Cascade-DETR: Delving into High-Quality Universal Object Detection

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

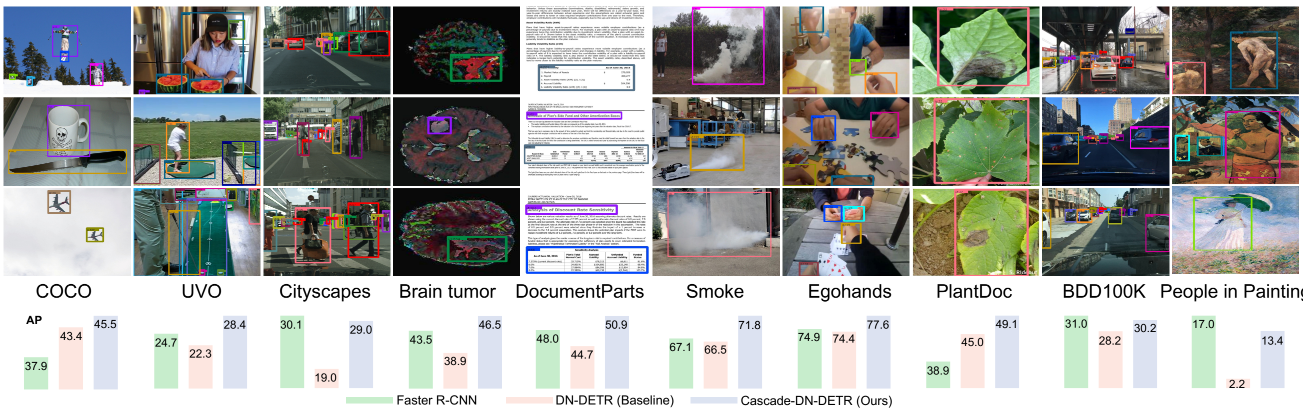


Figure 1. Cascade-DETR for high-quality universal object detection. We compare Faster R-CNN [35], DN-DETR [24] and our Cascade-DN-DETR on the constructed UDB10 benchmark. Cascade-DN-DETR gives a strong performance on a variety of benchmarks, spanning traffic, medical, art, open-world, etc. Taking the previous SOTA method DN-DETR [24] as baseline, our approach achieves 5.7 UniAP performance gain on the UDB10.

Abstract

Object localization in general environments is a fundamental part of vision systems. While dominating on the COCO benchmark, recent Transformer-based detection methods are not competitive in diverse domains. Moreover, these methods still struggle to very accurately estimate the object bounding boxes in complex environments.

We introduce Cascade-DETR for high-quality universal object detection. We jointly tackle the generalization to diverse domains and localization accuracy by proposing the Cascade Attention layer, which explicitly integrates object-centric information into the detection decoder by limiting the attention to the previous box prediction. To further enhance accuracy, we also revisit the scoring of queries. Instead of relying on classification scores, we predict the expected IoU of the query, leading to substantially more well-calibrated confidences. Lastly, we introduce a universal object detection benchmark, UDB10, that contains 10 datasets from diverse domains. While also advancing the state-of-the-art on COCO, Cascade-DETR substantially improves DETR-based detectors on all datasets in UDB10, even by over 10 mAP in some cases. The improvements under stringent quality requirements are even more pronounced. Our code and pretrained models are at https://github.com/SysCV/cascade-detr.

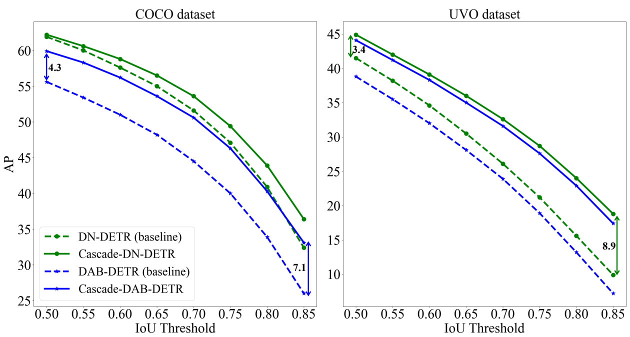
*Equal contribution.

1. Introduction

Object detection is a fundamental computer vision task with a wide range of real-life applications, such as self-driving and medical imaging. With remarkable progress since the emergence of DETR [5], Transformer-based detectors [55, 13, 38] have achieved ever increasing performance. The recent DETR-based methods [24, 52, 29] outperform CNN-based detectors [34, 18, 35, 41] on the *de facto* COCO challenge by a substantial margin.

Despite the notable progress of DETR-based detectors, there are still significant limitations that need to be addressed. Figure 1 shows that DETR-based methods severely struggle when applied outside of the conventional COCO benchmarks. This can be attributed to the limited number of training samples and diverse styles encountered in more task-specific domains, resulting in a drop in performance even below their CNN-based predecessors. In particular, we find that on e.g., Cityscapes [12] and Brain tumor [17] benchmarks, the performance of DN-DETR [24] is substantially poorer than Faster R-CNN despite its superior performance on COCO. Moreover, the prediction of highly accurate bounding boxes remains challenging. In Figure 2, given

Figure 2. Detection results comparison between DN-DETR [24] and our Cascade-DN-DETR, DAB-DETR [29] and our Cascade-DAB-DETR on COCO [27] (Left) and UVO [43] (Right), using IoU thresholds ranging from loose to strict. All comparisons are with the same training setting and schedule.



stricter IoU thresholds, existing DETR-based methods still have substantial room for improvement.

We partially attribute these two problems, namely, poor generalization to other datasets and limited bounding box accuracy, to a lack of a local object-centric prior. Following the general philosophy of transformers [15], DETR-based methods replace convolutions with global cross-attention layers in the detection head, thus removing the object-centric inductive bias. We argue that without such bias makes it difficult to accurately identify local object regions, thus limiting the bounding box accuracy. Additionally, the reliance on a purely data-driven approach to learn such bias places a heavy reliance on large annotated datasets, which are often unavailable in diverse real-world applications. Many detection tasks have distinct image domains, such as medical imaging or document analysis (as shown in Figure 1), which differ significantly from those in COCO or ImageNet, making pretraining on large annotated datasets even less effective.

The other attributing factor is the scoring of bounding box predictions which further exacerbates the high accuracy of DETR-based detectors. The query scoring in DETR decoder is purely based on the final classification confidence. However, these scores are largely oblivious of the quality of the predicted bounding box. Instead, we argue that correctly classified box proposals that better overlaps with the ground-truth should be assigned higher scores.

To address these two issues, this paper presents Cascade-DETR to promote high-quality universal detection performance for DETR-based models. To tackle the lack of local object-centric prior, we introduce cascade attention in the DETR decoder, which constrains the spatial cross attention layers to only inside the previously predicted bounding box of each query. Since DETR decoder has multiple decoder layers, the cascade structure iteratively refines the cross-attention region for each query, using a more accurate box prediction after every layer. To improve the scoring of box predictions, we propose an IoU-aware Query Recali-

Table 1. Datasets components in UDB10 benchmark for evaluating high-quality universal object detection. The UniAP metric computes the mean of AP for each individual dataset component. Training is done individually on each dataset. All comparing methods use ResNet50 as backbone. Our Cascade-DN-DETR is built on DN-DETR [24]. FR-CNN: Faster R-CNN; Paintings: People in paintings dataset [33]; Document: Document parts [31].

