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

Mingqiao Ye\textsuperscript{1}\textsuperscript{*} Lei Ke\textsuperscript{1,2}\textsuperscript{*} Siyuan Li\textsuperscript{1} Yu-Wing Tai\textsuperscript{3} Chi-Keung Tang\textsuperscript{2} Martin Danelljan\textsuperscript{1} Fisher Yu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \quad \textsuperscript{2}HKUST \quad \textsuperscript{3}Dartmouth College

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.

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...
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. During inference, we recalibrate the 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
In this section, we describe the architecture of 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. Cascade-DETR

We propose Cascade-DETR for high-quality and universal object detection. We first review the design of the conventional 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 $Q_i \in \mathbb{R}^{N \times D}$ 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,

$$Q_{i+1} = \sum_{j=1}^{H \times W} \frac{\exp(f_q(Q_i) \cdot K_j^i \cdot V_j^i)}{\sum_{k} \exp(f_q(Q_i) \cdot K_k^i \cdot V_k^i)} + Q_i,$$

where $K$ and $V$ 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 $Q_{i+1}$ are then used to predict bounding boxes $B_{(i+1)}$ and query scores $S_{(i+1)}$ by feeding them into two parallel linear layers $f_{box}$ and $f_{score}$ respectively, i.e., $B_{(i+1)} = f_{box}(Q_{(i+1)})$ and $S_{(i+1)} = f_{score}(Q_{(i+1)})$. The query score matrix $S_{(i+1)}$ 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
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.

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,

\[
Q_{i+1} = \frac{\exp(f_q(Q_i) \cdot K_i^c)}{\sum_{k \in S_i} \exp(f_q(Q_i) \cdot K_i^c)} V_i^c + Q_i, \tag{2}
\]

where \( S_i \) is the set of 2D locations inside the predicted bounding box \( B_i \) from the preceding decoder layer \( i \). The cascade structure utilizes the property that the predicted \( B_i \) will be more accurate after every decoder layer in DETR-based detectors [5]. Thus, the box-constrained cross-attention region \( S_i \) 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 randomly initialized query eventually converge 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 an object, as obtained from the classification probability. The expected IoU of a query is computed as

\[
E(\text{IoU}_q) = E(\text{IoU}_q | \text{obj}_q)P(\text{obj}_q) + E(\text{IoU}_q | \neg \text{obj}_q)P(\neg \text{obj}_q)
\]

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: \( E(\text{IoU}_q | \neg \text{obj}_q) = 0 \).

To predict the expected IoU (4), we introduce an additional branch that predicts the expected IoU for a present ground-truth object \( E(\text{IoU}_q | \text{obj}_q) \), 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 expected 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 \( \text{IoU}^\text{GT}_q \).

\[
L_{\text{IoU}} = \| E(\text{IoU}_q | \text{obj}_q) - \text{IoU}^\text{GT}_q \|^2.
\]

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, \textit{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.](image)

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,

\[
\mathcal{L}_{\text{Detect}} = \mathcal{L}_{\text{Box}} + \lambda_1 \mathcal{L}_{\text{Class}} + \lambda_2 \mathcal{L}_{\text{IoU}},
\]

where \( \mathcal{L}_{\text{Detect}} \) supervises both the position prediction and the category classification borrowed from the DETR [5] detector. The hyper-parameters \( \lambda_1 \) and \( \lambda_2 \) 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 pre-trained on ImageNet-1k, and train our model with an initial learning rate $1 \times 10^{-5}$ for backbone and $1 \times 10^{-4}$ for transformer. We use the AdamW optimizer with weight decay $1 \times 10^{-4}$. 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 multiscale 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.

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 over which is much larger than gain of 3.0 AP over 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 multiscale 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.

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 over which is much larger than gain of 3.0 AP over 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 multiscale 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.

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

<table>
<thead>
<tr>
<th>Model</th>
<th>UVO AP</th>
<th>UVO AP50</th>
<th>UVO AP75</th>
<th>Cityscapes AP</th>
<th>Cityscapes AP50</th>
<th>Cityscapes AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>DN-DETR [24]</td>
<td>✓</td>
<td>22.3</td>
<td>41.5</td>
<td>21.2</td>
<td>51.4</td>
<td>15.7</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>27.0</td>
<td>44.5</td>
<td>26.1</td>
<td>53.9</td>
<td>21.2</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>28.4</td>
<td>44.9</td>
<td>28.7</td>
<td>58.2</td>
<td>28.4</td>
</tr>
</tbody>
</table>

Table 3. Ablation study on the query recalibration loss on the UVO dataset. **Baseline:** DN-Deformable-DETR.

