Detection Transformers

DETR | Conditional DETR | DAB-DETR | DN-DETR | DINO (DETR)

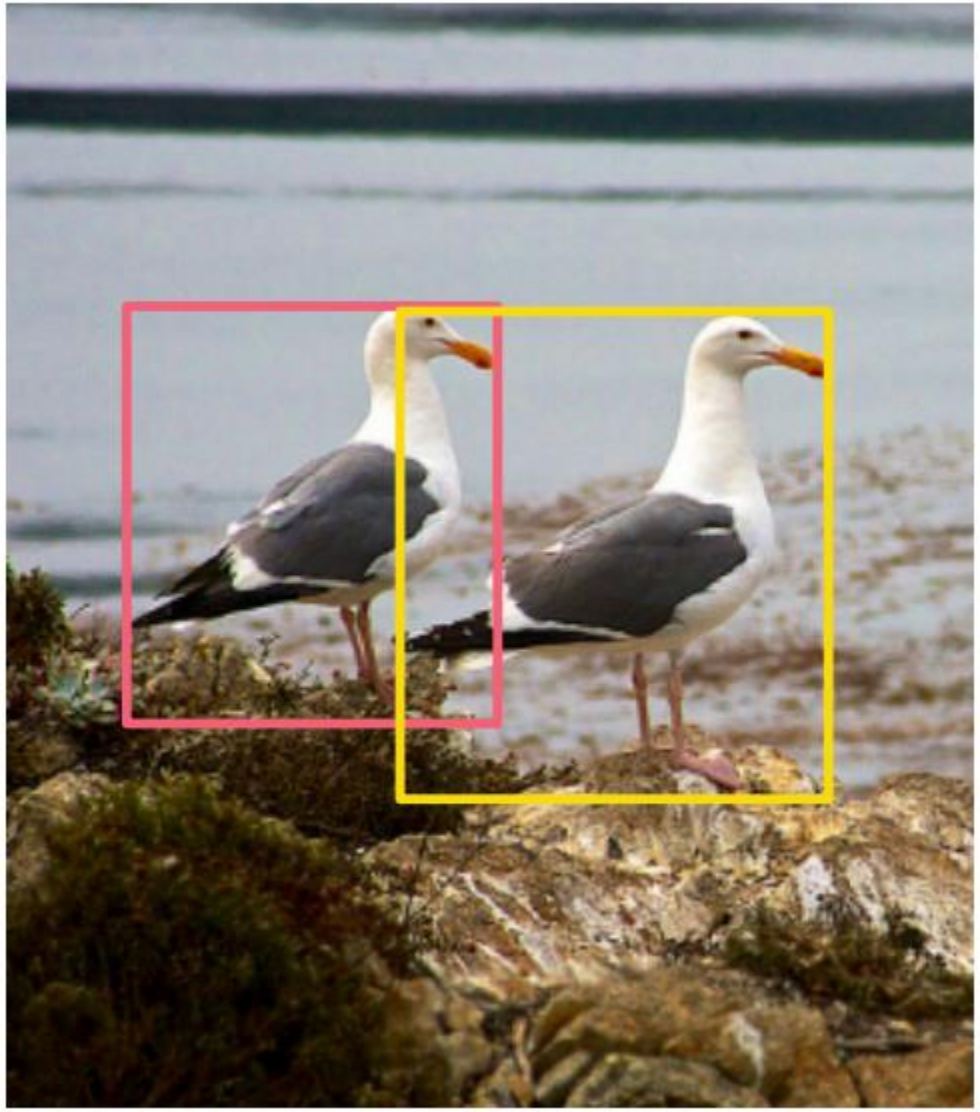
Denis Shepelev, March 2023
https://github.com/denzist/presentations

Object detection

For given image detect all objects, i.e. estimate:

- objects classes
- positions – bounding box (bbox)

This is *a set prediction problem*



Near-duplicate problem

- Previous methods utilizes a large set of anchors to obtain target bboxes, which leads to near-duplicate predictions
- To remove duplicates various hand-crafted heuristics are used, for example non-maximum suppression (NMS)
- Solution: construct end-to-end solution, that do not rely on hand-crafted blocks and reduces used prior knowledge

Plan

- DETR
- Conditional DETR
- DAB DETR
- DN DETR
- DINO

DETR

End-to-End Object Detection with Transformers (ECCV, 2020)

DETR

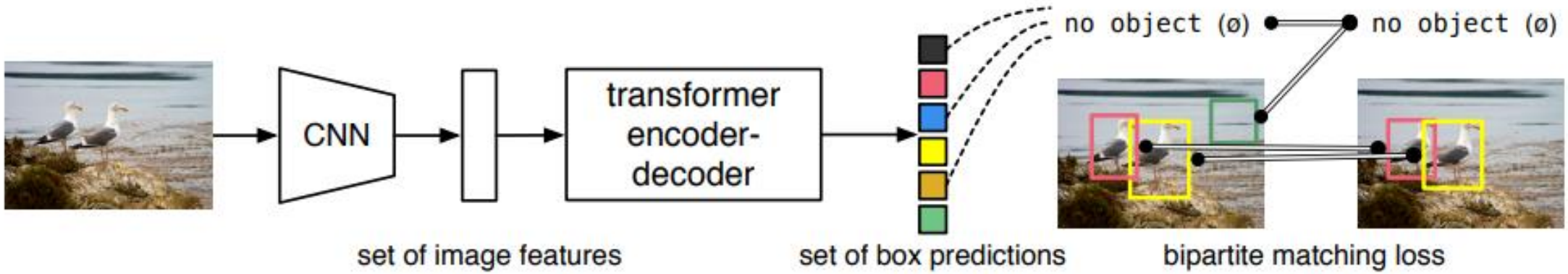
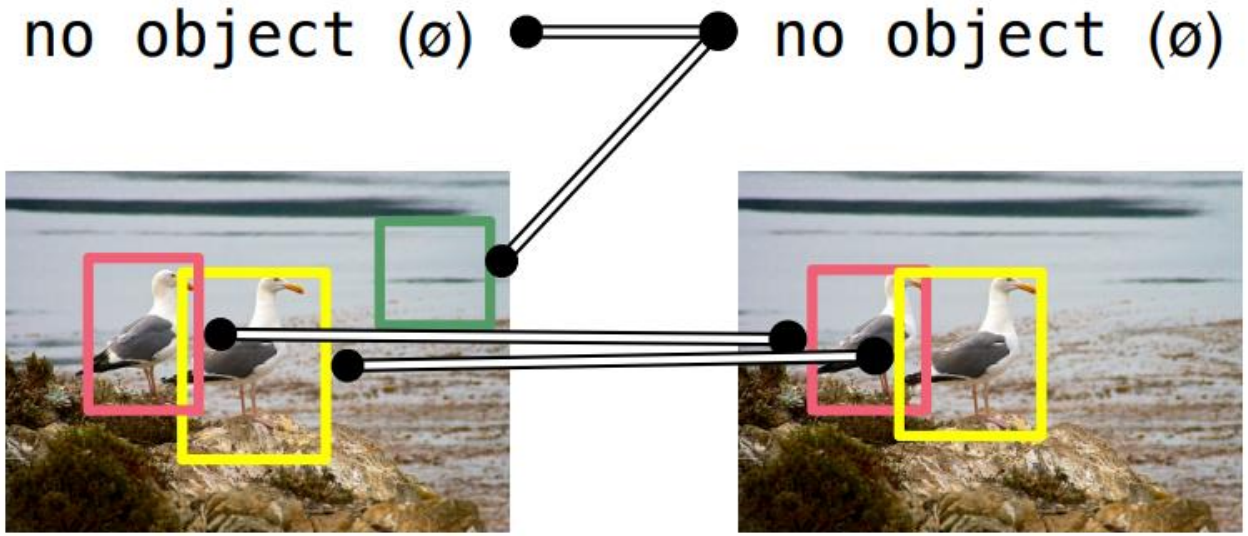


Fig. 1: DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a transformer architecture. During training, bipartite matching uniquely assigns predictions with ground truth boxes. Prediction with no match should yield a "`no object`" ($\varnothing$) class prediction.

Bipartite matching



Worker \ Task	Clean bathroom	Sweep floors	Wash windows
Alice	**$8**	$4	$7
Bob	$5	$2	**$3**
Dora	$9	**$4**	$8

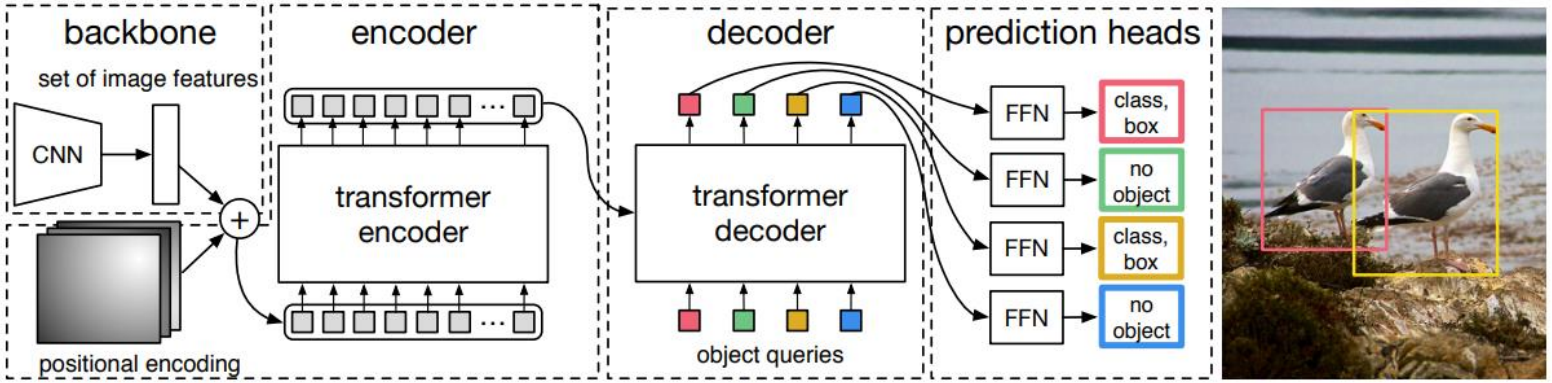
Assignment problem example can be solved by Hungarian algorithm

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right]$$

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = -\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$$