	Domain	# Images	FR-CNN [35]	DN-DETR [24]	Ours
COCO [27]	Natural	118k	37.9	43.4	$45.5_{\uparrow 2.1}$
UVO [43]	Open World	15k	24.7	22.3	$28.4_{\uparrow 6.1}$
Cityscapes [12]	Traffic	3k	30.1	19.0	$29.0_{\uparrow 10.0}$
BDD100K [48]	Traffic	70k	31.0	28.2	$30.2_{\uparrow 2.0}$
Brain tumor [17]	Medical	7k	43.5	38.9	$46.5_{\uparrow 7.6}$
Document [31]	Office	1k	48.0	44.7	$50.9_{\uparrow 6.2}$
Smoke [32]	Natural	0.5k	67.1	66.5	$71.8_{\uparrow 5.3}$
EgoHands [1]	Egoview	11k	74.9	74.4	$77.6_{\uparrow 3.2}$
PlantDoc [37]	Natural	2k	38.9	45.0	$49.1_{\uparrow 4.1}$
Paintings [33]	Art	0.6k	17.0	2.2	$13.4_{\uparrow 11.2}$
UniAP			41.3	38.5	$\mathbf{44.2}_{\uparrow 5.7}$

bration, by adding an IoU prediction branch to re-calibrate query scores. In parallel to the query classification and regression branches, the IoU prediction branch computes the box proposal IoU to the corresponding GT object. This enables each matched learnable query to be aware of its quality more accurately. During inference, we recalibrate the classification scores by the predicted localization scores as the final ones to rank proposals.

We further compose a new detection benchmark UDB10 and corresponding evaluation metric UniAP to support high-quality universal detection. We hope to facilitate the detection community not only focusing detection results on COCO but also in more wide real-life applications. As in Table 1, UDB10 consists of 10 datasets from various real-life domains. We compare the UniAP among Faster R-CNN [35], DN-DETR [24] and Cascade-DN-DETR, where our approach achieves the best 44.2 UniAP. With negligible model parameters increase, our method significantly promotes the detection quality of DETR-based models for 5.7 UniAP, especially on the domain-specific datasets. This is also validated by our large performance gain in Figure 2. On the large-scale COCO benchmark, Cascade-DN-DETR achieves significant 2.1 and 2.4 AP improvement over DN-DETR using R50 and R101 backbone respectively.

2. Related Work

DETR-based Object Detection Modern object detectors can be mainly divided into the classical CNN-based and more recent DETR-based models [9, 50, 26, 4, 42]. The convolutional detectors includes one-stage detectors [34, 41] and two/multi-stage models [35, 18, 2, 6]. For DETR-based models [5, 55, 30, 14, 36], recent works such as [29, 24, 52] outperform CNN-based detectors by a significant margin on COCO.

For improving the transformer decoder, Dynamic DETR [13] designs dynamic encoder for focusing on more important features on multi-scale feature maps while [39] even replaces the decoder with FCOS/RCNN networks. To enhance decoder queries, Efficient DETR [47] adopts the top-K locations from encoder's dense prediction prior. Anchor DETR [46] represents object queries based on anchor points, while DAB-DETR [29] adopts 4D anchor box coordinates. DN-DETR [24] further speeds up the DETR convergence by an additional denoising branch. Based on DN-DETR and DAB-DETR, DINO [52] includes contrastive denoising training and mixed query selection for anchor initialization.

In contrast to existing DETR-based methods [16, 39, 28], Cascade-DETR is targeted for high-quality object detection. The proposed cascade attention and IoU-aware query recalibration significantly improve AP performance under strict IoU thresholds. Besides only experimenting on COCO, we show the effectiveness of our approach on the constructed UDB10 benchmark, which contains a wide range of task-specific applications.

Cross-attention in DETR-based Decoder In addition to standard cross-attention [5] applied on global image features, Deformable DETR [55] proposes deformable attention. A set of 2D image locations are predicted, which are then used for attention. Mask2Former [10] proposes mask attention, only indented for segmentation. Different from these methods, our cascade attention utilizes the iteratively updated boxes to constrain the cross-attention on the image, and does not introduce any extra model parameters. We reveal our advantages to deformable attention and mask attention in the experiment section.

High-quality Object Detection Different from high-quality segmentation networks [23, 10, 21, 22] based on transformers, existing works [40, 2, 3, 6] on high-quality object detection are mainly R-CNN based. Specially, Cascade R-CNN [2] introduces multi-stage detectors trained with increasing IoU thresholds, while Dynamic R-CNN [51] designs dynamic labels and a regression loss. Wang et al. [19] improves R-CNN based segmentation via mask scoring. For localization quality estimation (LQE), previous works [41, 20, 54, 8, 25] mainly study it in FCOS or R-CNN based detectors. To our knowledge, we are the first DETR-based method to tackle the problem of predicting highly accurate boxes.

DETR-based Universal Object Detection Existing DETR-based methods [24, 52] mostly train and evaluate their performance on COCO. However, detectors should generalize well to wide and practical scenarios, such as medical imaging and document analysis. Typically these datasets contain around 1K to 20K images, where some contain images with very different styles than COCO/ImageNet. Different from previous detection work on adaptation learning [45, 7], few-shot setting [49] or mixed training [53], we focus on the fully supervised training setting per dataset to evaluate the detector performance in various application scenarios. To facilitate the research on universal object detection using DETR-based detectors, we construct a large-scale UDB10 benchmark containing 228k images, which doubles the size of UODB [45] for domain adaptation and has significantly more images per dataset component than [11]. We show that Cascade-DETR with injected local object-centric prior brings large performance gains to existing DETR-based models across wide and challenging domains, making DETR-based models more universally applicable.

3. Cascade-DETR

We propose Cascade-DETR for high-quality and universal object detection. We first review the design of the conventional DETR decoder in Section 3.1. Then we introduce our detection transformer Cascade-DETR in Figure 3. It is an iterative approach consisting of two novel components: **1)** Cascade attention, which constrains the cross-attention range in each decoder layer within the box region predicted from the preceding layer (Section 3.3); **2)** Query-recalibration, which recalibrates the learnable queries with the IoU prediction to enable more accurate query scoring (Section 3.4). Finally, we describe the training and inference details of our Cascade-DETR in Section 3.5.

3.1. Preliminaries: The DETR Decoder

We briefly review the design of the standard DETR decoder, which consists of a set of cross- and self-attention layers that iteratively updates a set of queries, initialized as learnable constants. At the i-th layer, the queries $\mathbf{Q}_i \in \mathbb{R}^{N\times D}$ are first input to a self-attention block, followed by cross-attention with the encoded image features of size $H \times W \times D$. The cross-attention is computed as the weighted sum over the global feature map,

$$\mathbf{Q}_{i+1} = \sum_{j=1}^{H\times W} \frac{\exp(f_q(\mathbf{Q}_i)\cdot \mathbf{K}_i^j)\mathbf{V}_i^j}{\sum_k \exp(f_q(\mathbf{Q}_i)\cdot \mathbf{K}_i^k)} + \mathbf{Q}_i\,, \qquad (1)$$

where $\mathbf{K}$ and $\mathbf{V}$ respectively denote key and value maps extracted from the image features. The index i denotes the cross-attention layer, j is the 2D spatial location on the image, and f_q denotes the query transformation function.

The updated queries $\mathbf{Q}_{i+1}$ are then used to predict bounding boxes $\mathbf{B}_{(i+1)}$ and query scores $\mathbf{S}_{(i+1)}$ by feeding them into two parallel linear layers f_{box} and f_{score} respectively, *i.e.*, $\mathbf{B}_{(i+1)} = f_{\text{box}}(\mathbf{Q}_{(i+1)})$ and $\mathbf{S}_{(i+1)} = f_{\text{cls}}(\mathbf{Q}_{(i+1)})$. The query score matrix $\mathbf{S}_{(i+1)}$ of size $N \times (C+1)$ contains the class probabilities for all input queries, where C is the number of classes of the dataset. This decoder design is generally used in [5, 29, 30, 24].

