Self-Attention Modeling for Visual Recognition

Han Hu

Visual Computing Group

Microsoft Research Asia (MSRA)

CVPR2020 Tutorial

Overview

- Part I: Applications of Self-Attention Models for Visual Recognition
  - Pixel-to-pixel relationship
  - Object-to-pixel relationship
  - Object-to-object relationship

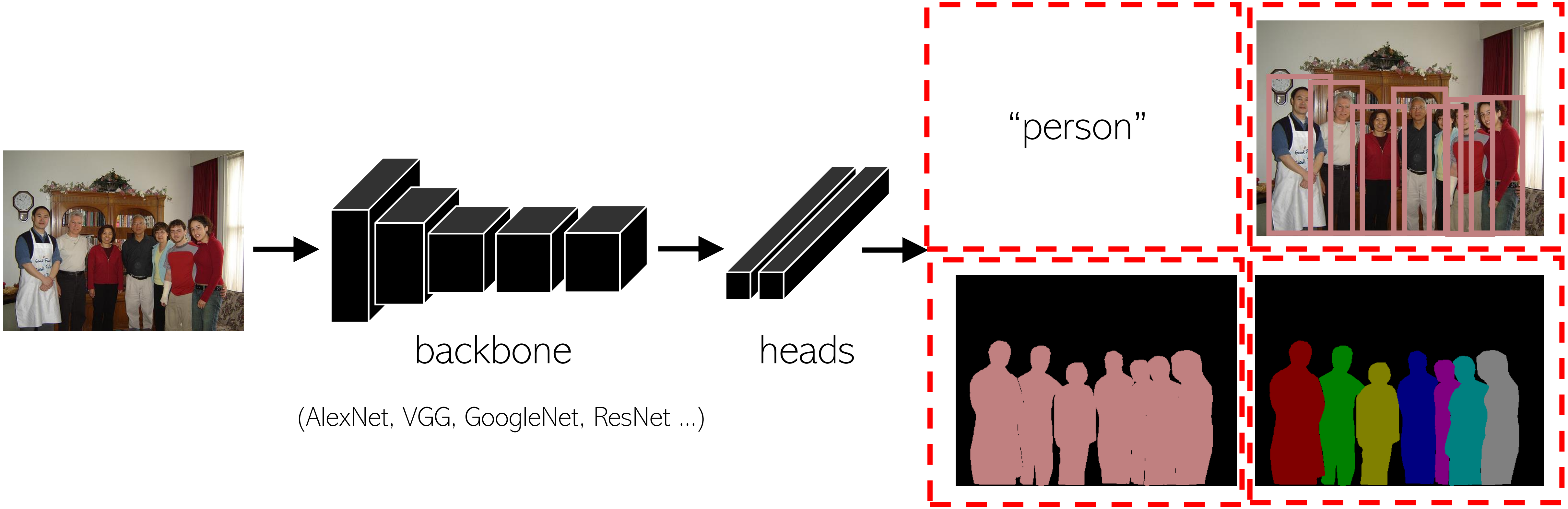
- Part II: Diagnosis and Improvement of Self-Attention Modeling
  - Are self-attention models learnt well on visual tasks?
  - How can it be more effective?

Overview

- Part I: Applications of Self-Attention Models for Visual Recognition
  - Pixel-to-pixel relationship
  - Object-to-pixel relationship
  - Object-to-object relationship

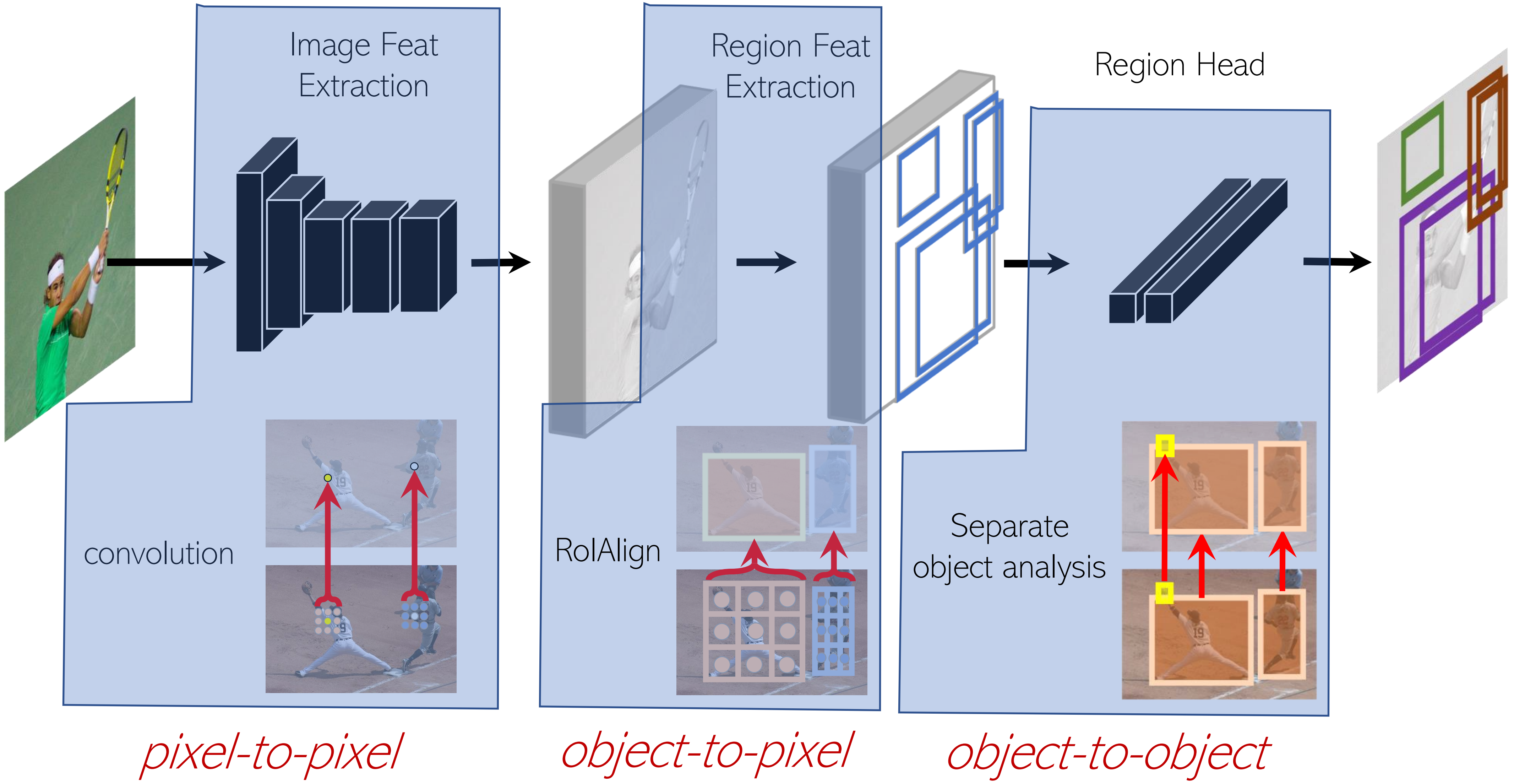
- Part II: Diagnosis and Improvement of Self-Attention Modeling
  - Are self-attention models learnt well on visual tasks?
  - How can it be more effective?

Visual Recognition Paradigm

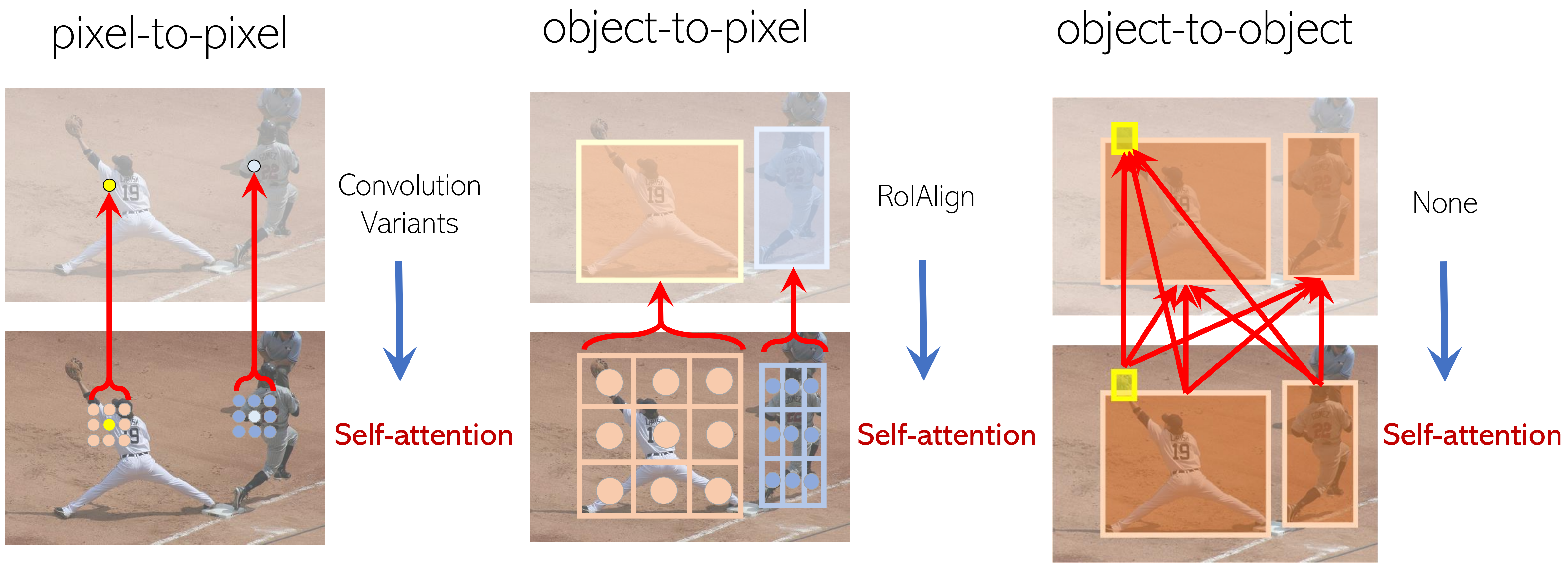


various recognition tasks

An Object Detection Example



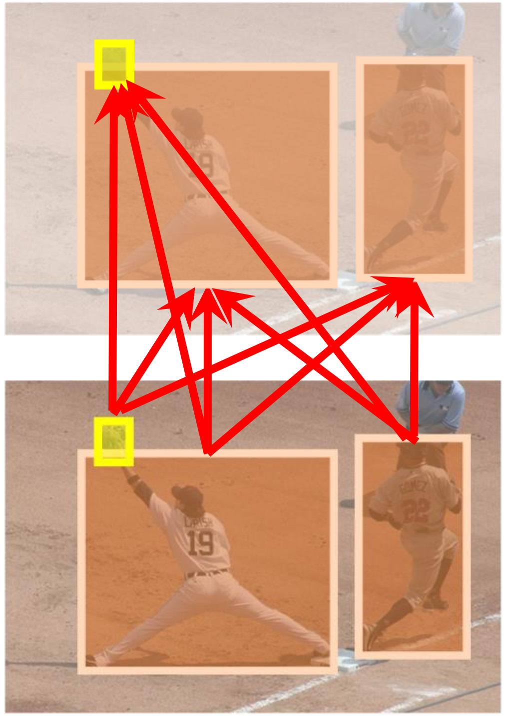
Relationship Modeling of Basic Visual Elements



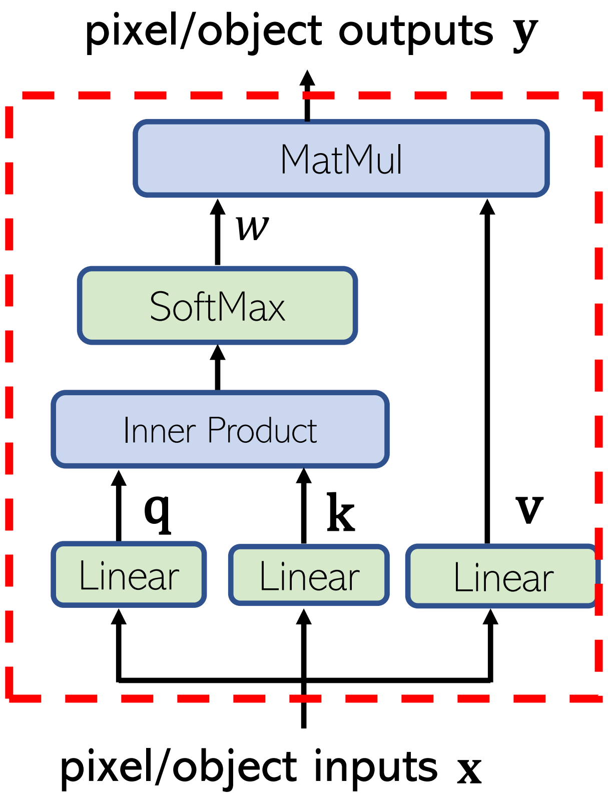
What is a Self-Attention Module?