Encoder-Decoder



Encoder self-attention

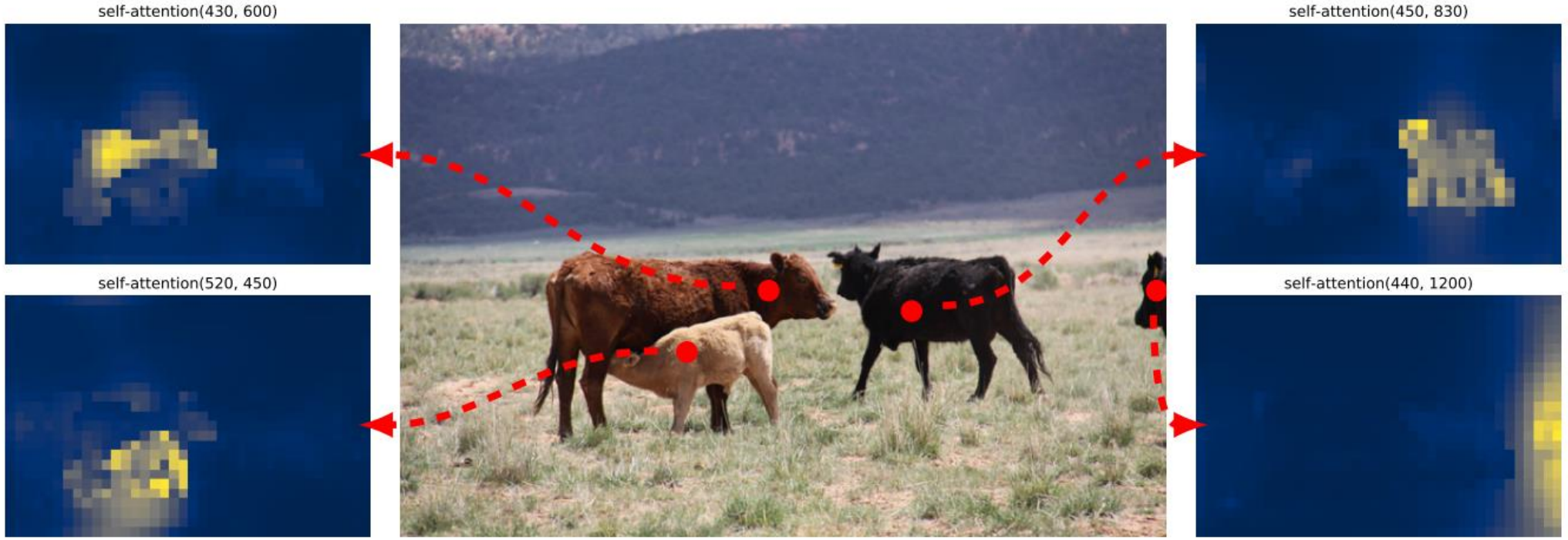


Fig. 3: Encoder self-attention for a set of reference points. The encoder is able to separate individual instances. Predictions are made with baseline DETR model on a validation set image.

Object queries (learnt positional encodings)

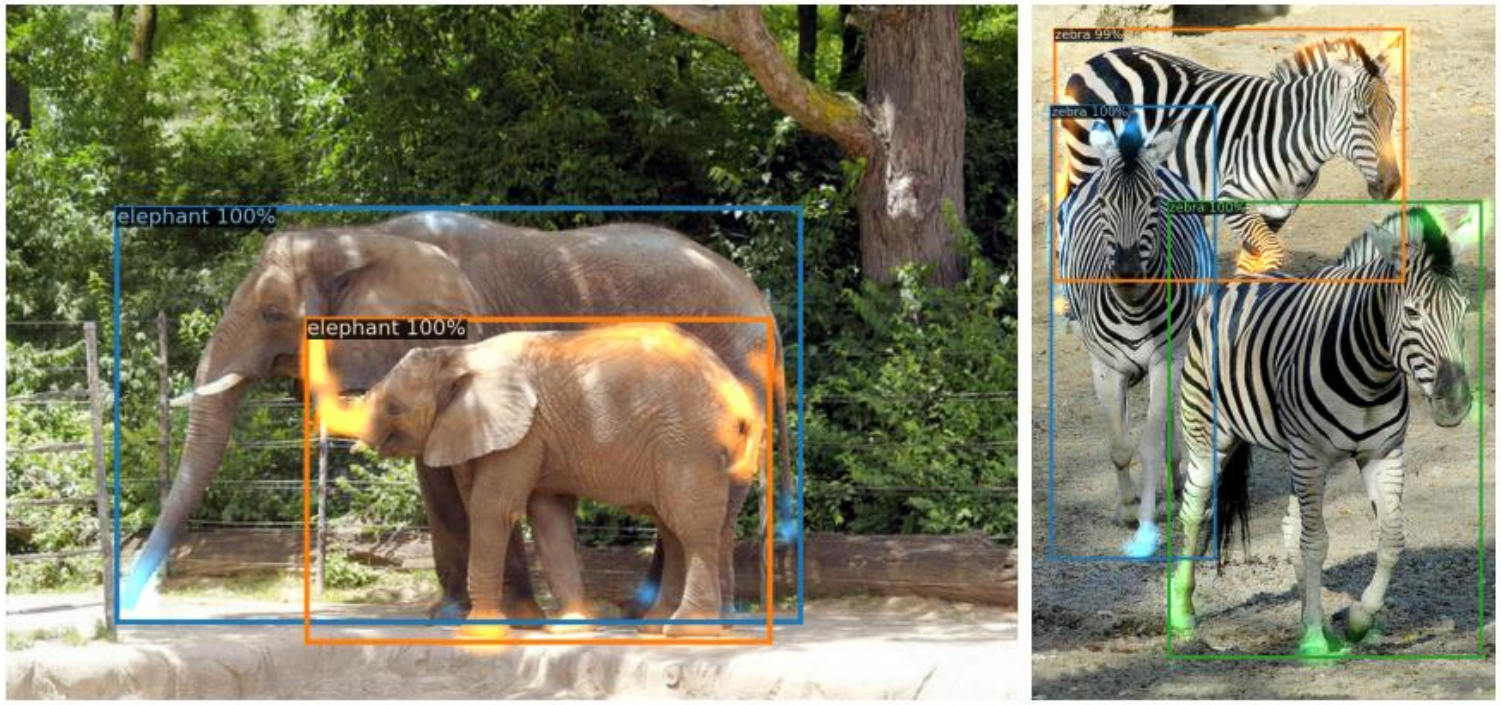


Fig. 6: Visualizing decoder attention for every predicted object (images from COCO `val` set). Predictions are made with DETR-DC5 model. Attention scores are coded with different colors for different objects. Decoder typically attends to object extremities, such as legs and heads. Best viewed in color.

Object queries (learnt positional encodings)

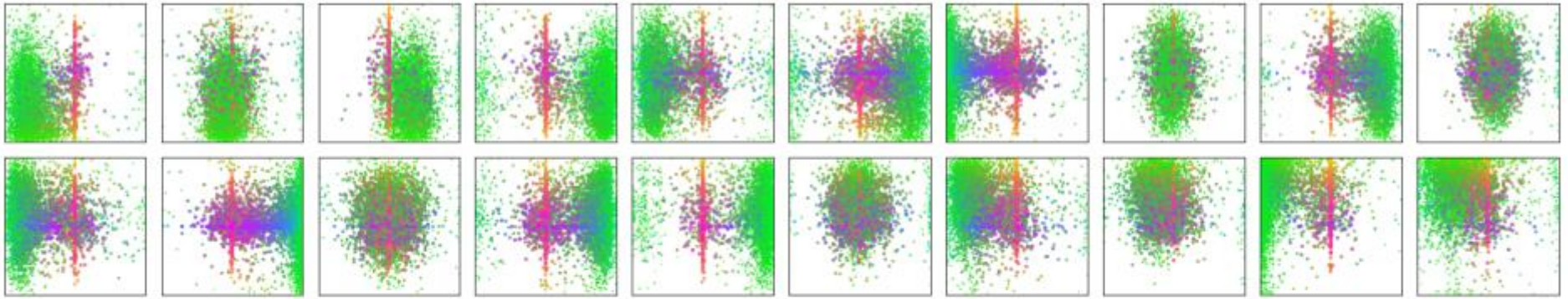


Fig. 7: Visualization of all box predictions on all images from COCO 2017 val set for 20 out of total $N = 100$ prediction slots in DETR decoder. Each box prediction is represented as a point with the coordinates of its center in the 1-by-1 square normalized by each image size. The points are color-coded so that green color corresponds to small boxes, red to large horizontal boxes and blue to large vertical boxes. We observe that each slot learns to specialize on certain areas and box sizes with several operating modes. We note that almost all slots have a mode of predicting large image-wide boxes that are common in COCO dataset.

Ablations

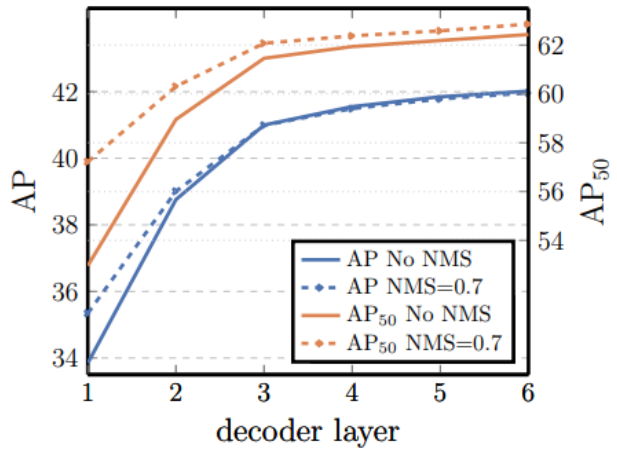


Fig. 4: AP and AP_{50} performance after each decoder layer. A single long schedule baseline model is evaluated. DETR does not need NMS by design, which is validated by this figure. NMS lowers AP in the final layers, removing TP predictions, but improves AP in the first decoder layers, removing double predictions, as there is no communication in the first layer, and slightly improves AP_{50}.

Table 3: Results for different positional encodings compared to the baseline (last row), which has fixed sine pos. encodings passed at every attention layer in both the encoder and the decoder. Learned embeddings are shared between all layers. Not using spatial positional encodings leads to a significant drop in AP. Interestingly, passing them in decoder only leads to a minor AP drop. All these models use learned output positional encodings.

spatial pos. enc. encoder	spatial pos. enc. decoder	output pos. enc. decoder	AP	Δ	AP_{50}	Δ
none	none	learned at input	32.8	-7.8	55.2	-6.5
sine at input	sine at input	learned at input	39.2	-1.4	60.0	-1.6
learned at attn.	learned at attn.	learned at attn.	39.6	-1.0	60.7	-0.9
none	sine at attn.	learned at attn.	39.3	-1.3	60.3	-1.4
sine at attn.	sine at attn.	learned at attn.	**40.6**	-	**61.6**	-

Table 4: Effect of loss components on AP. We train two models turning off ℓ_1 loss, and GIoU loss, and observe that ℓ_1 gives poor results on its own, but when combined with GIoU improves AP_M and AP_L. Our baseline (last row) combines both losses.

class	ℓ_1	GIoU	AP	Δ	AP_{50}	Δ	AP_S	AP_M	AP_L
✓	✓		35.8	-4.8	57.3	-4.4	13.7	39.8	57.9
✓		✓	39.9	-0.7	**61.6**	0	**19.9**	43.2	57.9
✓	✓	✓	**40.6**	-	**61.6**	-	**19.9**	**44.3**	**60.2**

