Attention Augmented Convolutional Networks

Paper Details

- Paper Title: Attention Augmented Convolutional Networks
- Publication Date: International Conference on Computer Vision (ICCV), 2019
- Publisher: Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens
- Affiliation: Google Brain

Objectives

Image Classifications

simpler task than that from DERT

Motivation

- Convolutional Neural Networks
  - locality via a limited receptive field + translation equivariance via weight sharing
  - Not capture the global contexts
- Self-attention

Motivation

- Convolutional Neural Networks
  - locality via a limited receptive field + translation equivariance via weight sharing
  - Not capture the global contexts
- Self-attention

Architechture

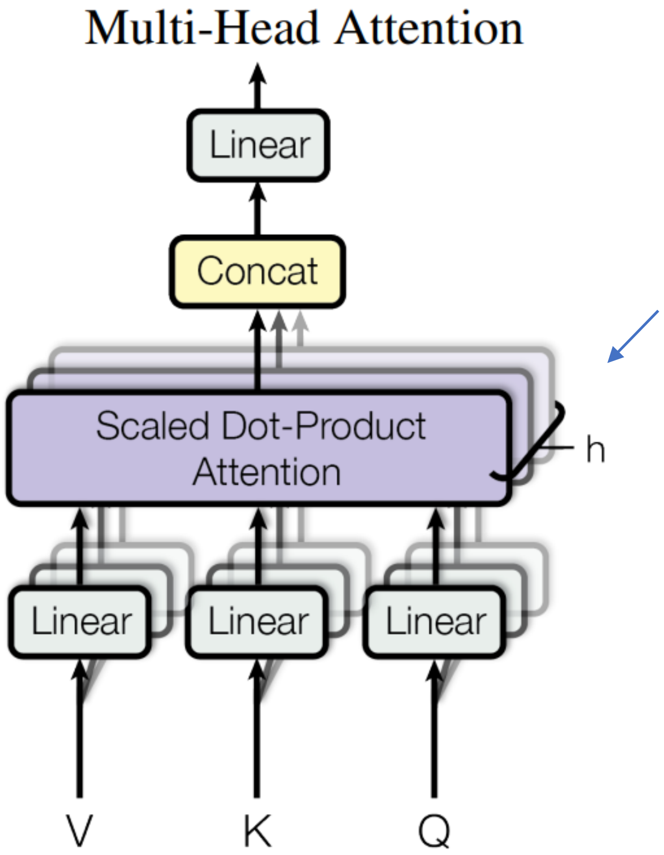
- Self-attention over images

tensor

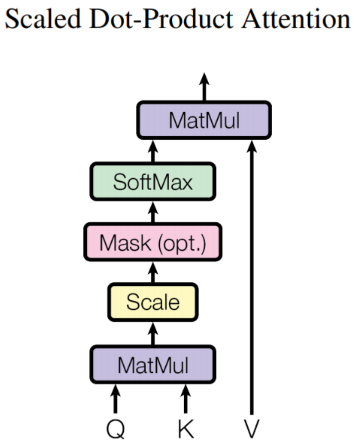
(H, W, F_{in})

flatten

$\mathbb{R}^{HW \times F_{in}}$



multihead attention



Architechture

- Self-attention over images

$$O_h = \text{Softmax}\left(\frac{(XW_q)(XW_k)^T}{\sqrt{d_k^h}}\right)(XW_v) \Rightarrow \text{MHA}(X) = \text{Concat}\left[O_1, \ldots, O_{Nh}\right]W^O$$

$$W^O \in \mathbb{R}^{d_v \times d_v}$$

Learned linear Transformation $W_q, W_k \in \mathbb{R}^{F_{in} \times d_k^h}$ $W_v \in \mathbb{R}^{F_{in} \times d_v^h}$

$Q = XW_q,\ K = XW_k\ \ V = XW_v.$

reshape $\Downarrow$

$$(H, W, d_v)$$

Architechture

- Self-attention over images
  - Two-dimensional Positional Encodings

Absolute Encoding

Relative Encoding

Architecture	Position Encodings	top-1	top-5
AA-ResNet-34	None	74.4	91.9
AA-ResNet-34	2d Sine	74.4	92.0
AA-ResNet-34	CoordConv	74.4	92.0
AA-ResNet-34	Relative (ours)	74.7	92.0
AA-ResNet-50	None	77.5	93.7
AA-ResNet-50	2d Sine	77.5	93.7
AA-ResNet-50	CoordConv	77.5	93.8
AA-ResNet-50	Relative (ours)	77.7	93.8

Position Encodings	mAP_{COCO}	mAP_{50}	mAP_{75}
None	37.7	56.0	40.2
CoordConv [29]	37.4	55.5	40.1
Relative (ours)	38.2	56.5	40.7

Architechture

- Self-attention over images
  - Two-dimensional Positional Encodings

Absolute Encoding

Relative Encoding

how much pixel $i = (i_x, i_y)$ attends to pixel $j = (j_x, j_y)$

$$l_{i,j} = \frac{q_i^T}{\sqrt{d_k^h}}(k_j + r^W_{j_x - i_x} + r^H_{j_y - i_y})$$

where q_i is the query vector for pixel i (the i-th row of Q), k_j is the key vector for pixel j (the j-th row of K) and $r^W_{j_x - i_x}$ and $r^H_{j_y - i_y}$ are learned embeddings for relative width $j_x - i_x$ and relative height $j_y - i_y$, respectively.

