Strojové učení II

Sequences

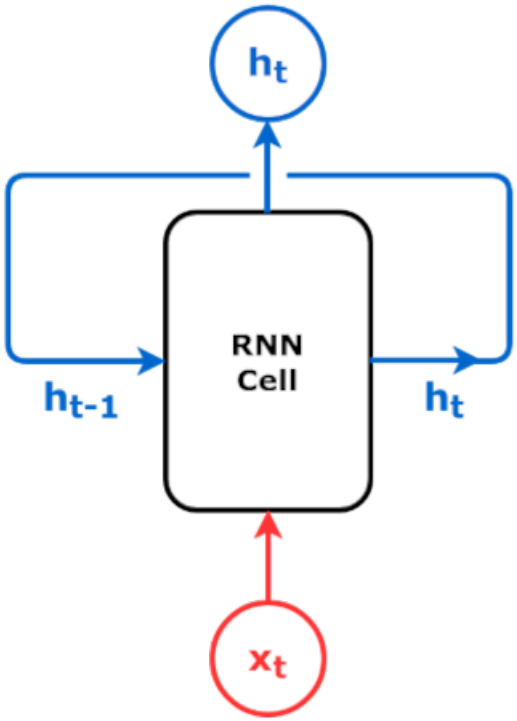
Institute of Information Theory and Automation of the AS CR

Sequences

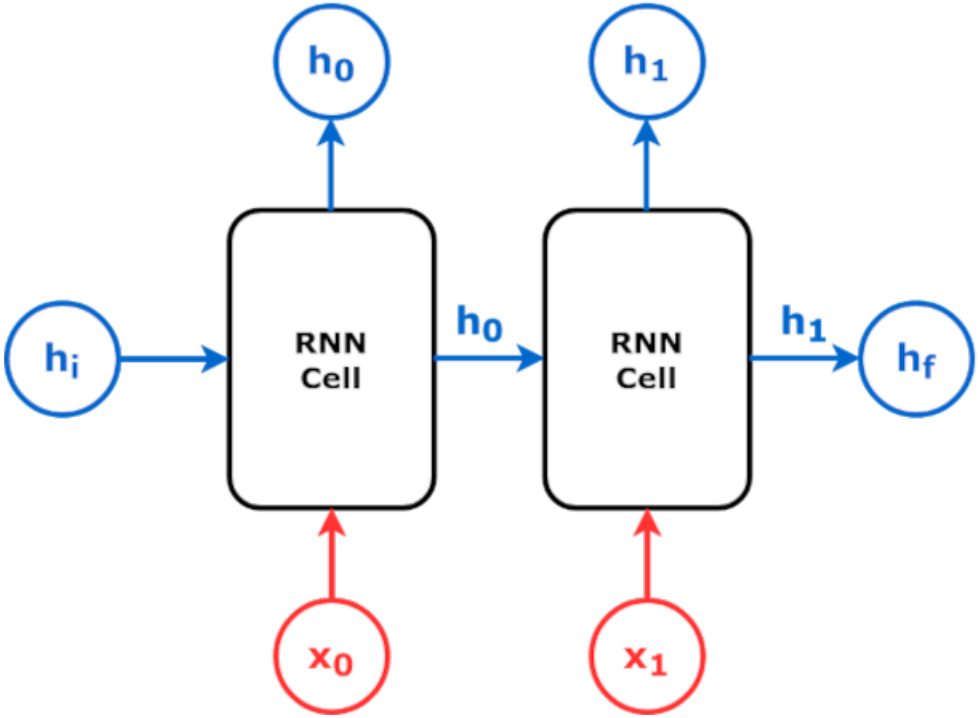
- Input:
  - sequence = an **ordered** set of data points
- Output:
  - single label: RNN
  - sequence (of different length): encoder-decoder model
- Application: time series, Natural Language Processing (NLP)

Recurrent Neural Network (RNN)

- Basic RNN

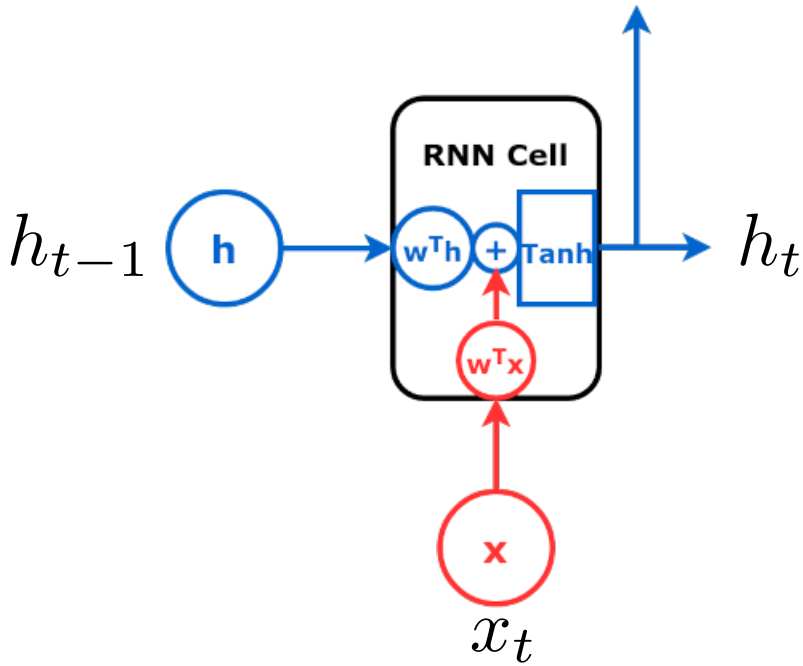


- Unfolded RNN



Godoy, Deep Learning with Pytorch, 2019

RNN cell



$$t_h = W_h h_{t-1} + b_h$$

$$h_t = \tanh(t_h + t_x)$$

$$t_x = W_i x_t + b_i$$

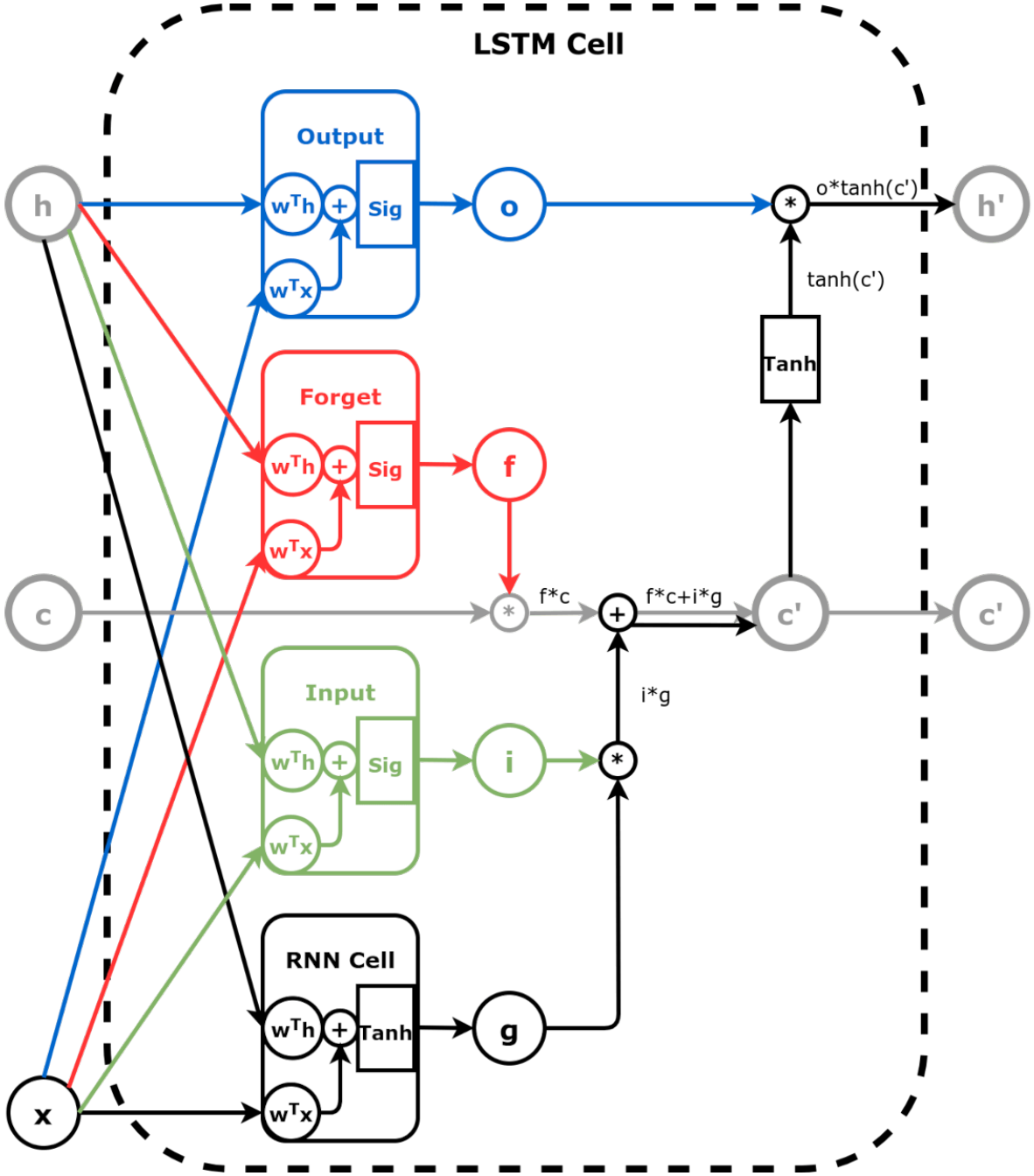
Problem: Long-term dependencies

Godoy, Deep Learning with Pytorch, 2019

Multiple Time Scales

- Long-term dependencies are weak in the basic RNN cell.
- Solution:
  - Skip connections through time
  - Long Short-Term Memory (LSTM)
  - Gated Recurrent Unit (GRU)

LSTM

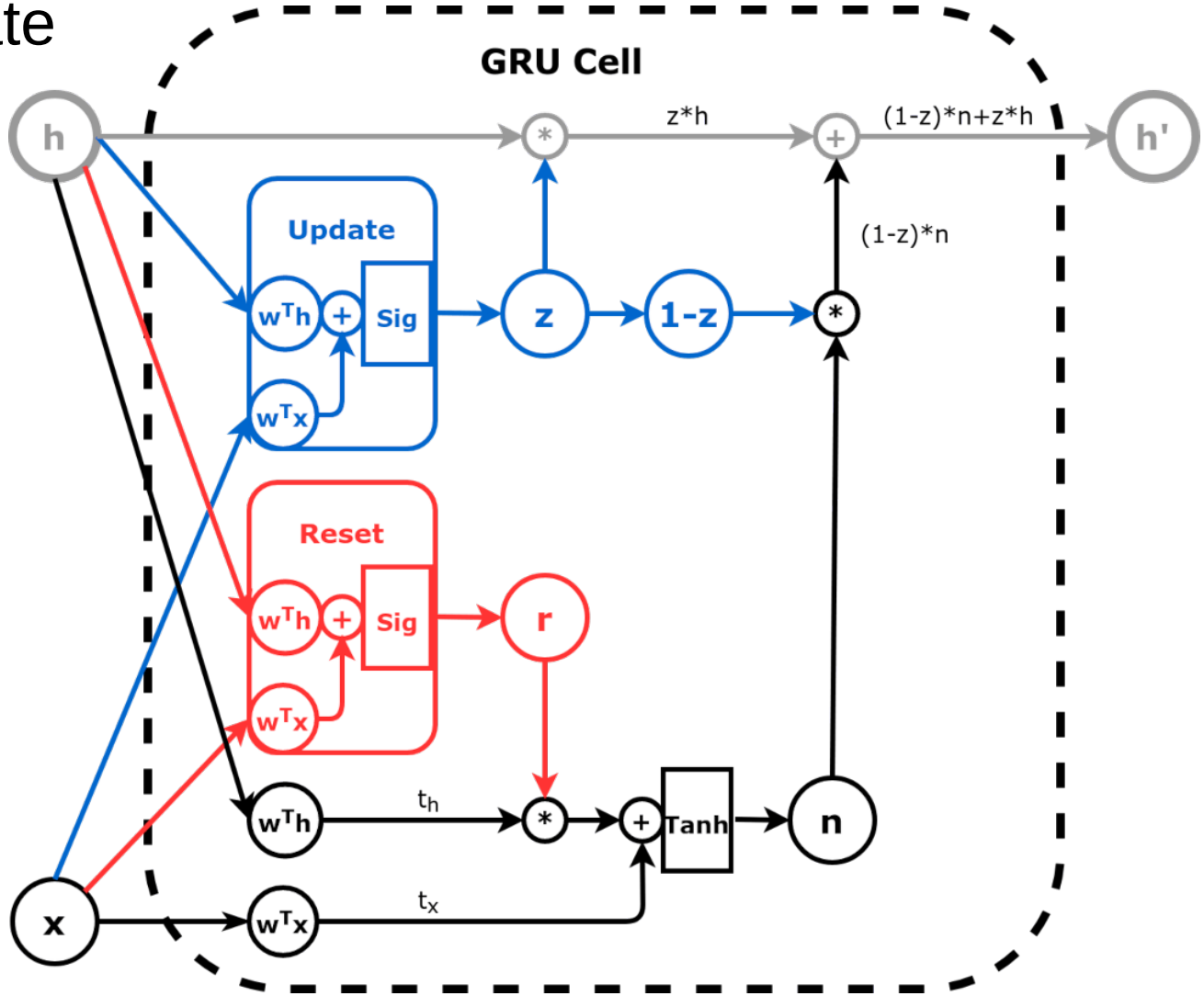


Discuss cases:
i=1, f = 0;
i=0, f = 1;

Godoy, Deep Learning with Pytorch, 2019

GRU

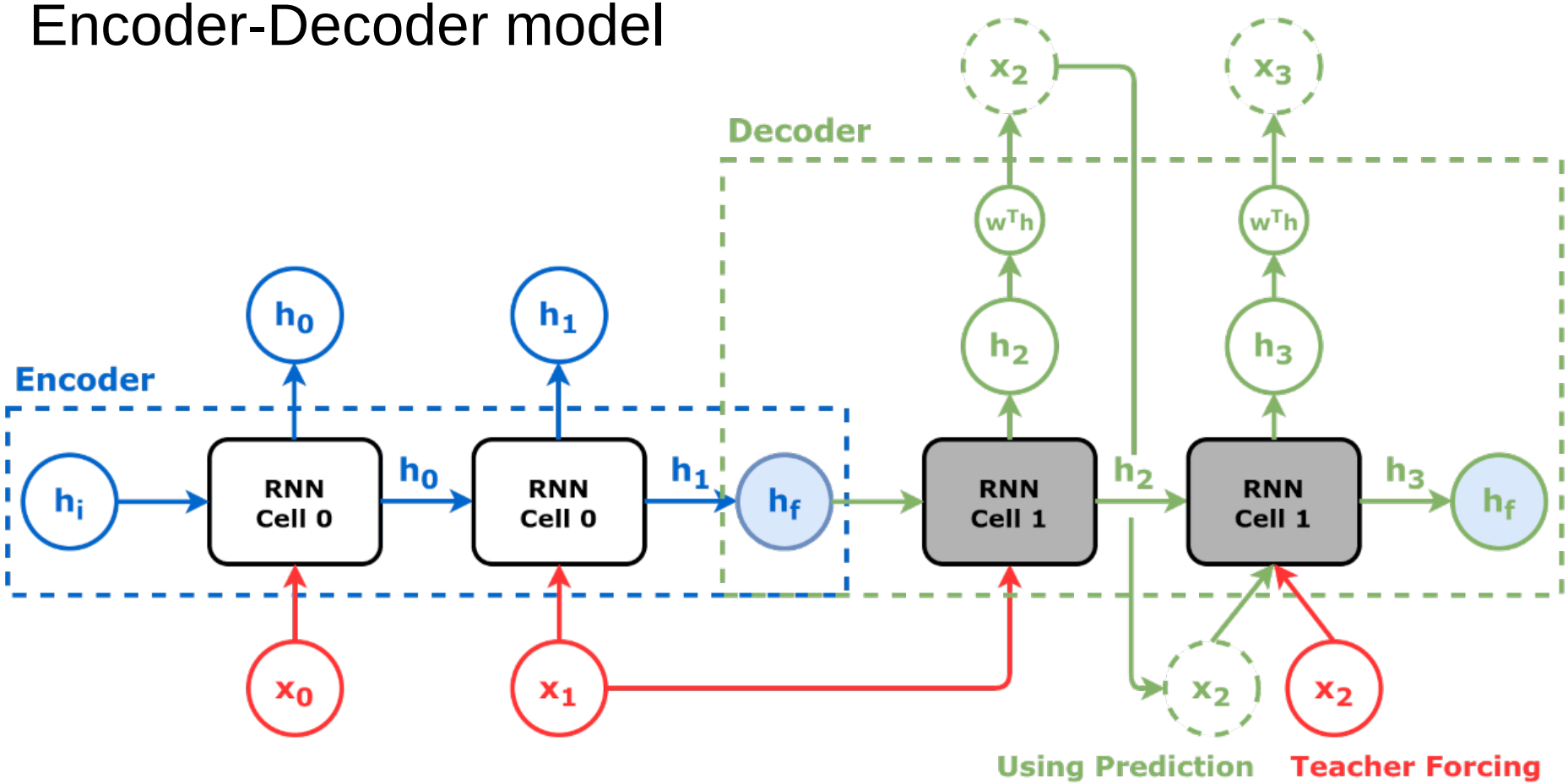
- No input gate
- No cell state



Godoy, Deep Learning with Pytorch, 2019

Sequence-to-Sequence

- Encoder-Decoder model



- The European economic zone → la zona económica europea
  - Word order
  - the → el / la: To which word do we pay **attention**?

