Towards Universal Models with NLP for Computer Vision

Transformer and Attention Mechanism

Han Hu

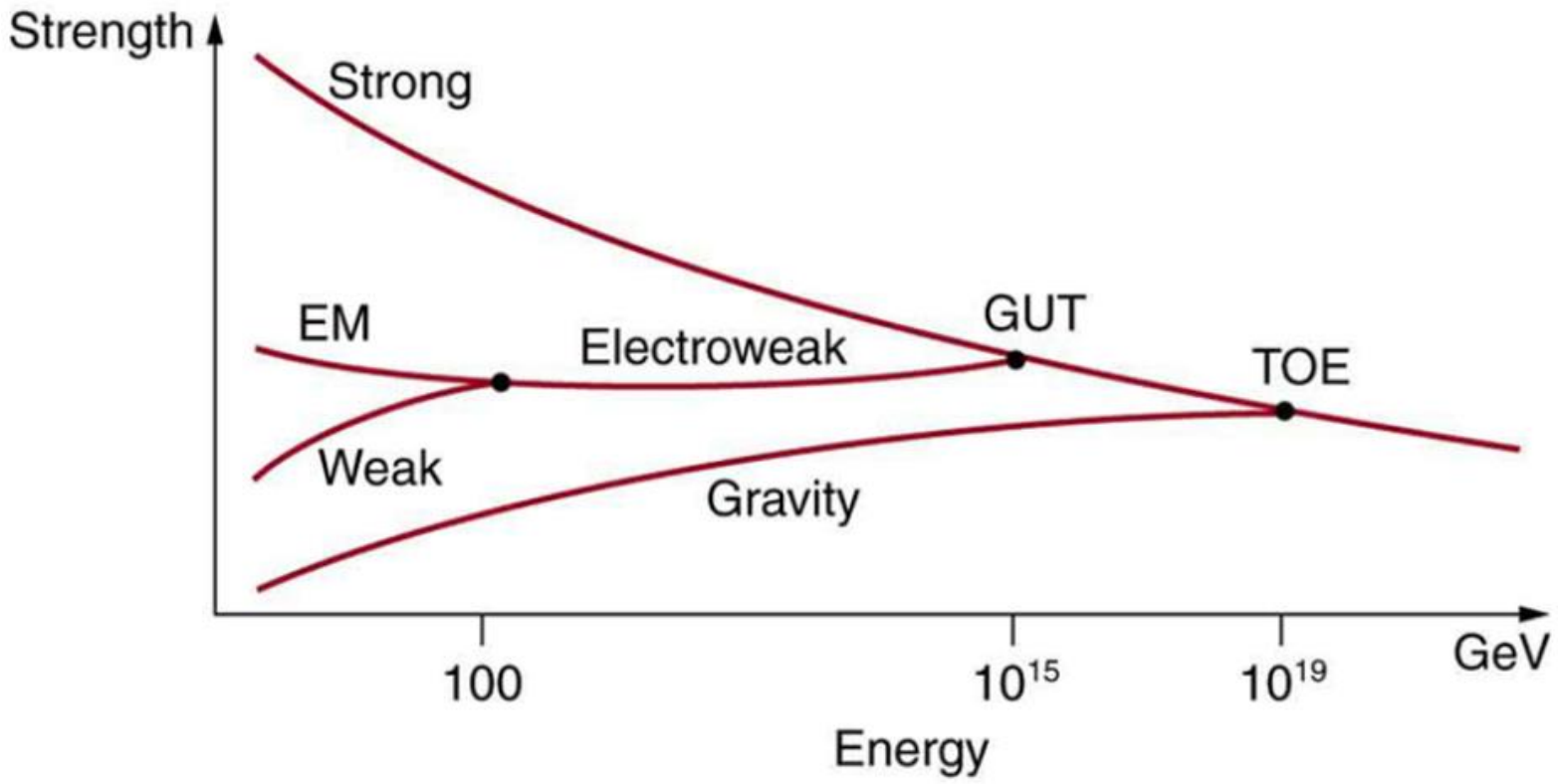
Visual Computing Group

Microsoft Research Asia (MSRA)

April 8th, 2021

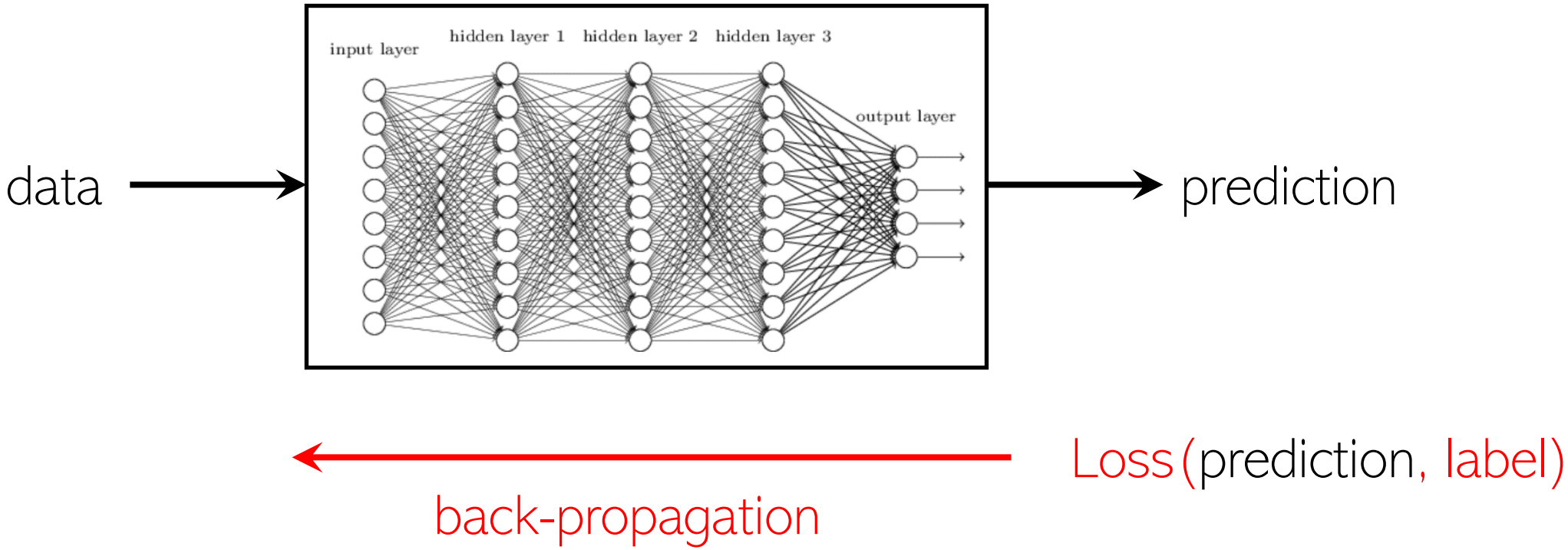
Grand Unification Theory in Physic

- The Holy Grail in Physics



A Unification Story for AI

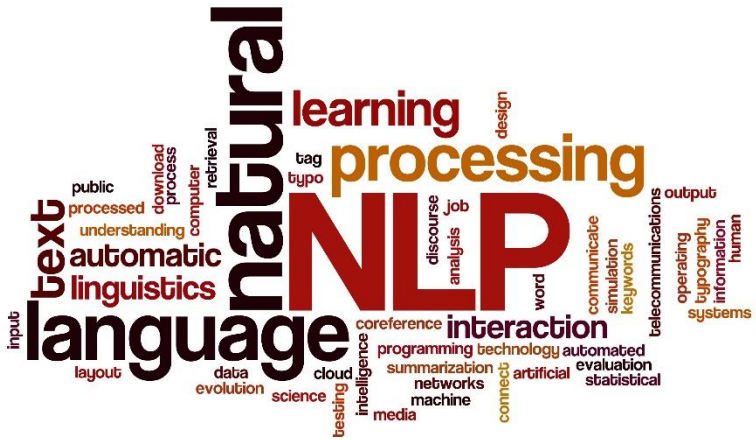
- The deep learning era
  - a universal pipeline



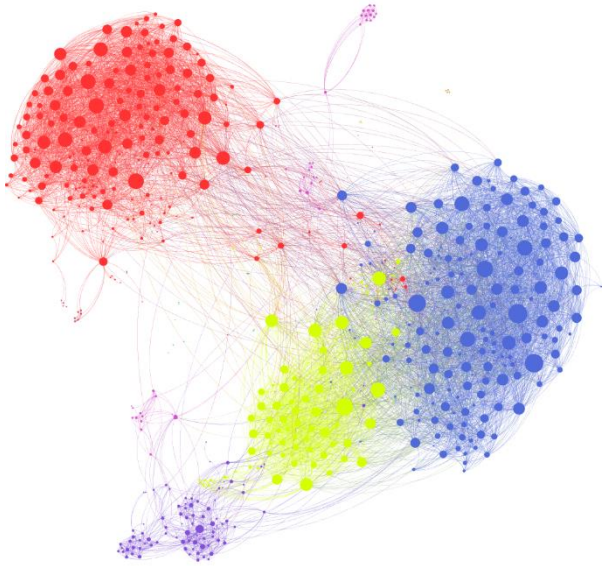
A Unification Story for AI

- What about models?

convolution



self-attention
(Transformers)



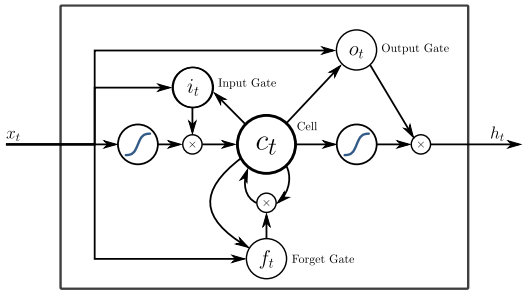
graph networks

Why universal models?

- Facilitate joint modeling of visual and textual signals
- Modeling and learning knowledge from both domains can be more deeply shared

- Pursuing universality, which is beauty itself

Model Evolution in NLP



1995

LSTM

Jürgen Schmidhuber

2014

GRU

Google

2017

Transformers (self-attention)

Google Brain

2014

Deep RNN

Baidu

2015

RNN+attention

CIFAR

Transformers

- The cornerstone architecture in NLP
- Used in the revolutionary unsupervised pretraining methods (BERT, GPT)

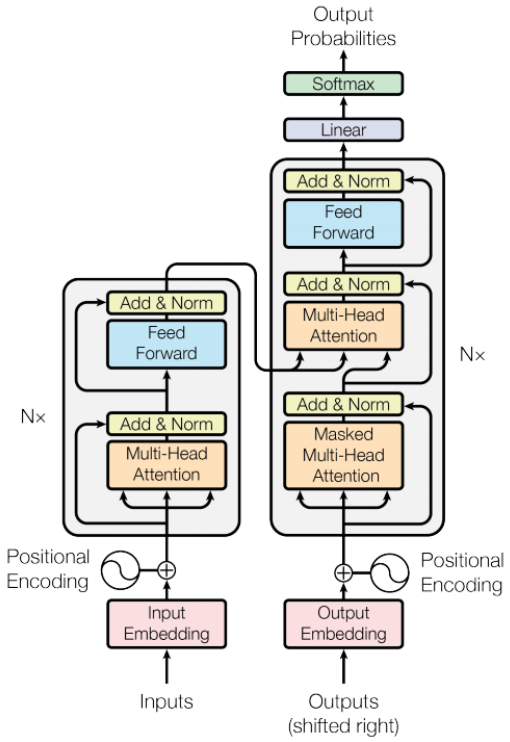
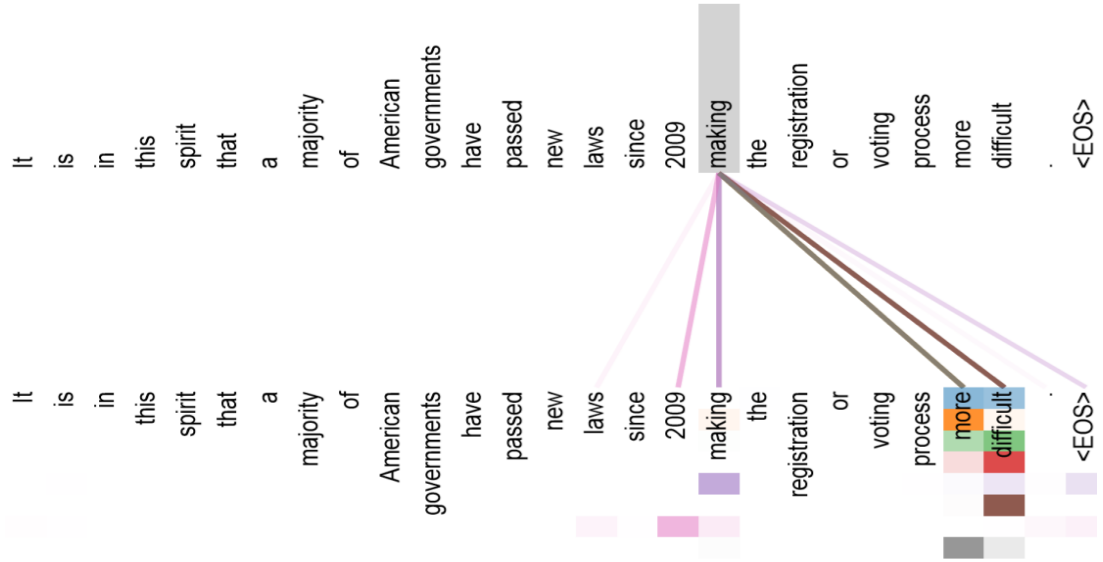


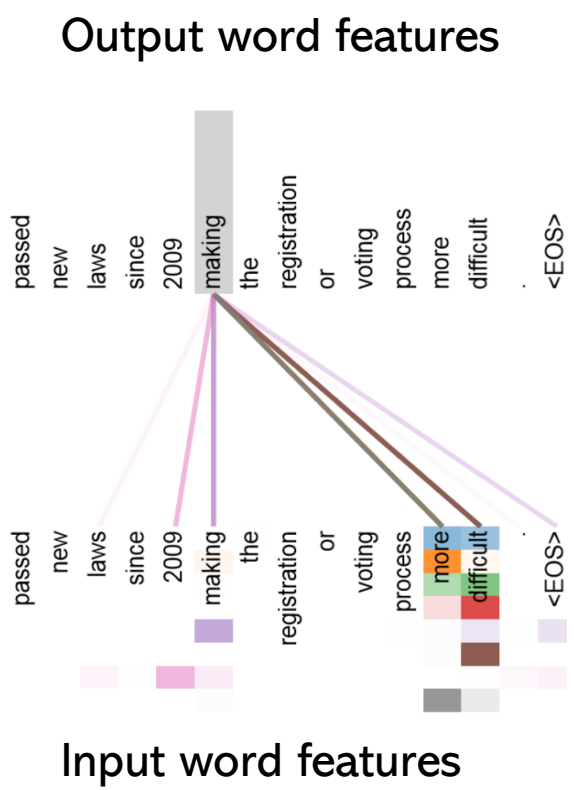
Figure 1: The Transformer - model architecture.

