Deformable DETR: Deformable Transformers for End-to-End Object Detection

Xizhou Zhu*, **Weijie Su***, Lewei Lu, Bin Li, Xiaogang Wang, Jifeng Dai

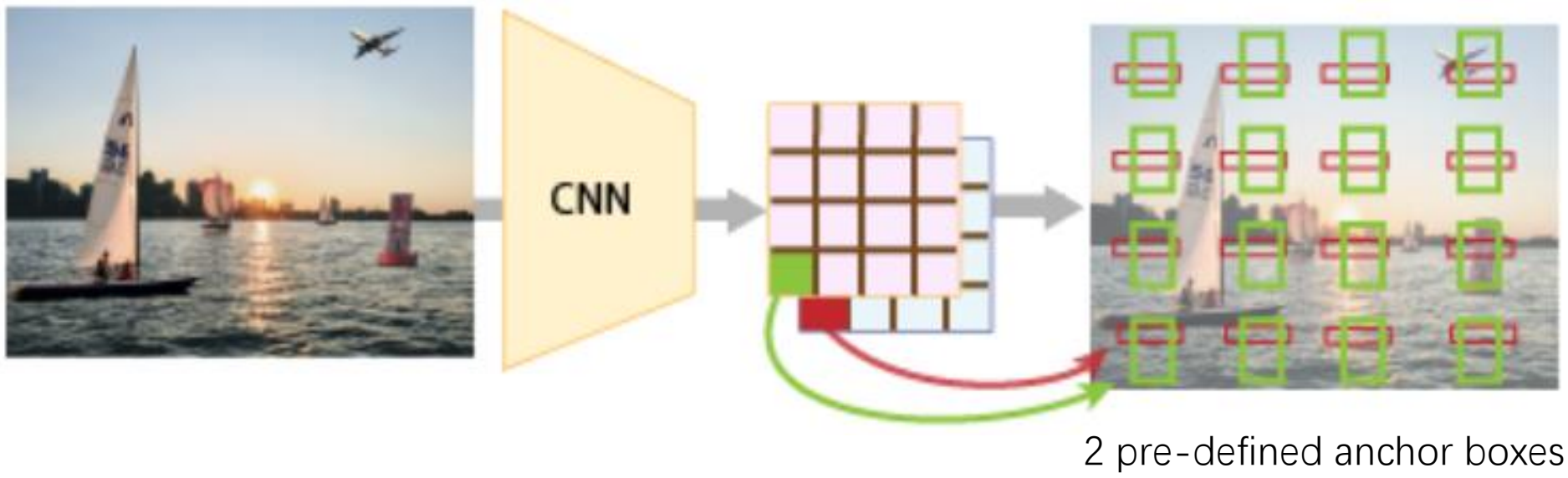
SenseTime Research; University of Science and Technology of China; The Chinese University of Hong Kong

Previous Modern Object Detectors

- Rely on Hand-Crafted Components

Previous Modern Object Detectors

- Rely on Hand-Crafted Components, e.g.,
  - anchor generation → anchor-free detector

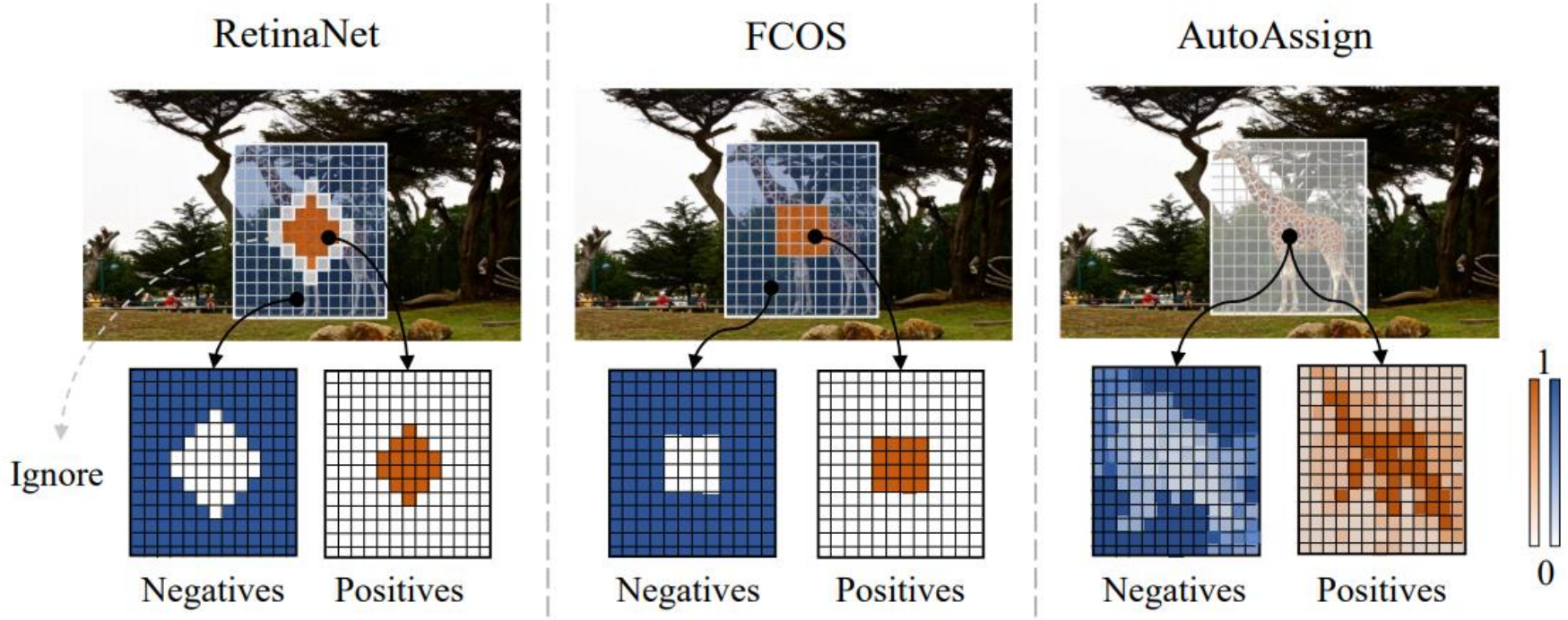


2 pre-defined anchor boxes

https://www.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html

Previous Modern Object Detectors

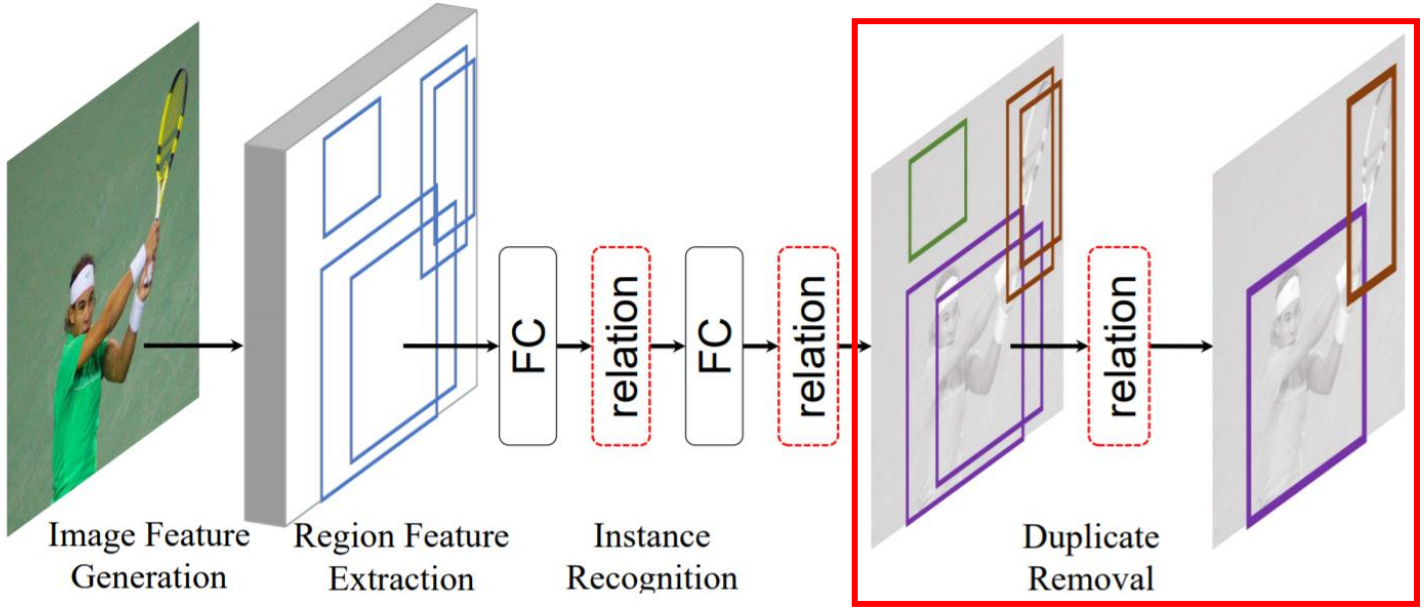
- Rely on Hand-Crafted Components, e.g.,
  - anchor generation → anchor-free detector
  - rule-based training target assignment → automatic target assignment



Zhu, Benjin, et al. "Autoassign: Differentiable label assignment for dense object detection." *arXiv:2007.03496* (2020).

Previous Modern Object Detectors

- Rely on Hand-Crafted Components, e.g.,
  - anchor generation → anchor-free detector
  - rule-based training target assignment → automatic target assignment
  - non-maximum suppression (NMS) post-processing → learnable deduplication



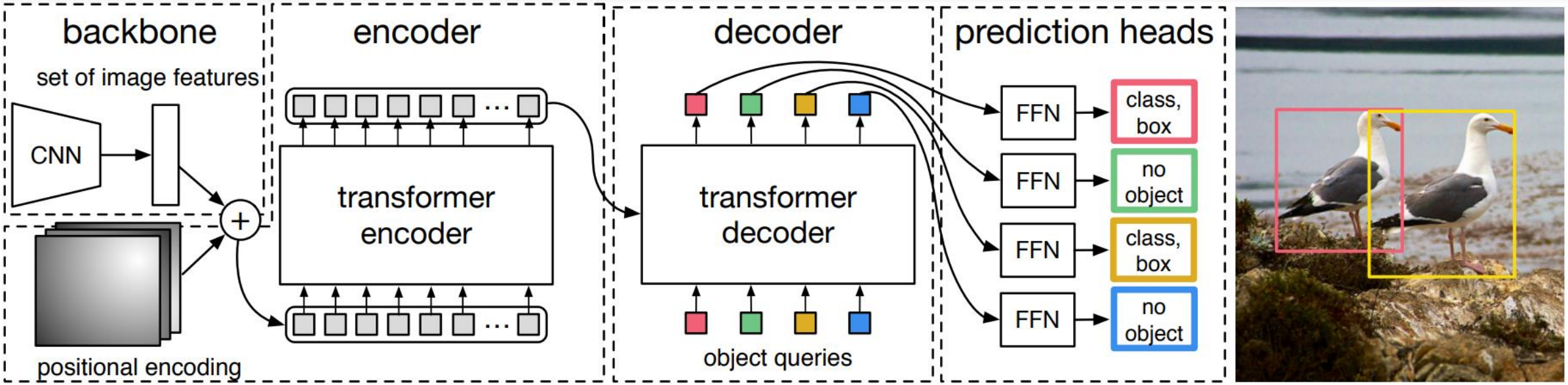
Hu, Han, et al. "Relation networks for object detection." *In CVPR* 2018.