Comparisons to other detectors

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**

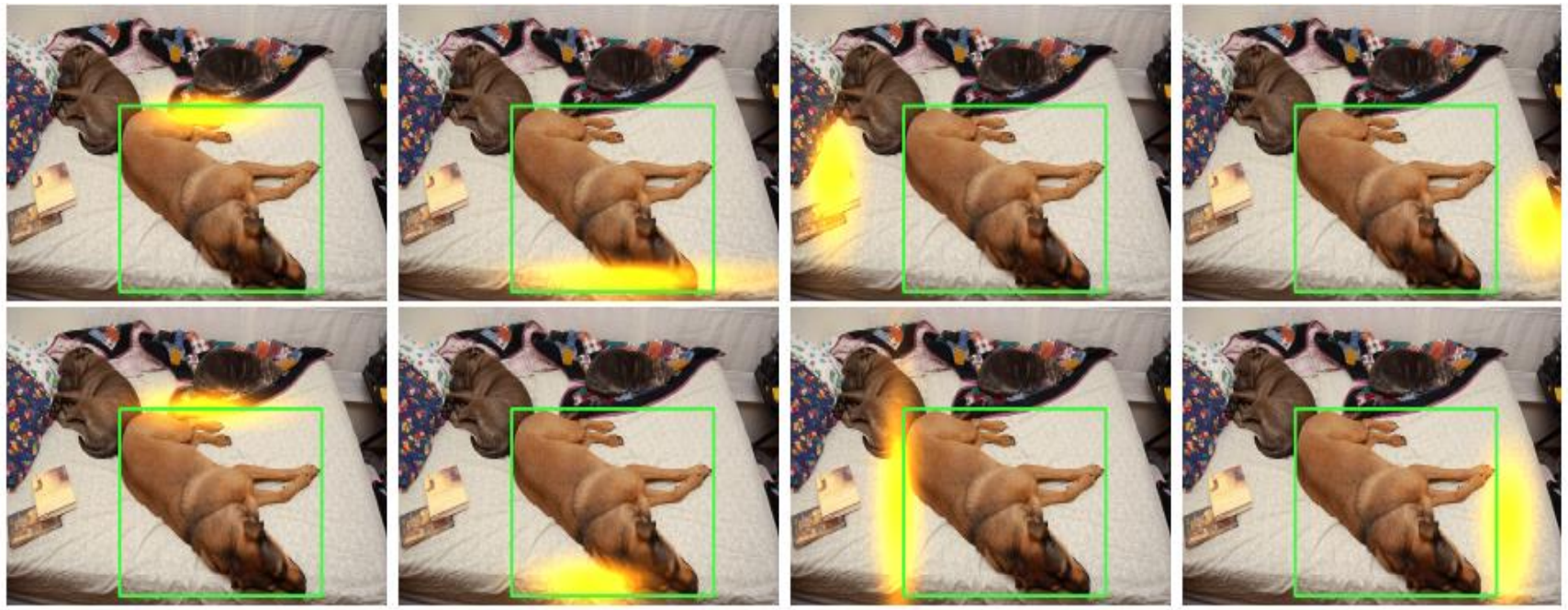
Still worse than Faster RCNN on small objects

DETR training time is long

- **Learnt positional encodings are not so good**

$(o_q^T + c_q)^T p_k$

$(o_q^T + c_q)^T p_k$



Attention map after training DETR for 50 epochs (first row) and 500 epochs (second row)

Conditional DETR

Conditional DETR for Fast Training Convergence (ICCV 2021)

Conditional DETR

Estimate bboxes as:

$$\mathbf{b} = \text{sigmoid}(\text{FFN}(\mathbf{f}) + [\mathbf{s}^\top\ 0\ 0]^\top)$$

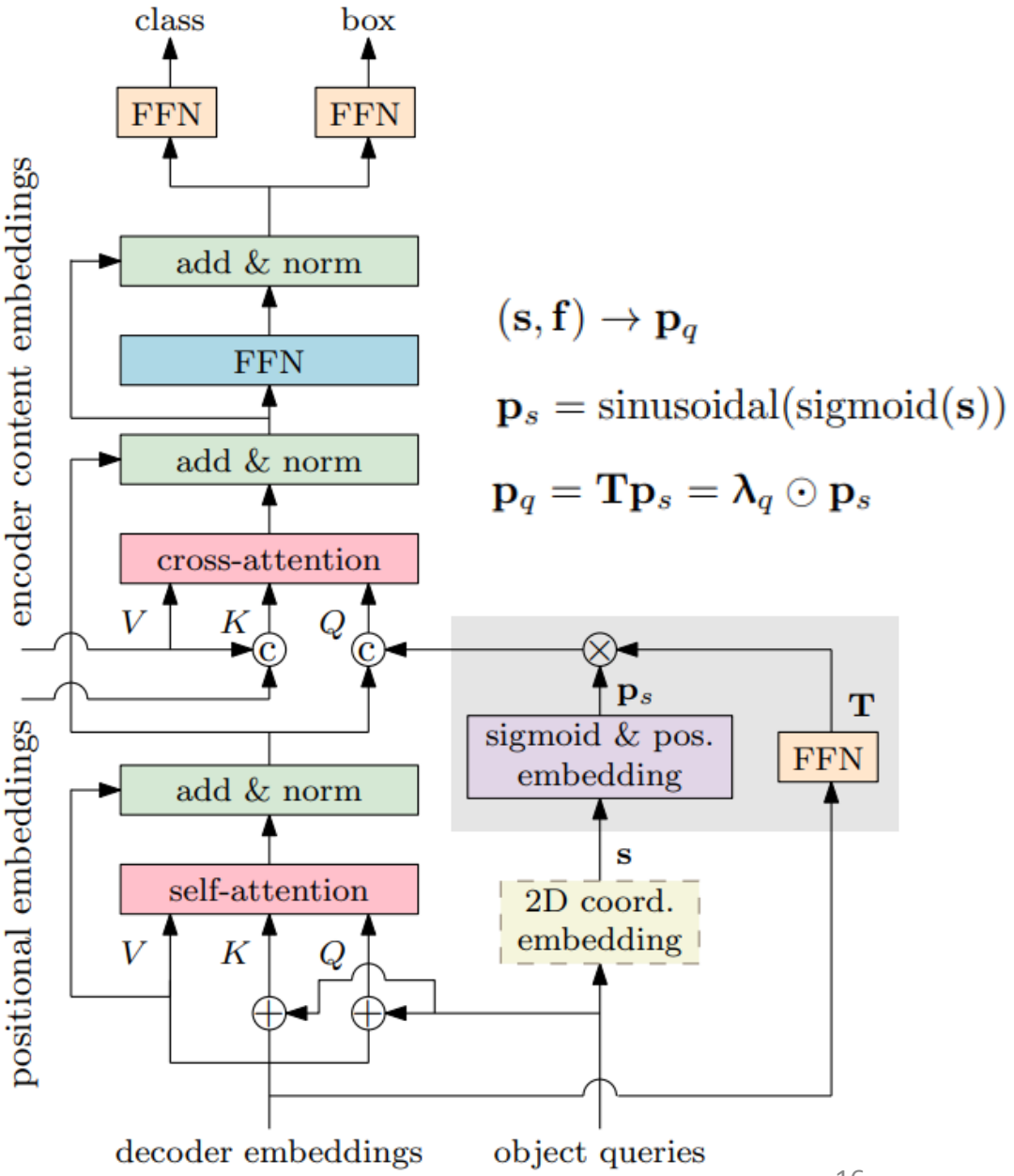
In DETR $\boldsymbol{s} = 0$
In Conditional DETR $\boldsymbol{s}$ is learnt

DETR attention estimation:

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

Conditional DETR attention estimation:

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



Conditional object queries

$\boldsymbol{p}_q^T \boldsymbol{p}_k$

$\boldsymbol{c}_q^T \boldsymbol{c}_k$

$\boldsymbol{c}_q^T \boldsymbol{c}_k + \boldsymbol{p}_q^T \boldsymbol{p}_k$

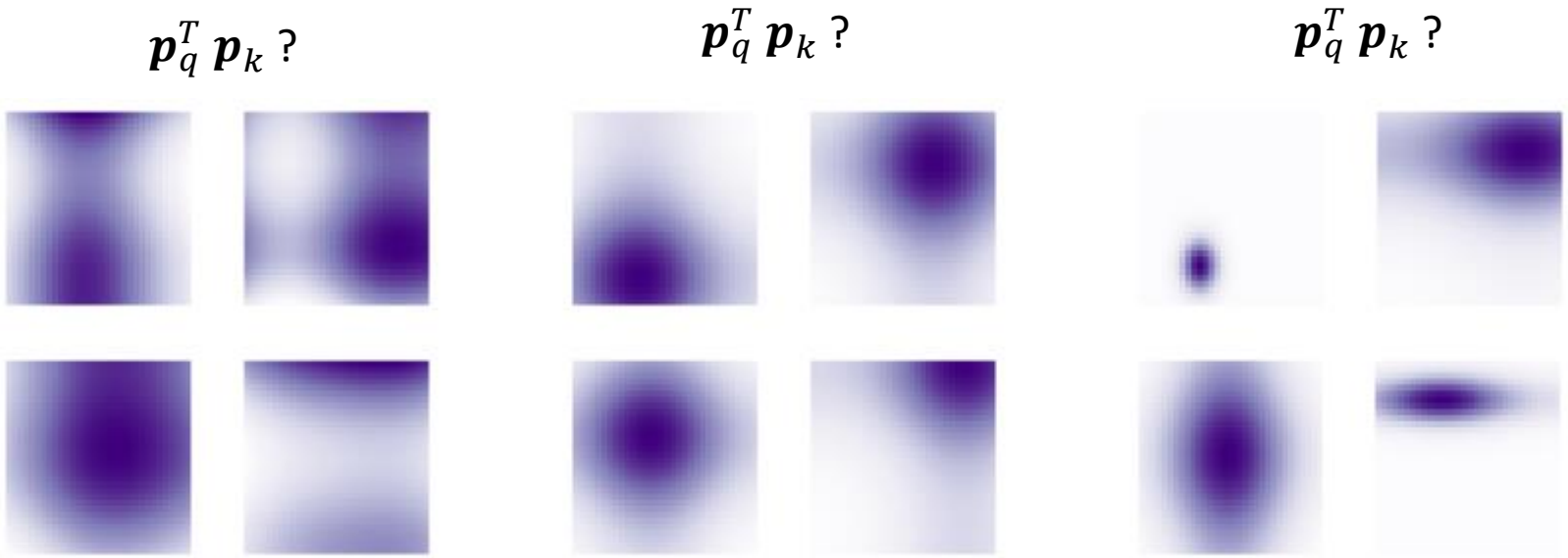
Comparison

Model	#epochs	GFLOPs	#params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-FPN-R50 [33]	36	180	42	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-FPN-R50 [33]	108	180	42	42.0	62.1	45.5	26.6	45.5	53.4
Deformable DETR-R50 [53]	50	173	40	43.8	62.6	47.7	26.4	47.1	58.0
TSP-FCOS-R50 [37]	36	189	–	43.1	62.3	47.0	26.6	46.8	55.9
TSP-RCNN-R50 [37]	36	188	–	43.8	63.3	48.3	28.6	46.9	55.7
TSP-RCNN-R50 [37]	96	188	–	45.0	64.5	49.6	29.7	47.7	58.0
Conditional DETR-DC5-R50	50	195	44	43.8	64.4	46.7	24.0	47.6	60.7
Conditional DETR-DC5-R50	108	195	44	45.1	65.4	48.5	25.3	49.0	62.2
Faster RCNN-FPN-R101 [33]	36	246	60	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-FPN-R101 [33]	108	246	60	44.0	63.9	47.8	27.2	48.1	56.0
TSP-FCOS-R101 [37]	36	255	–	44.4	63.8	48.2	27.7	48.6	57.3
TSP-RCNN-R101 [37]	36	254	–	44.8	63.8	49.2	29.0	47.9	57.1
TSP-RCNN-R101 [37]	96	254	–	46.5	66.0	51.2	29.9	49.7	59.2
Conditional DETR-DC5-R101	50	262	63	45.0	65.5	48.4	26.1	48.9	62.8
Conditional DETR-DC5-R101	108	262	63	45.9	66.8	49.5	27.2	50.3	63.3

DAB DETR

DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR (ICLR 2022)

Object queries vs positional encodings



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

Figure 4: We visualize the positional attention between positional queries and positional keys for DETR, Conditional DETR, and our proposed DAB-DETR. Four attention maps in (a) are randomly sampled, and we select figures with similar query positions as in (a) for (b) and (c). The darker the color, the greater the attention weight, and vice versa. (a) Each attention map in DETR is calculated by performing dot product between a learned query and positional embeddings from a feature map, and can have multiple modes and unconcentrated attentions. (b) The positional queries in Conditional DETR are encoded in the same way as the image positional embeddings, resulting in Gaussian-like attention maps. However, it cannot adapt to objects of different scales. (c) DAB-DETR explicitly modulates the attention map using the width and height information of an anchor, making it more adaptive to object size and shape. The modulated attentions can be regarded as helping perform soft ROI pooling.