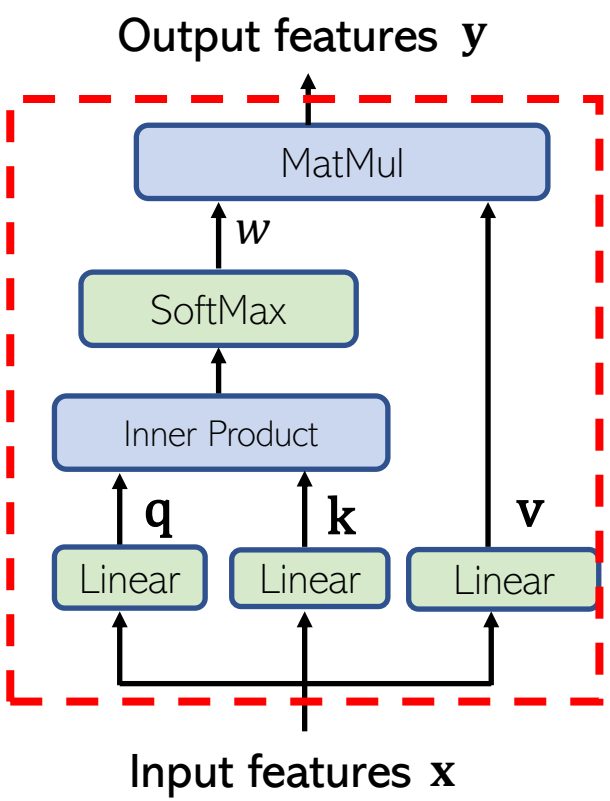


Ashish Vaswani et al, **Attention is all you need**, NeurIPS'2017

Self-Attention Unit

- Transforms the word/token input feature by encoding its relationship with other words/tokens
- A weighted average of **Value**, where the weight is the normalized inner product of **Query** and **Key**





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

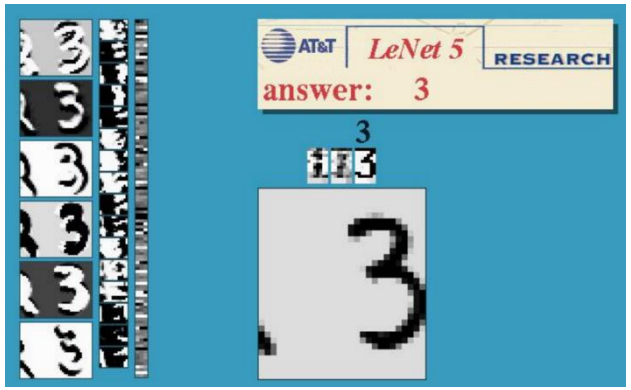
$$w(\mathbf{q}_i, \mathbf{k}_j) \sim exp(\mathbf{q}_i^T \mathbf{k}_j)$$

Model Evolution in CV

1989

Convolution

Yann LeCun



LeNet, AlexNet, GoogleNet, VGGNet, ResNet …

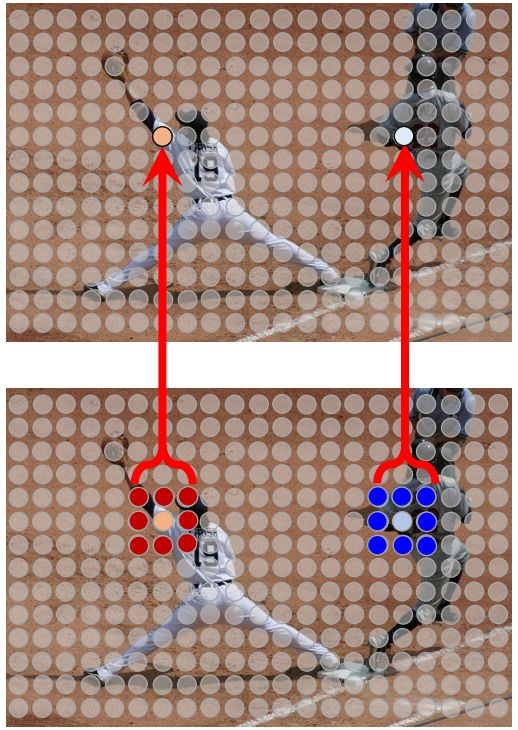
Deformable Convolution (2017)

regular convolution

Output feature
(grid)

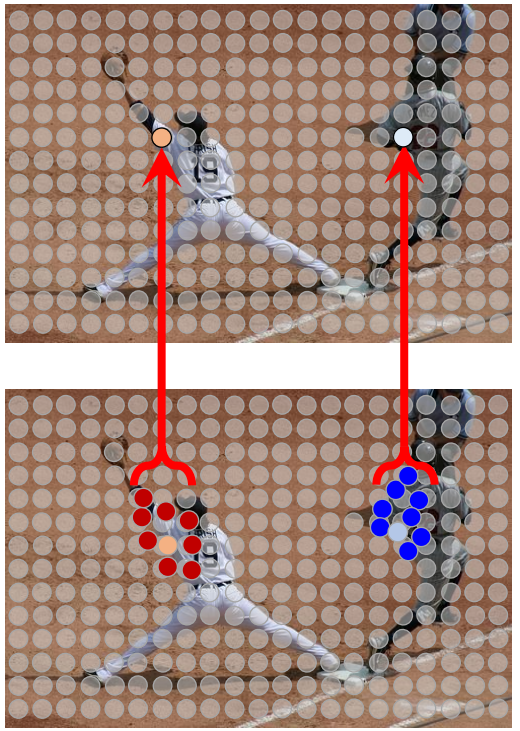
Sampling
(grid)

Input feature
(grid)



$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n)$$

deformable convolution



Output feature
(grid)

Sampling
(non-grid)

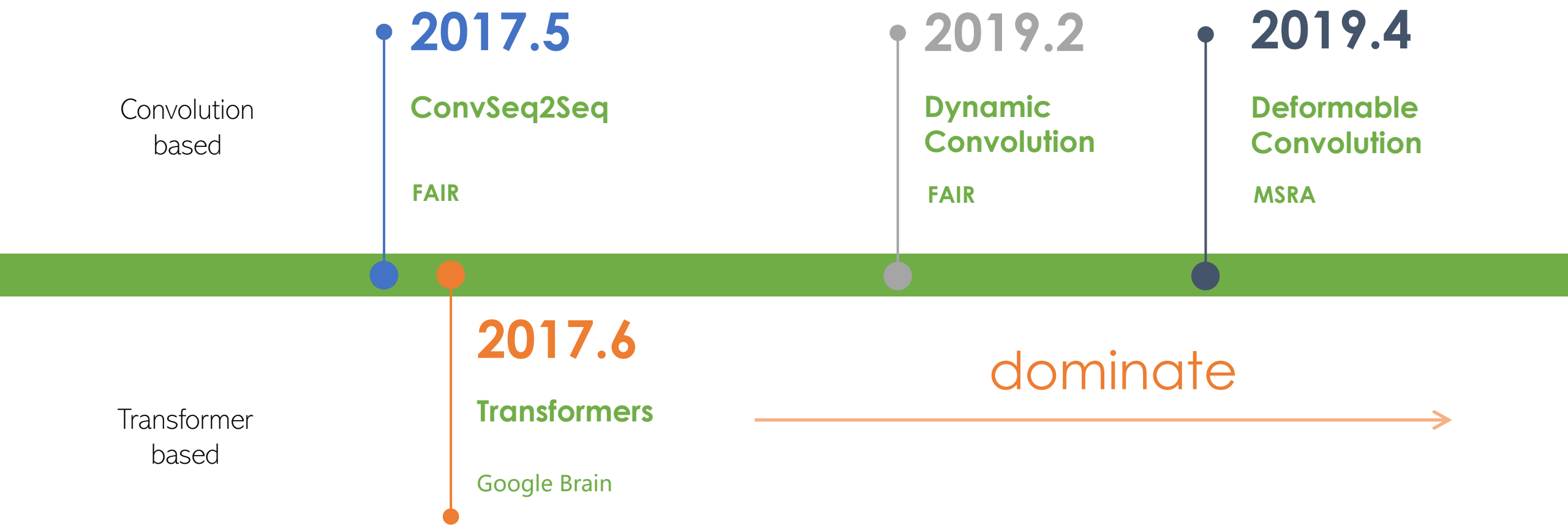
Input feature
(grid)

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n)$$

Dai et al. Deformable Convolution Networks. ICCV 2017

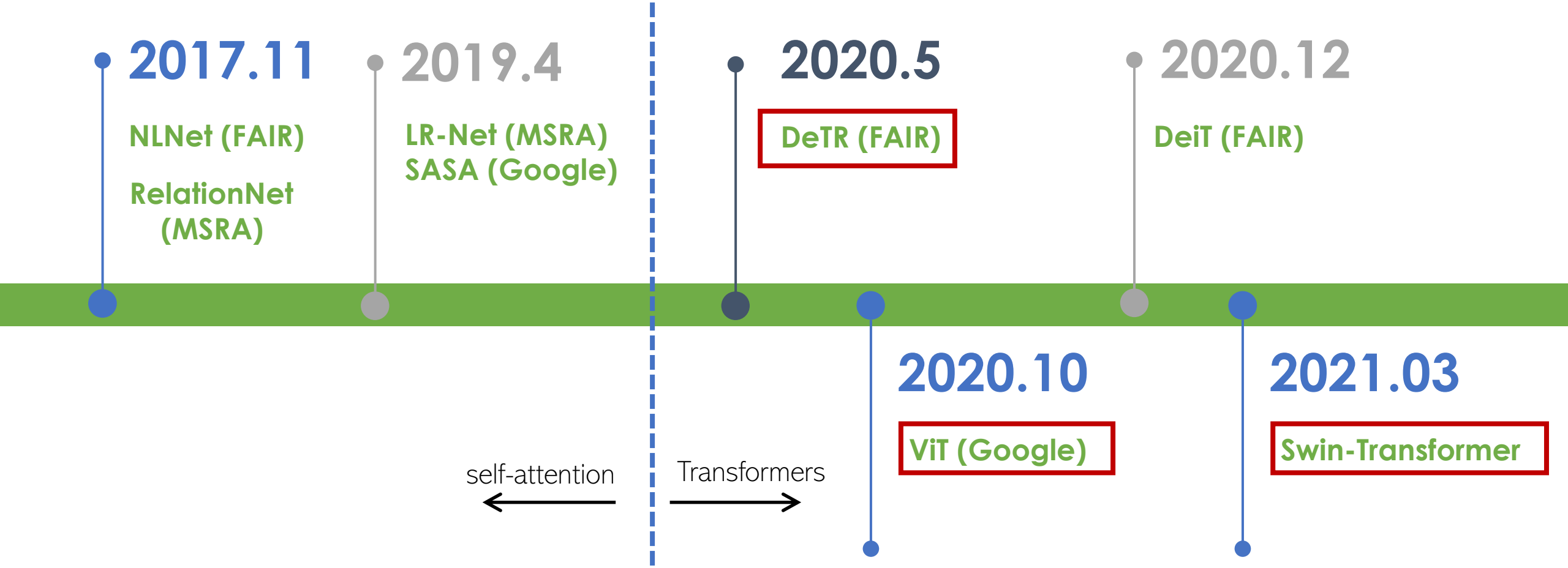
Can NLP/CV share the same basic modules?

- Adapting convolution layers for NLP modeling



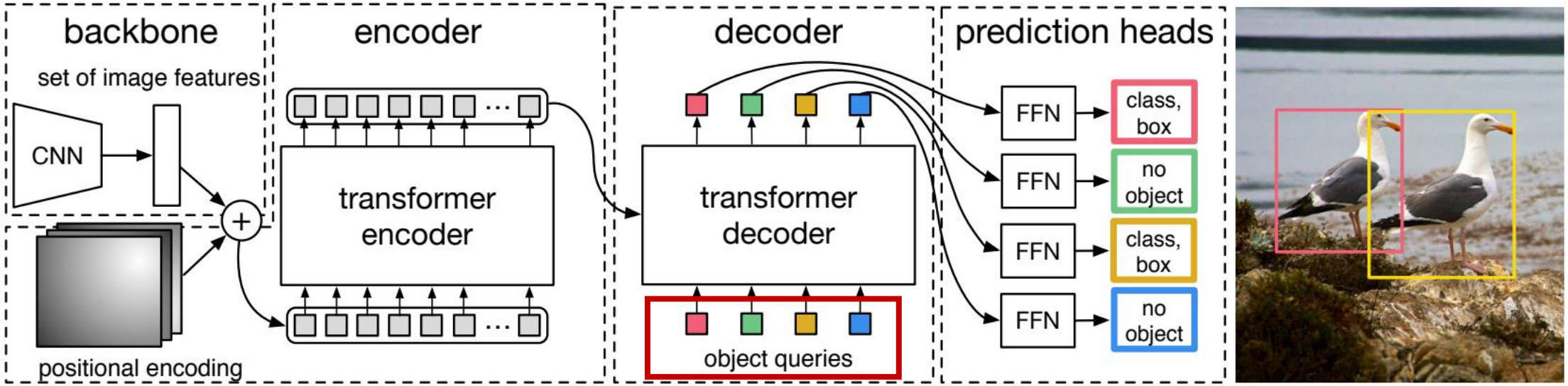
Can NLP/CV share the same basic modules?

- Adapting self-attention/Transformers models for visual modeling



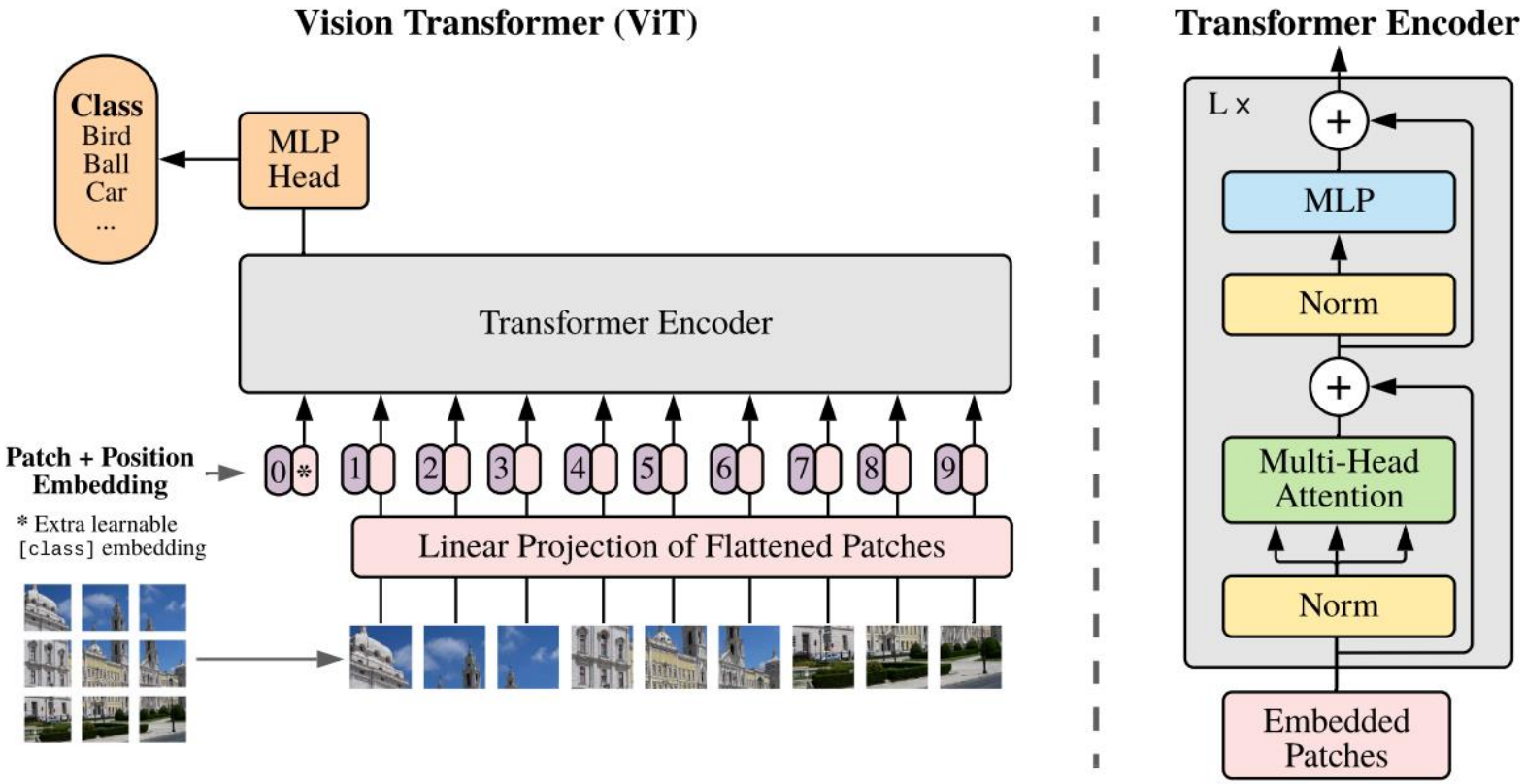
DETR for Object Detection

- End-to-end object detection without using priors



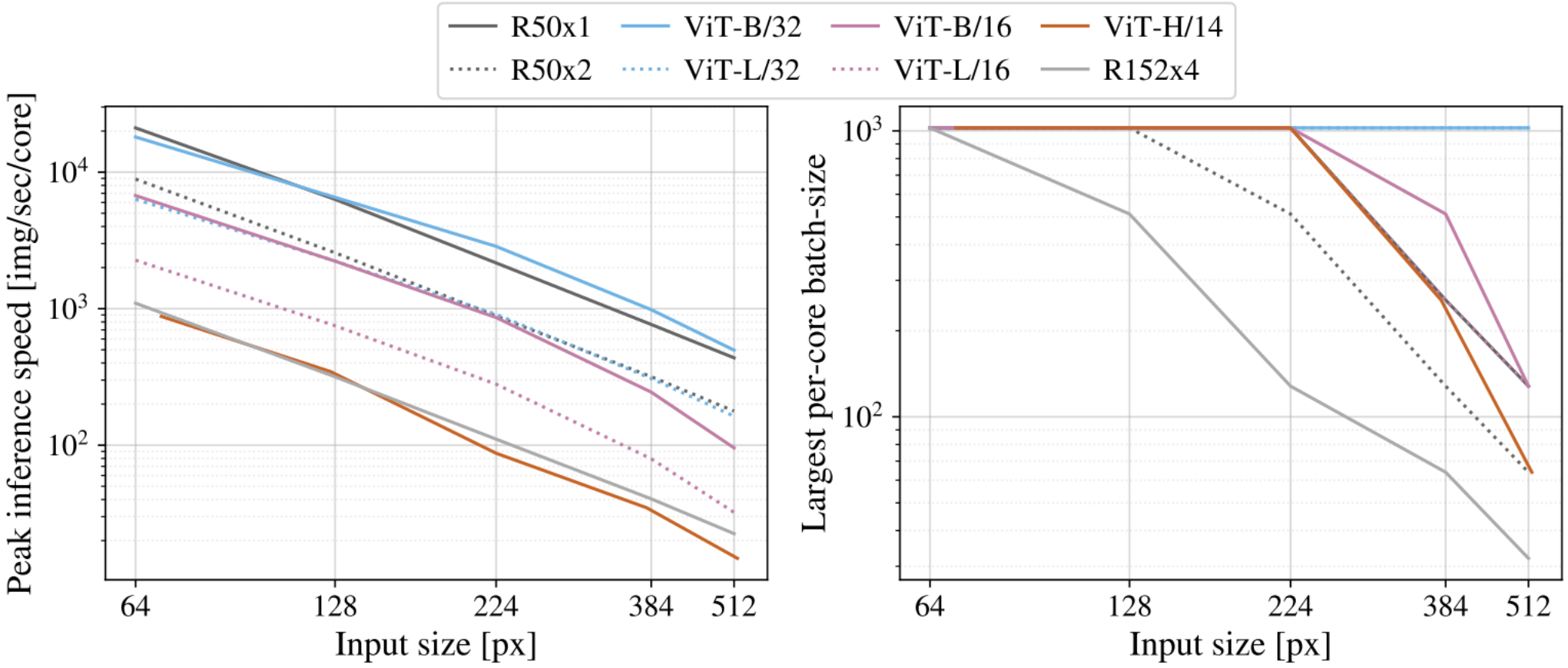
Nicolas Carion et al. End-to-End Object Detection with Transformers. ECCV 2020