Godoy, Deep Learning with Pytorch, 2019

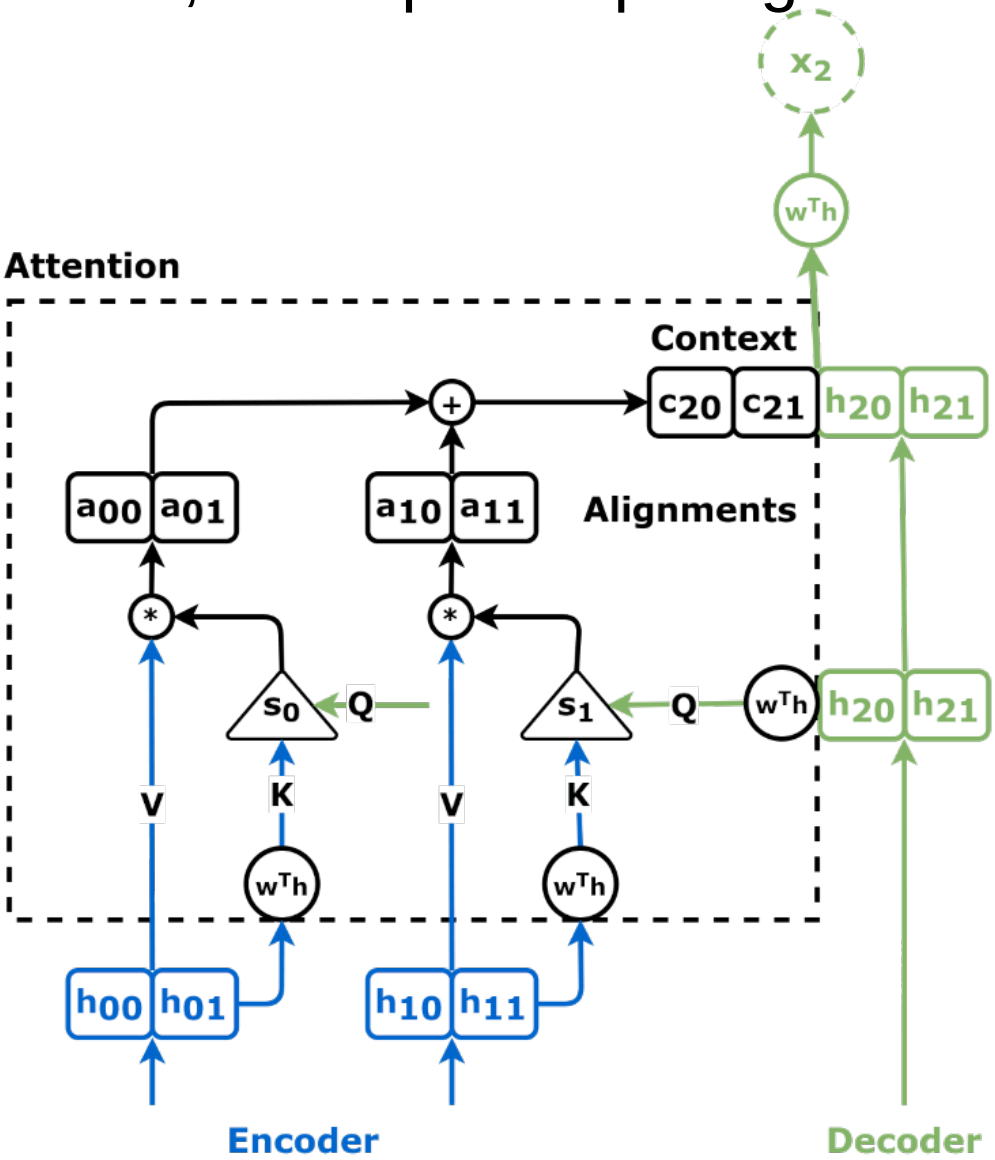
Attention

- V… values [D x L], D… hidden dim., L… input seq. length
- K… keys [D x L]
- q… query

$$c = V\left(K^T q\right)_{\text{ssmax}}$$

$$c = \sum_i \text{softmax}\left(\frac{\langle q, k_i \rangle}{\sqrt{D}}\right) v_i$$

What is D and L in the figure?



Godoy, Deep Learning with Pytorch, 2019

Self-Attention

- Forget RNN!

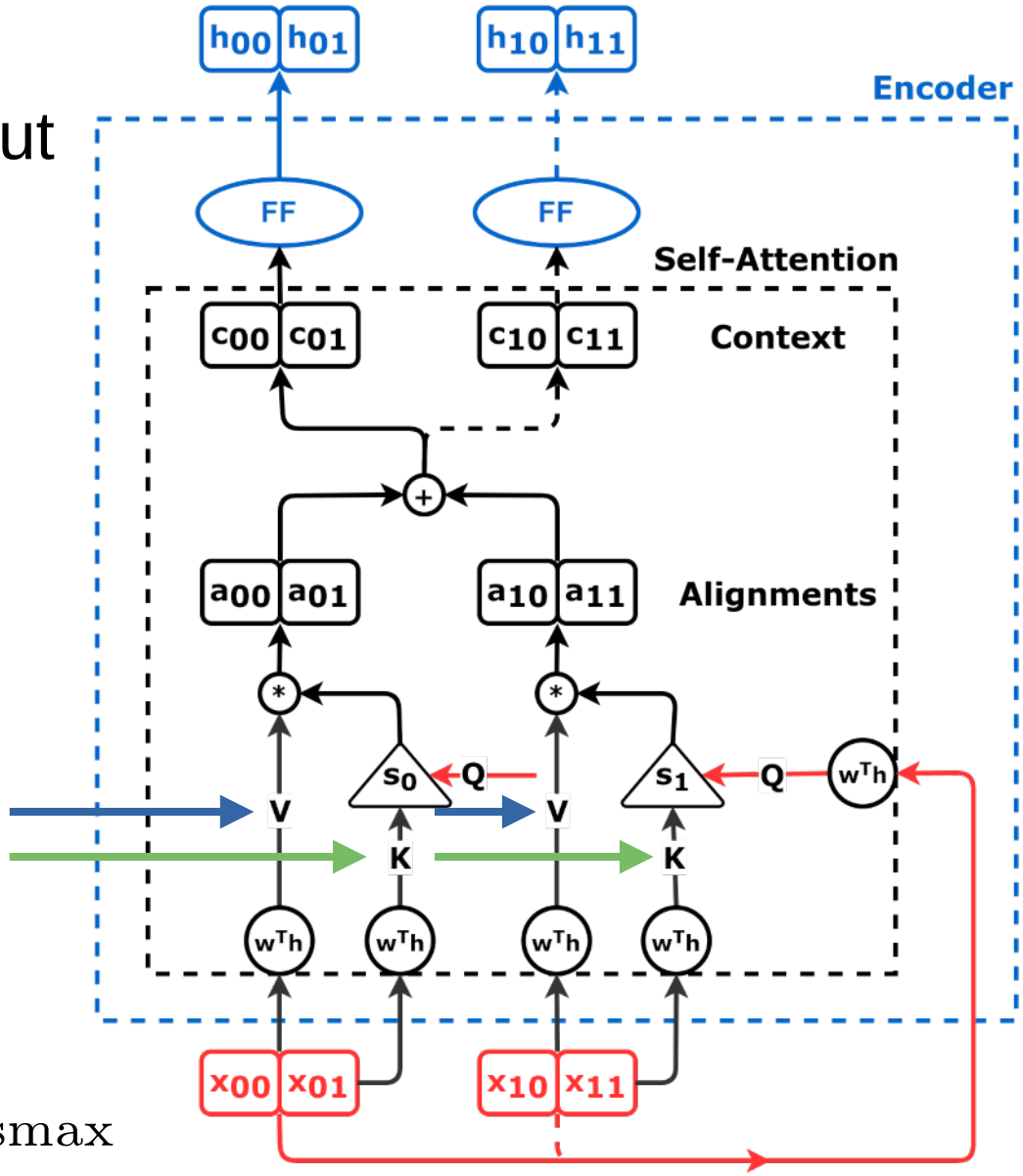
  Compute V, K, Q from the input sequence $X = [x_1, \ldots, x_L]$

- Attention

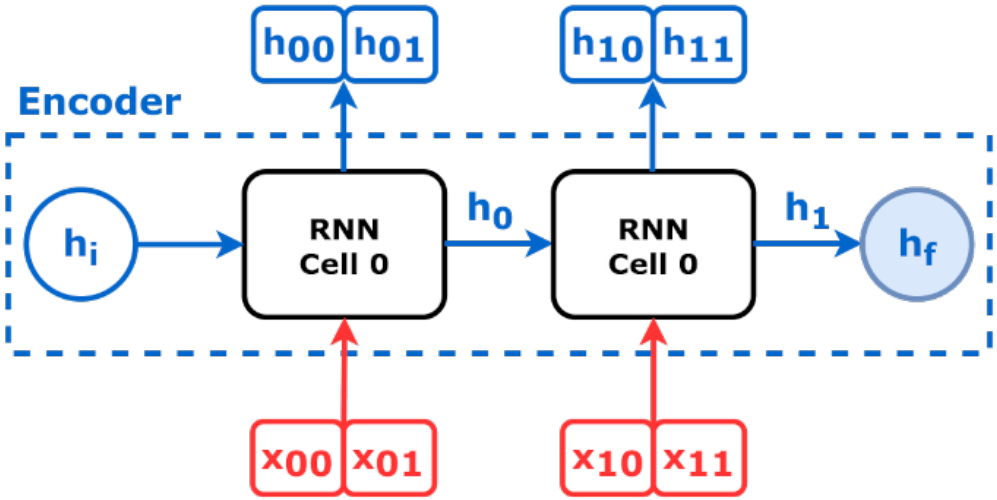
$$c = \underline{V}\left(\underline{K}^T \underline{q}\right)_{\text{ssmax}}$$

- Self-Attention

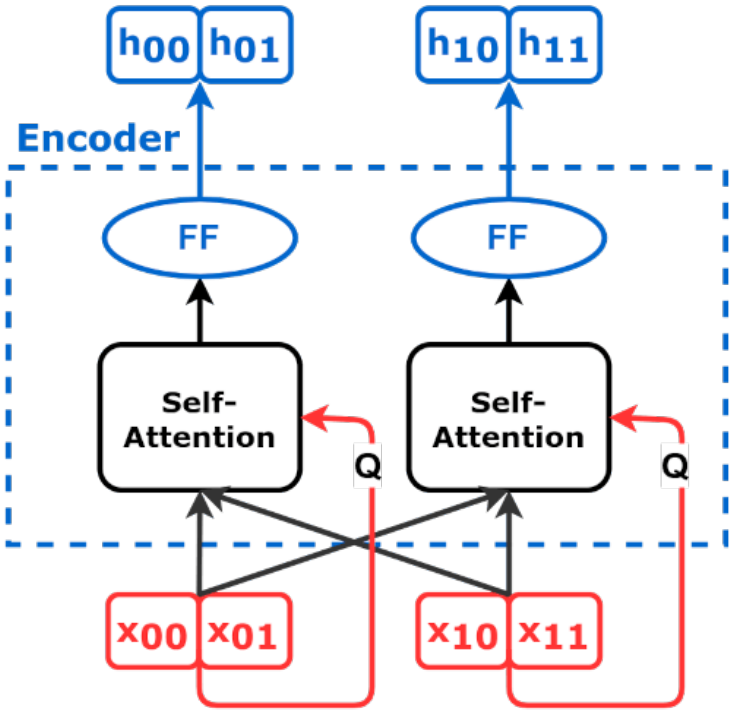
$$C = [\underline{W_v X}]\left([\underline{W_k X}]^T [\underline{W_q X}]\right)_{\text{ssmax}}$$



RNN vs Self-Attention



RNN



Self-Attention

Godoy, Deep Learning with Pytorch, 2019

Convolution vs Self-Attention

$$C = [W_v X] \left([W_k X]^T [W_q X] \right)_{\text{ssmax}}$$

$$c_i = \sum_j \langle q_i, k_j \rangle v_j$$

$$W_v = W_k = W_q = I$$

$$C = [X] \left([X]^T [X] \right)_{\text{ssmax}}$$

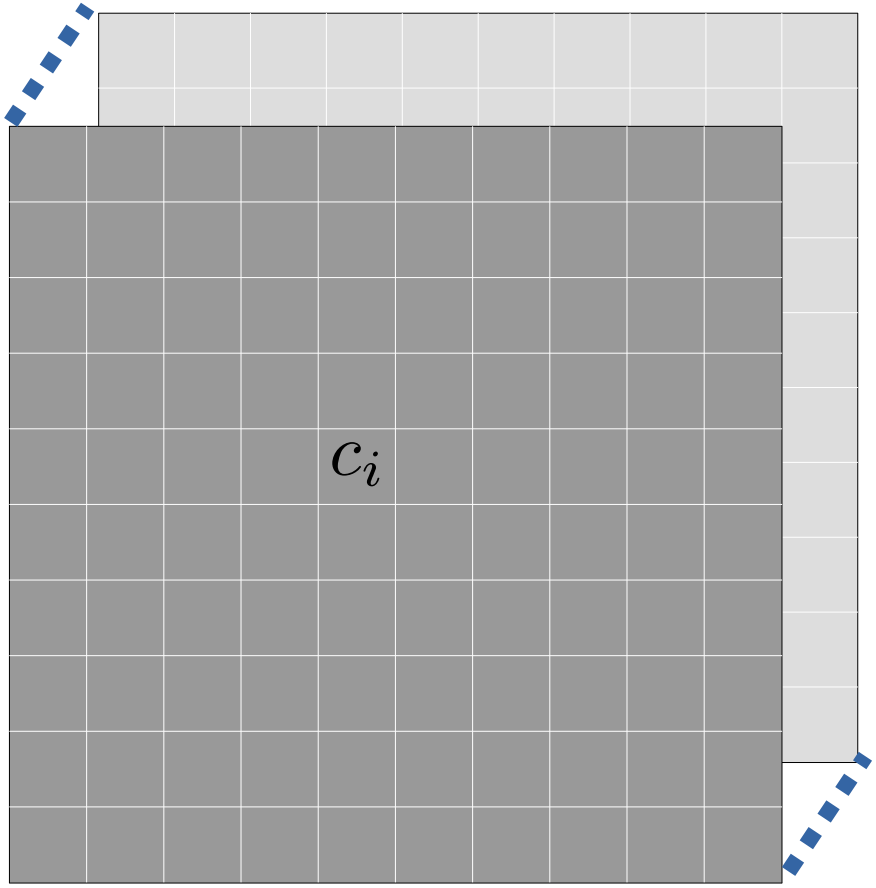
Convolution:

