Dense Distinct Query for End-to-End Object Detection(Supplementary Material)

Shilong Zhang[1,3 *], Xinjiang Wang[2 *], Jiaqi Wang[1], Jiangmiao Pang[1],
Chengqi Lyu[1], Wenwei Zhang[4,1], Ping Luo[3,1], Kai Chen[1]

[1]Shanghai AI Laboratory [2] SenseTime Research

[3] The University of Hong Kong [4] S-Lab, Nanyang Technological University

In the supplementary material, the analysis of dense distinct queries (DDQ) to Deformable DETR [8] is first sketched in Sec. 1. The details about auxiliary loss are added in Sec. 2. Sec. 3 gives more detailed ablation studies and analysis of pyramid shuffle. Sec. 6 gives more detailed ablation studies about the number of queries and refining stages in DDQ R-CNN. Sec. 4 show the details about DDQ R-CNN with encoder. Sec. 5 elaborate and improved Deformable DETR and discuss the difference with DINO. Sec. 7 gives the latency benchmark. Sec. 8 reports the results of DDQ R-CNN with other ways to construct the query. Sec. 9 show the results of traditional detectors with different IOU thresholds on CrowdHuman. At last, Sec. 10 illustrates our social impact.

1. Analysis of DDQ in Deformable DETR

For fast verification of Distinct Queries Selection(DQS) in such a heavy model, we adopt the standard 1x setting on COCO and keep other hyperparameters (such as learning rate and weight decay) the same with that in Deformable DETR [8].

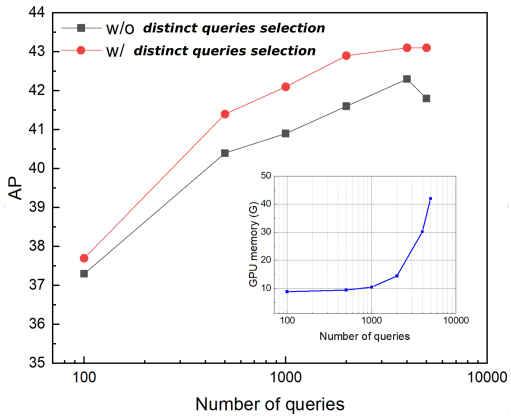

Figure 1. The performance comparison of Deformable DETR with and without distinct queries selection

As shown in Fig.1, when we increase the number of queries in Deformable DETR, there is a similar trend as it in Sparse R-CNN [4], as shown in the main manuscript. When the number of queries naively increases without distinct queries selection, the performance increases at the beginning but decreases as the number of queries reaches ~ 5000. It is due to the more difficult training with more similar queries out of the dense queries. By imposing a distinct queries selection pre-processing to filter out similar queries and keeping only distinct queries before each stage of iterative refinement, the performance is improved with a clear margin, and the performance margin consistently increases along with more queries.

Therefore, we believe Dense Distinct Queries (DDQ) is a principle of designing an object detector with a fast convergence based on recent end-to-end detectors.

2. Auxiliary Loss for Dense Queries

We follow the TOOD [3] to design our auxiliary loss. We select K samples with the smallest cost of each ground truth as positive samples. P means the index set of positive samples which correspond to the same ground truth. The classification score target of a sample i in this set is

$$\frac{score_i * IoU_i^6}{Max(score_j * IoU_j^6)_{j\in P}} * Max(IoU_j)_{j\in P} \quad (1)$$

The GIoU loss of each sample is reweighted by the classification target. The classification loss and regression loss weight keep consistent with the main loss weight (1 and 2 respectively) for distinct queries. The performance is quite stable for DDQ FCN when K ranges between 5 and 16. We adopt 8 and 4 for DDQ FCN and DDQ DETR respectively in this study. The auxiliary loss also works for refining heads in DDQ RCNN. Due to time issues, we will supplement relevant results in the future version.

3. More Analysis and Ablation for Pyramid Shuffle

We provide an in-depth analysis of pyramid shuffle. Firstly, we compare it with 3D MAX Filter in DeFCN [5]

in Table. 1. Then we visualize the change of score maps in different levels after adding pyramid shuffle operations in Fig. 2. At last, Table. 3 gives the results under different shuffle channels.

