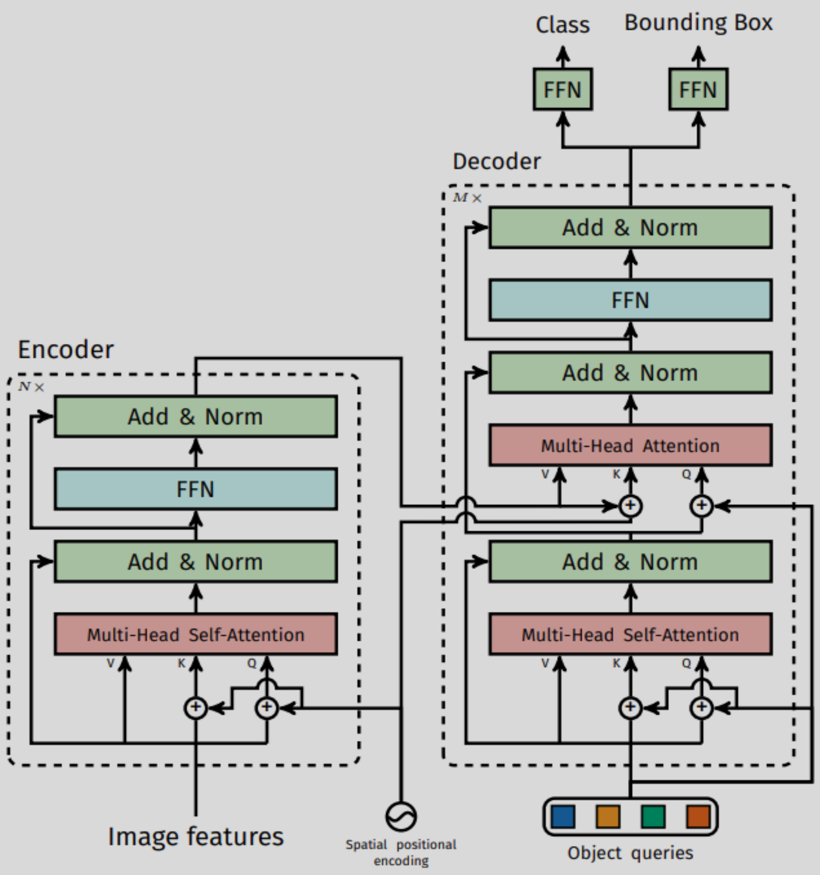
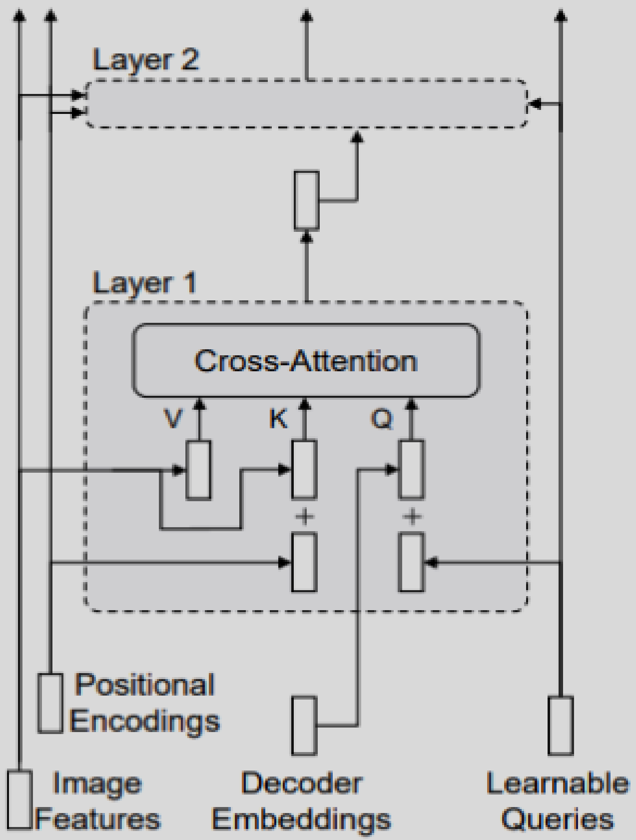
Computer Vision

Lecture 06: Object detection pipeline - 2

Deformable-DETR

DETR Transformer



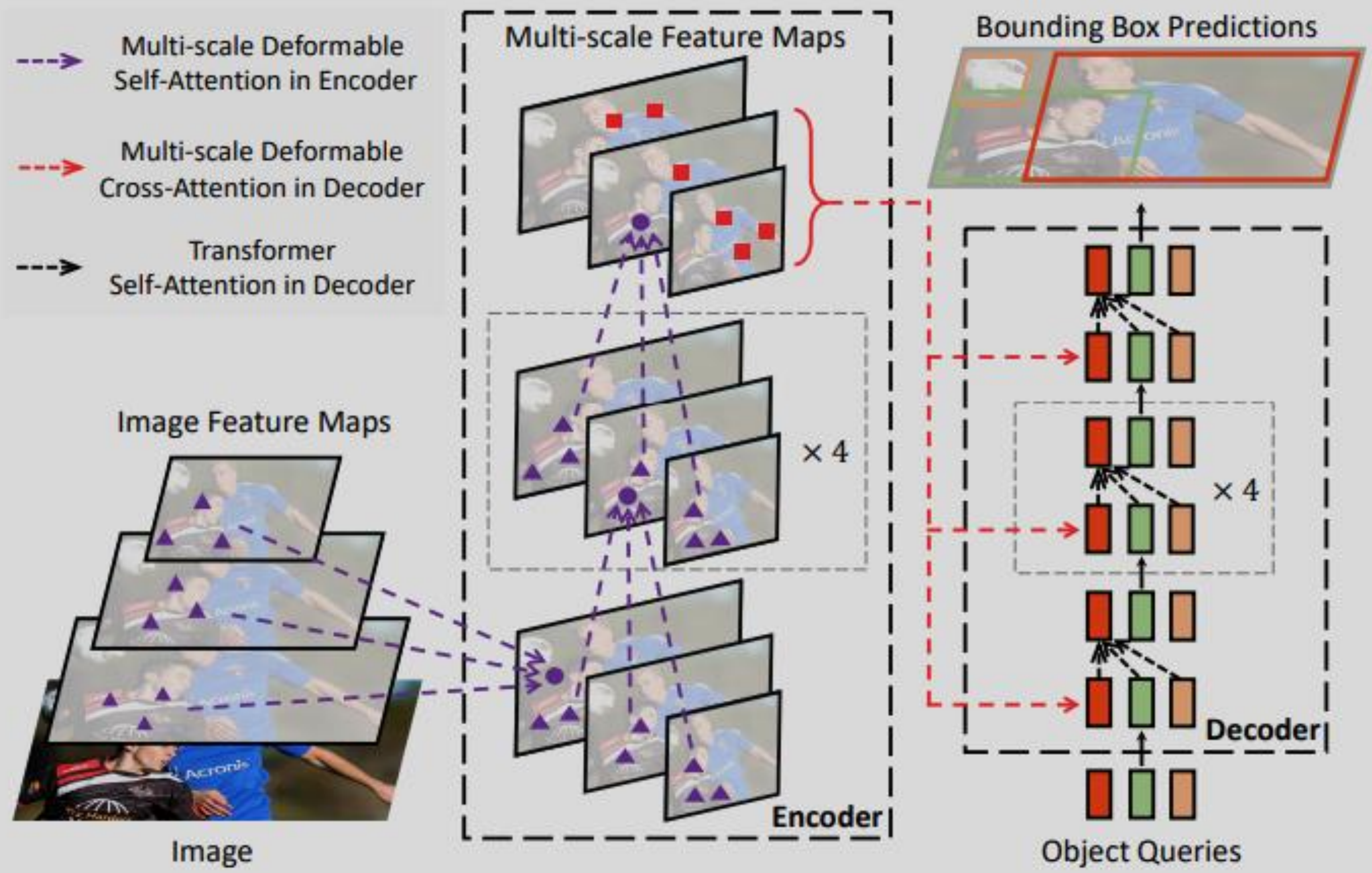


End-to-end object detection with Transformers, ECCV'20

Deformable-DETR

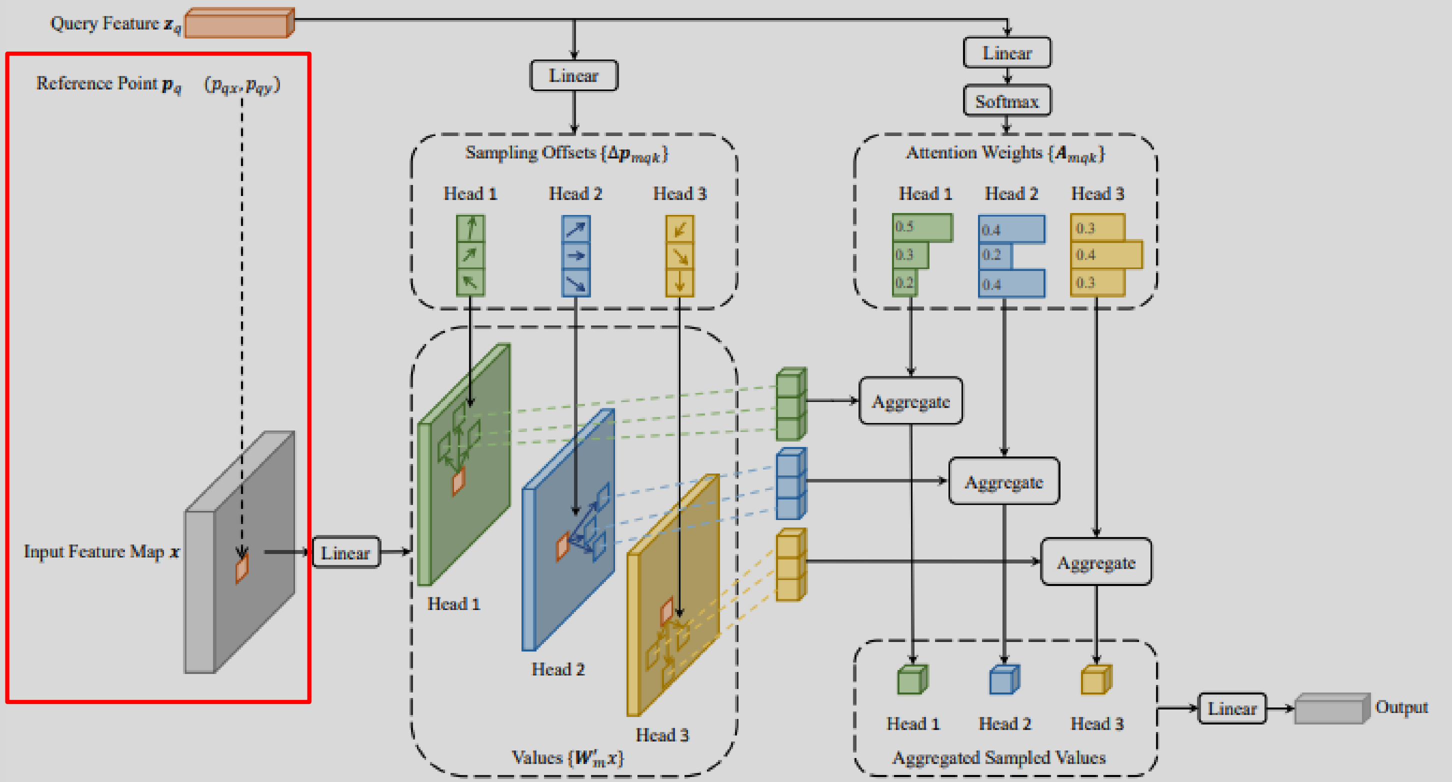
- **Limitations of DETR**
  - Slow convergence.
  - It takes 10-20 times slow compared to Faster-RCNN.
  - Scale-problem:
    - It cannot effectively detect small objects.
    - Recent detection models exhibit multi-scale encoders; while DETR already takes huge complexity for one scale. Thus, it is hard to use the DETR in multi-scales.
    - Attention mechanism requires quadratic computation w.r.t. pixel numbers.
  - Deformable attention is proposed to relieve the heavy complexity and slow convergence.