DAB-DETR

$$P_q = \text{MLP}(\text{PE}(A_q))$$

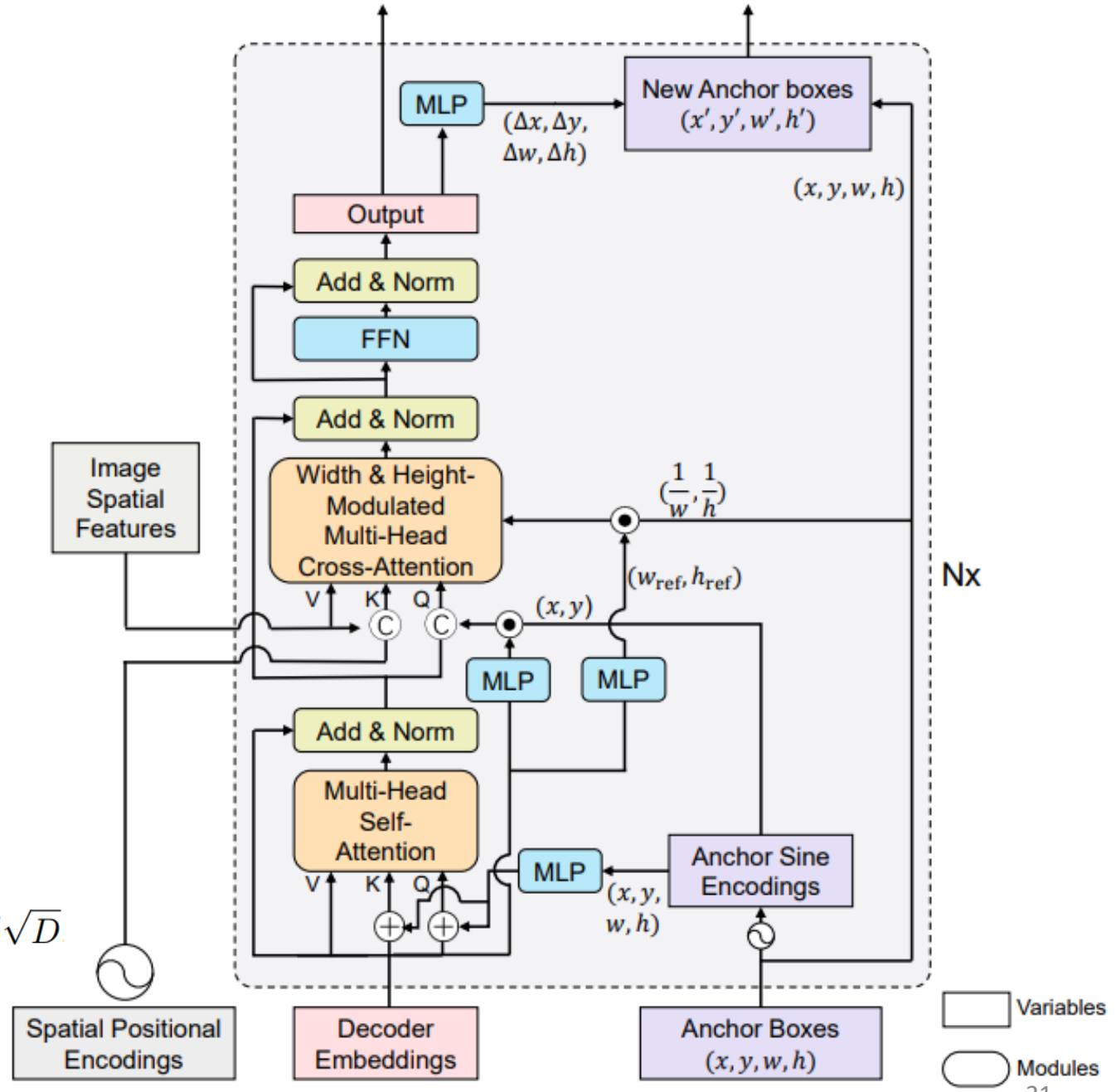
Self-Attn: $Q_q = C_q + P_q, \quad K_q = C_q + P_q, \quad V_q = C_q$

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

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

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

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



Modulation and temperature tuning

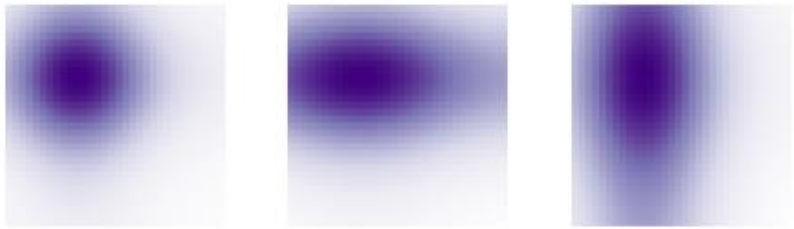


Figure 6: Positional attention maps modulated by width and height.

$$\text{ModulateAttn}((x, y), (x_{\text{ref}}, y_{\text{ref}})) =$$

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

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

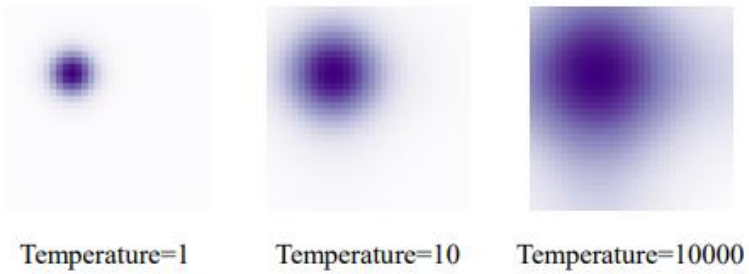
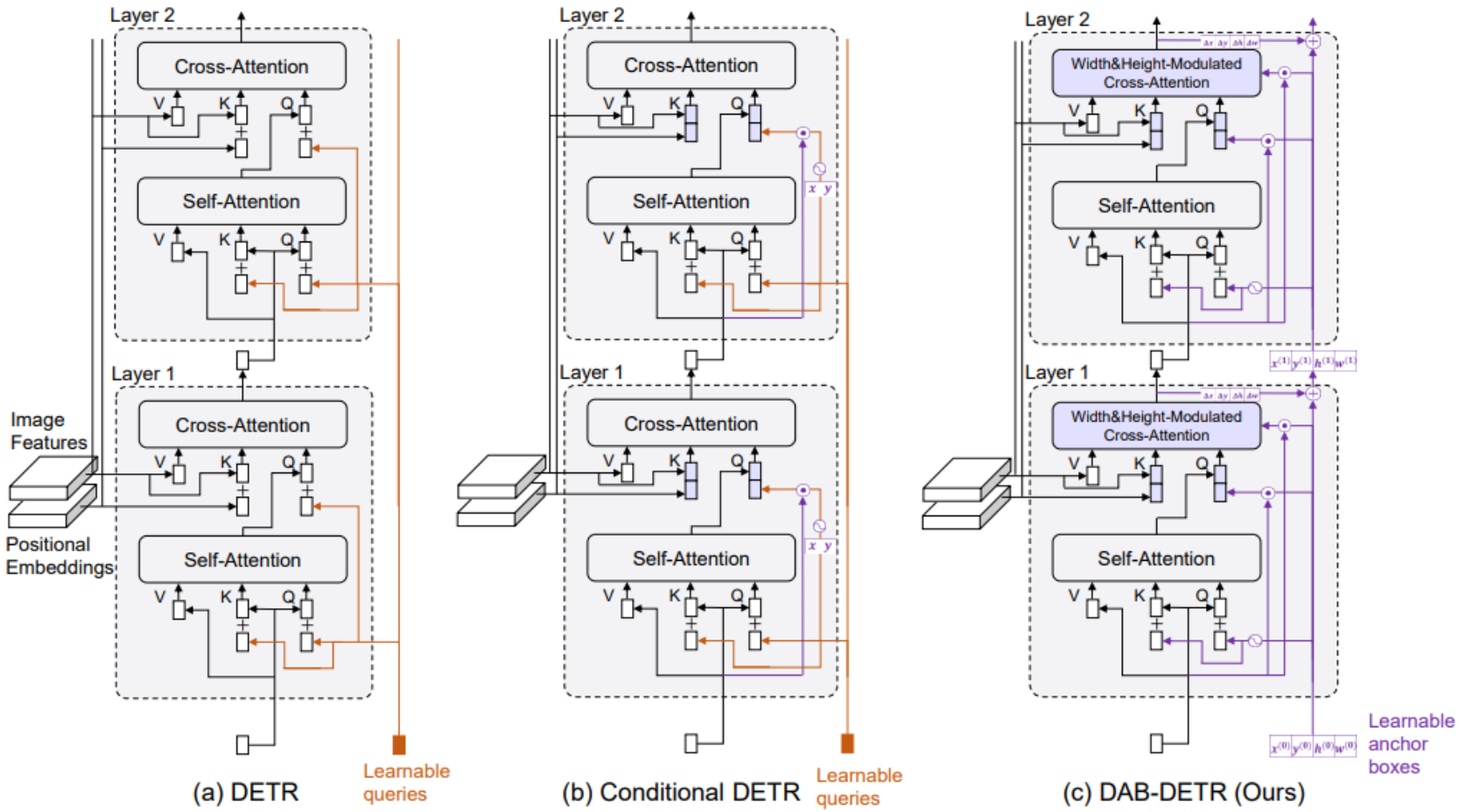


Figure 7: Positional attention maps with different temperatures.