Architechture

- Self-attention over images
  - Two-dimensional Positional Encodings

Absolute Encoding

how much pixel $i = (i_x, i_y)$ attends to pixel $j = (j_x, j_y)$

$$l_{i,j} = \frac{q_i^T}{\sqrt{d_k^h}}(k_j + r^W_{j_x - i_x} + r^H_{j_y - i_y})$$

Relative Encoding

$$O_h = \text{Softmax}\left(\frac{(XW_q)(XW_k)^T}{\sqrt{d_k^h}}\right)(XW_v) \qquad O_h = \text{Softmax}\left(\frac{QK^T + S_H^{rel} + S_W^{rel}}{\sqrt{d_k^h}}\right)V$$

Architechture

- Self-attention over images
  - Two-dimensional Positional Encodings

Absolute Encoding

$$O_h = \text{Softmax}\left(\frac{(XW_q)(XW_k)^T}{\sqrt{d_k^h}}\right)(XW_v)$$

Relative Encoding

$$O_h = \text{Softmax}\left(\frac{QK^T + S_H^{rel} + S_W^{rel}}{\sqrt{d_k^h}}\right)V$$

where $S_H^{rel}, S_W^{rel} \in \mathbb{R}^{HW \times HW}$ are matrices of relative position logits along height and width dimensions that satisfy $S_H^{rel}[i,j] = q_i^T r_{j_y - i_y}^H$ and $S_W^{rel}[i,j] = q_i^T r_{j_x - i_x}^W$. As we consider relative height and width information separately, S_H^{rel} and S_W^{rel} also satisfy the properties $S_W^{rel}[i,j] = S_W^{rel}[i, j+W]$ and $S_H^{rel}[i,j] = S_H^{rel}[i+H, j]$, which prevents from having to compute the logits for all (i,j) pairs.

Architechture

- Self-attention over images
  - Two-dimensional Positional Encodings

Absolute Encoding

$$O_h = \text{Softmax}\left(\frac{(XW_q)(XW_k)^T}{\sqrt{d_k^h}}\right)(XW_v)$$

Relative Encoding

$$O_h = \text{Softmax}\left(\frac{QK^T + S_H^{rel} + S_W^{rel}}{\sqrt{d_k^h}}\right)V$$

$$\text{Self-Attention}(\boldsymbol{X})_{t,:} := \text{softmax}\left(\boldsymbol{A}_{t,:}\right)\boldsymbol{X}\boldsymbol{W}_{val},$$

$$\boldsymbol{A} := (\boldsymbol{X} + \boldsymbol{P})\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}(\boldsymbol{X} + \boldsymbol{P})^{\top}$$

Absolute Encoding

$$\mathbf{A}_{\boldsymbol{q},\boldsymbol{k}}^{\text{abs}} = (\mathbf{X}_{\boldsymbol{q},:} + \mathbf{P}_{\boldsymbol{q},:})W_{qry}W_{key}^{\top}(\mathbf{X}_{\boldsymbol{k},:} + \mathbf{P}_{\boldsymbol{k},:})^{\top}$$

Relative Encoding

$$\mathbf{A}_{\boldsymbol{q},\boldsymbol{k}}^{\text{rel}} := \mathbf{X}_{\boldsymbol{q},:}^{\top}W_{qry}^{\top}W_{key}\,\mathbf{X}_{\boldsymbol{k},:} + \mathbf{X}_{\boldsymbol{q},:}^{\top}W_{qry}^{\top}\widehat{W}_{key}\,\boldsymbol{r_\delta} + \boldsymbol{u}^{\top}W_{key}\,\mathbf{X}_{\boldsymbol{k},:} + \boldsymbol{v}^{\top}\widehat{W}_{key}\,\boldsymbol{r_\delta}$$

Architechture

- Self-attention over images
  - Two-dimensional Positional Encodings

Absolute Encoding

$$O_h = \mathrm{Softmax}\left(\frac{(XW_q)(XW_k)^T}{\sqrt{d_k^h}}\right)(XW_v) \qquad O_h = \mathrm{Softmax}\left(\frac{QK^T + S_H^{rel} + S_W^{rel}}{\sqrt{d_k^h}}\right)V$$

$$O((HW)^2 N_h) \qquad O((HW)^2 d_k^h)$$

Relative Encoding

$$O(HW d_k^h)$$

Music transformer. In Advances in Neural Processing Systems, 2018

https://arxiv.org/pdf/1809.04281.pdf

Architecture

- Self-attention over images
- Attention Augmented Convolution
  - use an attention mechanism that can attend jointly to spatial and feature subspaces
  - introduce additional feature maps rather than refining them

