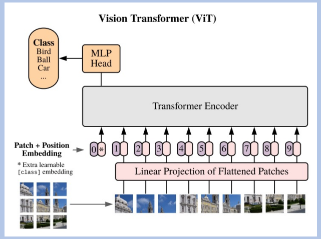
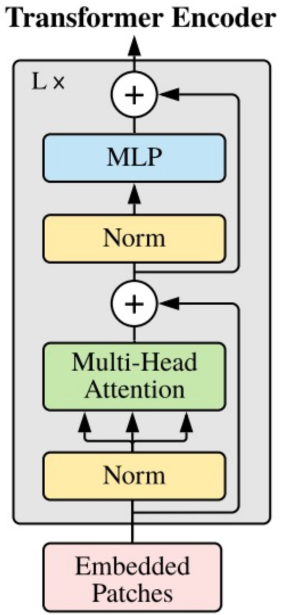
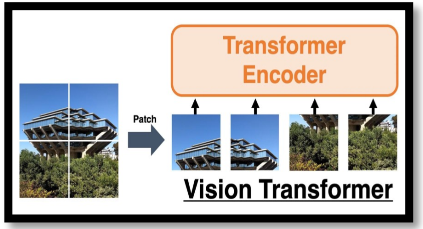
Outline

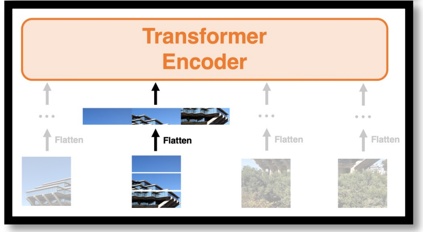
1. Attention and Vision Transformers (ViT)
    - NLP: Attention is all you need
    - Transformer Encoder ViT with Self Attention for image classification


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

Attention process in Vision







$$x \in \mathbb{R}^{H\times W\times C}$$

$$x_p \in \mathbb{R}^{N\times(P^2\cdot C)}$$

$$N = HW/P^2$$

CLS token

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\cdots;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E}\in\mathbb{R}^{(P^2\cdot C)\times D},\ \mathbf{E}_{pos}\in\mathbb{R}^{(N\boxed{+1})\times D}$$

$$\mathbf{z'}_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1\ldots L$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell, \qquad \ell = 1\ldots L$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0)$$

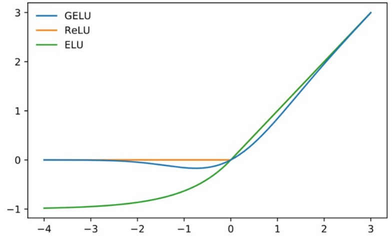
[class=CLS] token: a learnable embedding to the sequence of embedded patches

Layernorm (LN) before every block, and residual connections after every block

MSA: Multi Head Self Attention

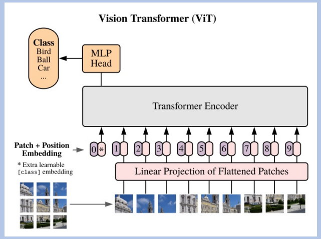
MLP: two layers with a GELU non-linearity

Hybrid Architecture : Raw image patches --> Feature map of a CNN



Outline

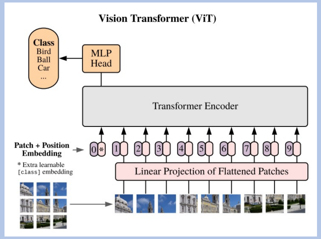
1. Attention and Vision Transformers (ViT)
    - NLP: Attention is all you need
    - Transformer Encoder ViT with Self Attention for image classification


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

2. **Transformer Decoder** for downstream tasks

Outline

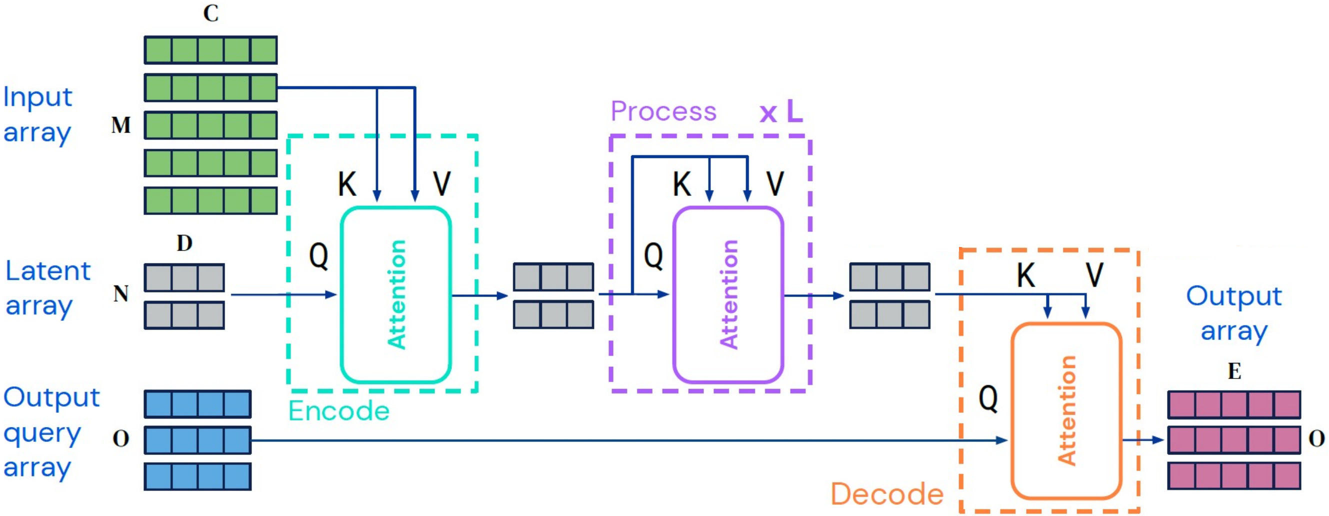
1. Attention and Vision Transformers (ViT)
   - NLP: Attention is all you need
   - Transformer Encoder ViT with Self Attention for image classification


