南京航空航天大學

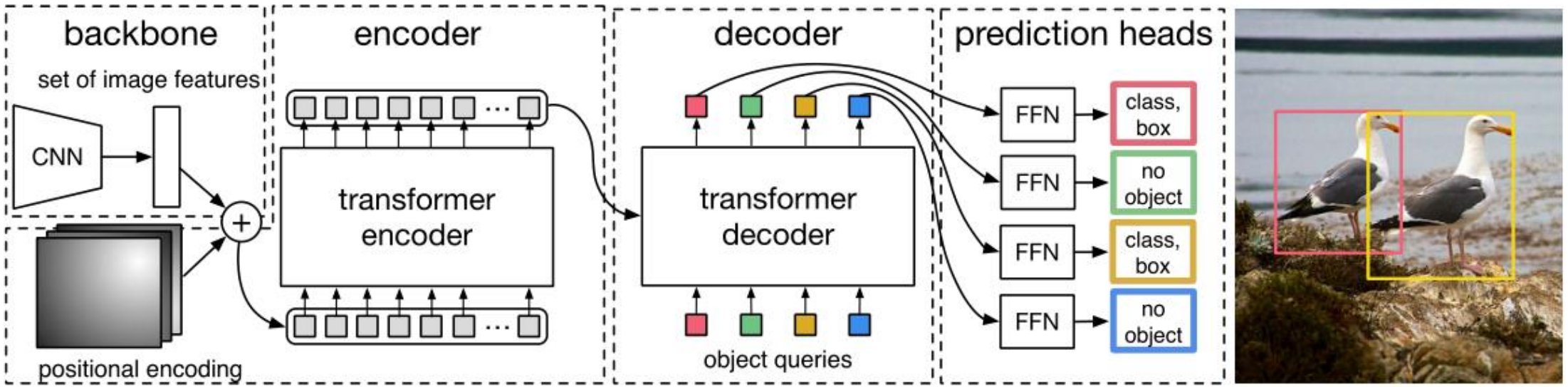
Nanjing University of Aeronautics and Astronautics

Conditional DETR for Fast Training Convergence

Depu Meng[1*] Xiaokang Chen[2*] Zejia Fan[2] Gang Zeng[2]
Houqiang Li[1] Yuhui Yuan[3] Lei Sun[3] Jingdong Wang[3†]
[1]University of Science and Technology of China [2]Peking University [3]Microsoft Research Asia

ICCV 2021

DETR



The DETR approach suffers from slow convergence on training, and needs 500 training epochs to get good performance.