3.2. Cascade-DETR Architecture

In this section, we describe the architecture of Cascade-DETR, which injects local object-centric bias into the conventional transformer decoder in Section 3.1. Similar to

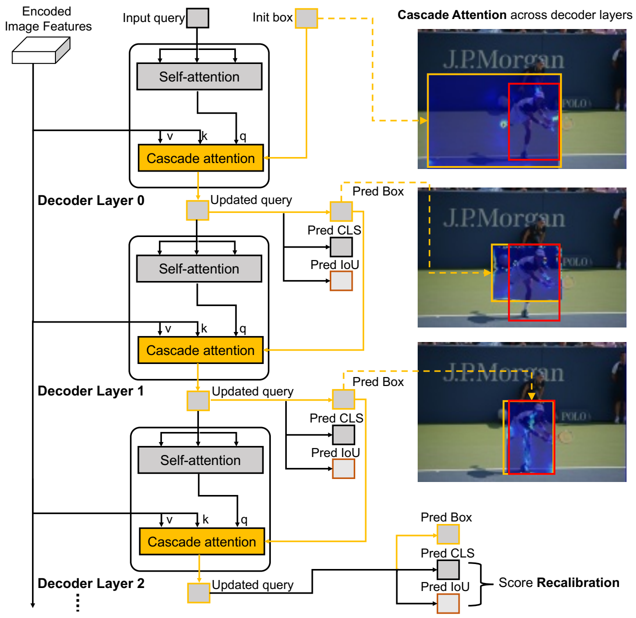


Figure 3. The transformer decoder of our Cascade DETR. We feed in the encoded image features from the transformer encoder along with learnable queries. The box-constrained cross-attention regions (inside the yellow predicted boxes) are iteratively refined per decoder layer, which, in turn, further promotes detection accuracy. The score recalibration is used in the last transformer decoder layer during inference. The red box denotes the ground truth object box. We omit the transformer encoder and positional embedding for clarity.

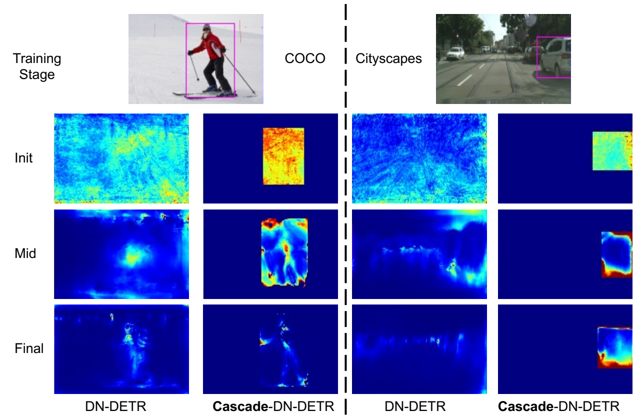


Figure 4. Visual comparison of cross-attention map between DN-DETR [24] and our Cascade-DN-DETR on COCO and Cityscapes datasets. At different network training stages, we visualize the cross-attention map of the last transformer decoder layer, where the learnable query corresponds to the object inside the red box.

existing DETR-based methods, such as DAB-DETR [29] and DN-DETR [24], our architecture contains a transformer encoder for extracting image features. The encoded features combined with the positional encoding are fed to the transformer decoder. The learnable queries are also fed into the decoder to localize and classify objects through cross-attention. The two new modules in our Cascade-DETR are cascade attention and IoU-aware query re-calibration, which only bring negligible computation overhead or model parameters while significantly improving the detection quality and generalizability.

3.3. Cascade Attention

In the standard DETR decoder, learnable queries attend globally over the entire image features, as in Eq. 1. However, to accurately classify and localize the object, we argue that local information around each object is most crucial. The global context can be extracted via self-attention between queries. In Figure 4, we observe that the cross-attention distribution during COCO training tends to converge to the surrounding regions of the predicted object locations. While the transformer model can learn this inductive bias end-to-end, it requires large amounts of data. This problem becomes more pronounced for small or task-specific datasets with image styles radically different from those exhibited in ImageNet.

To address the above issue, we treat the object-centric prior as a known constraint to incorporate into both the initialization and training procedures, as depicted in Figure 3. We design the cascade attention in layer $i+1$ as,

$$\mathbf{Q}_{i+1} = \sum_{j \in \mathbf{S}_i} \frac{\exp(f_q(\mathbf{Q}_i) \cdot \mathbf{K}_i^j)\mathbf{V}_i^j}{\sum_{k \in \mathbf{S}_i} \exp(f_q(\mathbf{Q}_i) \cdot \mathbf{K}_i^k)} + \mathbf{Q}_i\,, \quad (2)$$

$$\mathbf{S}_i = \mathcal{M}(\mathbf{B}_i) = \mathcal{M}(f_{\text{box}}(\mathbf{Q}_i))\,, \quad (3)$$

where $\mathbf{S}_i$ is the set of 2D locations inside the predicted bounding box $\mathbf{B}_i$ from the preceding decoder layer i. The cascade structure utilizes the property that the predicted $\mathbf{B}_i$ will be more accurate after every decoder layer in DETR-based detectors [5]. Thus, the box-constrained cross-attention region $\mathbf{S}_i$ not only brings object-centric bias, but will also be iteratively refined (see Figure 3). With more accurately cross-attended features per layer, cascade attention in turn promotes the detection accuracy per layer.

We validate our assumption by visualizing the attention map in Figure 4. The initial and final attention maps of a baseline DN-DETR model are shown both in COCO and Cityscapes. On COCO, we observe both the cross-attention of a randomly initialized query eventually converges on semantically distinct locations using DN-DETR or Cascade-DN-DETR. However, on Cityscapes, there is an obvious contrast between the two methods, where the integration of object-centric knowledge is more important to focus the attention on the most relevant parts of the image.

Unlike previous approaches such as DAB-DETR [29] and Deformable DETR [55], which utilize soft constraints, the design of our Cascade-DETR is much simpler. The prediction boxes in each layer of the DETR decoder is directly used as constraints to limit the cross-attention range in the following layer. This inductive bias enables DETR to converge quickly and achieve superior performance, especially for small and diverse datasets.

3.4. IoU-aware Query Recalibration

Most DETR-based detectors take 300 [24, 29] or even 900 [52] learnable queries as input to the transformer decoder and predict one box per query. When computing final detection results, classification confidence is adopted as a surrogate to rank all query proposals. However, the classification score does not explicitly account for the accuracy of the predicted bounding box, which is crucial for selecting high-quality proposals. We therefore introduce an IoU-aware scoring of the predicted queries in order to achieve more well-calibrated confidence, which better reflects the quality of the predictions.

Instead of scoring queries by classification confidence, we score them by the expected IoU with the ground-truth box. Let $E(\text{IoU}_q)$ be the expected ground-truth IoU of query q. Further, let $P(\text{obj}_q)$ denotes the probability of q indicating an object, as obtained from the classification probability. The expected IoU of a query is computed as

$$\begin{aligned} &E(\text{IoU}_q) \\ &= E(\text{IoU}_q \mid \text{obj}_q)P(\text{obj}_q) + E(\text{IoU}_q \mid \neg\text{obj}_q)P(\neg\text{obj}_q) \\ &= E(\text{IoU}_q \mid \text{obj}_q)P(\text{obj}_q) \end{aligned} \tag{4}$$

Here, $\neg$ denotes the negation of the binary random variable. The second equality follows from that the expected IoU for a prediction that is not an object is zero: $E(\text{IoU}_q \mid \neg\text{obj}_q) = 0$.

