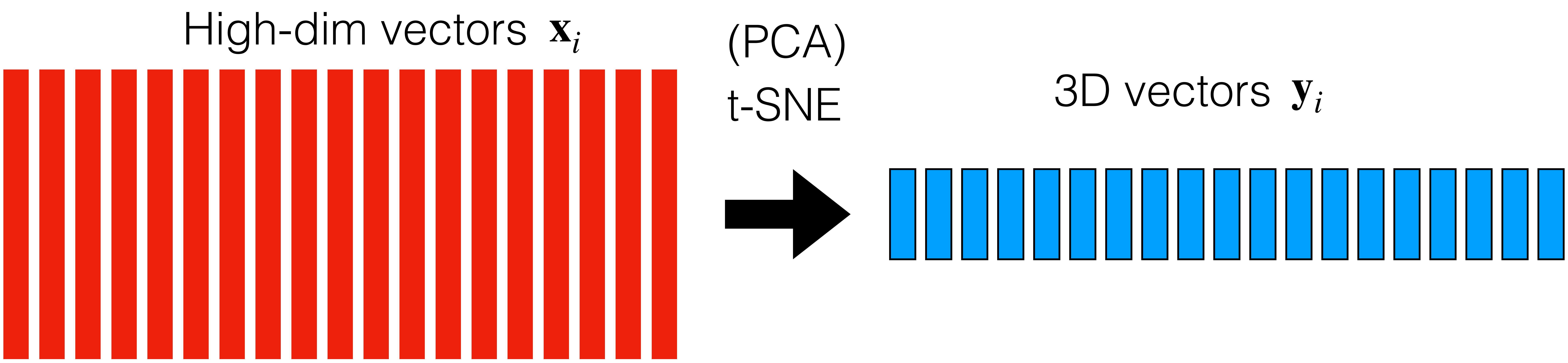
Transformers

Word and image embeddings with global attention.

Karel Zimmermann
Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

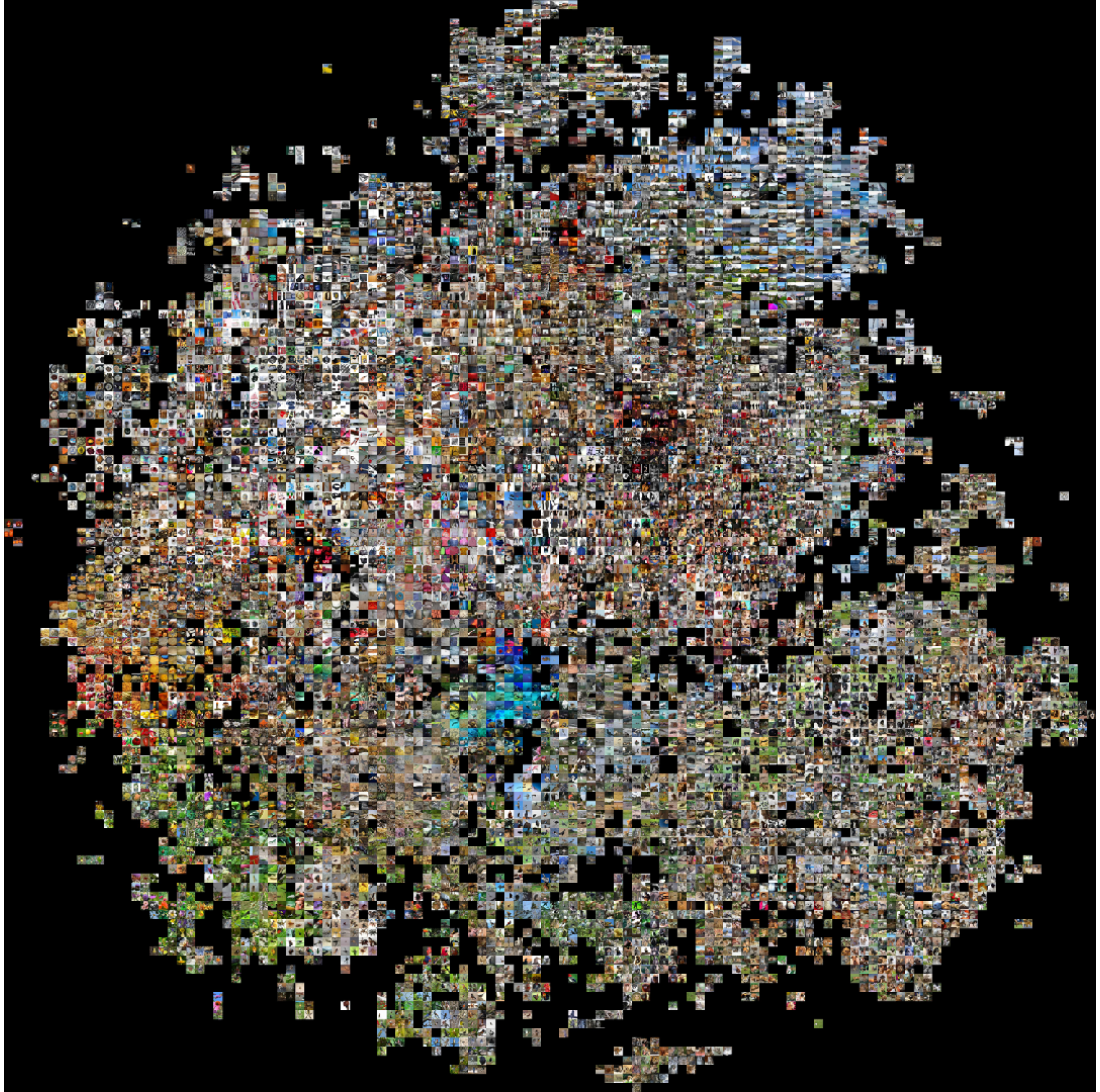
Pre-requisity: Visualizing high-dimensional data

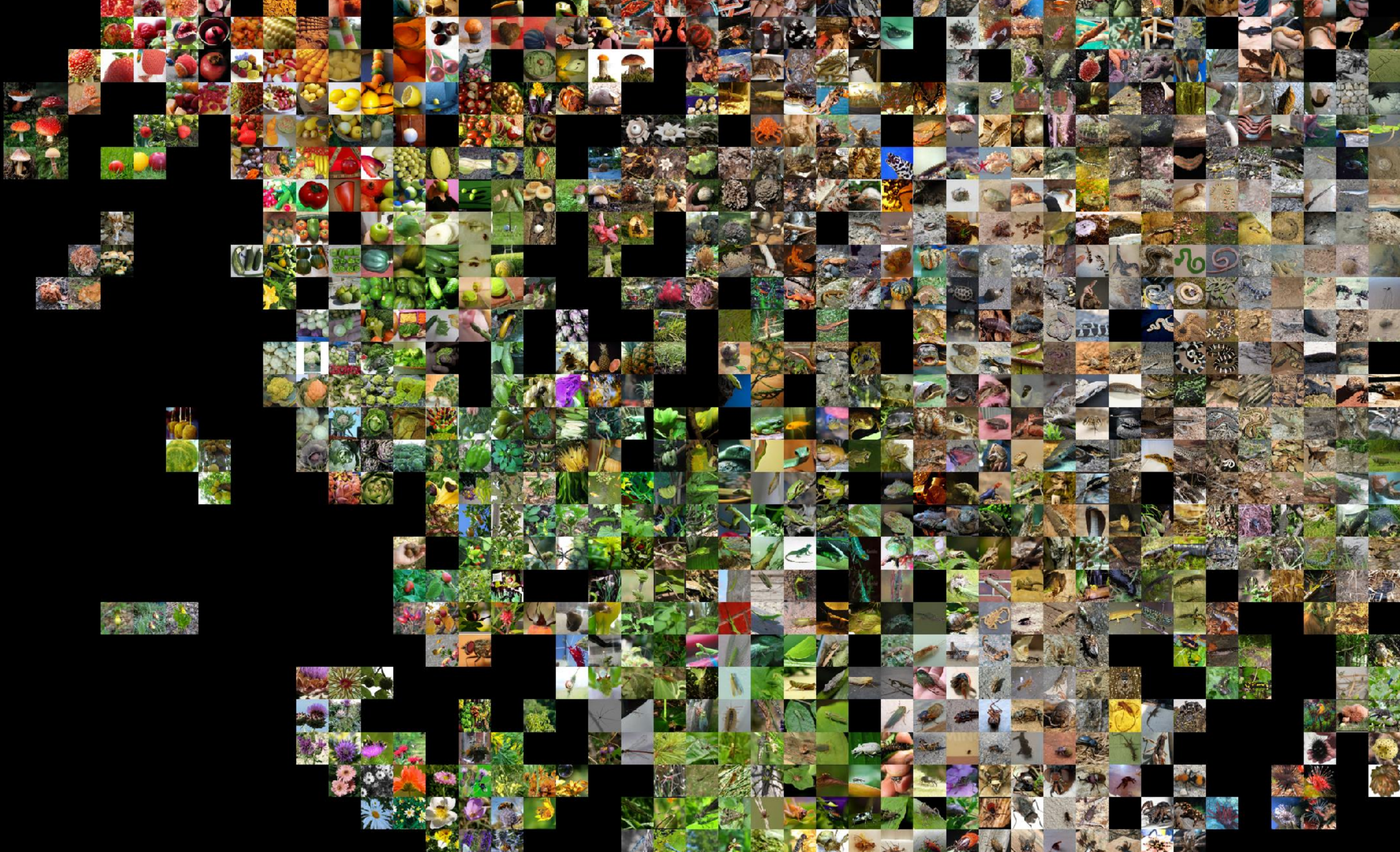
Visualizing high-dimensional embedding in 2D/3D world.



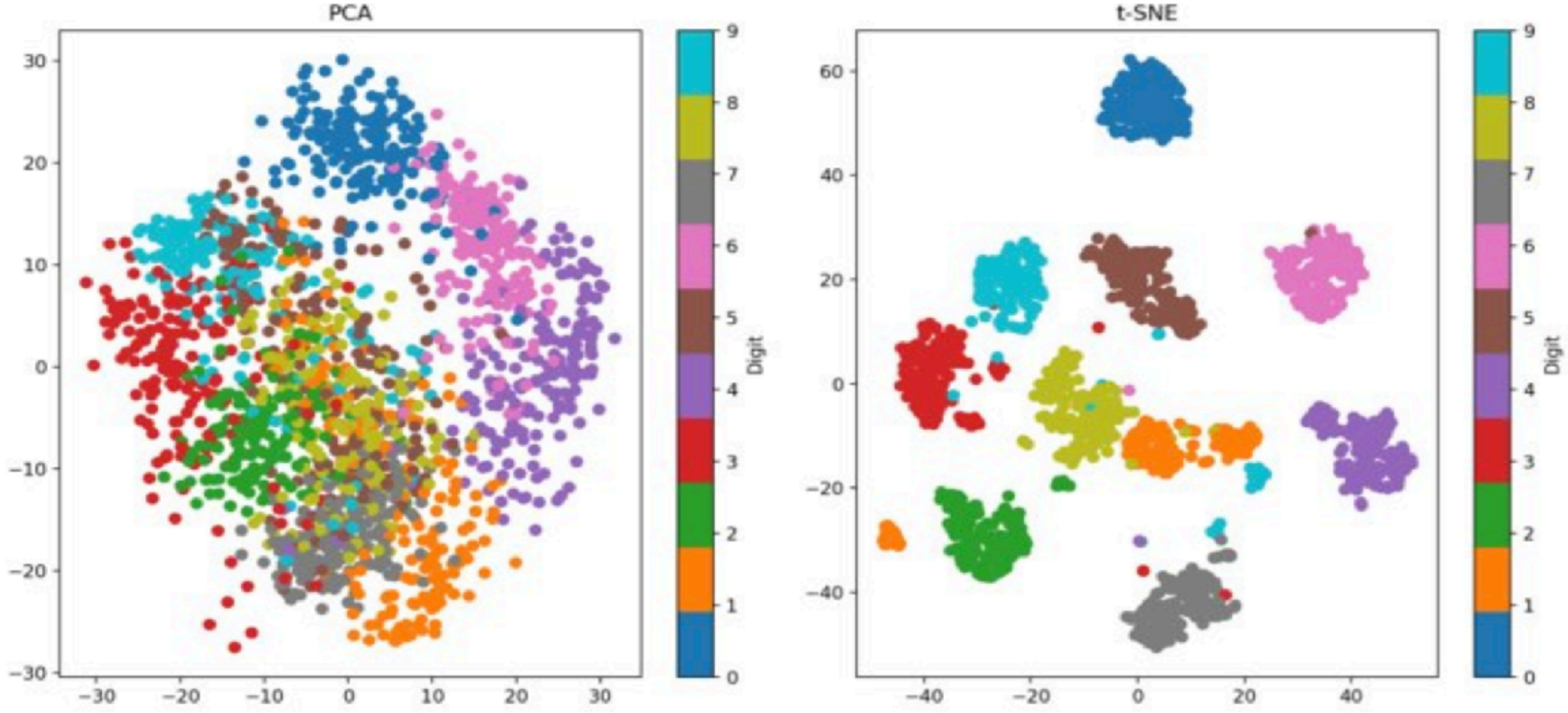
1. Randomly initialize $\mathbf{y}_i$ by normal zero-mean noise $\mathcal{N}(0, 0.001)$
2. Compute pair-wise probabilities in $\mathbf{x}_i$: $p_{ij} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$ Gaussian distribution
3. Compute pair-wise probabilities in $\mathbf{y}_i$: $q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$ t-distr. with heavier-tails (crowding)
4. Optimize $\mathbf{y}_i$ to get similar distribution $\mathrm{KL}(P\|Q) = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$

Visualizing high-dimensional embedding in 2D/3D world.





Visualizing high-dimensional embedding in 2D/3D world.



- t-SNE (t-distributed Stochastic Neighbor Embedding)
  - Captures **non-linear** relationships in data
  - Separate clusters based on their high-dimensional proximity
  - Outcome is stochastic and depends of perplexity σ

- PCA
  - Captures **linear** relationships in data
  - Deterministic and useful for preprocessing

Transformers in language (NLP)

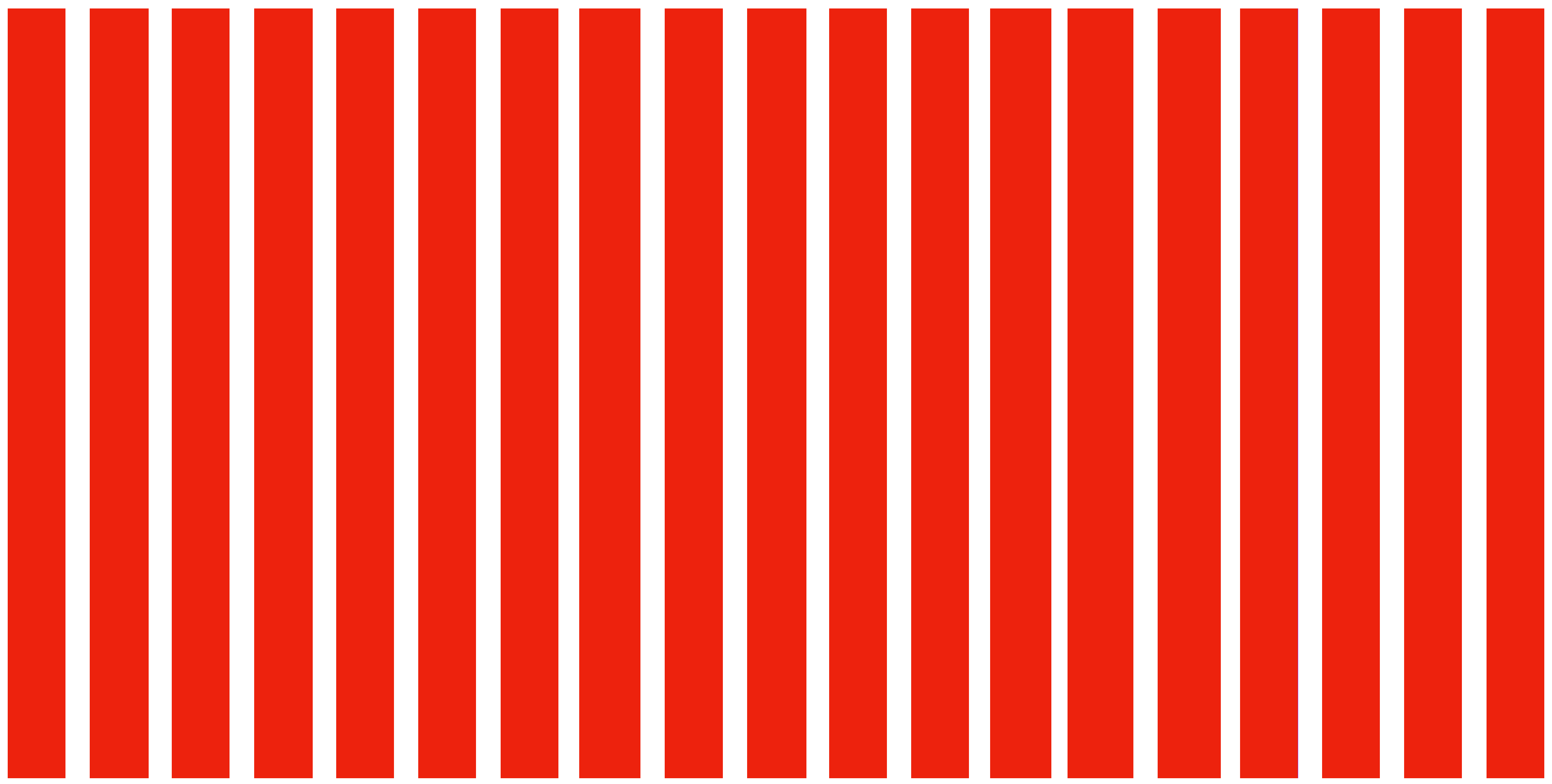
Word2vec represents words as low-dimensional continuous vectors [Mikolov NIPS 2013]

Word2vec represents words as low-dimensional continuous vectors

N-word vocabulary (one-hot enc.)

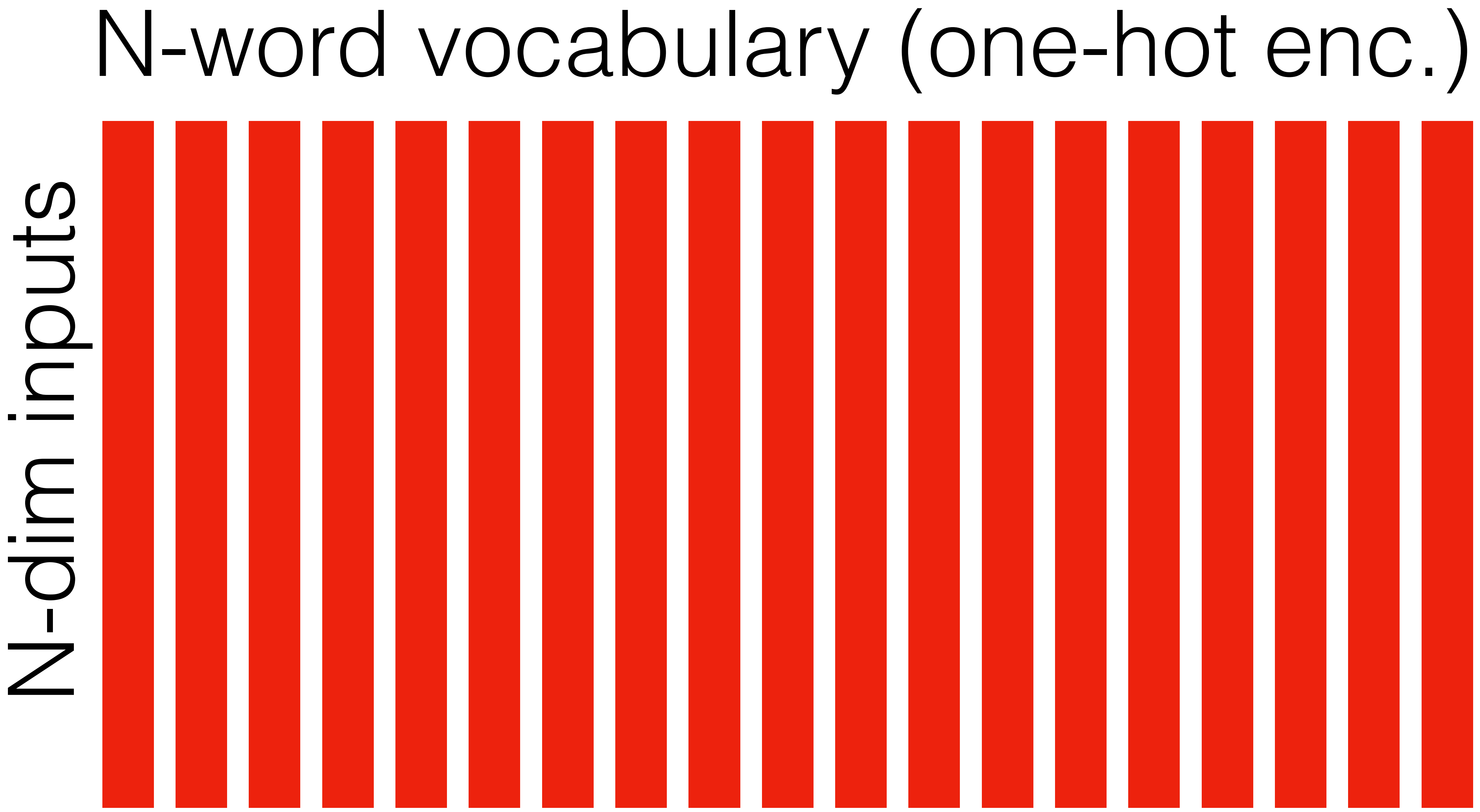
Karel is the best teacher in the whole world

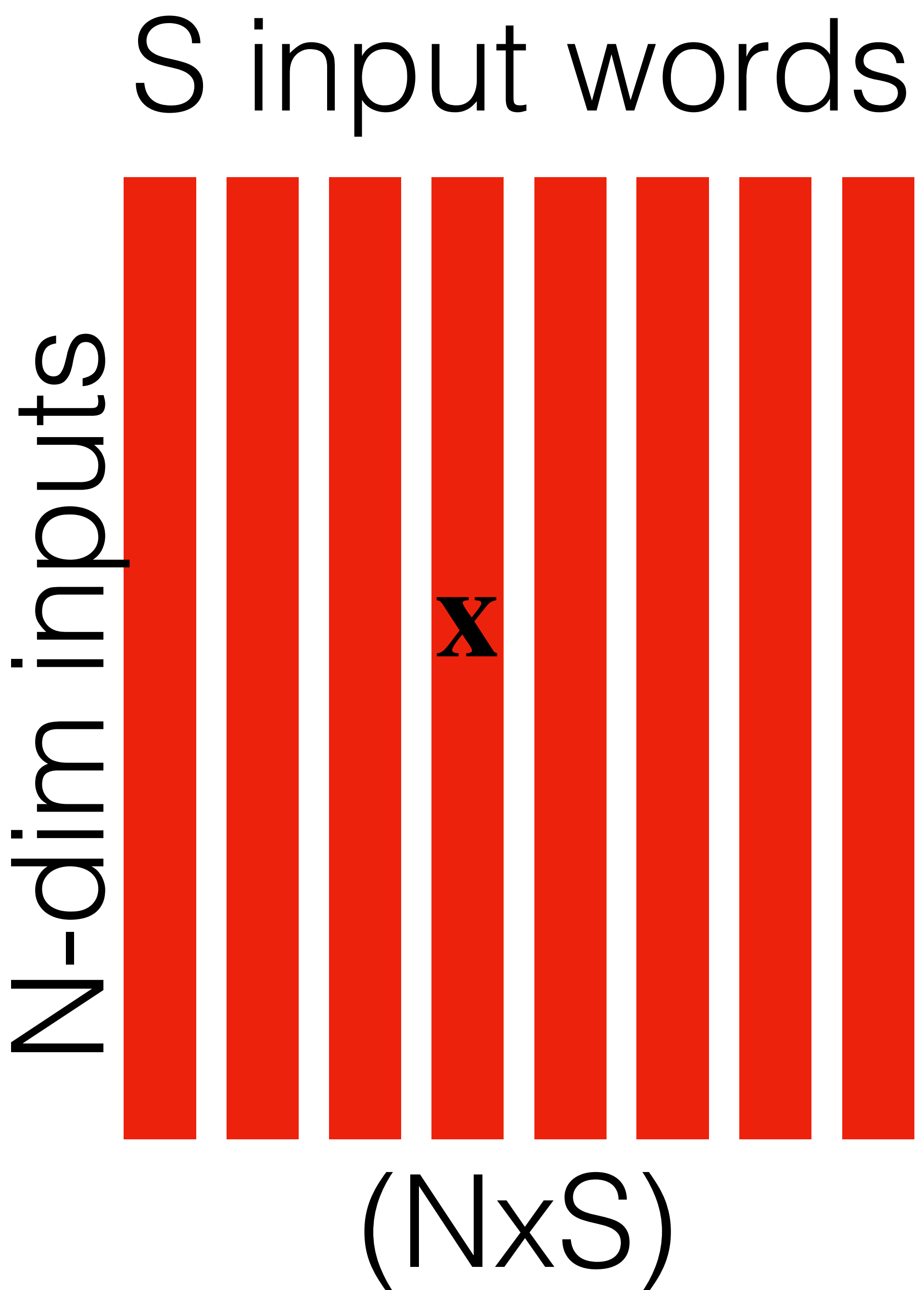
N-dim inputs



Word2vec represents words as low-dimensional continuous vectors

Word2vec represents words as low-dimensional continuous vectors





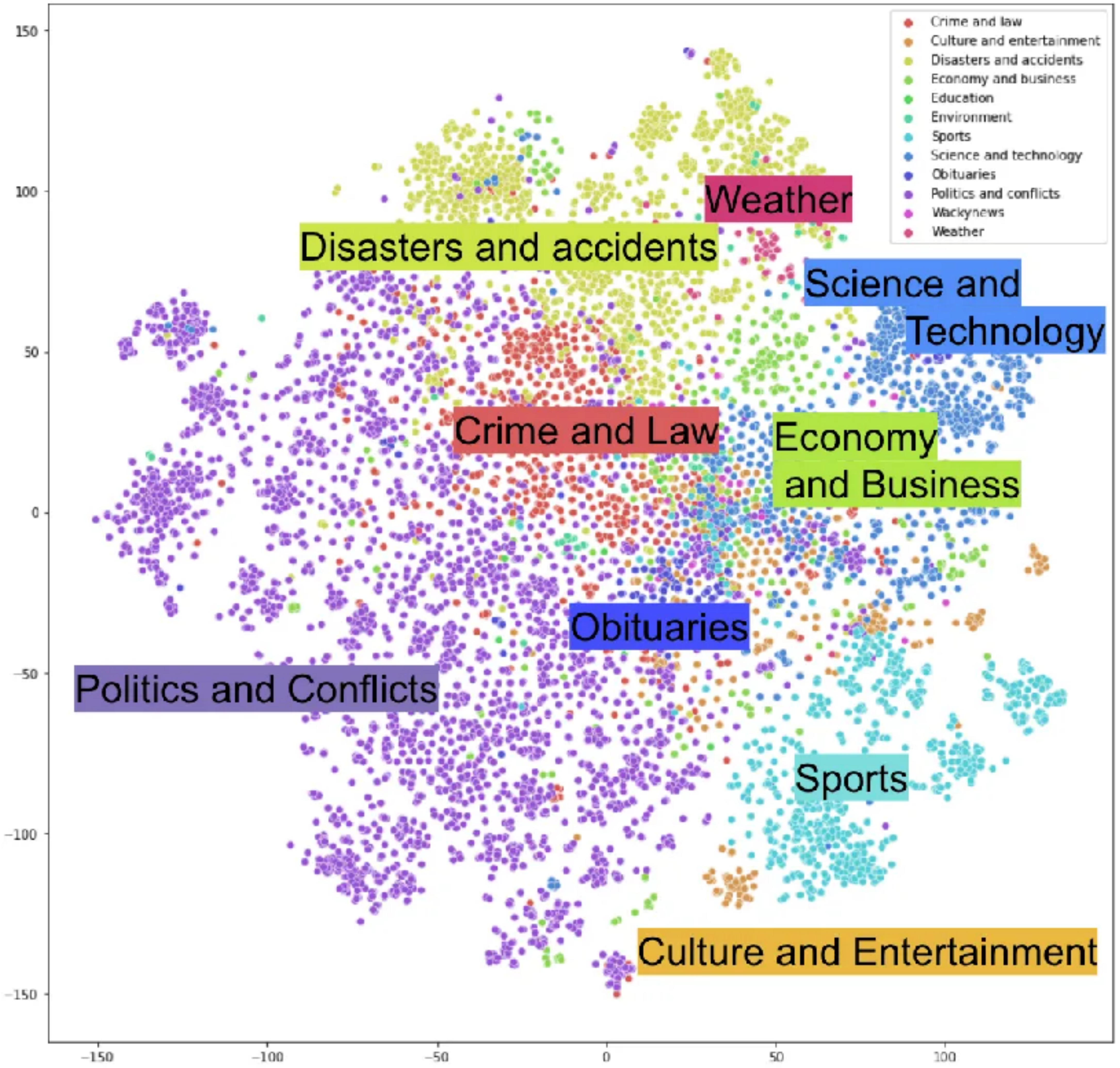
Word2vec represents words as low-dimensional continuous vectors

