UNIVERSITÀ PARTHENOPE
DiST
Dipartimento di Scienze e Tecnologie

Machine Learning (part II)

Transformers

Angelo Ciaramella

Introduction

- Transformers
  - transform a set of vectors in some representation space into a corresponding set of vectors, having the same dimensionality, in some new space
  - the fundamental concept that underpins a transformer is attention
- Attention
  - developed as an enhancement to RNNs for machine translation
- A transformer can be viewed as a richer form of embedding in which a given vector is mapped to a location that depends on the other vectors in the sequence

Attention

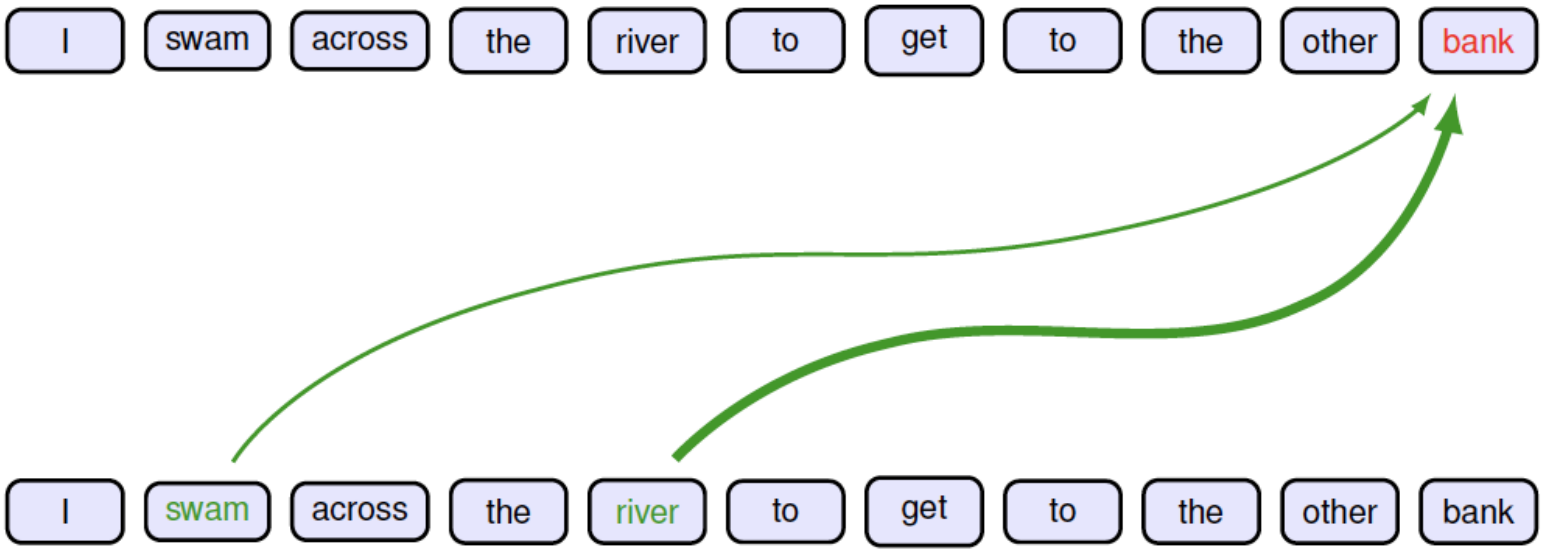
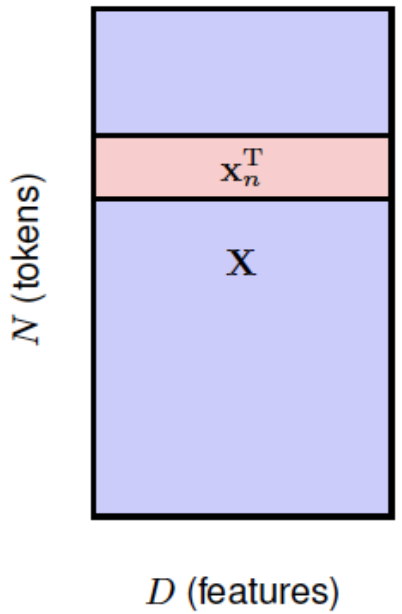


Figure 12.1 Schematic illustration of attention in which the interpretation of the word 'bank' is influenced by the words 'river' and 'swam', with the thickness of each line being indicative of the strength of its influence.

Introduction

Figure 12.3 The structure of the data matrix $\mathbf{X}$, of dimension $N \times D$, in which row n represents the transposed data vector $\mathbf{x}_n^{\mathrm{T}}$.



$$\widetilde{\mathbf{X}} = \mathrm{TransformerLayer}\left[\mathbf{X}\right]$$

function that takes a data matrix as input and creates a transformed matrix of the same dimensionality as the output

Deep networks

- Multiple transformer layers
  - to construct deep networks capable of learning rich internal representations
  - each transformer layer contains its own weights and biases, which can be learned using gradient descent using appropriate cost function

Attention coefficients

attention weights (non-negative)

$$\mathbf{y}_n = \sum_{m=1}^{N} a_{nm}\mathbf{x}_m$$

$$a_{nm} \geqslant 0$$

$$\sum_{m=1}^{N} a_{nm} = 1.$$

The coefficients should be close to zero for input tokens that have little influence on the output y_n and largest for inputs that have most influence

Self-Attention

- choosing which movie to watch in an online movie streaming service
  - associate each movie with a list of attributes
  - attributes of each movie in a vector called the key
  - the corresponding movie file itself is called a value
  - personal vector of values for the desired attributes called query
- movie service could then compare the query vector with all the key vectors to find the best match and send the corresponding movie to the user in the form of the value file (hard attention)

