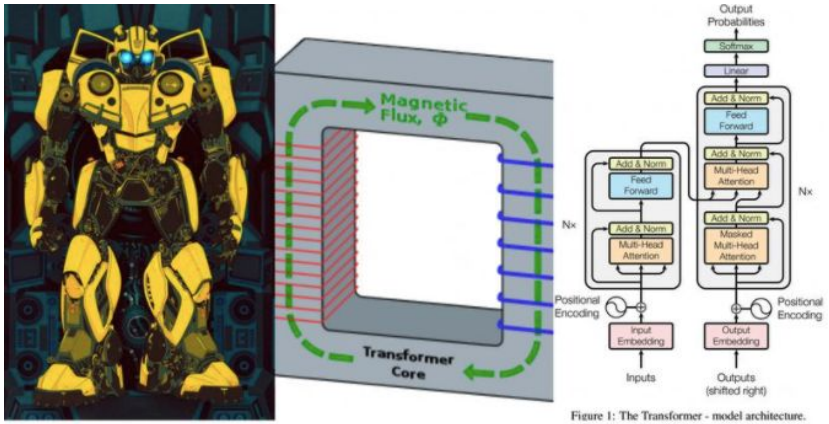
Transformers

Attention is all you need



Javier Selva Castelló

Image Source: https://twitter.com/_sam_sinha_/status/1369821141913767945
Original meme by: Messal Monem Dipto

Outline

- Motivation
  - No more recurrence
  - Attention Mechanisms
- The transformer
  - Overview
  - Queries, Keys and Values
  - Dot-product attention
- Vision Transformers
  - Pre-training
  - Efficient Designs
- Application
  - Image Transformers
  - Video Transformers

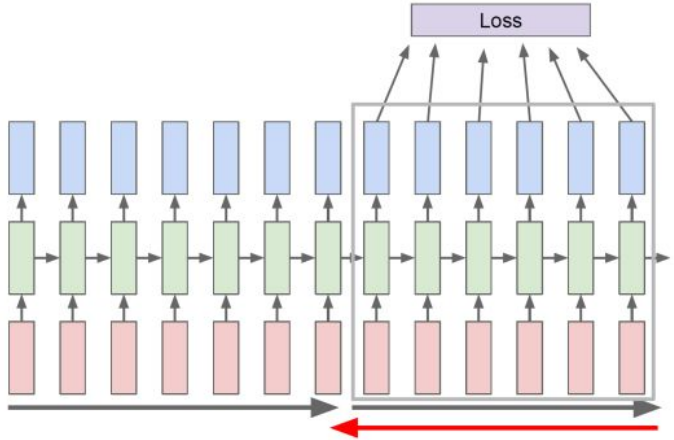
Motivation

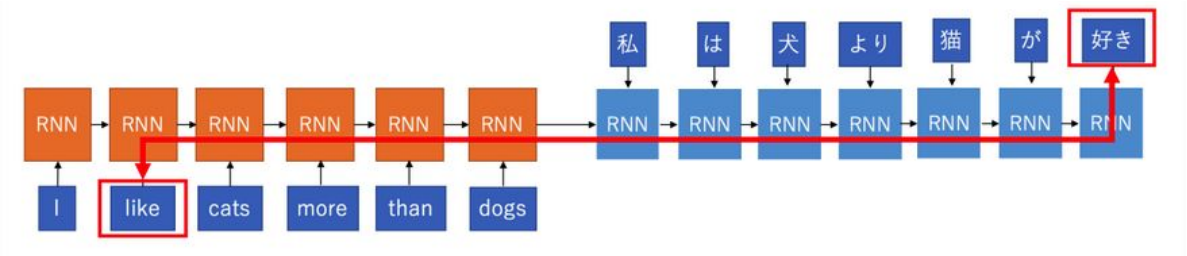
No more recurrence

Recurrent nature does not allow for **parallelism**.

Relating **distant elements** of the input may be tricky:

- #Operations grows with sequence length
- Truncated Backpropagation
- Vanishing gradients if full BPTT

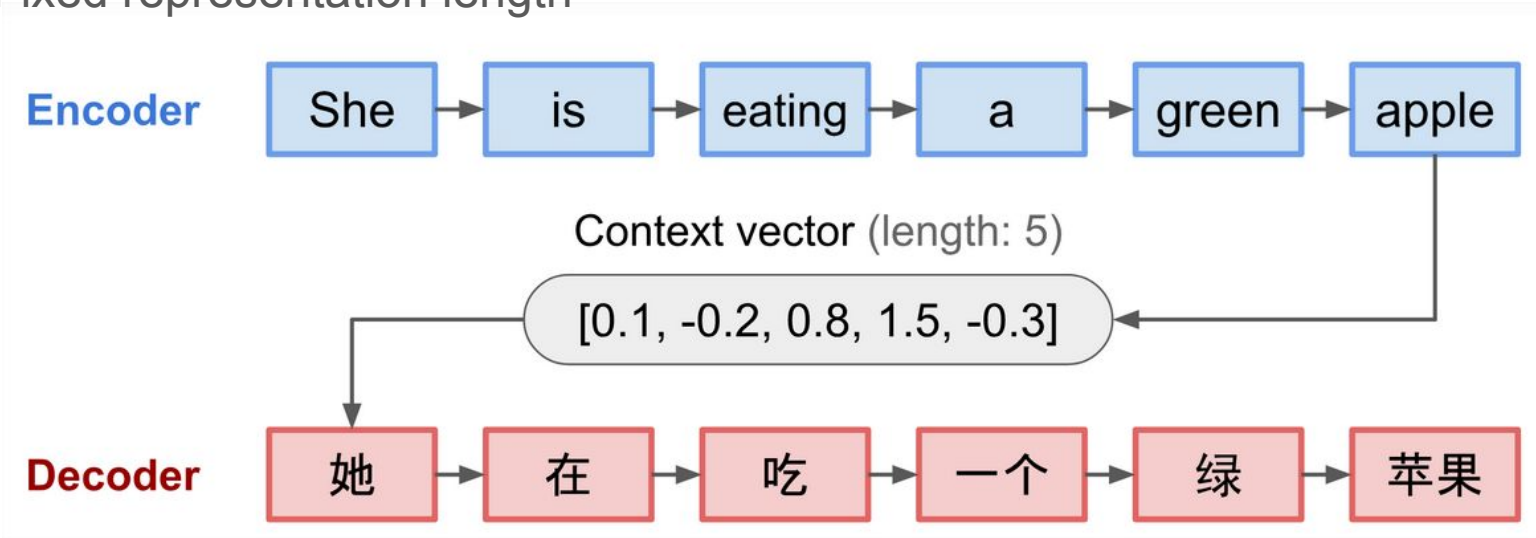




Source: https://medium.com/@b.terryjack/deep-learning-the-transformer-9ae5e9c5a190

Sequence to sequence (seq2seq)

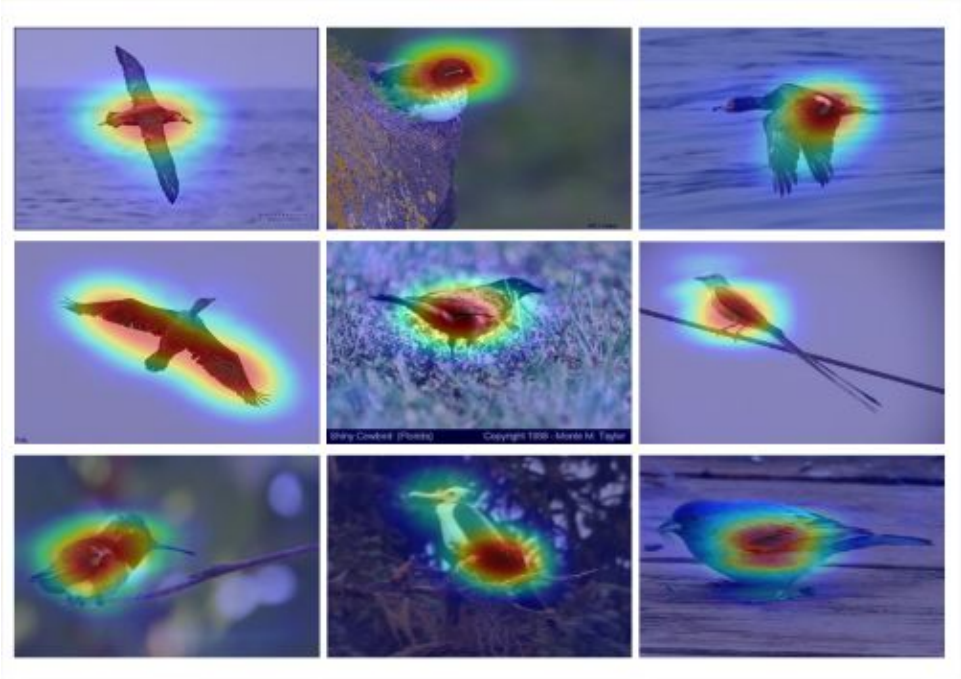
- Input a sequence, output another sequence (e.g. translation)
- Fixed representation length



Source: https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

Attention Mechanisms

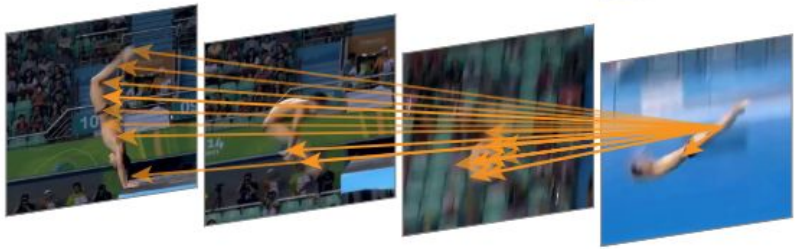
- Select important regions of an input.

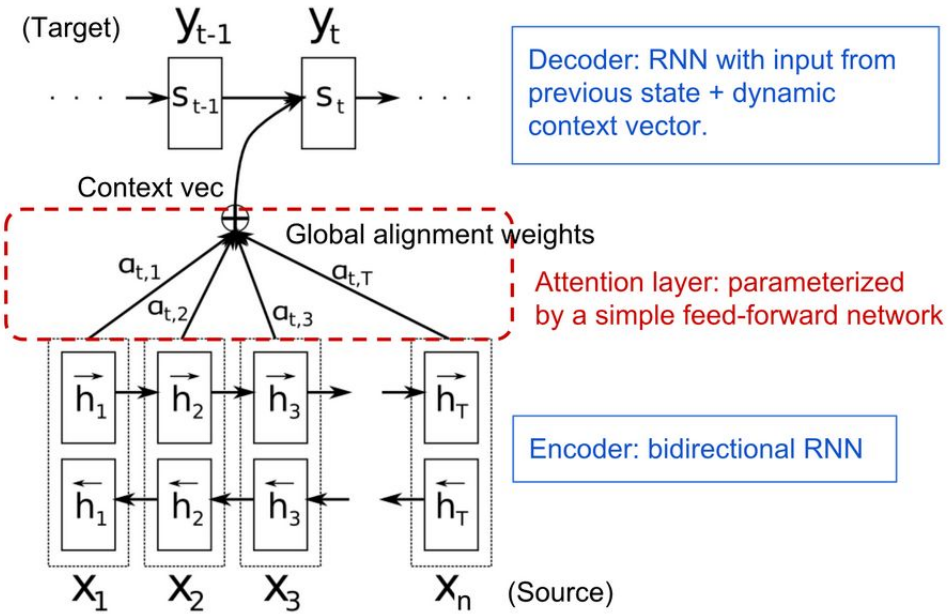


Source: **He, X.** et al. "*Fine-grained discriminative localization via saliency-guided faster R-CNN*". In the 25th ACM ICM (2017).

Attention Mechanisms

- Select important elements of a sequence.





$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i \quad ; \text{Context vector for output } y_t$$

$$\alpha_{t,i} = \text{align}(y_t, x_i) \quad ; \text{How well two words } y_t \text{ and } x_i \text{ are aligned.}$$

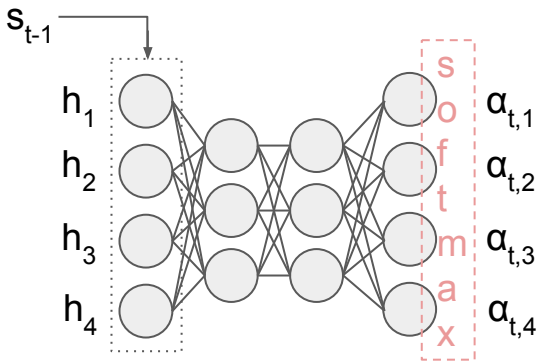
$$= \frac{\exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_i))}{\sum_{i'=1}^{n} \exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_{i'}))} \quad ; \text{Softmax of some predefined alignment score..}$$

Source: https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

[1] **Bahdanau, D.** et al. "*Neural machine translation by jointly learning to align and translate*". In CoRR (2014).

Attention Mechanisms

Fully connected layer → **Uses learned weights**.



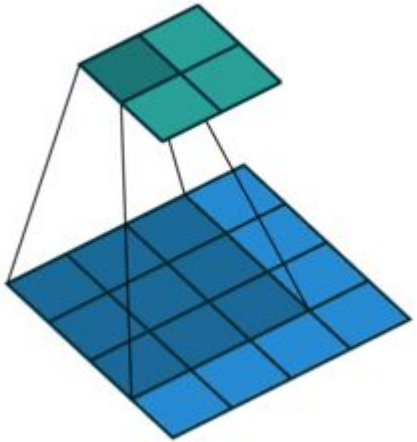
$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i$$

$$\alpha_{t,i} = \text{align}(y_t, x_i) = \frac{\exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_i))}{\sum_{i'=1}^{n} \exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_{i'}))}$$

Source: http://nlp.seas.harvard.edu/2018/04/03/attention.html

[2] **Wang, X.** et al. *"Non-local neural networks."* In CVPR (2018).

Local vs. Non-local



Local (Convolution)

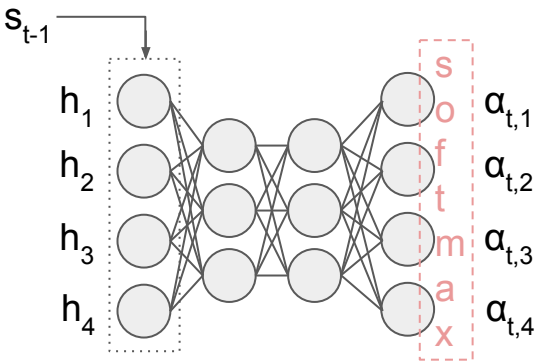
Non-local

Source: https://github.com/vdumoulin/conv_arithmetic

[2] **Wang, X.** et al. "*Non-local neural networks.*" In CVPR (2018).

Attention Mechanisms

Fully connected layer → **Uses learned weights**.



$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i$$

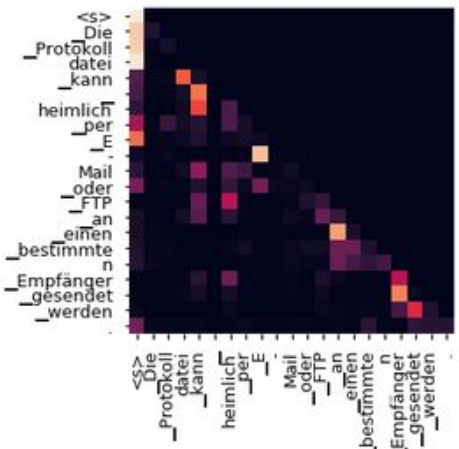
$$\alpha_{t,i} = \text{align}(y_t, x_i) = \frac{\exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_i))}{\sum_{i'=1}^{n} \exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_{i'}))}$$

Source: http://nlp.seas.harvard.edu/2018/04/03/attention.html

[2] **Wang, X.** et al. *"Non-local neural networks."* In CVPR (2018).

Non-local layer → Attention based on **relationships between input elements**

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j).$$



f → Computes the affinity

g → Computes representation of the input

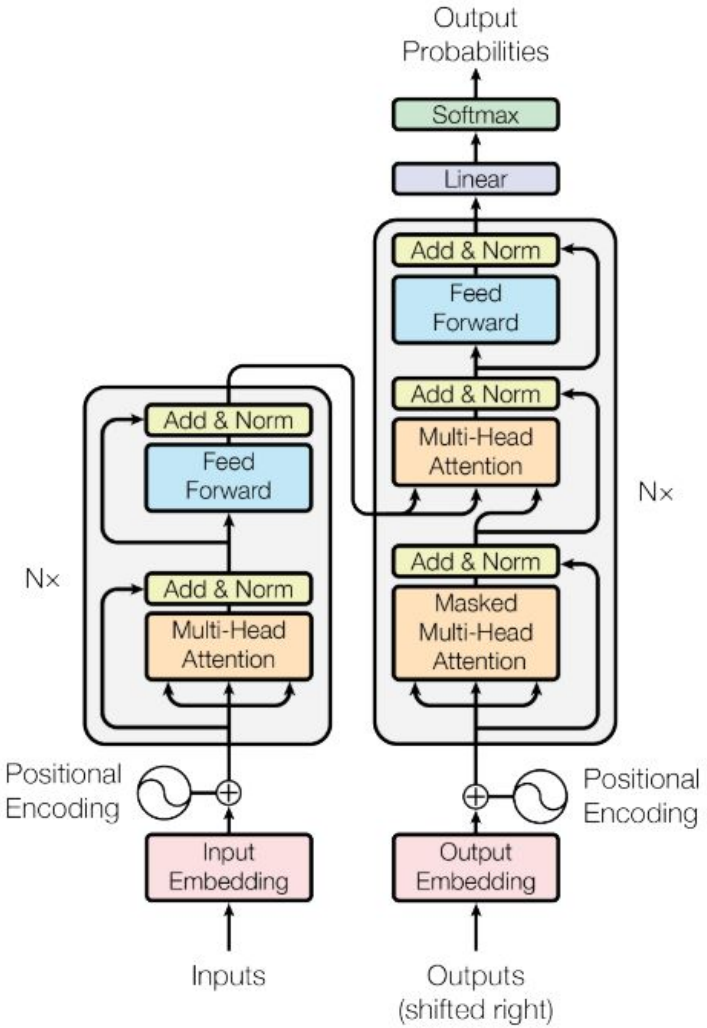
C(x) → Normalization factor

The Transfomer

The transformer

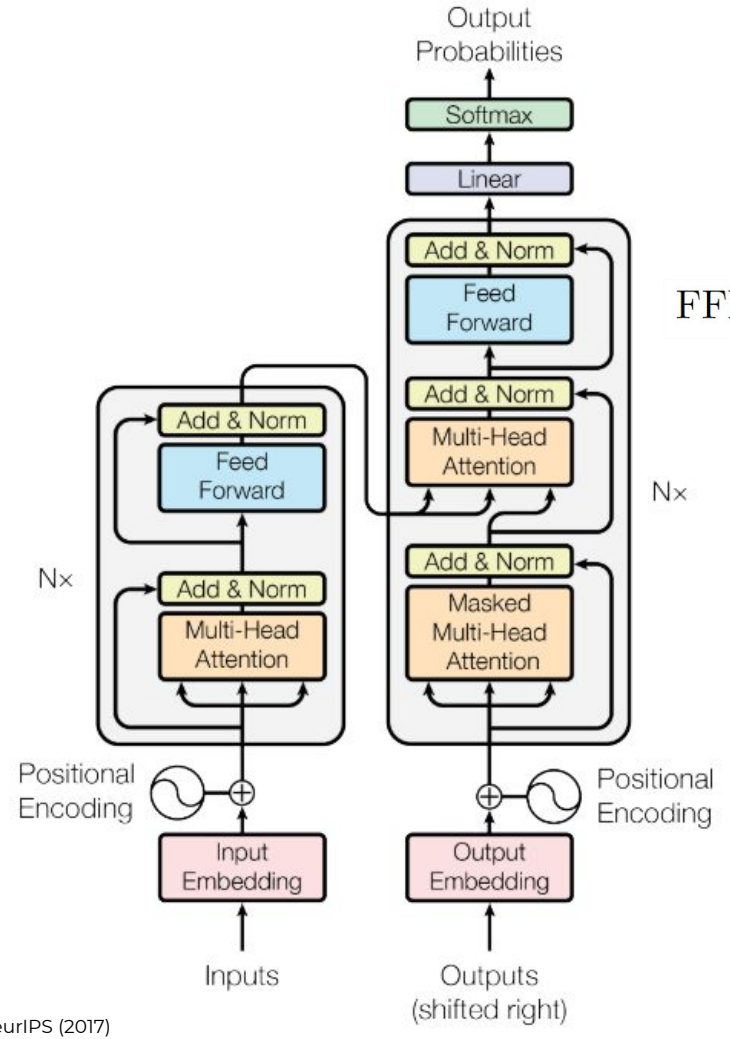
- Input whole sequences (tokenized)
- Add positional information

- Self-attention
- Linear transformation
- Residuals and Normalization



Source: [3] **Vaswani, A.** et al. "*Attention is all you need*". In NeurIPS (2017)

The transformer

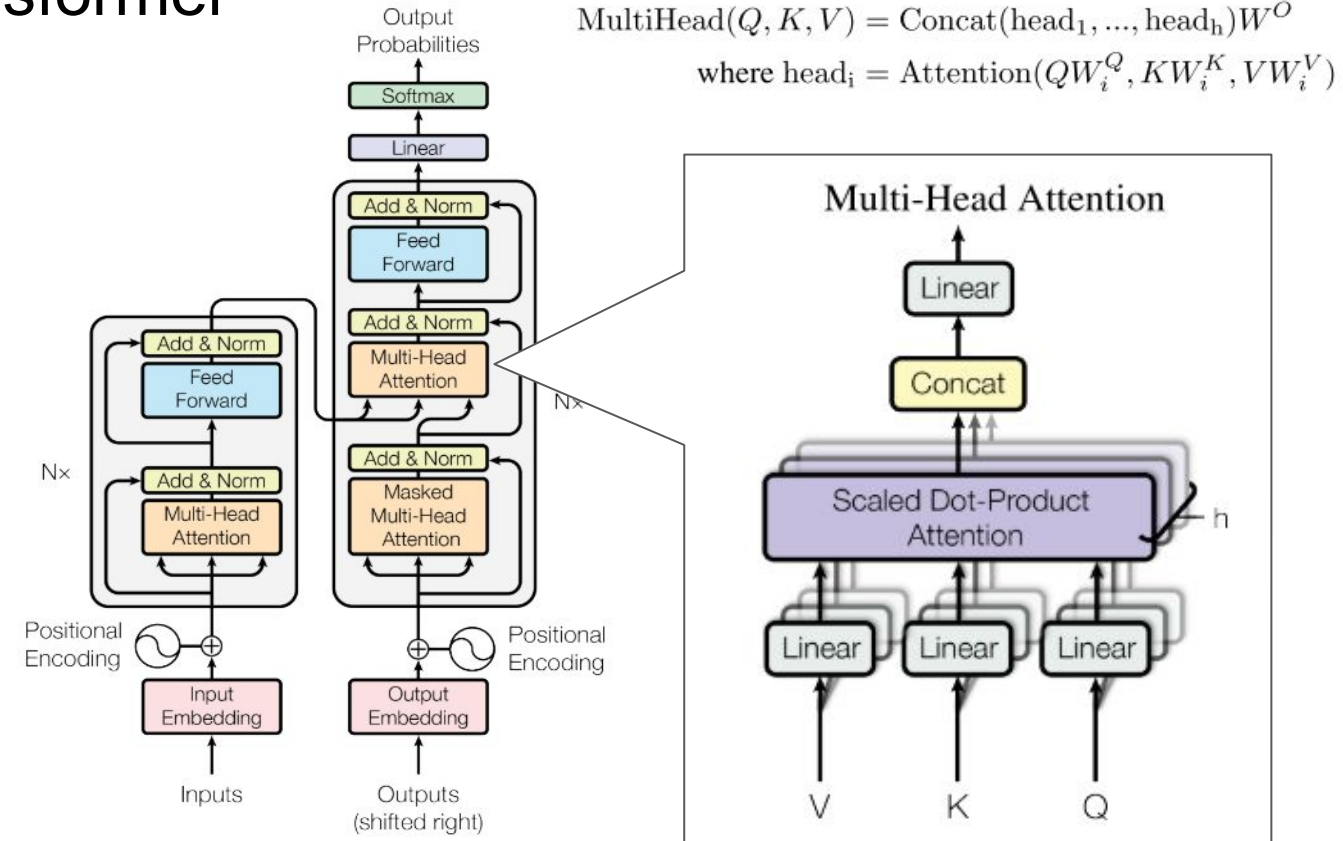


$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Source: [3] **Vaswani, A.** et al. "*Attention is all you need*". In NeurIPS (2017)

