KD-DETR: Knowledge Distillation for Detection Transformer with Consistent Distillation Points Sampling

Supplementary Material

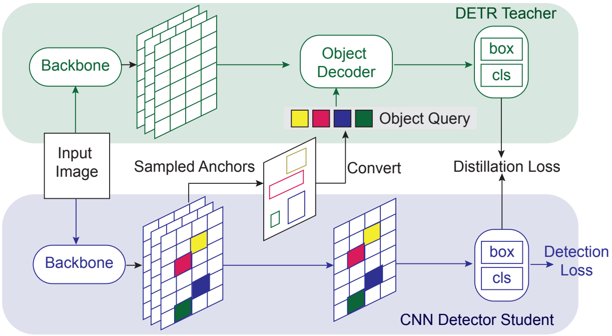

Figure 5. **Heterogeneous Distillation**

7. Heterogeneous Distillation

In heterogeneous distillation, the crucial part is to construct consistent distillation points between DETR and CNN-detector, indicating object queries and anchors respectively. KD-DETR propose the first idea in heterogeneous distillation by constructing the consistency between the object query and the anchor via spacial coordination: the anchor is generated through the sliding window strategy, and can be represented as $A = \{x_a, y_a, w_a, h_a\}$; while the object queries in most DETR, including DINO, are generated from anchor boxes: $Q = MLP(PE(cx, cy, w, h))$, where PE is positional encoding and MLP refers to a MLP projector. In this way, the anchor can be directly converted into the object query, and utilized as consistent distillation points:

$$Q_A = MLP(PE(x_a + \frac{w_a}{2}, y_a + \frac{h_a}{2}, w_a, h_a)) \qquad (7)$$

As shown on Figure 5, KD-DETR constructs distillation points by sampling anchors generated in CNN-detector (sampling details in Sec.4.2), then convert them to object queries of DETR via Eq. 7. With the predictions of distillation points from student and teacher, the distillation loss is Eq.3 and the total loss is Eq.4.

8. More Ablation Study and Analysis

8.1. Inheriting Stratgy

For DETR with multi-scale features, including Deformable DETR and DINO, we propose the inheriting strategy[12] by initialize the student's level embeddings with teacher's parameters. As shown in Table 8, inheriting strategy brings an additional 0.3% promotion on Deformable DETR Res18. Such phenomena also validate our analysis on consistent distillation points, as the level embeddings in DETR is a set of learnable embeddings for model to distinguish different scale of features, and are egocentirc. That is to say, on multi-scale DETR, the formulation of distillation points turns to $x = (I + LE, q)$, where LE denotes the level embeddings. In this way, inheriting the level embeddings from teacher to student can restrict the consistency of distillation points.

Model	Arch	AP	AP_{50}	AP_{75}
Deformable DETR	Res50	44.5	63.6	52.6
Deformable DETR	Res18	40.1	58.1	43.7
Ours	Res18	43.4	61.8	47.5
Ours†	Res18	**43.7**	**62.1**	**47.7**

Table 8. Level Embedding Inheriting Strategy on Deformable DETR. † means using inheriting strategy

Student	Arch	AP	AP_{50}	AP_{75}
DINO	Swin-T	50.7	67.9	55.0
Ours	Swin-T	52.6	70.3	57.5
Gains		**+1.9**	**+2.4**	**+2.5**
DINO	Swin-B	55.6	74.3	60.8
Ours	Swin-B	57.1	75.5	62.5
Gains		**+1.5**	**+1.2**	**+1.7**

Table 9. Distillation on DINO with Swin Transformer backbone

8.2. Generalization on Advanced Backbone

To validate the extensibility of KD-DETR, we conduct additional experiments with DINO Swin Transformer[] as backbone. As shown in Table 9, with a strong baseline, KD-DETR significantly boosts the performance of student models. For Swin-Tiny as student and Swin-Base as teacher, KD-DETR promotes the student's COCO mAP from 50.7% to $52.6\%(+1.9\%)$; For Swin-Base as student and Swin-Large as teacher, KD-DETR promotes student from 55.6% to $57.1\%(+1.5\%)$.

8.3. Distillation on the Transformer Layers

Besides the scale of backbone, the layer number of transformer encoder and decoder is also an important factor of the model size and computation cost in DETR. In this paper, we also conduct experiments to compress the layer numbers

Enc/Dec	AP	AP_{50}	AP_{75}	FPS	Params
6/6	36.2	56.1	37.9	76	31M
Ours	41.4(**+5.2**)	61.4	44.2	76	31M
2/6	36.2	56.3	38.4	102	27M
Ours	39.0(**+2.8**)	58.9	41.7	102	27M
6/2	31.8	49.8	33.5	82	25M
Ours	38.9(**+7.1**)	58.0	41.6	82	25M
2/2	29.3	46.6	37.0	113	17M
Ours	32.8(**+3.5**)	52.6	34.2	113	17M

Table 10. **Distillation on Transformer Layers**: Compressing the number of encoder layers and decoder layers with KD-DETR

of transformer to validate the scalability of KD-DETR. The FPS reported is measured on a single Nvidia A100 GPU.

Table 10 shows the results of KD-DETR on DAB-DETR, with backbone of ResNet-50 as teacher and ResNet-18 as student. While decreasing the number of transformer layers will cause great degradation in the performance, KD-DETR can significantly boost the student model. For example, the student model with 2 encoder layers and 6 decoder layers can outperform the full-scale model for 2.8$\%$ mAP with 1.2x FPS improvement.