Disentangled Non-Local Networks

ECCV 2020

Minghao Yin[1⋆], Zhuliang Yao[1,2⋆], Yue Cao[2], Xiu Li[1], Zheng Zhang[2], Stephen Lin[2], and Han Hu[2]

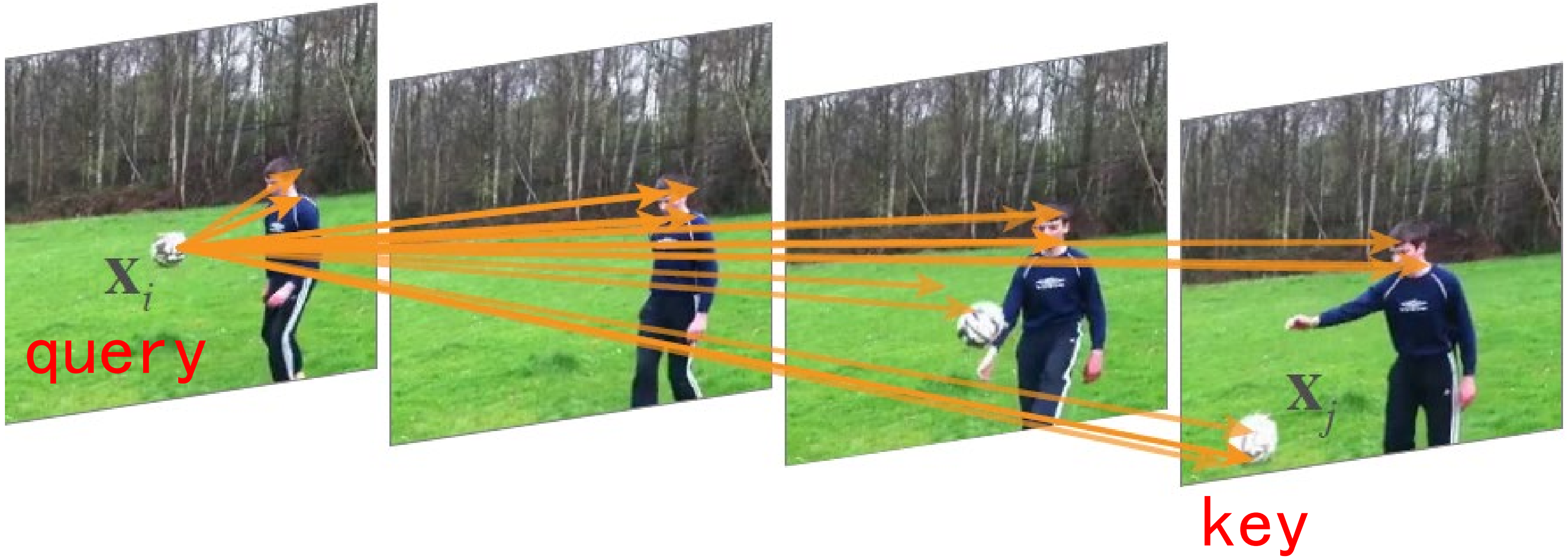
[1] Tsinghua University
{yinmh17,yzl17}@mails.tsinghua.edu.cn li.xiu@sz.tsinghua.edu.cn
[2] Microsoft Research Asia
{yuecao,zhez,stevelin,hanhu}@microsoft.com

Introduction

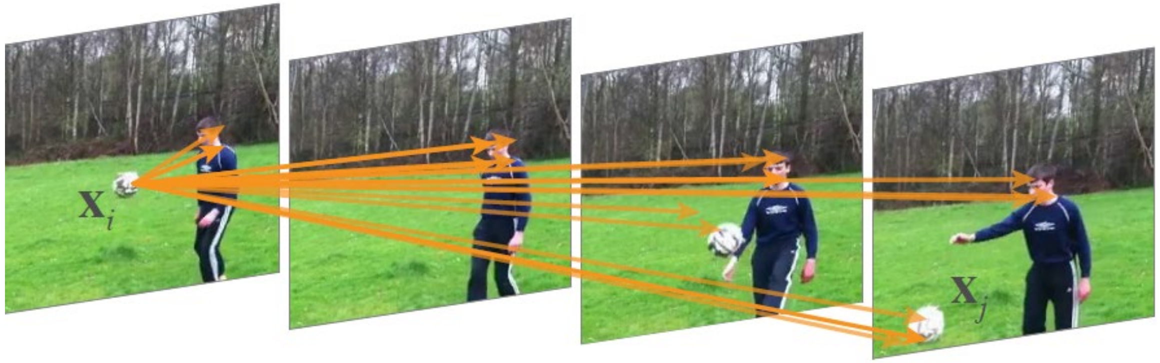
- From Non-Local to Disentangled Non-Local
- Theoretical and Experimental Analysis
- Disentangled Non-Local Block and Experiments
- Comparisons with Self Attention, Pairwise Attention, Non-Local