Vision Transformers (ViT)



Alexey Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR' 2021

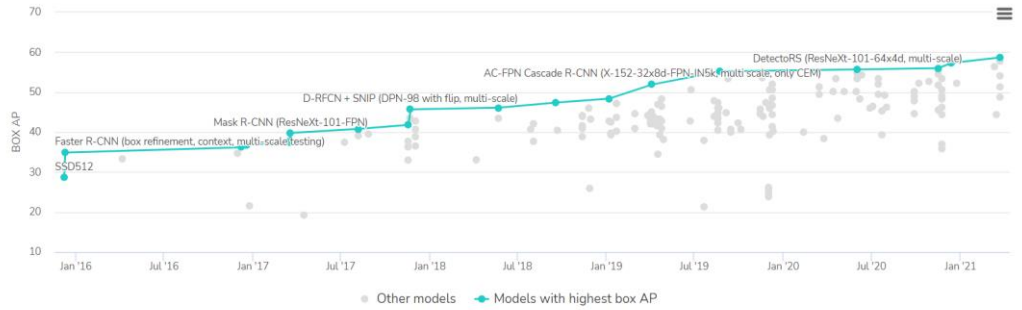
Good Real Speed of ViT



Swin Transformer: a general-purpose backbone

Object Detection on COCO test-dev

Leaderboard | Dataset



View: box AP | All models | Edit

RANK	MODEL	BOX AP	AP50	AP75	APS	APM	APL	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	Swin-L (HTC++, multi scale)	58.7						×	Swin Transformer: Hierarchical Vision Transformer using Shifted Windows			2021
2	Swin-L (HTC++, single scale)	57.7						×	Swin Transformer: Hierarchical Vision Transformer using Shifted Windows			2021
3	Cascade Eff-B7 NAS-FPN (1280, self-training Copy Paste, single-scale)	57.3						✓	Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation			2020
4	CenterNet2 (Res2Net-101-DCN-BiFPN, self-training, 1560 single-scale)	56.4	74.0	61.6	38.7	59.7	68.6	✓	Probabilistic two-stage detection			2021
5	YOLOv4-P7 (CSP-P7, multi-scale)	56.0	73.3	61.2	38.9	60.0	68.6	×	Scaled-YOLOv4: Scaling Cross Stage Partial Network			2020

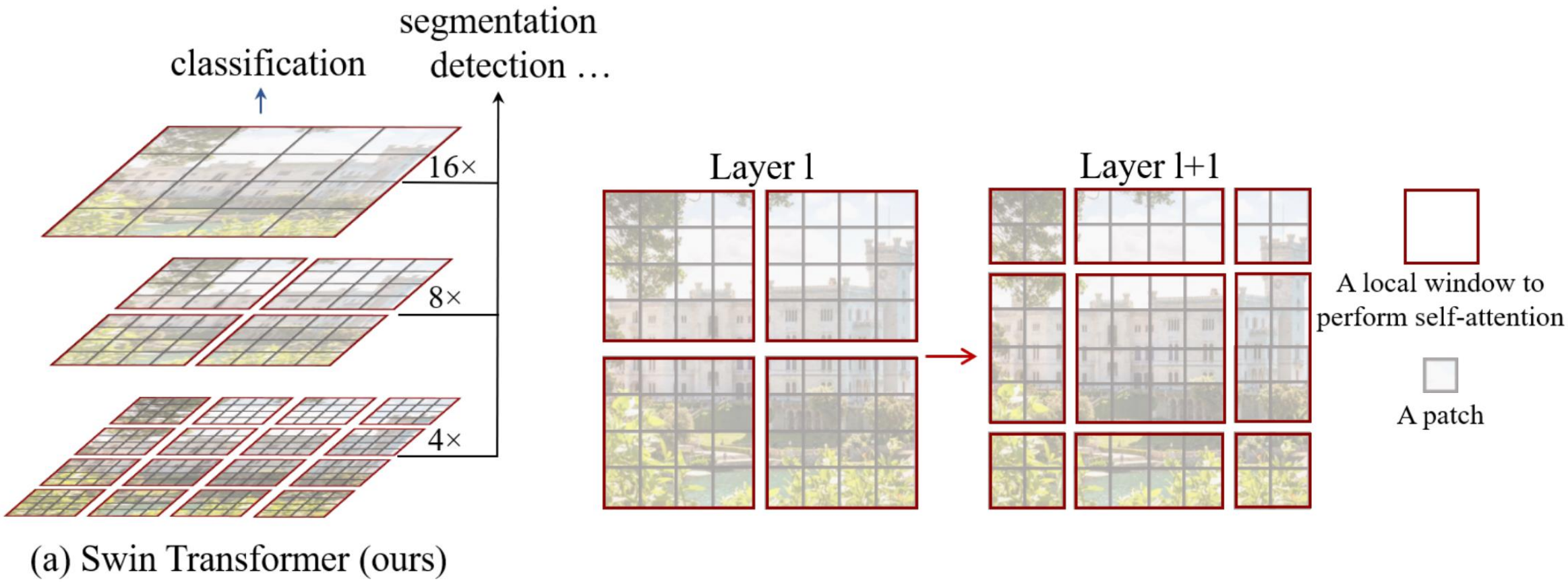
Semantic Segmentation on ADE20K val

Leaderboard | Dataset

View: mIoU | Edit

RANK	MODEL	MIOU	PIXEL ACCURACY	PAPER	CODE	RESULT	YEAR
1	Swin-L (UperNet, ImageNet-22k pretrain)	53.5		Swin Transformer: Hierarchical Vision Transformer using Shifted Windows			2021
2	DPT-Hybrid	49.02	83.11	Vision Transformers for Dense Prediction			2021
3	ResNeSt-200	48.36		ResNeSt: Split-Attention Networks			2020

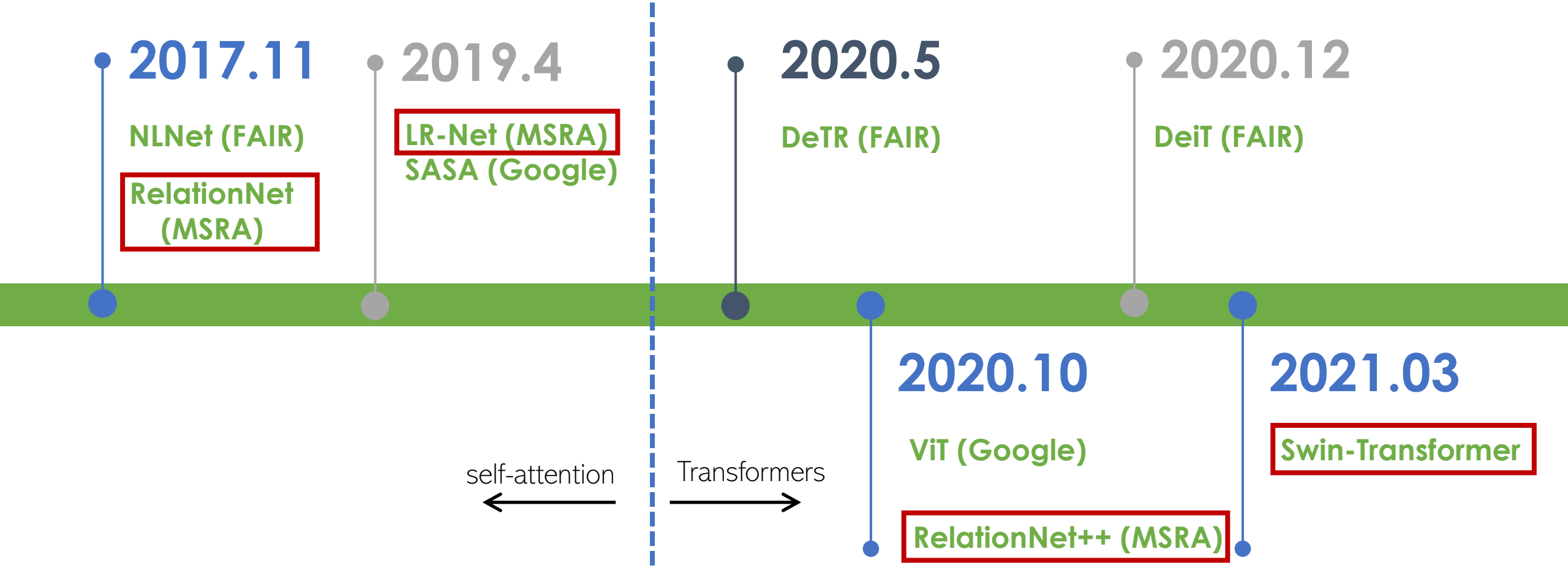
Swin Transformer: a general-purpose backbone



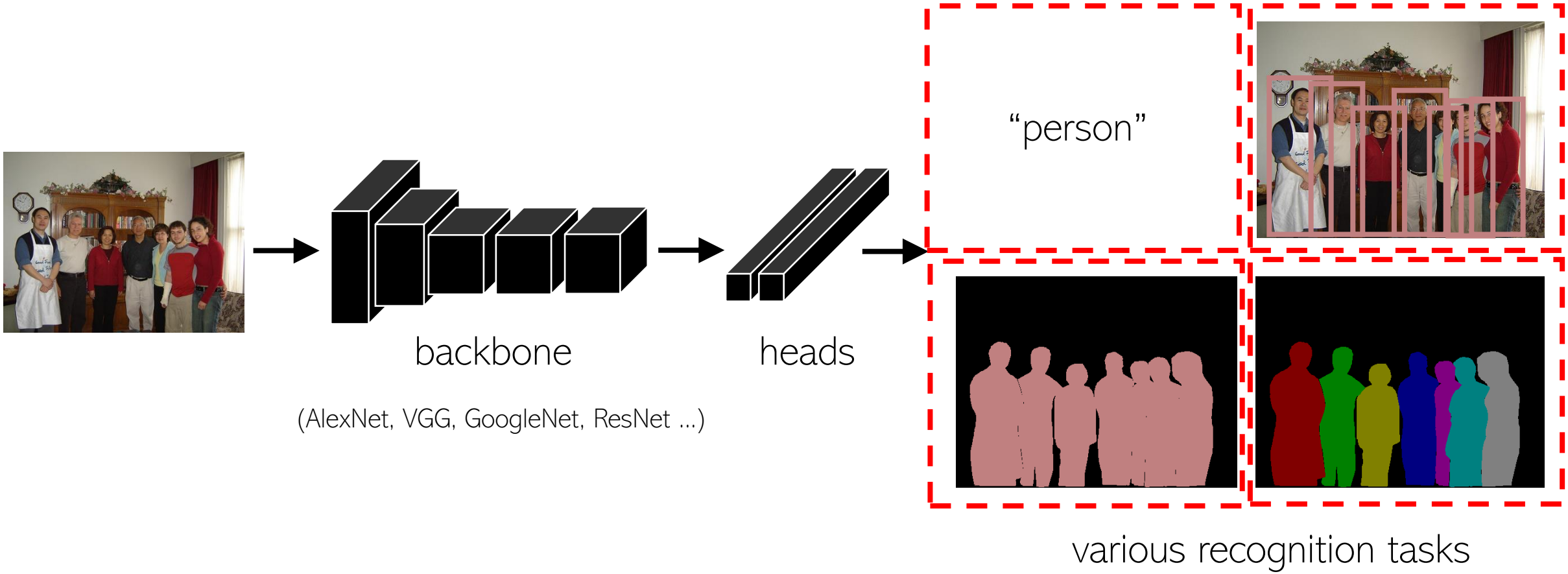
(a) Swin Transformer (ours)

https://github.com/microsoft/Swin-Transformer

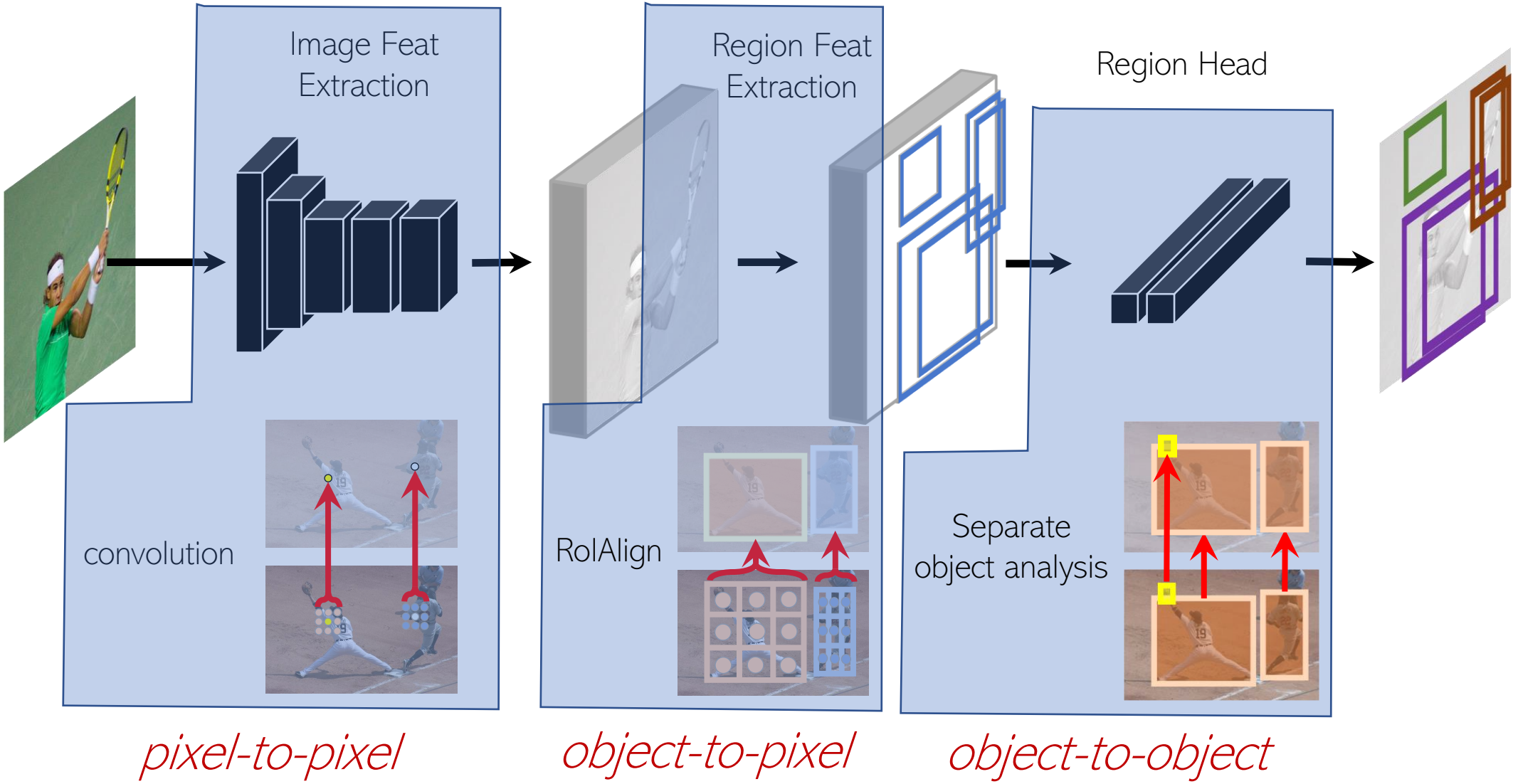
How did we get here?



