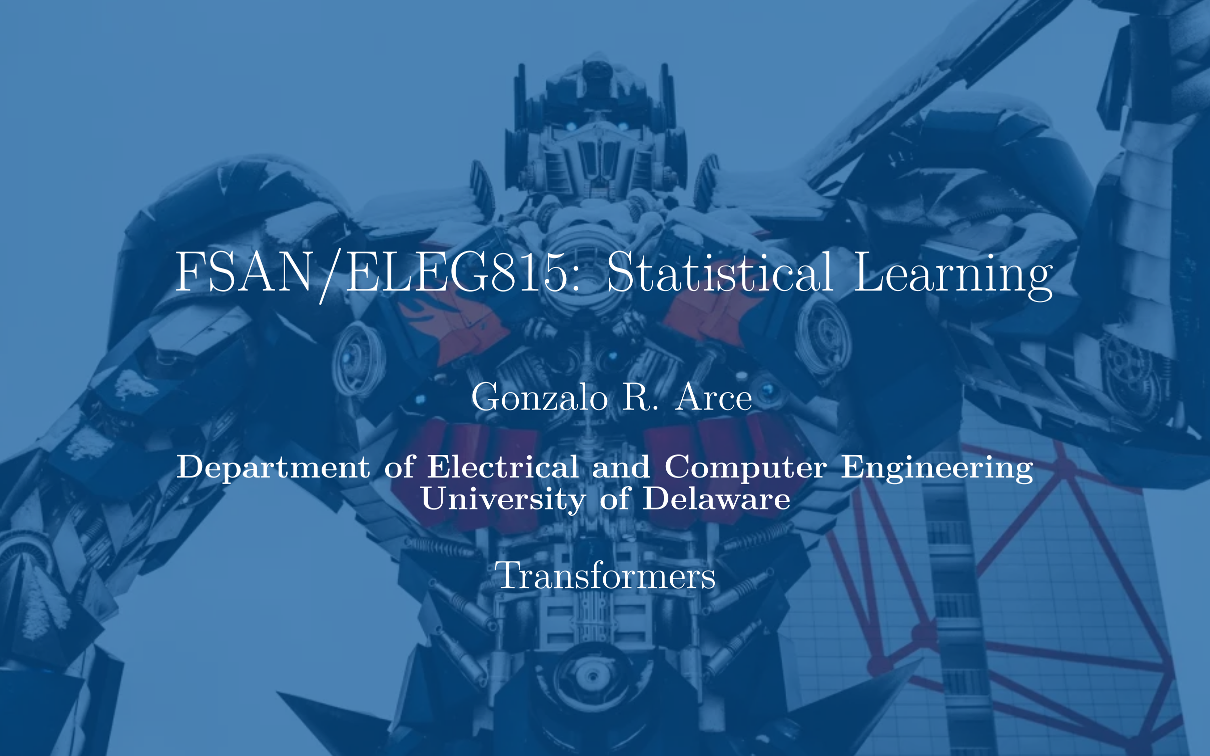


FSAN/ELEG815: Statistical Learning

Gonzalo R. Arce

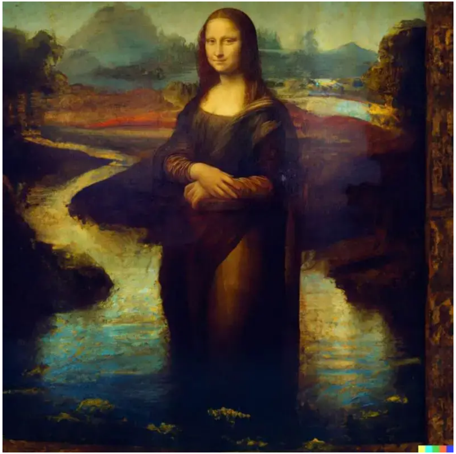
Department of Electrical and Computer Engineering
University of Delaware

Transformers

Transformers - Revolutionary Architecture

- ChatGPT is based on the GPT (Generative Pretrained Transformer) architecture.
- Introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017.
- Excel in NLP and Imaging tasks thanks to their capacity to incorporate extensive context. Outperforms in image classification, segmentation, and machine translation.
- The name "transformer" reflects the ability to seamlessly *transform* one sequence of data into another, thanks to its sophisticated self-attention mechanisms.