To predict the expected IoU (4), we introduce an additional branch that predicts the expected IoU for a present ground-truth object $E(\text{IoU}_q \mid \text{obj}_q)$, as illustrated in Figure 3. Specifically, we simply use another linear layer in parallel to the classification and box regression branches. As derived in Eq.(4), the final query score is then obtained as the product between the predicted IoU and the original classification confidence $P(\text{obj}_q)$.

We supervise the IoU prediction with an L_2 loss to the ground-truth IoU, denoted IoU_q^{GT},

$$L_{\text{IoU}} = \left\| E(\text{IoU}_q \mid \text{obj}_q) - \text{IoU}_q^{\text{GT}} \right\|^2 . \tag{5}$$

The loss is only applied for queries q with an assigned ground-truth, as we condition on the presence of the object in the expectation. Note that the L_2 loss implies learning the mean, *i.e*. expectation, of a Gaussian distribution over the IoU values. We ablate this choice of loss in Table 5 of the experiment section.

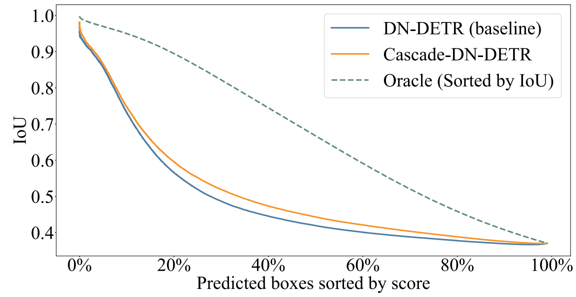


Figure 5. Sparsification plot between query localization quality (IoU to GT boxes) and query ranking (scoring). For 5k COCO validation images with 50 outputs for each image, we sort all the outputs by their confidence scores. We then compute the IoU with ground truth for each prediction and show a cumulative average of IoU. **Oracle:** Cumulative average of IoU sorted by IoU itself. Compared to the blue curve before recalibration, ours re-calibrated orange curve is closer to the Oracle and has a much higher localization quality.

To analyze the advantage of our IoU-aware query recalibration, we generate sparsification plots over all predictions on COCO in Figure 5. All predictions are sorted with respect to the confidence score. The average IoU with ground-truth is then plotted for the N predictions with the highest confidence score, by varying N across the x-axis. The Oracle represents the upper bound, obtained by taking the top N predictions in terms of ground-truth IoU. Compared to Cascade-DN-DETR without query recalibration (blue curve), our recalibrated result (orange curve) achieves a substantially better ranking of the results, leading to a higher IoU.

3.5. Training and Inference

Our Cascade-DETR is trained in an end-to-end manner using a multi-task loss function,

$$\mathcal{L}_{\text{Detect}} = \mathcal{L}_{\text{Box}} + \lambda_1 \mathcal{L}_{\text{Class}} + \lambda_2 \mathcal{L}_{\text{IoU}}, \tag{6}$$

where $\mathcal{L}_{\text{Detect}}$ supervises both the position prediction and the category classification borrowed from the DETR [5] detector. The hyper-parameters λ_1 and λ_2 balances the loss functions, and set to $\{1.0, 2.0\}$ respectively on the validation set. Following [24, 29], FFNs and the Hungarian loss are adopted after each decoder layer. FFNs share their model parameters in each prediction layer.

During inference, our cascade attention is consistently used as it only relies on the predicted boxes in each transformer decoder layer. For the query scoring calibration manner, as described in 4, we only apply it on the final transformer decoder layer.

4. Experiments

4.1. Experimental Setup

COCO We perform evaluation on the challenging MS COCO 2017 object detection benchmark [27]. Models are trained with 118k training images in *train2017* split and evaluated on the 5k validation images in *val2017*. We report the standard average precision (AP) result under different IoU thresholds.

UVO and Cityscapes To generalize on universal object detection, we also conduct experiments on two challenging datasets, UVO [43] and Cityscapes [12]. UVO is an exhaustively labeled open-world dataset with 15k training images and 7k validation images. Cityscapes is an urban street scene dataset which contains 3k training images and 500 validation images. We perform results comparison following the standard training and model evaluation setting on the two benchmarks.

UDB10 Benchmark There is a wide variety of detection applications in real-life scenarios. To facilitate the research on universal detection, we construct a large-scale UDB10 benchmark which is composed of 10 different datasets across wide domains. Besides the aforementioned COCO [27], Cityscapes [12] and UVO [43], the other 7 task-specific datasets includes BDD100K [48], Brain Tumor [17], Document Parts [31], Smoke [32], EgoHands [1], PlantDoc [37] and People in paintings [33]. UDB10 contains 228k images, which covers a great variety of domains such as medical, traffic, nature, office, art, ego-view, etc. We follow the official training/evaluation settings on each dataset component. Along with the UDB10 benchmark, we design UniAP metric to evaluate the detection performance among detectors. After detectors are trained individually on each dataset component, UniAP is computed as the mean over the AP scores across all datasets.

In Table 2, we compare UDB10 with two other existing universal detection benchmarks UODB [45] and Roboflow 100 [11], where we find UDB10 has significantly more images and annotated instances per dataset component. We establish UDB10 aims to evaluate the detection performance of data-sensitive DETR-based methods in diverse domains.

Implementation Details In our experiments, we use two different backbones: ResNet-50 and ResNet-101 pretrained on ImageNet-1k, and train our model with an initial learning rate 1×10^{-5} for backbone and 1×10^{-4} for transformer. We use the AdamW optimizer with weight decay 1×10^{-4}. We train on 8 Nvidia GeForce RTX 3090 GPUs with total batch size of 8, and adopt two training schedules. For small datasets (less than 10k images), we train DETR-based methods for 50 epochs with a learning rate decay after 40 epochs. For large datasets (greater than or equal to 10k images), we adopt DETR-based methods for 12 epochs with a learning rate decay after 10 epochs. The original DETR uses 100 queries, while in all other experiments we use 300 queries except DINO [52], where 900 queries are used to be consistent with their paper. For multi-scale features, we use DN-Deformable-DETR [55] with a deformable encoder. For the first layer cascade attention input box, we use the initial learnable anchor box proposed in DAB-DETR [29]. For Faster-RCNN, we use 1X schedule for large datasets and 3X schedule for small datasets. For mask attention ablation on COCO, we train an extra mask head with ground truth mask annotations and do not use query recalibration. More details are in the Supp. file.

Table 2. Comparison between the universal detection benchmarks. # Images / set denotes the average number of training images per dataset component, while # Instances / set is the average number of annotated boxes per dataset component.

Benchmarks	# Images	# Images / set	# Instances / set
UODB [45]	113k	10.2k	69k
Roboflow 100 [11]	224k	2.2k	25k
UDB10 (Ours)	228k	22.8k	239k

4.2. Ablation Study

We conduct detailed ablation studies for Cascade-DETR using ResNet-50 as backbone on the Cityscapes [12] and UVO [43] datasets. We analyze the impact of each proposed component of our Cascade-DETR.

Ablation on Cascade Attention (CA) In Table 3, we study the effect of Cascade Attention (CA). Built on the baseline DN-DETR, CA significantly promotes the performance for 3.7 AP on UVO and 9.9 AP on Cityscapes. In Table 4, we further compare our cascade attention in the transformer decoder to the mask attention [10]. We perform comparisons on both UVO and COCO as both these two datasets in UDB10 have corresponding GT mask labels per box. We design the mask attention by an additional mask prediction branch, which is supervised by the GT mask labels. This can be regarded as an oracle analysis as many object detection benchmarks have no annotated GT mask labels. Our cascade attention achieves similar results to mask attention by improving 0.6 AP on COCO but decreasing 0.6 AP on UVO. This indicates that accurate object mask shape is not necessary for object detection.