The transformer

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

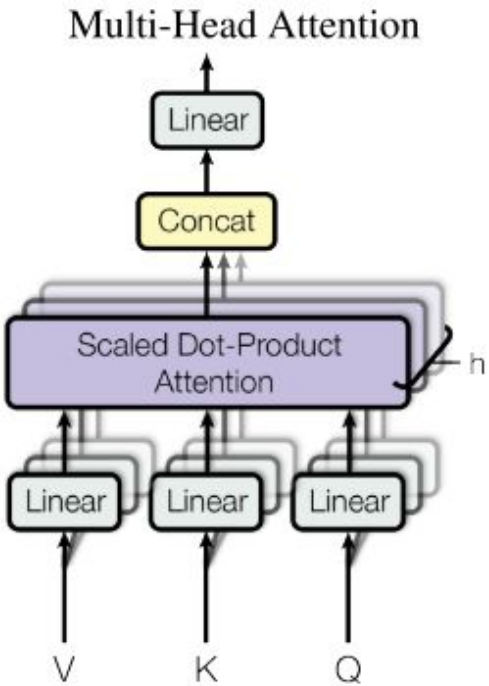


Source: [3] **Vaswani, A.** et al. "*Attention is all you need*". In NeurIPS (2017)

Multi-head Self-attention

- Map into different representation spaces (similar to convolutional filters)
- Attend at different features of the same regions
- Concatenate
- Map back to common representation space

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

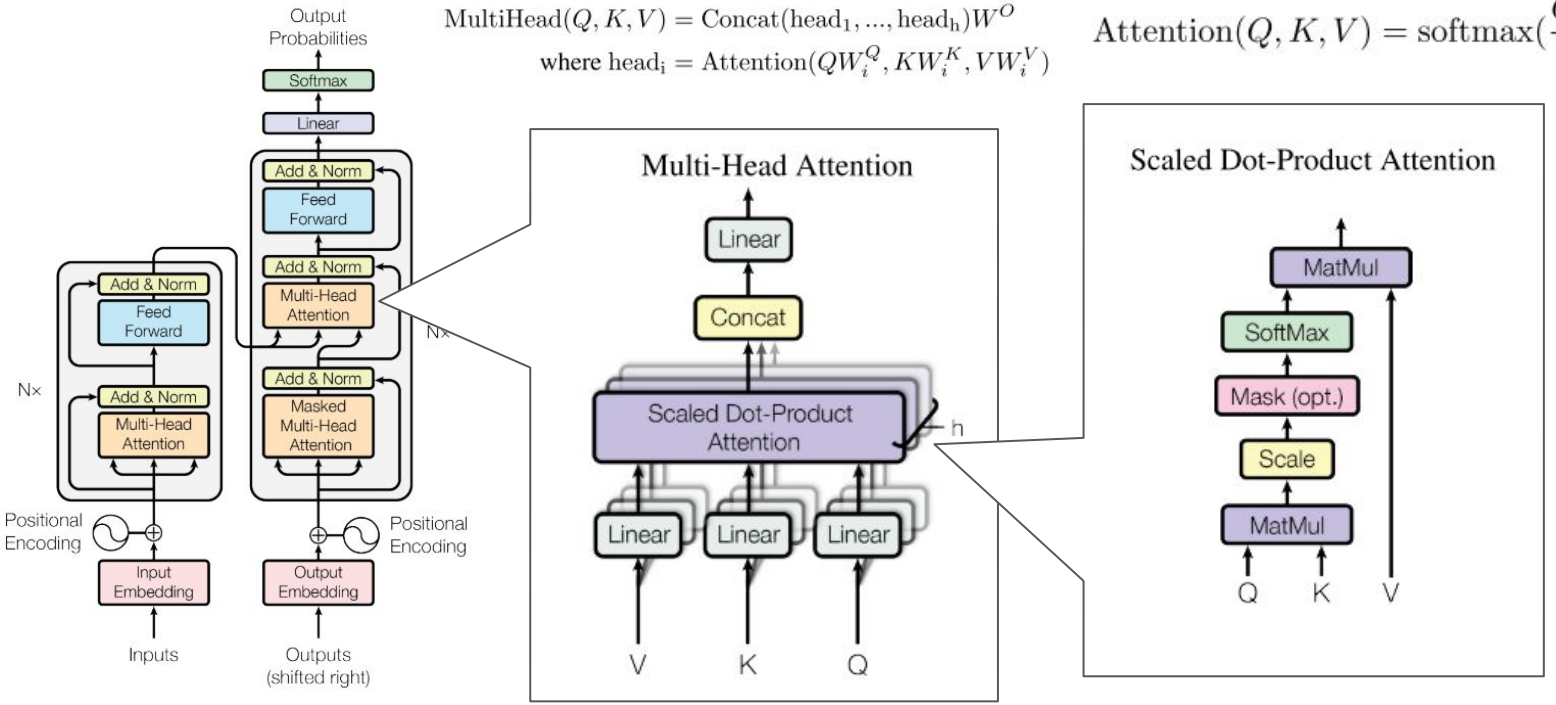


Source: [3] **Vaswani, A.** et al. "*Attention is all you need*". In NeurIPS (2017)

The transformer

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,...,\text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

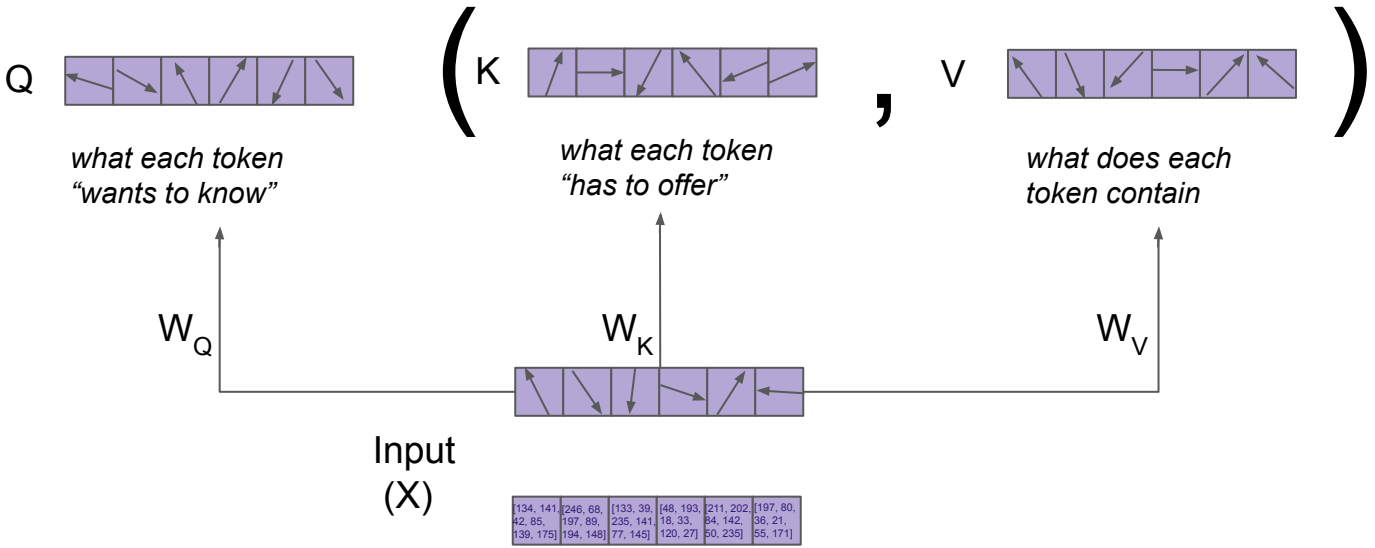
$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Source: [3] **Vaswani, A.** et al. "*Attention is all you need*". In NeurIPS (2017)

Queries, Keys and Values

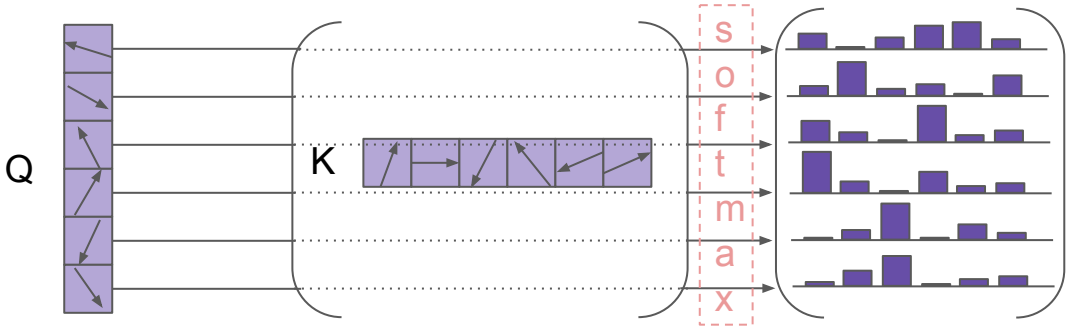
- Non-local: Mechanism used to **route information** from every token to any other
- Use tokens as **weighted contextual** information
- Set of **Queries**, and set of **pairs of Keys and Values**



Queries, Keys and Values

- Each **Query** is ***compared*** **with all Keys**
- Produce "***histogram***" of **Query alignment with the Keys**

$$a_{ij} = \mathrm{softmax}(\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_k}})$$

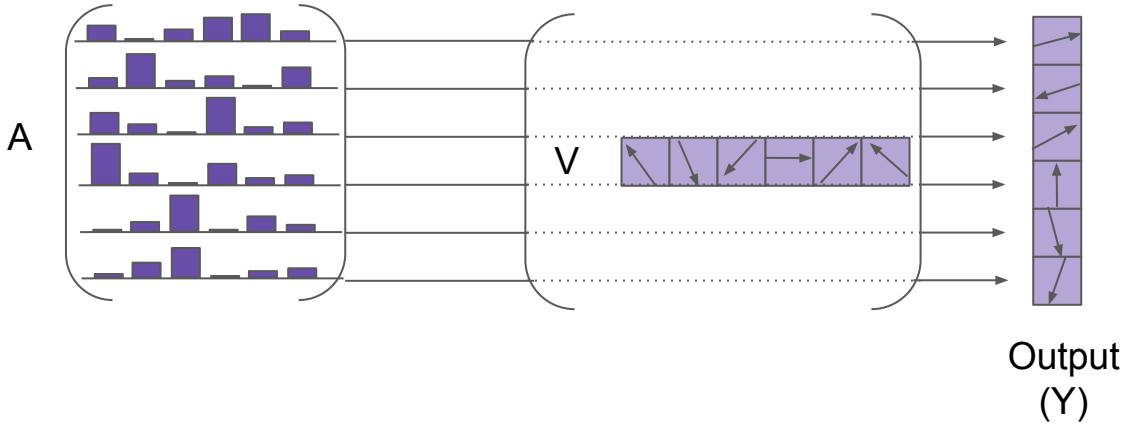


(A)ttention matrix

Queries, Keys and Values

- **Weighted summation**
- Each output token is the result of **contextual aggregation**

$$y_i = \sum_{j=0}^{T} v_i a_{ij}$$

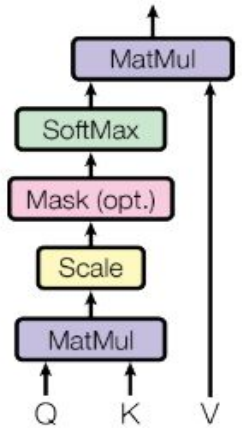


Scaled Dot-product

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

- The dot-product of two linearly independent vectors is 0.
- Dot-product is used to compute similarity among vectors.
- Softmax to turn it into an attention map - through probabilities (sum=1)

Scaled Dot-Product Attention



$$a_{ij} = \text{softmax}(\frac{\mathbf{q}_i {\mathbf{k}_j}^\top}{\sqrt{d_k}}) = \frac{\exp(\mathbf{q}_i {\mathbf{k}_j}^\top)}{\sqrt{d_k} \sum_{r \in S_i} \exp(\mathbf{q}_i {\mathbf{k}_r}^\top)}$$

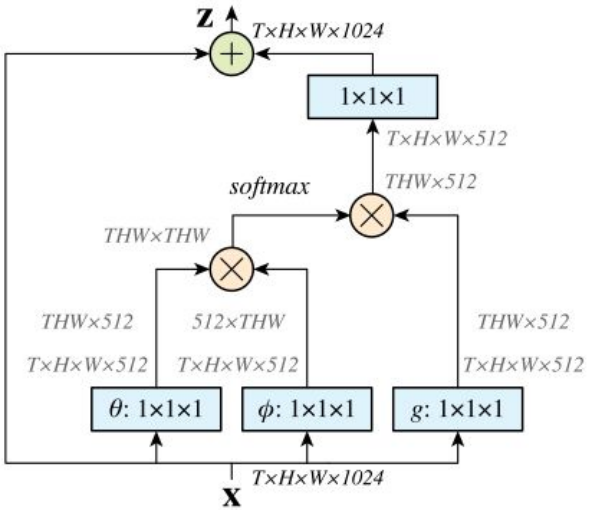
$$y_i = \sum_{j=0}^{T} v_i a_{ij}$$

Source: [3] **Vaswani, A.** et al. "*Attention is all you need*". In NeurIPS (2017)

Dot-product as non-local

- The dot-product is an instantiation of the non-local operation

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j).$$



$$a_{ij} = \text{softmax}(\frac{\mathbf{q}_i {\mathbf{k}_j}^\top}{\sqrt{d_k}}) = \frac{\exp(\mathbf{q}_i {\mathbf{k}_j}^\top)}{\sqrt{d_k} \sum_{r \in S_i} \exp(\mathbf{q}_i {\mathbf{k}_r}^\top)}$$

$$y_i = \sum_{j=0}^{T} v_i a_{ij}$$

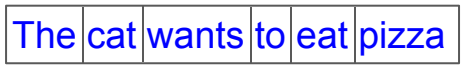
Source: [2,3]

Overview of the transformer

- Full **example** of original NLP Transformer for **machine translation**
- Main **difference with vision** transformers is **before the input** to the Transformer layers:
  - Tokenization
  - Embedding
  - Positional Encodings

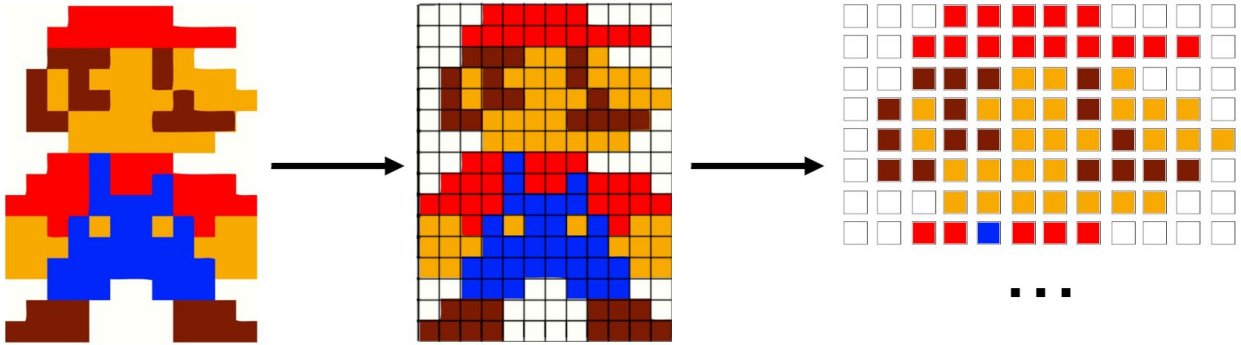
The cat wants to eat pizza

Overview of the transformer



TOKENIZATION

Tokenization (images)



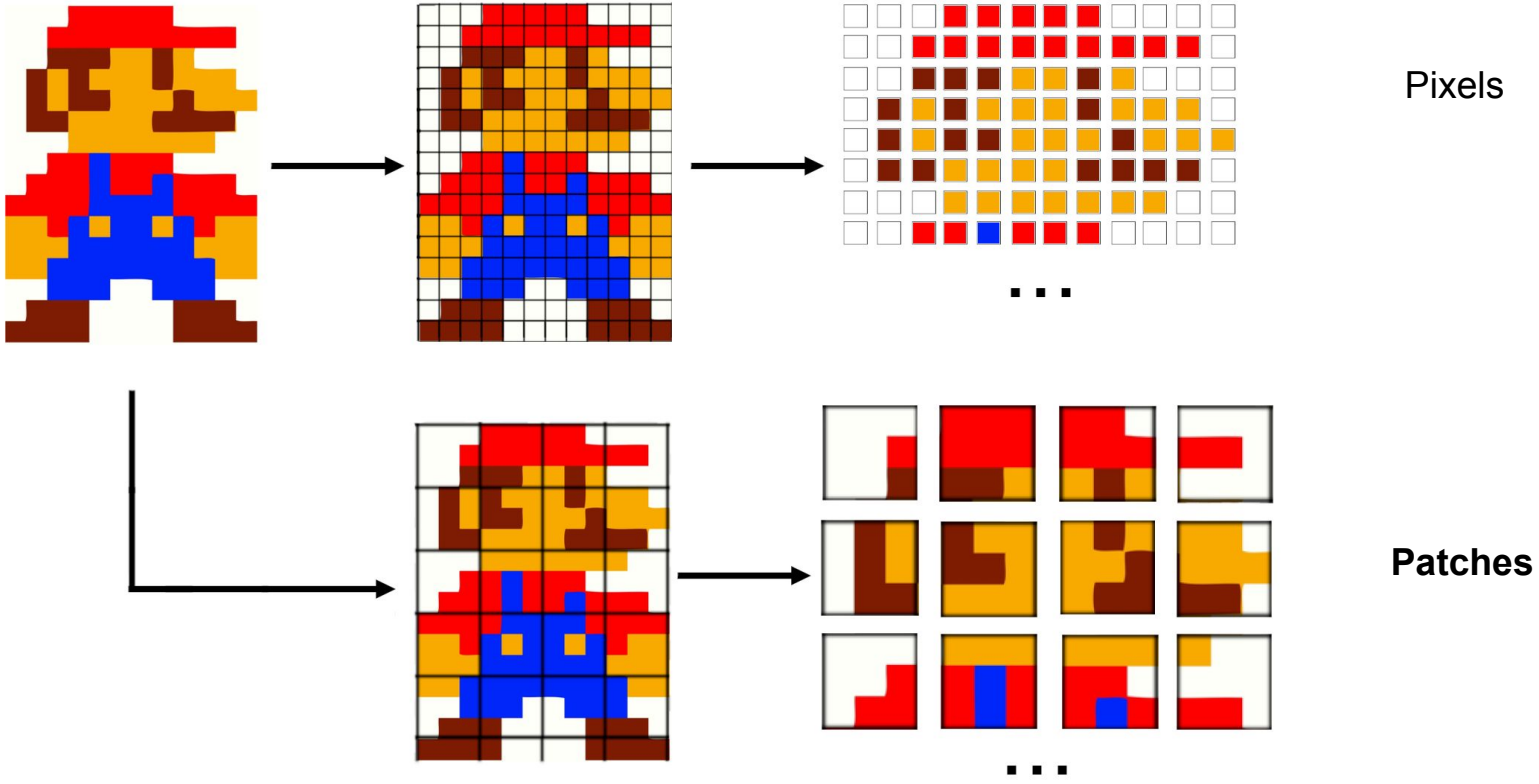
Pixels

But attention matrix grows **quadratically** with **input sequence length**!! → $O(n^2)$

A 256x256 image would have = 65.536 px

Attention matrix would have **65.536^2 elements**!!

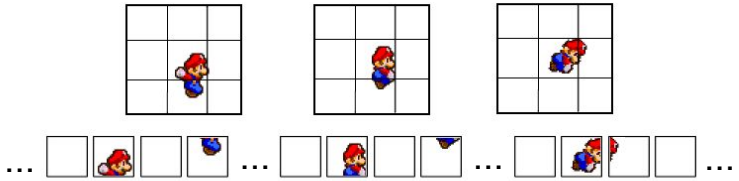
Tokenization (images)



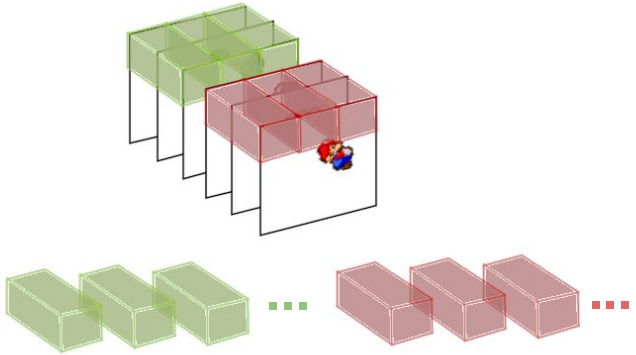
Tokenization (videos)

Patches

2D



3D

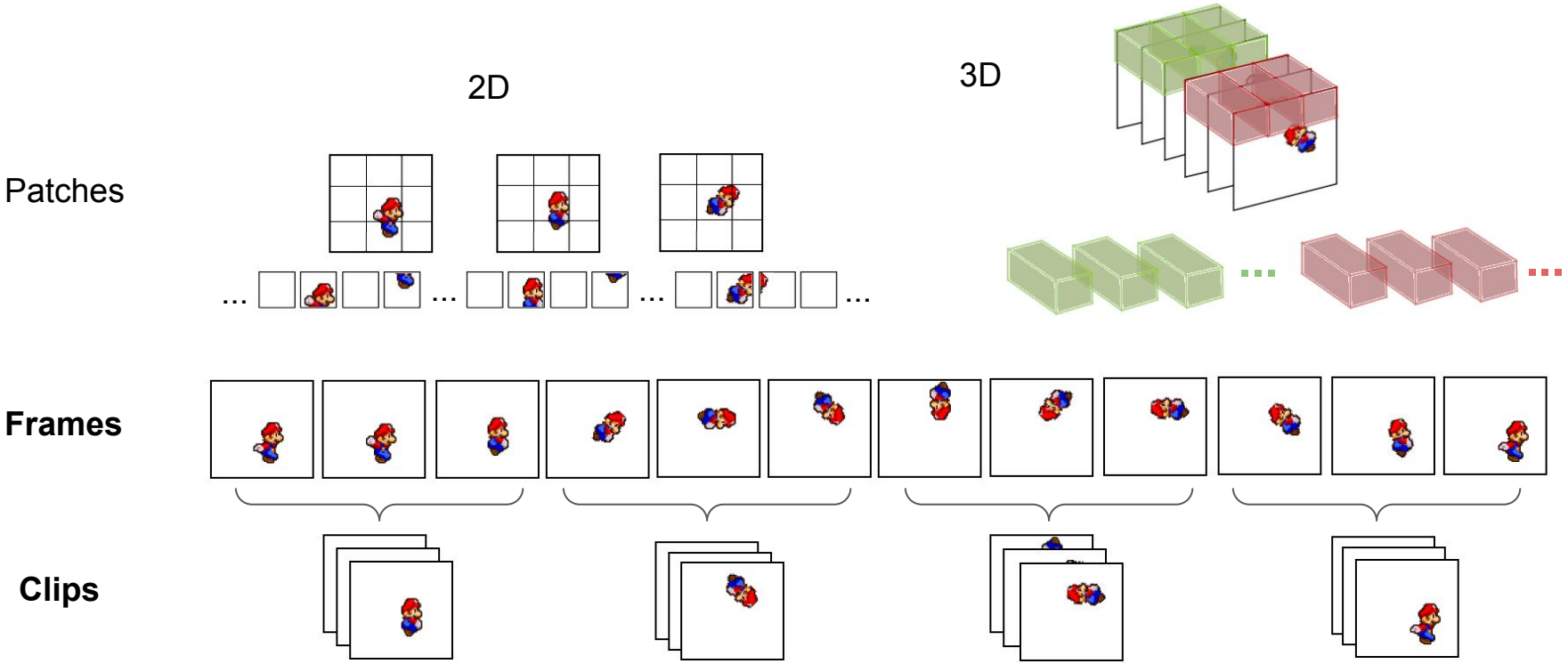


A 256x256 image tokenized as 16x16 patches, would have = 256 tokens per frame

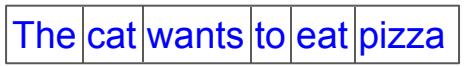
15 second video at 25 fps → Attention matrix would have **96.000^2 elements!!**

3D patch setting (16x16x16) alleviates, but still does not scale that well → **A has 6000^2**)

Tokenization (videos)

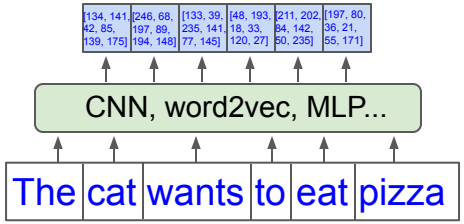


Overview of the transformer



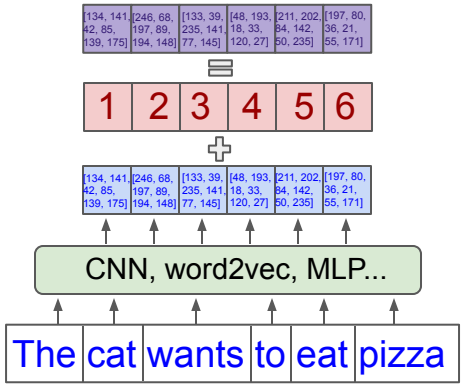
TOKENIZATION

Overview of the transformer



EMBEDDING

Overview of the transformer



POSITIONAL ENCODING

Positional Encodings

Transformers are not invariant to translation! No idea about position!!