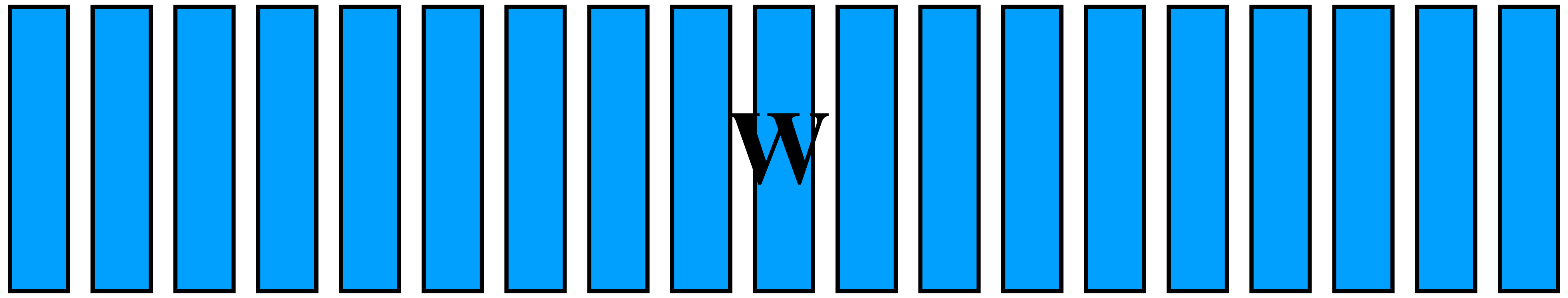
Word2vec represents words as low-dimensional continuous vectors

Word2vec represents words as low-dimensional continuous vectors

Word2vec represents words as low-dimensional continuous vectors

Word2vec represents words as low-dimensional continuous vectors

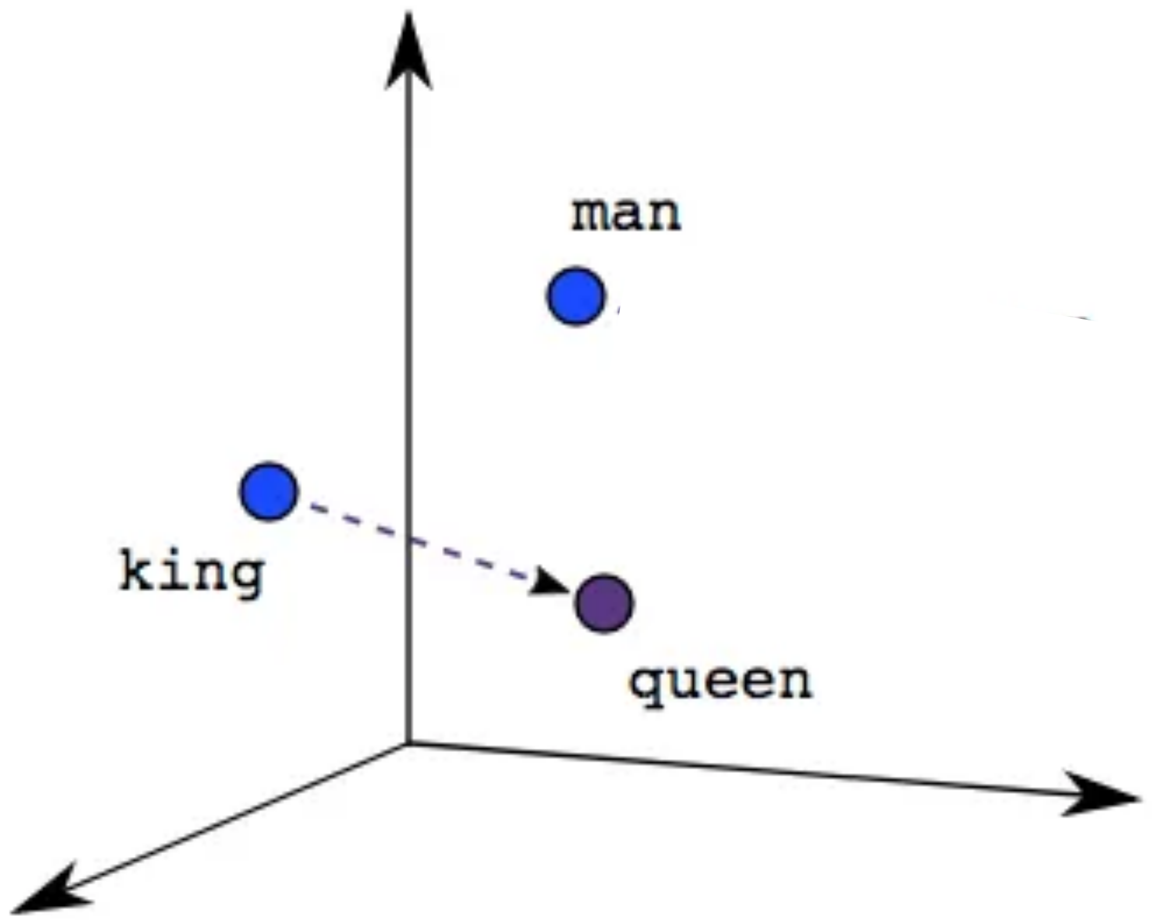




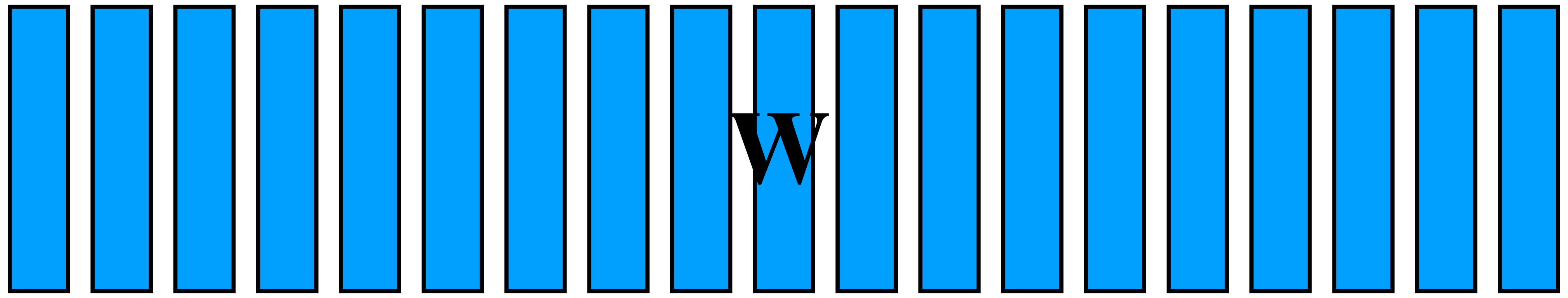
Word embedding

Word2vec represents words as low-dimensional continuous vectors

Word2vec represents words as low-dimensional continuous vectors

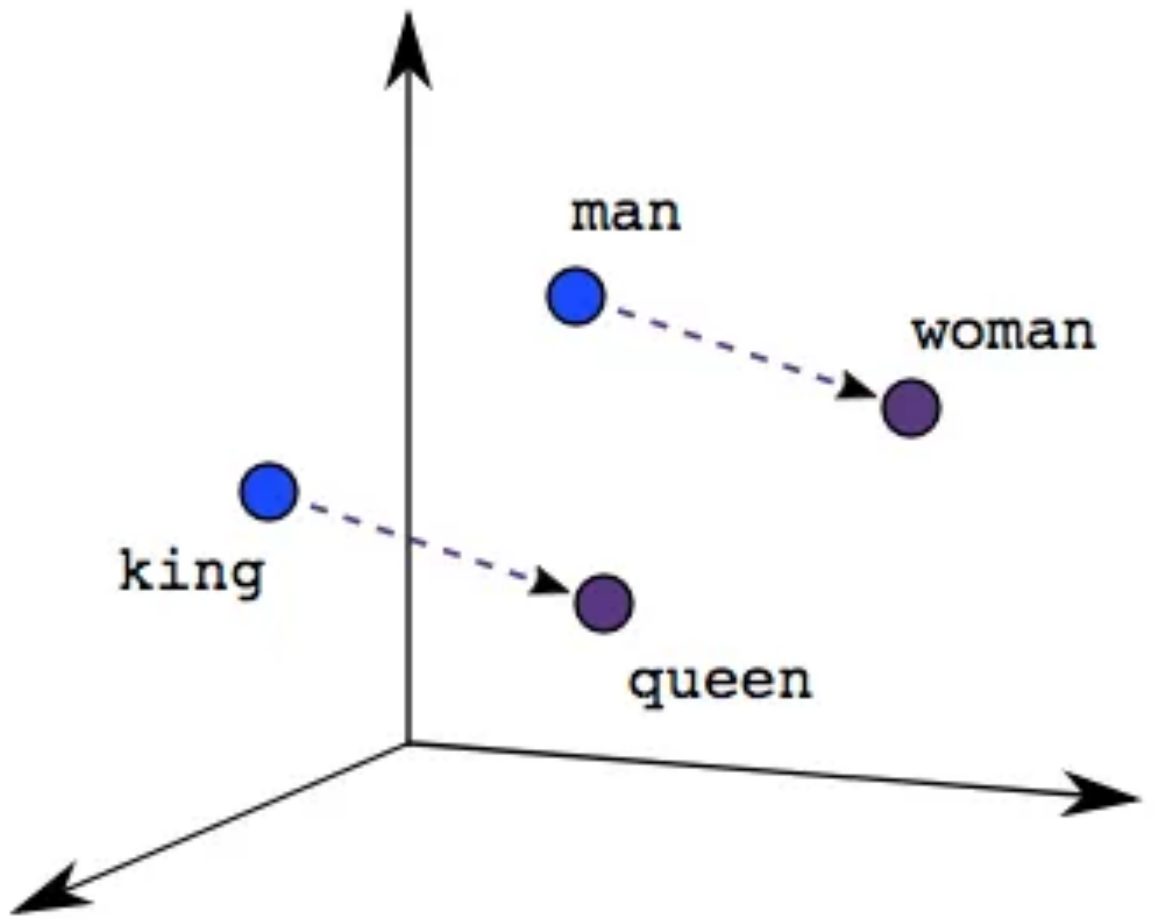


Male-Female



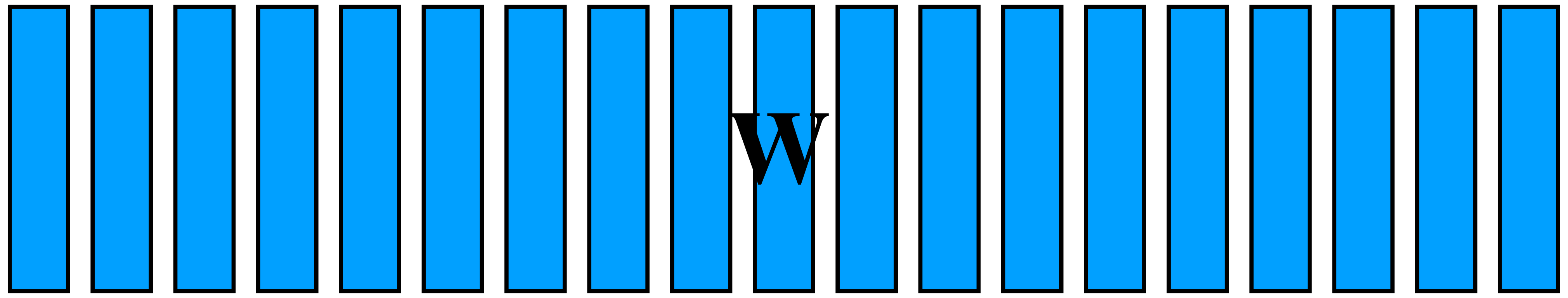
Word embedding

Word2vec represents words as low-dimensional continuous vectors



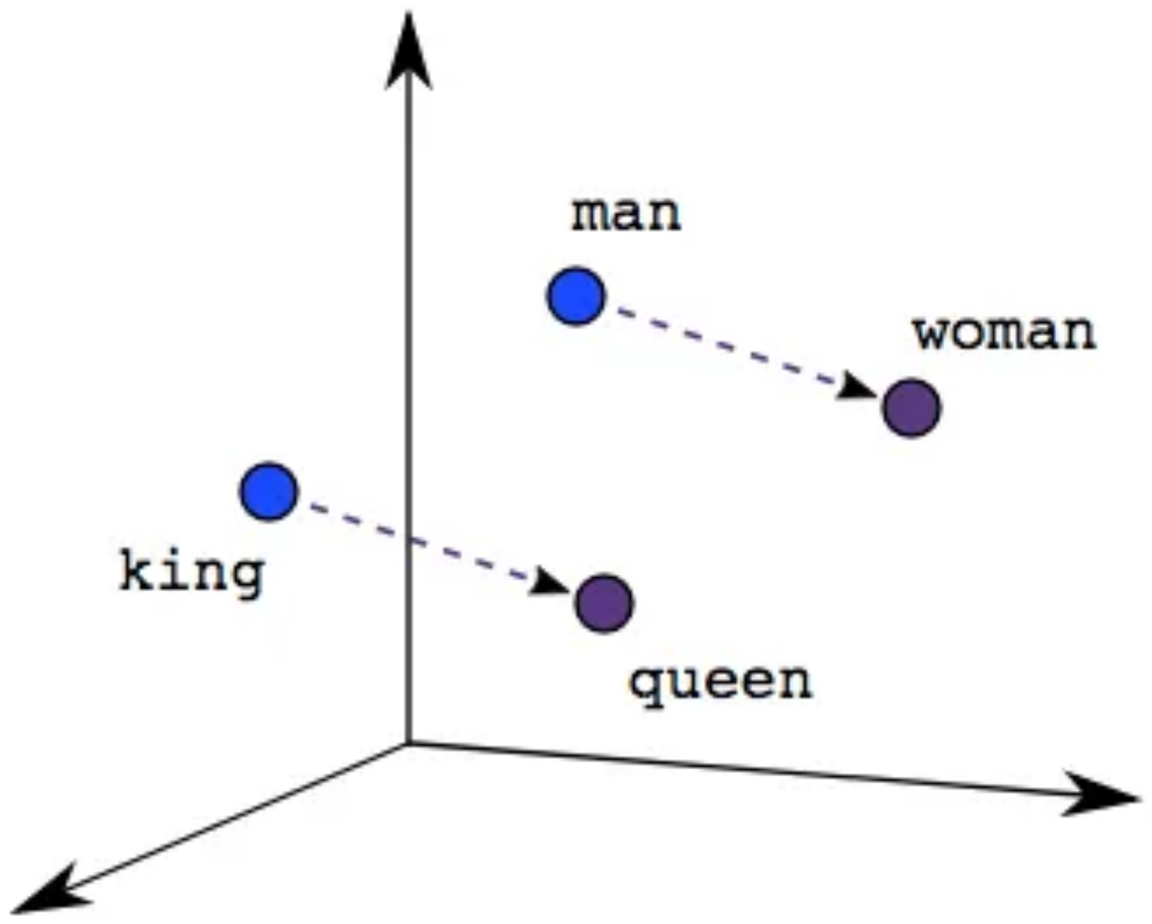
Male-Female

Word algebra: `king - man + woman = queen`

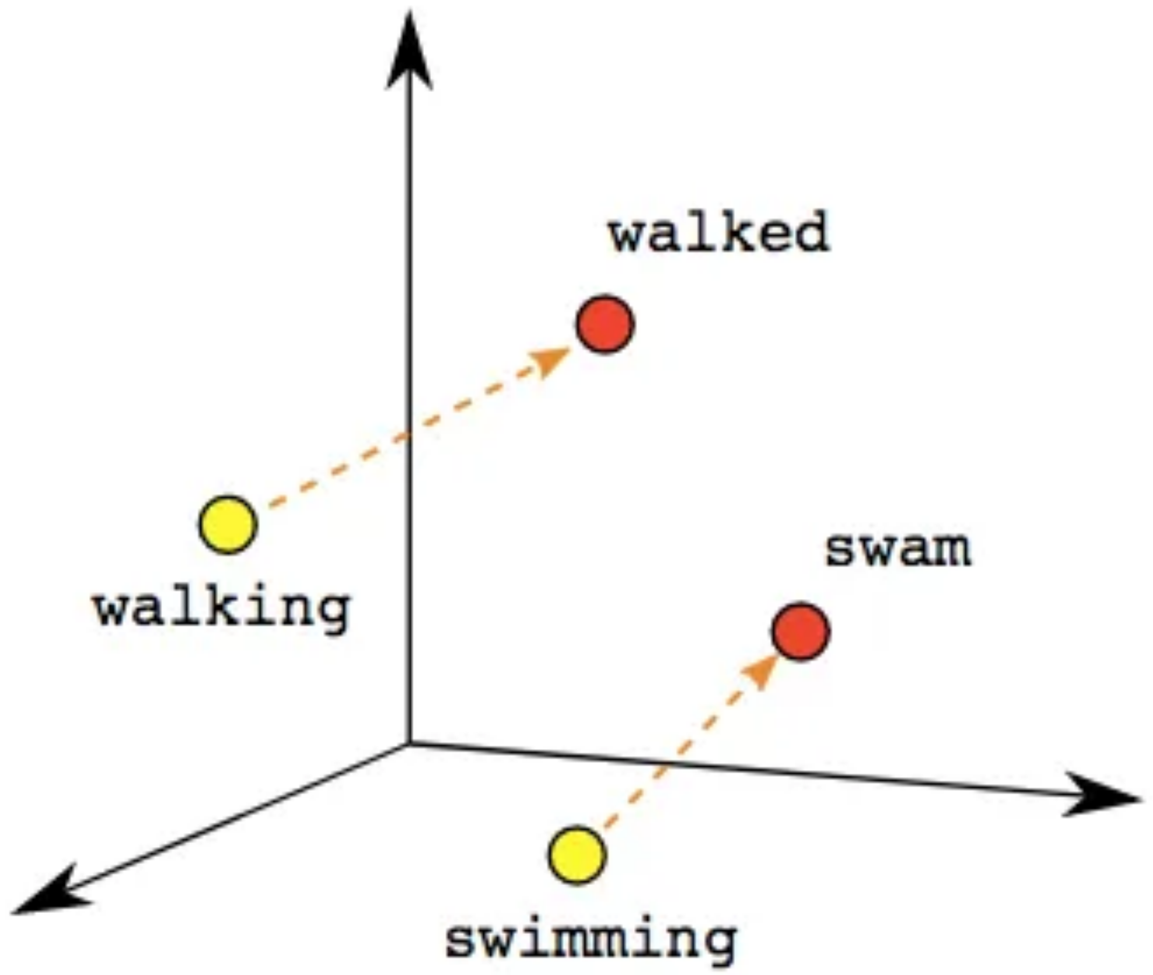


Word embedding

Word2vec represents words as low-dimensional continuous vectors

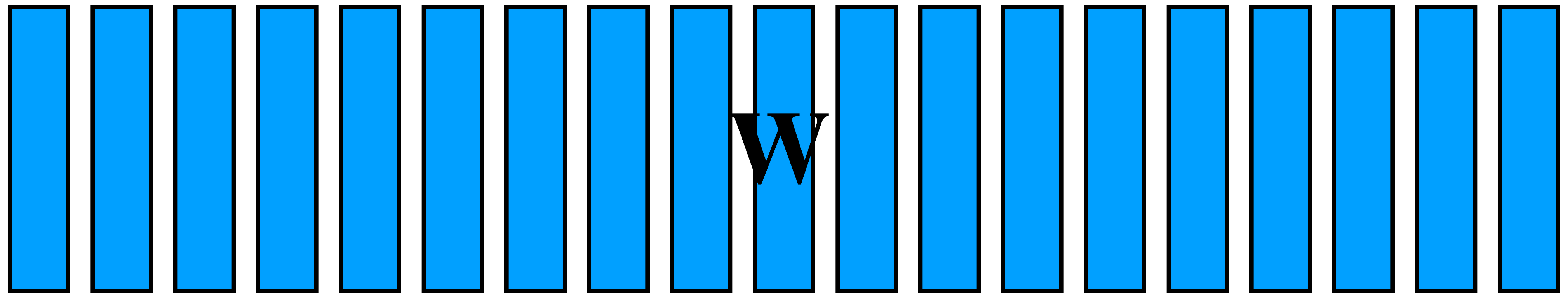


Male-Female



Verb Tense

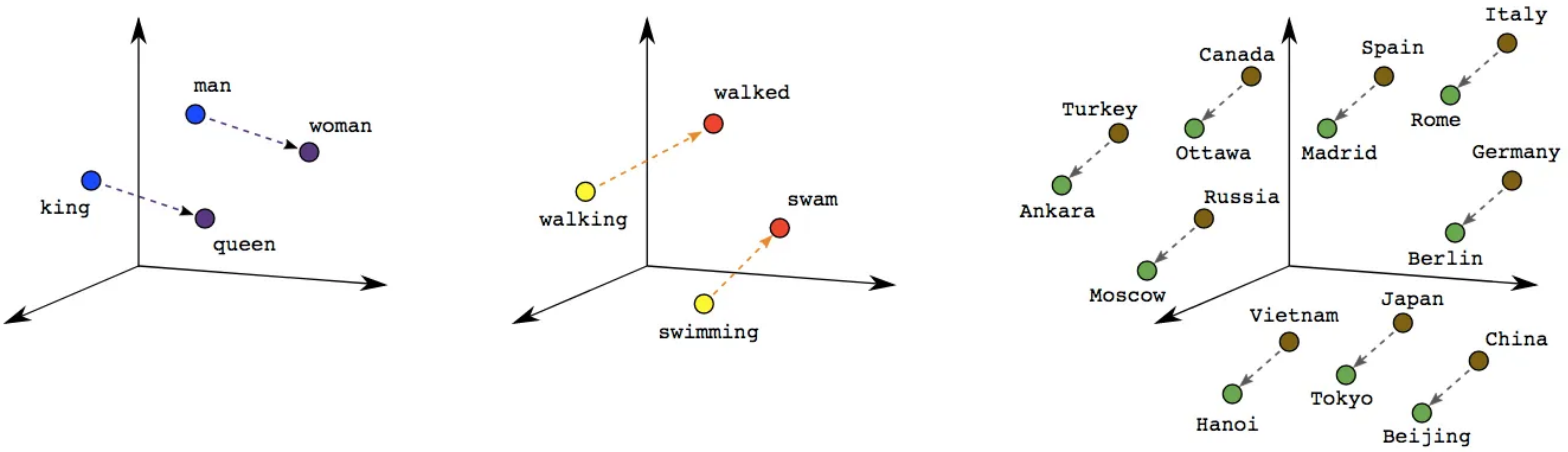
Word algebra: `king - man + woman = queen`



Word embedding

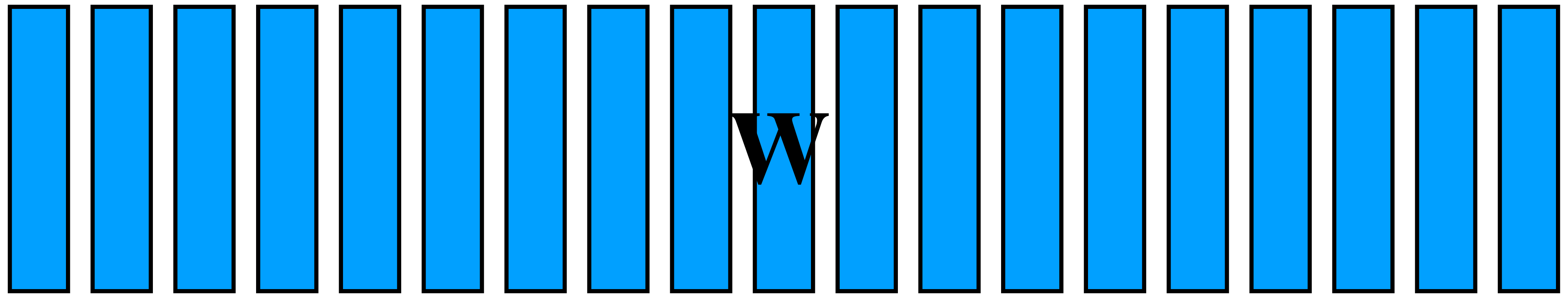
Word2vec represents words as low-dimensional continuous vectors

https://dash.gallery/dash-word-arithmetic/



Male-Female   Verb Tense   Country-Capital

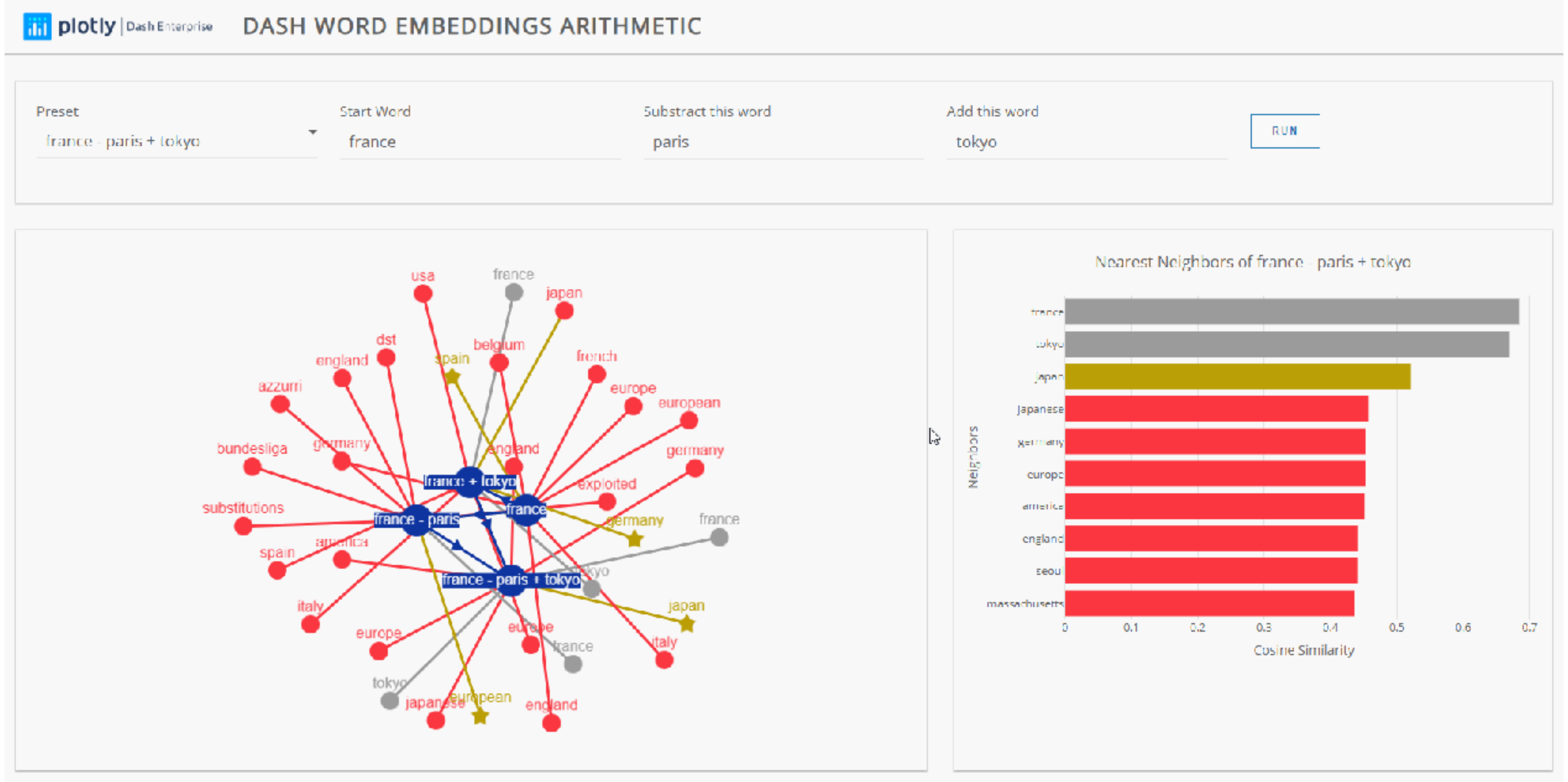
Word algebra: `king - man + woman = queen`