Deformable-DETR



Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

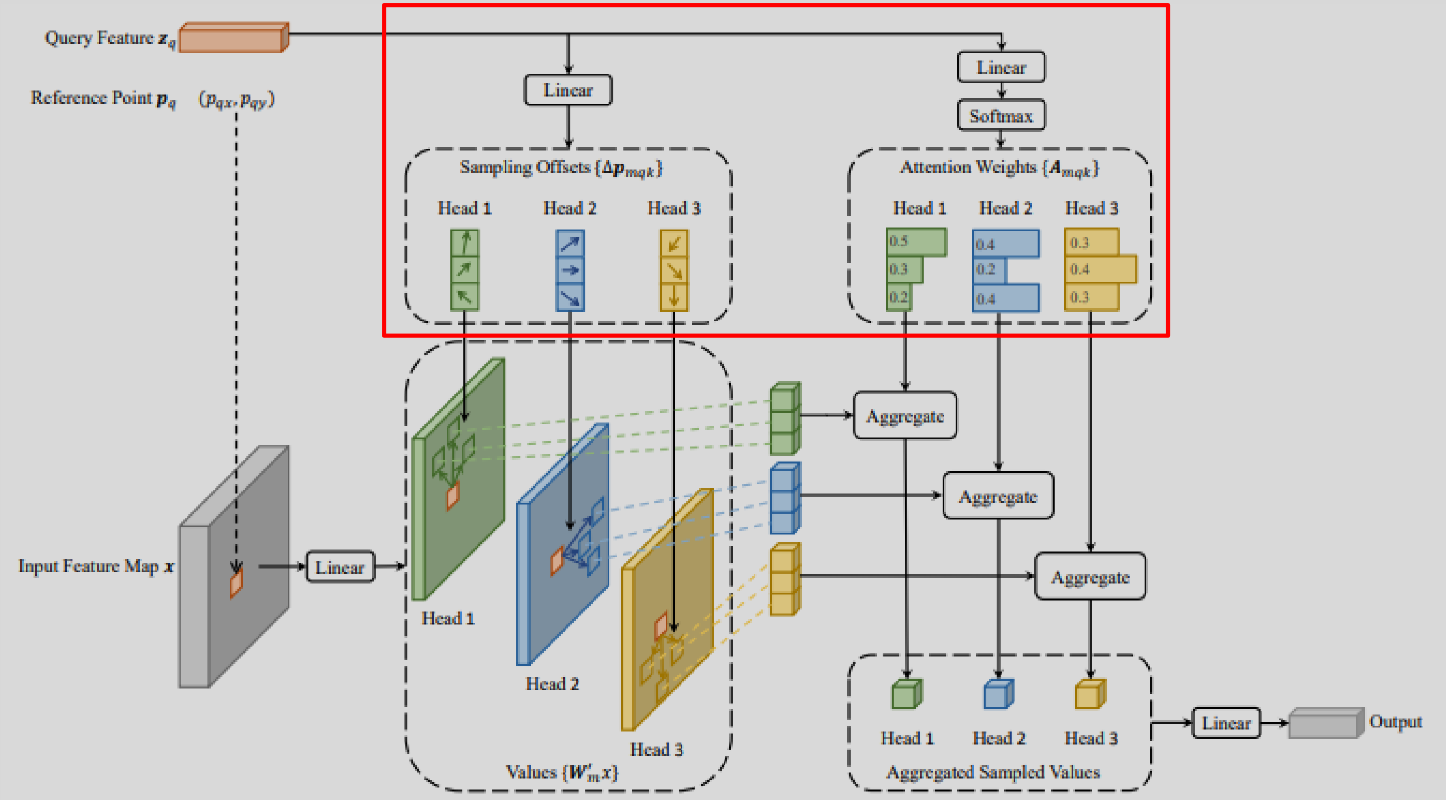
Deformable-DETR



1. Reference point is defined as: Coordinates of query pixel @ Encoder, Inferred via linear layer @ Decoder.

Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

Deformable-DETR

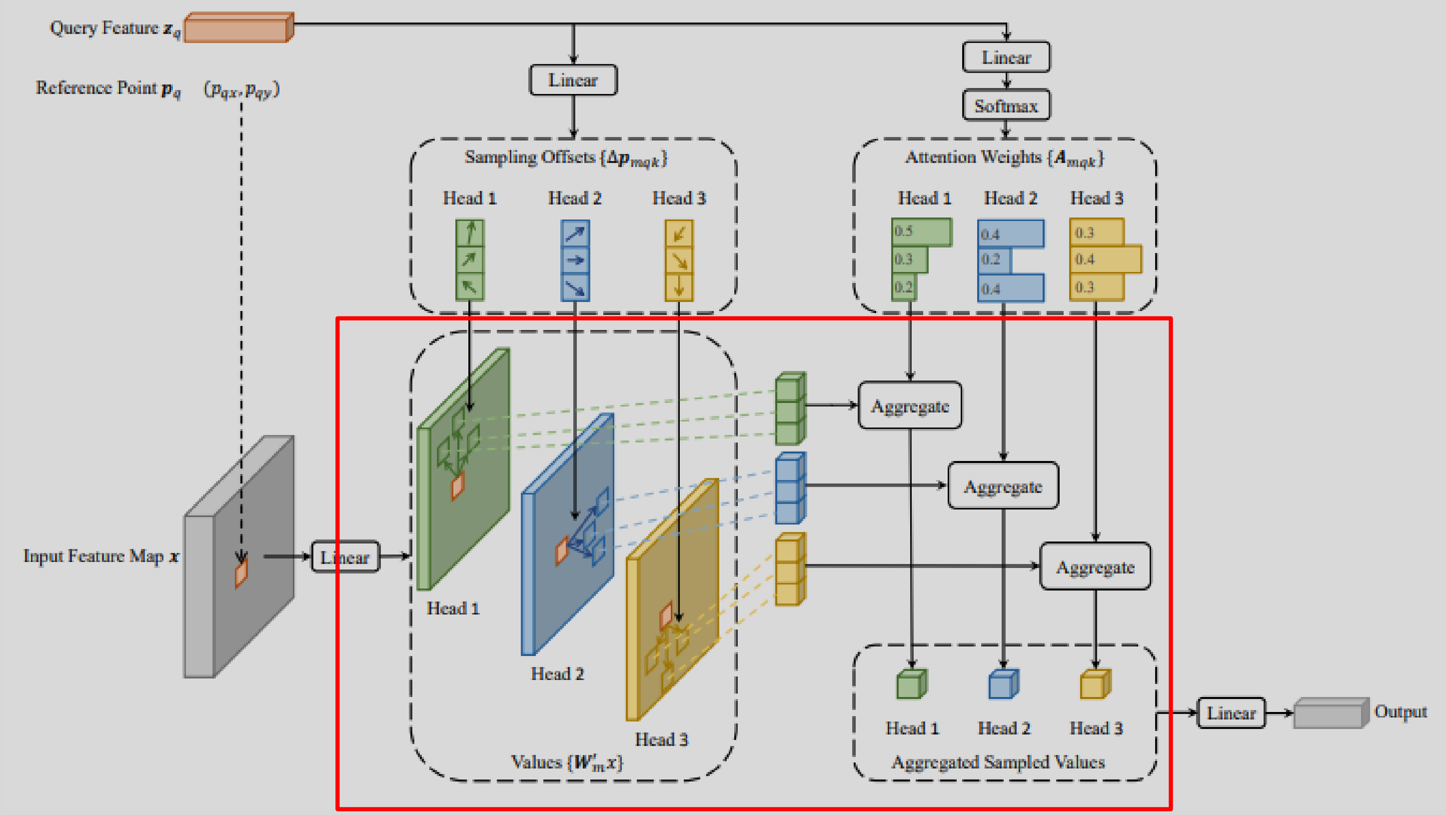


1. Reference point is defined as:
   Coordinates of query pixel @ Encoder,
   Inferred via linear layer @ Decoder.

2. Predict sampling offsets and attention weights.

Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

Deformable-DETR



1. Reference point is defined as:
   Coordinates of query pixel @ Encoder,
   Inferred via linear layer @ Decoder.

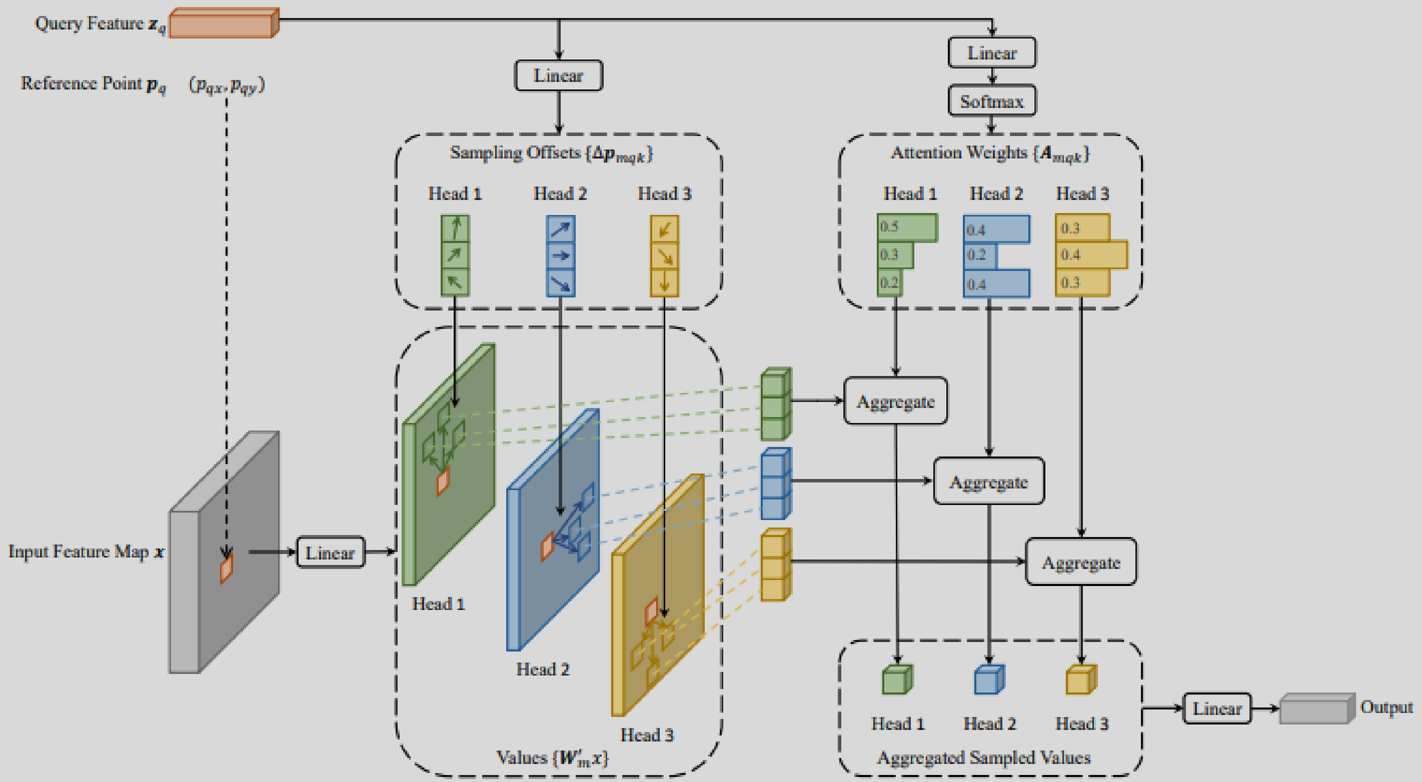
2. Predict sampling offsets and attention weights.

3. Attention is obtained for offset points of the reference points.

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

Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

Deformable-DETR



1. Reference point is defined as:
   Coordinates of query pixel @ Encoder,
   Inferred via linear layer @ Decoder.

2. Predict sampling offsets and attention weights.

3. Attention is obtained for offset points of the reference points.

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

4. Predicted bounding boxes have relative coordinates to the reference points.

$$\hat{\boldsymbol{b}}_q = \{\sigma\big(b_{qx} + \sigma^{-1}(\hat{p}_{qx})\big), \sigma\big(b_{qy} + \sigma^{-1}(\hat{p}_{qy})\big), \sigma(b_{qw}), \sigma(b_{qh})\}$$

Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

Deformable-DETR

https://wikidocs.net/31379

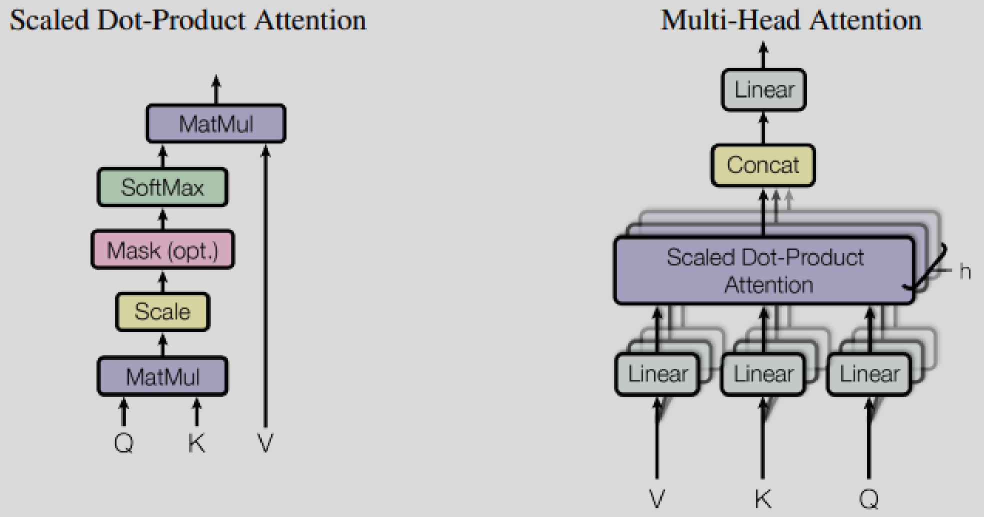


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\begin{aligned}\mathrm{MultiHead}(Q,K,V) &= \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O \\ \text{where } \mathrm{head_i} &= \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{aligned}$$

$$\mathrm{MultiHeadAttn}(\boldsymbol{z}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{k \in \Omega_k} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}_k \right]$$