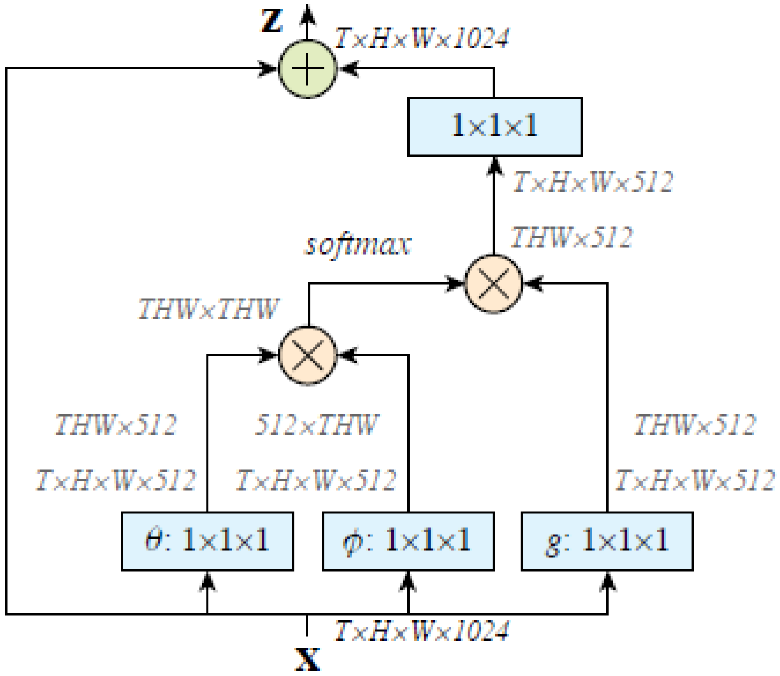
Non-Local Block

- Sequential data – Recurrent operations
- Image data – Convolutional operations
- Computationally inefficient, optimization difficulties, multi-hop dependency modelling



Non-Local Block





$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j)$$

$$z_i = W_z y_i + x_i$$

$$y_i = \frac{1}{C(\hat{x})} \sum_{\forall j} f(x_i, \hat{x}_j) g(\hat{x}_j)$$

$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

$$\theta(x_i) = W_\theta x_i, \phi(x_j) = W_\phi x_j$$

$$C(x) = \sum_{\forall j} f(x_i, x_j)$$

$$g(x_j) = W_g x_j$$

From Non-Local to Entangled Non-Local

$$y_i = \frac{1}{C(x)} \sum_{\forall j} \underline{f(x_i, x_j)} g(x_j)$$

may encode unary information as well

a pixel may have its own independent impact on all other pixels

a whitened pairwise term
accounts for impact of one pixel
specifically on another pixel

a unary term
influence of one pixel
generally over all pixels

Dividing Non-local Block

$$y_i = \sum_{j \in \Omega} \omega(x_i, x_j) g(x_j)$$

$$\omega(x_i, x_j) = \sigma(q_i^T k_j) = \frac{\exp(q_i^T k_j)}{\sum_{t \in \Omega} \exp(q_i^T k_t)}$$

$$q_i = W_q x_i, k_j = W_k x_j$$

Special case

- query vector is a constant over all image pixels, a key pixel will have global impact on all query pixels
- non-local blocks frequently degenerate into a pure unary term in several image recognition tasks where each key pixel in the image has the same similarity with all query pixels

Pure pairwise term

$$(q_i - \mu_q)^T (k_j - \mu_k)$$

$$\mu_q = \frac{1}{|\Omega|} \sum_{i \in \Omega} q_i \, , \mu_k = \frac{1}{|\Omega|} \sum_{j \in \Omega} k_j$$

- averaged query and key embedding over all pixels
- a whitened dot product between key and query
- determined by maximizing the normalized differences between query and key pixels

$$(q_i - \mu_q)^T (k_j - \mu_k)$$

$$\mu_q = \frac{1}{|\Omega|} \sum_{i \in \Omega} q_i \,, \mu_k = \frac{1}{|\Omega|} \sum_{j \in \Omega} k_j$$

Proposition 1: $\alpha^* = \frac{1}{|\Omega|} \sum_{i \in \Omega} \mathbf{q}_i$, $\beta^* = \frac{1}{|\Omega|} \sum_{m \in \Omega} \mathbf{k}_m$ is the optimal solution of the following optimization objective:

$$\begin{aligned} \arg\max_{\alpha,\beta} \quad & \frac{\sum_{i,m,n \in \Omega} \left((\mathbf{q}_i - \alpha)^T (\mathbf{k}_m - \beta) - (\mathbf{q}_i - \alpha)^T (\mathbf{k}_n - \beta) \right)^2}{\sum_{i \in \Omega} \left((\mathbf{q}_i - \alpha)^T (\mathbf{q}_i - \alpha) \right) \cdot \sum_{m,n \in \Omega} \left((\mathbf{k}_m - \mathbf{k}_n)^T (\mathbf{k}_m - \mathbf{k}_n) \right)} \\ & + \frac{\sum_{m,i,j \in \Omega} \left((\mathbf{k}_m - \beta)^T (\mathbf{q}_i - \alpha) - (\mathbf{k}_m - \beta)^T (\mathbf{q}_j - \alpha) \right)^2}{\sum_{m \in \Omega} \left((\mathbf{k}_m - \beta)^T (\mathbf{k}_m - \beta) \right) \cdot \sum_{i,j \in \Omega} \left((\mathbf{q}_i - \mathbf{q}_j)^T (\mathbf{q}_i - \mathbf{q}_j) \right)} \end{aligned} \tag{3}$$

Proof sketch: The Hessian of the objective function O with respect to α and β is a non-positive definite matrix. The optimal α^* and β^* are thus the solutions of the following equations: $\frac{\partial O}{\partial \alpha} = 0$, $\frac{\partial O}{\partial \beta} = 0$. Solving this yields $\alpha^* = \frac{1}{|\Omega|} \sum_{i \in \Omega} \mathbf{q}_i$, $\beta^* = \frac{1}{|\Omega|} \sum_{m \in \Omega} \mathbf{k}_m$. Please see the appendix for a detailed proof.