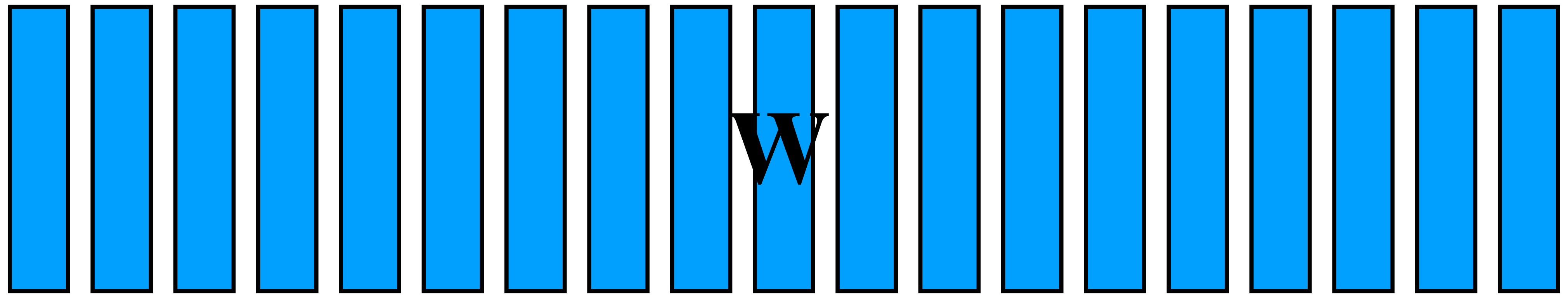
Word embedding

Word2vec represents words as low-dimensional continuous vectors

https://dash.gallery/dash-word-arithmetic/

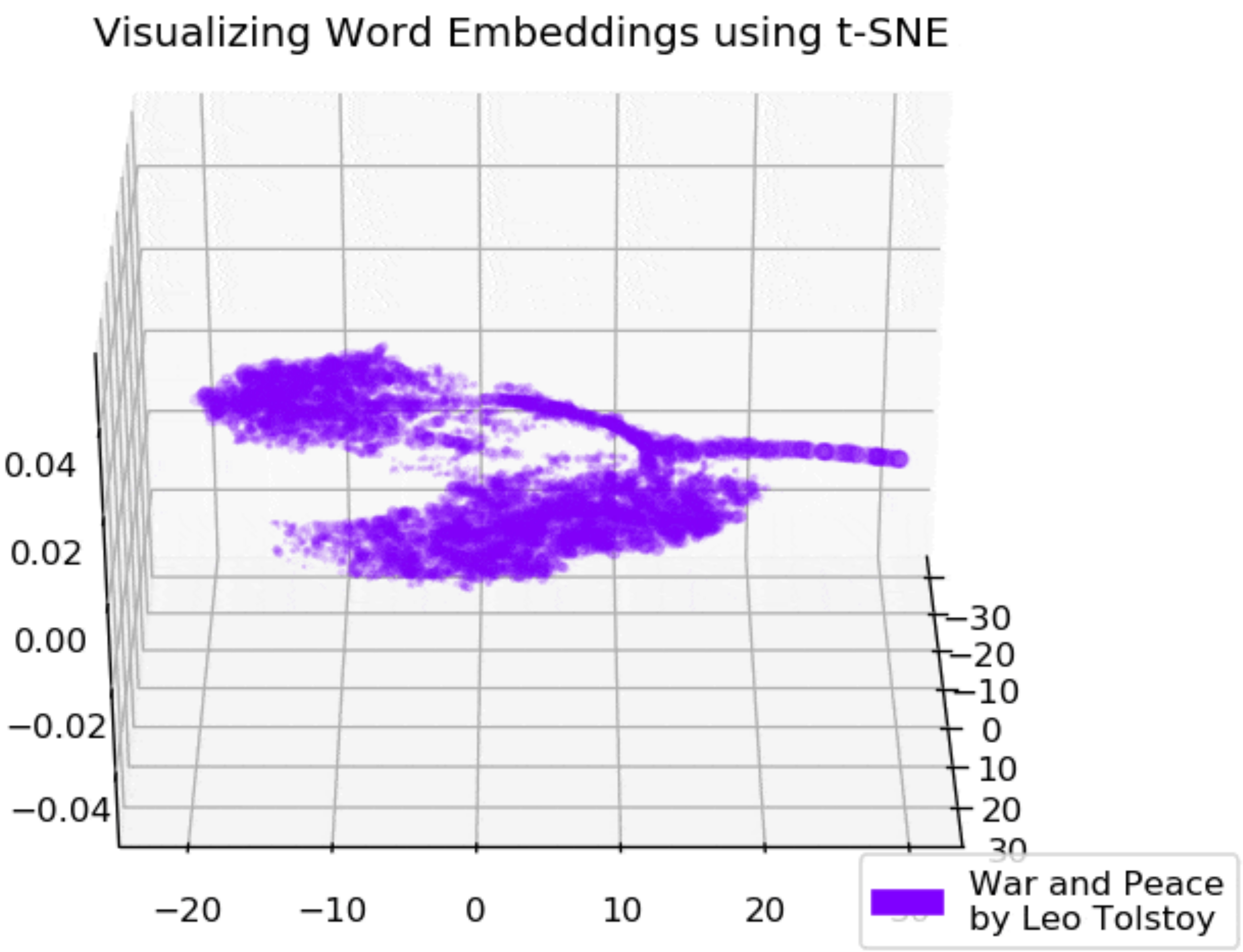


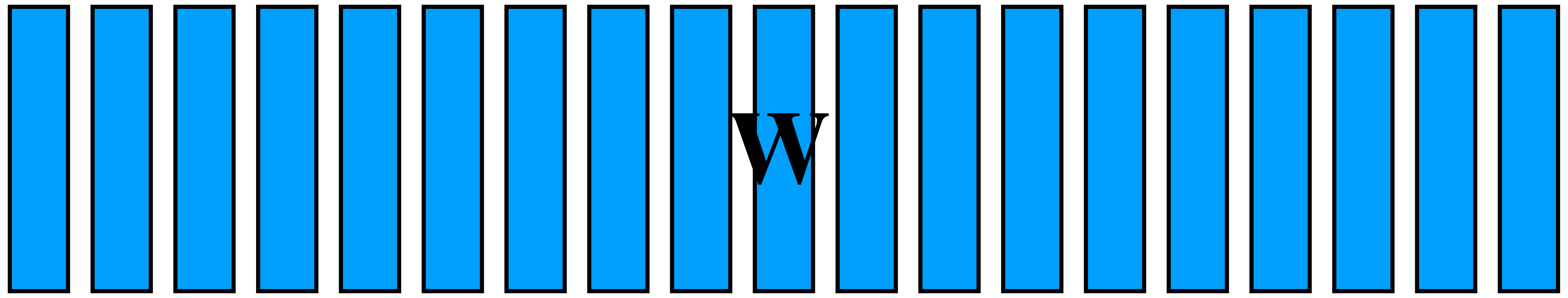
Word algebra: `king - man + woman = queen`



Word embedding

Word2vec represents words as low-dimensional continuous vectors

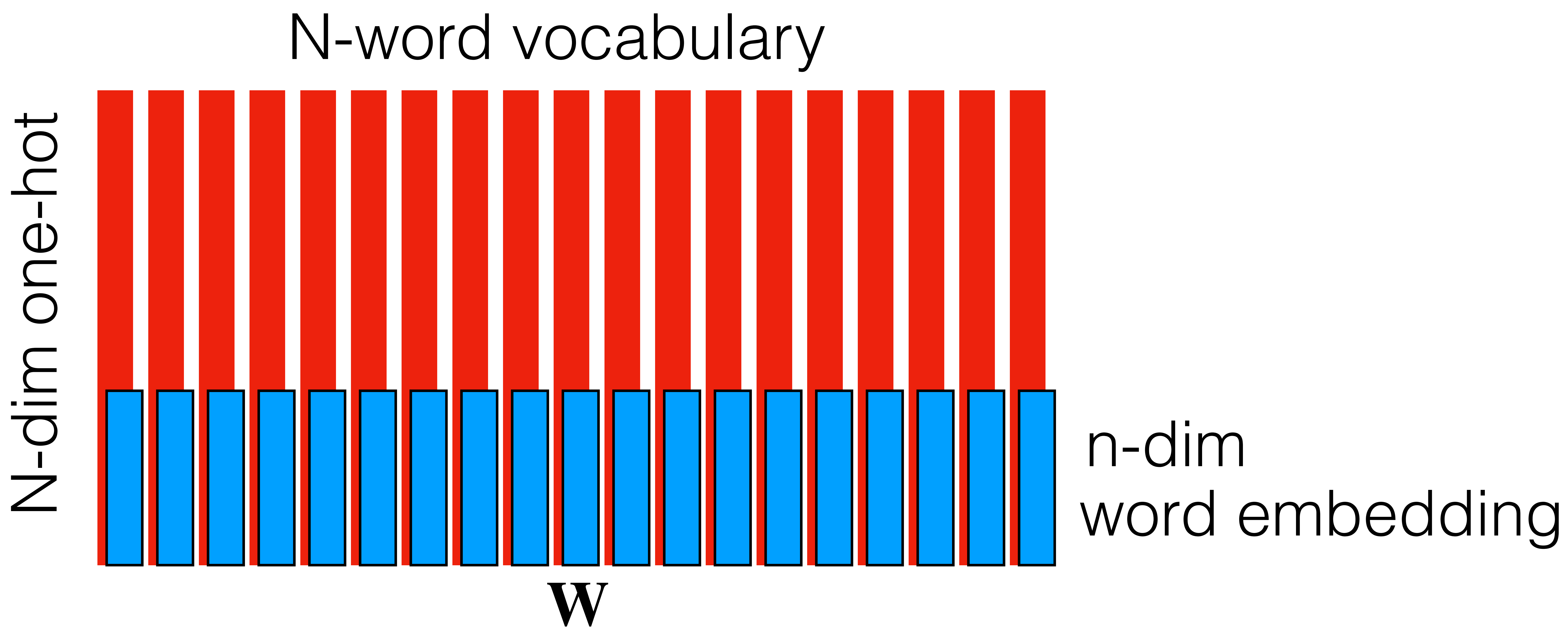




Word embedding

Word2vec represents words as low-dimensional continuous vectors

- **N N-dim orthonormal vectors** projected into **n-dim** space where **n<<N** (large-scale models BERT, GPT has n=768 –1024)
- How many orthogonal vectors in “n”-dim space?
- Can I represent only “n” independent concepts?

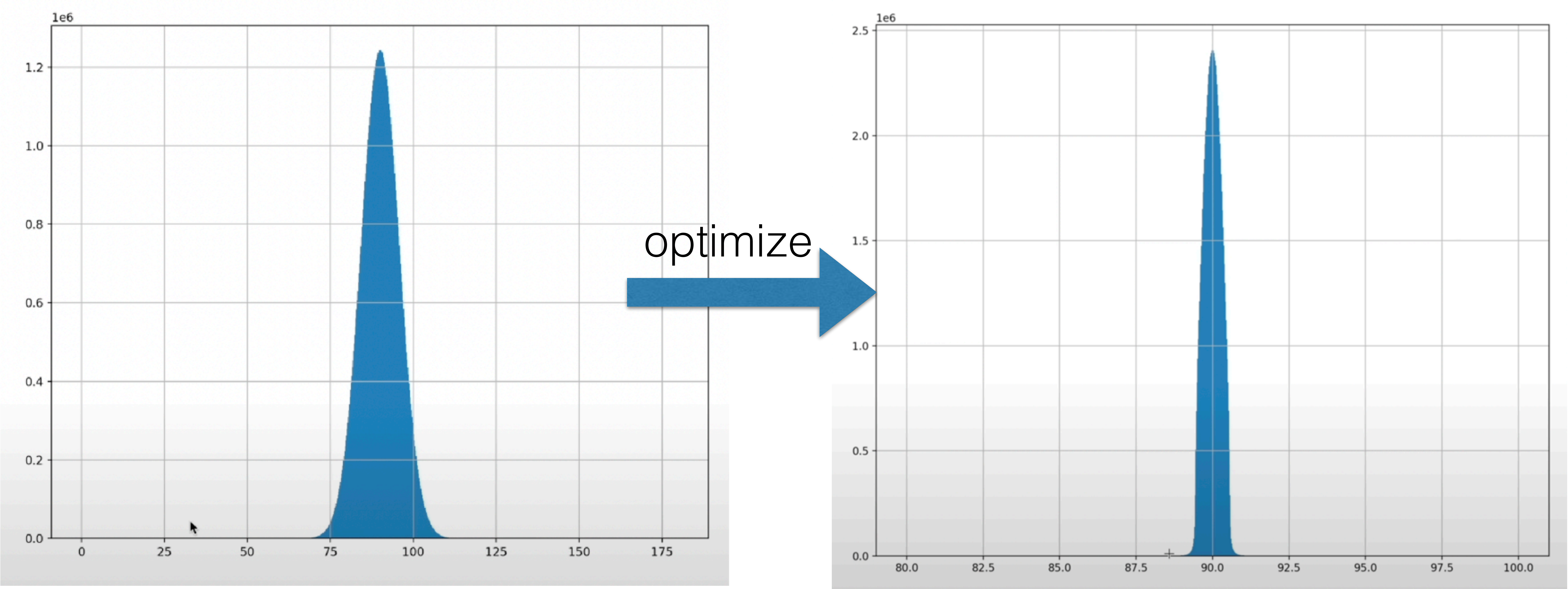


How many indepedent concepts do you fit in N-dimensional space?

- How many orthogonal vectors in “n”-dim space?

Angles between randomly generated 10000 vectors in 100-dimensional space

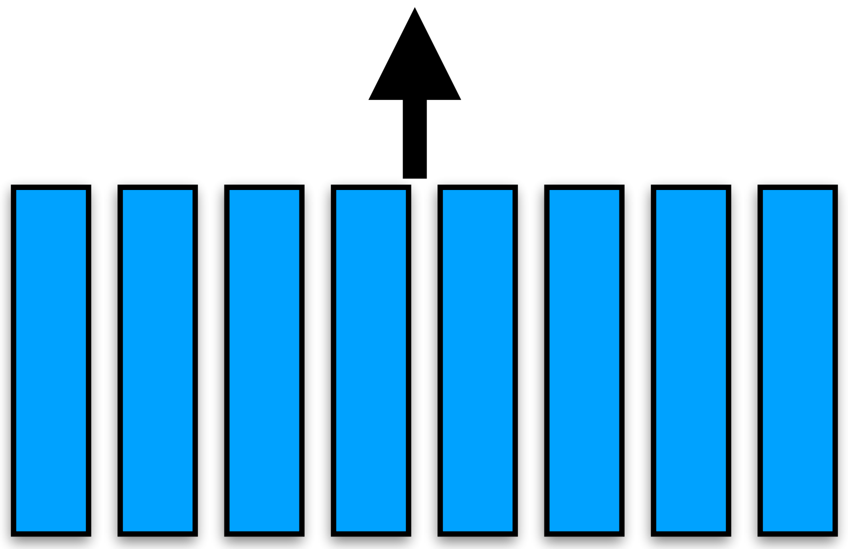
Optimized vectors are almost independent



consequence of Johnson-Lindenstrauss Lemma

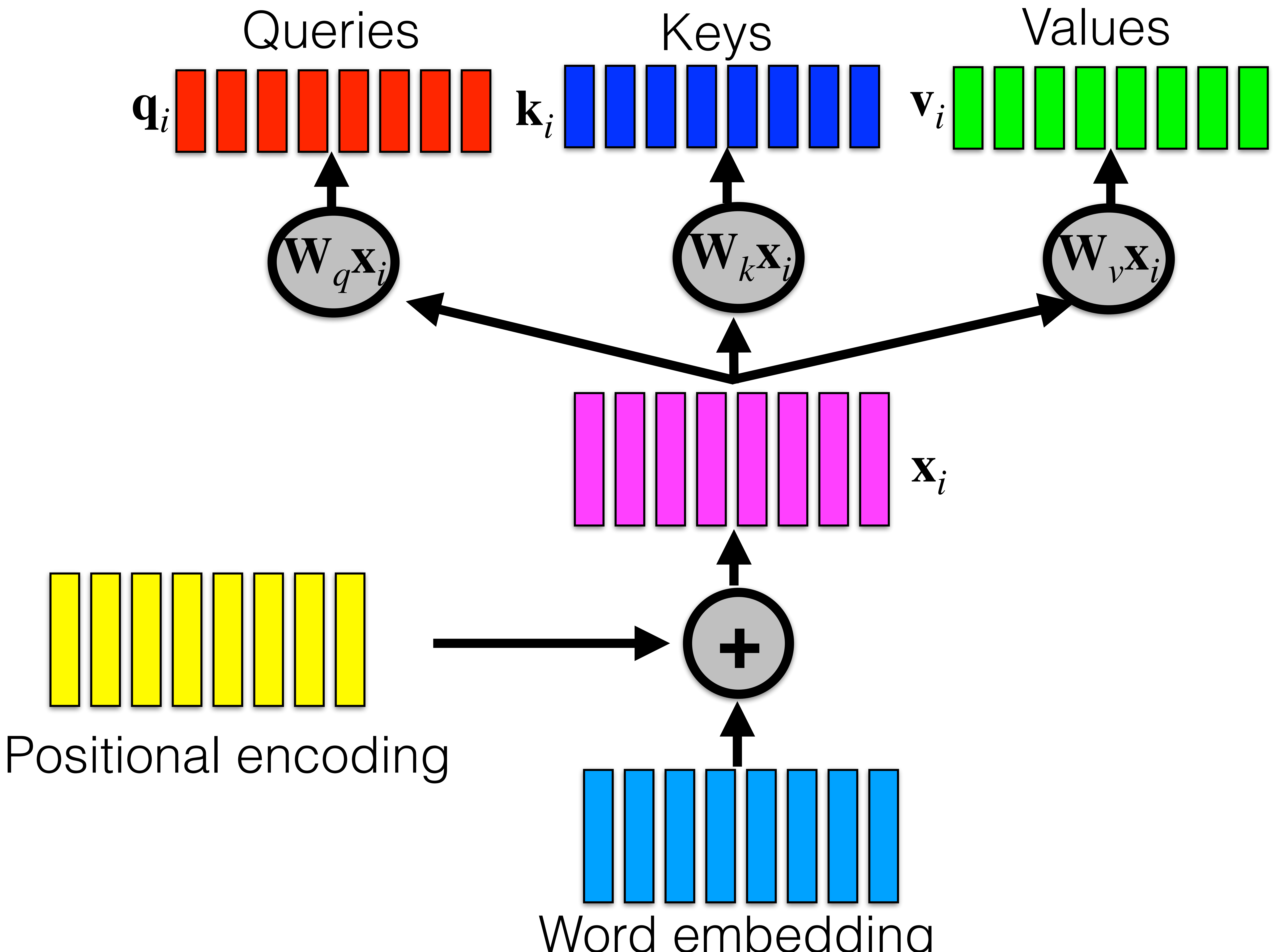
https://www.youtube.com/watch?v=9-Jl0dxWQs8

encoder

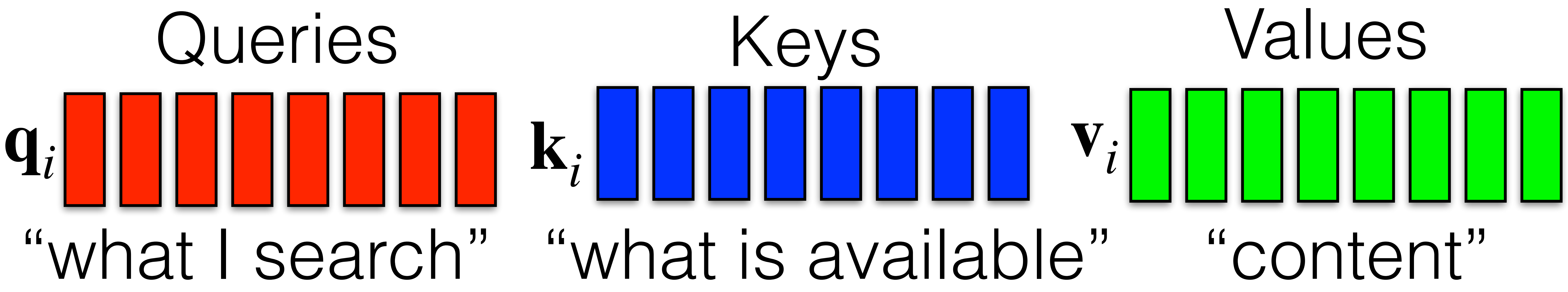


Word embedding

encoder

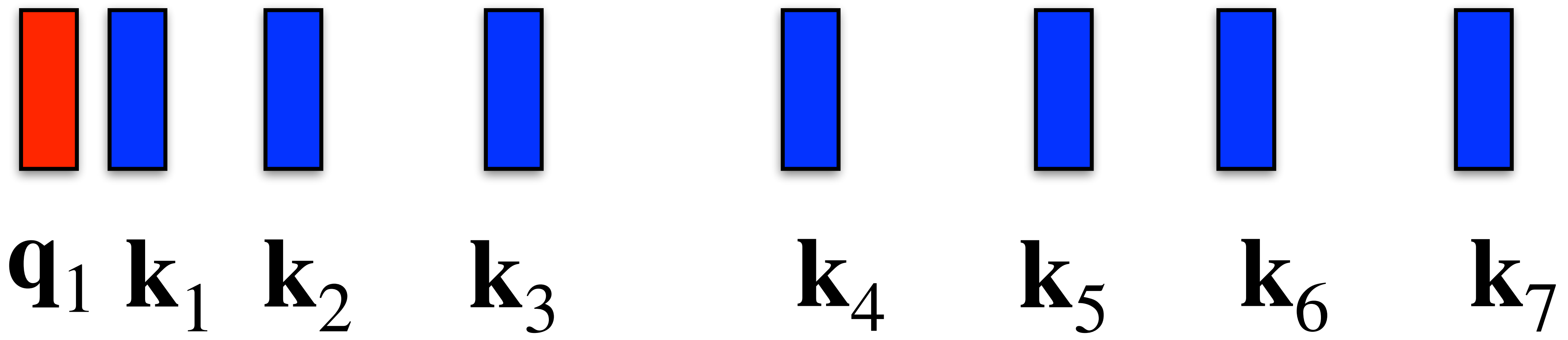

Queries
Keys
Values
$\mathbf{q}_i$
$\mathbf{k}_i$
$\mathbf{v}_i$
$\mathbf{W}_q\mathbf{x}_i$
$\mathbf{W}_k\mathbf{x}_i$
$\mathbf{W}_v\mathbf{x}_i$
$\mathbf{x}_i$
+
Positional encoding
Word embedding

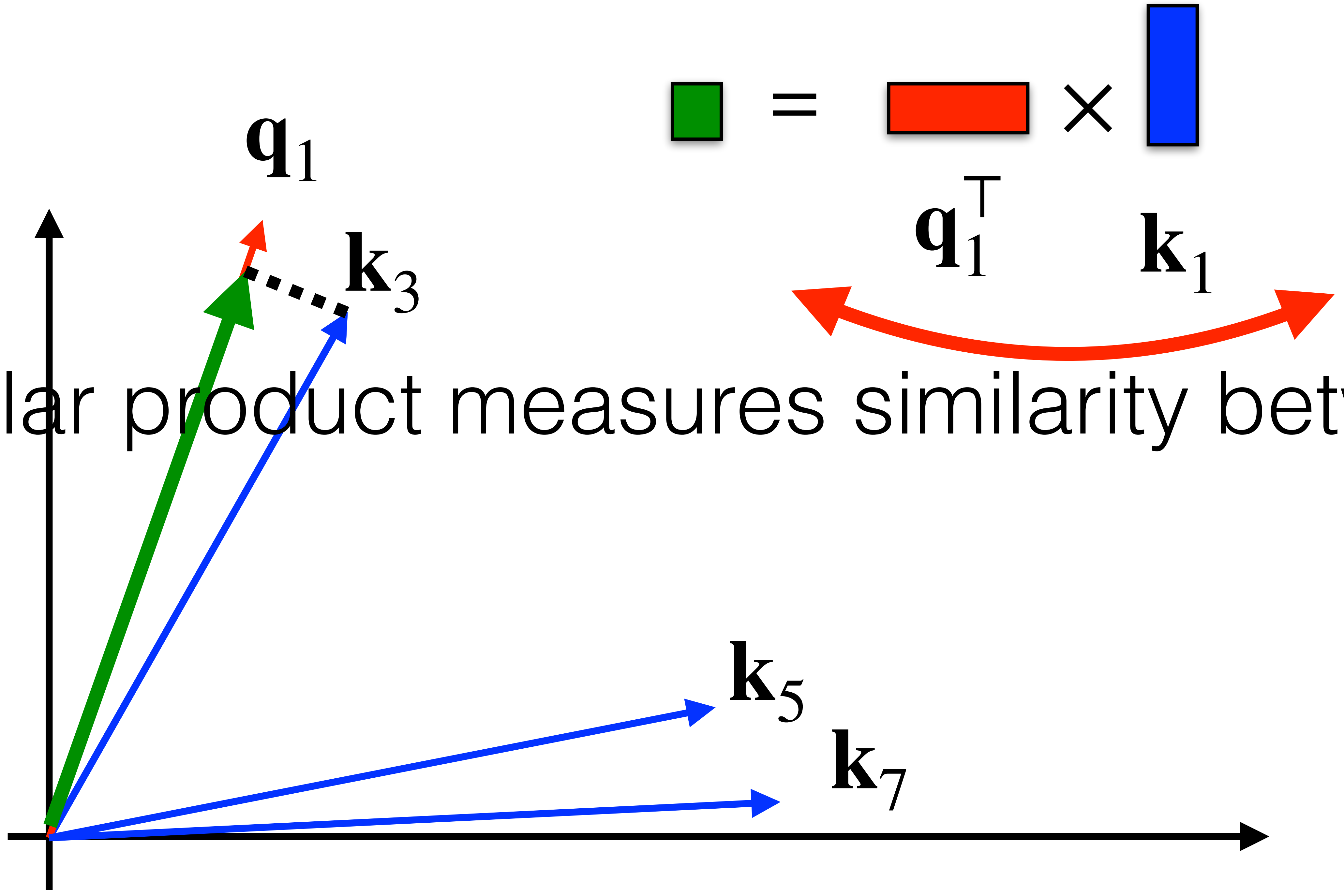
encoder



Karel is teacher and Mario is plumber.

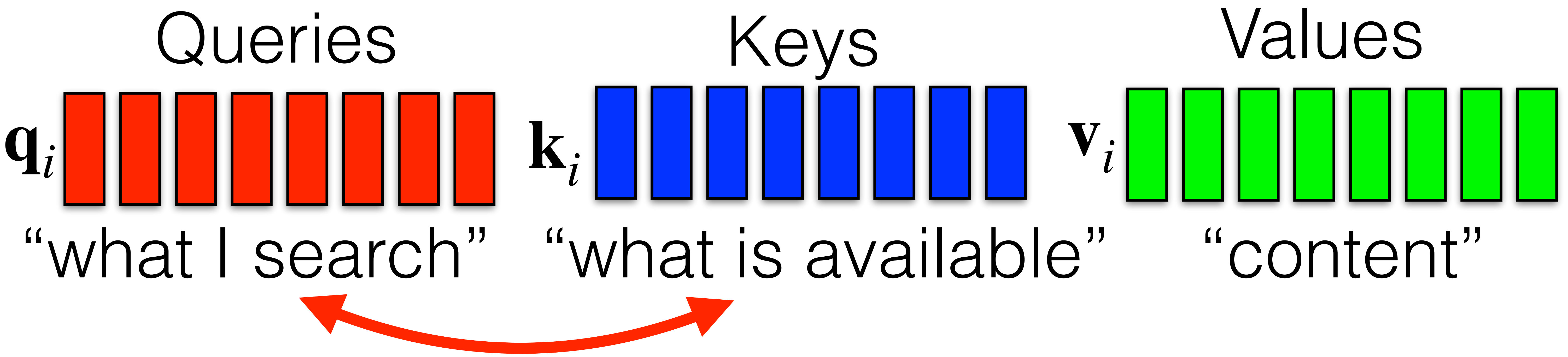
Which words contributes to meaning of Karel?





