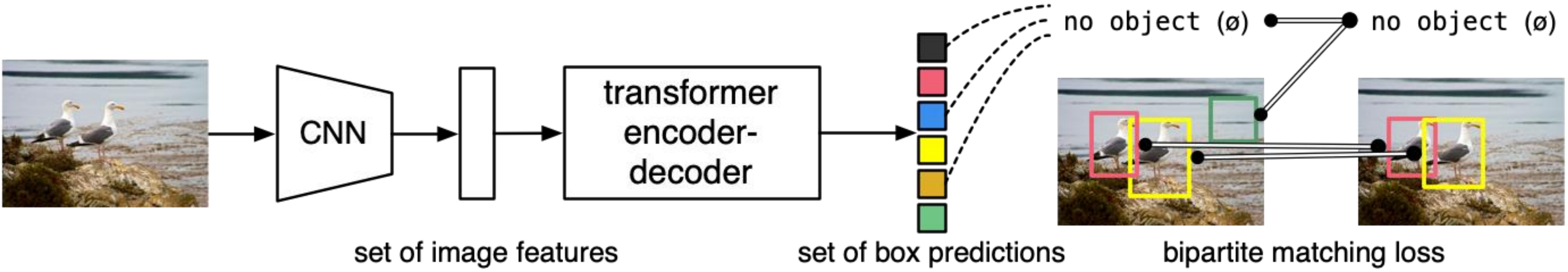
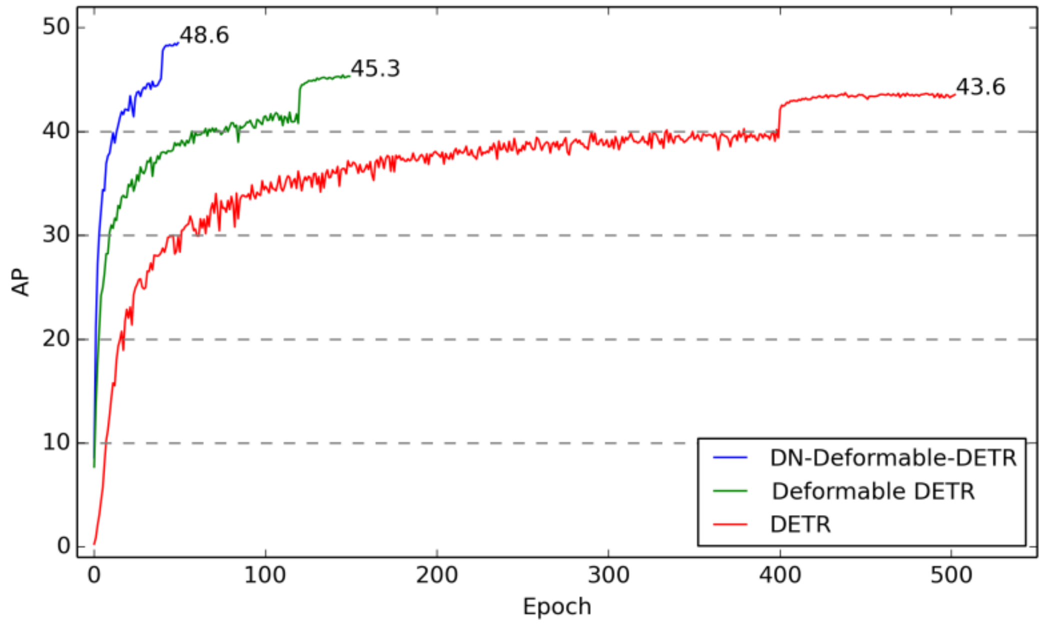
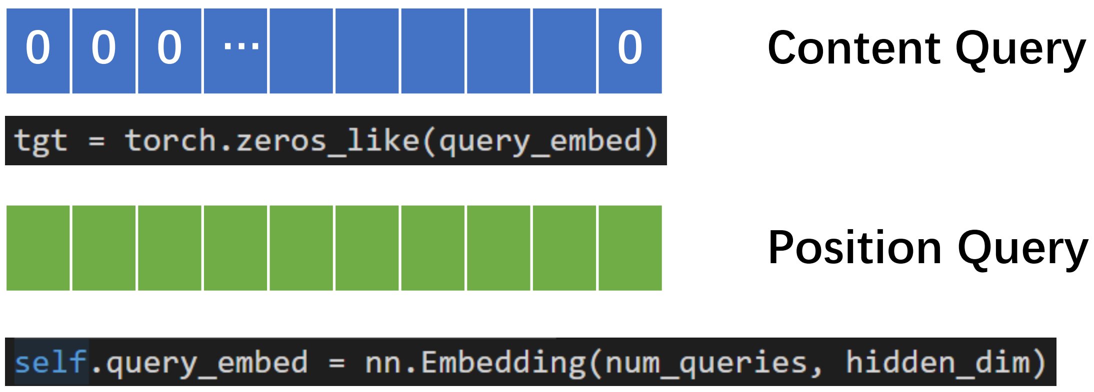