Transformers - Image Generation



Tokens and Input Tokenizing

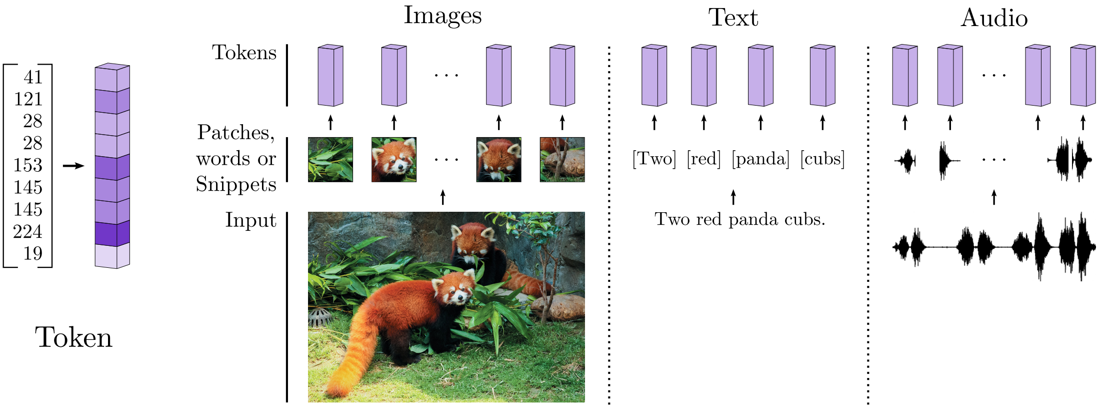
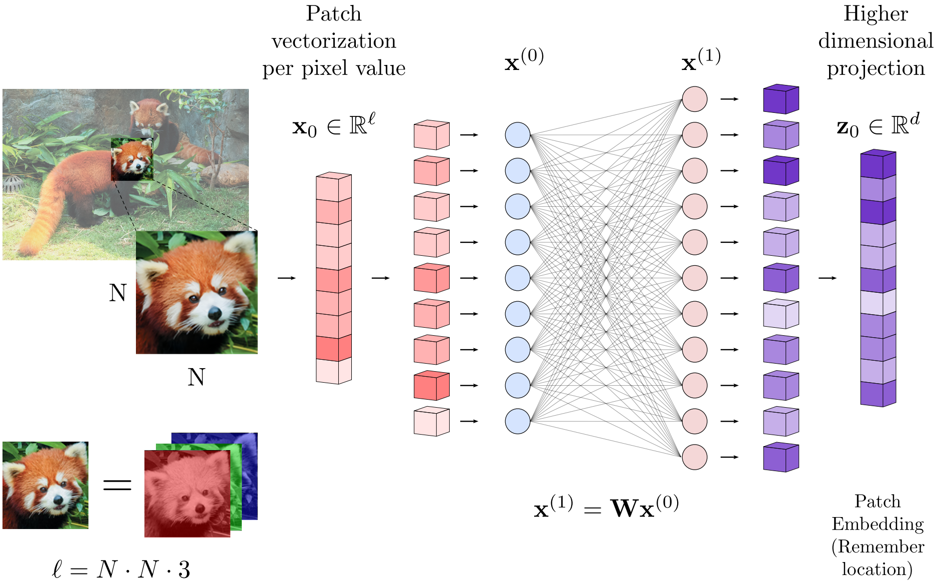
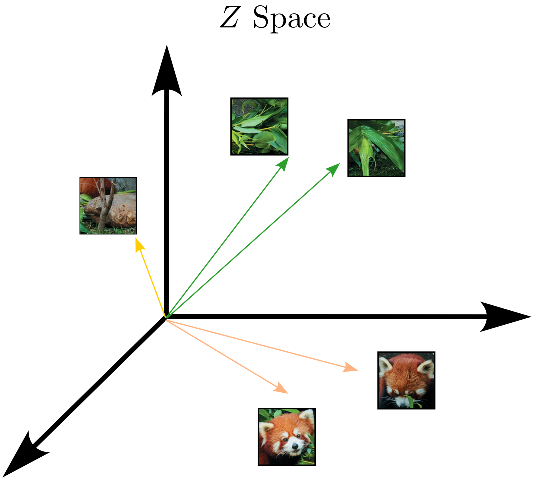


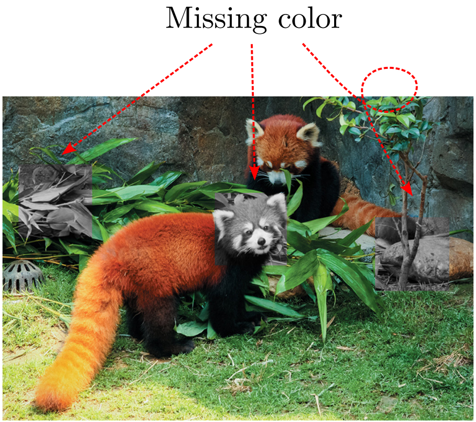
Image Tokenization and Linear Transformation