Deformable-DETR

$$\text{MultiHeadAttn}(\boldsymbol{z}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \big[\sum_{k \in \Omega_k} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}_k \big]$$

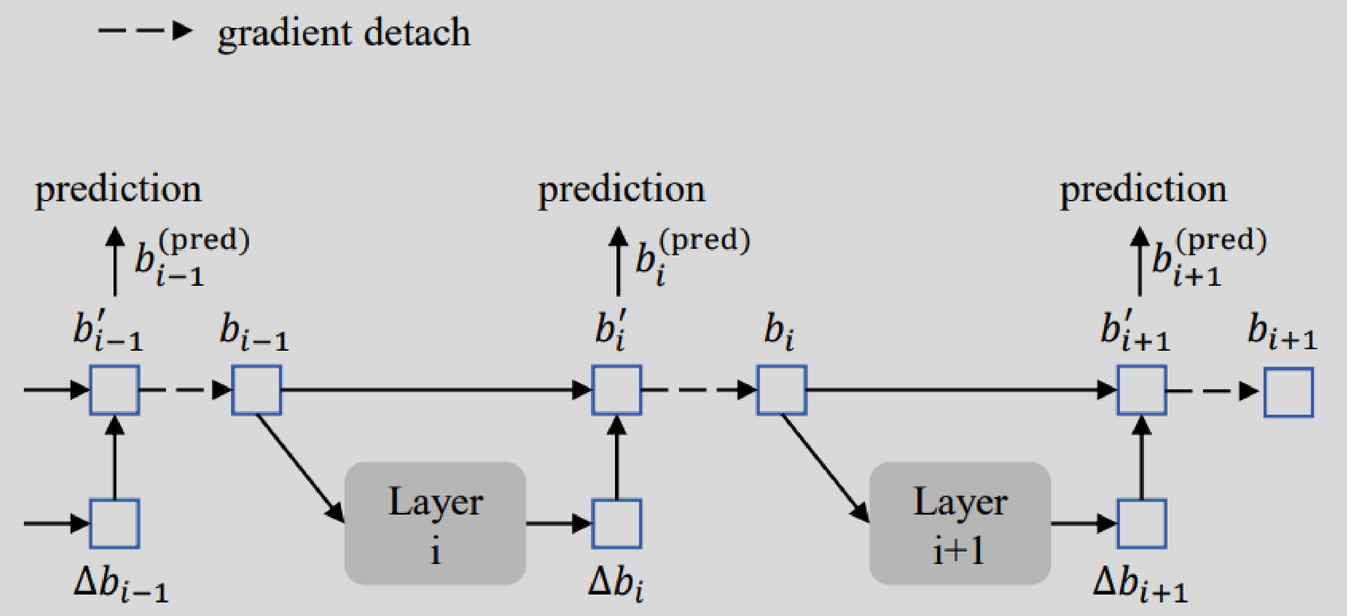
$$\text{DeformAttn}(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \big[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}) \big]$$

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

Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

Deformable-DETR

Iterative Bounding Box Refinement.



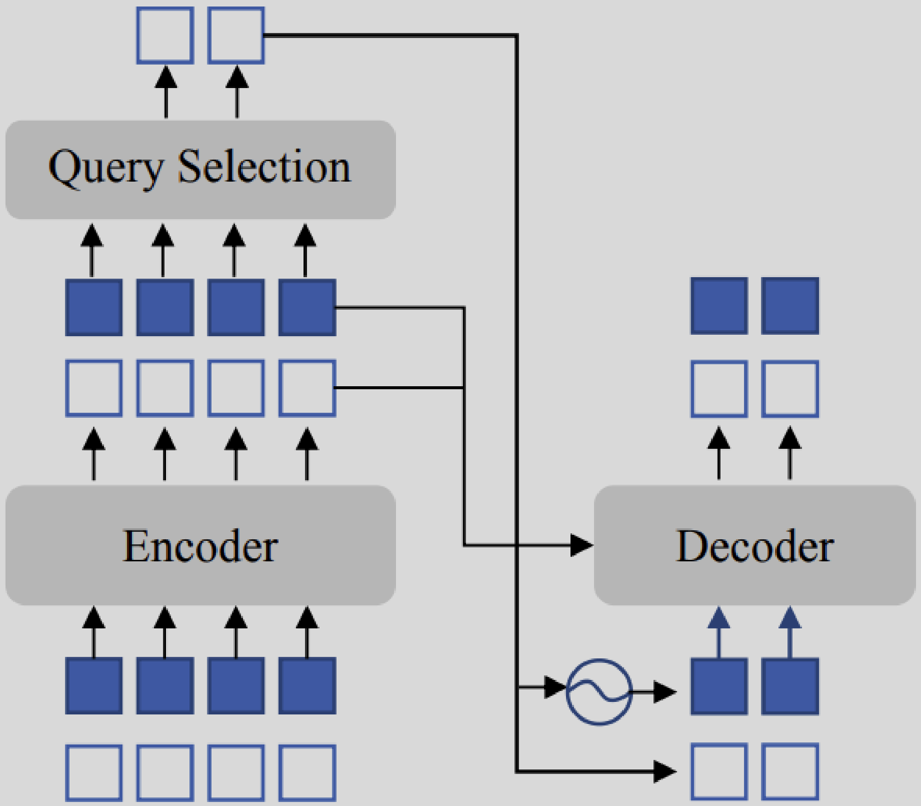
- Decoder layers refine bounding boxes based on previous decoder layer outputs.

$$b_i = \{\sigma(\Delta b_i^x + \sigma^{-1}(b_{i-1}^x)), \sigma(\Delta b_i^y + \sigma^{-1}(b_{i-1}^y)), \\ \sigma(\Delta b_i^w + \sigma^{-1}(b_{i-1}^w)), \sigma(\Delta b_i^h + \sigma^{-1}(b_{i-1}^h))\}$$

Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

Deformable-DETR

Two-Stage Deformable DETR.



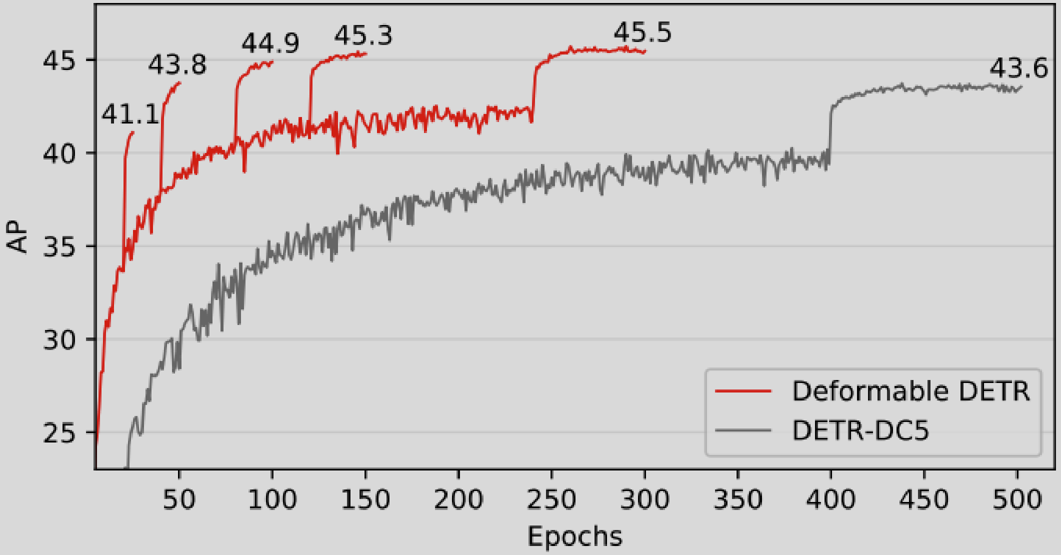
- Object queries are irrelevant to the input images.
- Motivated by the two-stage object detectors such as Faster-RCNN, region proposals are generated.
- Generated region proposals are further used as the input to the Decoder input.

Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

Deformable-DETR

Performance

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



Deformable DETR: Deformable Transformers for End-to-End Object Detection, ICLR'21.

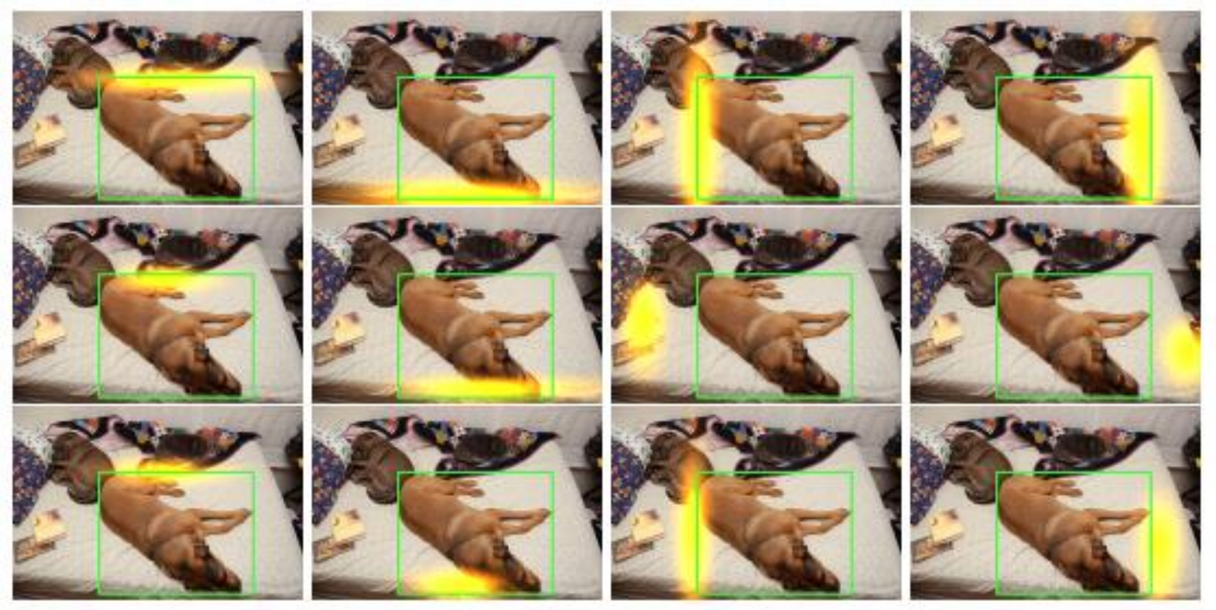
Conditional DETR

Problem: Slow convergence speed of DETRs

Conditional DETR

DETR (50 epoch)

DETR (500 epoch)



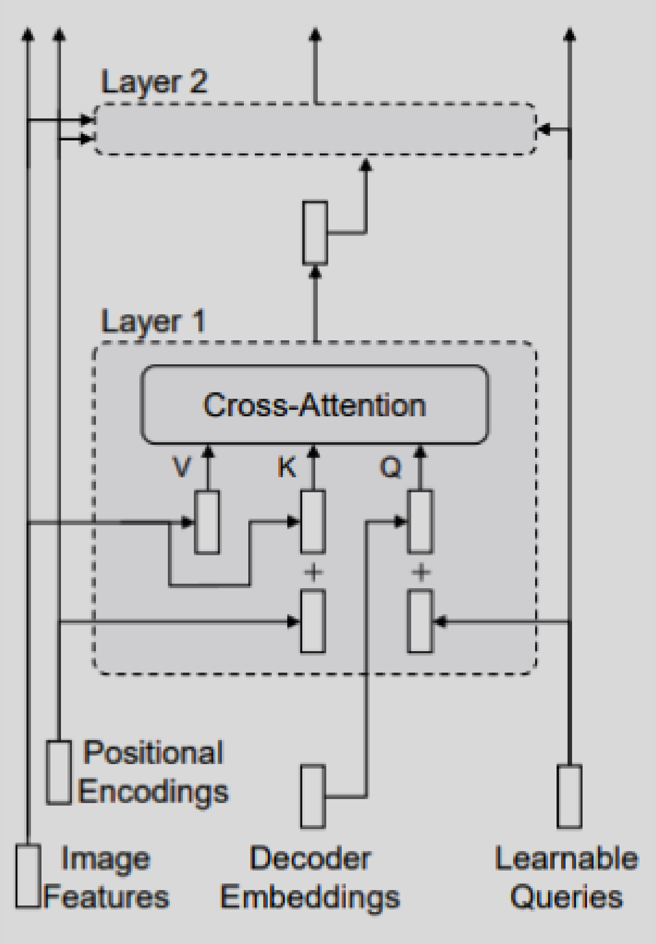
Object queries are not using the image specific information; while offering attention weight map.

It is hard to map queries to the spatial keys and it causes the slow convergence of DETRs.

Conditional DETR for Fast Training Convergence, ICCV 2021

Conditional DETR

Two-Stage Deformable DETR.



