Transformer/Self-attention Modeling in Computer Vision

Towards Universal Models for CV/NLP

Yue Cao

Microsoft Research Asia

Oct. 9th, 2021 @ VALSE

Self Introduction

- 2010.8 - 2019.7 B.S. & Ph.D. in Tsinghua Univ.
- 2019.7 – now Senior Researcher in MSRA

Yue Cao

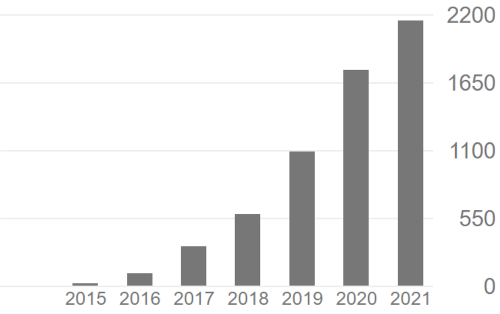
Senior Researcher, Microsoft Research Asia
在 microsoft.com 的电子邮件经过验证 - 首页
Deep Learning Computer Vision

已关注

标题	引用次数	年份
Learning Transferable Features with Deep Adaptation Networks M Long, Y Cao, J Wang, MI Jordan International Conference on Machine Learning (ICML), 2015	2657	2015
GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond Y Cao*, J Xu*, S Lin, F Wei, H Hu arXiv preprint arXiv:1904.11492	461	2019
Deep Hashing Network for Efficient Similarity Retrieval H Zhu, M Long, J Wang, Y Cao AAAI Conference on Artificial Intelligence (AAAI), 2016	460	2016
Vl-bert: Pre-training of generic visual-linguistic representations W Su*, X Zhu*, Y Cao, B Li, L Lu, F Wei, J Dai International Conference on Learning Representations (ICLR)	406	2020
Deep Visual-Semantic Quantization for Efficient Image Retrieval Y Cao, M Long, J Wang, S Liu IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017	238	2017
Swin transformer: Hierarchical vision transformer using shifted windows Z Liu, Y Lin, Y Cao, H Hu, Y Wei, Z Zhang, S Lin, B Guo International Conference on Computer Vision (ICCV), 2021 (Oral)	216	2021

引用次数

	总计	2016 年至今
引用	6093	6022
h 指数	25	25
i10 指数	29	29



可公开访问的出版物数量 查看全部

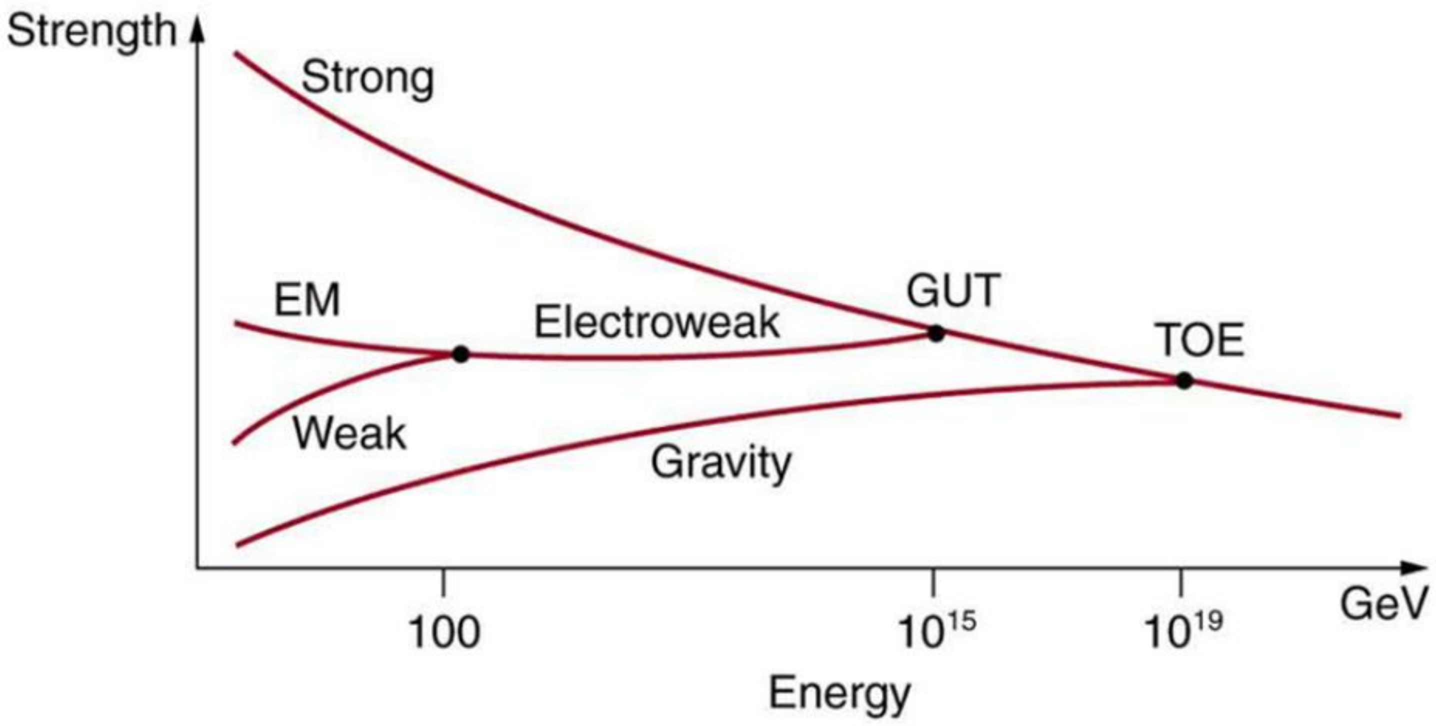
2 篇文章 16 篇文章
无法查看的文章 可查看的文章

根据资金授权书

合著作者 修改

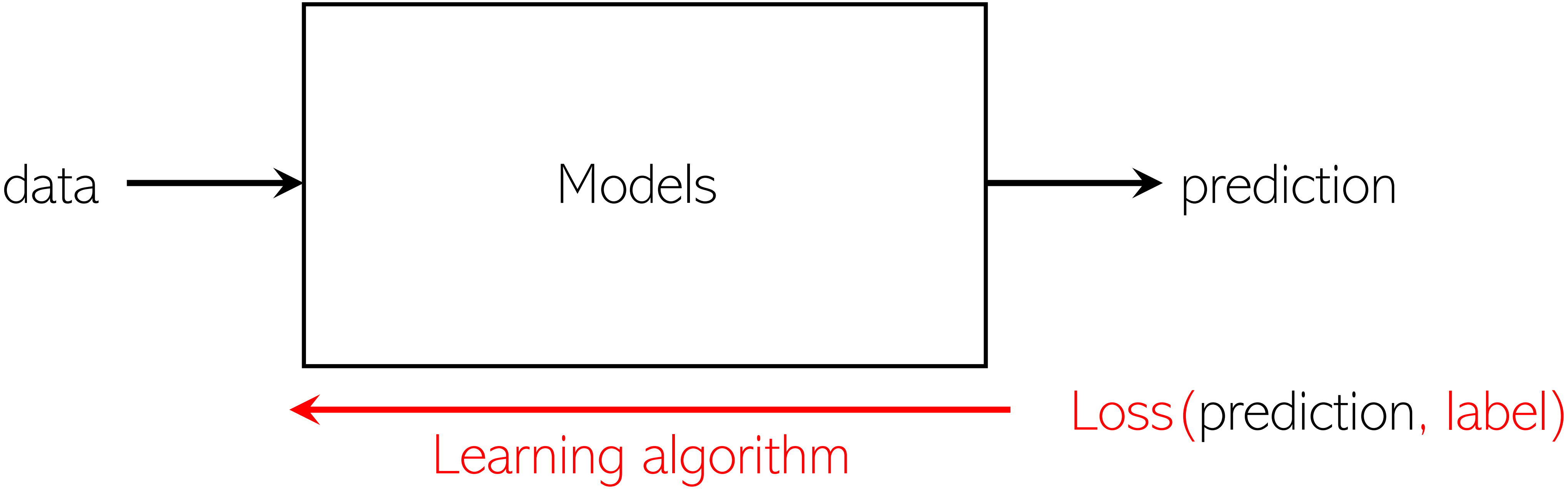
Grand Unification Theory in Physics

- The Holy Grail in Physics



A Unification Story for AI

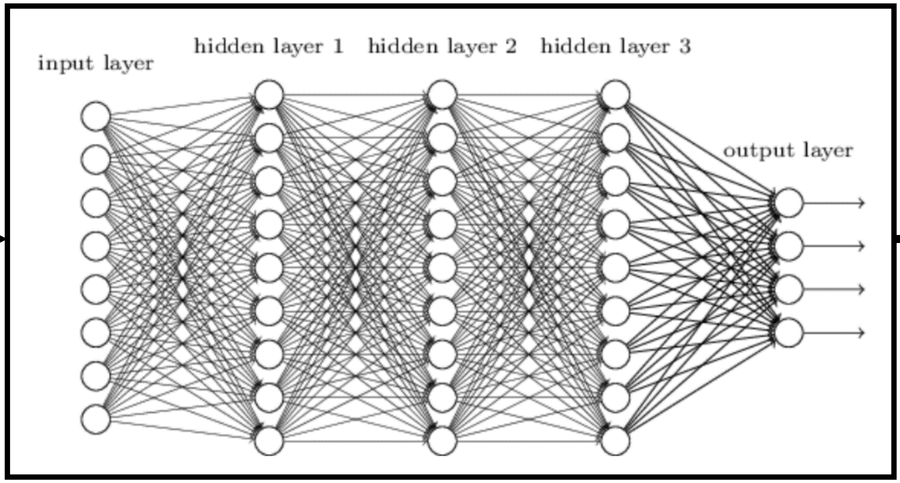
- The machine learning era
  - a **paradigm** unification: learning from **historical data** and make future predictions



A Unification Story for AI

- The machine learning era
  - a **paradigm** unification: learning from **historical data** and make future predictions
- The deep learning era
  - The unification of **architecture:** CNN, RNN, LSTM, Transformer

data



prediction

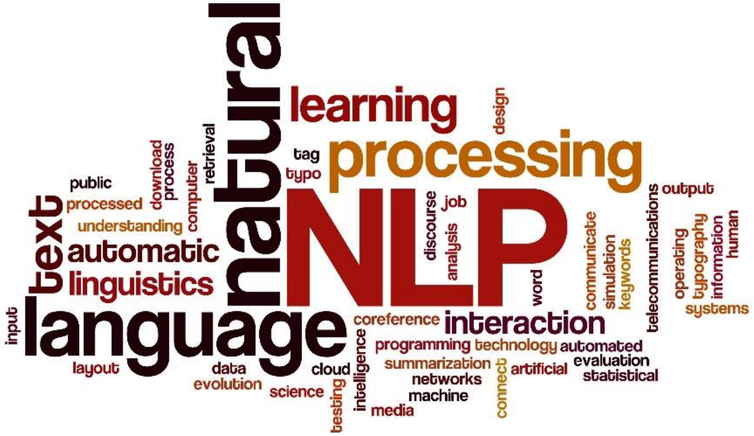
back-propagation

Loss (prediction, label)

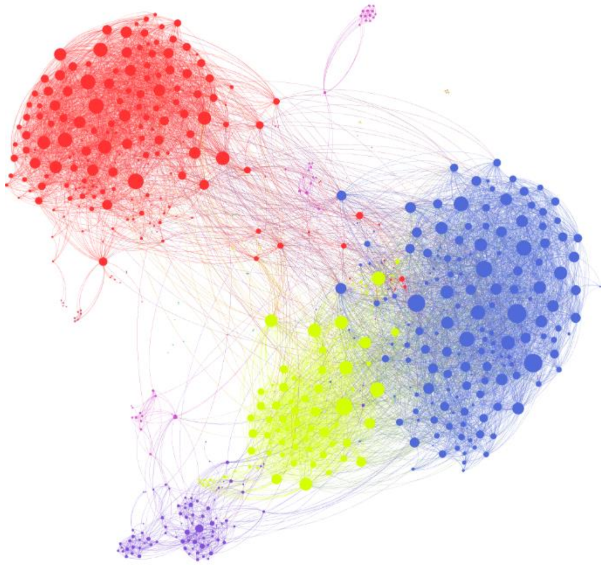
A Unification Story for AI

- What about models?

convolution

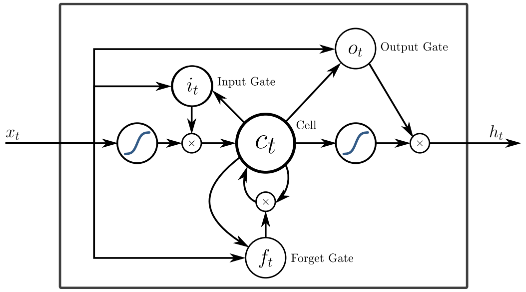


self-attention
(Transformers)



graph networks

Model Evolution in NLP



1995

LSTM

Jürgen Schmidhuber

2014

Deep RNN

Baidu

2014

GRU

Google

2015

RNN+attention

CIFAR

2017

Transformers (self-attention)

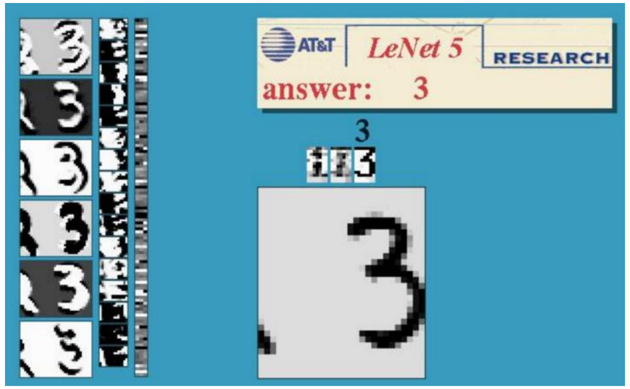
Google Brain

Model Evolution in CV

1989

Convolution

Yann LeCun



LeNet, AlexNet, GoogleNet, VGG, ResNet …

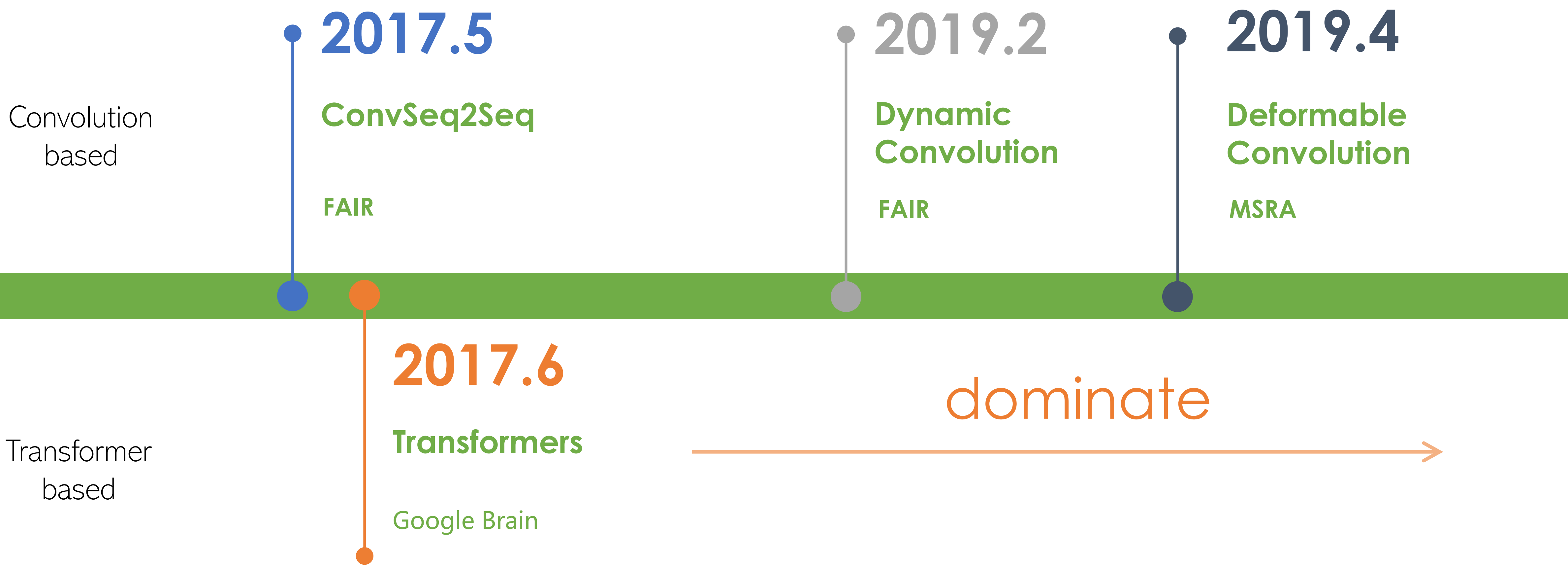
Can NLP/CV share the same basic modules?

- Why?
  - Facilitate joint modeling of visual and textual signals
  - Modeling knowledge from both domains can be more deeply shared
  - Easy for industry to perform specific optimization

  - Pursuing universality, which is beautiful itself

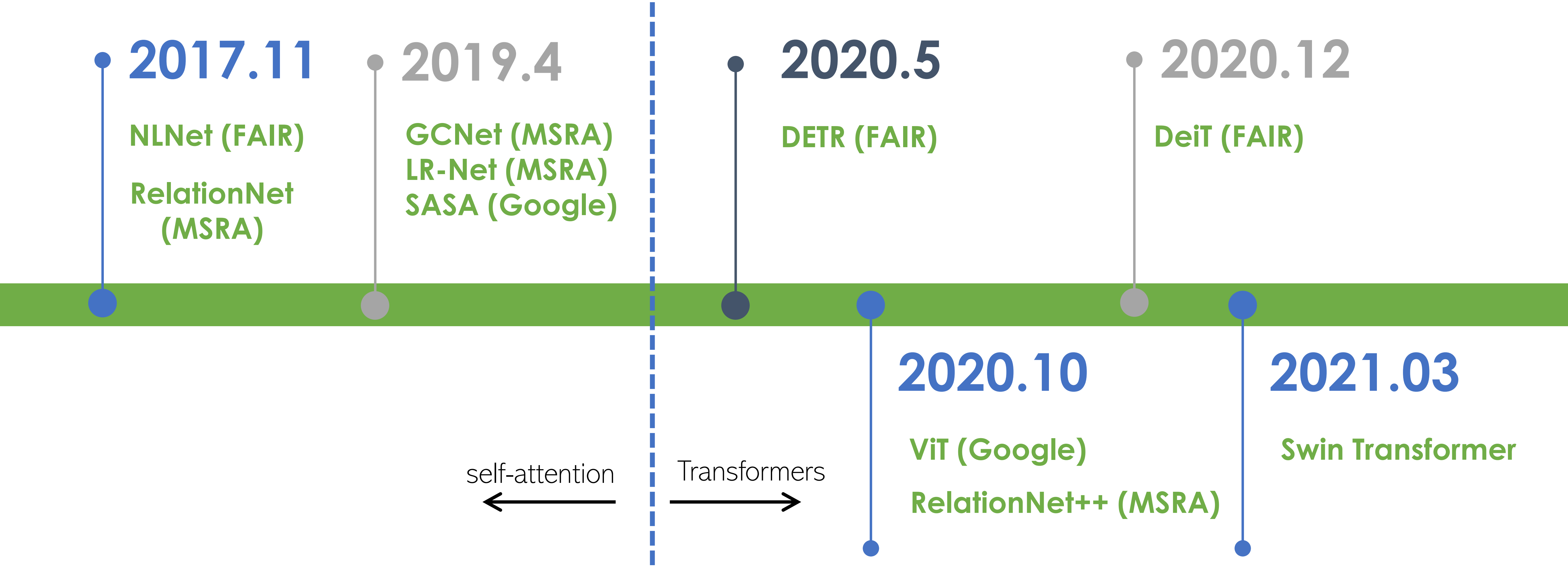
Can NLP/CV share the same basic modules?

- Adapting convolution layers for NLP modeling



Can NLP/CV share the same basic modules?

- Adapting self-attention/Transformer layers for CV modeling



Transformers

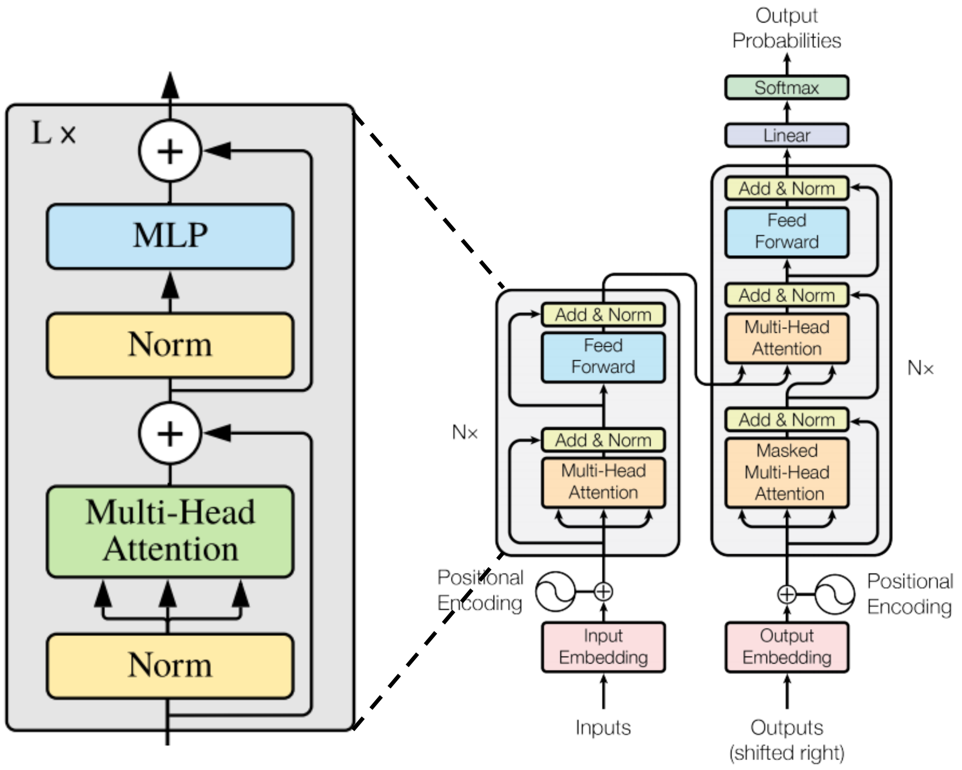
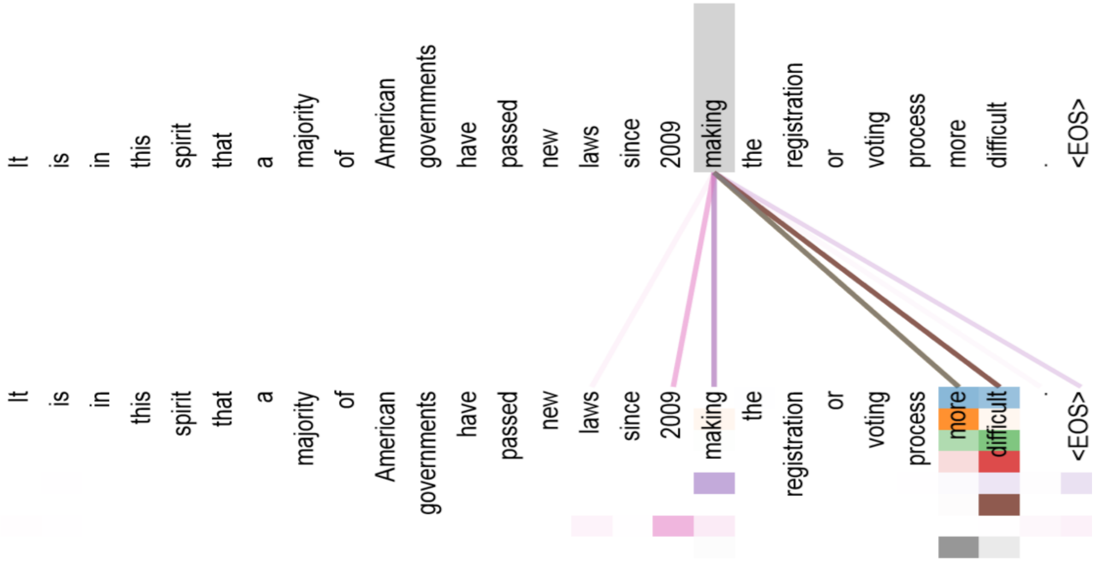


Figure 1: The Transformer - model architecture.

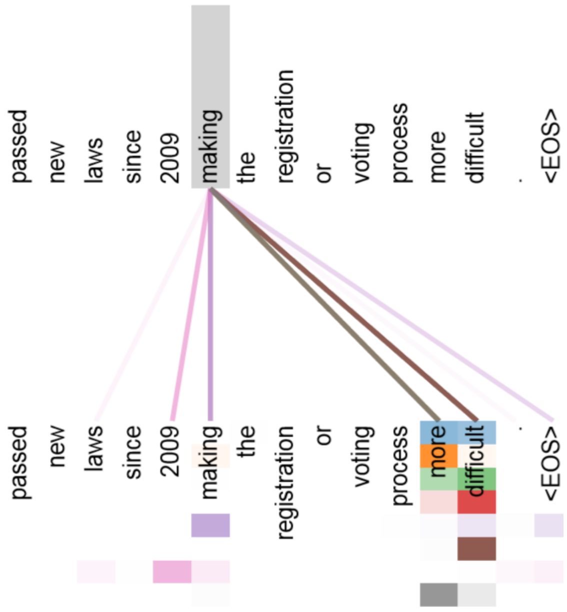


Ashish Vaswani et al, **Attention is all you need**, NeurIPS'2017

Self-Attention Unit

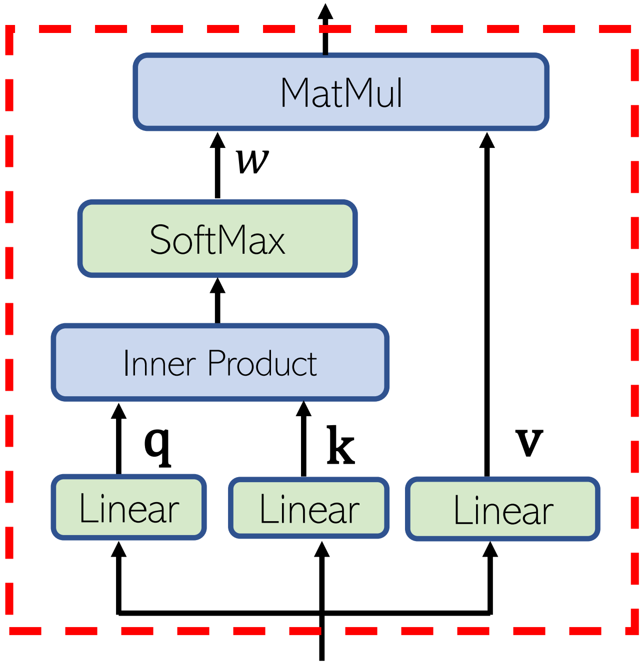
- Transforms the word/token input feature by encoding its relationship with other words/tokens
- A weighted average of **Value**, where the weight is the normalized inner product of **Query** and **Key**

Output word features



Input word features

Output features $\mathbf{y}$



Input features $\mathbf{x}$

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

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

Self-Attention Unit: An Example

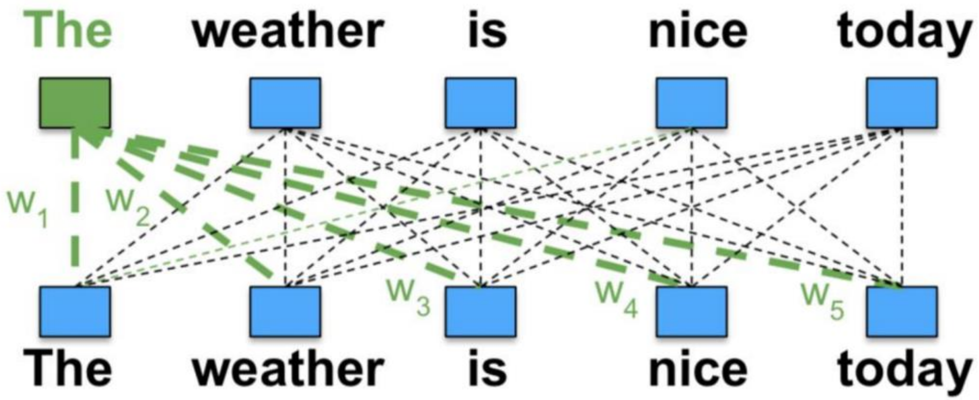


Figure Credit: http://fuyw.top/NLP_02_QANet/

Transformer Block

- Major Component
  - Multi-head self-attention block

- Others are also necessary
  - Positional encoding
  - Layer Normalization
  - Skip connection
  - Feed Forward Networks