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

2. **Transformer Decoder** for downstream tasks
   - Detection
   - Segmentation
   - Continual Learning, …

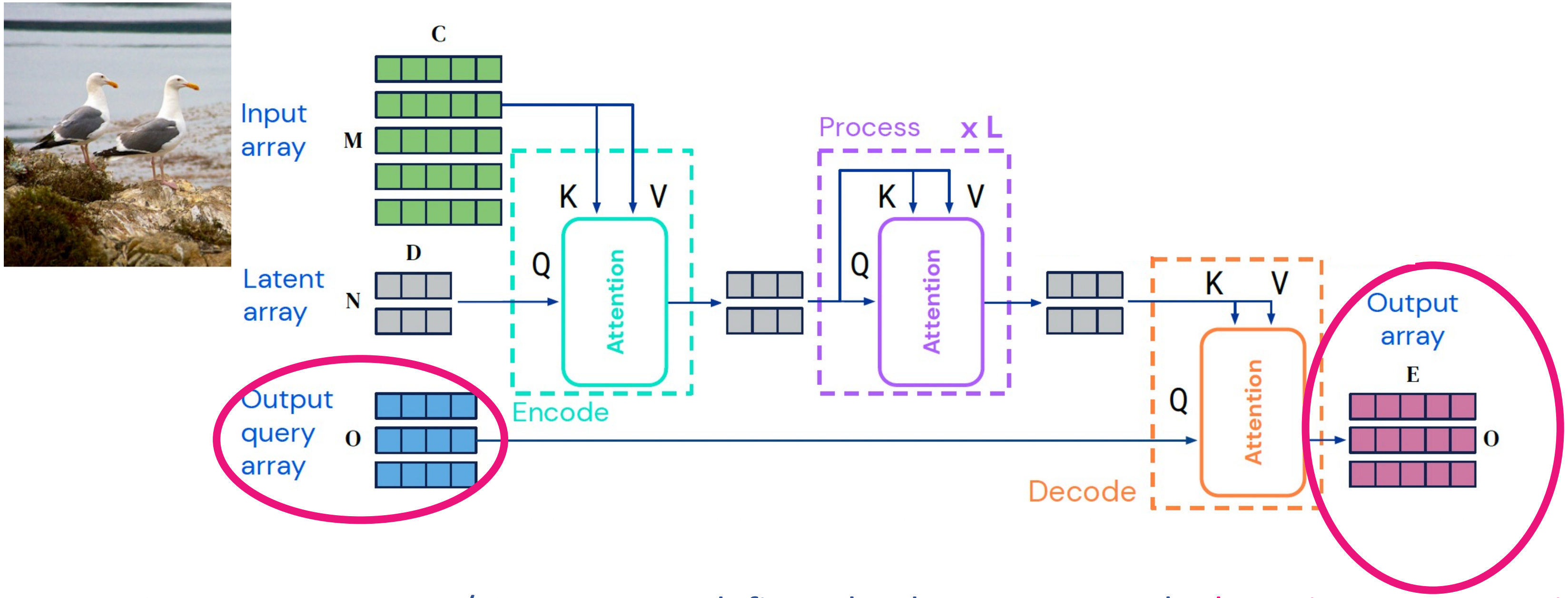
General Decoder

[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]



General Decoder

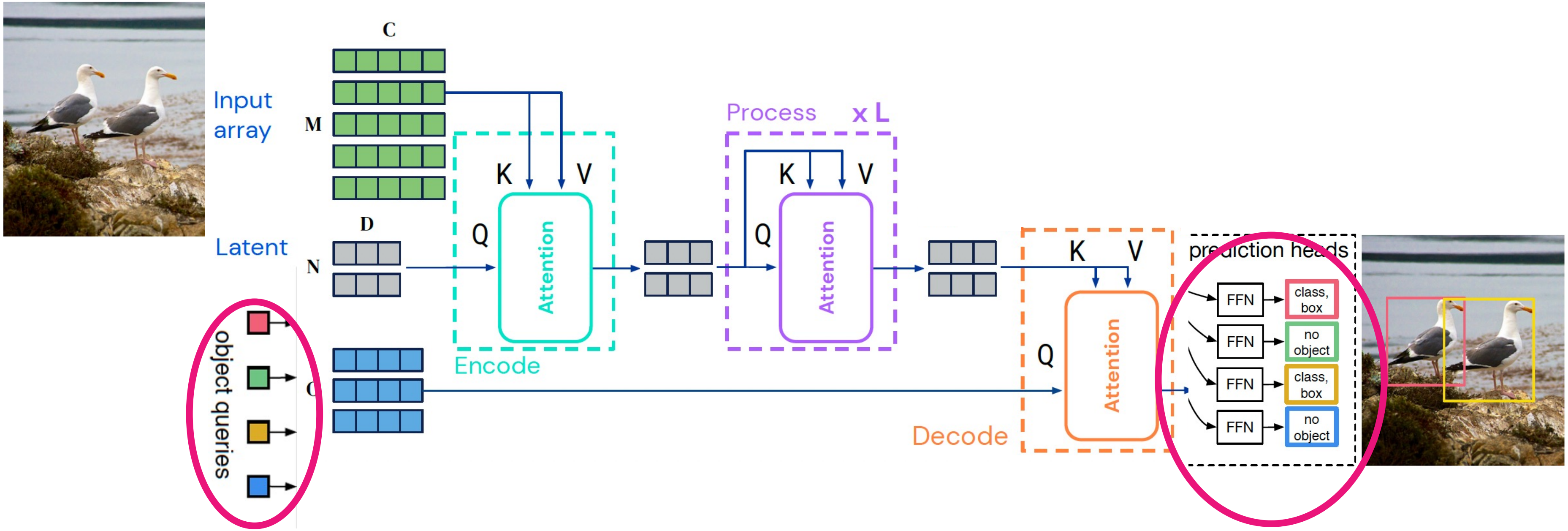
[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]