- Transforms the pixel/object input feature by encoding its relationship with other pixels/objects
- A weighted average of **Value**, where the weight is the normalized inner product of **Query** and **Key**

output feats



input feats

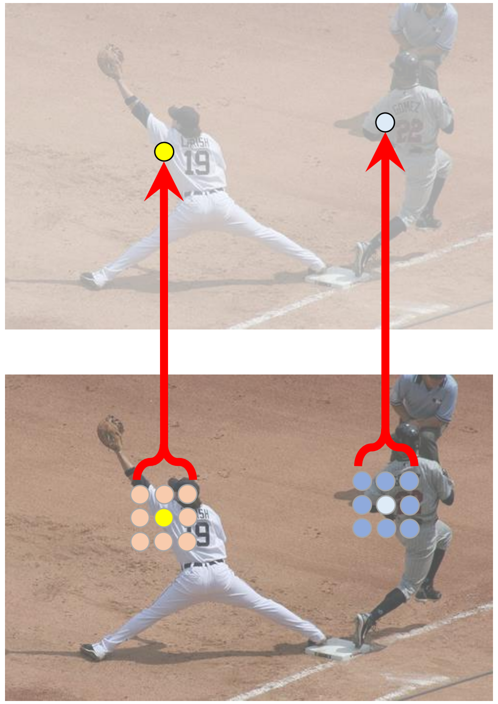


$$\mathbf{y}_i = \sum_{j \in \Omega} w(\boldsymbol{q}_i, \mathbf{k}_j)\, \mathbf{v}_j$$

$$w(\mathbf{q}_i, \mathbf{k}_j) \sim exp(\mathbf{q}_i^T \mathbf{k}_j)$$

Pixel-to-Pixel Relation Modeling

pixel-to-pixel



Convolution Variants → **Self-Attention**

Usage

✓Complement convolution

✓Replace convolution

Complement Convolution

- “Convolution is too local”

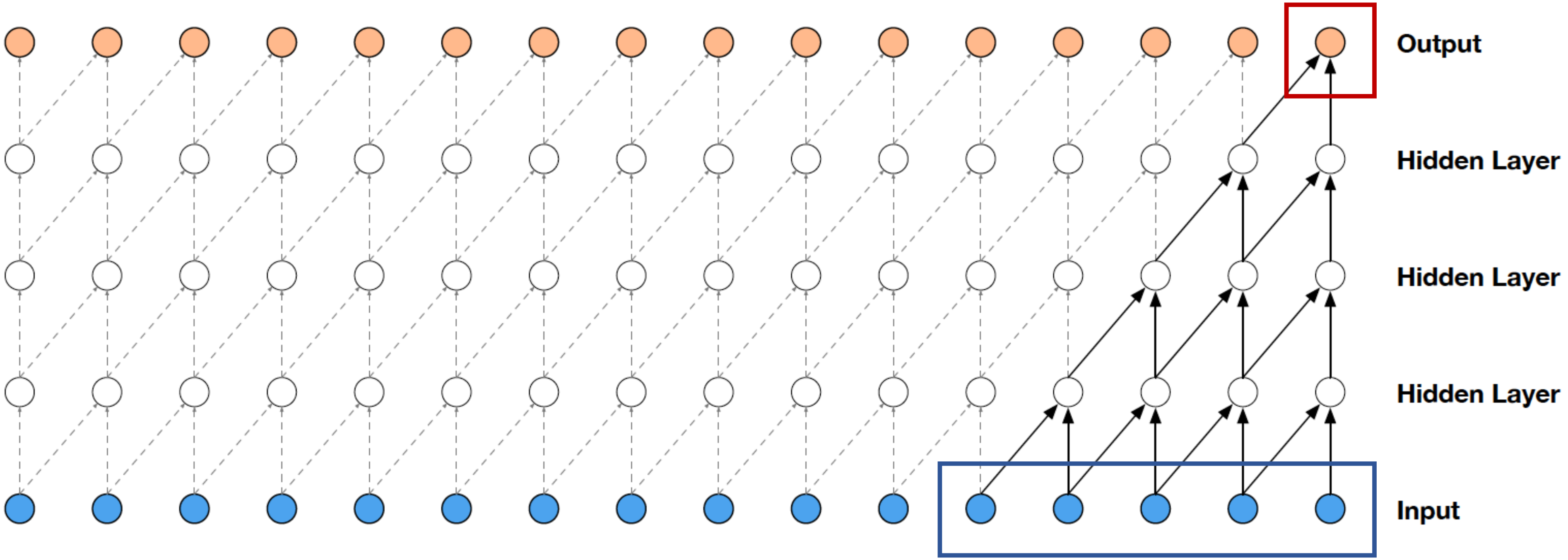
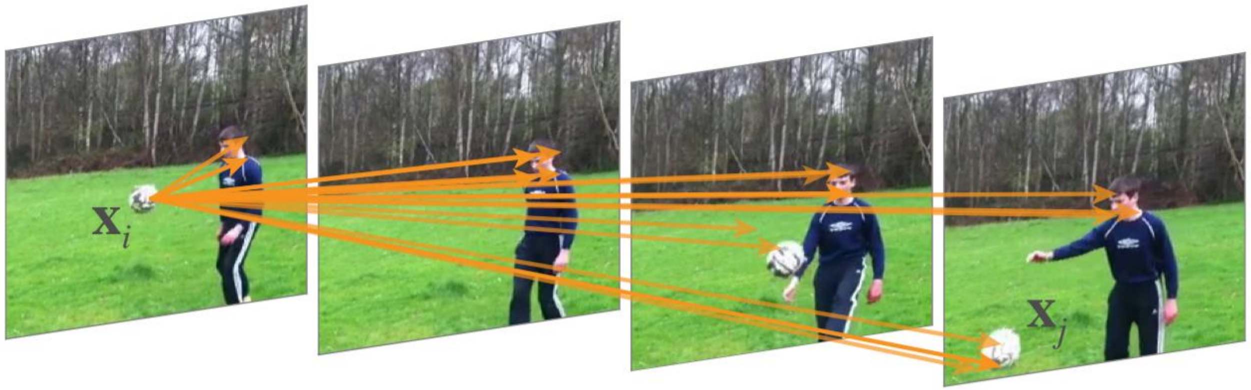
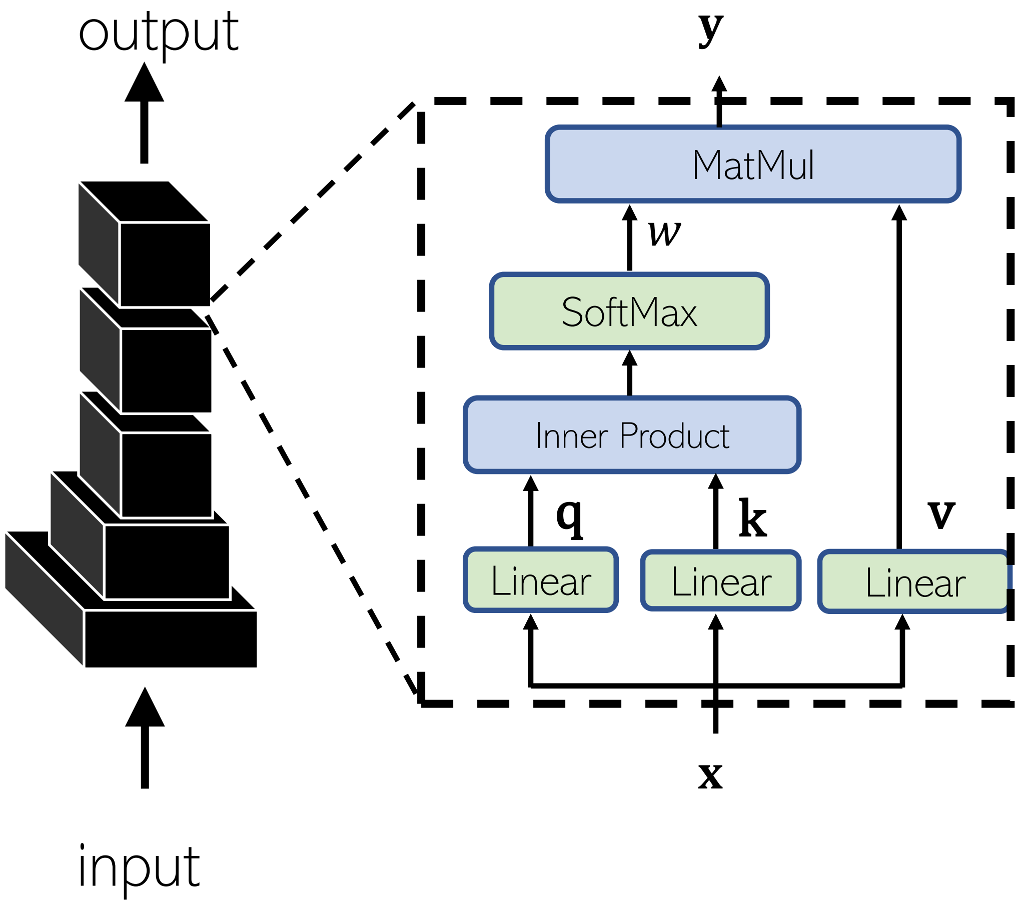


Figure credit: Van Den Oord et al.