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

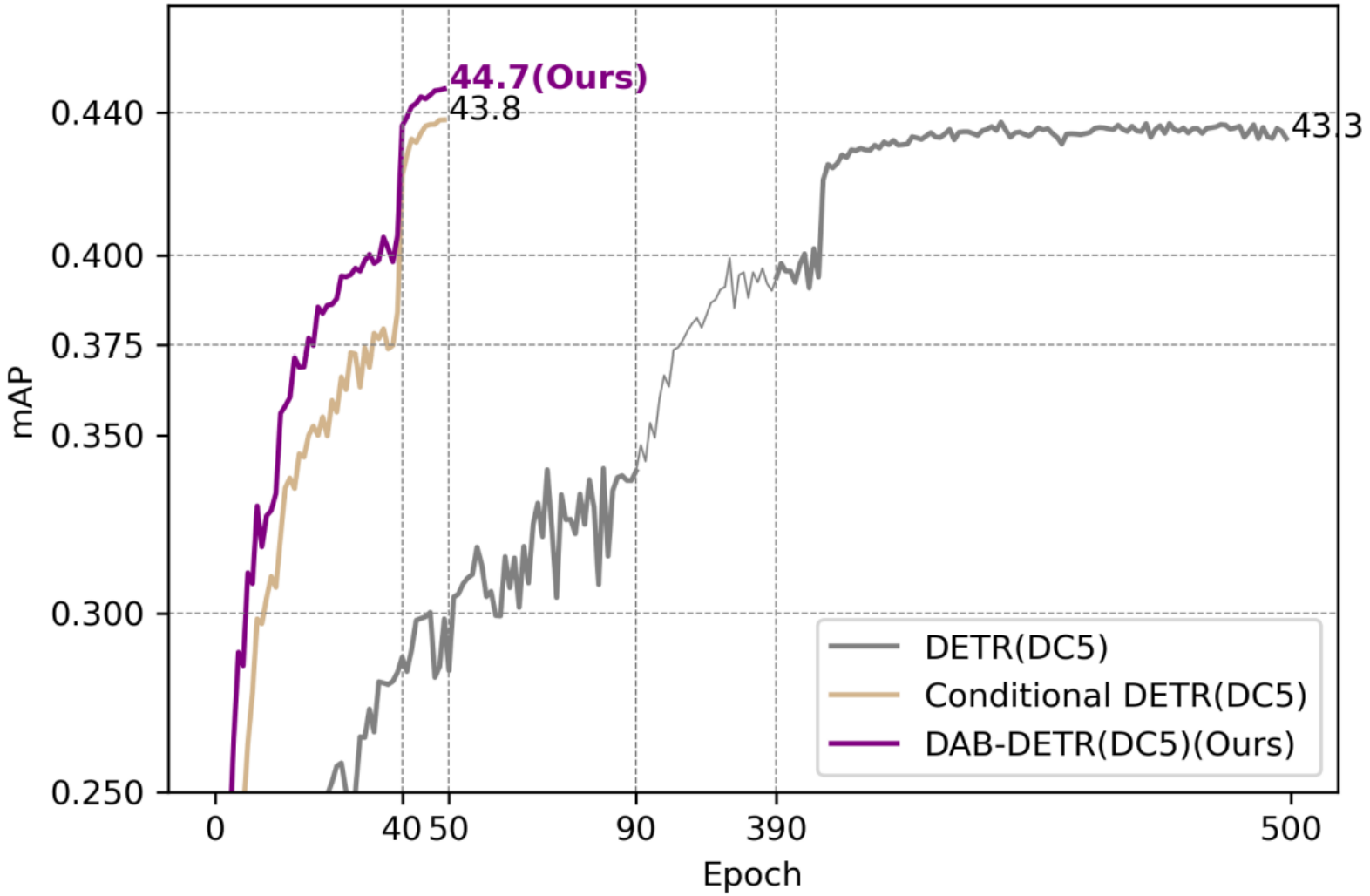
Architecture comparison



Comparison

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
DETR-R50		500	42.0	62.4	44.2	20.5	45.8	61.1	86	41M
Faster RCNN-FPN-R50		108	42.0	62.1	45.5	26.6	45.5	53.4	180	42M
Anchor DETR-R50*		50	42.1	63.1	44.9	22.3	46.2	60.0	–	39M
Conditional DETR-R50		50	40.9	61.8	43.3	20.8	44.6	59.2	90	44M
DAB-DETR-R50		50	42.2	63.1	44.7	21.5	45.7	60.3	94	44M
DAB-DETR-R50*		50	**42.6**	63.2	45.6	21.8	46.2	61.1	100	44M
DETR-DC5-R50		500	43.3	63.1	45.9	22.5	47.3	61.1	187	41M
Deformable DETR-R50	✓	50	43.8	62.6	47.7	26.4	47.1	58.0	173	40M
SMCA-R50	✓	50	43.7	63.6	47.2	24.2	47.0	60.4	152	40M
TSP-RCNN-R50	✓	96	45.0	64.5	49.6	29.7	47.7	58.0	188	–
Anchor DETR-DC5-R50*		50	44.2	64.7	47.5	24.7	48.2	60.6	151	39M
Conditional DETR-DC5-R50		50	43.8	64.4	46.7	24.0	47.6	60.7	195	44M
DAB-DETR-DC5-R50		50	44.5	65.1	47.7	25.3	48.2	62.3	202	44M
DAB-DETR-DC5-R50*		50	**45.7**	66.2	49.0	26.1	49.4	63.1	216	44M
DETR-R101		500	43.5	63.8	46.4	21.9	48.0	61.8	152	60M
Faster RCNN-FPN-R101		108	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
Anchor DETR-R101*		50	43.5	64.3	46.6	23.2	47.7	61.4	–	58M
Conditional DETR-R101		50	42.8	63.7	46.0	21.7	46.6	60.9	156	63M
DAB-DETR-R101		50	43.5	63.9	46.6	23.6	47.3	61.5	174	63M
DAB-DETR-R101*		50	**44.1**	64.7	47.2	24.1	48.2	62.9	179	63M
DETR-DC5-R101		500	44.9	64.7	47.7	23.7	49.5	62.3	253	60M
TSP-RCNN-R101	✓	96	46.5	66.0	51.2	29.9	49.7	59.2	254	–
SMCA-R101	✓	50	44.4	65.2	48.0	24.3	48.5	61.0	218	50M
Anchor DETR-R101*		50	45.1	65.7	48.8	25.8	49.4	61.6	–	58M
Conditional DETR-DC5-R101		50	45.0	65.5	48.4	26.1	48.9	62.8	262	63M
DAB-DETR-DC5-R101		50	45.8	65.9	49.3	27.0	49.8	63.8	282	63M
DAB-DETR-DC5-R101*		50	**46.6**	67.0	50.2	28.1	50.5	64.1	296	63M

Convergence

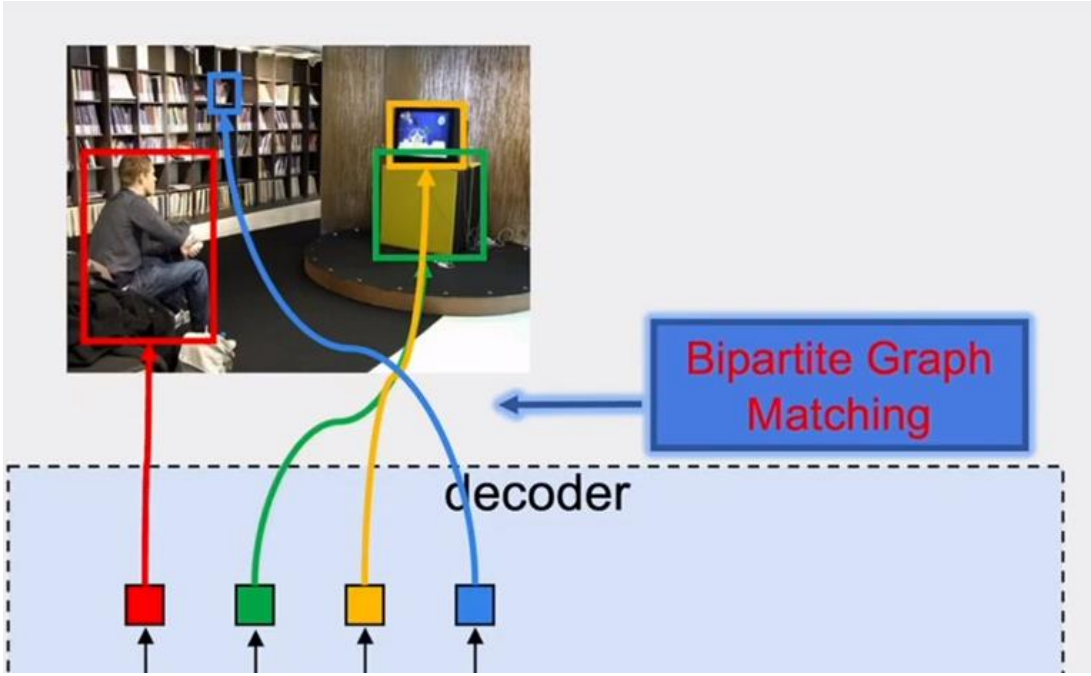


DN DETR

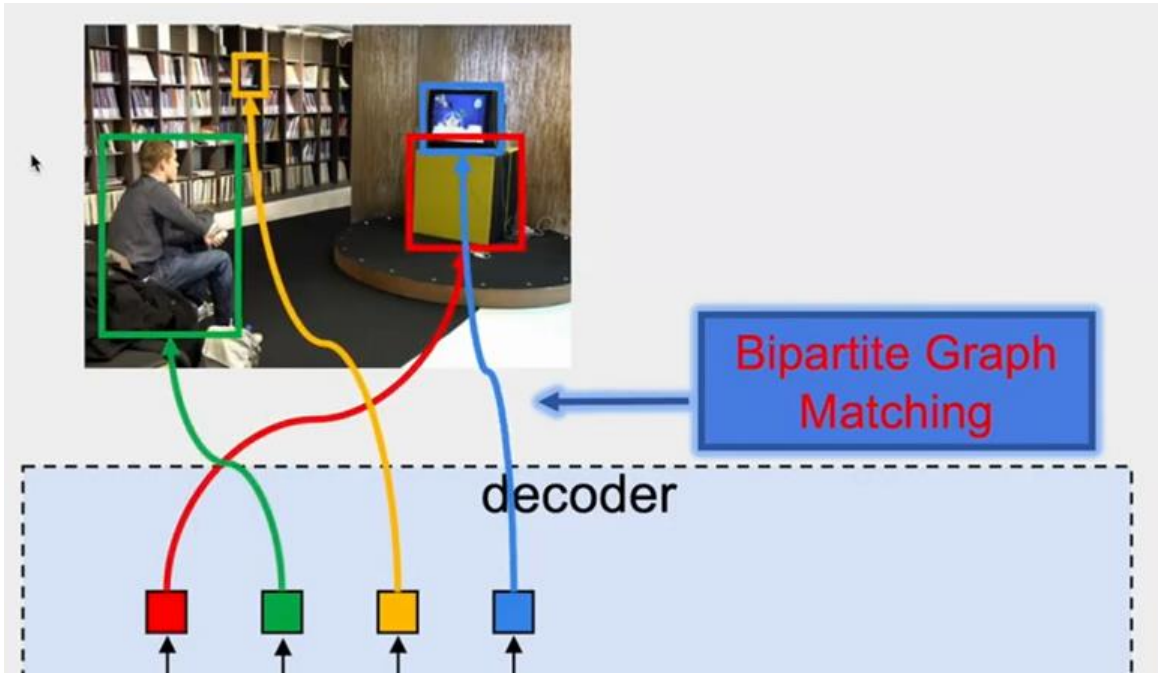
DN-DETR: Accelerate DETR Training by Introducing Query DeNoising (CVPR 2022)