Ablation on Query Recalibration (QR) In Table 3, we also validate the effect of Query Recalibration (QR), which promotes 3.6 AP on UVO and 4.1 AP on Cityscapes. Specifically, on UVO, QR improves 4.9 AP_{75} which is much larger than gain of 3.0 AP_{50}. We further perform detailed ablation experiments on the query recalibration loss types (Table 5), recalibration methods (Table 6) and recalibration training strategies (Table 7). In Table 5, the performance boost is similar using L2 or L1 loss while outperforming Huber Loss with 0.6 AP.

As derived in Eq. 4, our expected IoU is computed as a product between the classification confidence and IoU prediction. Table 6 compares this fusion with other strategies

Table 3. Ablation study on the Cascade Attention (CA) and Query Recalibration (QR). We use ResNet-50 based DN-DETR [24] with deformable encoder as our baseline.

			UVO				Cityscapes		
Model	CA	QR	AP	AP_{50}	AP_{75}	AR	AP	AP_{50}	AP_{75}
DN-DETR [24]			22.3	41.5	21.2	51.4	19.0	39.3	15.7
	✓		$26.0_{\uparrow\mathbf{3.7}}$	43.0	25.6	57.8	$28.9_{\uparrow\mathbf{9.9}}$	52.1	27.0
		✓	$25.9_{\uparrow\mathbf{3.6}}$	44.5	26.1	53.9	$23.1_{\uparrow\mathbf{4.1}}$	44.1	20.7
Ours	✓	✓	$28.4_{\uparrow\mathbf{6.1}}$	44.9	28.7	58.2	$29.0_{\uparrow\mathbf{10.0}}$	49.5	28.4

Table 4. Detection performance comparison between cascade and mask cross-attention schemes in the transformer decoder on UVO and COCO. Both two cross-attention schemes are taking DN-Deformable-DETR as baseline. **Oracle:** We add an extra mask head with GT mask supervision, and use predicted outputs as attention mask in the transformer decoder.

Cross-attention Type	UVO				COCO		
	AP	AP_{50}	AP_{75}	AR	AP	AP_{50}	AP_{75}
Mask Attention (**Oracle**)	29.0	46.4	29.5	56.8	44.2	61.1	47.8
Cascade Attention (**Ours**)	$28.4_{\downarrow\mathbf{0.6}}$	44.9	28.7	58.2	$44.8_{\uparrow\mathbf{0.6}}$	62.7	48.4

Table 5. Ablation study on the query recalibration loss on the UVO dataset. **Baseline:** DN-Deformable-DETR.

Loss type	AP	AP_{50}	AP_{75}	AR
Baseline	22.3	41.5	21.2	51.4
Huber Loss	$25.3_{\uparrow\mathbf{3.0}}$	43.8	25.4	53.6
L1 Loss	$25.9_{\uparrow\mathbf{3.6}}$	44.6	26.3	53.9
L2 Loss	$25.9_{\uparrow\mathbf{3.6}}$	44.5	26.1	53.9

Table 6. Comparison of various score recalibration methods between classification (cls) score and predicted IoUs on UVO dataset during testing. **Baseline:** DN-Deformable-DETR trained without query recalibration.

Scoring Manner	AP	AP_{50}	AP_{75}	AR
Baseline	22.3	41.5	21.2	51.4
Single score (cls. & IoU superv.)	$24.8_{\uparrow\mathbf{2.5}}$	44.3	24.4	53.7
Sum Fusion (cls. prob + IoU)	$23.5_{\uparrow\mathbf{1.2}}$	38.5	24.4	51.4
Expected IoU (cls. prob × IoU)	$25.9_{\uparrow\mathbf{3.6}}$	44.5	26.1	53.9

Table 7. Ablation study on the training strategies for query recalibration on UVO. **Baseline:** Default DN-Deformable-DETR training manner. **All:** Input all queries for query recalibration loss computation. **Positive:** Only input Hungarian matched outputs for loss computation. We assign GT IoU scores to the unmatched queries by greedy matching to the GT boxes.

Training strategies	AP	AP_{50}	AP_{75}	AR
Baseline	22.3	41.5	21.2	51.4
All	22.5	38.7	22.6	49.4
Positive	25.9	44.5	26.1	53.9

for computing the final query score. Our principled approach achieves the best performance of 25.9 AP. It outperforms the baseline classification-only by 3.6 AP and the sum fusion by a large margin of 2.4 AP. We also compare with directly predicting a single confidence score, supervised both by the baseline classification loss and our IoU loss (second row). While achieving a significant gain of 2.5 AP over the baseline, it does not reach the performance of our derived expected IoU based fusion.

Since the expected IoU scores in Eq. 5 are conditioned on the presence of the object, we only add this loss on predictions which are matched with ground-truth boxes. We ablate this choice in Table 7 by adding the loss to all predictions. The latter results in a performance only marginally above the baseline without IoU-awareness. Again, this demonstrates the advantage of our principled IoU-based query scoring.

4.3. Comparison with State-of-the-art

We compare Cascade-DETR with the state-of-the-art object detection methods on COCO, UVO, Cityscapes and our constructed UDB10 benchmark. We integrate Cascade-DETR on three representative methods [24, 29, 52], and find that Cascade-DETR attains consistent large gains over the strong baselines.

COCO Table 8 compares Cascade-DETR with state-of-the-art object detection methods on COCO benchmark. By integrating with SOTA DETR-based detectors, Cascade-DETR achieves consistent improvement on different backbones with negligible increase in model parameters, demonstrating its effectiveness by outperforming DN-Def-DETR [24] by 2.1 AP and 2.4 AP respectively on R50 and R101 backbone. Cascade-DETR consistently attains larger increase in the strict AP_{75} than the loose AP_{50}, which reveals our advantages in predicted box quality. Using R50 as backbone, we also compare Cascade-DINO to DINO [52] by replacing its deformable attention [55] in the transformer decoder with our cascade attention. Cascade-DINO outperforms DINO by 1.0 AP_{75} with a much simpler attention design, removing the necessity for predicting 2D anchor points and sampling offsets.

UVO and Cityscapes Table 9 tabulates the results on UVO benchmark, and Table 10 tabulates the results on Cityscapes benchmark. Cascade-DETR achieves the best 28.4 AP on UVO, where our approach significantly surpasses the strong baselines DN-DETR [24] and DAB-DETR [29], respectively with a large margin of 8.7 and 7.5 points in AP_{75}. The significant increase in AP_{75} is also consistent on Cityscapes. Comparing to our baseline DN-DETR, in Table 10, Cascade-DN-DETR substantially improves the AP_{75} from 15.7 to 28.4.

UDB10 Benchmark Table 11 shows the detailed results comparison between Faster R-CNN [35], DN-DETR [24] and our Cascade-DN-DETR on the constructed UDB10 benchmark. We compute UniAP as the mean of AP scores for each individual dataset component, where Cascade-DN-DETR obtains the highest 44.2 AP by improving the baseline performance for 5.7 AP and outperforms Faster R-CNN by 2.9 AP under the same R-50 backbone. The significant advancements reveal the generalizability of our approaches, without requiring any domain adaptation designs.