Visual Recognition Paradigm

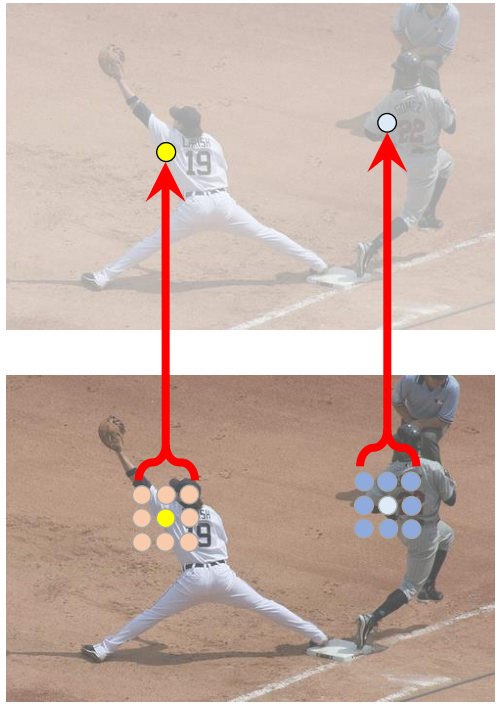


An Object Detection Example



Relationship Modeling of Basic Visual Elements

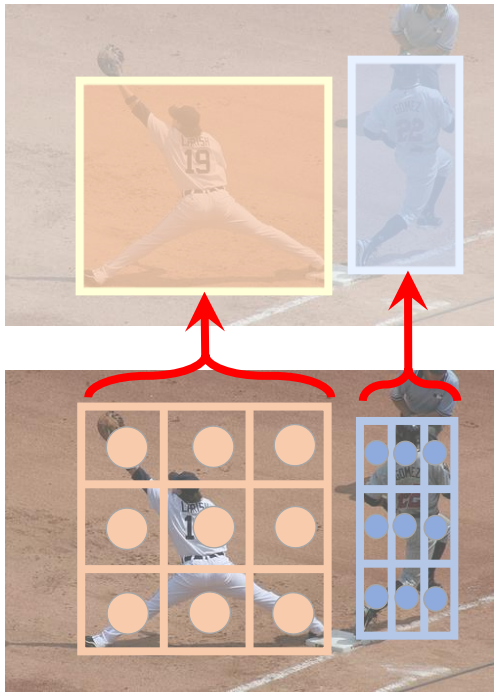
pixel-to-pixel



Convolution Variants

Self-attention

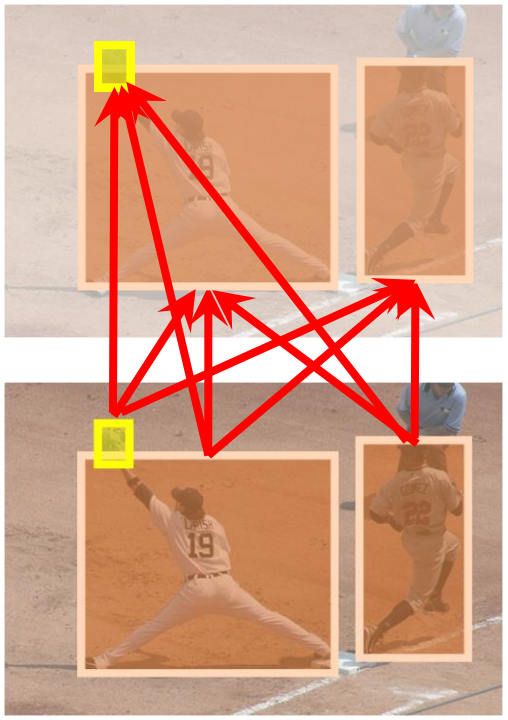
object-to-pixel



RoIAlign

Self-attention

object-to-object

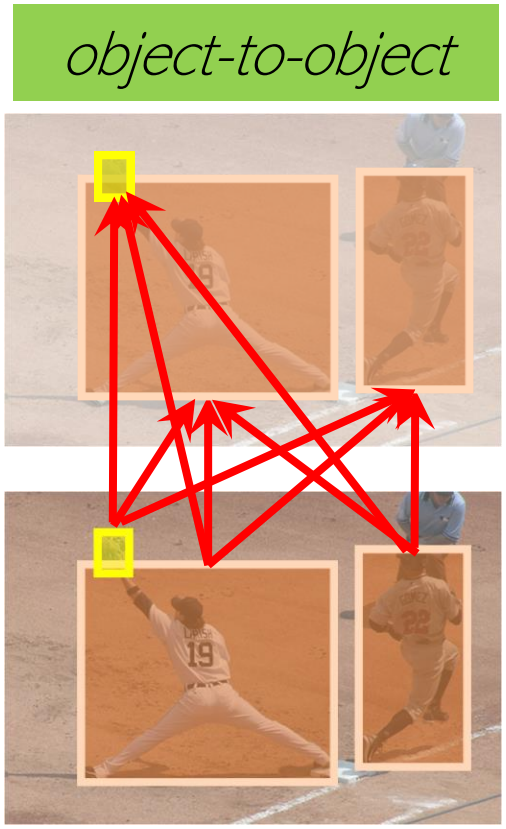


None

Self-attention

our study timeline

Object-to-Object Relation Modeling

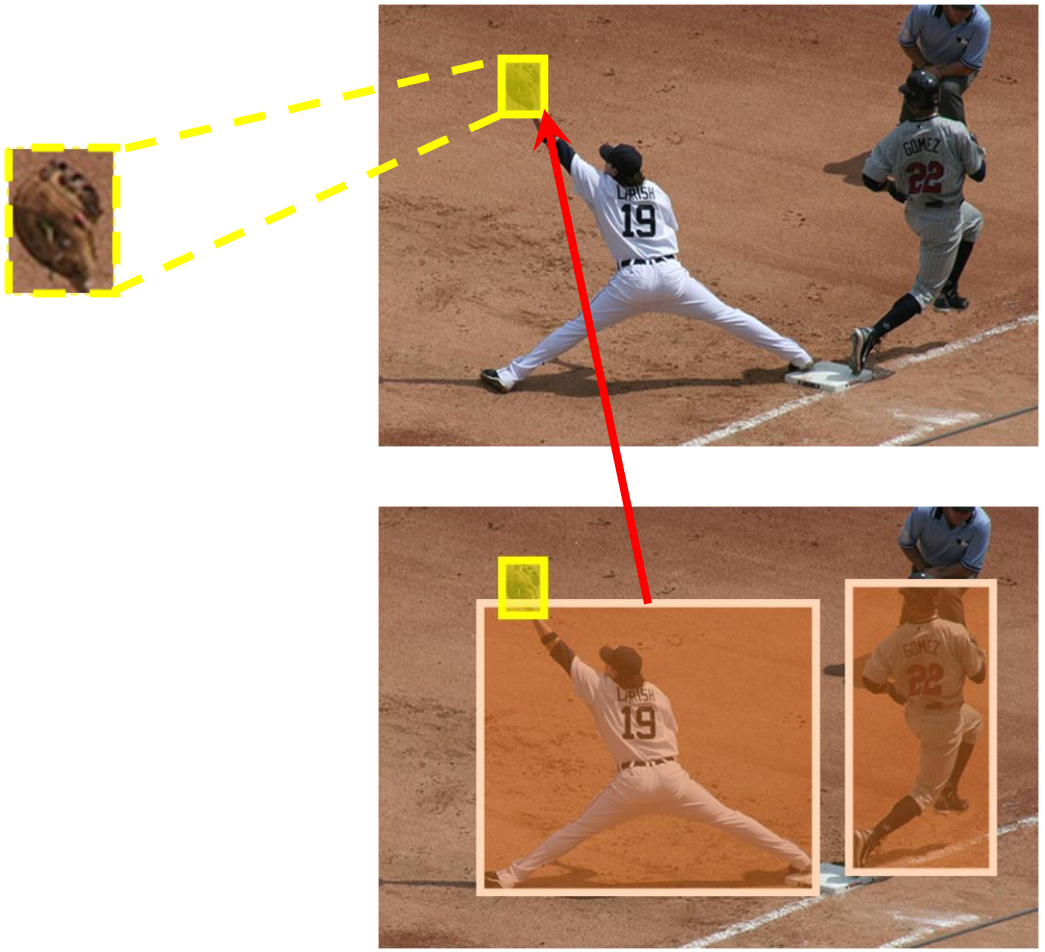


None ⟶ **Self-Attention**

- Object Detection
  - RelationNet [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]

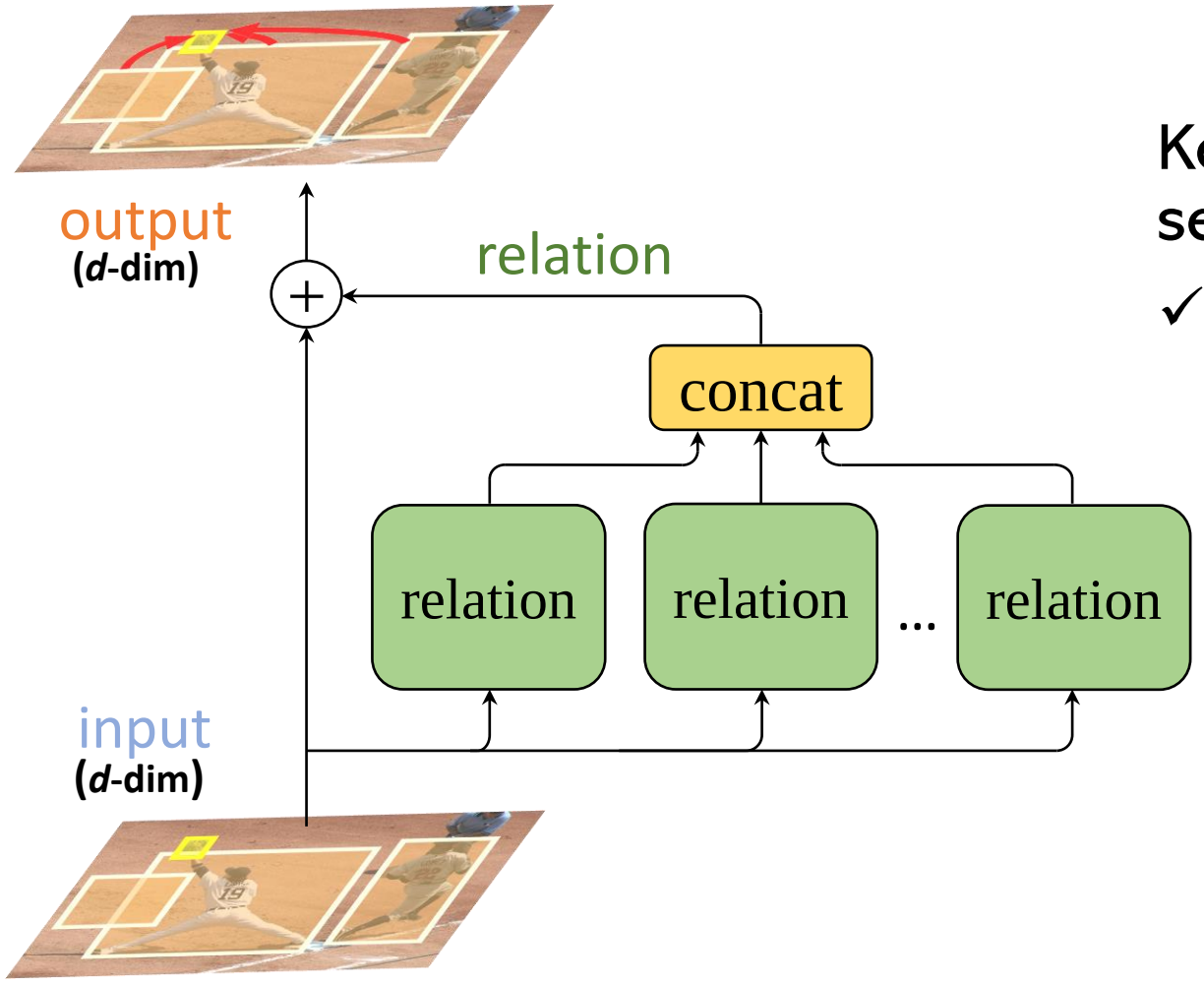
Object-to-Object Relation Modeling

Object-to-Object Relation Modeling



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

Object Relation Module

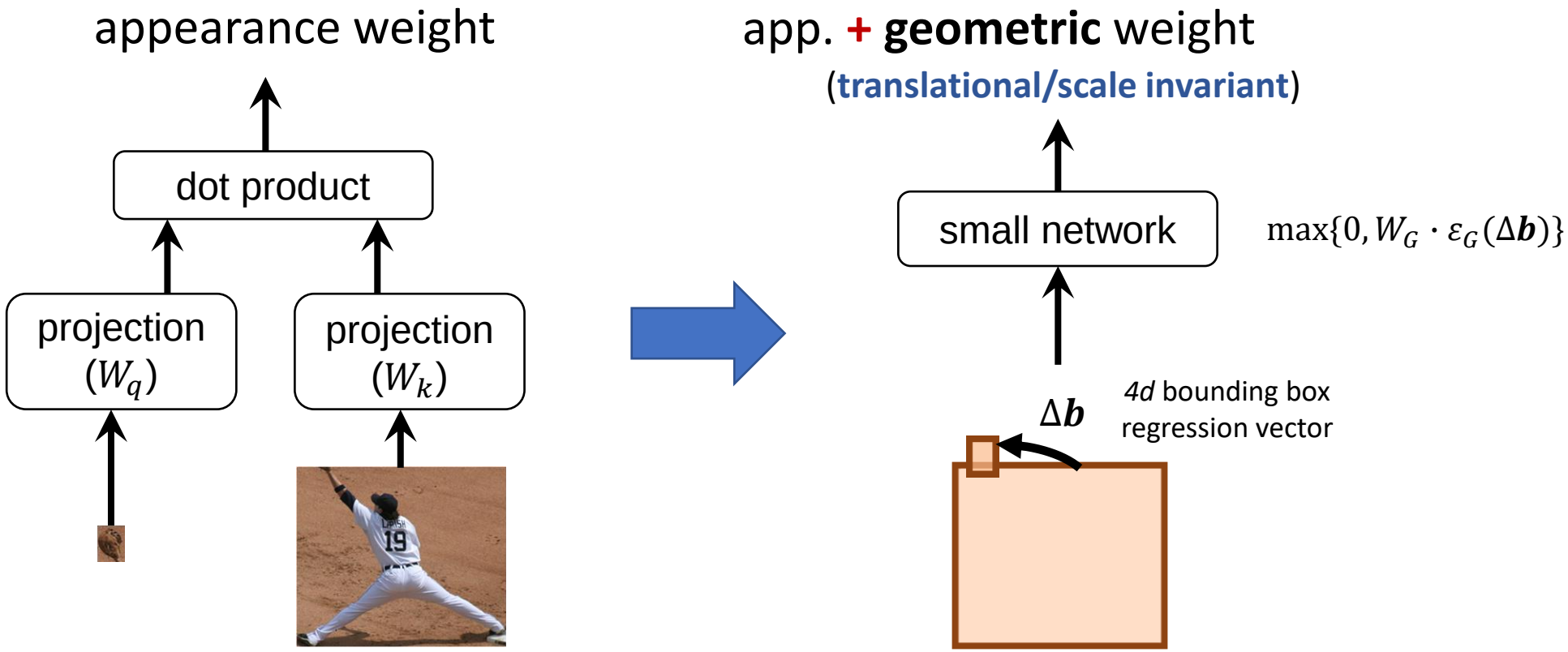


Key Different with standard self-attention models

✓Relative position

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

Relative Position for Relation Modeling



in standard ***attention*** module

in **object relation** module

Relative Position for NLP modeling (2020)

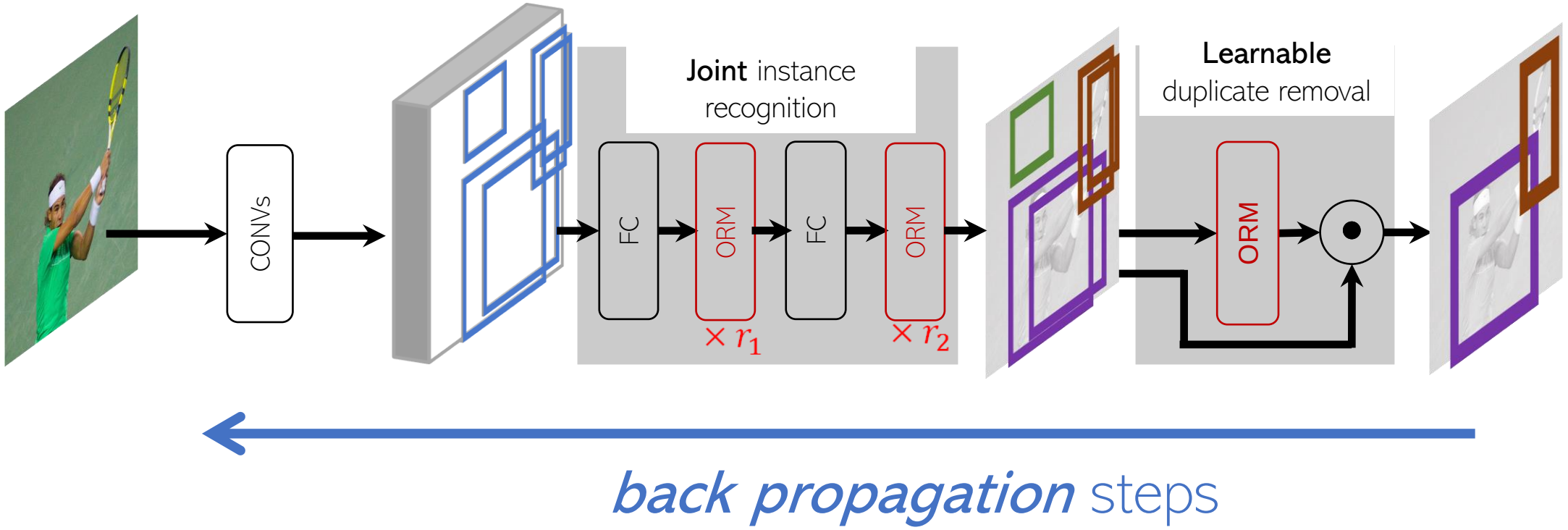
RETHINKING POSITIONAL ENCODING IN LANGUAGE PRE-TRAINING

Guolin Ke, Di He & Tie-Yan Liu
Microsoft Research
{guolin.ke, dihe, tyliu}@microsoft.com