DETR

$$
\begin{aligned}
&(\mathbf{c}_q + \mathbf{p}_q)^\top (\mathbf{c}_k + \mathbf{p}_k) \\
=\ & \mathbf{c}_q^\top \mathbf{c}_k + \mathbf{c}_q^\top \mathbf{p}_k + \mathbf{p}_q^\top \mathbf{c}_k + \mathbf{p}_q^\top \mathbf{p}_k \\
=\ & \mathbf{c}_q^\top \mathbf{c}_k + \mathbf{c}_q^\top \mathbf{p}_k + \mathbf{o}_q^\top \mathbf{c}_k + \mathbf{o}_q^\top \mathbf{p}_k.
\end{aligned}
$$

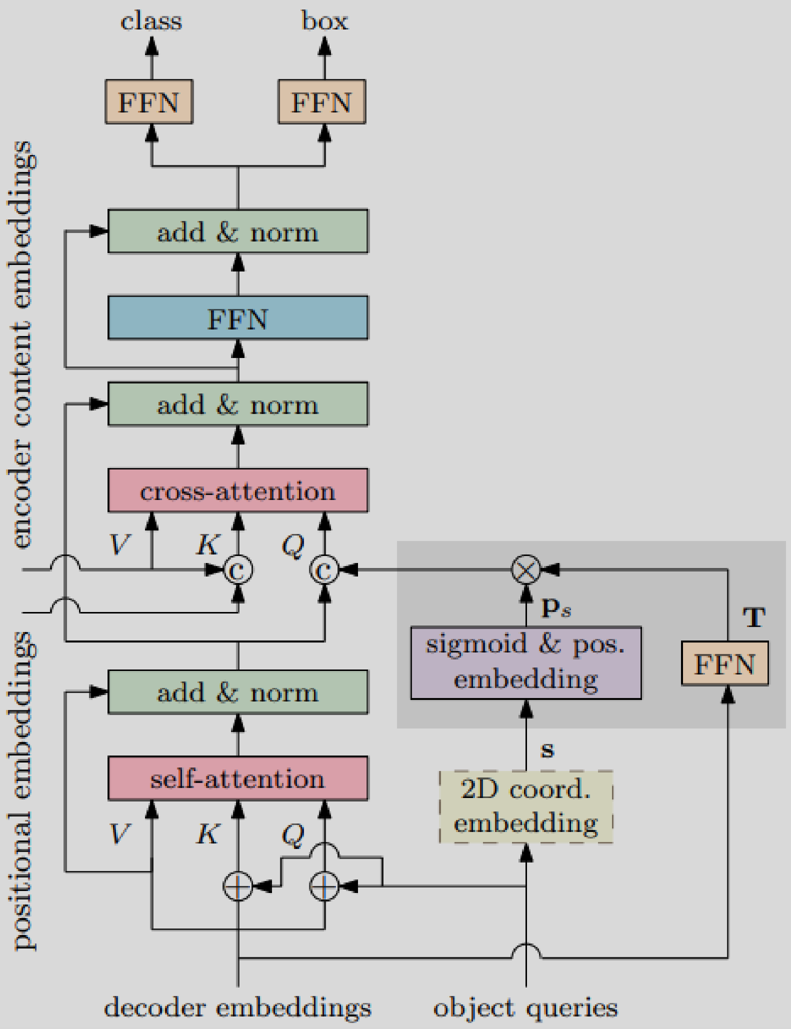
$\longrightarrow$

Conditional DETR

$$\mathbf{c}_q^\top \mathbf{c}_k + \mathbf{p}_q^\top \mathbf{p}_k.$$

Conditional DETR for Fast Training Convergence, ICCV 2021

Conditional DETR



$$(\mathbf{s}, \mathbf{f}) \rightarrow \mathbf{p}_q,$$

s: reference point obtained from object queries.
f: decoder embedding

$$\mathbf{p}_s = \text{sinusoidal}(\text{sigmoid}(\mathbf{s})).$$

To make it aligned with positional space of the key.

$$\mathbf{p}_q = \mathbf{T}\mathbf{p}_s = \boldsymbol{\lambda}_q \odot \mathbf{p}_s.$$

T : Diagonal matrix, this transforms the reference point to the embedding space.

$\boldsymbol{\lambda}_q$: element of T.

Conditional DETR for Fast Training Convergence, ICCV 2021

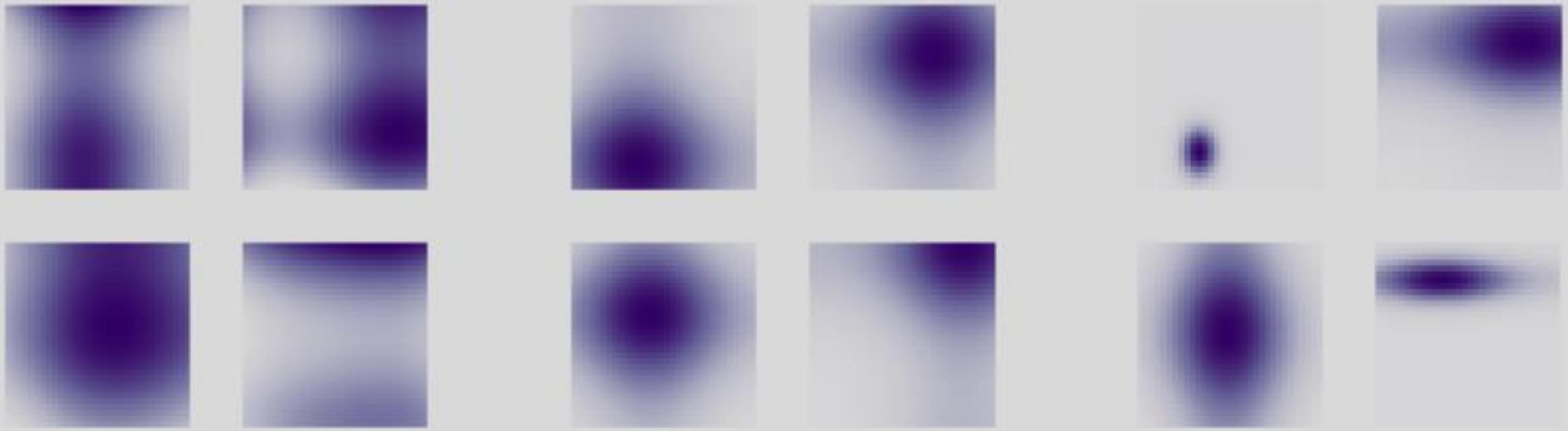
Conditional DETR

Model	#epochs	GFLOPs	#params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR-R50	500	86	41	42.0	62.4	44.2	20.5	45.8	61.1
DETR-R50	50	86	41	34.9	55.5	36.0	14.4	37.2	54.5
Conditional DETR-R50	50	90	44	40.9	61.8	43.3	20.8	44.6	59.2
Conditional DETR-R50	75	90	44	42.1	62.9	44.8	21.6	45.4	60.2
Conditional DETR-R50	108	90	44	43.0	64.0	45.7	22.7	46.7	61.5
DETR-DC5-R50	500	187	41	43.3	63.1	45.9	22.5	47.3	61.1
DETR-DC5-R50	50	187	41	36.7	57.6	38.2	15.4	39.8	56.3
Conditional DETR-DC5-R50	50	195	44	43.8	64.4	46.7	24.0	47.6	60.7
Conditional DETR-DC5-R50	75	195	44	44.5	65.2	47.3	24.4	48.1	62.1
Conditional DETR-DC5-R50	108	195	44	45.1	65.4	48.5	25.3	49.0	62.2
DETR-R101	500	152	60	43.5	63.8	46.4	21.9	48.0	61.8
DETR-R101	50	152	60	36.9	57.8	38.6	15.5	40.6	55.6
Conditional DETR-R101	50	156	63	42.8	63.7	46.0	21.7	46.6	60.9
Conditional DETR-R101	75	156	63	43.7	64.9	46.8	23.3	48.0	61.7
Conditional DETR-R101	108	156	63	44.5	65.6	47.5	23.6	48.4	63.6
DETR-DC5-R101	500	253	60	44.9	64.7	47.7	23.7	49.5	62.3
DETR-DC5-R101	50	253	60	38.6	59.7	40.7	17.2	42.2	57.4
Conditional DETR-DC5-R101	50	262	63	45.0	65.5	48.4	26.1	48.9	62.8
Conditional DETR-DC5-R101	75	262	63	45.6	66.5	48.8	25.5	49.7	63.3
Conditional DETR-DC5-R101	108	262	63	45.9	66.8	49.5	27.2	50.3	63.3
Other single-scale DETR variants									
Deformable DETR-R50-SS*	50	78	34	39.4	59.6	42.3	20.6	43.0	55.5
UP-DETR-R50 [5]	150	86	41	40.5	60.8	42.6	19.0	44.4	60.0
UP-DETR-R50 [5]	300	86	41	42.8	63.0	45.3	20.8	47.1	61.7
Deformable DETR-DC5-R50-SS*	50	128	34	41.5	61.8	44.9	24.1	45.3	56.0

Conditional DETR for Fast Training Convergence, ICCV 2021

DAB-DETR

Visualization of attentions of positional queries and positional keys.

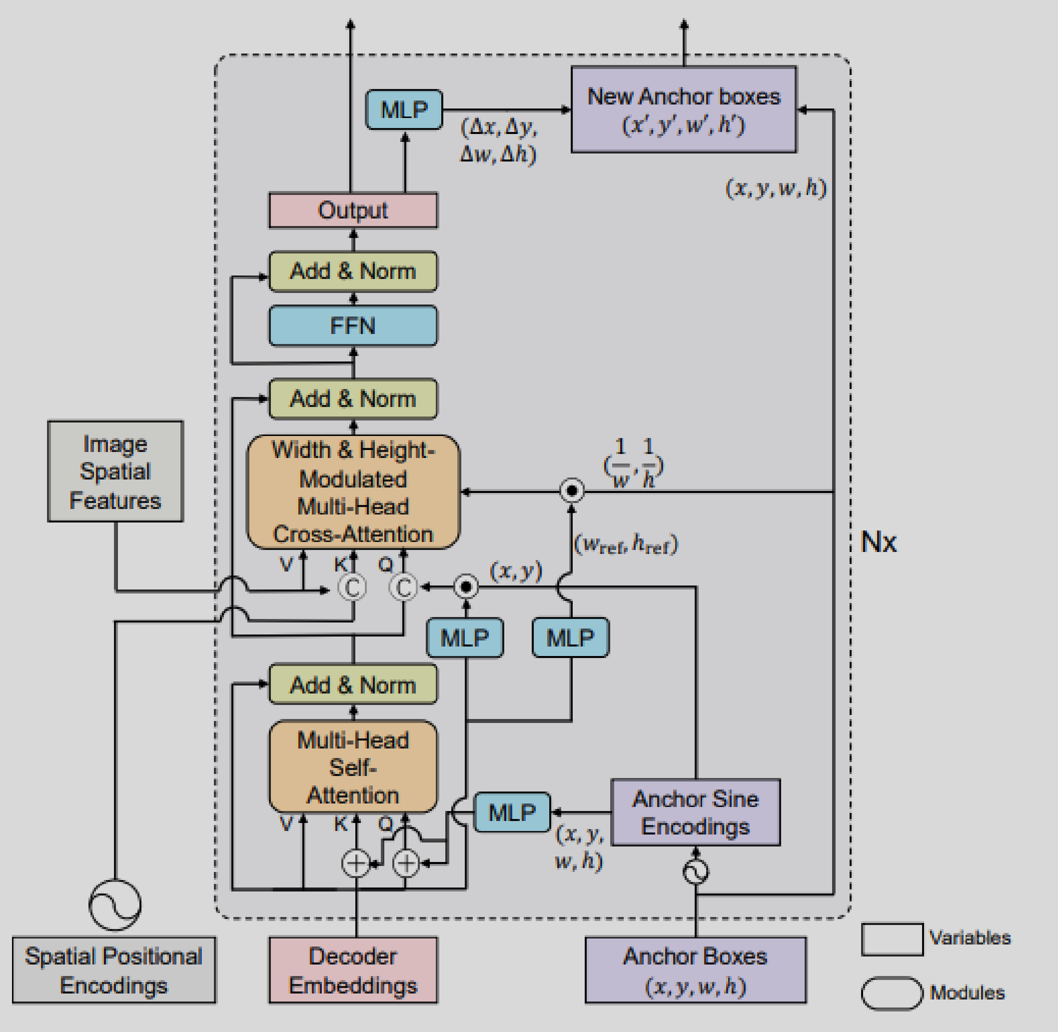


(a) DETR (b) Conditional DETR (c) DAB-DETR

(a) Unconcentrated and too small/too big attentions.
(b) Gaussian-like attentions, object scale is not considered.
(c) Sharp attentions.