$$c_i = \sum_j h_{i-j} x_j$$

$$c_i = \sum_j \langle x_i, x_j \rangle x_j$$

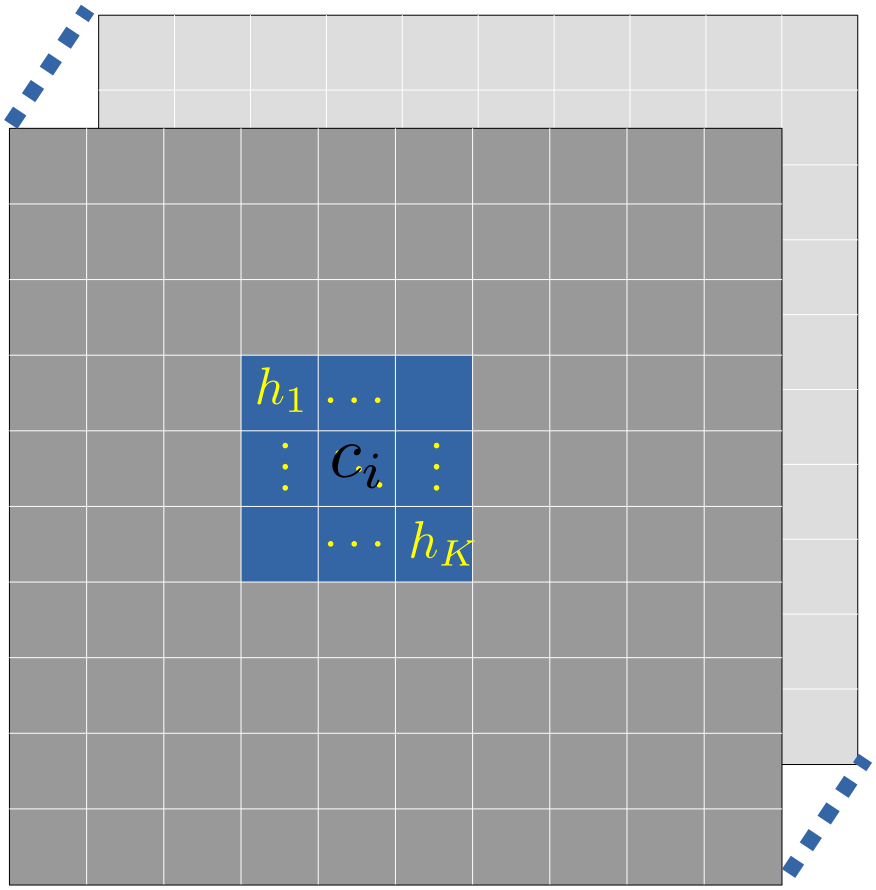
Convolution vs Self-Attention

$$c_i = \sum_j h_{i-j} x_j$$



Convolution vs Self-Attention

$$c_i = \sum_j h_{i-j} x_j$$

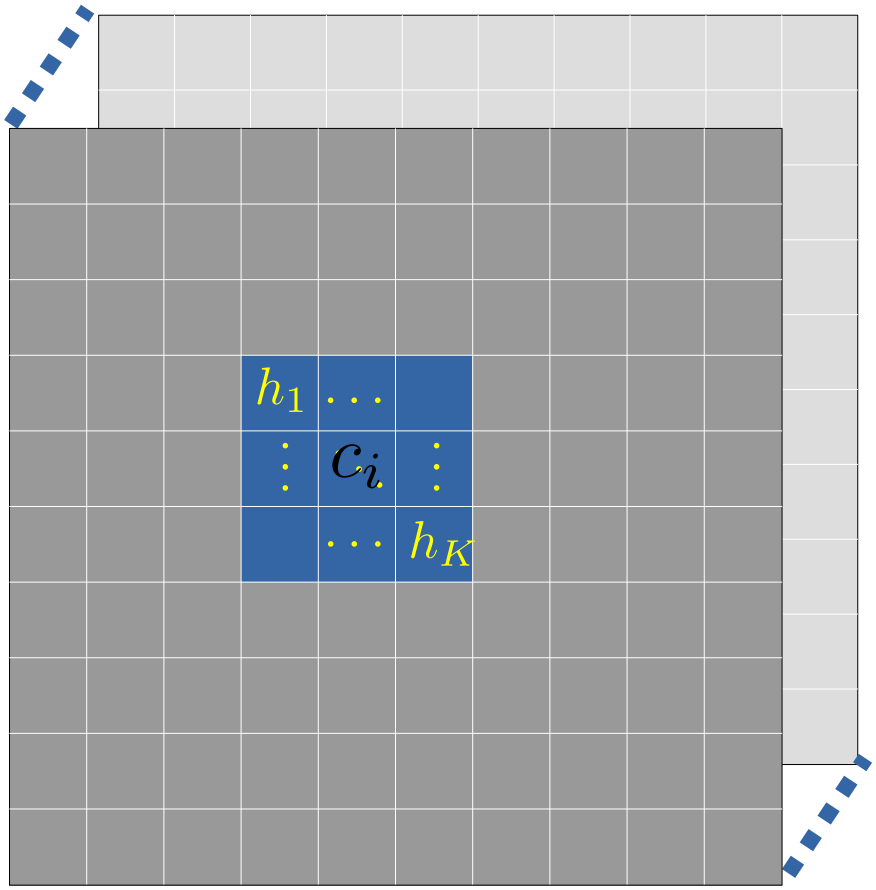


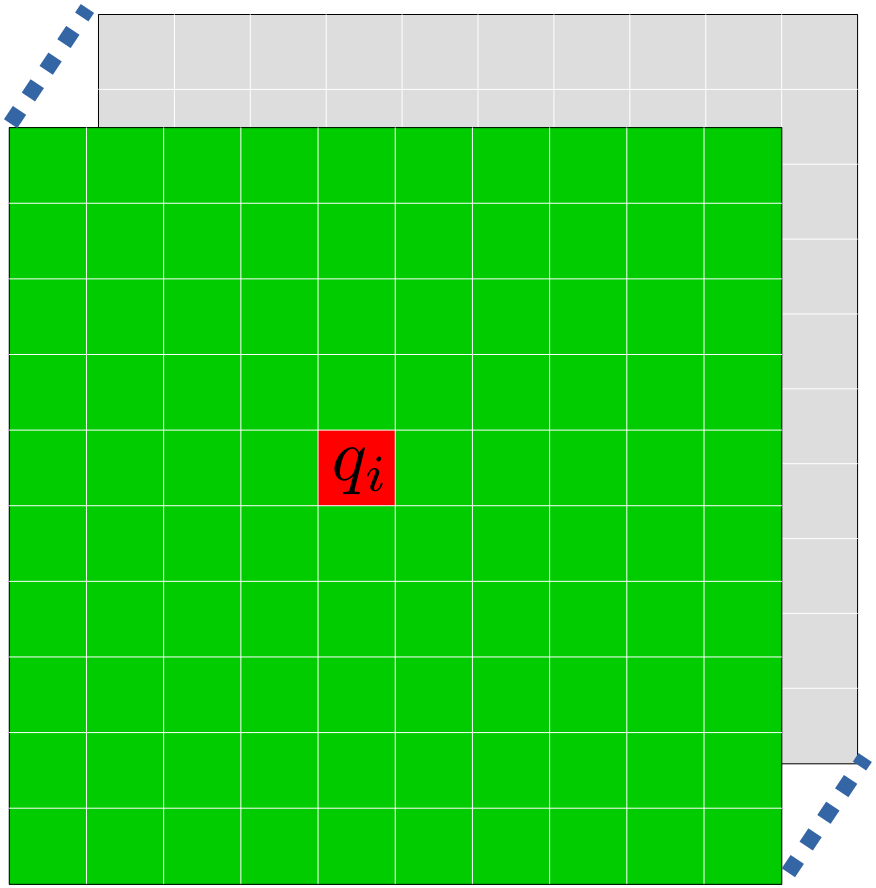
Local and fixed
during inference

Convolution vs Self-Attention

$$c_i = \sum_j h_{i-j} x_j$$

$$c_i = \sum_j \overbrace{\langle q_i, k_j \rangle}^{\alpha_{ij}} v_j$$



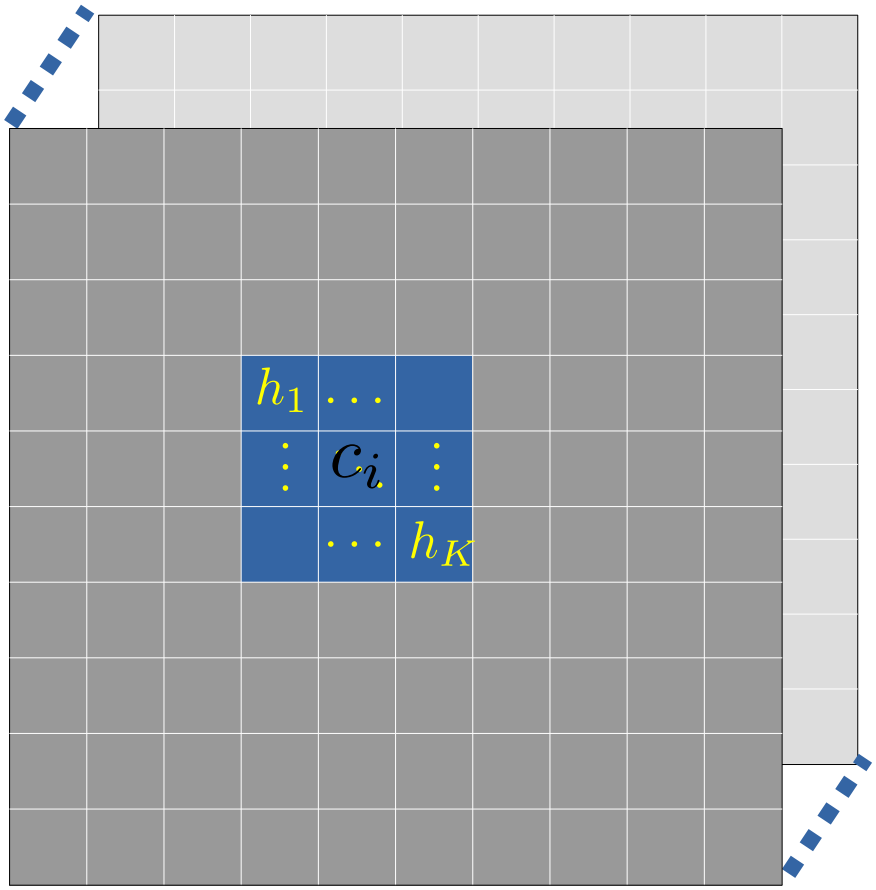


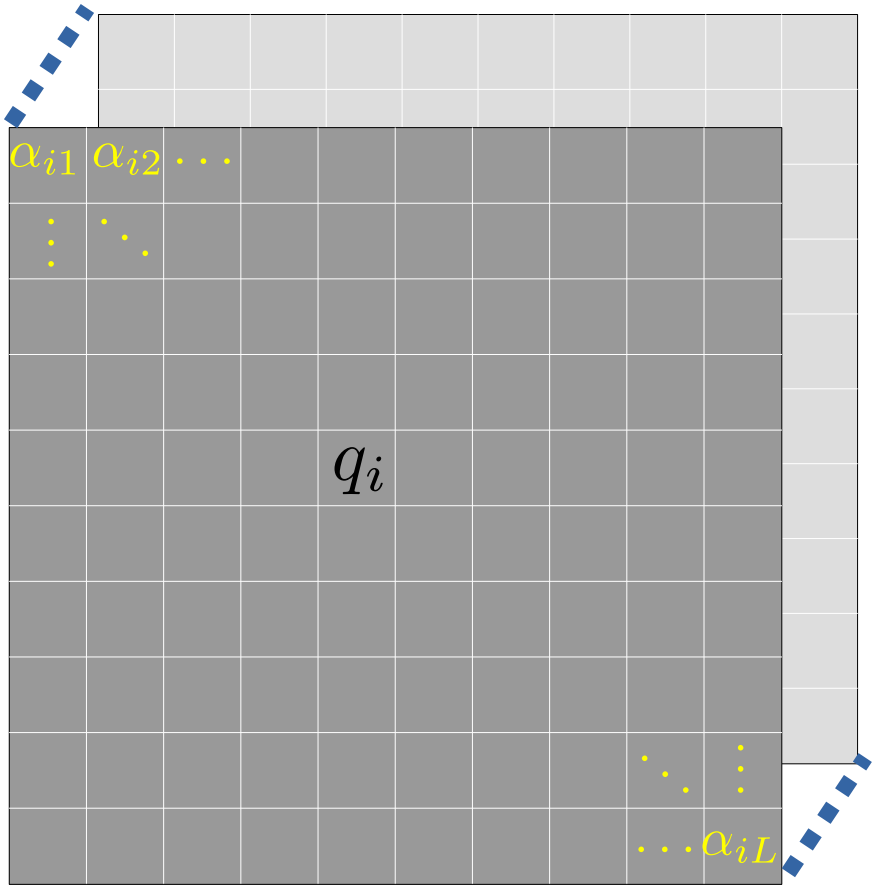
Local and fixed during inference

Convolution vs Self-Attention

$$c_i = \sum_j h_{i-j} x_j$$

$$c_i = \sum_j \overbrace{\langle q_i, k_j \rangle}^{\alpha_{ij}} v_j$$



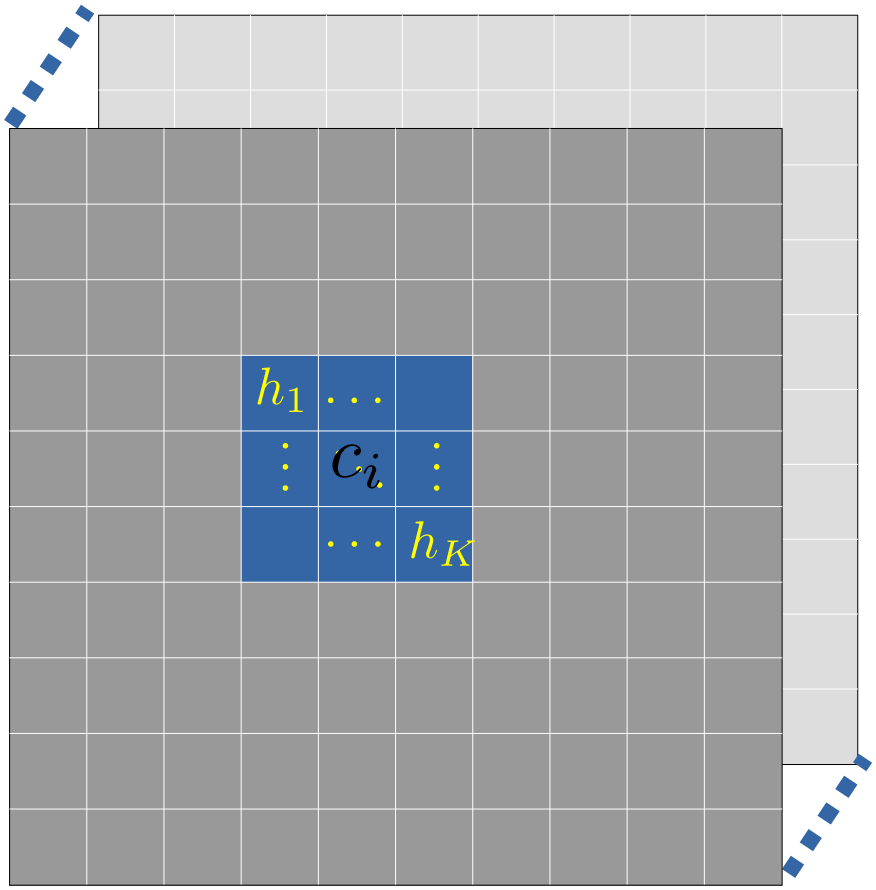


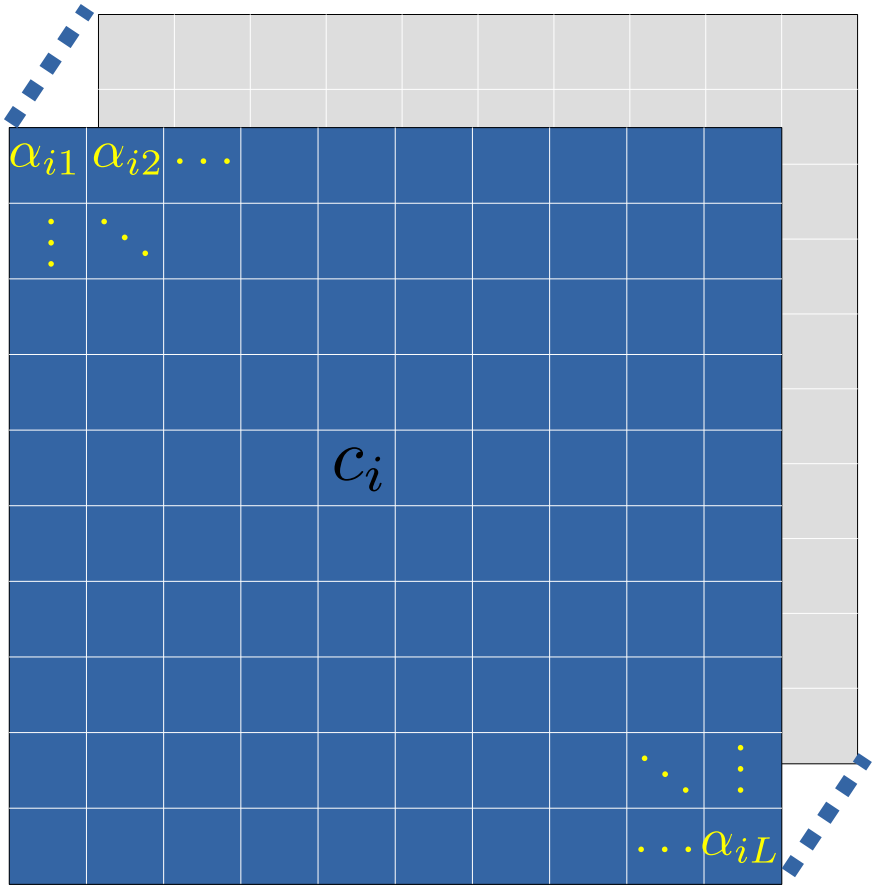
Local and fixed during inference

Convolution vs Self-Attention

$$c_i = \sum_j h_{i-j} x_j$$

$$c_i = \sum_j \overbrace{\langle q_i, k_j \rangle}^{\alpha_{ij}} v_j$$



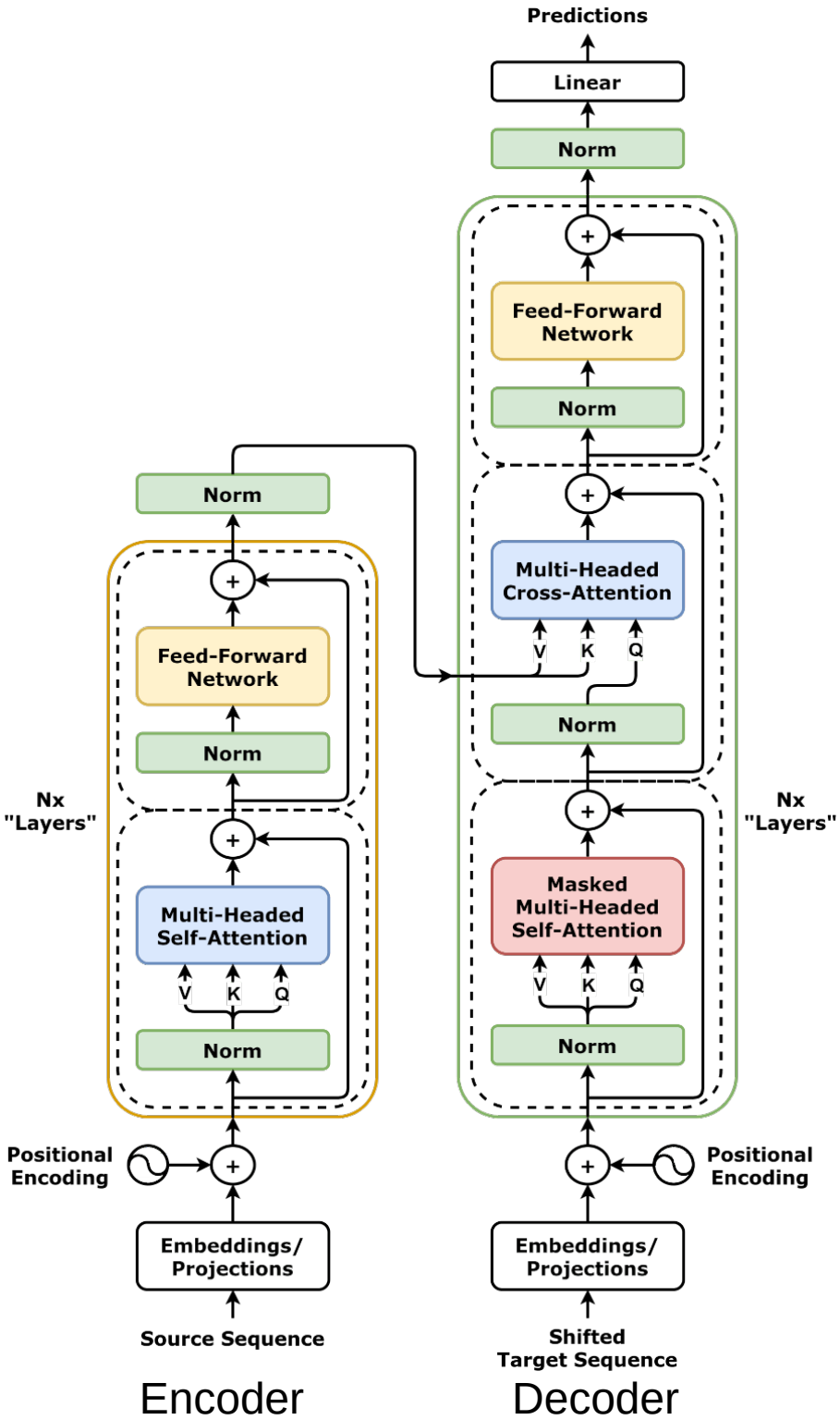


Local and fixed during inference

Global and variable during inference

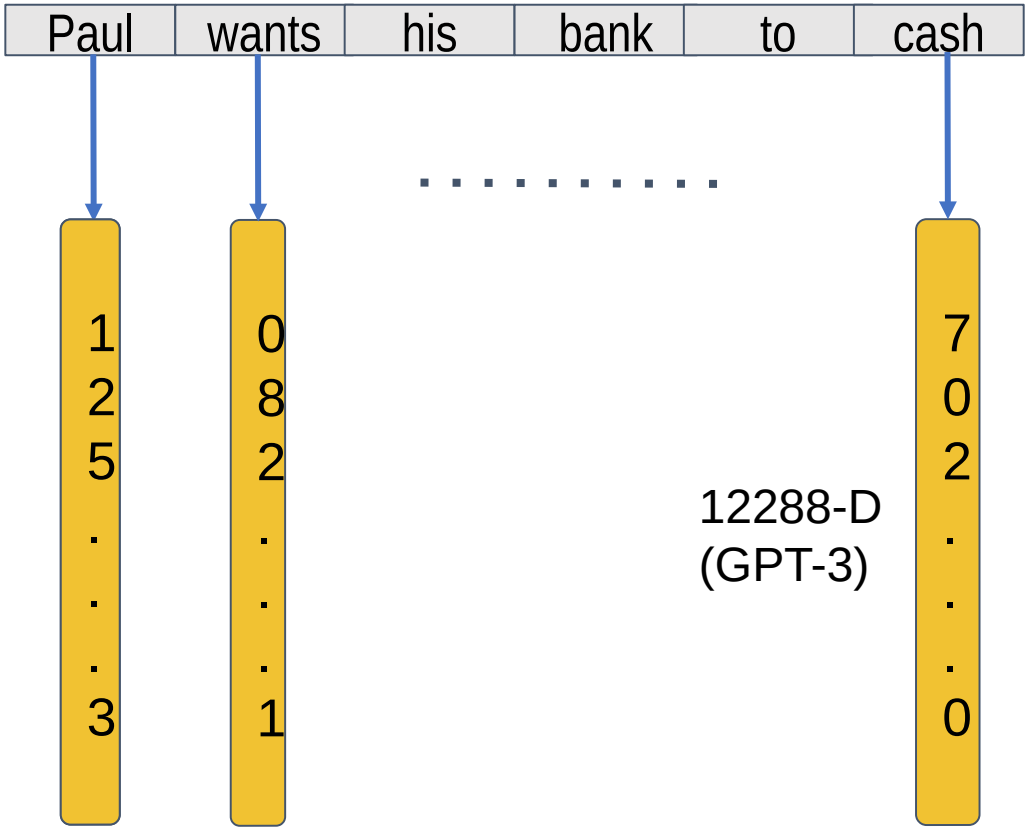
The Transformer