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel* CRAFFEL@GMAIL.COM
Noam Shazeer* NOAM@GOOGLE.COM
Adam Roberts* ADAROB@GOOGLE.COM
Katherine Lee* KATHERINELEE@GOOGLE.COM
Sharan Narang SHARANNARANG@GOOGLE.COM
Michael Matena MMATENA@GOOGLE.COM
Yanqi Zhou YANQIZ@GOOGLE.COM
Wei Li MWEILI@GOOGLE.COM
Peter J. Liu PETERJLIU@GOOGLE.COM
Google, Mountain View, CA 94043, USA

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	

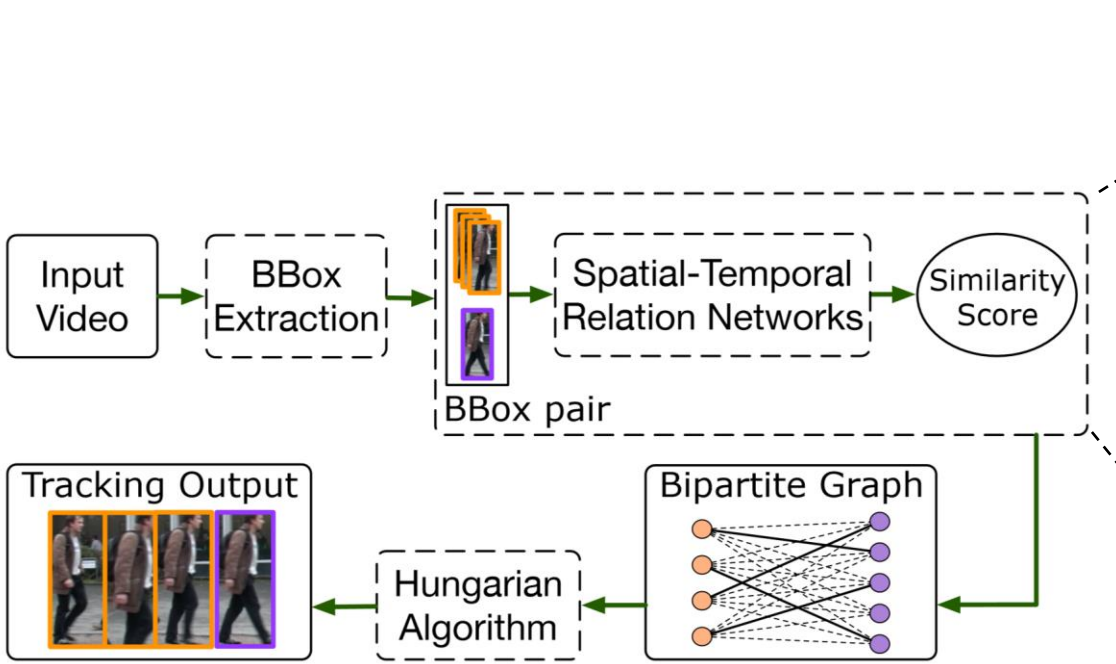
*Faster R-CNN with ResNet-101 model are used (evaluation on *minival*/ *test-dev* are reported)

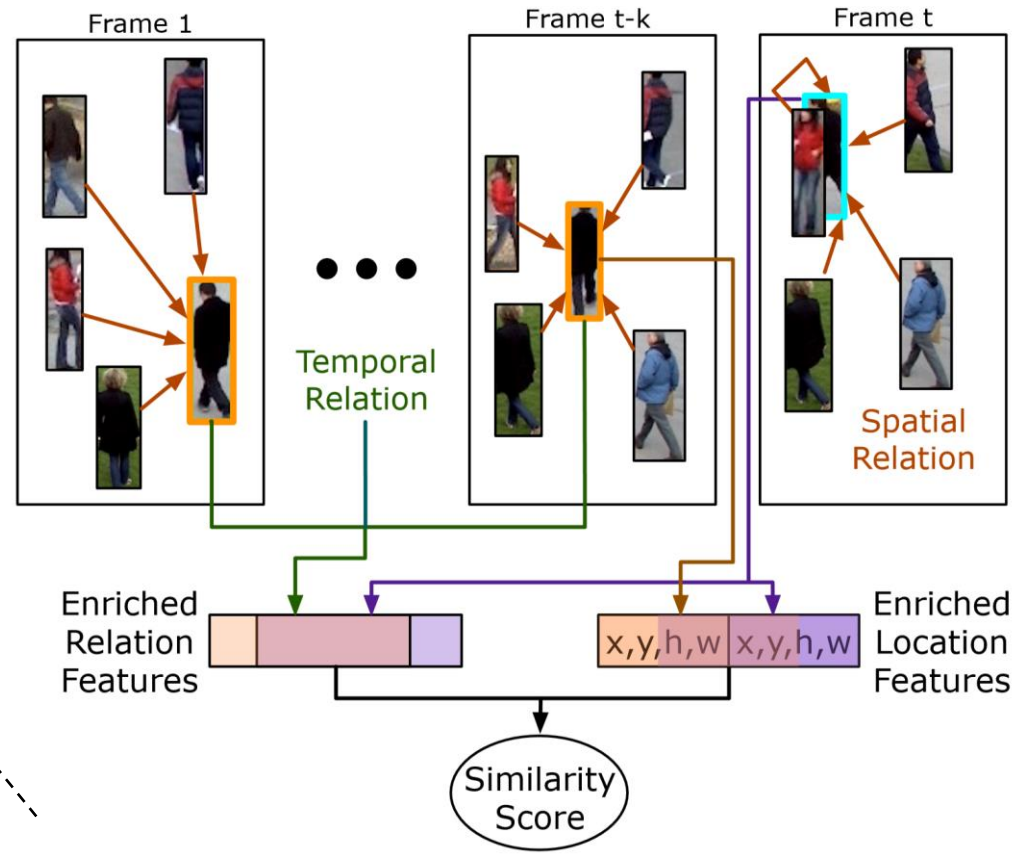
Relation Networks

ResNeXt-101-64x4d-FPN-DCN 45.0 → 45.9

+0.9 mAP

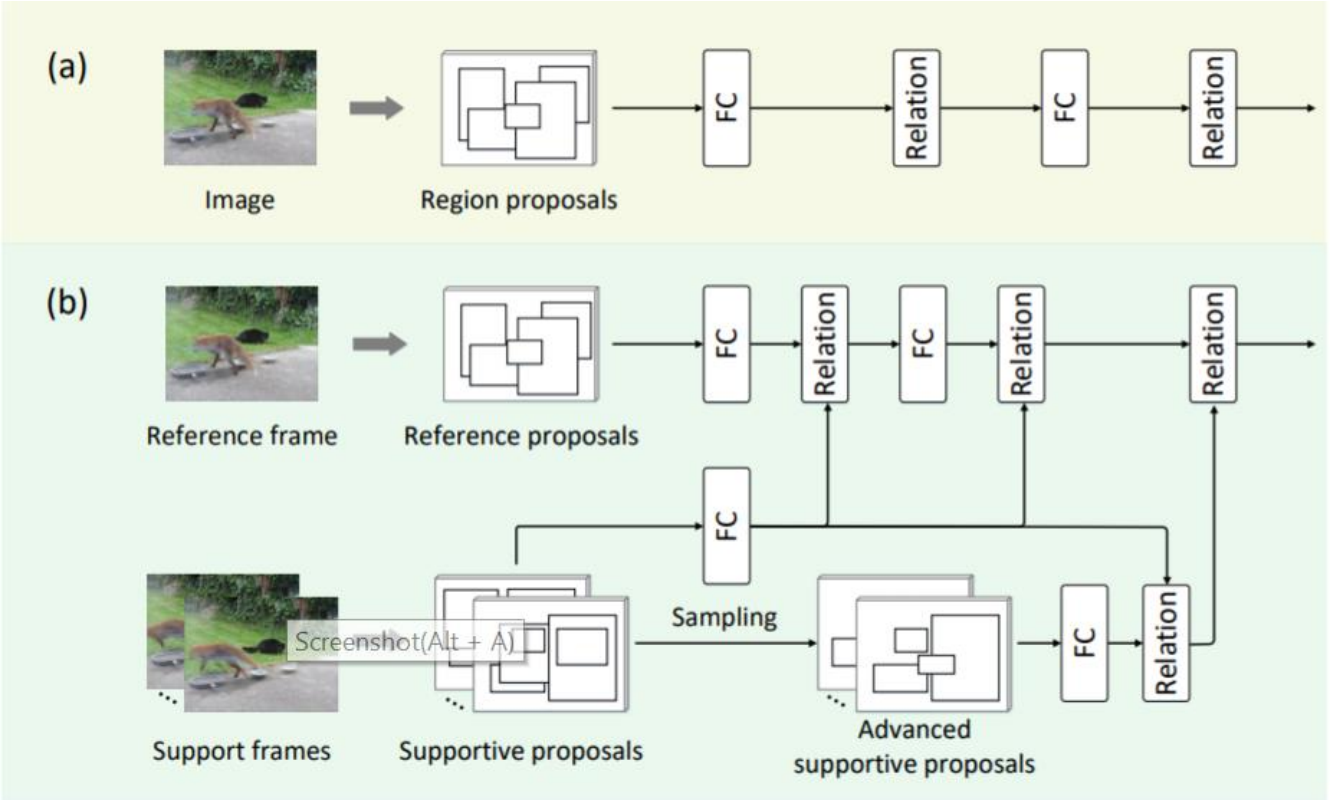
Multi-Object Tracking





Jiarui Xu, Yue Cao, Zheng Zhang and Han Hu. *Spatial-Temporal Relation Networks for Multi-Object Tracking.* ICCV, 2019

Video Object Detection

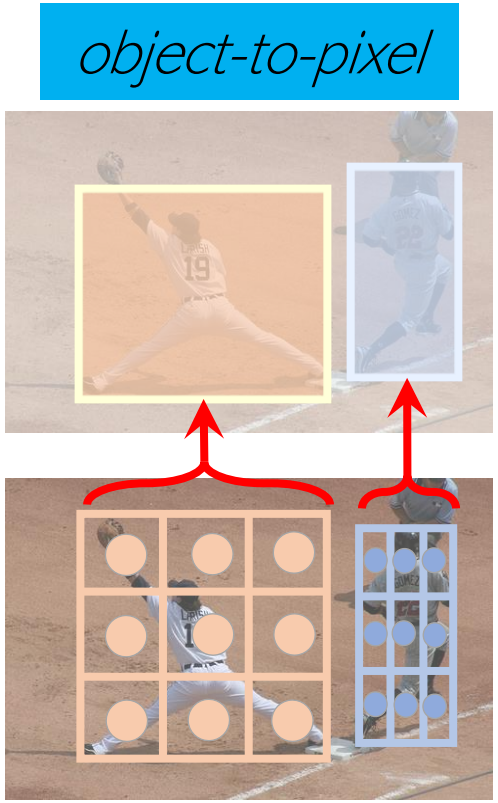


Jiajun Deng, et al. *Relation Distillation Networks for Video Object Detection*. ICCV, 2019

Haiping Wu, et al. *Sequence Level Semantics Aggregation for Video Object Detection*. ICCV, 2019

Yihong Chen, et al. *Memory Enhanced Global-Local Aggregation for Video Object Detection*. CVPR, 2020

Object-to-Pixel Relation Modeling

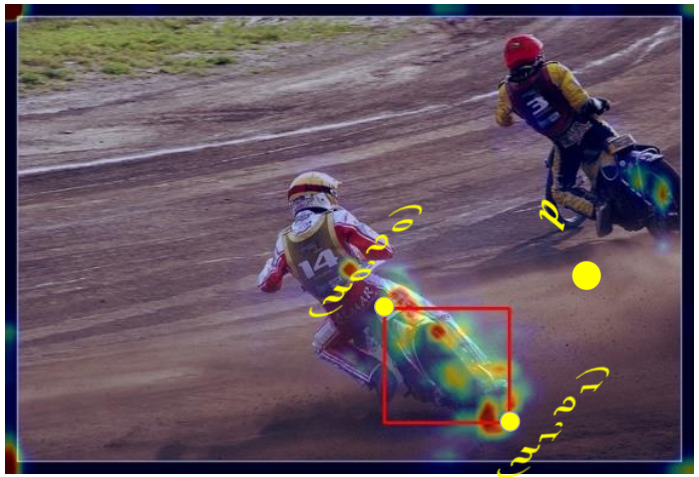


RoIAlign ⟶ **Self-Attention**

- Learn Region Features [ECCV'2018]
- Transformer Detector [ECCV'2020]
- RelationNet++ [NeurIPS'2020]

Learnable Object-to-Pixel Relation

Image Feature to Region Feature

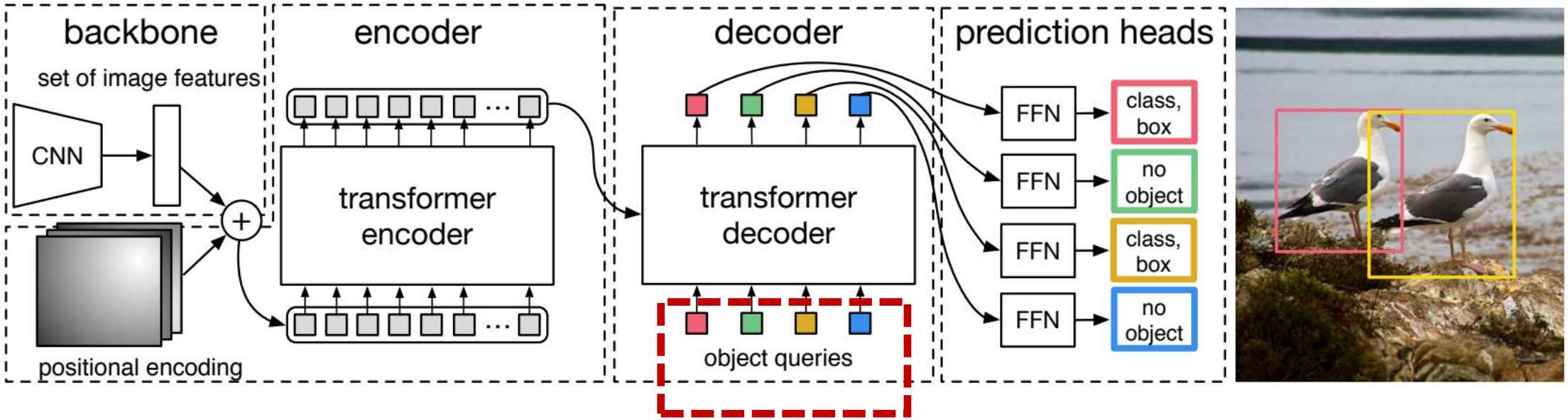


Geometric

Appearance

Jiayuan Gu et al. *Learning Region Features for Object Detection.* ECCV 2018

Transformer Detectors (DETR)