$$\omega(x_i, x_j) = \sigma(q_i^T k_j) = \frac{\exp(q_i^T k_j)}{\sum_{t\in\Omega} \exp(q_i^T k_t)}$$

$$q_i^T k_j = (q_i - \mu_q)^T (k_j - \mu_k) + \mu_q^T k_j + \underline{q_i^T \mu_k + \mu_q^T \mu_k}$$

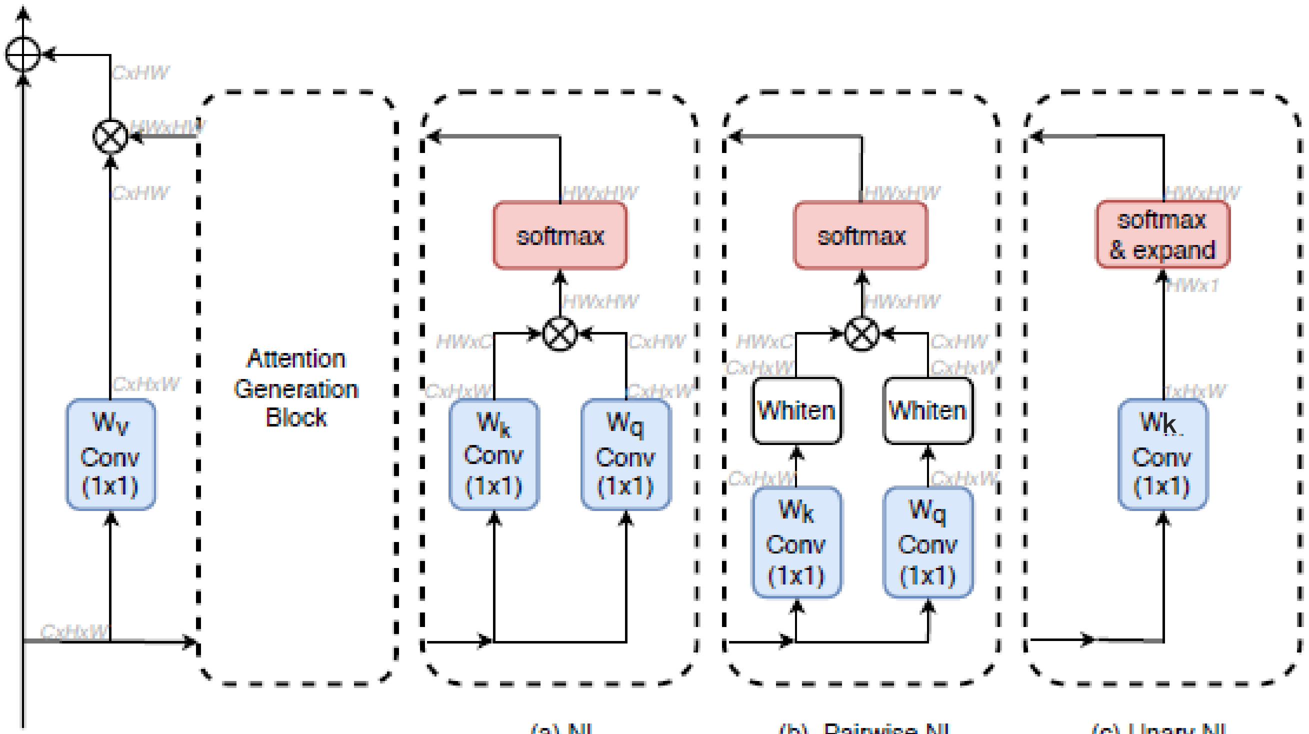
can be eliminated

$$\omega(x_i, x_j) = \sigma(q_i^T k_j) = \sigma(\underbrace{(q_i - \mu_q)^T (k_j - \mu_k)}_{\text{pairwise}} + \underbrace{\mu_q^T k_j}_{\text{unary}})$$

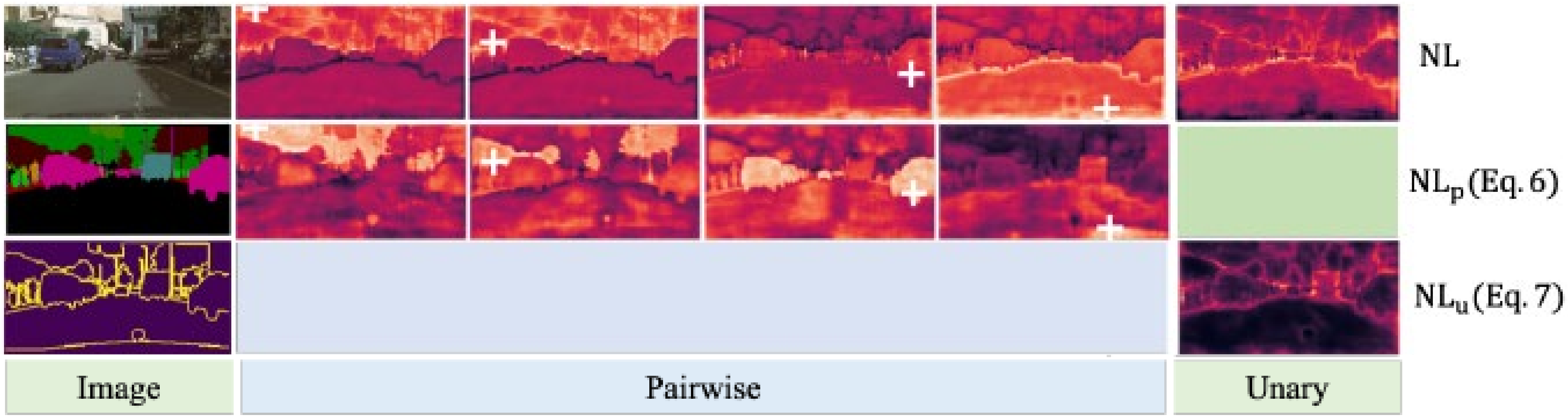
Expect to learn what ?