0 :	0 0 0 0	8 :	1 0 0 0
1 :	0 0 0 1	9 :	1 0 0 1
2 :	0 0 1 0	10 :	1 0 1 0
3 :	0 0 1 1	11 :	1 0 1 1
4 :	0 1 0 0	12 :	1 1 0 0
5 :	0 1 0 1	13 :	1 1 0 1
6 :	0 1 1 0	14 :	1 1 1 0
7 :	0 1 1 1	15 :	1 1 1 1

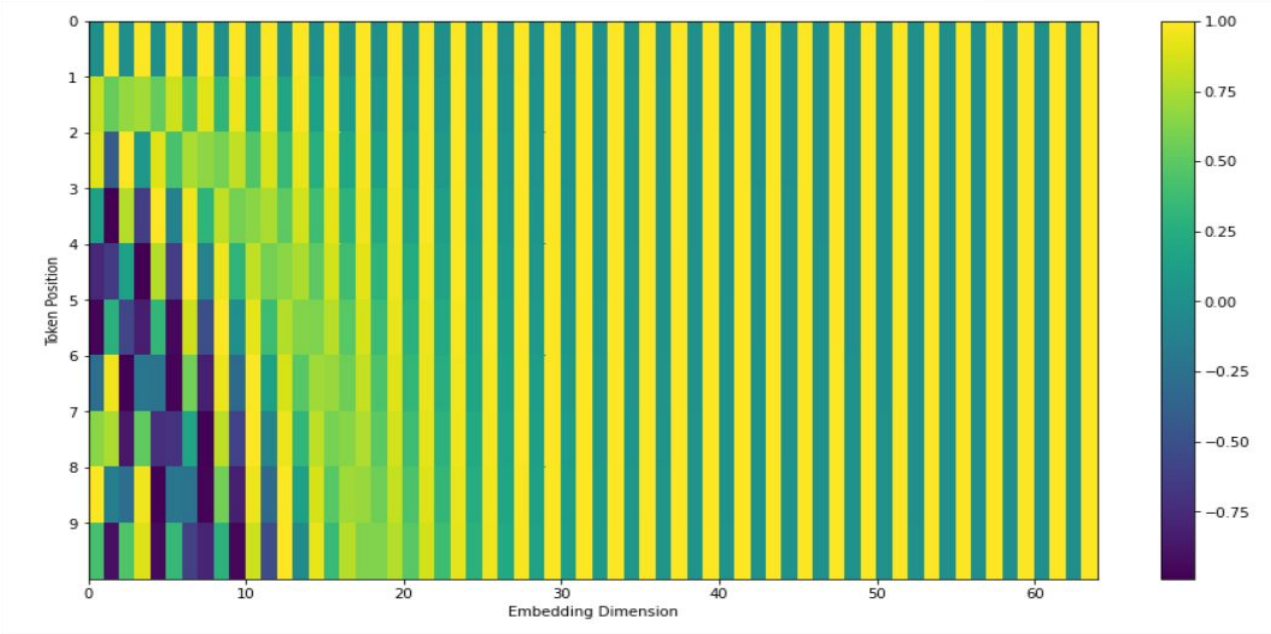
Binary

Sinusoidal

Sources: https://kazemnejad.com/blog/transformer_architecture_positional_encoding/
http://nlp.seas.harvard.edu/2018/04/03/attention.html

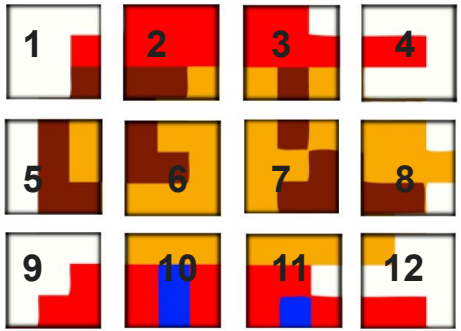
Positional Encodings

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

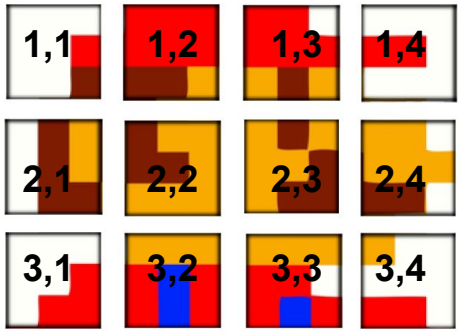


Sources: https://lilianweng.github.io/lil-log/2020/04/07/the-transformer-family.html
https://jalammar.github.io/illustrated-transformer/

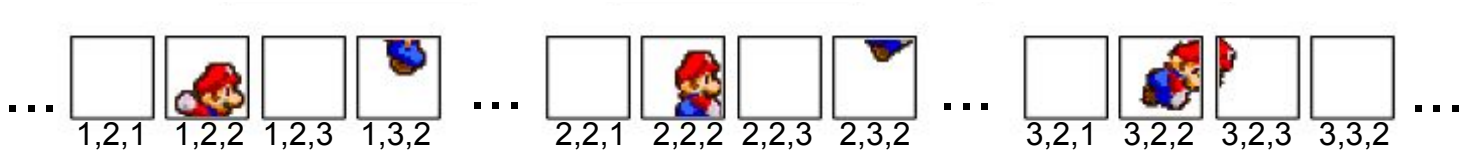
Positional Encodings



1D (Raster order)

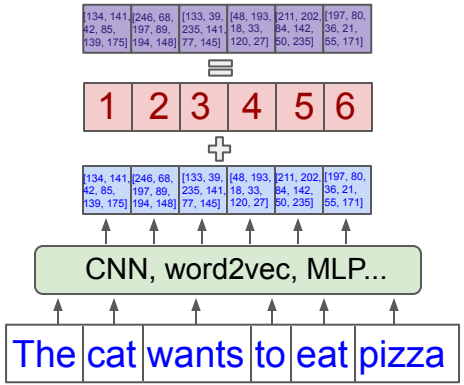


2D (Coordinates)



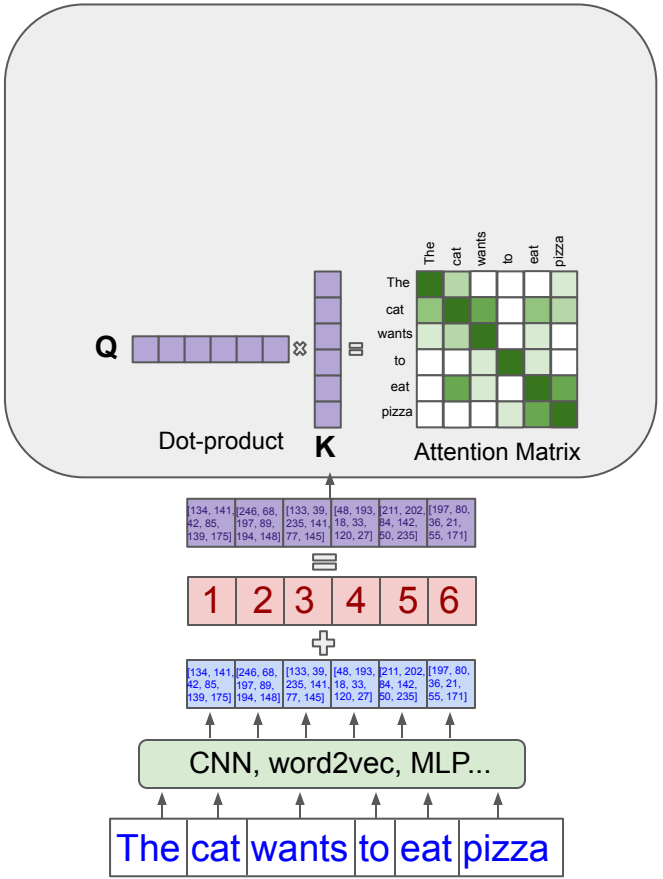
3D
(Coordinates for video patches)

Overview of the transformer



POSITIONAL ENCODING

Overview of the transformer

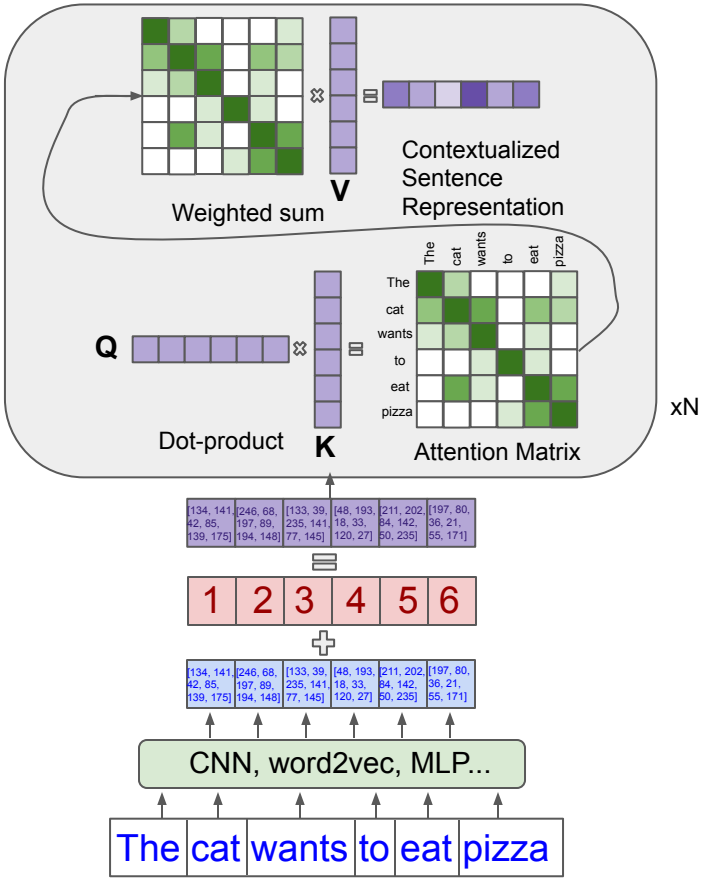


ENCODING (Self-Attention)

Computing Affinity

$$a_{ij} = \text{softmax}(\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_k}}) = \frac{\exp(\mathbf{q}_i \mathbf{k}_j^\top)}{\sqrt{d_k} \sum_{r \in S_i} \exp(\mathbf{q}_i \mathbf{k}_r^\top)}$$

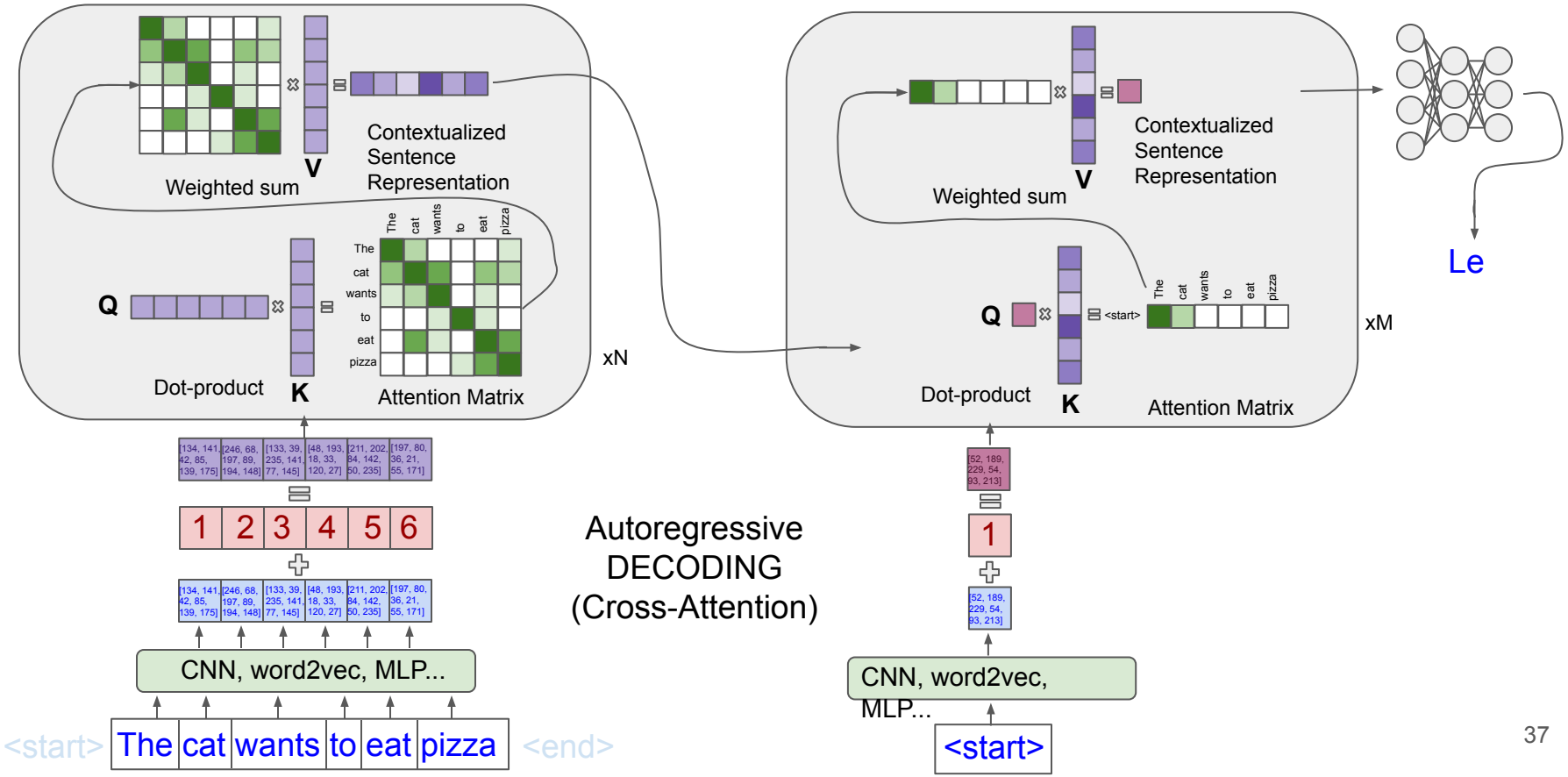
Overview of the transformer



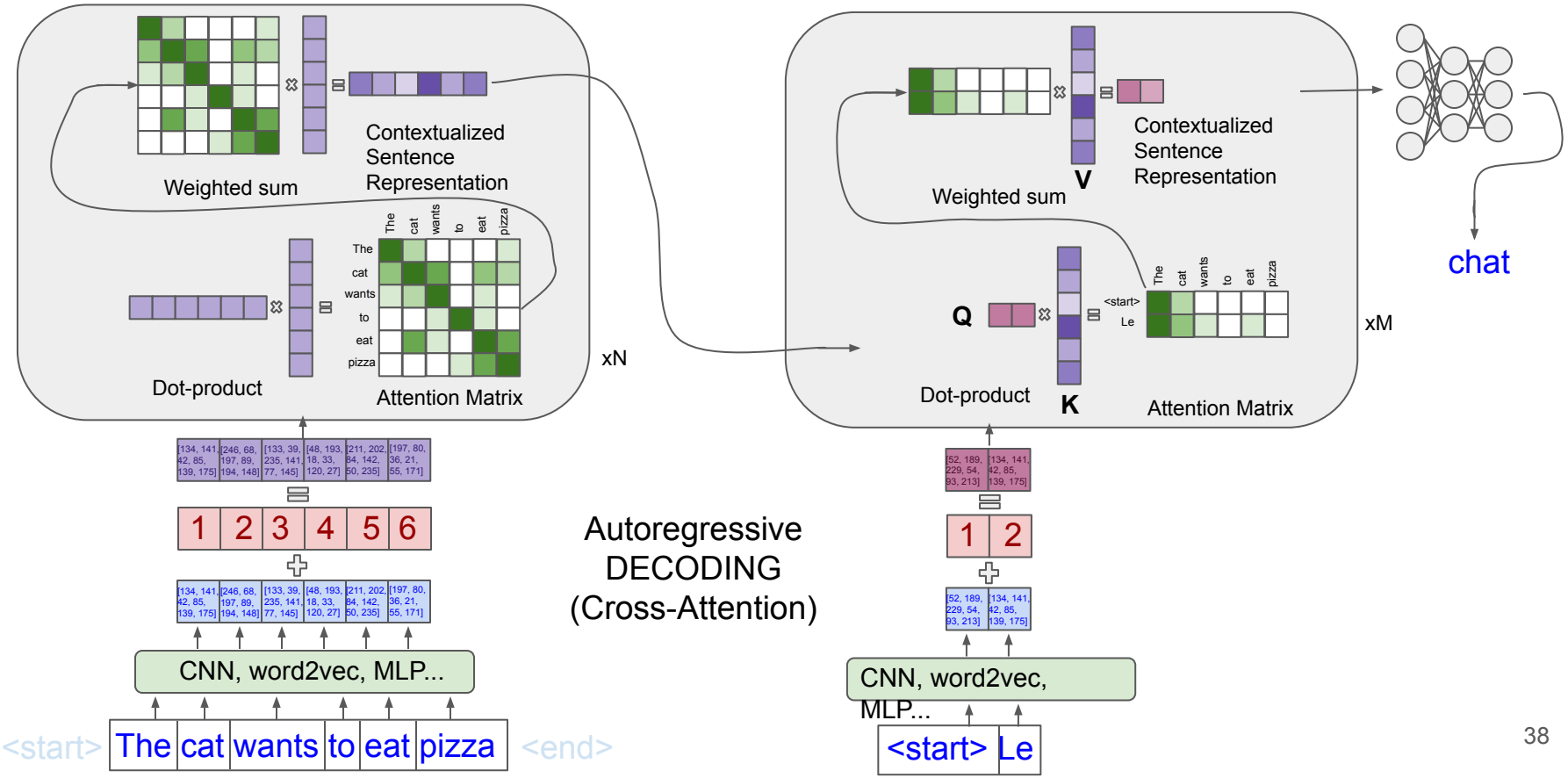
Aggregation $y_i = \sum_{j=0}^{T} v_i a_{ij}$

ENCODING (Self-Attention)

