Cascade-DETR: Delving into High-Quality Universal Object Detection (Supplemental material)

Mingqiao Ye[1*] Lei Ke[1,2*] Siyuan Li[1] Yu-Wing Tai[3]
Chi-Keung Tang[2] Martin Danelljan[1] Fisher Yu[1]
[1]ETH Zürich [2]HKUST [3]Dartmouth College

1. Appendix

In this supplementary material, we first present detailed training convergence curve comparison of Cascade-DINO in Section 1.1. We also provide more studies on deformable attention and alternative training manners. In Section 1.2, we then show qualitative results comparisons of our Cascade-DN-DETR to DN-DETR [3], including localization quality, occlusion cases and multi-object attention map. Finally, we provide more implementation/training details in Section 1.3.

1.1. Supplementary Experiments

Convergence Speed Comparison In Figure 1, we show the detection results comparison between Cascade R-CNN [1], DINO [8] (Baseline) and Cascade-DINO (Ours) per training epoch. We sample three dataset components UVO [6], BDD [7] and Braintumor [2] out of UDB10, which belong to various domains in open-world, self-driving and medical analysis. On UVO and BDD datasets, we find that Cascade R-CNN converges faster in the first three epochs, but its performance is quickly saturated and surpassed by our Cascade-DINO after training for 3 epochs. Cascade-DINO achieves stable performance growth during training, and outperforms Cascade R-CNN and DINO consistently in all three domains.

Comparison to Deformable Attention We compare Cascade attention (CA) with Deformable attention (DA) in Tab. 1, where merely replacing the Deformable attention in DINO to Cascade attention (*i.e.* **w/o** QR) promotes the results from 30.2 to 31.9 on UVO.

Comparison to Box-constrained Deformable Attention In Table 2, we compare our cascade attention to both the standard deformable attention and box-constrained deformable attention. We design box-constrained deformable attention by normalizing its learnable offsets within the predicted bounding boxes. Box-constrained deformable attention slightly increases the result of standard deformable attention by 0.8 AP. However, its performance is still 1.7 AP lower than our cascade-attention. This gap may be due to the insufficient/limited sampling points of deformable attention in the box regions.

Adopting GT boxes in the Initial Training Stage We try an alternative training manner to replace the predicted boxes with GT boxes for constraining attention in the initial training stage. The intuition is that at the beginning stage, the predicted boxes by learnable queries are inaccurate, which can be replaced by the corresponding GT box via greedy matching. We take Cascade-DINO as baseline and set the ratio of using GT boxes at the first iteration being 25%. This ratio then linearly decreases to 0 at the last training iteration after 12 epochs. However, we find it leads to 3.0 AP performance decrease (32.7 $\rightarrow$ 29.7) on UVO dataset compared to using predicted boxes for the whole network training stage.

1.2. More Qualitative Comparisons

Visualization Comparison in Box Quality In Figure 2, we compare the predicted box quality between baseline DN-DETR and our Cascade-DN-DETR. On COCO validation set, we visualize the predicted boxes satisfying IoU (to GT) thresholds 0.5 and 0.75 in the first row and second row respectively. The predicted boxes differences are highlighted in red. The detected lower-quality predictions by the baseline, such as 'toilet' (left example) and 'refrigerator' (right example) are highlighted in the red boxes.

Visualization Comparison in Occlusion Cases In Figure 3, comparing to DN-DETR, we find that Cascade-DN-DETR has a higher recall rate for detection in the challenging heavy occlusion cases. The box-constrained cascade attention can better attend to the target occluded objects with less distraction from their neighboring overlapping objects.

Visualization of Multi-object Attention Map We provide cross-attention maps for multiple objects in Fig. 4. Cascade-DN-DETR's query attention focuses on the most relevant parts of the detected objects, while DN-DETR has a more scattered attention distribution.