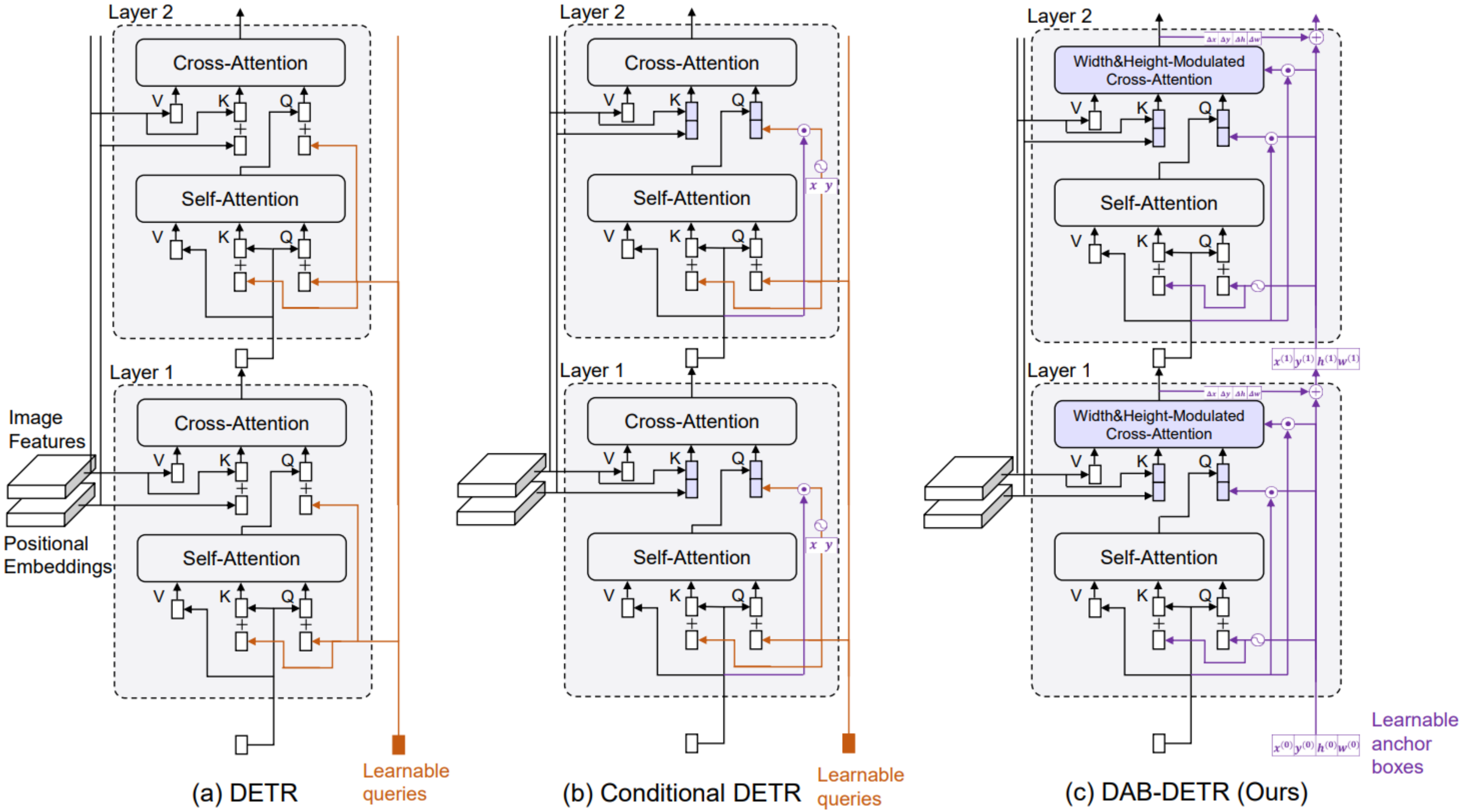
DN-DETR: Accelerate DETR Training by Introducing Query DeNoising