Output query array / Output array defines the downstream task: detection, segmentation …

General Decoder

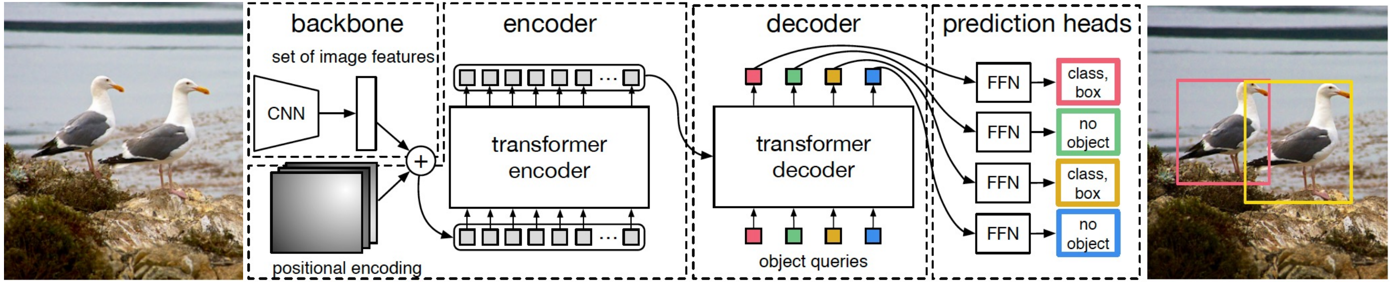
[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]

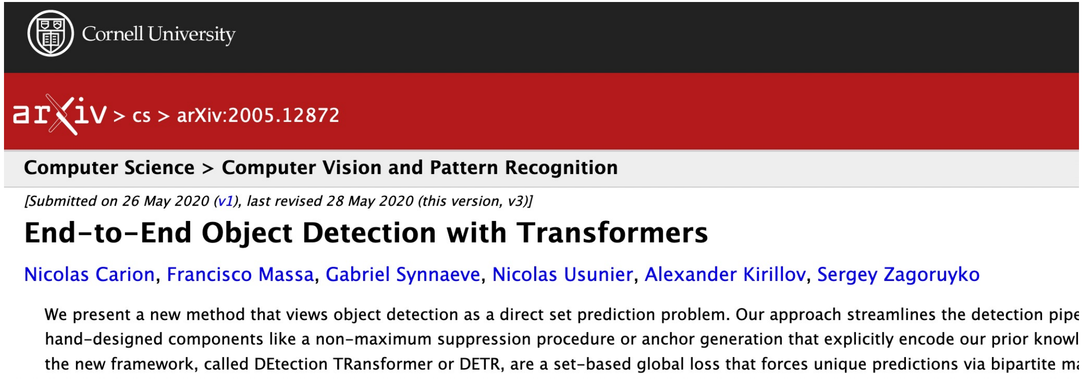


Output query array / Output array defines the downstream task: detection

Transformer Decoder for detection

Just another scheme for DETR model





Cornell University

arXiv > cs > arXiv:2005.12872

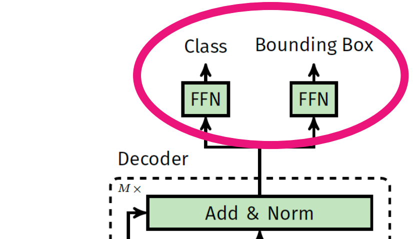
Computer Science > Computer Vision and Pattern Recognition

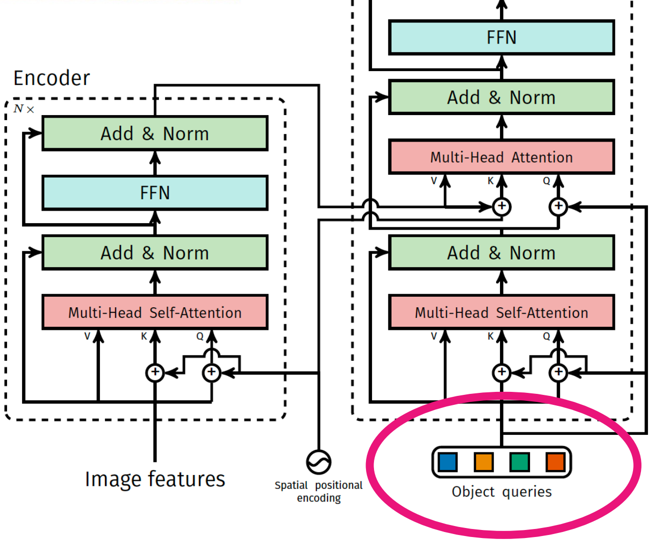
[Submitted on 26 May 2020 (v1), last revised 28 May 2020 (this version, v3)]