Reasons for slow convergence of DETR

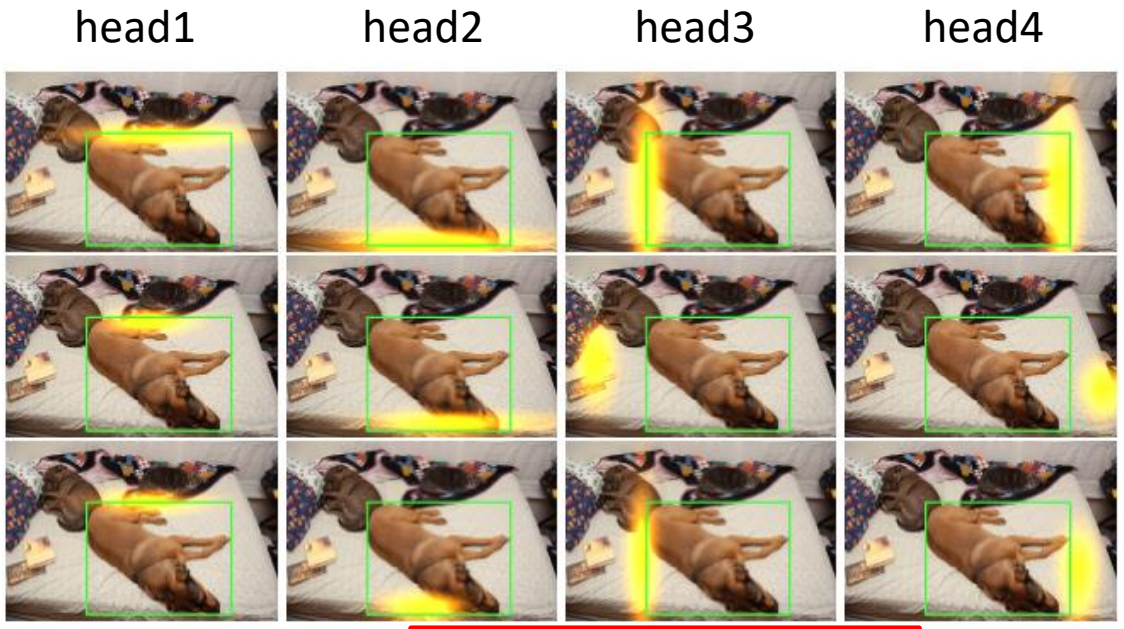


Figure 1. Comparison of spatial attention weight maps for our conditional DETR-R50 with 50 training epochs (the first row), the original DETR-R50 with 50 training epochs (the second row), and the original DETR-R50 with 500 training epochs (the third row). The maps for our conditional DETR and DETR trained with 500 epochs are able to highlight the four extremity regions satisfactorily. In contrast, the spatial attention weight maps responsible for the left and right edges (the third and fourth images in the second row) from DETR trained with 50 epochs cannot highlight the extremities satisfactorily. The green box is the ground-truth box.

The empirical results in DETR show that if removing the positional embeddings in keys and the object queries from the second decoder layer and only using the content embeddings in keys and queries, the detection AP drops slightly.

query：content embedding+spatial embedding
key：content embedding+spatial embedding

Conditional DETR

1.Decouple the cross-attention function of DETR decoder.
content embedding spatial embedding

2.We propose a conditional cross-attention mechanism with introducing conditional spatial queries for improving the localization capability and accelerating the training process.

Decouple the cross-attention function

The DETR decoder cross-attention mechanism takes three inputs: queries, keys and values.

key： content key $\mathbf{c}_k$ (the content embedding output from the encoder)
spatial key $\mathbf{p}_k$ (the positional embedding of the corresponding normalized 2D coordinate)

query： content query $\mathbf{c}_q$ (the embedding output from the decoder self-attention)
spatial query $\mathbf{p}_q$ (the object query o_q)

value： content query $\mathbf{c}_k$ (the content embedding output from the encoder)

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

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

Prediction

Box Regression： A candidate box is predicted from each decoder embedding as follows,

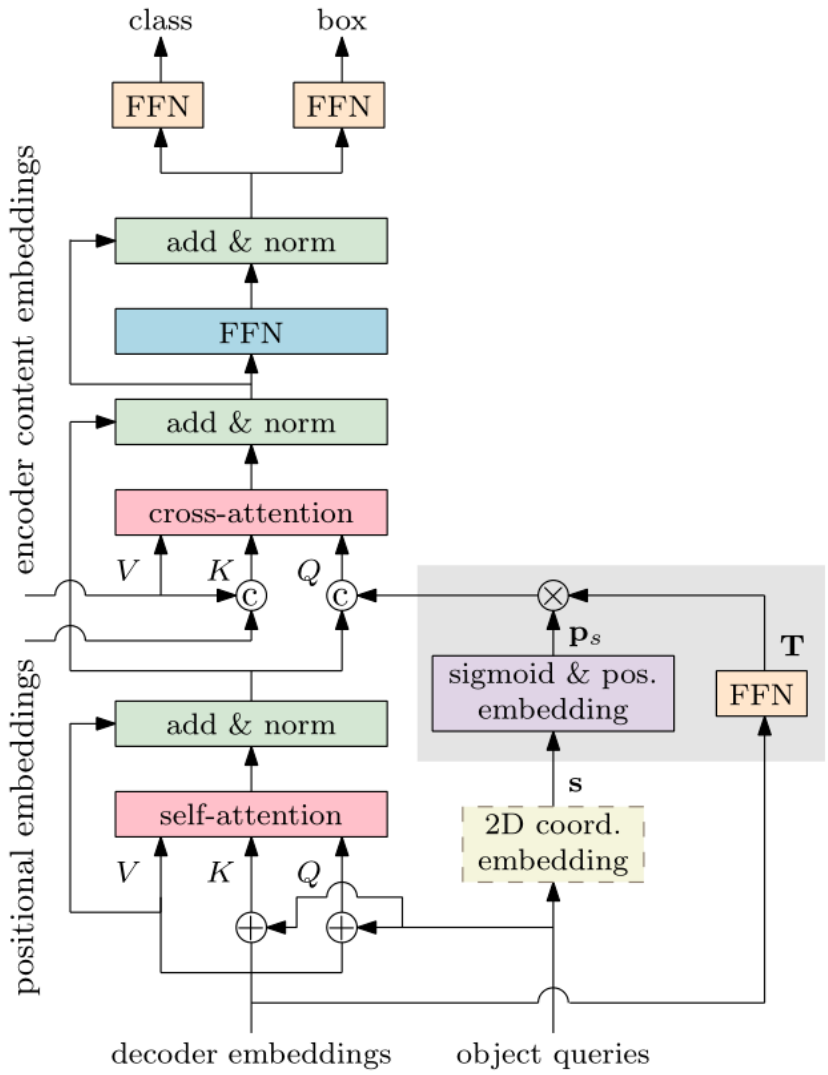
$$\mathbf{b} = \text{sigmoid}(\text{FFN}(\mathbf{f}) + [\mathbf{s}^\top\ 0\ 0]^\top).$$