For the six task-specific and small-scale datasets in

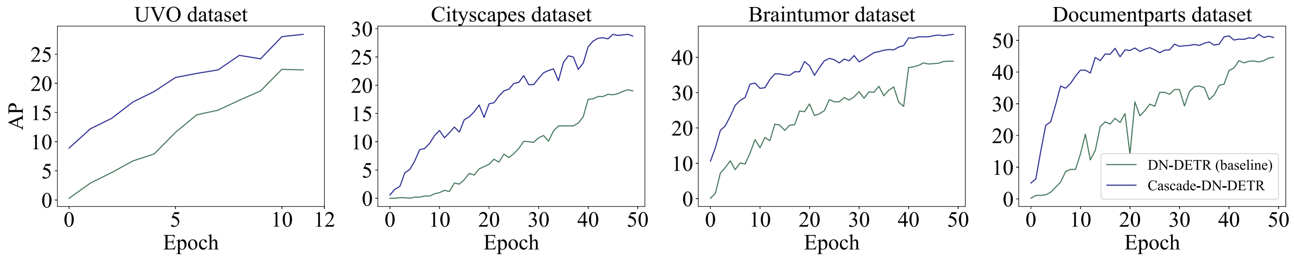


Figure 6. Detection results comparison between DN-DETR [24] (Baseline) and Cascade-DN-DETR (Ours) per training epoch on UVO [43], Cityscapes [12], Brain tumor [17], and Documentparts [31] datasets. These datasets cover four various detection application domains. Cascade-DN-DETR achieves stable performance growth during training, and consistently outperforms the strong baseline DN-DETR [24] with a significant margin. Note that DN-DETR has already been significantly sped up during training by its denoising branch.

Table 8. Comparison with SOTA methods on COCO *val2017*. All comparing methods are trained for 12 epochs. Asterisked models (*) were trained by ourselves. **Def**: deformable. We implement DAB-Deformable-DETR by removing the dn part in DN-Deformable-DETR.

Model	Base	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params
Faster-RCNN [35]	R50	37.9	58.8	41.1	22.4	41.1	49.1	40M
Cascade-RCNN [2]	R50	40.4	58.9	44.1	22.8	43.7	54.0	69M
DETR [5]	R50	15.5	29.4	14.5	4.3	15.1	26.7	41M
Def DETR [55]	R50	37.2	55.5	40.5	21.1	40.7	50.5	40M
CondDETR [30]	R50	32.0	51.8	33.5	14.1	34.7	47.9	43M
DAB-DETR(DC5) [29]	R50	38.0	60.3	39.8	19.2	40.9	55.4	44M
DE-CondDETR [44]	R50	35.6	55.2	37.8	20.6	38.5	48.3	44M
DN-Def-DETR [24]	R50	43.4	61.9	47.2	24.8	46.8	59.4	48M
Cascade-DN-Def-DETR	R50	45.5$_{\uparrow 2.1}$	62.2$_{\uparrow 0.3}$	49.4$_{\uparrow 2.2}$	27.3	50.0	62.4	48M
DINO* [52]	R50	48.8	66.2	53.1	31.1	52.0	63.0	47M
Cascade-DINO	R50	49.7$_{\uparrow 0.9}$	67.1$_{\uparrow 0.9}$	54.1$_{\uparrow 1.0}$	32.4	53.5	65.1	48M
DAB-Def-DETR* [29]	R101	37.1	55.6	40.0	19.3	41.2	51.6	67M
Cascade-DAB-Def-DETR	R101	42.7$_{\uparrow 5.6}$	60.0$_{\uparrow 4.4}$	46.3$_{\uparrow 6.3}$	24.5	47.7	58.4	67M
DN-Def-DETR [24]	R101	44.1	62.8	47.9	26.0	47.8	61.3	67M
Cascade-DN-Def-DETR	R101	46.5$_{\uparrow 2.4}$	63.7$_{\uparrow 0.9}$	50.4$_{\uparrow 2.5}$	27.5	50.6	63.8	67M

UDB10, we further compare model finetuning results by taking their corresponding COCO pretrained model as initialization. We find that the result of Faster R-CNN with COCO pretraining only has a slight increase in most dataset components. However, the COCO finetuning is much more crucial for DETR-based approaches. For example, with COCO initialization, the AP_{75} of DN-Def-DETR on Paintings [33] improves drastically from 1.2 to 19.9, while Cascade-DN-Def-DETR boosts from 9.0 to 21.5. However, Cascade-DN-Def-DETR still consistently outperforms the strong baseline DN-Def-DETR on all dataset components.

Convergence Speed Comparison In Figure 6, we provide the convergence speed comparison on four task-specific benchmarks UVO [43], Cityscapes [12], Brain tumor [17] and Documentparts [31]. Note that DN-DETR has already been significantly sped up by its denoising branch during training. Our Cascade-DN-DETR outperforms the strong baseline DN-DETR across all datasets by a significant margin at various training stages, and converges much faster.

4.4. More Results Comparison on UDB10

In Table 13, we provide comprehensive and detailed experiment results comparison on all 10 dataset compo-

Table 9. State-of-the-art results comparison on UVO [43]. All comparing methods are trained for 12 epochs. Both Cascade-DN-Def-DETR and Cascade-DAB-Def-DETR significantly surpass their strong baselines for over 7.0 AP_{75}.

Model	Base	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR
Faster-RCNN [35]	R50	24.7	48.4	22.1	11.1	21.0	32.9	43.6
Def DETR [55]	R50	17.7	34.9	16.3	6.7	15.8	25.3	51.4
DE-CondDETR [44]	R50	17.4	32.1	16.4	7.5	14.0	25.6	52.5
DAB-Def-DETR [29]	R101	20.0	38.8	18.9	7.4	17.0	28.2	50.4
Cascade-DAB-Def-DETR	R101	27.3$_{\uparrow 7.3}$	44.1$_{\uparrow 5.3}$	27.6$_{\uparrow 8.7}$	11.0	23.1	37.7	58.0
DN-Def-ETR [24]	R50	22.3	41.5	21.2	7.1	16.9	33.1	51.4
Cascade-DN-Def-DETR	R50	28.4$_{\uparrow 6.1}$	44.9$_{\uparrow 3.4}$	28.7$_{\uparrow 7.5}$	10.7	22.5	40.7	58.2

Table 10. State-of-the-art results comparison on Cityscapes [12]. Both Cascade-DN-Def-DETR and Cascade-DAB-Def-DETR achieve over 10.0 AP_{75} performance gain over their counterparts.

Model	Base	Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN [35]	R50	36	30.1	53.2	30.3	8.5	31.2	51.0
DETR [5]	R50	300	11.5	26.7	8.6	2.5	9.5	25.1
Def DETR [55]	R50	50	27.3	49.2	26.3	8.7	28.2	45.7
CondDETR [30]	R50	50	12.1	28.0	9.1	2.2	9.8	27.0
DE-CondDETR [44]	R50	50	26.8	47.8	25.4	6.8	25.6	46.6
DAB-Def-DETR [29]	R101	50	17.3	34.5	15.0	4.1	17.9	32.3
Cascade-DAB-Def-DETR	R101	50	25.4$_{\uparrow 8.1}$	43.9$_{\uparrow 9.4}$	25.0$_{\uparrow 10.0}$	6.7	25.8	46.4
DN-Def-DETR [24]	R50	50	19.0	39.3	15.7	4.9	19.8	35.5
Cascade-DN-Def-DETR	R50	50	29.0$_{\uparrow 10.0}$	49.5$_{\uparrow 10.2}$	28.4$_{\uparrow 12.7}$	9.1	28.4	51.5

Table 11. Detailed results comparison on the proposed UDB10 benchmark using R50 backbone. All methods are initialized from ImageNet pretrained model. We take DN-Def-FETR as the strong baseline to build our Cascade-DN-Def-FETR.