Previous Modern Object Detectors

- Rely on Hand-Crafted Components, e.g.,
  - anchor generation → anchor-free detector
  - rule-based training target assignment → automatic target assignment
  - non-maximum suppression (NMS) post-processing → learnable deduplication

- Not Fully End-to-End
  - complex combination of hand-crafted components
  - requiring manually adjustment (e.g., anchor size and NMS threshold) for specific datasets

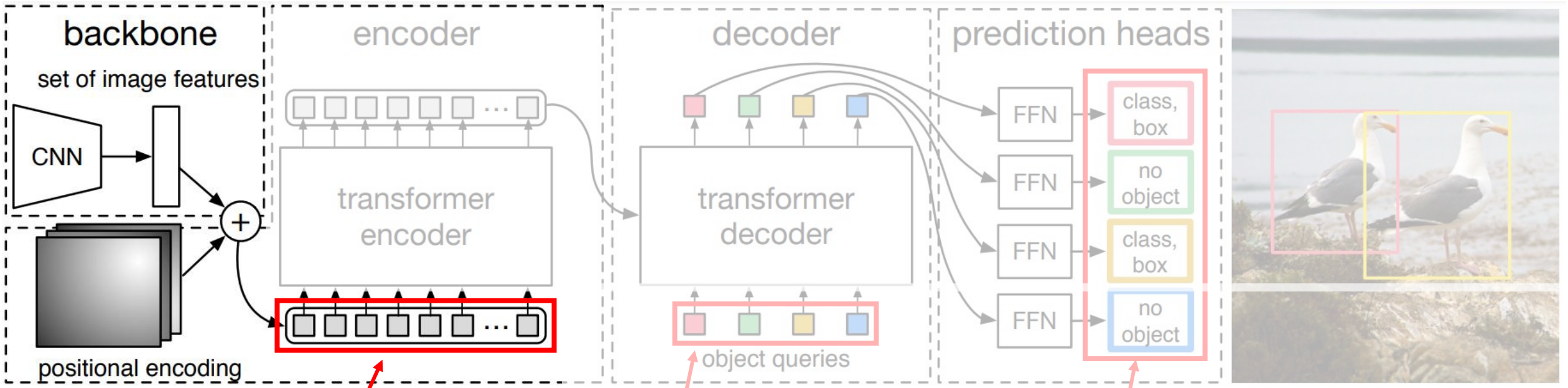
DETR - The First End-to-End Object Detector



- Eliminate the need for hand-crafted components
- Achieve very competitive performance with previous modern object detectors

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

DETR - Architecture



$HW \times C$

flattened image features

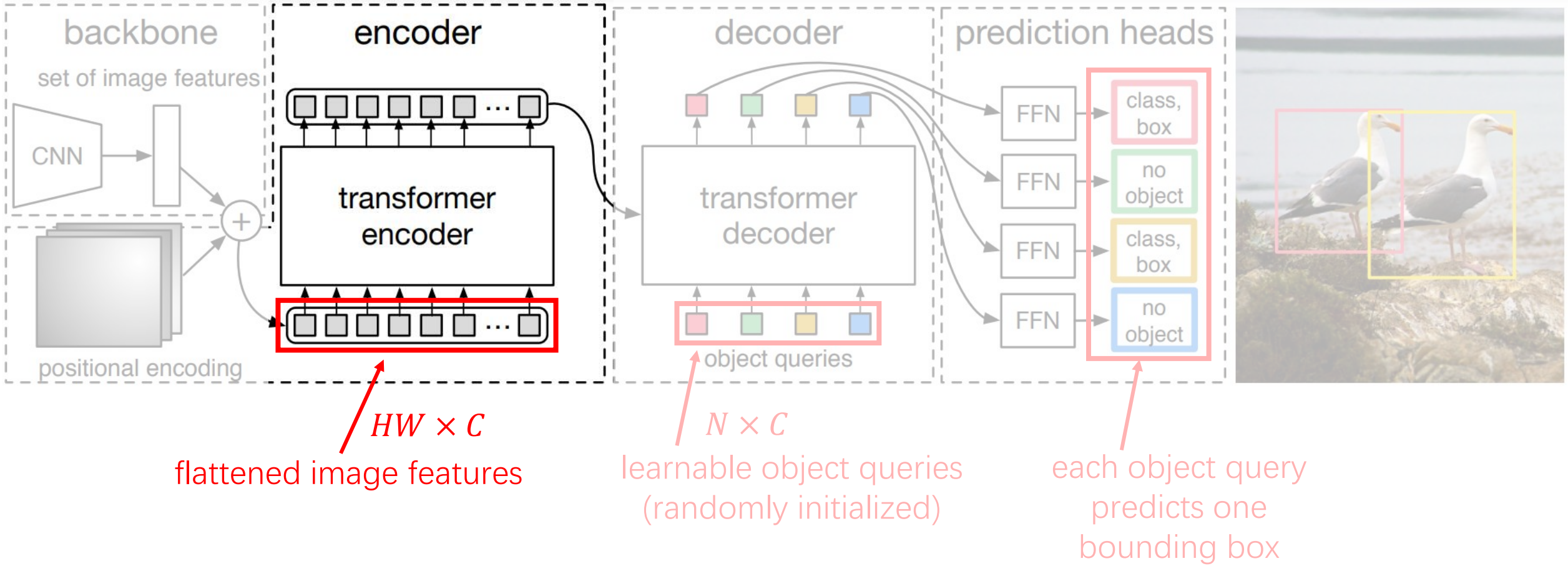
$N \times C$

learnable object queries (randomly initialized)

each object query predicts one bounding box

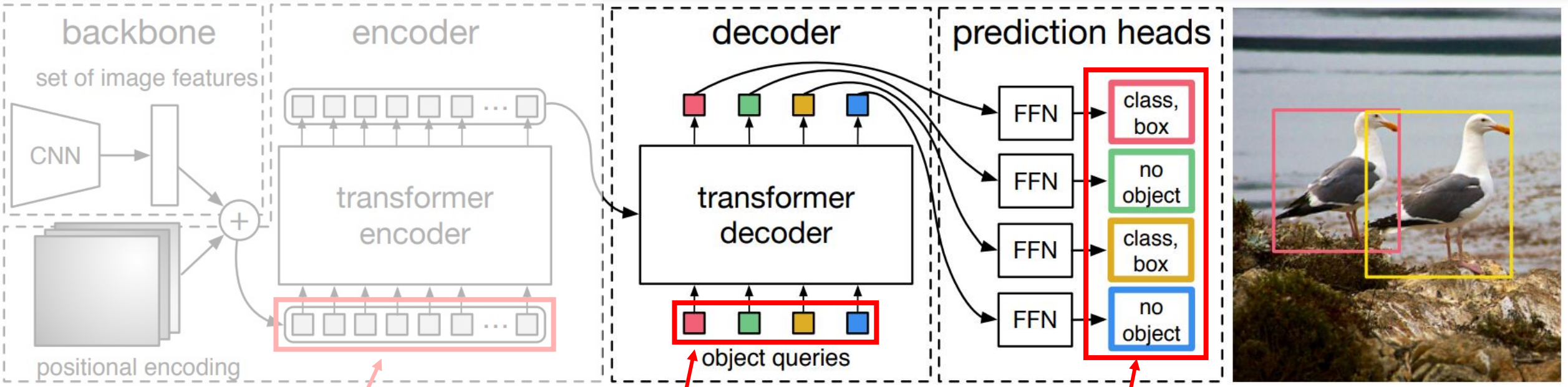
Carion, Nicolas, et al. "End-to-End Object Detection with Transformers." In ECCV 2020.

DETR - Architecture



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

DETR - Architecture



$HW \times C$

flattened image features

$N \times C$

learnable object queries (randomly initialized)

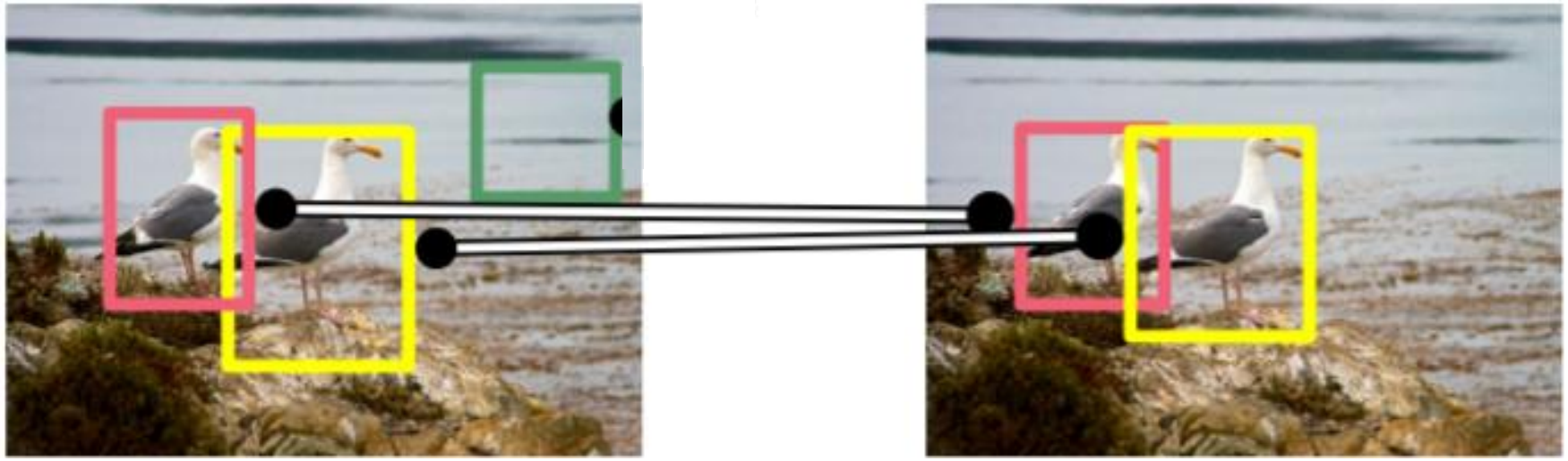
each object query predicts one bounding box

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

DETR - Training Target Assignment (Bipartite Matching)