End-to-End Object Detection with Transformers

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko

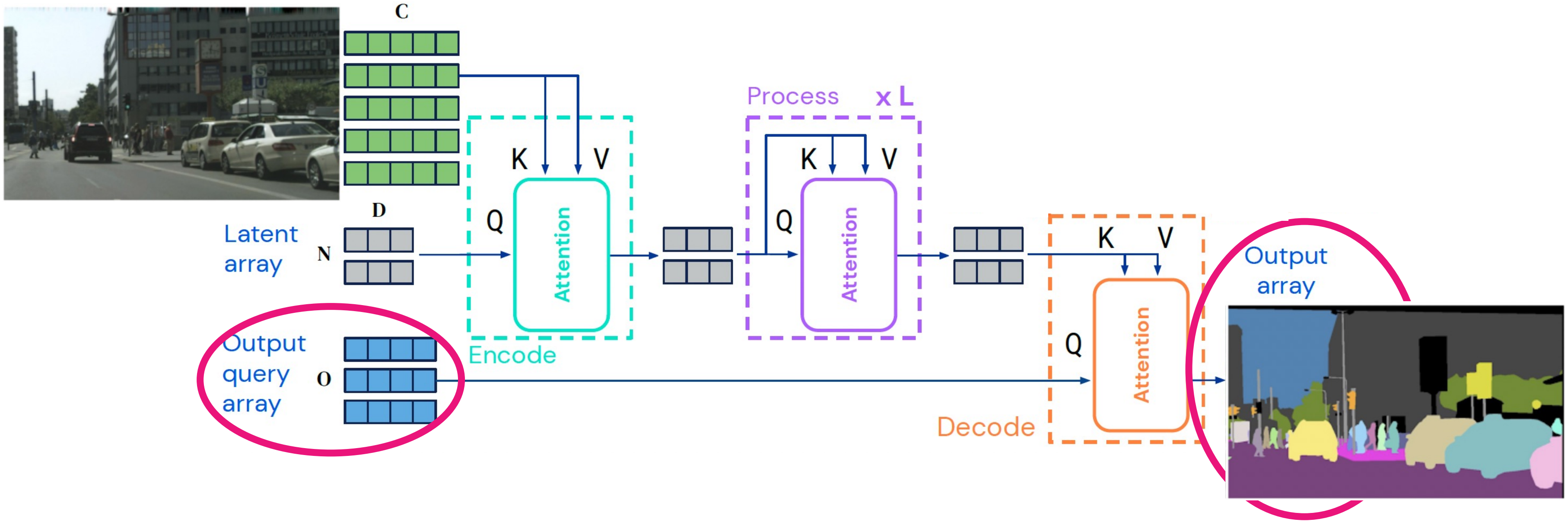
We present a new method that views object detection as a direct set prediction problem. Our approach streamlines the detection pipe
hand-designed components like a non-maximum suppression procedure or anchor generation that explicitly encode our prior knowl
the new framework, called DEtection TRansformer or DETR, are a set-based global loss that forces unique predictions via bipartite ma





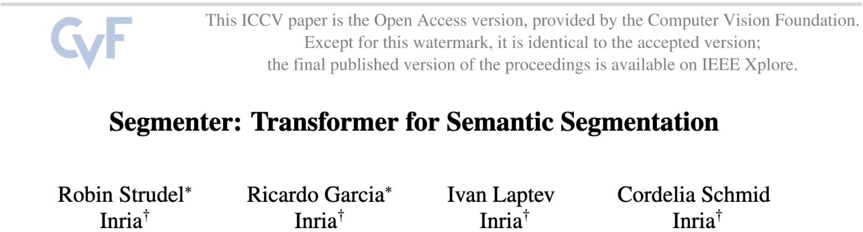
General Decoder

[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]



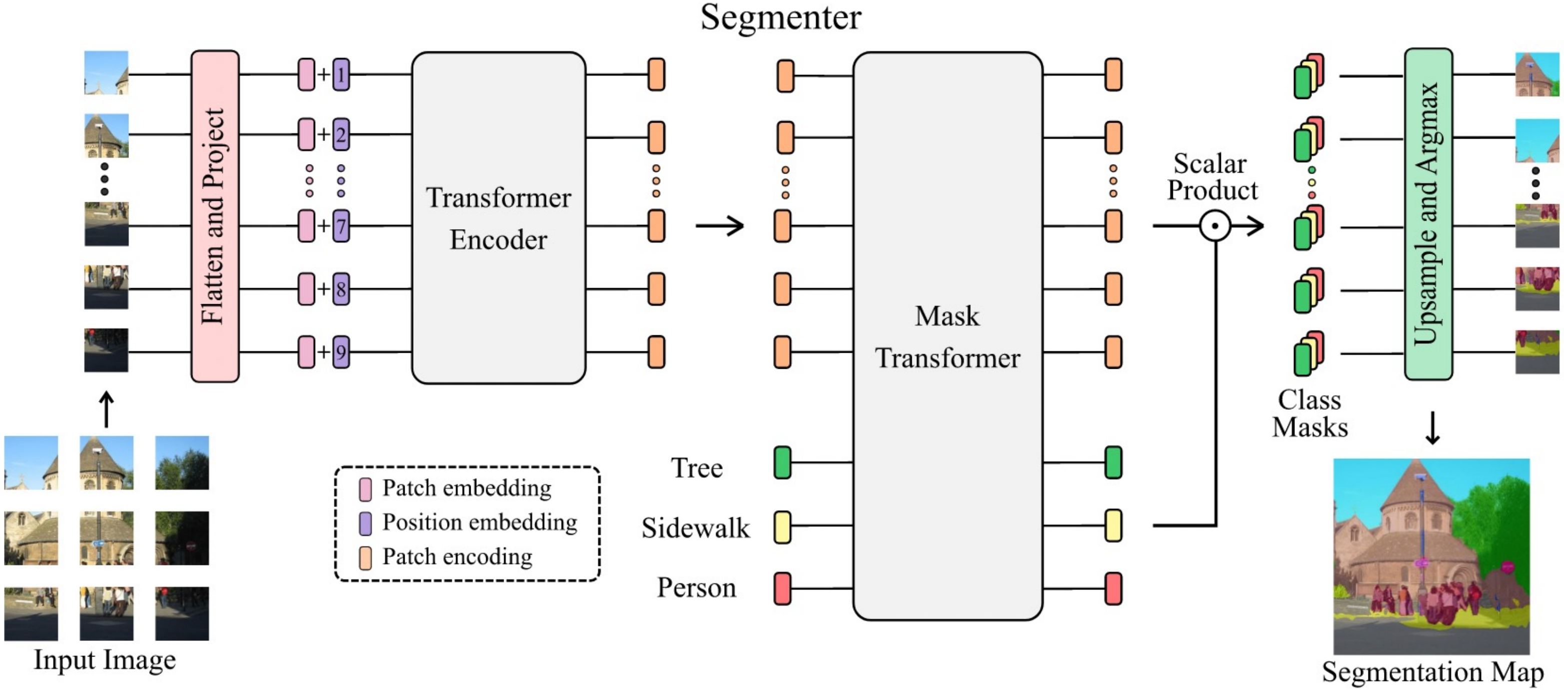
Output query array / Output array defines the downstream task: segmentation …