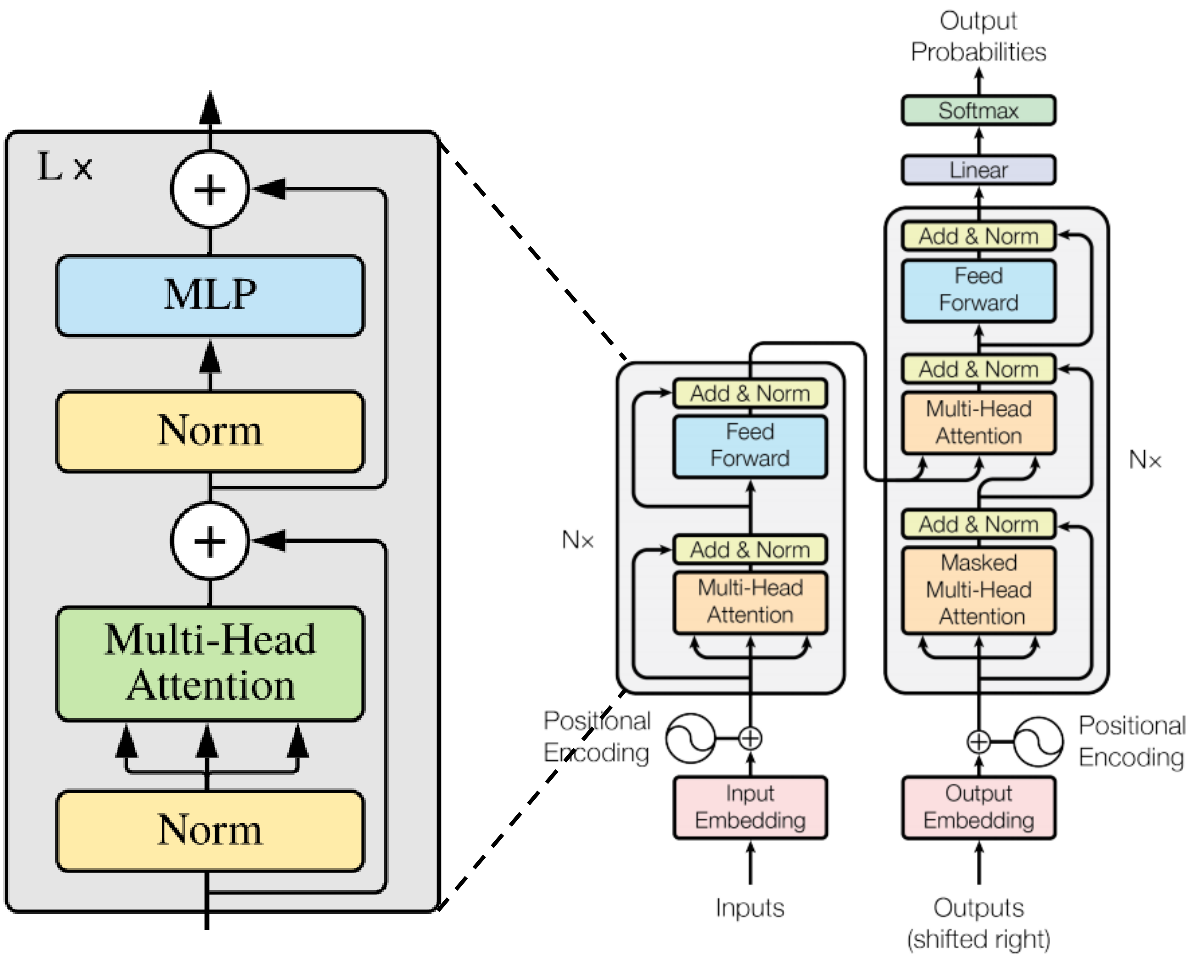
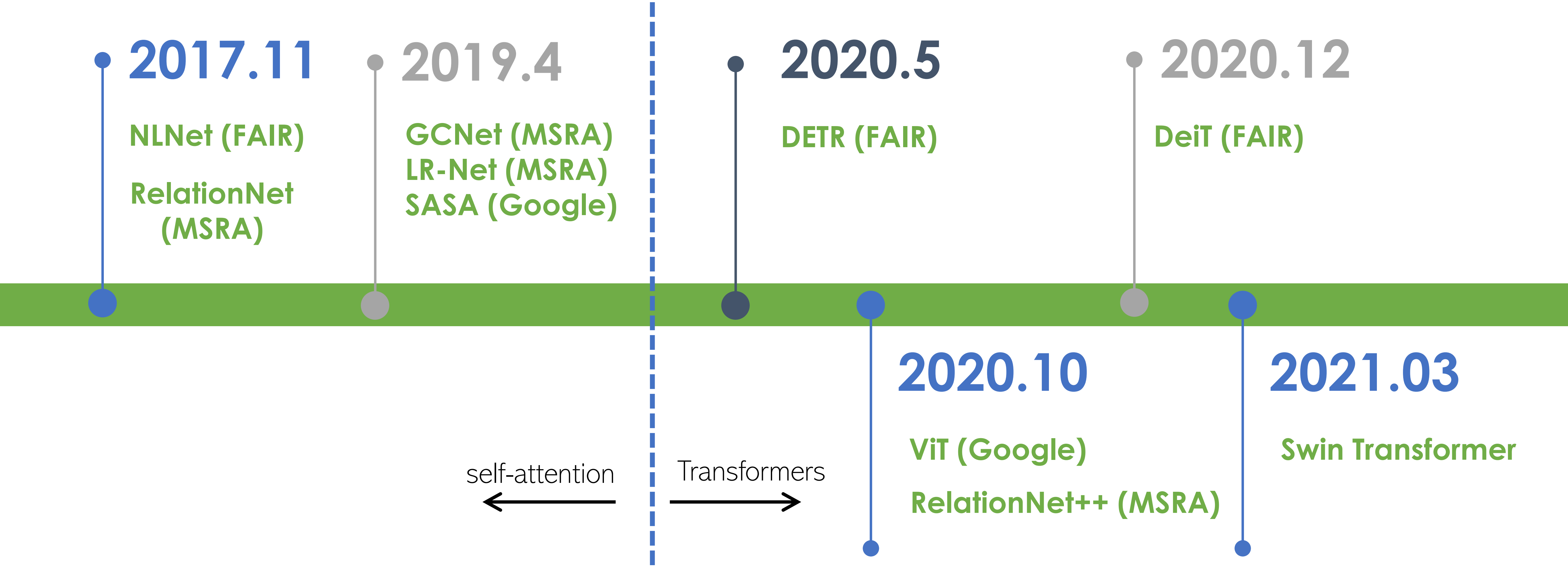


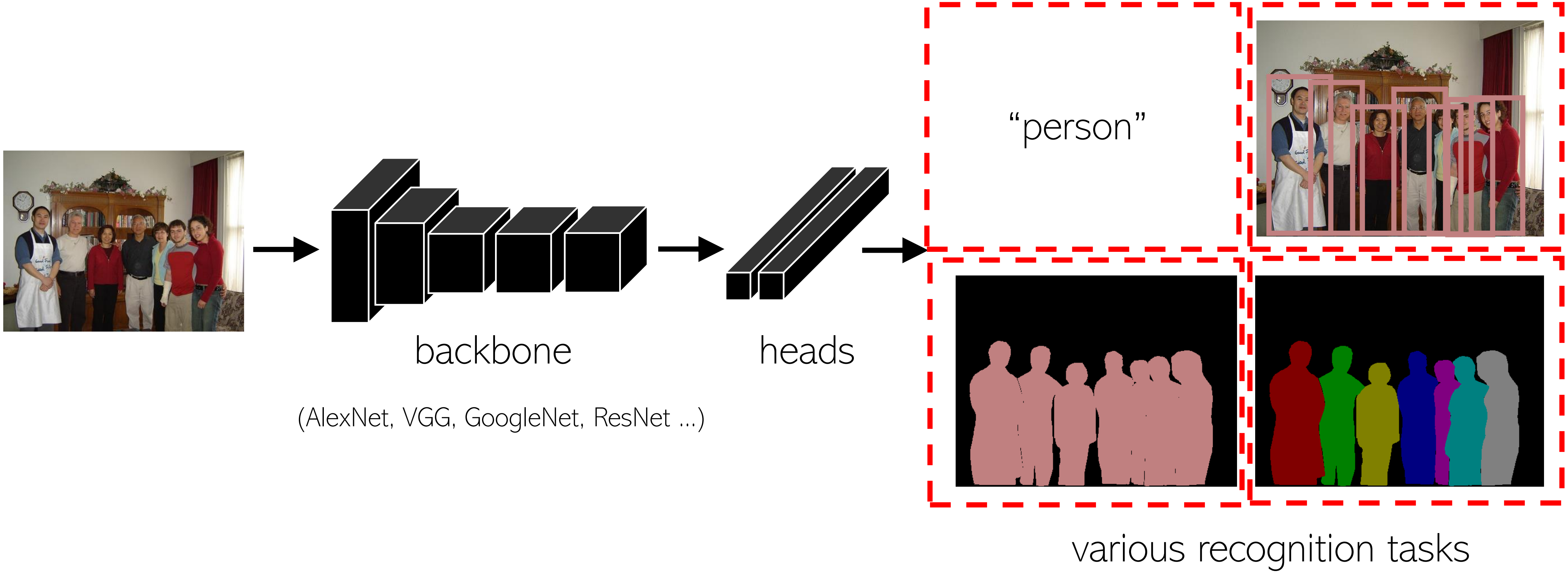
Figure 1: The Transformer - model architecture.

Can NLP/CV share the same basic modules?

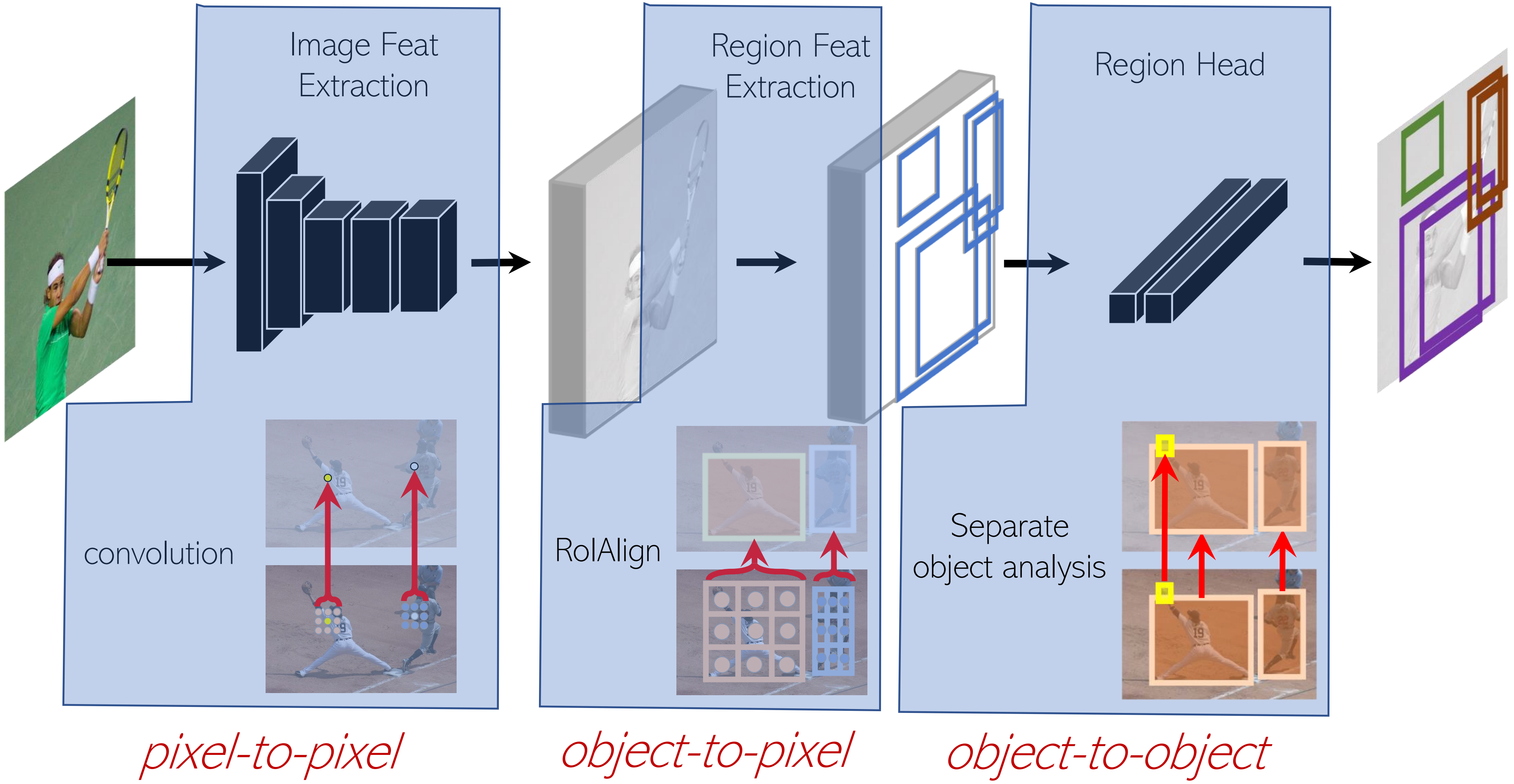
- Adapting self-attention/Transformer layers for CV modeling



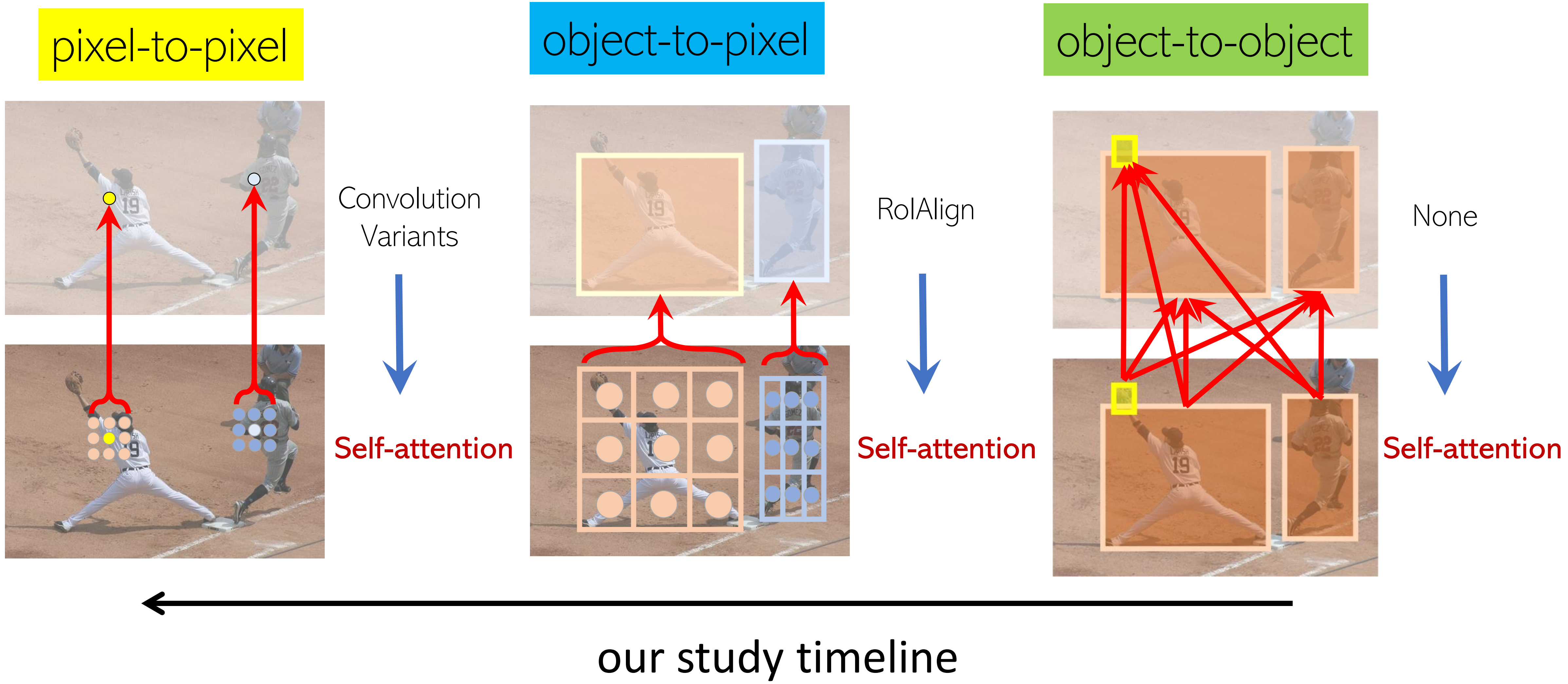
Visual Recognition Paradigm



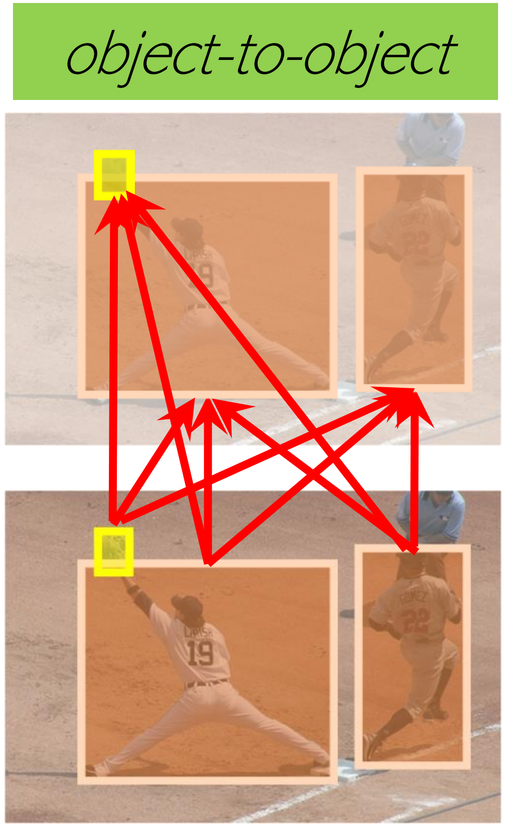
An Object Detection Example



Relationship Modeling of Basic Visual Elements



Object-to-Object Relation Modeling

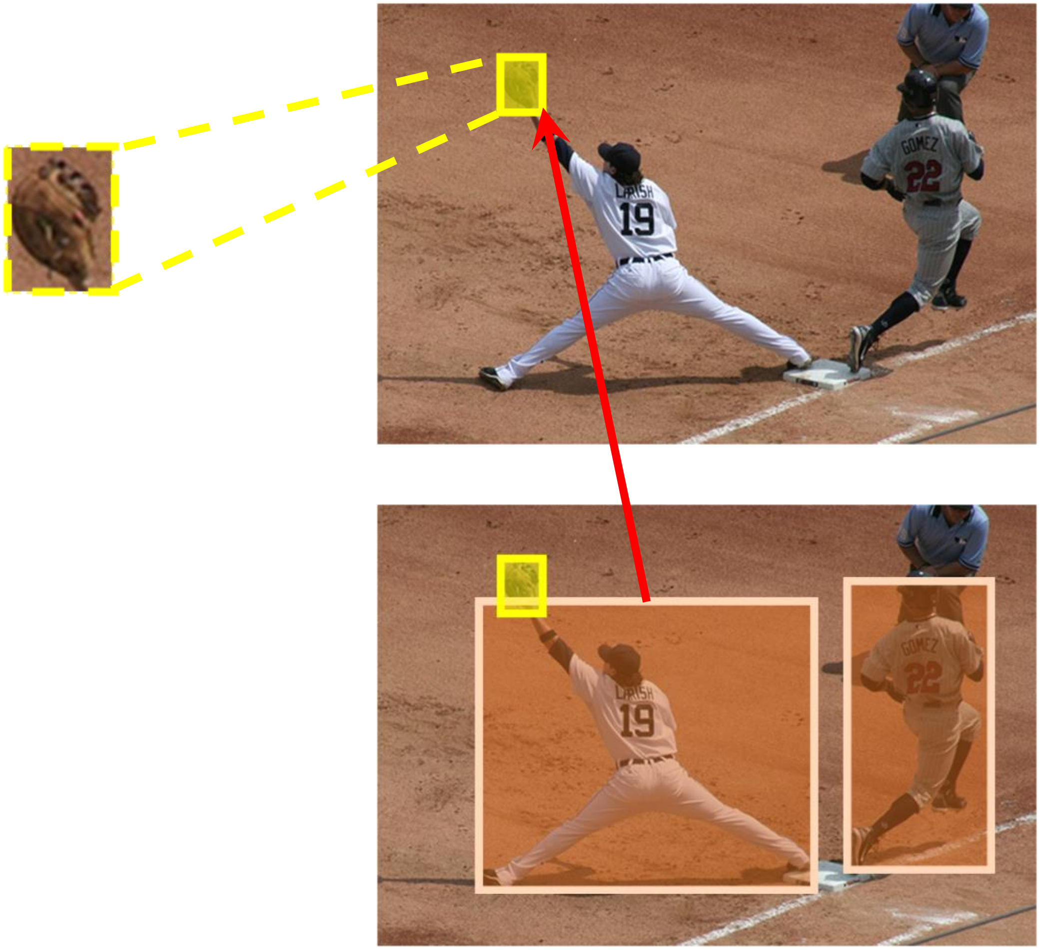


None ⟶ **Self-Attention**

- 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]

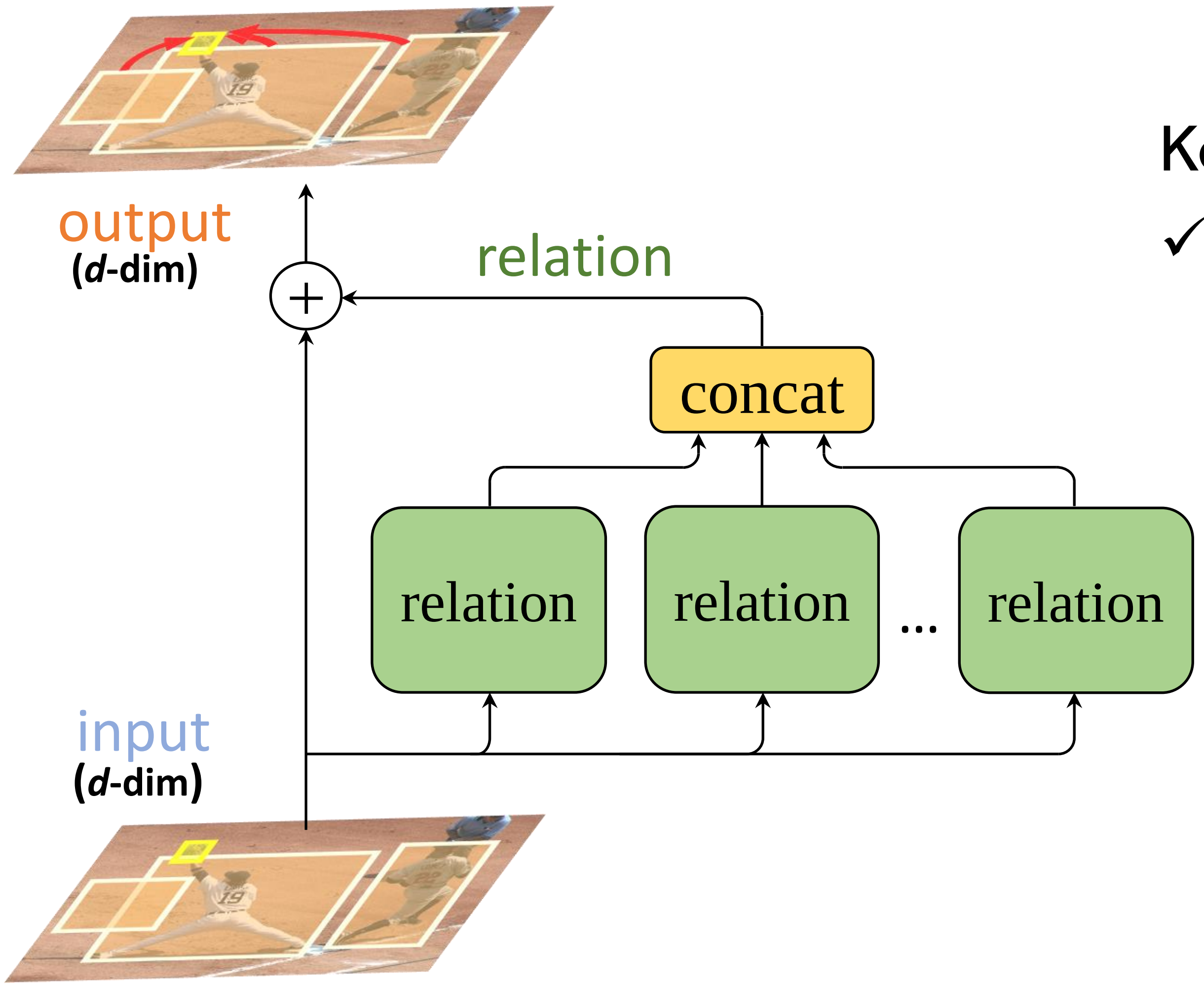
Object-to-Object Relation Modeling

Object-to-Object Relation Modeling



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

Object Relation Module

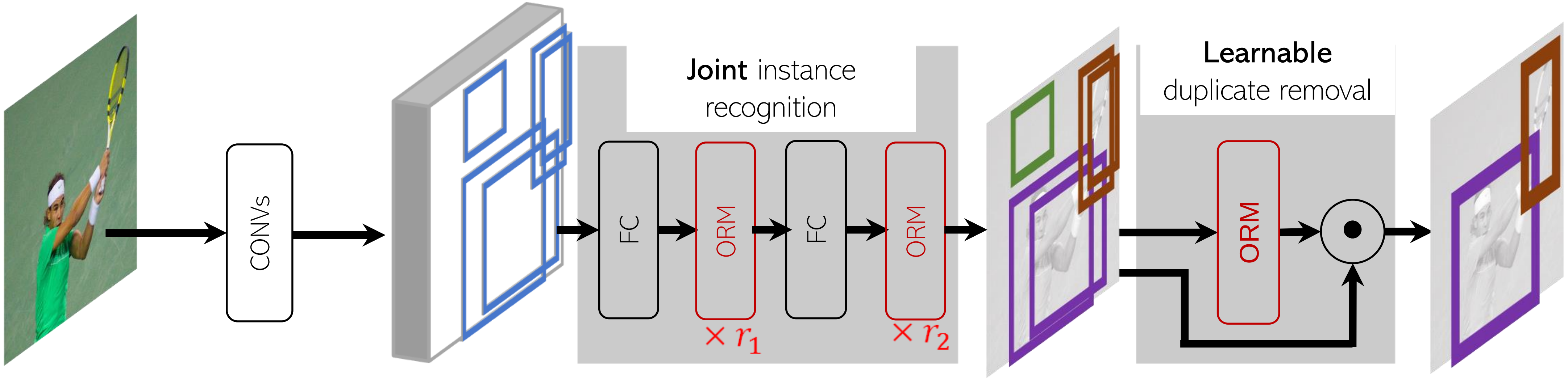


Key Feature

✓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



Relation Networks

ResNeXt-101 + FPN + DCN 45.0 → 45.9

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

Multi-Object Tracking

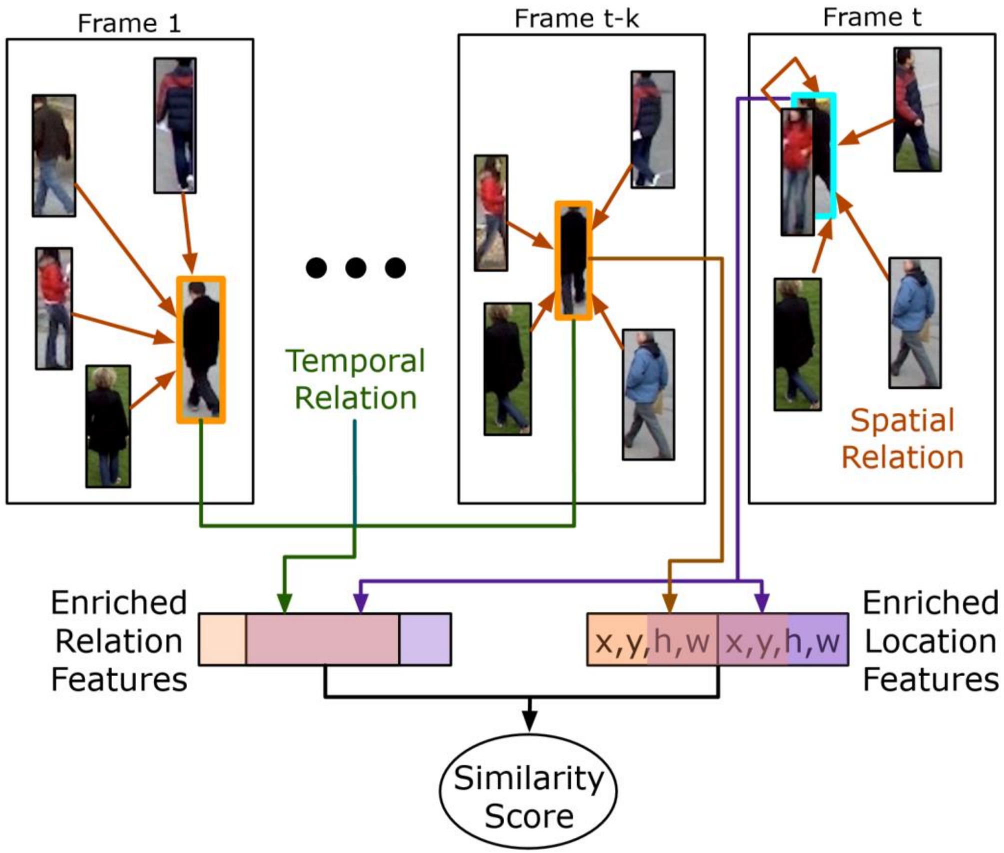


Table 4. Tracking Performance on MOT16 benchmark dataset.

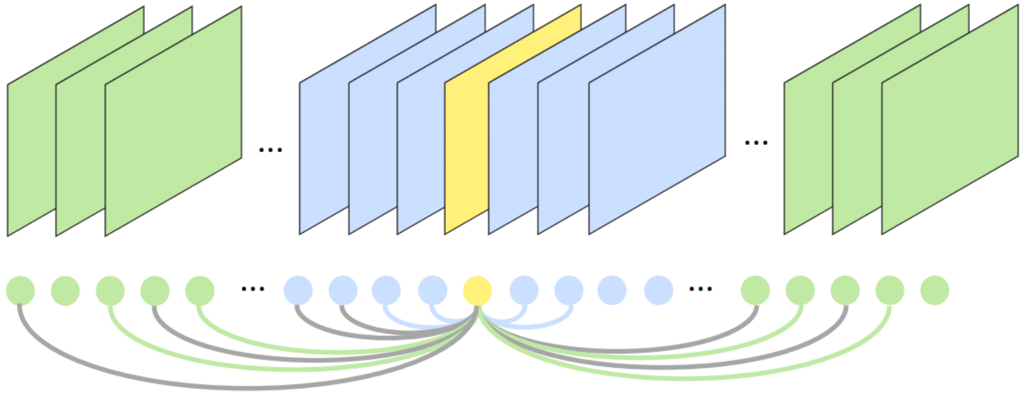
Mode	Method	MOTA↑	MOTP↑	IDF↑	IDP↑	IDR↑	MT(%)↑	ML(%)↓	FP↓	FN↓	IDS↓	Frag↓	AR↓
Offline	NOMT [11]	46.4	76.6	**53.3**	73.2	**41.9**	18.3	41.4	9,753	87,565	**359**	**504**	18.6
	MCjoint [23]	47.1	76.3	52.3	**73.9**	40.4	**20.4**	46.9	6,703	89,368	370	598	19.8
	NLLMPa [30]	47.6	78.5	47.3	67.2	36.5	17.0	40.4	5,844	89,093	629	768	18.8
	FWT [18]	47.8	75.5	44.3	60.3	35	19.1	**38.2**	8,886	**85,487**	852	1,534	24.8
	GCRA [32]	48.2	77.5	48.6	69.1	37.4	12.9	41.1	**5,104**	88,586	821	1,117	21.9
	LMP [54]	**48.8**	**79.0**	51.3	71.1	40.1	18.2	40.1	6,654	86,245	481	595	**17.8**
Online	oICF [24]	43.2	74.3	49.3	73.3	37.2	11.3	48.5	6,651	96,515	**381**	**1,404**	31.8
	STAM [12]	46.0	74.9	50	71.5	38.5	14.6	43.6	6,895	91,117	473	1,422	29.3
	DMAN [63]	46.1	73.8	**54.8**	**77.2**	42.5	**17.4**	42.7	7,909	89,874	532	1,616	23.4
	AMIR [46]	47.2	**75.8**	46.3	68.9	34.8	14.0	41.6	**2,681**	92,856	774	1,675	22.9
	MOTDT [10]	47.6	74.8	50.9	69.2	40.3	15.2	38.3	9,253	85,431	792	1,858	23.5
	ours	$\boxed{\mathbf{48.5}}$	73.7	53.9	72.8	**42.8**	17.0	**34.9**	9,038	**84,178**	747	2,919	**15.4**

Table 5. Tracking Performance on MOT17 benchmark dataset.