Bipartite matching instability (1)

Before parameters update



After parameters update



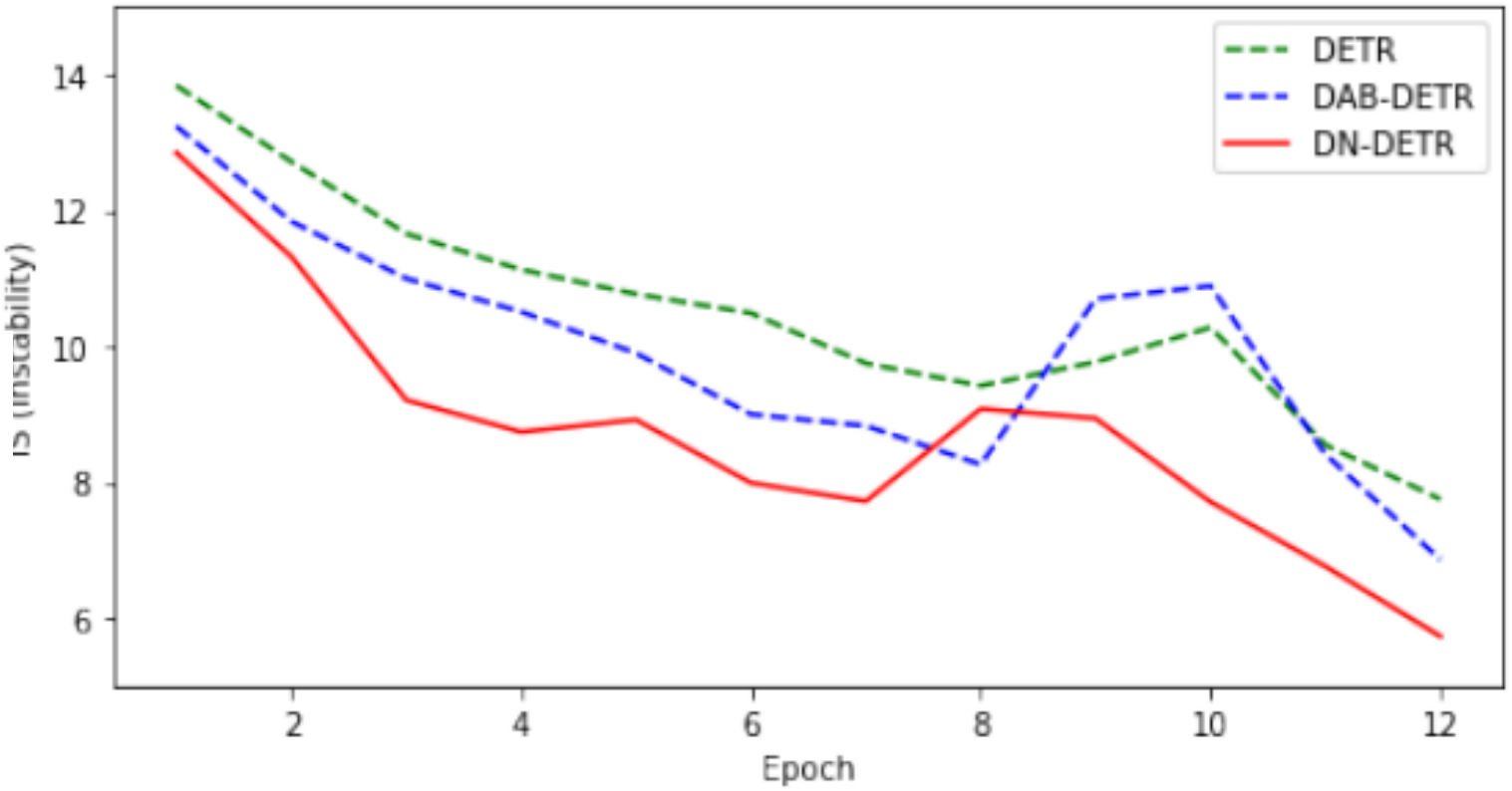
Bipartite matching instability (2)

Bipartite matching is unstable and might lead to slow convergence!

$$\mathbf{O^i} = \{O_0^i, O_1^i, ..., O_{N-1}^i\}$$
$$\mathbf{T} = \{T_0, T_1, T_2, ..., T_{M-1}\}$$

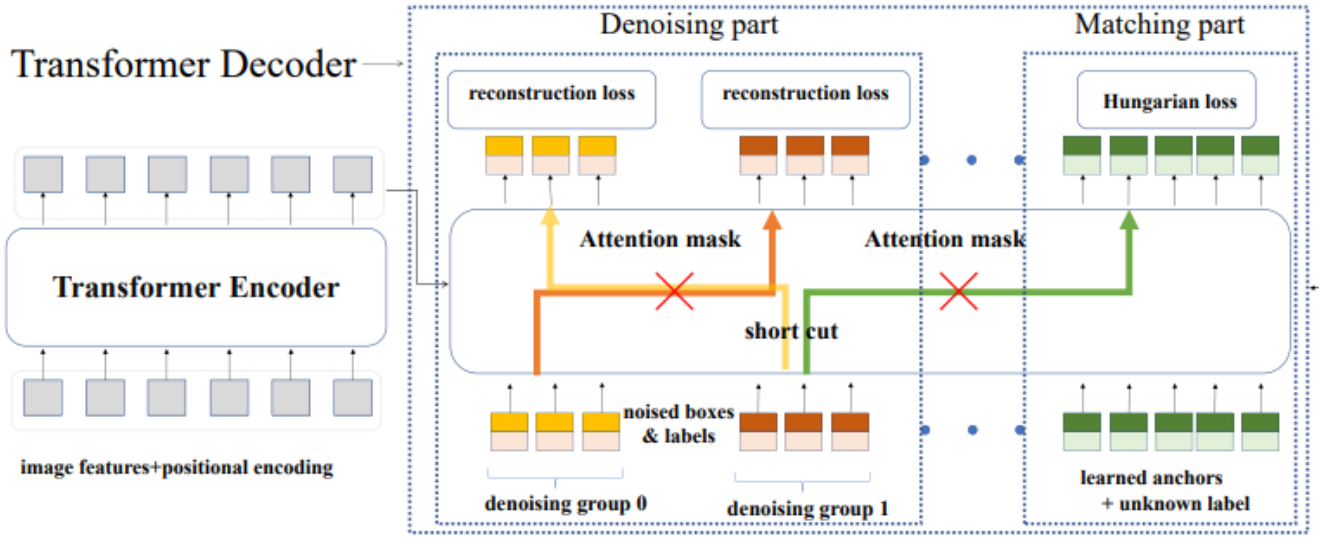
$$V_n^i = \begin{cases} m, & \text{if } O_n^i \text{ matches } T_m \\ -1, & \text{if } O_n^i \text{ matches nothing} \end{cases}$$

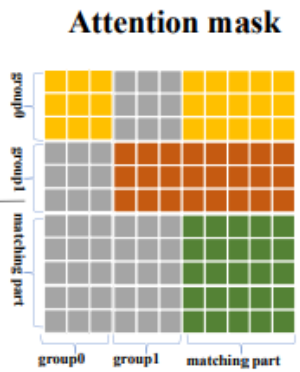
$$IS^i = \sum_{j=0}^{N} \mathbb{1}(V_n^i \neq V_n^{i-1})$$

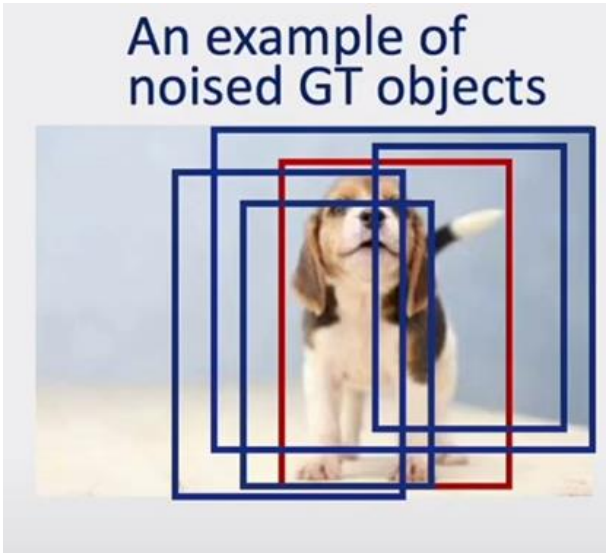


Denoising (DN) training

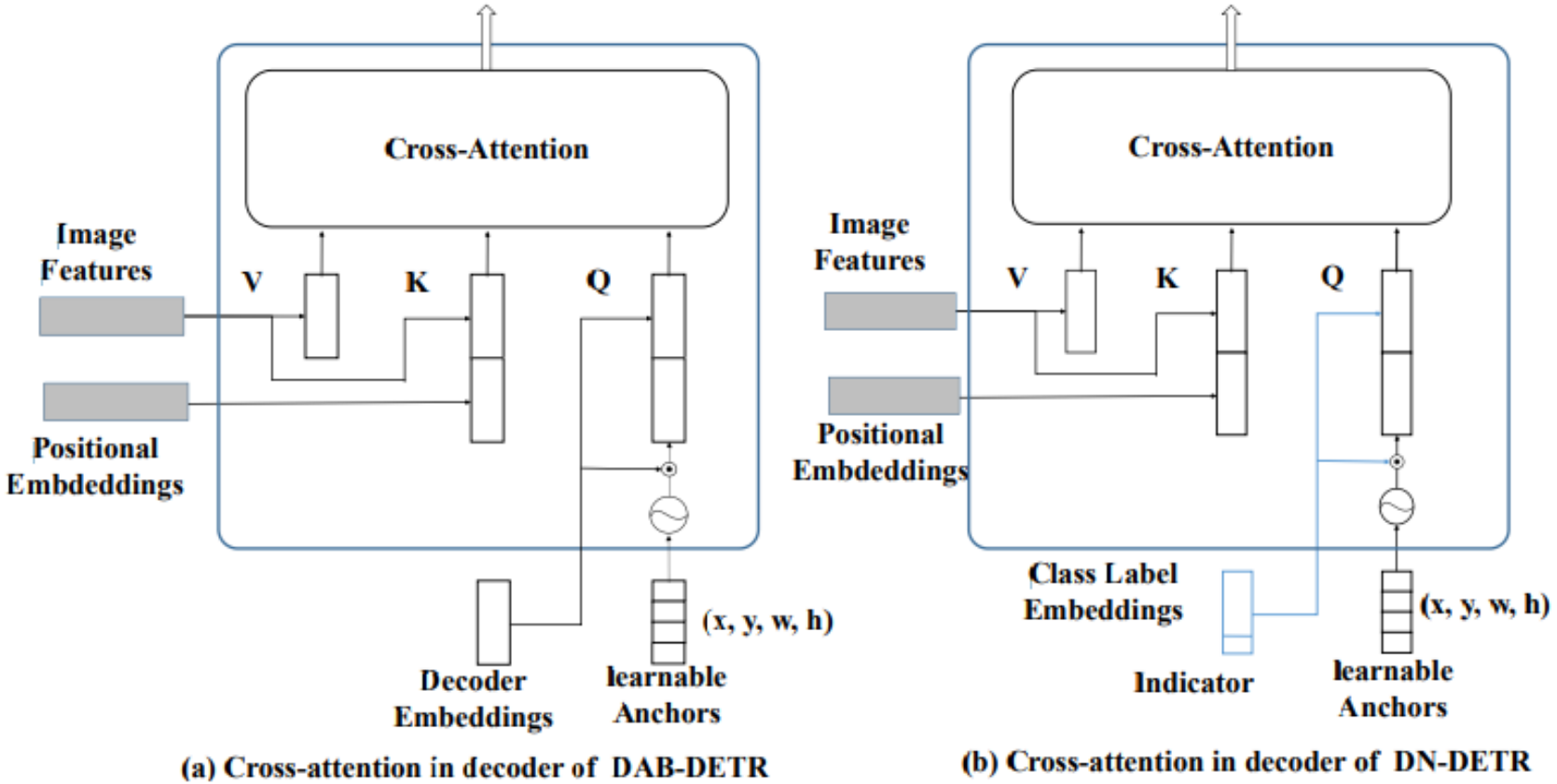
- Solution: solve **denoising task** additionally to matching