	Faster RCNN [35]			DN-Def-DETR [24]			**Cascade-DN-Def-DETR**		
Dataset	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
COCO [27]	37.9	58.8	41.1	43.4	61.9	47.2	45.5$_{\uparrow 2.1}$	62.2$_{\uparrow 0.3}$	49.4$_{\uparrow 2.2}$
UVO [43]	24.7	48.4	22.1	22.3	41.5	21.2	28.4$_{\uparrow 6.1}$	44.9$_{\uparrow 3.4}$	28.7$_{\uparrow 7.5}$
Cityscapes [12]	30.1	53.2	30.3	19.0	39.3	15.7	29.0$_{\uparrow 10.0}$	49.5$_{\uparrow 10.2}$	28.4$_{\uparrow 12.7}$
BDD100K [48]	31.0	55.9	29.4	28.2	53.9	24.8	30.2$_{\uparrow 2.0}$	55.0$_{\uparrow 1.1}$	27.9$_{\uparrow 3.1}$
Brain tumor [17]	43.5	75.1	45.0	38.9	71.6	38.6	46.5$_{\uparrow 7.6}$	75.6$_{\uparrow 4.0}$	49.4$_{\uparrow 10.8}$
Document [31]	48.0	66.2	55.6	44.7	64.1	50.3	50.9$_{\uparrow 16.2}$	66.6$_{\uparrow 2.5}$	58.4$_{\uparrow 8.1}$
Smoke [32]	67.1	92.9	80.6	66.5	91.6	77.8	71.8$_{\uparrow 5.3}$	91.9$_{\uparrow 0.3}$	82.9$_{\uparrow 5.1}$
EgoHands [32]	74.9	96.9	90.4	74.4	97.4	89.2	77.6$_{\uparrow 3.3}$	98.3$_{\uparrow 0.9}$	91.5$_{\uparrow 2.3}$
PlantDoc [37]	38.9	60.8	44.9	45.0	61.7	53.7	49.1$_{\uparrow 4.1}$	63.9$_{\uparrow 2.2}$	56.5$_{\uparrow 2.8}$
Paintings [33]	17.0	50.1	6.3	2.2	5.8	1.2	13.4$_{\uparrow 11.2}$	33.1$_{\uparrow 27.3}$	9.0$_{\uparrow 7.8}$
UniAP	41.3			38.5			**44.2**$_{\uparrow 5.7}$		

Table 12. Finetuning results for the six small-scale task-specific datasets in our UDB10 benchmark using R50 backbone. All finetuned methods are from their COCO pretrained model.

	Faster RCNN [35]			DN-Def-DETR [24]			**Cascade-DN-Def-DETR**		
Dataset	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Brain tumor [17]	44.2	74.1	46.3	48.0	78.1	51.7	51.5$_{\uparrow 3.5}$	79.6	56.0
Document [31]	49.7	67.7	58.0	51.6	67.7	60.8	52.7$_{\uparrow 1.1}$	68.0	61.1
Smoke [32]	68.3	90.3	82.4	72.9	94.9	86.0	74.3$_{\uparrow 1.4}$	93.7	86.8
EgoHands [1]	75.8	96.9	90.0	77.4	98.7	92.1	79.4$_{\uparrow 2.0}$	97.9	93.4
PlantDoc [37]	37.5	57.3	39.9	50.2	62.9	57.3	52.7$_{\uparrow 2.5}$	66.5	61.1
Paintings [33]	24.1	59.5	15.4	28.6	66.3	19.9	29.4$_{\uparrow 0.8}$	66.0	21.5

Table 13. Quantative Results Comparison on the constructed UDB10 benchmark using R50 backbone. All methods are initialized from ImageNet pretrained model. We take DN-Def-DETR [24] as the baseline to build our Cascade-DN-Def-DETR. We also take DINO [52] as a stronger baseline, replacing the deformable transformer decoder with our cascade transformer decoder and building our Cascade-DINO. The UniAP metric computes the mean of AP for each individual dataset component.

	Faster RCNN [35]			DN-Def-DETR [24]			**Cascade-DN-Def-DETR**			Cascade RCNN [2]			DINO [52]			**Cascade-DINO**		
Dataset	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
COCO [27]	37.9	58.8	41.1	43.4	61.9	47.2	$45.5_{\uparrow\mathbf{2.1}}$	62.2	49.4	40.4	58.9	44.1	48.8	66.2	53.1	$49.7_{\uparrow\mathbf{0.9}}$	67.1	54.1
UVO [43]	24.7	48.4	22.1	22.3	41.5	21.2	$28.4_{\uparrow\mathbf{6.1}}$	44.9	28.7	26.2	46.7	25.2	30.2	46.9	30.5	$32.7_{\uparrow\mathbf{2.5}}$	50.2	33.4
Cityscapes [12]	30.1	53.2	30.3	19.0	39.3	15.7	$29.0_{\uparrow\mathbf{10.0}}$	49.5	28.4	31.8	54.4	30.9	34.5	56.6	34.5	$34.8_{\uparrow\mathbf{0.3}}$	57.3	33.7
BDD100K [48]	31.0	55.9	29.4	28.2	53.9	24.8	$30.2_{\uparrow\mathbf{2.0}}$	55.0	27.9	32.4	56.3	31.6	34.4	60.7	32.7	$35.6_{\uparrow\mathbf{1.2}}$	61.8	34.0
Brain tumor [17]	43.5	75.1	45.0	38.9	71.6	38.6	$46.5_{\uparrow\mathbf{7.6}}$	75.6	49.4	46.2	74.2	49.6	46.4	76.8	49.1	$48.6_{\uparrow\mathbf{2.2}}$	77.8	52.2
Document [31]	48.0	66.2	55.6	44.7	64.1	50.3	$50.9_{\uparrow\mathbf{16.2}}$	66.6	58.4	50.3	66.3	58.9	47.7	63.2	55.9	$49.6_{\uparrow\mathbf{1.9}}$	65.8	58.1
Smoke [32]	67.1	92.9	80.6	66.5	91.6	77.8	$71.8_{\uparrow\mathbf{5.3}}$	91.9	82.9	70.4	91.3	83.5	69.4	92.4	80.7	$69.7_{\uparrow\mathbf{0.3}}$	92.6	80.4
EgoHands [32]	74.9	96.9	90.4	74.4	97.4	89.2	$77.6_{\uparrow\mathbf{3.3}}$	98.3	91.5	76.4	96.9	91.5	77.7	97.9	91.8	$78.0_{\uparrow\mathbf{0.3}}$	98.0	91.6
PlantDoc [37]	38.9	60.8	44.9	45.0	61.7	53.7	$49.1_{\uparrow\mathbf{4.1}}$	63.9	56.5	37.5	55.3	43.6	35.1	49.7	39.9	$38.3_{\uparrow\mathbf{3.2}}$	53.8	44.2
Paintings [33]	17.0	50.1	6.3	2.2	5.8	1.2	$13.4_{\uparrow\mathbf{11.2}}$	33.1	9.0	18.0	50.7	8.1	12.0	30.3	6.7	$13.4_{\uparrow\mathbf{1.4}}$	34.3	7.9
UniAP	41.3			38.5			$\mathbf{44.2}_{\uparrow\mathbf{5.7}}$			43.0			43.6			$\mathbf{45.0}_{\uparrow\mathbf{1.4}}$		
UniAP$_{75}$			44.6			42.0			$\mathbf{48.2}_{\uparrow\mathbf{6.2}}$			46.7			47.5			$\mathbf{49.0}_{\uparrow\mathbf{1.5}}$