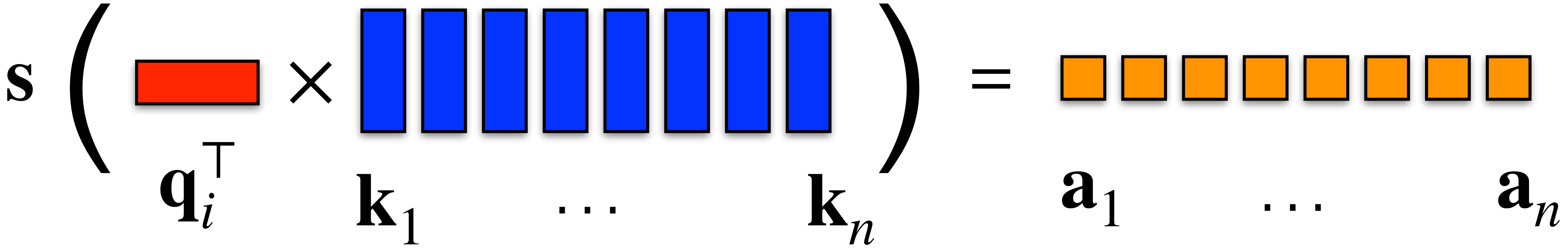
Scalar product measures similarity between vectors.

encoder

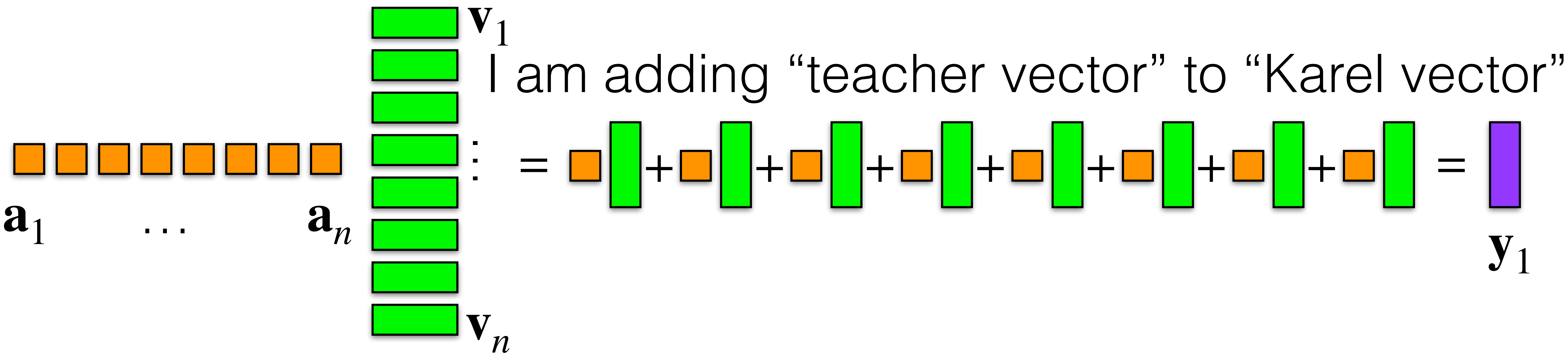


Scalar product measures similarity between vectors.

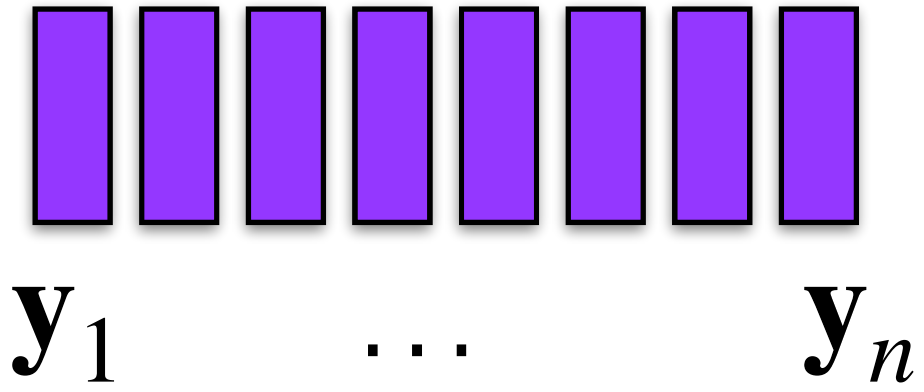
Get attention weights



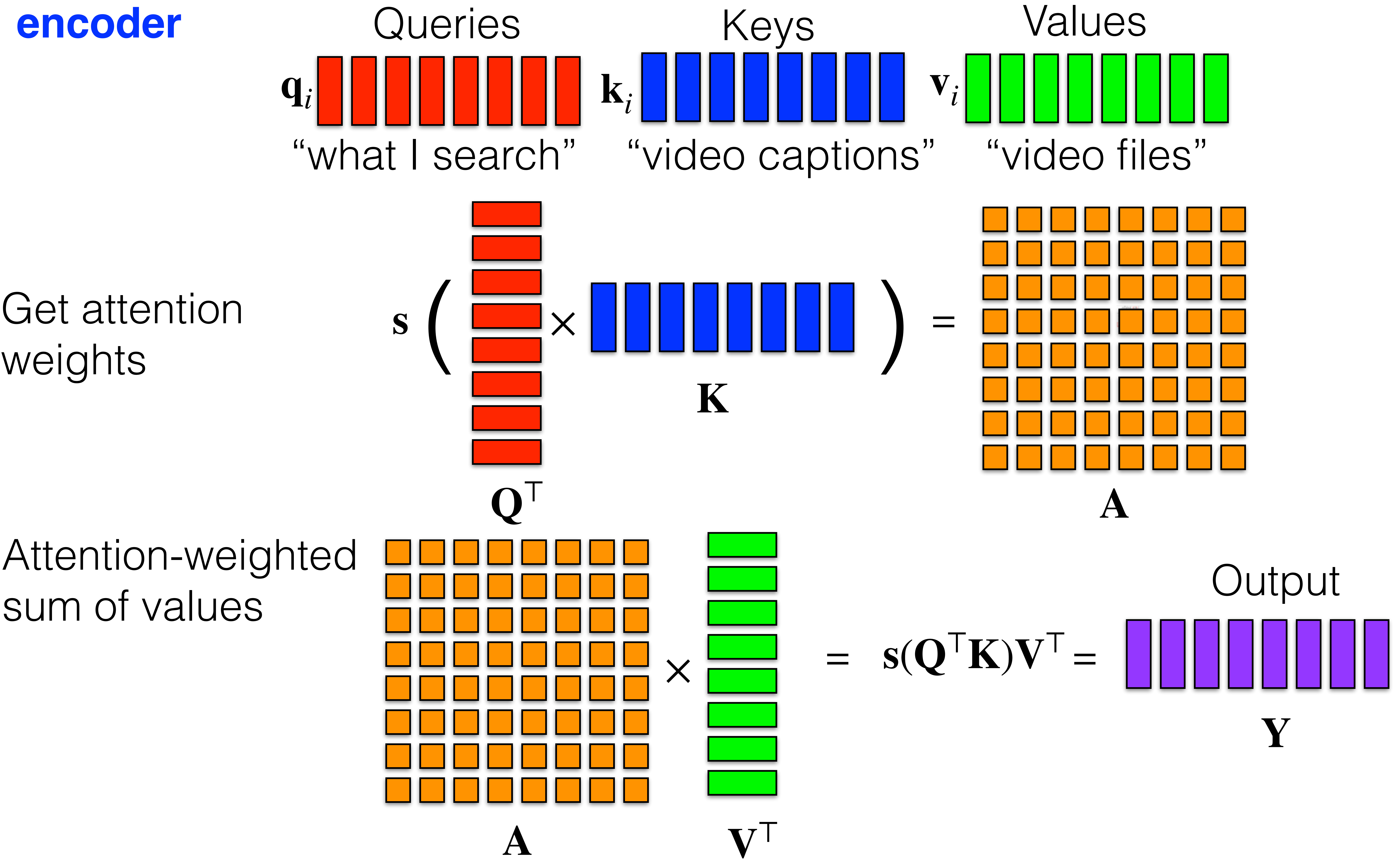
Attention-weighted sum of values



Outputs:

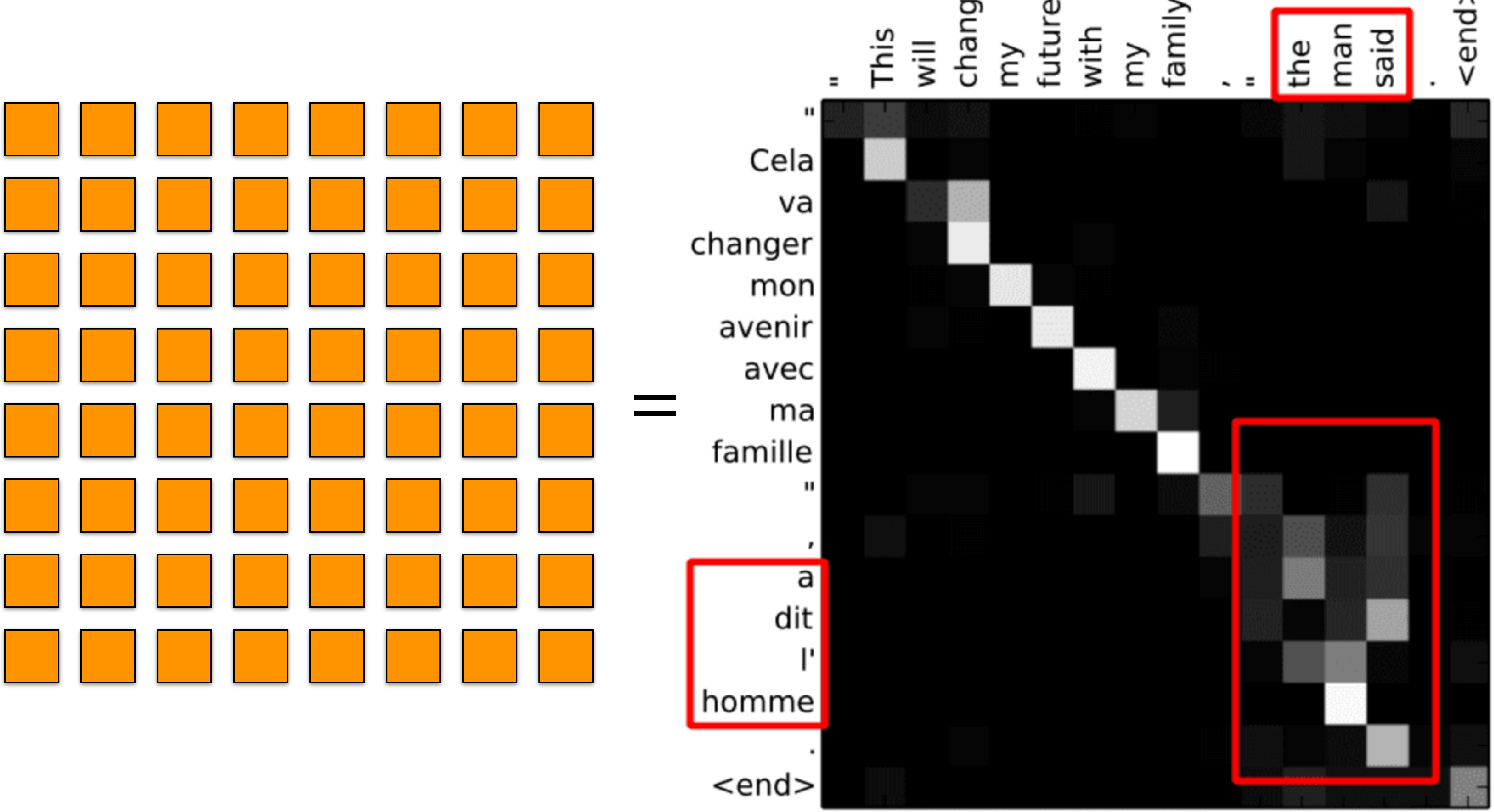


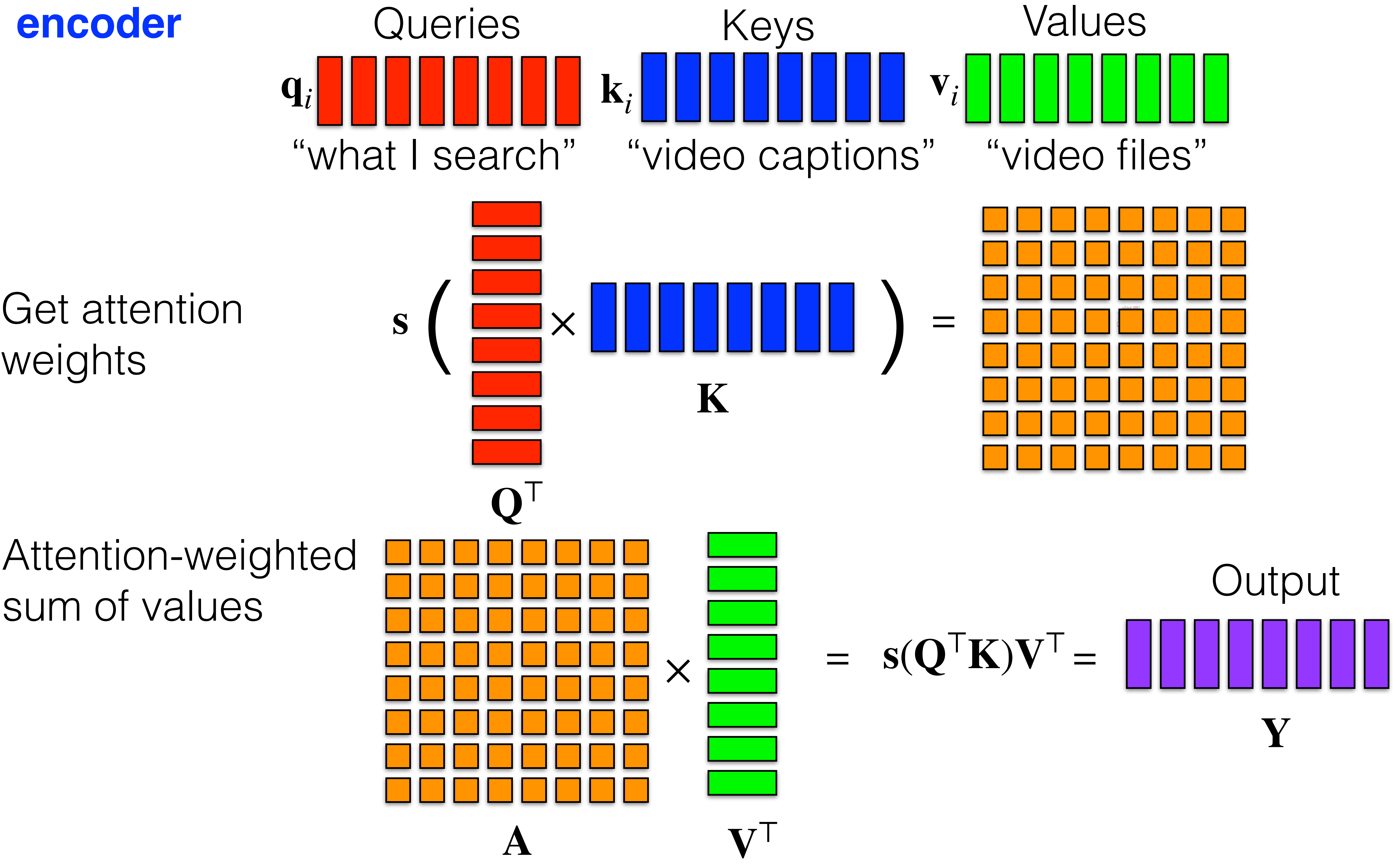
Avoid for-loop by smart matrix multiplication



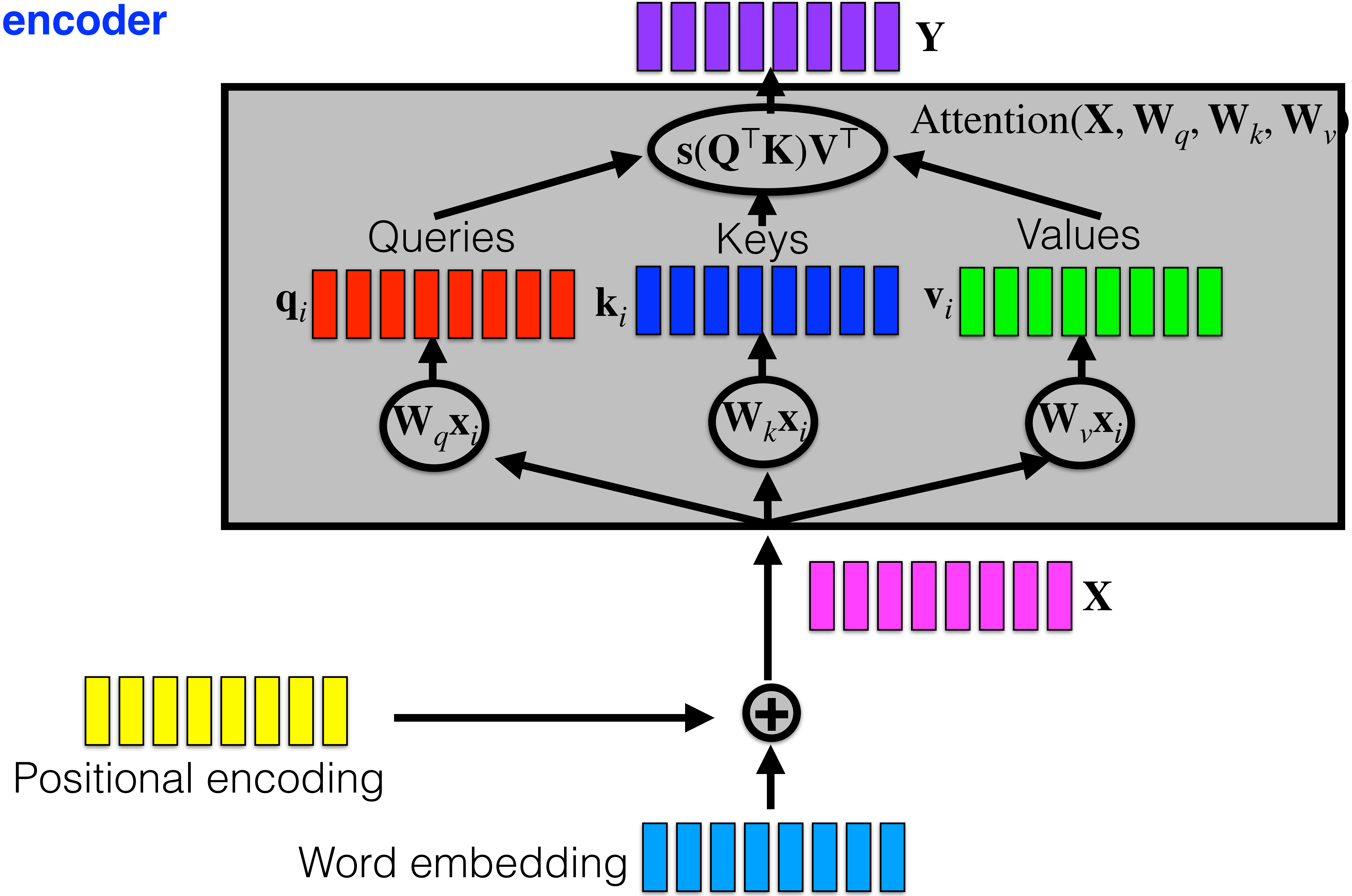
encoder
Queries
Keys
Values
$\mathbf{q}_i$
$\mathbf{k}_i$
$\mathbf{v}_i$
“what I search”
“video captions”
“video files”
Get attention weights
$\mathbf{s}\left(\mathbf{Q}^{\top} \times \mathbf{K}\right) = \mathbf{A}$
Attention-weighted sum of values
$\mathbf{A} \times \mathbf{V}^{\top} = \mathbf{s}(\mathbf{Q}^{\top}\mathbf{K})\mathbf{V}^{\top} = \mathbf{Y}$
Output

encoder

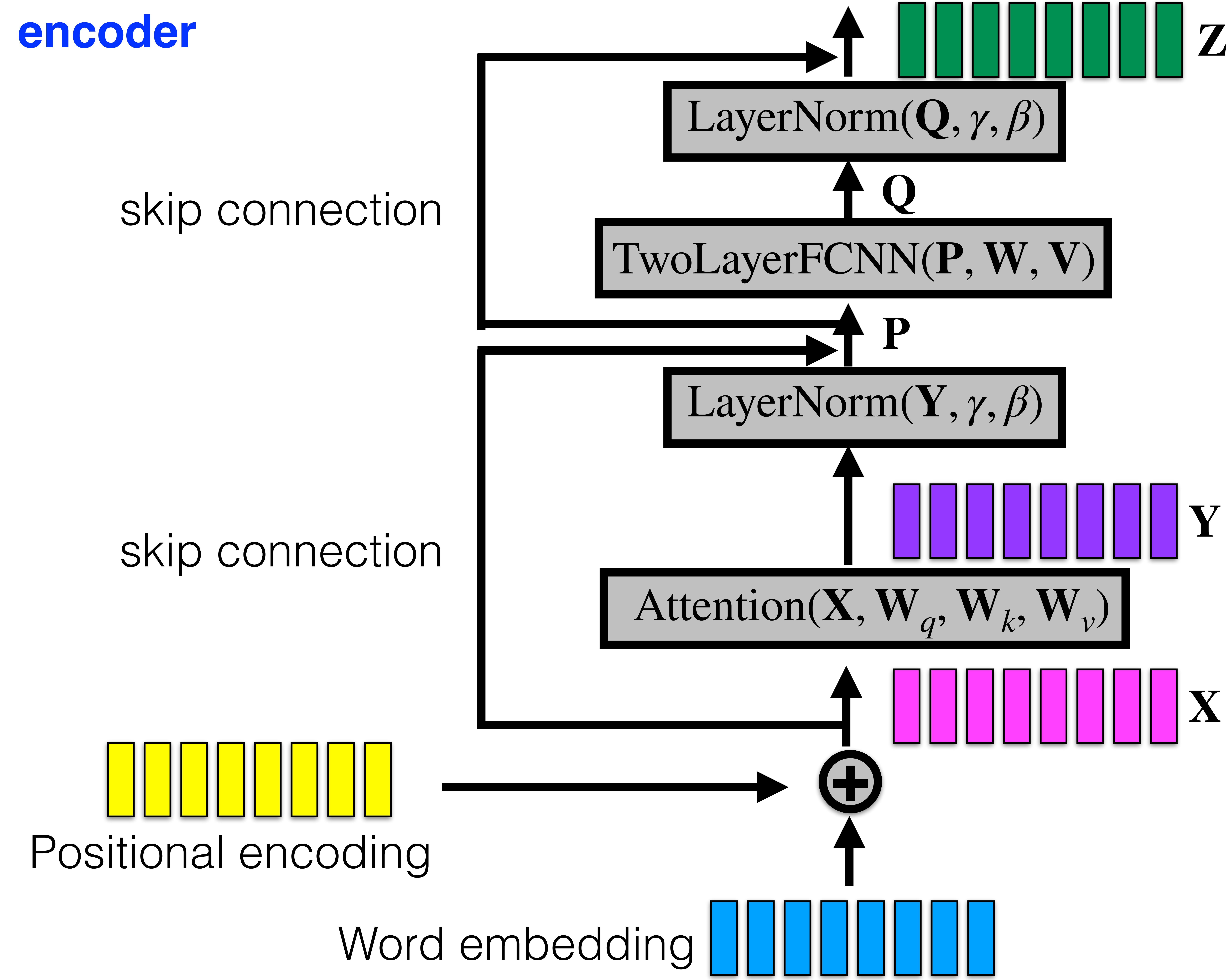

=
"
This
will
change
my
future
with
my
family
,
"
the
man
said
.
<end>
"
Cela
va
changer
mon
avenir
avec
ma
famille
"
,
a
dit
l'
homme
.
<end>