b： a four-dimensional vector $[b_{cx}\ b_{cy}\ b_w\ b_h]^T$
f： decoder embedding
s： the unnormalized 2D coordinate of the reference point, and is (0, 0) in the original DETR.

In our approach, we consider two choices: learn the reference point s as a parameter for each candidate box prediction, or generate it from the corresponding object query.

Category prediction： e = FFN(f)

Conditional spatial query



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

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

$$\mathbf{s} = \text{FFN}(\mathbf{o}_q)$$

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

$$\mathbf{T} = \text{FFN}(\text{f})$$

$$\mathbf{p}_q = \mathbf{T}\mathbf{p}_q$$

Illustrating one decoder layer in conditional DETR.

Visualization

row1： the spatial attention weight maps $\longrightarrow p_q^{\mathrm{T}} p_k$

row2： the content attention weight maps $\longrightarrow c_q^{\mathrm{T}} c_k$

row3： the combined attention weight maps $\longrightarrow p_q^{\mathrm{T}} p_k + c_q^{\mathrm{T}} c_k$

(i) Translate the highlight positions to the four extremities and the position inside the object box: interestingly the highlighted positions are spatially similarly distributed in the object box.

(ii) Scale the spatial spread for the extremity highlights: large spread for large objects and small spread for small objects.

Experiment

Table 1. Comparison of conditional DETR with DETR on COCO 2017 `val`. Our conditional DETR approach for high-resolution backbones DC5-R50 and DC5-R101 is 10× faster than the original DETR, and for low-resolution backbones R50 and R101 6.67× faster. Conditional DETR is empirically superior to other two single-scale DETR variants. *The results of deformable DETR are from the GitHub repository provided by the authors of deformable DETR [53].

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

Experiment

Table 2. Results for multi-scale and higher-resolution DETR variants. We do not expect that our approach performs on par as our approach (single-scale, 16× resolution) does not use a strong multi-scale or 8× resolution encoder. Surprisingly, the AP scores of our approach with DC5-R50 and DC5-R101 are close to the two multi-scale and higher-resolution DETR variants.

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

Ablations

Table 3. Ablation study for the ways forming the conditional spatial query. CSQ = our proposed conditional spatial query scheme. Please see the first two paragraphs in Section 5.3 for the meanings of CSQ variants. Our proposed CSQ manner performs better. The backbone ResNet-50 is adopted.

Exp.	CSQ-C	CSQ-T	CSQ-P	CSQ-I	CSQ
GFLOPs	89.3	89.5	89.3	89.5	89.5
AP	37.1	37.6	37.8	40.2	40.9

(i) CSQ-P - only the positional embedding p_s,
(ii) CSQ-T - only the transformation T,
(iii) CSQ-C - the decoder content embedding f,
(iv) CSQ-I - the element-wise product of the transformation predicted from the decoder self-attention output c_q and the positional embedding p_s.

Thanks