Soft attention

- we use continuous variables to measure the degree of match between queries and keys
  - variables to weight the influence of the value vectors on the outputs

dot product

$$a_{nm} = \frac{\exp(\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m)}{\sum_{m'=1}^{N}\exp(\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_{m'})}$$

softmax

normalization

  - if all the input vectors are orthogonal

$$\mathbf{y}_m = \mathbf{x}_m \text{ for } m = 1, \ldots, N$$

Self attention

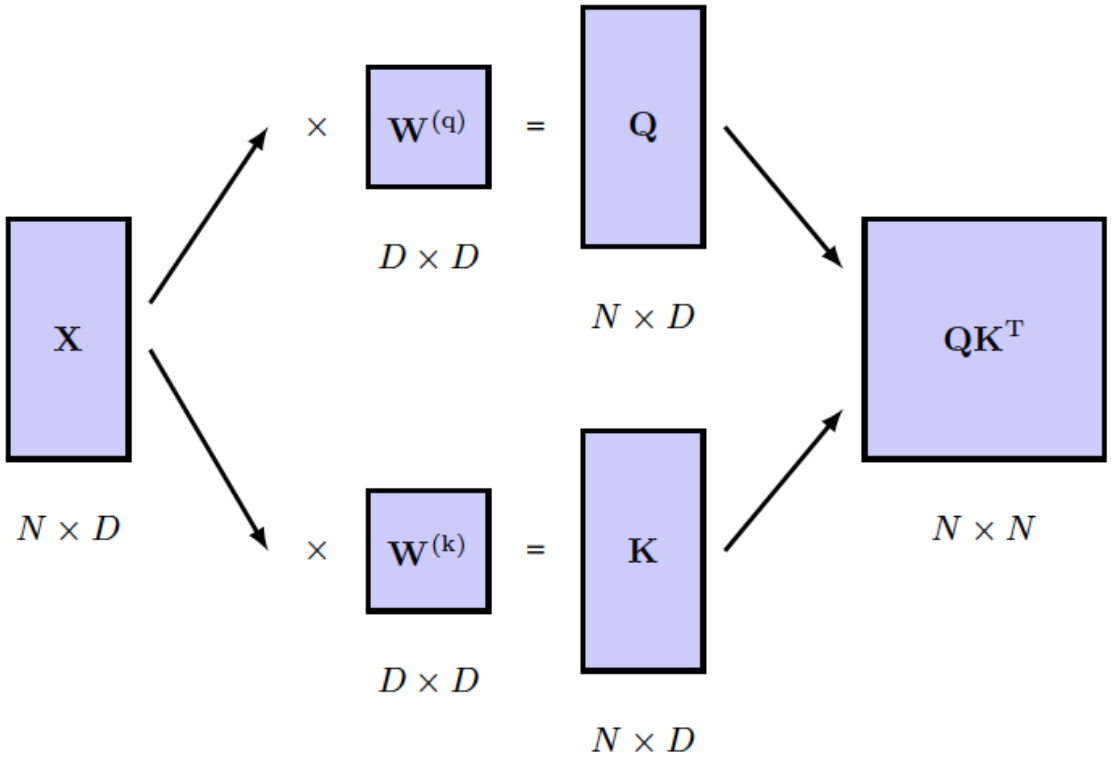
- Matrix notation

dot product self-attention

$$\mathbf{Y} = \text{Softmax}\left[\mathbf{X}\mathbf{X}^{\mathrm{T}}\right]\mathbf{X}$$

- we are using the same sequence to determine the queries, keys, and values

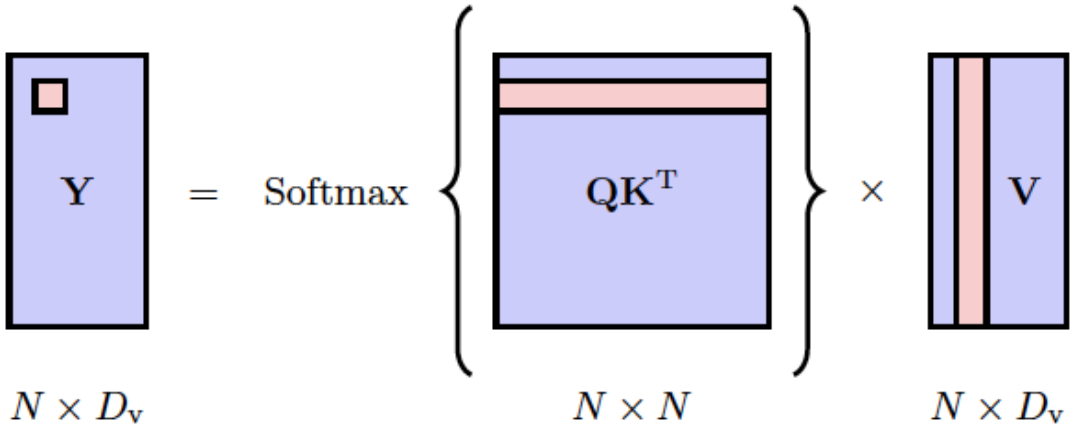
Self attention



dot product self-attention

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^{(q)}$$
$$\mathbf{K} = \mathbf{X}\mathbf{W}^{(k)}$$
$$\mathbf{V} = \mathbf{X}\mathbf{W}^{(v)}$$