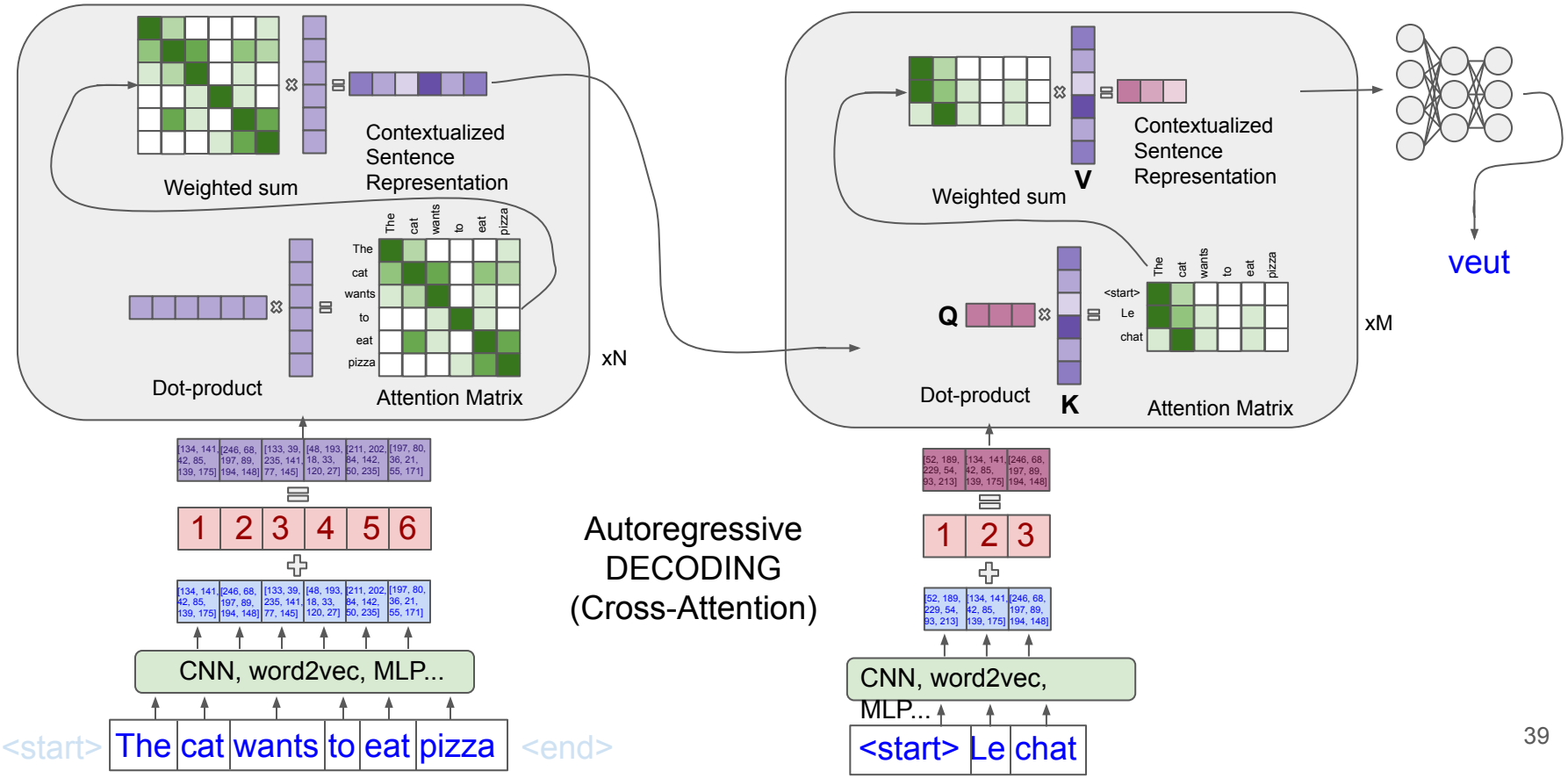
Overview of the transformer



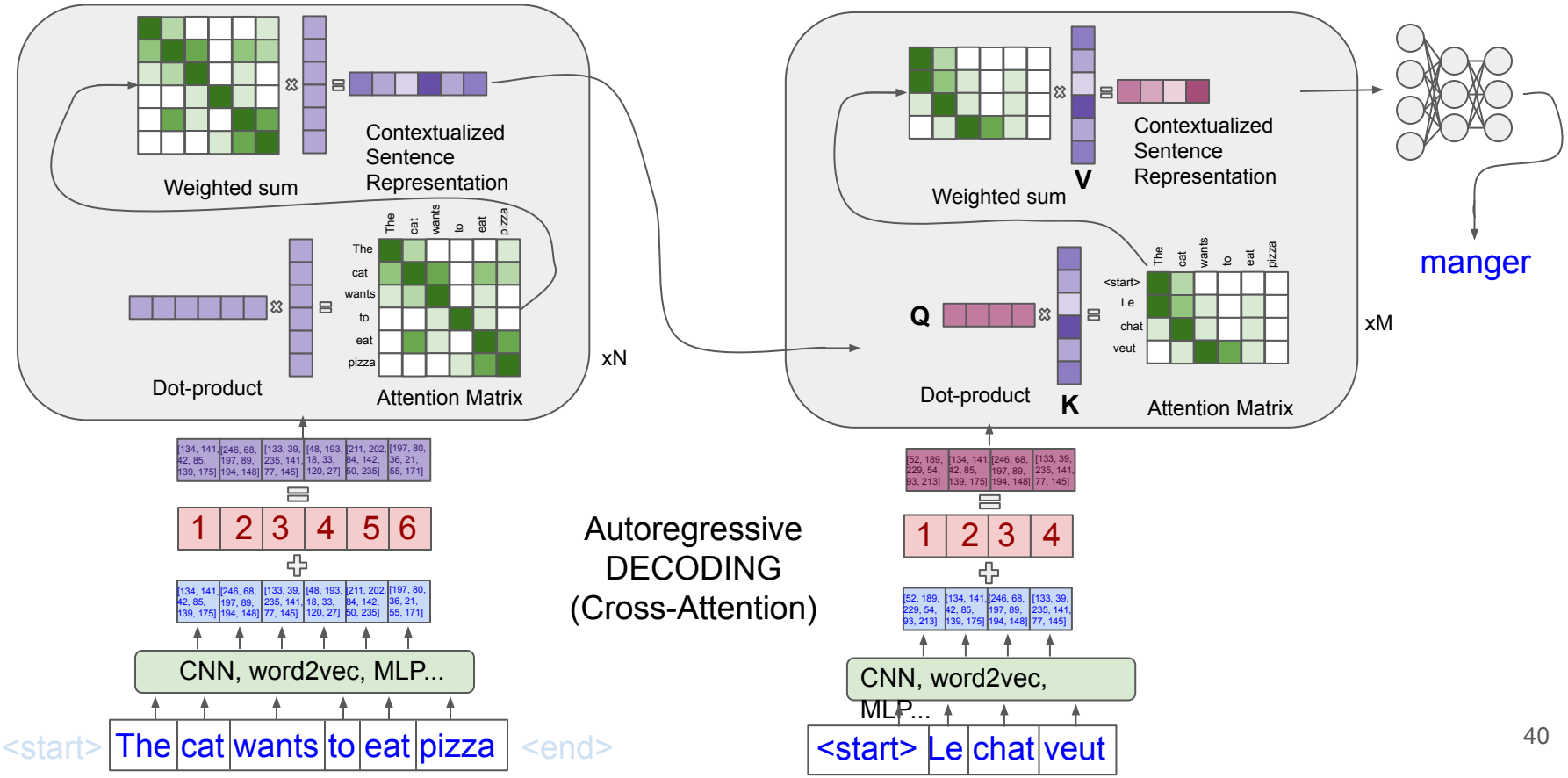
Overview of the transformer



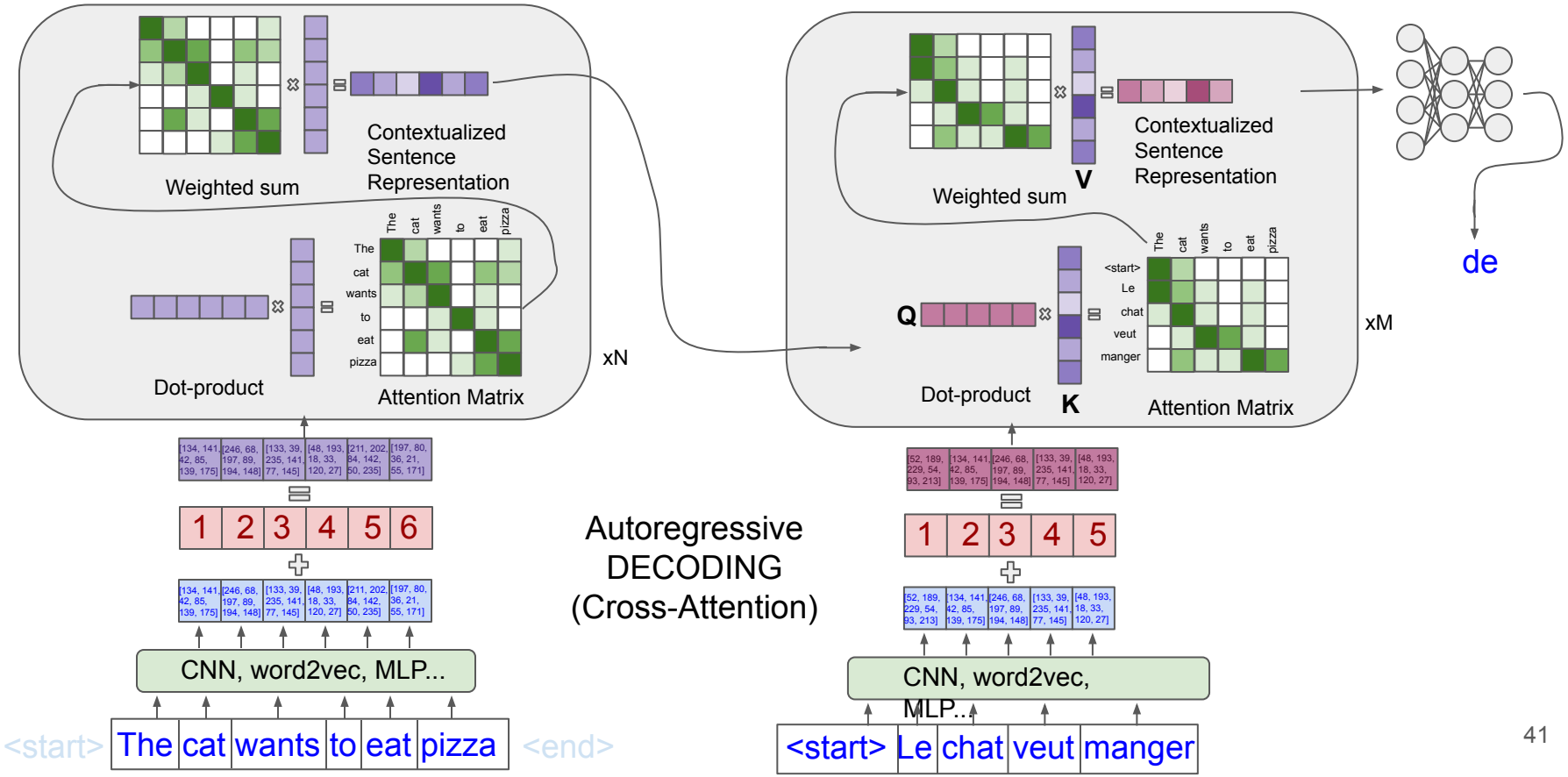
Overview of the transformer



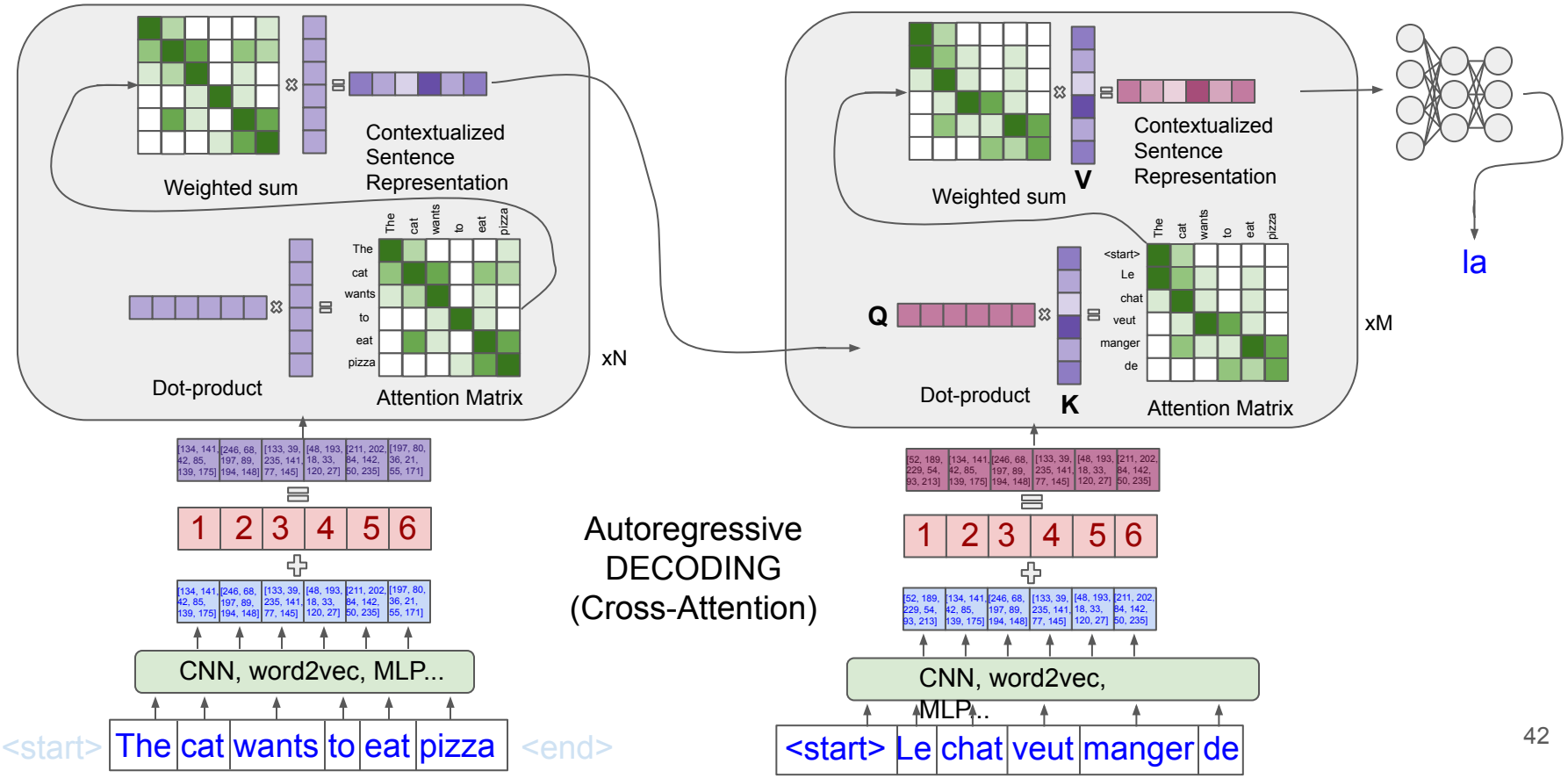
Overview of the transformer



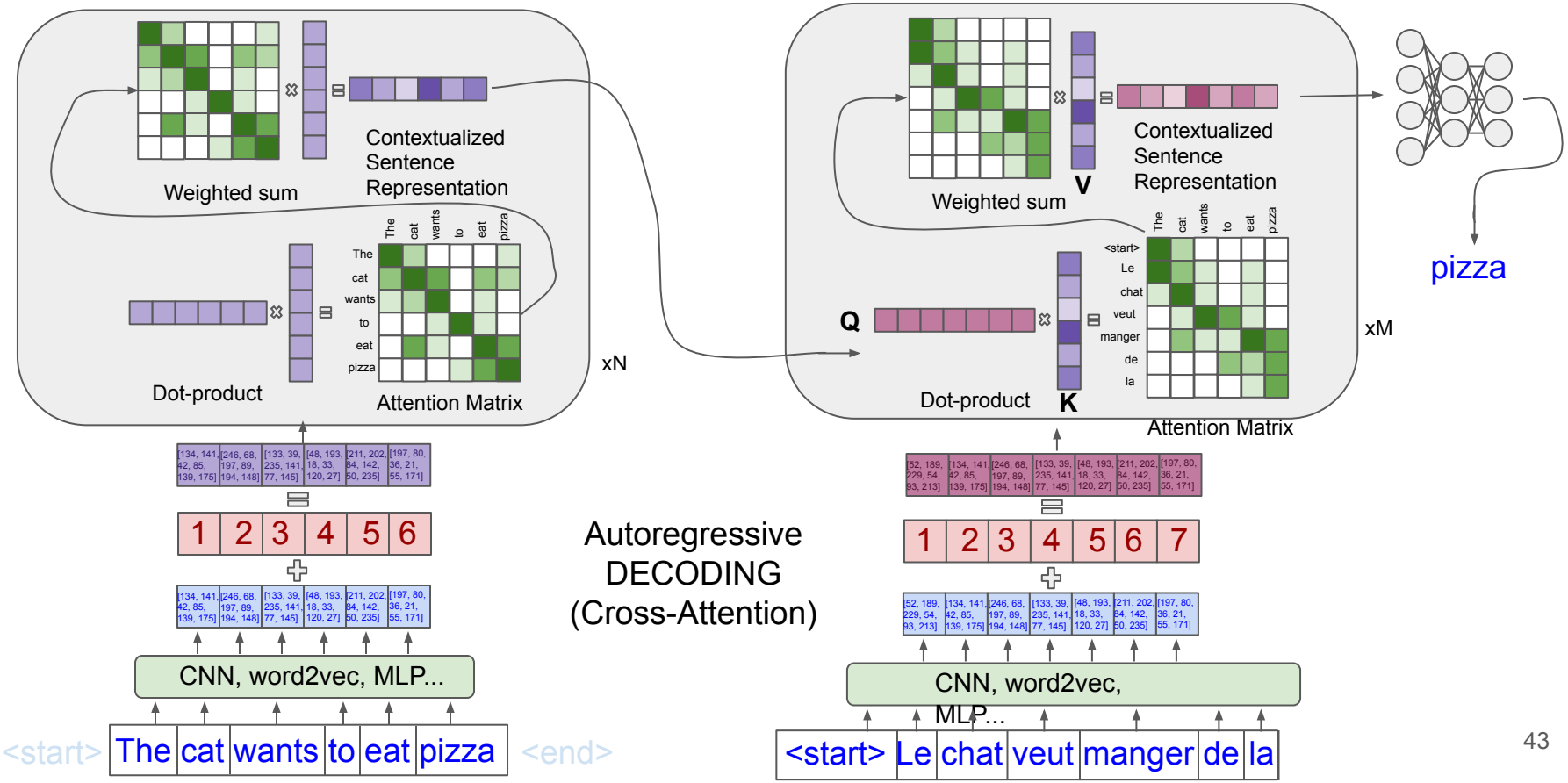
Overview of the transformer



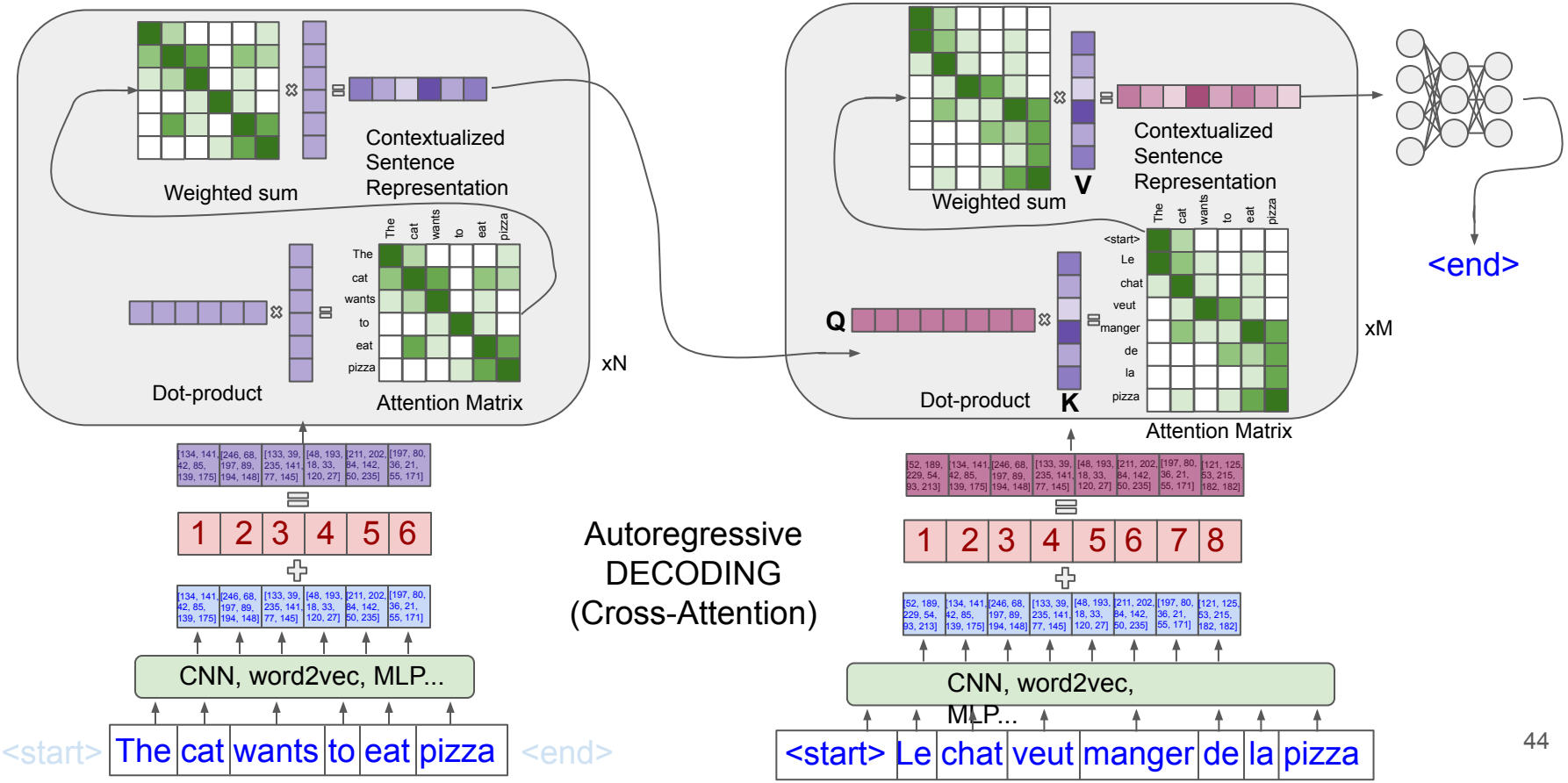
Overview of the transformer



Overview of the transformer



Overview of the transformer



AI Dungeon (GPT2)

play.aidungeon.io

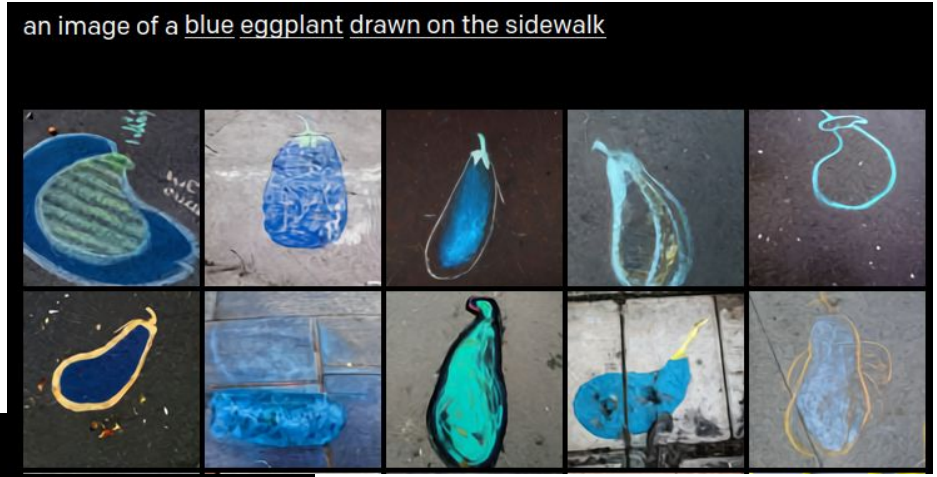
```
You are Sven, a rogue living in the kingdom of Larion. You have a long steel
 dagger and a length of rope. You walk down the city street looking for somehwere
 to steal from. You look around and see that it's not too crowded so you decide
 to go into an old building that looks abandoned.
As you enter the door, you hear a voice behind you
> turn around

You turn around and see a man with a beard and wearing a black robe standing
 there. He has a very stern face but his eyes tell you he is friendly.
"Hello," he says. "I'm Doctor Zook".
> stab the doctor

You quickly pull out your knife and stab the doctor in the back. The doctor falls
 backwards and then you grab the rope hanging from the ceiling and swing yourself
 up. You land on top of the doctor who is still struggling to get free.

> stab the doctor again

You stab the doctor in the chest three more times. The doctor screams as blood
 pours out of him. You stand over him and look at the bloody body.
"Well done," Dr. Zook says.
> Say "Wait! I killed you!"

You shout "I didn't kill you!" and point your finger at the doctor.
>
```

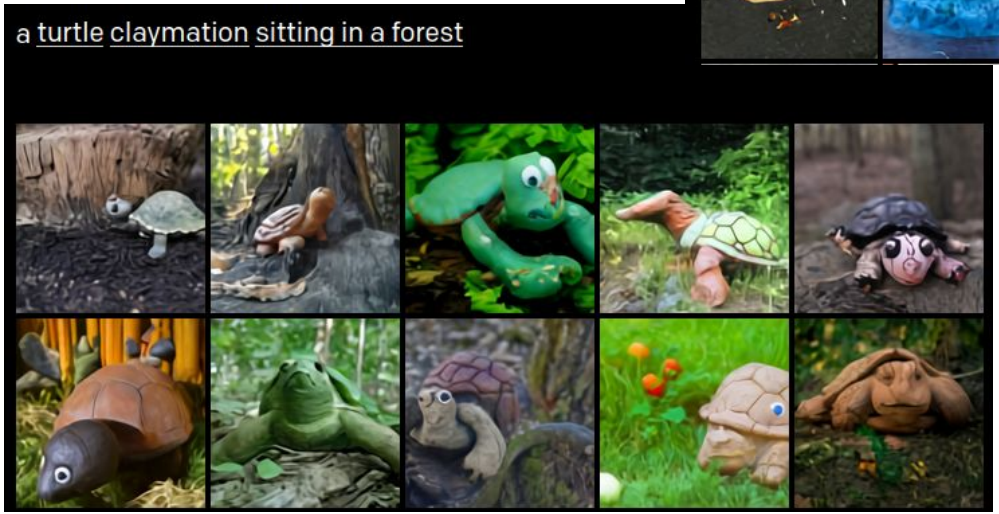
GPT2

Human Prompt	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
Machine Completion	The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. ...

(Radford et al. 2019)

DALL-E



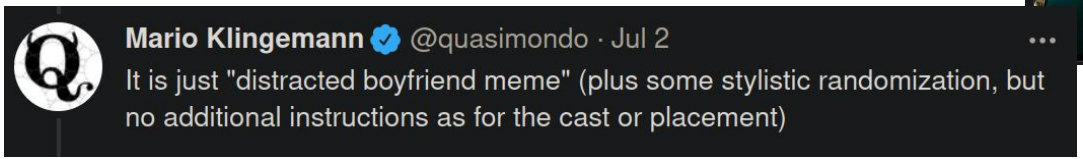


Source: https://openai.com/blog/dall-e/

CLIP

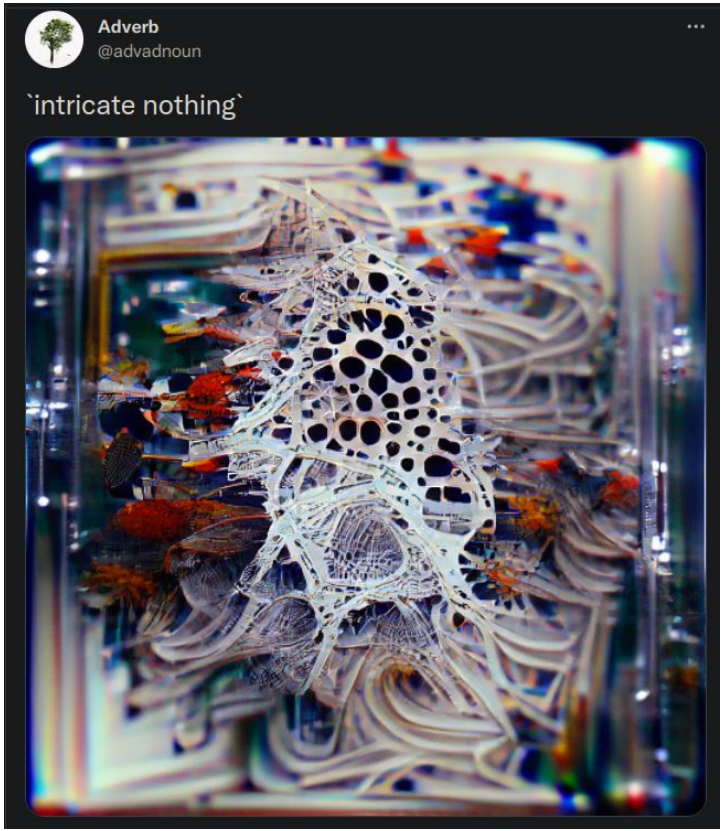
Source: https://twitter.com/quasimondo/status/1410933306640699398

CLIP



Source: https://twitter.com/quasimondo/status/1410933306640699398

Impressive results in many modalities!



Source: https://twitter.com/advadnoun/status/1389834237793112065

Impressive results in many modalities!

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Source: [5] **Dosovitskiy, A.** et al. "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*". In ICLR (2020).

Impressive results in many modalities!