Architecture

- Self-attention over images
- Attention Augmented Convolution
  - Concatening convolutional and attentional feature maps

$$\text{AAConv}(X) = \text{Concat}\Big[\text{Conv}(X), \text{MHA}(X)\Big]$$

Architecture

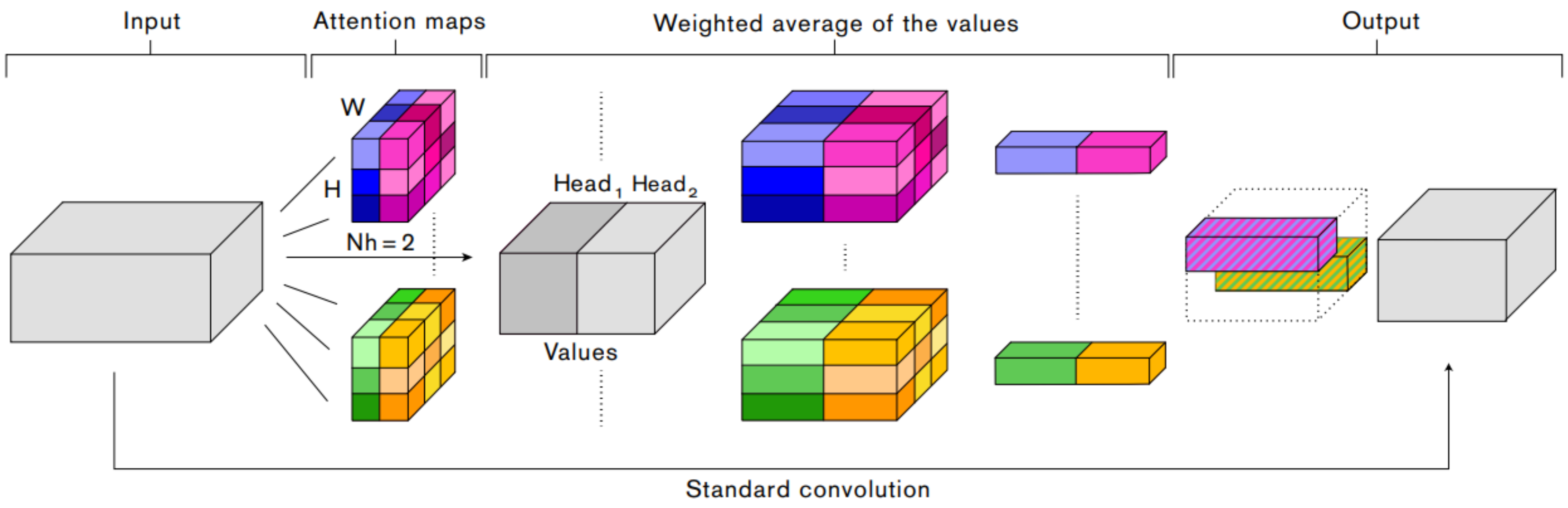
- Self-attention over images
- Attention Augmented Convolution
  - Effect on number of parameters

$$\text{AAConv}(X) = \text{Concat}\Big[\text{Conv}(X), \text{MHA}(X)\Big]$$

Overall Architecture

(H, W, F_{in})