Table 1. **Comparison between 3D MAX Filter and Pyramid Shuffle.** * means the results is unstable

Flops	Parameters	Operations	AP
-	-	-	41.0*
12.2 G	0.59M	Conv&GN&Relu&MaxPool3d	41.2
0.2 G	0.00M	Shuffle x3	**41.5**
12.2 G	0.59M	Conv&GN&Relu& Shuffle x3	**42.0**

Comparision with 3D MAX Filter Table. 1 shows the comparison between pyramid shuffle and 3D MAX Filtering in DeFCN. DeFCN believes there should be extra parameters and max pooling operation to facilitate the optimization under the one-to-one assignment. However, we argue that only the interaction of cross-level queries matters, and extra parameters or max operations are unnecessary. Our pyramid shuffle is more lightweight and with better performance. When we add an extra convolution to regression branches to fair compare with the 3D MAX Filter, we can surpass it by 0.8 AP.

Visualization of Adjacent Level Score Map Fig. 2 shows the scores map of adjacent levels. The left side of each subfigure(with blue background) shows score maps with pyramid shuffles, and the corresponding right side (with yellow background) means without pyramid shuffles. The top-left corner marks the feature level. The red circle represents the duplication predictions in the adjacent level. We can find pyramid shuffle effectively reduces the cross-level high score false positives.

Results under Different Shuffle Channels Table 3 gives the results under different shuffle channels; we can find that the number of channels can even be reduced to 16 when there is already cross-level distinct queries selection, making it more lightweight. When the number of shuffle channels is 128, which means no remaining channels for the current level, the performance will dramatically drop 1.5 AP because of missing information on the current level queries in the interaction.

Table 2. Performance of one-stage DDQ with different numbers of shuffle channels.

Number	AP	AP_{50}	AP_{75}
0	41.0*	59.9	45.1
8	41.2	60.3	45.4
16	41.3	60.6	45.4
32	41.4	60.6	45.5
64	**41.5**	60.9	45.4
96	41.6	61.1	45.7
128	39.9	59.9	43.6

3.1. Number of Pyramid Shuffle Operations

We report the results of DDQ FCN with the different number of pyramid shuffle operations in the classification and regression branches. When no pyramid shuffle is adopted in the DDQ FCN, its performance is unstable and fluctuates between 40.8 AP and 41.1 AP. We report an average performance of 41.0 AP. Even though there has been a cross-level distinct queries selection operation, compared to adopting only 2 and 1 operations to two branches respectively, there is still a 0.5 AP drop.

Table 3. Different number Pyramid shuffle operations in DDQ FCN. Cls means the classification branch and Reg means the regression branch. * indicate the performance is unstable

Cls	Reg	AP	AP_{50}	AP_{75}
0	0	41.0*	59.9	45.1
1	0	41.3	60.1	45.6
0	1	41.3	60.6	45.4
0	2	41.3	60.0	45.6
2	0	41.2	60.1	45.5
1	1	41.3	60.6	45.8
2	1	**41.5**	60.9	45.4
2	2	41.4	60.6	45.5
4	4	41.5	61.1	45.6

4. DDQ R-CNN with Encoder

The encoder can provide a more powerful feature representation for the decoder head which is explored in [2, 6, 8]. We simply add 6 dynamic blocks in DyHead [1] as our encoder. 500 queries and 3 refinement stages are adopted. It is observed in the main manuscript that there is about 3 AP improvement for DDQ R-CNN.

5. Details of Improved DeformableDETR and Comparison with DINO

DINO [7] adopt some techniques that significantly improve the Deformable DETR. We remove the CDN and mix query selection from DINO to form our baseline. DDQ is a concurrent work of DINO. The contrastive denoising training (CDN) is not intended to relieve the optimization difficulty of very similar queries among dense queries. In their implementation, the generated positive and negative samples in each pair are always significantly distinct from each other. Mix query selection increases the distinctness of queries by additionally initializing content embeddings, but the position embeddings are still created from top-k dense regression predictions which can be very similar and still hinder the optimization. We have shown our components can be combined with DINO.