DAB-DETR: Dynamic Anchor Boxes are better queries for DETR, ICLR 2022

DAB-DETR



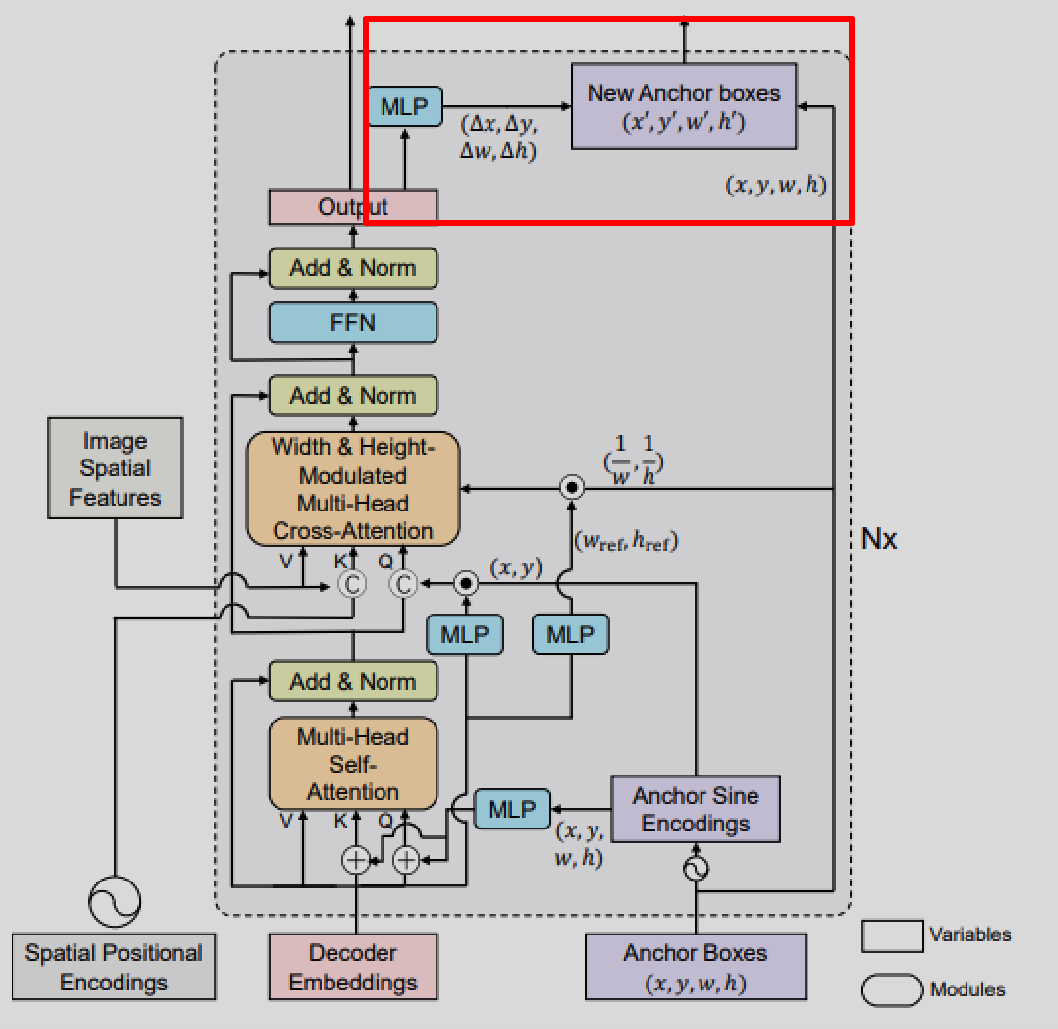
q-th anchor $A_q = (x_q, y_q, w_q, h_q)$

$$\mathrm{PE}(A_q) = \mathrm{PE}(x_q, y_q, w_q, h_q) = \mathrm{Cat}(\mathrm{PE}(x_q), \mathrm{PE}(y_q), \mathrm{PE}(w_q), \mathrm{PE}(h_q)).$$

$\mathrm{PE}: \mathbb{R} \to \mathbb{R}^{D/2}$: Positional encoding function that generates sinusoidal embedding from float numbers.

DAB-DETR: Dynamic Anchor Boxes are better queries for DETR, ICLR 2022

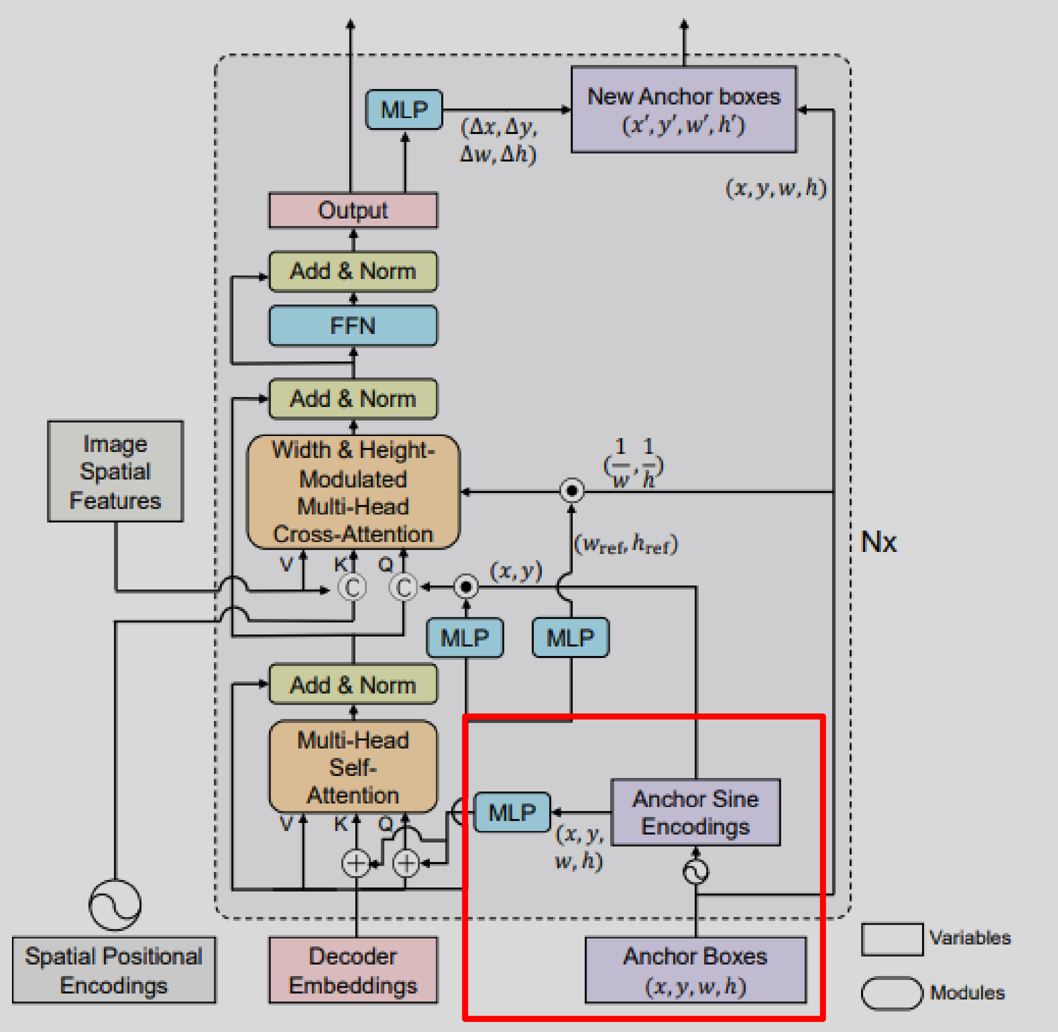
DAB-DETR



- DETR, Conditional DETR -> High-dimensional query
- No direct relationship between query values and bounding box locations.
- Directly used bounding box values as the queries.

DAB-DETR: Dynamic Anchor Boxes are better queries for DETR, ICLR 2022

DAB-DETR



positional query $P_q = \mathrm{MLP}(\mathrm{PE}(A_q)),$

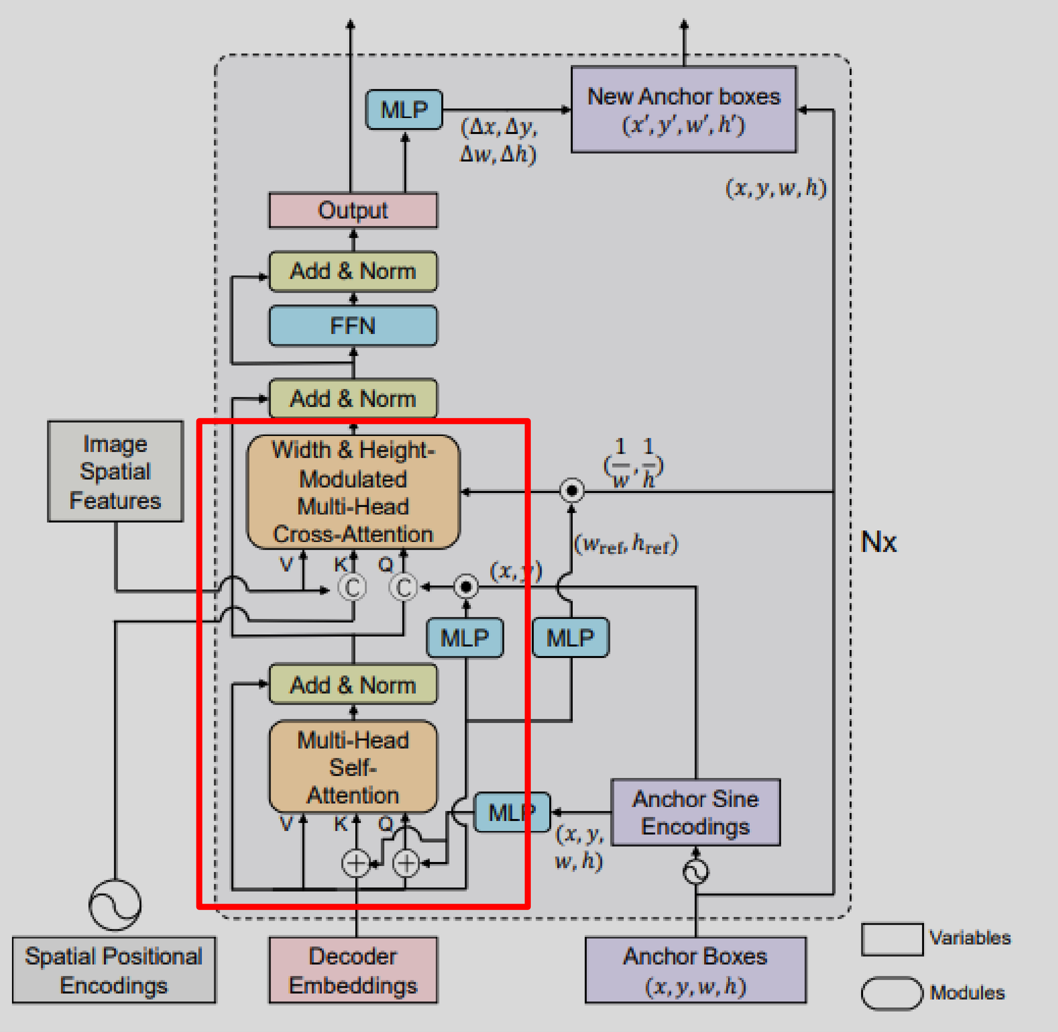
q-th anchor $A_q = (x_q, y_q, w_q, h_q)$

$$\mathrm{PE}(A_q) = \mathrm{PE}(x_q, y_q, w_q, h_q) = \mathrm{Cat}(\mathrm{PE}(x_q), \mathrm{PE}(y_q), \mathrm{PE}(w_q), \mathrm{PE}(h_q)).$$

$\mathrm{PE}: \mathbb{R} \to \mathbb{R}^{D/2}$: Positional encoding function that generates sinusoidal embedding from float numbers.