$\mathbb{R}^{HW \times F_{in}}$

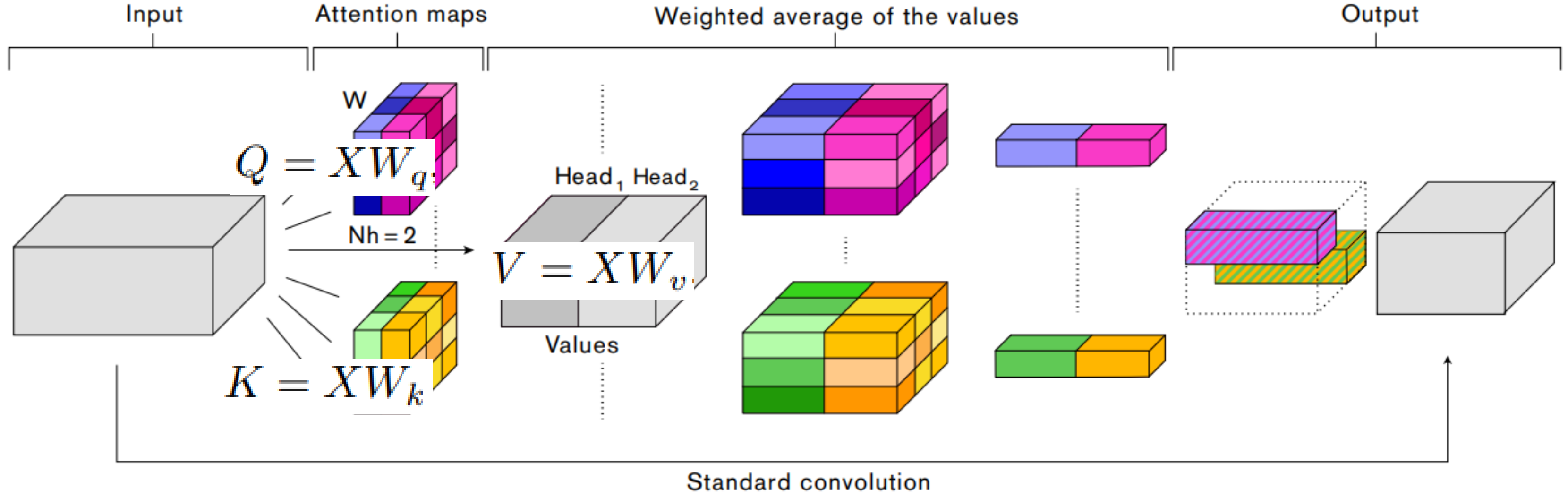


Overall Architecture

$$O_h = \text{Softmax}\left(\frac{(XW_q)(XW_k)^T}{\sqrt{d_k^h}}\right)(XW_v)$$

(H, W, F_{in})

$\mathbb{R}^{HW \times F_{in}}$

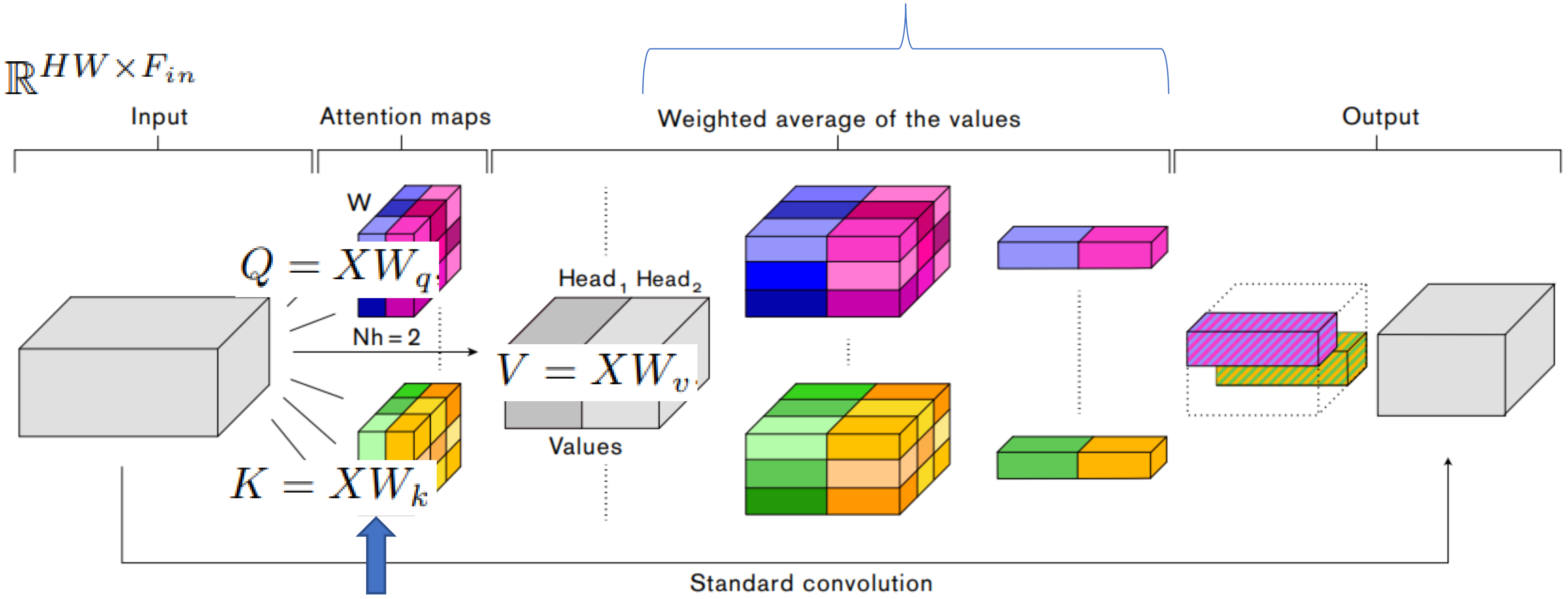


Overall Architecture

$$O_h = \text{Softmax}\left(\frac{QK^T + S_H^{rel} + S_W^{rel}}{\sqrt{d_k^h}}\right) V$$

(H, W, F_{in})