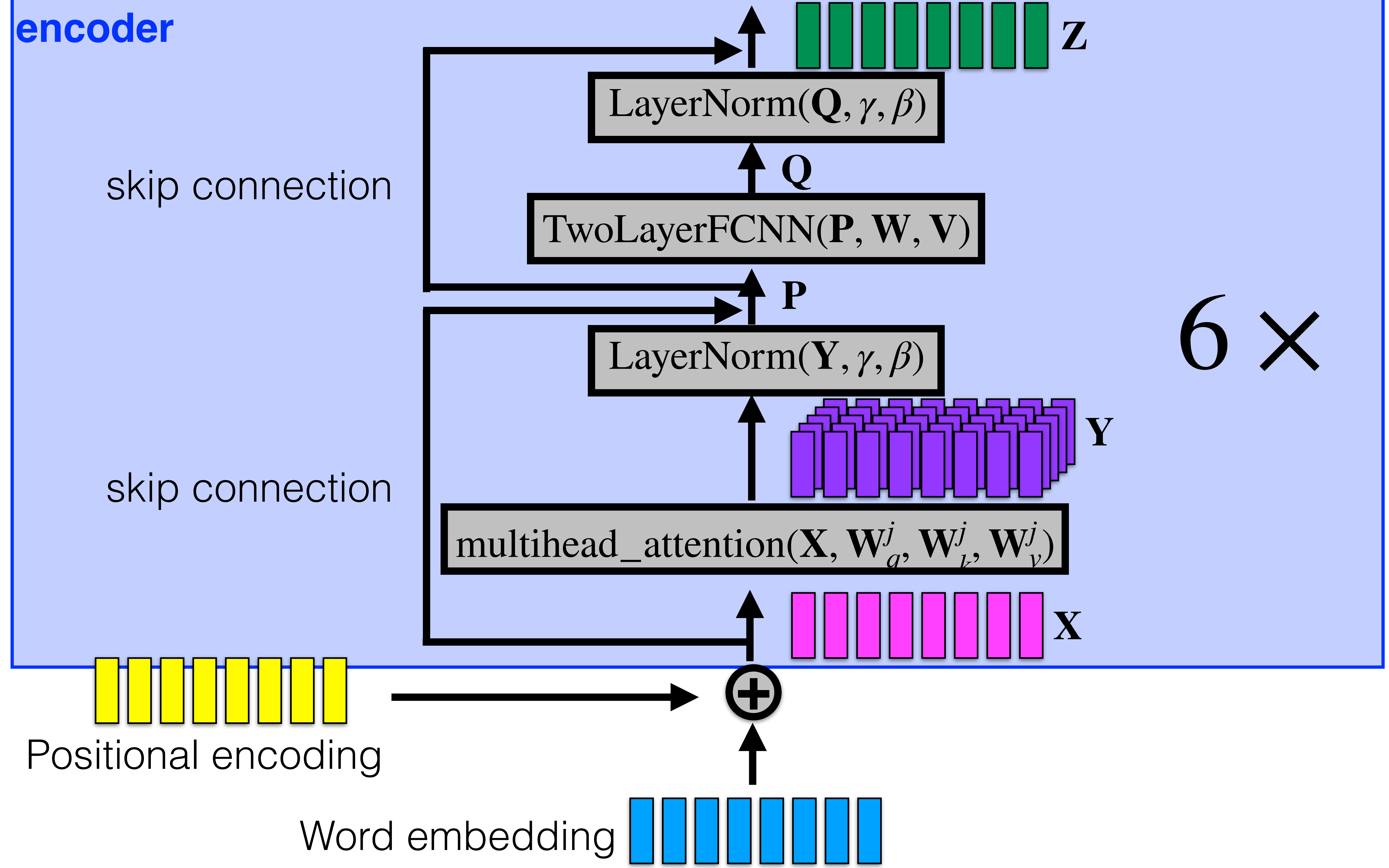

encoder
Queries
Keys
Values
$\mathbf{q}_i$
$\mathbf{k}_i$
$\mathbf{v}_i$
“what I search”
“video captions”
“video files”
Get attention weights
$\mathbf{s}\left(\mathbf{Q}^\top \times \mathbf{K}\right) =$
Attention-weighted sum of values
$\mathbf{A} \times \mathbf{V}^\top = \mathbf{s}(\mathbf{Q}^\top\mathbf{K})\mathbf{V}^\top =$
Output
$\mathbf{Y}$

encoder

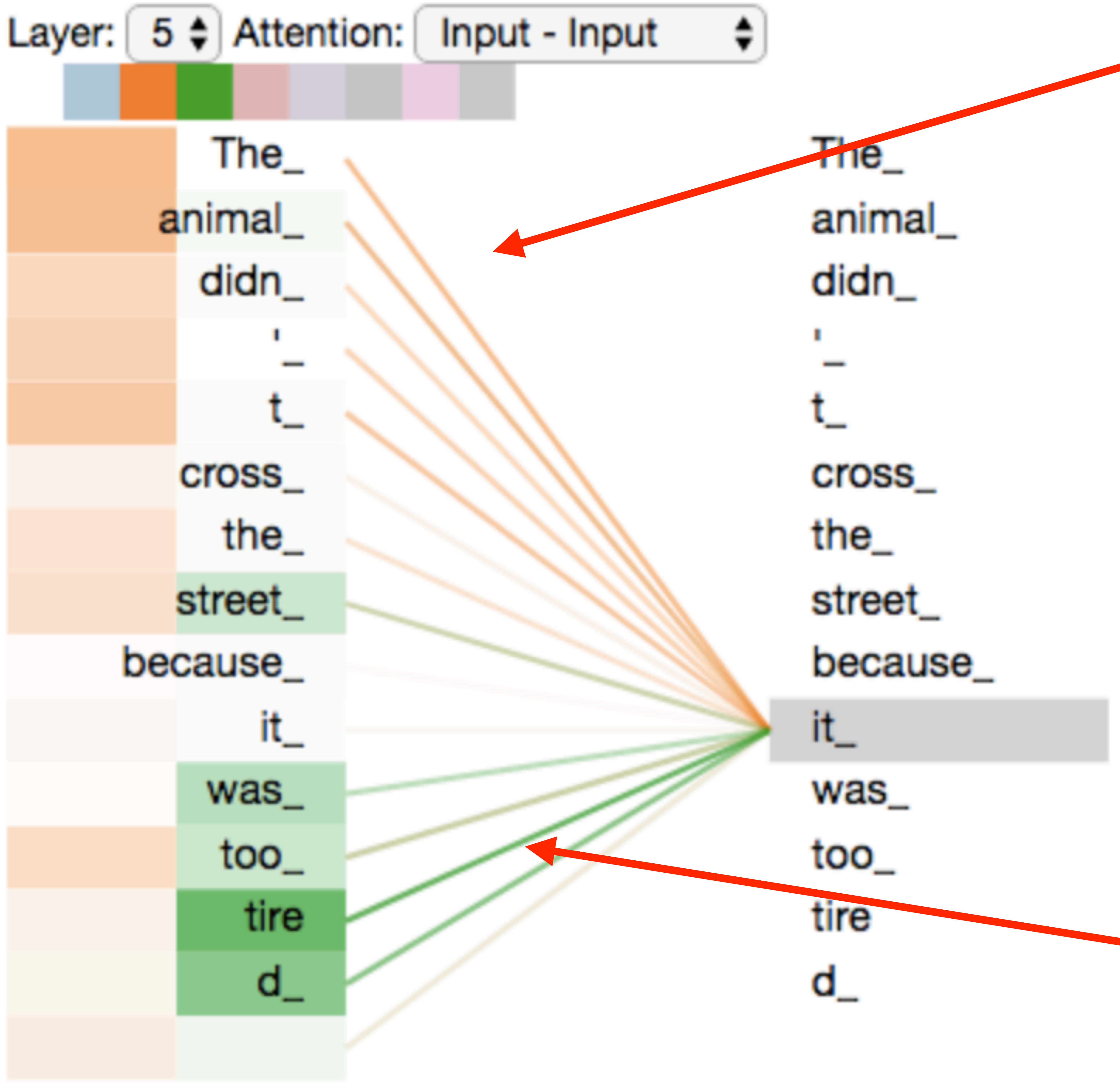


Y
$s(Q^\top K)V^\top$
Attention($\mathbf{X}, \mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$)
Queries
Keys
Values
$\mathbf{q}_i$
$\mathbf{k}_i$
$\mathbf{v}_i$
$\mathbf{W}_q\mathbf{x}_i$
$\mathbf{W}_k\mathbf{x}_i$
$\mathbf{W}_v\mathbf{x}_i$
X
Positional encoding
Word embedding


encoder
Z
LayerNorm(Q, γ, β)
Q
skip connection
TwoLayerFCNN(P, W, V)
P
LayerNorm(Y, γ, β)
Y
skip connection
Attention(X, W_q, W_k, W_v)
X
Positional encoding
Word embedding


encoder
Z
LayerNorm(Q, γ, β)
Q
skip connection
TwoLayerFCNN(P, W, V)
P
LayerNorm(Y, γ, β)
6 ×
Y
skip connection
multihead_attention(X, W^j_q, W^j_k, W^j_v)
X
Positional encoding
Word embedding

BertViz



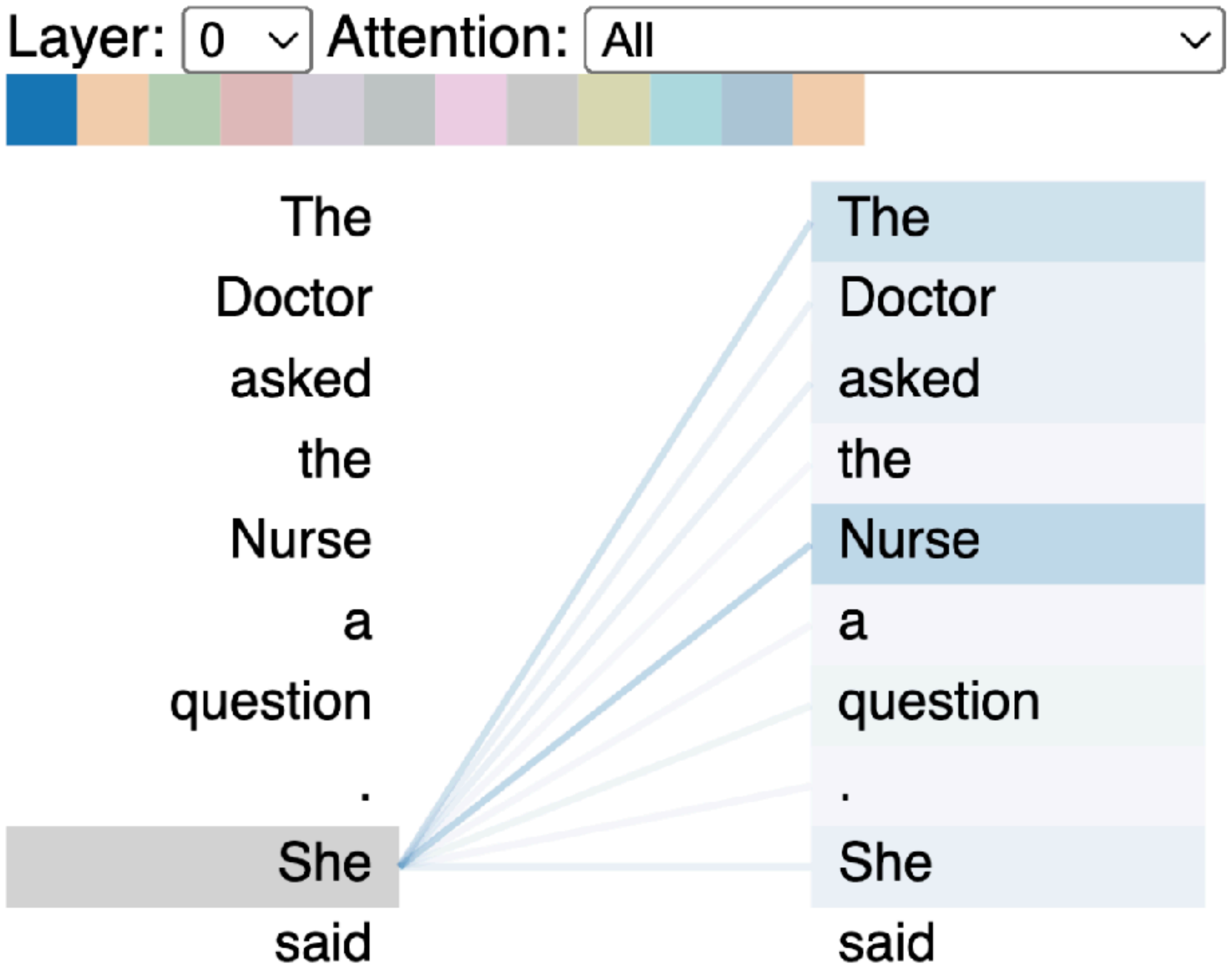
attention weights of **orange** attention head

"it=animal" vs "it=street"??

attention weights of **green** attention head

https://colab.research.google.com/github/tensorflow/tensor2tensor/blob/master/tensor2tensor/notebooks/hello_t2t.ipynb

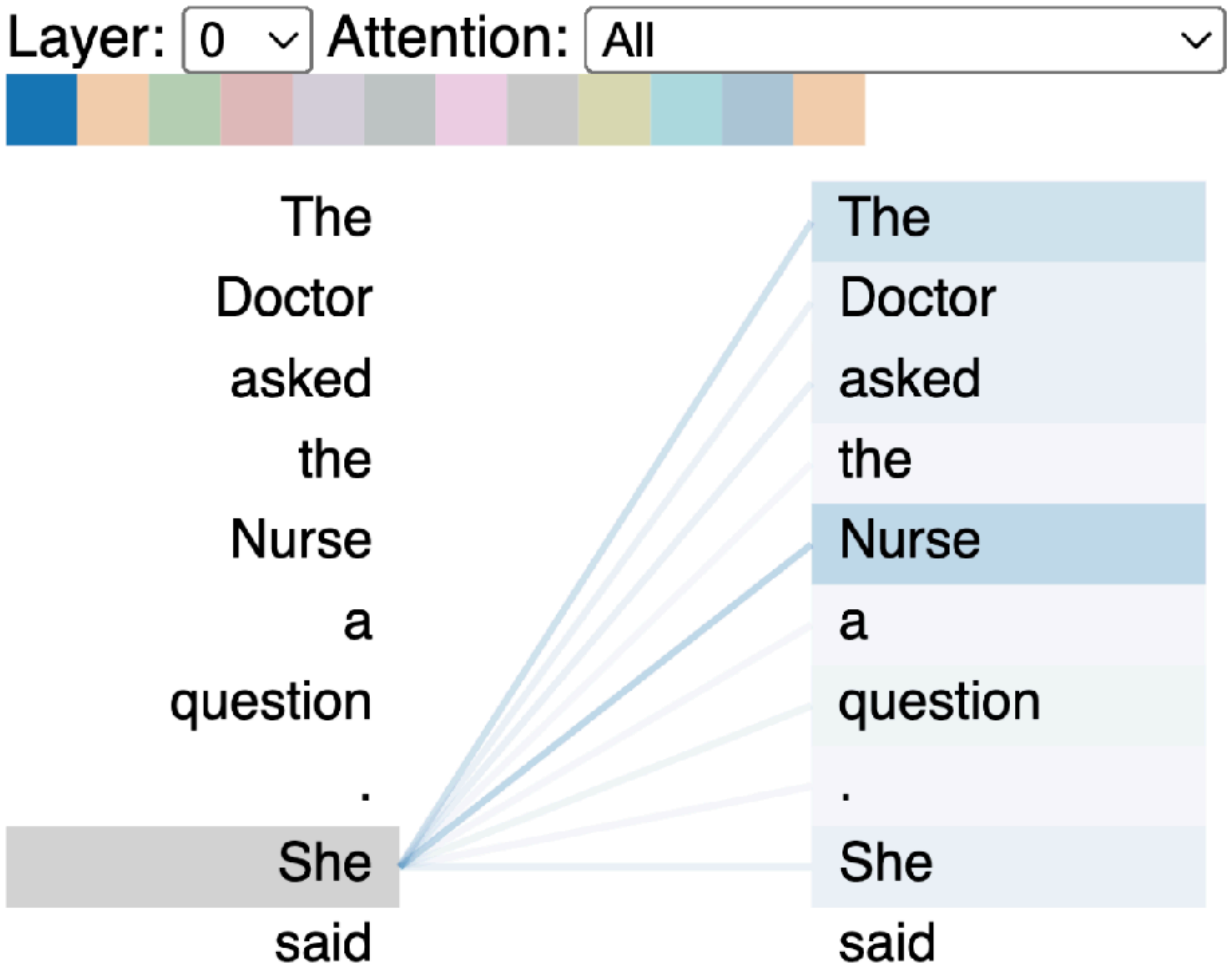
BertViz



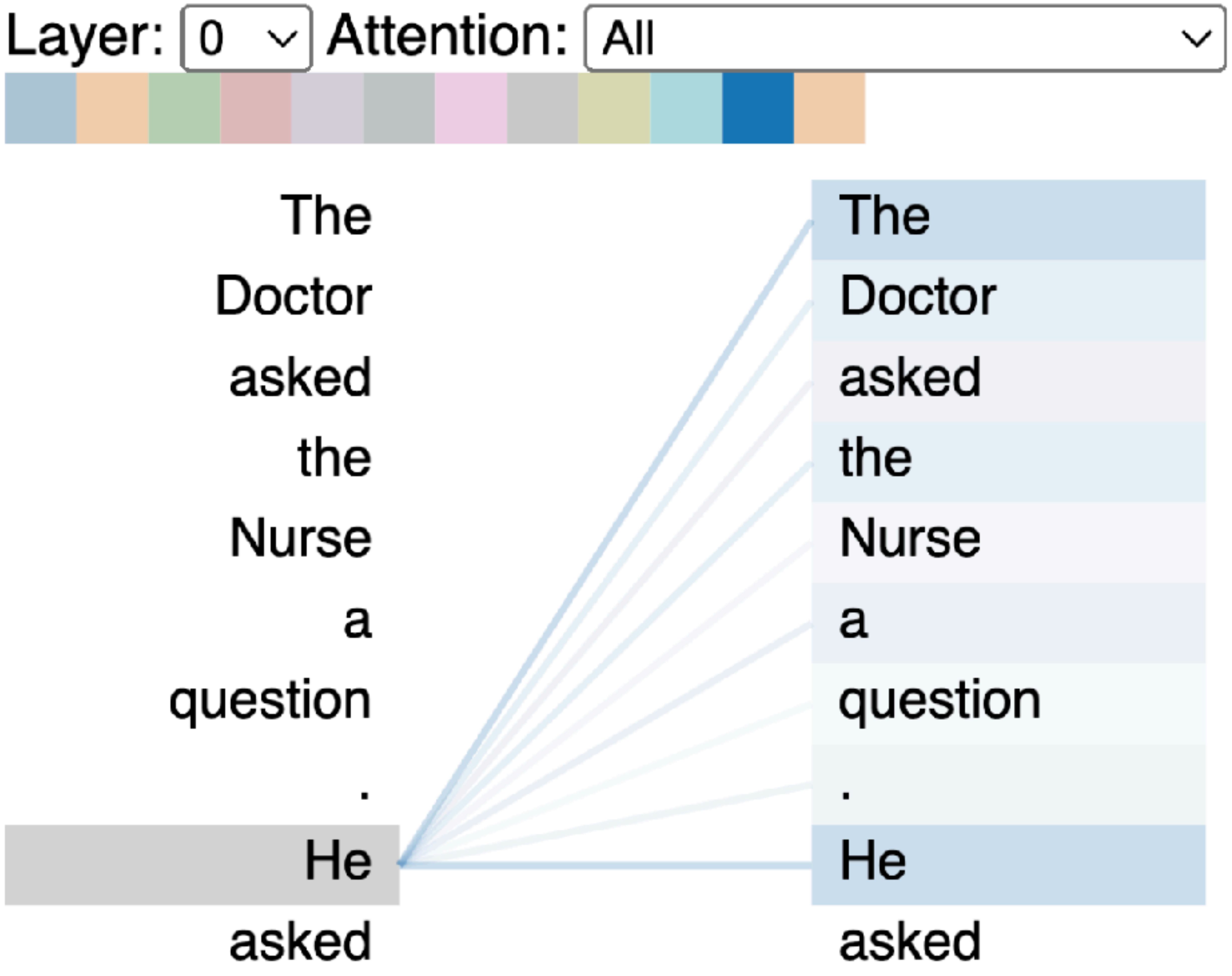
Model assumes “she=nurse”

https://www.comet.com/site/blog/explainable-ai-for-transformers/

BertViz (GPT2 model)



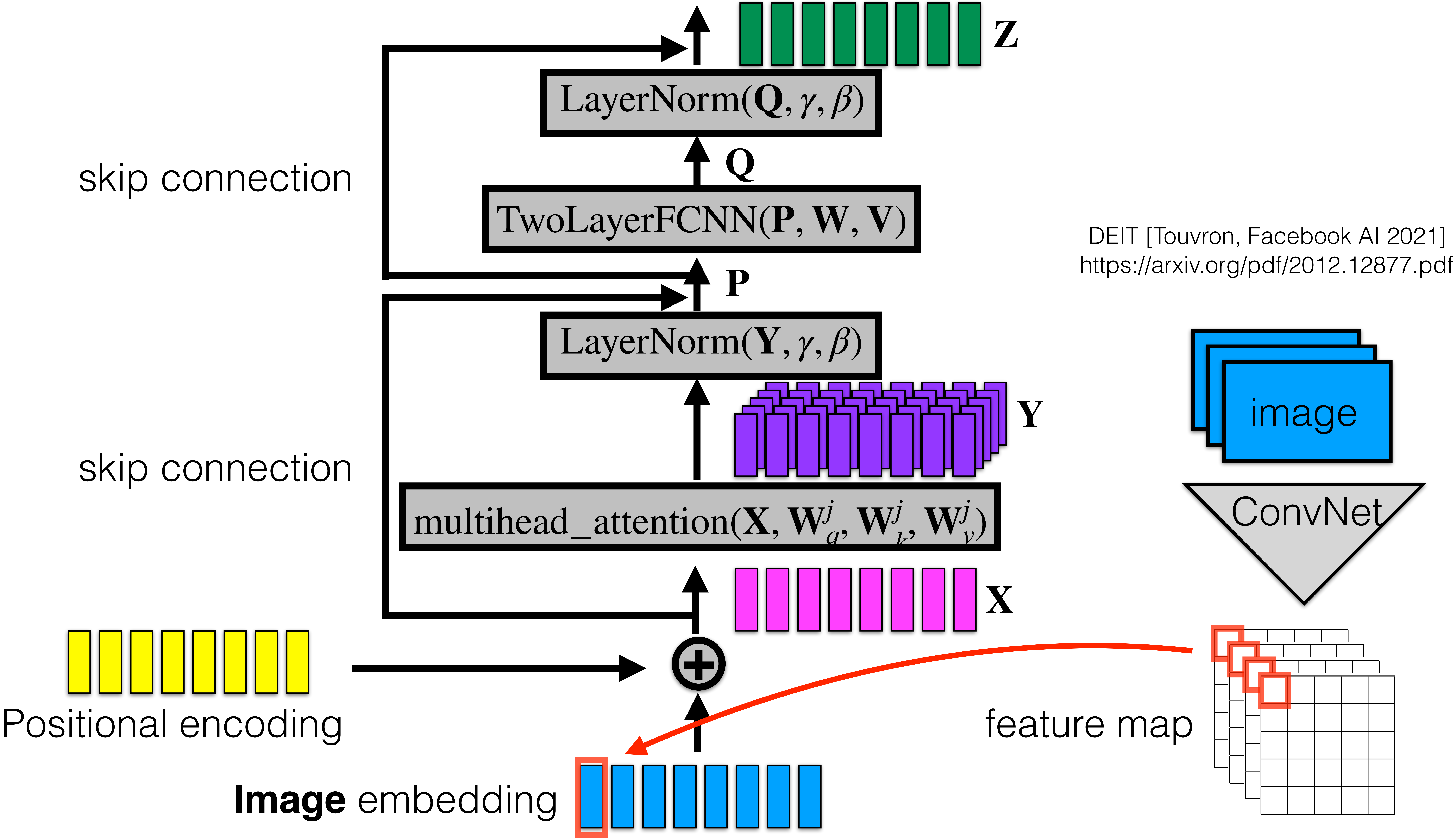
Model assumes "she=nurse"



Model assumes "he=doctor"

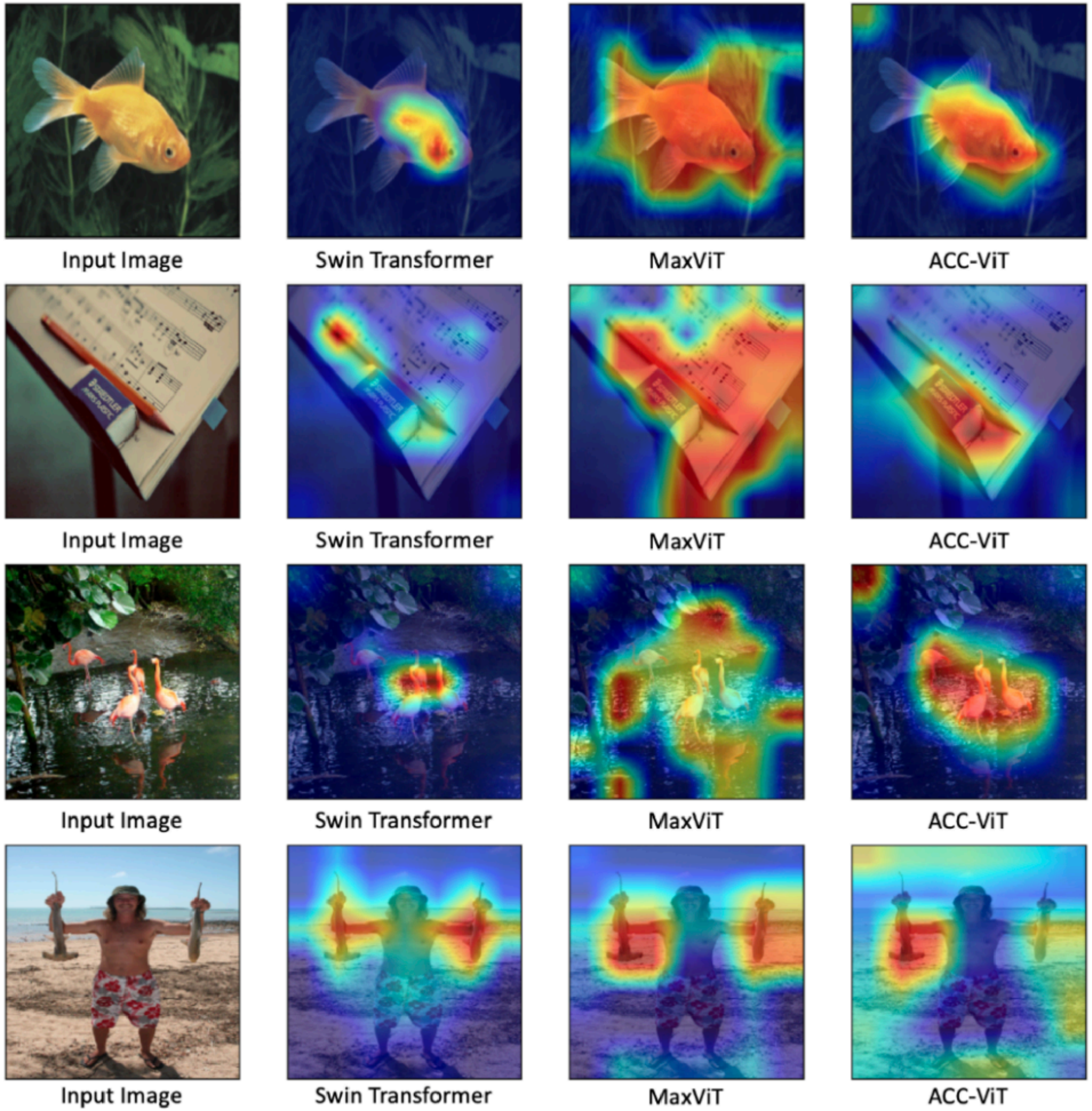
https://www.comet.com/site/blog/explainable-ai-for-transformers/

Transformers in images



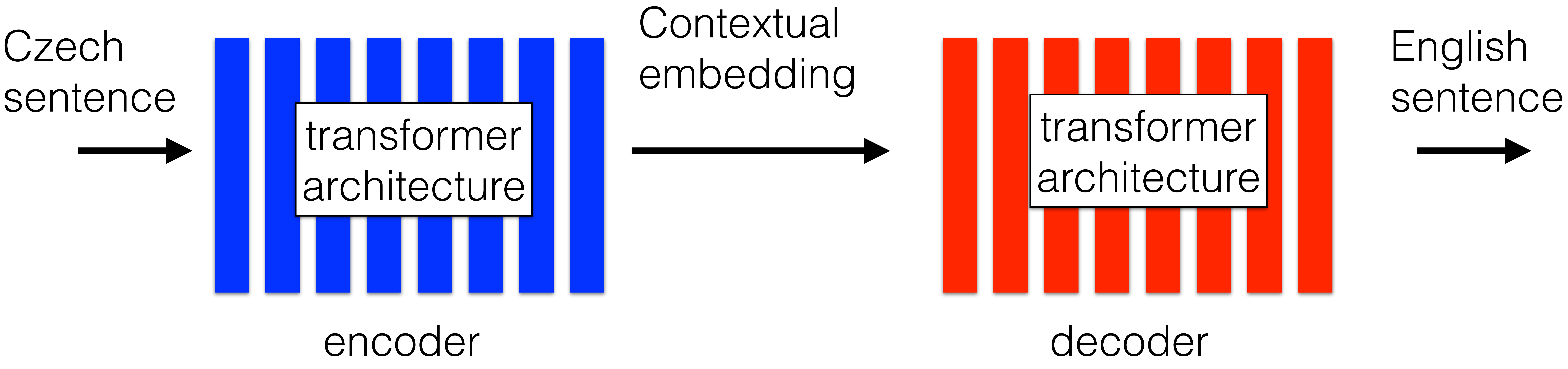
Z
LayerNorm(Q, γ, β)
Q
TwoLayerFCNN(P, W, V)
P
LayerNorm(Y, γ, β)
Y
multihead_attention(X, W^j_q, W^j_k, W^j_v)
X
skip connection
skip connection
Positional encoding
Image embedding
DEIT [Touvron, Facebook AI 2021]
https://arxiv.org/pdf/2012.12877.pdf
image
ConvNet
feature map

Attention in different Vision transformers (visu by gradCAM)