Predicted boxes

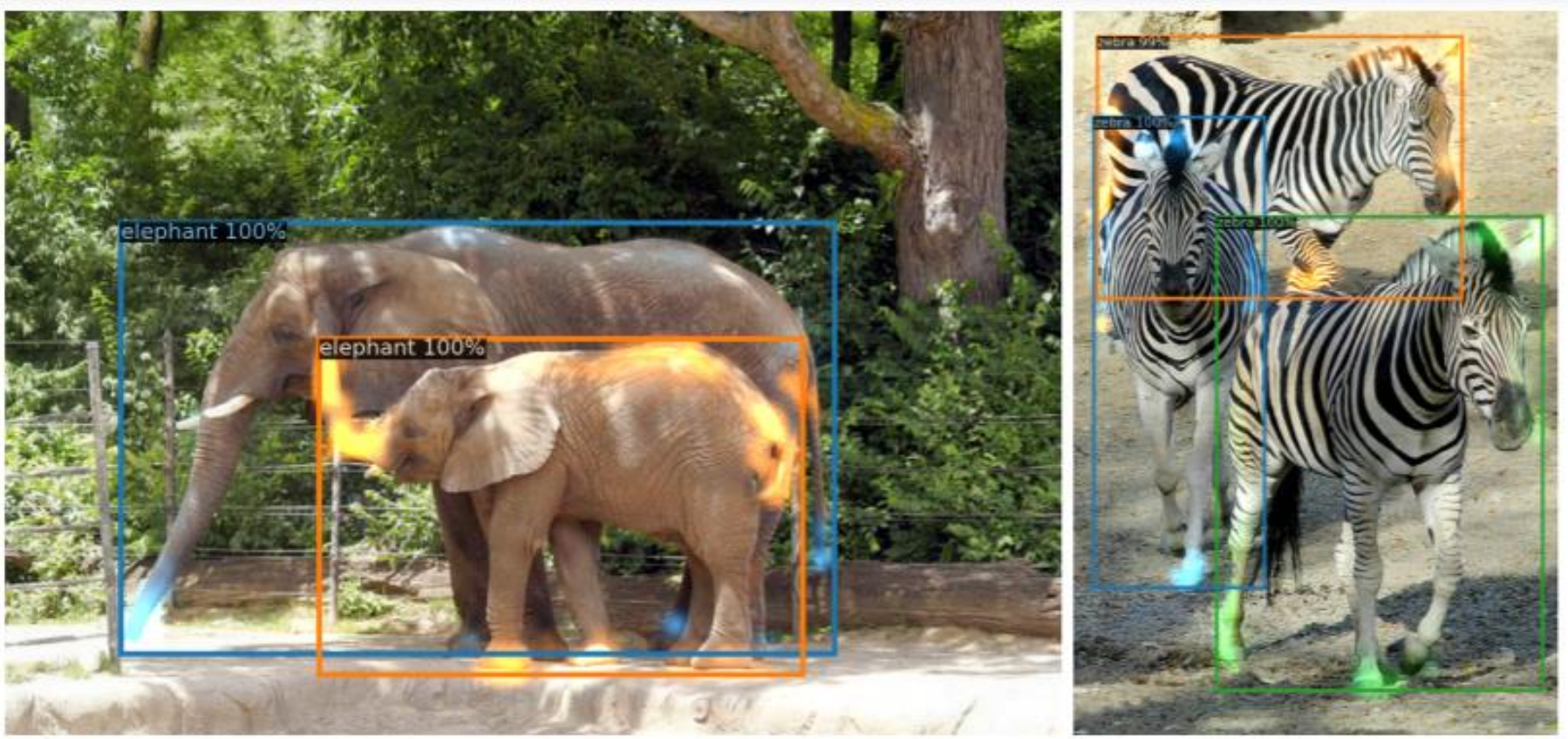
Ground-truth boxes



Bipartite matching assignment to minimize the classification and regression costs

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

DETR - Visualizing Decoder Cross-Attention



- Different color indicates different object query
- Each object query attends to the object extremities of each object instance

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

Two main issues of DETR

- Slow convergence
  - 500 epochs to converge on COCO, 10x~20x slower than Faster R-CNN
- Limited feature resolution
  - 4.1 lower AP on small objects than Faster R-CNN+FPN

Two main issues of DETR

- Slow convergence
  - 500 epochs to converge on COCO, 10x~20x slower than Faster R-CNN
- Limited feature resolution
  - 4.1 lower AP on small objects than Faster R-CNN+FPN

Core reason: Limitation of Transformer attention in processing image feature maps

- Quadratic complexity (encoder self-attention)
  - memory and computation complexity: $O(N^2)$, N is the number of pixels
  - couldn't afford high resolution feature maps

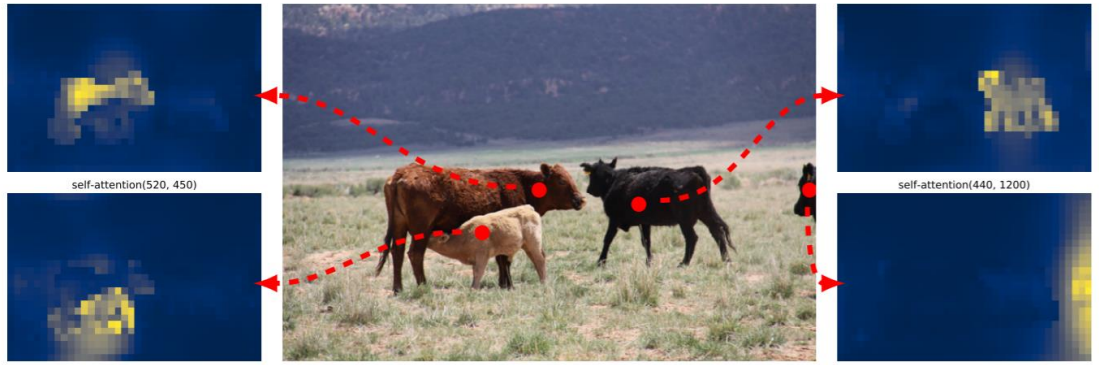
- Needs long training schedule so that attention weight focus on specific keys
  - $A_{mqk} \approx \frac{1}{N_k}$ at initialization, nearly uniform attention weights to all the pixels
    - N_k is the number of key elements
  - In the image domain, where the key elements are usually of image pixels, N_k can be very large and the convergence is tedious

Core reason: Limitation of Transformer attention in processing image feature maps

- Quadratic complexity (encoder self-attention)
  - memory and computation complexity: $O(N^2)$, N is the number of pixels
  - couldn't afford high resolution feature maps

- Needs long training schedule so that attention weight focus on specific keys
  - $A_{mqk} \approx \frac{1}{N_k}$ at initialization, nearly uniform attention weights to all the pixels
    - N_k is the number of key elements
  - In the image domain, where the key elements are usually of image pixels, N_k can be very large and the convergence is tedious

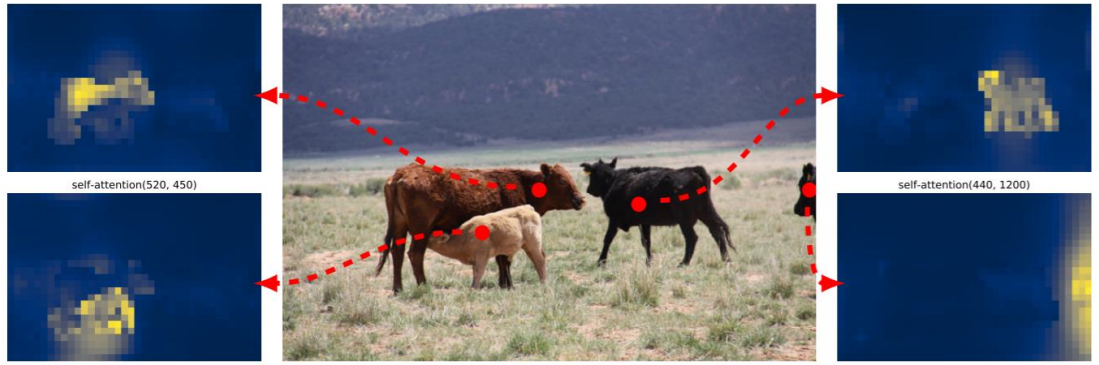
Efficient Sparse Attention in Image Domain



Dense attention (e.g., Transformer[1], Non-Local[2])

look over **all possible** spatial locations

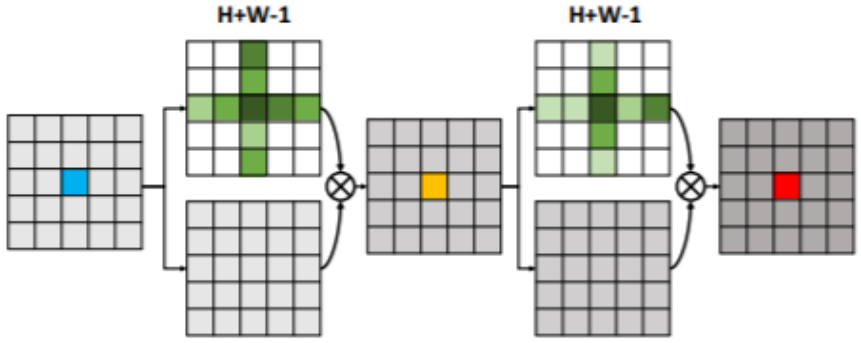
Efficient Sparse Attention in Image Domain



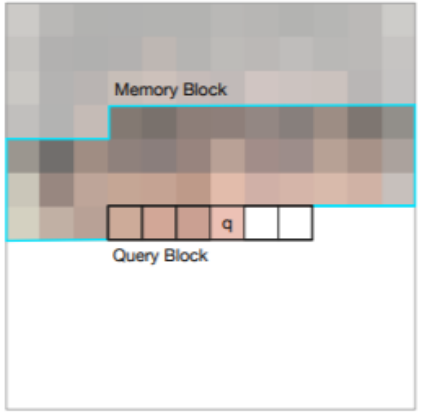
Dense attention (e.g., Transformer[1], Non-Local[2])

look over **all possible** spatial locations

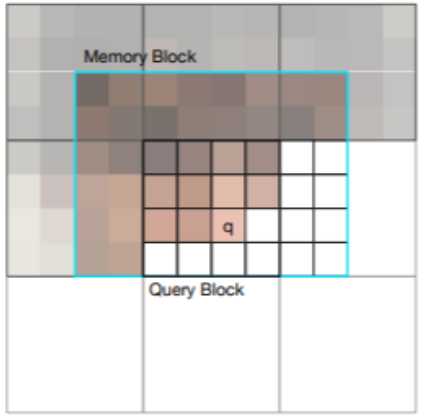
look over **pre-defined sparse** spatial locations



attention along each axis
(e.g., Axial Attention[3], CCNet[4])



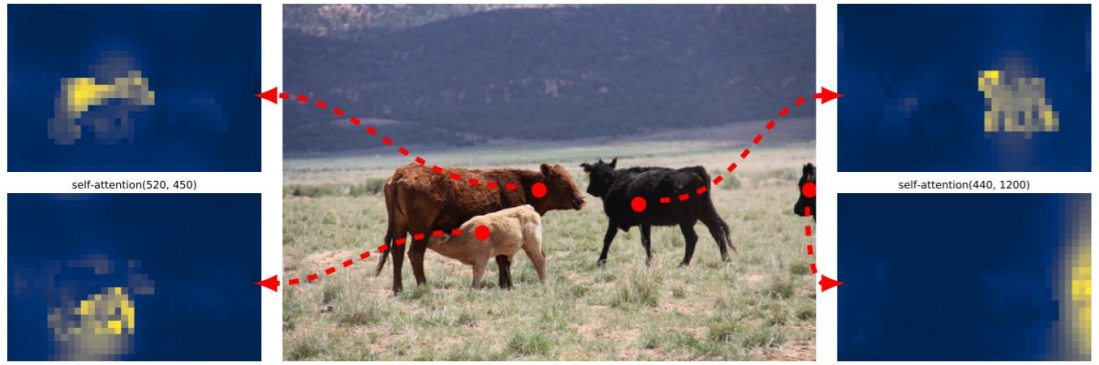
1D local attention
(e.g., Image Transformer[5])



2D local attention
(e.g., Image Transformer[5], Stand-Alone[6], Local Relation[7])

[1] Vaswani, Ashish, et al. “Attention is all you need.” In NeurIPS 2017
[2] Wang, Xiaolong, et al. "Non-local neural networks." In CVPR 2018.
[3] Ho, Jonathan, et al. "Axial Attention in Multidimensional Transformers.“ In ICLR 2020.
[4] Huang, Zilong, et al. "Ccnet: Criss-cross attention for semantic segmentation.“ In ICCV 2019.
[5] Parmar, Niki, et al. "Image transformer." In PMLR 2018.
[6] Ramachandran, Prajit, et al. "Stand-alone self-attention in vision models.“ In NeurIPS 2019.
[7] Hu, Han, et al. "Local relation networks for image recognition.“ In ICCV 2019.