General Decoder: or not!



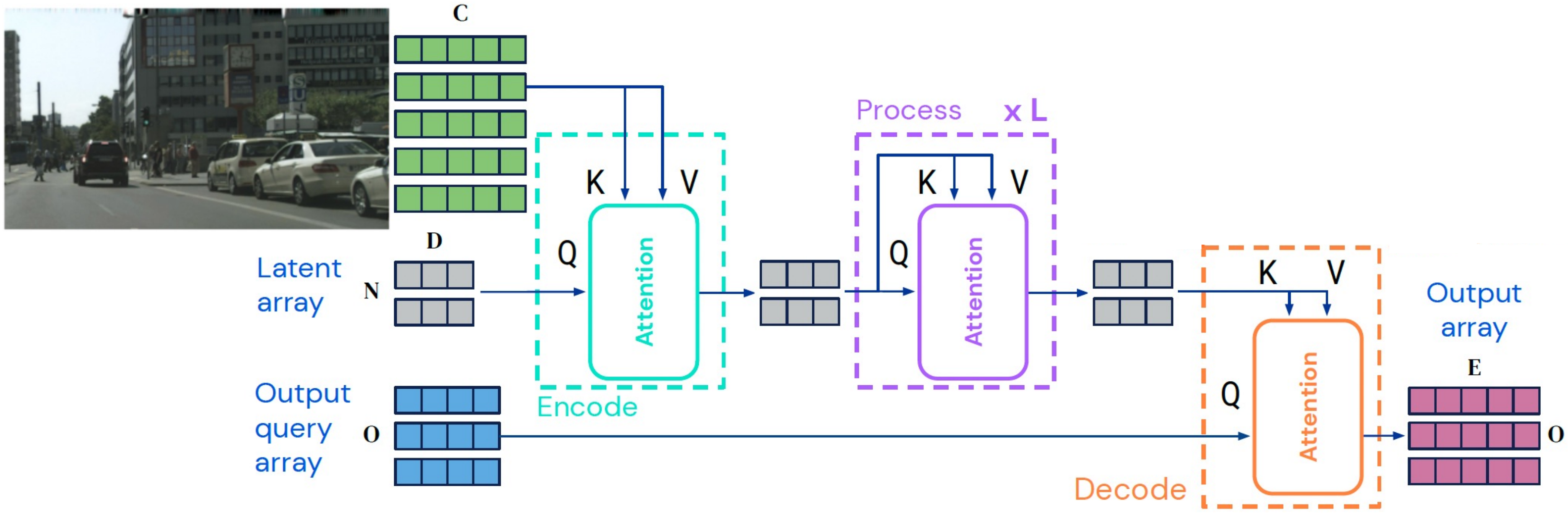
Segmenter: Transformer for Semantic Segmentation

Robin Strudel[*] Inria[†]
Ricardo Garcia[*] Inria[†]
Ivan Laptev Inria[†]
Cordelia Schmid Inria[†]



General Decoder

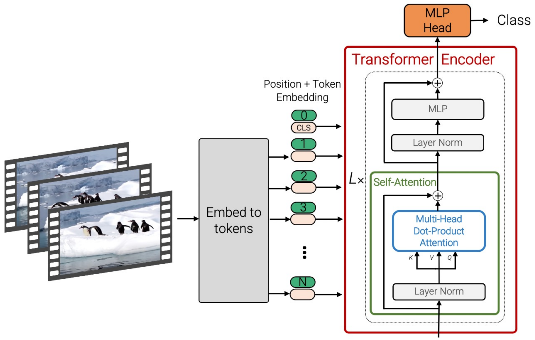
[Perceiver IO A General Architecture for Structured Inputs & Outputs ICLR22]

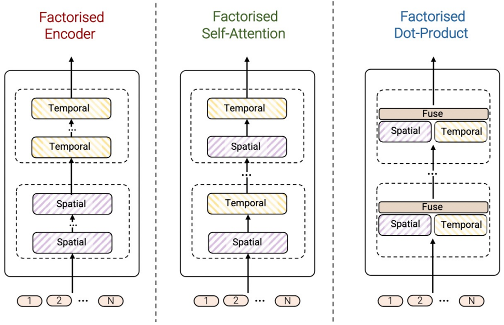


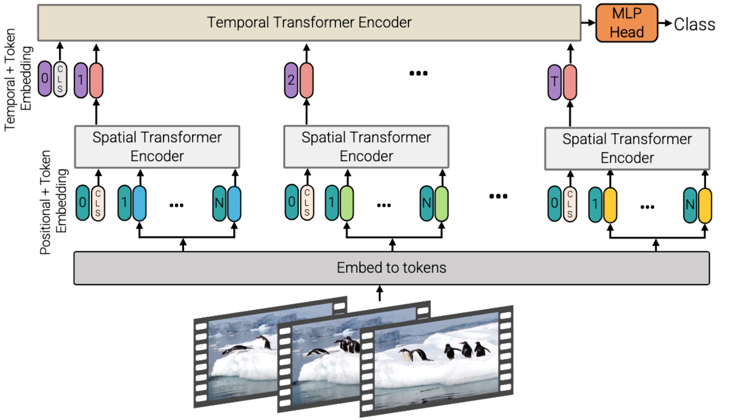
Output query array / Output array defines the downstream task: continual learning