Pariwise NL $$\omega_p(x_i, x_j) = \sigma((q_i - \mu_q)^T (k_j - \mu_k))$$

Unary NL $$\omega_u(x_i, x_j) = \sigma(\mu_q^T k_j)$$

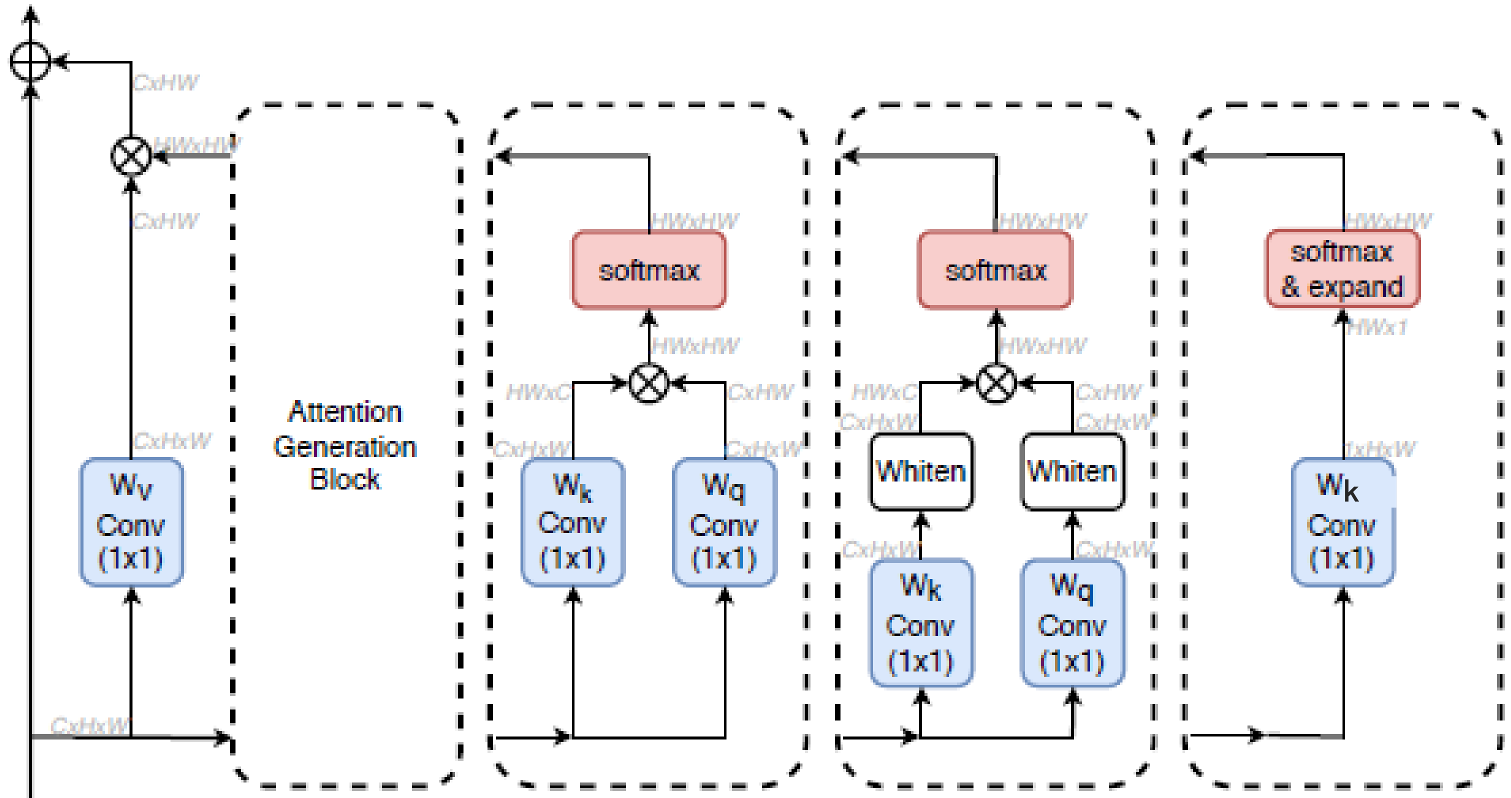


(a) NL (b) Pairwise NL (c) Unary NL



method	pair ∩ within-category	pair ∩ boundary	unary ∩ boundary
random	0.259	0.132	0.135
pairwise NL (Eq. 6)	0.635	0.141	-
unary NL (Eq. 7)	-	-	0.460
NL (Eq. 2)	0.318	0.160	0.172
DNL* (Eq. 13)	0.446	0.146	0.305
DNL† (Eq. 14)	0.679	0.137	0.657
DNL (Eq. 12)	0.759	0.130	0.696

Why Non-Local 👎 ?