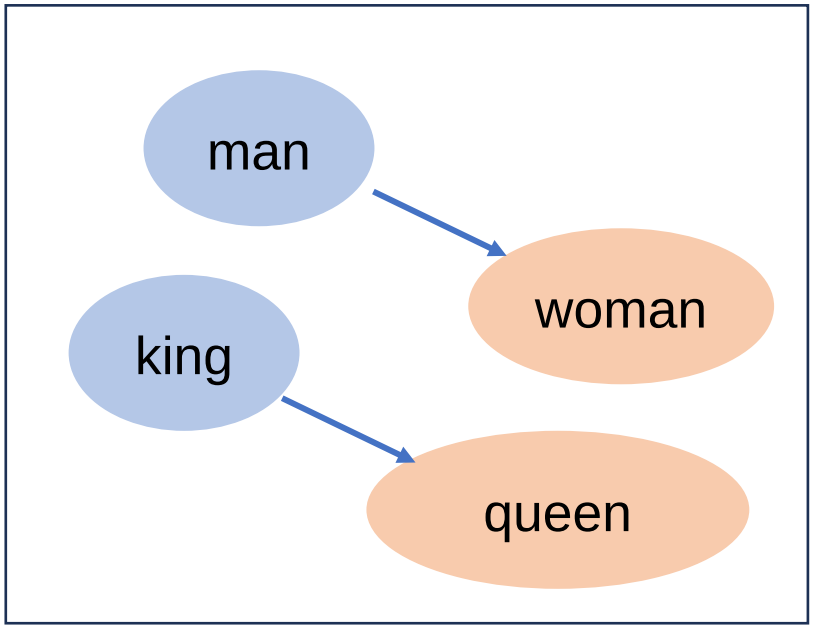
- Multi-Headed Self- and Cross-Attention
- Masked Multi-Headed Self-Attention
- Layer Normalization
- Linear + ReLU
- Positional Encoding
- Residual Connection
- Dropouts



Vaswani et al., Attention is all you need, 2017

Embedding





king – man + woman = queen

Byte pair encoding → tokens

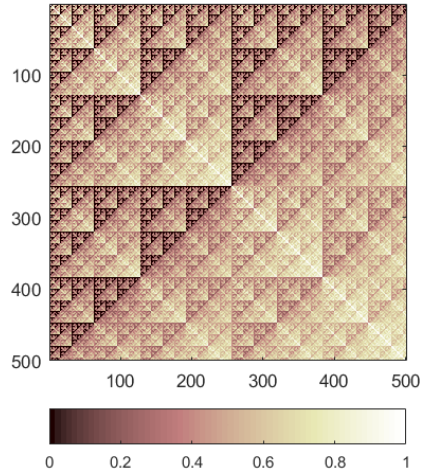
50257 tokens in GPT-3

Mikolov et al., Word2Vec, 2013

Positional Encoding

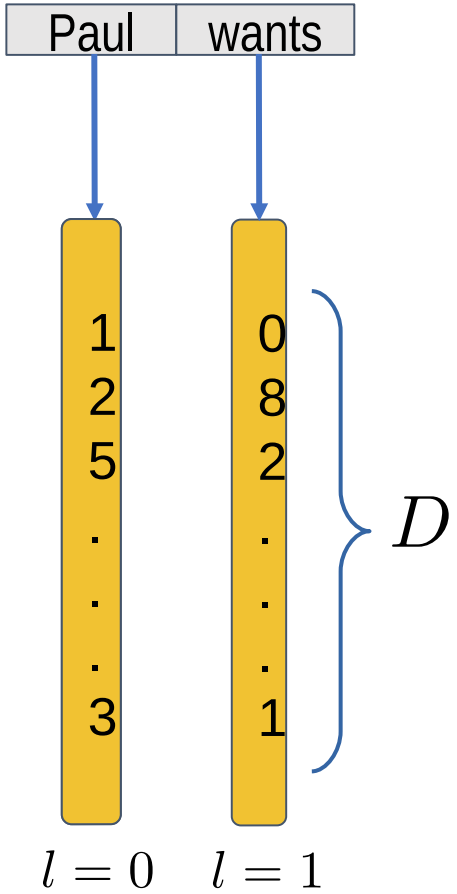
- **Unique** encoding for each position
- Encoding distance between two positions **consistent**
- **Generalize** to any sequence length

- Binary code?



$$\omega_k = \frac{1}{10000^{2k/D}}$$

$$\mathbf{p}_l = \begin{bmatrix} \sin(\omega_1 l) \\ \cos(\omega_1 l) \\ \sin(\omega_2 l) \\ \cos(\omega_2 l) \\ \vdots \\ \sin(\omega_{D/2} l) \\ \cos(\omega_{D/2} l) \end{bmatrix}$$

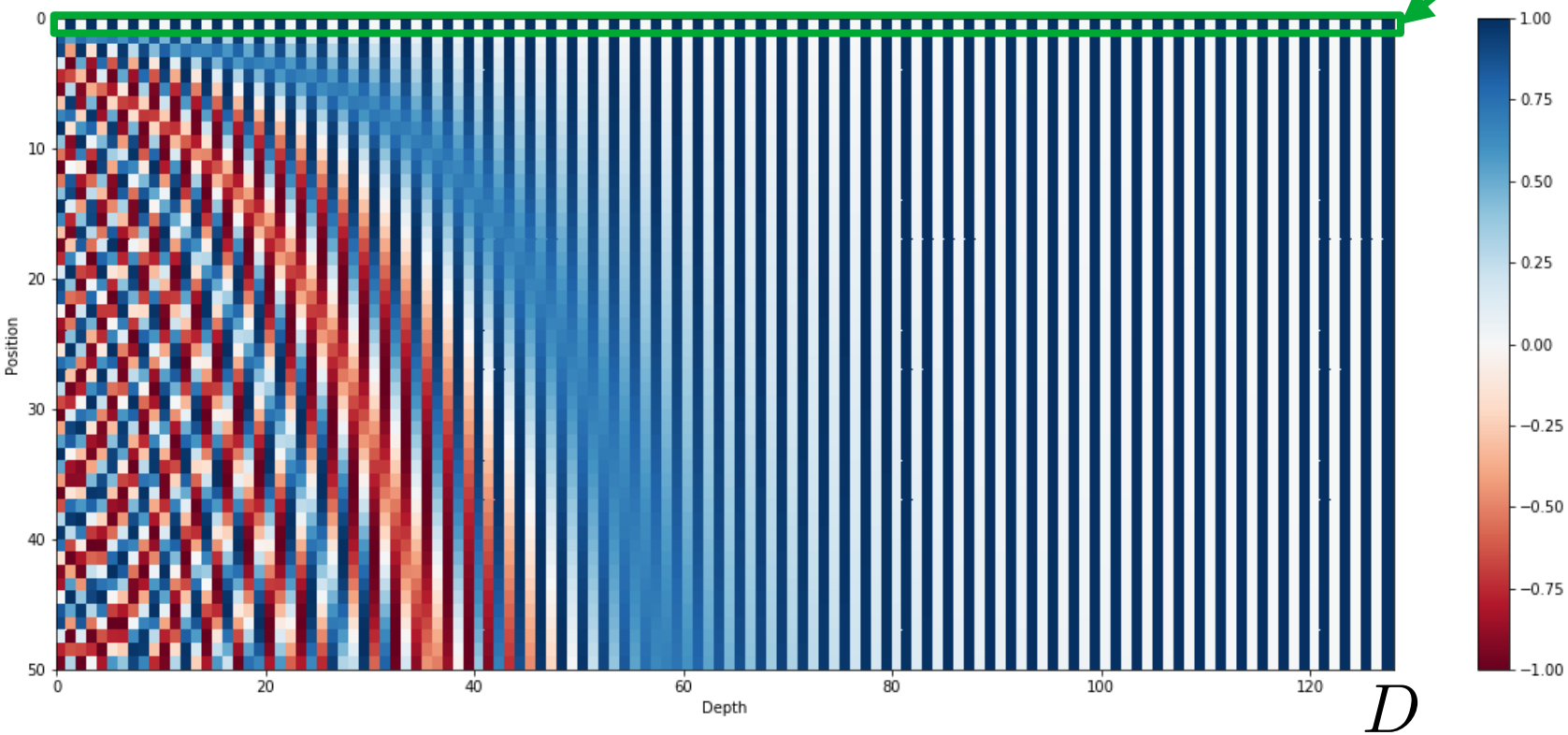


Kazemnejad: Transformer Architecture: The Positional Encoding

Handcrafted Positional Encoding

$$\mathbf{p}_l = \begin{bmatrix} \sin(\omega_1 l) \\ \cos(\omega_1 l) \\ \sin(\omega_2 l) \\ \cos(\omega_2 l) \\ \vdots \\ \sin(\omega_{D/2} l) \\ \cos(\omega_{D/2} l) \end{bmatrix}$$