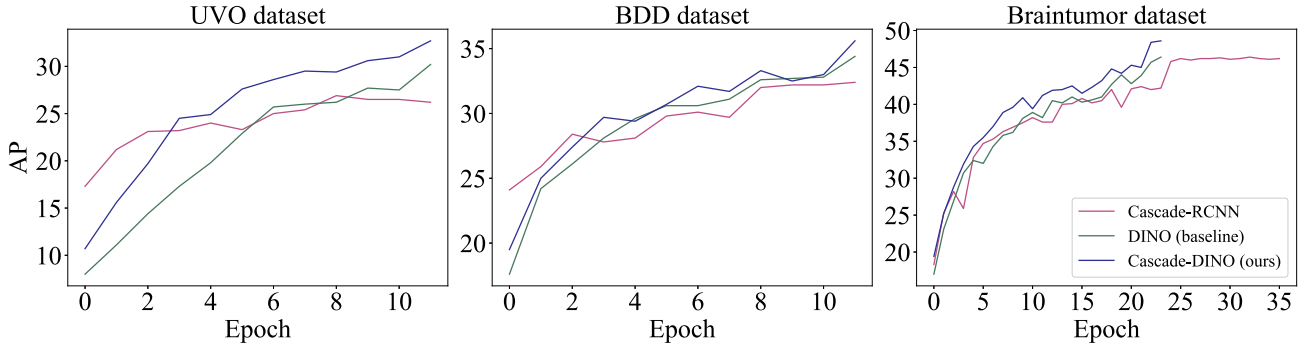

Figure 1. Quantitative results comparison between Cascade R-CNN [1], DINO [8] (Baseline) and Cascade-DINO (Ours) per training epoch on UVO [6], BDD [7], Brain tumor [2]. These datasets cover three various detection application domains. Cascade-DINO achieves stable performance growth during training, and outperforms Cascade R-CNN and DINO consistently on all three domains. Note that DINO is a very recent work which has already been significantly sped up by the usage of denoising branch and two-stage training.