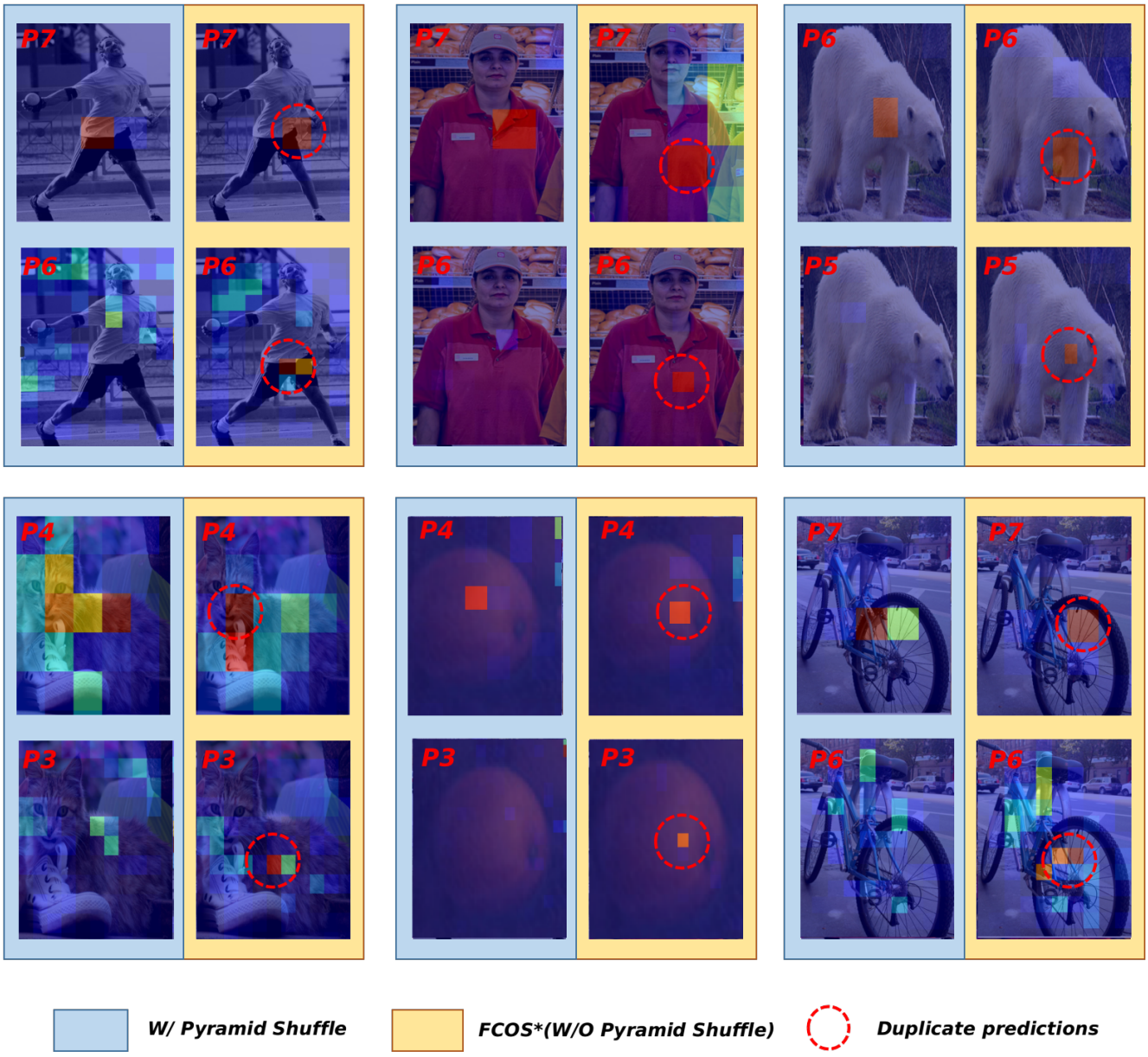

Figure 2. **Visualization of score map of adjacent levels**. We visualize classification scores with the rainbow color system. The left side of each subfigure(with blue background) shows score maps with pyramid shuffles, and the corresponding right side (with yellow background) means without pyramid shuffles. The top-left corner marks the feature level. The red circle represents the duplication predictions in the adjacent level.