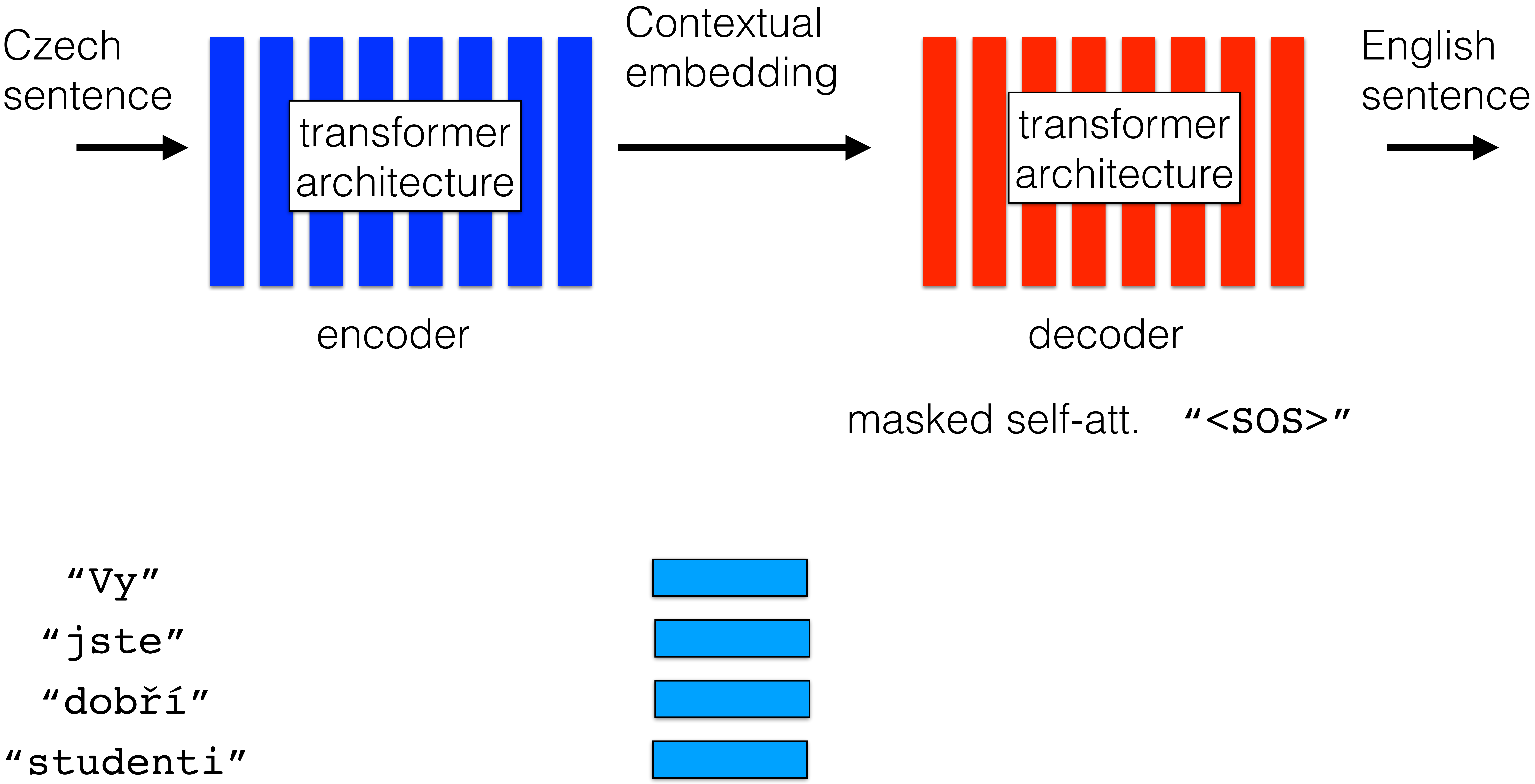
https://arxiv.org/pdf/2406.08859v1

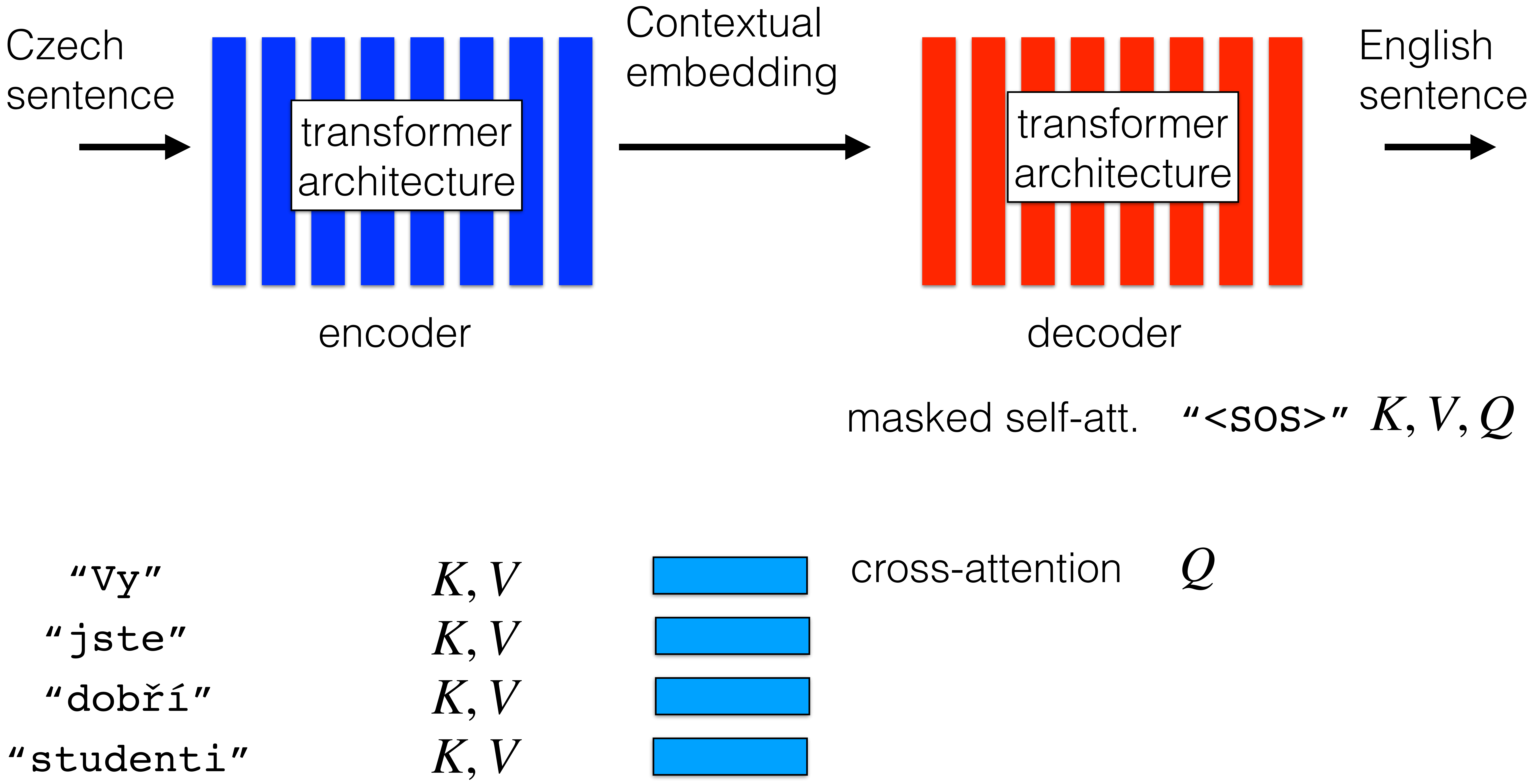
Machine translation

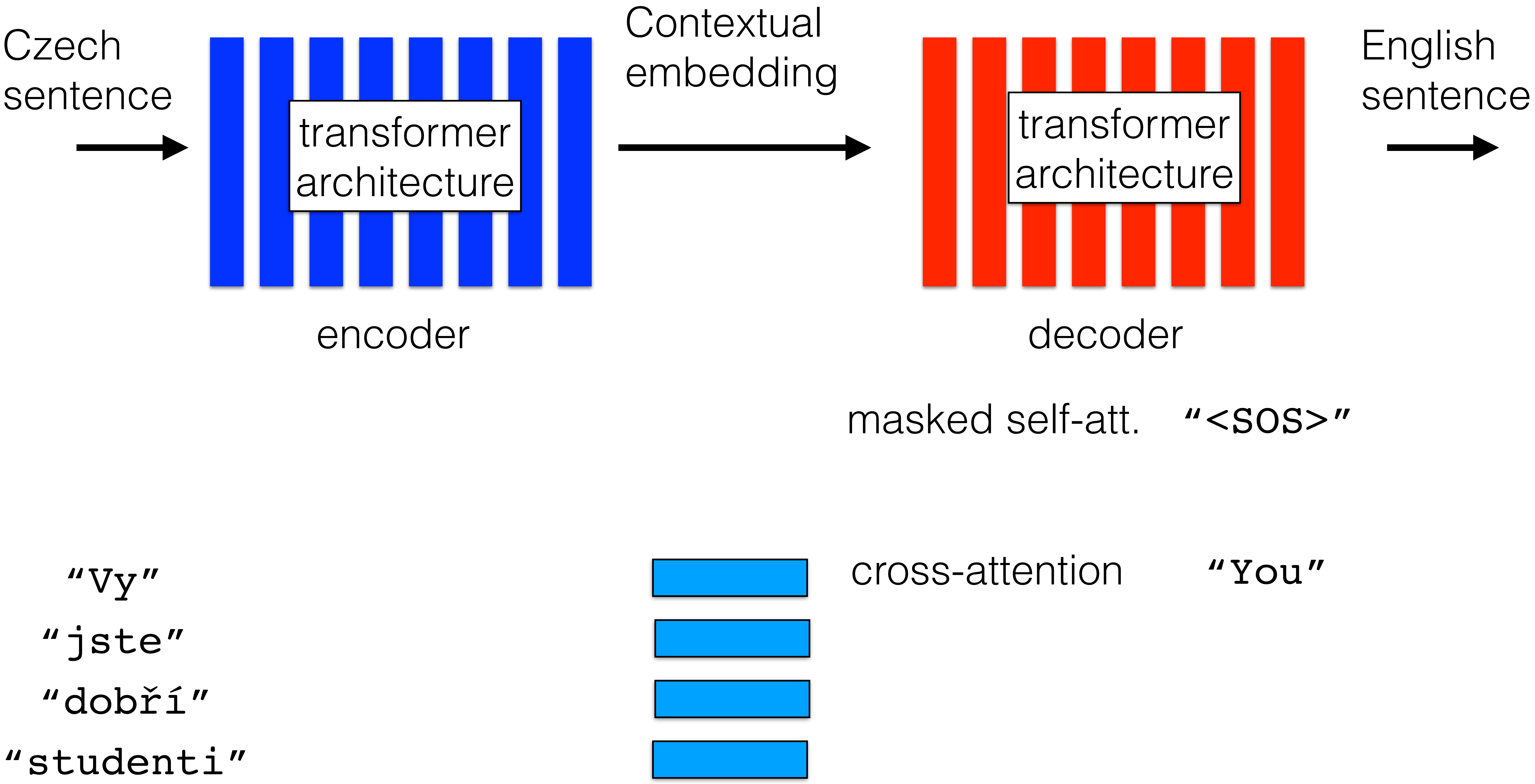


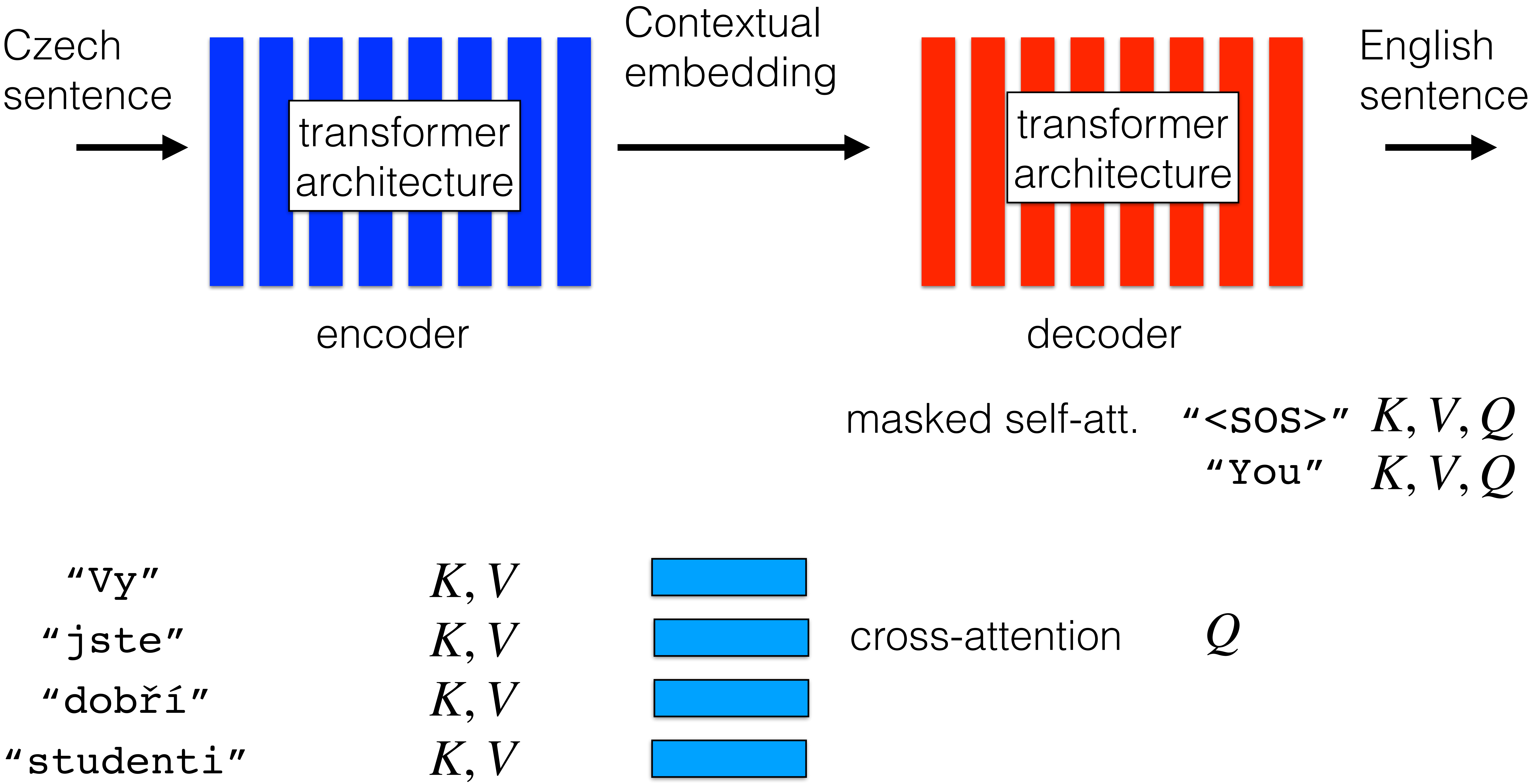
“Vy”
“jste”
“dobří”
“studenti”

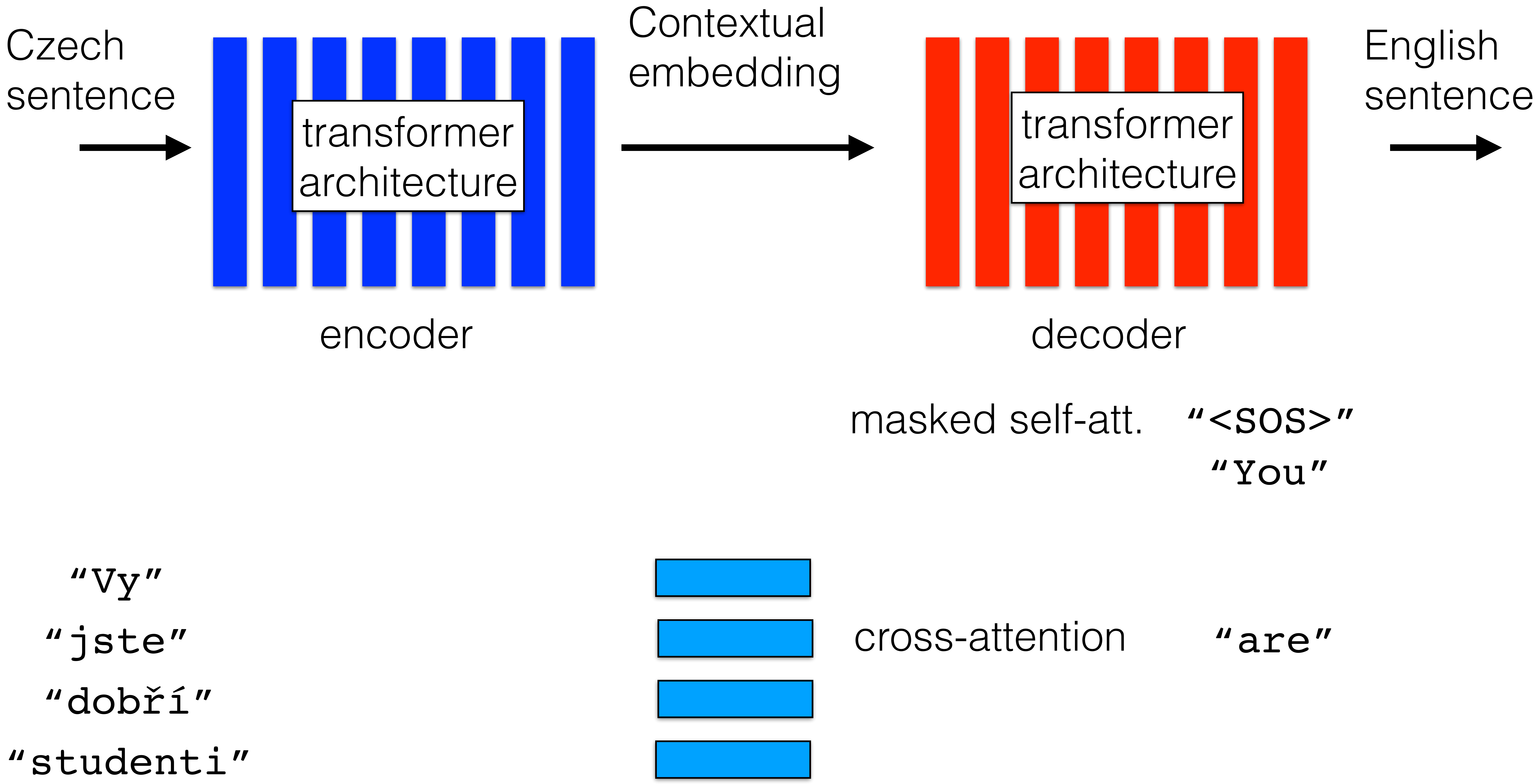
- Encoder is standard transformer with self-attention
- Decoder auto-regressively generates output sentence
- Decoder requires special attentions

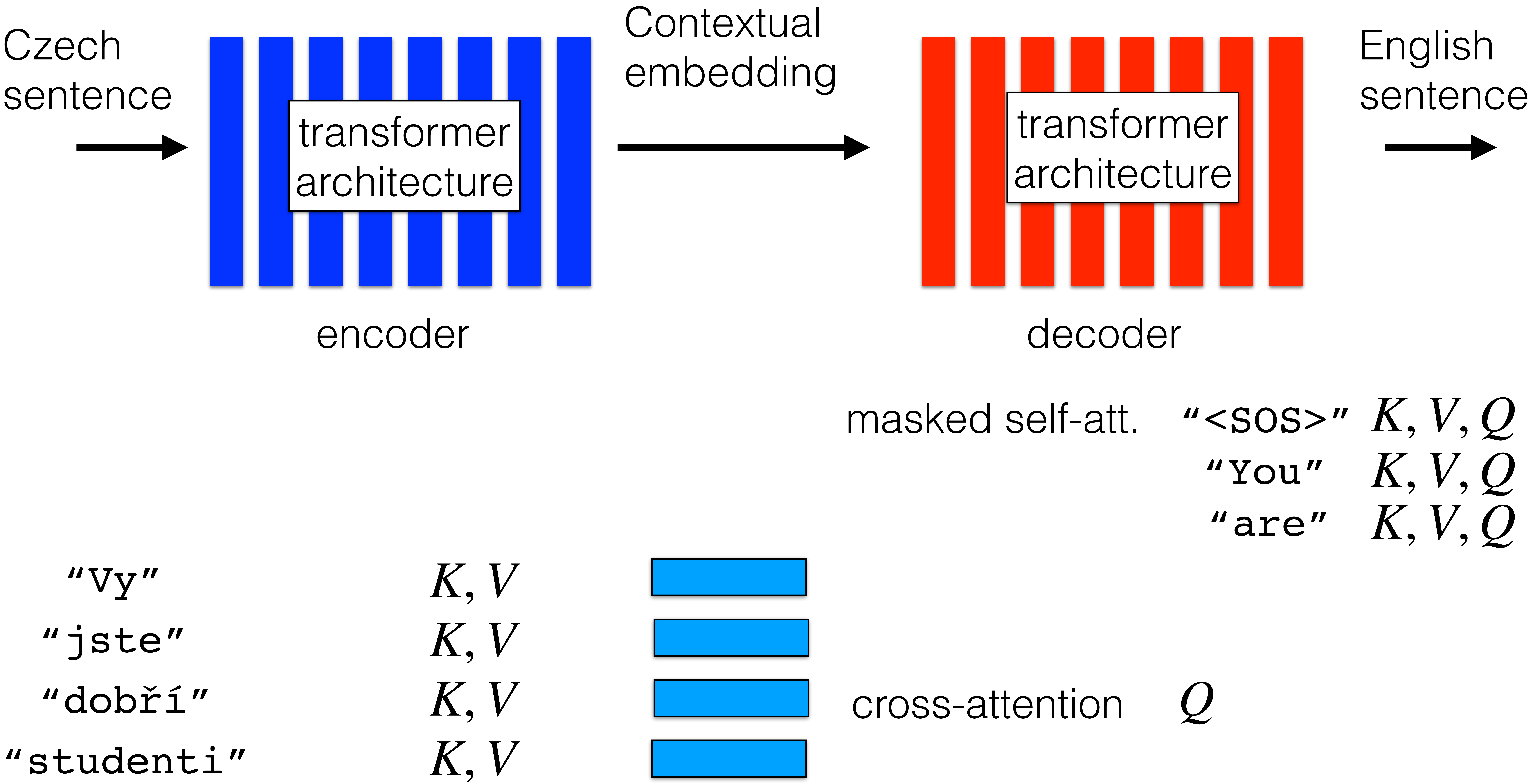

Czech
sentence
transformer
architecture
encoder
Contextual
embedding
transformer
architecture
decoder
English
sentence
masked self-att.
“<SOS>”
“Vy”
“jste”
“dobří”
“studenti”

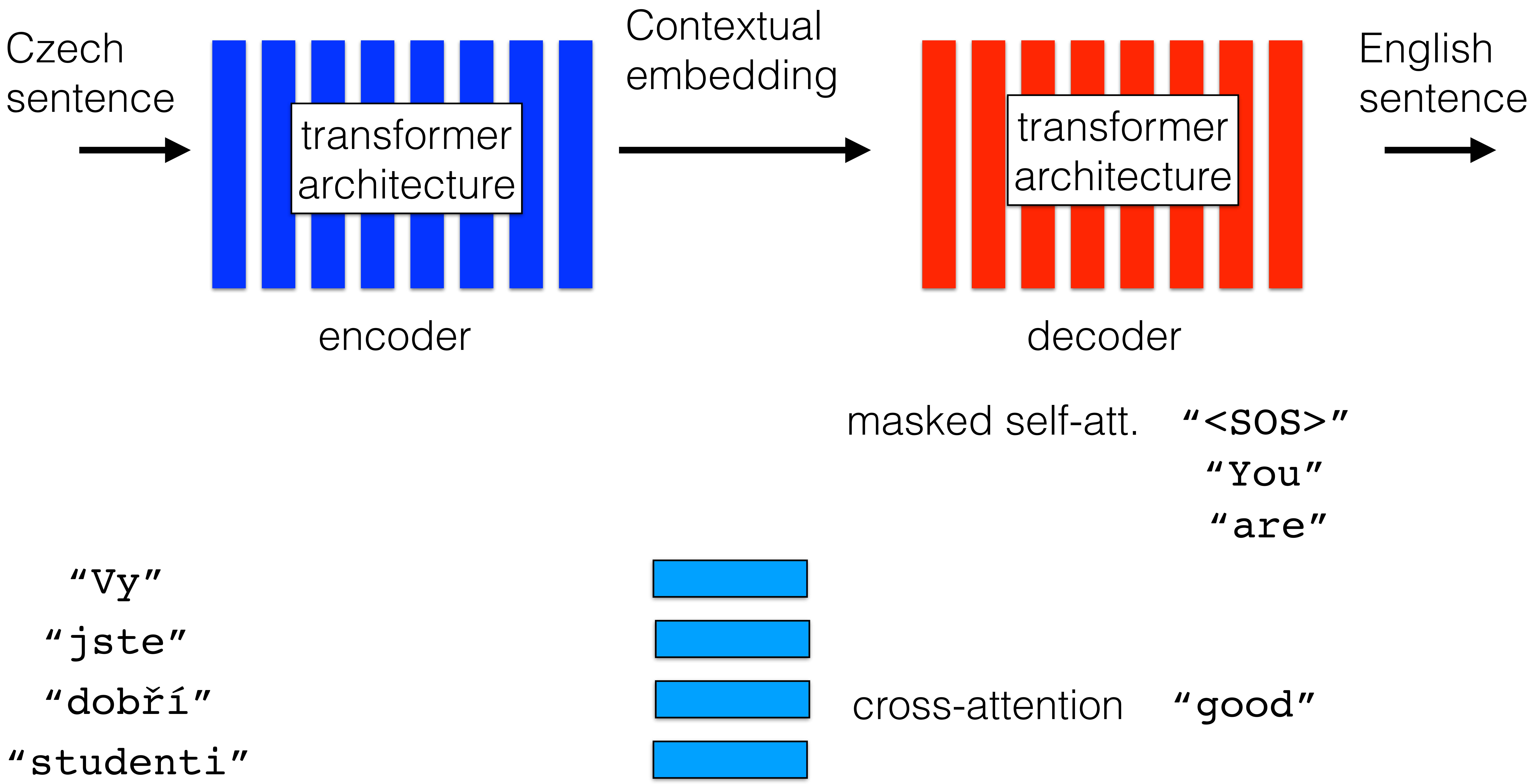

Czech sentence
transformer architecture
encoder
Contextual embedding
transformer architecture
decoder
English sentence
masked self-att. “<sos>” K, V, Q
“Vy” K, V
“jste” K, V
“dobří” K, V
“studenti” K, V
cross-attention Q


Czech sentence
transformer architecture
encoder
Contextual embedding
transformer architecture
decoder
English sentence
masked self-att.
“<SOS>”
“Vy”
“jste”
“dobří”
“studenti”
cross-attention
“You”

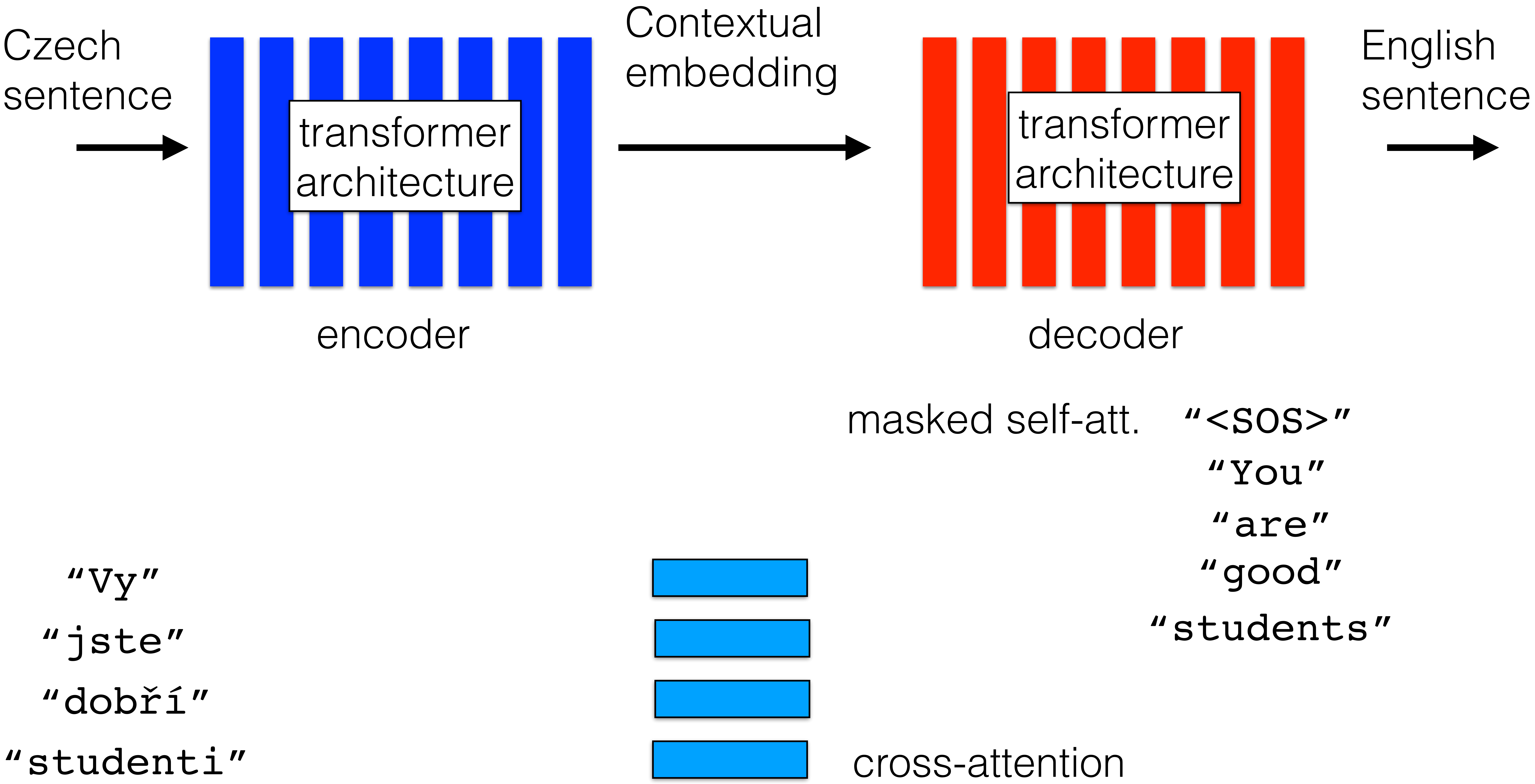

Czech sentence
transformer architecture
encoder
Contextual embedding
transformer architecture
decoder
English sentence
masked self-att.
"<sos>" K, V, Q
"You" K, V, Q
"Vy" K, V
"jste" K, V
"dobří" K, V
"studenti" K, V
cross-attention
Q