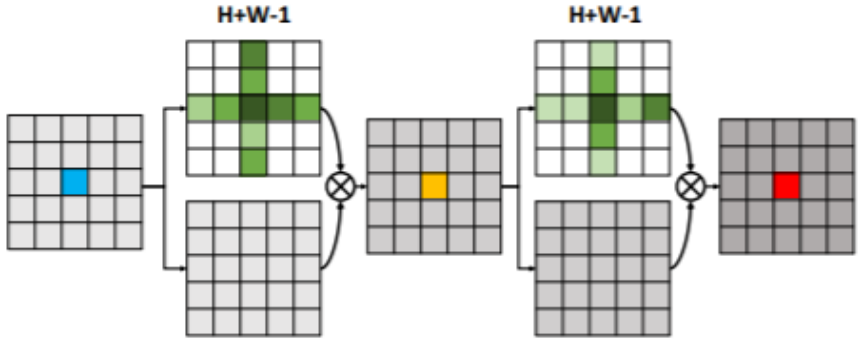
Efficient Sparse Attention in Image Domain



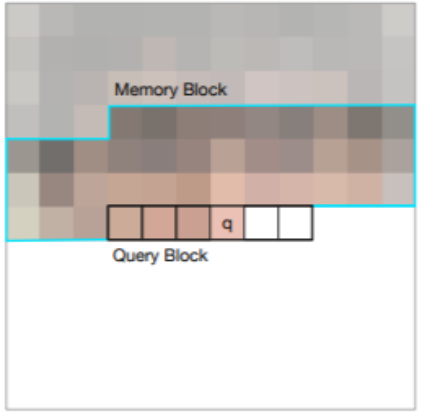
Dense attention (e.g., Transformer[1], Non-Local[2])

look over **all possible** spatial locations

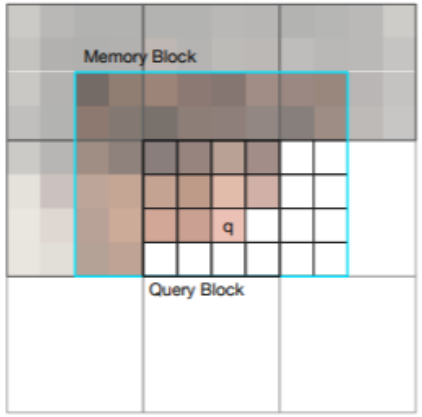
look over **pre-defined sparse** spatial locations



attention along each axis
(e.g., Axial Attention[3], CCNet[4])



1D local attention
(e.g., Image Transformer[5])

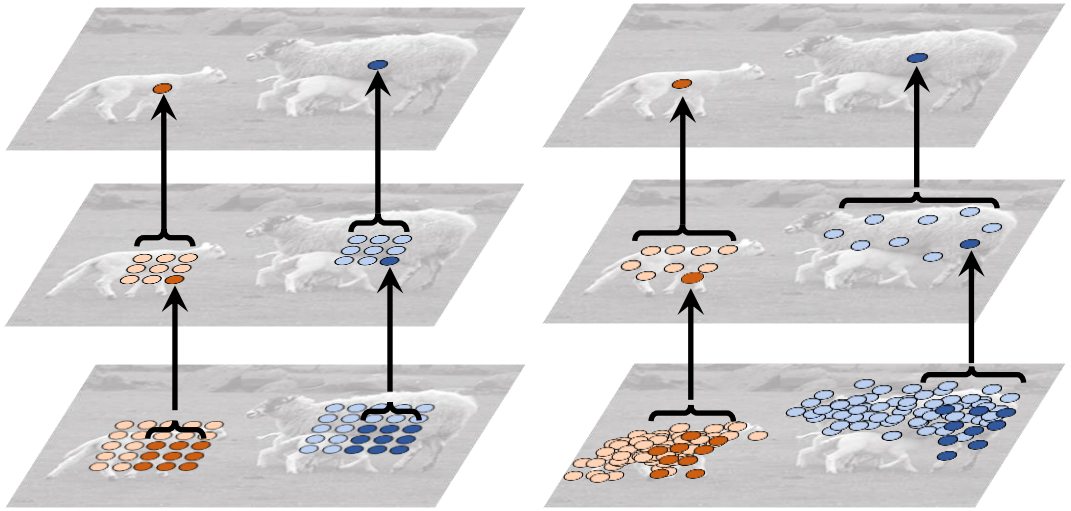


2D local attention
(e.g., Image Transformer[5], Stand-Alone[6], Local Relation[7])

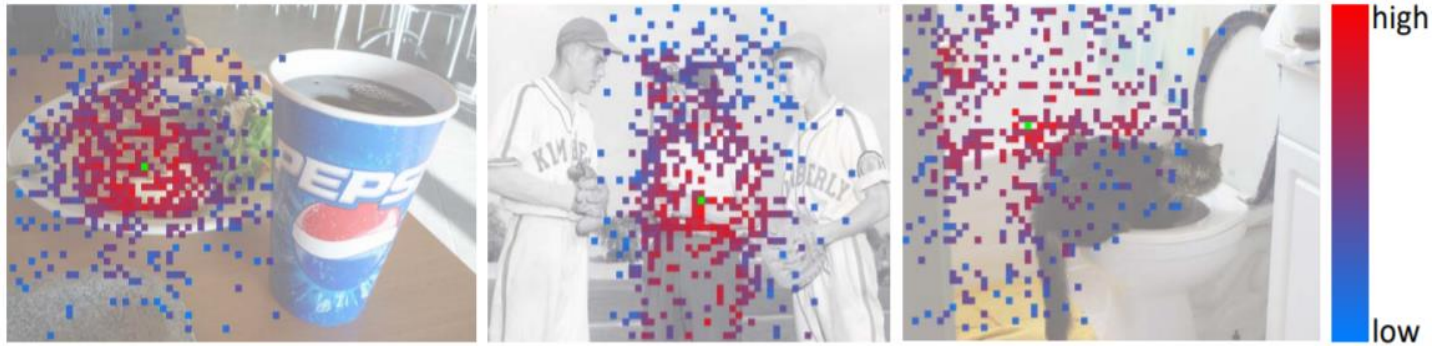
[1] Vaswani, Ashish, et al. “Attention is all you need.” In NeurIPS 2017
[2] Wang, Xiaolong, et al. "Non-local neural networks." In CVPR 2018.
[3] Ho, Jonathan, et al. "Axial Attention in Multidimensional Transformers.“ In ICLR 2020.
[4] Huang, Zilong, et al. "Ccnet: Criss-cross attention for semantic segmentation.“ In ICCV 2019.
[5] Parmar, Niki, et al. "Image transformer." In PMLR 2018.
[6] Ramachandran, Prajit, et al. "Stand-alone self-attention in vision models.“ In NeurIPS 2019.
[7] Hu, Han, et al. "Local relation networks for image recognition.“ In ICCV 2019.

much slower in implementation than traditional convolution with the same FLOPs (at least 3× slower)

Deformable Convolution as Self-Attention



(a) standard convolution (b) deformable convolution



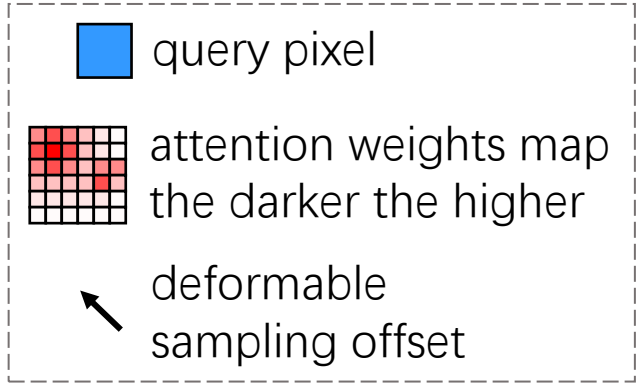
(c) effective sampling locations in deformable convolutions

Deformable convolution is effective and efficient on image recognition

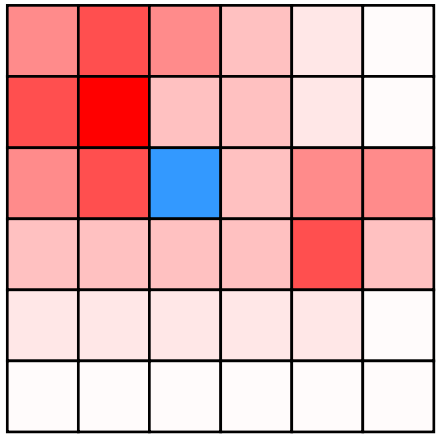
However, it lacks the element relation modeling mechanism.

[1] Dai, Jifeng, et al. "Deformable convolutional networks.“ In ICCV 2017.
[2] Zhu, Xizhou, et al. "Deformable convnets v2: More deformable, better results.“ In CVPR 2019.
[3] Zhu, Xizhou, et al. "An empirical study of spatial attention mechanisms in deep networks.“ In ICCV 2019.