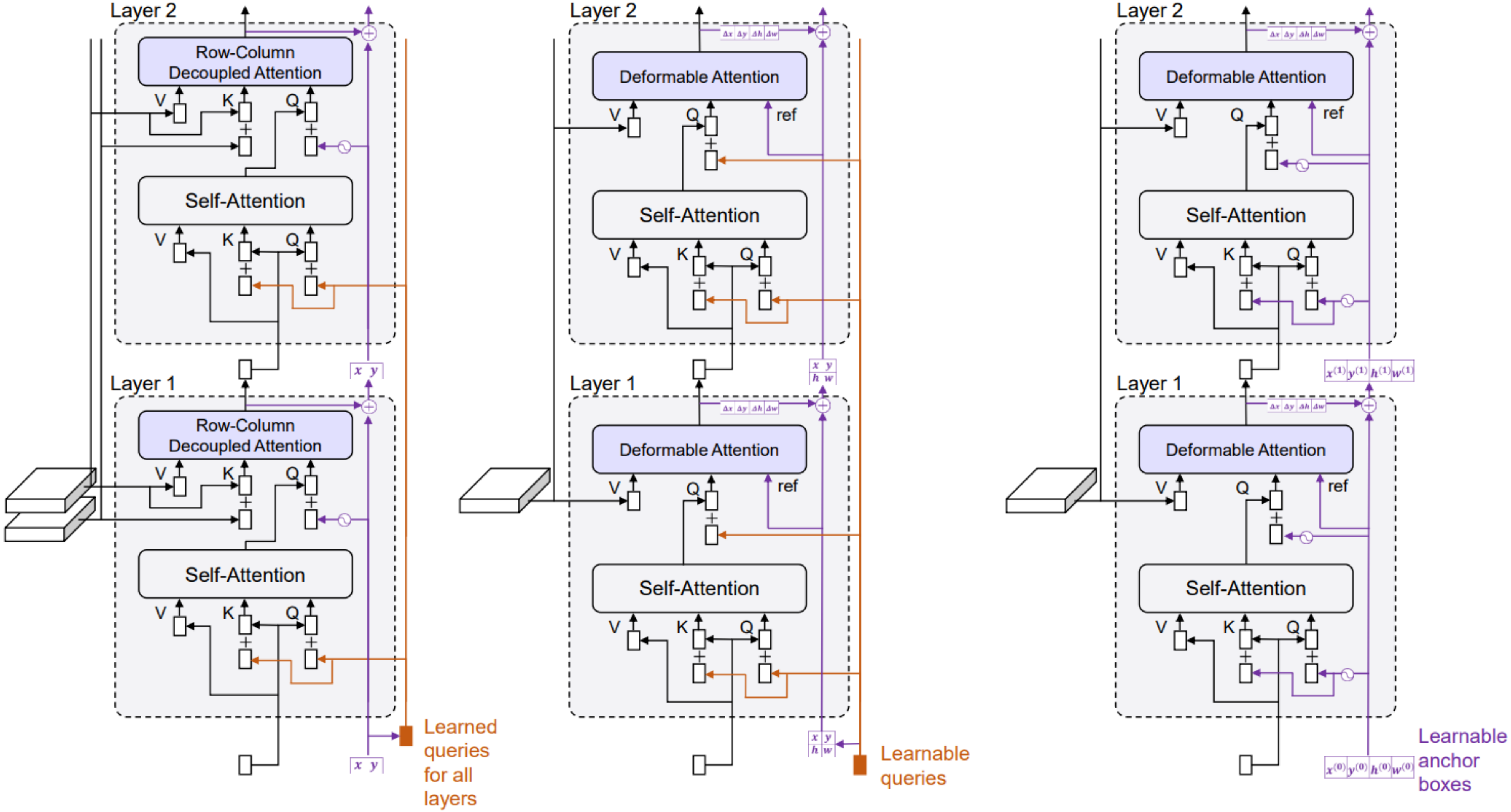
Feng Li[1,2*†], Hao Zhang[1,2*†], Shilong Liu[2,3†], Jian Guo[2], Lionel M. Ni[1,4], Lei Zhang[2‡]
[1]The Hong Kong University of Science and Technology.
[2]International Digital Economy Academy (IDEA).
[3]Tsinghua University.
[4]The Hong Kong University of Science and Technology (Guangzhou).
{fliay,hzhangcx}@connect.ust.hk
{liusl20}@mails.tsinghua.edu.cn
{ni}@ust.hk
{guojian,leizhang}@idea.edu.cn