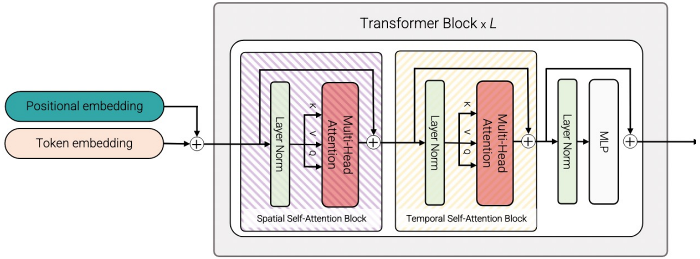
Video Transformer

[ViViT: A Video Vision Transformer ICCV 2021]



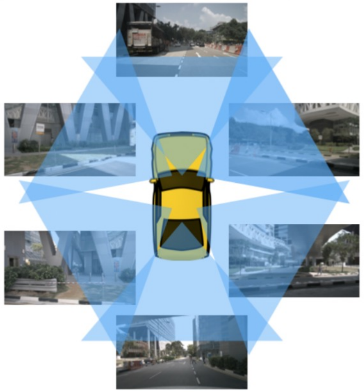



Temporal Transformer Encoder
MLP Head
Class
Temporal + Token Embedding
0
CLS
1
2
T
Spatial Transformer Encoder
Spatial Transformer Encoder
Spatial Transformer Encoder
Positional + Token Embedding
0
CLS
1
N
Embed to tokens