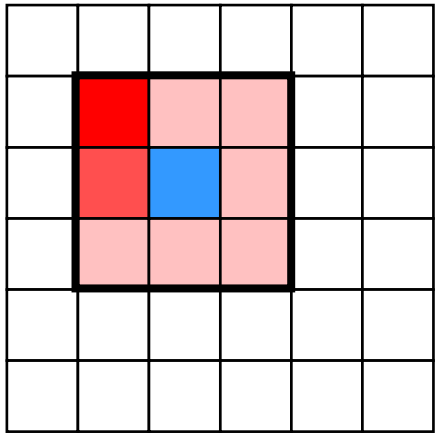
Deformable Attention



- Self-attention patterns for each attention head

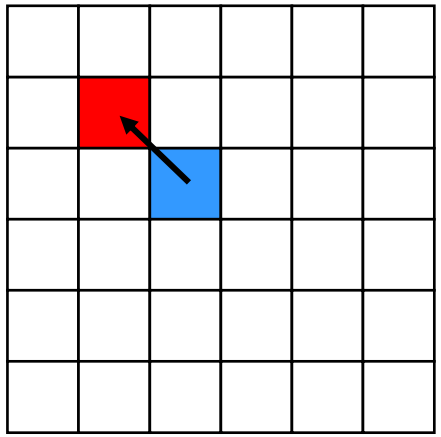


(a) Transformer Attention



(b) Sparse Local Attention

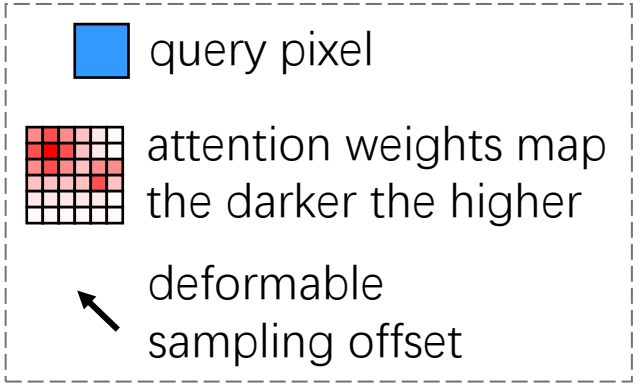
inefficient in implementation



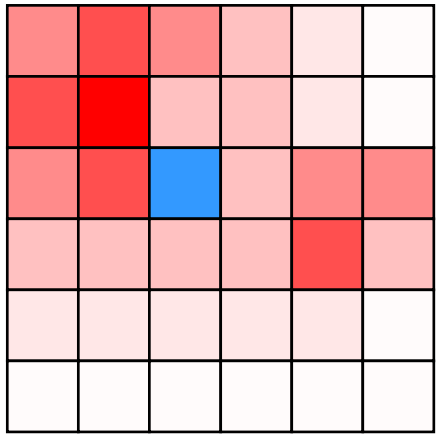
(c) Deformable Convolution

lacks the relation modeling

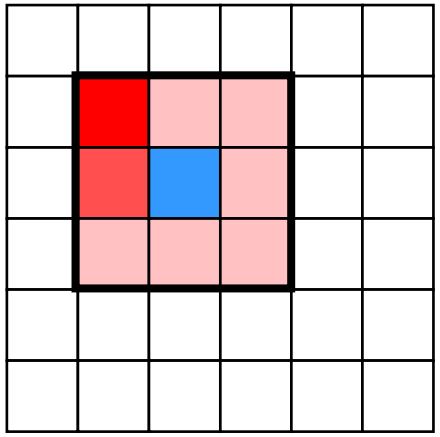
Deformable Attention



- Self-attention patterns for each attention head

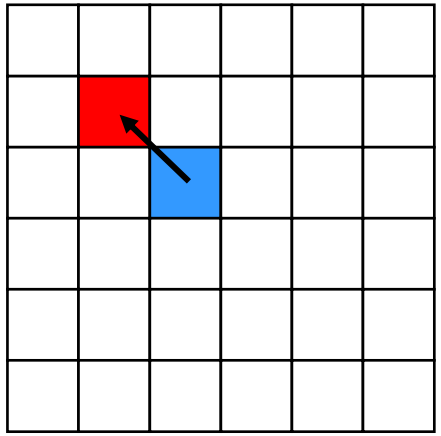


(a) Transformer Attention



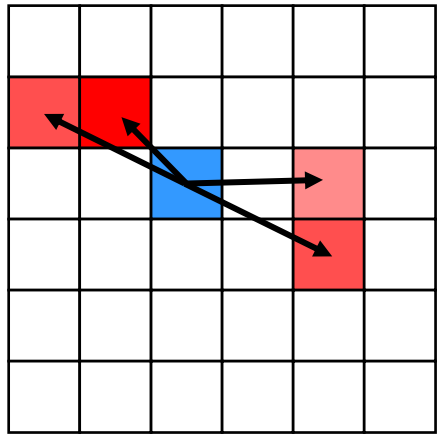
(b) Sparse Local Attention

inefficient in implementation



(c) Deformable Convolution

lacks the relation modeling

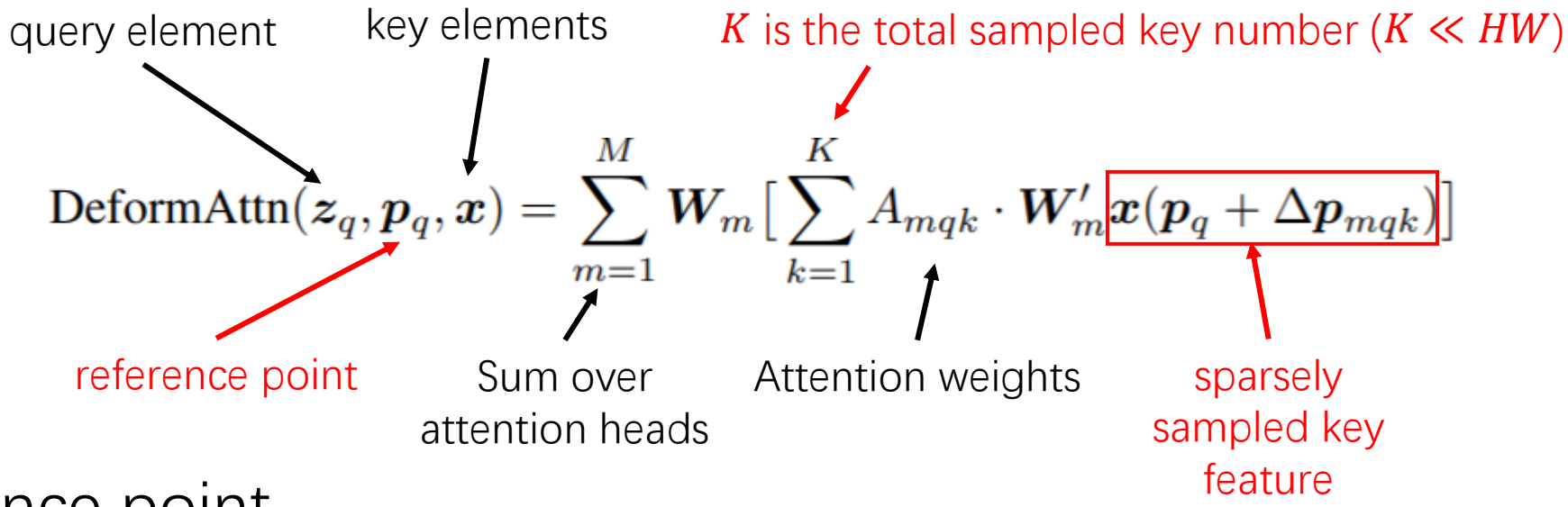


(d) Deformable Attention **(K=4)**

Ours

Deformable Attention Module

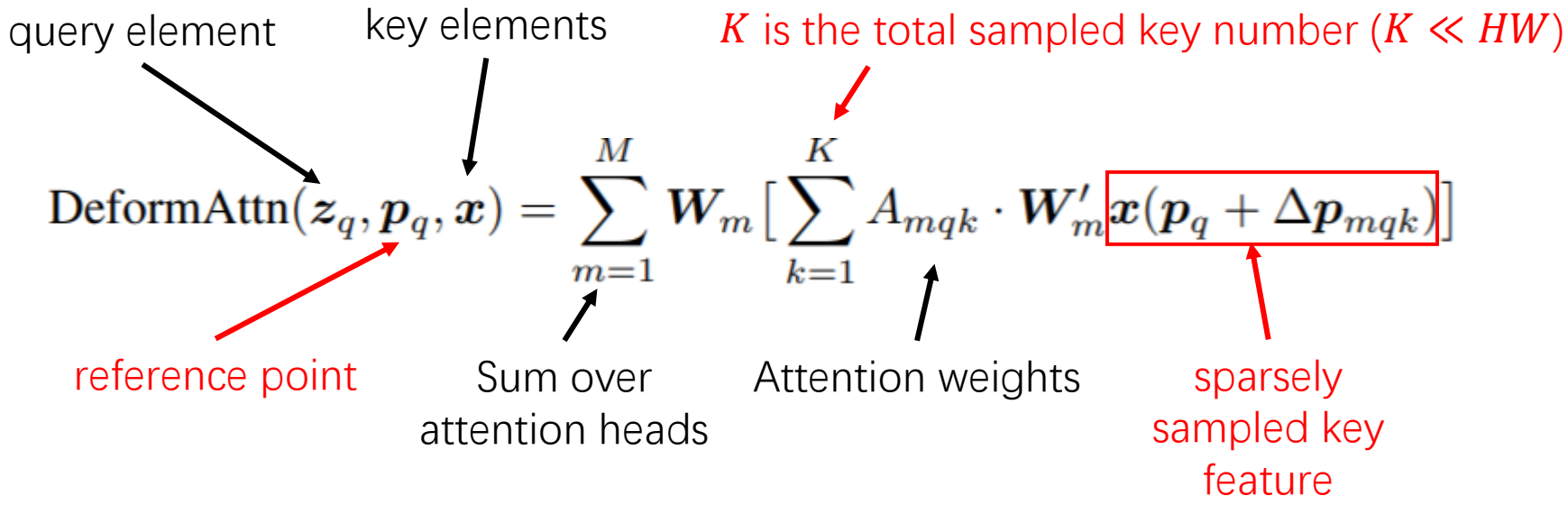
- Formulation



- Reference point
  - In encoder: the query point itself
  - In decoder: predicted from object query embedding