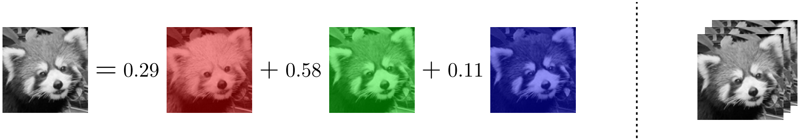


Patch Embedding Has Meaning

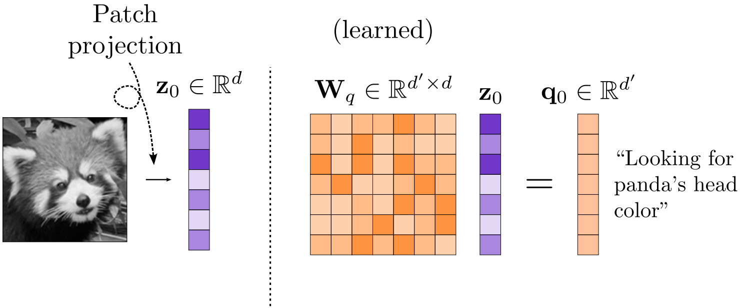


Coloring Problem



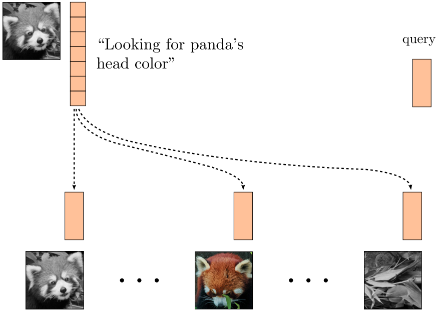


Coloring Problem - Query Vector

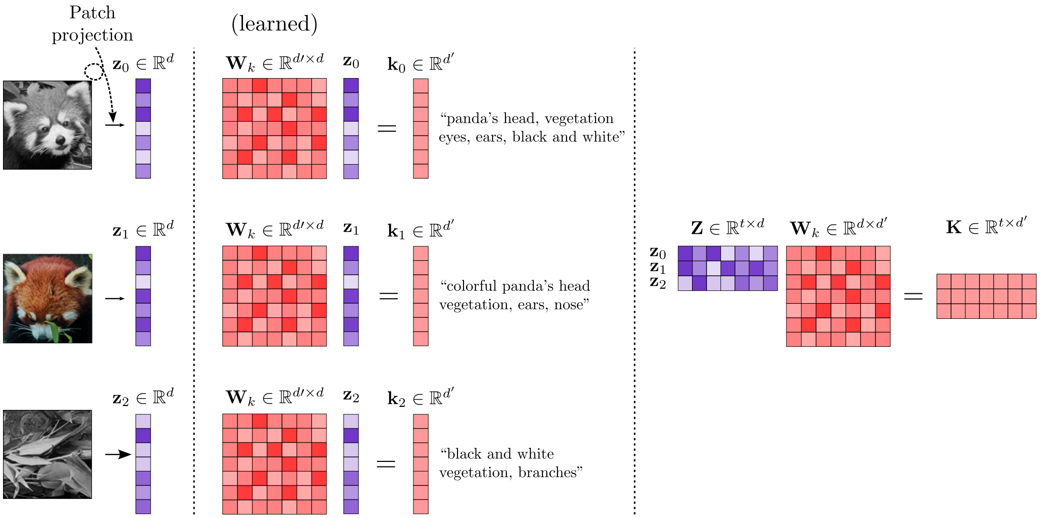


In general, $d' \geq d$

Coloring Problem



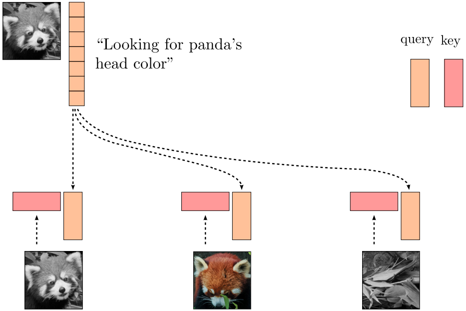
Coloring Problem - Key Matrix



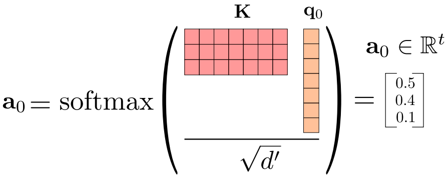
$$\mathbf{k}_i = \mathbf{W}_k \mathbf{z}_i$$

$$\mathbf{K} = \mathbf{Z}\mathbf{W}_k$$

Coloring Problem



Coloring Problem - Self-Attention

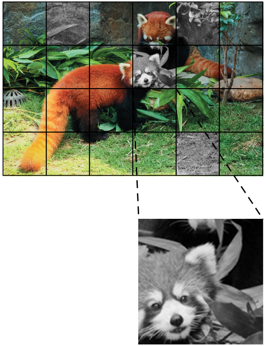


$$\mathbf{a}_0 = \text{softmax}\left(\frac{\mathbf{K}^T \mathbf{q}_0}{\sqrt{d'}}\right)$$

$\sqrt{d'}$ Scales the dot product for numerical stability on the softmax function and balancing signal magnitudes with respect to the dimensionality

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{t} e^{x_j}}$$

Coloring Problem - Self-Attention Heat Map

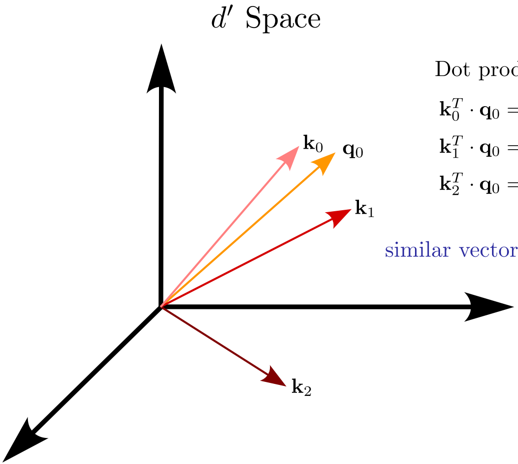


Attention to each patch

To generate vegetation's color

To generate panda's color

Why Dot Product for Similarity?



Dot product

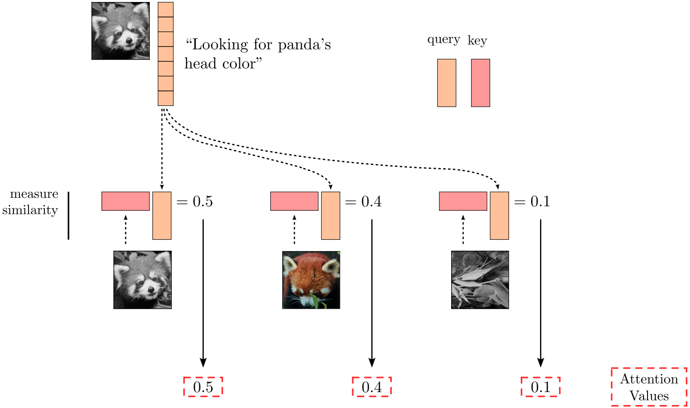
$\mathbf{k}_0^T \cdot \mathbf{q}_0 = 35$