DAB-DETR: Dynamic Anchor Boxes are better queries for DETR, ICLR 2022

DAB-DETR



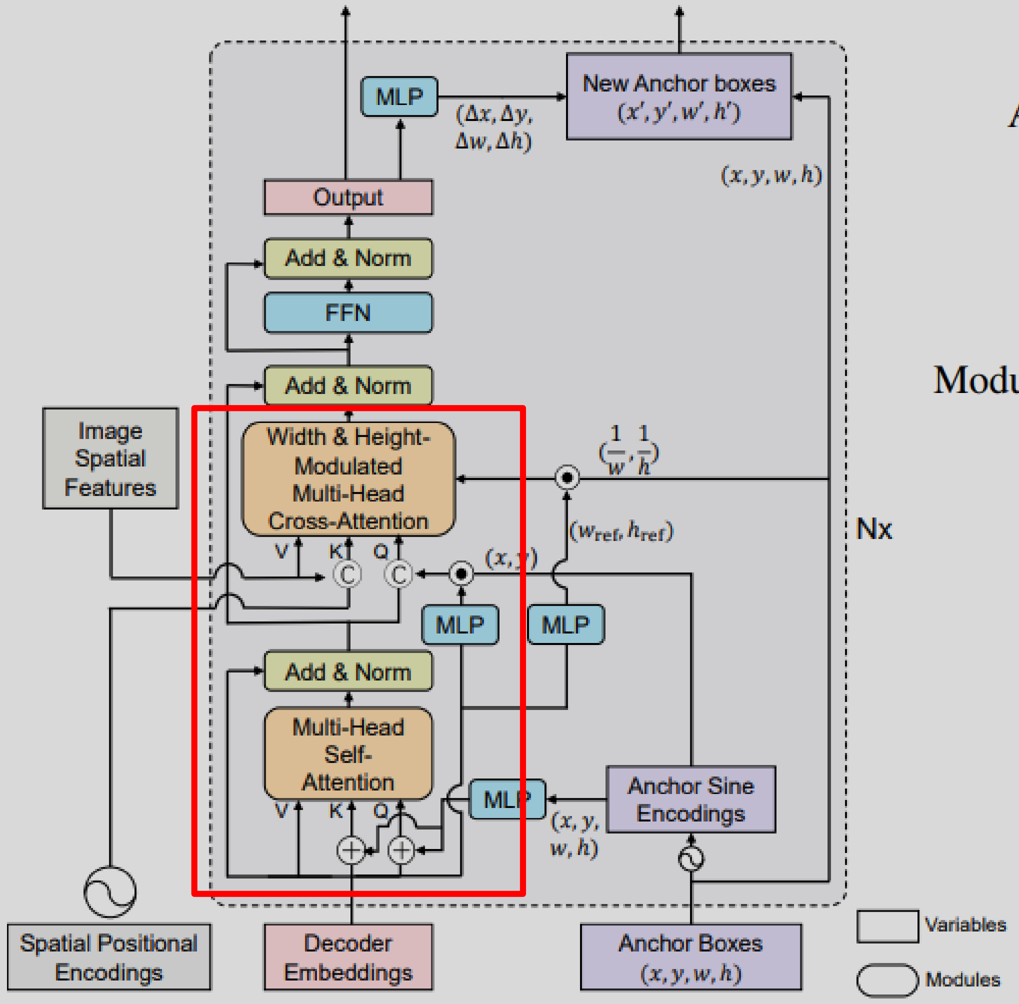
Self-Attn: $Q_q = C_q + P_q, \quad K_q = C_q + P_q, \quad V_q = C_q,$

Cross-Attn: $$Q_q = \mathrm{Cat}(C_q, \mathrm{PE}(x_q, y_q) \cdot \mathrm{MLP}^{(\mathrm{csq})}(C_q)),$$
$$K_{x,y} = \mathrm{Cat}(F_{x,y}, \mathrm{PE}(x, y)), \quad V_{x,y} = F_{x,y},$$

-> $\mathrm{MLP}^{(\mathrm{csq})}$ is used similarly to Conditional DETR.

DAB-DETR: Dynamic Anchor Boxes are better queries for DETR, ICLR 2022

DAB-DETR



Original positional attention map

$$\mathrm{Attn}((x,y),(x_{\text{ref}},y_{\text{ref}})) = (\mathrm{PE}(x)\cdot\mathrm{PE}(x_{\text{ref}}) + \mathrm{PE}(y)\cdot\mathrm{PE}(y_{\text{ref}}))/\sqrt{D},$$

Modulate positional attention maps

$$\mathrm{ModulateAttn}((x,y),(x_{\text{ref}},y_{\text{ref}})) = (\mathrm{PE}(x)\cdot\mathrm{PE}(x_{\text{ref}})\frac{w_{q,\text{ref}}}{w_q} + \mathrm{PE}(y)\cdot\mathrm{PE}(y_{\text{ref}})\frac{h_{q,\text{ref}}}{h_q})/\sqrt{D},$$

$$w_{q,\text{ref}}, h_{q,\text{ref}} = \sigma(\mathrm{MLP}(C_q)).$$

Temperature Tuning

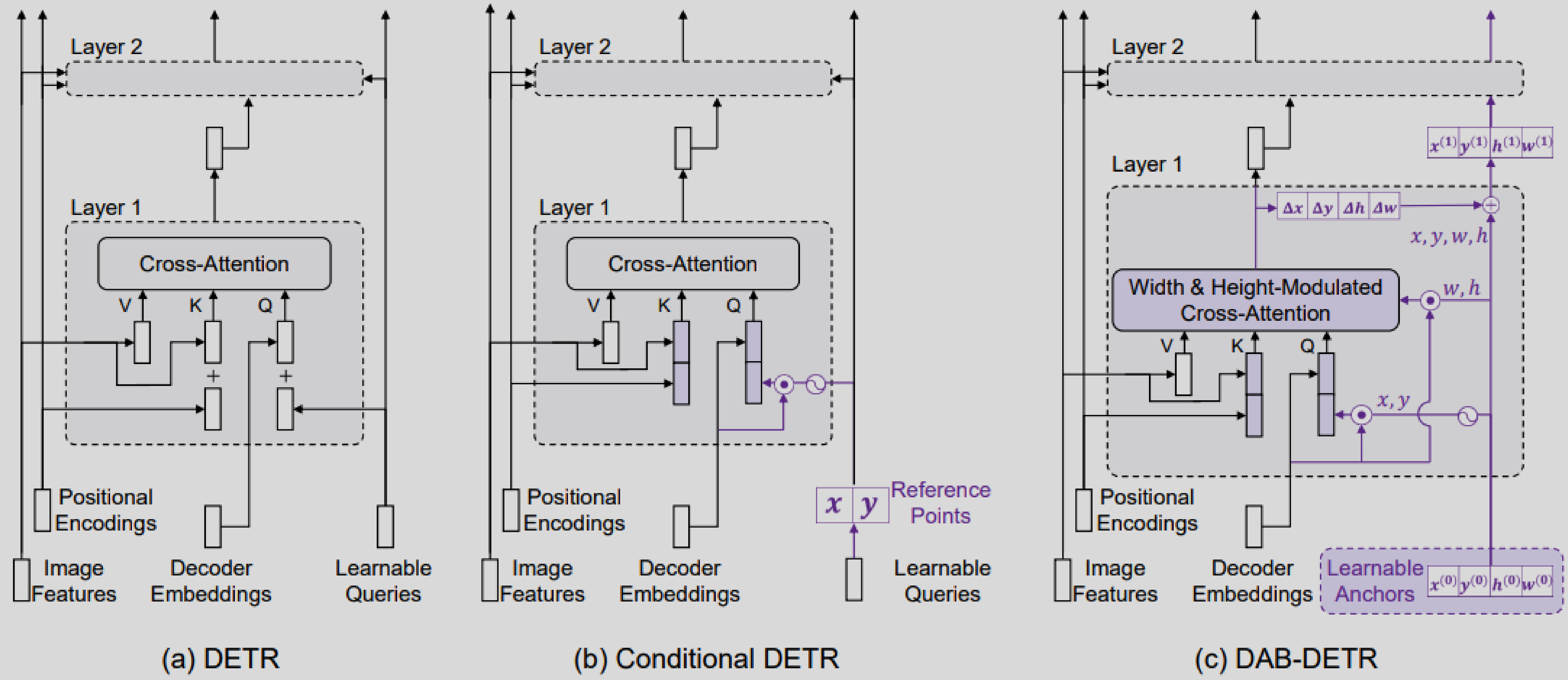
$$\mathrm{PE}(x)_{2i} = \sin\left(\frac{x}{T^{2i/D}}\right), \quad \mathrm{PE}(x)_{2i+1} = \cos\left(\frac{x}{T^{2i/D}}\right),$$

It affects the size of positional priors.

H=1, W=1 H=1, W=3 H=3, W=1

Temperature=1 Temperature=10 Temperature=10000

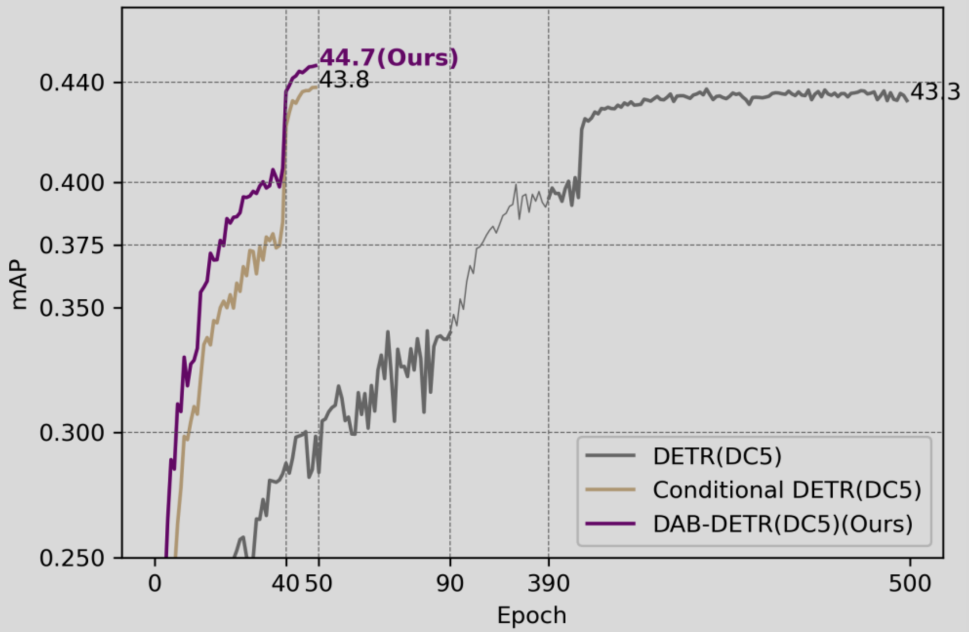
DAB-DETR



DAB-DETR: Dynamic Anchor Boxes are better queries for DETR, ICLR 2022

DAB-DETR

# row	Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params
1	Deformable DETR	43.8	62.6	47.7	26.4	47.1	58.0	40M
2	Deformable DETR+	45.4	64.7	49.0	26.8	48.3	61.7	40M
3	Deformable DETR+ (open source)	46.3	65.3	50.2	28.6	49.3	62.1	47M
4	DAB-Deformable-DETR(Ours)	**46.8**	**66.0**	**50.4**	**29.1**	**49.8**	**62.3**	47M



DAB-DETR: Dynamic Anchor Boxes are better queries for DETR, ICLR 2022

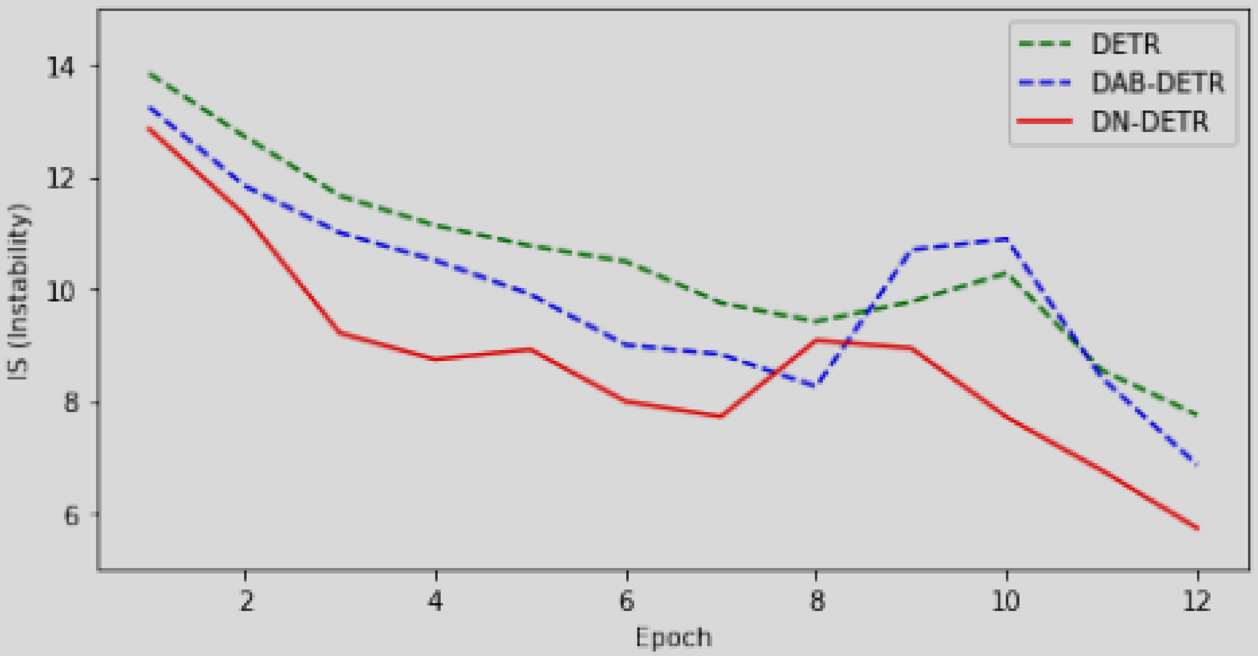
DN-DETR

- Previous works focused on improving the decoder and its queries.
- Slow convergence issue is caused by unstable bipartite graph matching in the early stage.
- For the same image, queries unstably matched with different objects in different epochs; thus the training goes unstable.
- DN-DETR proposed the query denoising method that stabilizes the bipartite graph matching during the training process.
- It can be easily applicable to DETR-like methods.