Complement Convolution

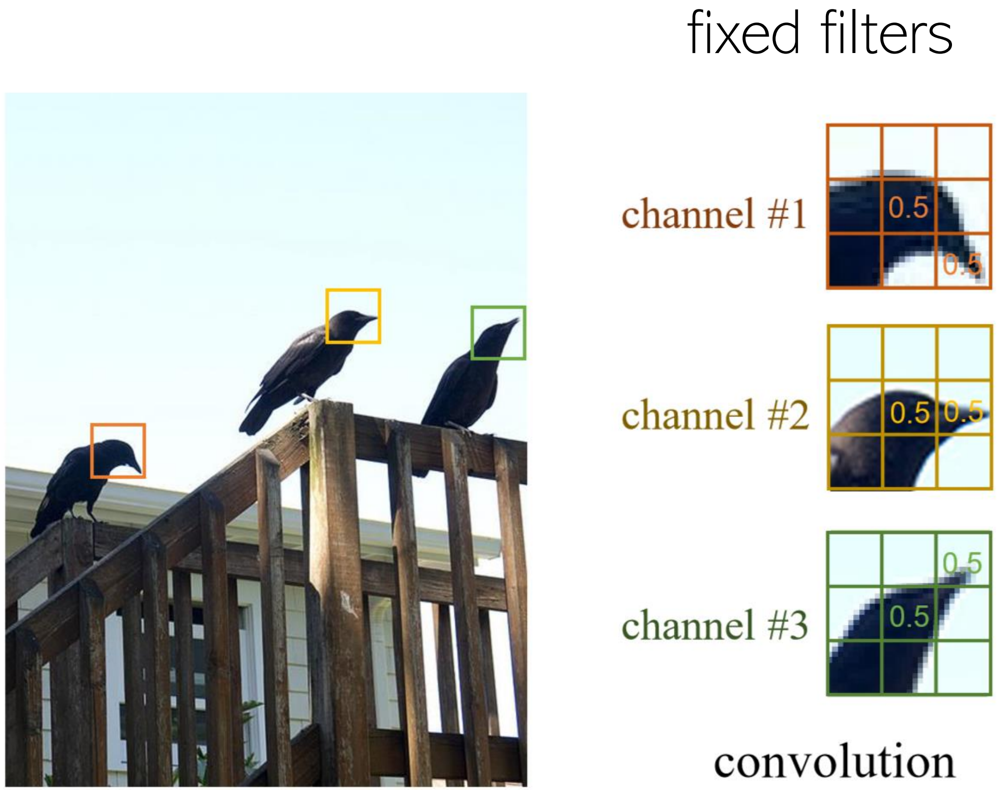
- Non-Local Networks [Wang et al, CVPR'2018]





Replace Convolution

- "Convolution is exponentially inefficient"



Convolution
=Template Matching

We need 3 channels/filters/templates to encode these bird heads!

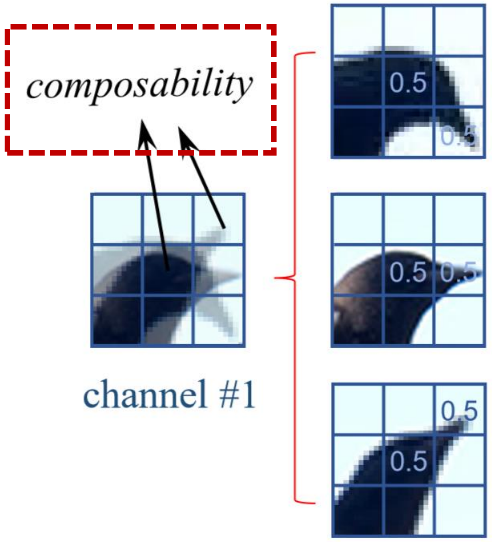
Inefficient!

Replace Convolution

- **Adaptive filters (composition)** vs. fixed filters (template)

local relation layer

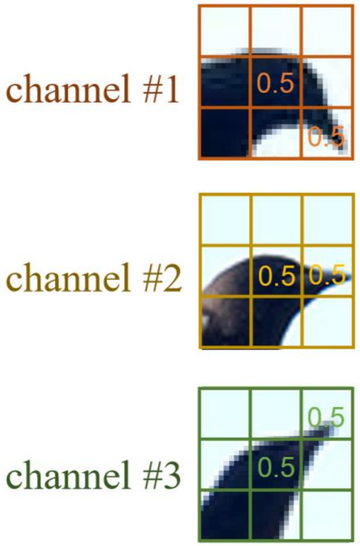
adaptive filters



(1 channel)

VS.

convolution layer



fixed filters

(3 channels)

Han Hu, Zheng Zhang, Zhenda Xie and Stephen Lin. *Local Relation Networks for Visual Recognition. ICCV 2019*

Local Relation Network (LR-Net)

- Replace all **convolution layers** by **local relation layers**

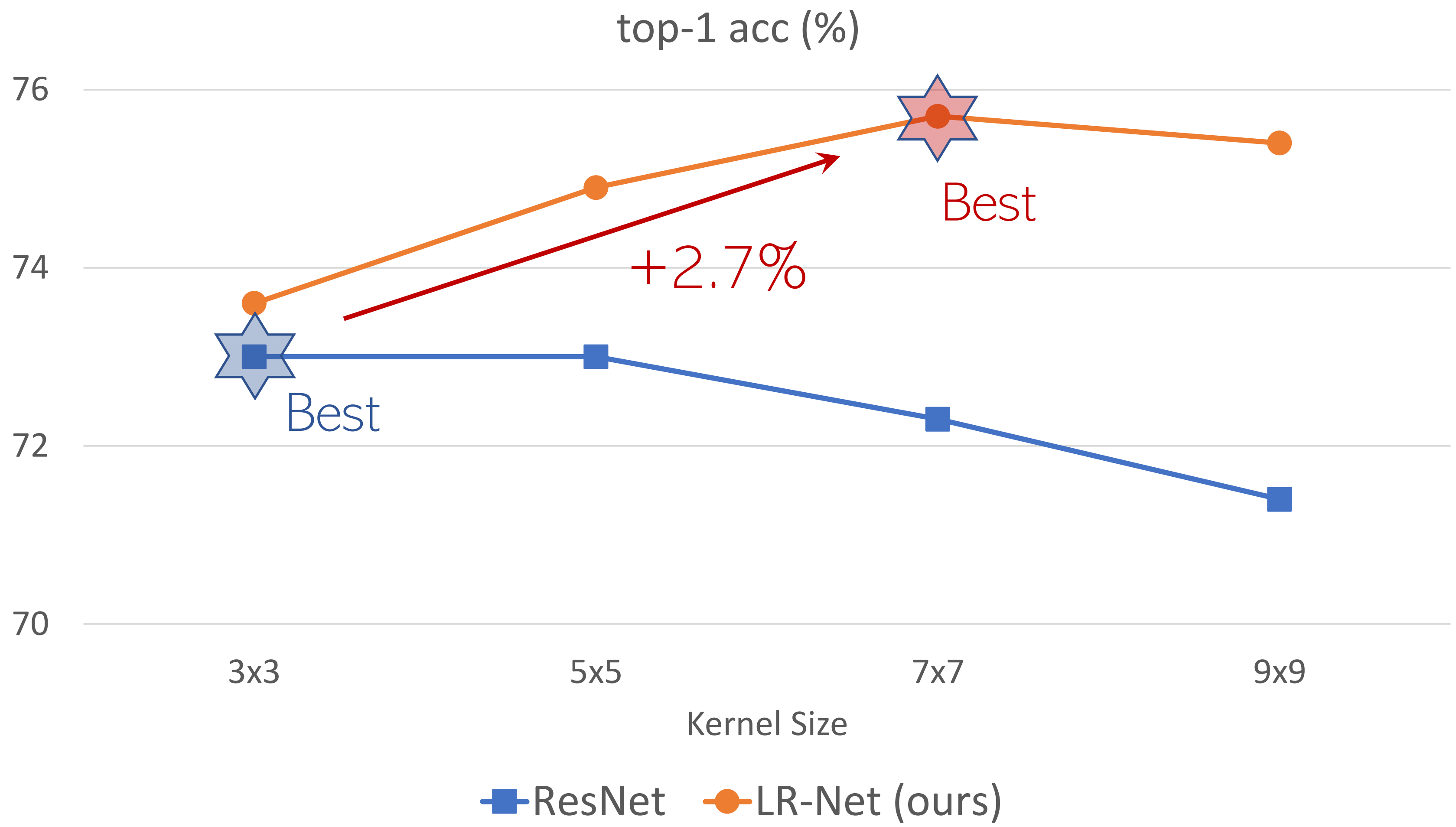
ResNet

LR-Net

stage	output	ResNet-50	LR-Net-50 (7×7, m=8)
res1	112×112	7×7 conv, 64, stride 2	**1×1, 64** **7×7 LR, 64, stride 2**
res2	56×56	3×3 max pool, stride 2 [1×1, 64 3×3 conv, 64 1×1, 256] ×3	3×3 max pool, stride 2 [1×1, 100 **7×7 LR, 100** 1×1, 256] ×3
res3	28×28	[1×1, 128 3×3 conv, 128 1×1, 512] ×4	[1×1, 200 **7×7 LR, 200** 1×1, 512] ×4
res4	14×14	[1×1, 256 3×3 conv, 256 1×1, 1024] ×6	[1×1, 400 **7×7 LR, 400** 1×1, 1024] ×6
res5	7×7	[1×1, 512 3×3 conv, 512 1×1, 2048] ×3	[1×1, 800 **7×7 LR, 800** 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params		$\mathbf{25.5}\times10^6$	$\mathbf{23.3}\times10^6$
FLOPs		$\mathbf{4.3}\times10^9$	$\mathbf{4.3}\times10^9$

Han Hu, Zheng Zhang, Zhenda Xie and Stephen Lin. *Local Relation Networks for Visual Recognition. ICCV 2019*

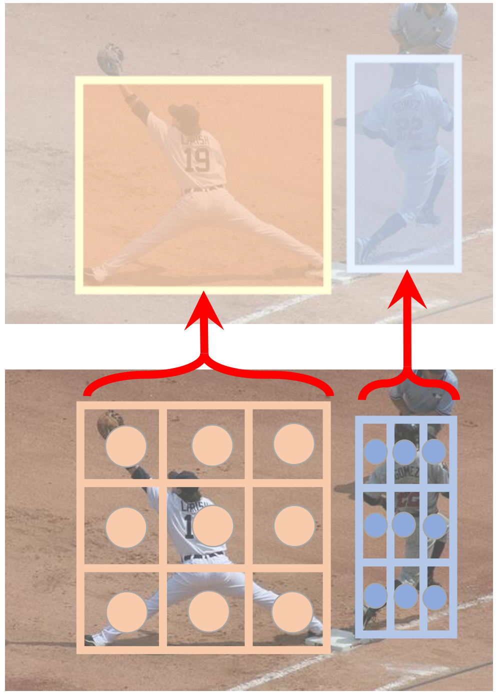
Classification on ImageNet (26 Layers)



Han Hu, Zheng Zhang, Zhenda Xie and Stephen Lin. *Local Relation Networks for Visual Recognition. ICCV 2019*

Object-to-Pixel Relation Modeling

object-to-pixel



RoIAlign ⟶ **Self-Attention**

- Learn Region Features [ECCV'2018]
- Transformer Detector [Tech Report'2020]