Implicitly learnt

Nicolas Carion et al. *End-to-End Object Detection with Transformers*. ECCV'2020

RelationNet++

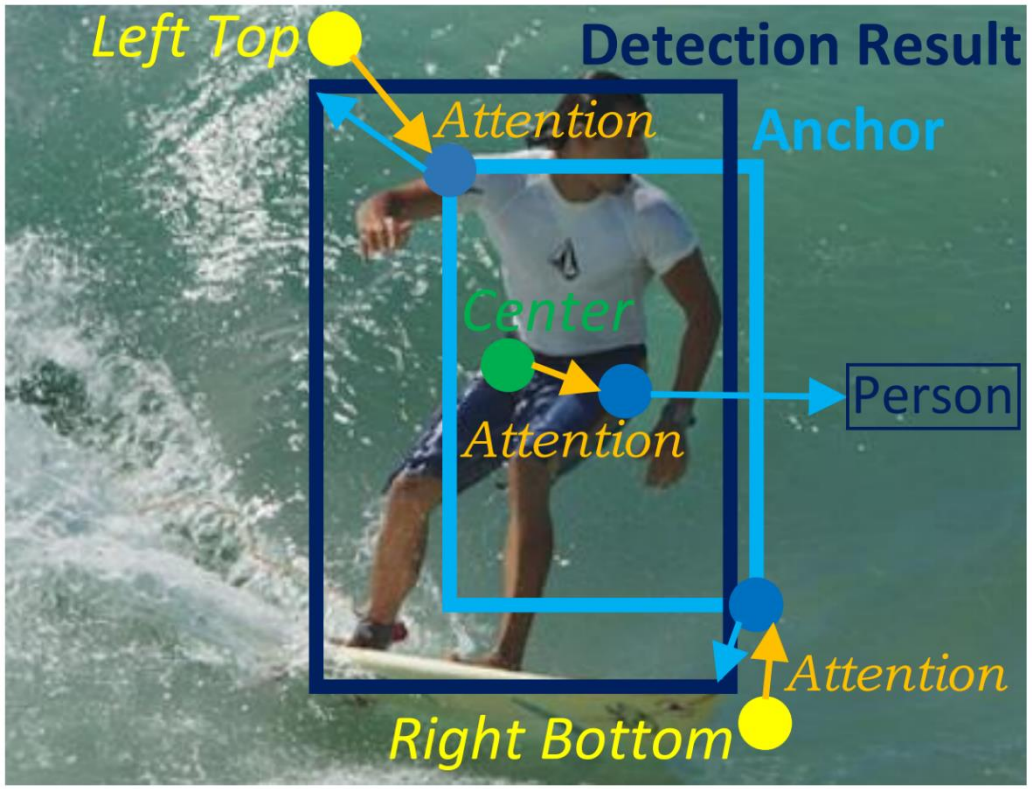


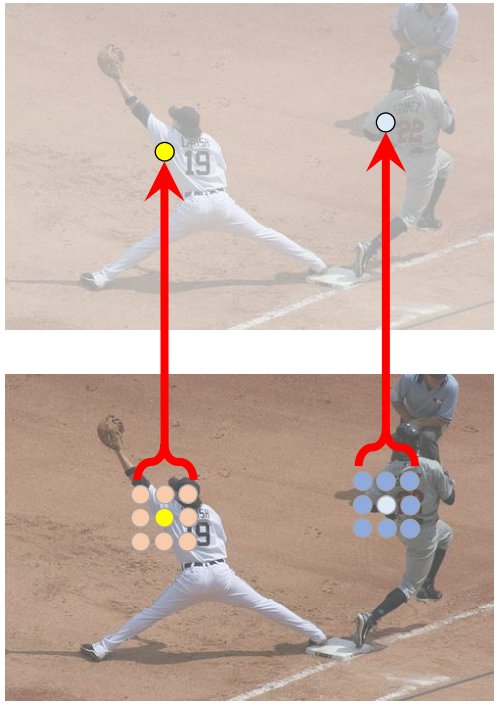
Table 12: Results on MS COCO `test-dev` set, '∗' denotes the m

method	backbone	AP	AP_{50}	AP_{75}
DCN v2* [40]	ResNet-101-DCN	46.0	67.9	50.8
SNIPER* [27]	ResNet-101	46.5	67.5	52.2
RepPoints* [35]	ResNet-101-DCN	46.5	67.4	50.9
MAL* [13]	ResNeXt-101	47.0	66.1	51.2
CentripetalNet* [6]	Hourglass-104	48.0	65.1	51.8
ATSS* [37]	ResNeXt-64x4d-101-DCN	50.7	68.9	56.3
TSD* [28]	SENet154-DCN	51.2	71.9	56.0
RelationNet++ (our)	ResNeXt-64x4d-101-DCN	50.3	69.0	55.0
RelationNet++ (our)*	ResNeXt-64x4d-101-DCN	**52.7**	**70.4**	**58.3**

Cheng Chi et al. *RelationNet++: Bridging Visual Representations for Object Detection via Transformer Decoder.* NeurIPS'2020

Pixel-to-Pixel Relation Modeling

pixel-to-pixel



Convolution Variants → **Self-Attention**

Usage

✓Complement convolution

✓Replace convolution

Complement Convolution

- "Convolution is too local"

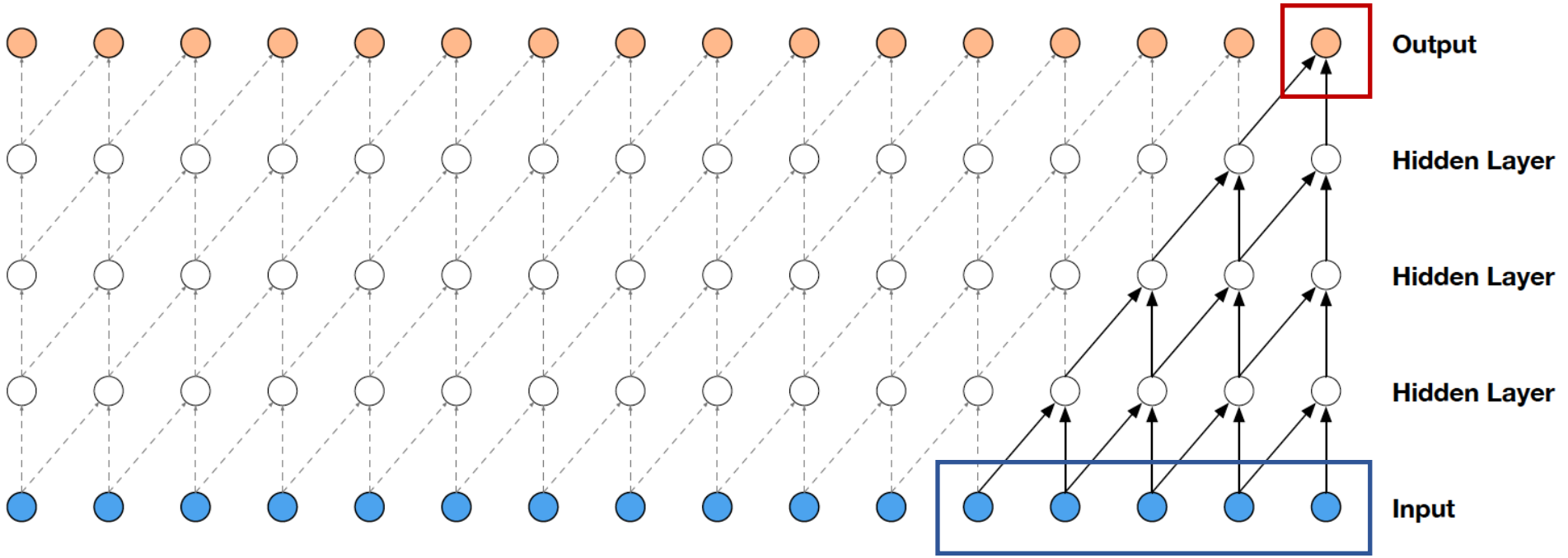
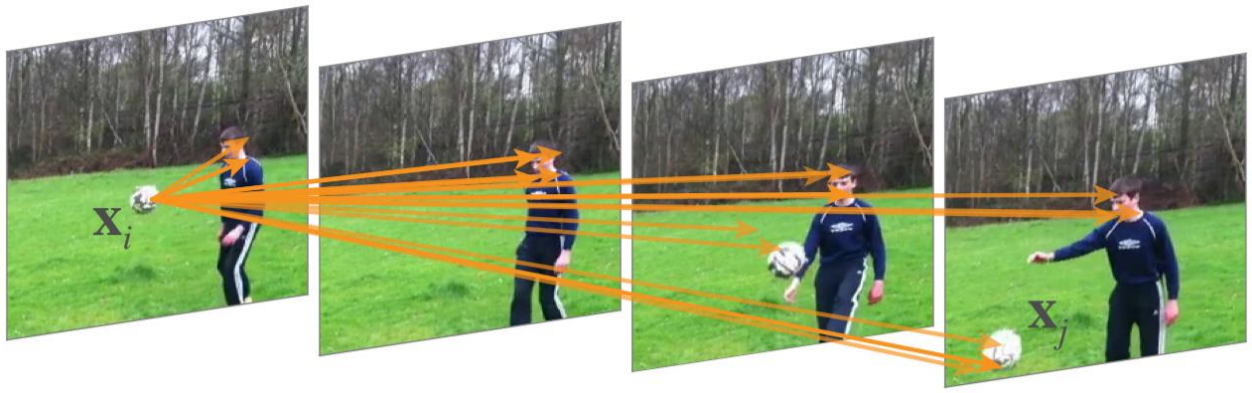


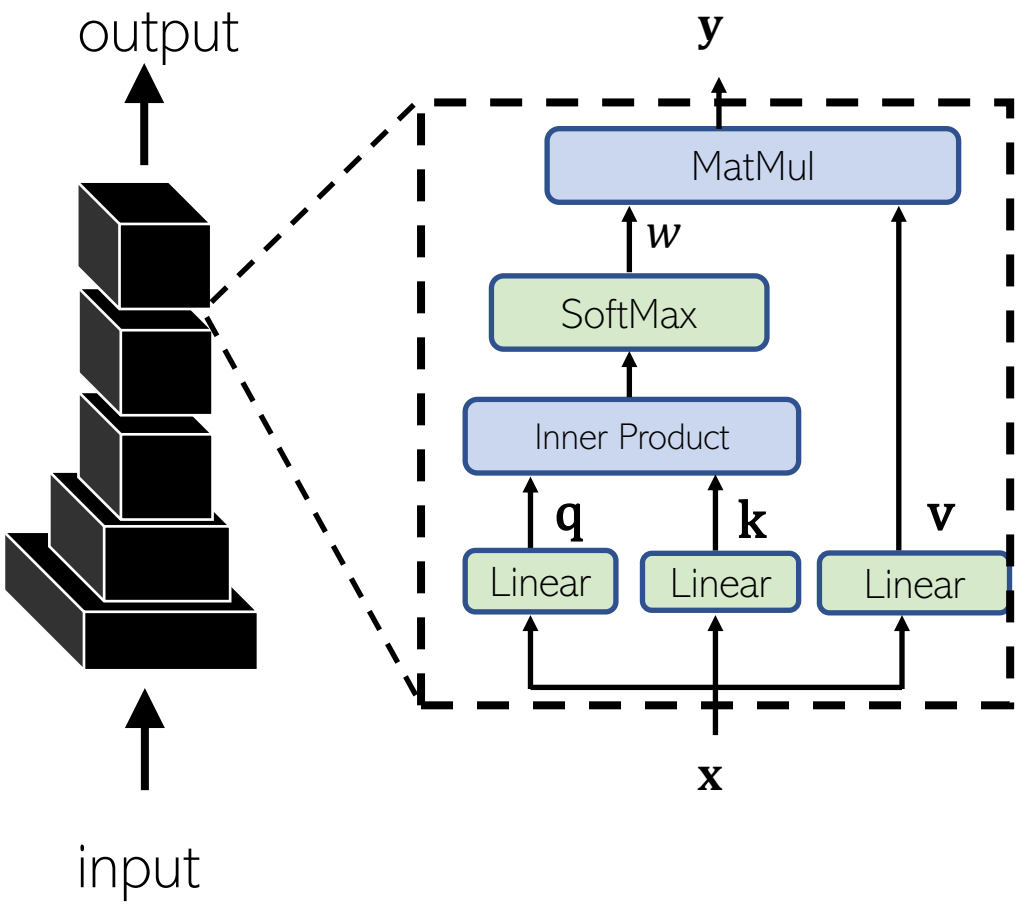
Figure credit: Van Den Oord et al.

Complement Convolution

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

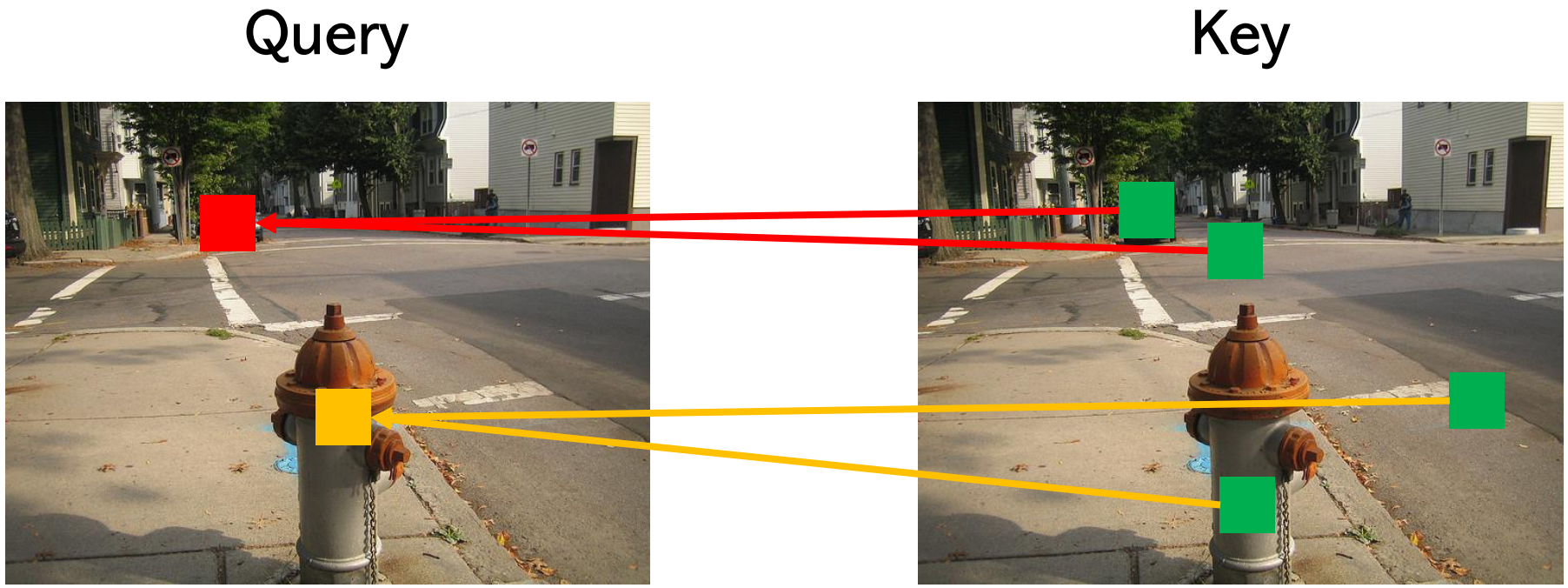


non-local block



The Degeneration Problem (2019)

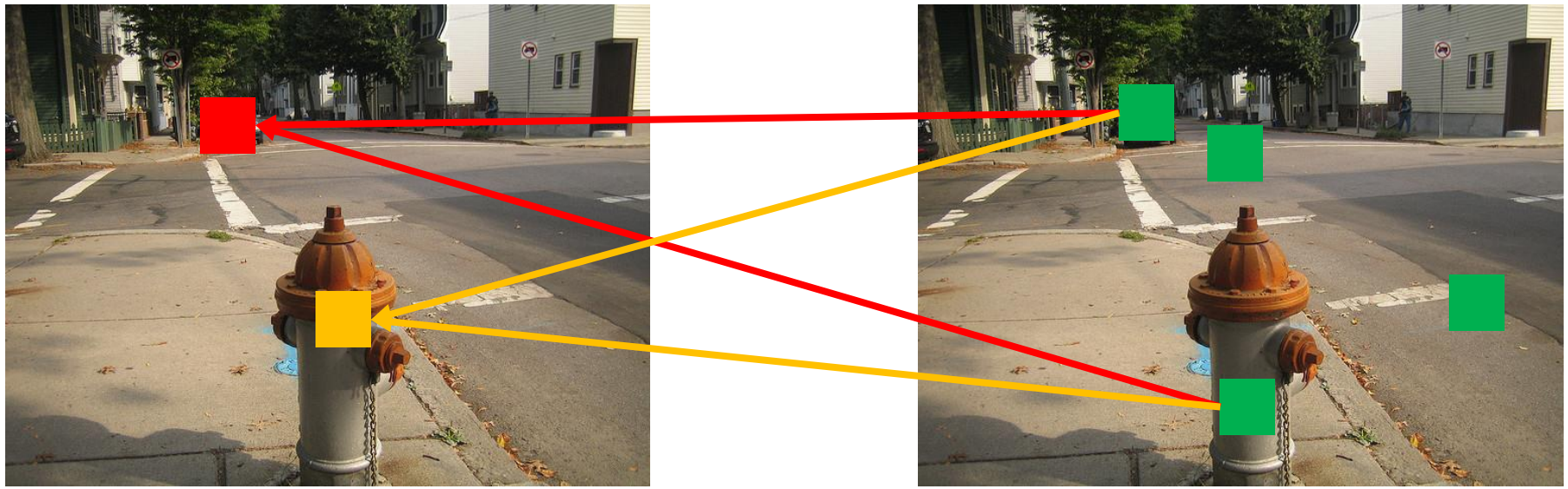
- Expectation of Ideally Learnt Relation
  - Different queries affected by **different** key



The Degeneration Problem (2019)