Deformable Attention Module

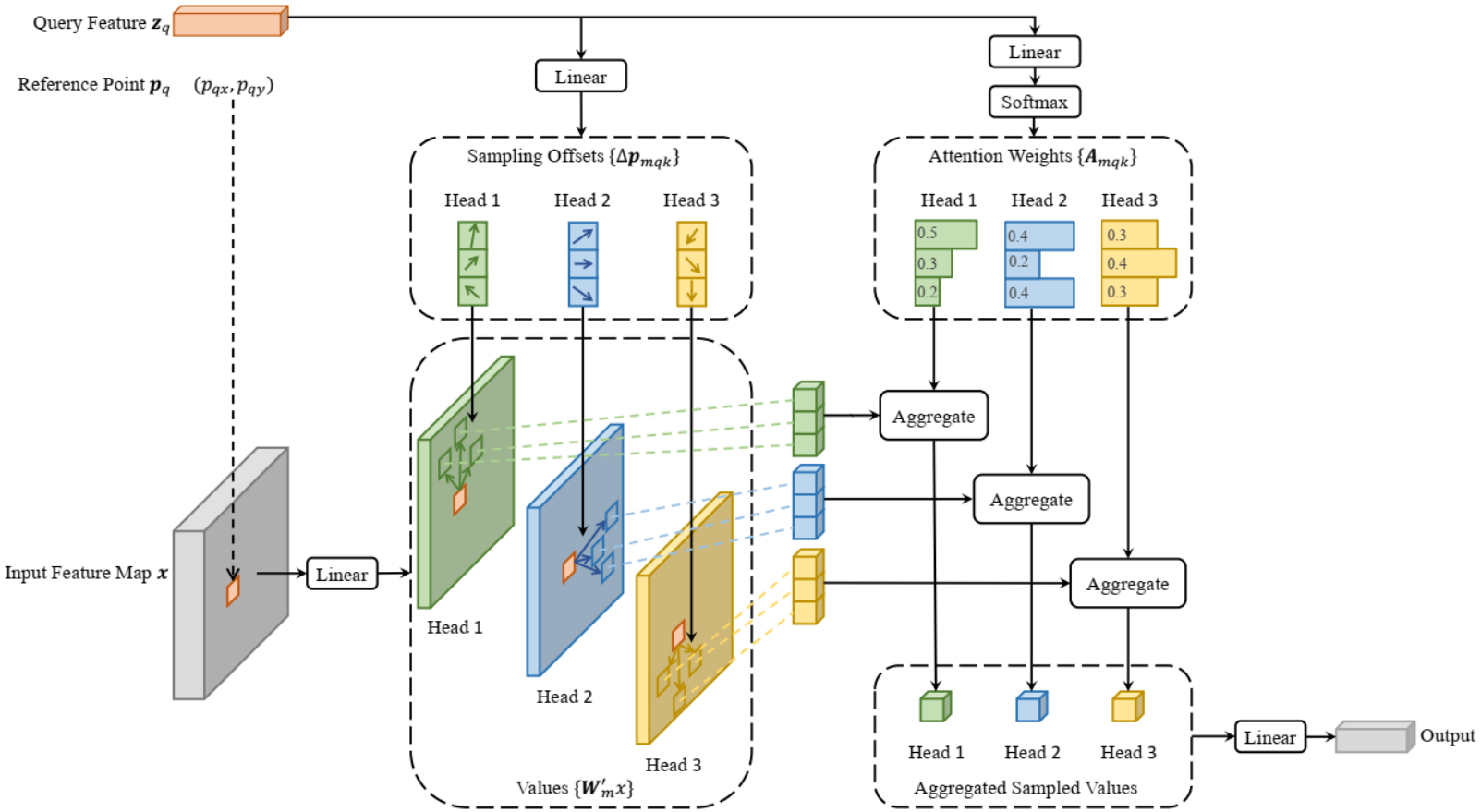
- Formulation



- Equivalent to **Deformable Convolution**, when $\boldsymbol{K} = \boldsymbol{1}$ and W'_m is fixed as an identity matrix
- Equivalent to **Transformer Attention**, when $\boldsymbol{K} = \boldsymbol{HW}$ and the sampling points traverse all possible locations

Deformable Attention Module

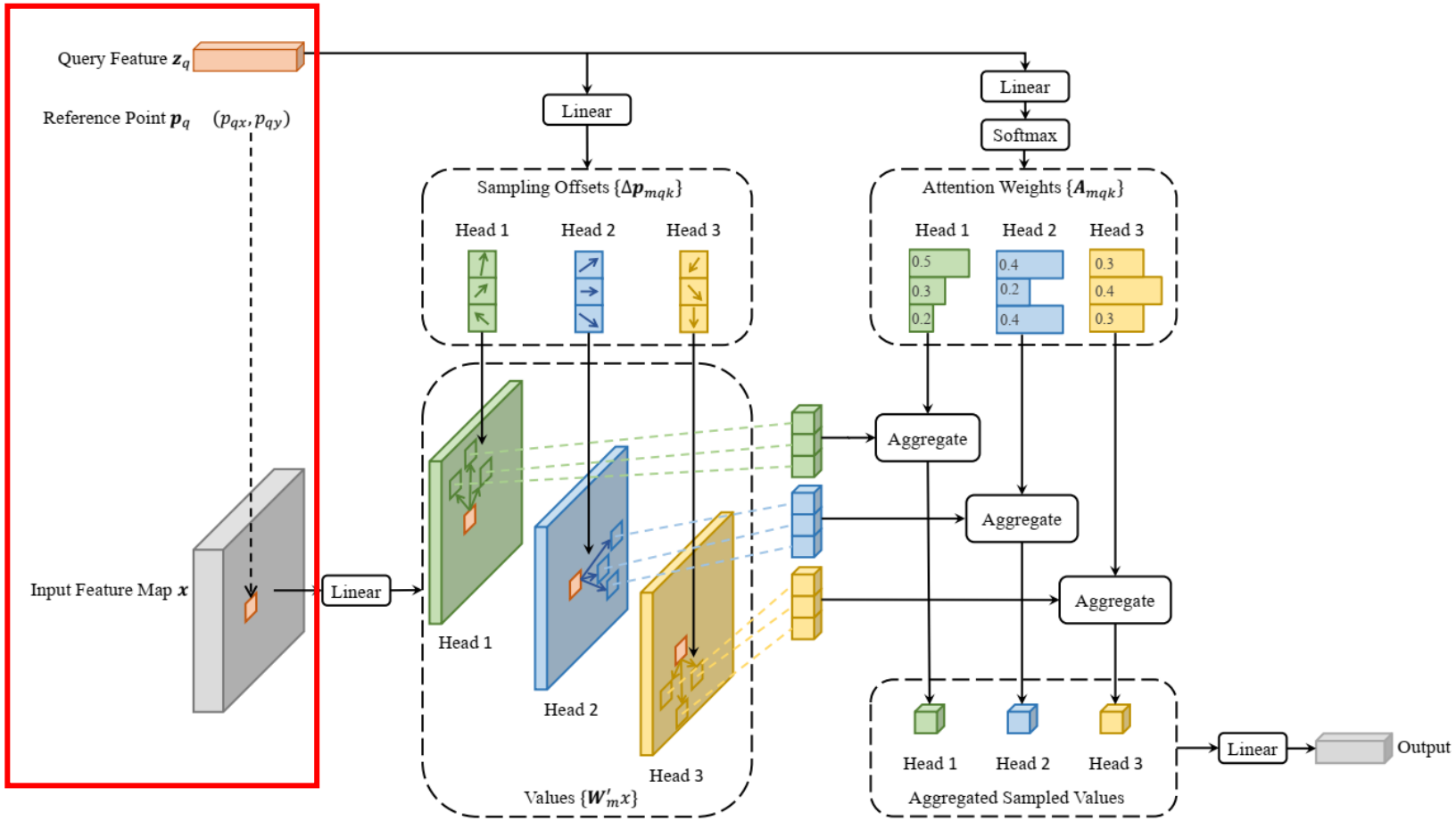
$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big], \tag{2}$$



Deformable Attention Module

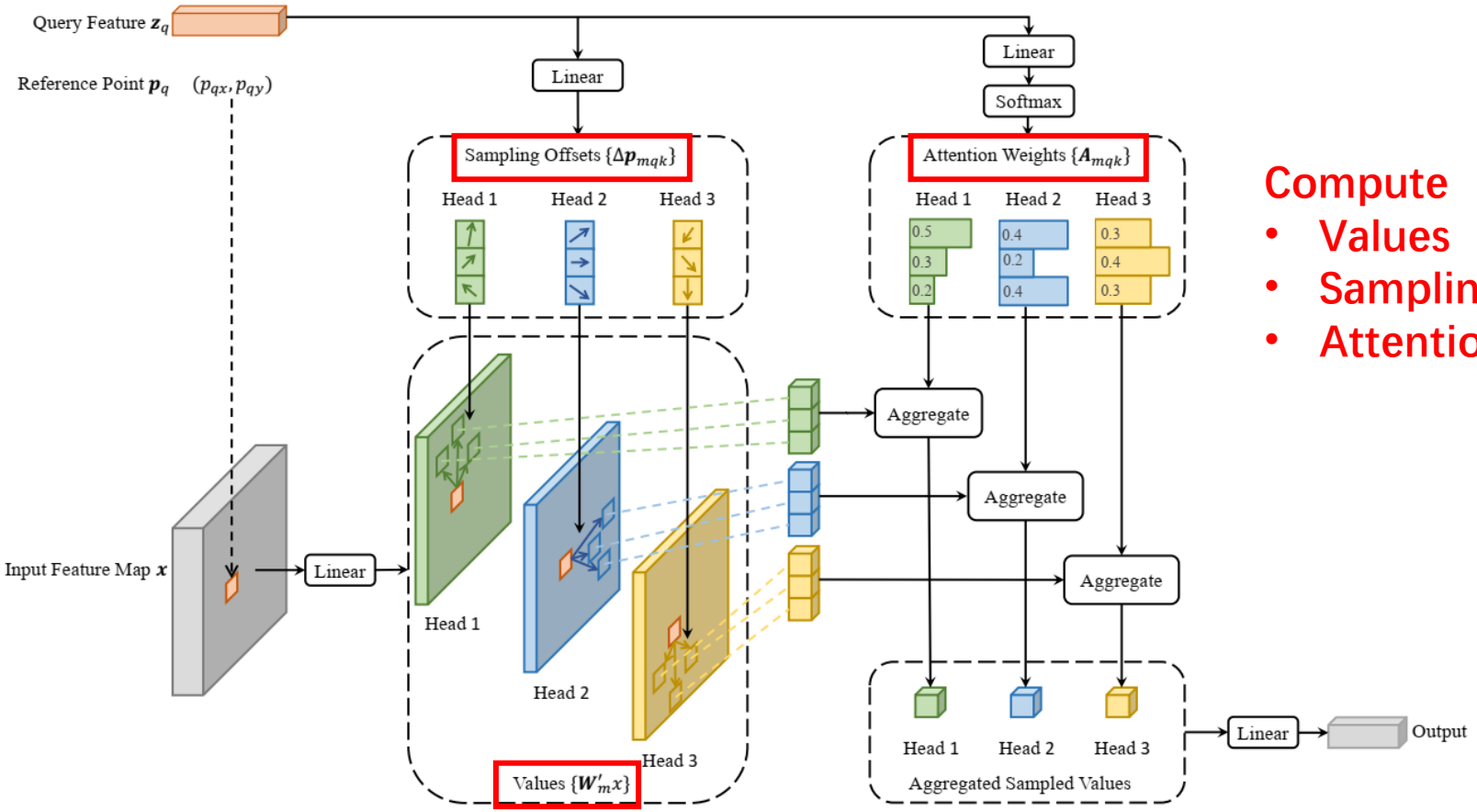
$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big], \qquad (2)$$