Learnable Object-to-Pixel Relation

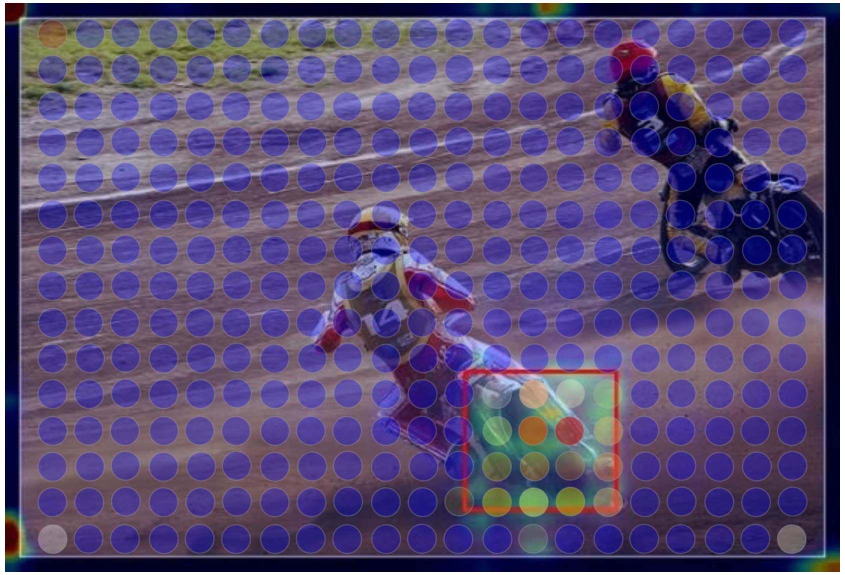


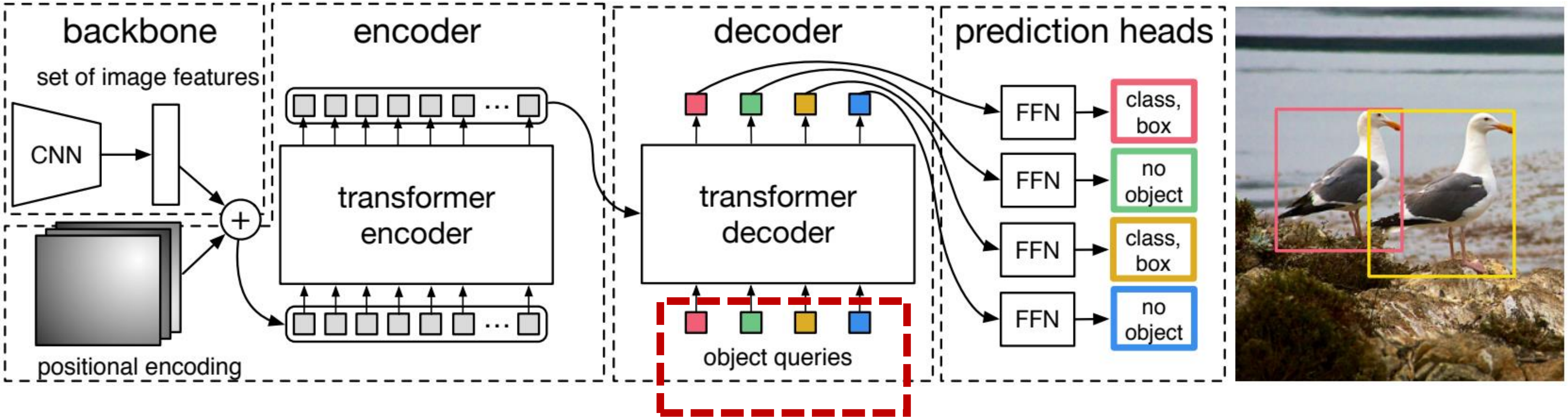
Image Feature to Region Feature

Geometric

Appearance

Jiayuan Gu, Han Hu, Liwei Wang, Yichen Wei and Jifeng Dai. *Learning Region Features for Object Detection.* ECCV 2018

Transformer Detectors (DETR)

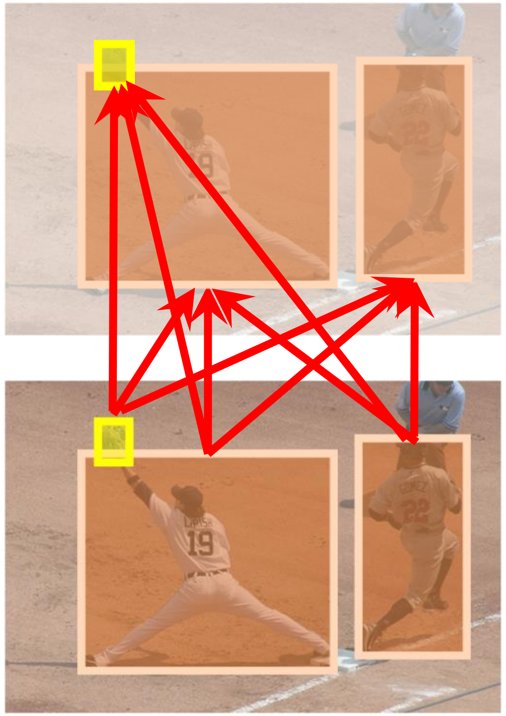


Implicitly learnt

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. *End-to-End Object Detection with Transformers*. Tech Report 2020

Object-to-Object Relation Modeling

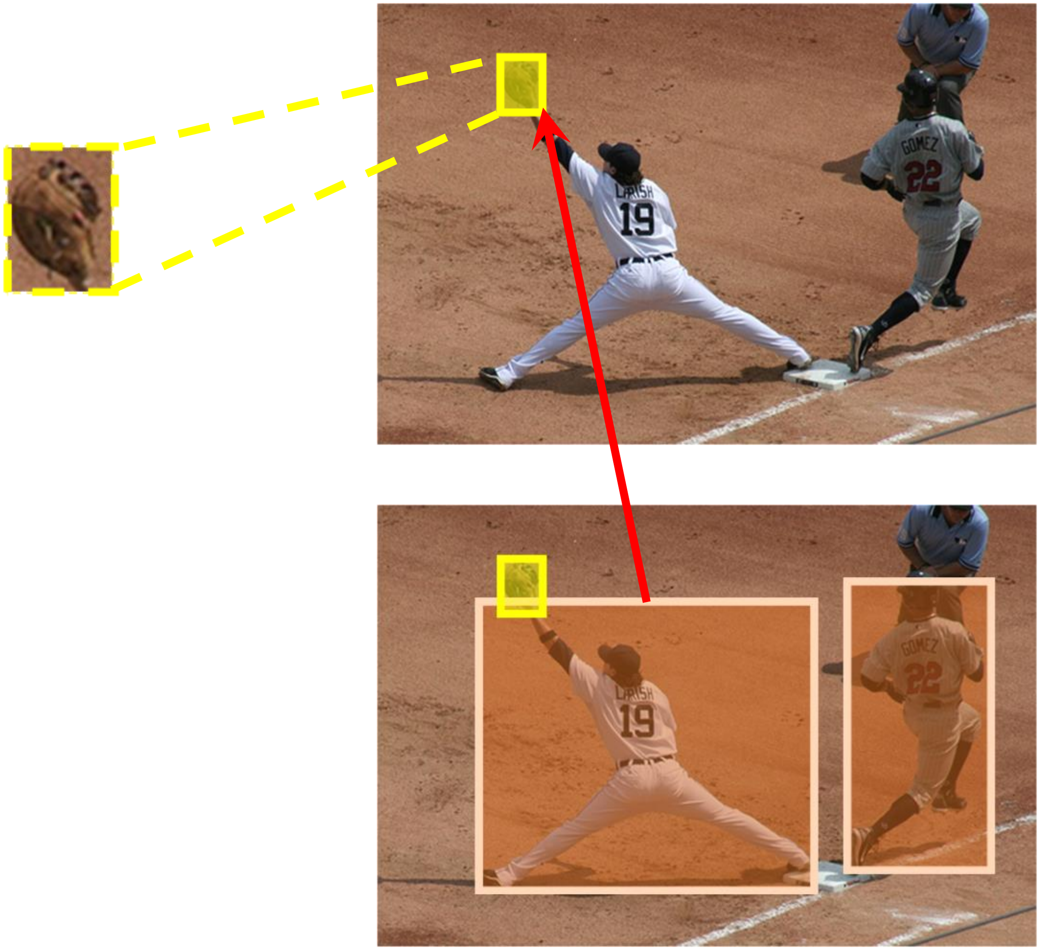
object-to-object



None ⟶ **Self-Attention**

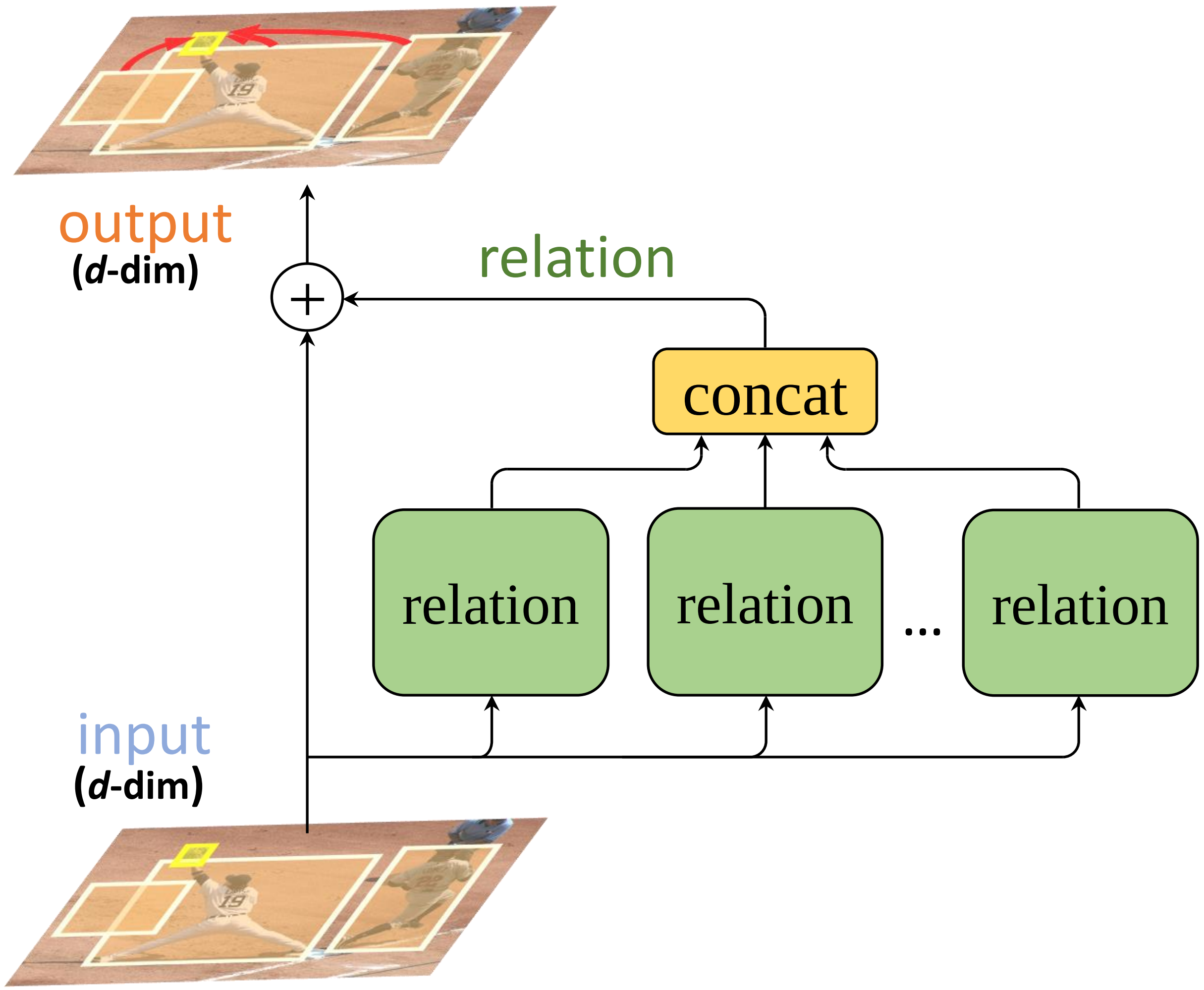
- Object Detection
  - Relation Networks [CVPR'2018]
- Video Action Recognition
  - Videos as Space-Time Region Graphs [ECCV'2018]
- Multi-Object Tracking
  - Spatial-Temporal Relation Network [ICCV'2019]
- Video Object Detection
  - RDN [ICCV'2019]
  - MEGA [CVPR'2020]

Object-to-Object Relation Modeling



It is much easier to detect the ***glove*** if we know there is a ***baseball player***.