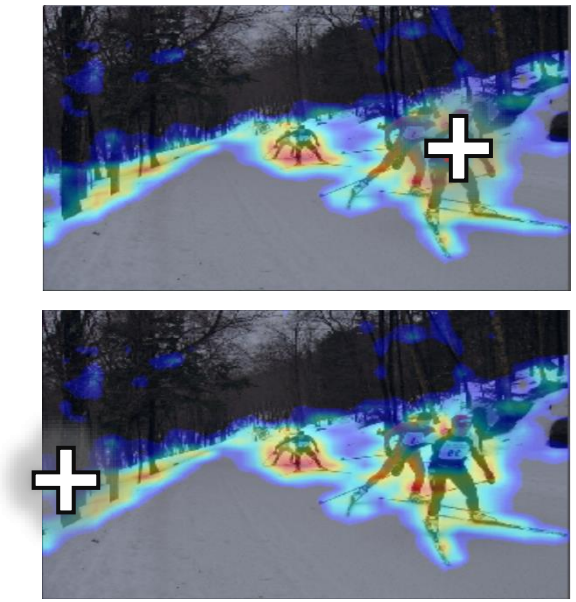
- What does the Self-Attention Learn?
  - Different queries affected by the **same** keys

Query **Key**

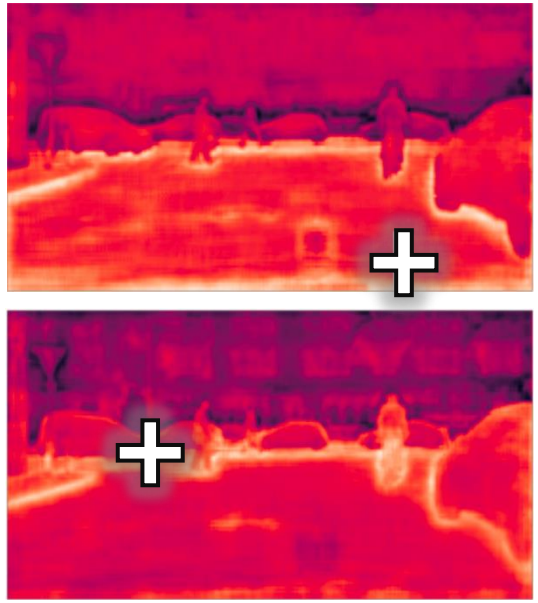


Visualizations on Real Tasks

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



Object Detection

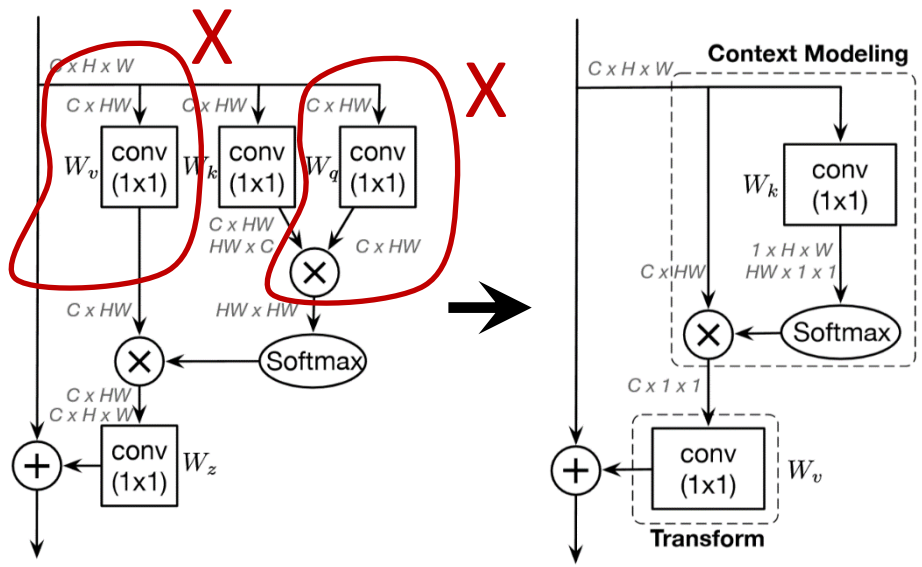


Semantic Segmentation

[GCNet, ICCVW'2019]

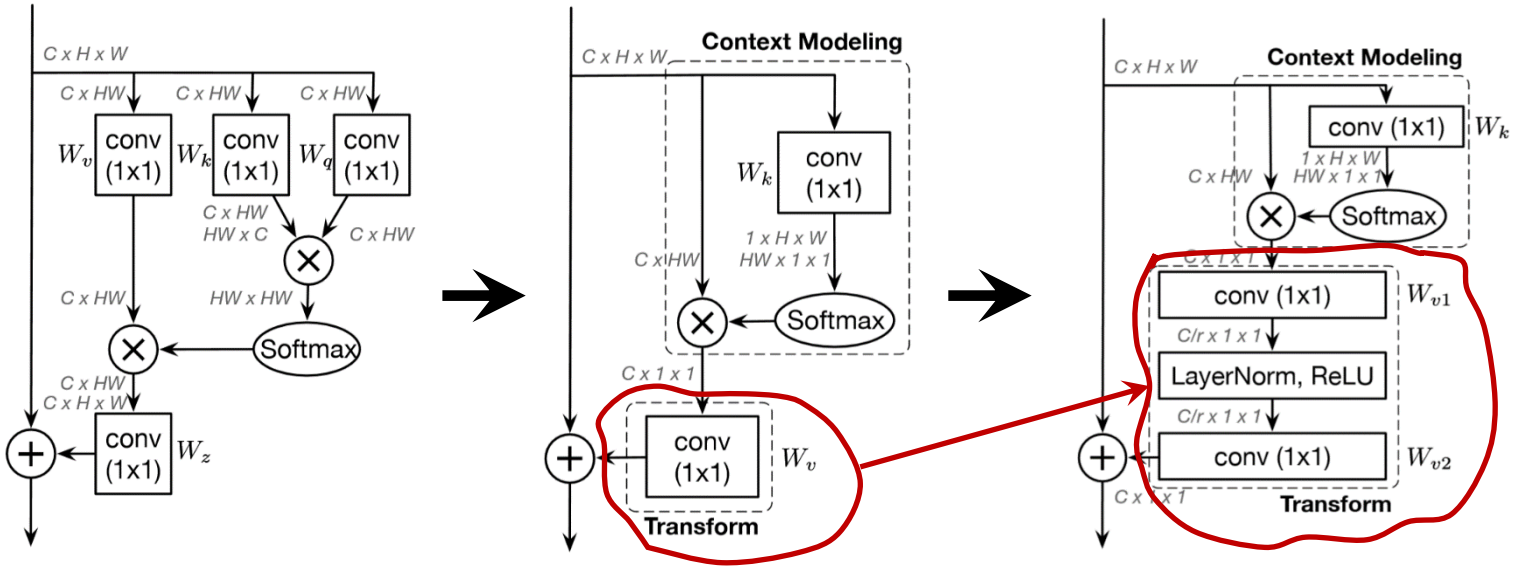
https://arxiv.org/pdf/1904.11492.pdf

GCNet: Explicitly Use the Same Attention Map



FLOPs	9.3G	→	**5.0M**
model size	2.1M		1.0M
accuracy (mAP)	38.0		38.1

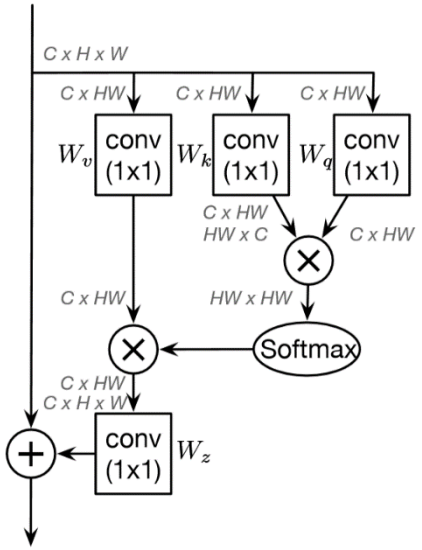
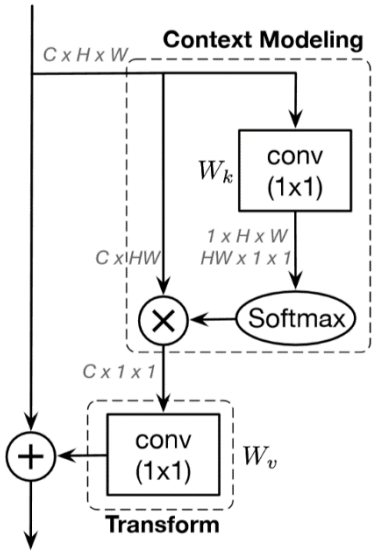
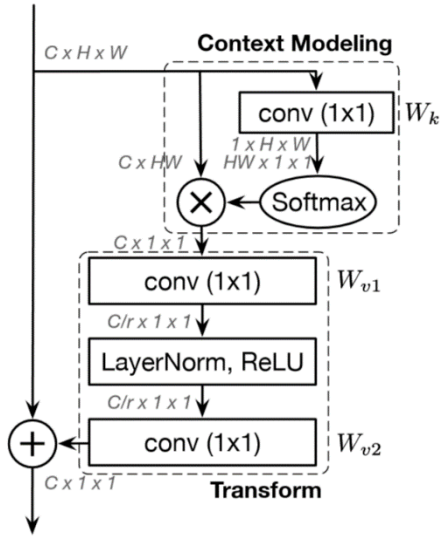
GCNet: Explicitly Use the Same Attention Map



FLOPs	9.3G	5.0M	4.0M
model size	2.1M	1.0M	**0.1M**
accuracy (mAP)	38.0	38.1	**38.1**

borrowed from SE-Net (champion of 2017 ImageNet Challenge)

GCNet: Explicitly Use the Same Attention Map

	Non-Local Block	Simplification	**Global Context Block**	**reduction**
				
FLOPs	9.3G	5.0M	**4.0M**	**2,300x**
model size	2.1M	1.0M	**0.1M**	**20x**
accuracy (mAP)	38.0	38.1	**38.1**	**unchanged**

COCO Object Detection Results

- Baseline: Mask R-CNN + ResNet50 + FPN

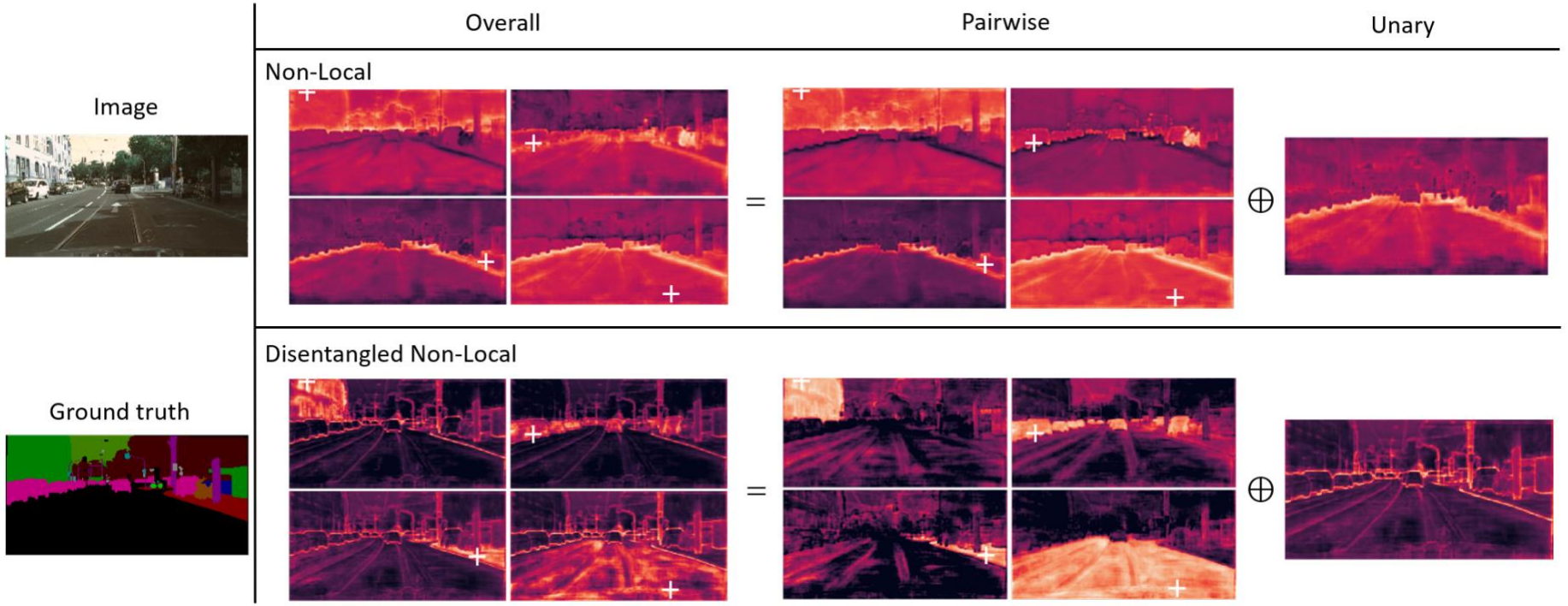
method	AP (bbox)	AP (mask)	#param	FLOPs
baseline	37.2	33.8	44.4M	279.4G
NL-Net	38.0	34.7	46.5M	288.7G
SE-Net	38.2	34.7	46.9M	279.5G
GC-Net (1 layer)	38.1	34.9	44.5M	279.4G
GC-Net (all layers)	**39.4**	**35.7**	46.9M	279.6G

+2.2 mAP **+1.9 mAP**

with little computation and model size overhead!

DNL: How to Effectively Model Pairwise?

- Disentangled design (ECCV'2020)



Minghao Yin et al. Disentangled Non-Local Neural Networks. ECCV'2020

Replace Convolution

- "Convolution is exponentially inefficient"

ResNet

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

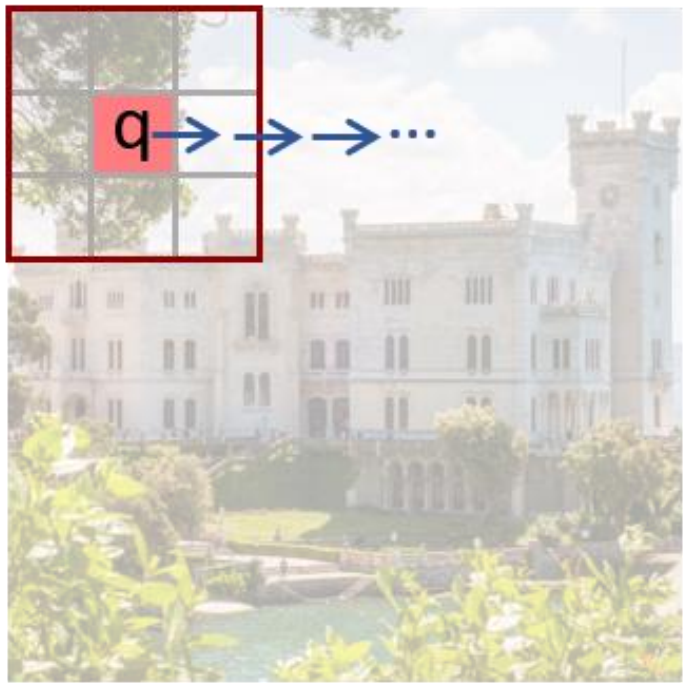
LR-Net-50 (7×7, m=8)
1×1, 64 **7×7 LR, 64, stride 2**
3×3 max pool, stride 2
[1×1, 100 **7×7 LR, 100** 1×1, 256] ×3
[1×1, 200 **7×7 LR, 200** 1×1, 512] ×4
[1×1, 400 **7×7 LR, 400** 1×1, 1024] ×6
[1×1, 800 **7×7 LR, 800** 1×1, 2048] ×3
global average pool 1000-d fc, softmax
$\mathbf{23.3}\times10^6$
$\mathbf{4.3}\times10^9$

