projection of Flattened Patches

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Advantages & Limitations

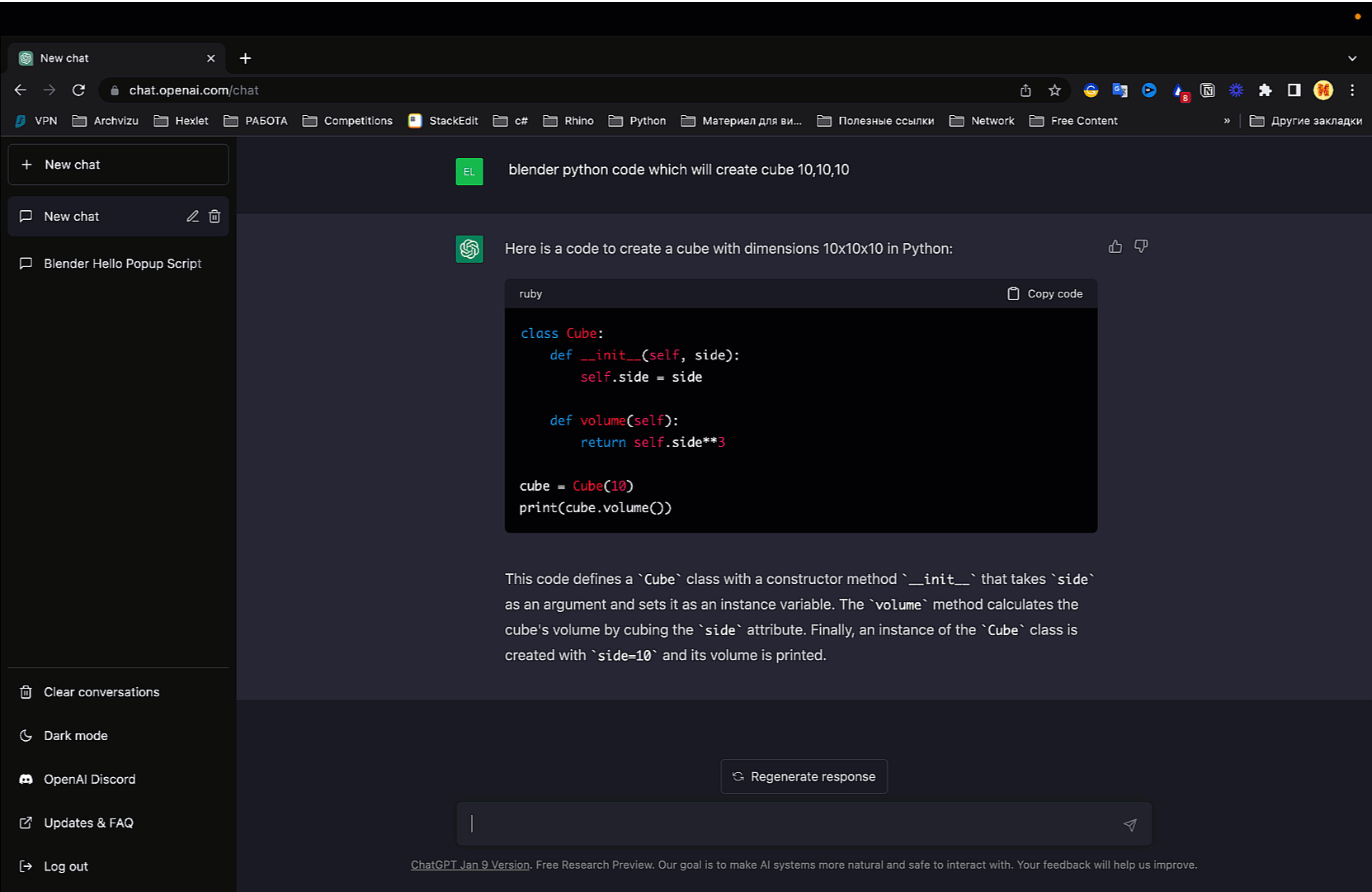
- Parallel processing
- Better at capturing long-range dependencies
- No vanishing gradient problem
- Scales well with data and compute
- Modality-agnostic