no object (ø)
no object (ø)
CNN
transformer encoder-decoder
set of image features
set of box predictions
bipartite matching loss


48.6
45.3
43.6
AP
Epoch
DN-Deformable-DETR
Deformable DETR
DETR


0 0 0 ··· 0
Content Query
tgt = torch.zeros_like(query_embed)
Position Query
self.query_embed = nn.Embedding(num_queries, hidden_dim)


Layer 2
Cross-Attention
Self-Attention
V
K
Q
Layer 1
Image Features
Positional Embeddings
Learnable queries
(a) DETR
(b) Conditional DETR
Width&Height-Modulated Cross-Attention
Learnable anchor boxes
(c) DAB-DETR (Ours)



(d) Anchor-DETR (e) Deformable DETR (f) DAB-Deformable-DETR (Ours)

DAB-DETR

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

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

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

$$\mathrm{ModulateAttn}((x, y), (x_{\mathrm{ref}}, y_{\mathrm{ref}})) = (\mathrm{PE}(x) \cdot \mathrm{PE}(x_{\mathrm{ref}}) \frac{w_{q,\mathrm{ref}}}{w_q} + \mathrm{PE}(y) \cdot \mathrm{PE}(y_{\mathrm{ref}}) \frac{h_{q,\mathrm{ref}}}{h_q}) / \sqrt{D}, \quad (6)$$