inputs



Deformable Attention Module

$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big], \qquad (2)$$

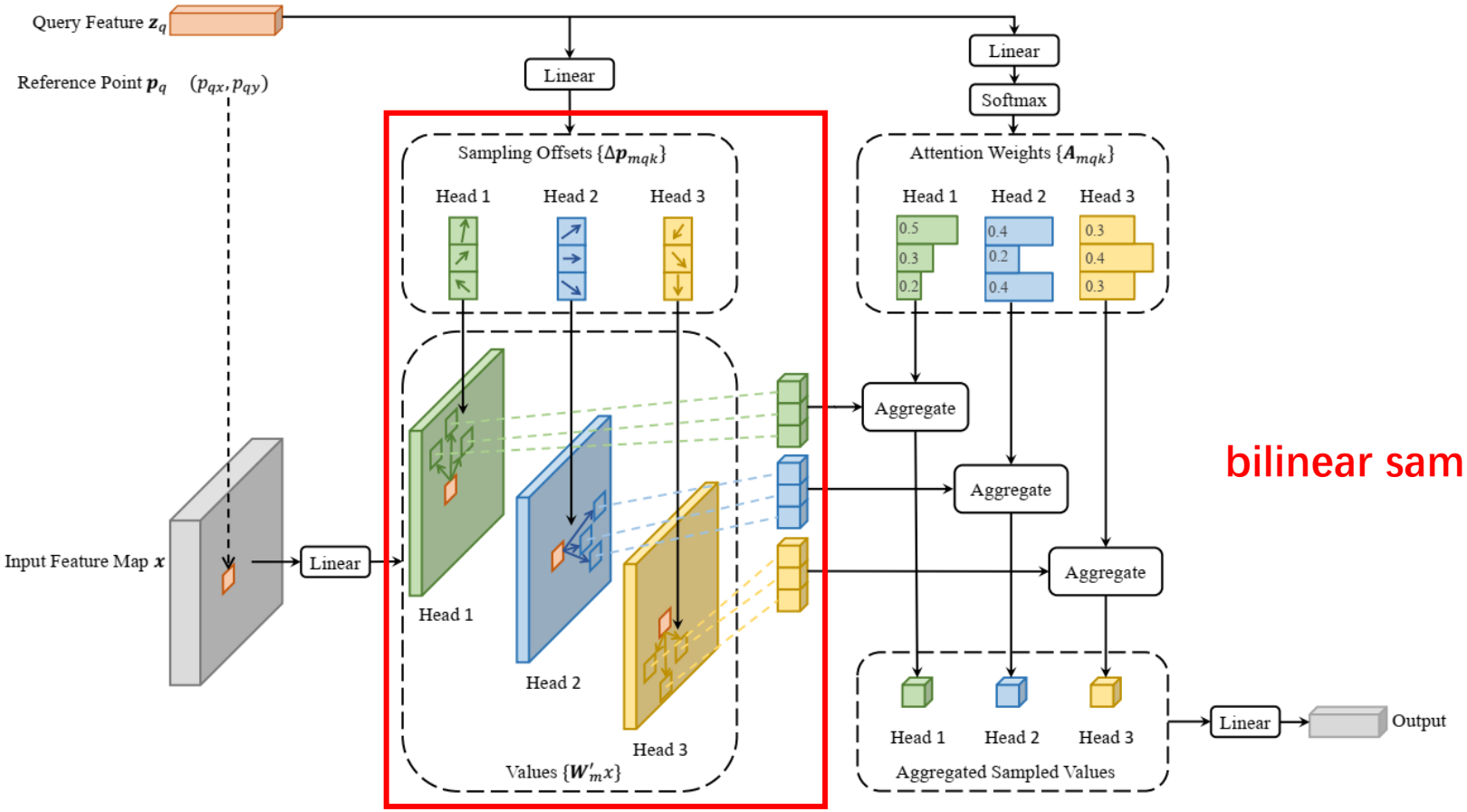


Compute

- **Values**
- **Sampling offsets**
- **Attention weights**

Deformable Attention Module

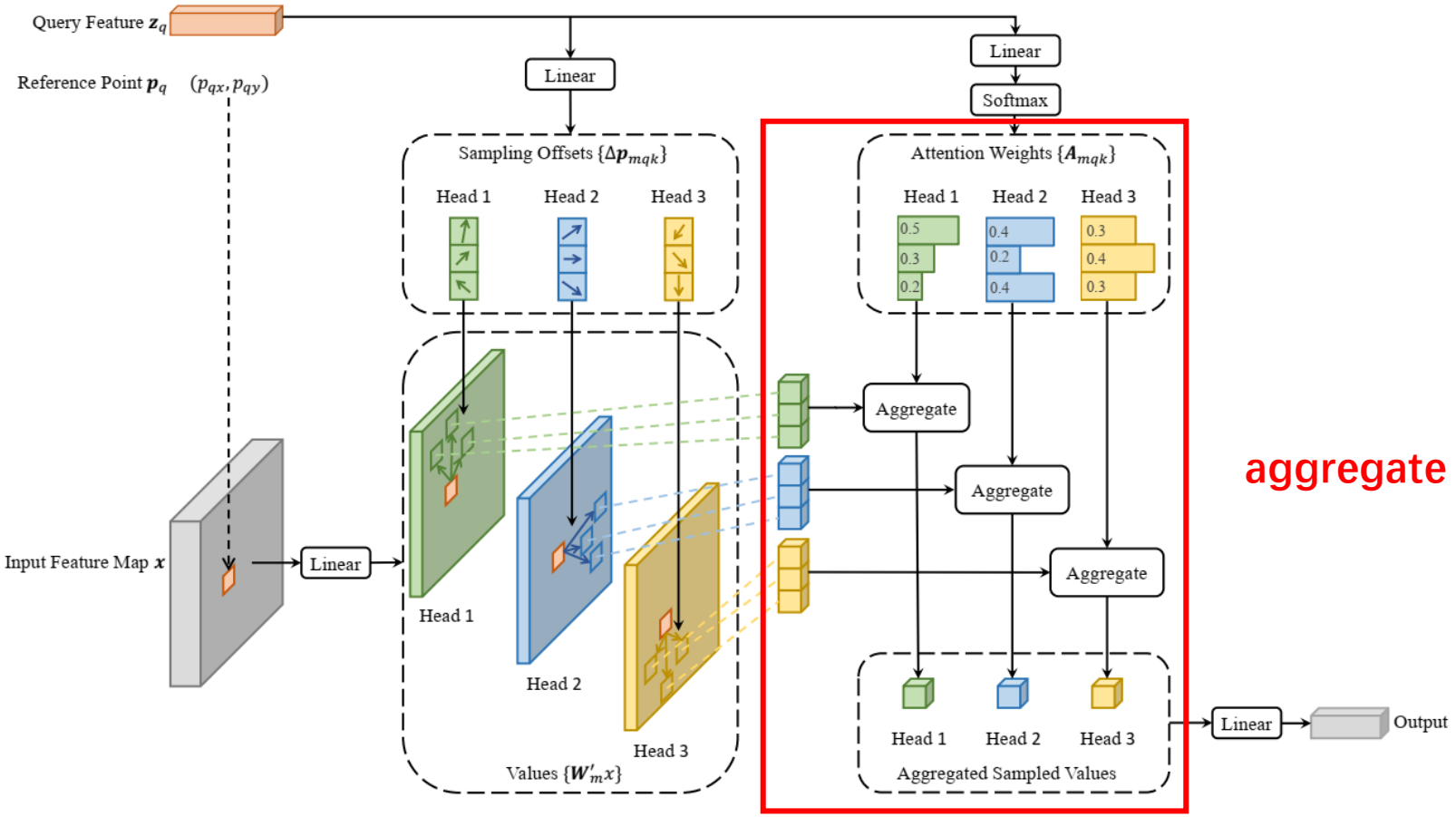
$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big], \qquad (2)$$



bilinear sampling of sparse values

Deformable Attention Module

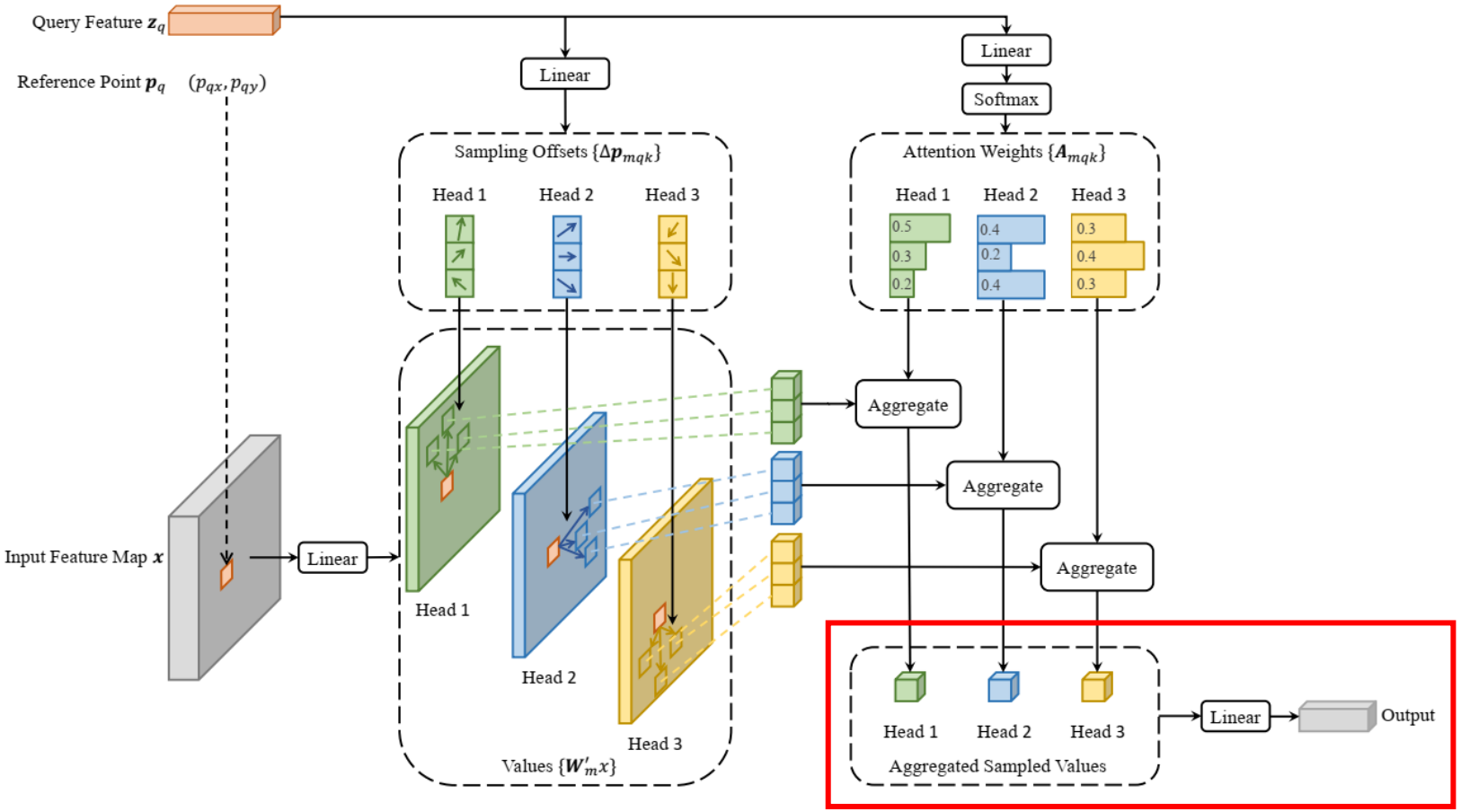
$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big], \qquad (2)$$



aggregate by attention weights

Deformable Attention Module

$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big], \qquad (2)$$



output projection

Multi-scale Deformable Attention Module

$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \Big], \tag{2}$$

$$\text{MSDeformAttn}(\boldsymbol{z}_q, \hat{\boldsymbol{p}}_q, \{\boldsymbol{x}^l\}_{l=1}^{L}) = \sum_{m=1}^{M} \boldsymbol{W}_m \Big[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}^l(\phi_l(\hat{\boldsymbol{p}}_q) + \Delta \boldsymbol{p}_{mlqk}) \Big], \tag{3}$$