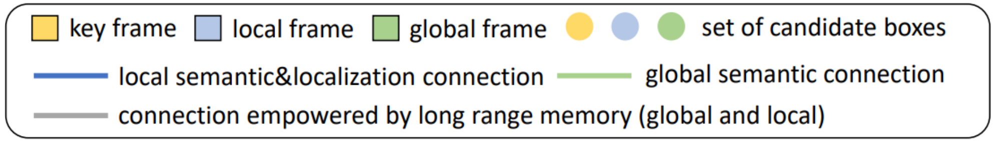
Mode	Method	MOTA↑	MOTP↑	IDF↑	IDP↑	IDR↑	MT(%)↑	ML(%)↓	FP↓	FN↓	IDS↓	Frag↓	AR↓
Offline	IOU [5]	45.5	76.9	39.4	56.4	30.3	15.7	40.5	**19,993**	281,643	5,988	7,404	36.5
	MHT_DLSTM [26]	47.5	**77.5**	51.9	71.4	40.8	18.2	41.7	25,981	268,042	2,069	3,124	28.8
	EDMT [8]	50.0	77.3	51.3	67	41.5	**21.6**	36.3	32,279	**247,297**	2,264	3,260	24.0
	MHT_DAM [25]	50.7	**77.5**	47.2	63.4	37.6	20.8	36.9	22,875	252,889	2,314	**2,865**	25.4
	jCC [22]	51.2	75.9	**54.5**	**72.2**	**43.8**	20.9	37	25,937	247,822	**1,802**	2,984	**20.3**
	FWT [18]	**51.3**	77	47.6	63.2	38.1	21.4	**35.2**	24,101	247,921	2,648	4,279	24.2
Online	PHD_GSDL [16]	48.0	**77.2**	49.6	68.4	39	17.1	**35.6**	23,199	265,954	3,998	8,886	32.5
	AM_ADM [49]	48.1	76.7	52.1	71.4	41	13.4	39.7	25,061	265,495	2,214	5,027	27.3
	DMAN [63]	48.2	75.9	55.7	**75.9**	44	19.3	38.3	26,218	263,608	2,194	5,378	26.6
	HAM_SADF [61]	48.3	**77.2**	51.1	71.2	39.9	17.1	41.7	**20,967**	269,038	**1,871**	**3,020**	25.2
	MOTDT [10]	**50.9**	76.6	52.7	70.4	42.1	17.5	35.7	24,069	250,768	2,474	5,317	23.1
	ours	$\boxed{\mathbf{50.9}}$	75.6	**56.5**	74.5	**45.5**	**20.1**	37.0	27,532	**246,924**	2,593	9,622	**18.2**

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

Video Object Detection

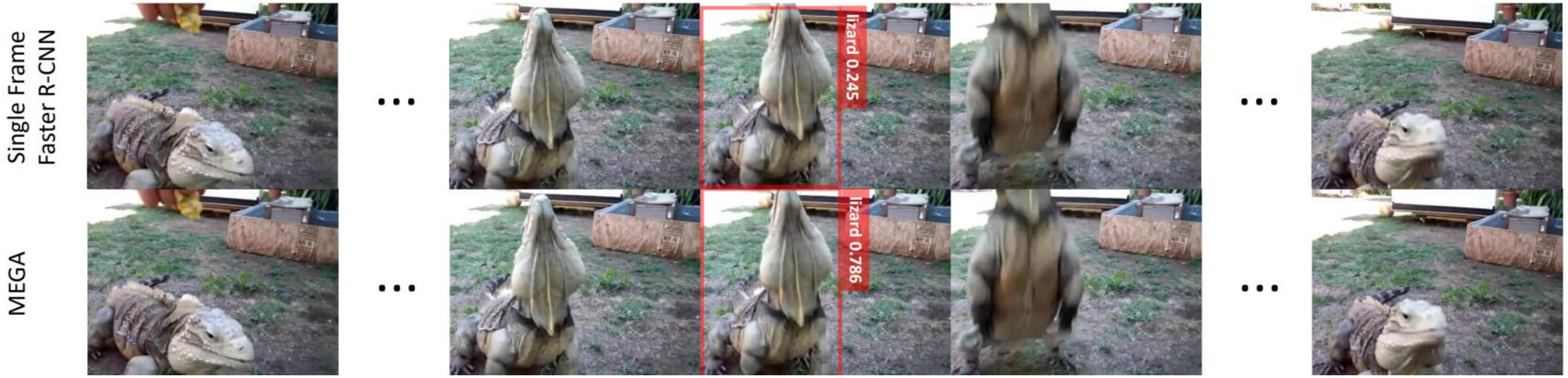


(d) the aggregation size of our proposed methods.



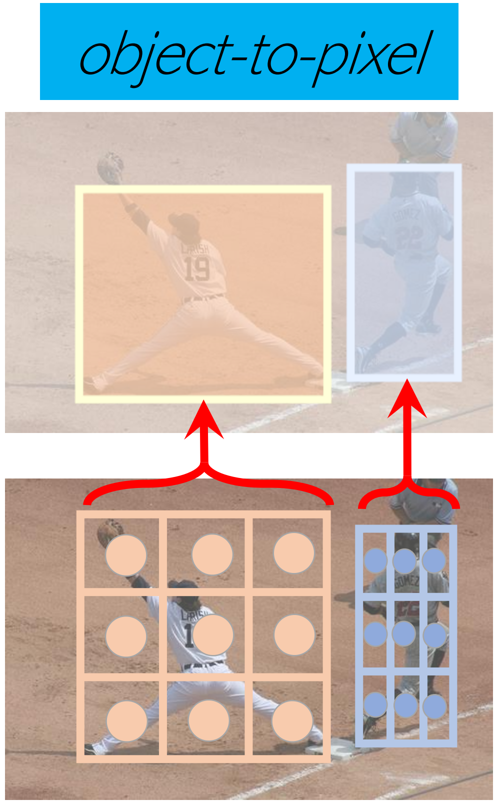
Methods	Backbone	mAP(%)
FGFA [36]	ResNet-101	78.4
ST-Lattice [4]	ResNet-101	79.6
MANet [27]	ResNet-101	80.3
D&T [9]	ResNet-101	80.2
STSN [1]	ResNet-101+DCN	80.4
STMN [31]	ResNet-101	80.5
SELSA [30]	ResNet-101	80.5
OGEMN [6]	ResNet-101+DCN	81.6
RDN [7]	ResNet-101	83.8
MEGA (ours)	ResNet-101	**84.5**
FGFA [36]	Inception-ResNet	80.1
D&T [9]	Inception-v4	82.0
RDN [7]	ResNeXt-101	84.7
MEGA (ours)	ResNeXt-101	**85.4**

Table 2. Performance comparison with state-of-the-art video object detection models with post-processing methods (*e.g.* Seq-NMS, Tube Rescoring, BLR).



Yihong Chen, Yue Cao, Han Hu, Liwei Wang. *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 (DETR) [ECCV'2020]
- RelationNet++ [NeurIPS'2020]

Learnable Object-to-Pixel Relation

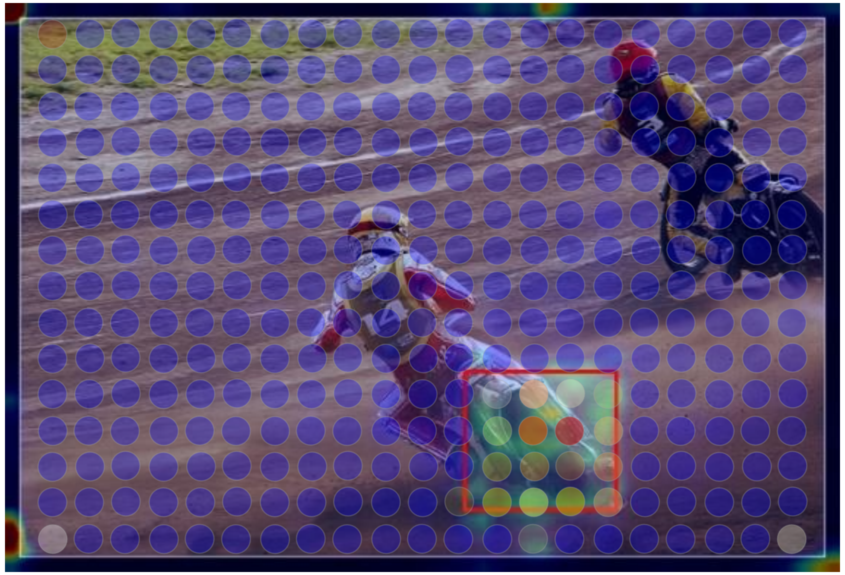
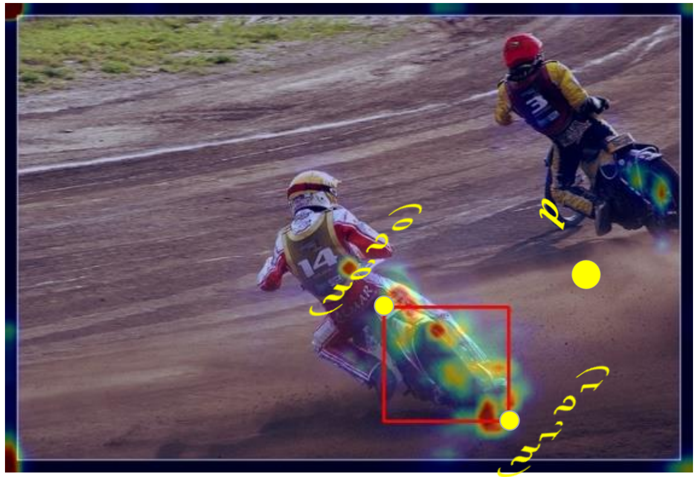


Image Feature to Region Feature



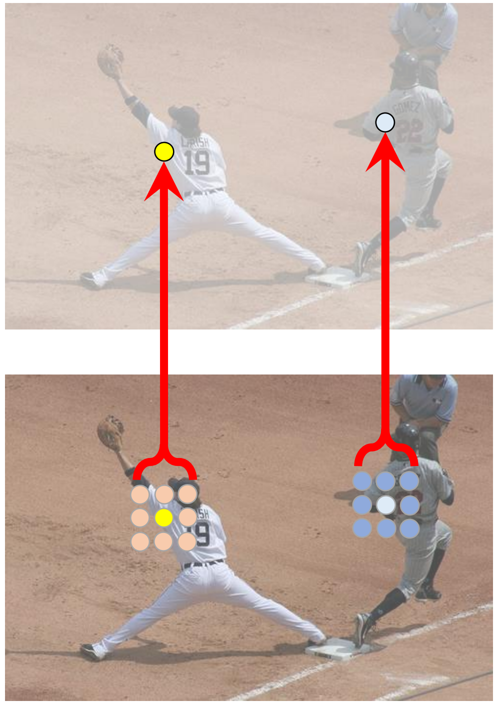
Geometric

Appearance

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

Pixel-to-Pixel Relation Modeling

pixel-to-pixel



Convolution Variants → **Self-Attention**

Usage

✓Complement convolution

✓Replace convolution

Complement Convolution

- "Convolution is too local"

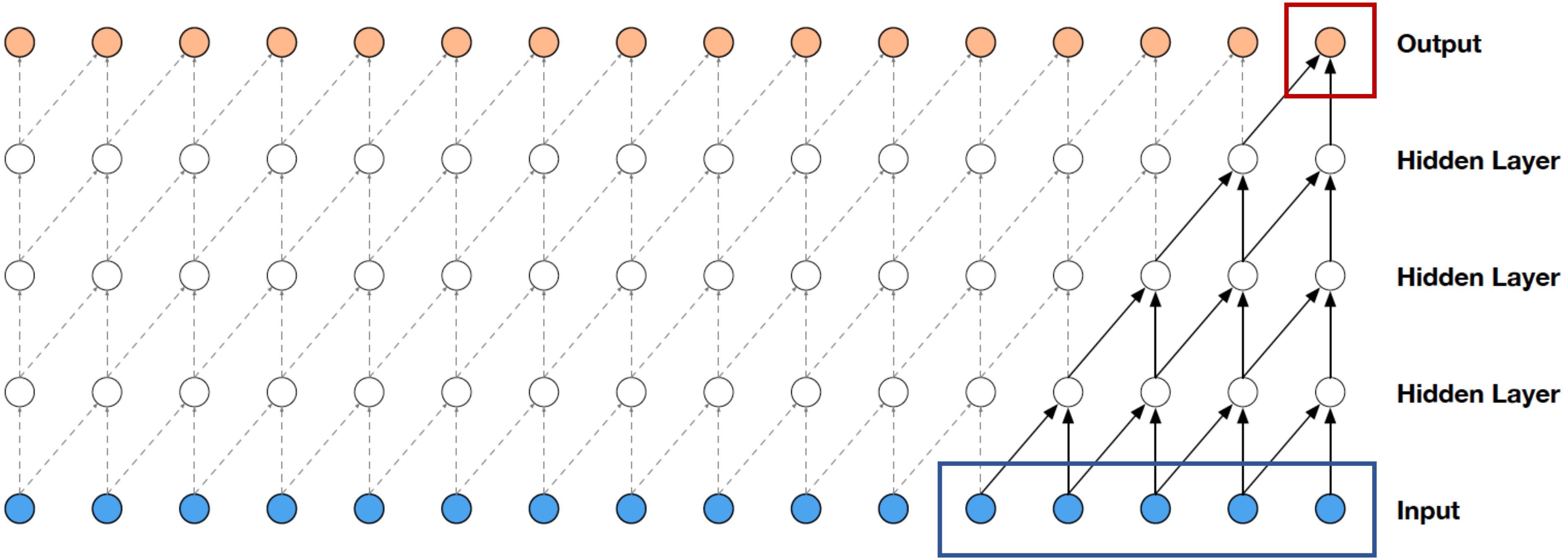
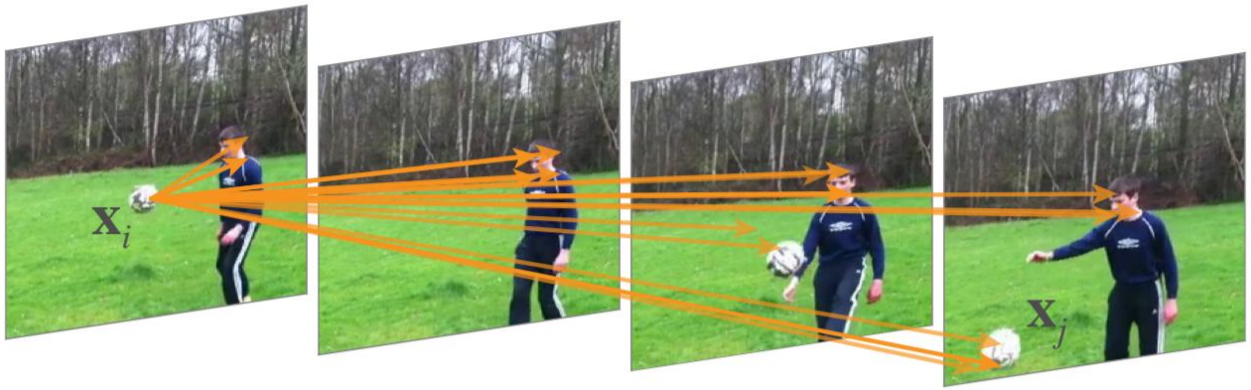
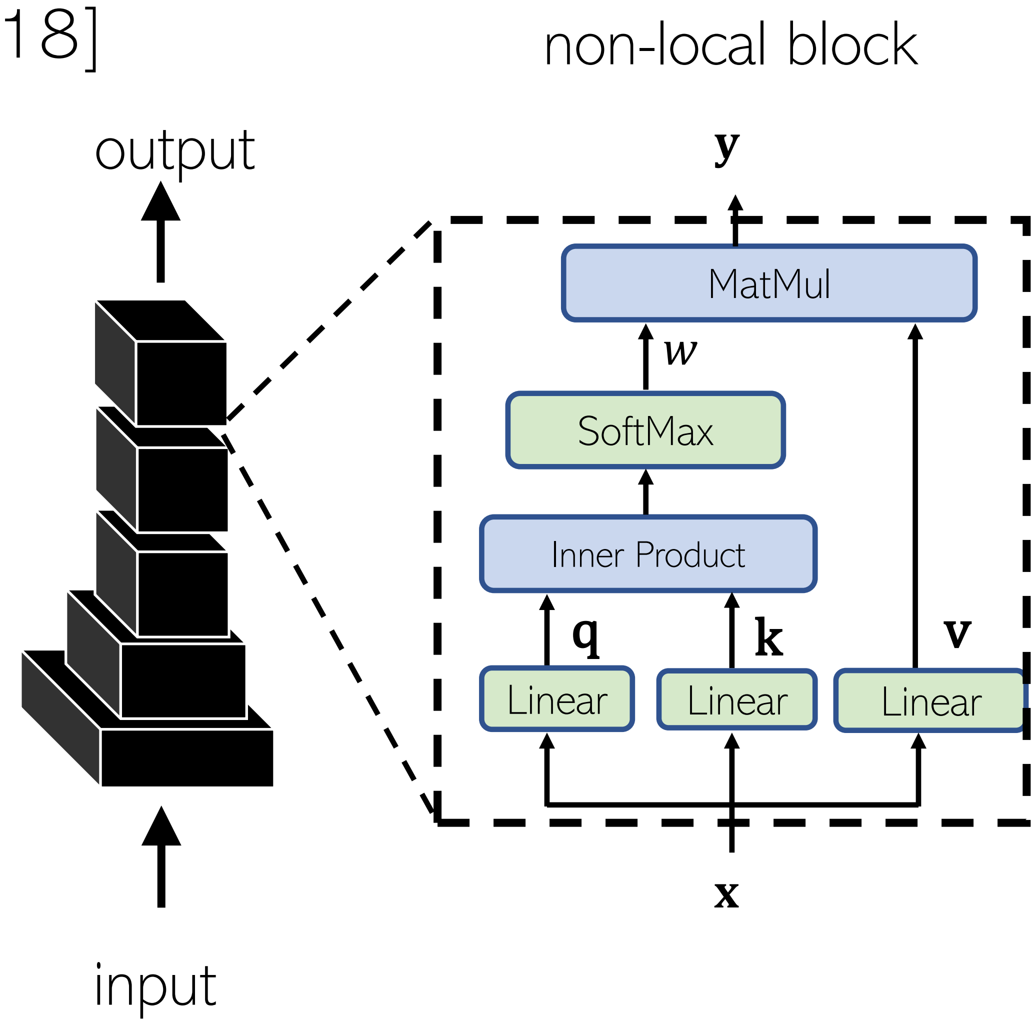


Figure credit: Van Den Oord et al.

Complement Convolution

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





The Degeneration Problem (2019)

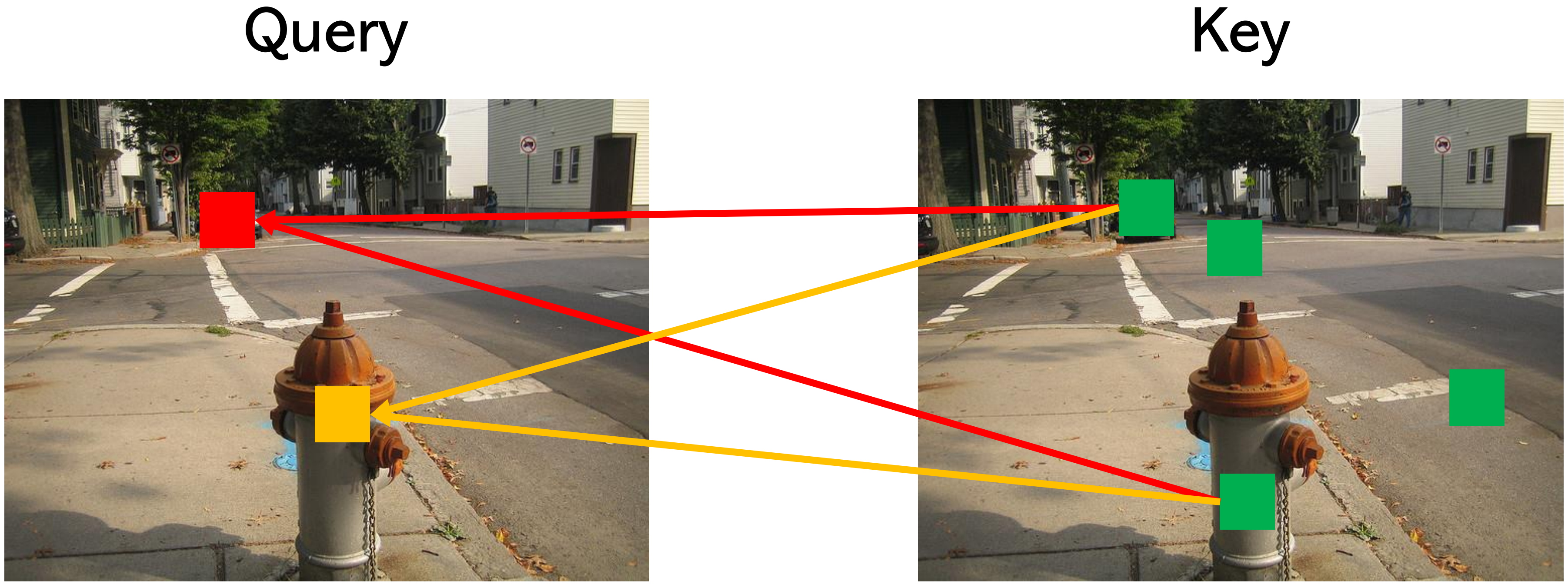
- Expectation of Ideally Learnt Relation
  - Different queries affected by **different** key

Query **Key**

Yue Cao, Jiarui Xu et al. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond. ICCVW'2019, TPAMI'2020

The Degeneration Problem (2019)

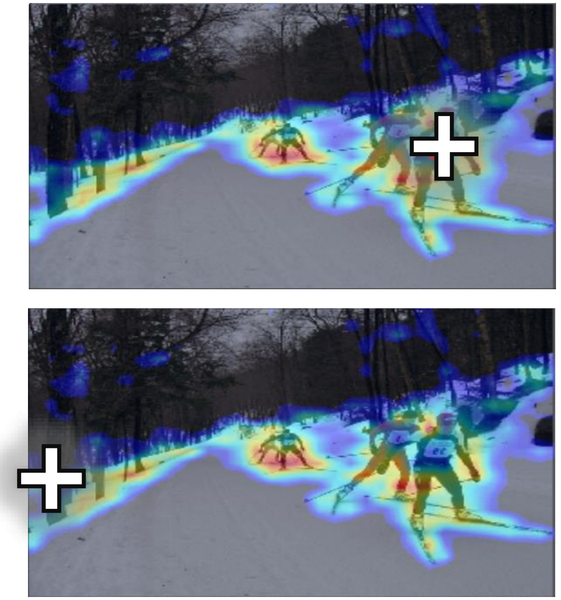
- What does the Self-Attention Learn?
  - Different queries affected by the **same** keys



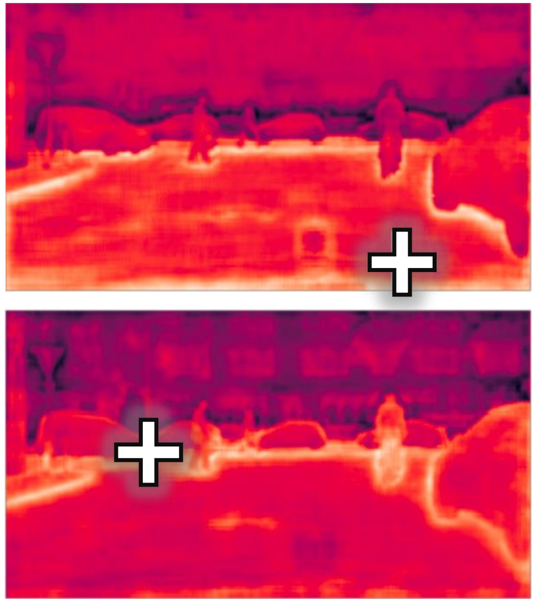
Yue Cao, Jiarui Xu et al. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond. ICCVW'2019, TPAMI'2020

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

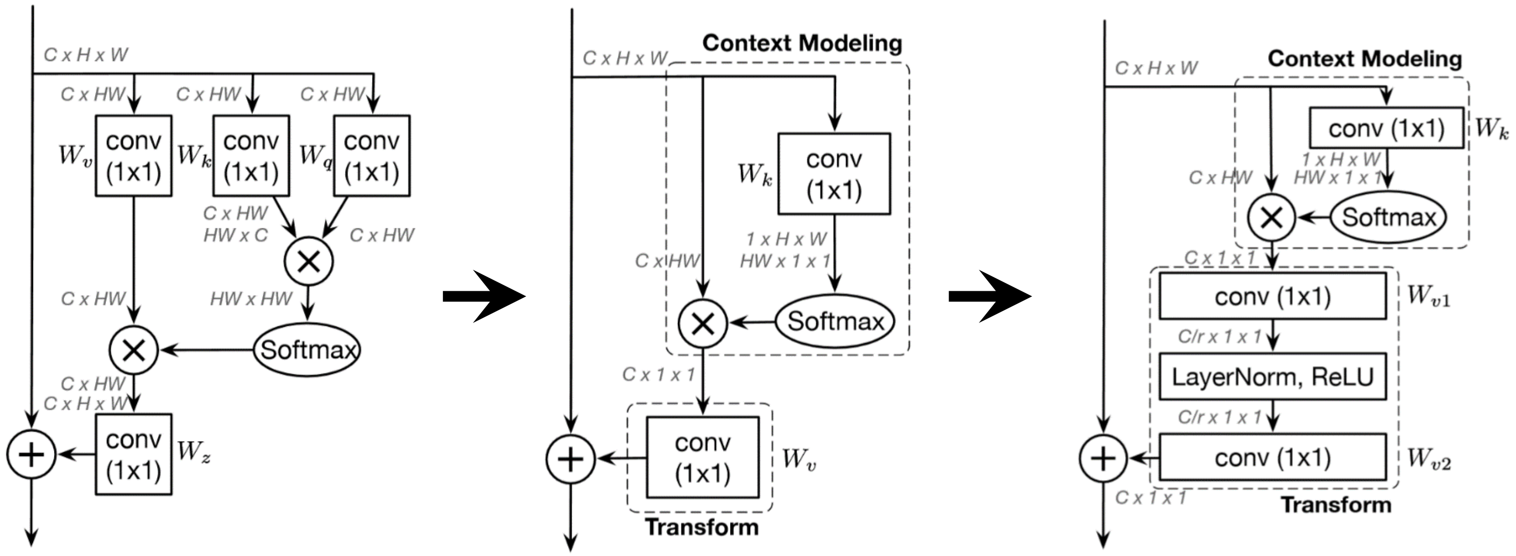


Semantic Segmentation

[GCNet, ICCVW'2019 & TPAMI'2020]

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

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 + ResNet-50 + 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

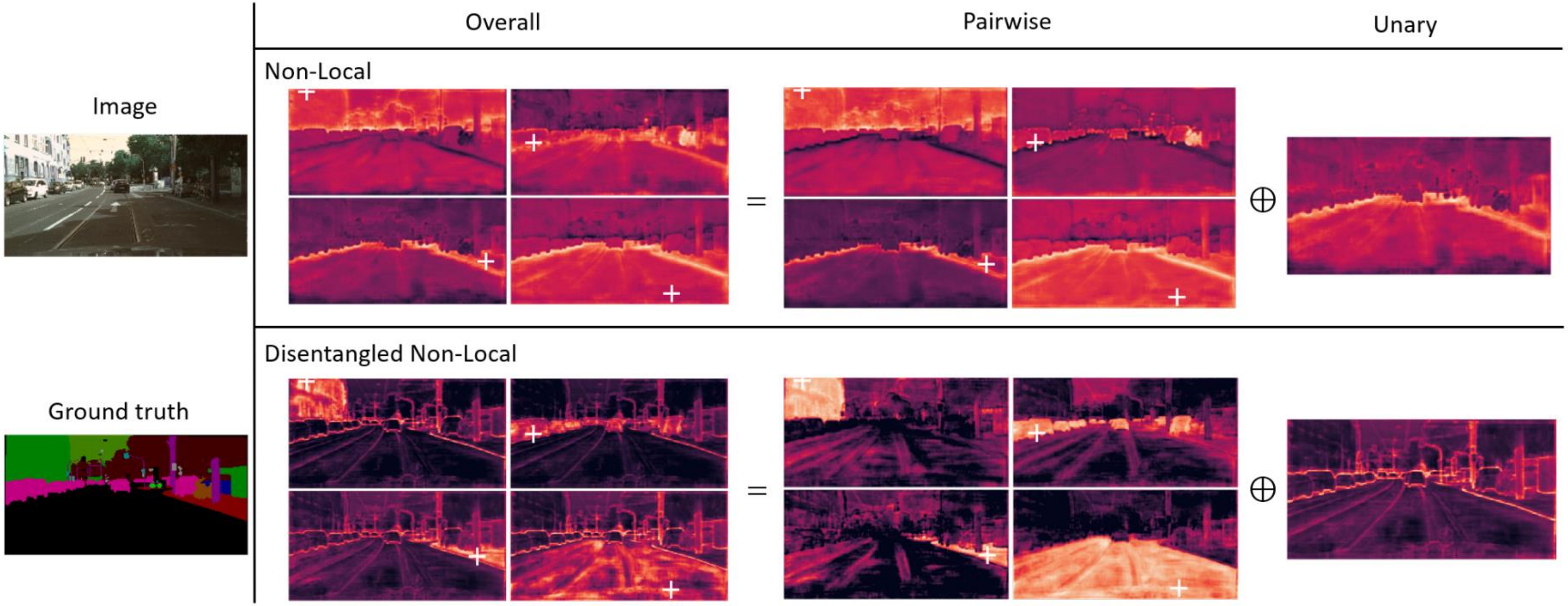
+2.2 mAP **+1.9 mAP**

with little computation and model size overhead!

Yue Cao, Jiarui Xu et al. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond. ICCVW'2019, TPAMI 2020

DNL: How to Effectively Model Pairwise Relationship?

- Disentangled design (ECCV'2020)



Minghao Yin, Zhuliang Yao, Yue Cao et al. Disentangled Non-Local Neural Networks. ECCV'2020

DNL: How to Effectively Model Pairwise?