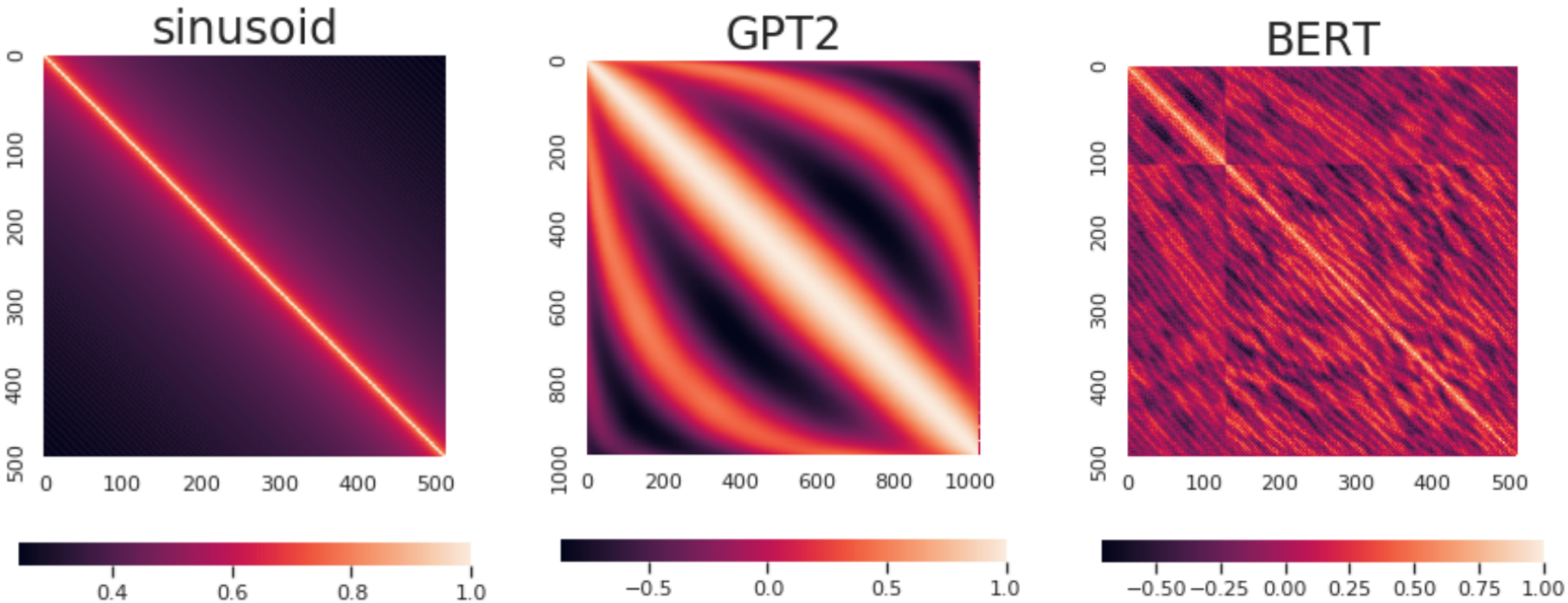
l



D

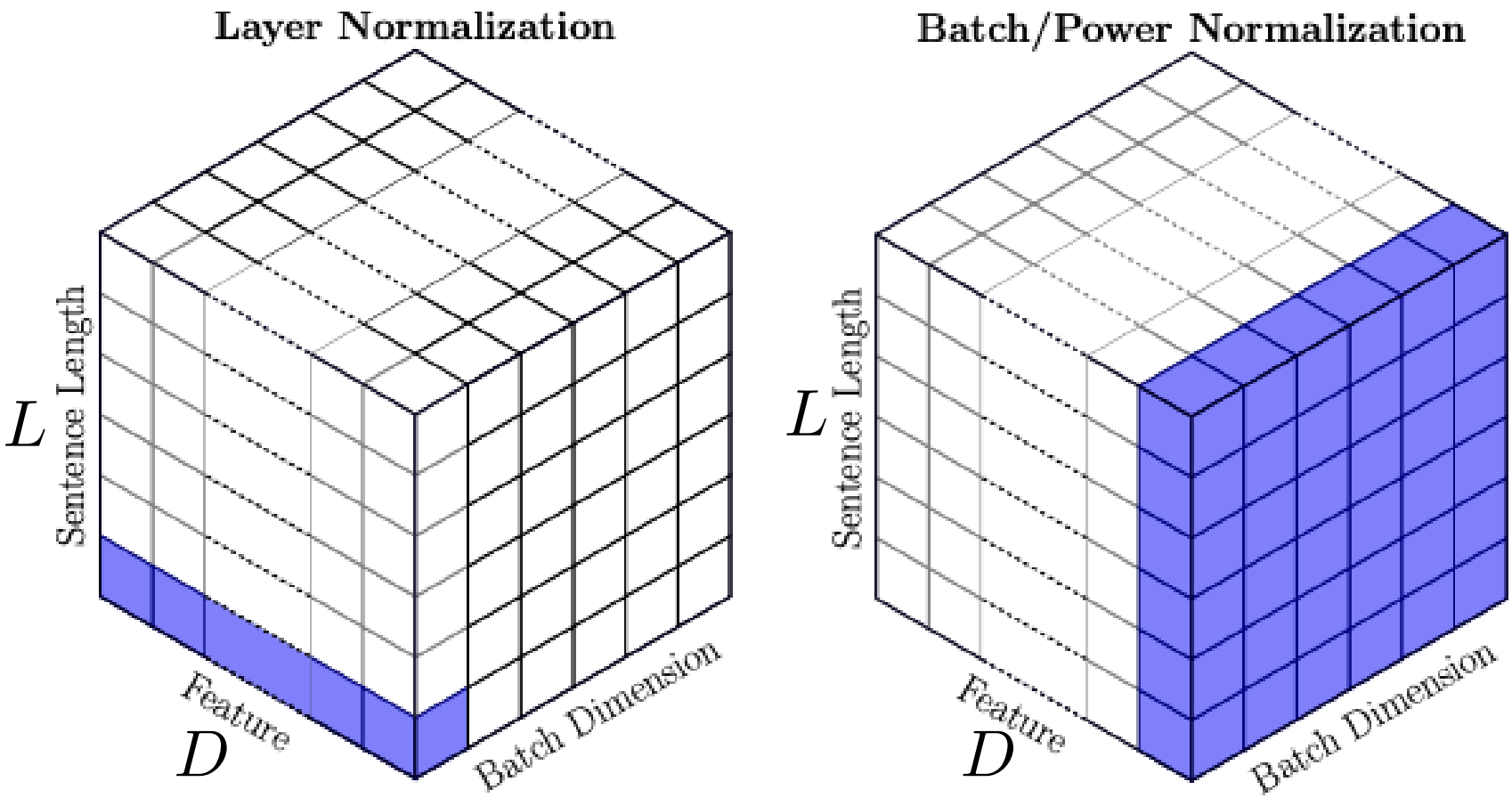
Learned Positional Encoding

- PE Correlation between different positions

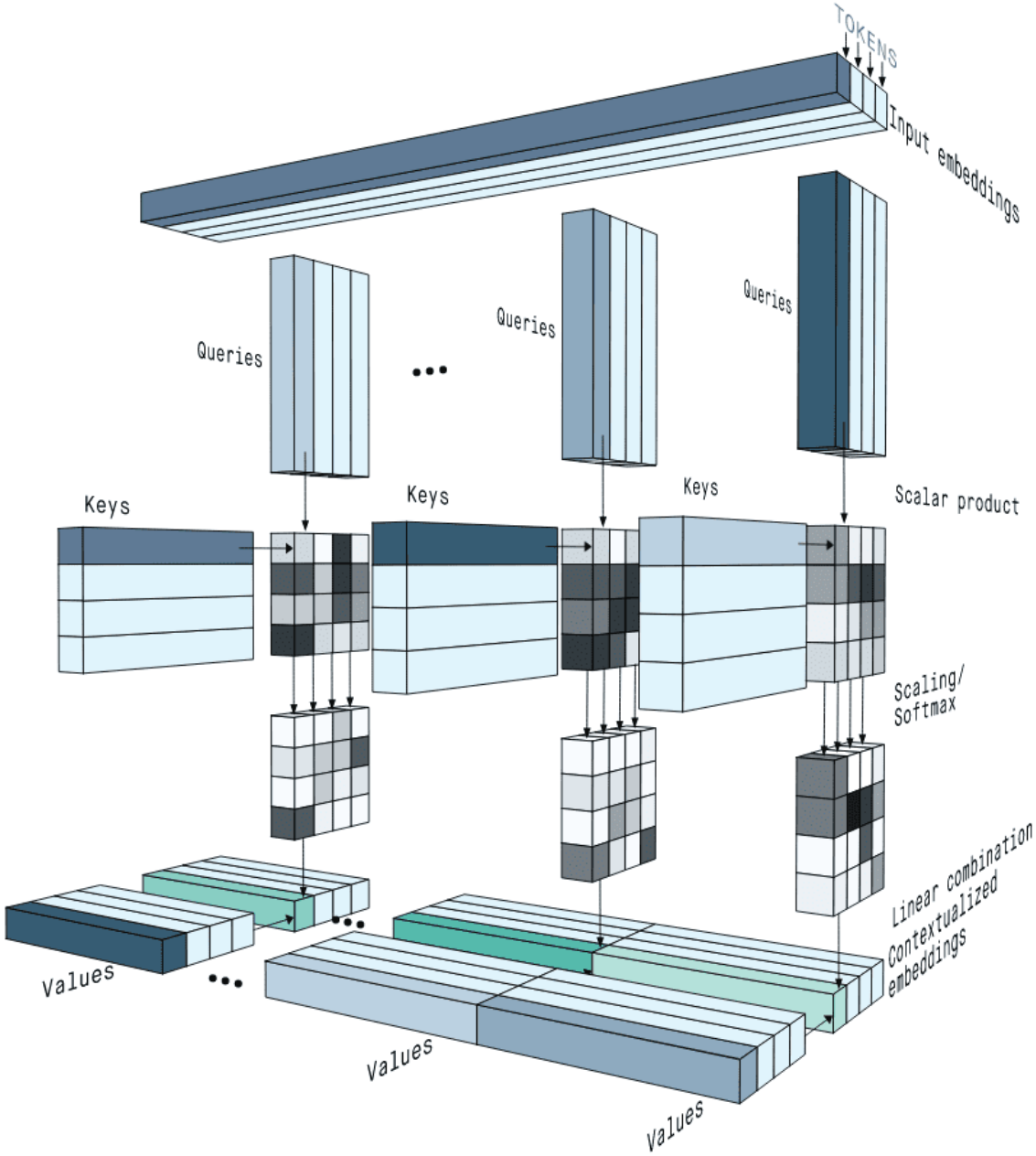


Wang, Chen: What do position embeddings learn?, EMNLP 2020

Layer Normalization



Multi-headed Self-Attention



Masked Attention

$$i = 1 \qquad 2 \qquad\qquad L$$

$$c_i = \sum_j \overbrace{\langle q_i, k_j \rangle}^{\alpha_{ij}} v_j$$

$$i, j = 1, \ldots, L$$

α_{11}	x	x	x	x	x	x	x	x	x
α_{21}	α_{22}	x	x	x	x	x	x	x	x
$\vdots$	$\vdots$	$\ddots$	x	x	x	x	x	x	x
				x	x	x	x	x	x
					x	x	x	x	x
						x	x	x	x
							x	x	x
								x	x
									x
α_{L1}	α_{L2}	$\ldots$							α_{LL}

Large Language Models

1) **Pretraining**

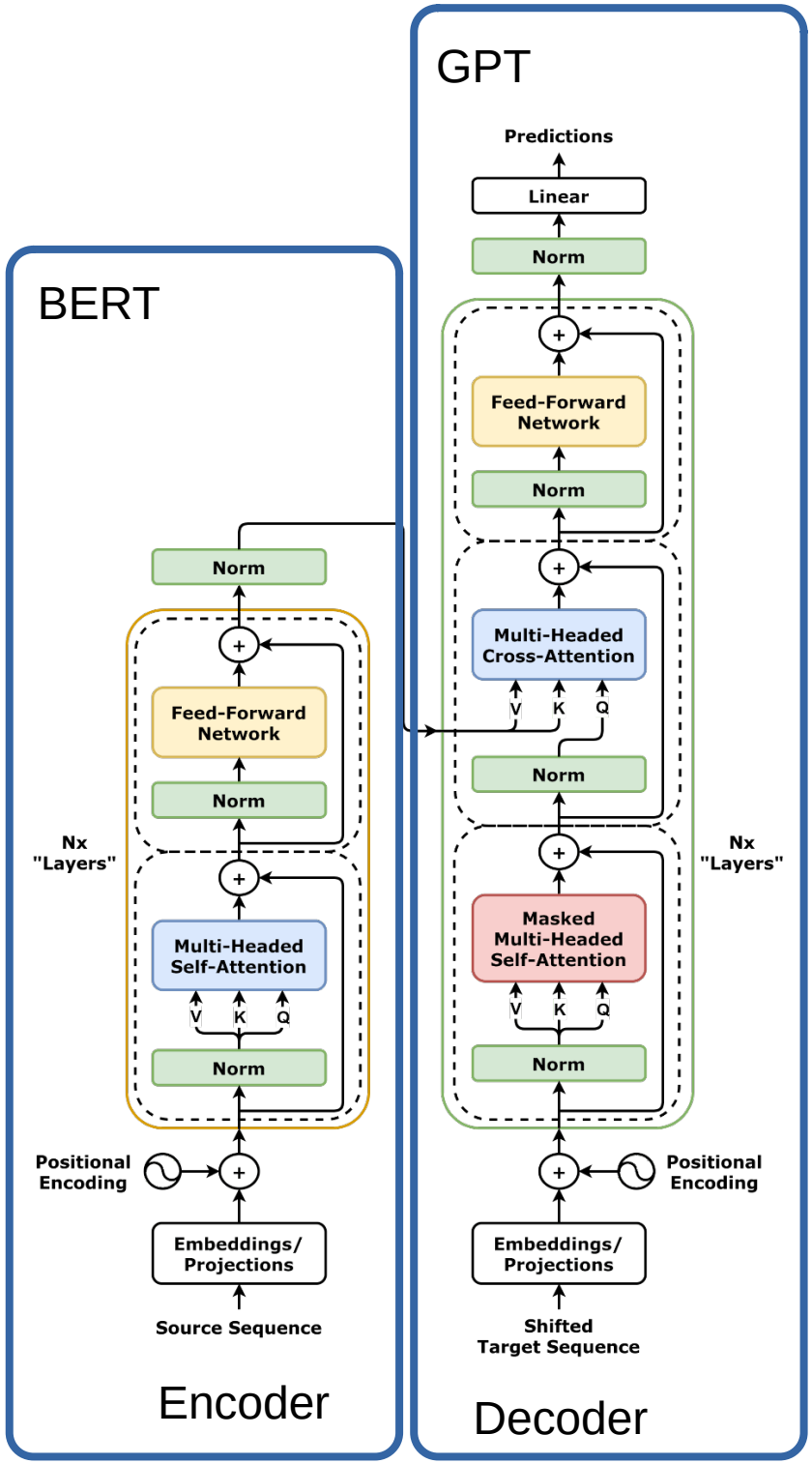
- Annotation is a curse
- Task - “Predict a next (masked) word in an incomplete sentence.“

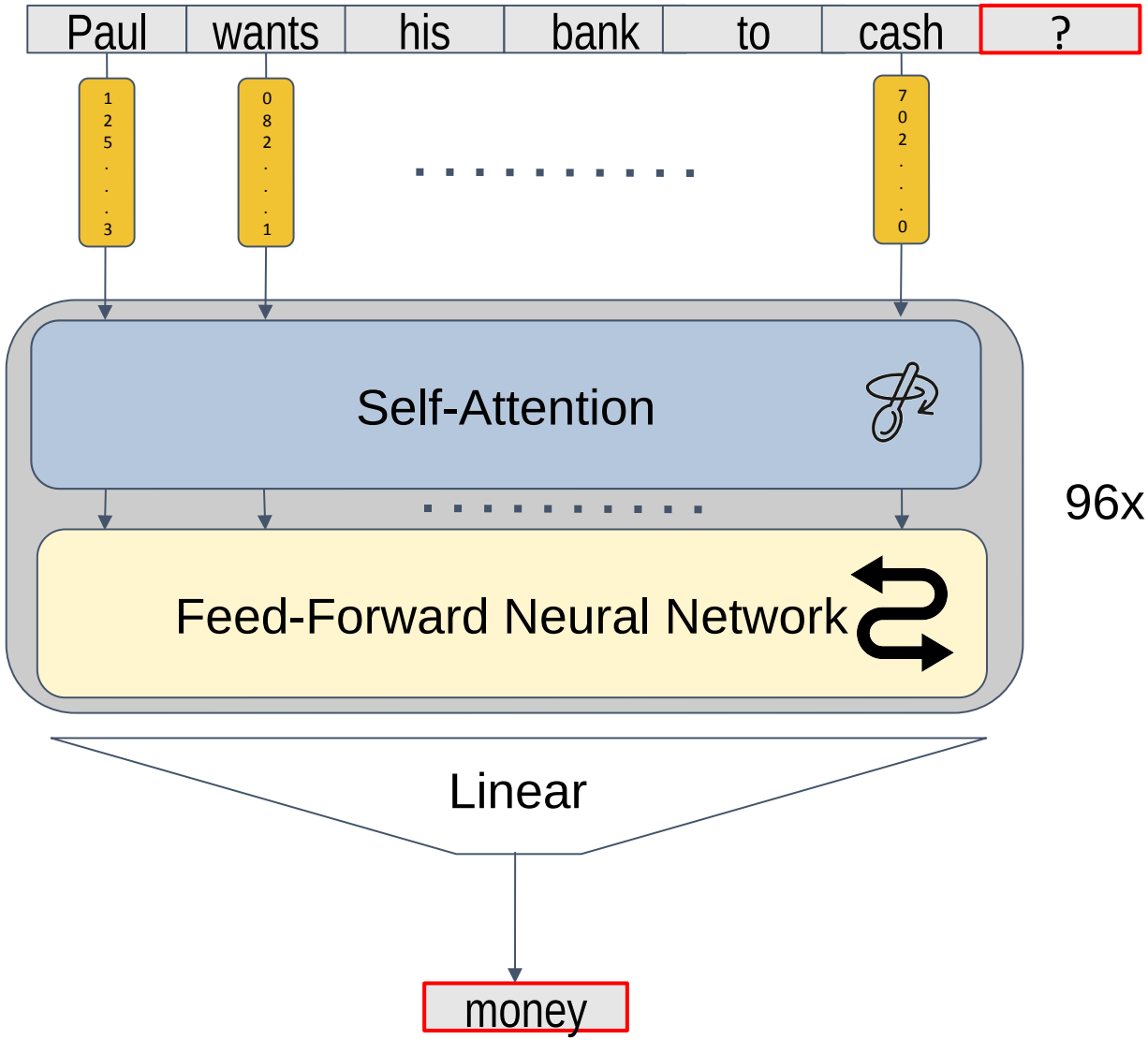
2) **Fine-tuning on a downstream task**

- Small annotation dataset
- Reinforcement learning from Human Feedback
- Immitate human preferences - Reward Model