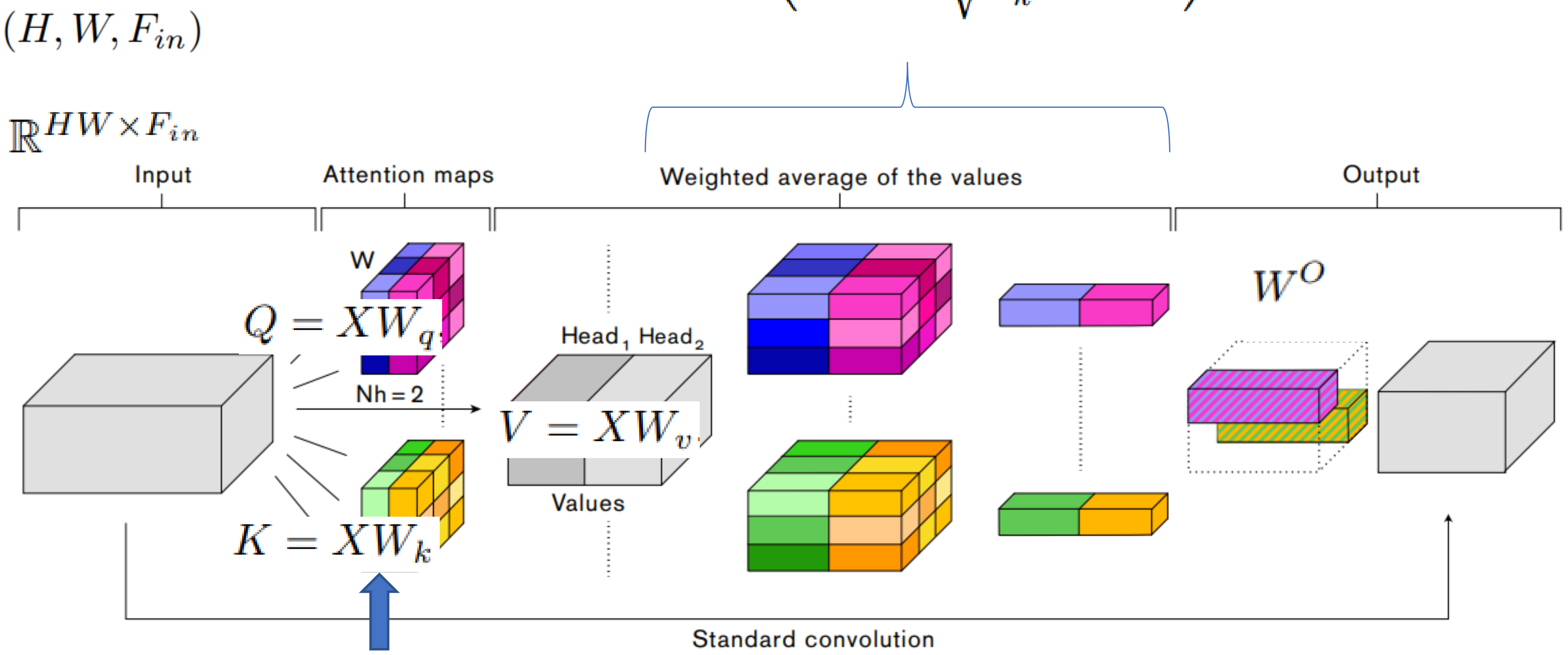
$\mathbb{R}^{HW \times F_{in}}$



Attention Maps

Overall Architecture

$$O_h = \mathrm{Softmax}\left(\frac{QK^T + S_H^{rel} + S_W^{rel}}{\sqrt{d_k^h}}\right)V$$