- Disentangled design (ECCV'2020)

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	**53.7**
HRNet v2	HRNetV2-W48	54.0
Self-Attention	HRNetV2-W48	54.2
Ours	HRNetV2-W48	**55.3**

ADE20K

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

COCO

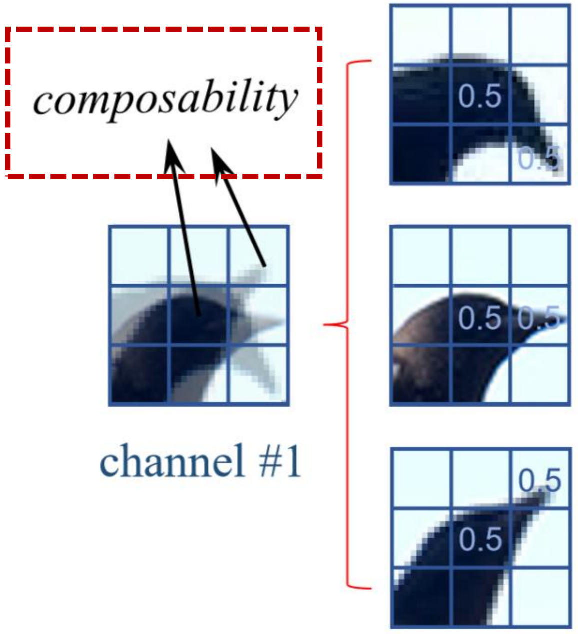
method	**Top-1 Acc**	**Top-5 Acc**
Baseline	74.9	91.9
Self-Attention	75.9	92.2
Ours	**76.3**	**92.7**

Kinetics-400

Minghao Yin, Zhuliang Yao, Yue Cao et al. Disentangled Non-Local Neural Networks. ECCV'2020

Replace Convolution

Local relation layer

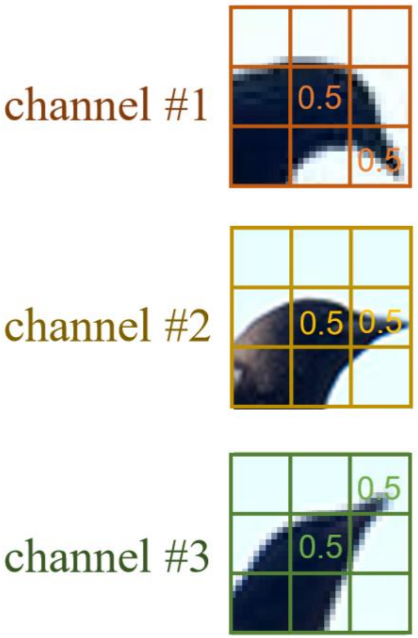


(1 channel)

Adaptive filters

VS.

convolution layer



(3 channels)

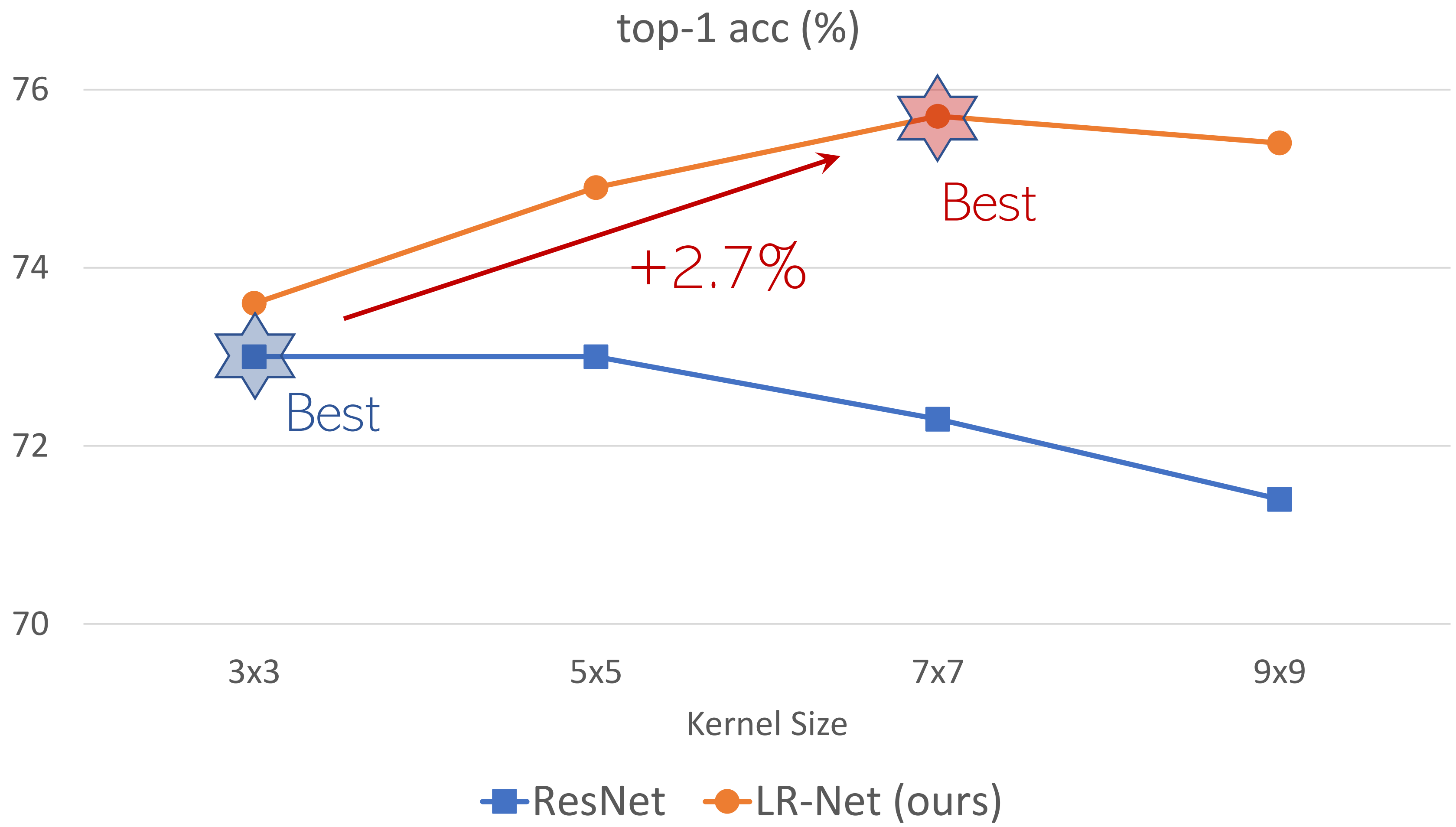
Fixed filters

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		**25.5**$\times 10^6$	**23.3**$\times 10^6$
FLOPs		**4.3**$\times 10^9$	**4.3**$\times 10^9$

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

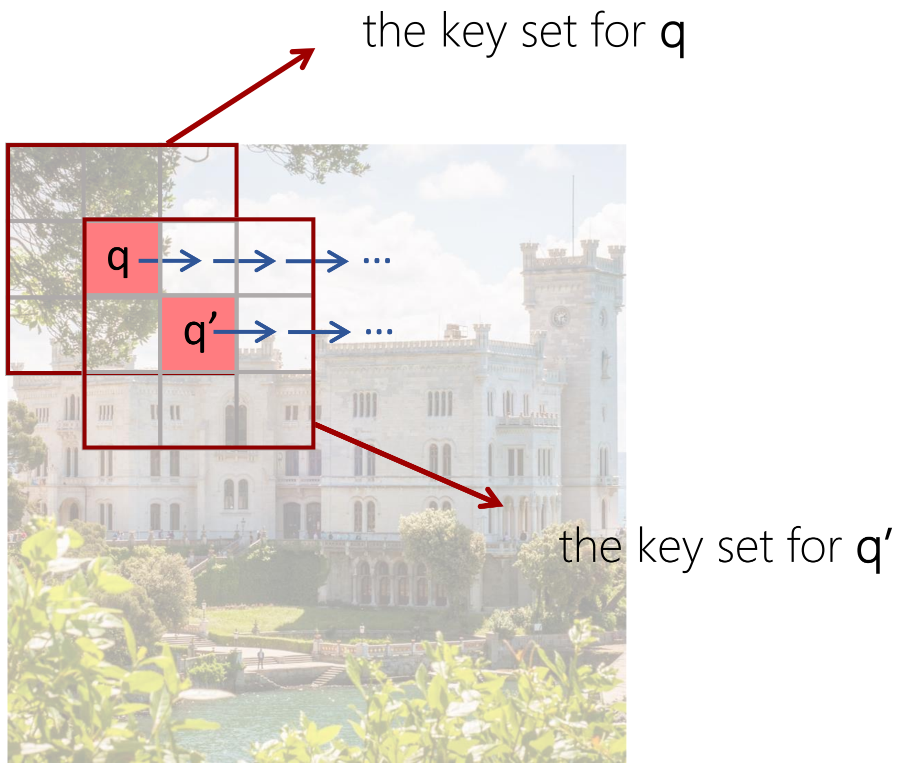
Classification on ImageNet (26 Layers)



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

But … slow in real computation

- Because different queries use different key sets



Stand-alone Self-attention (SASA)

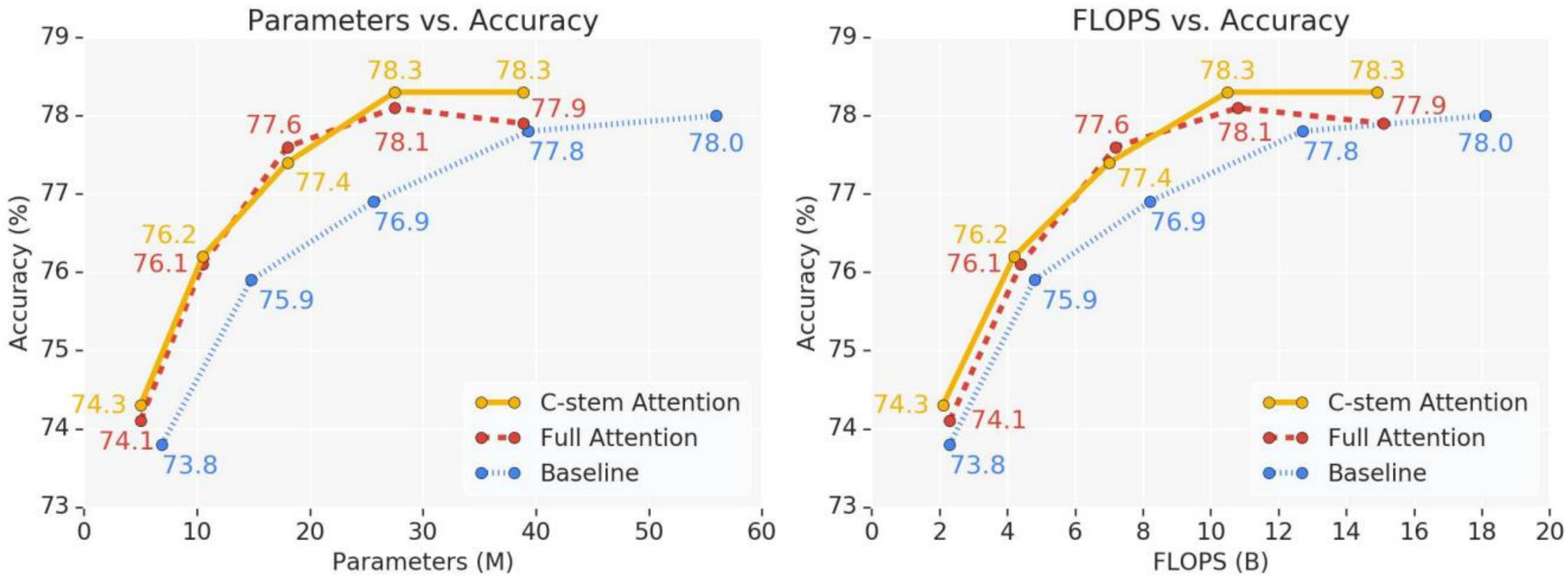


Figure 5: Comparing parameters and FLOPS against accuracy on ImageNet classification across a range of network widths for ResNet-50. Attention models have fewer parameters and FLOPS while improving upon the accuracy of the baseline.

Prajit Ramachandran et al. *Stand-Alone Self-Attention in Vision Models.* NeurIPS 2019

Self-attention Networks(SAN)

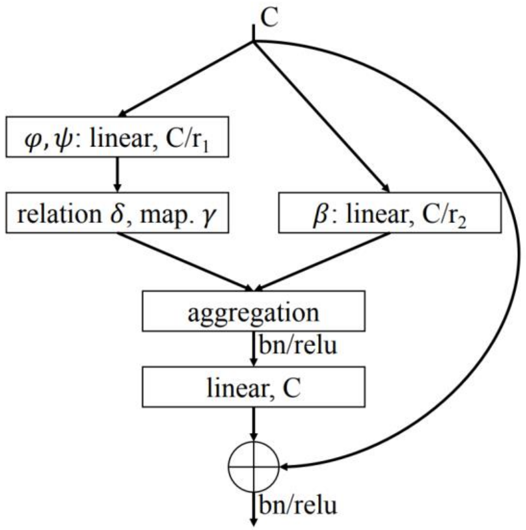


Figure 1. Our self-attention block. C is the channel dimensionality. The left stream evaluates the attention weights α, the right stream transforms the features via a linear mapping β. Both streams reduce the channel dimensionality for efficient processing. The outputs of the streams are aggregated via a Hadamard product and the dimensionality is subsequently expanded back to C.

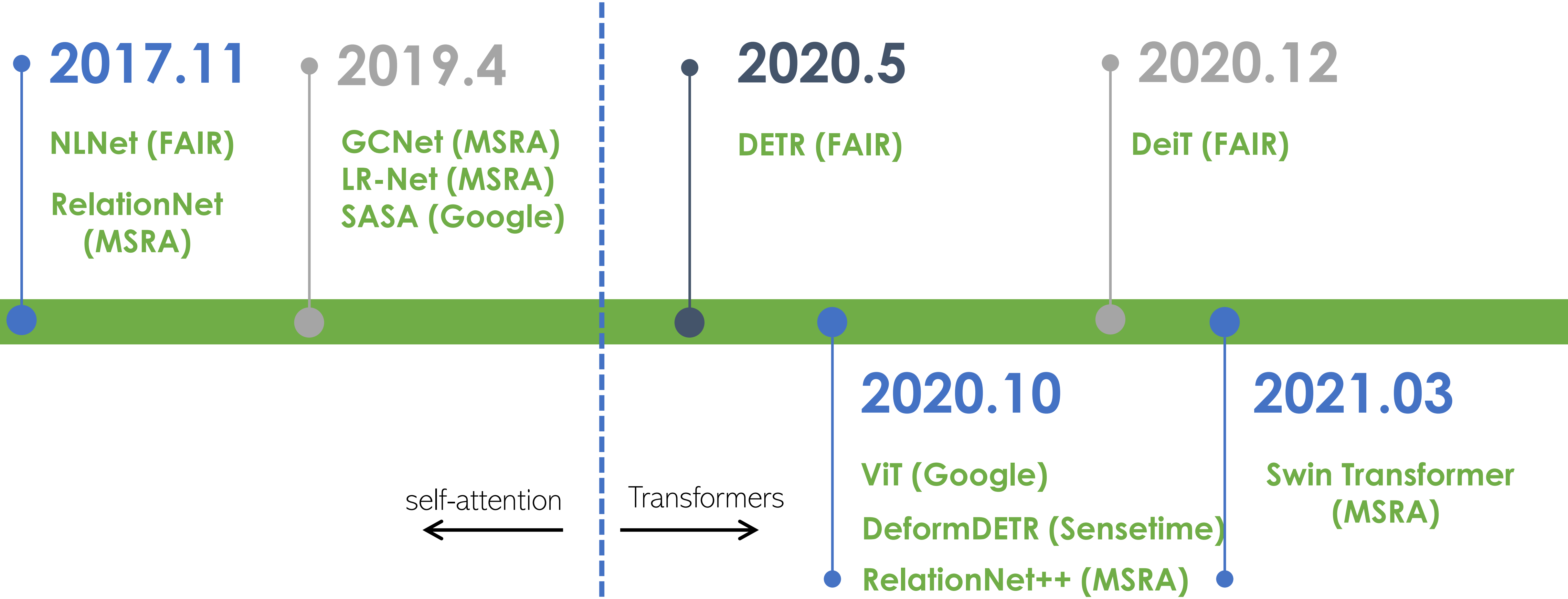
Method	no rotation		clockwise 90°		clockwise 180°	
	top-1	top-5	top-1	top-5	top-1	top-5
ResNet26	73.6	91.7	49.1(24.5)	72.7(19.0)	50.6(23.0)	75.4(16.3)
SAN10-pair.	74.9	92.1	51.8(**23.1**)	74.6(**17.5**)	54.7(**20.2**)	78.5(**13.6**)
SAN10-patch.	77.1	93.5	53.1(24.0)	75.7(17.8)	54.6(22.5)	78.4(15.1)
ResNet38	76.0	93.0	51.2(24.8)	74.2(18.8)	52.2(23.8)	76.9(16.1)
SAN15-pair.	76.6	93.1	54.5(**22.1**)	77.1(**16.0**)	57.9(**18.7**)	80.8(**12.3**)
SAN15-patch.	78.0	93.9	53.7(24.5)	76.1(17.8)	56.0(22.2)	79.5(14.4)
ResNet50	76.9	93.5	52.6(24.3)	75.3(18.2)	52.9(24.0)	77.4(16.2)
SAN19-pair.	76.9	93.4	54.7(**22.2**)	77.1(**16.3**)	58.0(**18.9**)	80.4(**13.0**)
SAN19-patch.	78.2	93.9	54.2(24.0)	76.3(17.6)	56.2(22.0)	79.5(14.4)

Robustness

Hengshuang Zhao et al. *Exploring Self-attention for Image Recognition.* CVPR 2020

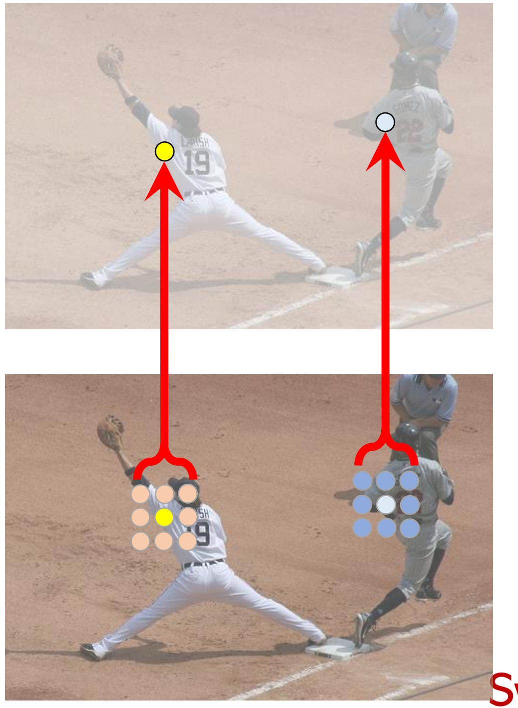
Can NLP/CV share the same basic modules?

- Adapting self-attention/Transformer layers for CV modeling



Relationship Modeling of Basic Visual Elements

pixel-to-pixel

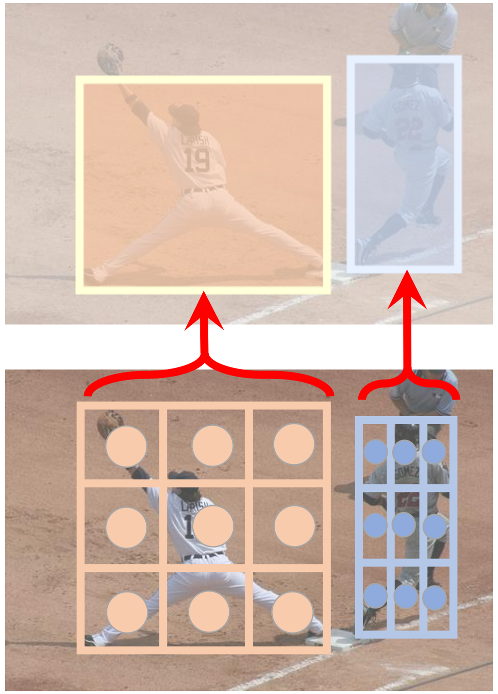


Convolution Variants, Self-att

Transformer

(ViT, DeiT Swin Transformer, CvT, PVT, T2T-ViT, Twins, CSwin …)

object-to-pixel



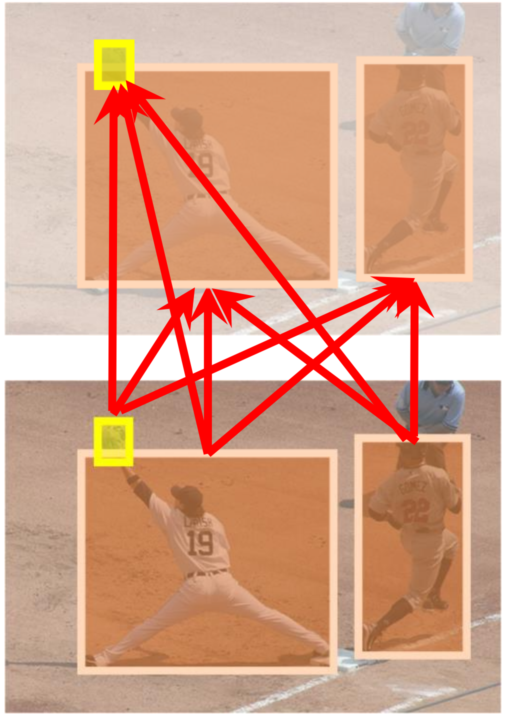
RoIAlign, Self-att

Self-att

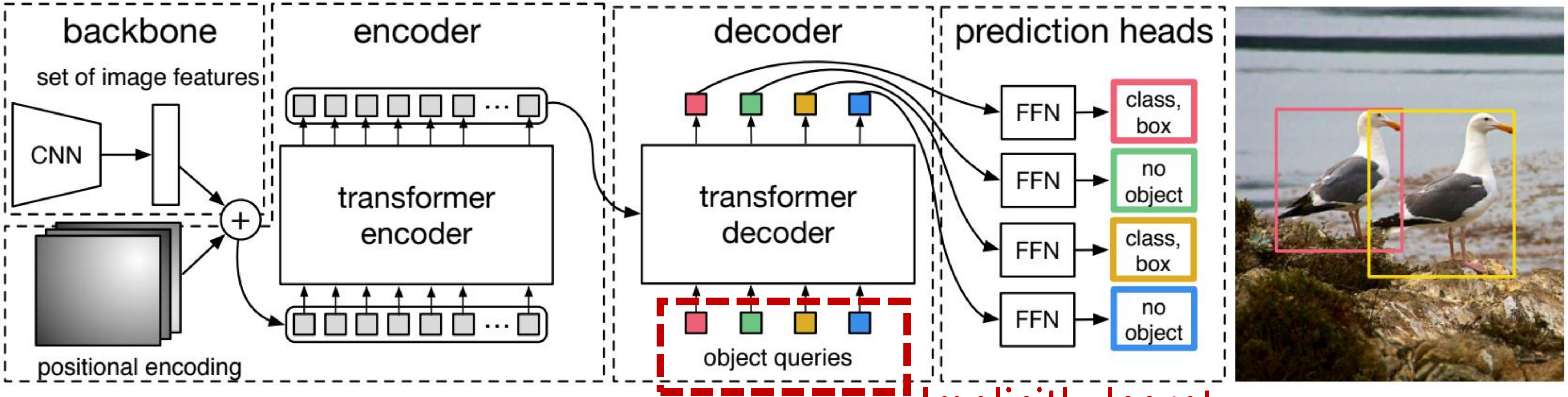
Transformer

(DETR, Deformable DETR, Cond DETR, Pix2Seq …)

object-to-object



Transformer Detectors (DETR)



Implicitly learnt

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

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

Deformable Transformer Detectors

- global att -> deformable att

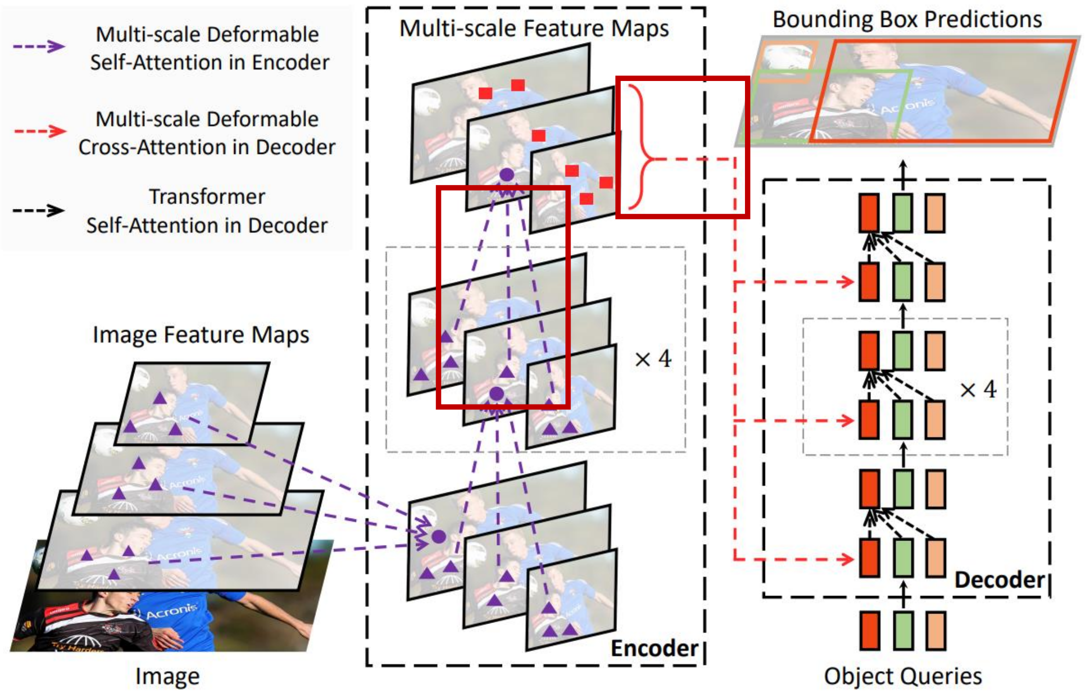
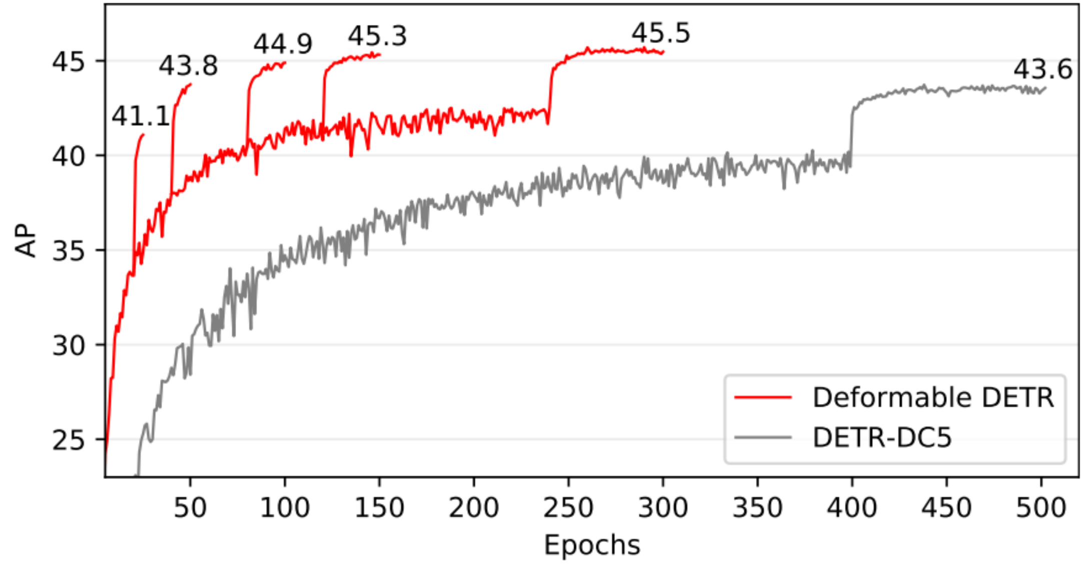


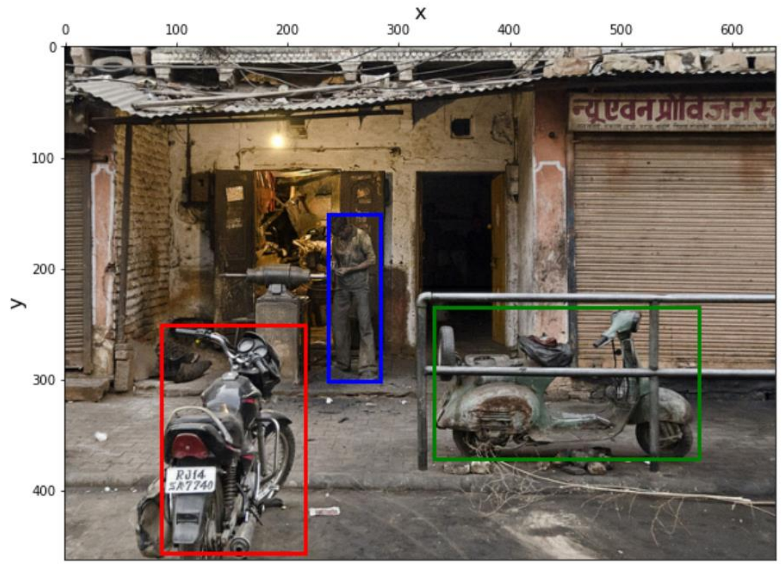
Figure 1: Illustration of the proposed Deformable DETR object detector.



Method	Backbone	TTA	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FCOS (Tian et al., 2019)	ResNeXt-101		44.7	64.1	48.4	27.6	47.5	55.6
ATSS (Zhang et al., 2020)	ResNeXt-101 + DCN	✓	50.7	68.9	56.3	33.2	52.9	62.4
TSD (Song et al., 2020)	SENet154 + DCN	✓	51.2	71.9	56.0	33.8	54.8	64.2
EfficientDet-D7 (Tan et al., 2020)	EfficientNet-B6		52.2	71.4	56.3	-	-	-
Deformable DETR	ResNet-50		46.9	66.4	50.8	27.7	49.7	59.9
Deformable DETR	ResNet-101		48.7	68.1	52.9	29.1	51.5	62.0
Deformable DETR	ResNeXt-101		49.0	68.5	53.2	29.7	51.7	62.8
Deformable DETR	ResNeXt-101 + DCN		50.1	69.7	54.6	30.6	52.8	64.7
Deformable DETR	ResNeXt-101 + DCN	✓	52.3	71.9	58.1	34.4	54.4	65.6

Xizhou Zhu et al. Deformable DETR: Deformable Transformers for end-to-end object detection. *ICLR 2021*

Pix2Seq



Random ordering (multiple samples):

327 370 653 444 1001 544 135 987 338 1004 508 518 805 892 1004 0

544 135 987 338 1004 327 370 653 444 1001 508 518 805 892 1004 0

508 518 805 892 1004 544 135 987 338 1004 327 370 653 444 1001 0

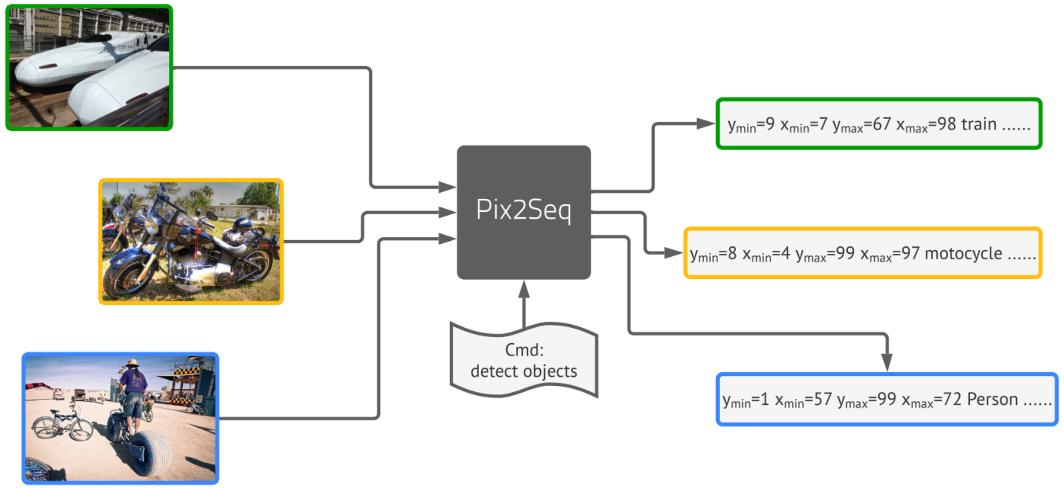


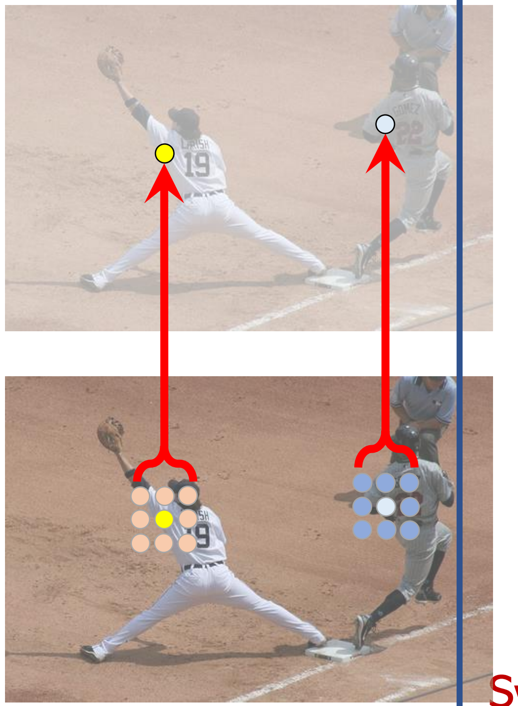
Figure 1: Illustration of Pix2Seq framework for object detection. The neural net perceives an image and generates a sequence of tokens that correspond to bounding boxes and class labels.