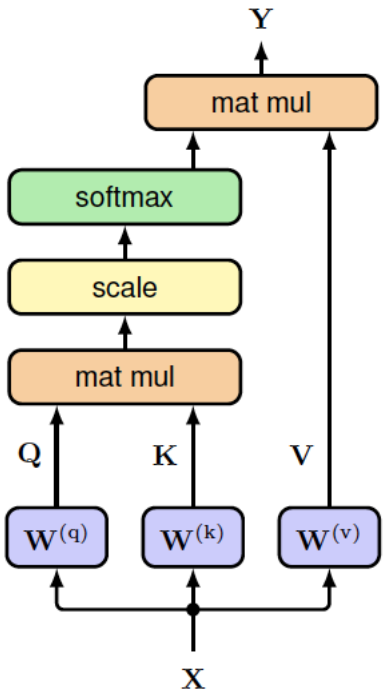
Self attention



output from an attention layer

Scaled self-attention

- the gradients of the softmax function become exponentially small for inputs of high magnitude
- variance of the dot product would be D_K



Scaled dot-product self-attention

$$\mathbf{Y} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \equiv \text{Softmax}\left[\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{D_{\mathrm{k}}}}\right]\mathbf{V}$$

Multi-head attention

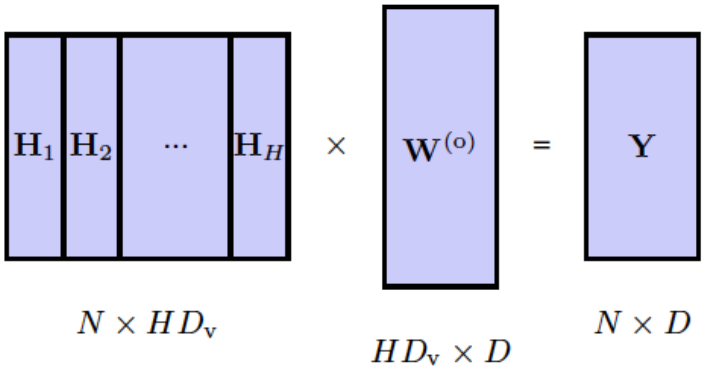
- in natural language some patterns might be relevant to tense whereas others might be associated with vocabulary

$$\mathbf{H}_h = \text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h)$$

$$\mathbf{Q}_h = \mathbf{X}\mathbf{W}_h^{(\text{q})}$$
$$\mathbf{K}_h = \mathbf{X}\mathbf{W}_h^{(\text{k})}$$
$$\mathbf{V}_h = \mathbf{X}\mathbf{W}_h^{(\text{v})}$$

$$\mathbf{Y}(\mathbf{X}) = \text{Concat}\left[\mathbf{H}_1, \ldots, \mathbf{H}_H\right]\mathbf{W}^{(\text{o})}$$

H heads



Multi-head attention

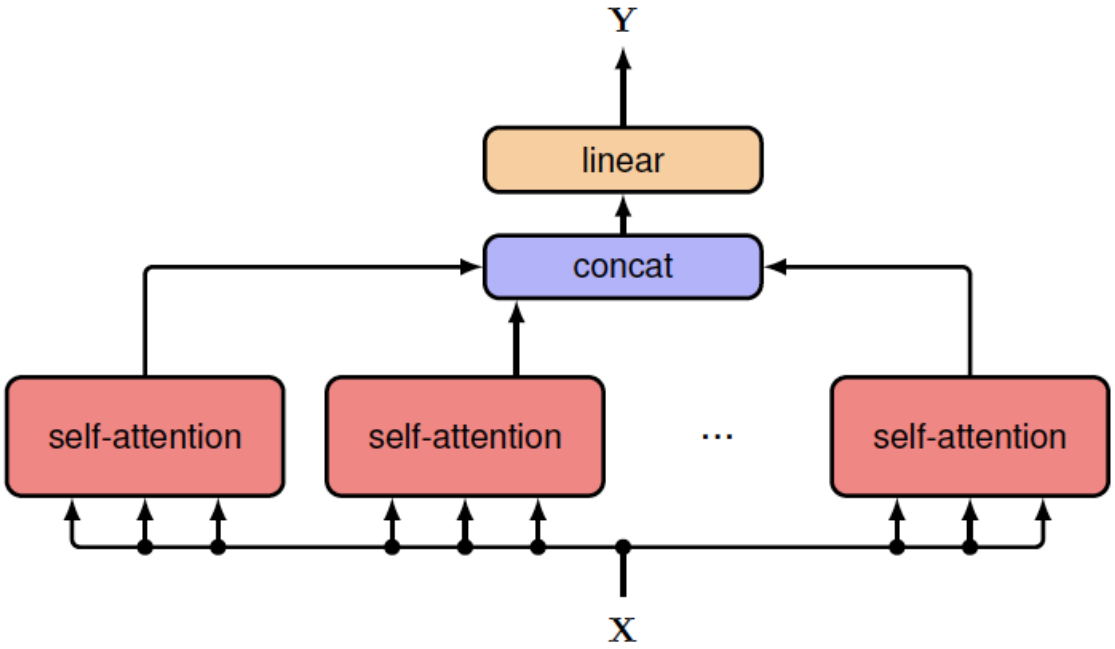


Figure 12.8 Information flow in a multi-head attention layer. The associated computation, given by Algorithm 12.2, is illustrated in Figure 12.7.

