Appendix: Decoupled DETR For Few-shot Object Detection

Zeyu Shangguan[0000−0003−1435−6959], Lian Huai *, Tong Liu, Yuyu Liu, and Xingqun Jiang

BOE Technology Group Co. Ltd., Beijing, China
{shangguanzeyu,huailian,liutongcto,liuyuyu,jiangxingqun}@boe.com.cn

1 Deformable attention

The DETR baseline suffers from a slow convergence rate. To address this issue, we applied deformable attention, following the approach in DINO [2]. We utilize a single-layer deformable attention as the framework of our decoupling module. The concept of deformable attention was initially introduced in Deformable DETR [1]. In order to expedite the convergence process of DETR, the deformable attention module only samples a few points around the reference points as the k value of the attention. The equation for deformable attention is in Eq. 1.

$$DeformAttn(z_q, p_q, x) = \sum_{m=1}^{M} W_m[\sum_{k=1}^{K} A_{mqk} \cdot W_m^{'} x(p_q + \Delta p_{mqk})] \tag{1}$$

- z_q: content query
- p_q: 2-D reference point
- k: the sampled keys
- K: total sampled key number
- Δp_{mqk}: sampling offset
- A_{mqk}: attention wight of the k-th sampling point in the m-th attention head

As a variation of the standard Transformer attention that provides a pre-filtering mechanism for the reference points; it is equivalent to the standard Transformer only if the sampling reference points traverse all possible locations.

2 Effectiveness of decoupling modules

The effectiveness of our decoupling module is primarily determined by the ratio of novel and base instances. This is because the gradient updating process on the novel branch is most effective when the loss originates from the novel instances, which is also true for the base branch. Therefore, we record the ratio of the three cases depicted in Sec. 3.3 on the PASCAL VOC split1, resulting in:

* Corresponding author

	Trainable parameters	Model size
DETR baseline	23,746,994	455.9MB
DeDETR	26,360,696	487.3MB
Meta-DETR	10,704,028	306.0MB
Meta-DeDETR	11,649,788	317.2MB

Table 1: Extra computation overhead of our methods

$$base(case1) : novel(case2) : both(case3) = 10 : 3 : 1, \tag{2}$$

which indicates that the decoupling process is primarily influenced by the conditions of Case 1 and Case 2, resulting in a substantial increase in accuracy. Additionally, we expect that manually dividing the training sample into Case 1 and Case 2 will enhance the effectiveness of the decoupling module.

3 Extra computation overhead

Since we have introduced a additional trainable module in our decoupling module, we report the extra computation overhead of our model for reference, details are listed in Tab. 1

4 Performance on base classes.

Our experimental results on VOC split1 5-shot indicate that compared to the baseline method, our decoupled module benefits to **both** base and novel classes, but **mostly** on the novel classes. As shown in Tab. 2. The average loss value for base categories is lower than for novel categories, resulting in a lower gradient and, consequently, relatively smaller improvement than the few-shot novel categories.

	bAP	nAP
DETR baseline	69.2	49.8
DeDETR (our)	72.4 (+3.2)	59.1 (+9.3)
Meta-DETR	73.8	59.2
Meta-DeDETR (our)	76.4 (+2.6)	65.1 (+5.9)

Table 2: Performance on base and novel classes.

5 Generalization of baseline

We chose ResNet101 as the backbone to align with the existing settings and ensure a fair comparison with previous works. Additionally, ResNet101 reaches the upper limit of our computing resources. We also experimented with smaller backbones, such as ResNet50 based on VOC split1 5-shot, shown in Tab. 3, which resulted in a slight decrease in the overall performance of our model. However, the improvements introduced by our method remained consistent and steady, demonstrating its robustness across different backbone architectures.

	R50	R101
DETR baseline	48.1	49.8
DeDETR	57.9 (+9.8)	59.1 (+9.3)
Meta-DETR (our)	57.6	59.2
Meta-DeDETR (our)	63.2 (+5.6)	65.1 (+5.9)

Table 3: Performance on different backbone.

6 Discussion

6.1 Incremental setting

Regarding how the decoupling module will work in an incremental setting, there may be multiple stages requiring the model to adapt to new categories using only a few samples. For this incremental setting, for example, assume we still have 15 base categories that are many-shot, and 5 novel categories that are few-shot in stage #1, and we have additional 5 novel categories in stage #2. For stage #1, which is the same case in our paper, our decoupling module will assign a base category feature extractor for the 15 base categories and assign a novel category feature extractor for the 5 novel categories. When it goes to stage #2, both the base and novel category feature extractor will be kept, but the only difference is that the novel category feature extractor will respond for all 10 few-shot categories. The reason is that, in our mechanism, the base category feature extractor is responsible for the categories that have abundant training data, and the novel category feature extractor is for the categories that have limited training data.

6.2 Top N predictions

One may consider the performance of the adaptive decoder layer selection mechanism compared to a simpler approach of considering the top N predictions

from each decoder layer. However, selecting the Top N candidates based on confidence values is a non-learning approach. During the early stages of training or fine-tuning, the confidence scores predicted by the decoder tend to be relatively unreliable. As a result, relying on these scores in the early training stage could introduce additional challenges for the model, making convergence more difficult.

References

1. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: CVPR. pp. 13609–13617 (2022)
2. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., Shum, H.Y.: DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In: The Eleventh International Conference on Learning Representations (2023)