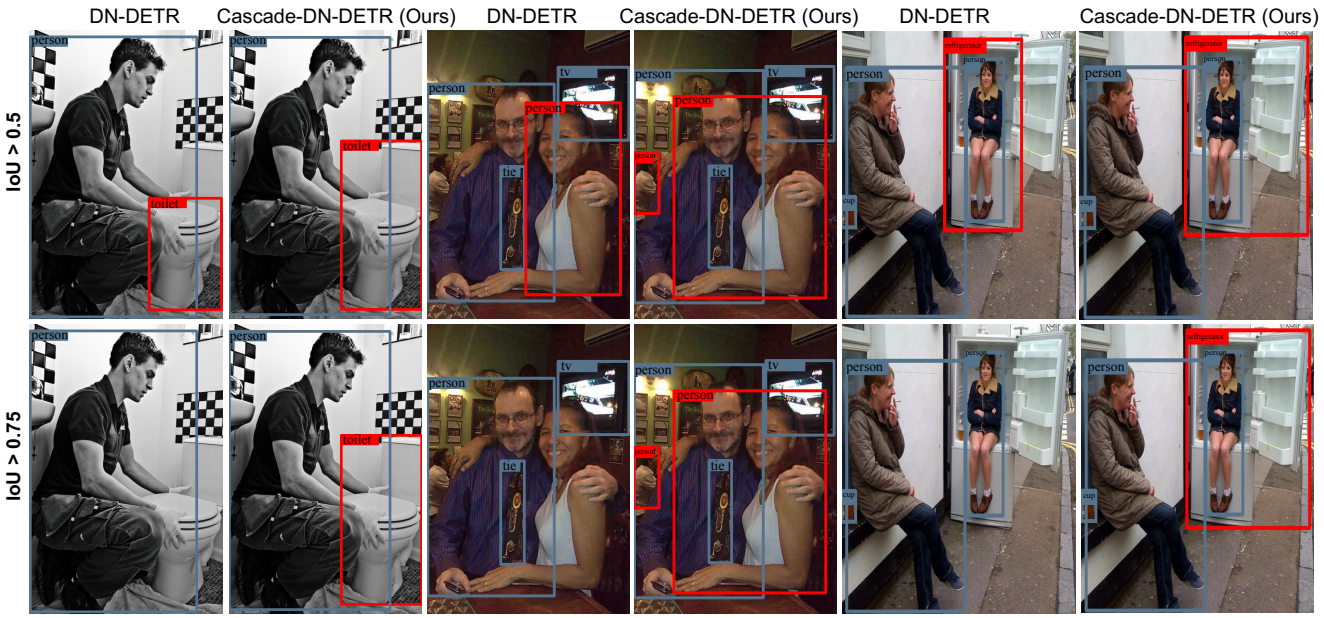

Figure 2. Qualitative results comparison between DN-DETR [3] and our Cascade-DN-DETR on COCO validation set. The box prediction differences are highlighted in red color. The first row shows box predictions satisfying IoU threshold (to GT box) larger than 0.5 while the bottom row shows IoU threshold larger than strict IoU (to GT) threshold 0.75. Taking the first column as an example, the detected 'toilet' object by baseline DN-DETR is filtered when using strict threshold 0.75.

Table 1. Cascade attention (CA) vs. Deformable attention (DA) on UVO. QR denotes Query Re-calibration.

Model	DA	CA	QR	AP	AP_{50}	AP_{75}	AR
DINO [46]	✓			30.2	46.9	30.5	63.4
Cascade-DINO	✓		✓	31.5	47.9	32.1	63.4
		✓		31.9	49.5	32.8	63.4
		✓	✓	**32.7**	50.2	33.4	63.0

Table 2. Ablation study on the box-constrained Deformable attention on UVO. **Baseline**: DINO with standard deformable attention in the transformer decoder. **Box-constrained deformable attention**: the learnable offsets around reference points are constrained inside the predicted boxes by normalization. Our Cascade-DINO adopts the cascade-attention.

Model	AP	AP_{50}	AP_{75}	AR
DINO (Baseline, standard deformable attention)	30.2	46.9	30.5	63.4
Box-constrained Deformable attention	$31.0_{\uparrow \mathbf{0.8}}$	47.2	31.7	62.8
Casacde-DINO (Ours)	$32.7_{\uparrow \mathbf{2.5}}$	50.2	33.4	63.0

1.3. More Implementation Details

Implementation Details In the paper, we mainly adopt DN-DETR [3] as our baseline. We use 4-scale feature maps with the help of a deformable encoder. For the transformer decoder, we apply cascade attention to each feature map and perform fusion in each layer. For query recalibration, we add one IoU head which is similar to the classification head but the output of the IoU head has only one channel for each query. We employ 300 queries with one pattern to keep the same as 300 queries of traditional DETR methods. We perform the same Hungarian matching as traditional DETR-based methods and record the IoU of matched boxes and ground truth. We use this as the regression target for our IoU head. L2Loss is used for IoU regression.

Figure 3. Qualitative results comparison between DN-DETR [3] and our Cascade-DN-DETR. The box prediction differences are highlighted in red. Compared to DN-DETR (Baseline), our Cascade-DN-DETR can better detect objects for challenging occlusion cases.