Method	Backbone	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN	R50-FPN	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster R-CNN+	R50-FPN	42M	42.0	62.1	45.5	26.6	45.4	53.4
DETR	R50	41M	42.0	62.4	44.2	20.5	45.8	61.1
Pix2seq (Ours)	R50	37M	**43.0**	61.0	45.6	25.1	46.9	59.4
Faster R-CNN	R101-FPN	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster R-CNN+	R101-FPN	60M	44.0	63.9	47.8	27.2	48.1	56.0
DETR	R101	60M	43.5	63.8	46.4	21.9	48.0	61.8
Pix2seq (Ours)	R101	56M	**44.5**	62.8	47.5	26.0	48.2	60.3
Faster R-CNN	R50-DC5	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster R-CNN+	R50-DC5	166M	41.1	61.4	44.3	22.9	45.9	55.0
DETR	R50-DC5	41M	**43.3**	63.1	45.9	22.5	47.3	61.1
Pix2seq (Ours)	R50-DC5	38M	**43.2**	61.0	46.1	26.6	47.0	58.6
DETR	R101-DC5	60M	**44.9**	64.7	47.7	23.7	49.5	62.3
Pix2seq (Ours)	R101-DC5	57M	**45.0**	63.2	48.6	28.2	48.9	60.4

Ting Chen, Geoffrey Hinton et al. *Pix2Seq: A Language Modeling Framework for Object Detection*. Arxiv 2021

Relationship Modeling of Basic Visual Elements

pixel-to-pixel

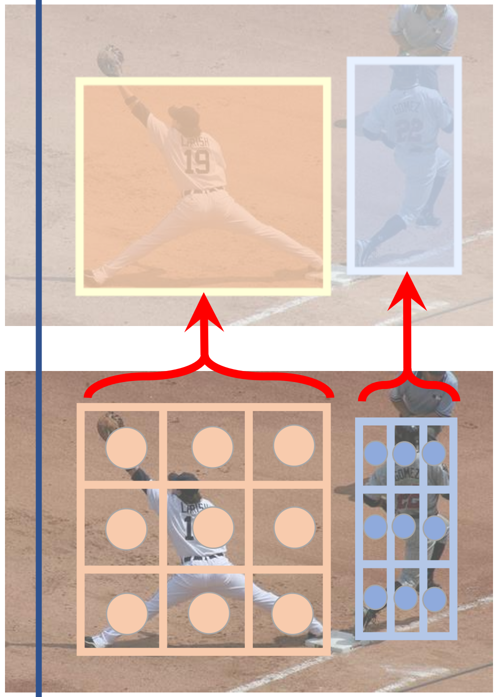


Convolution Variants, Self-att

Transformer

(ViT, DeiT Swin Transformer, CvT, PVT, DPT, T2T-ViT, Twins, Cswin …)

object-to-pixel



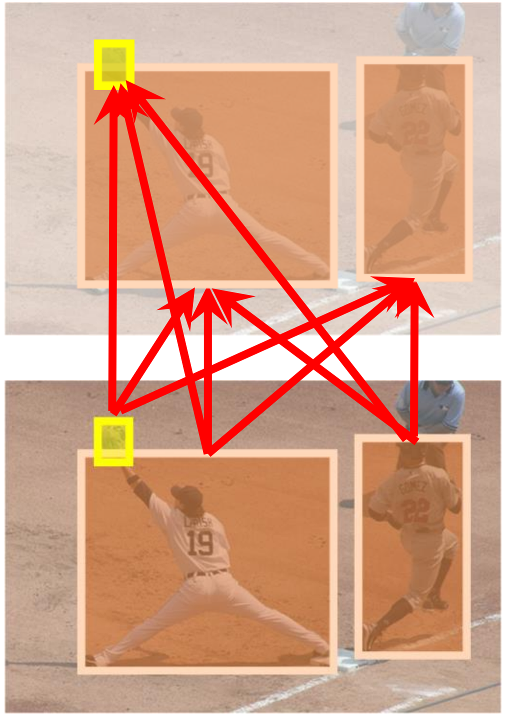
RoIAlign, Self-att

Self-att

Transformer

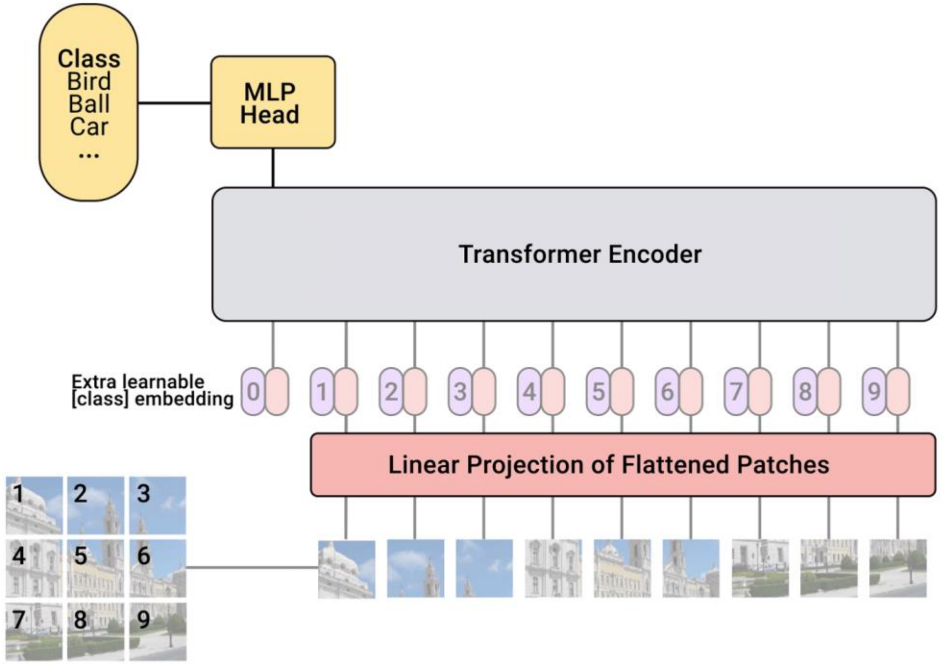
(DETR, Deformable DETR, Cond DETR, PixSeq …)

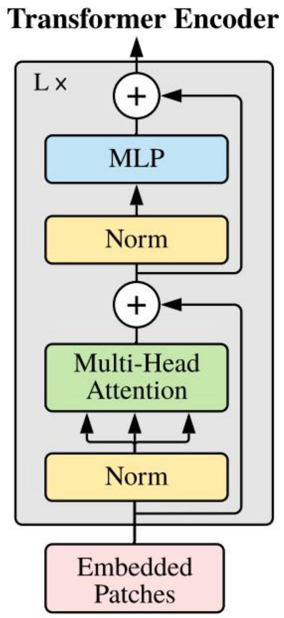
object-to-object



Vision Transformer (ViT)

- by Google Brain (2020.10)



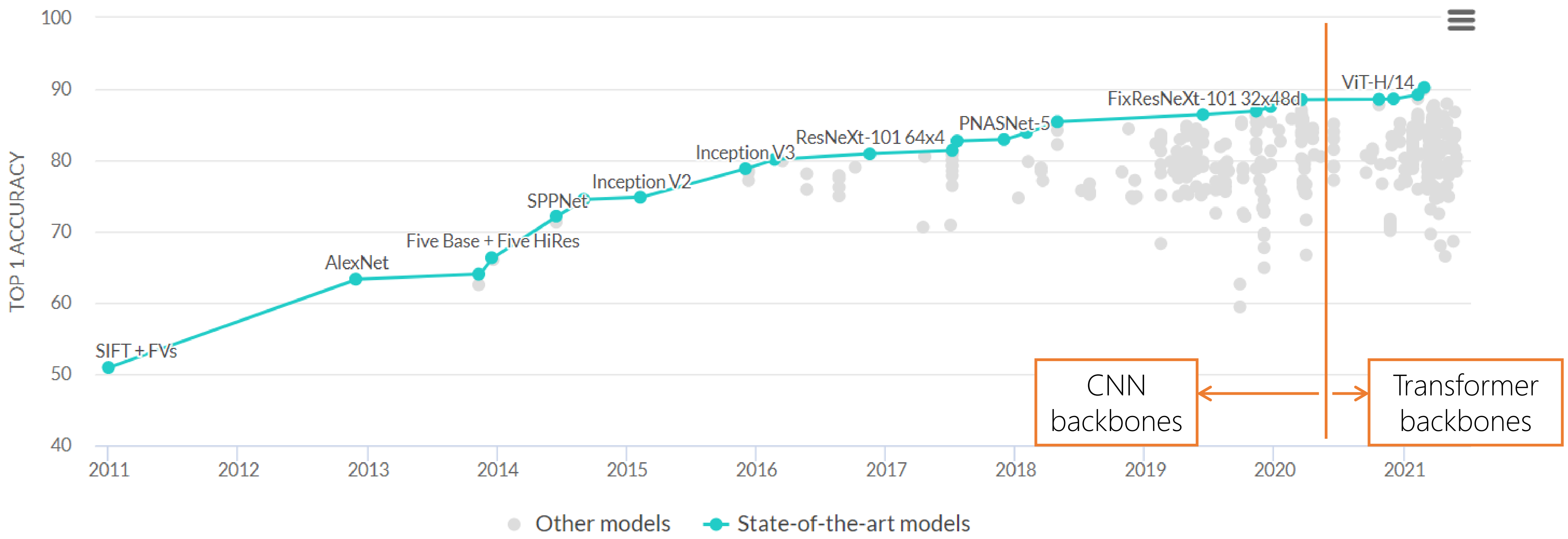


Model	FLOPs	Speed (TPU)
R50	4.3G	~2100 im/s
ViT-B/32	4.3G	~3000 im/s

Even with +50% speed-up due to shared a key set (globally) for all queries

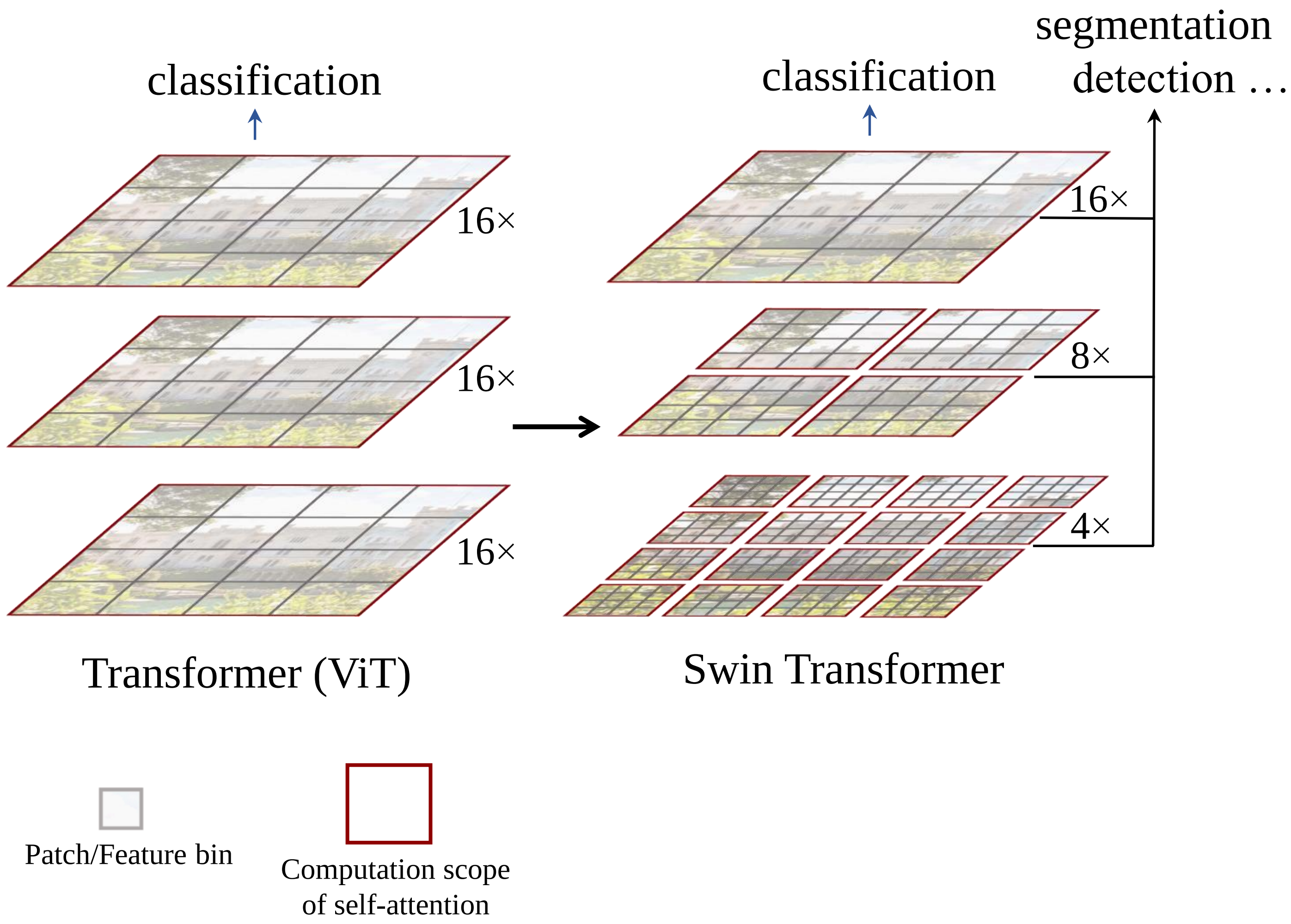
Alexey Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR' 2021
GiF Credit: https://github.com/lucidrains/vit-pytorch

Image Classification on IMAGENET



Swin Transformer =

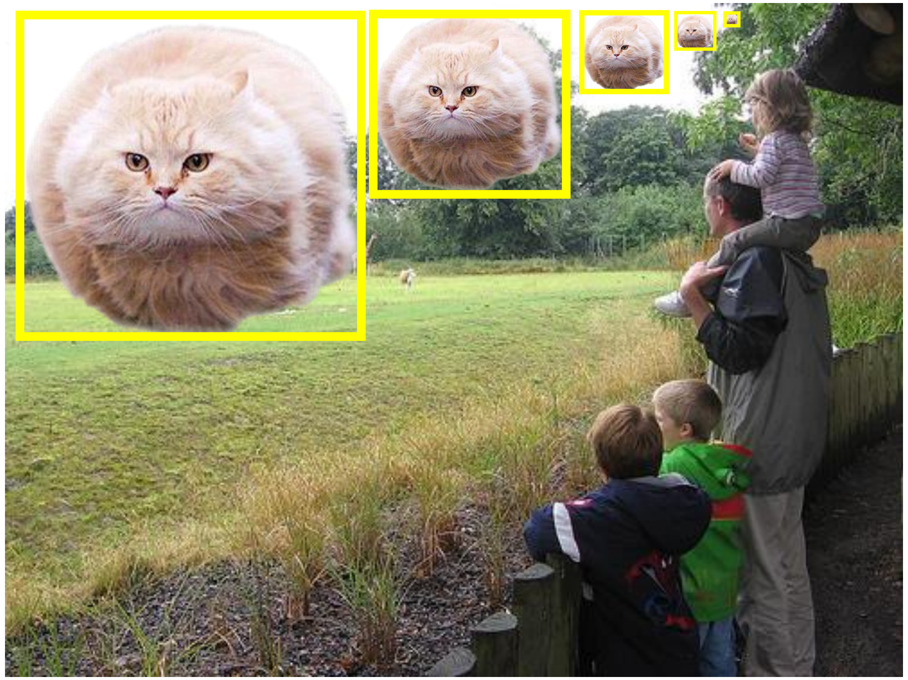
- Transformer
  - Strong modeling power
- + good priors for visual modeling
  - Hierarchy
  - Locality
  - Translational invariance

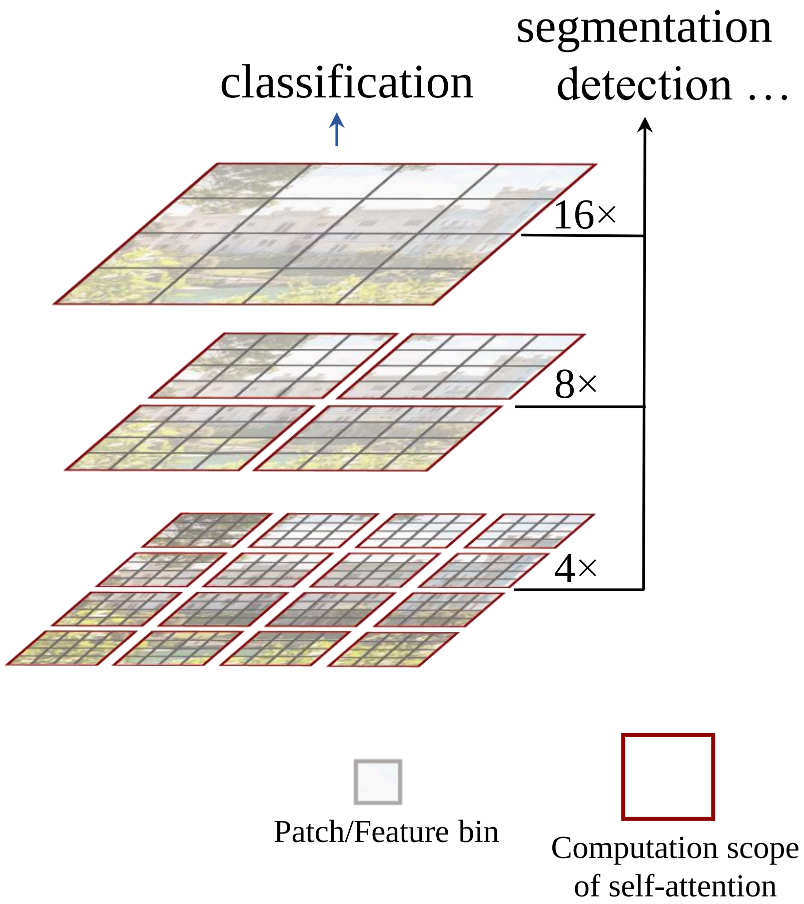


Ze Liu, Yutong Lin, Yue Cao, Han Hu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. ICCV 2021*

Hierarchy

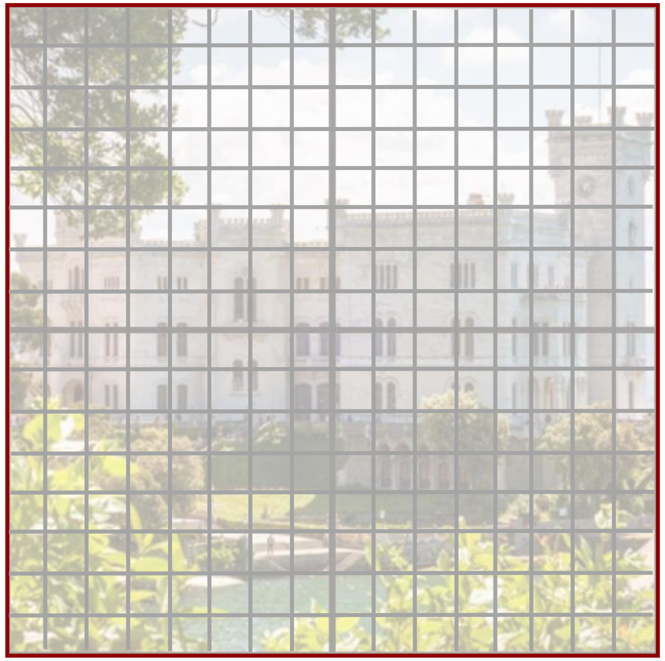
- Processing objects of different scales





Locality by non-overlapped windows

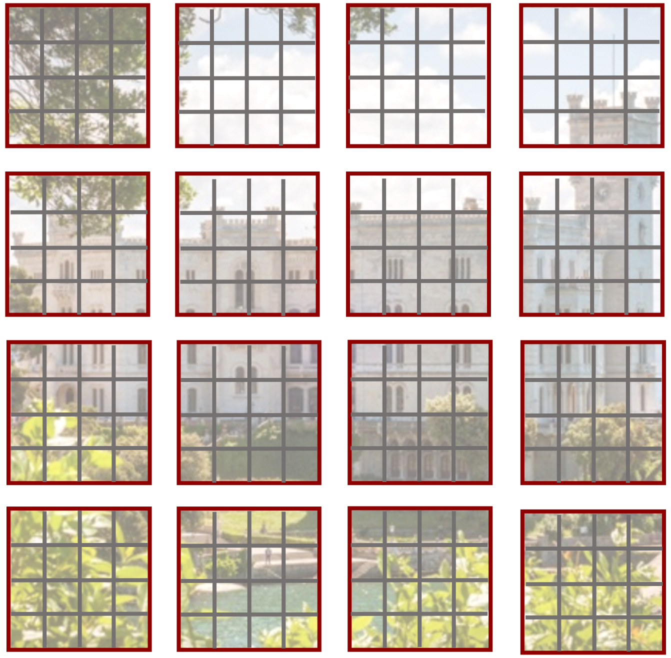
- Proves beneficial in modeling the high correlation in visual signals (Yann LeCun)
- Linear complexity with increasing image resolution: from $O(n^2)$ to $O(n)$



ViT: 256^2=65536 (Global)

16x less computation

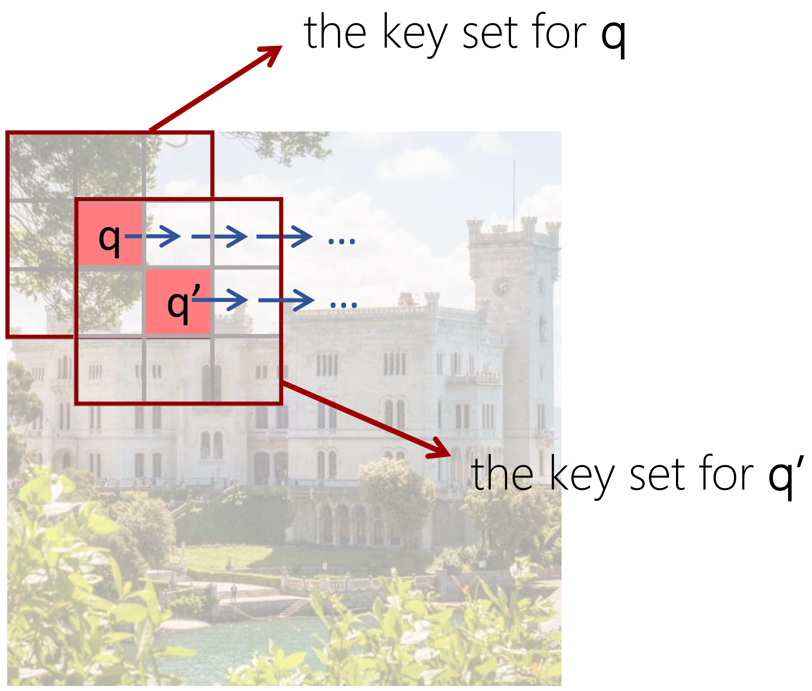
→



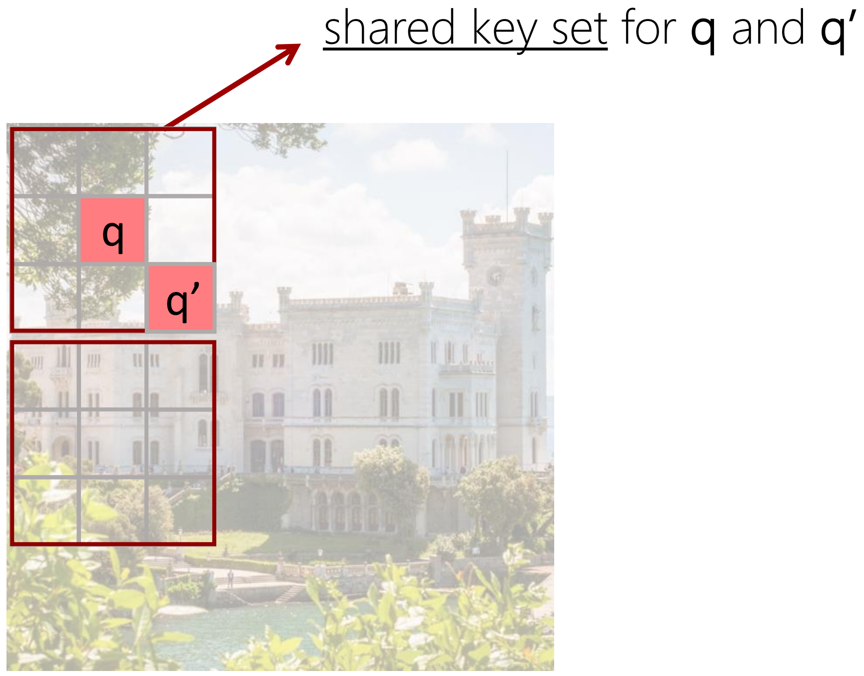
Swin Transformer: 16×16^2=4096 (Local)

Locality by non-overlapped windows

- Compared to sliding window (LR-Net)
  - Shared key set enables friendly memory access and is thus good for speed (larger than 3x)



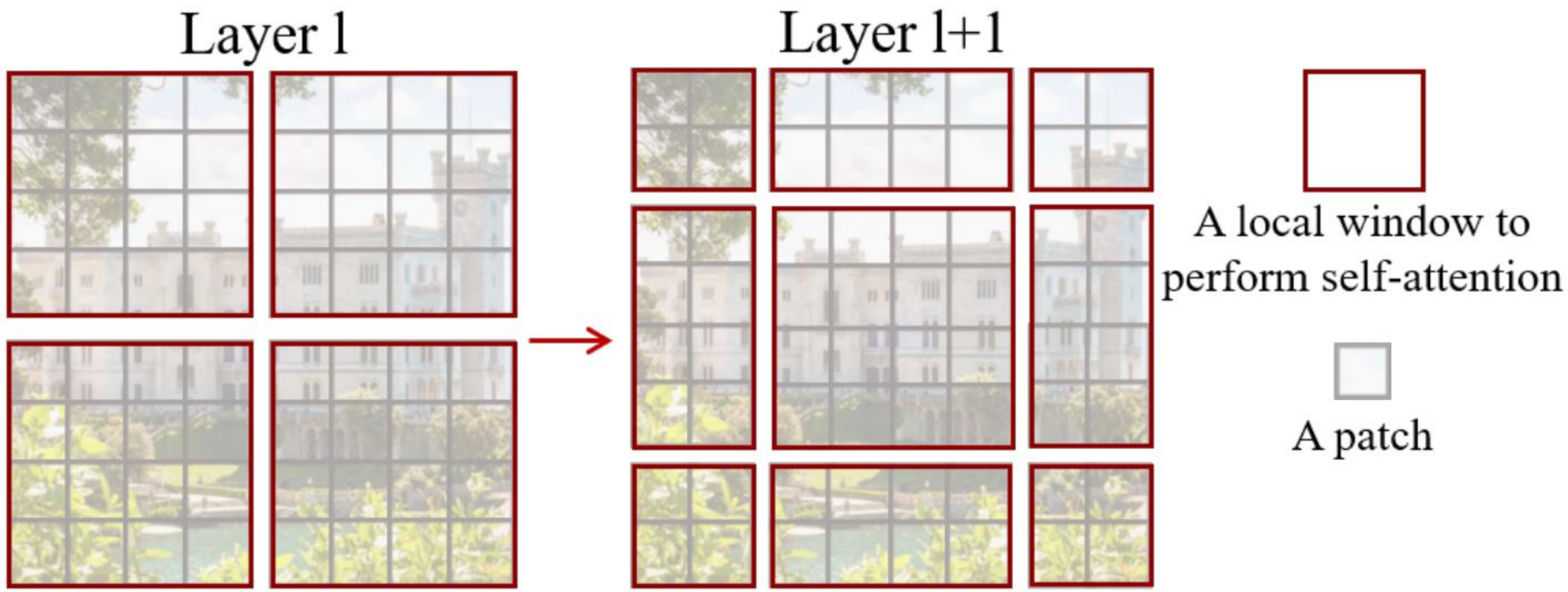
sliding window (LR-Net)



Non-overlapped window (Swin Transformer)

Shifted non-overlapped windows

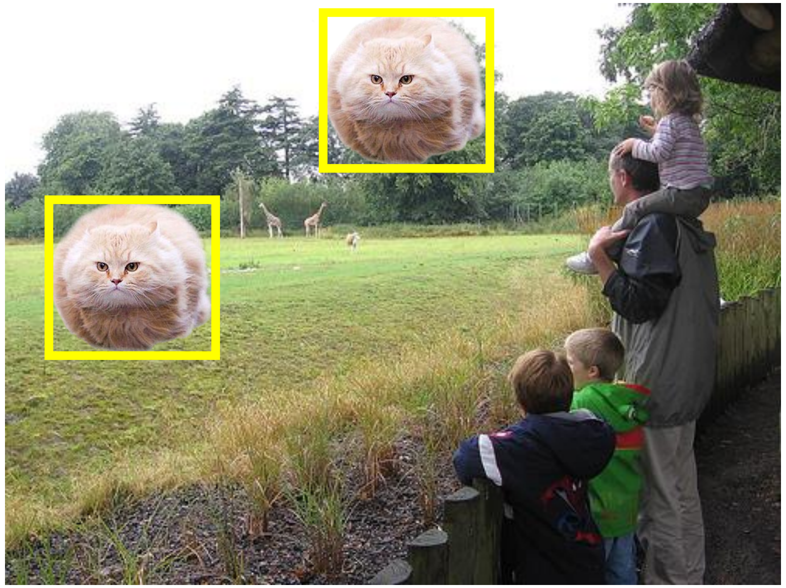
- Enable cross-window connection
  - Non-overlapped windows will result in no connection between windows
  - Performs as effective or even slightly better than the sliding window approach, due to regularization effects

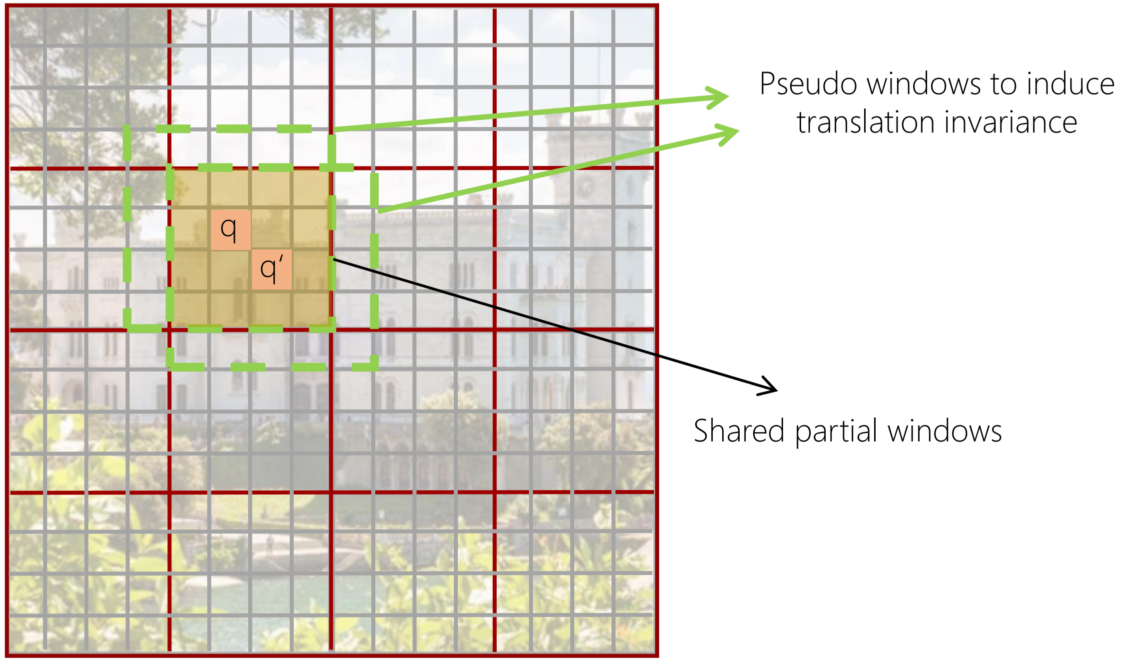


Translational semi-invariance

- Relative position bias plays a more important role in vision than in NLP

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + \boxed{B})V,$$

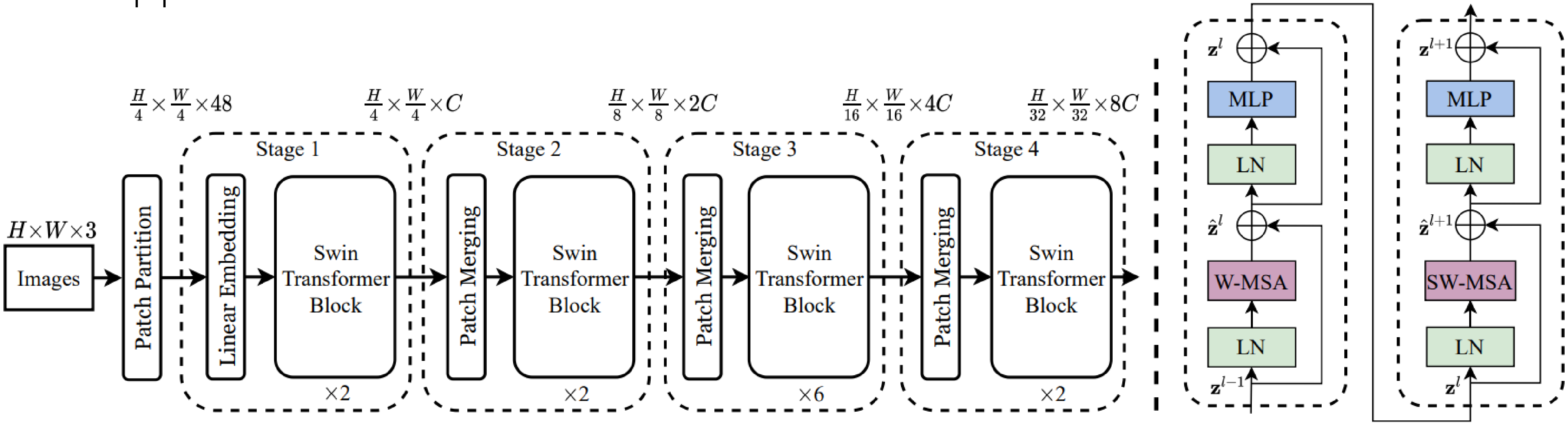




semi-invariance is as effective as full-invariance in our experiments

Architecture instantiations

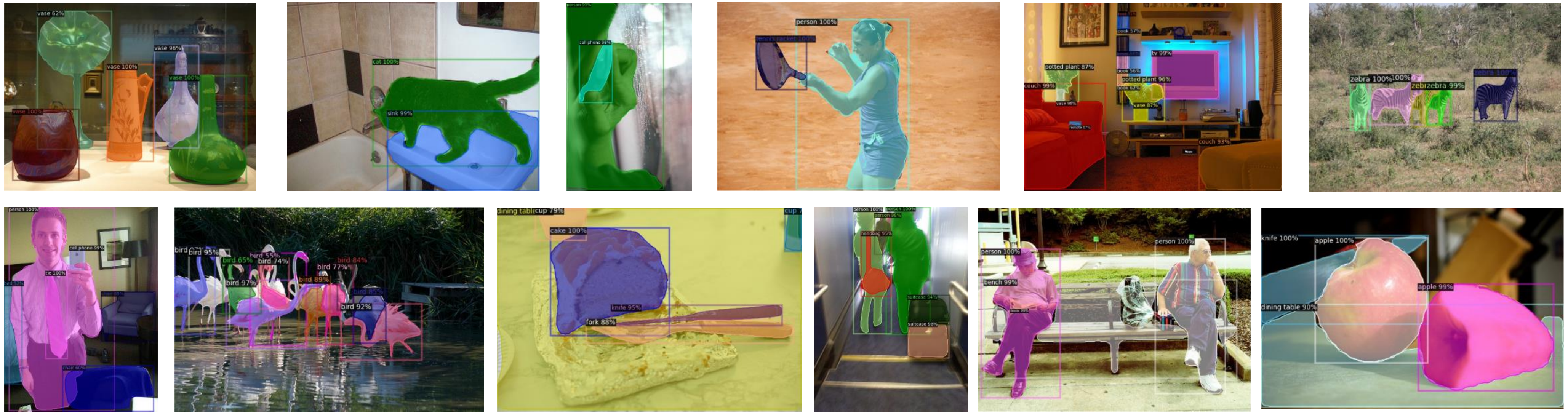
- Resolution of each stage is set similar as ResNet, to facilitate application to down-stream tasks