Object Relation Module

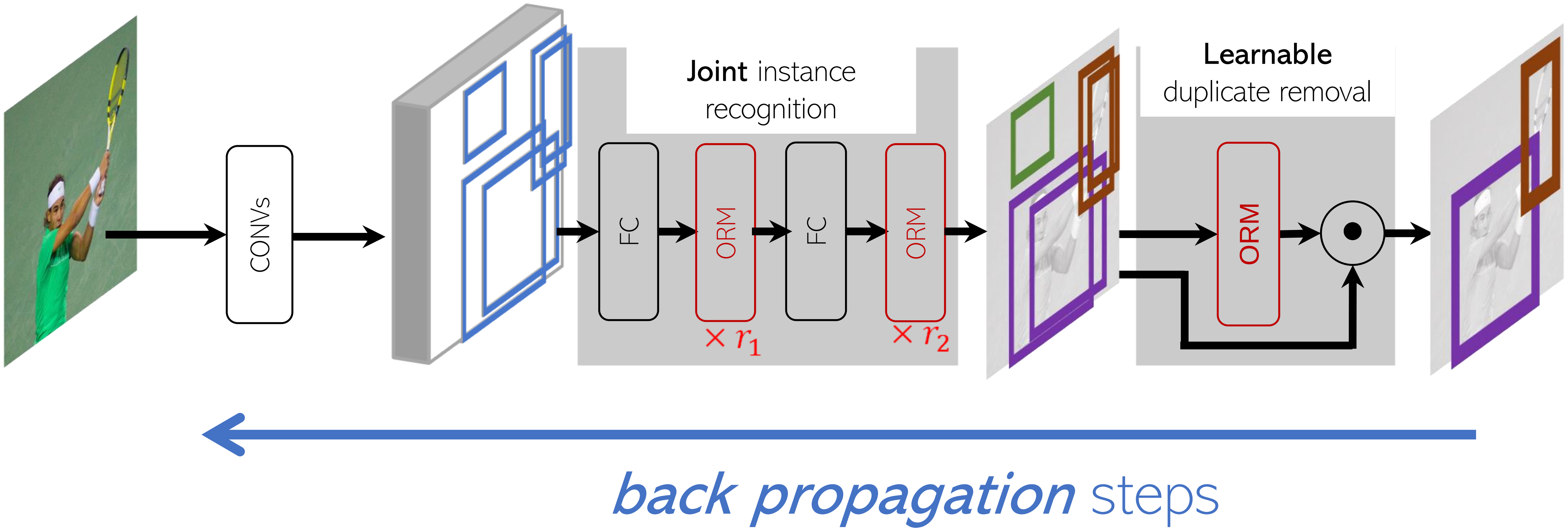


Key Feature

✓Relative position

Han Hu*, Jiayuan Gu*, Zheng Zhang*, Jifeng Dai and Yichen Wei. *Relation Networks for Object Detection.* CVPR 2018

The First Fully End-to-End Object Detector



Han Hu*, Jiayuan Gu*, Zheng Zhang*, Jifeng Dai and Yichen Wei. *Relation Networks for Object Detection.* CVPR 2018

On Stronger Base Detectors

backbone	setting	mAP	mAP_{50}	mAP_{75}	#. params	FLOPS	
faster RCNN	2fc+SoftNMS	32.2/32.7	52.9/53.6	34.2/34.7	58.3M	122.2B	
	2fc+RM+SoftNMS	34.7/35.2	55.3/**56.2**	37.2/37.8	64.3M	124.6B	+3.0 mAP
	2fc+RM+e2e	**35.2/35.4**	**55.8**/56.1	**38.2/38.5**	64.6M	124.9B	
FPN	2fc+SoftNMS	36.8/37.2	57.8/58.2	40.7/41.4	56.4M	145.8B	
	2fc+RM+SoftNMS	38.1/38.3	59.5/59.9	41.8/42.3	62.4M	157.8B	+2.0 mAP
	2fc+RM+e2e	**38.8/38.9**	**60.3/60.5**	**42.9/43.3**	62.8M	158.2B	
DCN	2fc+SoftNMS	37.5/38.1	57.3/58.1	41.0/41.6	60.5M	125.0B	
	2fc+RM+SoftNMS	38.1/38.8	57.8/**58.7**	41.3/42.4	66.5M	127.4B	+1.0 mAP
	2fc+RM+e2e	**38.5/39.0**	**57.8**/58.6	**42.0/42.9**	66.8M	127.7B	

*Faster R-CNN with ResNet-101 model are used (evaluation on *minival*/ *test-dev* are reported)

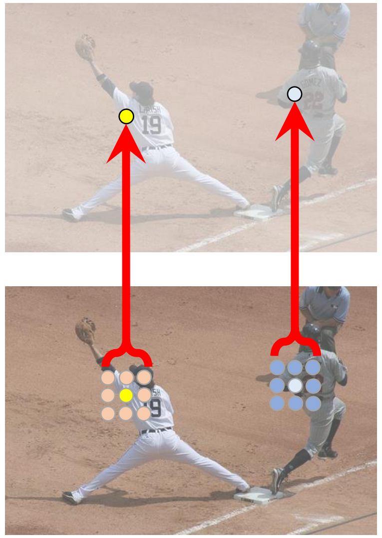
Relation Networks

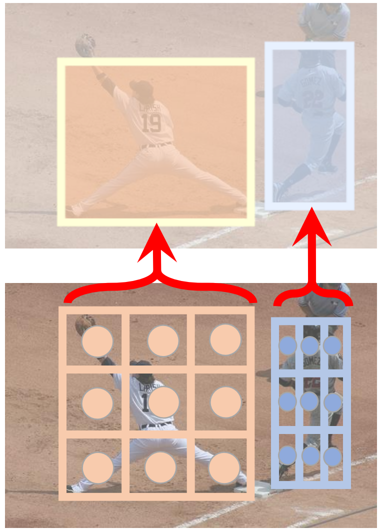
ResNeXt-101-64x4d-FPN-DCN 45.0 → 45.9

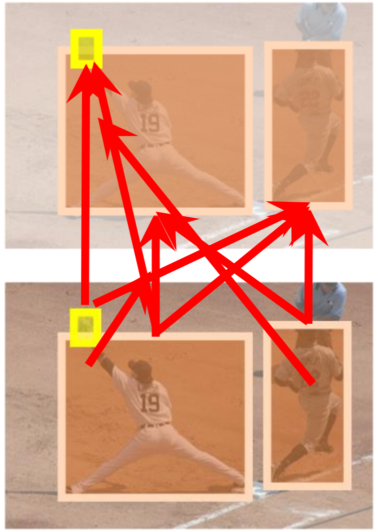
+0.9 mAP

Part I Summary

- Part I: Self-Attention Models for Visual Recognition (Application View)
  - Pixel-to-Pixel, Object-to-Pixel, Object-to-Object
  - **A strong competitor; complementary to existing architectures; SOTA in video applications**
  - **There is still much room to improve!**



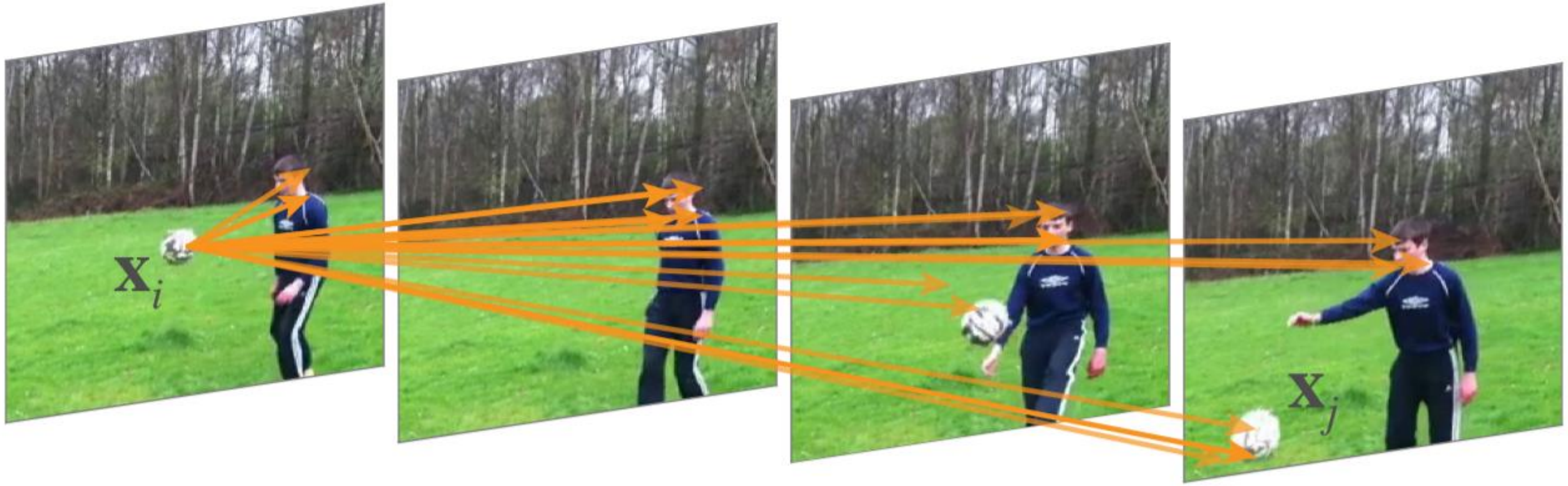




Overview

- Part I: Applications of Self-Attention Models for Visual Recognition
  - Pixel-to-Pixel
  - Object-to-Pixel
  - Object-to-Object
- Part II: Diagnosis and Improvement of Self-Attention Modeling
  - Are self-attention models learnt well on visual tasks?
  - How can it be more effective?
  - [GCNet, ICCVW'2019] https://arxiv.org/pdf/1904.11492.pdf
  - [Disentangled Non-Local Networks, Arxiv'2020] https://arxiv.org/pdf/2006.06668.pdf

Self-Attention Encodes **Pairwise** Relationship



Does it learn pairwise relationship well?

Expectation of Learnt Relation

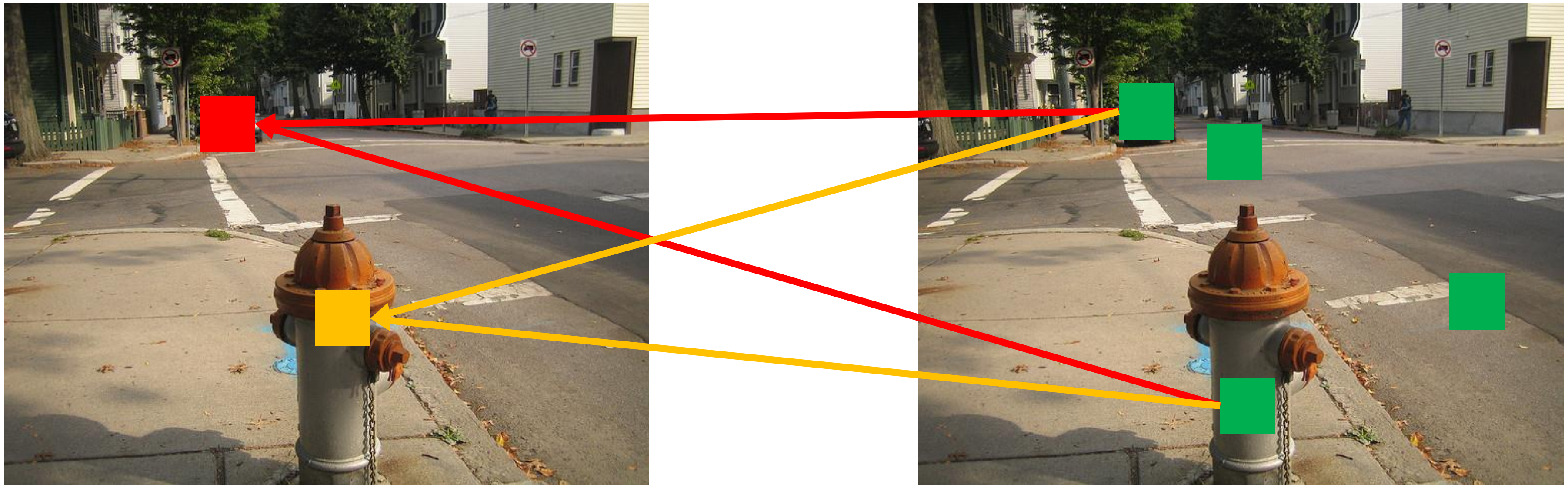
- Different queries affected by **different** key

Query **Key**

What does the Self-Attention Learn?