$\mathbf{k}_1^T \cdot \mathbf{q}_0 = 22$

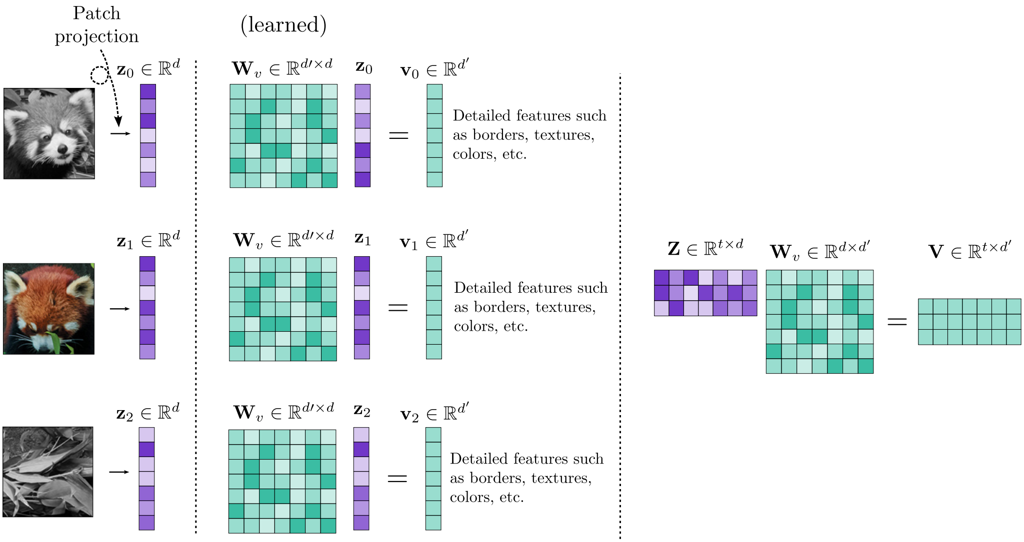
$\mathbf{k}_2^T \cdot \mathbf{q}_0 = 4$

similar vectors $\rightarrow$ larger dot product

Coloring Problem



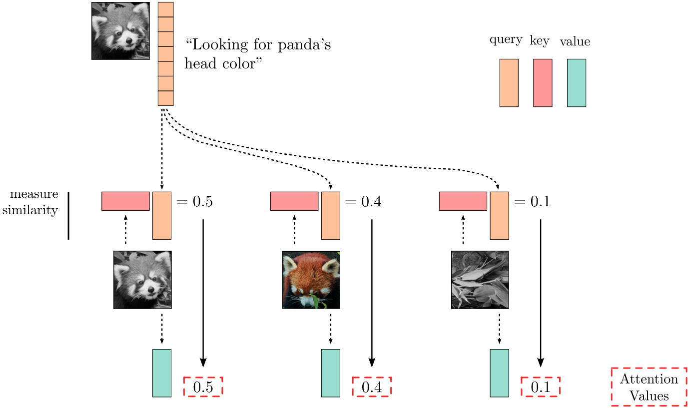
Coloring Problem - Value Matrix



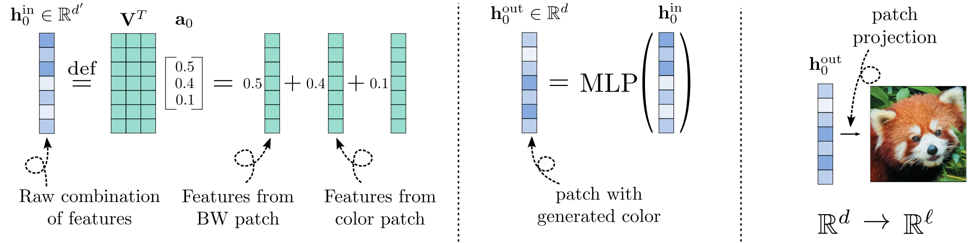
$$\mathbf{k}_i = \mathbf{W}_k \mathbf{z}_i$$

$$\mathbf{V} = \mathbf{Z}\mathbf{W}_v$$

Coloring Problem



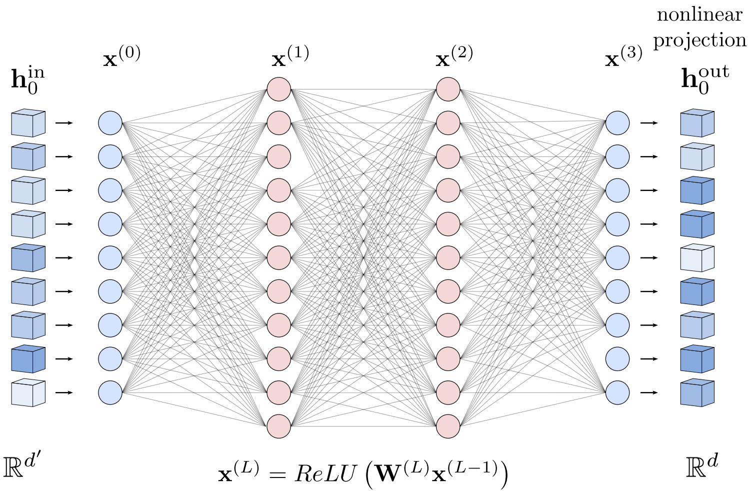
Coloring Problem - Hidden Representation