Line color denotes model size

- Self-attention cost and memory scales quadratically with seq. length
- Needs a lot of data
- Expensive and difficult to train

Generative Pretrained Transformers



blender python code which will create cube 10,10,10

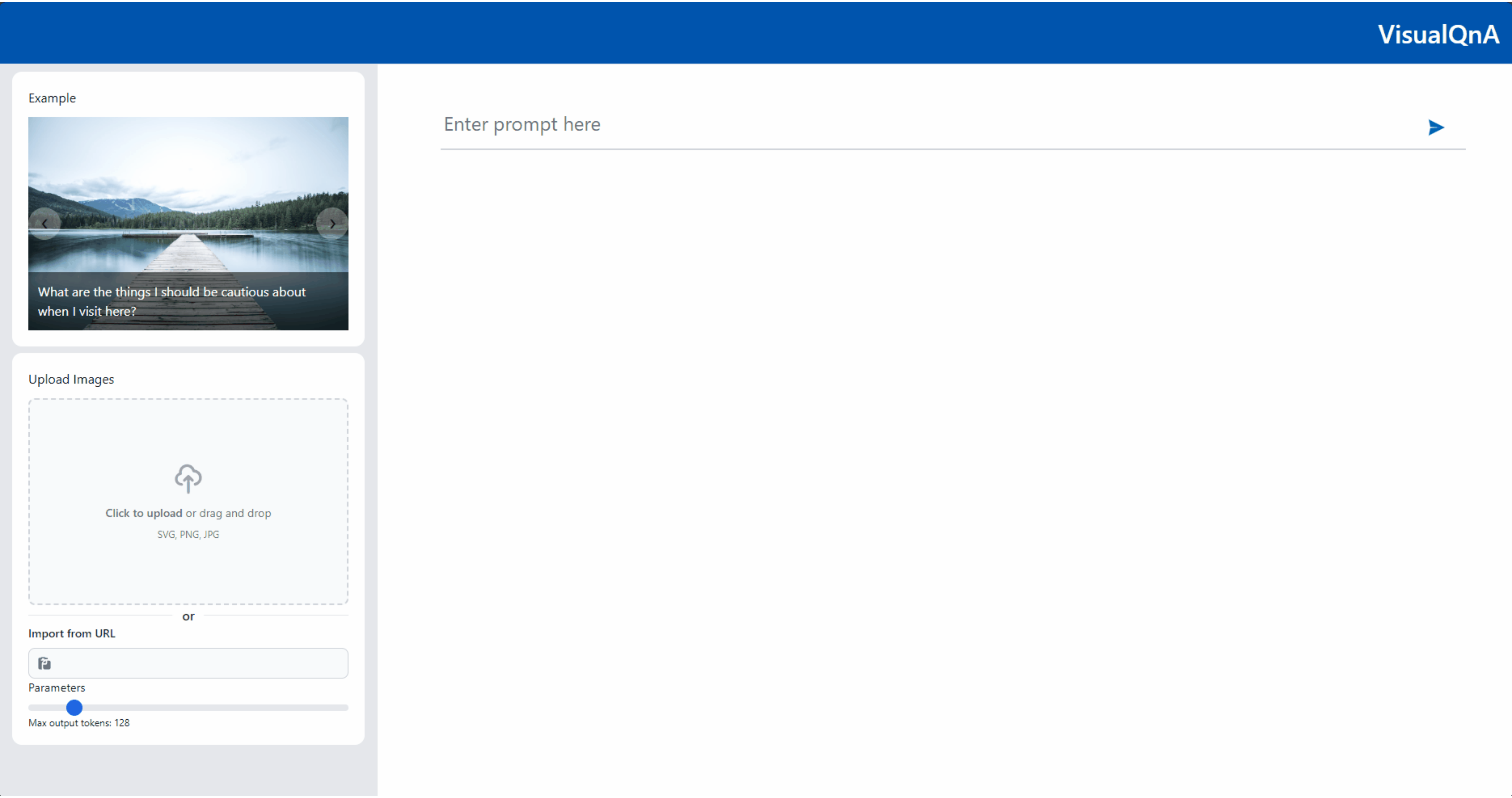
Here is a code to create a cube with dimensions 10x10x10 in Python:

```ruby
class Cube:
    def __init__(self, side):
        self.side = side

    def volume(self):
        return self.side**3

cube = Cube(10)
print(cube.volume())
```

This code defines a `Cube` class with a constructor method `__init__` that takes `side` as an argument and sets it as an instance variable. The `volume` method calculates the cube's volume by cubing the `side` attribute. Finally, an instance of the `Cube` class is created with `side=10` and its volume is printed.