Residual connections

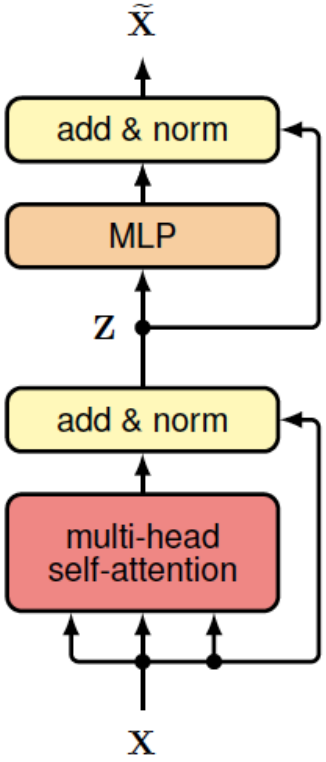
- Improving the learning

$$\mathbf{Z} = \text{LayerNorm}\left[\mathbf{Y}(\mathbf{X}) + \mathbf{X}\right]$$

H heads

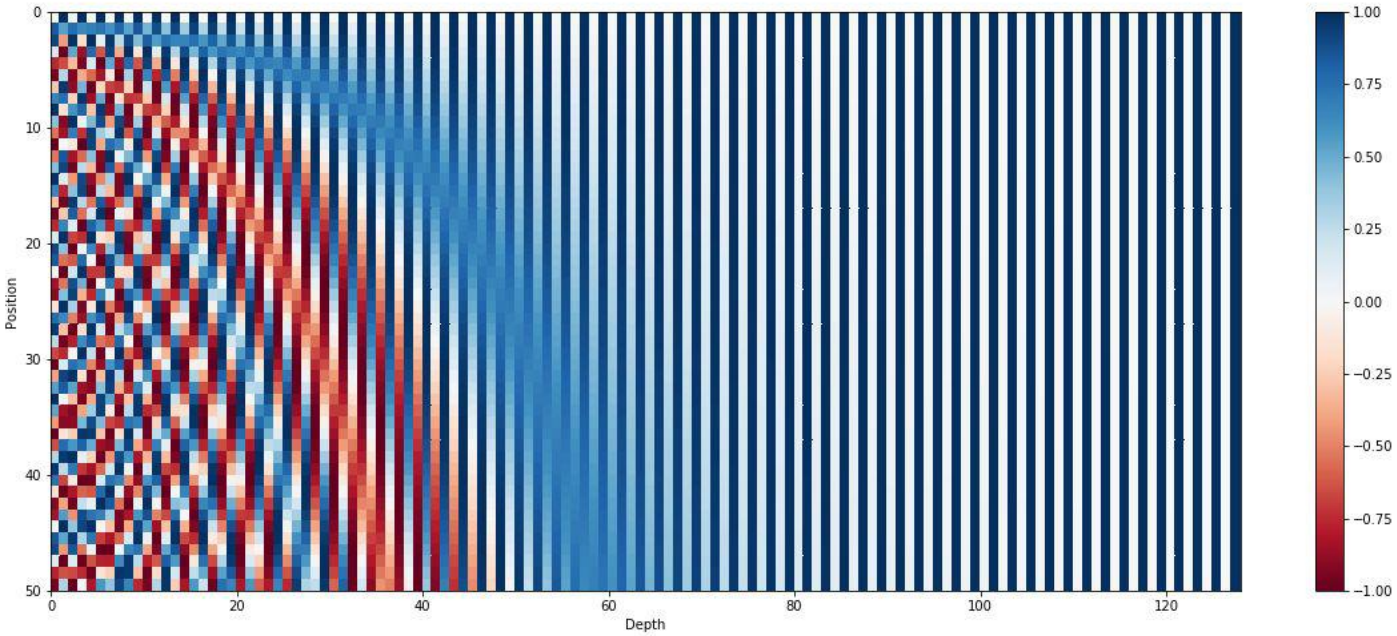
- Adding MLP

$$\tilde{\mathbf{X}} = \text{LayerNorm}\left[\text{MLP}\left[\mathbf{Z}\right] + \mathbf{Z}\right]$$



Positional embeddings

$$\mathrm{PE}(i,\delta) = \begin{cases} \sin\left(\frac{i}{10000^{2\delta'/d}}\right) & \text{if } \delta = 2\delta' \\ \cos\left(\frac{i}{10000^{2\delta'/d}}\right) & \text{if } \delta = 2\delta' + 1 \end{cases}$$