DN-DETR: Accelerate DETR Training by Introducing Query DeNoising, CVPR 2022

DN-DETR

- Training process of DETR-like models -> two stage : learning “good anchor” and learning “relative offset”
- Learning offsets could be difficult when “good anchor“ is not clear.
- It makes relative offset learning works better as denoising task can bypass the bipartite matching. Noised query is similar to GT anchor and it can be regarded as good anchor.



prediction $\mathbf{O^i} = \{O_0^i, O_1^i, ..., O_{N-1}^i\}$

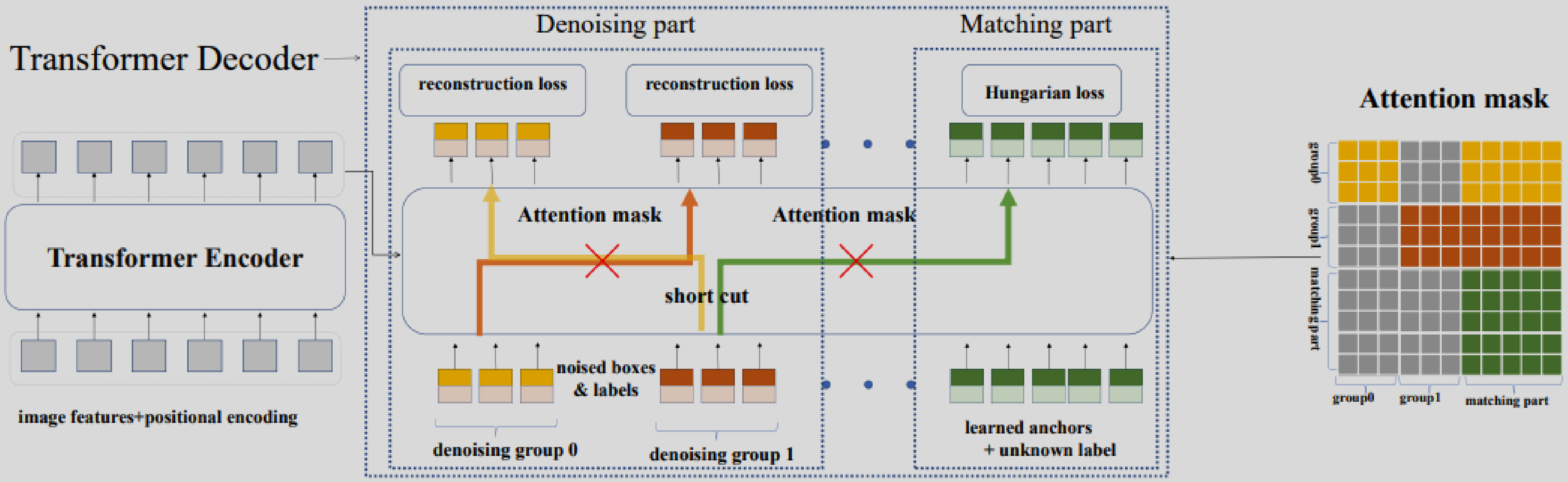
GT $\mathbf{T} = \{T_0, T_1, T_2, ..., T_{M-1}\}$

$$V_n^i = \begin{cases} m, & \text{if } O_n^i \text{ matches } T_m \\ -1, & \text{if } O_n^i \text{ matches nothing} \end{cases}$$

$$IS^i = \sum_{j=0}^{N} \mathbb{1}(V_n^i \neq V_n^{i-1})$$

DN-DETR: Accelerate DETR Training by Introducing Query DeNoising, CVPR 2022

DN-DETR



- Decoder is composed of two parts : Denosing part, Matching part
- Matching part is samely trained as previous DETRs via bipartite matching.
- Denoising part : input - noised GT object.

DN-DETR: Accelerate DETR Training by Introducing Query DeNoising, CVPR 2022

DN-DETR

Attention mask

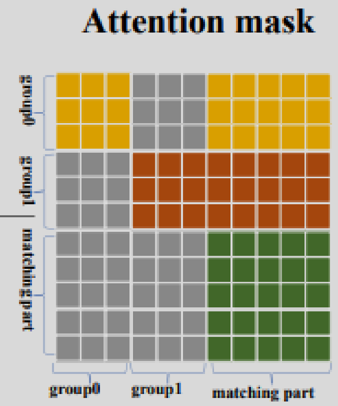
- the noised GT objects are divided into the group.

$$\mathbf{q} = \{\mathbf{g_0}, \mathbf{g_1}, ..., \mathbf{g_{P-1}}\}$$

- Each denoising group includes N number of queries

$$\mathbf{g_p} = \{q_0^p, q_1^p, ..., q_{M-1}^p\}$$

- The aim of attention mask is to prevent the information leakage
  - It prevents matching part refer to denoising part.
  - It prevents the reference among denoising group.



DN-DETR: Accelerate DETR Training by Introducing Query DeNoising, CVPR 2022

DN-DETR

Denoising

1. Box noising :
    a. center shifting,

$$|\Delta x| < \frac{\lambda_1 w}{2} \text{ and } |\Delta y| < \frac{\lambda_1 h}{2}, \text{ where } \lambda_1 \in (0,1)$$

the center of noise box belongs to the original box.

    a. box scaling

$$[(1-\lambda_2)w, (1+\lambda_2)w] \quad [(1-\lambda_2)h, (1+\lambda_2)h] \qquad \lambda_2 \in (0,1)$$

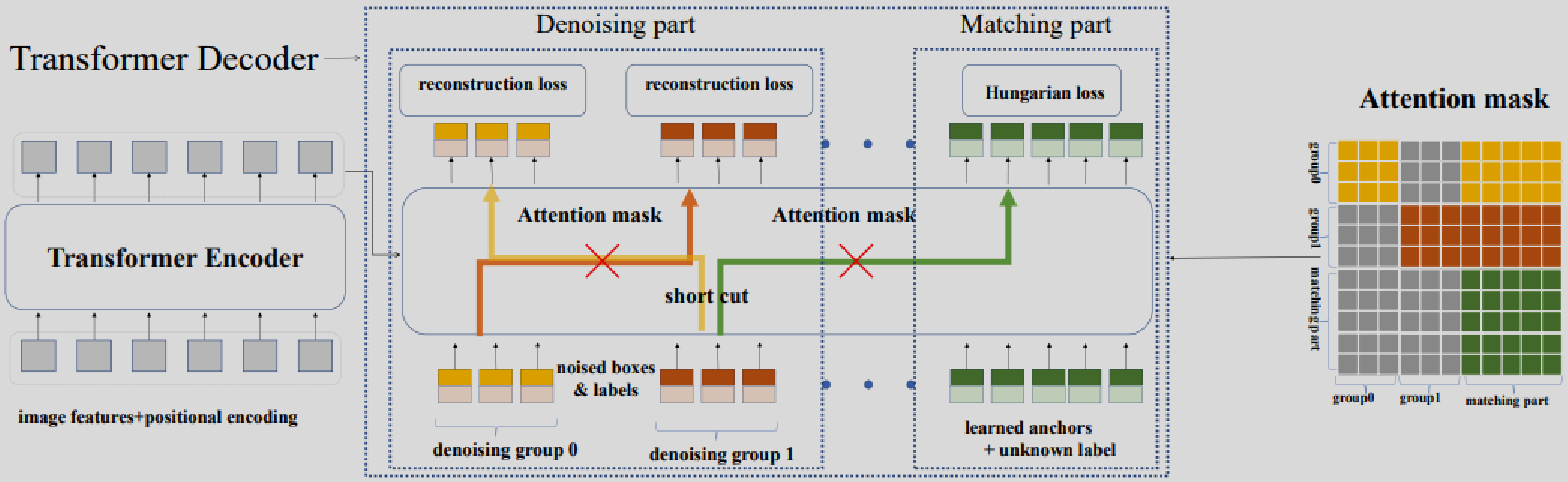
Randomly sampled within the ranges.

2. Label noising
    a. randomly flip

3. Reconstruction losses
    a. L1, GIOU loss -> box
    b. focal loss -> class

DN-DETR: Accelerate DETR Training by Introducing Query DeNoising, CVPR 2022

DN-DETR



- Decoder is composed of two parts : Denosing part, Matching part
- Matching part is samely trained as previous DETRs via bipartite matching.
- Denoising part : input - noised GT object. Only used in training.

DN-DETR: Accelerate DETR Training by Introducing Query DeNoising, CVPR 2022

DN-DETR

Model	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
DETR-R50 [1]	500	42.0	62.4	44.2	20.5	45.8	61.1	86	41M
Faster RCNN-FPN-R50 [15]	108	42.0	62.1	45.5	26.6	45.5	53.4	180	42M
Anchor DETR-R50 [18]	50	42.1	63.1	44.9	22.3	46.2	60.0	–	39M
Conditional DETR-R50 [12]	50	40.9	61.8	43.3	20.8	44.6	59.2	90	44M
DAB-DETR-R50 [11]	50	42.2	63.1	44.7	21.5	45.7	60.3	94	44M
DN-DETR-R50	50	**44.1(+1.9)**	64.4	46.7	22.9	48.0	63.4	94	44M
DETR-R101 [1]	500	43.5	63.8	46.4	21.9	48.0	61.8	152	60M
Faster RCNN-FPN-R101 [15]	108	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
Anchor DETR-R101 [18]	50	43.5	64.3	46.6	23.2	47.7	61.4	–	58M
Conditional DETR-R101 [12]	50	42.8	63.7	46.0	21.7	46.6	60.9	156	63M
DAB-DETR-R101 [11]	50	43.5	63.9	46.6	23.6	47.3	61.5	174	63M
DN-DETR-R101	50	**45.2(+1.7)**	65.5	48.3	24.1	49.1	65.1	174	63M
DETR-DC5-R50 [1]	500	43.3	63.1	45.9	22.5	47.3	61.1	187	41M
Anchor DETR-DC5-R50 [18]	50	44.2	64.7	47.5	24.7	48.2	60.6	151	39M
Conditional DETR-DC5-R50 [12]	50	43.8	64.4	46.7	24.0	47.6	60.7	195	44M
DAB-DETR-DC5-R50 [11]	50	44.5	65.1	47.7	25.3	48.2	62.3	202	44M
DN-DETR-DC5-R50	50	**46.3(+1.8)**	66.4	49.7	26.7	50.0	64.3	202	44M
DETR-DC5-R101 [1]	500	44.9	64.7	47.7	23.7	49.5	62.3	253	60M
Anchor DETR-R101 [18]	50	45.1	65.7	48.8	25.8	49.4	61.6	–	58M
Conditional DETR-DC5-R101 [12]	50	45.0	65.5	48.4	26.1	48.9	62.8	262	63M
DAB-DETR-DC5-R101 [11]	50	45.8	65.9	49.3	27.0	49.8	63.8	282	63M
DN-DETR-DC5-R101	50	**47.3(+1.5)**	67.5	50.8	28.6	51.5	65.0	282	63M

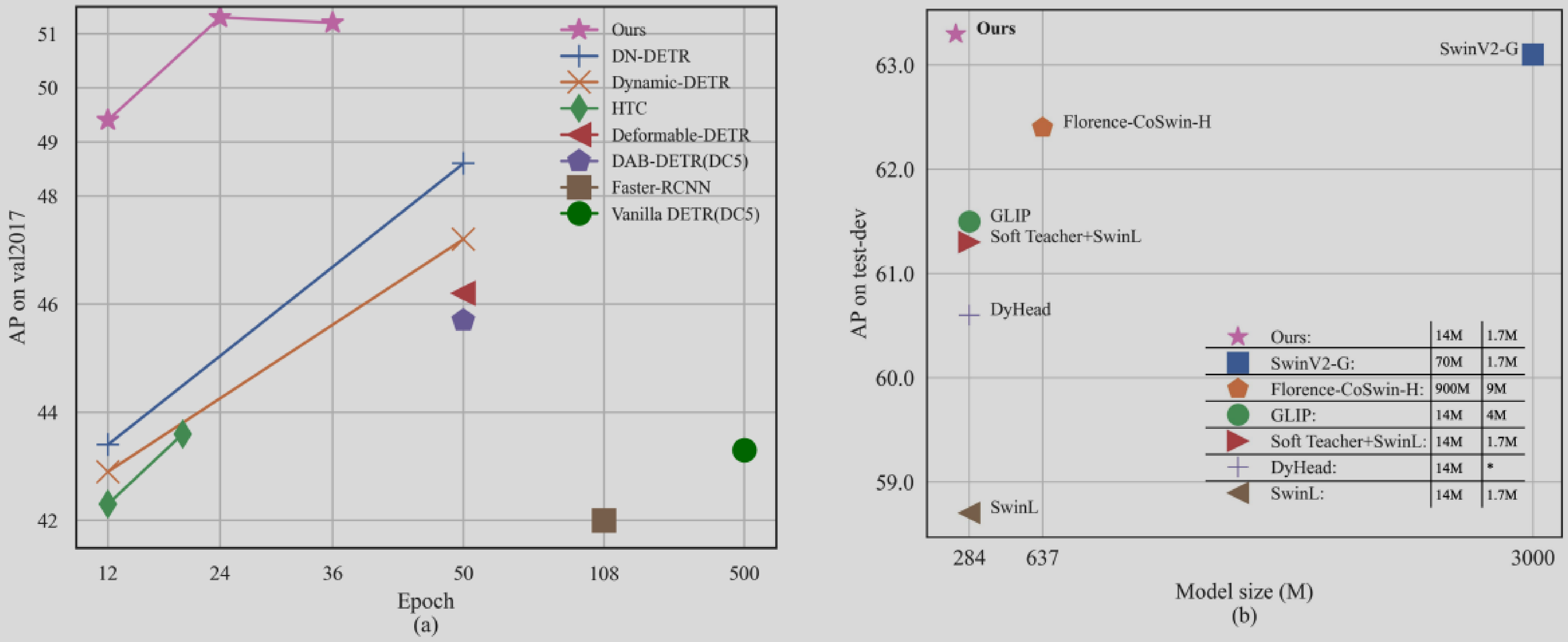
DN-DETR: Accelerate DETR Training by Introducing Query DeNoising, CVPR 2022