GPT (OpenAI) , BERT (Google),

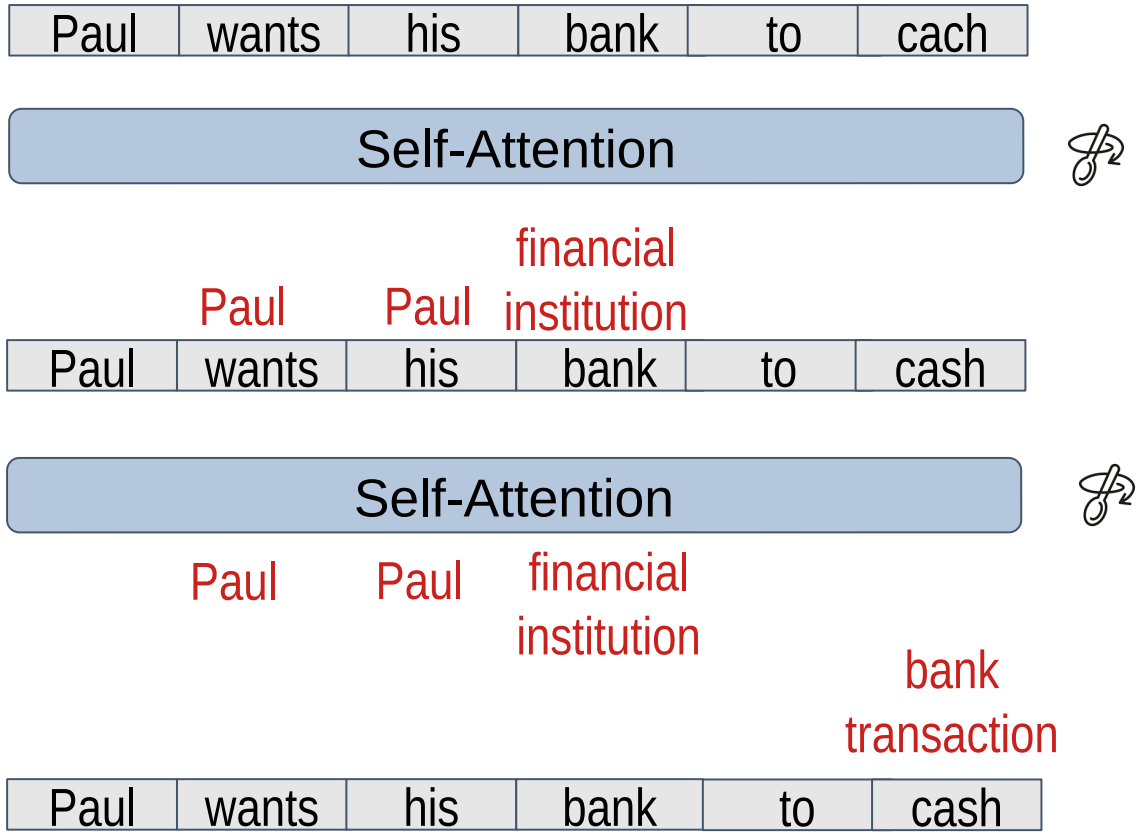
LLaMA (Meta AI), Titan (Amazon)


GPT
Predictions
Linear
Norm
Feed-Forward Network
Norm
Multi-Headed Cross-Attention
V
K
Q
Norm
Masked Multi-Headed Self-Attention
V
K
Q
Norm
Nx "Layers"
Positional Encoding
Embeddings/ Projections
Shifted Target Sequence
Decoder
BERT
Norm
Feed-Forward Network
Norm
Multi-Headed Self-Attention
V
K
Q
Norm
Nx "Layers"
Positional Encoding
Embeddings/ Projections
Source Sequence
Encoder


Paul
wants
his
bank
to
cash
?
125...3
082...1
702...0
Self-Attention
96x
Feed-Forward Neural Network
Linear
money

Self-Attention

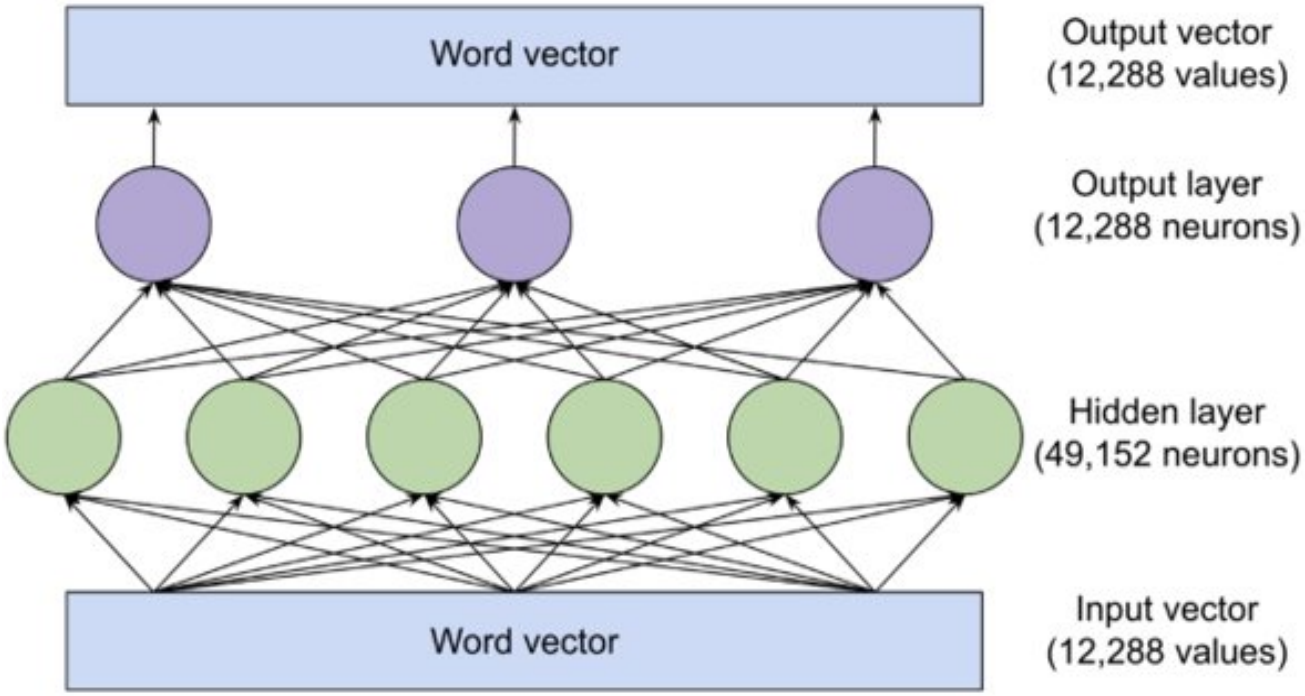
- Understanding, relations, reasoning

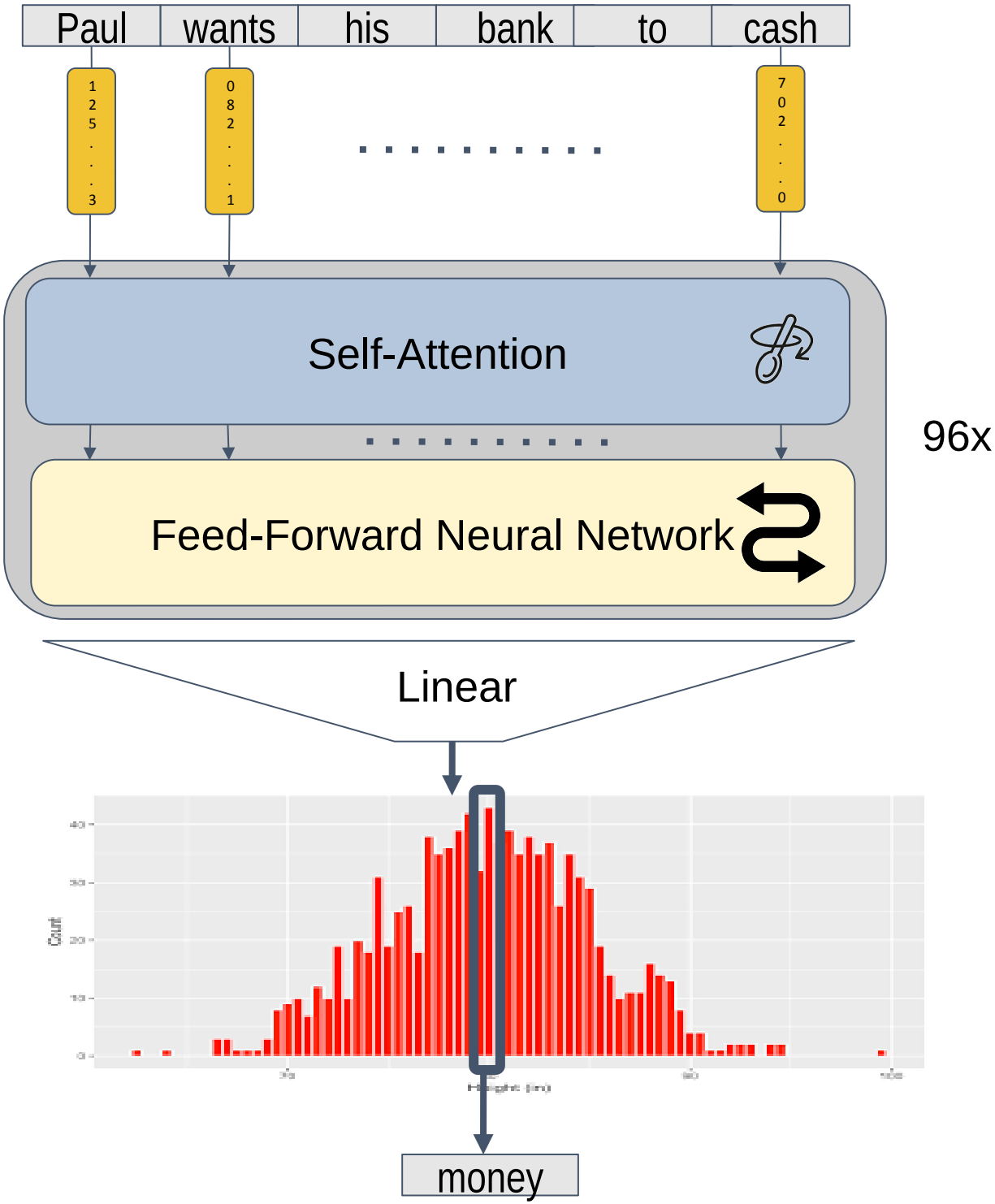


Feed-Forward Neural Network

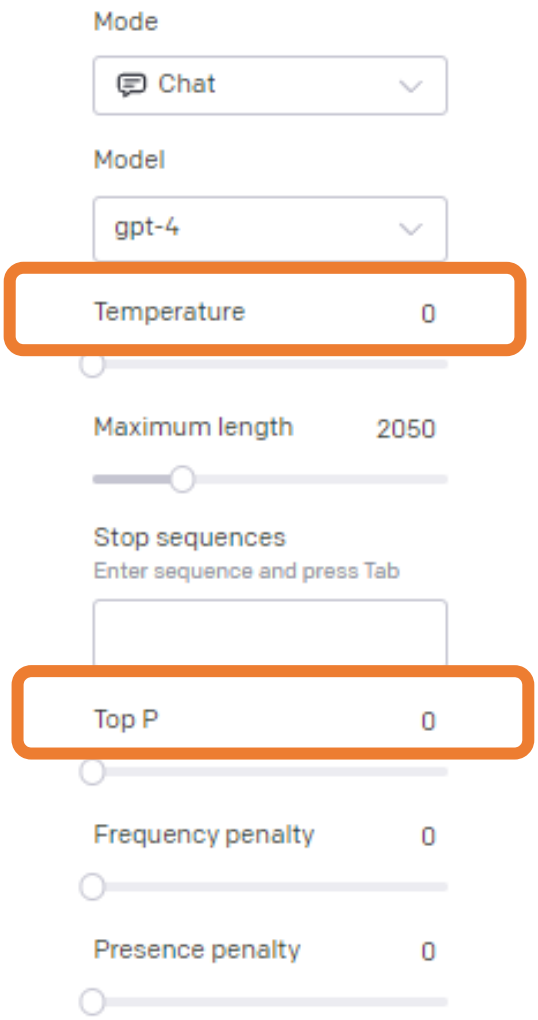
- Long-term Memory – facts from the training data

- GPT-3:
  - 175 ($x10^9$) billion of parameters
- GPT-4:
  - 1 700 billion
- Human brain:
  - 700 000 billion

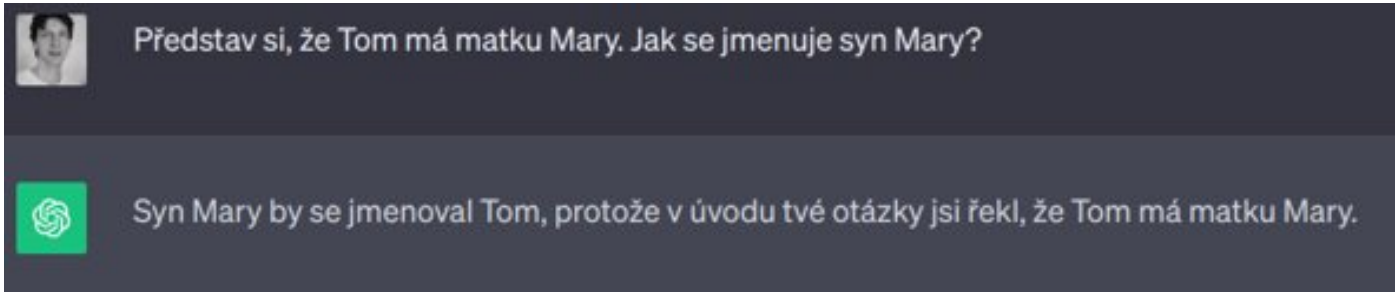




Non-deterministic!

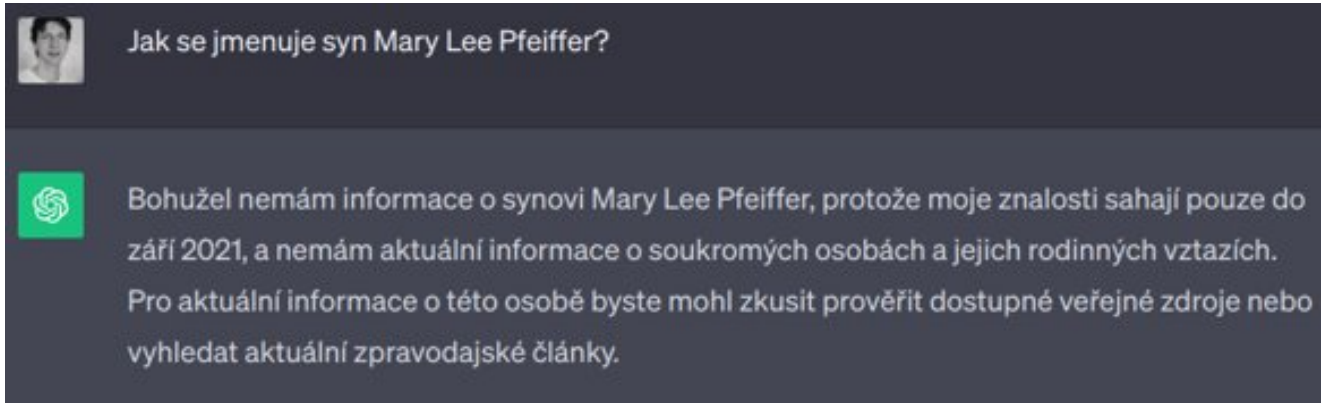


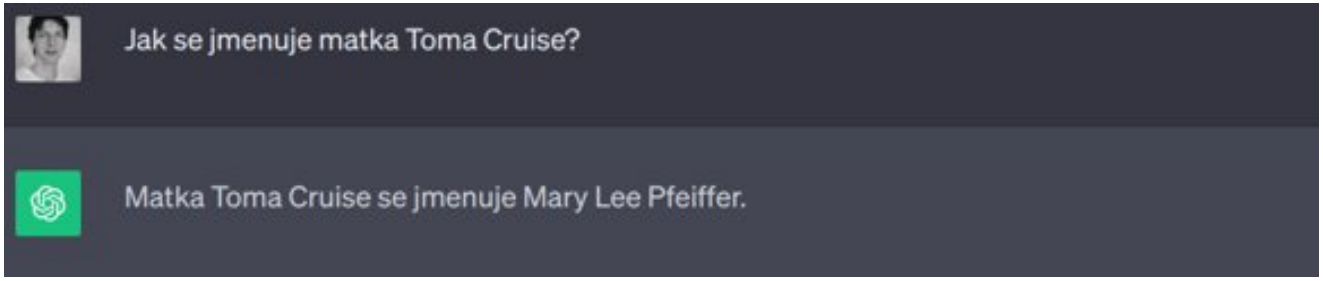
Context in Prompting



Self-Attention can infer opposite implication using the context.

Long-term memory in FF does not have this ability.