$$\mathbf{h}_0^{\text{in}} = \mathbf{V}^{\mathbf{T}}\mathbf{a}_a$$

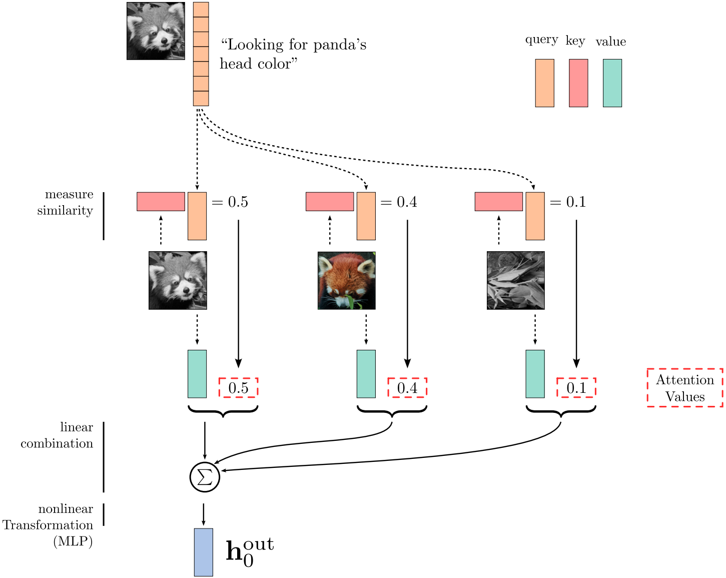
$$\mathbf{h}_0^{\text{out}} = \text{MLP}(\mathbf{h}_0^{\text{in}})$$

Nonlinear Transformation of Tokens - MLP

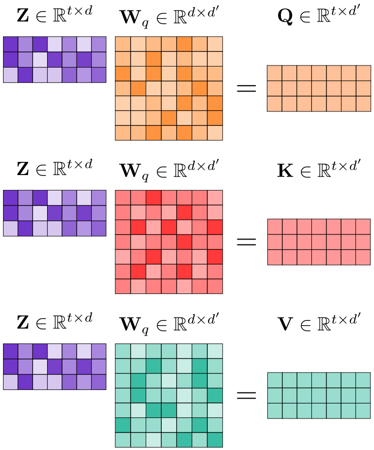


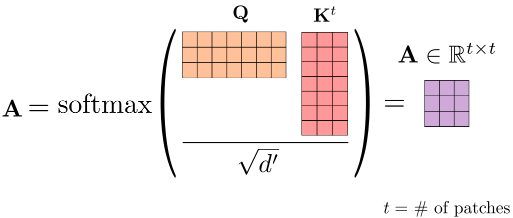
$$\mathbf{x}^{(L)} = ReLU\left(\mathbf{W}^{(L)}\mathbf{x}^{(L-1)}\right)$$

Coloring Problem



Coloring Problem - $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V}$ Matrices

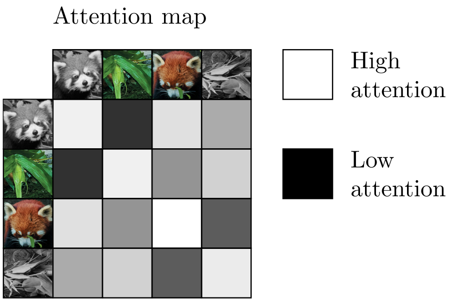


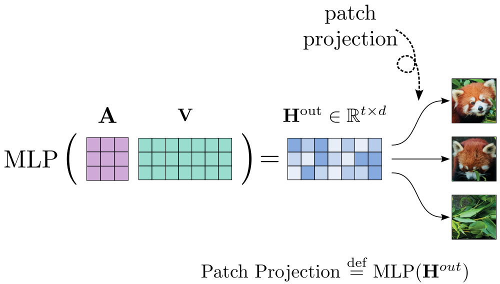


$\mathbf{Q} = \mathbf{Z}\mathbf{W}_q \quad \mathbf{K} = \mathbf{Z}\mathbf{W}_k \quad \mathbf{V} = \mathbf{Z}\mathbf{W}_v$

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}^T\mathbf{K}}{\sqrt{d'}}\right)$$