Attention is all You Need

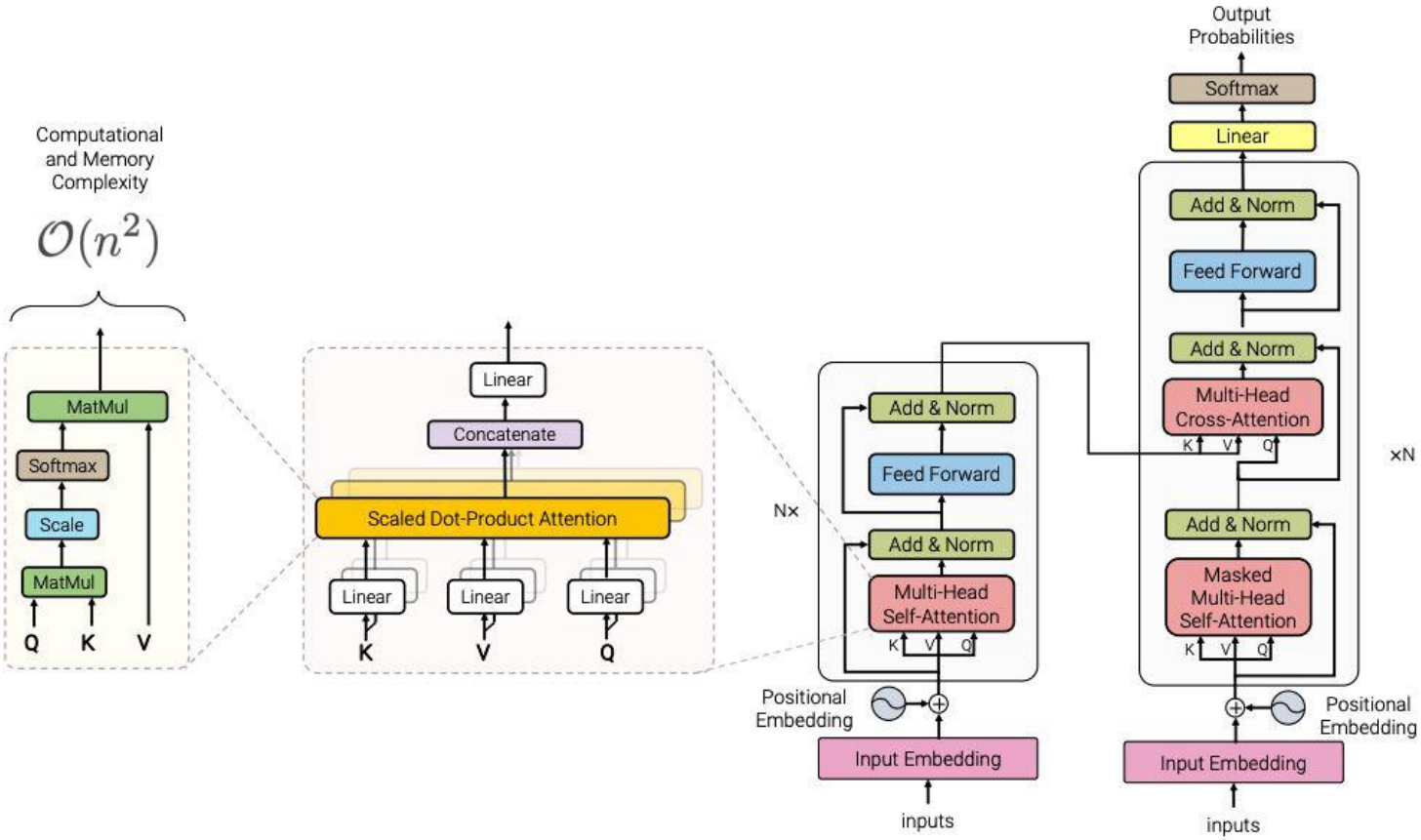
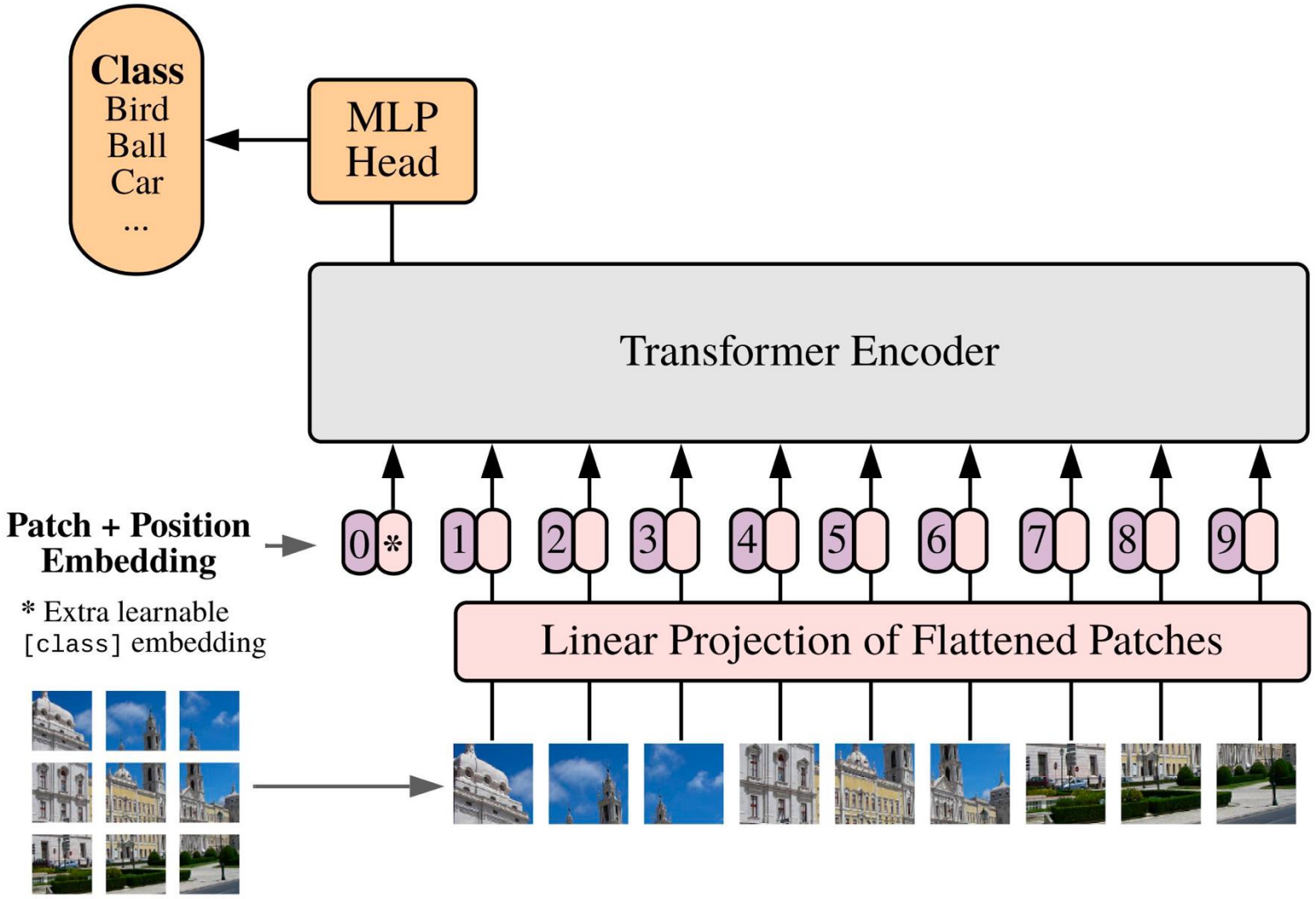