(a) Architecture

(b) Two Successive Swin Transformer Blocks

Application: object detection



- COCO object detection: #1 for single model (61.3 mAP)
  - Significantly surpass all previous CNN models (+5.3 mAP vs. Google's EfficientDet at CVPR21)
- COCO instance segmentation: #1 for single model (53.0 mAP)
  - Significantly surpass all previous CNN models (+3.9 mAP vs. Google's EfficientDet at CVPR21)

Application: object detection

- Performs consistently better than CNN on various object detectors and various model sizes (+3~4.5 mAP)

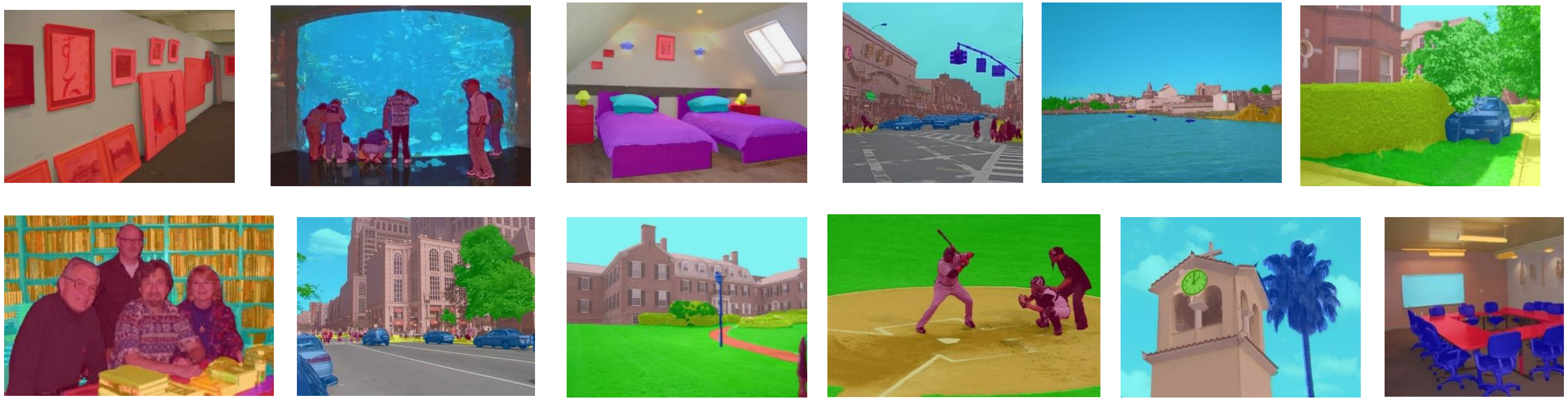
(a) Various frameworks

Method	Backbone	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	#param.	FLOPs	FPS	
Cascade	R-50	46.3	64.3	50.5	82M	739G	18.0	
Mask R-CNN	Swin-T	**50.5**	**69.3**	**54.9**	86M	745G	15.3	+4.2
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	+3.7
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	+3.5
Sparse	R-50	44.5	63.4	48.2	106M	166G	21.0	
R-CNN	Swin-T	**47.9**	**67.3**	**52.3**	110M	172G	18.4	+3.4

(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	+4.2
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	+3.7
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	+3.6

Application: semantic segmentation



- ADE20K semantic segmentation: #1 for single model (57.0 mIoU) with Swin-H
  - The largest and most difficult semantic segmentation benchmark
    - 20,000 training images, 150 categories
  - Significantly surpass all previous models (+8.6 mIoU vs. the previous best CNN model)

3rd-party application: medical image segmentation

Table 1. Segmentation accuracy of different methods on the Synapse multi-organ CT dataset.

Methods	DSC↑	HD↓	Aorta	Gallbladder	Kidney(L)	Kidney(R)	Liver	Pancreas	Spleen	Stomach
V-Net [35]	68.81	-	75.34	51.87	77.10	**80.75**	87.84	40.05	80.56	56.98
DARR [36]	69.77	-	74.74	53.77	72.31	73.24	94.08	54.18	89.90	45.96
R50 U-Net [2]	74.68	36.87	87.74	63.66	80.60	78.19	93.74	56.90	85.87	74.16
U-Net [3]	76.85	39.70	89.07	**69.72**	77.77	68.60	93.43	53.98	86.67	75.58
R50 Att-UNet [2]	75.57	36.97	55.92	63.91	79.20	72.71	93.56	49.37	87.19	74.95
Att-UNet [37]	77.77	36.02	**89.55**	68.88	77.98	71.11	93.57	**58.04**	87.30	75.75
R50 ViT [2]	71.29	32.87	73.73	55.13	75.80	72.20	91.51	45.99	81.99	73.95
TransUnet [2]	77.48	31.69	87.23	63.13	81.87	77.02	94.08	55.86	85.08	75.62
SwinUnet	**79.13**	**21.55**	85.47	66.53	**83.28**	79.61	**94.29**	56.58	**90.66**	**76.60**

Table 2. Segmentation accuracy of different methods on the ACDC dataset.

Methods	DSC	RV	Myo	LV
R50 U-Net	87.55	87.10	80.63	94.92
R50 Att-UNet	86.75	87.58	79.20	93.47
R50 ViT	87.57	86.07	81.88	94.75
TransUnet	89.71	88.86	84.53	95.73
SwinUnet	**90.00**	88.55	**85.62**	**95.83**

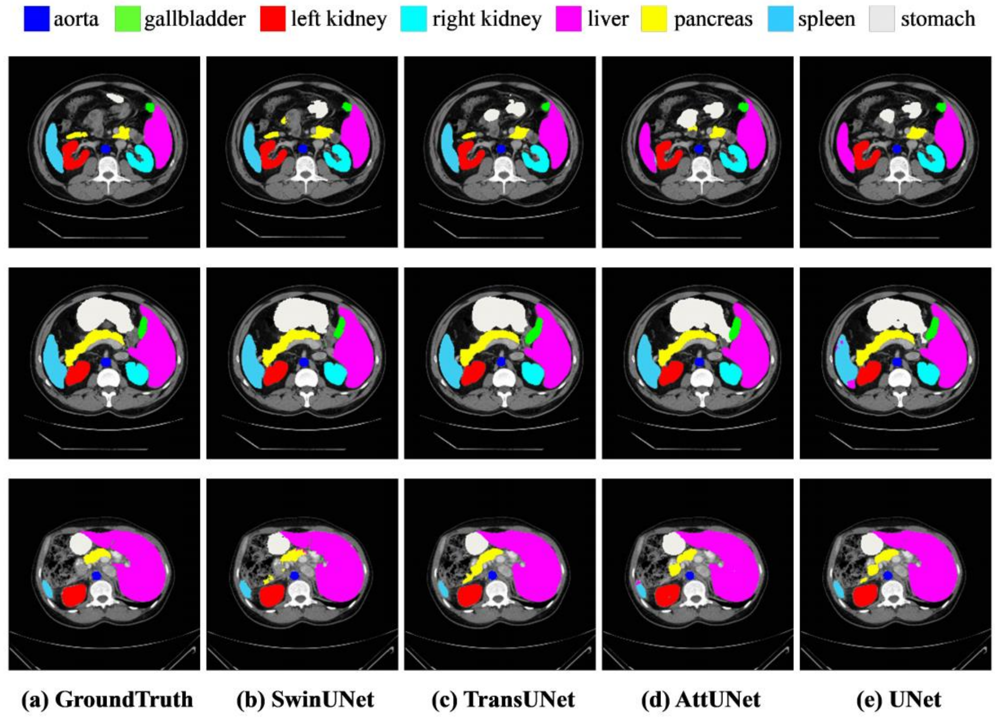


Fig. 3. The segmentation results of different methods on the Synapse multi-organ CT dataset.

https://arxiv.org/abs/2105.05537

3rd-party application: Image Restoration (SwinIR)

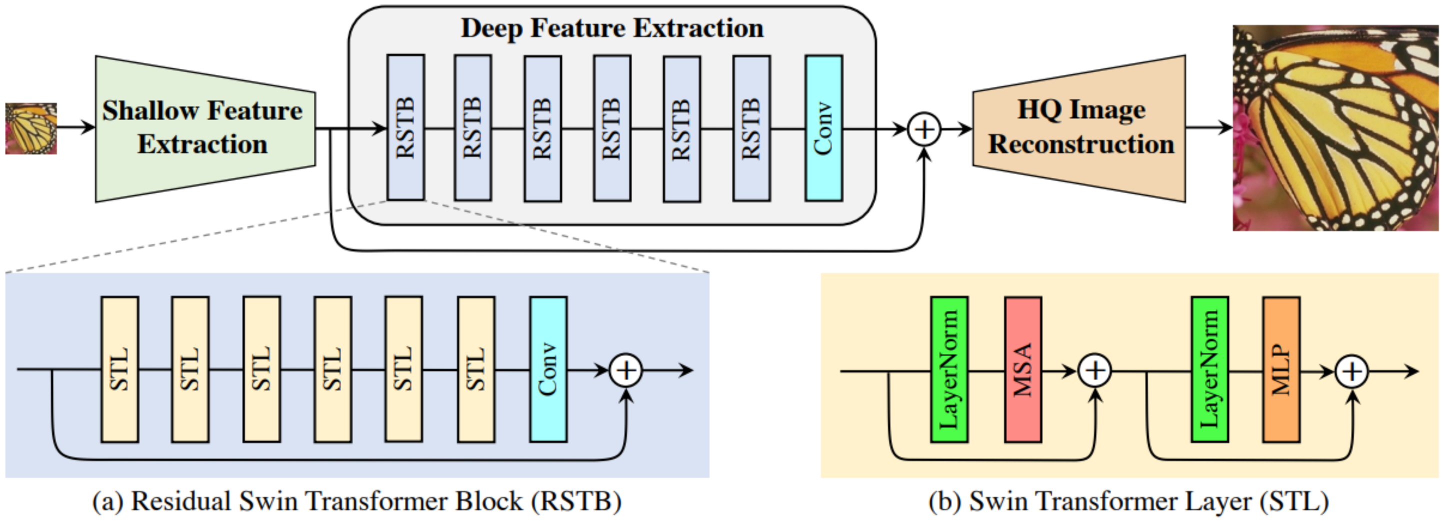


Figure 2: The architecture of the proposed SwinIR for image restoration.

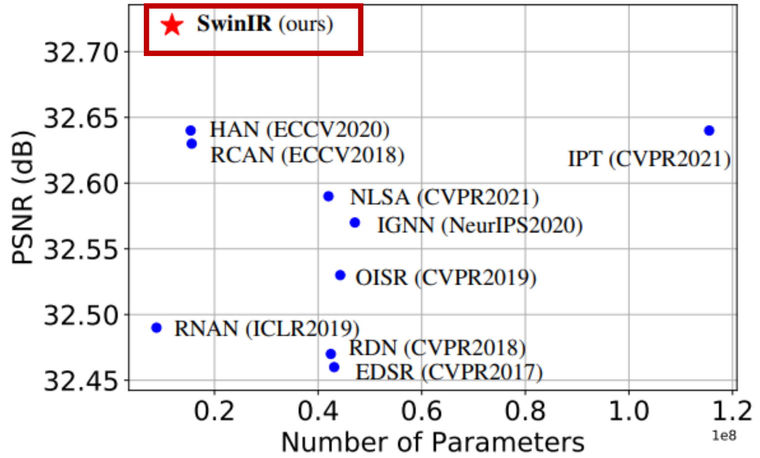
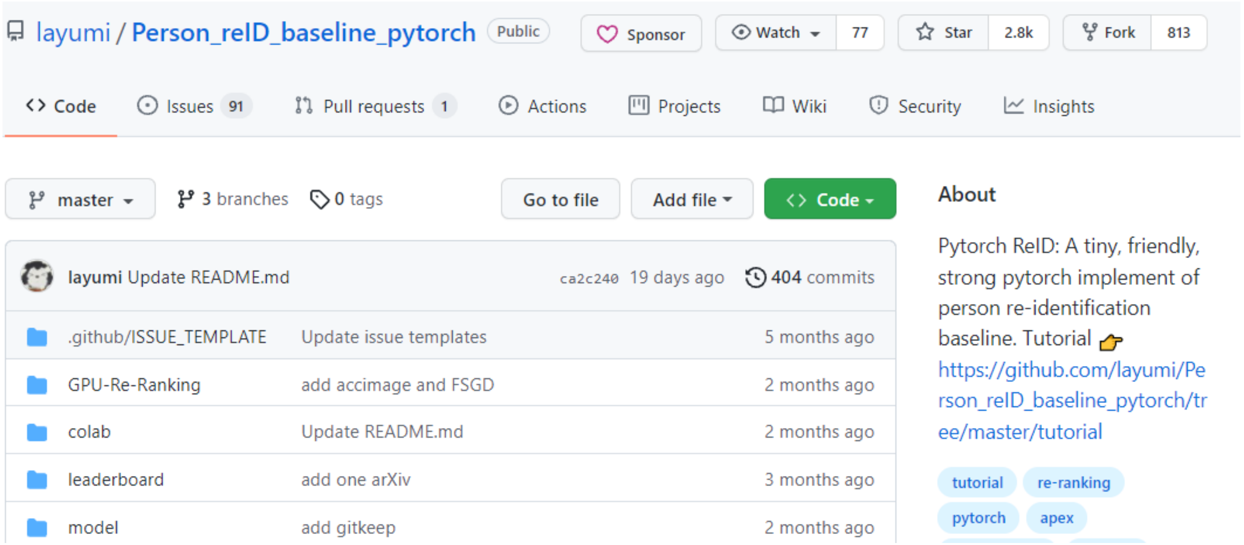


Figure 1: PSNR results v.s the total number of parameters of different methods for image SR (×4) on Set5 [3].

Jingyun Liang, Jiezhang Cao et al. SwinIR: Image Restoration Using Swin Transformer, Arxiv 2021

3rd-party application: Person Re-ID

Methods	Rank@1	mAP
ResNet-50	$\boxed{88.84\%}$	71.59%
ResNet-50 (all tricks+Circle)	92.13%	79.84%
ResNet-50 (all tricks+Circle+DG)	92.13%	80.13%
Swin	$\boxed{92.73\%}$	79.71%
Swin (all tricks+Circle)	93.65%	83.65%
Swin (all tricks+Circle+b16)	93.91%	85.17%
Swin (all tricks+Circle+b16+DG)	94.00%	85.36%



https://github.com/layumi/Person_reID_baseline_pytorch

Application: self-supervised learning (MoBY)

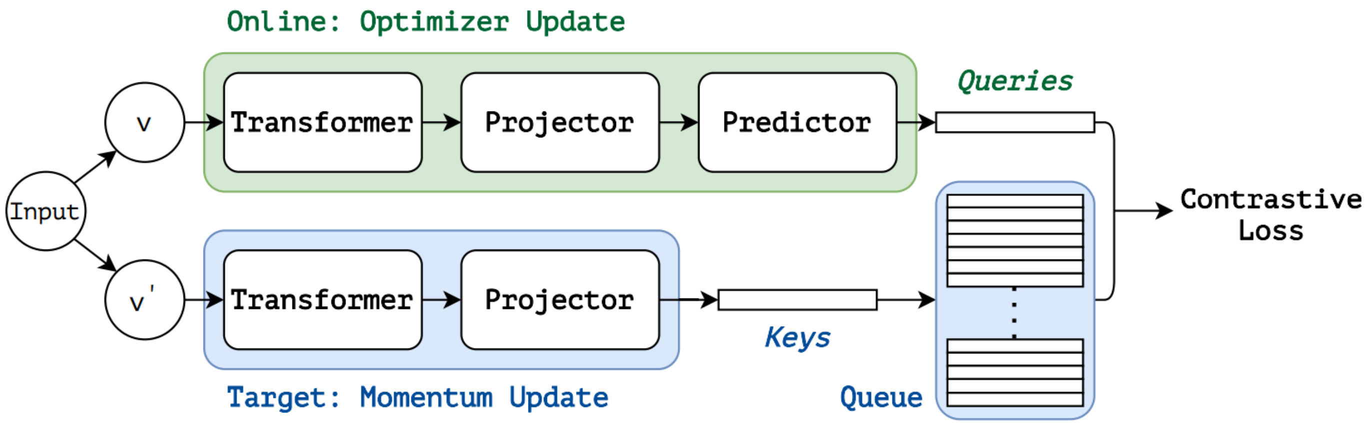


Figure 1: The pipeline of MoBY.

Method	Arch.	Epochs	Params (M)	FLOPs (G)	img/s	Top-1 acc (%)
Sup.	DeiT-S	300	22	4.6	940.4	79.8
Sup.	Swin-T	300	29	4.5	755.2	81.3
MoCo v3	DeiT-S	300	22	4.6	940.4	72.5
DINO	DeiT-S	300	22	4.6	940.4	72.5
DINO†	DeiT-S	300	22	4.6	940.4	75.9
MoBY	DeiT-S	300	22	4.6	940.4	72.8
MoBY	Swin-T	100	29	4.5	755.2	70.9
MoBY	Swin-T	300	29	4.5	755.2	**75.0**

Table 1: Comparison of different SSL methods and different Transformer architectures on ImageNet-1K linear evaluation. † denotes DINO with a multi-crop scheme in training.

Zhenda Xie, Yutong Lin et al. *Self-supervised Learning with Swin Transformers*, Arxiv 2021

Application: self-supervised learning (EsViT)

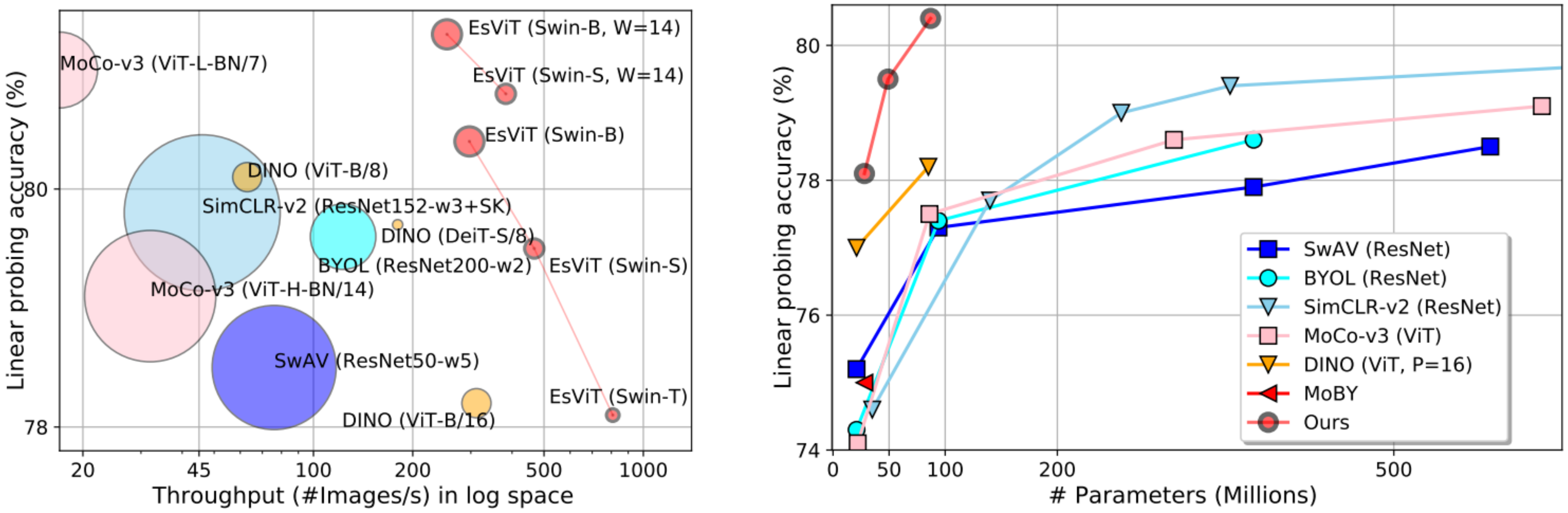


Figure 1: Efficiency vs accuracy comparison under the linear classification protocol on ImageNet. Left: Throughput of all SoTA SSL vision systems, circle sizes indicates model parameter counts; Right: performance over varied parameter counts for models with moderate (throughout/#parameters) ratio. Please refer Section 4.1 for details.

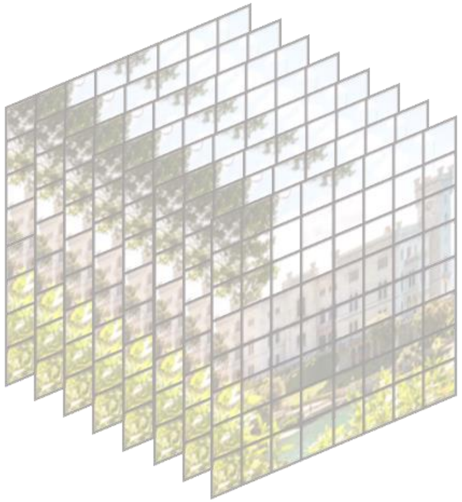
Chunyuan Li, Jianwei Yang et al. *Efficient Self-supervised Vision Transformers for Representation Learning*, Arxiv 2021

Application: video recognition

			
	An image		A video
# pixels	224x224	$\xrightarrow{8\ x}$	224x224x8
# tokens	14x14	$\xrightarrow{8\ x}$	14x14x8
Complexity (global att)	(14x14)^2	$\xrightarrow{64\ x}$	(14x14x8)^2

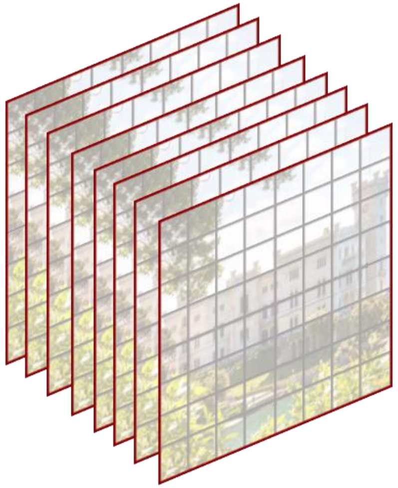
Too expensive!

How to approximate Global Self-Attention?



input: $T \times H \times W = 8 \times 8 \times 8$

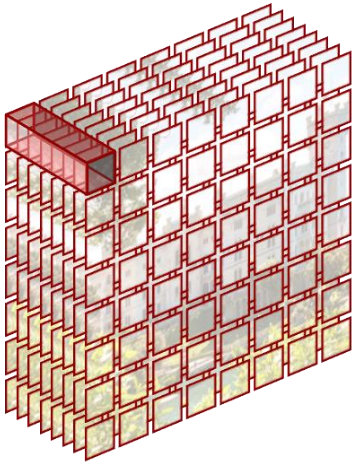
Global Attention: $(T \times H \times W)^2$



Spatial-only Attention

$T \times (H \times W)^2$

+



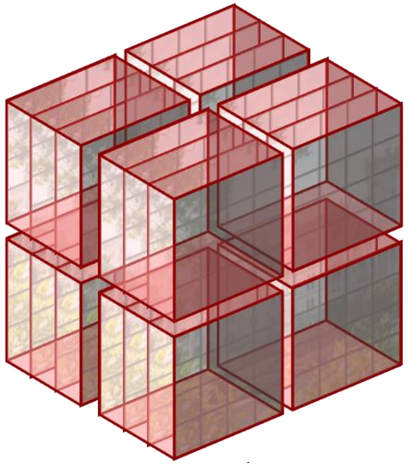
Temporal-only Attention

$(T)^2 \times H \times W$

= ViViT (Google)

Timesformer (FAIR)

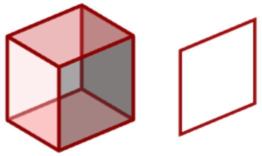
......



Spatial-Temporal Local Attention

$T \times H \times W \times Kt \times Ks \times Ks$

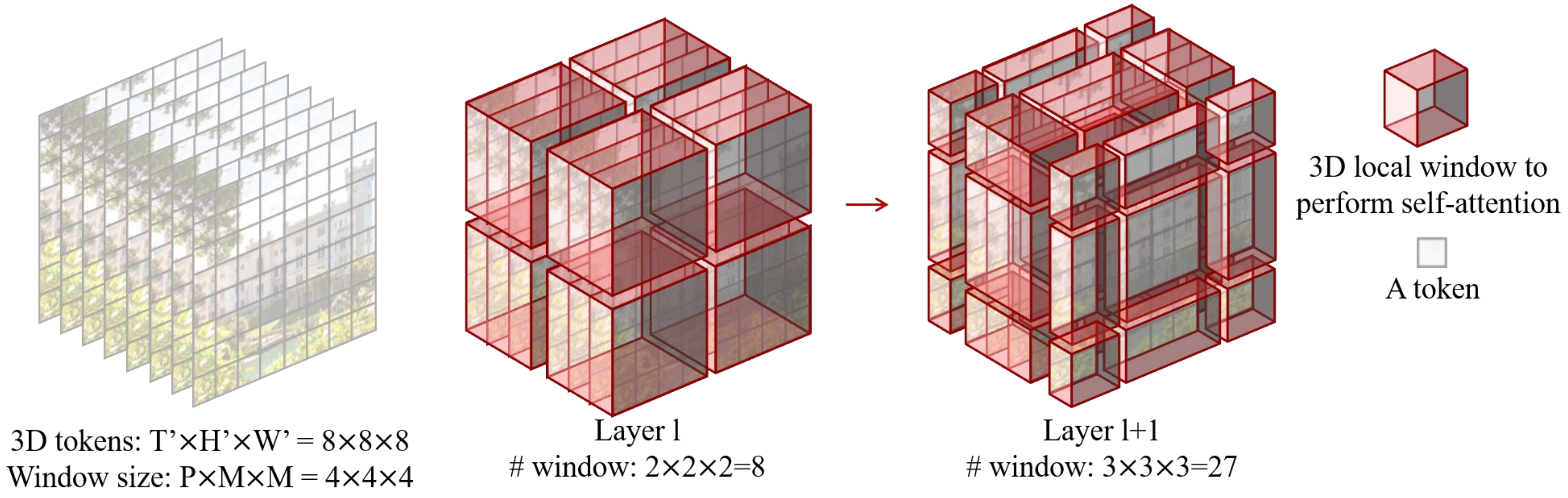
= Video Swin Transformer



2D/3D window to perform self-attention

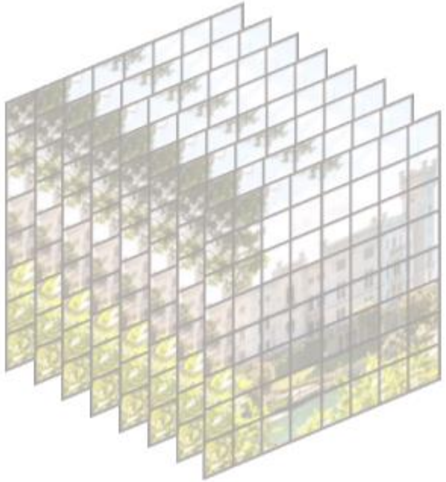
Video Swin Transformer

- 2D Locality -> 3D Locality
- Keep Hierarchy & Translation Semi-invariance



Ze Liu, Jia Ning, Yue Cao et al. Video Swin Transformer, Arxiv 2021

Video Swin Transformer



3D tokens: T'×H'×W' = 8×8×8
Window size: P×M×M = 4×4×4

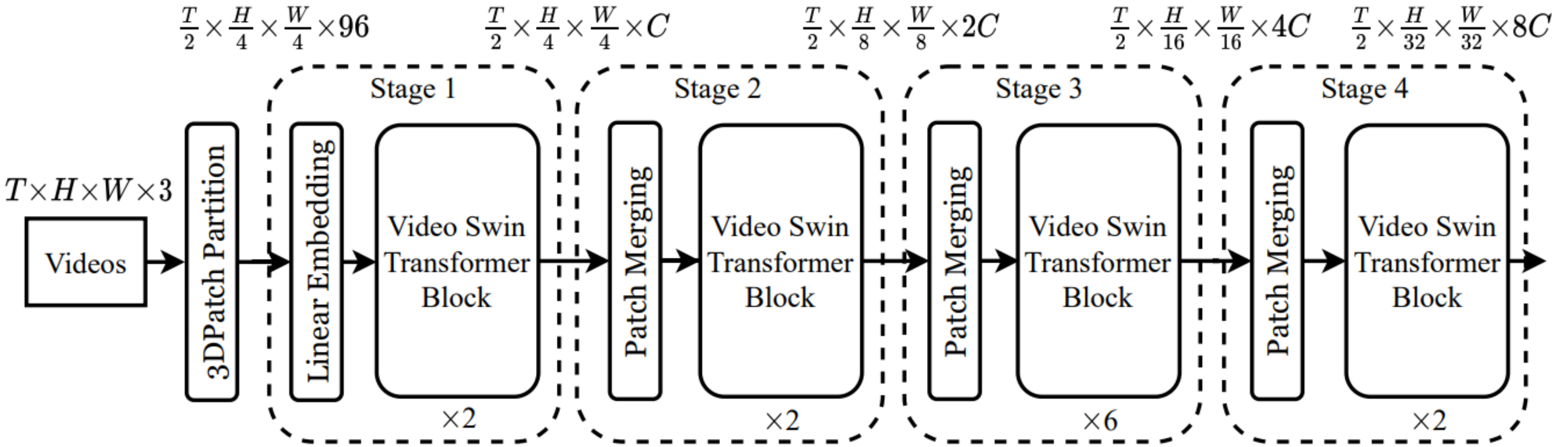


Figure 2: Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

Experiments

- Swin Transformer achieves **SOTA** on major video benchmarks with 20x less pre-training data and 3x smaller model size