DN-DETR

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
Faster R50-FPN 1x [15]	✓	12	37.9	58.8	41.1	22.4	41.1	49.1	180	40M
DETR-R50 1x [1]		12	15.5	29.4	14.5	4.3	15.1	26.7	86	41M
DAB-DETR-DC5-R50 [11]		12	38.0	60.3	39.8	19.2	40.9	55.4	216	44M
DN-DETR-DC5-R50		12	**41.7(+3.7)**	61.4	44.1	21.2	45.0	60.2	216	44M
Deformable DETR-R50 1x [20]	✓	12	37.2	55.5	40.5	21.1	40.7	50.5	173	40M
Dynamic DETR-R50† 1x (without dynamic encoder)	✓	12	40.2	58.6	43.4	−−	−	−	−	−
Dynamic DETR-R50† 1x [4]	✓	12	42.9	61.0	46.3	24.6	44.9	54.4	−	−
DN-Deformable-DETR-R50 [4]	✓	12	**43.4**	61.9	47.2	24.8	46.8	59.4	195	48M
DAB-DETR-DC5-R101 [11]		12	40.3	62.6	42.7	22.2	44.0	57.3	282	63M
DN-DETR-DC5-R101		12	**42.8(+2.5)**	62.9	45.7	23.3	46.6	61.3	282	63M
Faster R101 FPN [15]	✓	108	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
DN-Deformable-DETR-R101	✓	12	**44.1**	62.8	47.9	26.0	47.8	61.3	275	67M

DN-DETR: Accelerate DETR Training by Introducing Query DeNoising, CVPR 2022

DINO

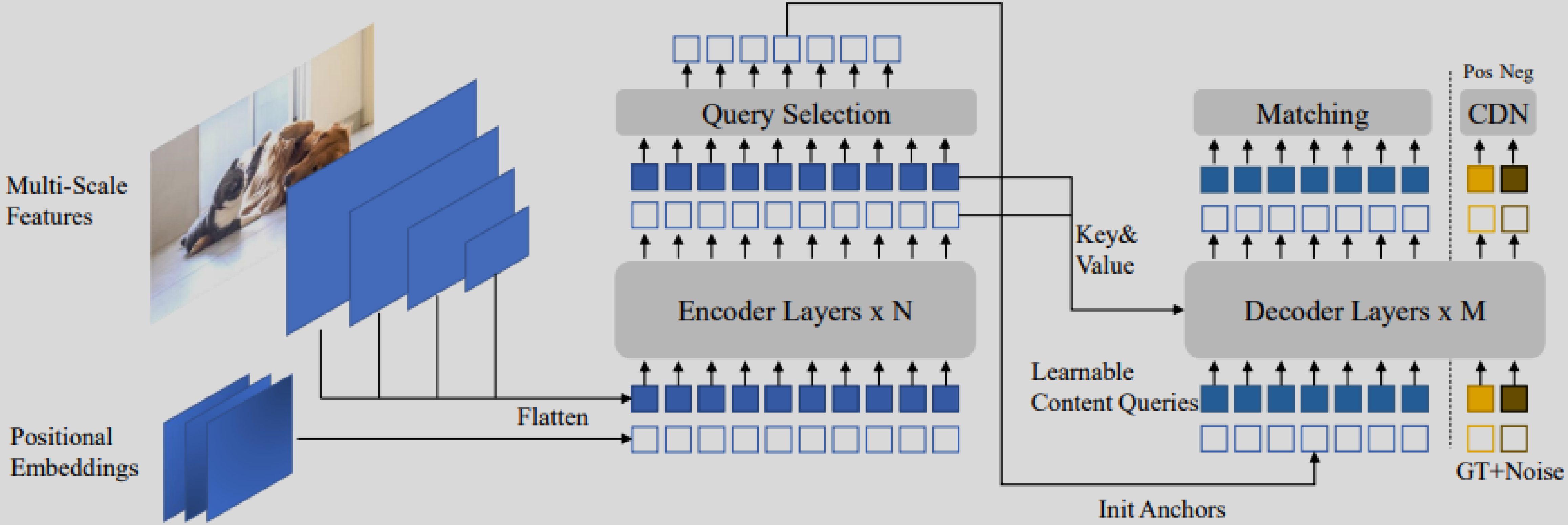


Based on deformable, DAB, DN DETR

- Denoising training
- Query initialization
- iterative box prediction

DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection, ICLR'23

DINO

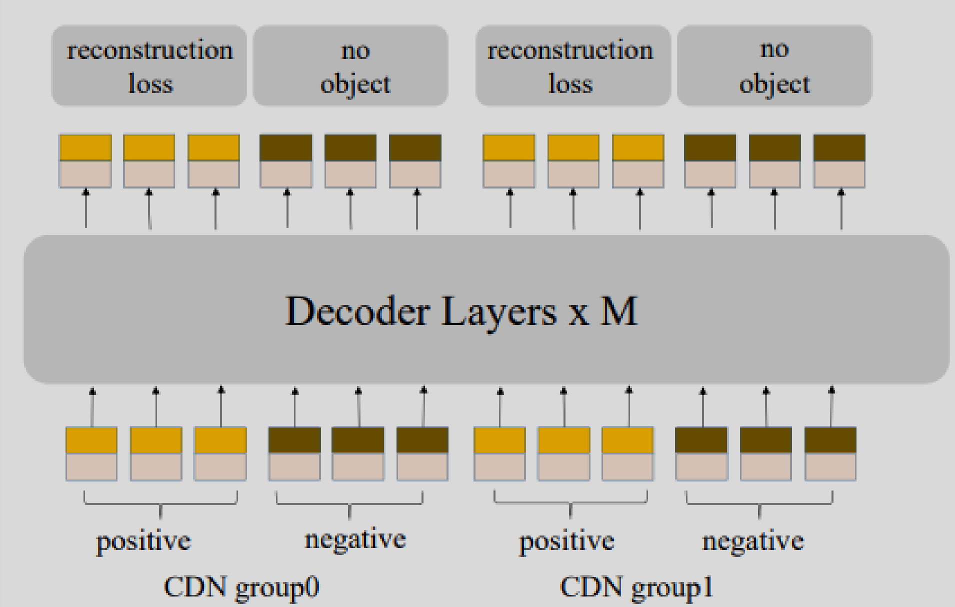


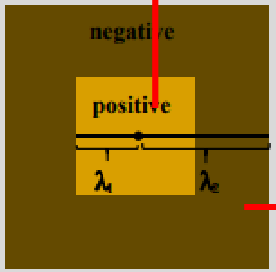
DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection, ICLR'23

DINO

Contrastive DeNoising Training

- DN-DETR is effective for queries that have GT box, while it lacks the ability to estimate background as “no object“ since there is no GT box for background.
- This can be relieved via CDN training.





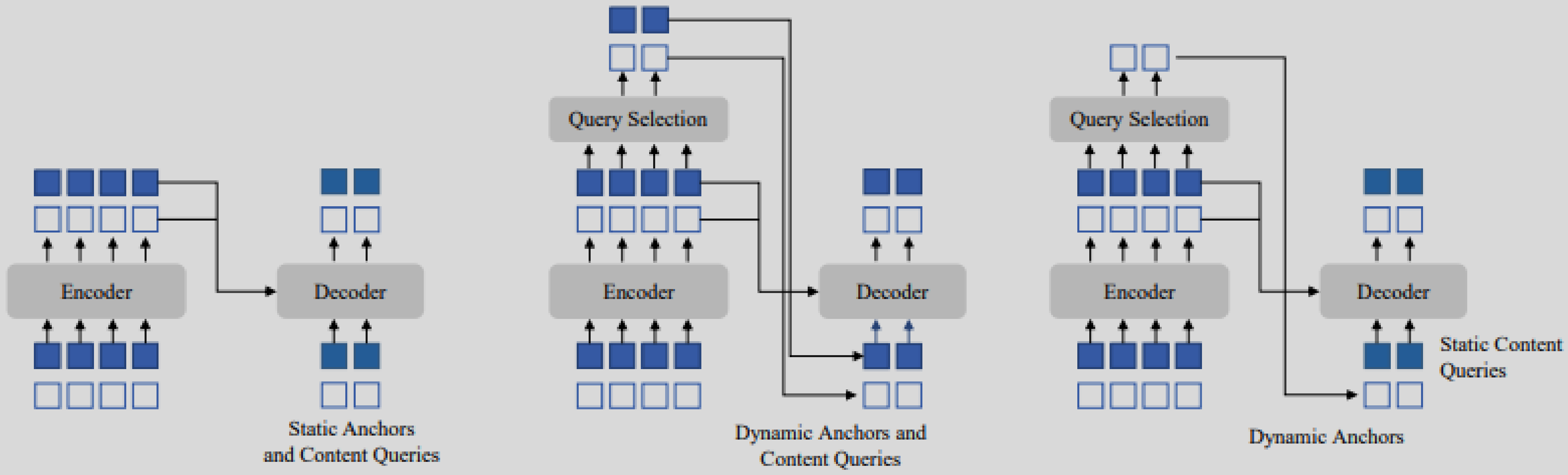
Positive queries within the inner square: noise scale is smaller than lambda_1

Negative queries between the inner and outer squares: noise scale is larger than lambda_1 but smaller than lambda_2

DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection, ICLR’23

DINO

Mixed Query Selection



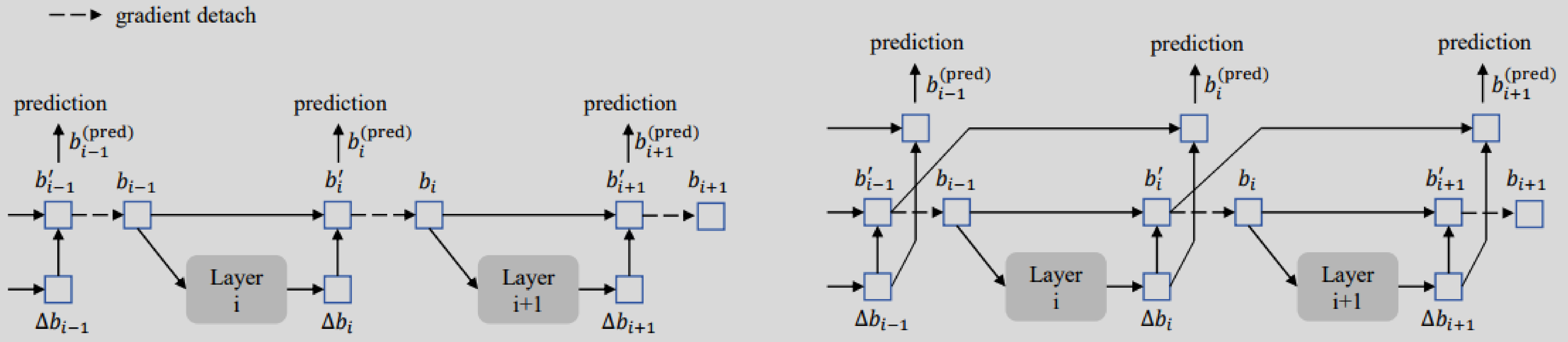
(a) Static Queries (b) Pure Query Selection (c) Mixed Query Selection

- Selected features could include multiple objects or contains only partial object. This causes confusion.
- Exploiting only positional queries while retaining contents queries as before.

DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection, ICLR'23

DINO

Look Forward Twice



(a) Look Forward Once

(b) Look Forward Twice

- In Deformable DETR, iterative box refinement does not involve the gradient back propagation for stability.
- It helps to improve the early layer's box prediction if we involve later layer's improved box information as input.

DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection, ICLR'23

DINO

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN [30]	108	42.0	62.4	44.2	20.5	45.8	61.1
DETR(DC5) [41]	500	43.3	63.1	45.9	22.5	47.3	61.1
Deformable DETR [41]	50	46.2	65.2	50.0	28.8	49.2	61.7
SMCA-R [11]	50	43.7	63.6	47.2	24.2	47.0	60.4
TSP-RCNN-R [34]	96	45.0	64.5	49.6	29.7	47.7	58.0
Dynamic DETR(5scale) [7]	50	47.2	65.9	51.1	28.6	49.3	59.1
DAB-Deformable-DETR [21]	50	46.9	66.0	50.8	30.1	50.4	62.5
DN-Deformable-DETR [17]	50	48.6	67.4	52.7	31.0	52.0	63.7
DINO-4scale	24	**50.4**(+1.8)	68.3	54.8	33.3	53.7	64.8
DINO-5scale	24	**51.3**(+2.7)	69.1	56.0	34.5	54.2	65.8
DINO-4scale	36	**50.9**(+2.3)	69.0	55.3	34.6	54.1	64.6
DINO-5scale	36	**51.2**(+2.6)	69.0	55.8	35.0	54.3	65.3

DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection, ICLR'23

Thank you!