Figure 1: Architecture of the standard Transformer (Vaswani et al., 2017)

Vision Transformers



An Image is Worth 16x16 Words

Vision Transformers

Image captioning

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

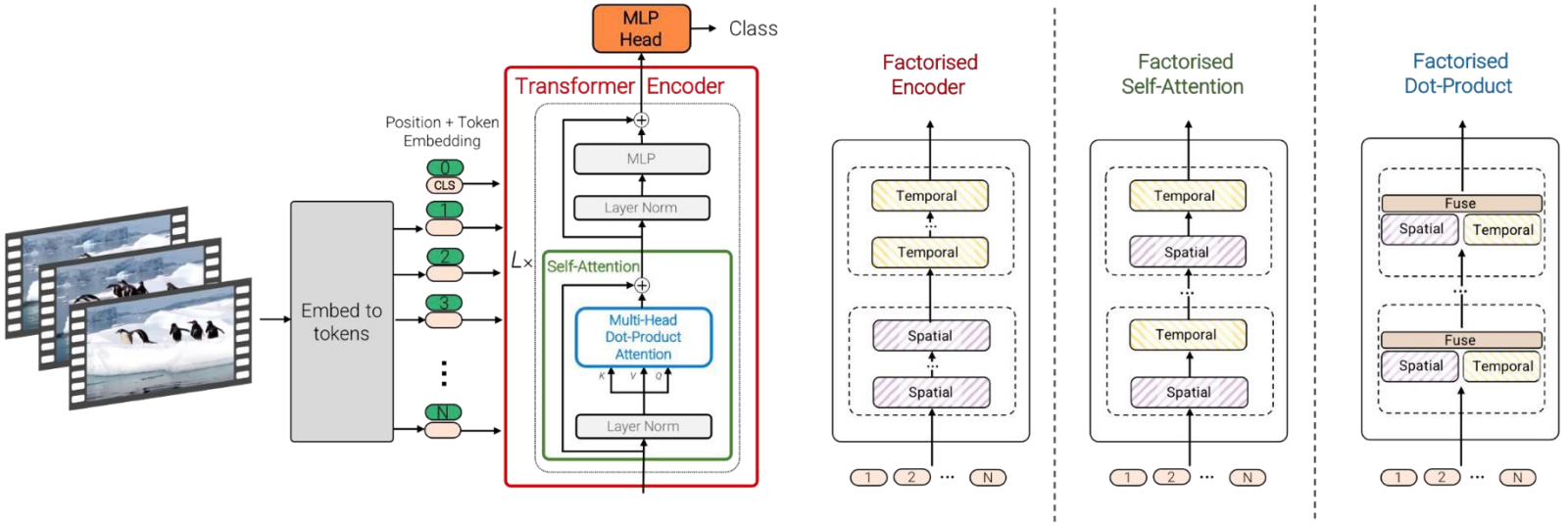
A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

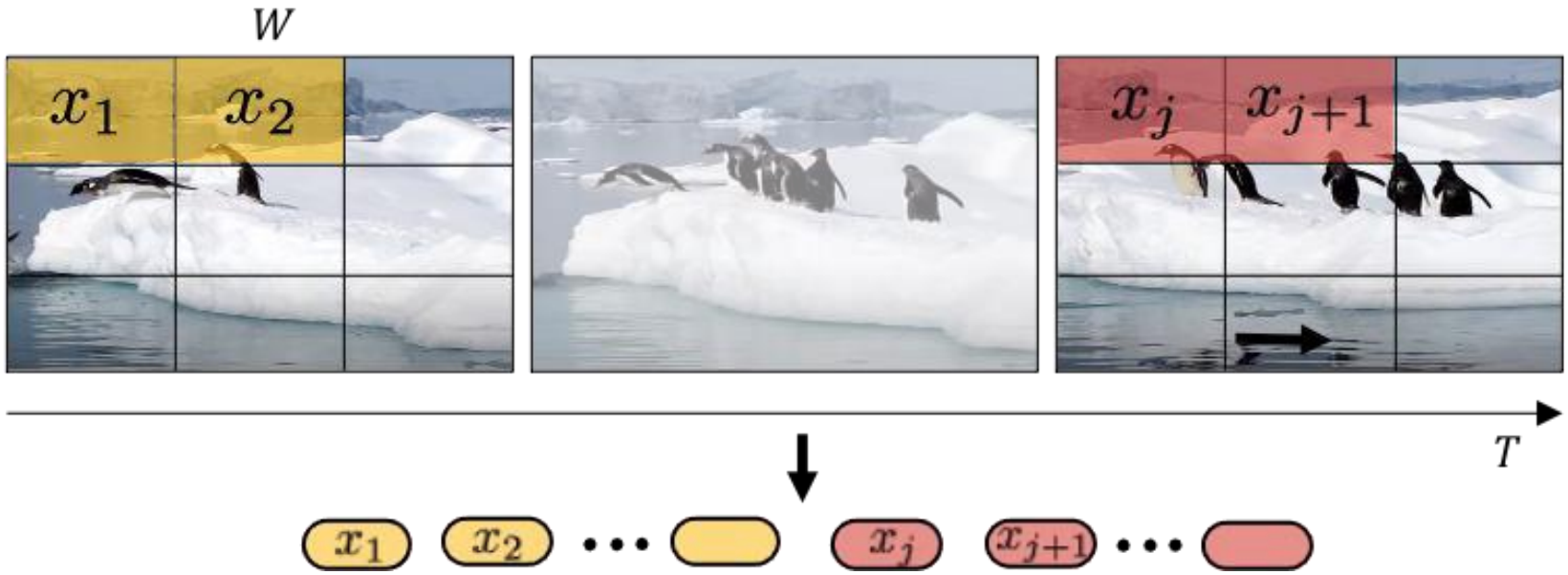
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