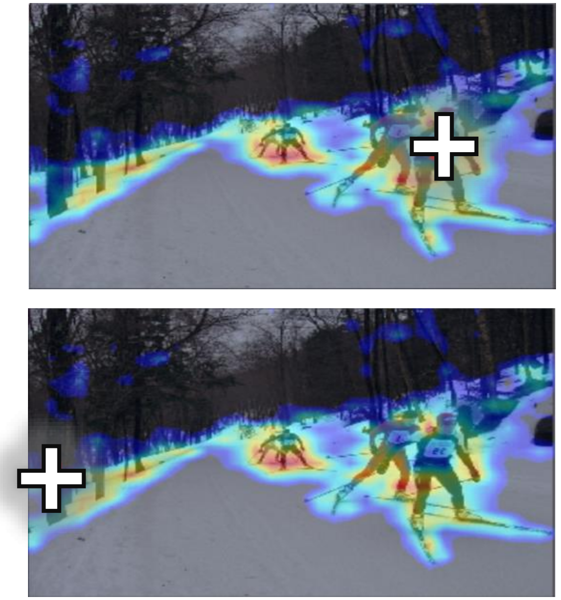
- Different queries affected by the **same** keys
- **Pairwise** in expectation → **Unary** in actual

Query **Key**

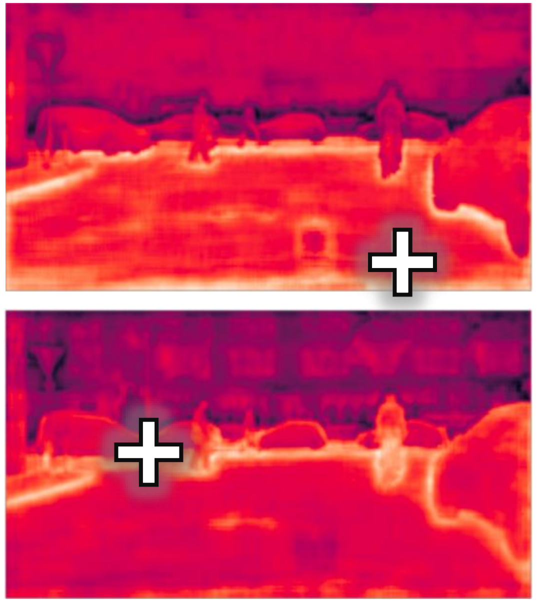


Visualizations on Real Tasks

- indicates the query point
- The activation map for different queries are similar
- The self-attention model degenerates to a unary model



Object Detection



Semantic Segmentation

[GCNet, ICCVW'2019]

https://arxiv.org/pdf/1904.11492.pdf

WHY?

Revisit Self-Attention Formulation

- The self-attention formulation has a '*hidden*' unary term:

$$w\left(\mathbf{q}_i, \mathbf{k}_j\right) \sim exp\left(\mathbf{q}_i^T \mathbf{k}_j\right) = exp\left(\underbrace{(\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)}_{\text{(whitened) pairwise}} + \underbrace{\boldsymbol{\mu}_q^T \mathbf{k}_j}_{\text{(hidden) unary}}\right)$$

* $\boldsymbol{\mu}_q$ and $\boldsymbol{\mu}_k$ are global average of $\mathbf{q}$ **and** $\mathbf{k}$

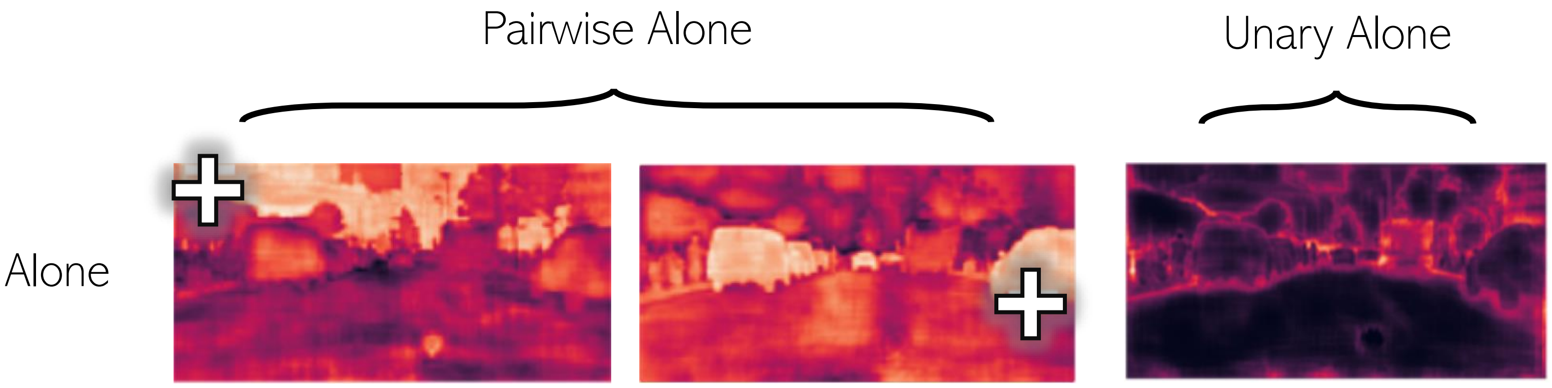
Behavior of the Pairwise and Unary Terms

method	fomulation	mIoU
Baseline	none	75.8%
Joint (Self-Attention)	$\sim exp(\mathbf{q}_i^T \mathbf{k}_j)$	$\boxed{78.5\%}$
Pairwise Alone	$\sim exp((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k))$	77.5%
Unary Alone	$\sim exp(\boldsymbol{\mu}_q^T \mathbf{k}_j)$	$\boxed{\mathbf{79.3\%}}$

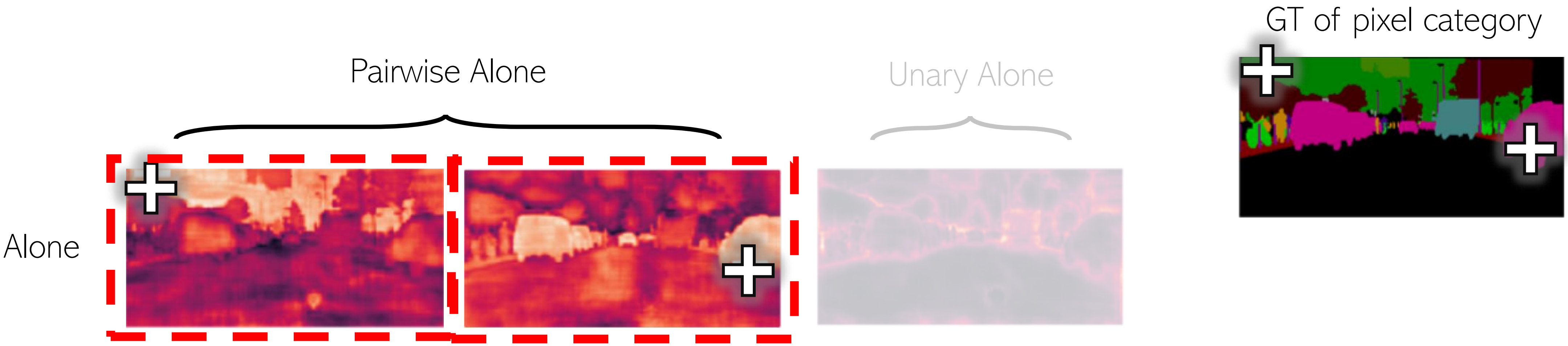
Quantitative results on semantic segmentation (Cityscapes)

- The **unary** term alone outperforms **the standard joint model**
- The pairwise and unary terms are **not well learnt** when combined in the self-attention formulation