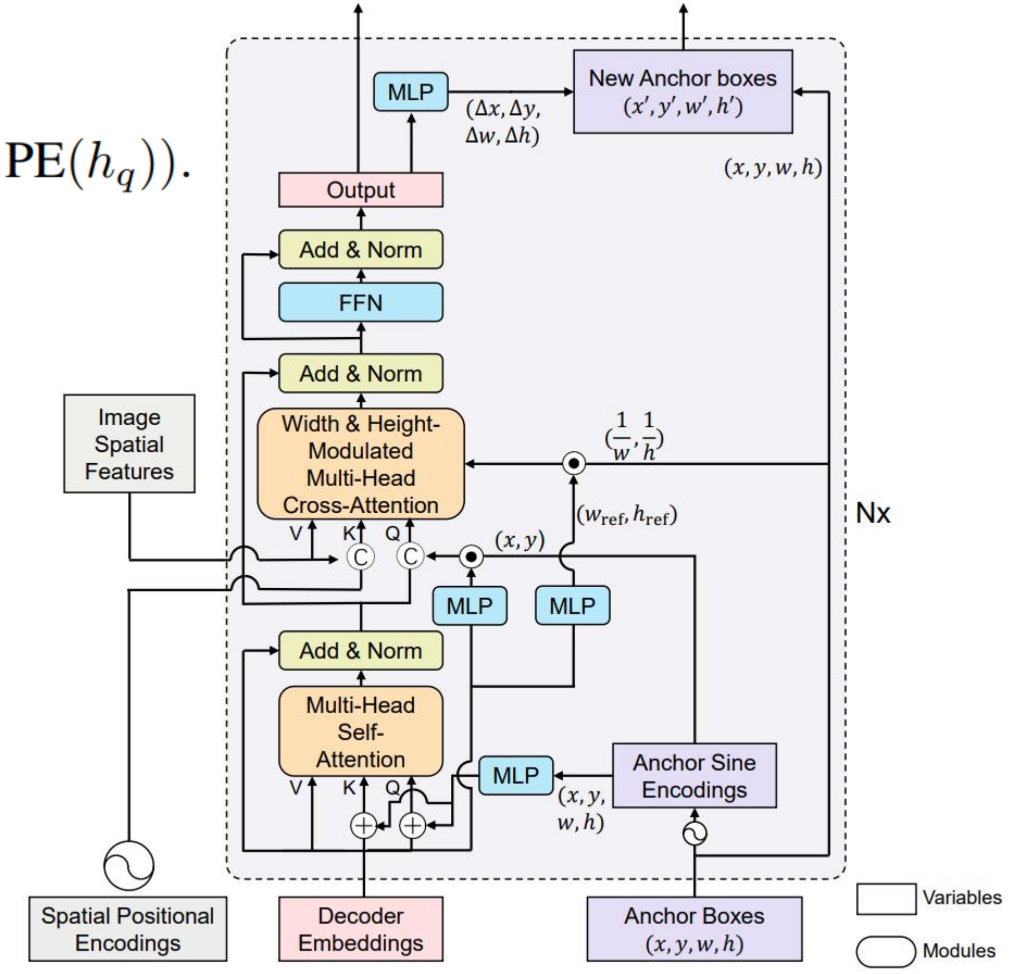
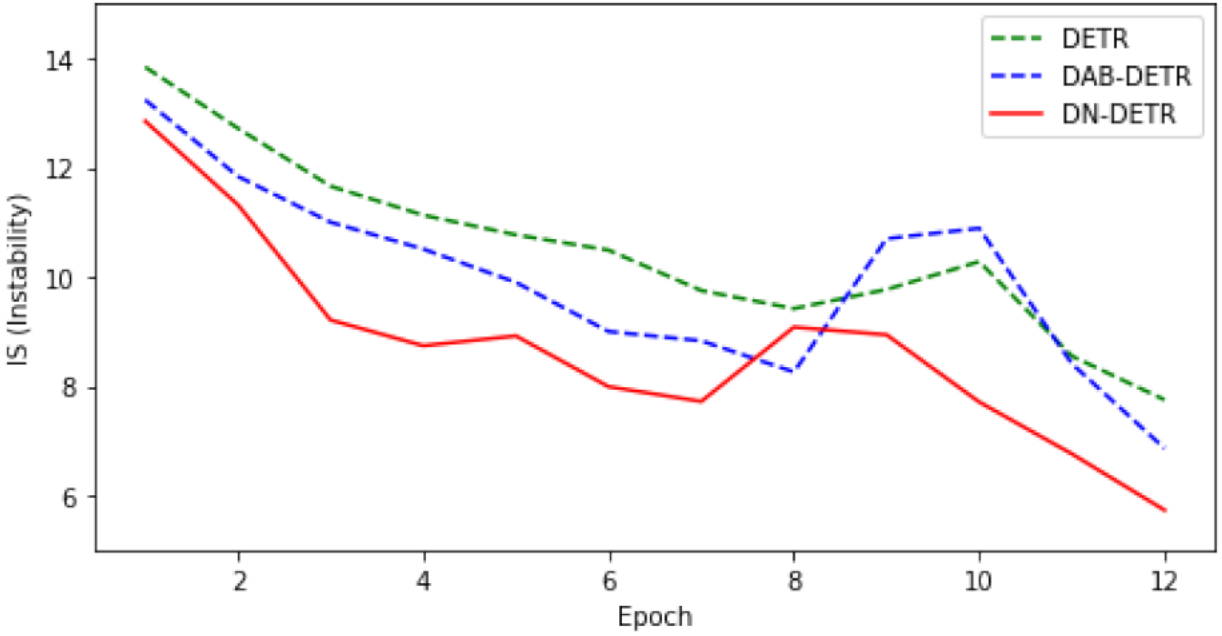


Figure 5: Framework of our proposed DAB-DETR.

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

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



	清洁浴室	打扫地板	洗窗
吉姆	$2	$3	$3
史提夫	$3	$2	$3
艾伦	$3	$3	$2

```
# Final cost matrix
C = self.cost_bbox * cost_bbox + self.cost_class * cost_class + self.cost_giou * cost_giou
C = C.view(bs, num_queries, -1).cpu()

sizes = [len(v["boxes"]) for v in targets]
indices = [linear_sum_assignment(c[i]) for i, c in enumerate(C.split(sizes, -1))]
return [(torch.as_tensor(i, dtype=torch.int64), torch.as_tensor(j, dtype=torch.int64)) for i, j in indices]
```

Box Denoising

noise scale of these 2 noises. For center shifting, we add a random noise $(\Delta x, \Delta y)$, to the box center and make sure that $|\Delta x| < \frac{\lambda_1 w}{2}$ and $|\Delta y| < \frac{\lambda_1 h}{2}$, where $\lambda_1 \in (0, 1)$

parameter $\lambda_2 \in (0, 1)$. The width and height of the box are randomly sampled in $[(1 - \lambda_2)w, (1 + \lambda_2)w]$ and $[(1 - \lambda_2)h, (1 + \lambda_2)h]$, respectively.