ViT for video

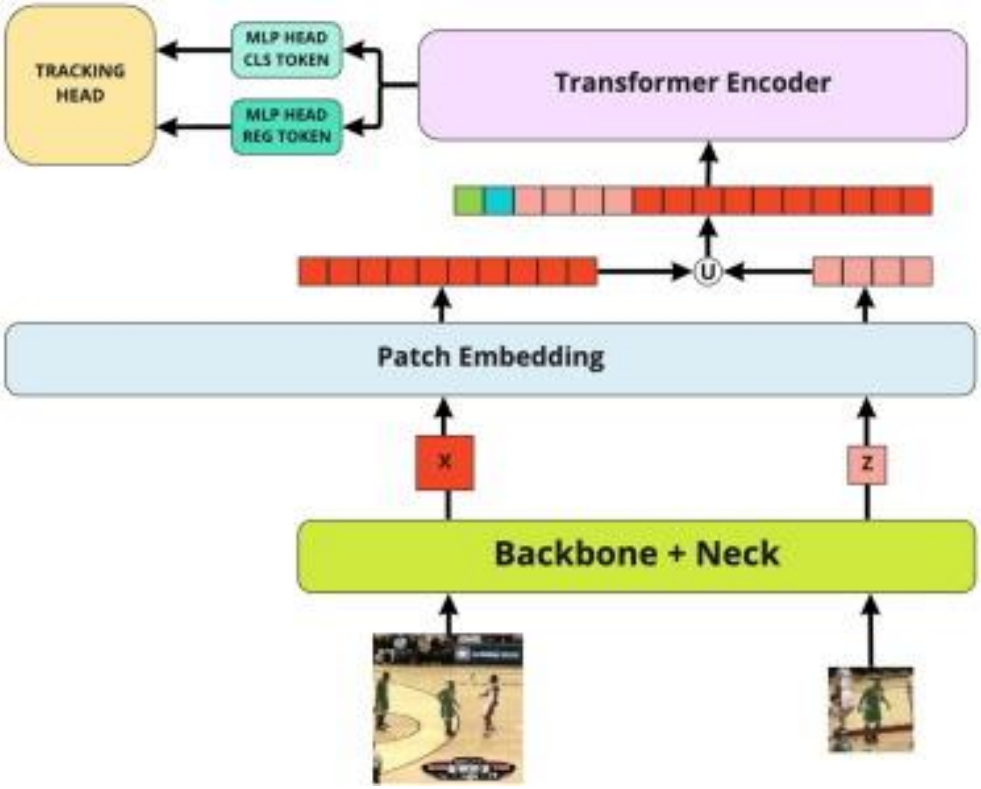


ViT for video



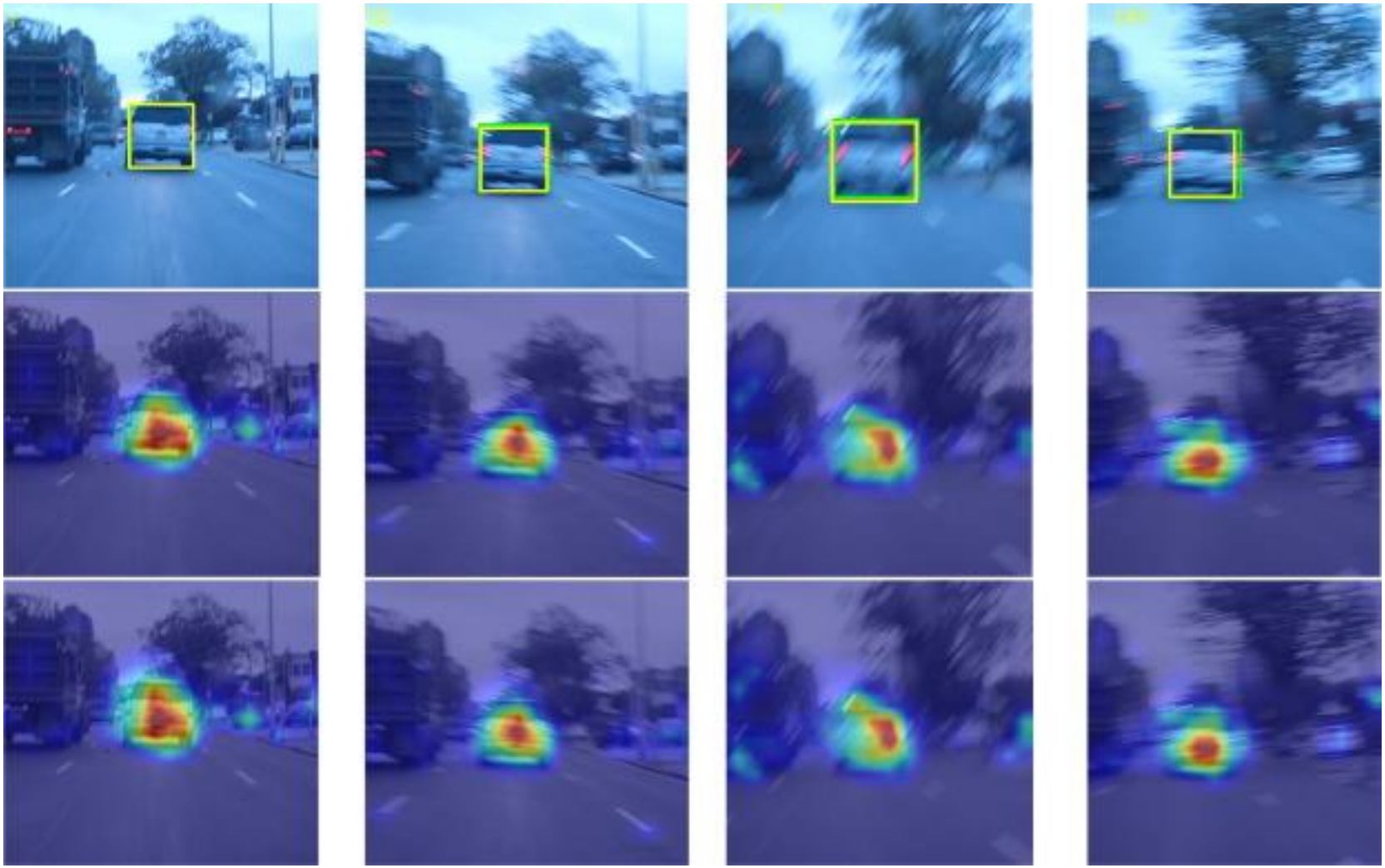
Sample frames, extract 2D patches and linearly project (as in ViT). Consider a video as a “big image”