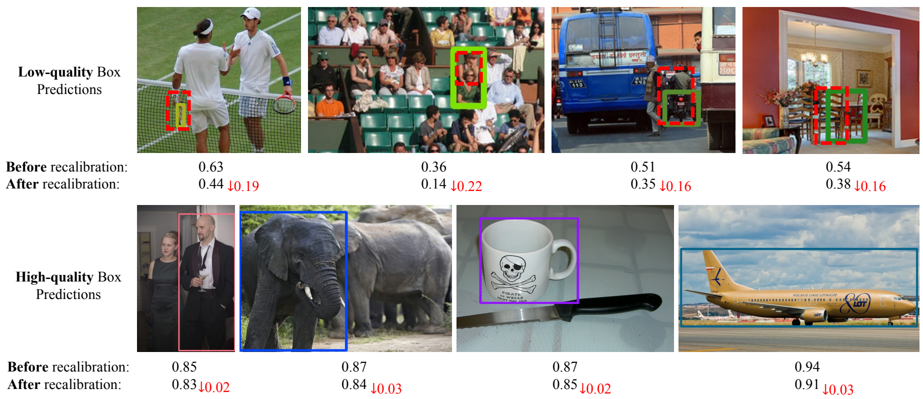


Figure 7. Predicted boxes and corresponding scores of Cascade-DN-DETR before and after IoU-aware query re-calibration. In the first row, we visualize both the box prediction by our Cascade-DN-DETR and the corresponding GT boxes (in a dotted line of red color). The first row shows that for low-quality predicted boxes (with small IoUs to the GT boxes), their confidence scores after re-calibration will have an obvious decrease to align with the low localization quality. The second row shows that for high-quality box predictions with high IoUs to GT boxes (not shown here due to overlapping), the re-calibration has a negligible influence on the original classification score.

nents of the constructed UDB10. The compared six methods include Faster R-CNN [35], Cascade R-CNN [2], DN-Def-DETR [24], Cascade-DN-Def-DETR (Ours), the most recent DINO [52] and Cascade-DINO (Ours). Cascade-DINO achieves the best UniAP 45.0 and UniAP$_{75}$ 49.0 among all comparing methods. It's worth mentioning that both Cascade-DINO and Cascade-DN-Def-DETR boost the performance of their strong baselines on all 10 dataset components consistently. This shows the generalizability and effectiveness of our proposed cascade attention and IoU-aware query re-calibration. Interestingly, we observe that although DINO obtains over 3.3 AP advantage over Cascade-DN-Def-DETR on the COCO dataset, its UniAP is 0.6 points lower than our Cascade-DN-Def-DETR (43.6 *vs.* 44.2). This indicates that the robustness of the most recent DINO [52] across domains still has improvement space.

4.5. Qualitative Analysis

In Figure 7, we visualize the predicted boxes and corresponding confidence scores before and after IoU-aware query re-calibration. For the low-quality box predictions with small IoUs to GT, their scores typically have an obvious decrease of around 0.2. However, for the high-quality boxes, the re-calibration has minor influences (around 0.02) on the predicted scores. The recalibration adjusts the box confidence score to better reveal its localization quality.

5. Conclusion

We present Cascade-DETR, the first DETR-based detector targeting for high-quality universal detection. To benefit future research on universal detection, we propose a large-scale universal object detection benchmark UDB10, which is composed of 10 sub-datasets from various real-life domains. Injected with local object-centric prior, Cascade-DETR achieves significant advantages in a wide range of detection applications, especially in higher IoU thresholds. We hope the detection community to focus more on real-life and practical applications when evaluating the detector performance, not only considering the *de facto* COCO, especially for the data-sensitive DETR-based approaches.

References

[1] Sven Bambach, Stefan Lee, David Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *ICCV*, 2015.

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

[3] Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. D2det: Towards high quality object detection and instance segmentation. In *CVPR*, 2020.

[4] Xipeng Cao, Peng Yuan, Bailan Feng, and Kun Niu. Cf-detr: Coarse-to-fine transformers for end-to-end object detection. In *AAAI*, 2022.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[6] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, 2019.

[7] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018.

[8] Zehui Chen, Chenhongyi Yang, Qiaofei Li, Feng Zhao, Zheng-Jun Zha, and Feng Wu. Disentangle your dense object detector. In *ACM MM*, 2021.

[9] Zhe Chen, Jing Zhang, and Dacheng Tao. Recurrent glimpse-based decoder for detection with transformer. In *CVPR*, 2022.

[10] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[11] Floriana Ciaglia, Francesco Saverio Zuppichini, Paul Guerrie, Mark McQuade, and Jacob Solawetz. Roboflow 100: A rich, multi-domain object detection benchmark. *arXiv preprint arXiv:2211.13523*, 2022.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[13] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *ICCV*, 2021.

[14] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[16] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *ICCV*, 2021.

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

[18] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[19] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019.

[20] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018.

[21] Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask transfiner for high-quality instance segmentation. In *CVPR*, 2022.

[22] Lei Ke, Henghui Ding, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Video mask transfiner for high-quality video instance segmentation. In *ECCV*, 2022.

[23] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv:2306.01567*, 2023.

[24] 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.

[25] Xiang Li, Wenhai Wang, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection, 2021.

[26] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[28] Fanfan Liu, Haoran Wei, Wenzhe Zhao, Guozhen Li, Jingquan Peng, and Zihao Li. Wb-detr: transformer-based detector without backbone. In *ICCV*, 2021.

[29] 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.

[30] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, 2021.

[31] Christian Green. Oliver Giesecke. Document parts dataset. *Roboflow Universe*, 2022.

[32] Matteo Pacini. Smoke dataset. *Roboflow Universe*, 2022.

[33] AI Raya. People in paintings dataset. *Roboflow Universe*, 2022.

[34] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

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

[36] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse detr: Efficient end-to-end object detection with learnable sparsity. In *ICLR*, 2022.

[37] Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. Plantdoc: A dataset

for visual plant disease detection. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, 2020.

[38] Hwanjun Song, Deqing Sun, Sanghyuk Chun, Varun Jampani, Dongyoon Han, Byeongho Heo, Wonjae Kim, and Ming-Hsuan Yang. Vidt: An efficient and effective fully transformer-based object detector. In *ICLR*, 2022.

[39] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *ICCV*, 2021.

[40] Christian Szegedy, Scott Reed, Dumitru Erhan, Dragomir Anguelov, and Sergey Ioffe. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2014.

[41] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: A simple and strong anchor-free object detector. *TPAMI*, 2020.

[42] Tao Wang, Li Yuan, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Pnp-detr: Towards efficient visual analysis with transformers. In *ICCV*, 2021.

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

[44] Wen Wang, Jing Zhang, Yang Cao, Yongliang Shen, and Dacheng Tao. Towards data-efficient detection transformers. In *ECCV*, 2022.

[45] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *CVPR*, 2019.

[46] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022.

[47] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021.

[48] 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.

[49] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P Xing. Meta-detr: Image-level few-shot detection with inter-class correlation exploitation. *TPAMI*, 2022.

[50] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. In *CVPR*, 2022.

[51] Hongkai Zhang, Hong Chang, Bingpeng Ma, Naiyan Wang, and Xilin Chen. Dynamic r-cnn: Towards high quality object detection via dynamic training. In *ECCV*, 2020.

[52] 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.

[53] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *CVPR*, 2022.

[54] Li Zhu, Zihao Xie, Liman Liu, Bo Tao, and Wenbing Tao. Iou-uniform r-cnn: Breaking through the limitations of rpn. In *Pattern Recognition*, 2021.

[55] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.