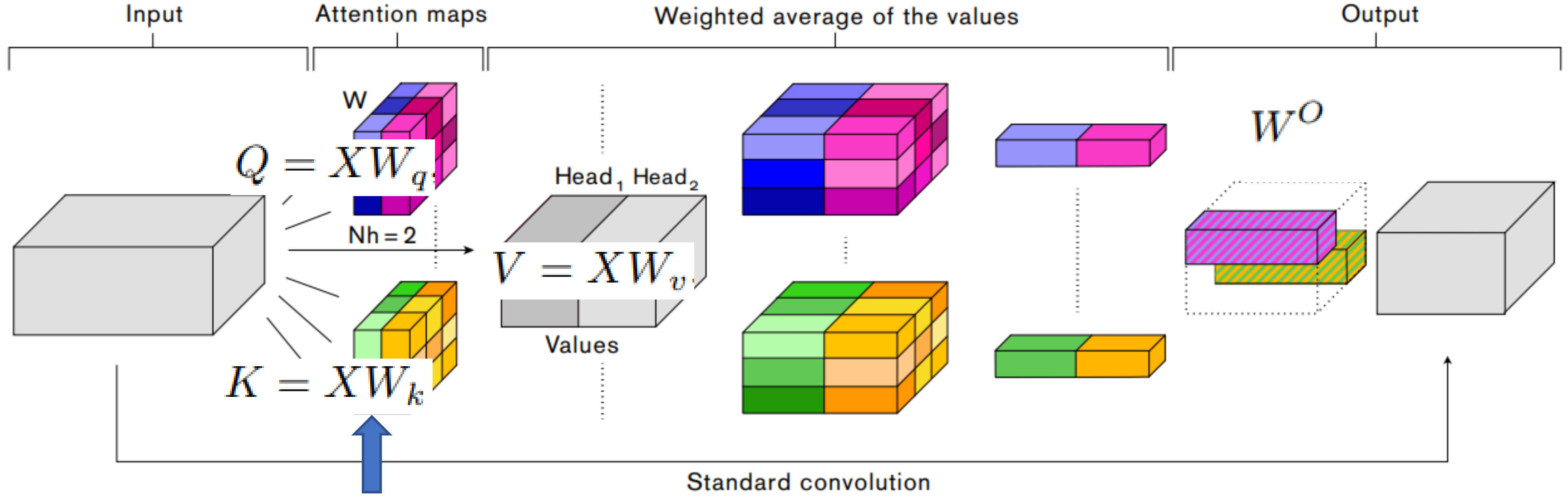
Attention Maps

Overall Architecture

$\text{MHA}(X) = \text{Concat}\Big[O_1, \ldots, O_{Nh}\Big]W^O$

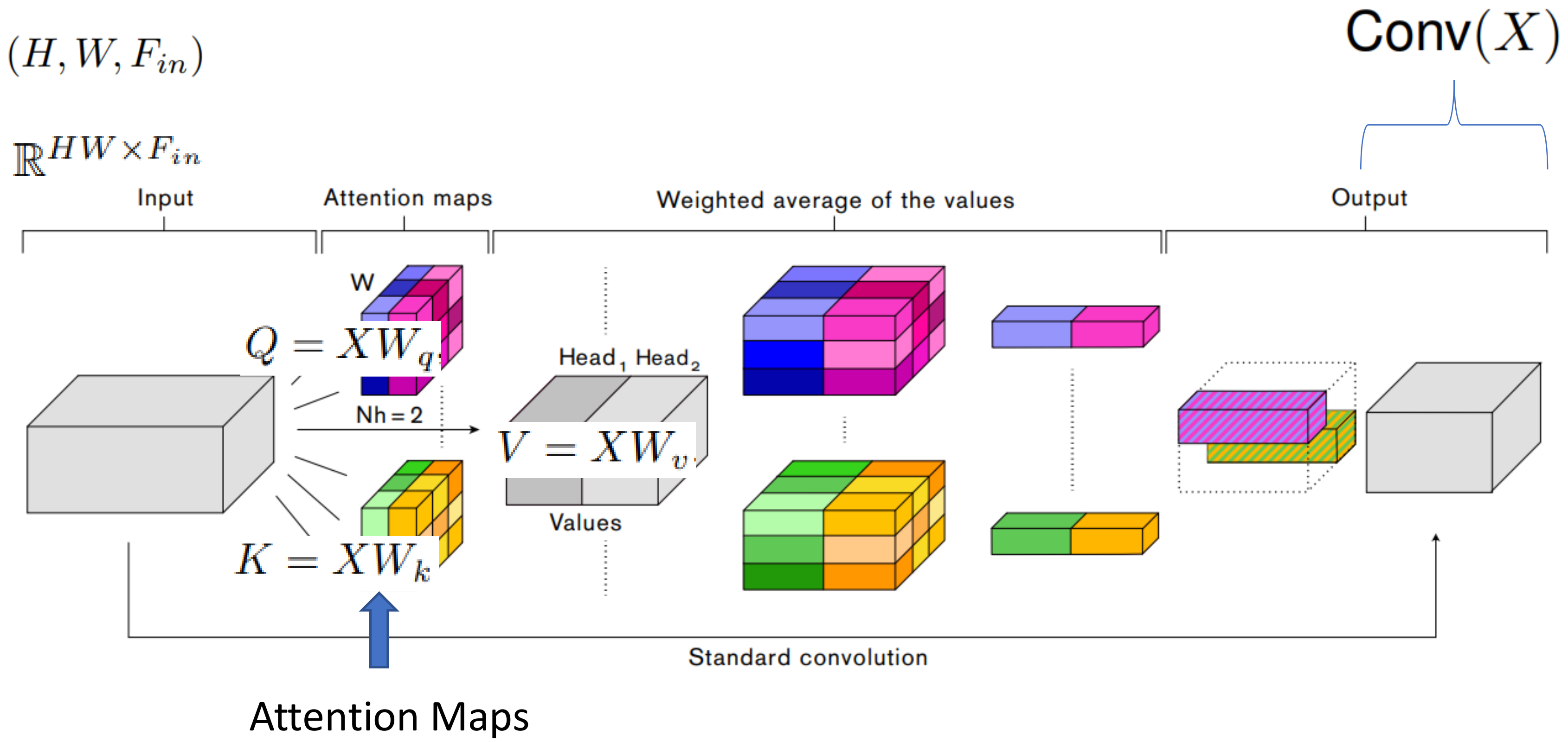
(H, W, F_{in})