DN-DETR


Cross-Attention
Image Features
V
K
Q
Positional Embdeddings
Decoder Embeddings
learnable Anchors
(x, y, w, h)
(a) Cross-attention in decoder of DAB-DETR
Cross-Attention
Image Features
V
K
Q
Positional Embdeddings
Class Label Embeddings
Indicator
learnable Anchors
(x, y, w, h)
(b) Cross-attention in decoder of DN-DETR

Comparison

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 [18]	108	42.0	62.1	45.5	26.6	45.5	53.4	180	42M
Anchor DETR-R50 [21]	50	42.1	63.1	44.9	22.3	46.2	60.0	–	39M
Conditional DETR-R50 [15]	50	40.9	61.8	43.3	20.8	44.6	59.2	90	44M
DAB-DETR-R50 [14]	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 [18]	108	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
Anchor DETR-R101 [21]	50	43.5	64.3	46.6	23.2	47.7	61.4	–	58M
Conditional DETR-R101 [15]	50	42.8	63.7	46.0	21.7	46.6	60.9	156	63M
DAB-DETR-R101 [14]	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 [21]	50	44.2	64.7	47.5	24.7	48.2	60.6	151	39M
Conditional DETR-DC5-R50 [15]	50	43.8	64.4	46.7	24.0	47.6	60.7	195	44M
DAB-DETR-DC5-R50 [14]	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 [21]	50	45.1	65.7	48.8	25.8	49.4	61.6	–	58M
Conditional DETR-DC5-R101 [15]	50	45.0	65.5	48.4	26.1	48.9	62.8	262	63M
DAB-DETR-DC5-R101 [14]	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

Convergence

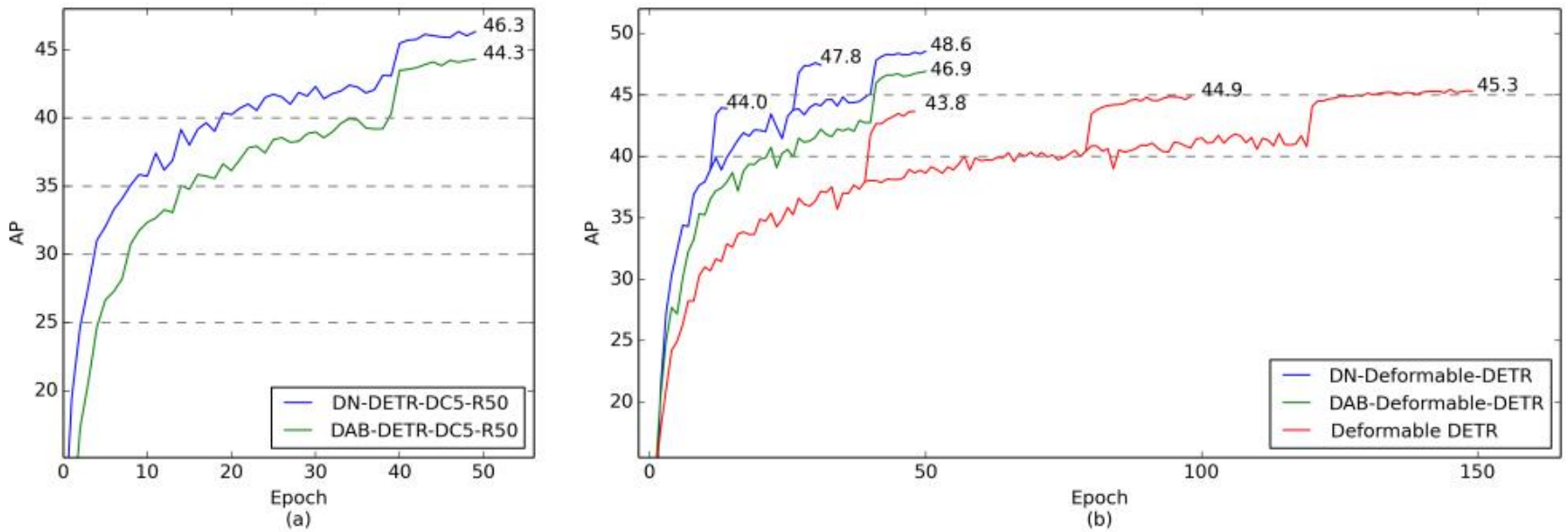
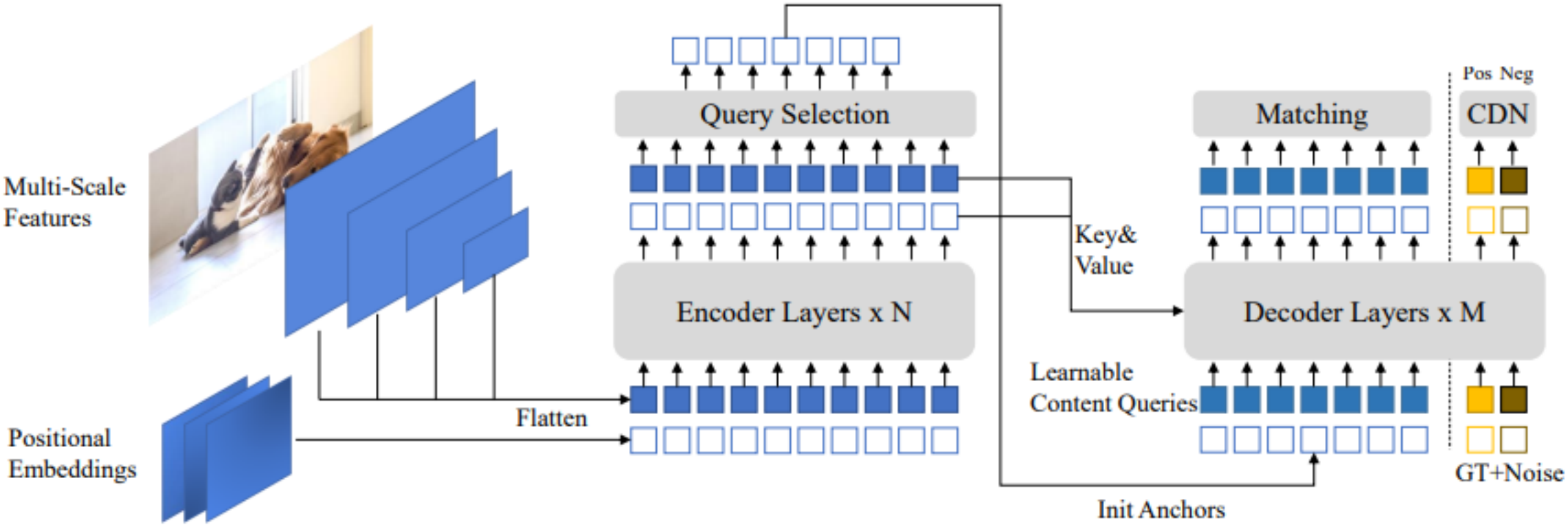


Figure 5. (a) Convergence curves of DAB-DETR and DN-DETR with ResNet-DC5-50. Before learning rate drop, DN-DETR achieves 40 AP in 20 epochs, while DAB-DETR needs 40 epochs. (b) Convergence curves of multi-scale models with ResNet-50. With learning rate drop, DN-Deformable-DETR achieves 47.8 AP in 30 epochs, which is 0.9 AP higher than the converged DAB-Deformable-DETR.

DINO

DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection (ICLR 2023)

DINO


Multi-Scale Features
Positional Embeddings
Flatten
Query Selection
Encoder Layers x N
Key& Value
Learnable Content Queries
Matching
Pos Neg
CDN
Decoder Layers x M
GT+Noise
Init Anchors