(a) NL (b) Pairwise NL (c) Unary NL

$\omega_p(x_i, x_j) \cdot \omega_u(x_i, x_j)$

ose to 0 <-> hard to learn

Modification

- Multiplication → Addition

$$\omega(\mathbf{x}_i, \mathbf{x}_j) = \omega_{\mathrm{p}}(\mathbf{x}_i, \mathbf{x}_j) \cdot \omega_{\mathrm{u}}(\mathbf{x}_i, \mathbf{x}_j)$$
$$\Rightarrow \quad \omega(\mathbf{x}_i, \mathbf{x}_j) = \omega_{\mathrm{p}}(\mathbf{x}_i, \mathbf{x}_j) + \omega_{\mathrm{u}}(\mathbf{x}_i, \mathbf{x}_j).$$

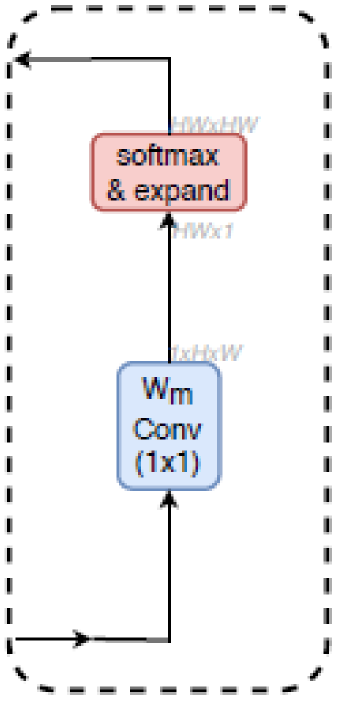
$$\frac{\partial L}{\partial \sigma(\omega_{\mathrm{p}})} = \frac{\partial L}{\partial \sigma(\omega)}, \frac{\partial L}{\partial \sigma(\omega_{\mathrm{u}})} = \frac{\partial L}{\partial \sigma(\omega)}.$$

- Unary term → independent linear transformation

$$\boldsymbol{\mu}_q^T \mathbf{k}_j = \boldsymbol{\mu}_q^T W_k \mathbf{x}_j \Rightarrow m_j = W_m \mathbf{x}_j.$$

- DNL formulation

$$\omega^{\mathrm{D}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left(\left(\mathbf{q}_i - \boldsymbol{\mu}_q\right)^T \left(\mathbf{k}_j - \boldsymbol{\mu}_k\right)\right) + \sigma(m_j)$$
$$\omega^{\mathrm{D}*}(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left(\left(\mathbf{q}_i - \boldsymbol{\mu}_q\right)^T \left(\mathbf{k}_j - \boldsymbol{\mu}_k\right) + m_j\right),$$
$$\omega^{\mathrm{D}\dagger}(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left(\left(\mathbf{q}_i - \boldsymbol{\mu}_q\right)^T \left(\mathbf{k}_j - \boldsymbol{\mu}_k\right)\right) + \sigma(\boldsymbol{\mu}_q^T \mathbf{k}_j),$$