Table 1: Comparison to state-of-the-art on Kinetics-400. "384↑" signifies that the model uses a larger spatial resolution of 384×384. "Views" indicates # temporal clip × # spatial crop. The magnitudes are Giga (10^9) and Mega (10^6) for FLOPs and Param respectively.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
R(2+1)D [37]	-	72.0	90.0	10 × 1	75	61.8
I3D [6]	ImageNet-1K	72.1	90.3	-	108	25.0
NL I3D-101 [40]	ImageNet-1K	77.7	93.3	10 × 3	359	61.8
ip-CSN-152 [36]	-	77.8	92.8	10 × 3	109	32.8
CorrNet-101 [39]	-	79.2	-	10 × 3	224	-
SlowFast R101+NL [13]	-	79.8	93.9	10 × 3	234	59.9
X3D-XXL [12]	-	80.4	94.6	10 × 3	144	20.3
MViT-B, 32×3 [10]	-	80.2	94.4	1 × 5	170	36.6
MViT-B, 64×3 [10]	-	81.2	95.1	3 × 3	455	36.6
TimeSformer-L [3]	ImageNet-21K	80.7	94.7	1 × 3	2380	121.4
ViT-B-VTN [29]	ImageNet-21K	78.6	93.7	1 × 1	4218	11.04
ViViT-L/16x2 [1]	ImageNet-21K	80.6	94.7	4 × 3	1446	310.8
ViViT-L/16x2 320 [1]	ImageNet-21K	81.3	94.7	4 × 3	3992	310.8
ip-CSN-152 [36]	IG-65M	82.5	95.3	10 × 3	109	32.8
ViViT-L/16x2 [1]	JFT-300M	82.8	95.5	4 × 3	1446	310.8
ViViT-L/16x2 320 [1]	JFT-300M	83.5	95.5	4 × 3	3992	310.8
ViViT-H/16x2 [1]	JFT-300M	84.8	95.8	4 × 3	8316	647.5
Swin-T	ImageNet-1K	78.8	93.6	4 × 3	88	28.2
Swin-S	ImageNet-1K	80.6	94.5	4 × 3	166	49.8
Swin-B	ImageNet-1K	80.6	94.6	4 × 3	282	88.1
Swin-B	ImageNet-21K	82.7	95.5	4 × 3	282	88.1
Swin-L	ImageNet-21K	83.1	95.9	4 × 3	604	197.0
Swin-L (384↑)	ImageNet-21K	**84.9**	**96.6**	10 × 5	2107	200.0

+3.6% using the same pre-training data

Table 2: Comparison to state-of-the-art on Kinetics-600.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
SlowFast R101+NL [13]	-	81.8	95.1	10 × 3	234	59.9
X3D-XL [12]	-	81.9	95.5	10 × 3	48	11.0
MViT-B-24, 32×3 [9]	-	83.8	96.3	5 × 1	236	52.9
TimeSformer-HR [3]	ImageNet-21K	82.4	96	1 × 3	1703	121.4
ViViT-L/16x2 320 [1]	ImageNet-21K	83.0	95.7	4 × 3	3992	310.8
ViViT-H/16x2 [9]	JFT-300M	85.8	96.5	4 × 3	8316	647.5
Swin-B	ImageNet-21K	83.8	96.4	4 × 3	282	88.1
Swin-L (384↑)	ImageNet-21K	**85.9**	**97.1**	4 × 3	2107	200.0

+2.9% using the same pre-training data

Table 3: Comparison to state-of-the-art on Something-Something v2.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
TimeSformer-HR [3]	ImageNet-21K	62.5	-	1 × 3	1703	121.4
SlowFast R101, 8×8 [13]	Kinetics-400	63.1	87.6	1 × 3	106	53.3
TSM-RGB [27]	Kinetics-400	63.3	88.2	2 × 3	62	42.9
MSNet [23]	ImageNet-21K	64.7	89.4	1 × 1	67	24.6
TEA [26]	ImageNet-21K	65.1	89.9	10 × 3	70	-
bIVNet [11]	SSv2	65.2	90.3	1 × 1	129	40.2
ViViT-L/16x2 [1]	-	65.4	89.8	-	903	352.1
MViT-B, 64×3 [10]	Kinetics-400	67.7	90.9	1 × 3	455	36.6
MViT-B-24, 32×3 [10]	Kinetics-600	68.7	91.5	1 × 3	236	53.2
Swin-B	Kinetics-400	**69.6**	**92.7**	1 × 3	321	88.8

+1.9% using the same pre-training data

Resources

Swin Transformer

- State of the Art | Object Detection on COCO test-dev (using additional training data)
- State of the Art | Instance Segmentation on COCO test-dev (using additional training data)
- State of the Art | Object Detection on COCO minival (using additional training data)
- State of the Art | Instance Segmentation on COCO minival (using additional training data)
- Ranked #8 | Semantic Segmentation on ADE20K (using additional training data)
- Ranked #9 | Semantic Segmentation on ADE20K val
- State of the Art | Action Recognition on Something-Something V2 (using additional training data)
- State of the Art | Action Classification on Kinetics-400 (using additional training data)
- State of the Art | Action Classification on Kinetics-600 (using additional training data)

By Ze Liu*, Yutong Lin*, Yue Cao*, Han Hu*, Yixuan Wei, Zheng Zhang, Stephen Lin and Baining Guo.

This repo is the official implementation of "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows".

It currently includes code and models for the following tasks:

> **Image Classification**: Included in this repo. See get_started.md for a quick start.
>
> **Object Detection and Instance Segmentation**: See Swin Transformer for Object Detection.
>
> **Semantic Segmentation**: See Swin Transformer for Semantic Segmentation.
>
> **Self-Supervised Learning**: See Transformer-SSL.
>
> **Video Action Recognition**: See Video Swin Transformer.
>
> **Semi-Supervised Object Detection**: See Soft Teacher.

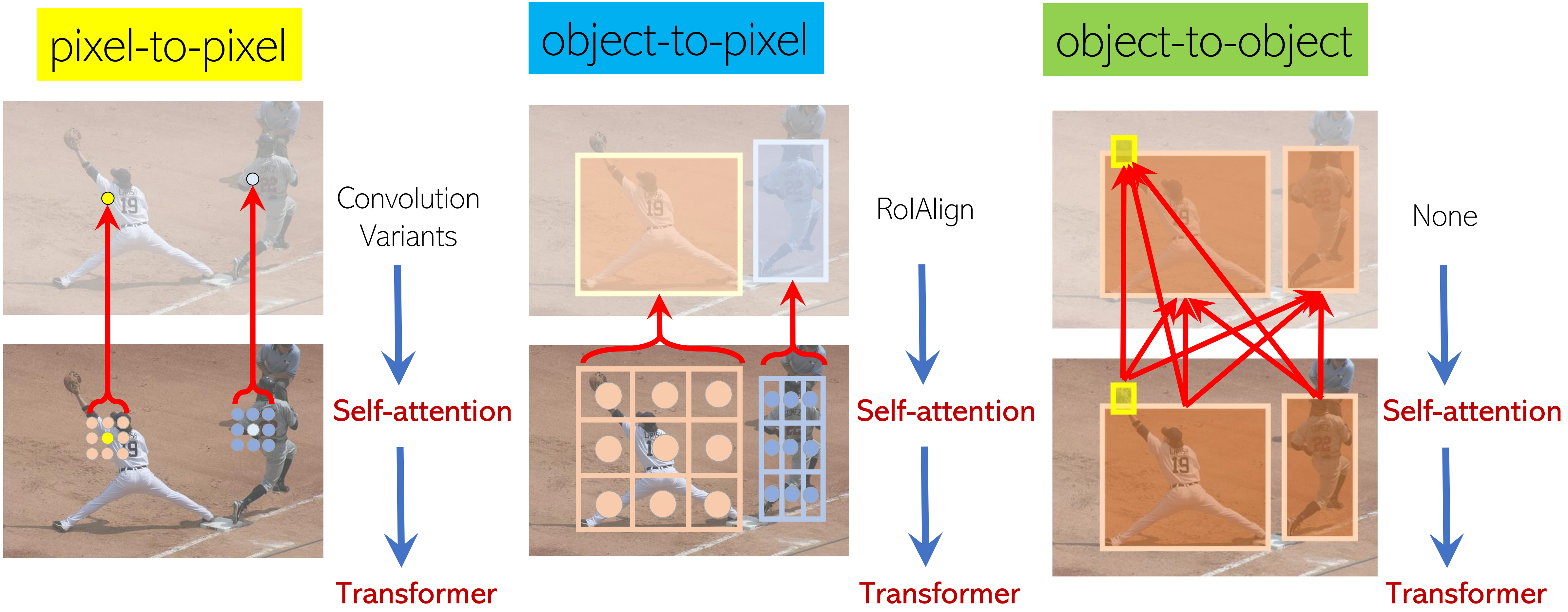
Already Got 4.6k stars and >200 citations!
ICCV 2021 Oral Presentation

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

Paper

Code

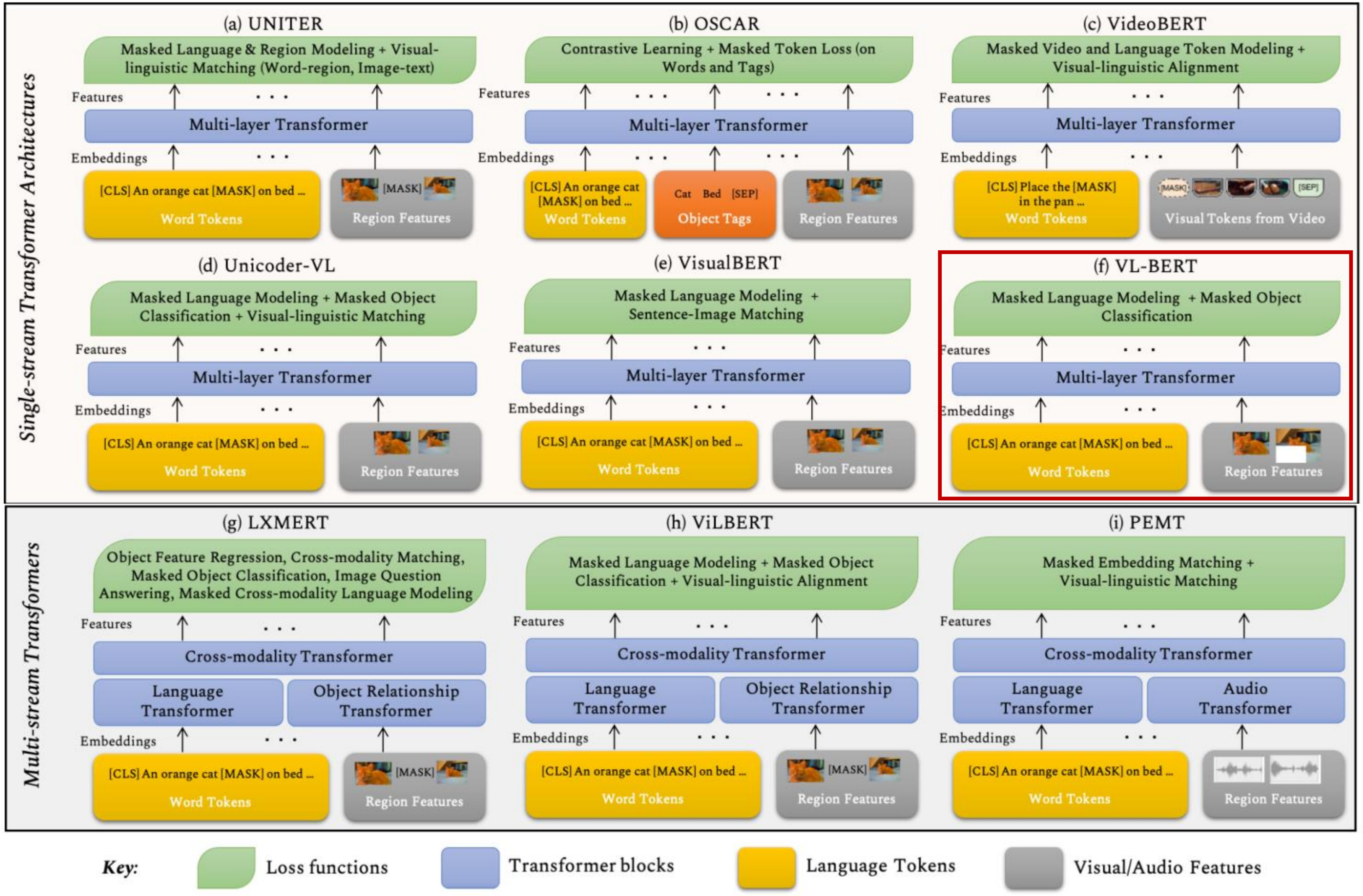
Relationship Modeling of Basic Visual Elements



Transformers: General Relationship Modeling

- Visual elements
  - Pixel-to-pixel
  - Pixel-to-object
  - Object-to-object
- Multi-modality
  - Visual token (patches/objects/pixels) & language token (words)
- Point clouds
  - Points in point clouds

Tokens from multiple modalities



Multi-Modality Learning

- **Humans routinely perform tasks which always involve multiple modalities.** Every time you ask someone to imagine a scene, or describe what you're seeing, you're performing a task bridging both linguistic and visual representation.

- But the interaction between language and vision, despite seeing traction as of late, is still **largely unexplored**.

What is visual-linguistic task?

- Visual-linguistic tasks are related to the systems that can demonstrate their **visual understanding by generating or responding to natural language** in the context of images and videos.

Visual-linguistic understanding

Who is wearing glasses?

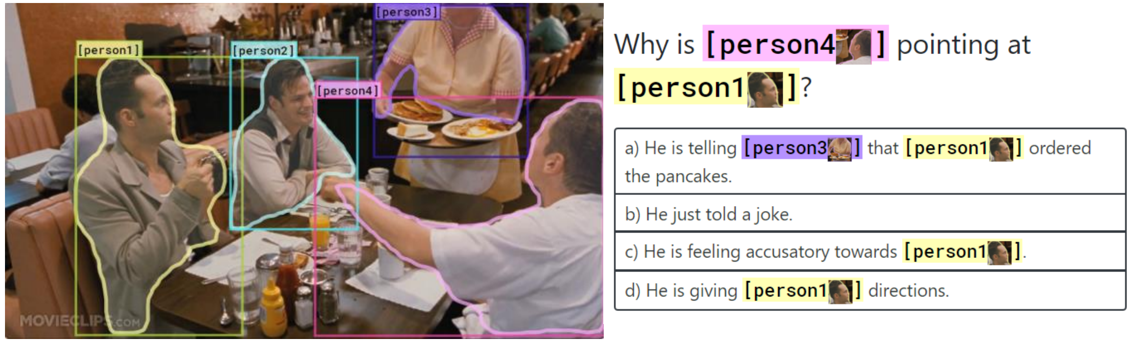
man woman

Visual Question Answering

Visual-linguistic generation

Image Captioning

Visual-linguistic reasoning



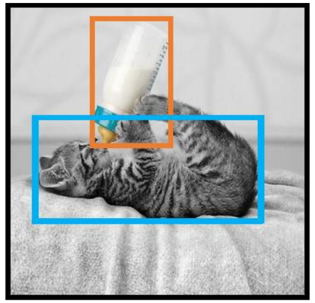
Visual Commonsense Reasoning

Common Need

- Better **align visual and linguistic clues**, which is the **main and common need** for all visual-linguistic tasks

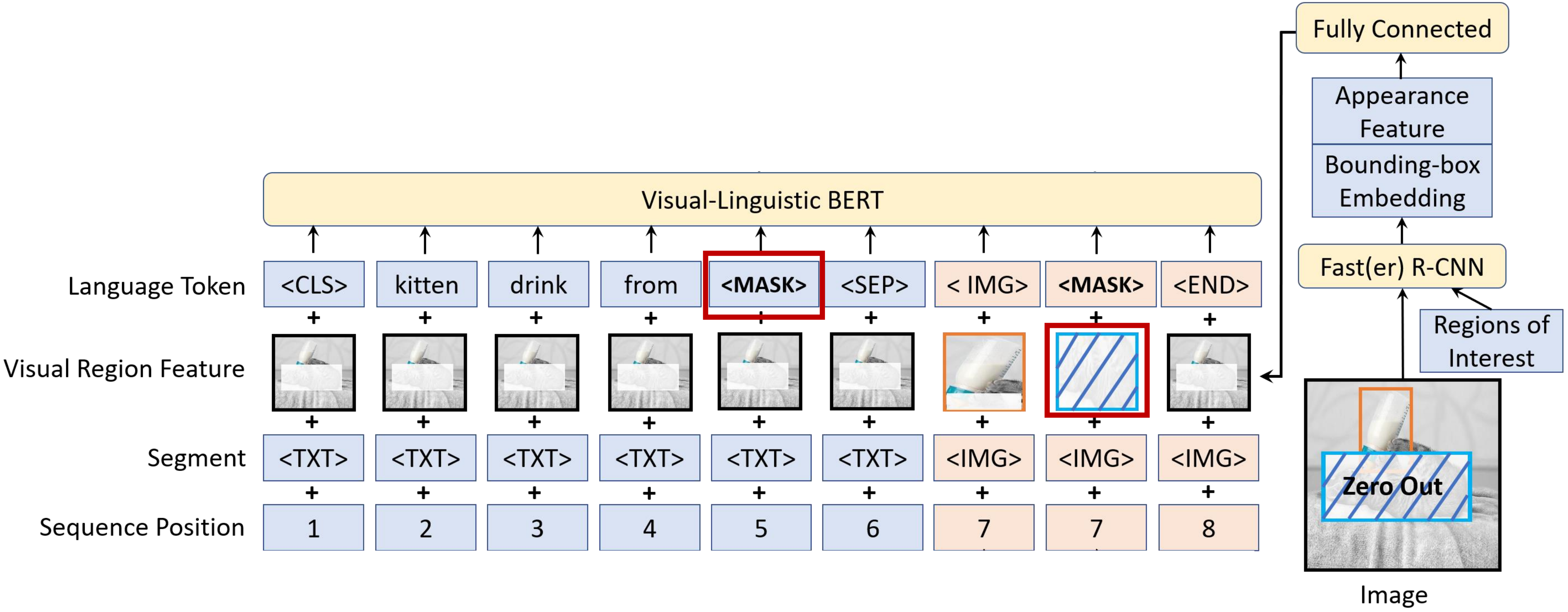
VL-BERT

Language Token | <CLS> | kitten | drink | from | bottle | <SEP>



Image

VL-BERT

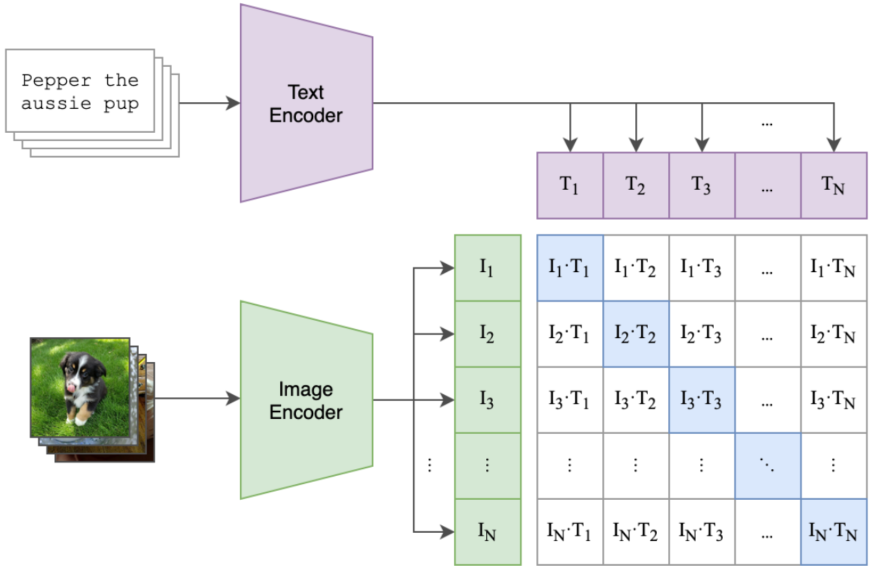


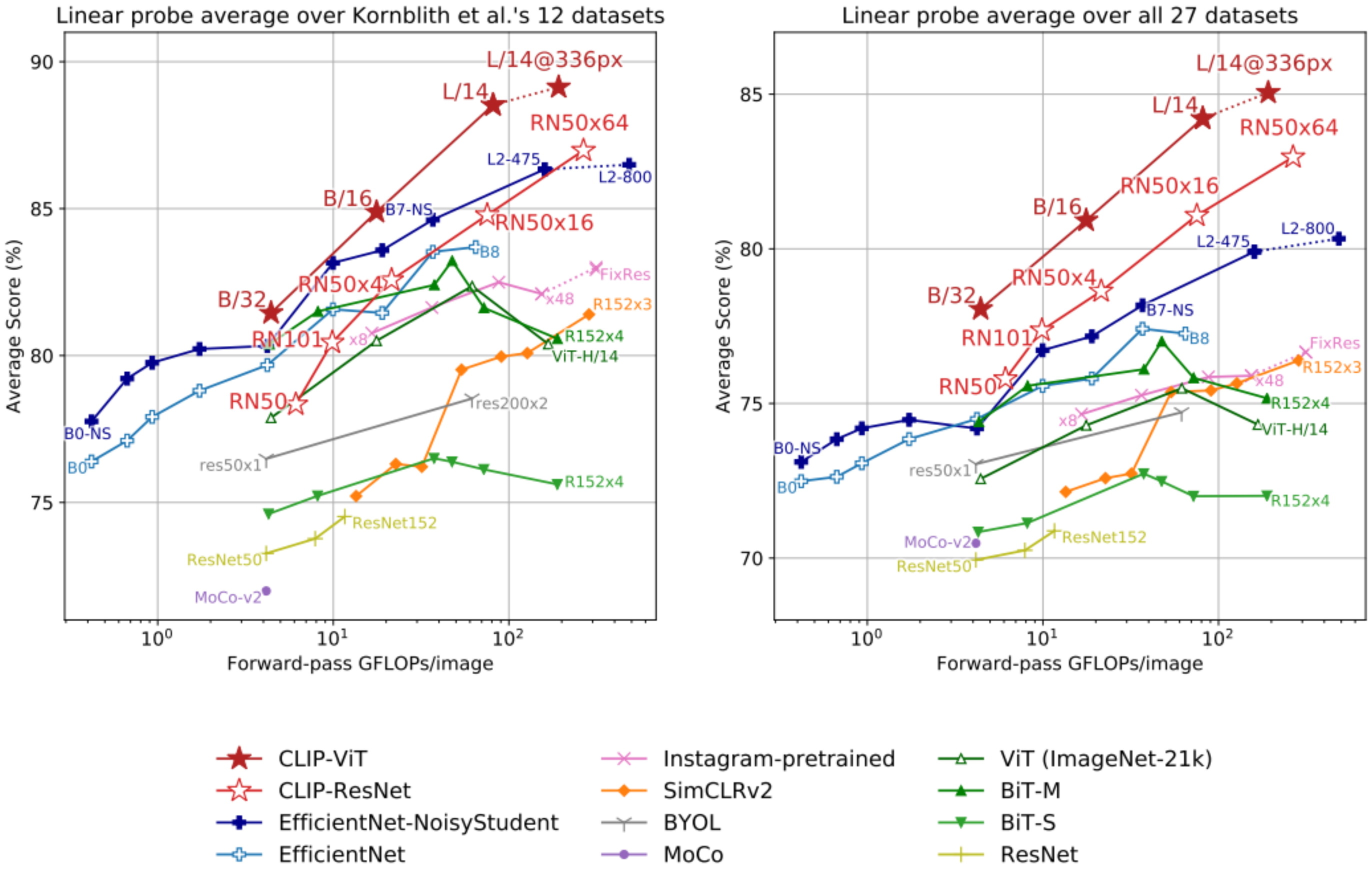
Fully Connected
Appearance Feature
Bounding-box Embedding
Fast(er) R-CNN
Regions of Interest
Zero Out
Image
Visual-Linguistic BERT
Language Token
<CLS> kitten drink from <MASK> <SEP> < IMG> <MASK> <END>
Visual Region Feature
Segment
<TXT> <TXT> <TXT> <TXT> <TXT> <TXT> <IMG> <IMG> <IMG>
Sequence Position
1 2 3 4 5 6 7 7 8

Method	Pre-training Data	Visual Question Answering		Referred Expression Comprehension		Image-Text Retrieval		Visual Commonsense Reasoning		
		Test-dev Acc	Test-std Acc	Labeled Regions Acc	Detected Regions Acc	Text-to-Image Recall@1	Image-to-Text Recall@1	Q -> A Acc	QA -> R Acc	Q -> AR Acc
DFAF (Gao et al., 2018)	-	70.22	70.34	-	-	-	-	-	-	-
MAttNet (Yu et al., 2018)	-	-	-	75.13	71.62	-	-	-	-	-
Concept Graph (Shi et al., 2019)	-	-	-	-	-	61.4	76.6	-	-	-
R2C (Zellers et al., 2019)	-	-	-	-	-	-	-	65.1	67.3	44.0
ViLBERT (Lu et al., 2019)	CC	70.55	70.92	-	78.52	-	-	73.3	74.6	54.8
VisualBERT (Li et al., 2019b)	COCO	70.80	71.00	-	-	-	-	71.6	73.2	52.4
B2T2 (Alberti et al., 2019)	CC	-	-	-	-	-	-	72.6	75.7	55.0
VL-BERT (Su et al., 2020)	CC	71.79	72.22	83.62	78.57	69.1	85.4	75.8	78.4	59.7
UNITER (Chen et al., 2020)	CC+COCO+VG +SBU	73.24	73.40	85.87	81.37	77.5	88.2	77.3	80.8	62.8

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, Jifeng Dai. **VL-BERT: Pre-training of Generic Visual-Linguistic Representations**. ICLR 2020.

Dual-tower Visual-linguistic Model (CLIP)





Alec Radford, Jong Wook Kim et al. *Learning Transferable Visual Modes From Natural Language Supervision*. ICML 2021.

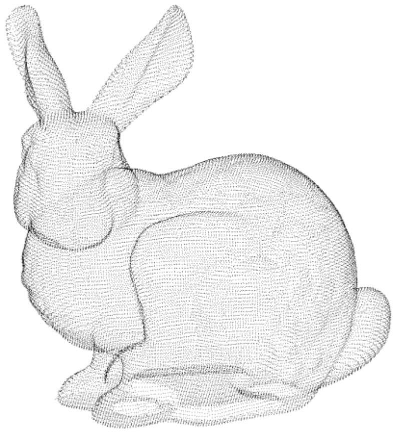
Transformers: General Relationship Modeling

- Visual elements
  - Pixel-to-pixel
  - Pixel-to-object
  - Object-to-object
- Multi-modality
  - Visual token (patches/objects/pixels) & language token (words)
- **Point clouds**
  - **Points** in point clouds