LR-Net

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

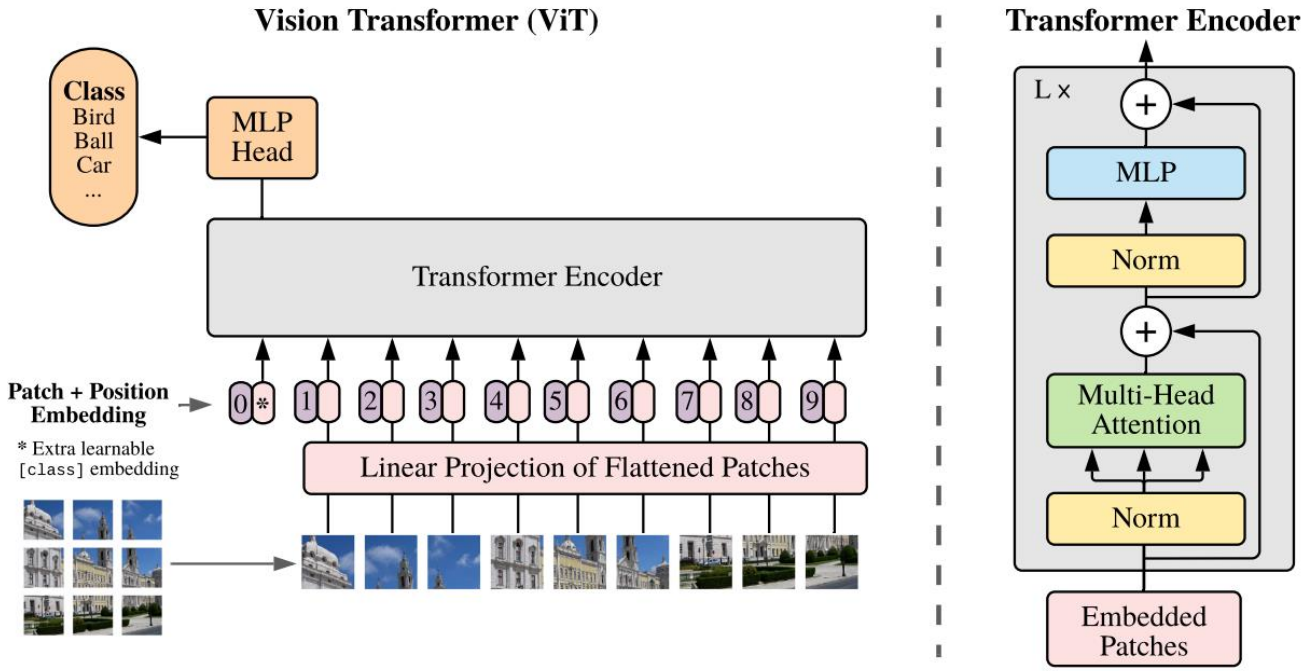
But … Slow in Real Computation

- Different queries use different key sets



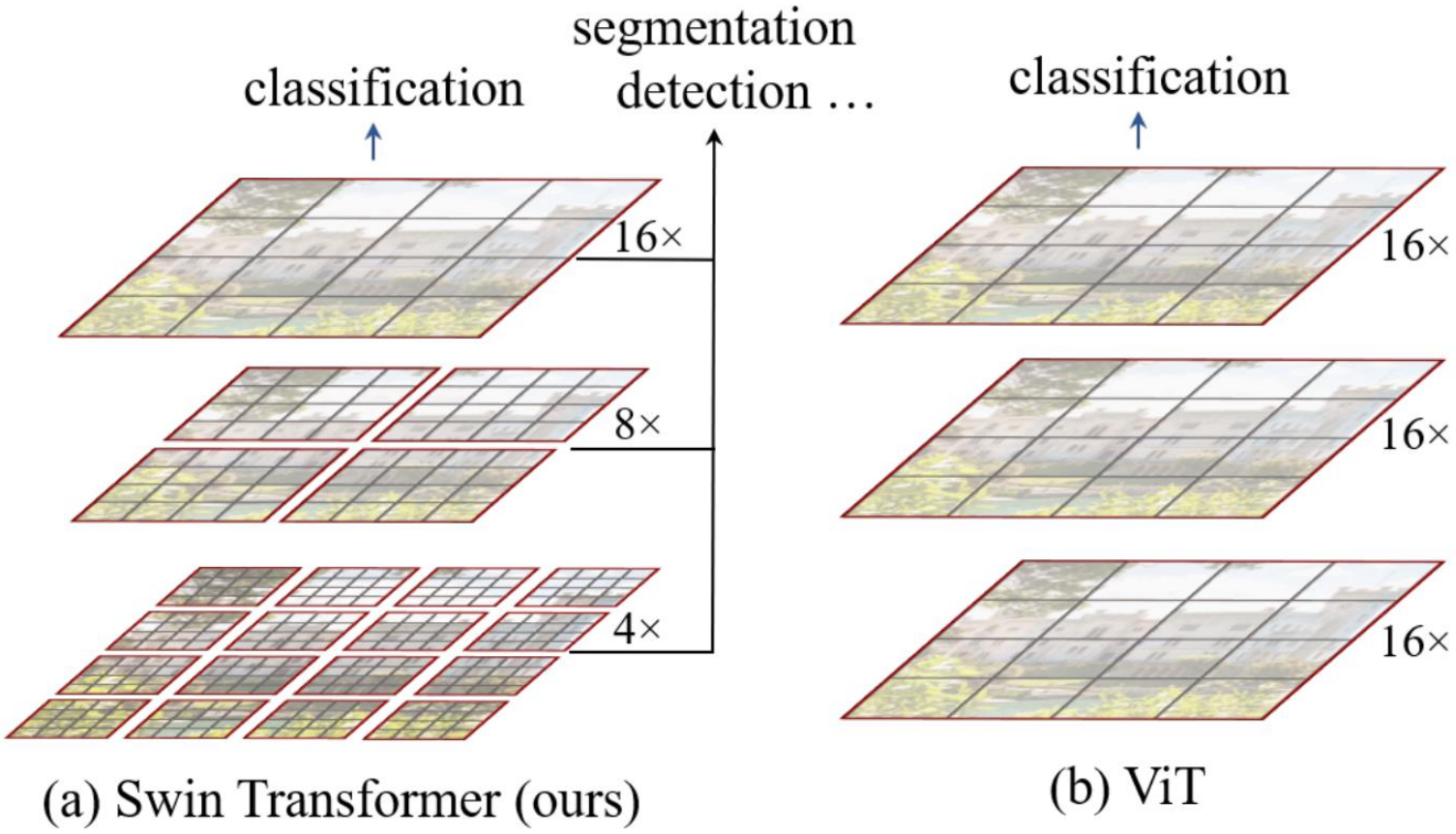
Vision Transformers for Image Recognition

- ICLR'2021 by Google Brain

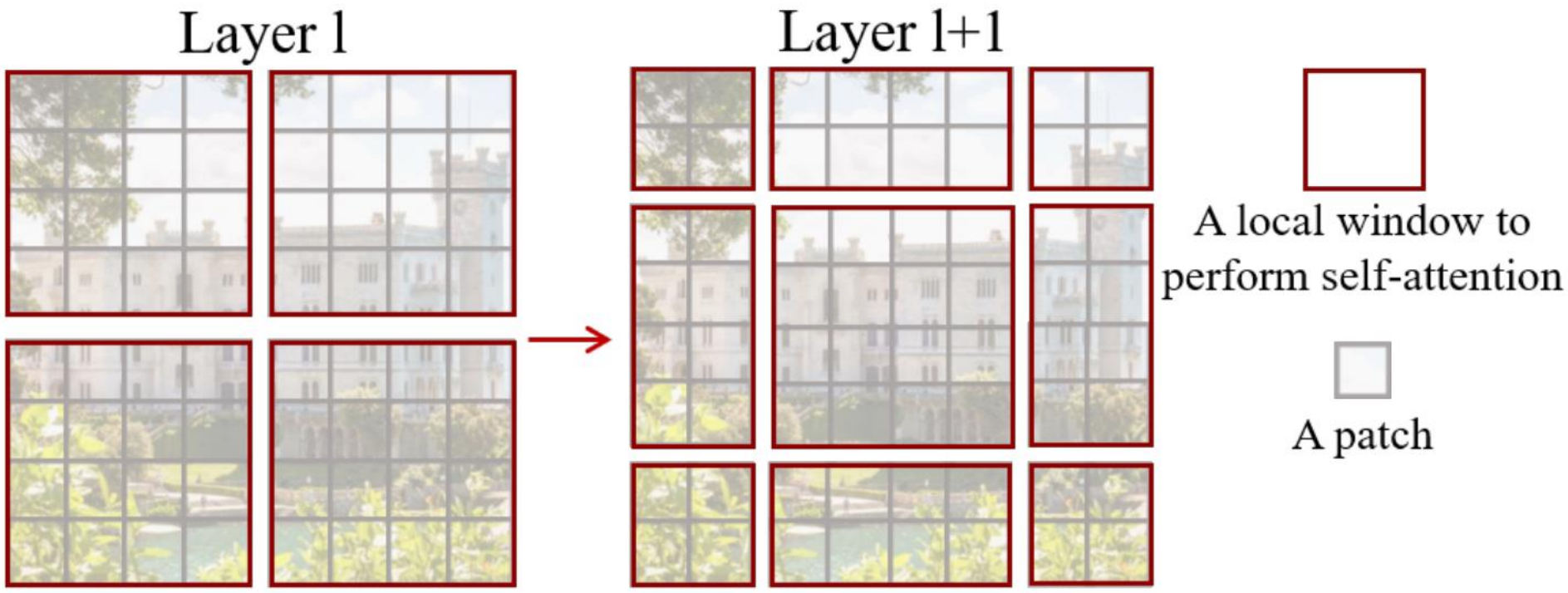


Alexey Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Tech Report 2020

Swin Transformer: a general-purpose backbone



Swin Transformer: a general-purpose backbone



Swin Transformer: a general-purpose backbone

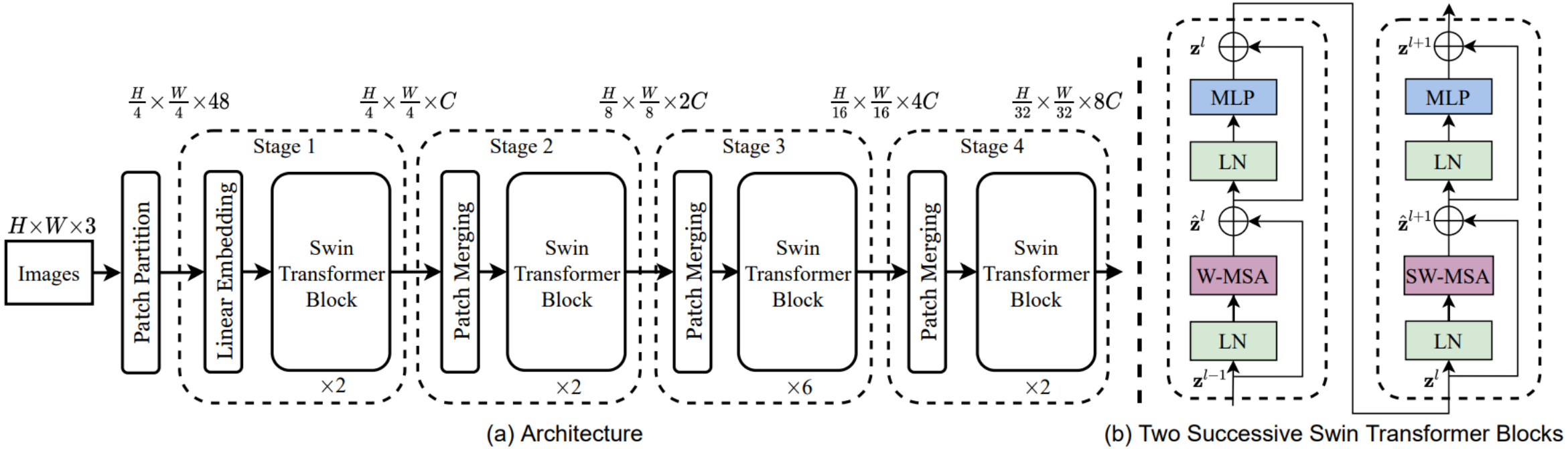


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Performance

(c) System-level Comparison

Method	mini-val AP^{box}	mini-val AP^{mask}	test-dev AP^{box}	test-dev AP^{mask}	#param.	FLOPs
RepPointsV2* [11]	-	-	52.1	-	-	-
GCNet* [6]	51.8	44.7	52.3	45.4	-	1041G
RelationNet++* [12]	-	-	52.7	-	-	-
SpineNet-190 [20]	52.6	-	52.8	-	164M	1885G
ResNeSt-200* [75]	52.5	-	53.3	47.1	-	-
EfficientDet-D7 [58]	54.4	-	55.1	-	77M	410G
DetectoRS* [45]	-	-	55.7	48.5	-	-
YOLOv4 P7* [3]	-	-	55.8	-	-	-
Copy-paste [25]	55.9	47.2	56.0	47.4	185M	1440G
X101-64 (HTC++)	52.3	46.0	-	-	155M	1033G
Swin-B (HTC++)	56.4	49.1	-	-	160M	1043G
Swin-L (HTC++)	57.1	49.5	57.7	50.2	284M	1470G
Swin-L (HTC++)*	**58.0**	**50.4**	**58.7**	**51.1**	284M	-

+2.7

Table 2. Results on COCO object detection and instance segmentation. †denotes that additional decovolution layers are used to produce hierarchical feature maps. * indicates multi-scale testing.

Method	Backbone	ADE20K val mIoU	ADE20K test score	#param.	FLOPs	FPS
DANet [22]	ResNet-101	45.2	-	69M	1119G	15.2
DLab.v3+ [10]	ResNet-101	44.1	-	63M	1021G	16.0
ACNet [23]	ResNet-101	45.9	38.5	-		
DNL [68]	ResNet-101	46.0	56.2	69M	1249G	14.8
OCRNet [70]	ResNet-101	45.3	56.0	56M	923G	19.3
UperNet [66]	ResNet-101	44.9	-	86M	1029G	20.1
OCRNet [70]	HRNet-w48	45.7	-	71M	664G	12.5
DLab.v3+ [10]	ResNeSt-101	46.9	55.1	66M	1051G	11.9
DLab.v3+ [10]	ResNeSt-200	48.4	-	88M	1381G	8.1
SETR [78]	T-Large‡	50.3	61.7	308M	-	-
UperNet	DeiT-S†	44.0	-	52M	1099G	16.2
UperNet	Swin-T	46.1	-	60M	945G	18.5
UperNet	Swin-S	49.3	-	81M	1038G	15.2
UperNet	Swin-B‡	51.6	-	121M	1841G	8.7
UperNet	Swin-L‡	**53.5**	**62.8**	234M	3230G	6.2

+3.2

Table 3. Results of semantic segmentation on the ADE20K val and test set. † indicates additional deconvolution layers are used to produce hierarchical feature maps. ‡ indicates that the model is pre-trained on ImageNet-22K.

Performance

(a) Regular ImageNet-1K trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [47]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [47]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [47]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [57]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [57]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [57]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [57]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [57]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [19]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [19]	384^2	307M	190.7G	27.3	76.5
DeiT-S [60]	224^2	22M	4.6G	940.4	79.8
DeiT-B [60]	224^2	86M	17.5G	292.3	81.8
DeiT-B [60]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.3
Swin-B	384^2	88M	47.0G	84.7	84.2

(b) ImageNet-22K pre-trained models

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [37]	384^2	388M	204.6G	-	84.4
R-152x4 [37]	480^2	937M	840.5G	-	85.4
ViT-B/16 [19]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [19]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.0
Swin-L	384^2	197M	103.9G	42.1	86.4

Table 1. Comparison of different backbones on ImageNet-1K classification. Throughput is measured using the GitHub repository of [65] and a V100 GPU, following [60].

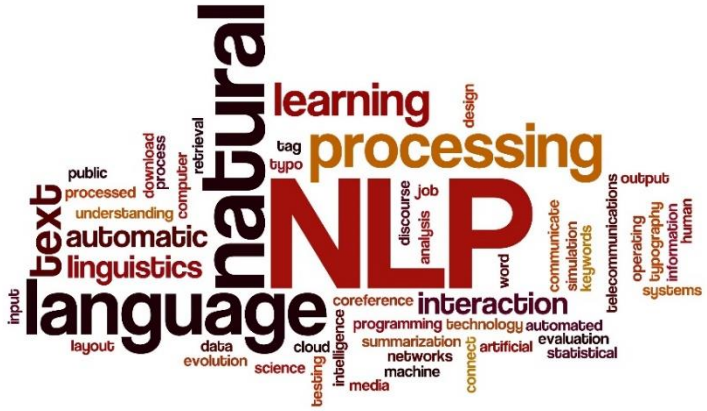
(a) Various frameworks

Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS
Cascade Mask R-CNN	R-50	46.3	64.3	50.5	82M	739G	18.0
	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3
ATSS	R-50	43.5	61.9	47.0	32M	205G	28.3
	Swin-T	**47.2**	**66.5**	**51.3**	36M	215G	22.3
RepPointsV2	R-50	46.5	64.6	50.3	42M	274G	13.6
	Swin-T	**50.0**	**68.5**	**54.2**	45M	283G	12.0
Sparse R-CNN	R-50	44.5	63.4	48.2	106M	166G	21.0
	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4

(b) Various backbones w. Cascade Mask R-CNN

	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	param	FLOPs	FPS
DeiT-S†	48.0	67.2	51.7	41.4	64.2	44.3	80M	889G	10.4
R50	46.3	64.3	50.5	40.1	61.7	43.4	82M	739G	18.0
Swin-T	**50.5**	**69.3**	**54.9**	**43.7**	**66.6**	**47.1**	86M	745G	15.3
X101-32	48.1	66.5	52.4	41.6	63.9	45.2	101M	819G	12.8
Swin-S	**51.8**	**70.4**	**56.3**	**44.7**	**67.9**	**48.5**	107M	838G	12.0
X101-64	48.3	66.4	52.3	41.7	64.0	45.1	140M	972G	10.4
Swin-B	**51.9**	**70.9**	**56.5**	**45.0**	**68.4**	**48.7**	145M	982G	11.6

Take-Home Message



Universal Models for NLP/CV

Thanks All!
Q & A