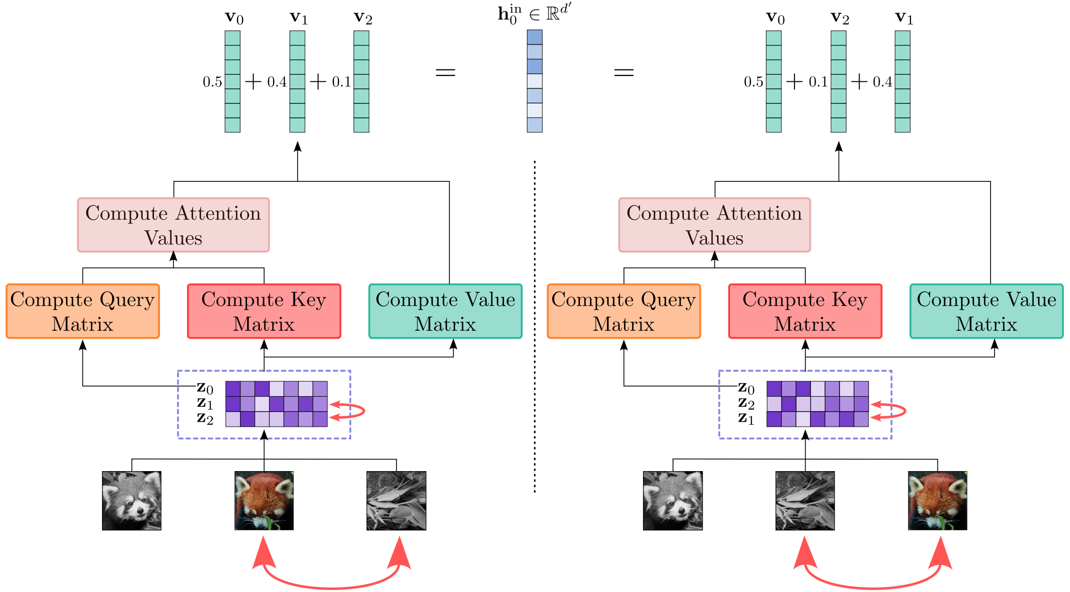
Coloring Problem - Attention Map





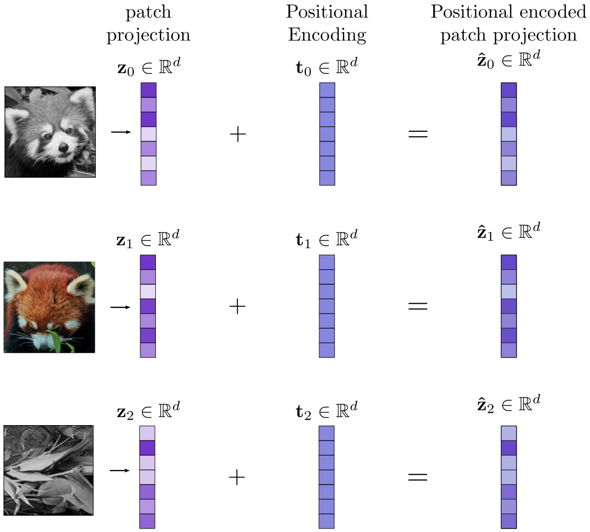
$$\mathbf{H}^{\text{out}} = \text{MLP}(\mathbf{AV})$$

Position in the Matrix Has no Influence



Positional Embedding - Encode Position

- Solution: Add a positional vector to each patch projection.



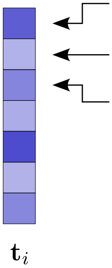
Positional embedding - Sinusoidal Waves

$$PE(i, 2j) = \sin\left(\frac{i}{10000^{2j/d}}\right)$$

$$PE(i, 2j+1) = \cos\left(\frac{i}{10000^{2j/d}}\right)$$

where i is the position of the patch in the sequence, j for $j = 0, \dots, d$ is the dimension within the positional vector and d is the size of the positional vector.

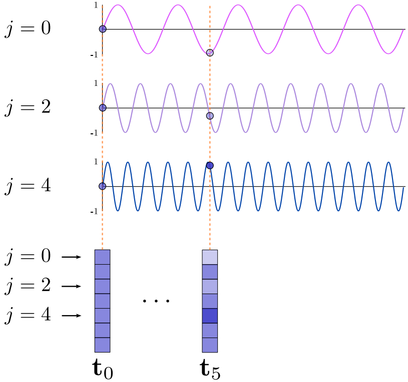
For each dimension of $\mathbf{t}_i$, there is a sinusoidal wave with different frequency



$$\mathbf{t}_{i,0} = \sin\left(\frac{i}{10000^{2(0)/d}}\right)$$

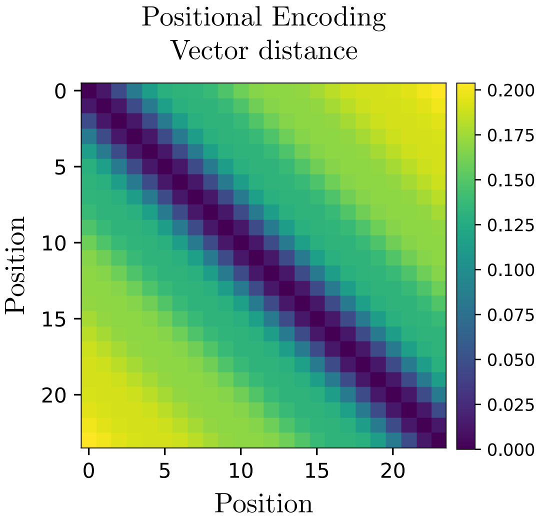
$$\mathbf{t}_{i,1} = \cos\left(\frac{i}{10000^{2(0)/d}}\right)$$

$$\mathbf{t}_{i,2} = \sin\left(\frac{i}{10000^{2(1)/d}}\right)$$

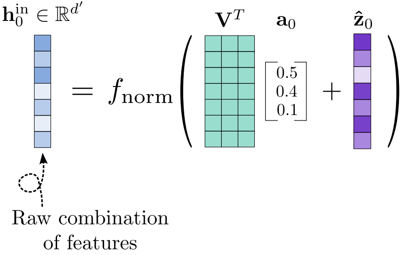


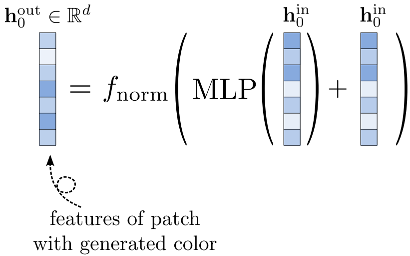
Positional embedding - Sinusoidal Waves

- This function provides a large number of vectors with a constant distance between them, i.e. $\|\mathbf{t}_i - \mathbf{t}_{i+1}\|_2$ is constant for all i.