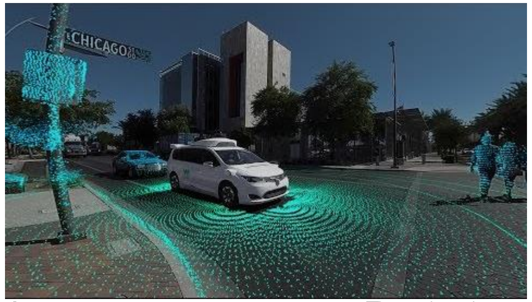
Background

Lidar

Depth Sensor

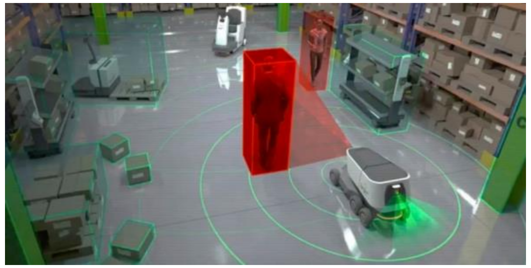


Point Cloud



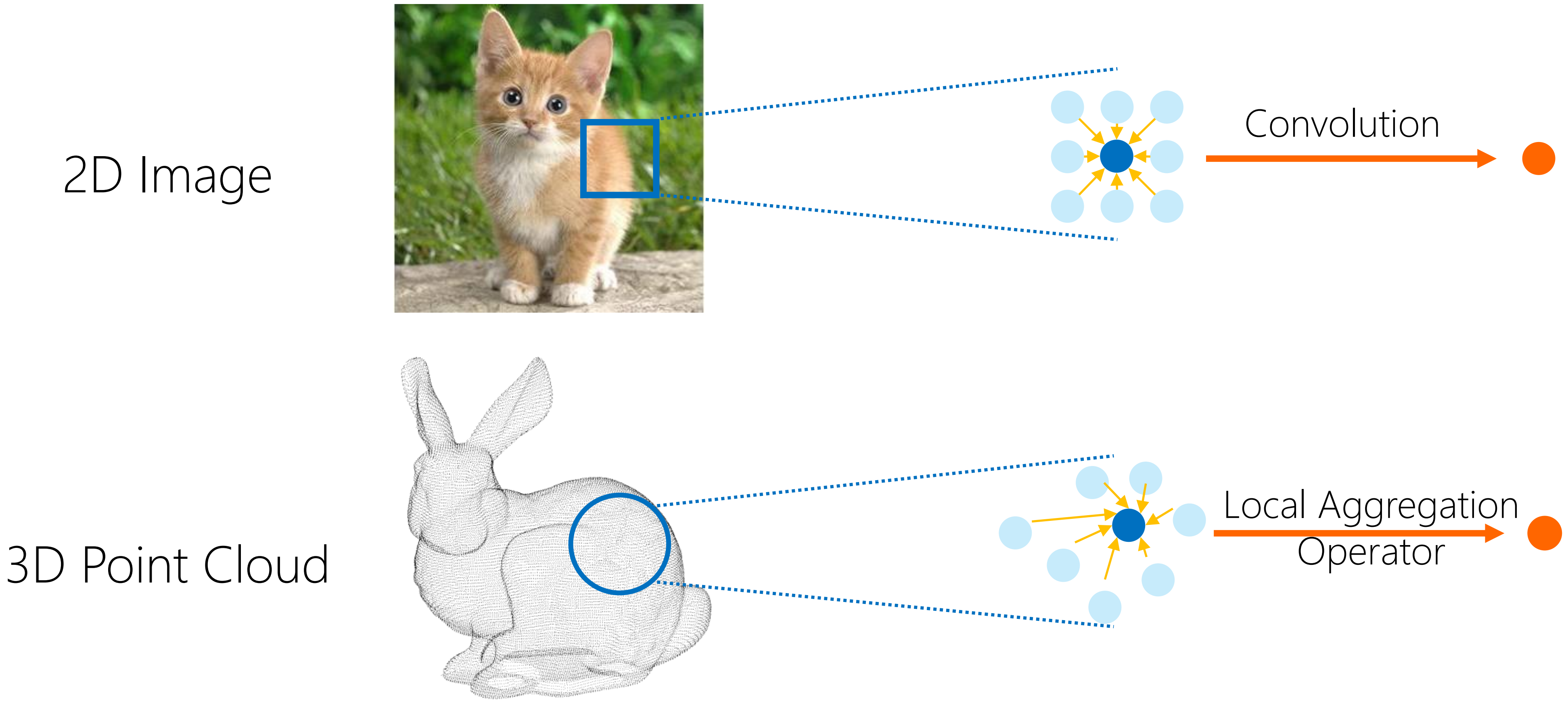
Autonomous Driving

Robot Navigation

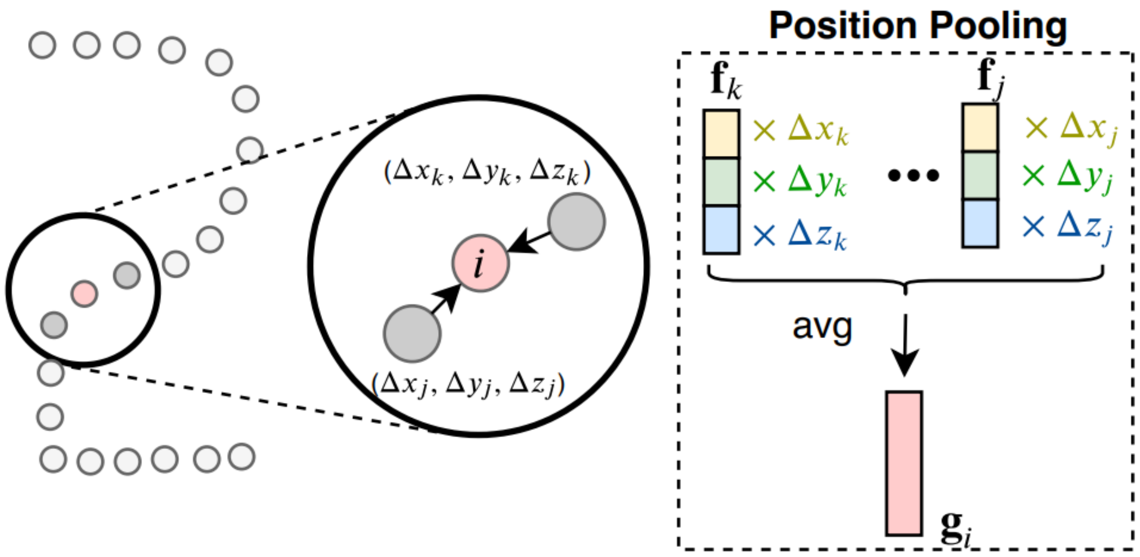


Augmented Reality

Background


2D Image
Convolution
3D Point Cloud
Local Aggregation
Operator

Position Pooling

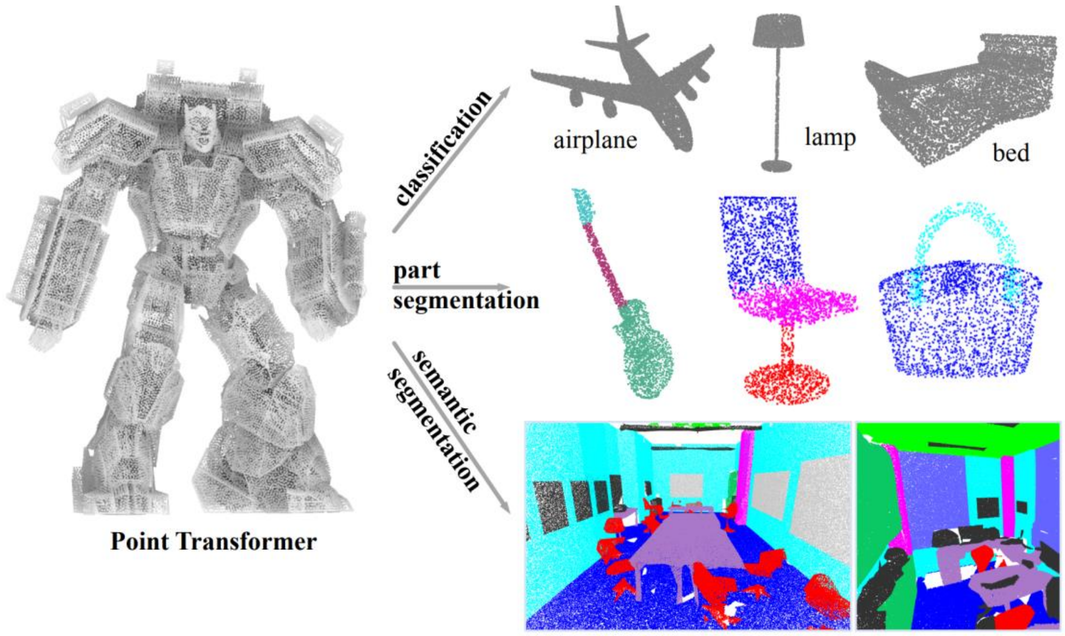


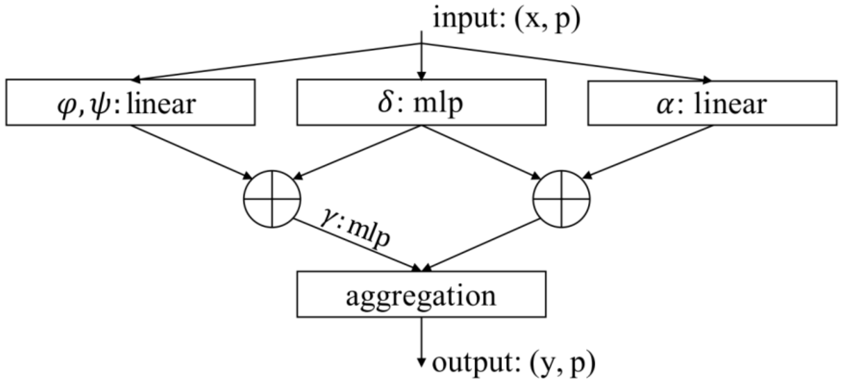
$$G(\Delta p_{ij}, f_j) = Concat[\Delta x_{ij} f_j^0; \Delta y_{ij} f_j^1; \Delta z_{ij} f_j^2]$$

method	ModelNet40	S3DIS	PartNet	
	acc	mIoU	val	test
DensePoint [15]	93.2	-	-	-
KPConv [30]	92.9	65.7	-	-
PointCNN [14]	92.5	65.4	-	46.4
baseline*	91.4	51.5	42.5	44.6
baseline† (AVG)	91.4	51.0	44.2	45.8
baseline† (MAX)	91.8	58.4	45.4	47.4
point-wise MLP	92.8	66.2	48.1	51.5
pseudo grid	93.0	65.9	50.8	53.0
adapt weights	93.0	66.5	50.1	53.5
PosPool (PPNet)	92.9	66.5	50.0	53.4
PosPool* (PPNet*)	**93.2**	**66.7**	**50.6**	**53.8**

Ze Liu, Han Hu, Yue Cao et al. *A Closer Look at Local Aggregation Operators in Point Cloud Analysis*, ECCV 2020

Point Transformer





Method	input	mAcc	OA
3DShapeNets [47]	voxel	77.3	84.7
VoxNet [23]	voxel	83.0	85.9
Subvolume [26]	voxel	86.0	89.2
MVCNN [34]	image	–	90.1
PointNet [25]	point	86.2	89.2
A-SCN [48]	point	87.6	90.0
Set Transformer [17]	point	–	90.4
PAT [50]	point	–	91.7
PointNet++ [27]	point	–	91.9
SpecGCN [40]	point	–	92.1
PointCNN [20]	point	88.1	92.2
DGCNN [44]	point	90.2	92.2
PointWeb [55]	point	89.4	92.3
SpiderCNN [49]	point	–	92.4
PointConv [46]	point	–	92.5
Point2Sequence [21]	point	90.4	92.6
KPConv [37]	point	–	92.9
InterpCNN [22]	point	–	93.0
PointTransformer	point	**90.6**	**93.7**

ModelNet40

S3DIS

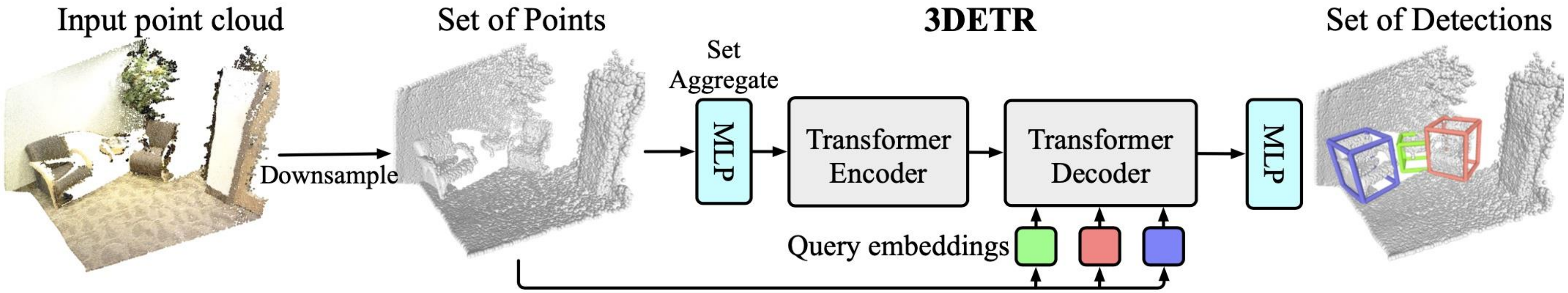
Method	OA	mAcc	mIoU
PointNet [25]	78.5	66.2	47.6
RSNet [12]	–	66.5	56.5
SPGraph [15]	85.5	73.0	62.1
PAT [50]	–	76.5	64.3
PointCNN [20]	88.1	75.6	65.4
PointWeb [55]	87.3	76.2	66.7
ShellNet [53]	87.1	–	66.8
RandLA-Net [37]	88.0	82.0	70.0
KPConv [37]	–	79.1	70.6
PointTransformer	**90.2**	**81.9**	**73.5**

Method	cat. mIoU	ins. mIoU
PointNet [25]	80.4	83.7
A-SCN [48]	–	84.6
PCNN [42]	81.8	85.1
PointNet++ [27]	81.9	85.1
DGCNN [44]	82.3	85.1
Point2Sequence [21]	–	85.2
SpiderCNN [49]	81.7	85.3
SPLATNet [33]	83.7	85.4
PointConv [46]	82.8	85.7
SGPN [43]	82.8	85.8
PointCNN [20]	84.6	86.1
InterpCNN [22]	84.0	86.3
KPConv [37]	**85.1**	86.4
PointTransformer	83.7	**86.6**

ShapeNetPart

Hengshuang Zhao, Li Jiang et al. *Point Transformer*, Arxiv 2020

3D Object Detection with Transformers



Method	ScanNetV2		SUN RGB-D	
	AP25	AP50	AP25	AP50
BoxNet	49.0	21.1	52.4	25.1
3DETR (ICCV21)	62.7	37.5	58.0	30.3
VoteNet (ICCV19)	60.4	37.5	58.3	33.4
3DETR-m (ICCV21)	65.0	47.0	59.1	32.7
H3DNet (ECCV20)	67.2	48.1	60.1	39.0
GFTrans (ICCV21)	**69.1**	**52.8**	**63.0**	**45.2**

Ze Liu, Zheng Zhang, Yue Cao et al. *Group-Free 3D Object Detection via Transformers*, ICCV 2021
Misra Ishan et al. *An End-to-End Transformer Model for 3D Object Detection*, ICCV 2021

Summary: The trend of Transformer

- Visual elements
  - Pixel-to-pixel
  - Pixel-to-object
  - Object-to-object
- Multi-modality
  - Visual token (patches/objects/pixels) & language token (words)
- Point clouds
  - Points in point clouds

The trend of Transformer

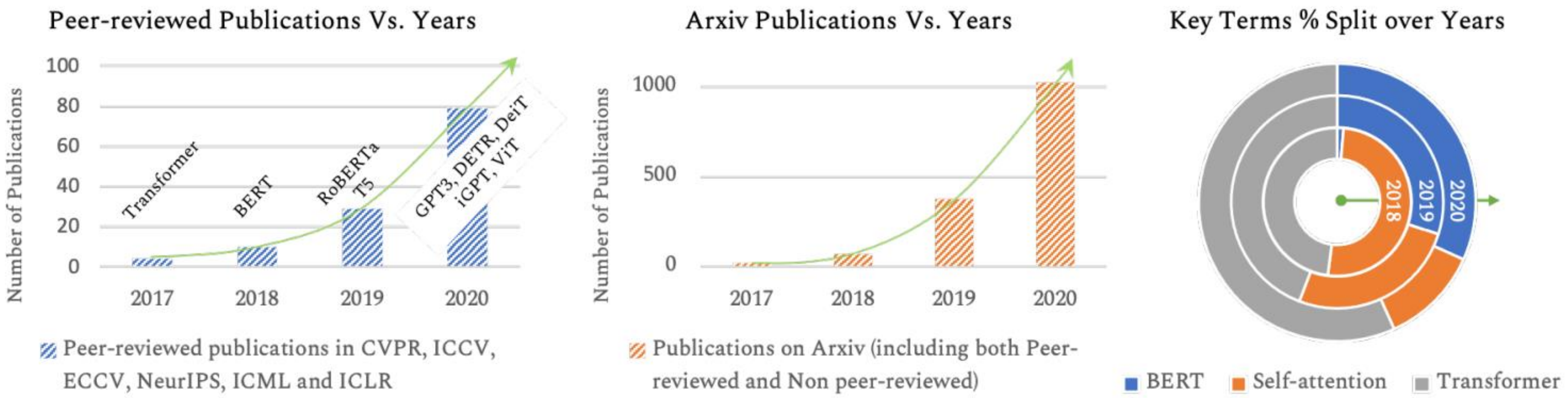
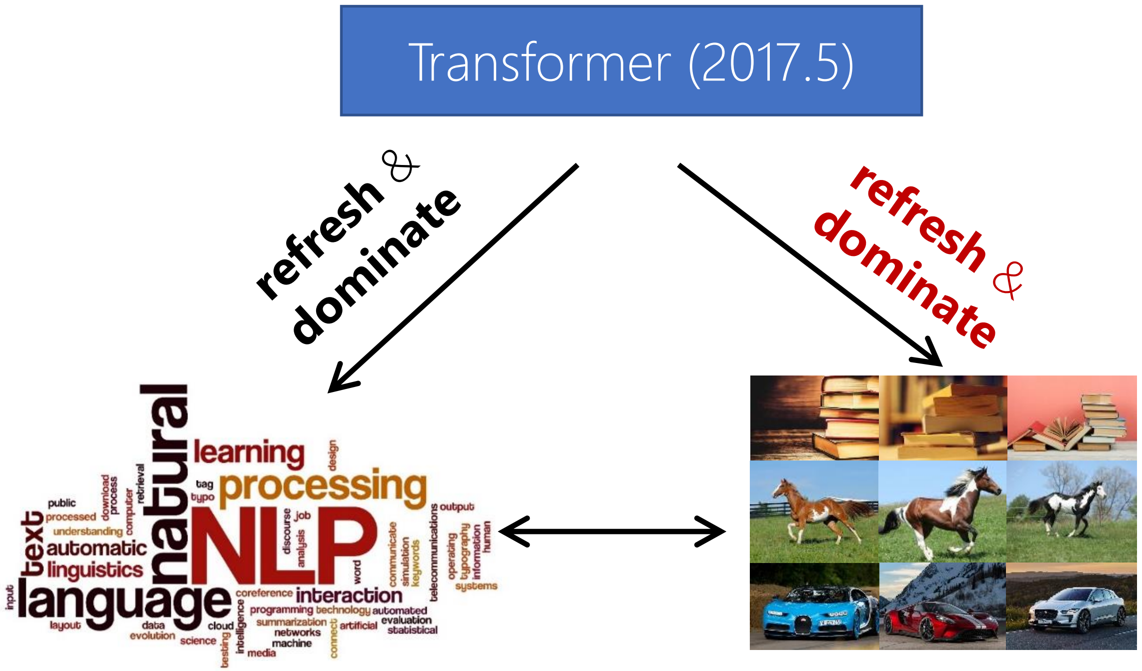


Fig. 1: Statistics on the number of times keywords such as BERT, Self-Attention, and Transformers appear in the titles of Peer-reviewed and arXiv papers over the past few years (in Computer Vision and Machine Learning). The plots show consistent growth in recent literature. This survey covers recent progress on Transformers in the computer vision domain.

Credit to: Salman Khan et al. *Transformers in Vision: A Survey.* Arxiv 2021

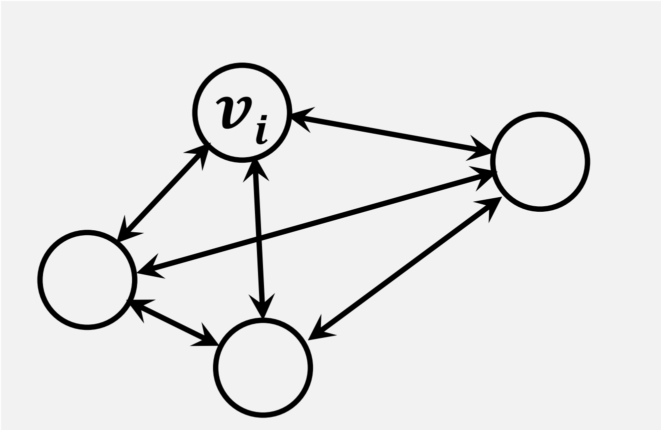
Why Transformer?

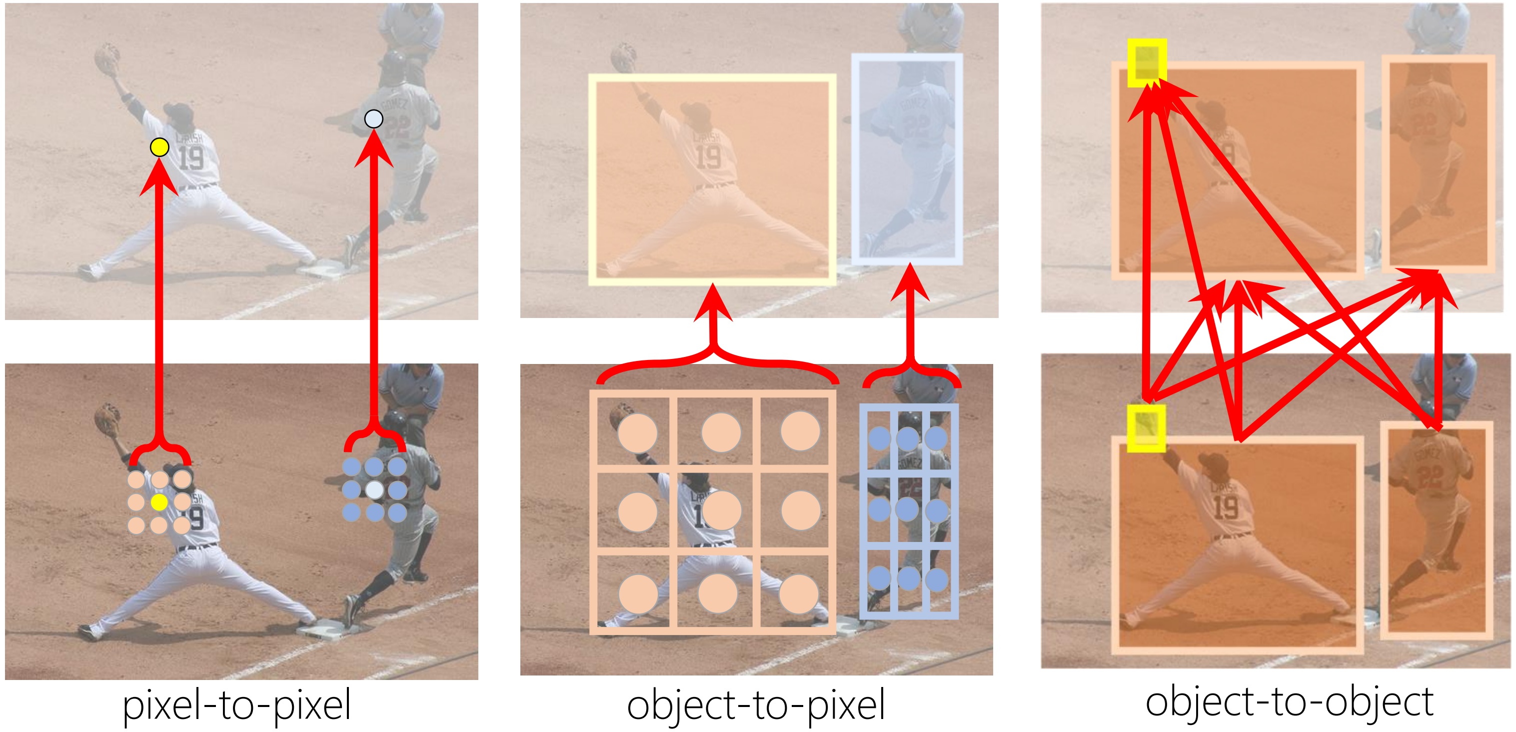
- Three merits to use Transformer vs. CNNs
  - Merit I: **General** modeling capability
  - Merit II: More **powerful** modeling capability
  - Merit III: **Scaling** to large model and large data



Three merits to use Transformer

- <u>Merit I</u>: General modeling capability
  - All concepts (concrete or abstract) and relationships can be modeled





Three merits to use Transformer

- Merit I: General modeling capability
  - All concepts (concrete or abstract) and relationships can be modeled
  - **Perceiver-IO**: A general architecture for structured inputs & outputs

Domain	Input Modality	Encoder KV input	Encoder KV channels	Decoder query input	Decoder query channels
Language (MLM)	Text	byte/token encoding + learned pos	768	learned pos	1280
Language (Perceiver IO++ MLM)	Text	byte/token encoding + learned pos	768	learned pos	1536
Language (GLUE)	Text	byte/token encoding + learned pos	768	Class query (per-task)	1280
Language (Perceiver IO++ GLUE)	Text	byte/token encoding + learned pos	768	Class query (per-task)	1536
Optical Flow	Video (image pairs)	[2 × conv features, 3D FFs]	450	[2 × conv features, 3D FFs]	450
Optical Flow (pixels)	Video (image pairs)	[Linear(2 × RGB), 2D FFs]	322	[Linear(2 × RGB), 2D FFs]	322
Kinetics	Video, Audio, Label	[Linear(RGB), 3D FFs, learned modality feat.] [sound pressure, 1D FF, learned modality feat.] [one-hot label, learned modality feat.]	704 704 704	[3D FFs, learned modality feat.] [1D FF, learned modality feat.] [learned modality feat.]	1026 1026 1026
StarCraft II	SC2 entities	Entity features	128	Entity features	128
ImageNet	Image	[RGB, 2D FFs]	261	Class query (single)	1024
ImageNet (learned pos)	Image	[Linear(RGB), learned pos]	512	Class query (single)	1024
ImageNet (conv)	Image	[Conv features, 2D FFs]	322	Class query (single)	1024

Three merits to use Transformer

- Merit II: More powerful modeling
  - "Convolution is too local!" -> larger receptive field with lower computation
  - "Convolution is exponentially inefficient! " -> adaptive computation

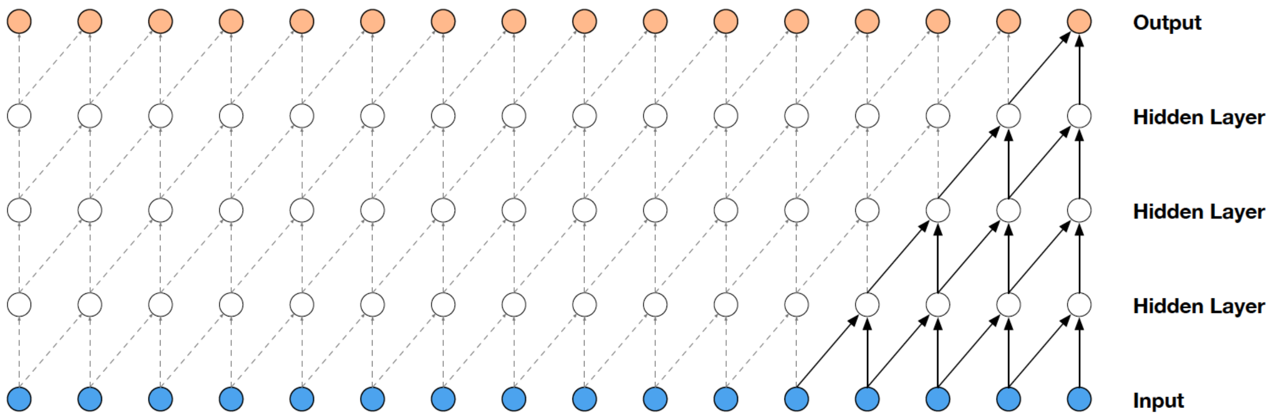
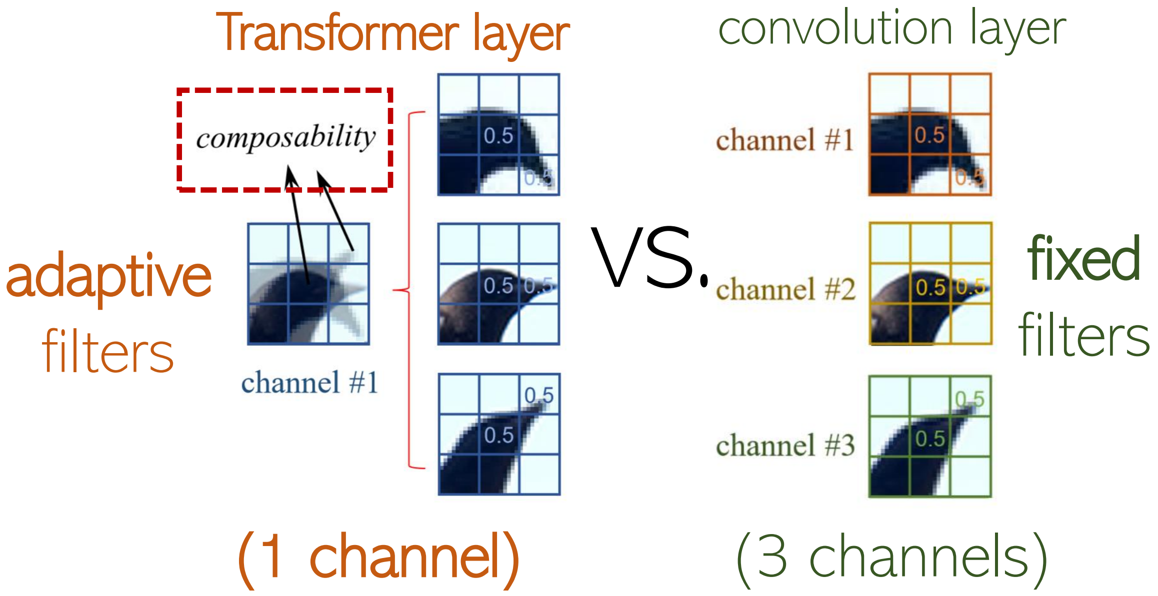
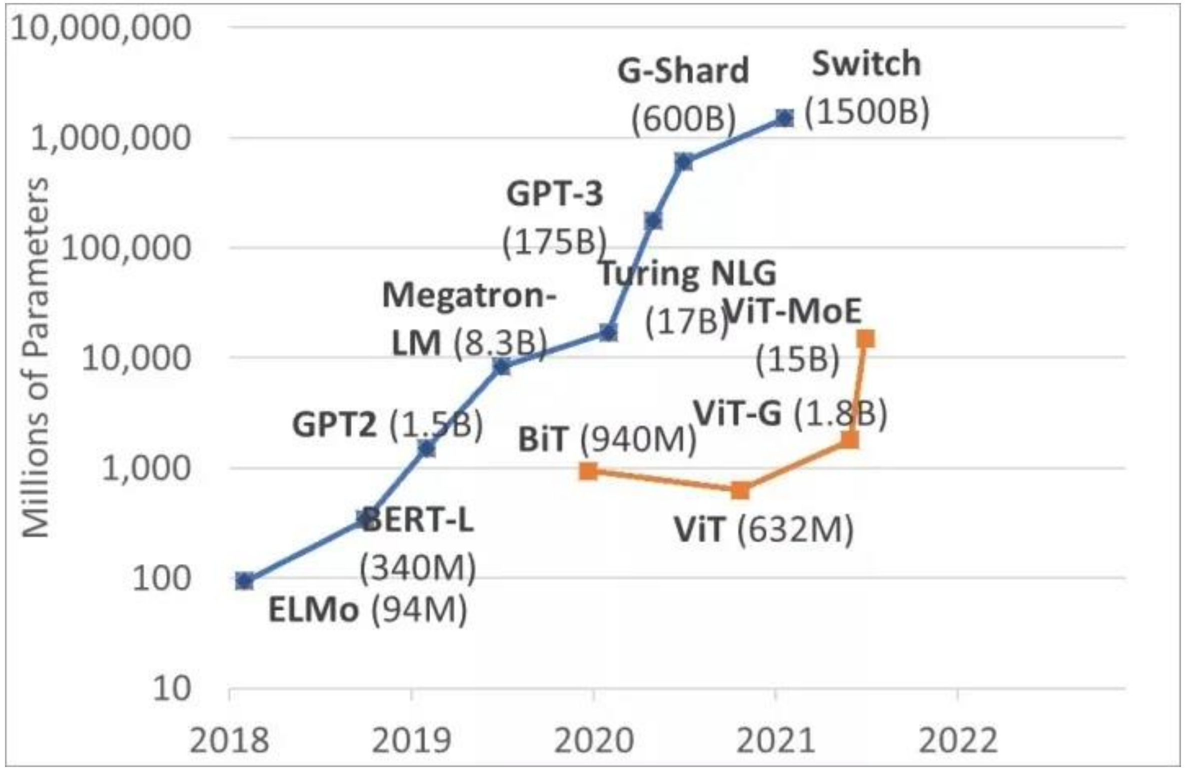


Figure credit: Van Den Oord et al.



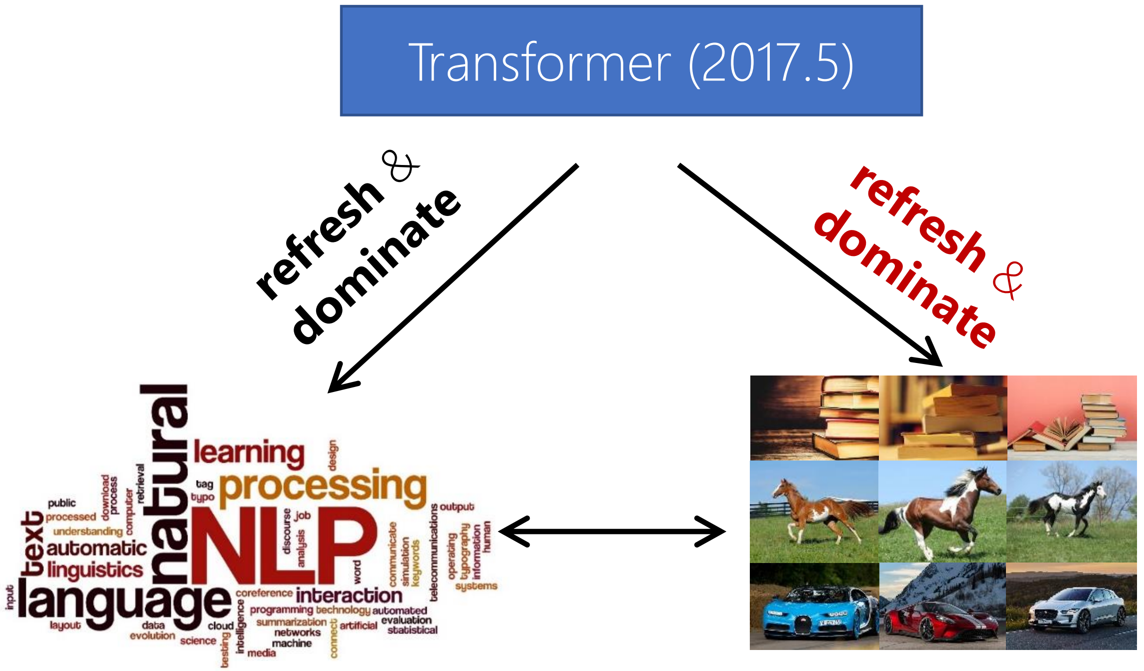
Three merits to use Transformer

- <u>Merit III</u>: Scaling to large model and large data



Why Transformer?

- Three merits to use Transformer vs. CNNs
  - Merit I: **General** modeling capability
  - Merit II: More **powerful** modeling capability
  - Merit III: **Scaling** to large model and large data



What is next?

- The machine learning era
  - a **paradigm** unification: learning from **historical data** and make future predictions
- The deep learning era
  - The unification of **architecture:** CNN, RNN, LSTM -> Transformer
- The universal model era?
  - The unification of **model** itself: ONE model for multimodal input and multi tasks

The new era just begins, and there is a long road in the front

Thanks All!
Q & A