(a) Kinetics 400

Method	Top 1	Top 5	Views
blVNet [16]	73.5	91.2	–
STM [30]	73.7	91.6	–
TEA [39]	76.1	92.5	10 × 3
TSM-ResNeXt-101 [40]	76.3	–	–
I3D NL [72]	77.7	93.3	10 × 3
CorrNet-101 [67]	79.2	–	10 × 3
ip-CSN-152 [63]	79.2	93.8	10 × 3
LGD-3D R101 [48]	79.4	94.4	–
SlowFast R101-NL [18]	79.8	93.9	10 × 3
X3D-XXL [17]	80.4	94.6	10 × 3
TimeSformer-L [2]	80.7	94.7	1 × 3
ViViT-L/16x2	80.6	94.7	4 × 3
ViViT-L/16x2 320	**81.3**	**94.7**	4 × 3
Methods with large-scale pretraining			
ip-CSN-152 [63] (IG [41])	82.5	95.3	10 × 3
ViViT-L/16x2 (JFT)	82.8	95.5	4 × 3
ViViT-L/16x2 320 (JFT)	83.5	95.5	4 × 3
ViViT-H/16x2 (JFT)	**84.8**	**95.8**	4 × 3

(b) Kinetics 600

Method	Top 1	Top 5	Views
AttentionNAS [73]	79.8	94.4	–
LGD-3D R101 [48]	81.5	95.6	–
SlowFast R101-NL [18]	81.8	95.1	10 × 3
X3D-XL [17]	81.9	95.5	10 × 3
TimeSformer-HR [2]	82.4	**96.0**	–
ViViT-L/16x2	82.5	95.6	4 × 3
ViViT-L/16x2 320	**83.0**	95.7	4 × 3
ViViT-L/16x2 (JFT)	84.3	96.2	4 × 3
ViViT-H/16x2 (JFT)	**85.8**	**96.5**	4 × 3

(c) Moments in Time

	Top 1	Top 5
TSN [69]	25.3	50.1
TRN [83]	28.3	53.4
I3D [6]	29.5	56.1
blVNet [16]	31.4	59.3
AssembleNet-101 [51]	34.3	62.7
ViViT-L/16x2	**38.0**	**64.9**

(d) Epic Kitchens 100 Top 1 accuracy

Method	Action	Verb	Noun
TSN [69]	33.2	60.2	46.0
TRN [83]	35.3	65.9	45.4
TBN [33]	36.7	66.0	47.2
TSM [40]	38.3	**67.9**	49.0
SlowFast [18]	38.5	65.6	50.0
ViViT-L/16x2 Fact. encoder	**44.0**	66.4	**56.8**

(e) Something-Something v2

Method	Top 1	Top 5
TRN [83]	48.8	77.6
SlowFast [17, 77]	61.7	–
TimeSformer-HR [2]	62.5	–
TSM [40]	63.4	88.5
STM [30]	64.2	89.8
TEA [39]	65.1	–
blVNet [16]	65.2	**90.3**
ViViT-L/16x2 Fact. encoder	**65.4**	89.8

Source: [23] **Arnab, A.** et al. "*Vivit: A video vision transformer*". In CoRR (2021).

Transformers for CV

Transformers are complex!

- Transformers are highly parametric!
  - Each **head** has $\mathbf{W_Q}$, $\mathbf{W_K}$ and $\mathbf{W_V}$ each of size (d_w x d_k).
  - Each **layer** also includes $\mathbf{W_O}$, $\mathbf{W_{F1}}$ and $\mathbf{W_{F2}}$ of sizes (d_w x d_w), (d_w x cd_w) and (cd_w x d_w), respectively.
  - By setting the **dimensionality** of the input to $\boldsymbol{d_w}$ **= 512**, the number of **heads** to **h = 8**, and **c = 4** *[3]* **each layer ends up with ~2.5M parameters**.
  - **6** transformer **layers** quickly turn into **~14.7M parameters**.

	# layers	# params
ResNet	20	0.27M
ResNet	32	0.46M
ResNet	44	0.66M
ResNet	56	0.85M
ResNet	110	1.7M
ResNet	1202	19.4M

Source: [3] **Vaswani, A.** et al. "*Attention is all you need*". In NeurIPS (2017)
[4] **He, K.** et al. "*Deep residual learning for image recognition*". In CVPR (2016).

Transformers are complex!

- Transformers are highly **parametric**!
- Transformers lack **inductive biases**:
  - That is why we need positional encodings.
  - Opposed to convolutions, it has to learn **translation equivariance** or importance **local relations** from the data *[5]*.

Source: [5] **Dosovitskiy, A.** et al. "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*". In ICLR (2020).

Transformers are complex!

- Transformers are highly **parametric**!
- Transformers lack **inductive biases**.
- Need a lot of resources!
  - **Big datasets** to allow the Transformer to learn
  - Long **training times** / **Big** computational clusters (**GPUs**)

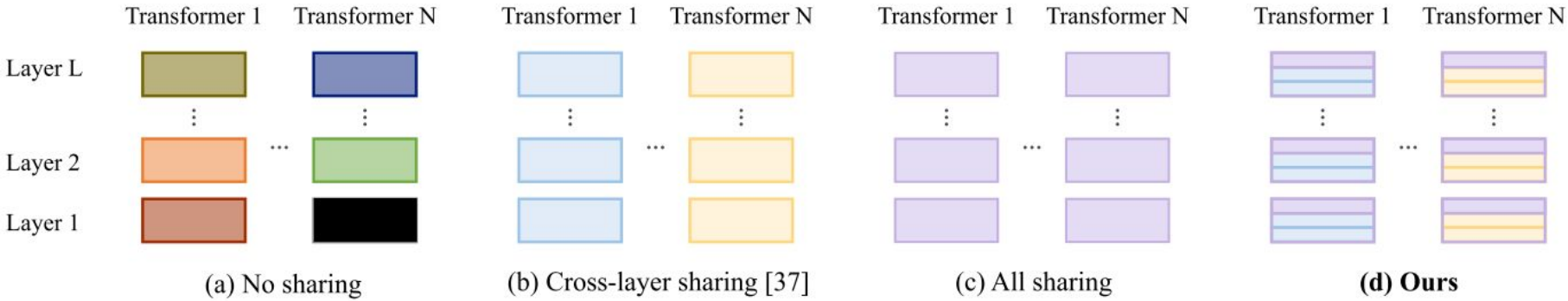
Source: [5] **Dosovitskiy, A.** et al. "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*". In ICLR (2020).

Transformers are complex!

- Transformers are highly **parametric**!
- Transformers lack **inductive biases**.
- Need a lot of resources!
- Solutions:
  - **Efficient designs**: Share weights.
  - **Pre-training**: Use **unsupervised** / self-supervised techniques to leverage more data

Source: [5] **Dosovitskiy, A.** et al. "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*". In ICLR (2020).

Weight Sharing



c) Model	X.-L	X.-T	Params	top-1/5
Multi-2	✗	✗	7M	60.3 / 88.9
Multi-6	✓	✗	21M	65.7 / 89.9
Multi-6	✓	✓(All)	7M	67.1 / 92.3
Multi-6	✓	✓(Part†)	**4M**	**67.5 / 92.3**

$$W = U\Sigma V^T$$

Source: [6] **Lee, S.** et al. "*Parameter Efficient Multimodal Transformers for Video Representation Learning*". In ICLR (2020).

Pre-training

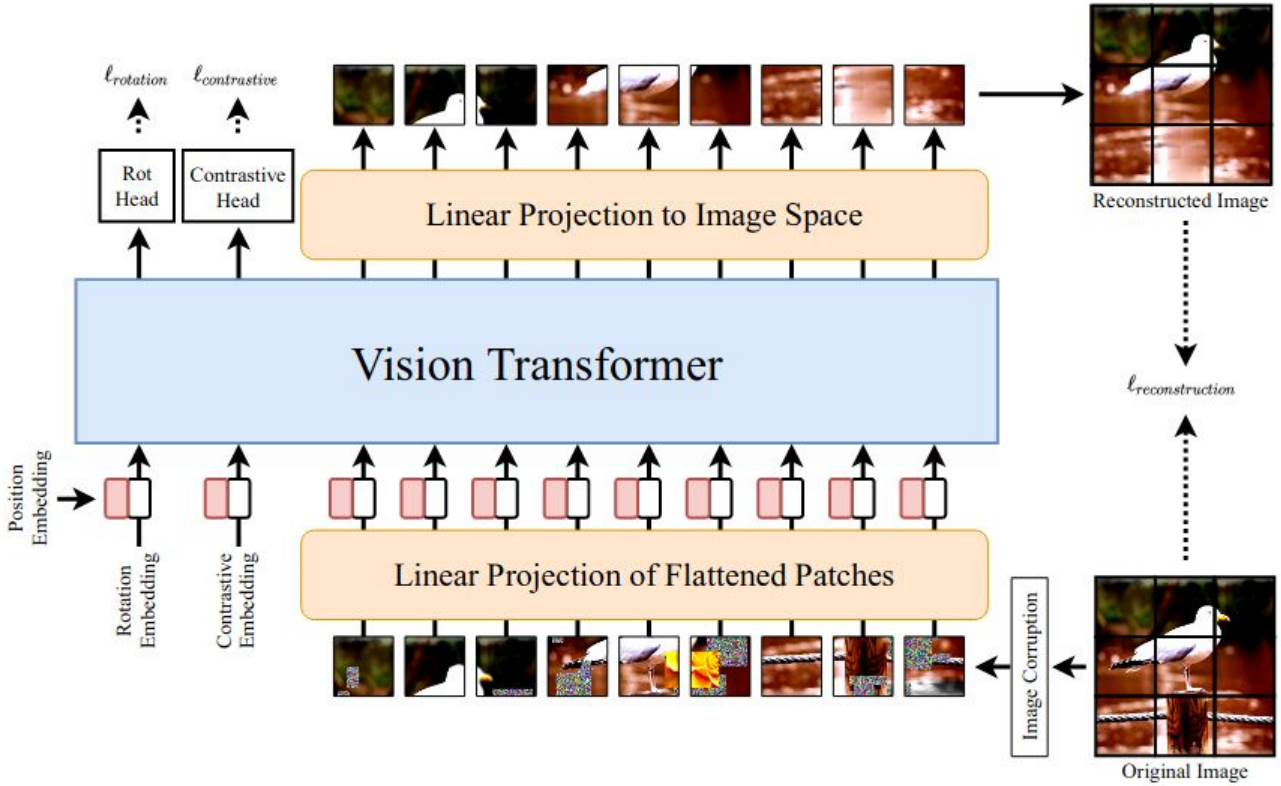
Use **large unlabeled** datasets to pre-train the model:

- **Internal structure** of the data.

Train with a **small** subset of **labeled dataset** for a specific task:

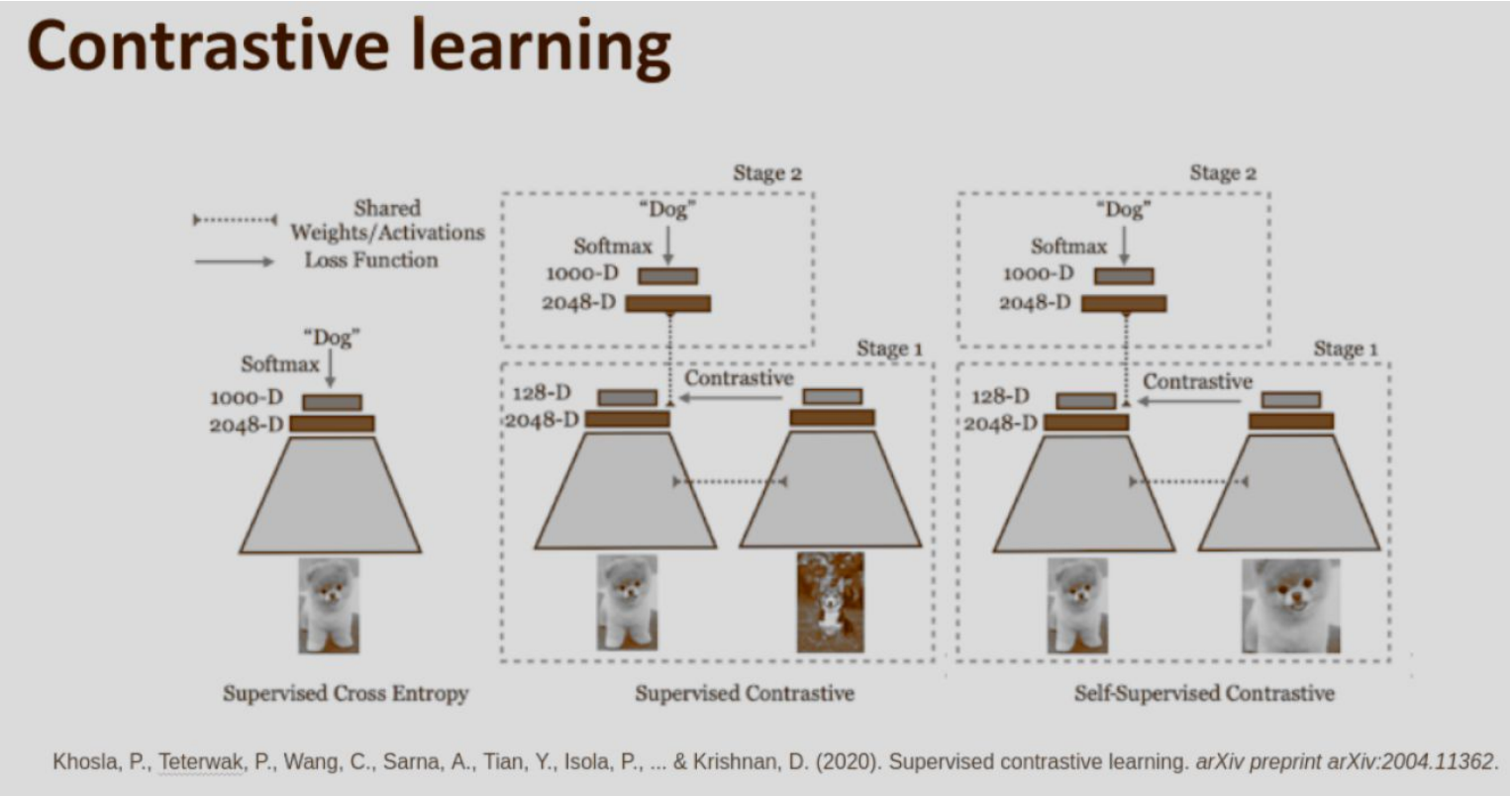
- Relevant **task specific information**.

Pre-training: Reconstruction



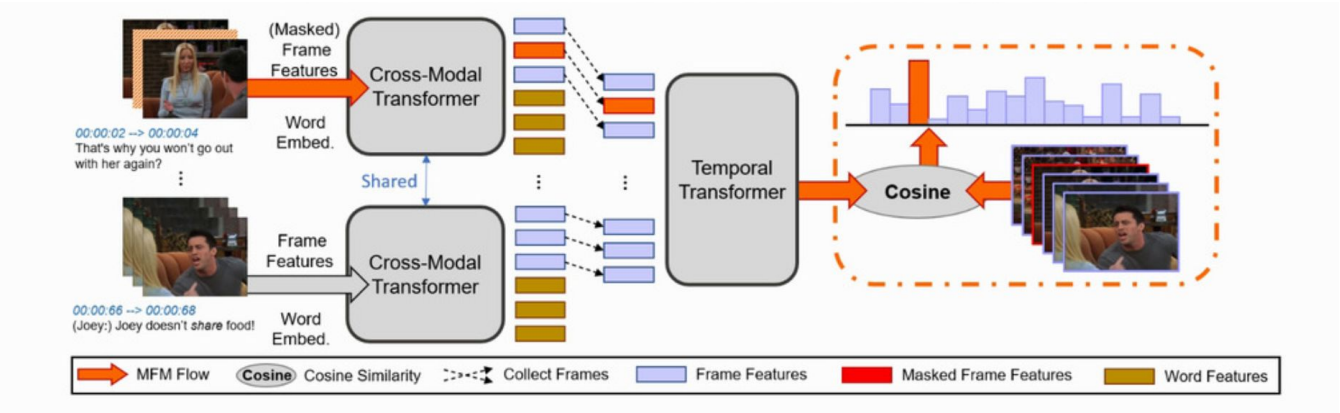
Source: [7] **Atito, S.** et al. "*Sit: Self-supervised vision transformer*". In CoRR (2021).

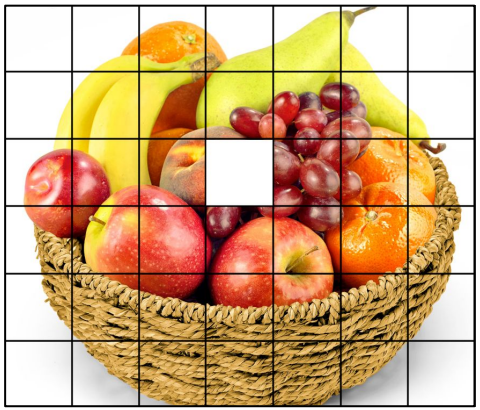
Previously…



Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., ... & Krishnan, D. (2020). Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Pre-training: Contrastive



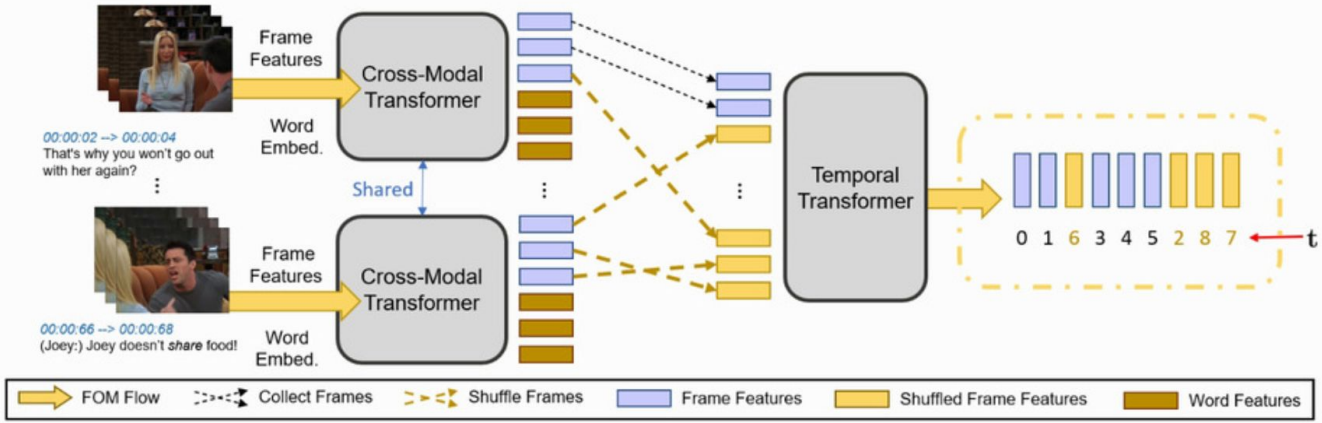


- Maximize Mutual information between input and output.
- Generally through **Noise Contrastive Estimation**

$$\mathcal{L}_{\text{NCE}}(\mathbf{x}, \tilde{\mathbf{o}}) = -\mathbb{E}_{\mathbf{x}} \left[\sum_t \log \frac{\mathbf{I}(\mathbf{x}_t, \tilde{\mathbf{o}}_t)}{\mathbf{I}(\mathbf{x}_t, \tilde{\mathbf{o}}_t) + \sum_{j \in \text{neg}(t)} \mathbf{I}(\mathbf{x}_j, \tilde{\mathbf{o}}_t)} \right]$$

Sources:[8] **Li, L.** et al. "*HERO: Hierarchical Encoder for Video+ Language Omni-representation Pre-training*". In EMNLP (2020).
[9] **Oord, A. V. D.** et al. "*Representation learning with contrastive predictive coding*". In CoRR (2018).

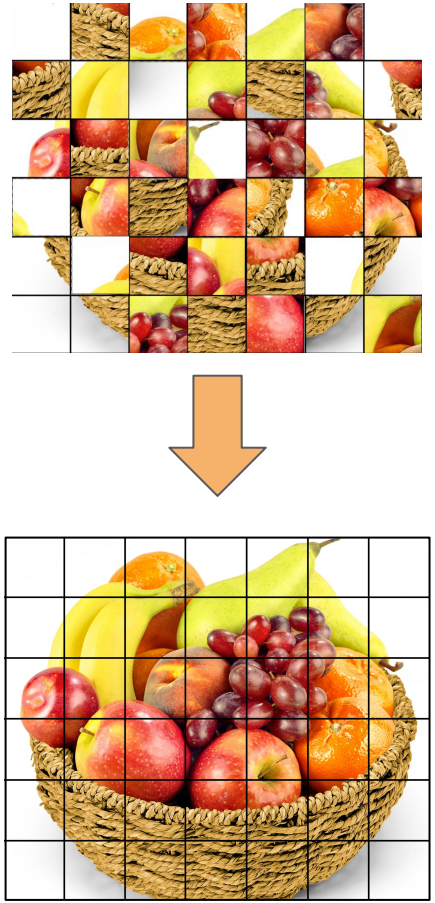
Pre-training: Token shuffling



Reorder Indices: $\mathbf{r} = \{r_i\}_{i=1}^{R} \in \mathbb{N}^R$

Original timestamp: $\mathbf{t} = \{t_i\}_{i=1}^{R}$

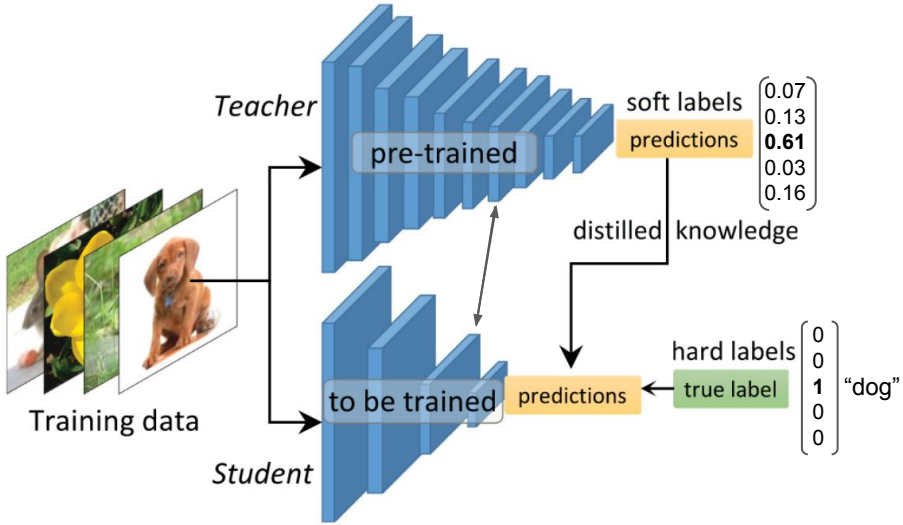
Loss Function of FOM: $\mathcal{L}_{\text{FOM}} = -\mathbb{E}_D \sum_{i=1}^{R} \log \mathbf{P}[r_i, t_i]$

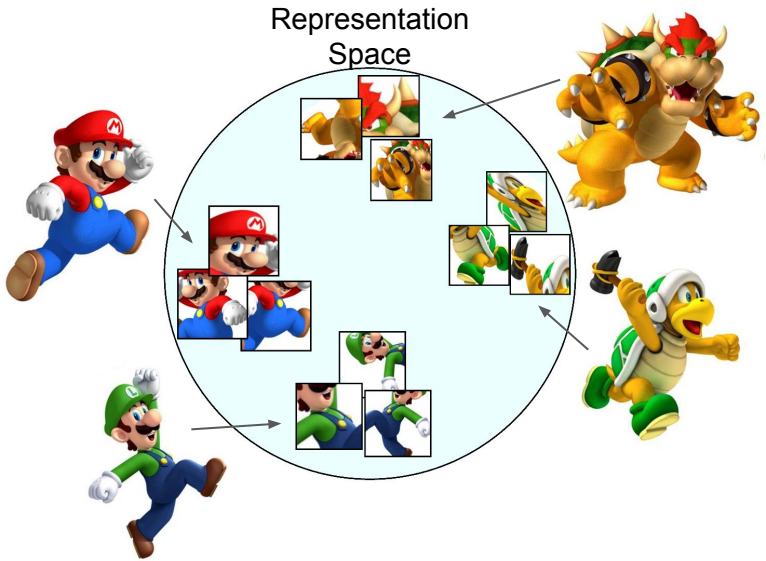


Source: [8] **Li, L.** et al. "*HERO: Hierarchical Encoder for Video+ Language Omni-representation Pre-training*". In EMNLP (2020).

Pre-training: Momentum Contrast

Borrows ideas from **knowledge distillation**, **data augmentation** and **contrastive/self-supervised losses**.

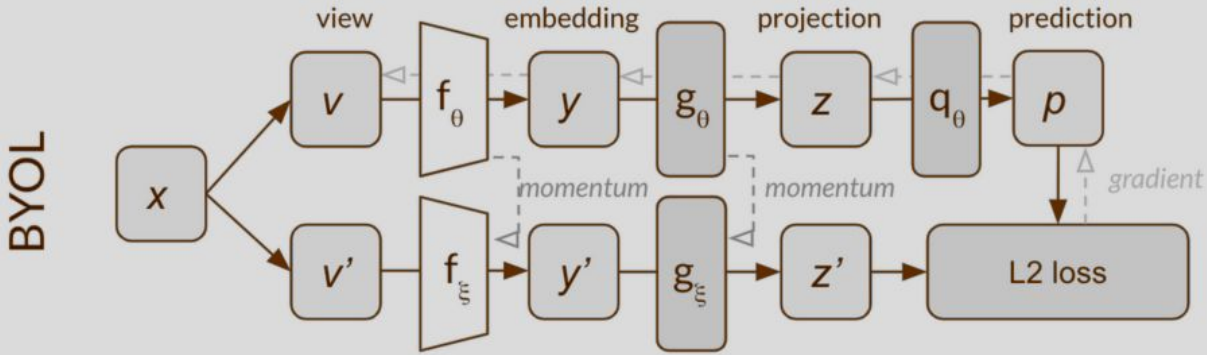




Source: [10] **Wang, J.** et al. "*DeepVID: Deep Visual Interpretation and Diagnosis for Image Classifiers via Knowledge Distillation*". In IEEE Tr-VCG (2019).