Box Denoising	Label Denoising	Attention Mask	AP
✓	✓	✓	43.4
✓		✓	43.0
		✓	42.2
✓	✓		24.0

Label Denoising

For label noising, we adopt label flipping, which means we randomly flip some ground-truth labels to other labels.

the ratio of labels to flip. The reconstruction losses are l_1 loss and GIOU loss for boxes and focal loss [9] for class labels as in DAB-DETR. We use a function $\delta(\cdot)$ to denote

Why Denoising accelerates DETR training

我们可以把decoder看成在学习两个东西:

- good anchors（anchor位置）(x, y, w, h)
- relative offset（相对偏移）$(\Delta x, \Delta y, \Delta w, \Delta h)$

decoder queries可以看成是anchor位置的学习，而不稳定的匹配会导致不稳定的anchor，从而使得相对偏移的学习变得困难。因此，我们使用一个denoising task作为一个**shortcut**来学习相对偏移，它**跳过了匹配**过程直接进行学习。如果我们把query像上述一样看作四维坐标，可以通过

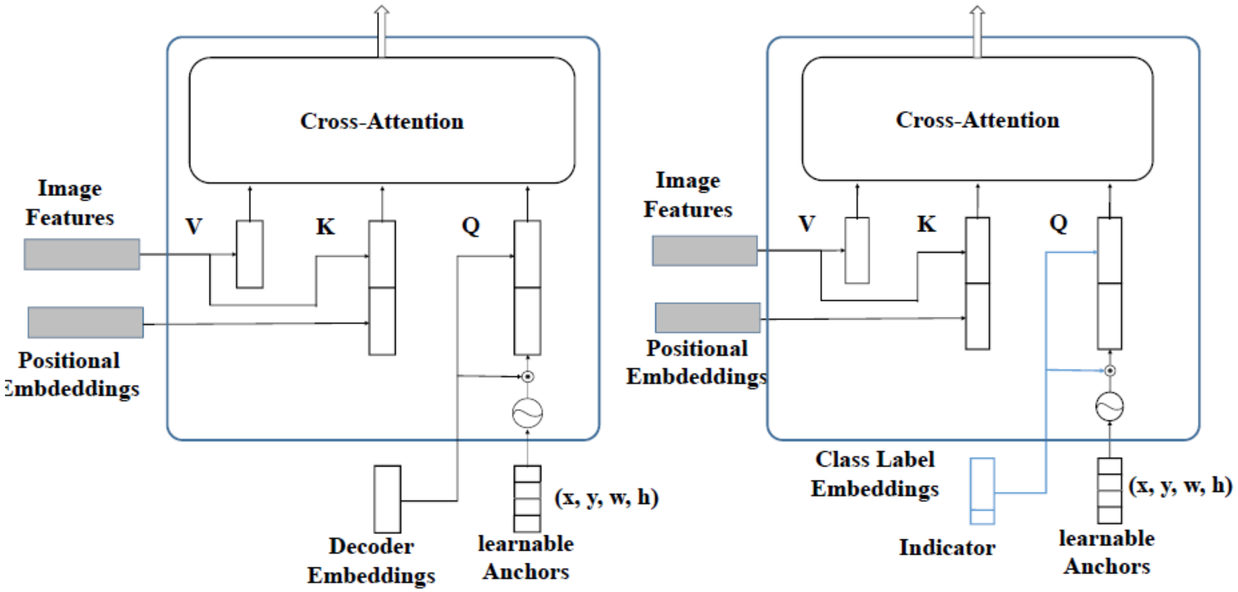
Denoising Group

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

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

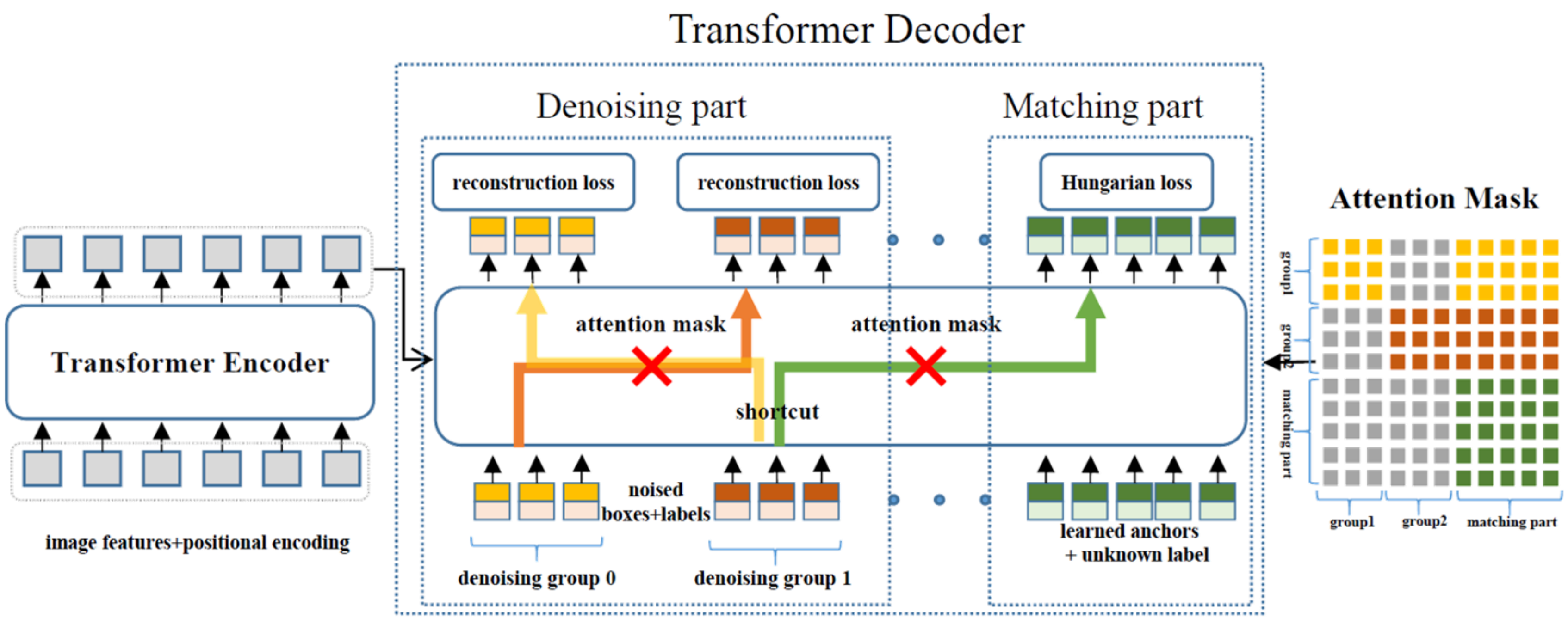
	No Group	1 Group	5 Groups
R50	42.2	43.4	44.1
R50-DC5	44.5	45.6	46.3
R101	43.5	45.0	45.2
R101-DC5	45.8	46.5	47.3

Label Embedding



(a) Cross-attention in decoder of DAB-DETR (b) Cross-attention in decoder of DN-DETR