Training with Human Feedback

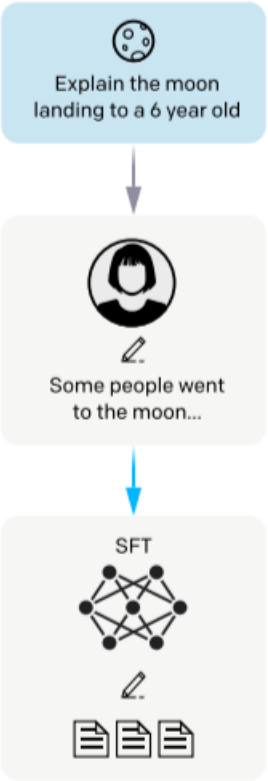
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



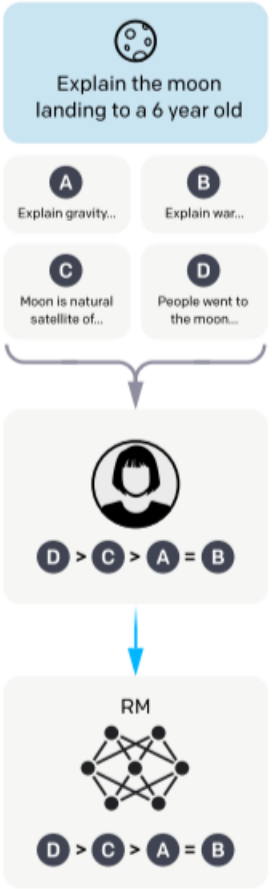
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

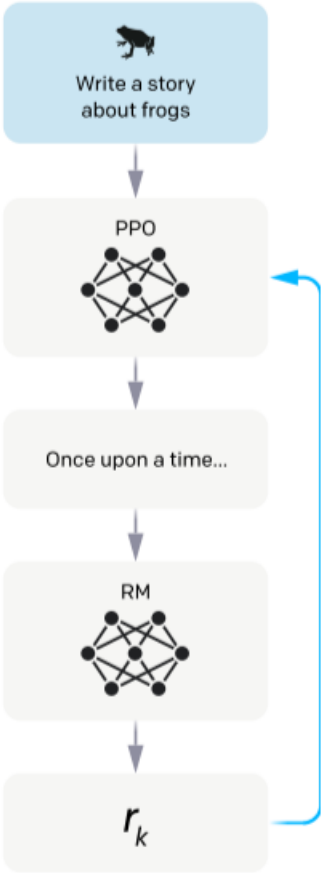
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

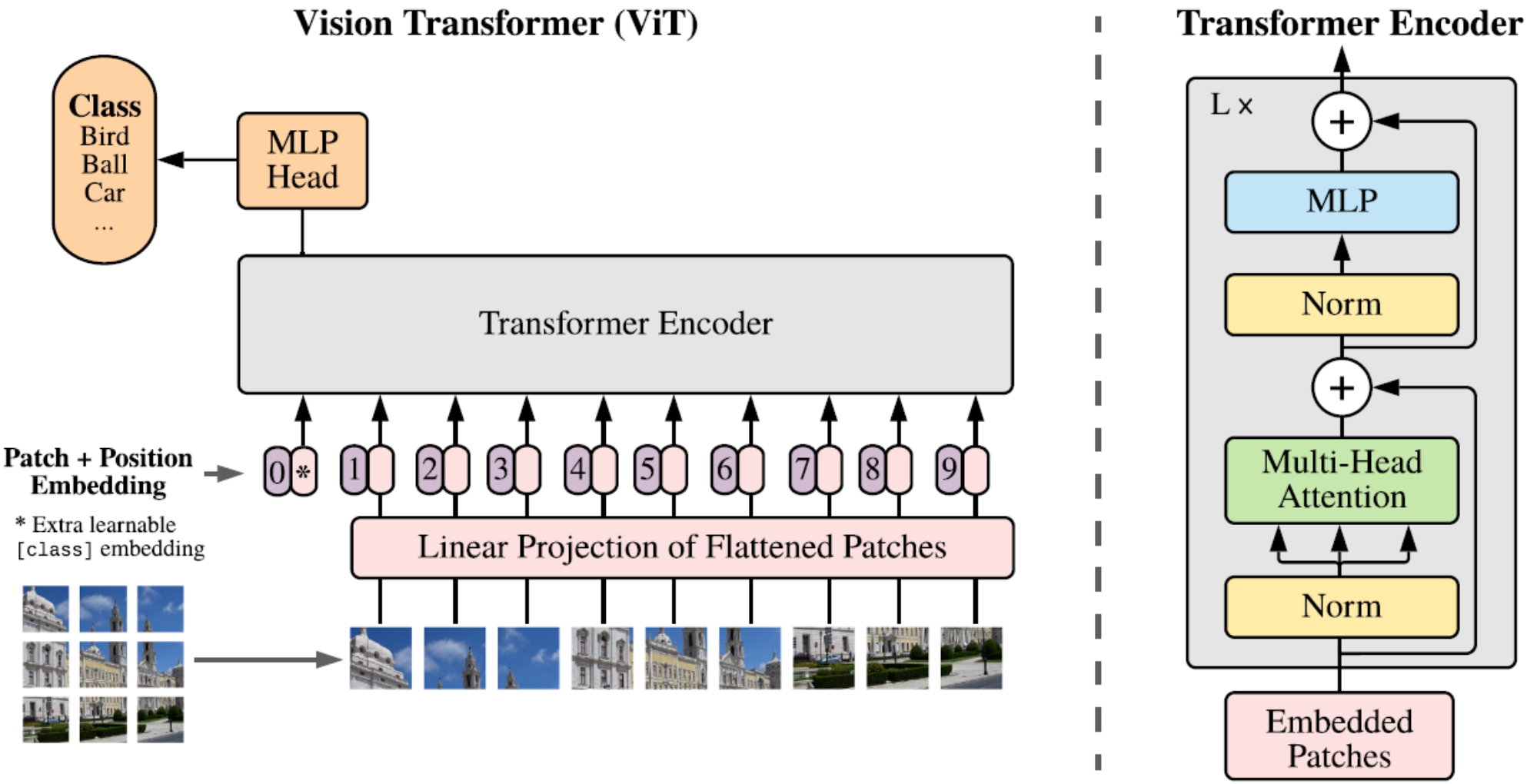


Ouyang et al., Training with Human Feedback, 2022

Attention in Image Processing

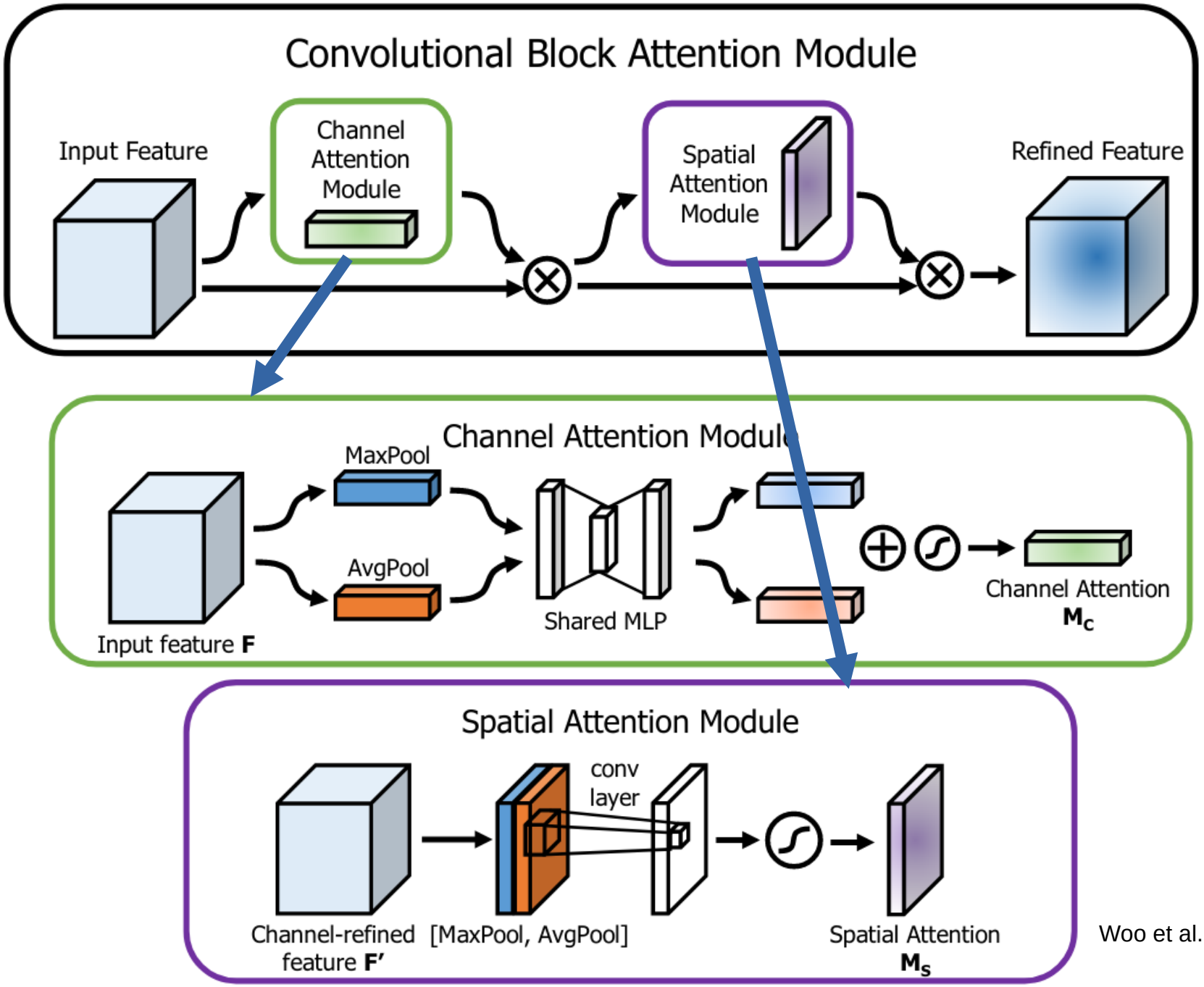
- Image Classification & Detection:
  - CBAM (Convolutional Block Attention Module)
  - Dual Attention (Spatial and Channel)
  - ViT (Vision Transformer)
  - CoAtNet (Convolution with Self-Attention)

Vision Transformer (ViT)



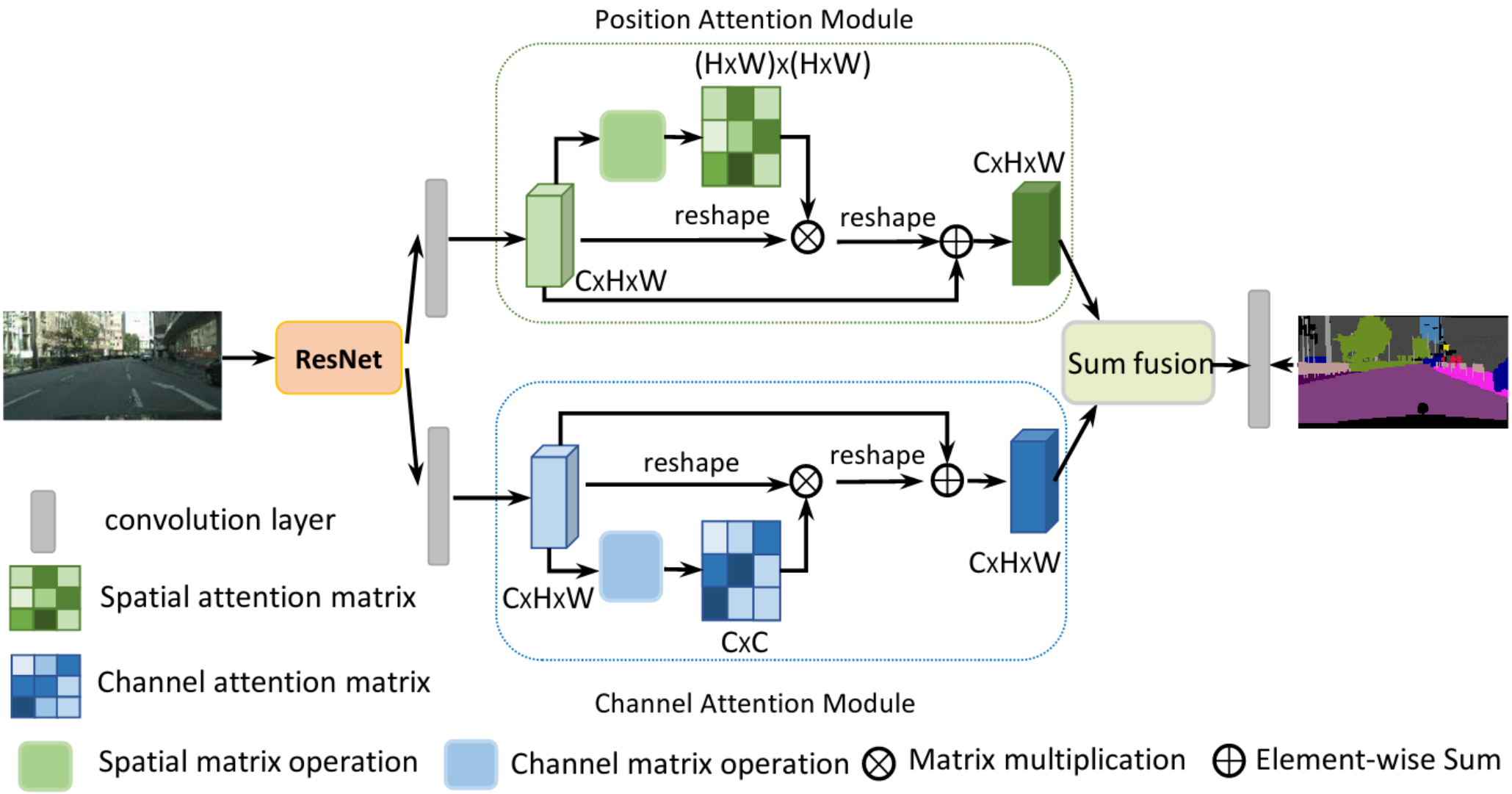
Dosovitskiy et al., An Image is Worth 16x16 Words, ICLR 2021