Previously…

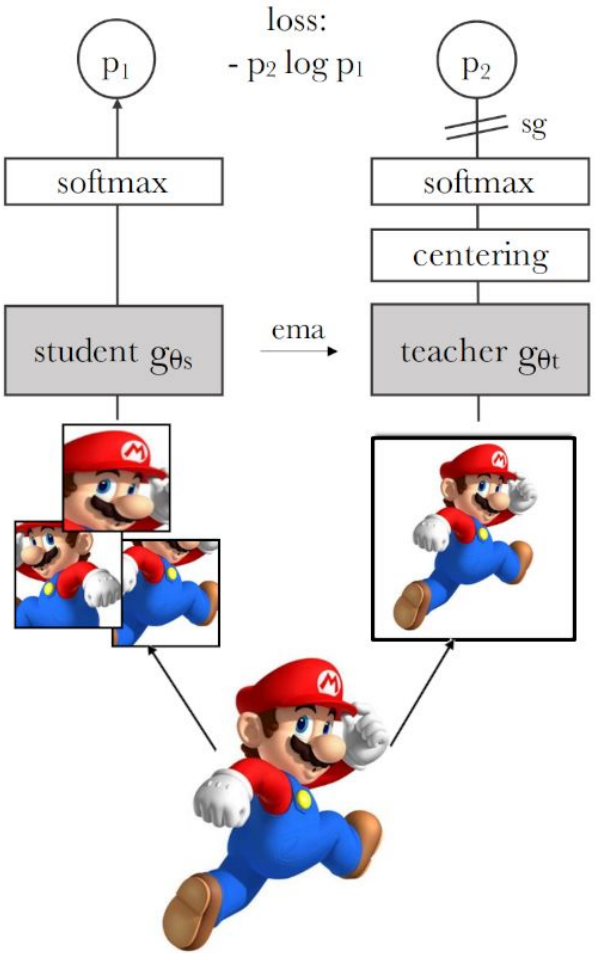
Distillation



Grill, J. B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., ... & Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Pre-training: Momentum Contrast

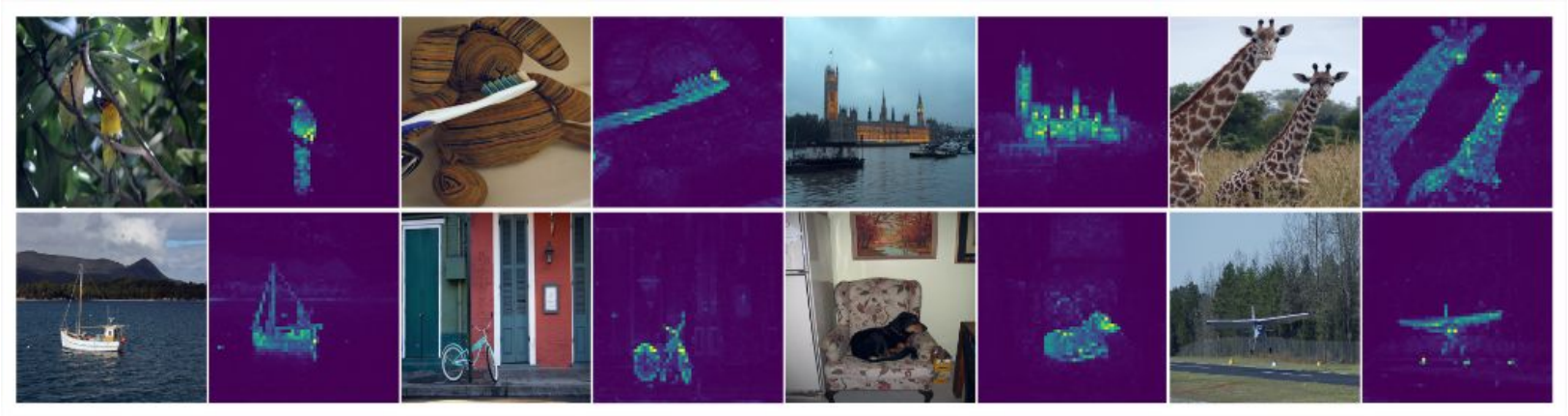
DINO



Sources: "*Advancing the state of the art in computer vision with self-supervised Vision Transformers*" https://www.youtube.com/watch?v=8I1ReInsgMw
[11] **Caron, M.** et al. "*Emerging Properties in Self-Supervised Vision Transformers*". In CoRR (2021).

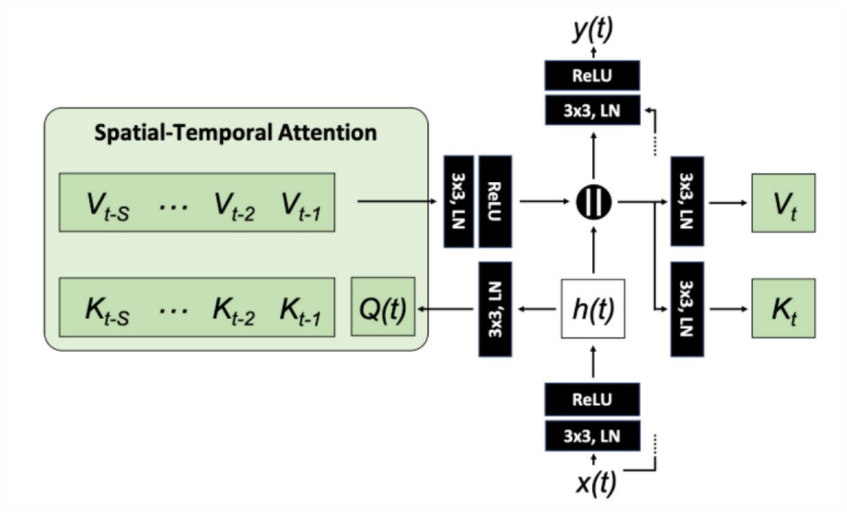
Pre-training: Momentum Contrast

DINO

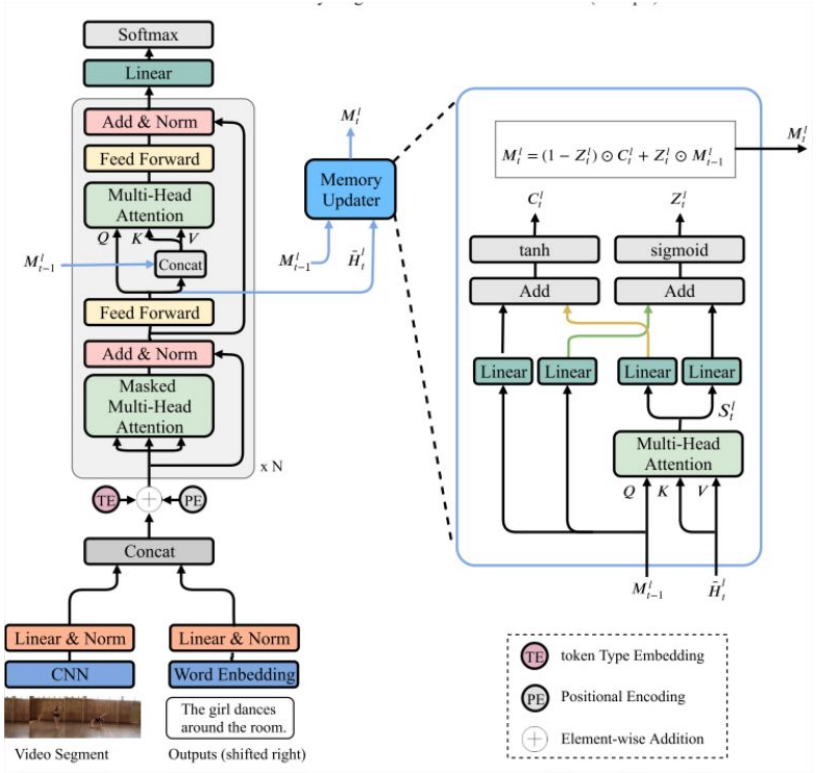


Sources: [11] **Caron, M.** et al. "*Emerging Properties in Self-Supervised Vision Transformers*". In CoRR (2021).

Recurrent Transformers



HORST



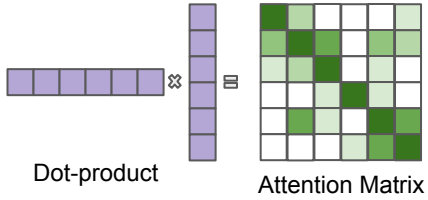
MART

Source:

Efficient Transformers

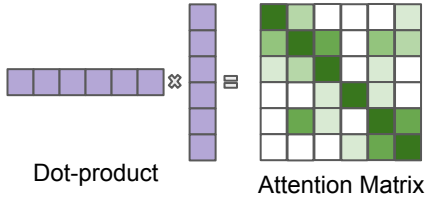
Attention matrix grows fast!

The size of the attention matrix grows
quadratically with the number of input tokens
(sequence length T) → $O(T^2)$



Transformers could adapt to arbitrarily long inputs, but this limits them to do so!!

Attention matrix grows fast!



The size of the attention matrix grows
quadratically with the number of input tokens
(sequence length T) → $O(T^2)$

Transformers could adapt to arbitrarily long inputs, but this limits them to do so!!

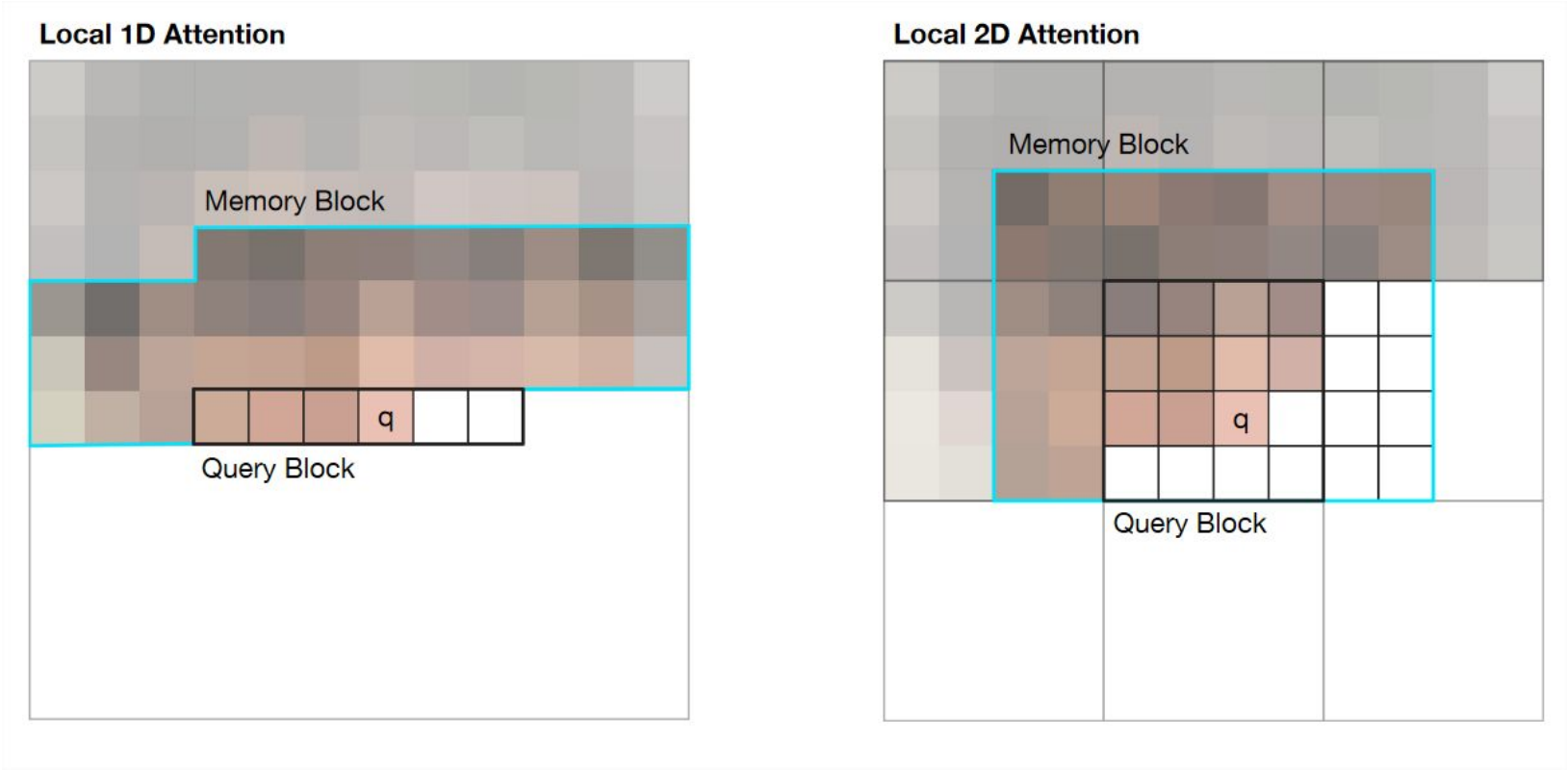
Solution: **Sparse / Decomposed attention**.

Limit the number of tokens in a single attention operation.

Efficient Designs (Locality/Sparsity)

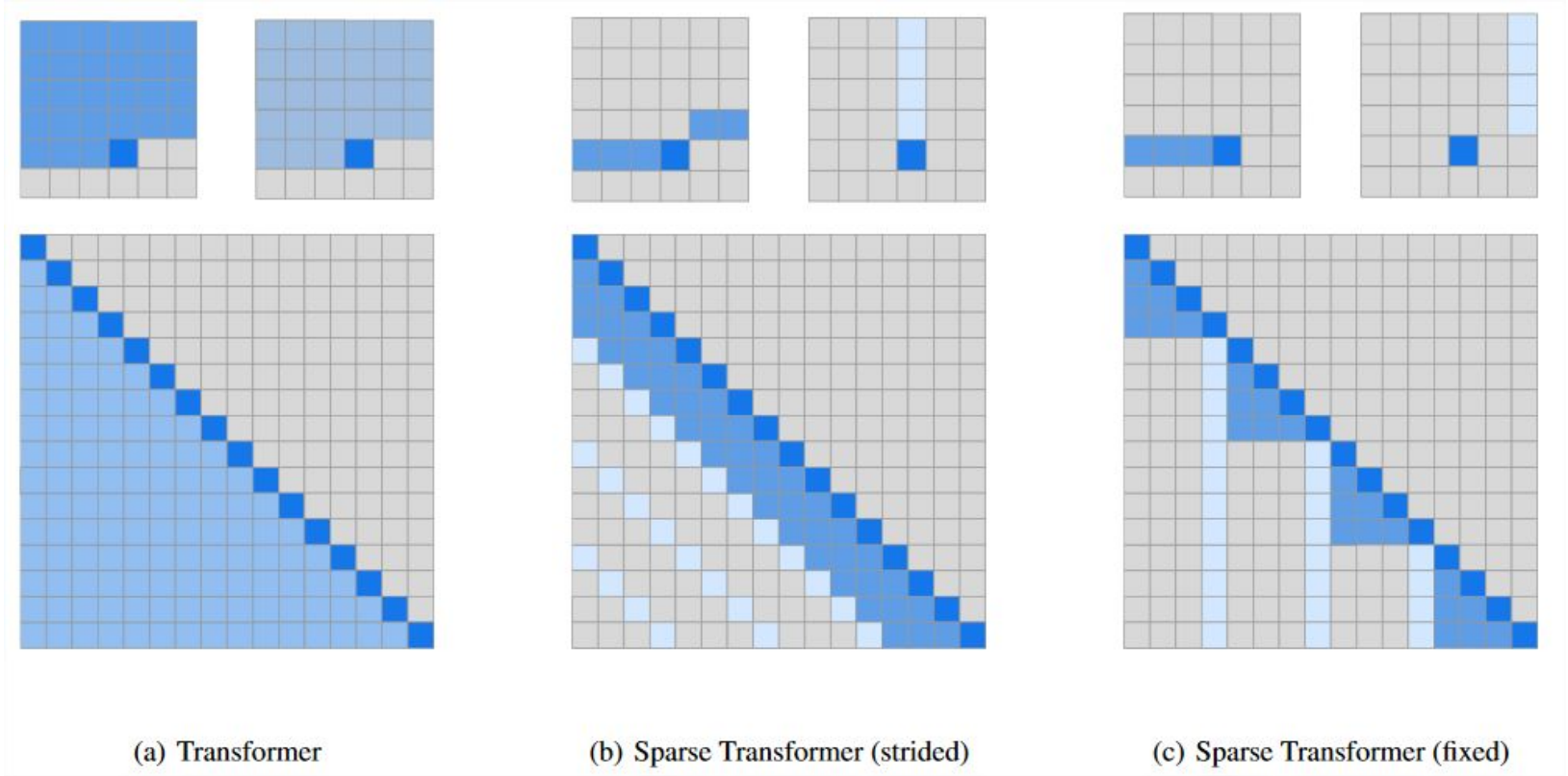
- Limit the number of tokens accessible in a single operation
- **Local**: Only allow to attend to neighboring tokens
- **Sparse**: Tokens can be distant, but only every other token is accessible.

Efficient Designs (Locality/Sparsity)



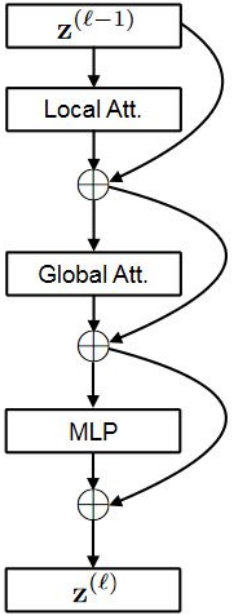
Source: [19] **Parmar, N.** et al. "*Image transformer*". In ICML (2018).

Efficient Designs (Locality/Sparsity)



Source: [20] **Child, R.** et al. "*Generating long sequences with sparse transformers*". In CoRR (2019).

Efficient Designs (Locality/Sparsity)

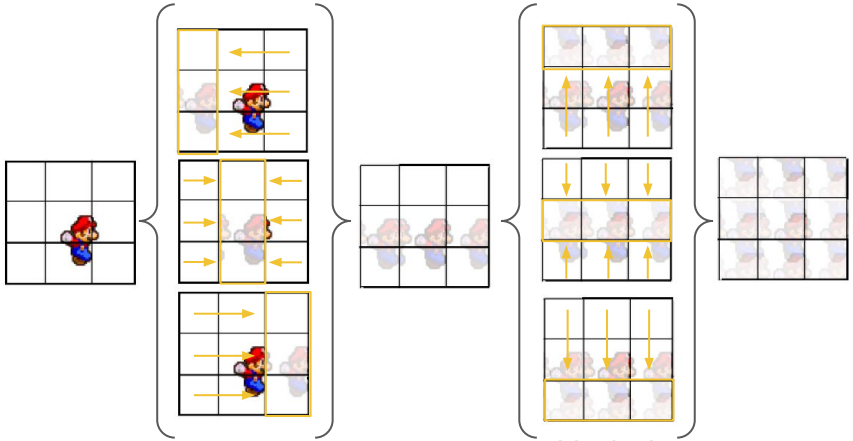


Sparse Local Global Attention (L+G)

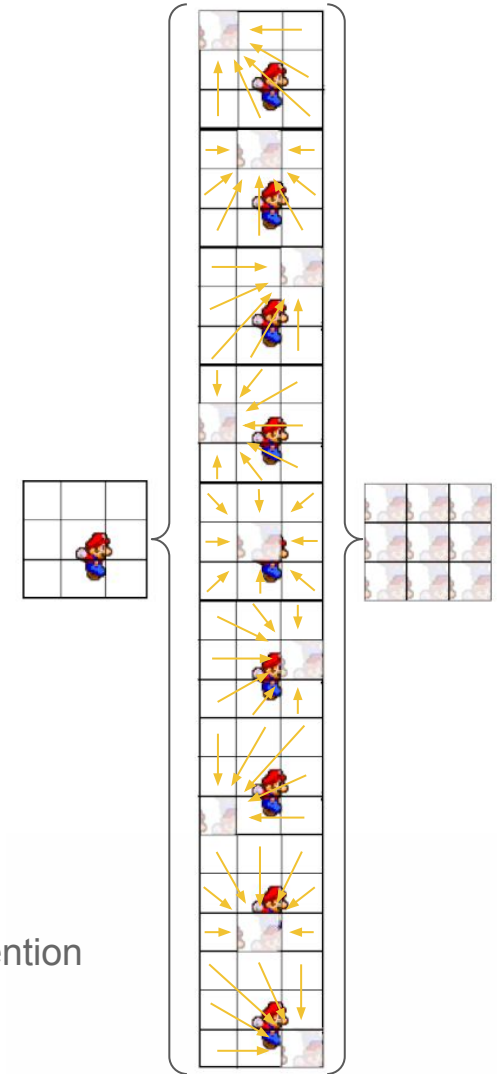
Source: [21] **Bertasius, G.** et al. "*Is Space-Time Attention All You Need for Video Understanding?*". In CoRR (2021).