Visual Meaning of Each Term

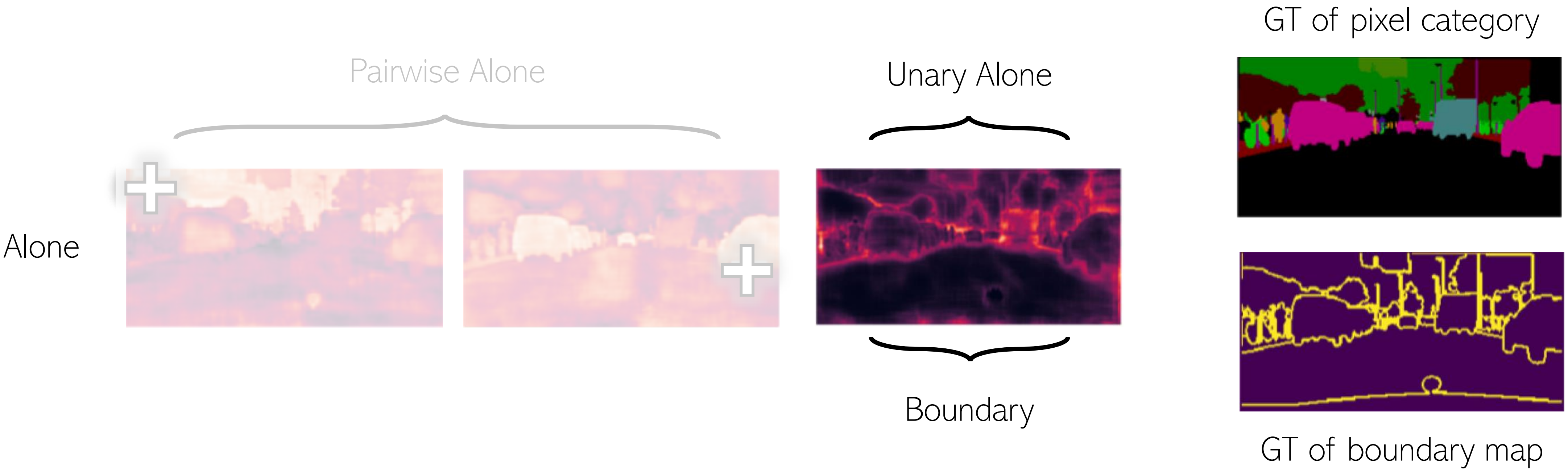


Visual Meaning of Each Term



- The pairwise term tends to learn relations within the **same category region**

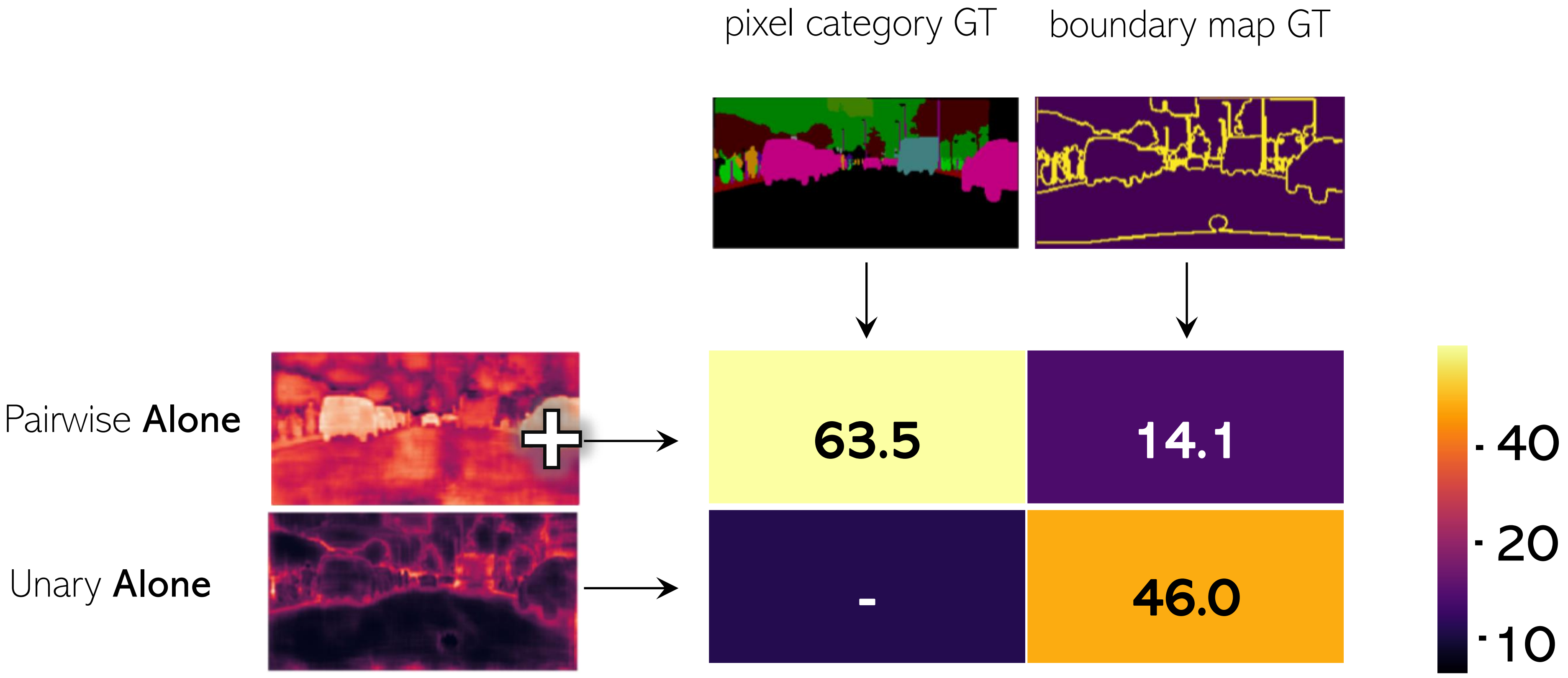
Visual Meaning of Each Term



- The pairwise term tends to learn relations within the **same category region**
- The unary term tends to focus on **boundary pixels**

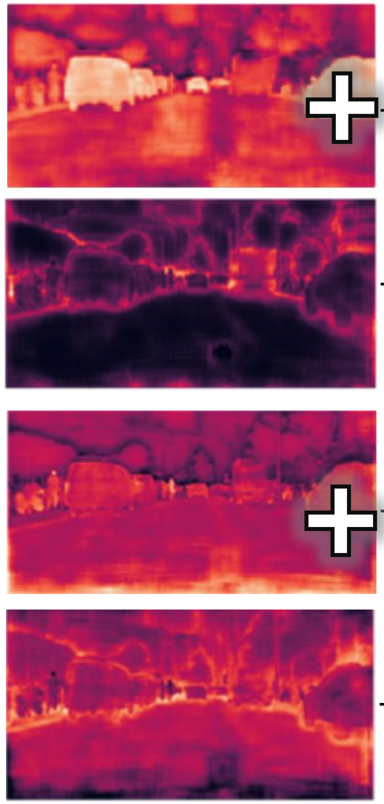
Visual Meaning of Each Term

- Statistical correlation

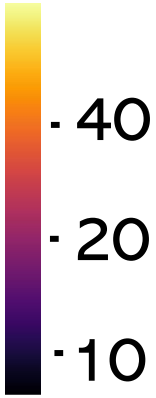


Comparison with Standard 'Joint' Model

- Statistical correlation



	pixel category GT	boundary map GT
Pairwise **Alone**	63.5	14.1
Unary **Alone**	-	46.0
Pairwise **Joint**	31.8	16.0
Unary **Joint**	-	17.2



Why is 'Joint' Worse than 'Alone'?

- Self-Attention is the **multiplicative** combination of pairwise term($\boldsymbol{w_p}$) and unary term ($\boldsymbol{w_u}$) :

$$w(\mathbf{q}_i, \mathbf{k}_j) \sim exp\left((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_q^T \mathbf{k}_j\right)$$

$$= \underbrace{exp\left((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)\right)}_{\text{Pairwise } \boldsymbol{w_p}} \times \underbrace{exp(\boldsymbol{\mu}_q^T \mathbf{k}_j)}_{\text{Unary } \boldsymbol{w_u}}$$

Combination by Multiplication is Bad

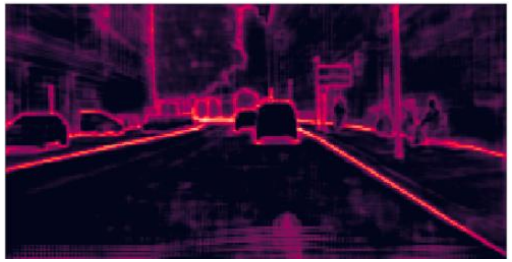
- Multiplication couples two terms in gradient computation

$$\boxed{\frac{\partial L}{\partial \boldsymbol{w_p}}} = \frac{\partial L}{\partial \boldsymbol{w}}\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{w_p}} \sim \frac{\partial L}{\partial \boldsymbol{w}}\boxed{\boldsymbol{w_u}} \qquad \boxed{\frac{\partial L}{\partial \boldsymbol{w_u}}} = \frac{\partial L}{\partial \boldsymbol{w}}\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{w_u}} \sim \frac{\partial L}{\partial \boldsymbol{w}}\boxed{\boldsymbol{w_p}}$$

- Multiplication acts like **intersection**, resulting in empty if two terms encode different visual clues

Pairwise
(Same category region)

$\cap$



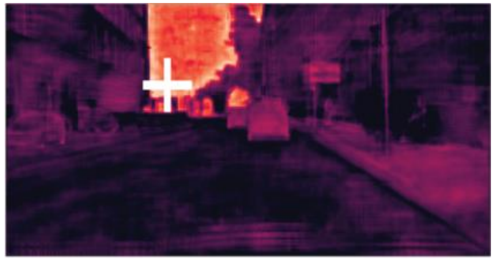
Unary
(Boundary)

$=$

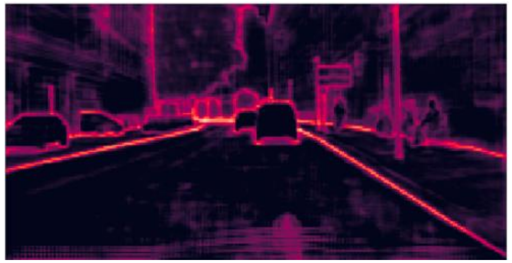
Empty

From Intersection (Mul) to Union (Add)

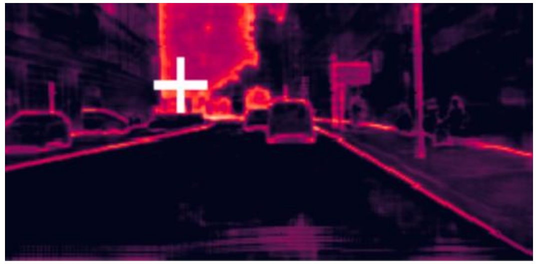
- **Union** instead of intersection:



$\cup$



$=$



- Implement by **addition**

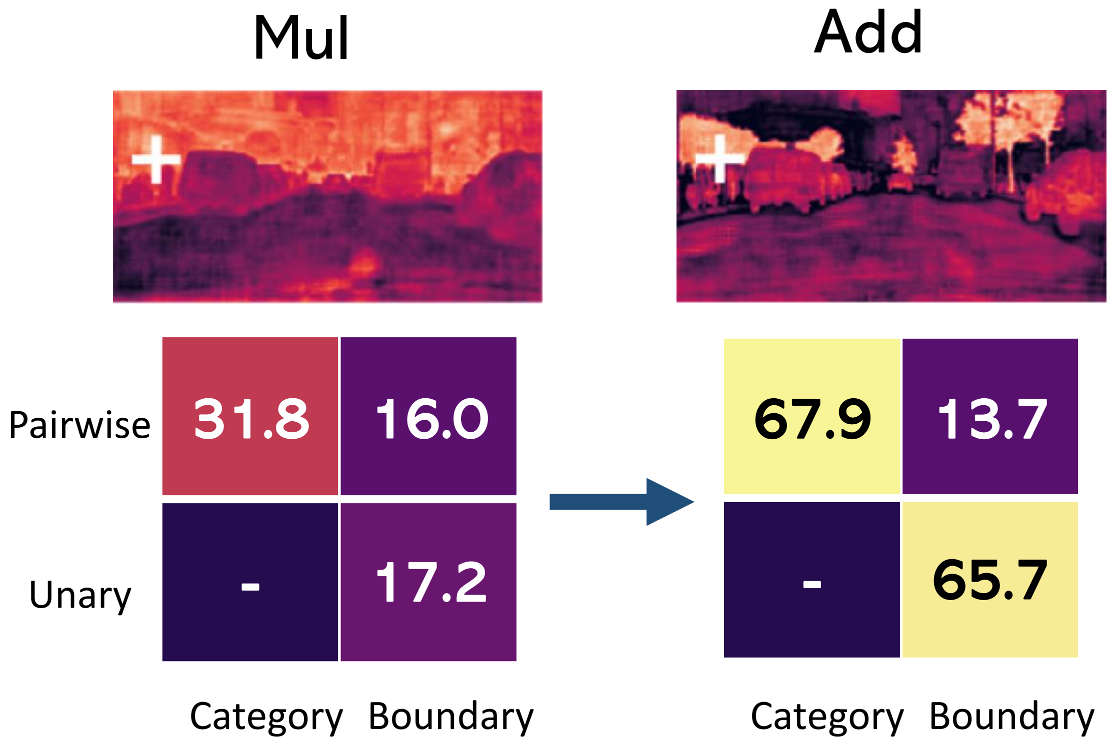
$$w(\mathbf{q}_i, \mathbf{k}_j) \sim exp\big((\mathbf{q}_i - \boldsymbol{\mu}_q)^T (\mathbf{k}_j - \boldsymbol{\mu}_k)\big) + exp(\boldsymbol{\mu}_q^T \mathbf{k}_j)$$

- Gradients are **disentangled** by **addition**

From Intersection (Mul) to Union (Add)

- 0.7 mIoU improvements on Cityscapes
- Significantly clearer visual meaning

method	mIoU
Baseline	75.8%
Mul (Self-Attention)	78.5%
Add (Ours)	**79.2%**



Are There Other Coupling Factors?