$\mathbb{R}^{HW \times F_{in}}$



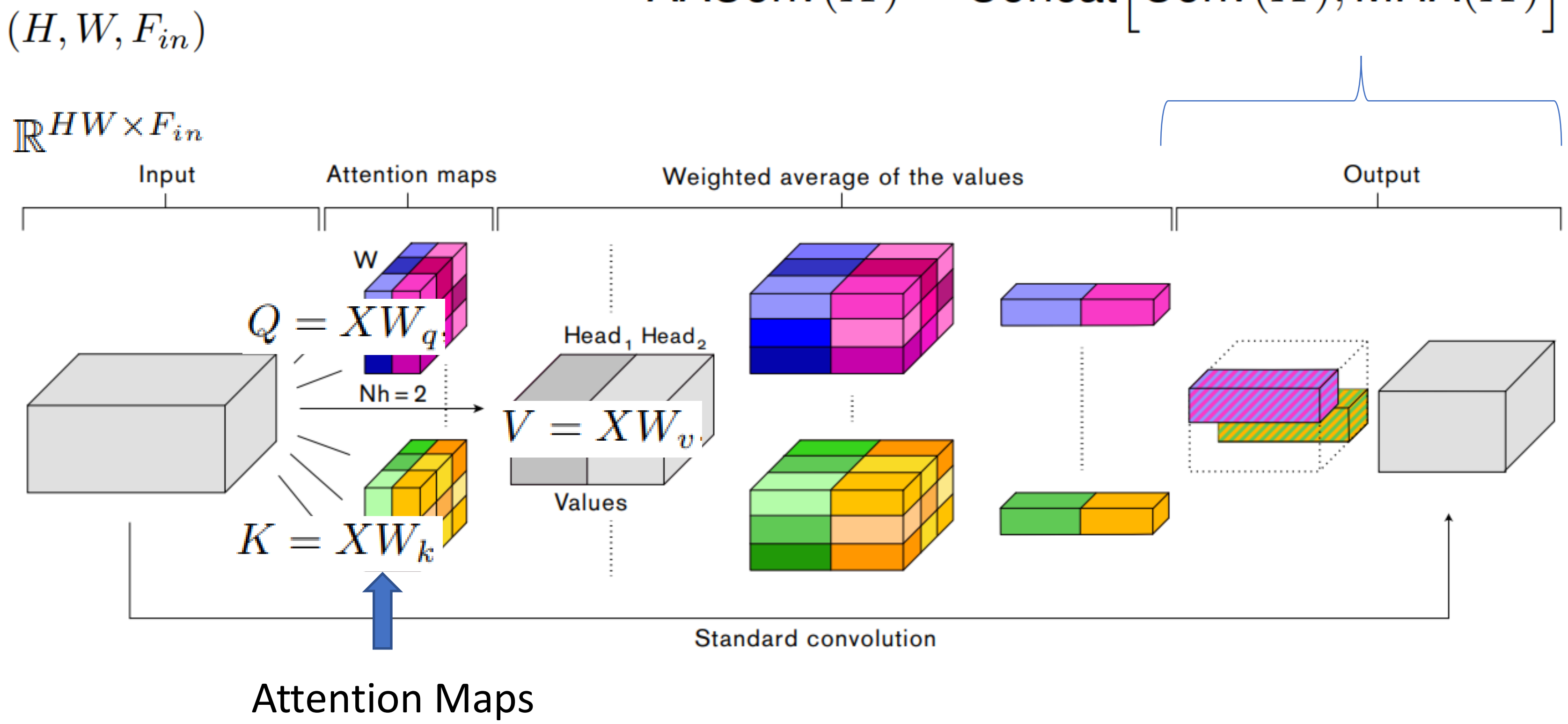
Attention Maps

Overall Architecture



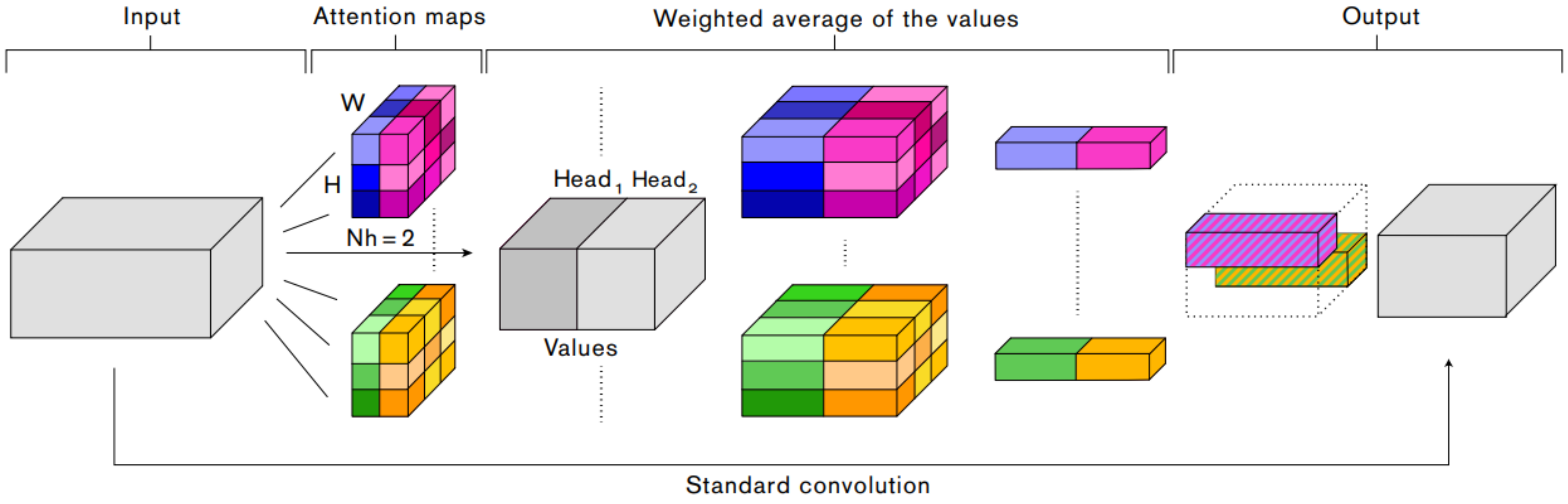
Overall Architecture

$$\mathrm{AAConv}(X) = \mathrm{Concat}\Big[\mathrm{Conv}(X), \mathrm{MHA}(X)\Big]$$



Overall Architecture (Question)

modeling tasks. The key idea behind self-attention is to produce a weighted average of values computed from hidden units. Unlike the pooling or the convolutional operator, the weights used in the weighted average operation are produced dynamically via a similarity function between hidden units. As a result, the interaction between input signals depends on the signals themselves rather than being predetermined by their relative location like in convolutions. In



Experiments

- Images classification (with ResNet, with MnasNet)
- Object detection (with Resnet)

Future work

- fully attentional regime
- how different attention mechanisms trade off computational efficiency versus representational power
- if using Attention Augmentation as a primitive in automated architecture search procedures proves useful to find even better models
- which degree fully attentional models can replace convolutional networks for visual tasks.

Questions

Effect on number of parameters: Multihead attention introduces a 1x1 convolution with F_{in} input filters and $(2d_k+d_v) = F_{out}(2\kappa+\upsilon)$ output filters to compute queries, keys and values and an additional 1x1 convolution with $d_v = F_{out}\upsilon$ input and output filters to mix the contribution of different heads. Considering the decrease in filters in the convolutional part, this leads to the following change in parameters:

$$\Delta_{params} \sim F_{in}F_{out}(2\kappa + (1 - k^2)\upsilon + \frac{F_{out}}{F_{in}}\upsilon^2), \qquad (5)$$