Efficient Designs (Decomposition)

Approximate full attention with two or more **consecutive attention** operations

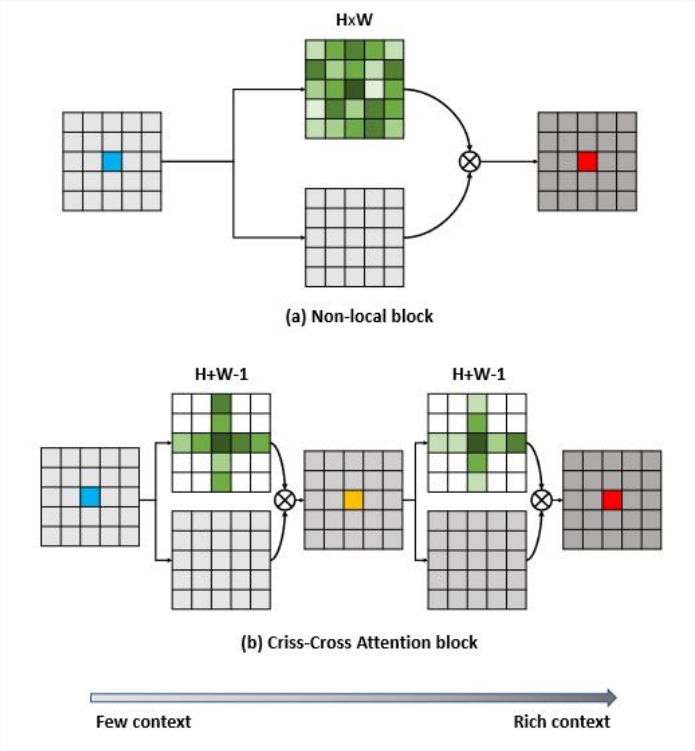


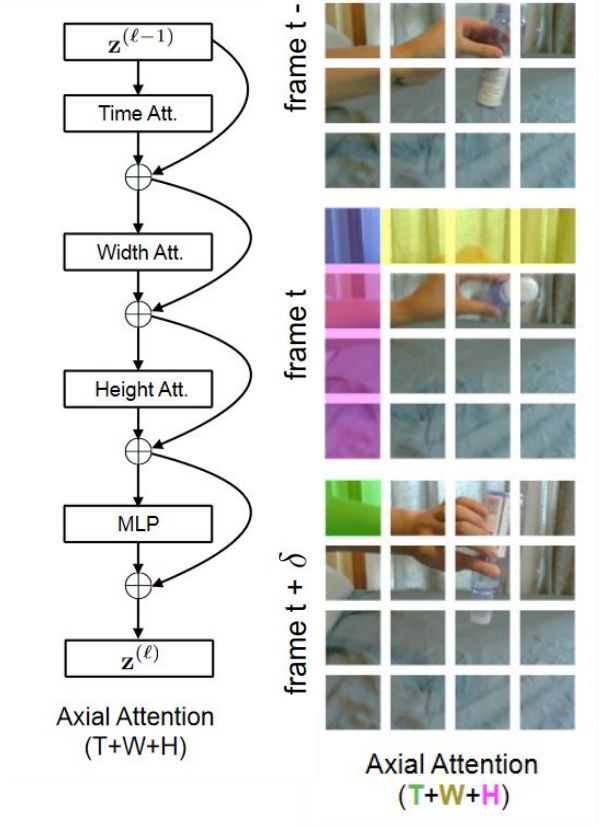
Axial Attention



Full Attention

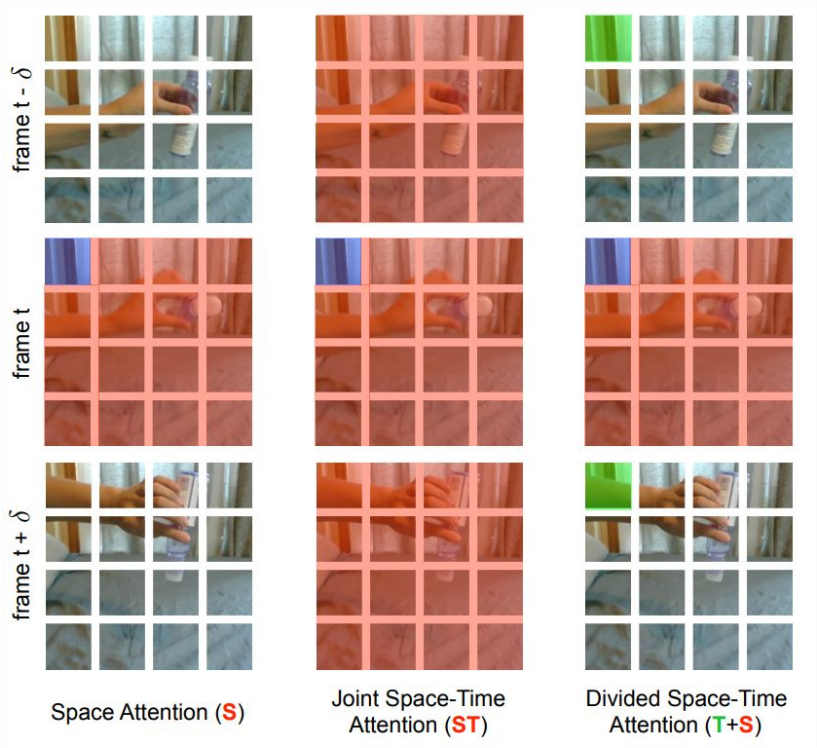
Efficient Designs (Decomposition - Axial)

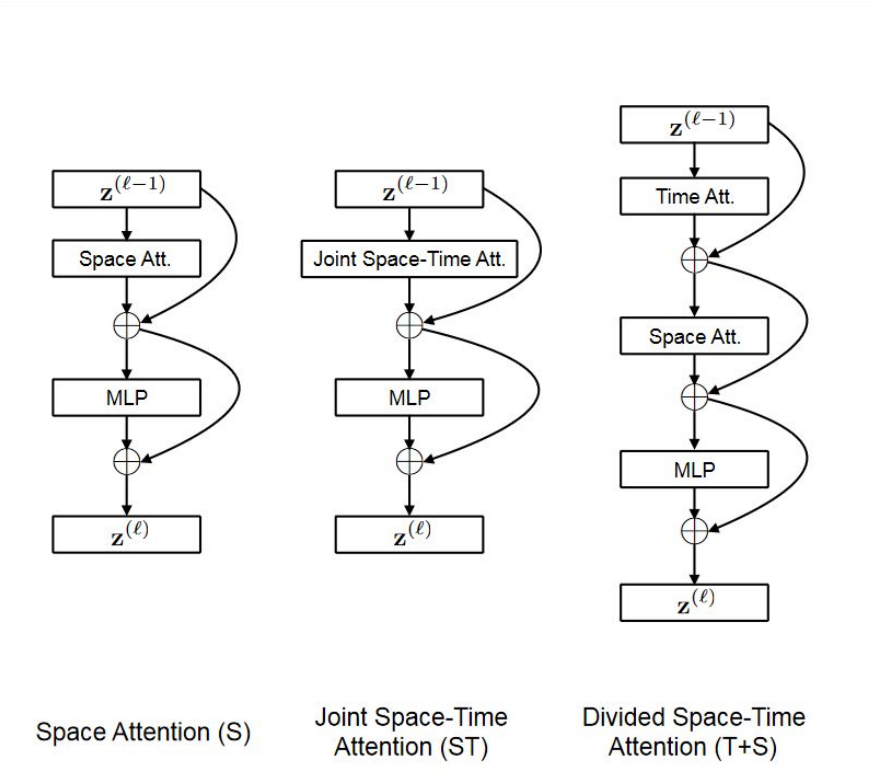




Source: [22] **Huang, Z.** et al. "*Ccnet: Criss-cross attention for semantic segmentation*". In ICCV (2019).
[21] **Bertasius, G.** et al. "*Is Space-Time Attention All You Need for Video Understanding?*". In CoRR (2021).

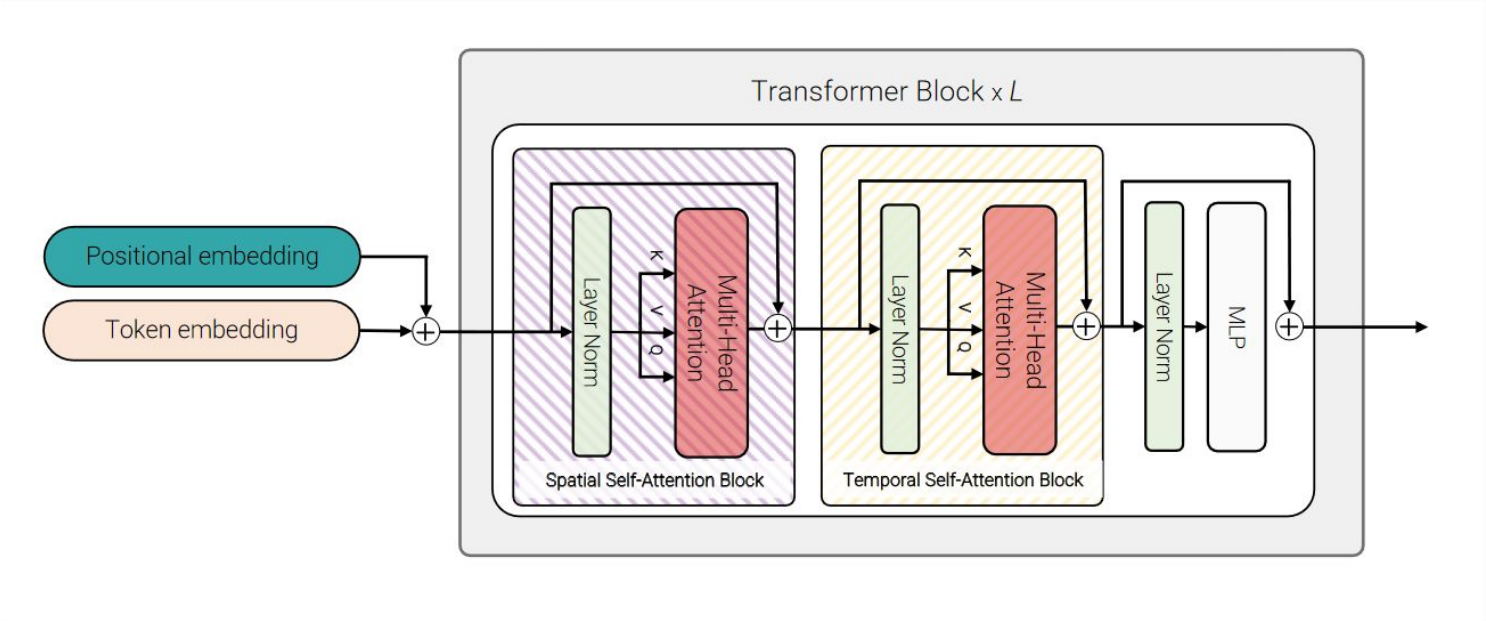
Efficient Designs (Decomposition - SpatioTemporal)





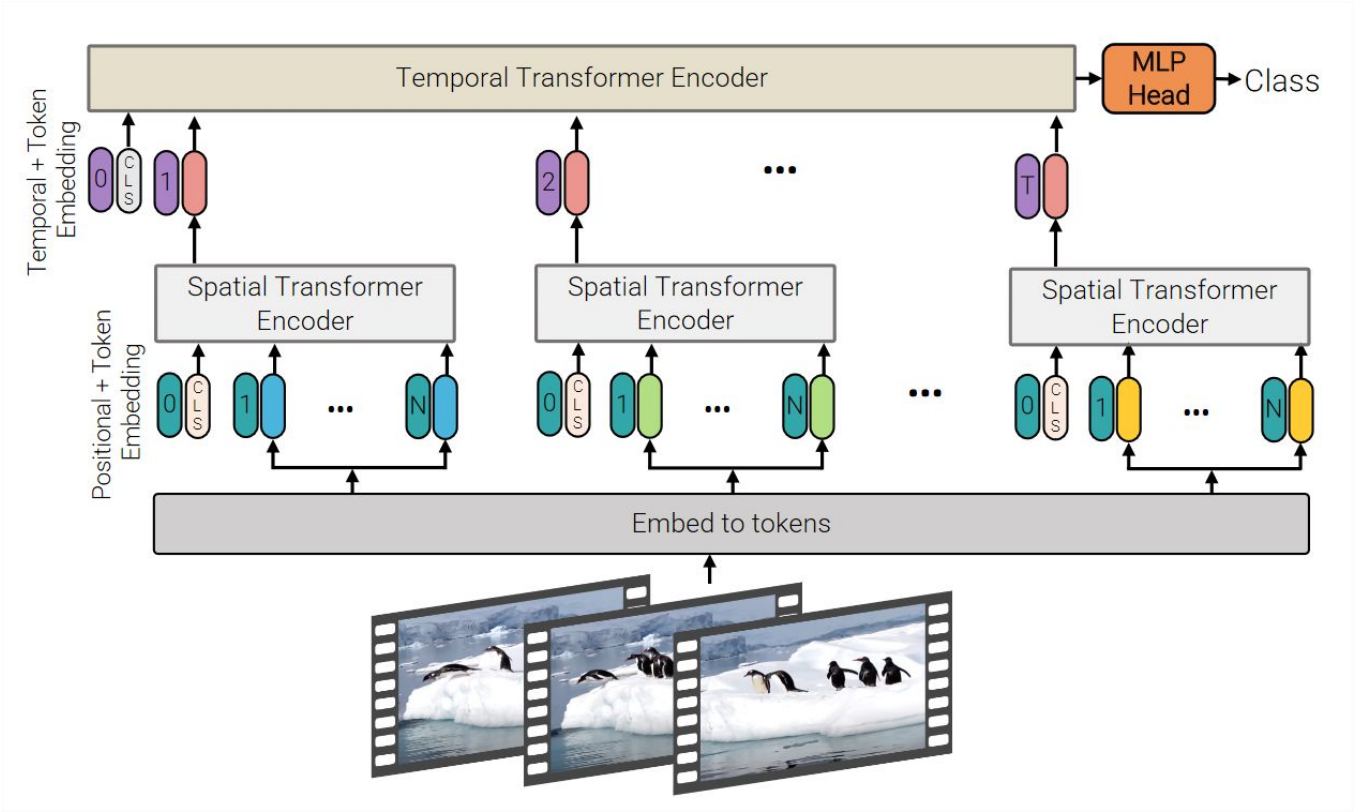
Source: [21] **Bertasius, G.** et al. "*Is Space-Time Attention All You Need for Video Understanding?*". In CoRR (2021).

Efficient Designs (Decomposition - SpatioTemporal)



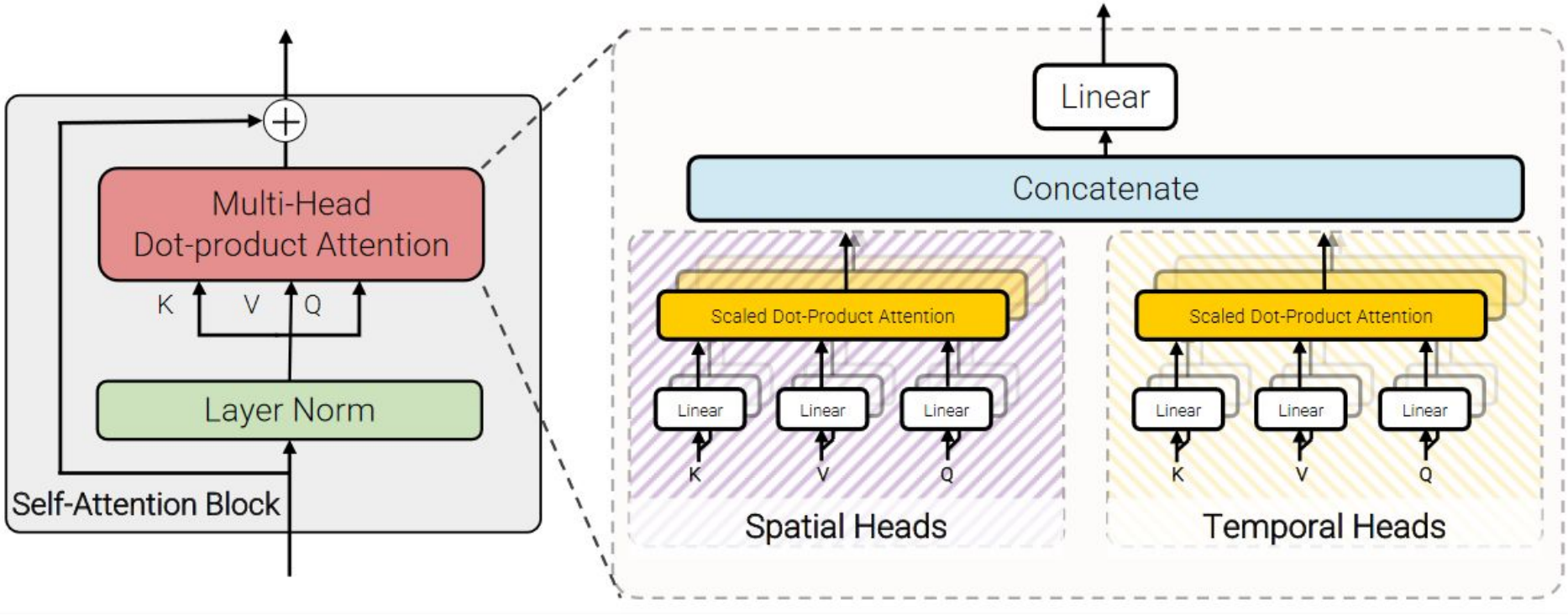
Source: [23] **Arnab, A.** et al. "*Vivit: A video vision transformer*". In CoRR (2021).

Efficient Designs (Decomposition - SpatioTemporal)



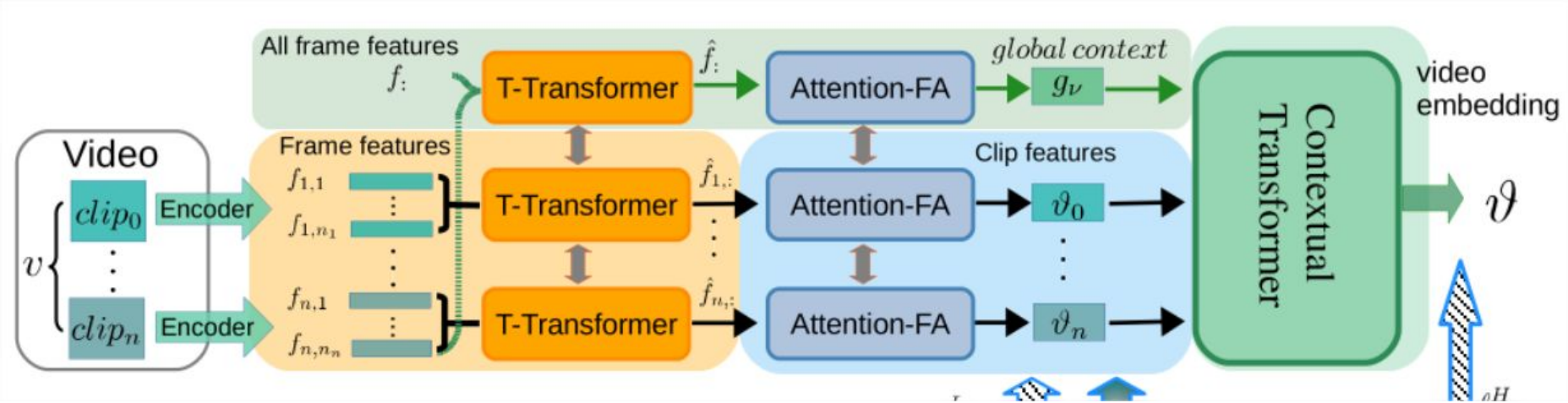
Source: [23] **Arnab, A.** et al. "*Vivit: A video vision transformer*". In CoRR (2021).

Efficient Designs (Decomposition - SpatioTemporal)



Source: [23] **Arnab, A.** et al. "*Vivit: A video vision transformer*". In CoRR (2021).

Efficient Designs (Hierarchy)

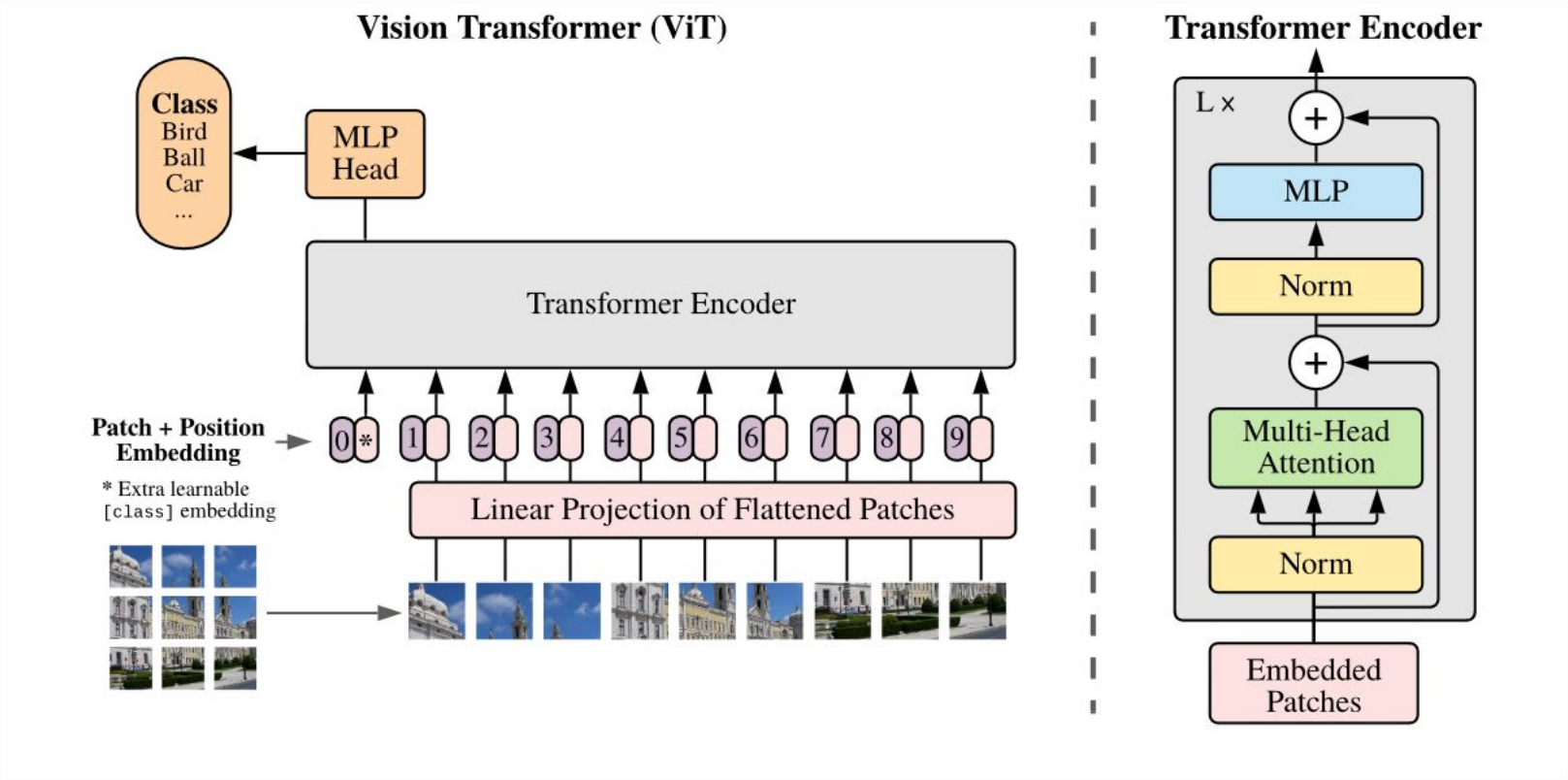


$$A = \text{softmax}(W_2 Q + b_2)^T, \qquad Q = GELU(W_1 K^T + b_1),\ K = X$$

Source: [24] **Ging, S.** et al. "*Coot: Cooperative hierarchical transformer for video-text representation learning*". In NeurIPS (2020).

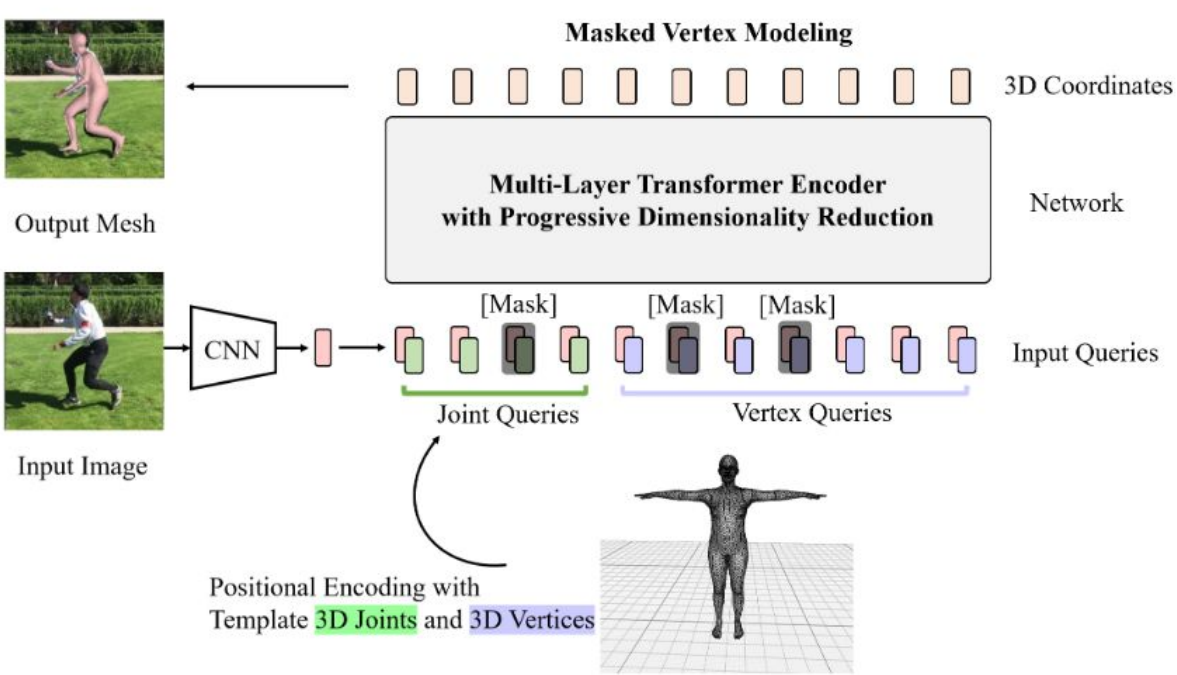
Applications

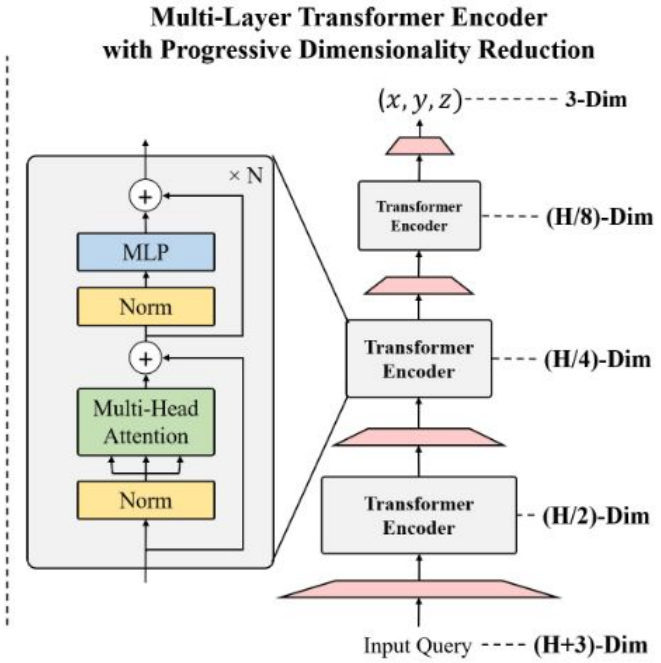
Vision Transformer (for classification)



Source: [5] **Dosovitskiy, A.** et al. "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*". In ICLR (2020).