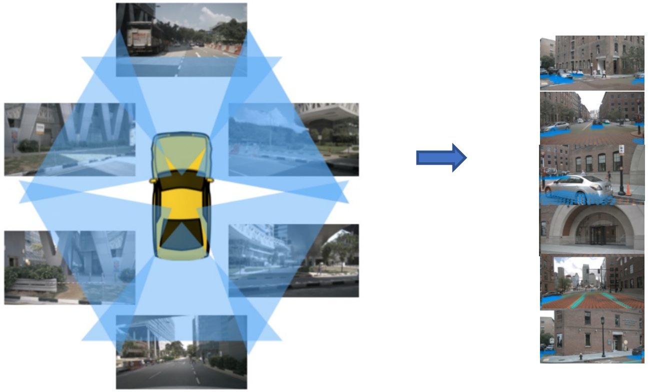
General Encoder / Decoder

Input array = N cameras



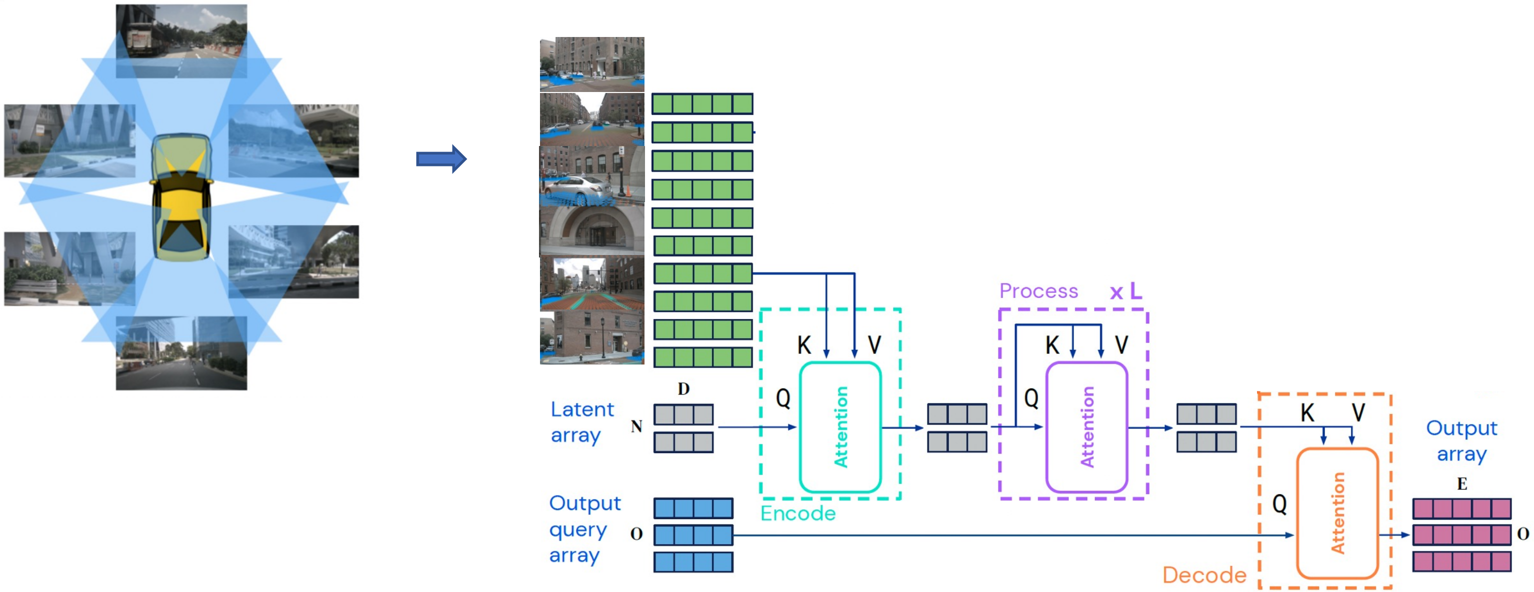
General Encoder / Decoder

Input array = N cameras



General Encoder / Decoder

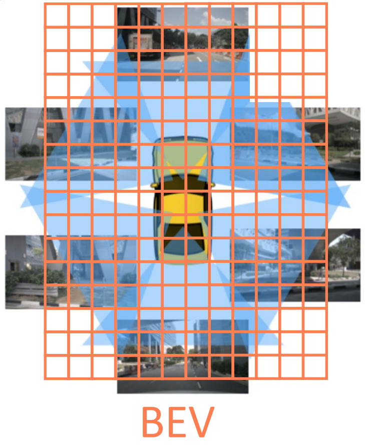
Input array = N cameras

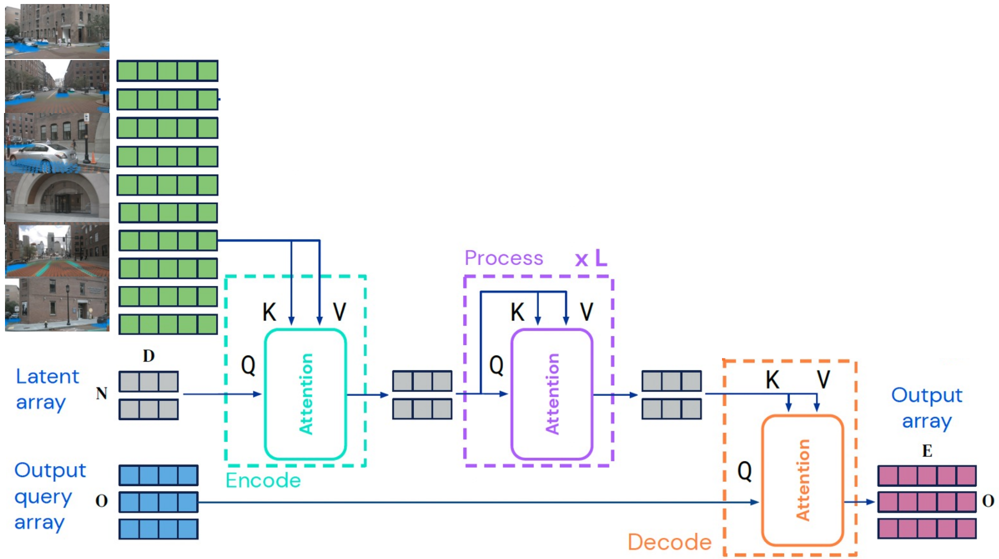


General Encoder / Decoder

Input array = N cameras

Output array = Bird Eye View (BEV) representation

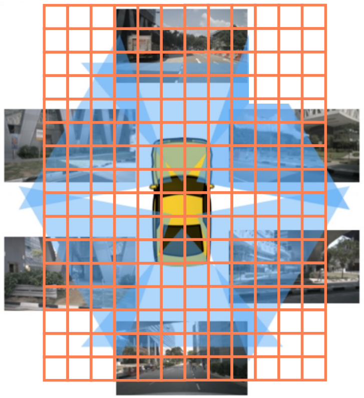




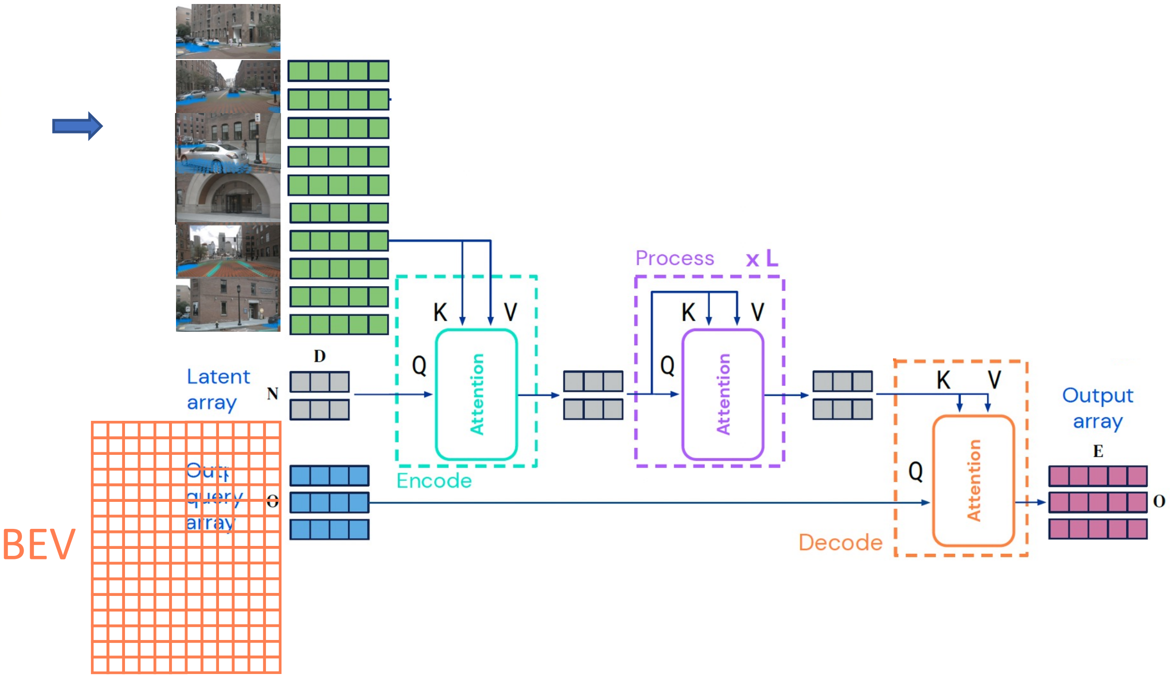
General Encoder / Decoder

Input array = N cameras

Output array = Bird Eye View (BEV) representation



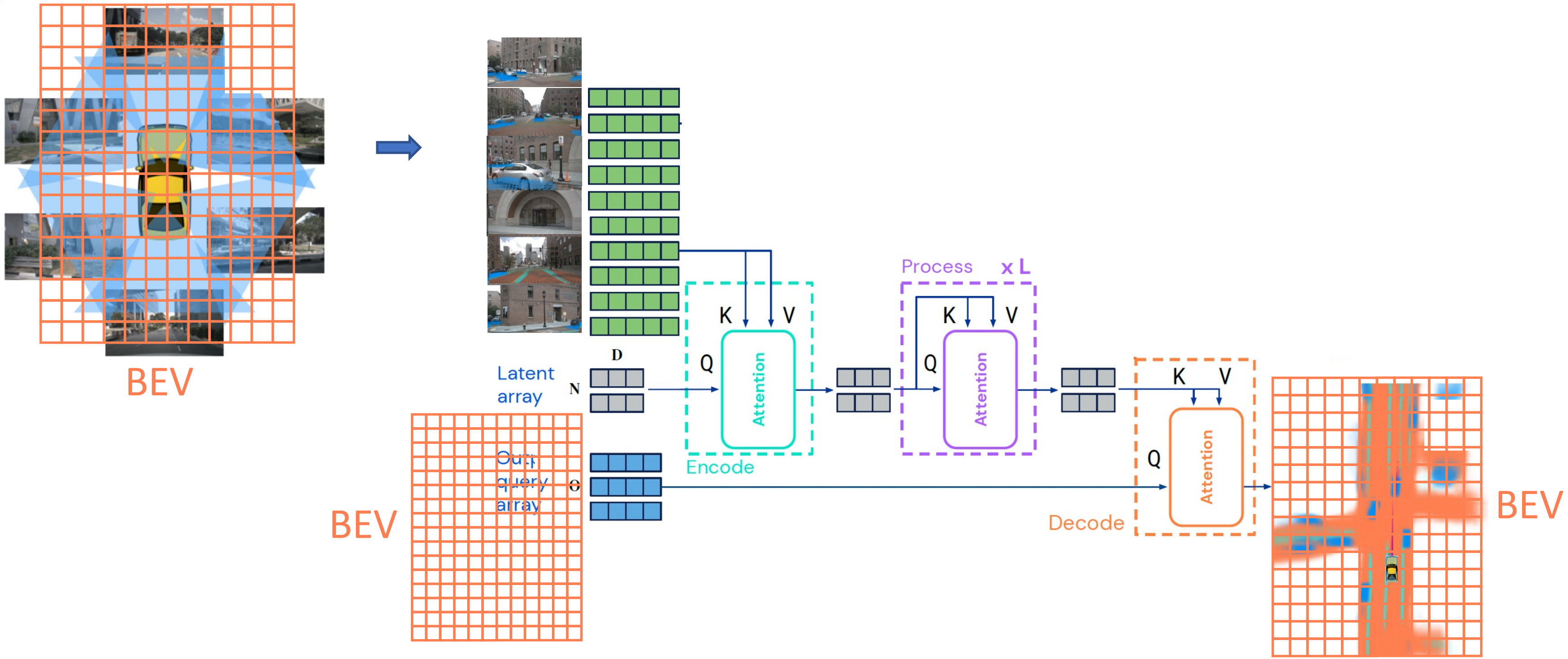
BEV



General Encoder / Decoder

Input array = N cameras