ChatGPT Jan 9 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.

Multi-modal Transformers

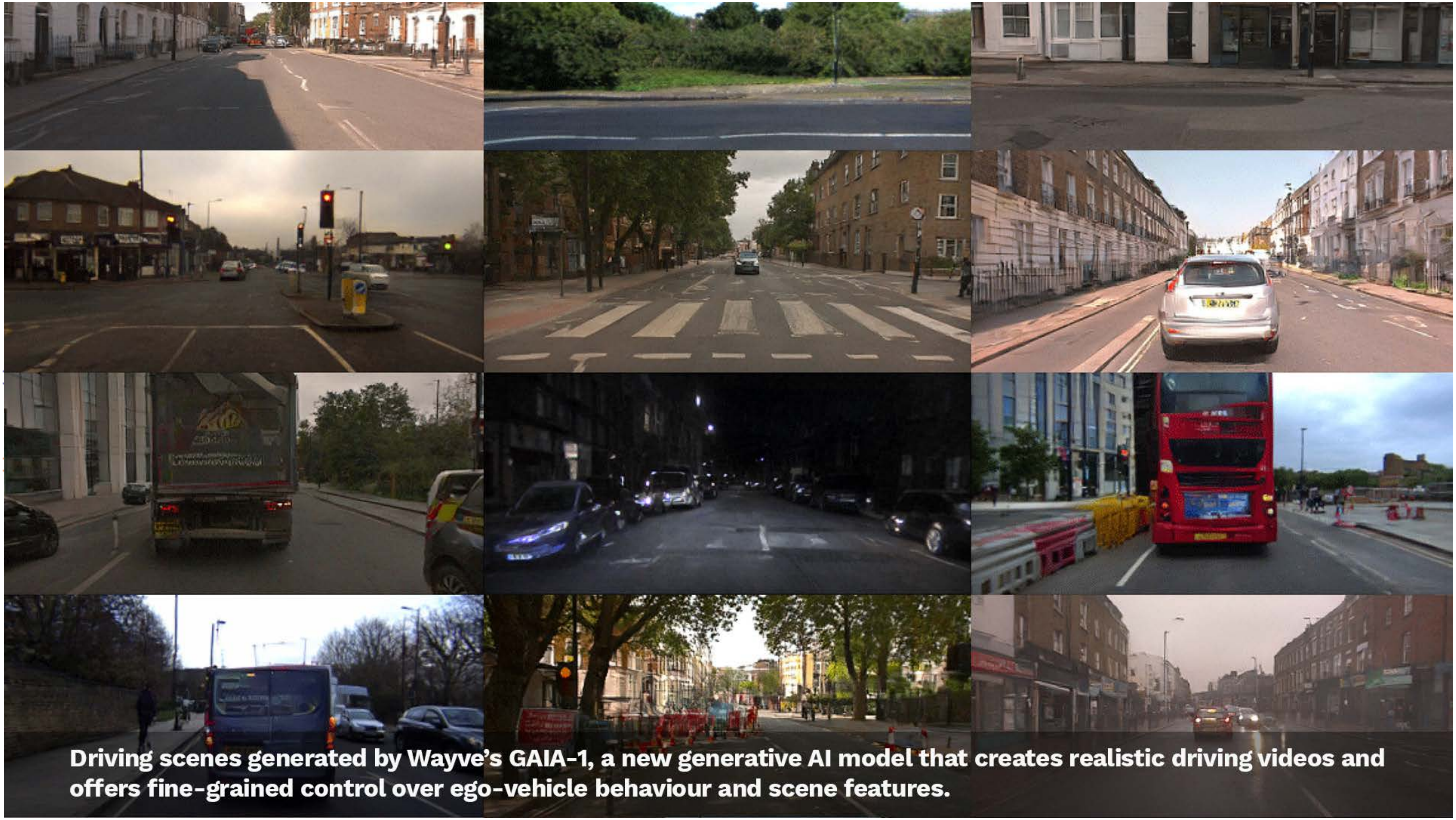


What are the things I should be cautious about when I visit here?