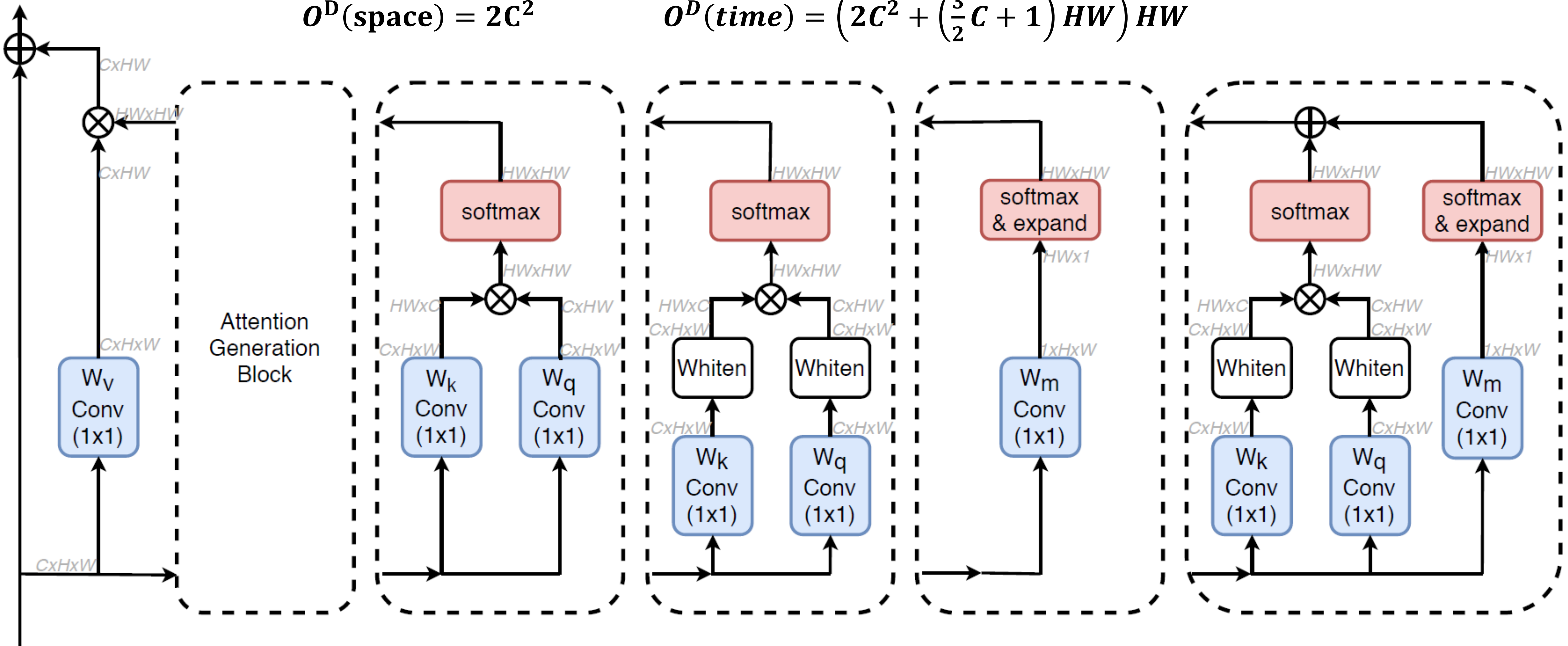


(c) Unary NL

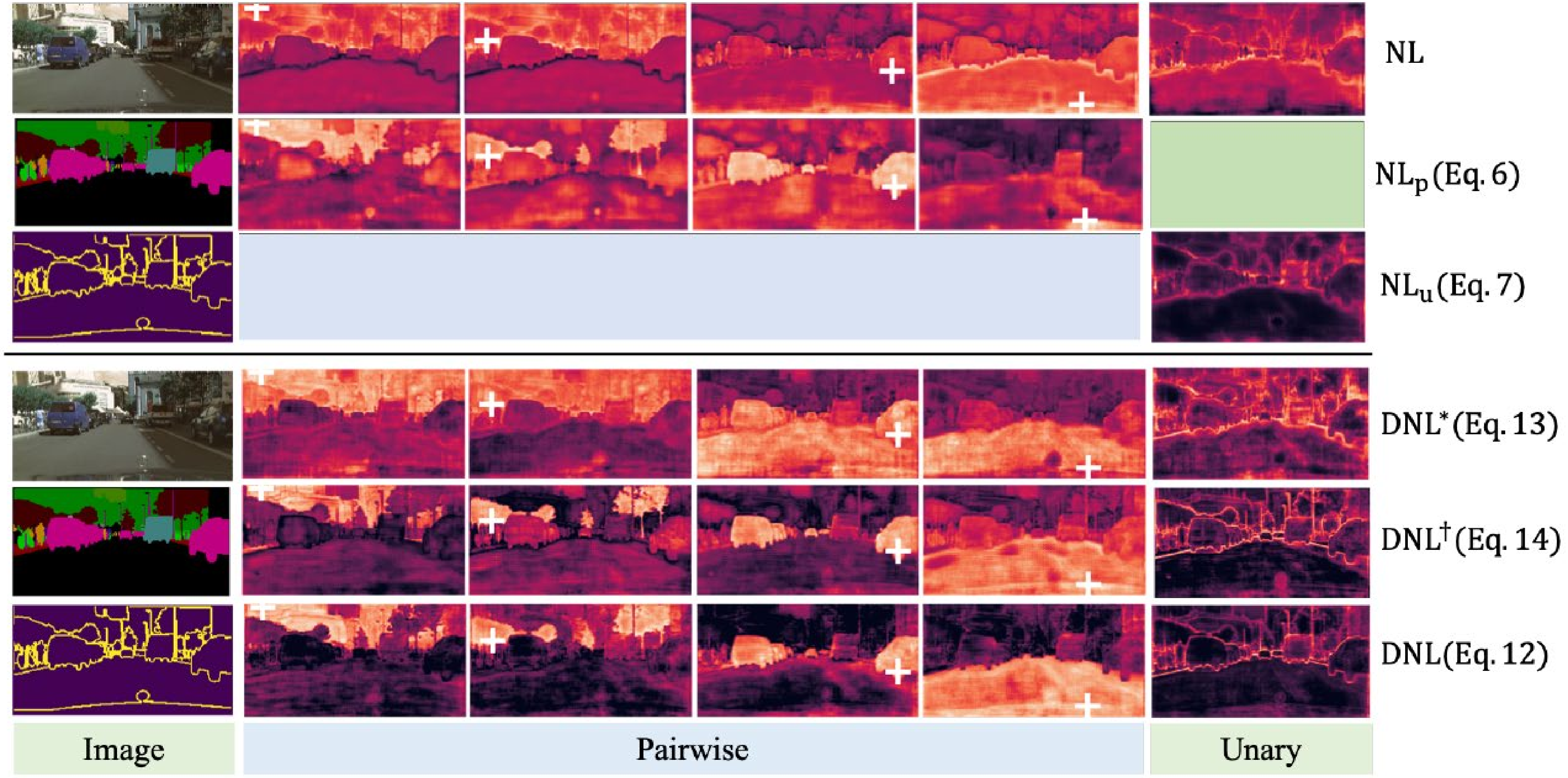
$$\omega^{\mathrm{D}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left(\left(\mathbf{q}_i - \boldsymbol{\mu}_q\right)^T \left(\mathbf{k}_j - \boldsymbol{\mu}_k\right)\right) + \sigma(m_j)$$

$$O^{\mathrm{D}}(\mathrm{space}) = (2C+1)C \qquad O^{D}(time) = \left((2C+1)C + \left(\tfrac{3}{2}C + 2\right)HW\right)HW$$

$$O^{\mathrm{D}}(\mathrm{space}) = 2C^2 \qquad O^{D}(time) = \left(2C^2 + \left(\tfrac{3}{2}C + 1\right)HW\right)HW$$



(a) NL (b) Pairwise NL (c) Unary NL (d) Disentangled NL



$$\omega^{\mathrm{D}*}(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left(\left(\mathbf{q}_i - \boldsymbol{\mu}_q\right)^T \left(\mathbf{k}_j - \boldsymbol{\mu}_k\right) + m_j\right),$$

$$\omega^{\mathrm{D}\dagger}(\mathbf{x}_i, \mathbf{x}_j) = \sigma\left(\left(\mathbf{q}_i - \boldsymbol{\mu}_q\right)^T \left(\mathbf{k}_j - \boldsymbol{\mu}_k\right)\right) + \sigma(\boldsymbol{\mu}_q^T \mathbf{k}_j),$$