Czech sentence
transformer architecture
encoder
Contextual embedding
transformer architecture
decoder
English sentence
masked self-att.
“<SOS>”
“You”
“Vy”
“jste”
“dobří”
“studenti”
cross-attention
“are”


Czech sentence
transformer architecture
encoder
Contextual embedding
transformer architecture
decoder
English sentence
masked self-att.
"<sos>" K, V, Q
"You" K, V, Q
"are" K, V, Q
"Vy" K, V
"jste" K, V
"dobří" K, V
"studenti" K, V
cross-attention Q



Czech
sentence
transformer
architecture
encoder
Contextual
embedding
transformer
architecture
decoder
English
sentence
masked self-att.
“<SOS>”
“You”
“are”
“Vy”
“jste”
“dobří”
“studenti”
cross-attention
“good”


Czech
sentence
transformer
architecture
encoder
Contextual
embedding
transformer
architecture
decoder
English
sentence
masked self-att.
“<SOS>”
“You”
“are”
“good”
“students”
“Vy”
“jste”
“dobří”
“studenti”
cross-attention

Self-attention

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

Fill in **gaps** (<unknown> words tokens) in sentences (BERT)

Masked self-attention

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}+\mathbf{mask}\right)V$$

$$\mathrm{Mask}=\begin{bmatrix}0 & -\infty & -\infty\\ 0 & 0 & -\infty\\ 0 & 0 & 0\end{bmatrix}$$

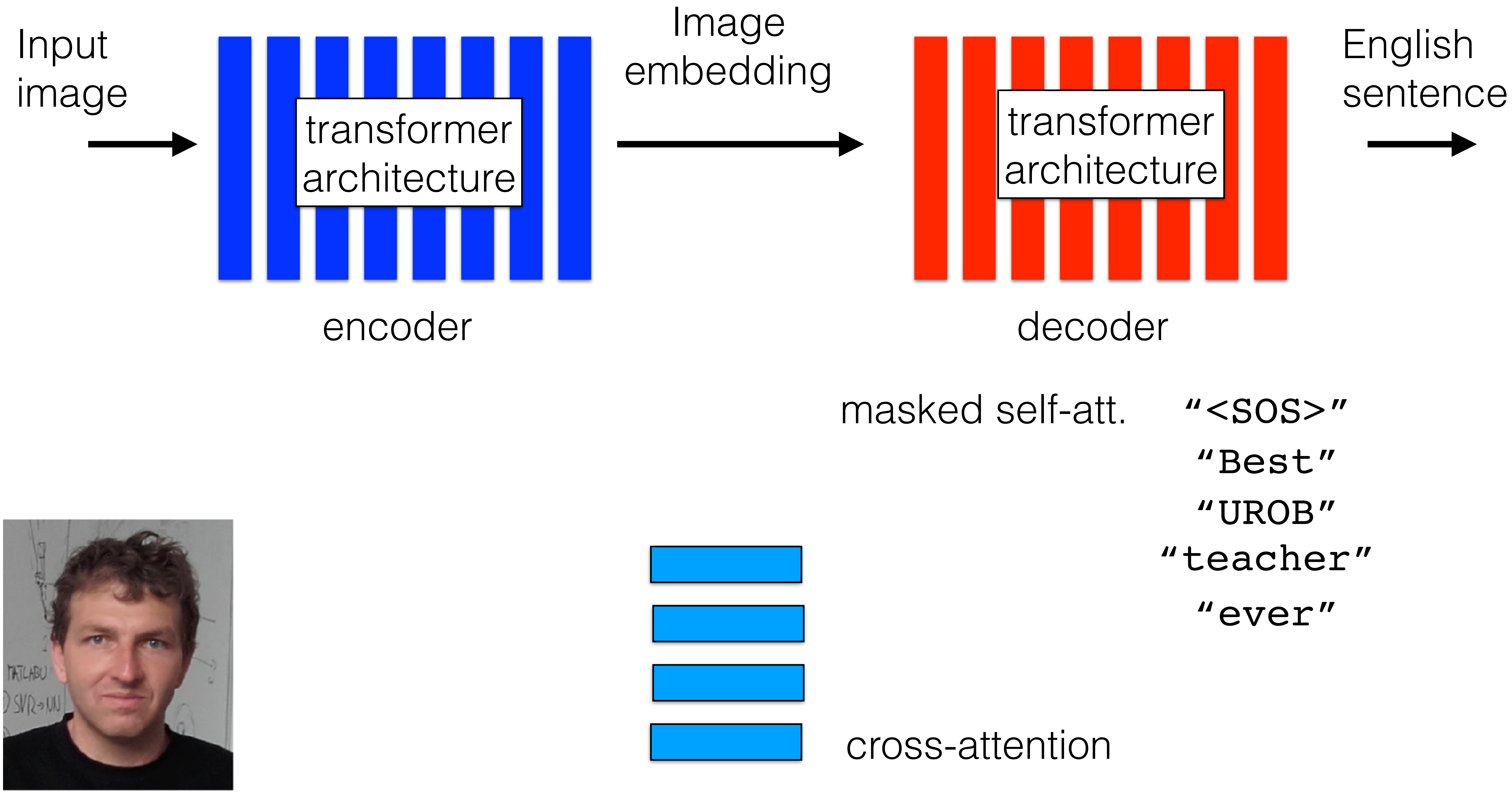
- Assures temporal coherence without creating dependence on the correct number of <unknown> words in the input.
- Assures better **paralelization** and **generalization** in autoregressive text generation (GPT)

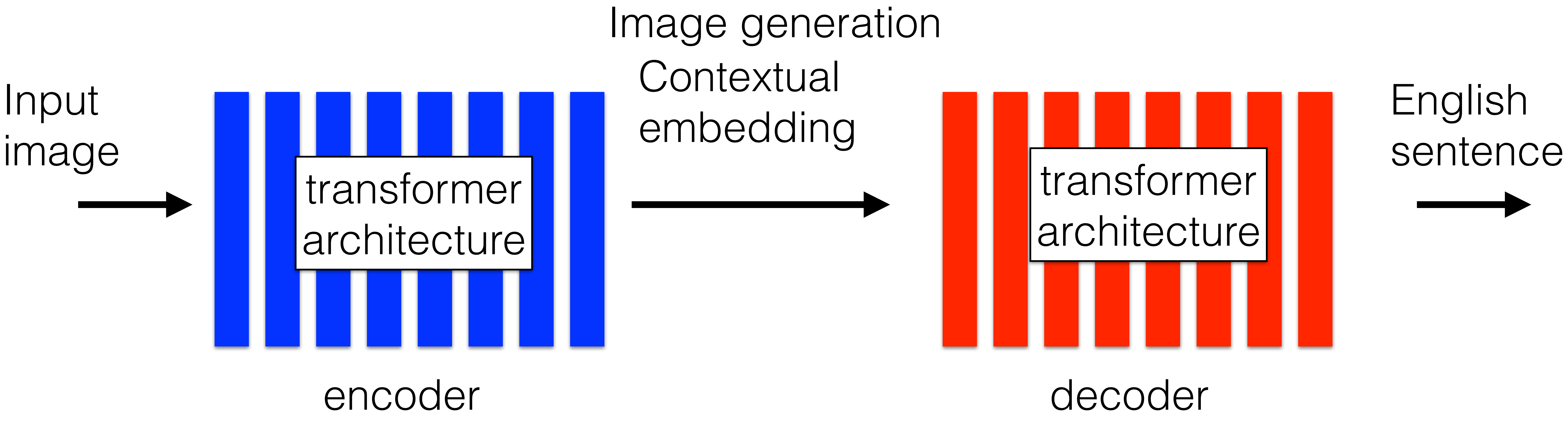
$$\mathrm{Cross\text{-}Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Q … from decoder

K, V … from encoder

Image captioning

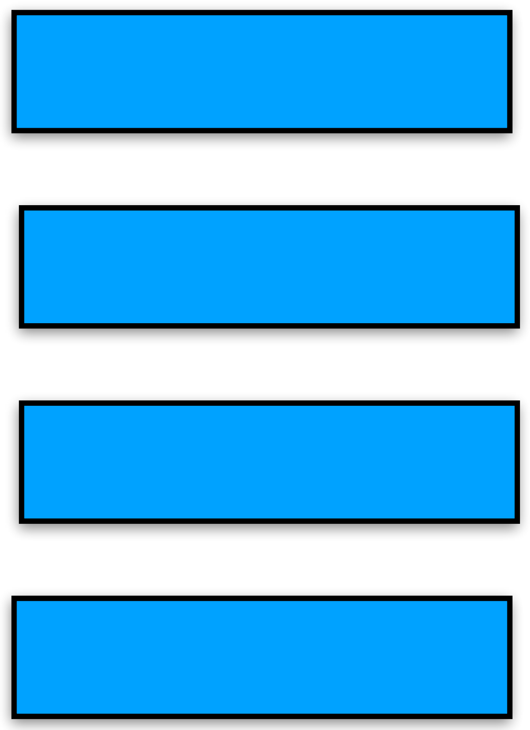


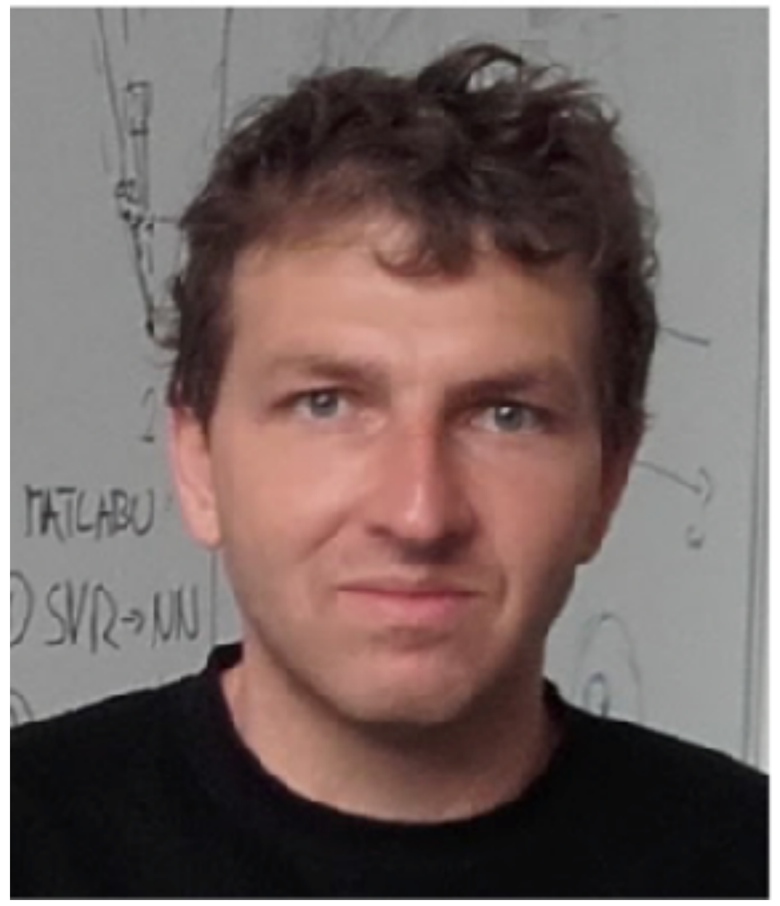


Text encoder
(BERT, GPT)

Generative model
(Diffusion, VQ-VAE, GAN)

“<SOS>”
“Best”
“UROB”
“teacher”
“ever”





SAM: Segment anything

[Facebook 2023] https://arxiv.org/pdf/2304.02643.pdf

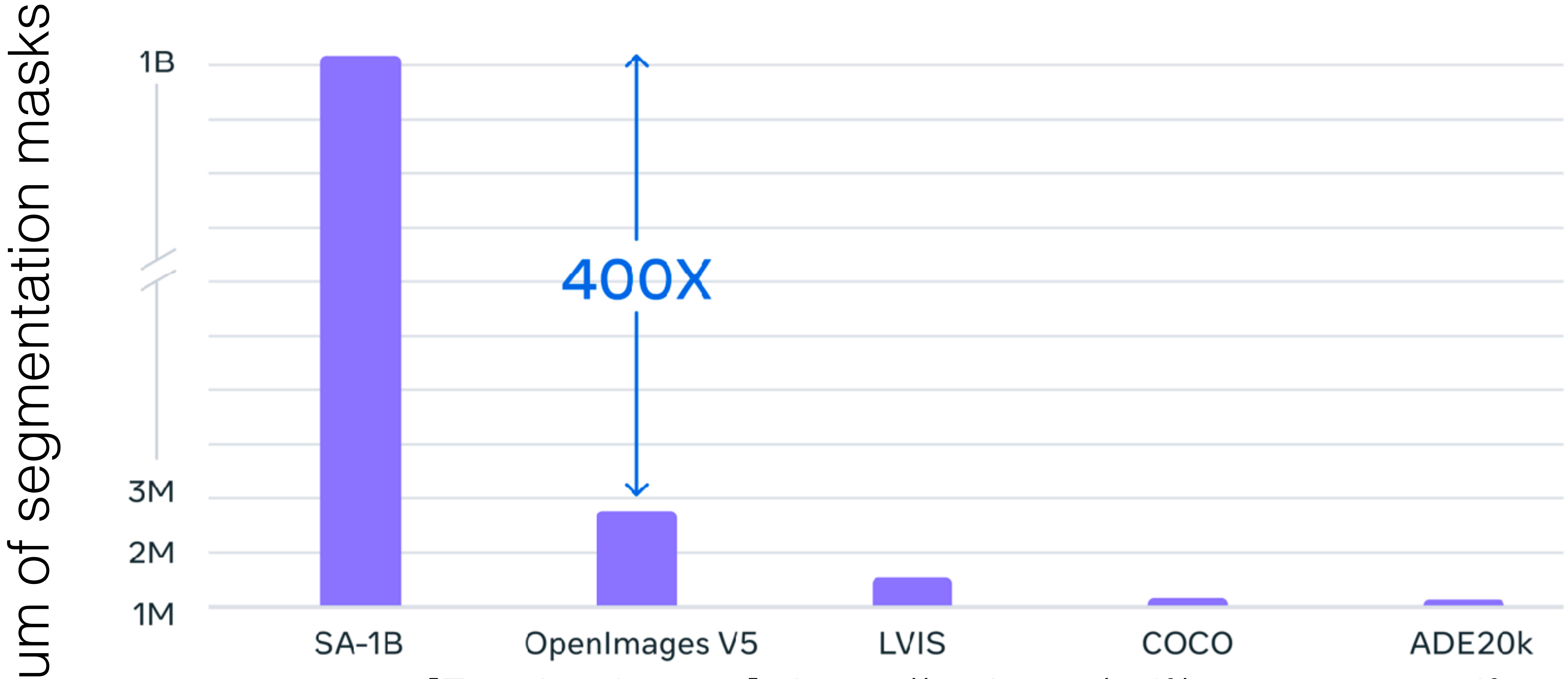
SAM: Segment anything

[Facebook 2023]   https://arxiv.org/pdf/2304.02643.pdf

SAM: Segment anything

Convolution is actually **structurally-enforced local attention.**

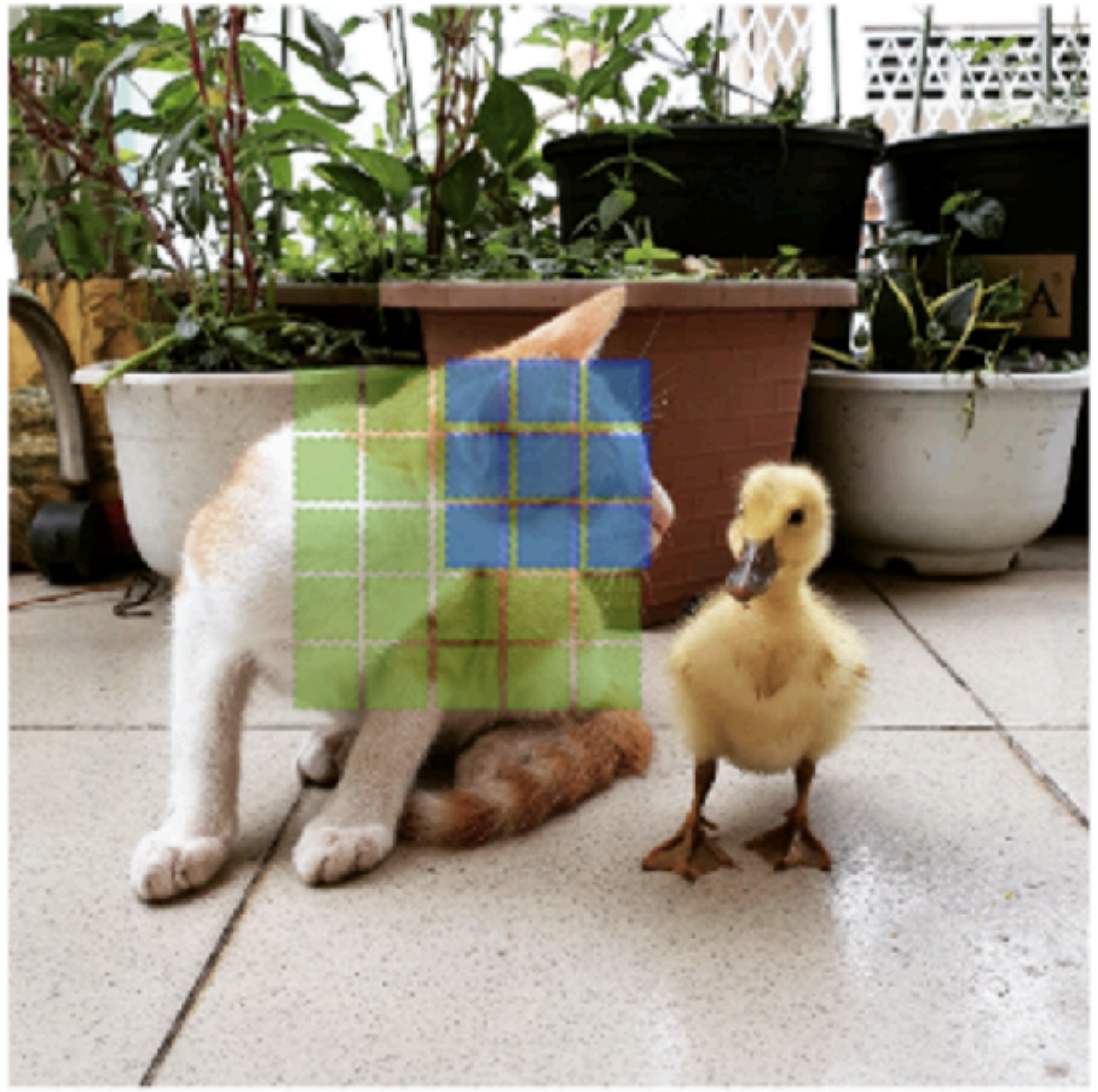
Transformers allow **global attention** and have to **learn** it **from data.**



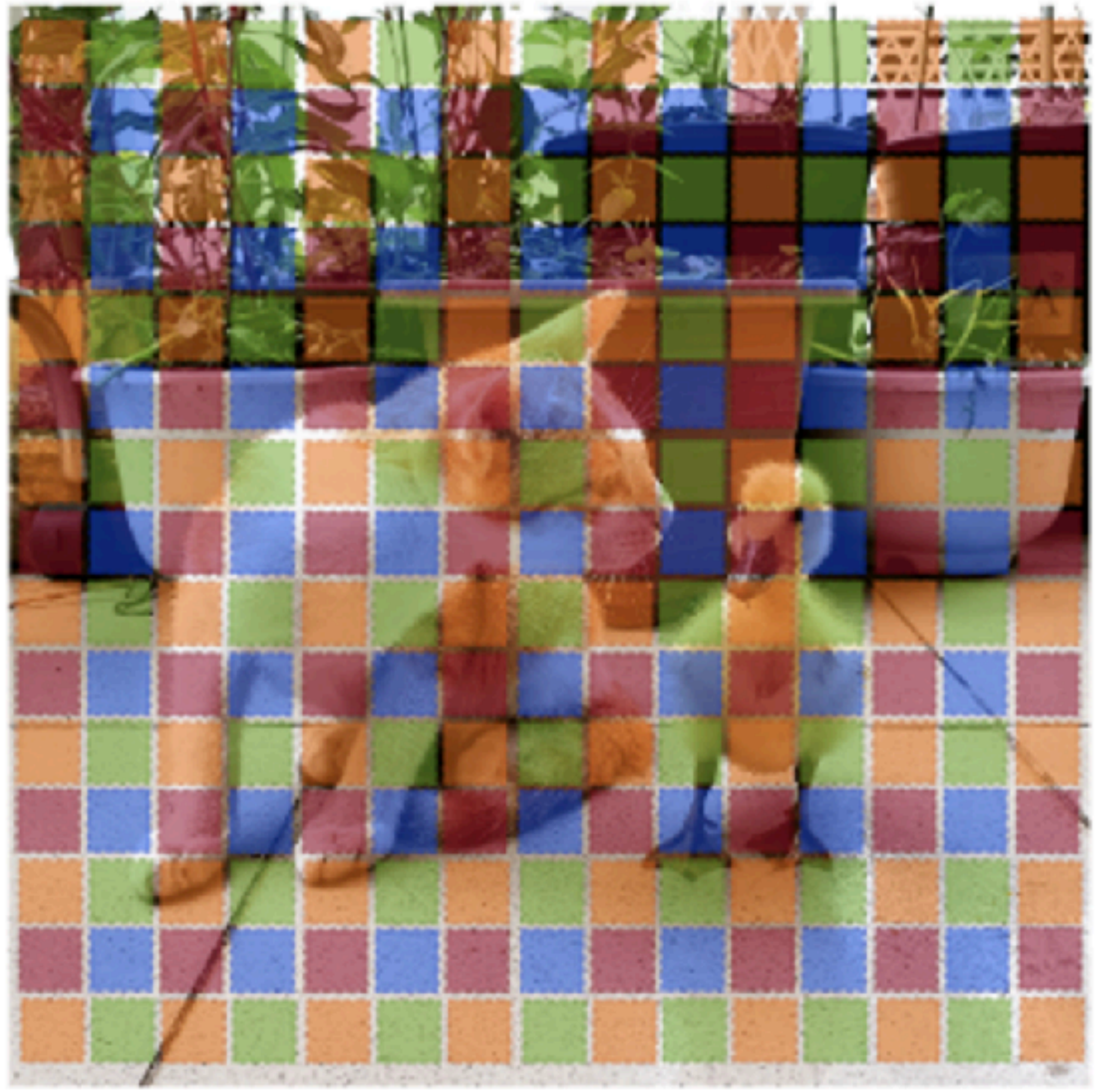
[Facebook 2023] https://arxiv.org/pdf/2304.02643.pdf

Attention used for images

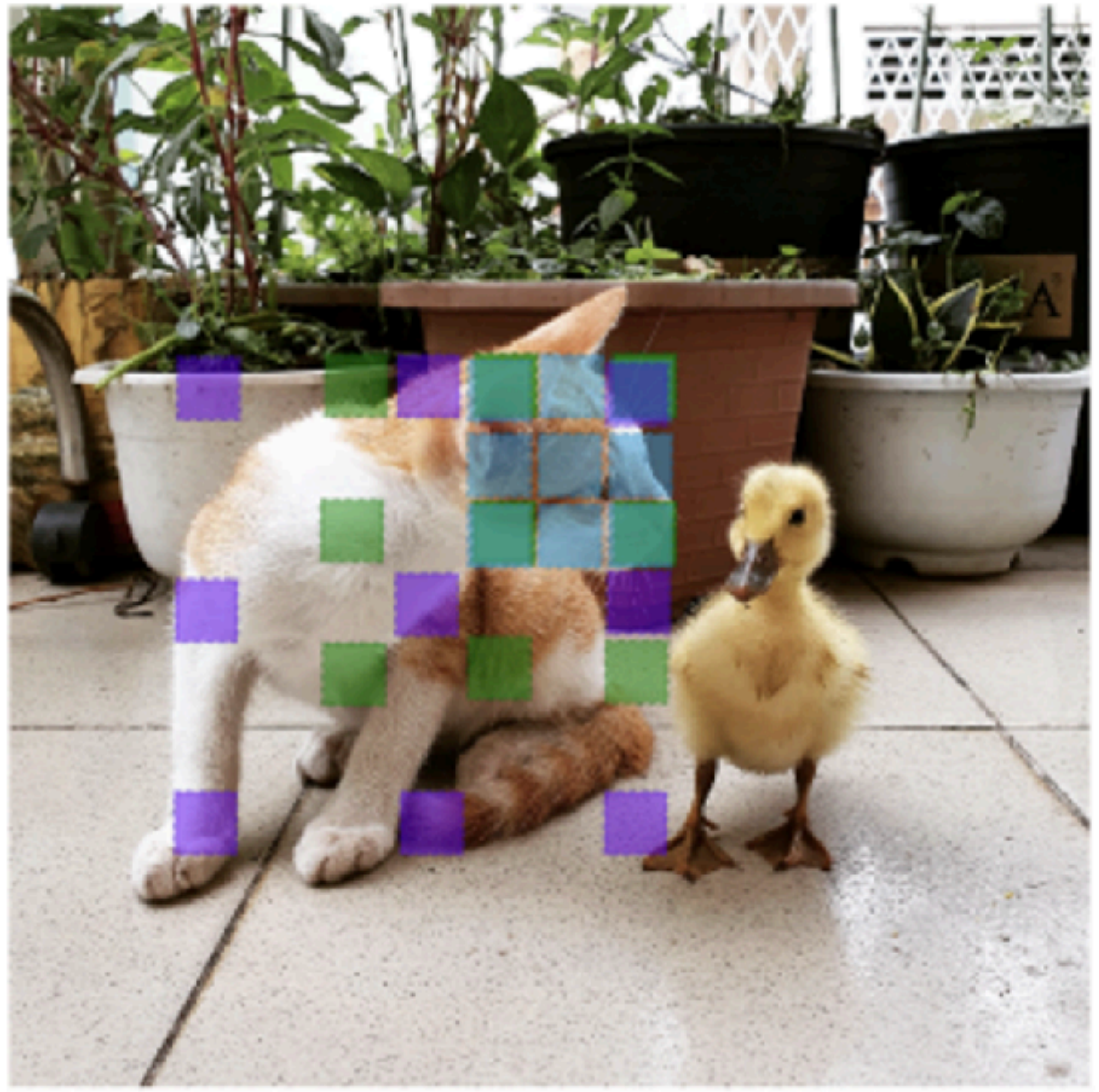
Global attention in early layers can be replaced by **local** attention