Output array = Bird Eye View (BEV) representation



Vision Transformers

Global Attention mechanism at every layer of the deep archi

Very **competitive architectures** in image classification with the best Convnets

Fusion/Merging by mixing thanks to cross attention process

Somehow universal deep structure around encoding/decoding for many vision tasks as classification (1 class token), object detection, segmentation, …

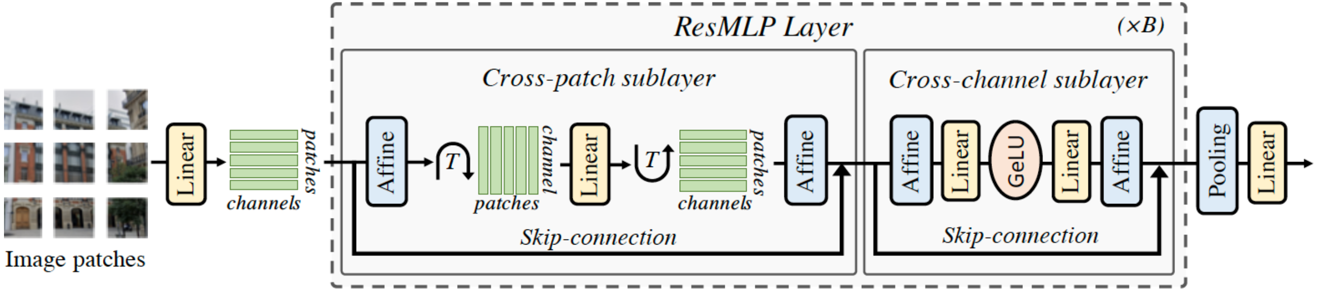


Image patches
Linear
channels
patches
ResMLP Layer
(×B)
Cross-patch sublayer
Affine
T
channel
patches
Linear
T
channels
patches
Affine
Skip-connection
Cross-channel sublayer
Affine
Linear
GeLU
Linear
Affine
Skip-connection
Pooling
Linear