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

Table 1. Results of our method trained for 25 epochs and our baseline method trained for 50 epochs under the same settings. The results shows we achieve 2x acceleration with denoising training.

7. Appendix

7.1. Acceleration Analysis

We show how much our method can speedup training exactly in Table 1. Our method achieves results comparable to the baseline with only half of the training epochs, resulting in 2x acceleration.

7.2. The training wall clock time and GFLOPs

We tested the training wall clock time and GFLOPs with 8 NVIDIA A100 GPUs as shown in Table 2. The total training time is calculated by multiplying the number of training epochs and the training time for each epoch. The training time per epoch is 51.1min and 57.7min for DAB-DETR-R50 and DN-DAB-DETR-R50, respectively. While denoising training introduces minor training cost increase, it only needs about half number of training epochs (25 epochs) to achieve the same performance as DAB-DETR-R50. The practical training speedup is indeed remarkable.

Model	Total Training time (min)	Training GFLOPs
DAB-DETR-R50	2555(50 epochs)	94.4
DN-DAB-DETR-R50	1443(25 epochs)	94.5

Table 2. We adopted five denoising groups for DN-DAB-DETR. The results are tested on the same GPUs for fair comparison.

7.3. Other tasks and future work

7.3.1 Other Tasks

In addition to regular detection, our design of queries as anchor box + label makes the detection model capable of handling other tasks. For example, known objects detection and known labels detection. Note that results shown in this section are just a preliminary exploration and not based on our well trained model with best hyper-parameters.

Known Objects Detection: Assume we know a part of the objects in an image and want to predict the remaining objects. We want the known objects to help predict the unknown objects through co-occurrence relations. We did some preliminary exploration. We randomly divide the 80 classes of MSCOCO2017 into 2 parts including known classes and unknown classes. We put objects of known classes in denoising part and want the matching part to predict the objects of the unknown classes. We do not use attention mask so that the matching part can get useful information from the denoising part. Our experimental results is shown in Table 3. Compared with the evaluation without known boxes, the known objects evaluation improves the performance, which indicates that co-occurrence helps the prediction of unknown boxes. Moreover, our DN-DETR trained with known objects exceeds DAB-DETR only trained on unknown classes when evaluating without known objects. This means the denoising of extra boxes from extra (known) classes also helps the performance on the unknown objects.

Method	Seting	AP	AP(Cond)
DAB-DETR	0.7/0.3	38.4	-
DN-DETR	0.7/0.3	42.1	42.9
DAB-DETR	0.5/0.5	37.8	-
DN-DETR	0.5/0.5	39.1	40.3

Table 3. Extra label prediction on COCO. We split the annotation of COCO class into known/unknown classes, where objects of known classes only appear in denoising part, and we evaluate the performance on the unknown classes. Cond means the result is evaluated with known objects.

Known Labels Detection: For each image, we assume we know all the class labels in the image without box information. Since our model has interpreted the query embedding into class label embedding, we can seamlessly utilize these known labels to detect the boxes of each class label. For each class c in the image, we concatenate its label embedding with the indicator 1 which denotes known label. We feed the concatenated vector into the decoder and let the decoder output all boxes of class c. To compare with methods without known boxes. We concatenate outputs of all classes and evaluate the result as shown in Table 4. Within 10 epochs of finetuning on pretrained DN-DETR, the known label detection performance is improved to 46.6.

7.3.2 future work

There are two potential future works to be mentioned here. One is zeroshot detection and the other is progressive inference.

Zeroshot Detection : Since we have decoupled decoder

Method	Seting	AP
DAB-DETR	no knwon labels	42.2
DN-DETR	no knwon labels	43.4
DN-DETR	known label (1ep)	43.8
DN-DETR	known label (10ep)	46.6

Table 4. Known label detection results under ResNet-50 with 1 denoising group. 1ep and 10ep means finetuned 1 or 10 epochs from pretrained DN-DETR.

queries as anchor boxes and class labels, pretrained class label embeddings can be fed into the class label part of the queries. To enable zeroshot detection, one can take 80 classes of MSCOCO as phrases and collect phrase embeddings from a pretrained language model as the class label embedding. With the pretrained label embedding, it is possible to train a given class detector which takes a class label embedding as input and detect objects of the given classes. In inference time, class label embeddings from unseen classes can be fed into the decoder to achieve zeroshot detection.

Progressive inference Based on the known objects detection, a progressive inference method can be designed. For example, we can train a DN-DETR capable of doing known objects detection. In inference time, we let the detector predict objects and then, we can choose the objects with highest score and treat them as known objects to do known objects detection. For each step of prediction, we choose objects with highest score and add them to the known boxes set. After repeating for many times, we get the final prediction.