Attention Mask



$$\mathbf{A} = [\mathbf{a}_{ij}]_{W \times W}$$

$$W = P \times M + N.$$

$$a_{ij} = \begin{cases} 1, & \text{if } j < P \times M \text{ and } \lfloor \frac{i}{M} \rfloor \neq \lfloor \frac{j}{M} \rfloor; \\ 1, & \text{if } j < P \times M \text{ and } i \geq P \times M; \\ 0, & \text{otherwise.} \end{cases}$$

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

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

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
DAB-DETR-DC5-R50		50	44.5	65.1	47.7	25.3	48.2	62.3	202	44M
DN-DETR-DC5-R50		25	44.4	64.5	47.3	24.4	48.0	63.0	202	44M
DAB-Deformable-DETR-R50	✓	50	46.9	66.0	50.8	30.1	50.4	62.5	195	48M
DN-Deformable-DETR-R50	✓	25	46.8	65.5	50.8	28.9	50.2	62.5	195	48M

those that do not support anchors like the vanilla DETR [1], we can do linear transformation to map 4D anchor boxes to the same latent space as for other learnable queries.

linear embedding将噪声框嵌入到与DETR query相同的维度中。由于 Vanilla DETR 没有明确的content part和position part，两部分信息是混在一起的，因此我们可以将label embedding和box embedding加在一起一起作为 DETR query。我们的初步结果表明，这些模型的收敛速度也有提高。