Attention Layer - Normalization

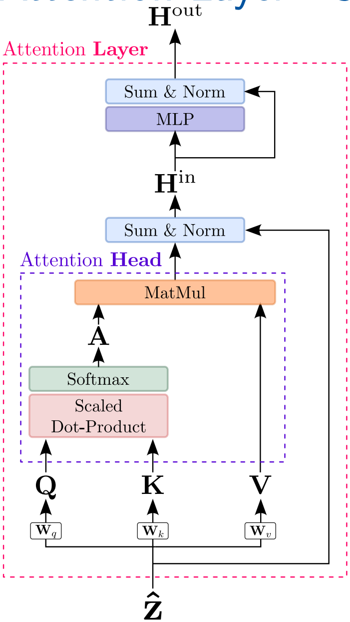


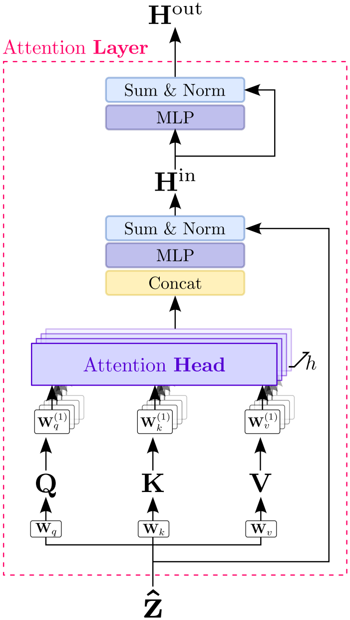


$$\mathbf{h}_0^{\text{in}} = f_{\text{norm}}(\mathbf{V}^T \mathbf{a}_0 + \hat{\mathbf{z}}_0)$$

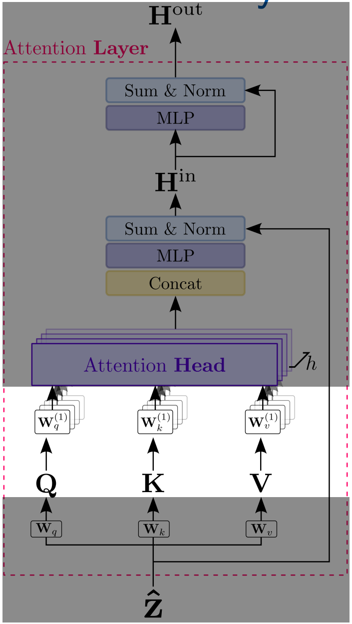
$$\mathbf{h}_0^{\text{out}} = f_{\text{norm}}(\text{MLP}(\mathbf{h}_0^{\text{in}}) + \mathbf{h}_0^{\text{in}})$$

Single Attention Layer - Single-Head and Multi Head

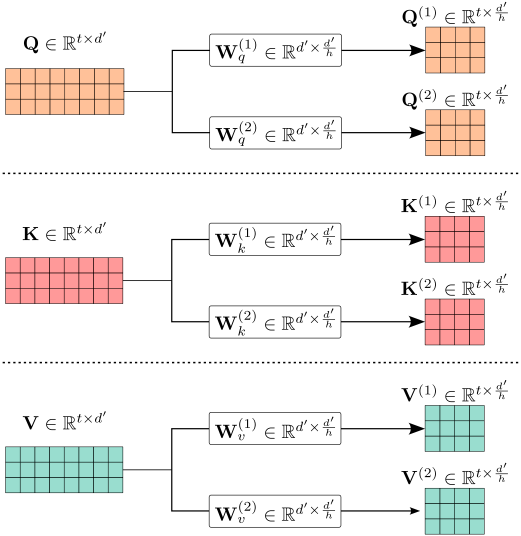




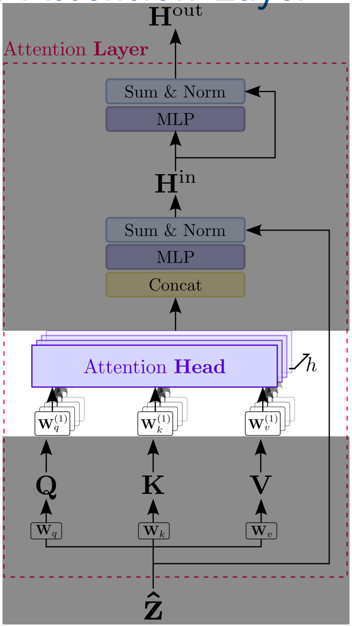
Single Attention Layer - Q, K and V split



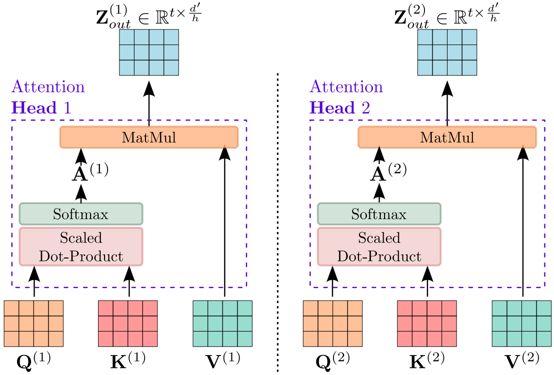
for $h = 2$ (two heads)



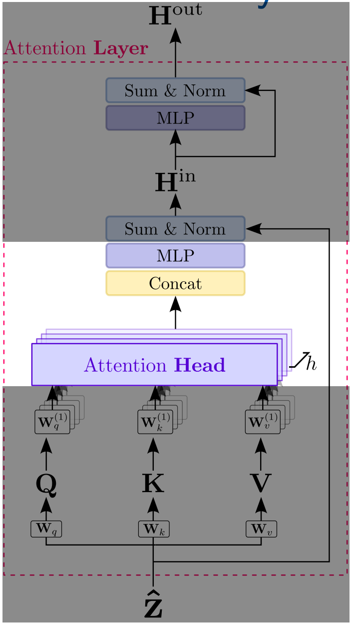
Single Attention Layer - Multi-Head Attention