Table 3. Comparisons with state-of-the-art approaches on the Cityscapes test set

Method	Backbone	ASPP	Coarse	mIoU (%)
PSANet [44]	ResNet-101			80.1
DANet [13]	ResNet-101			81.5
HRNet [31]	HRNetV2-W48			81.9
SeENet [29]	ResNet-101			81.2
SPGNet [7]	ResNet-101			81.1
CCNet [23]	ResNet-101			81.4
ANN [47]	ResNet-101			81.3
DenseASPP [38]	DenseNet-161	✓		80.6
OCNet [39]	ResNet-101	✓		81.7
ACFNet [40]	ResNet-101	✓		81.8
PSPNet [43]	ResNet-101		✓	81.2
PSANet [44]	ResNet-101		✓	81.4
DeepLabv3 [5]	ResNet-101	✓	✓	81.3
NL	ResNet-101		✓	80.8
DNL (ours)	ResNet-101		✓	82.0
NL	HRNetV2-W48		✓	82.5
DNL (ours)	HRNetV2-W48		✓	83.0

(a) Decoupling strategy

	mul $\rightarrow$ add	non-shared W_k	mIoU
Baseline	-	-	75.8
NL	$\times$	$\times$	78.5
DNL† (14)	$\surd$	$\times$	79.2
DNL* (13)	$\times$	$\surd$	79.0
DNL	$\surd$	$\surd$	80.5

(b) Pairwise and unary terms

	pairwise term	unary term	mIoU
Baseline	-	-	75.8
NL	$\surd$	$\surd$	78.5
NL_p	$\surd$	$\times$	77.5
NL_u	$\times$	$\surd$	79.3
DNL	$\surd$	$\surd$	80.5

Table 4. Comparisons with state-of-the-art approaches on the validation set and test set of ADE20K, and test set of PASCAL-Context

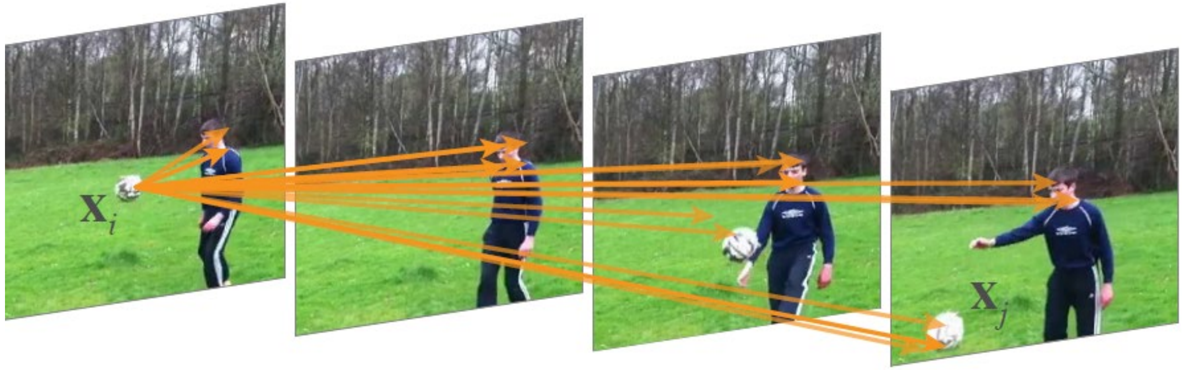
Method	Backbone	ADE20K val mIoU (%)	ADE20K test mIoU (%)	PASCAL-Context test mIoU (%)
PSANet [44]	ResNet-101	43.77	55.46	-
CCNet [23]	ResNet-101	45.22	-	-
OCNet [39]	ResNet-101	45.45	-	-
SVCNet [11]	ResNet-101	-	-	53.2
EMANet [25]	ResNet-101	-	-	53.1
HRNetV2 [31]	HRNetV2-W48	42.99	-	54.0
EncNet [41]	ResNet-101	44.65	55.67	52.6
DANet [13]	ResNet-101	45.22	-	52.6
CFNet [42]	ResNet-101	44.89	-	54.0
ANN [47]	ResNet-101	45.24	-	52.8
DMNet [17]	ResNet-101	45.50	-	54.4
ACNet [14]	ResNet-101	45.90	55.84	54.1
NL	ResNet-101	44.67	55.58	50.6
DNL (ours)	ResNet-101	45.97	56.23	54.8
NL	HRNetV2-W48	44.82	55.60	54.2
DNL (ours)	HRNetV2-W48	45.82	55.98	55.3

Table 5. Complexity comparisons

	#param(M)	FLOPs(G)	latency(s/img)
baseline	70.960	691.06	0.177
NL	71.484	765.07	0.192
DNL	71.485	765.16	0.194

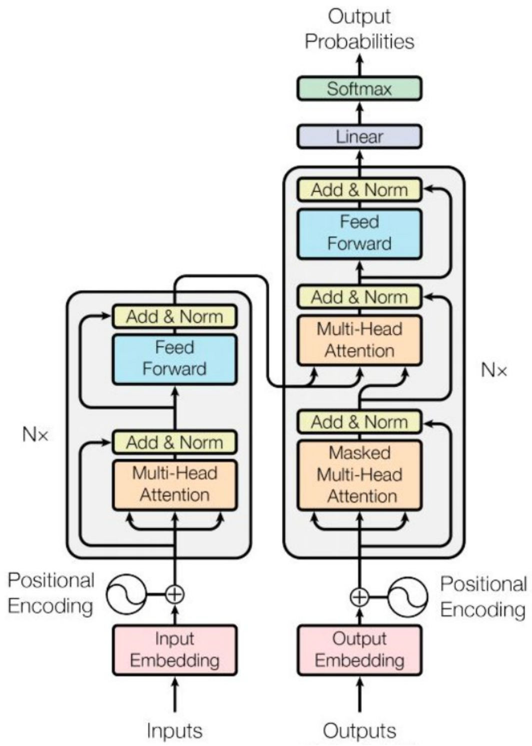
Table 7. Results based on Slow-only baseline using R50 as backbone on Kinetics validation set