(a) Regional Attention



(b) Sparse Attention



(c) Atrous Attention

https://arxiv.org/pdf/2406.08859v1

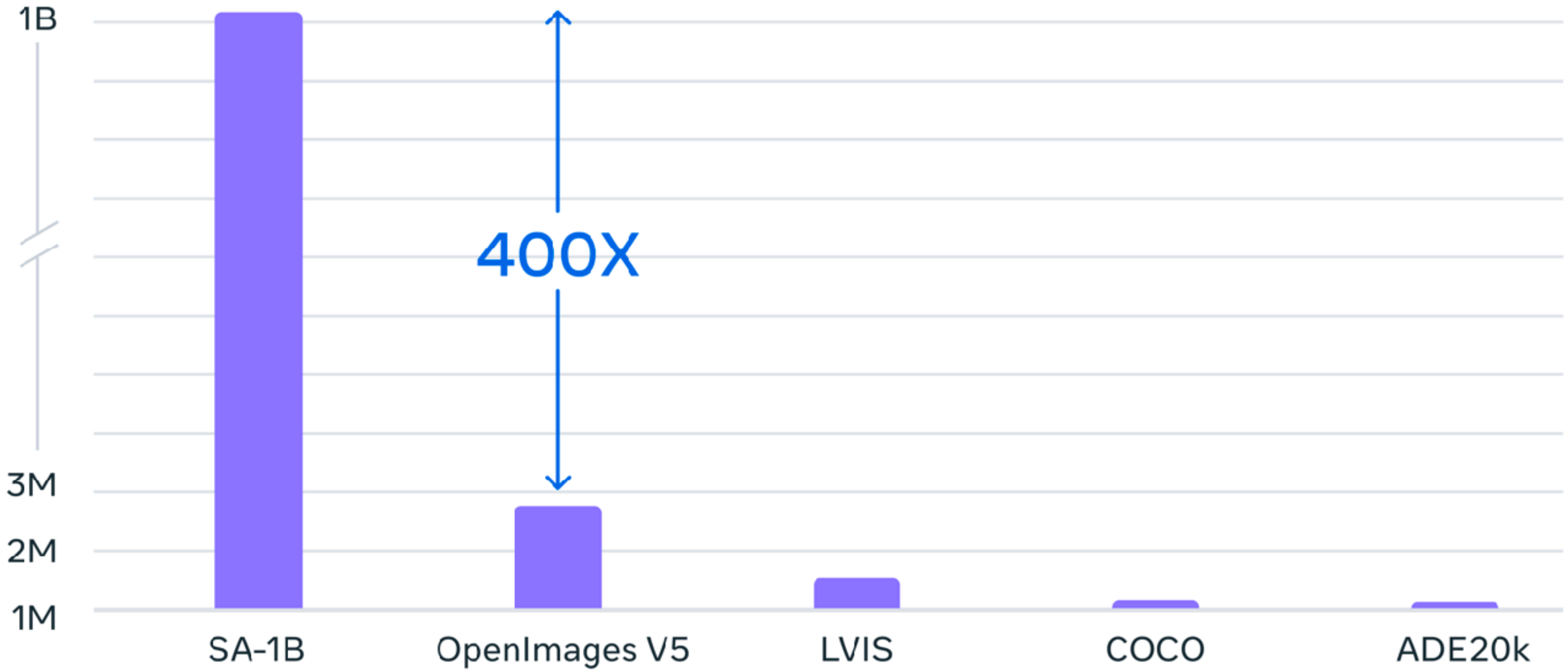
SAM: Segment anything

Convolution is actually **structurally-enforced local attention.**

Transformers allow **global attention** and have to **learn** it **from data.**

Change of paradigm:

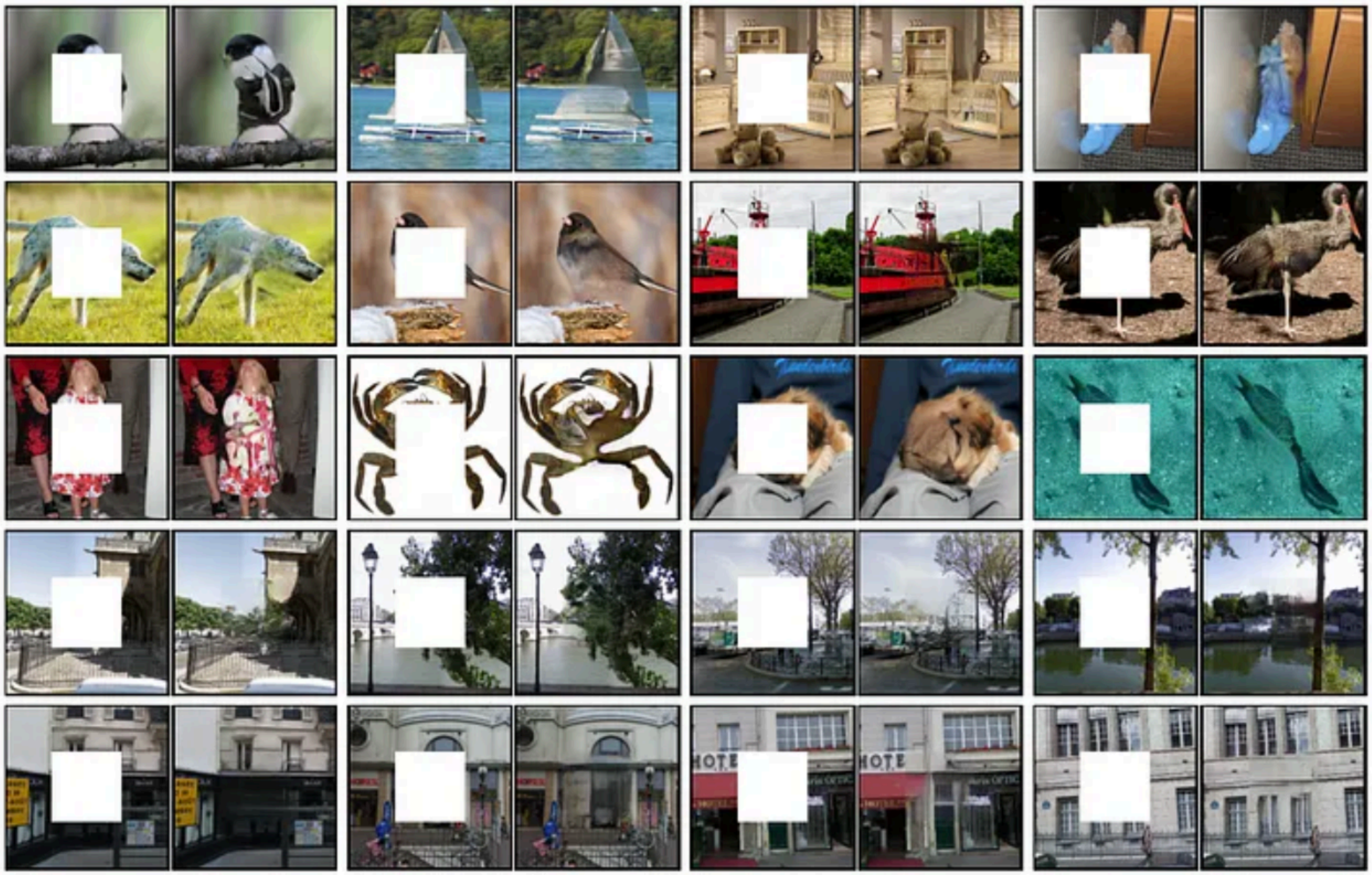
- **small** datasets => use **simple** models (strong inductive prior such as convolution)
- **huge** dataset with cheap or free training data => **complex** model **learn everything**
- **semi**-supervision / **self**-supervision



[Facebook 2023]   https://arxiv.org/pdf/2304.02643.pdf

Image impainting

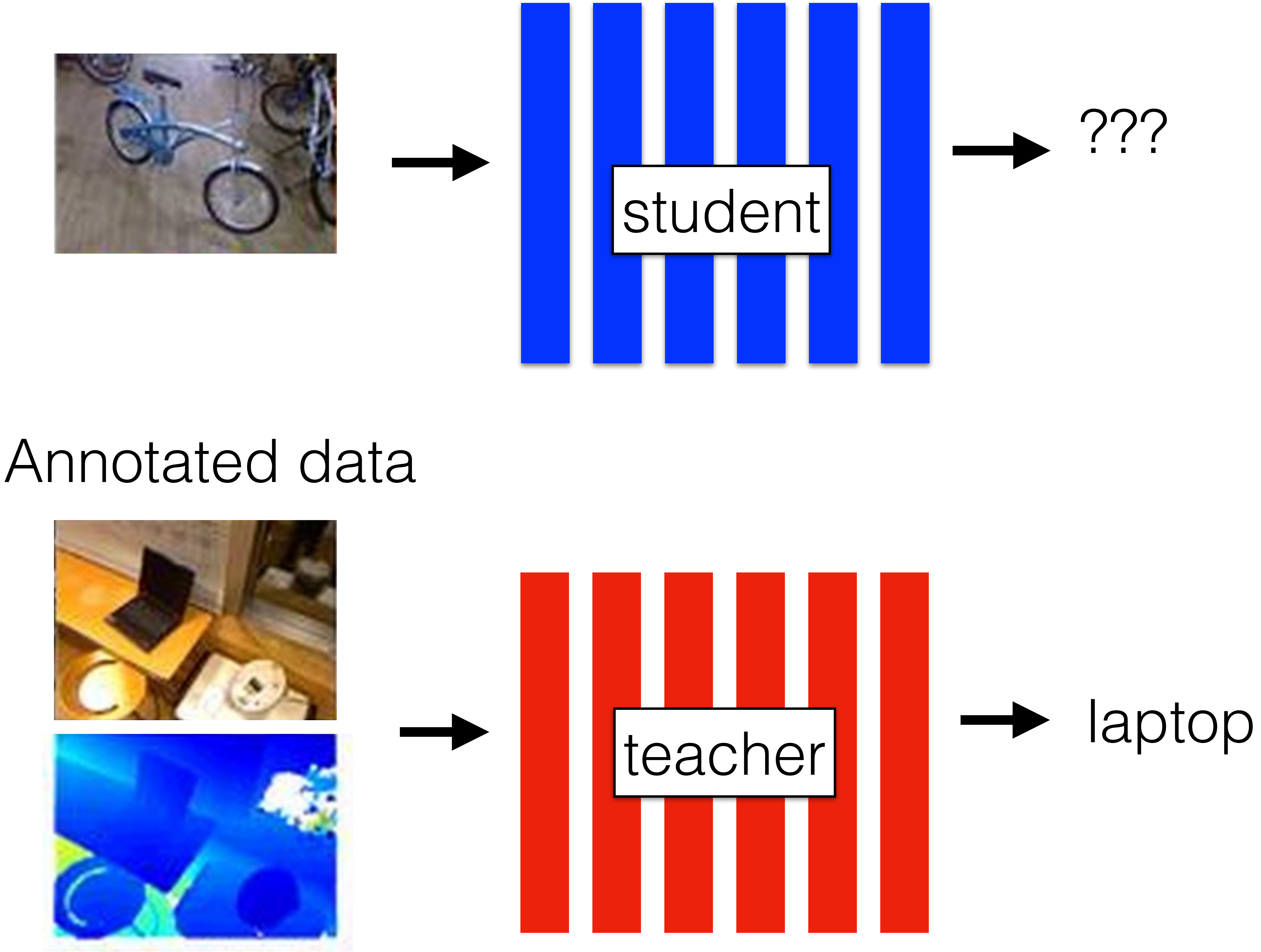
Image inpainting



Context encoders CVPR 2016 https://arxiv.org/abs/1604.07379

LUPI

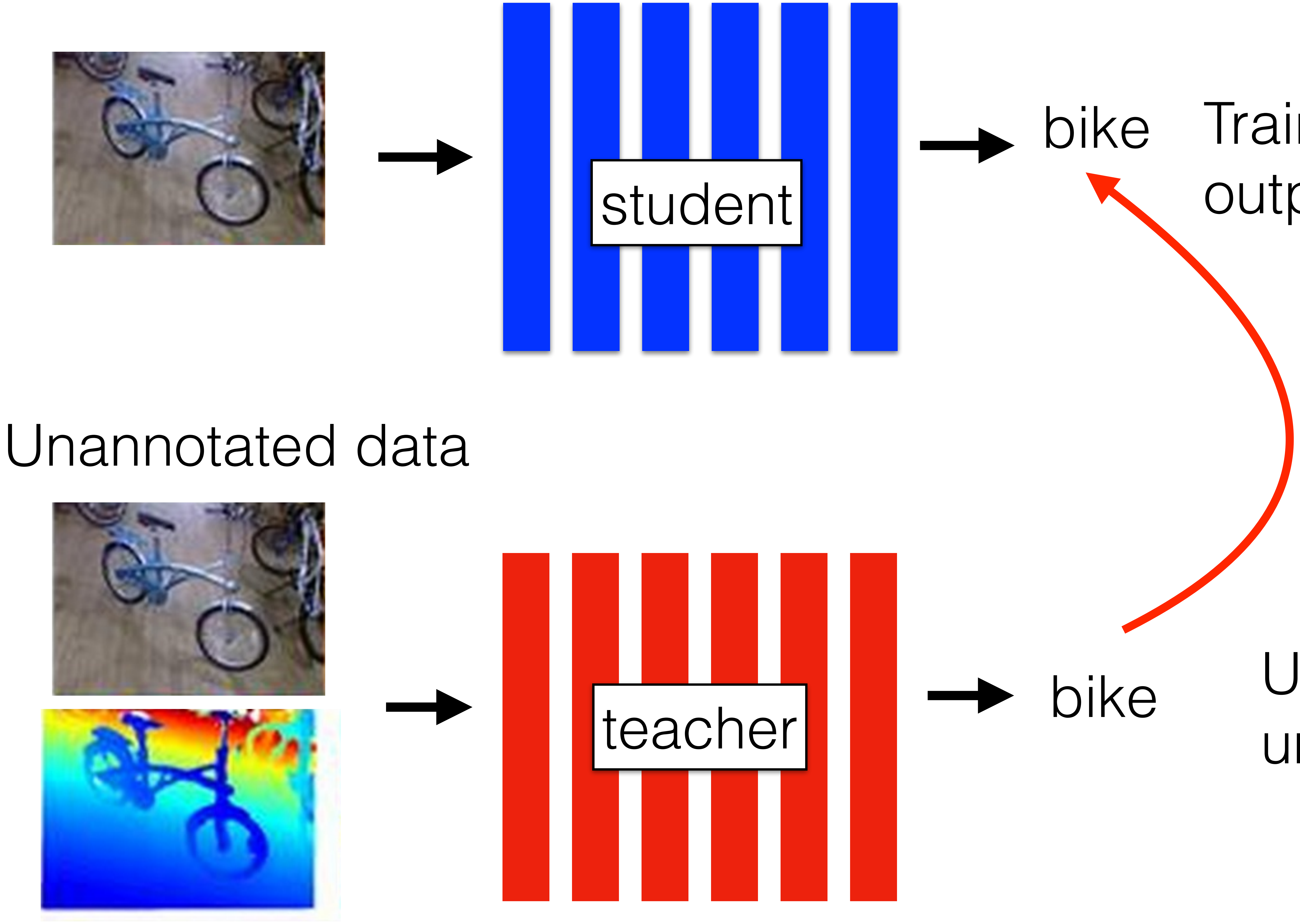
learning using privileged information



Train **teacher** on easier task with the access to **privileged** information

LUPI

learning using privileged information



Train **student** on teacher's outputs on **not annotated** data

Use teacher to classify unannotated data.

DINO

learning class-level features through the contrastive learning objective applied to the class token

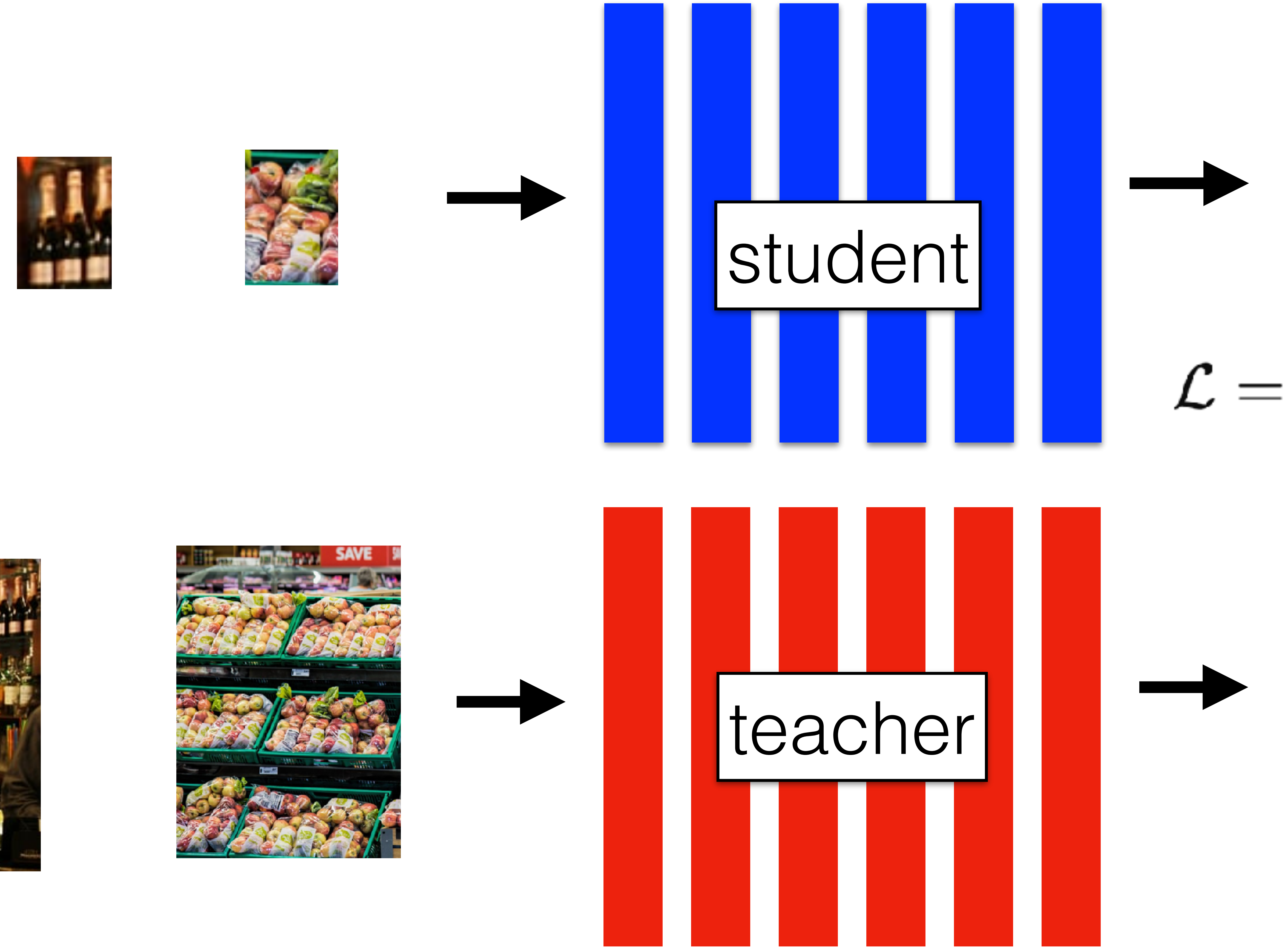
DINO

learning class-level features through the contrastive learning objective applied to the class token

Collapse avoided by:

- Sharpening $p_k^{(\mathbf{t}_i)} = \frac{\exp(\mathbf{t}_{i,k}/\tau_t)}{\sum_j \exp(\mathbf{t}_{i,j}/\tau_t)}$
- Centering $\mathbf{t}'_i = \mathbf{t}_i - \mathbf{c}$

Cropped patches

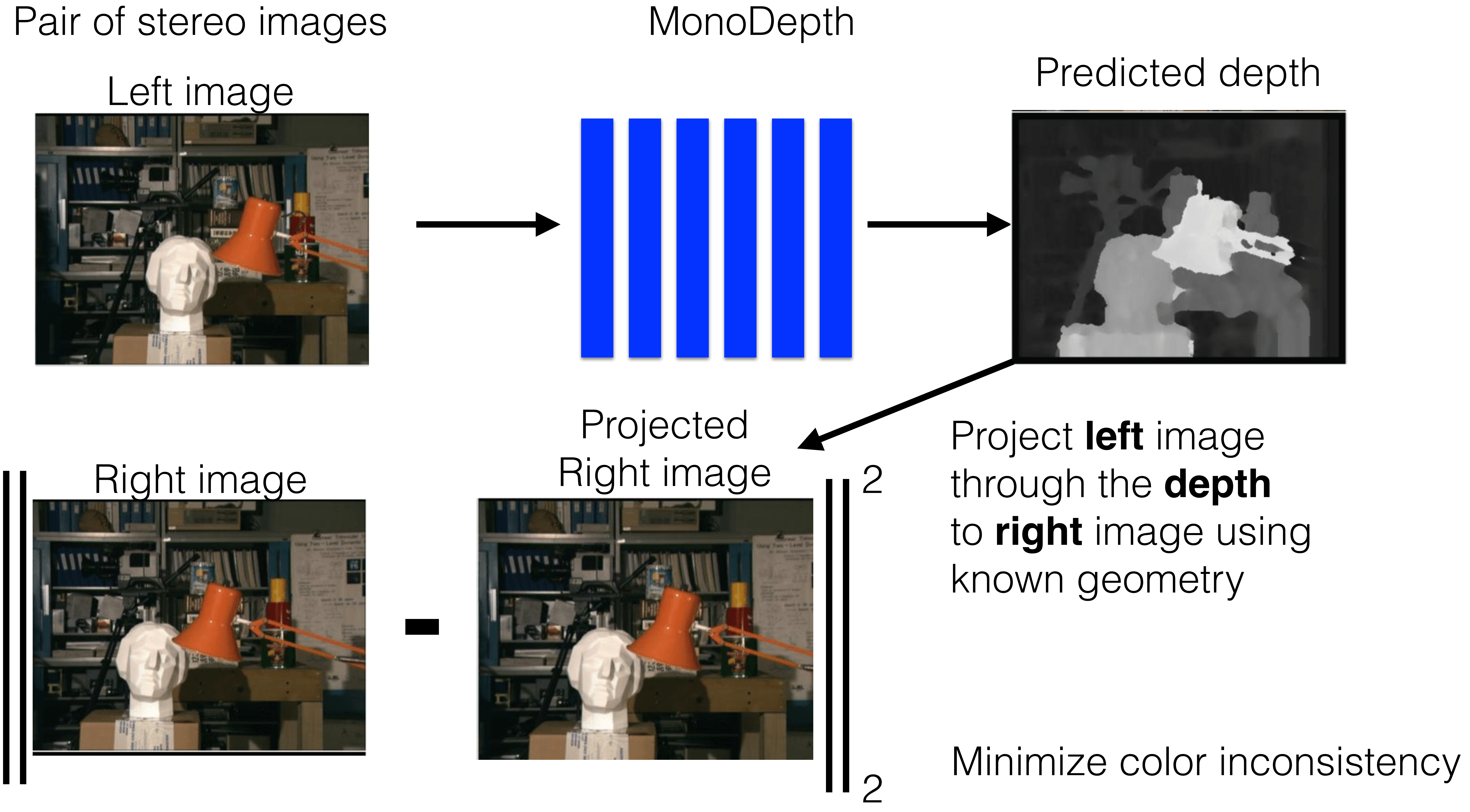


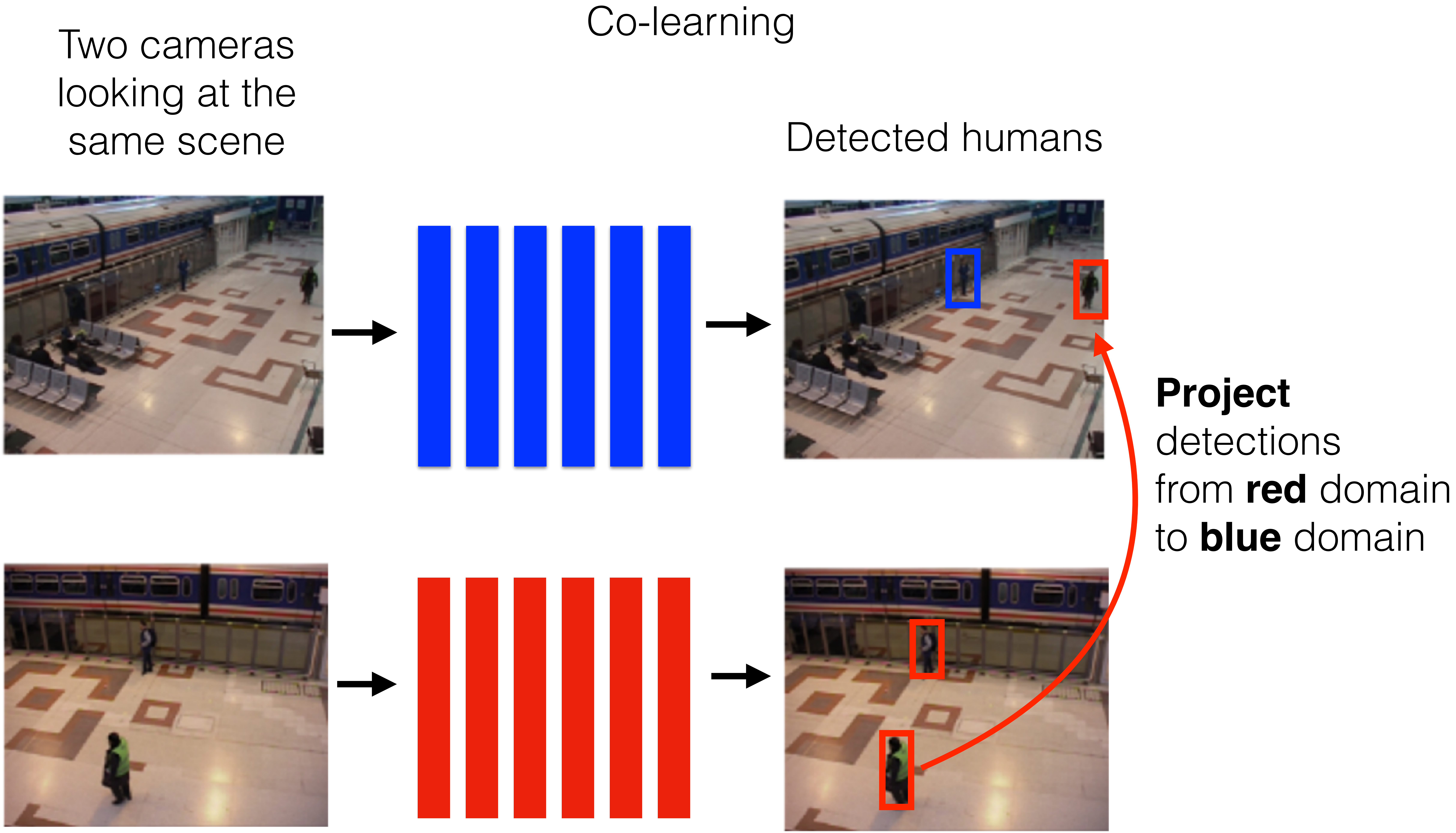
$\mathbf{s}_1$ $\mathbf{s}_2$

$$\mathcal{L} = \sum_i \mathrm{KL}\left(\mathrm{softmax}(\mathbf{t}_i) \,\|\, \mathrm{softmax}(\mathbf{s}_i)\right)$$

$\mathbf{t}_1$ $\mathbf{t}_2$

Forces the student to predict context-aware embedding


Pair of stereo images
MonoDepth
Left image
Predicted depth
Projected
Right image
Right image
2
2
Project **left** image
through the **depth**
to **right** image using
known geometry
Minimize color inconsistency


Co-learning
Two cameras looking at the same scene
Detected humans
Project detections from **red** domain to **blue** domain

Learning to mimic other sensor

Conclusions

- **ConvNets**
  - enforced **local** attention
  - data-independent **fixed attention**
  - data-independent **fixed** kernel **weights**

- **Transformers**
  - learned **global** attention
  - data-dependent **dynamic attention** (different for different content from **Q**,**K**)
  - data-dependent **dynamic weights** (different for different content from **V**)

- Big **foundation models** (such as SAM) delivered for various modalities images, depthmaps, pointclouds, text, speech

- In order to deliver billions of training data **self-supervision** is required