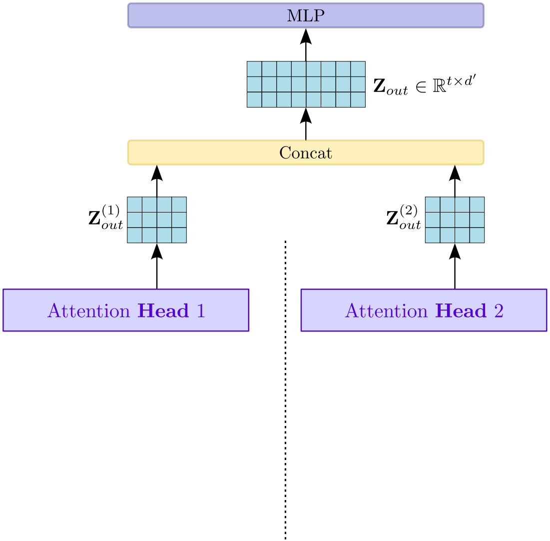
for $h = 2$ (two heads)



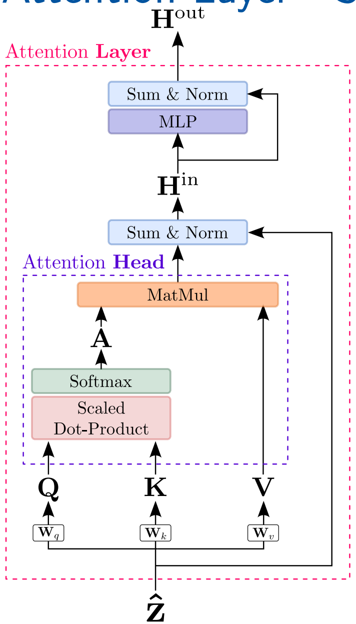
Single Attention Layer - Multi-Head Concat

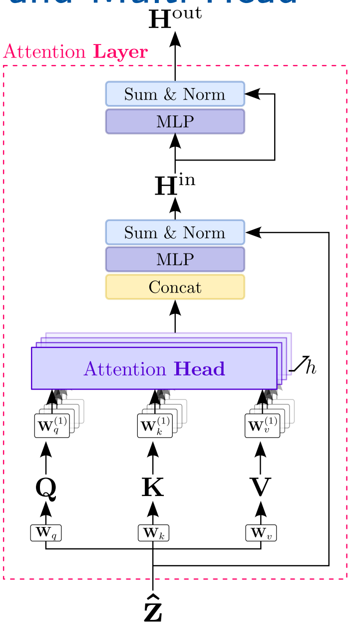


for $h = 2$ (two heads)

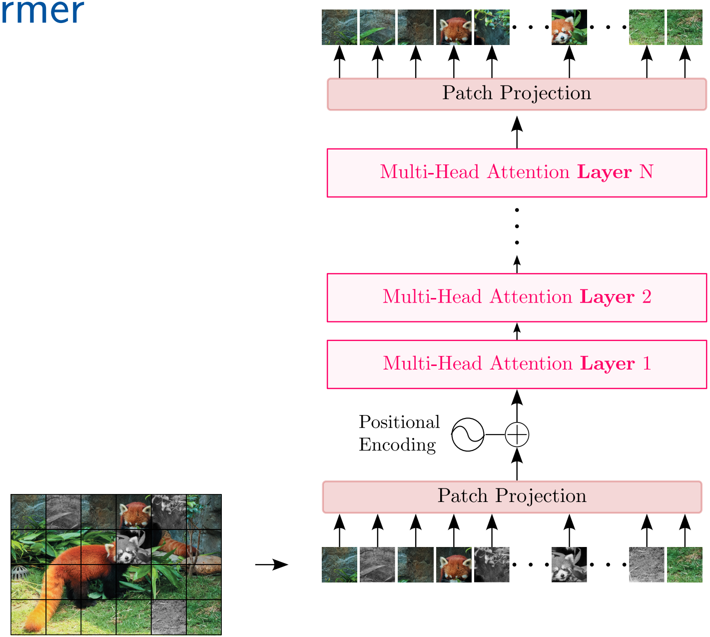


Single Attention Layer - Single Head and Multi Head

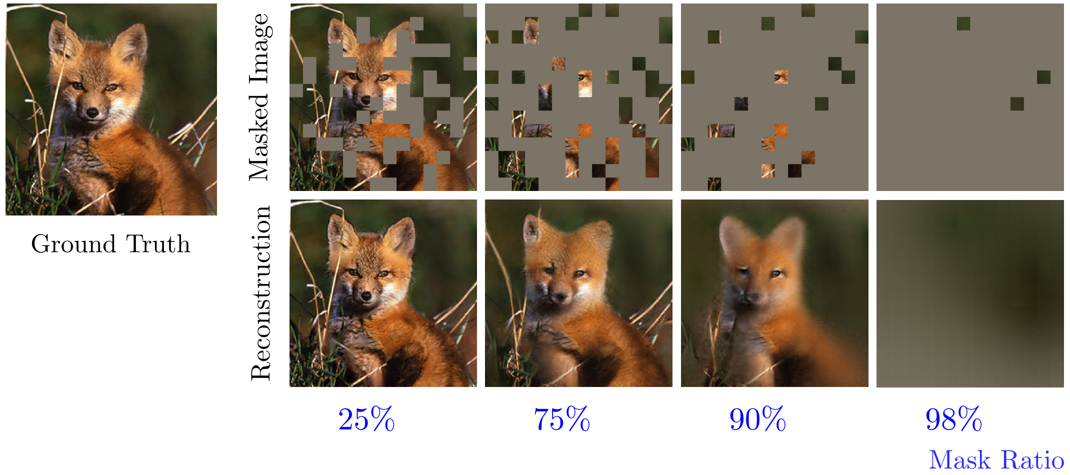




Transformer



Vision Transformer - Reconstructing Masked Image



Vision Transformer - Reconstructing Masked Image