Table 4. Latency(ms) of different models with batch size 1

DDQ FCN	Cascade R-CNN	Sparse R-CNN	Deformable DETR	**DDQ R-CNN**	DINO	**DDQ DETR**
44.8 AP	44.3 AP	45.0 AP	46.2 AP	**48.1** AP	50.9 AP	**52.0** AP
22.4 ms	28.5 ms	31.0 ms	40.0 ms	**31.3** ms	46 ms	**58** ms

6. Number of Queries and Stages in DDQ R-CNN

We analyze the combination of different numbers of stages and queries for DDQ R-CNN. Table. 5 shows that the best number of stages is proportional to the number of queries. This is easy to understand. When the number of queries increases, the newly added queries are of low quality, and more stages are needed to refine these queries. It is

worth emphasizing that we use 2 stages and 300 queries to trade off the performance and latency. When using 3 stages with the same number of queries, our method achieves an even higher performance of 45.1 AP on MS COCO.

Table 5. Performance with different number of refine stages(S) and queries(Q).

	100 Q	200 Q	300 Q	400 Q
S=1	43.0	43.2	43.4	43.2
S=2	44.2	44.2	**44.6**	44.8
S=3	44.4	44.8	45.1	44.9
S=4	44.5	45.0	45.0	45.2

7. Latency Benchmark

As for the details of latency calibration, We compare the speed (forwarding + post-processing) of different methods with batch size 1 in Table. 4. All evaluations were performed on Tesla A100 GPU with Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz. The Pytorch version is 1.9.0 with CudaToolkit 11.1 and Cudnn 8.0.5. An average of 200 iterations during model inference is adopted as the latency reported in this study.

8. Impact of Query Construction in DDQ R-CNN

We try five ways to construct queries in DDQ R-CNN. As shown in Table. 6, None means all query embeddings are set to a zero tensor, and the refinement stages only get meaningful query bounding boxes. This attempt reduces the performance to 43.2 AP. Simply constructing queries from the FPN results in 1.0 AP degradation. Reg means only using the last feature map of the regression branch, which drops the performance by 0.6 AP. Constructing queries from the last feature map in the classification branch can be an alternative as it can get a comparable performance(only 0.3 AP drop).

Table 6. Impact of Query Construction in DDQ RCNN

	AP	AP_{50}	AP_{75}
None	43.2	61.0	47.9
FPN	43.6	61.7	48.0
Cls	44.3	62.2	48.6
Reg	44.0	62.5	48.3
Cls&Reg	**44.6**	63.0	48.8

9. DQS with Different IoU Threshold in CrowdHuman

In this section, we show the robustness of distinct queries selection(DQS) with different IoU thresholds in CrowdHuman. We can find there is a clear performance bottleneck for traditional detector ATSS even with carefully adjusting the threshold of NMS.

Table 7. Performance of DDQ on COCO when DQS adopts different IoU thresholds. Results of ATSS adopting different IoU thresholds in post-processing are also reported. None means we remove DQS or post-processing from the inference pipeline.

CrowdHuman	0.5	0.6	0.7	0.8	0.9	None
DDQ FCN	88.0	91.8	**92.7**	92.8	92.3	91.7
DDQ RCNN	91.8	92.9	**93.5**	93.3	93.2	93.2
ATSS	88.3	**89.6**	88.4	85.3	78.9	42.7

10. Social Impact

The potential social impact of this work inherits from object detection. Because human behaviors often cause crowded scenes, and DDQ achieves excellent performance in such scenes, it may be applied to some applications that violate human privacy, such as surveillance.

References

[1] Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L.: Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7373–7382 (2021) 2

[2] Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L., Zhang, L.: Dynamic detr: End-to-end object detection with dynamic attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2988–2997 (2021) 2

[3] Feng, C., Zhong, Y., Gao, Y., Scott, M.R., Huang, W.: Tood: Task-aligned one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3510–3519 (2021) 1

[4] Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al.: Sparse r-cnn: End-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14454–14463 (2021) 1

[5] Wang, J., Song, L., Li, Z., Sun, H., Sun, J., Zheng, N.: End-to-end object detection with fully convolutional network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15849–15858 (2021) 1

[6] Wang, X., Zhang, S., Yu, Z., Feng, L., Zhang, W.: Scale-equalizing pyramid convolution for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) 2

[7] Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., Shum, H.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In: International Conference on Learning Representations 2

[8] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020) 1, 2