where we ignore the parameters introduced by relative position embeddings for simplicity as these are negligible. In practice, this causes a slight decrease in parameters when replacing 3x3 convolutions and a slight increase in parameters when replacing 1x1 convolutions. Interestingly, we find in experiments that attention augmented networks still significantly outperform their fully convolutional counterparts while using less parameters.

- Most notably, Bahdanau et al. [2] first proposed to combine attention with a Recurrent Neural Network [15] for alignment in Machine Translation. Attention was further extended by Vaswani et al. [43], where the self-attentional Transformer architecture achieved state-of-the-art results in Machine Translation. Using self-attention in cooperation with convolutions is a theme shared by recent work in Natural Language Processing [49] and Reinforcement Learning [52].

- In non-local neural networks [45], improvements are shown in video classification and object detection via the additive use of a few non-local residual blocks that employ self-attention in convolutional architectures. However, nonlocal blocks are only added to the architecture after ImageNet pretraining and are initialized in such a way that they do not break pretraining.
- In contrast, our attention augmented networks do not rely on pretraining of their fully convolutional counterparts and employ self-attention along the entire architecture.

- The use of multi-head attention allows the model to attend jointly to both spatial and feature subspaces. Additionally, we enhance the representational power of self-attention over images by extending relative self-attention [37, 18] to two dimensional inputs allowing us to model translation equivariance in a principled way. Finally our method produces additional feature maps, rather than recalibrating convolutional features via addition [45, 53] or gating [17, 16, 31, 46]. This property allows us to flexibly adjust the fraction of attentional channels and consider a spectrum of architectures, ranging from fully convolutional to fully attentional models.