- The key is **shared** in the pairwise term and unary term
- The shared key can be further **disentangled**:

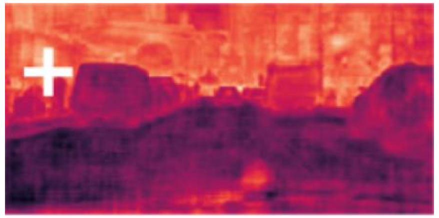
$$w(\mathbf{q}_i, \mathbf{k}_j) \sim \overbrace{exp\Big((\mathbf{q}_i - \boldsymbol{\mu}_q)^T(\boxed{\mathbf{k}_j} - \boldsymbol{\mu}_k)\Big)}^{\text{pairwise}} + \overbrace{exp(\boxed{\mathbf{k}_j})}^{\text{unary}}$$

$$exp\Big((\mathbf{q}_i - \boldsymbol{\mu}_q)^T(\boxed{\mathbf{W}^{\color{red}{p}}\mathbf{k}_j} - \boldsymbol{\mu}_k)\Big) + exp(\boxed{\mathbf{W}^{\color{red}{u}}\mathbf{k}_j})$$

Disentangle the Key Transformations

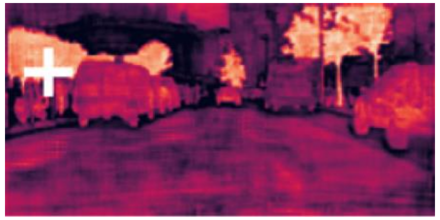
- The pairwise and unary terms learn clearer visual meaning

Mul



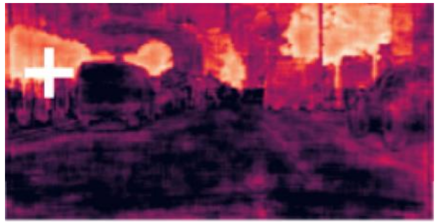
	Category	Boundary
Pairwise	31.8	16.0
Unary	-	17.2

Add (Key Shared)



	Category	Boundary
Pairwise	67.9	13.7
Unary	-	65.7

Add (Separate Keys)



	Category	Boundary
Pairwise	75.9	13.0
Unary	-	69.6

Results by Two Disentangle Techniques

- **2.0** mIoU improvements than self-attention
- **4.7** mIoU improvements than baseline

method	mIoU
Baseline	75.8%
Mul (Self-Attention)	78.5%
Add(Shared key)	79.2%
Add(Disentangled key)	**80.5%**

On Three Semantic Segmentation Benchmarks

- Disentangled Non-Local Neural Networks
  - Multiplication to Addition
  - Shared keys to Disentangled keys

method	backbone	mIoU(%)
Deeplab v3	ResNet101	81.3
OCNet	ResNet101	81.7
Self-Attention	ResNet101	80.8
Ours	ResNet101	**82.0**
HRNet	HRNetV2-W48	81.9
Self-Attention	HRNetV2-W48	82.5
Ours	HRNetV2-W48	**83.0**

Cityscapes

method	backbone	mIoU(%)
ANN	ResNet101	52.8
EMANet	ResNet101	53,1
Self-Attention	ResNet101	50.3
Ours	ResNet101	**54.8**
HRNet v2	HRNetV2-W48	54.0
Self-Attention	HRNetV2-W48	54.2
Ours	HRNetV2-W48	**55.3**

ADE20K

method	backbone	mIoU(%)
ANN	ResNet101	45.24
OCNet	ResNet101	45.45
Self-Attention	ResNet101	44.67
Ours	ResNet101	**45.90**
HRNet v2	HRNetV2-W48	42.99
Self-Attention	HRNetV2-W48	44.82
Ours	HRNetV2-W48	**45.82**

PASCAL-Context

Disentangled Non-Local Network is General

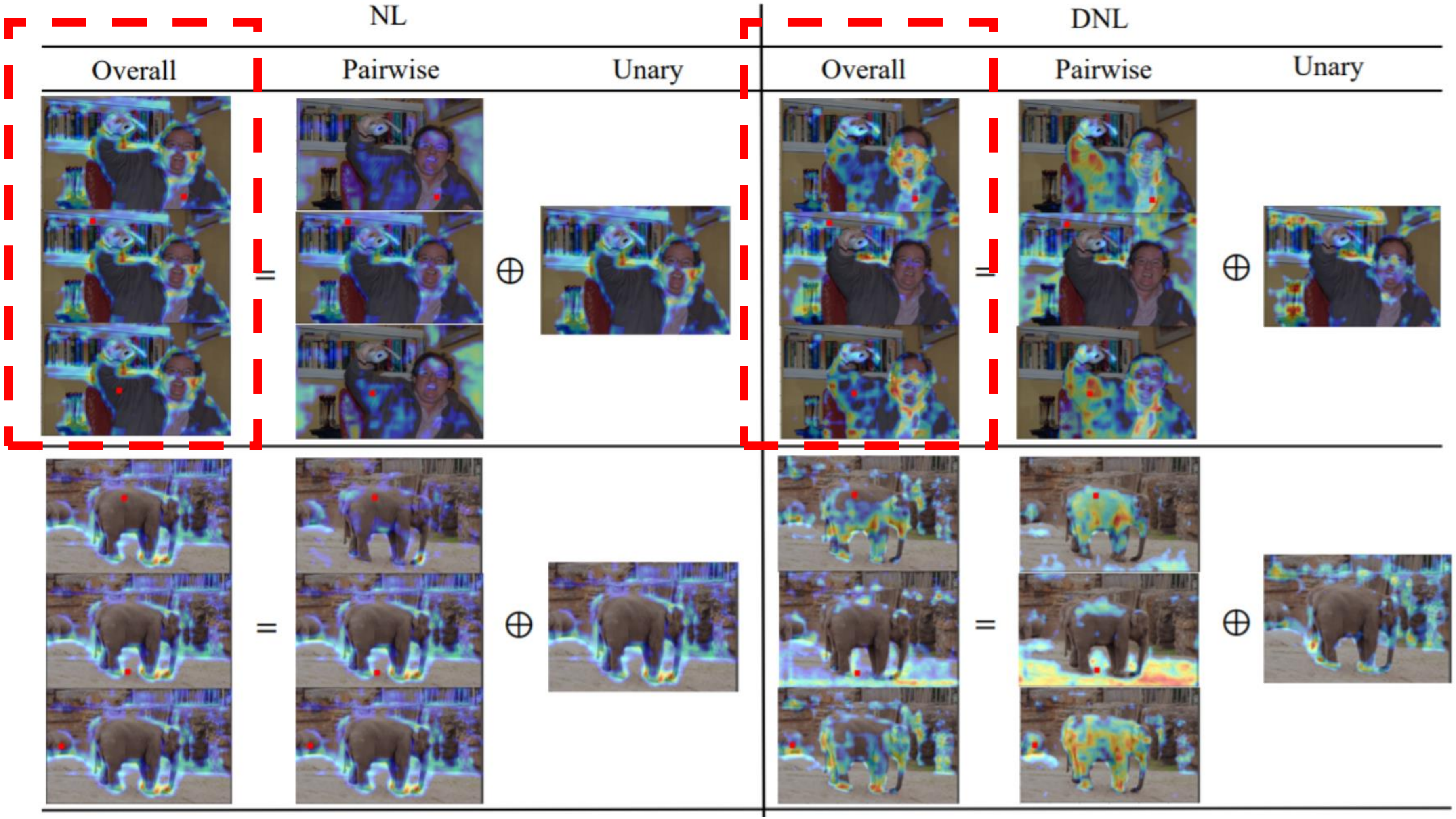
- Object detection & instance segmentation, COCO2017 dataset

method	$\mathrm{mAP}^{\mathrm{bbox}}$	$\mathrm{mAP}^{\mathrm{mask}}$
Baseline	38.8	35.1
Self-Attention	40.1	36.0
Disentangled Self-Attention (ours)	**41.4**	**37.3**

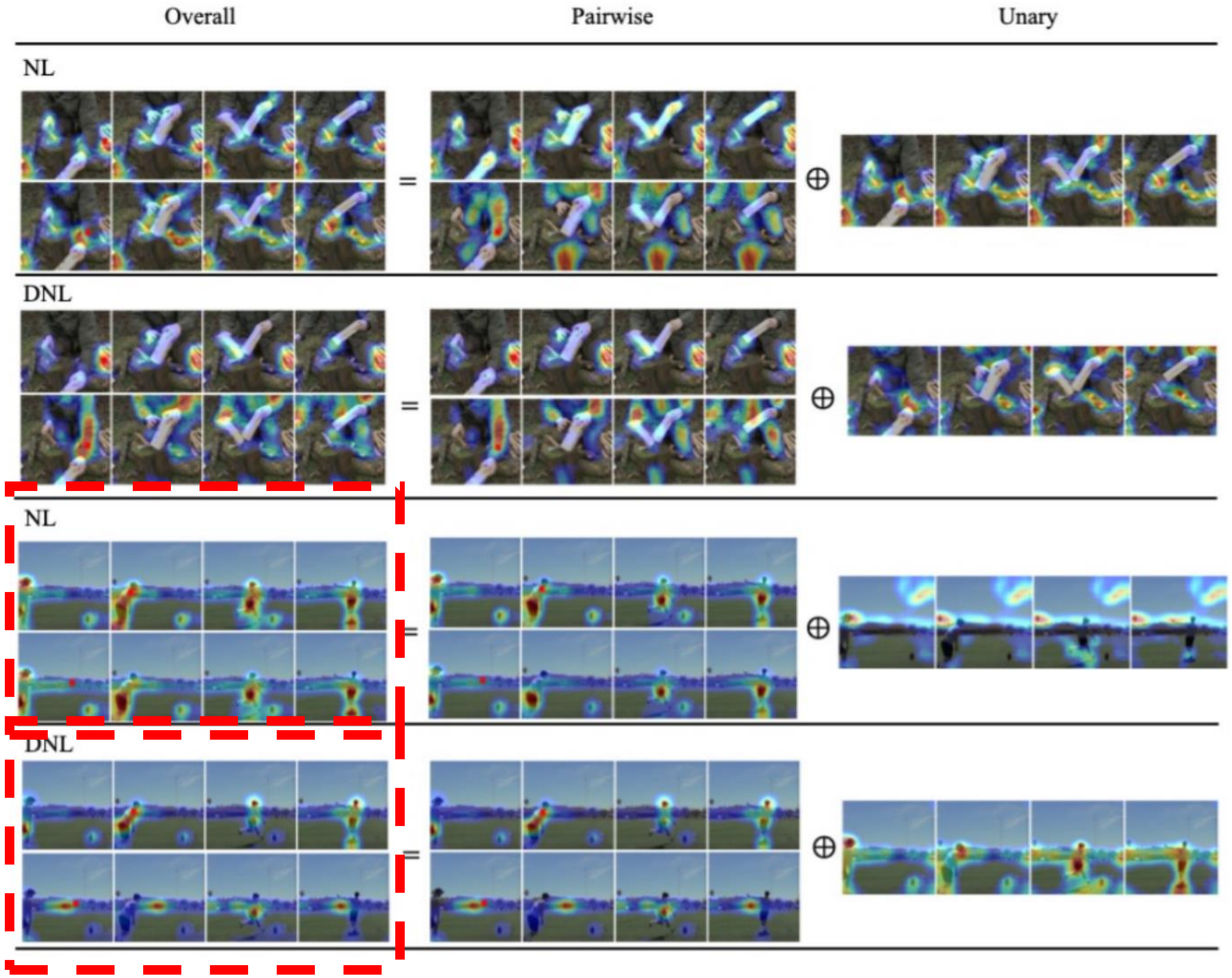
- Action recognition, Kinetics dataset

method	**Top-1 Acc**	**Top-5 Acc**
Baseline	74.9	91.9
Self-Attention	75.9	92.2
Disentangled Self-Attention (ours)	**76.3**	**92.7**

Visualization (Object Detection)



Visualization（Action Recognition)



Summary

- Part I: Self-Attention Models for Visual Recognition (Application View)
  - Pixel-to-Pixel, Object-to-Pixel, Object-to-Object
  - **A strong competitor; complementary to existing architectures; SOTA in video applications**
  - **There is still much room to improve!**
- Part II: Diagnosis and Improvement (Modeling View)
  - Are self-attention models learnt well on visual tasks?
    - **No [GCNet, ICCVW2019],**
  - How can it be more effective?
    - **[DNL, Tech Report 2020]**

Yue Cao*, Jiarui Xu* , Stephen Lin, Fangyun Wei and Han Hu. ***GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond***. ICCVW'2019

Minghao Yin *, Zhuliang Yao*, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. ***Disentangled Non-Local Neural Networks***. Tech Report 2020

Thanks All!