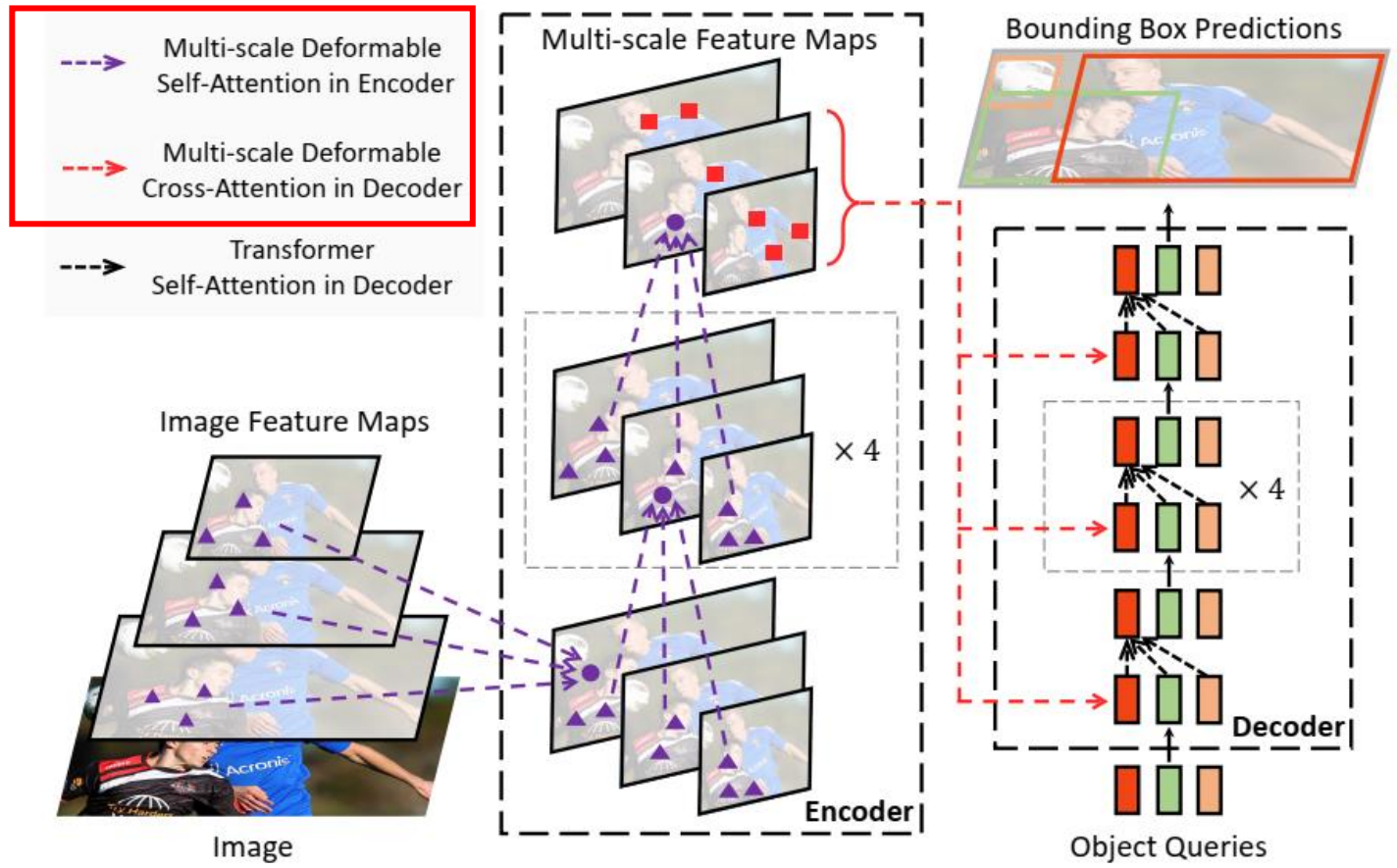
$\phi_l(\hat{\boldsymbol{p}}_q)$ normalized coordinate (ranged in [0, 1]) of $\hat{\boldsymbol{p}}_q$

single-scale feature map → multi-scale feature maps

K sampling points → $L \times K$ sampling points (i.e., K sampling points per feature level)

Deformable DETR object detector

Replace Transformer attention in DETR by multi-scale deformable attention while processing image feature maps



Experiment: Comparison with DETR

Table 1: Comparision of Deformable DETR with DETR on COCO 2017 val set. DETR-DC5+ denotes DETR-DC5 with Focal Loss and 300 object queries.

Method	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	params	FLOPs	Training GPU hours	Inference FPS
Faster R-CNN + FPN	109	42.0	62.1	45.5	26.6	45.4	53.4	42M	180G	380	26
DETR	500	42.0	62.4	44.2	20.5	45.8	61.1	41M	86G	2000	28
DETR-DC5	500	43.3	63.1	45.9	22.5	47.3	61.1	41M	187G	7000	12
DETR-DC5	50	35.3	55.7	36.8	15.2	37.5	53.6	41M	187G	700	12
DETR-DC5+	50	36.2	57.0	37.4	16.3	39.2	53.9	41M	187G	700	12
Deformable DETR	50	43.8	62.6	47.7	26.4	47.1	58.0	40M	173G	325	19
+ iterative bounding box refinement	50	45.4	64.7	49.0	26.8	48.3	61.7	40M	173G	325	19
++ two-stage Deformable DETR	50	46.2	65.2	50.0	28.8	49.2	61.7	40M	173G	340	19

10×

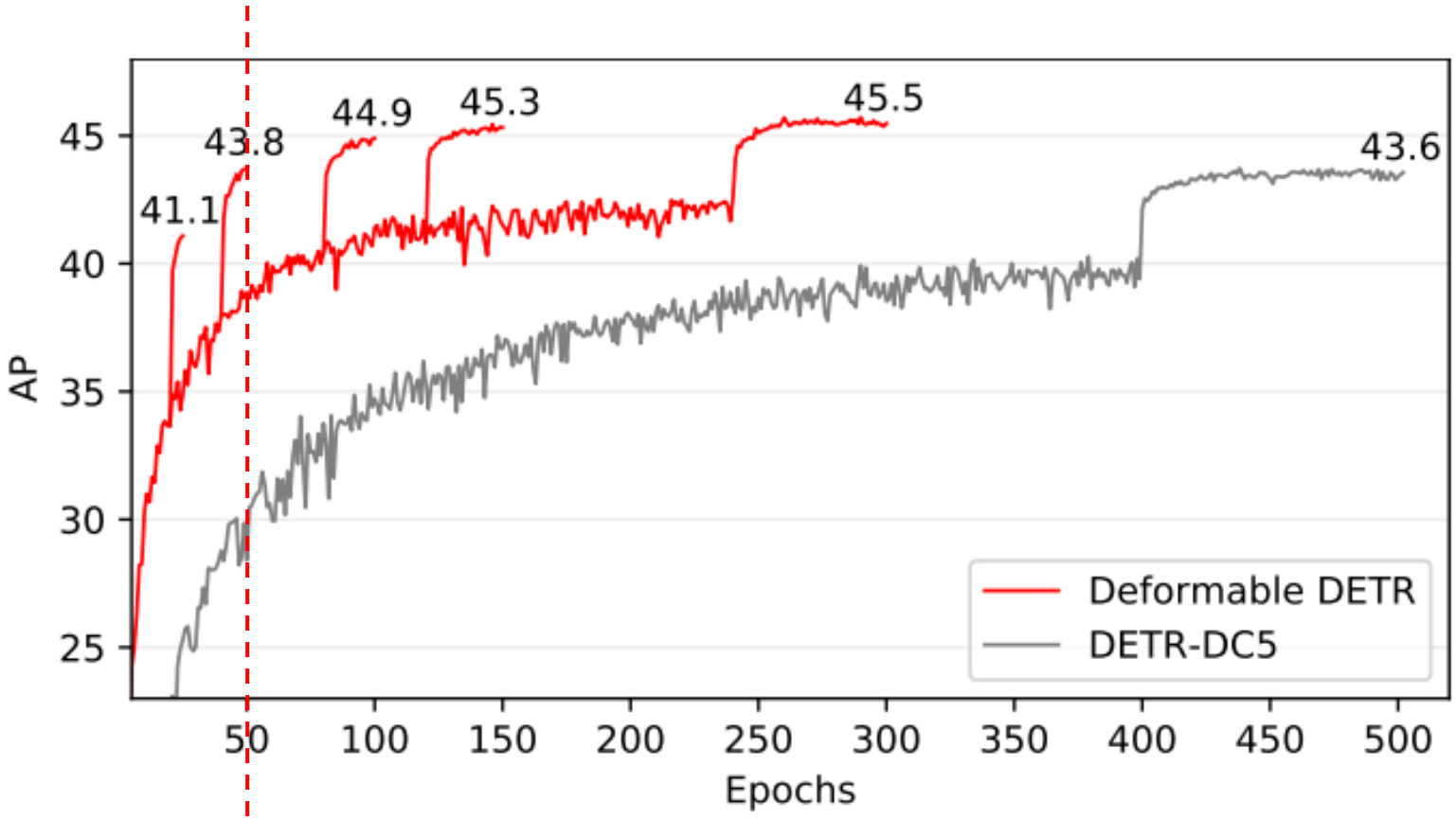
20×

1.6×

better performance (especially on small objects)

with 10x less training epoch, 20x less training time, 1.6x faster inference speed

Experiment: Convergence curve



Our default training schedule

Experiment: Comparison with SOTA methods

Table 3: Comparison of Deformable DETR with state-of-the-art methods on COCO 2017 test-dev set. "TTA" indicates test-time augmentations including horizontal flip and multi-scale testing.

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., 2020b)	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

Comparable with SOTA, the first high-performance end-to-end object detector

Conclusion

- Deformable DETR is an end-to-end object detector, which is efficient and fast-converging.
- Compared with DETR, Deformable DETR can achieve better performance (especially on small objects) with 10× less training epochs.
- It enables us to explore more interesting and practical variants of end-to-end object detectors.

- We hope our work opens up new possibilities in exploring end-to-end object detection.

code

paper

We are hiring @ SenseTime Research!

- Intern / Researcher / Developer / Ph.D Program
  (related to computer vision or deep learning)

- Email:

  Jifeng Dai : daijifeng@sensetime.com

  Xizhou Zhu : zhuwalter@sensetime.com