Visual tracking



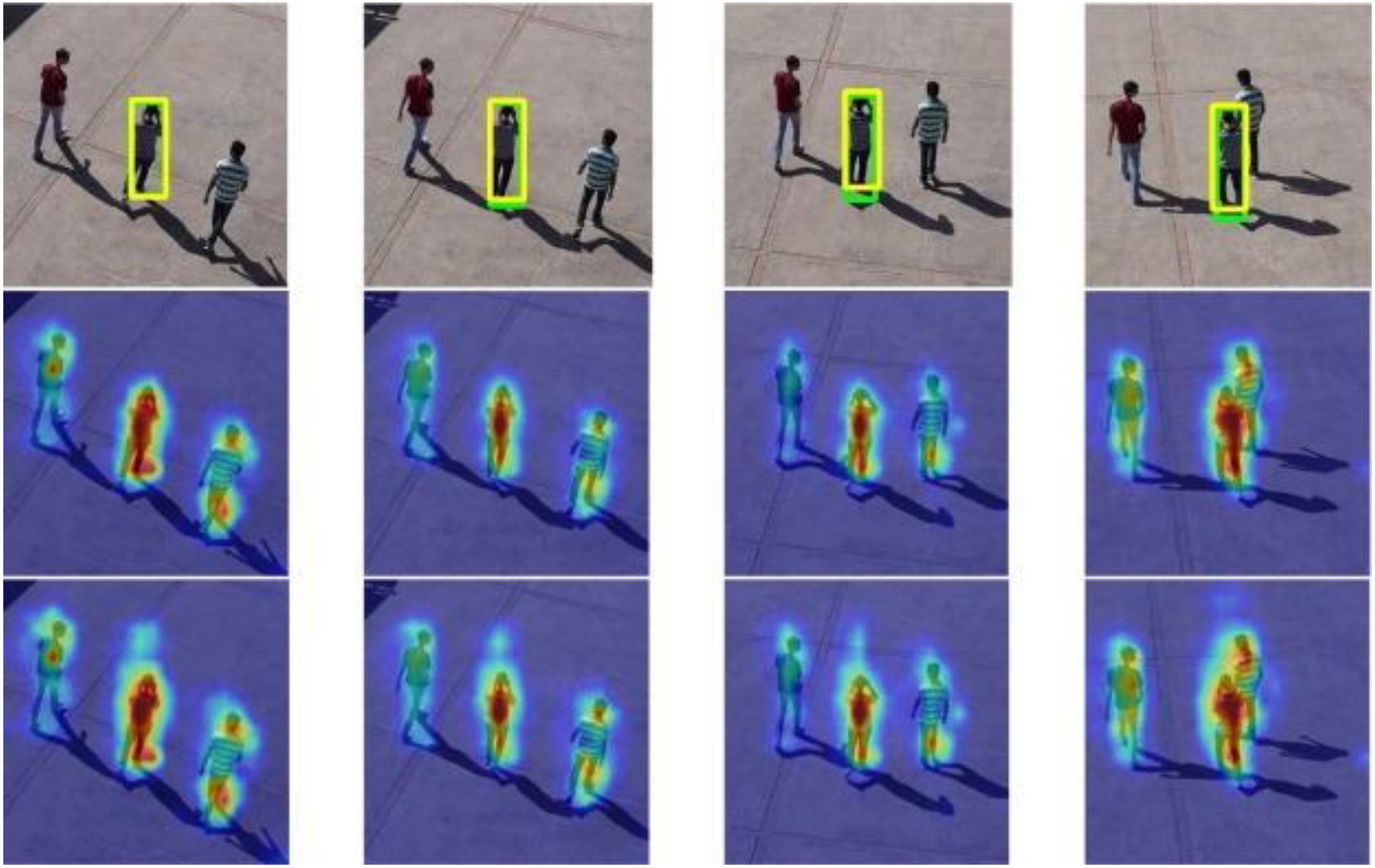
Tracking vision transformer with class and regression tokens

Visual tracking



Outputs for sequence *BlurCar1* in OTB100 with the attention of two tokens overlaying the image. *First row*: Bounding Box outputs. *Second row*: Regression token attention. *Third row*: Classification token attention

Visual tracking



Outputs for sequence *Group1_2* in UAV123 with the attention of two tokens overlaying the image. *First row*: Bounding Box outputs. *Second row*: Regression token attention. *Third row*: Classification token attention.