Max output tokens: 128

https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/

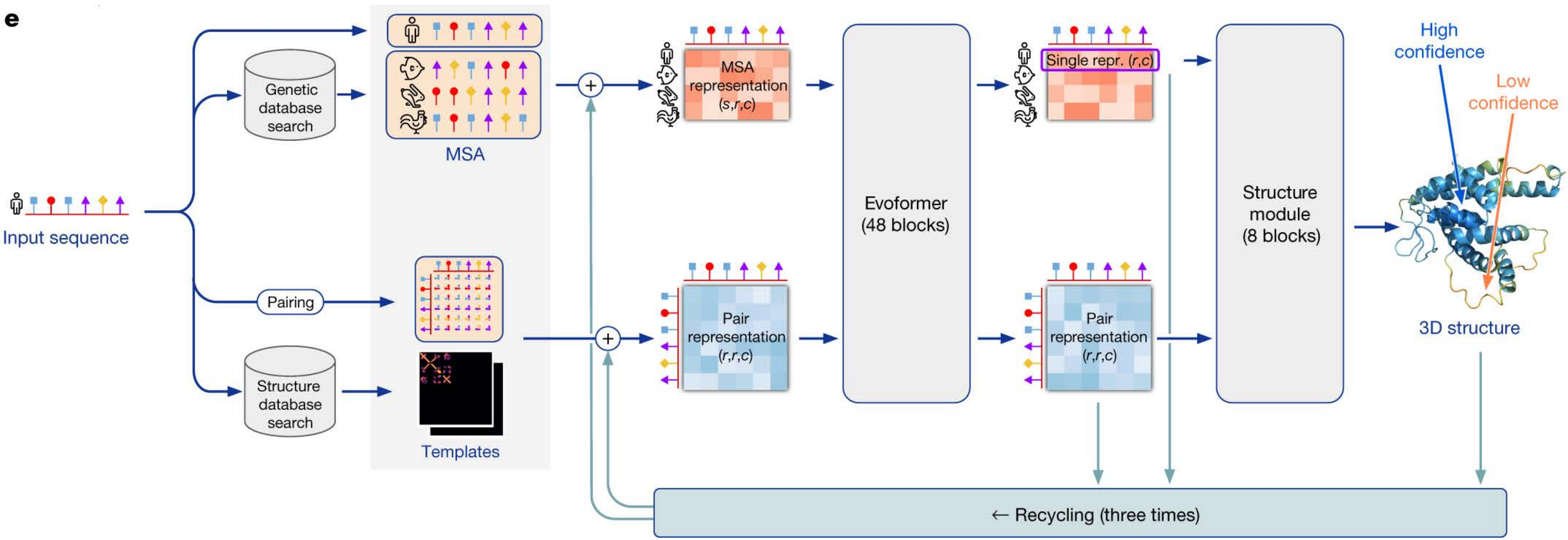
Multi-modal Transformers

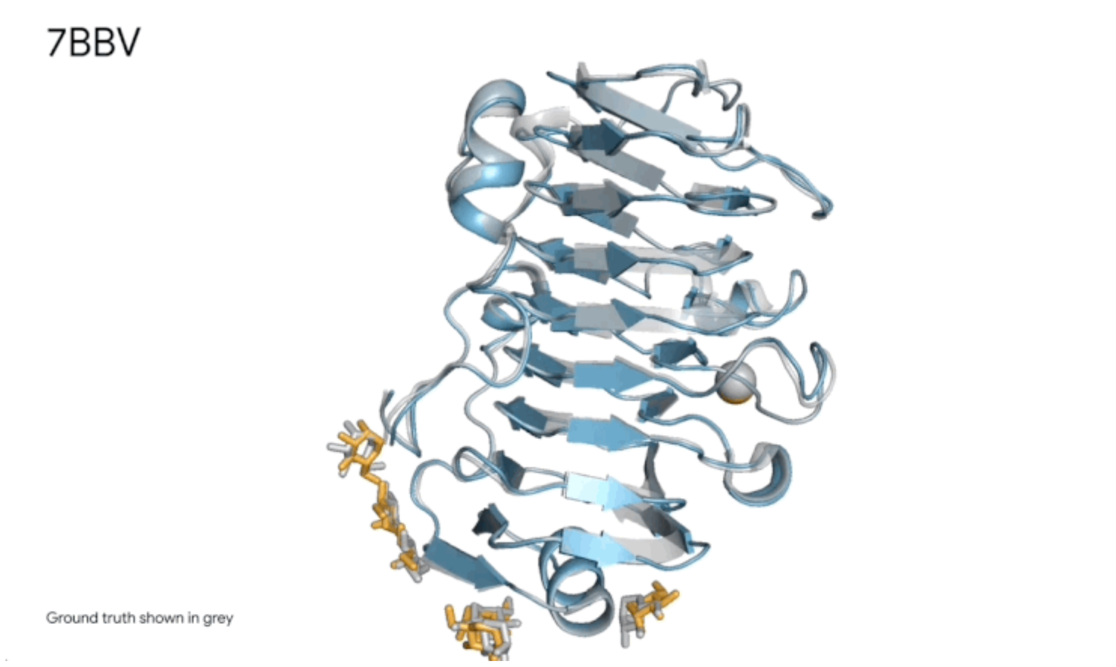


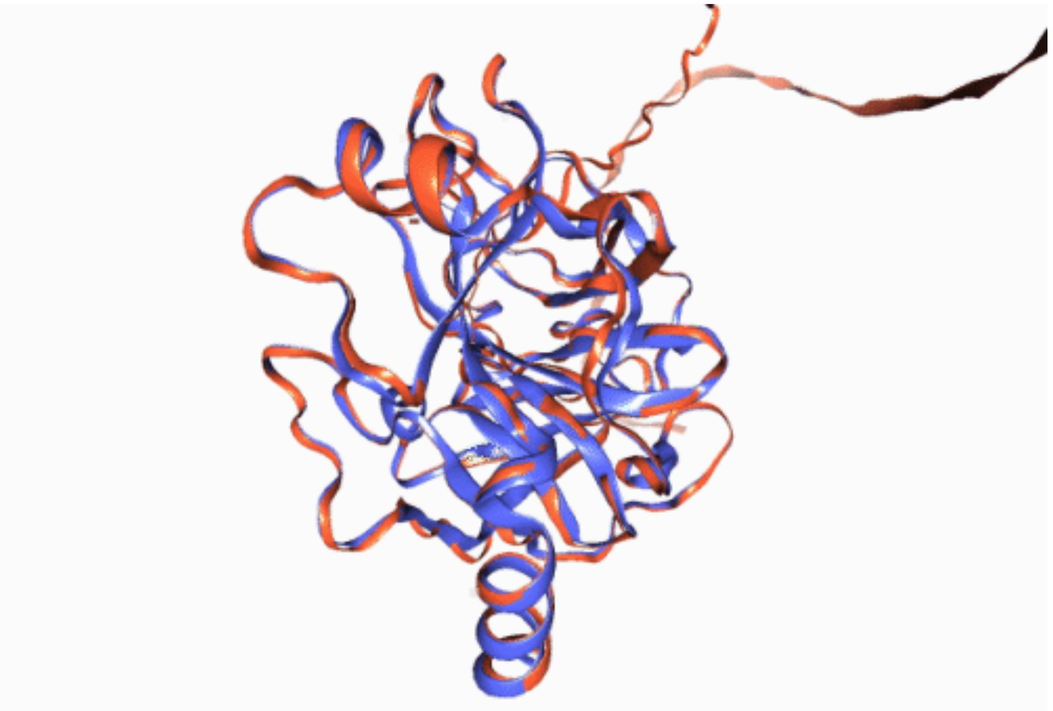
Driving scenes generated by Wayve's GAIA-1, a new generative AI model that creates realistic driving videos and offers fine-grained control over ego-vehicle behaviour and scene features.

https://wayve.ai/thinking/introducing-gaia1/

AlphaFold 3


e
Input sequence
Genetic database search
Pairing
Structure database search
MSA
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)





Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., ... & Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature, 1-3.

UvA Tutorial Notebooks

Build your own GPT from scratch