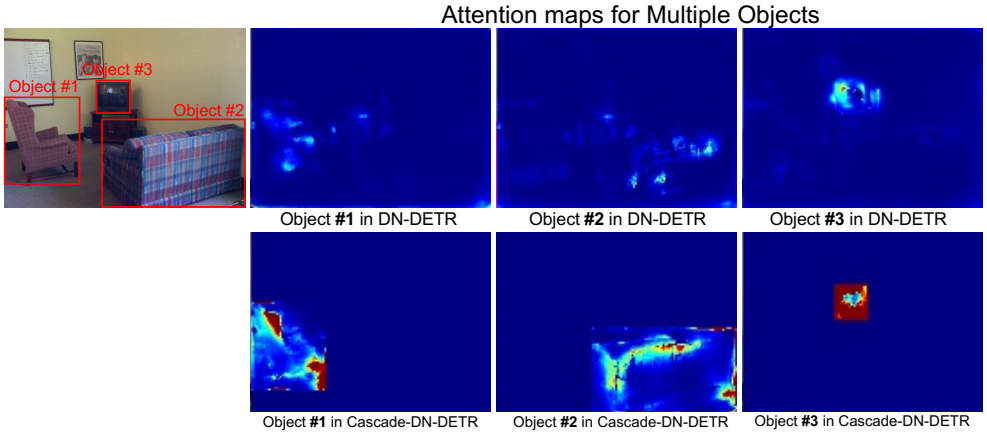

Figure 4. Visual comparison of cross-attention maps between DN-DETR [3] and our Cascade-DN-DETR on COCO for multiple objects.

We adopt auxiliary losses and the same loss coefficients as other DETR methods. The loss coefficients for classification loss, box L1 loss, box giou loss and our recalibration loss are $\{1.0, 5.0, 2.0, 2.0\}$. For DAB-DETR [4], we also use 4-scale feature maps and implement it by removing the dn-part of the previous DN-DETR. For Faster-RCNN [5] and Cascade R-CNN [1], we use the standard implementation of mmdetection.

For all dataset components in the UDB10 benchmark, we use the default data augmentation of Faster-RCNN, Cascade-RCNN and DETR-based methods on COCO. Large datasets (more than 10k images) in the UDB10 benchmark are COCO, UVO, BDD100K, and EgoHands. For them, we train DN-DETR for 12 epochs with an initial learning rate of 1×10^{-5} for the backbone and 1×10^{-4} for the transformer and drop the learning rate at the 10th epoch. We use the AdamW optimizer with weight decay 1×10^{-4}. We train on 8 Nvidia GeForce RTX 3090 GPUs with a total batch size of 8. We train Faster-RCNN for 12 epochs and drop the learning rate at the 8th and 11th epochs. For other small datasets, we train DN-DETR for 50 epochs and drop the learning rate at the 40th epoch. We train Faster-RCNN for 36 epochs and drop the learning rate at the 24th and 33rd epochs.

We also perform experiments on DINO [8] using Resnet50 backbone. We use 900 queries for the DINO baseline and Cascade-DINO. We employ 4-scale feature maps from the deformable encoder. We implement Cascade-DINO by replacing its deformable decoder with our cascade decoder, which is the same as Cascade-DN-DETR. Since DINO is a two-stage DETR detector, we also use the recalibrated score to select anchors from the transformer encoder. DINO is a strong SOTA method and we adopt the same 12-epoch and 24-epoch training schedules as their paper. For large datasets (more than 10k images) in the UDB10 benchmark, we train Cascade-RCNN, DINO, and Cascade-DINO for 12 epochs. For other small datasets, we train Cascade-RCNN for 36 epochs and DINO, Cascade-DINO for 24 epochs. For Box-constrained deformable attention, we also use IoU recalibration for comparison with Cascade-DINO.

Inference Details During inference, we predict expected IoU scores (Eq.4 of the paper) for all queries and take them as the recalibrated scores as discussed in the paper. We perform cascade attention with the initial boxes and predicted boxes of each layer. We also add cascade attention and IoU recalibration to dn-part in training and do not use dn-part in inference, which is the same as the original DN-DETR. For the other inference settings, we kept the same with our baseline methods DN-DETR and DINO.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[2] Yousef Ghanem. Brain tumor detection dataset. *Roboflow Universe*, 2022.

[3] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, 2022.

[4] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *ICLR*, 2022.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[6] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021.

[7] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.

[8] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Harry Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023.