	Top-1 Acc	Top-5 Acc
baseline	74.94	91.90
NL	75.95	92.29
NL_p	76.01	92.28
NL_u	75.76	92.44
DNL	76.31	92.69

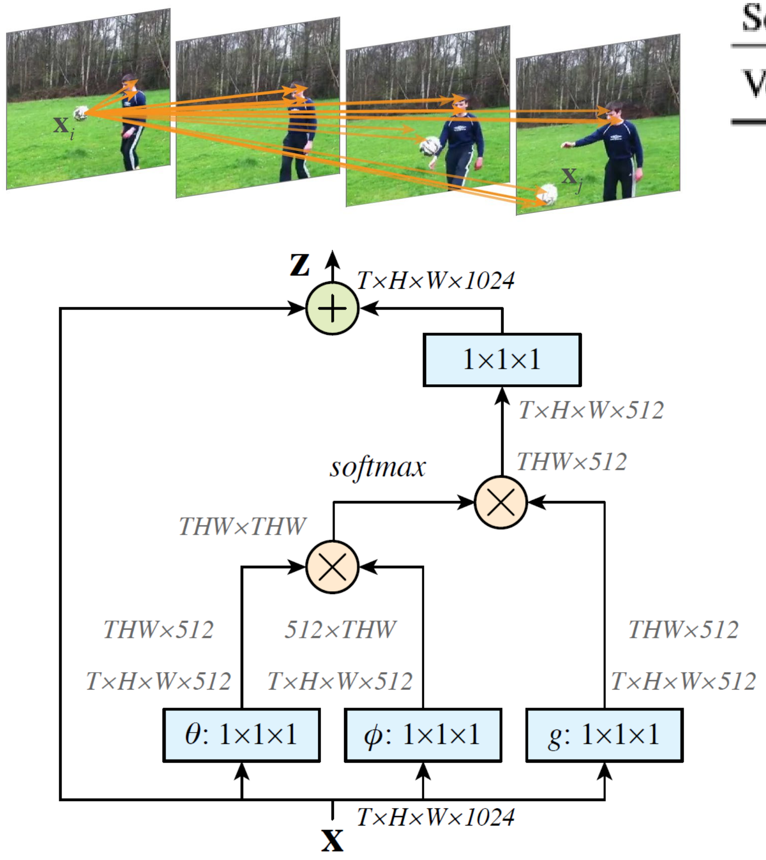


Comparisons

Self Attention

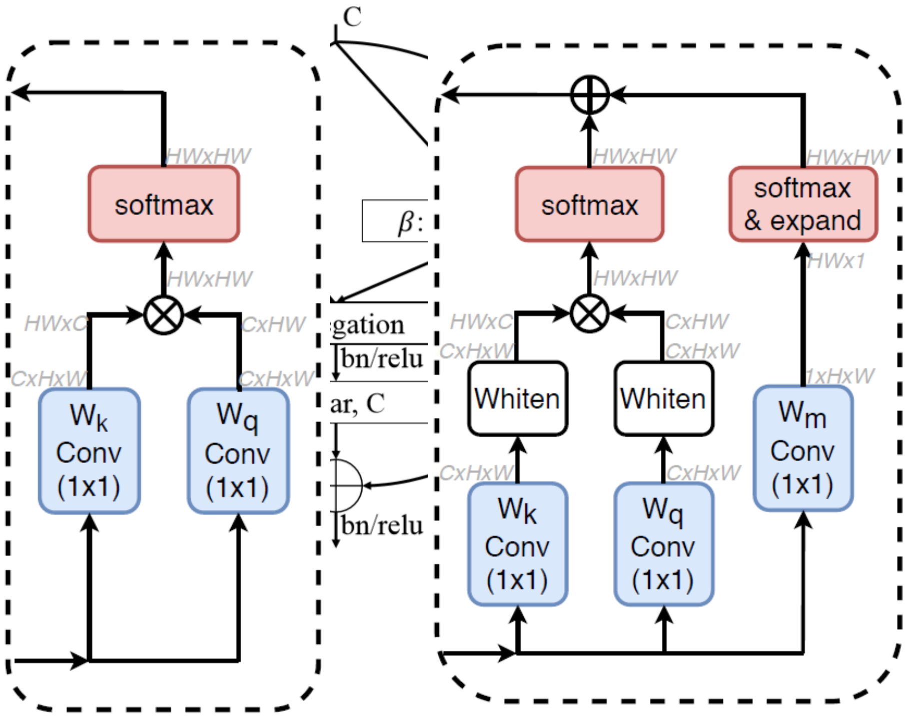


Non-Local



Pairwise Attention

Operation	Content adaptive	Channel adaptive
Convolution [19]	✗	✓
Scalar attention [33, 35, 27, 13]	✓	✗
Vector attention (ours)	✓	✓



(a) NL

(d) Disentangled NL

Disentangled Non-Local

Comments

- Combination of theoretical analysis and experimental analysis
- Reasonable extension and modelling

Limitations

- Limited improvements
- Only improvement of Non-Local, not outside the framework
- “Long”-range not long enough

Thanks