Convolutional Block Attention Module



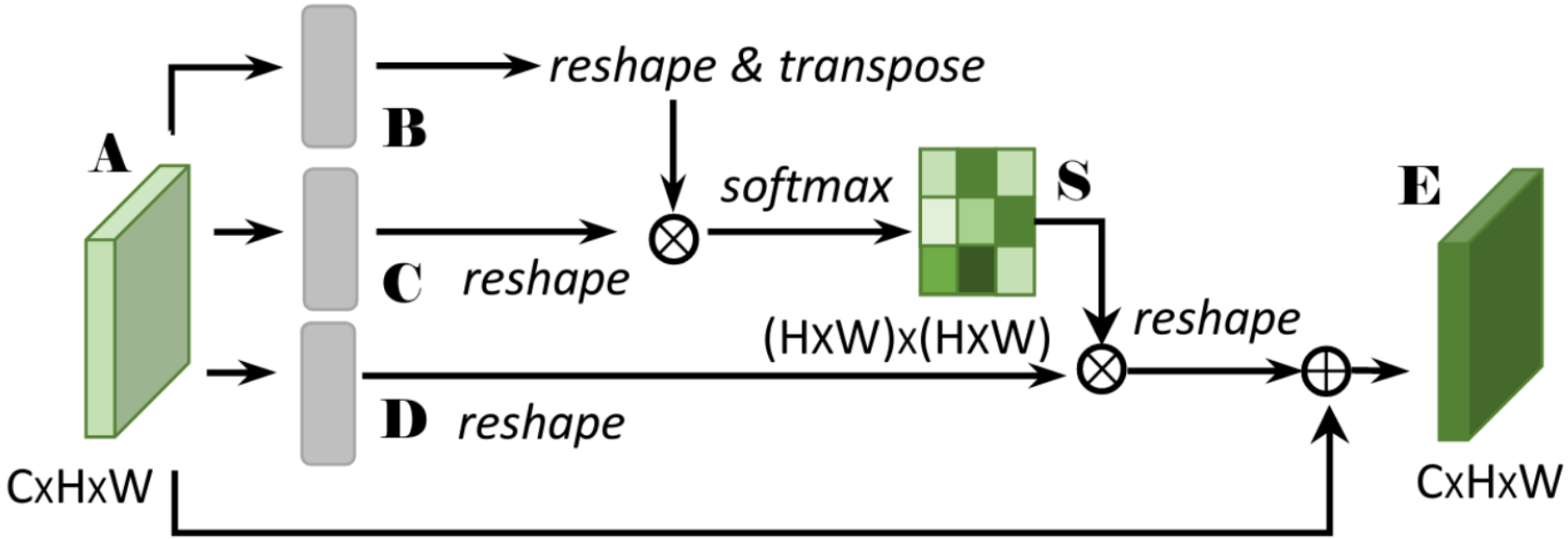
Woo et al., CBAM, 2018

Dual Attention

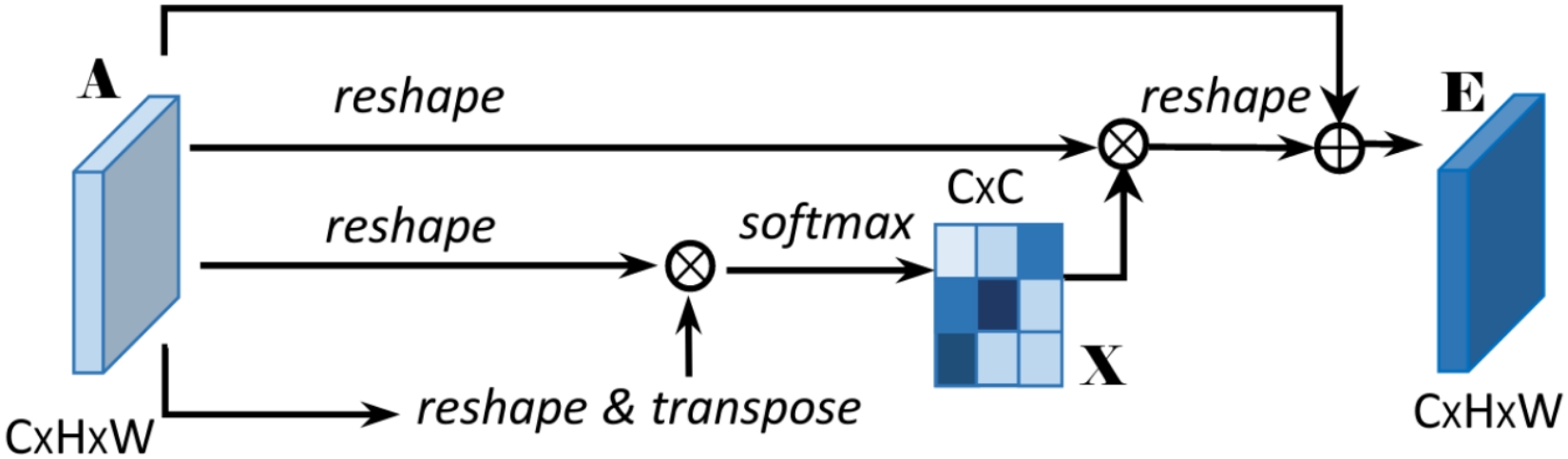


Fu et al., DANet, 2019

Dual Attention



A. Position attention module



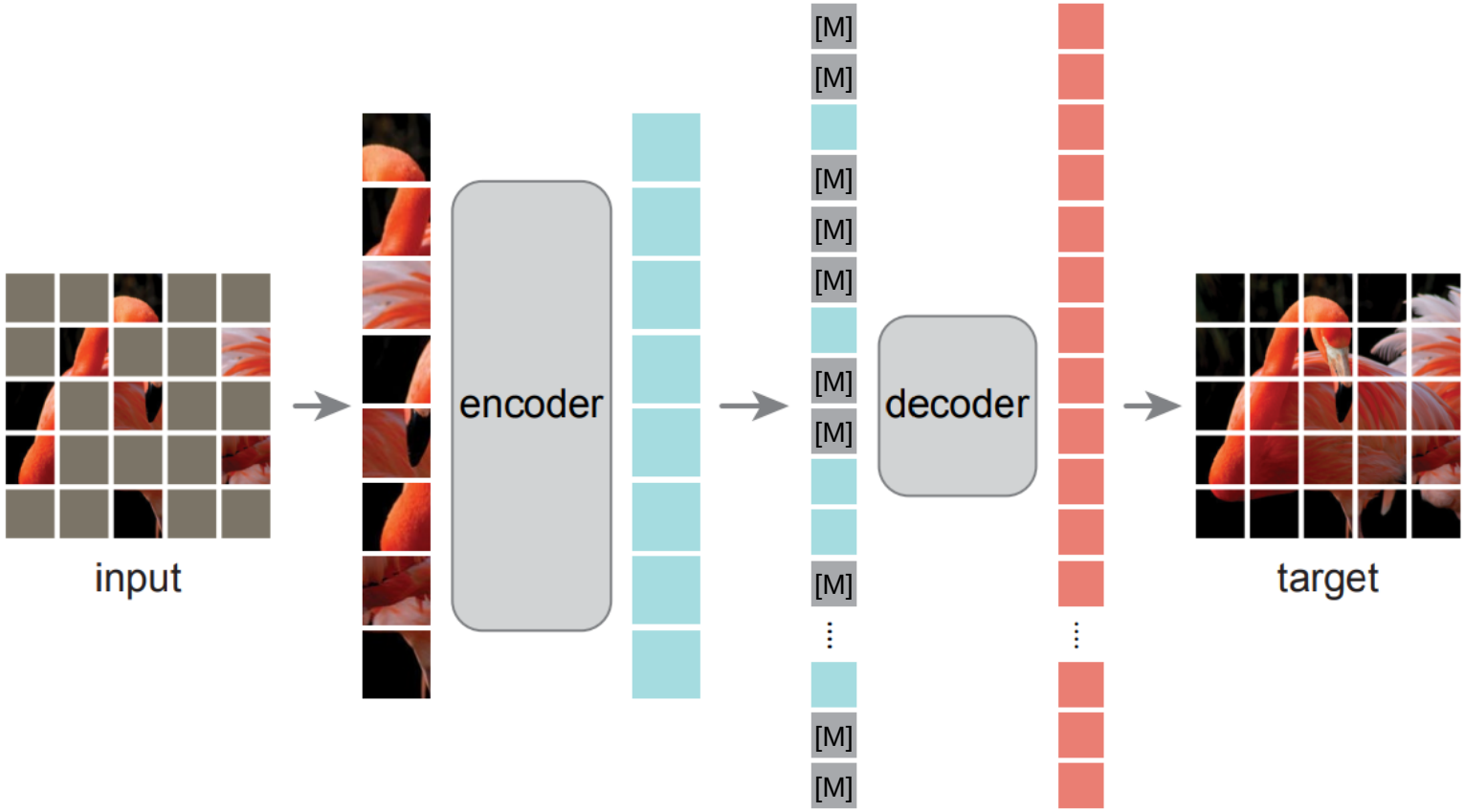
B. Channel attention module

Self-supervised training

- ViT requires a lot of training data
- Pretraining + fine-tuning
- Pretraining methods:
  - Mask Autoencoder (MAE)
  - Self-distillation with no labels (DINO)

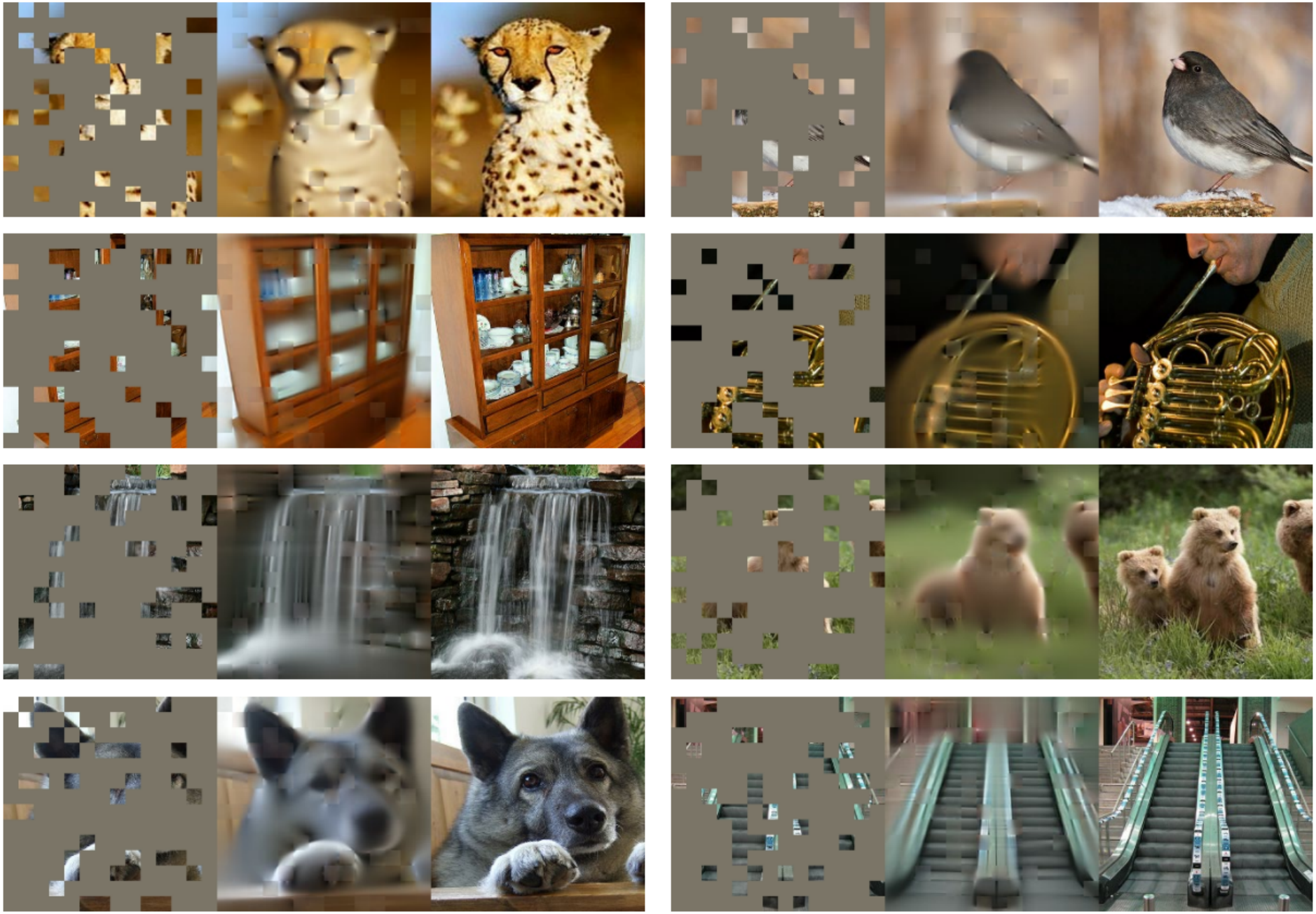
MAE

- Similar to BART pretraining



He et al., Masked Autoencoders, 2021

MAE

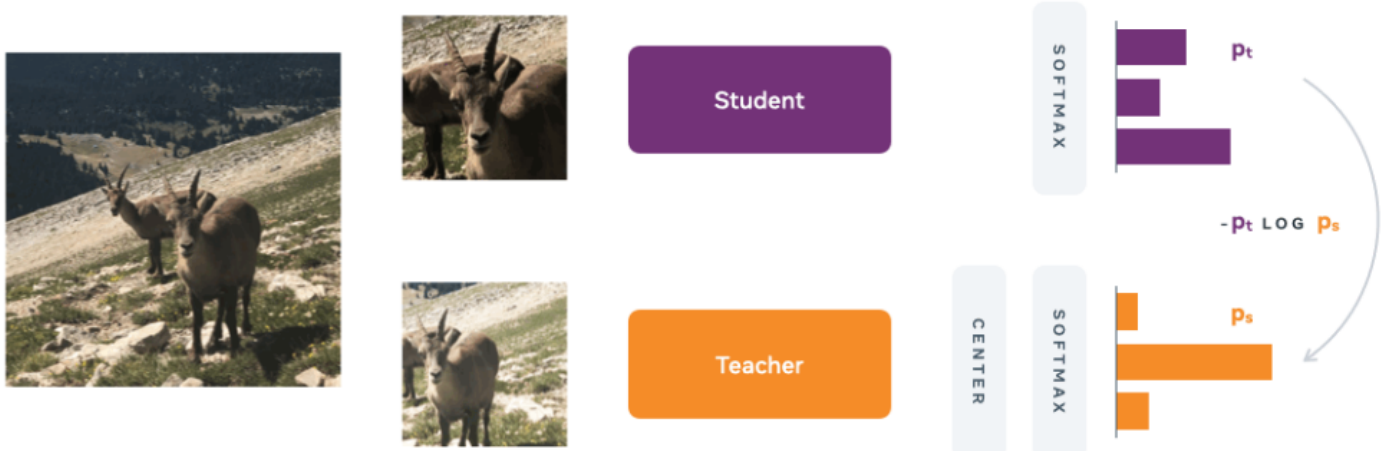


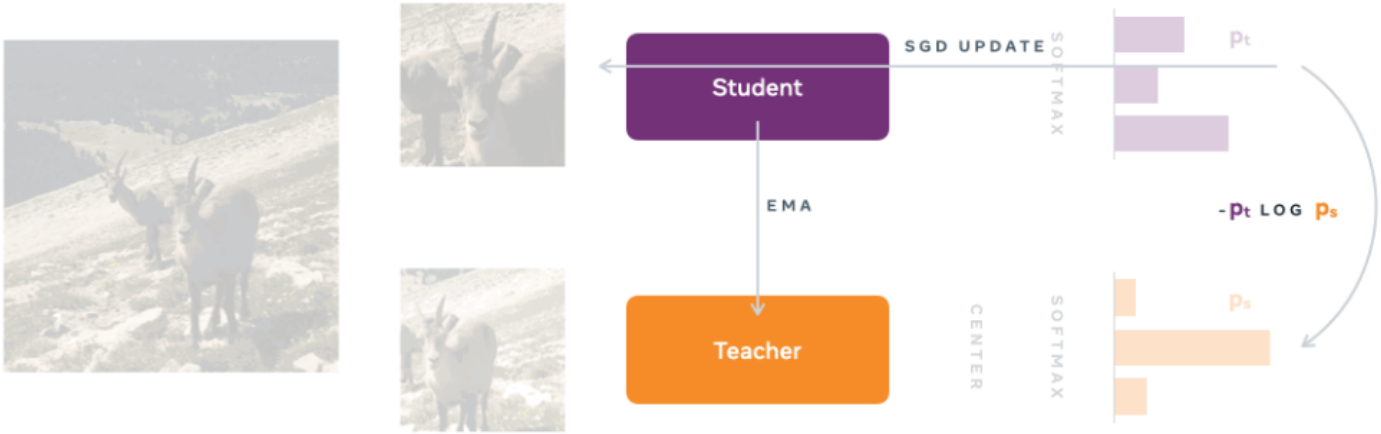
DINO

- Several local views
  - small areas of the original image (0.05 - 0.4)
  - 96x96 pixels
  - For the student
- Two global views
  - large area of the original image (0.4 – 1.0)
  - 224x224 pixels
  - Only for the teacher

Caron et al., DINO, 2021

Student
Teacher
SOFTMAX
SOFTMAX
CENTER
Pt
Ps

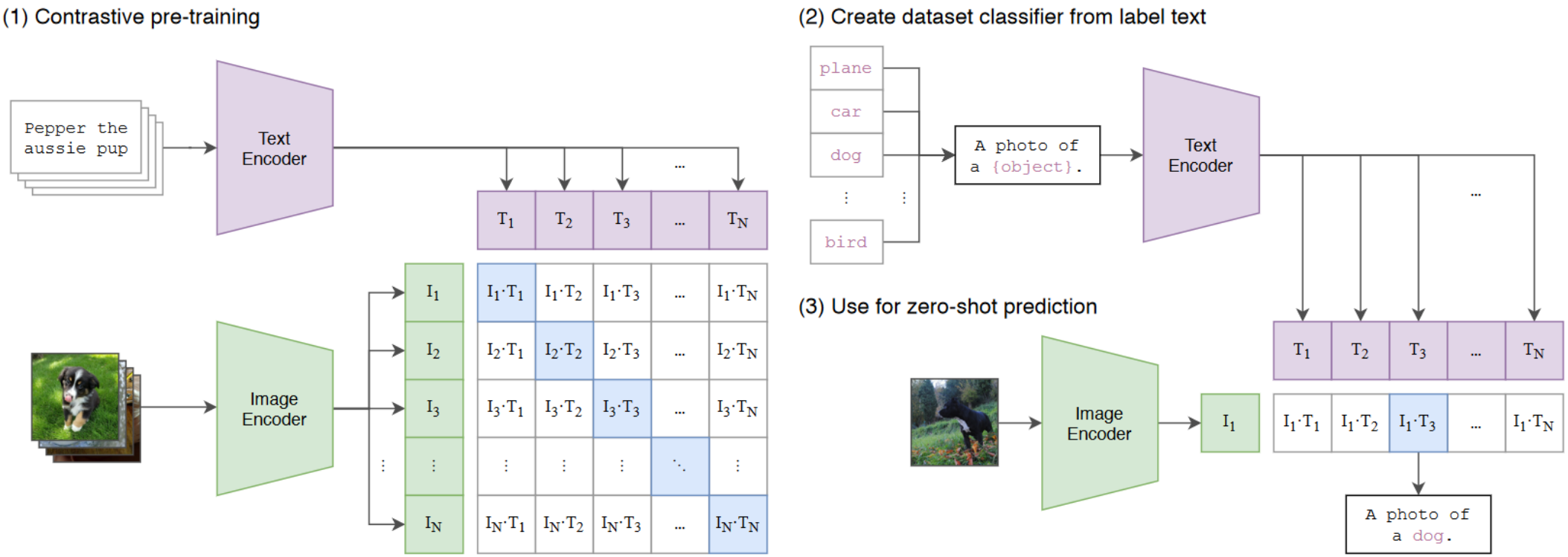

Student
Teacher
SOFTMAX
CENTER
SOFTMAX
pt
ps
-pt LOG ps



Avoid collapse: centering and sharpening

CLIP

- Contrastive Language-Image Pretraining



Radford et al., CLIP, 2021