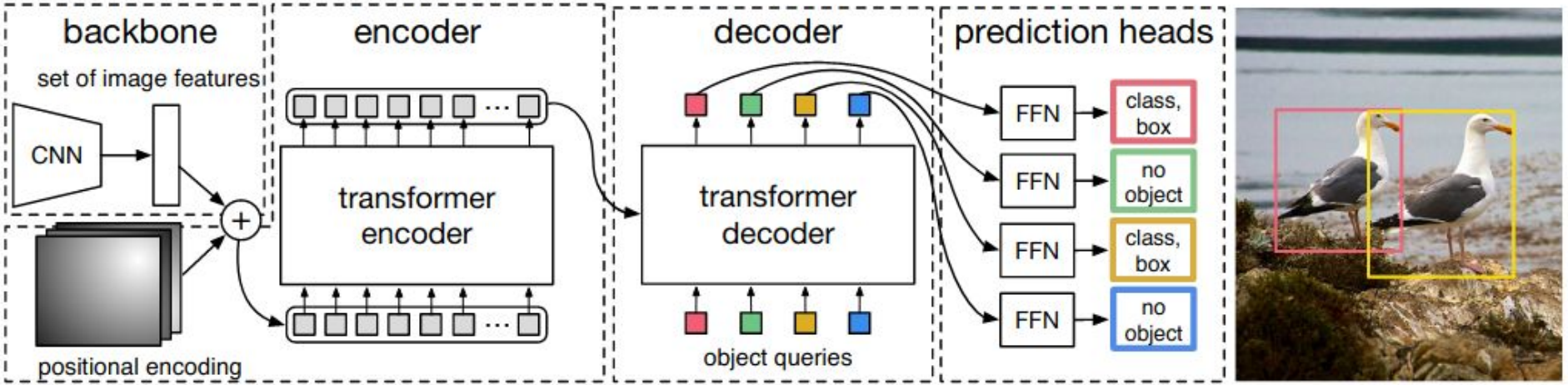
Mesh Reconstruction





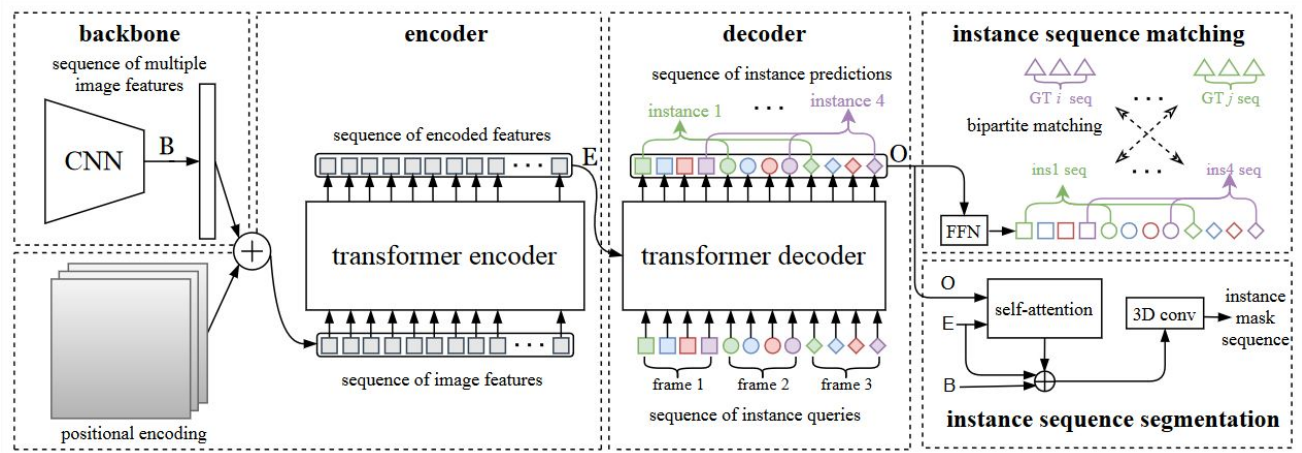
Source: [13] **Lin, K.** et al. "*End-to-end human pose and mesh reconstruction with transformers*". In CVPR (2021).

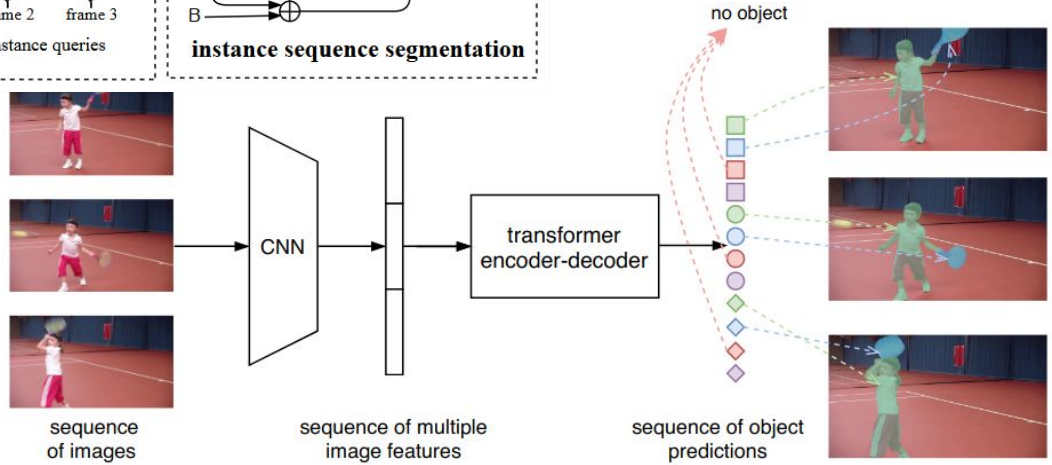
DETR (Transformer for object recognition)



Source: [12] **Carion, N.** et al. "*End-to-end object detection with transformers*". In ECCV (2020).

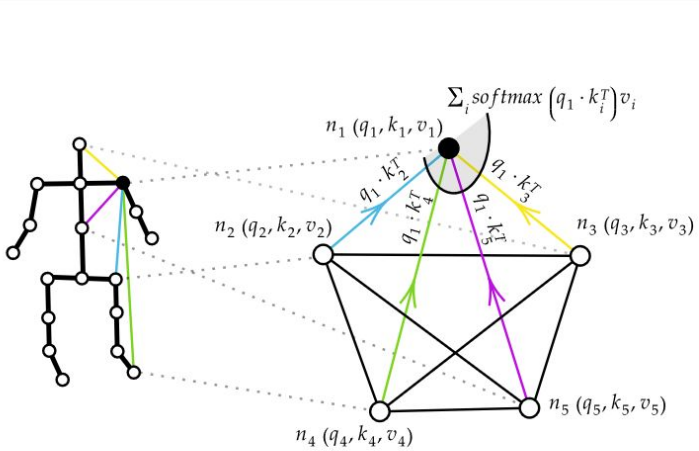
Transformer for video Segmentation



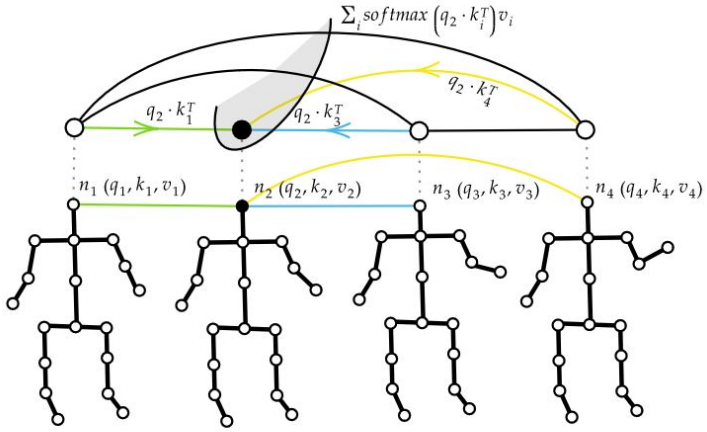


Source: [16] **Wang, Y.** et al. "*End-to-end video instance segmentation with transformers*". In CVPR (2021).

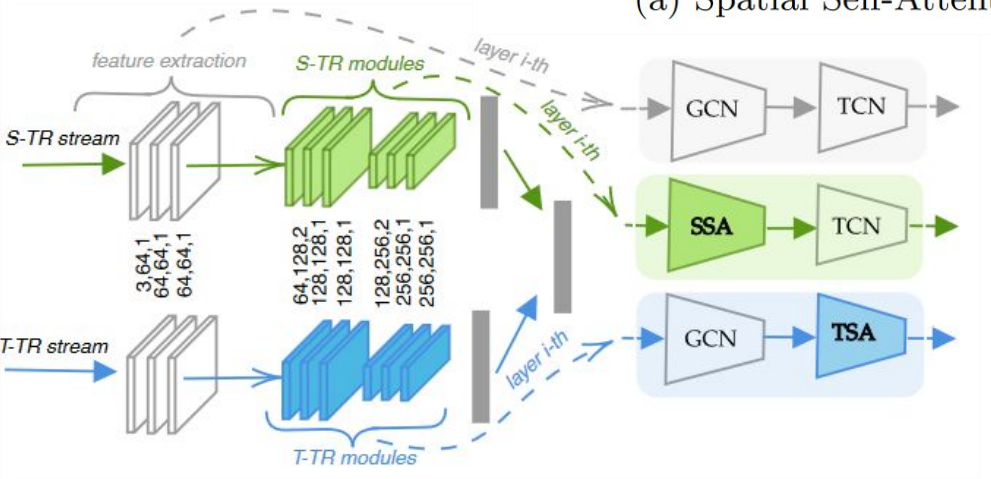
Pose (Action Recognition)



(a) Spatial Self-Attention

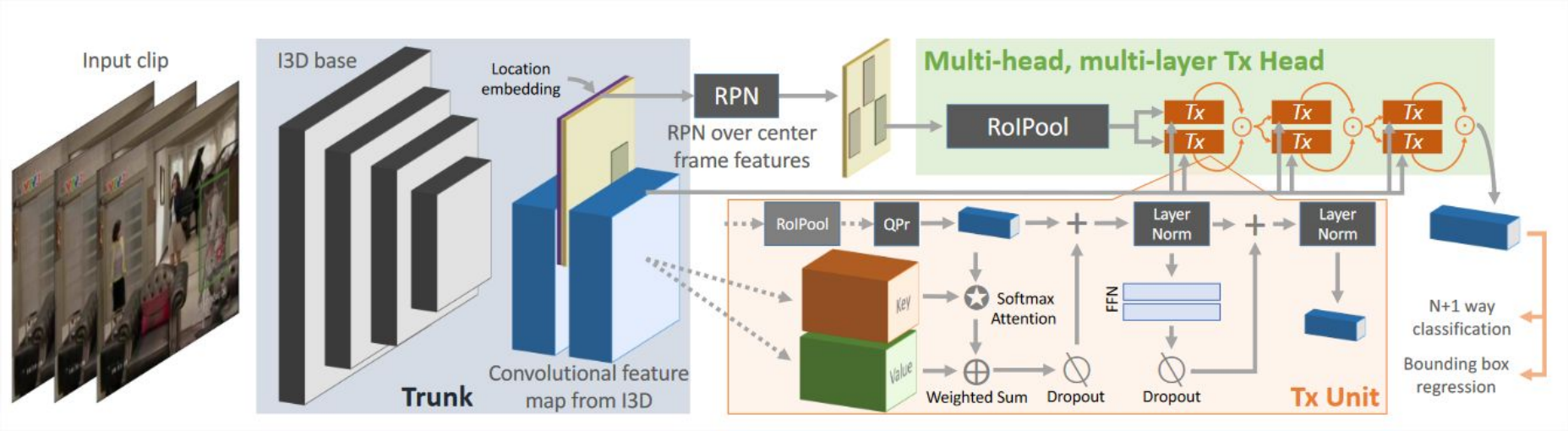


(b) Temporal Self-Attention



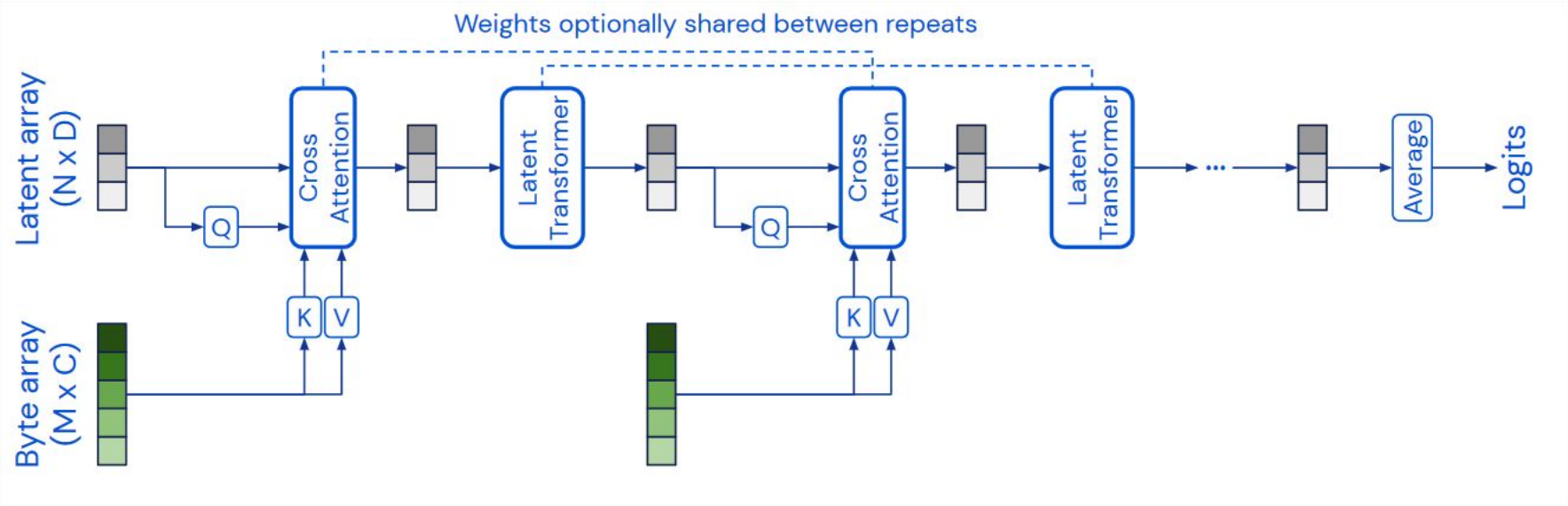
Source: [15] **Plizzari, C.** et al. "*Spatial temporal transformer network for skeleton-based action recognition*". In ICPR (2021).

Video Action Transformer



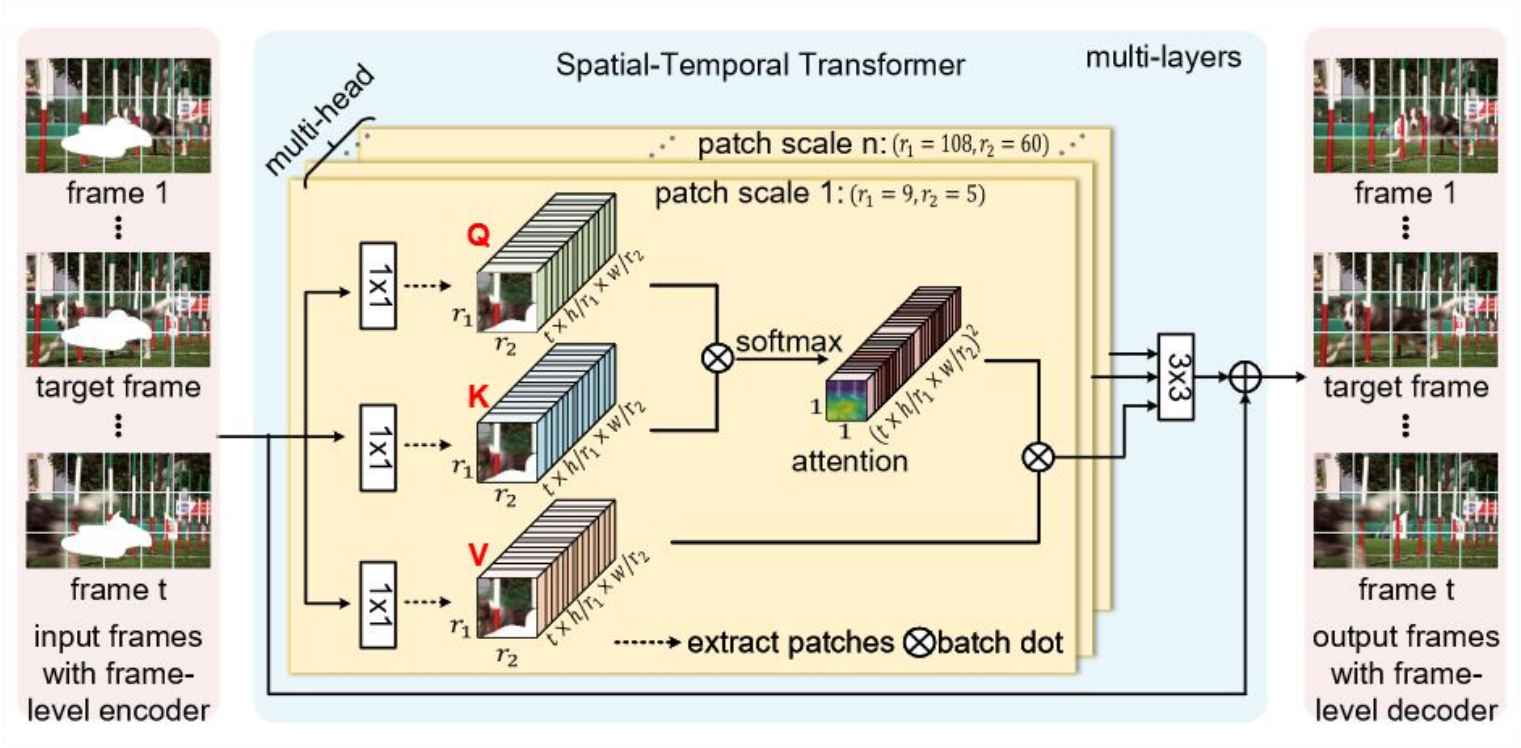
Source: [17] **Girdhar, R.** et al. "*Video Action Transformer Network*". In CVPR

Perceiver



Source: [14] **Jaegle, A.** et al. "*Perceiver: General Perception with Iterative Attention*". In ICML (2021).

Inpainting

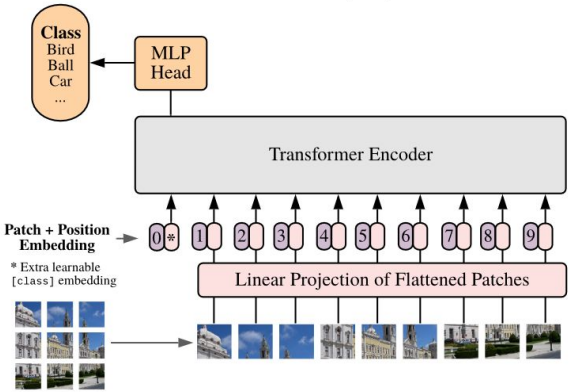


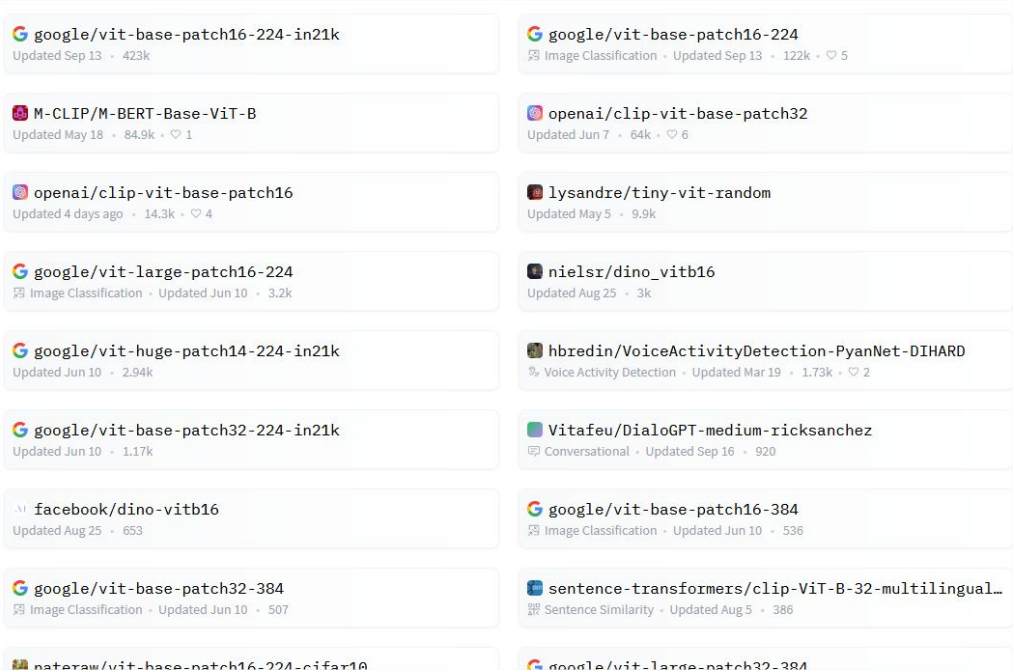
Source: [18] **Zeng, Y.** et al. "*Learning joint spatial-temporal transformations for video inpainting*". In ECCV (2020).

How to Transformer?

Use pre-trained models! GitHub!

https://colab.research.google.com/





https://huggingface.co/models?sort=downloads&search=vit

References

Publications

[1] **Bahdanau, D.** et al. "*Neural machine translation by jointly learning to align and translate"*. In CoRR (2014).
[2] **Wang, X.** et al. "*Non-local neural networks.*" In CVPR (2018).
[3] **Vaswani, A.** et al. "*Attention is all you need*". In NeurIPS (2017)
[4] **He, K.** et al. "*Deep residual learning for image recognition*". In CVPR (2016).
[5] **Dosovitskiy, A.** et al. "*An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*". In ICLR (2020).
[6] **Lee, S.** et al. "*Parameter Efficient Multimodal Transformers for Video Representation Learning*". In ICLR (2020).
[7] **Atito, S.** et al. "*Sit: Self-supervised vision transformer*". In CoRR (2021).
[8] **Li, L.** et al. "*HERO: Hierarchical Encoder for Video+ Language Omni-representation Pre-training*". In EMNLP (2020).
[9] **Oord, A. V. D.** et al. "*Representation learning with contrastive predictive coding*". In CoRR (2018).
[10] **Wang, J.** et al. "*DeepVID: Deep Visual Interpretation and Diagnosis for Image Classifiers via Knowledge Distillation*". In IEEE Tr-VCG (2019).
[11] **Caron, M.** et al. "*Emerging Properties in Self-Supervised Vision Transformers*". In CoRR (2021).
[12] **Carion, N.** et al. "*End-to-end object detection with transformers*". In ECCV (2020).
[13] **Lin, K.** et al. "*End-to-end human pose and mesh reconstruction with transformers*". In CVPR (2021).
[14] **Jaegle, A.** et al. "*Perceiver: General Perception with Iterative Attention*". In ICML (2021).
[15] **Plizzari, C.** et al. "*Spatial temporal transformer network for skeleton-based action recognition*". In ICPR (2021).
[16] **Wang, Y.** et al. "*End-to-end video instance segmentation with transformers*". In CVPR (2021).
[17] **Girdhar, R.** et al. "*Video Action Transformer Network*". In CVPR
[18] **Zeng, Y.** et al. "*Learning joint spatial-temporal transformations for video inpainting*". In ECCV (2020).
[19] **Parmar, N.** et al. "*Image transformer*". In ICML (2018).
[20] **Child, R.** et al. "*Generating long sequences with sparse transformers*". In CoRR (2019).
[21] **Bertasius, G.** et al. "*Is Space-Time Attention All You Need for Video Understanding?*". In CoRR (2021).
[22] **Huang, Z.** et al. "*Ccnet: Criss-cross attention for semantic segmentation*". In ICCV (2019).
[23] **Arnab, A.** et al. "*Vivit: A video vision transformer*". In CoRR (2021).
[24] **Ging, S.** et al. "*Coot: Cooperative hierarchical transformer for video-text representation learning*". In NeurIPS (2020).

Blog Posts

Vincent Dumoulin, Francesco Visin "A guide to convolution arithmetic for deep learning" (2016) https://github.com/vdumoulin/conv_arithmetic
Mohammed Terry-Jack *"Deep Learning: The Transformer"* (2019) https://medium.com/@b.terryjack/deep-learning-the-transformer-9ae5e9c5a190
Lilian Weng "*Attention? Attention!*" (2018) https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html
Amirhossein Kazemnejad "*Transformer Architecture: The Positional Encoding*" (2019) https://kazemnejad.com/blog/transformer_architecture_positional_encoding/
Gillaume Klein et al. "*The Annotated Transformer*" (2018) http://nlp.seas.harvard.edu/2018/04/03/attention.html
Lilian Weng "*The Transformer Family*" (2020) https://lilianweng.github.io/lil-log/2020/04/07/the-transformer-family.html
Jay Alammar "*The Illustrated Transformer*" (2018) https://jalammar.github.io/illustrated-transformer/