Contrastive denoising (CDN) training

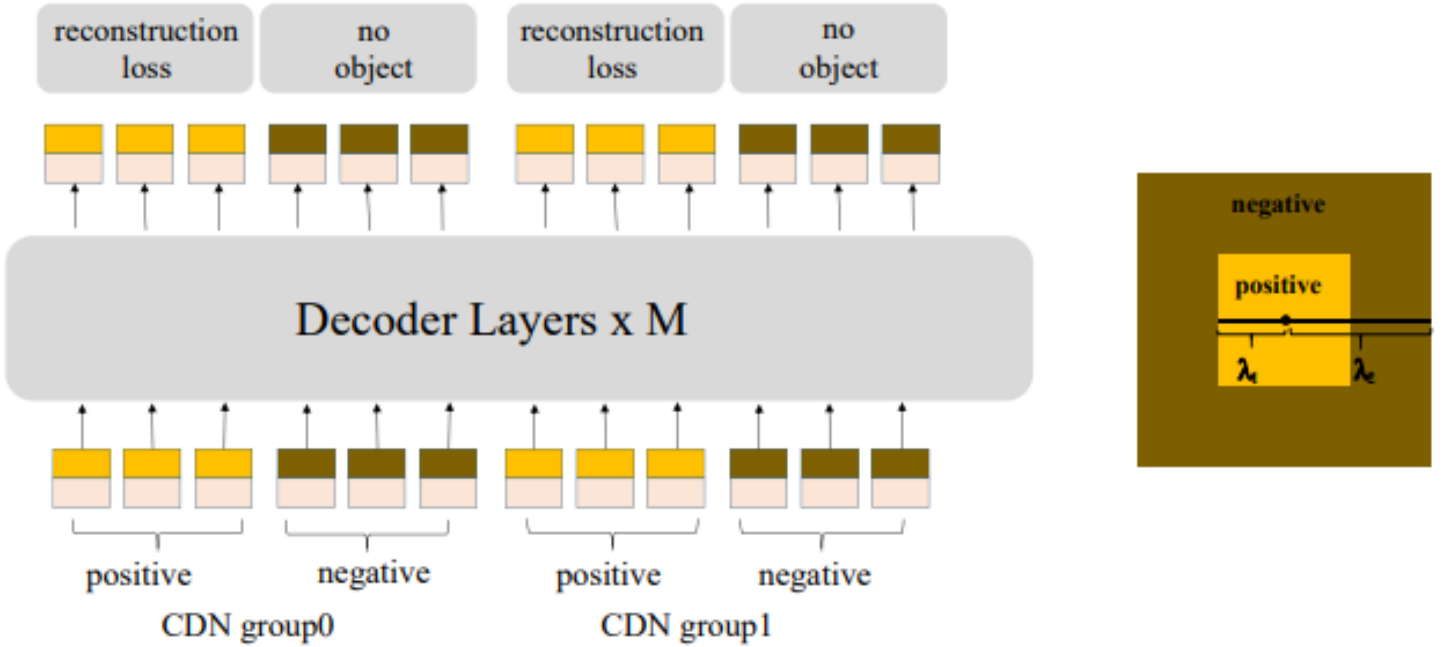
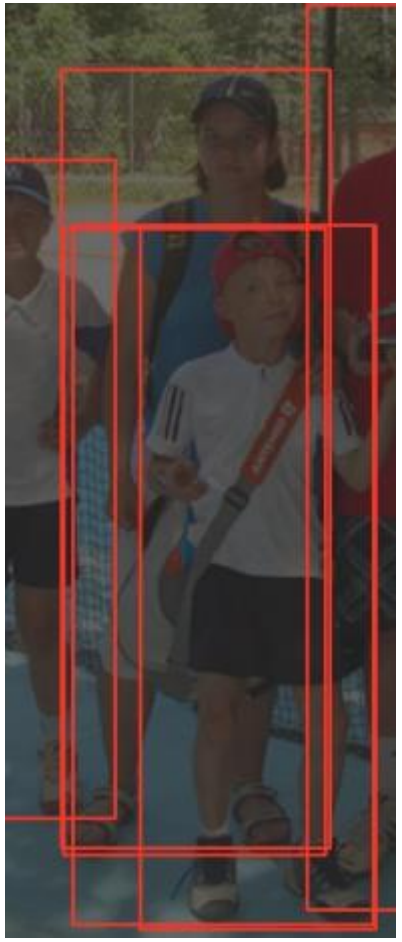


Figure 3: The structure of CDN group and a demonstration of positive and negative examples. Although both positive and negative examples are 4D anchors that can be represented as points in 4D space, we illustrate them as points in 2D space on concentric squares for simplicity. Assuming the square center is a GT box, points inside the inner square are regarded as a positive example and points between the inner square and the outer square are viewed as negative examples.

CDN reduces duplicates



DN

CDN

Mixed query selection

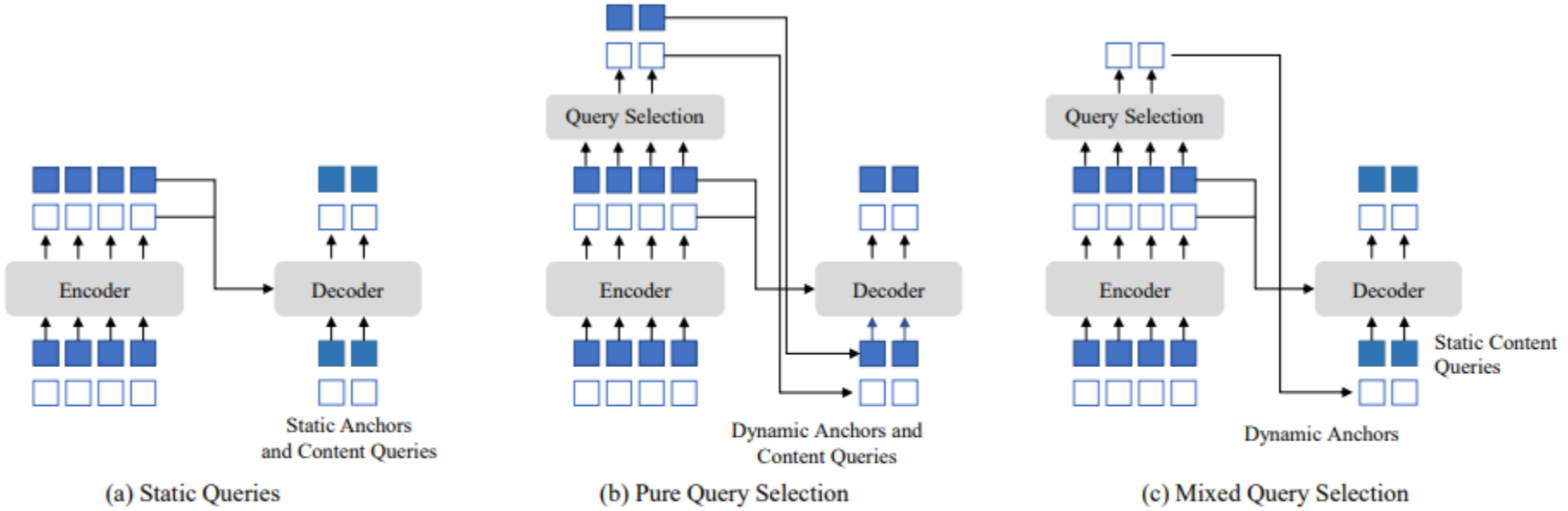


Fig. 5. Comparison of three different query initialization methods. The term "static" means that they will keep the same for different images in inference. A common implementation for these static queries is to make them learnable.

Comparison

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPS	Params	FPS
Faster-RCNN(5scale) [30]	12	37.9	58.8	41.1	22.4	41.1	49.1	207	40M	21*
DETR(DC5) [3]	12	15.5	29.4	14.5	4.3	15.1	26.7	225	41M	20
Deformable DETR(4scale)[41]	12	41.1	–	–	–	–		196	40M	24
DAB-DETR(DC5)† [21]	12	38.0	60.3	39.8	19.2	40.9	55.4	256	44M	17
Dynamic DETR(5scale) [8]	12	42.9	61.0	46.3	24.6	44.9	54.4	–	58M	–
Dynamic Head(5scale) [7]	12	43.0	60.7	46.8	24.7	46.4	53.9	–	–	–
HTC(5scale) [4]	12	42.3	–	–	–	–	–	441	80M	5*
DN-Deformable-DETR(4scale)† [17]	12	43.4	61.9	47.2	24.8	46.8	59.4	265	48M	23
DINO-4scale†	12	**49.0**(+5.6)	**66.6**	**53.5**	**32.0**(+7.2)	**52.3**	**63.0**	279	47M	24
DINO-5scale†	12	**49.4**(+6.0)	**66.9**	**53.8**	**32.3**(+7.5)	**52.5**	**63.9**	860	47M	10

Method	Params	Backbone Pre-training Dataset	Detection Pre-training Dataset	Use Mask	End-to-end	val2017 (AP)		test-dev (AP)	
						w/o TTA	w/ TTA	w/o TTA	w/ TTA
SwinL [23]	284M	IN-22K-14M	O365	✓		57.1	58.0	57.7	58.7
DyHead [7]	≥ 284M	IN-22K-14M	Unknown*			–	58.4	–	60.6
Soft Teacher+SwinL [38]	284M	IN-22K-14M	O365	✓		60.1	60.7	–	61.3
GLIP [18]	≥ 284M	IN-22K-14M	FourODs [18],GoldG+ [18,15]			–	60.8	–	61.5
Florence-CoSwin-H[40]	≥ 637M	FLD-900M [40]	FLD-9M [40]			–	62.0	–	62.4
SwinV2-G [22]	3.0B	IN-22K-ext-70M [22]	O365	✓		61.9	62.5	–	63.1
DINO-SwinL(Ours)	**218M**	IN-22K-14M	O365		✓	**63.1**	**63.2**	**63.2**	**63.3**

Convergence

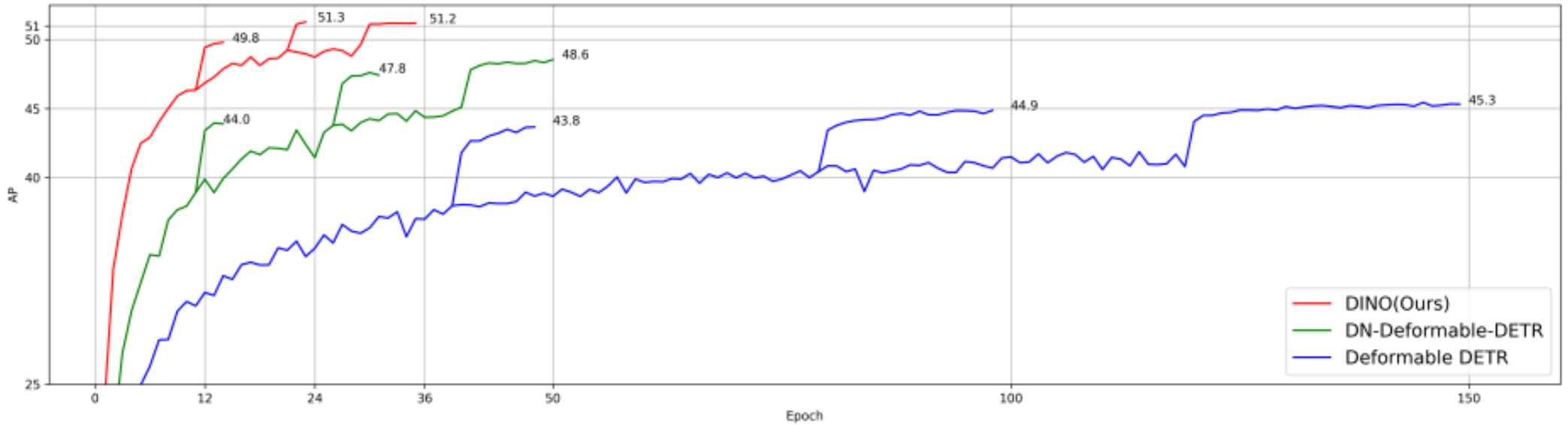


Fig. 7. Training convergence curves evaluated on COCO `val2017` for DINO and two previous state-of-the-art models with ResNet-50 using multi-scale features.

Ablation

#Row	QS	CDN	LFT	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1. DN-DETR [17]	No			43.4	61.9	47.2	24.8	46.8	59.4
2. Optimized DN-DETR	No			44.9	62.8	48.6	26.9	48.2	60.0
3. Strong baseline (Row2+pure query selection)	Pure			46.5	64.2	50.4	29.6	49.8	61.0
4. Row3+mixed query selection	Mixed			47.0	64.2	51.0	31.1	50.1	61.5
5. Row4+look forward twice	Mixed		✓	47.4	64.8	51.6	29.9	50.8	61.9
6. DINO (ours, Row5+contrastive DN)	Mixed	✓	✓	**47.9**	**65.3**	**52.1**	**31.2**	**50.9**	**61.9**

Conclusions

- DETR model converges too slowly
- Main reasons:
  - Inexplicit learnable positional queries
  - Bipartite matching
- Solutions:
  - For learnable positional queries use bboxes
  - In training learn additional tasks – e.g. CDN