<table>
<thead>
<tr>
<th>Cross-attention Type</th>
<th>UVO AP</th>
<th>UVO AP50</th>
<th>UVO AP75</th>
<th>COCO AP</th>
<th>COCO AP50</th>
<th>COCO AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask Attention (Oracle)</td>
<td>29.0</td>
<td>44.6</td>
<td>29.5</td>
<td>56.8</td>
<td>44.2</td>
<td>61.1</td>
</tr>
<tr>
<td>Cascade Attention (Ours)</td>
<td>28.4</td>
<td>44.9</td>
<td>28.7</td>
<td>58.2</td>
<td>44.5</td>
<td>62.7</td>
</tr>
</tbody>
</table>

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.

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

<table>
<thead>
<tr>
<th>Loss type</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
<th>AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>22.3</td>
<td>41.5</td>
<td>21.2</td>
<td>51.4</td>
</tr>
<tr>
<td>Huber Loss</td>
<td>25.3</td>
<td>43.8</td>
<td>25.4</td>
<td>53.6</td>
</tr>
<tr>
<td>L1 Loss</td>
<td>25.9</td>
<td>44.6</td>
<td>26.3</td>
<td>53.9</td>
</tr>
<tr>
<td>L2 Loss</td>
<td>25.9</td>
<td>44.5</td>
<td>26.1</td>
<td>53.9</td>
</tr>
</tbody>
</table>

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 AP75 than the loose AP50, 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 AP75 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 AP75. The significant increase in AP75 is also consistent on Cityscapes. Comparing to our baseline DN-DETR, in Table 10, Cascade-DN-DETR substantially improves the AP75 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
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-Def-DETR has already been significantly sped up by its denoising branch during training. Our Cascade-DN-DETR outperforms the strong baseline DN-Def-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 13. Quantitative 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP</td>
<td>AP50</td>
<td>AP75</td>
<td>AP</td>
<td>AP50</td>
<td>AP75</td>
</tr>
<tr>
<td>COCO</td>
<td>37.9</td>
<td>48.1</td>
<td>41.4</td>
<td>43.4</td>
<td>61.9</td>
<td>47.2</td>
</tr>
<tr>
<td>UVO</td>
<td>24.7</td>
<td>48.4</td>
<td>22.1</td>
<td>22.3</td>
<td>41.5</td>
<td>21.2</td>
</tr>
<tr>
<td>Cityscapes [12]</td>
<td>30.1</td>
<td>53.2</td>
<td>30.3</td>
<td>19.0</td>
<td>39.3</td>
<td>15.7</td>
</tr>
<tr>
<td>BDD100K [48]</td>
<td>31.0</td>
<td>53.9</td>
<td>29.4</td>
<td>28.2</td>
<td>53.9</td>
<td>24.8</td>
</tr>
<tr>
<td>Brain tumor [17]</td>
<td>43.5</td>
<td>75.1</td>
<td>45.0</td>
<td>38.9</td>
<td>71.6</td>
<td>38.6</td>
</tr>
<tr>
<td>Document [31]</td>
<td>48.0</td>
<td>66.2</td>
<td>55.6</td>
<td>44.7</td>
<td>64.1</td>
<td>50.3</td>
</tr>
<tr>
<td>Smoke [32]</td>
<td>67.1</td>
<td>92.9</td>
<td>80.6</td>
<td>66.5</td>
<td>91.6</td>
<td>77.8</td>
</tr>
<tr>
<td>EgoHands [32]</td>
<td>74.9</td>
<td>96.9</td>
<td>90.4</td>
<td>74.4</td>
<td>97.4</td>
<td>89.2</td>
</tr>
<tr>
<td>PlantDoc [37]</td>
<td>38.9</td>
<td>60.8</td>
<td>44.9</td>
<td>45.0</td>
<td>61.7</td>
<td>53.7</td>
</tr>
<tr>
<td>Paintings [33]</td>
<td>17.0</td>
<td>50.1</td>
<td>6.3</td>
<td>2.2</td>
<td>5.8</td>
<td>12.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UniAP</td>
<td>41.3</td>
<td>38.5</td>
<td>44.6</td>
<td>42.0</td>
<td>48.2</td>
<td>42.2</td>
</tr>
</tbody>
</table>

| UniAP50         | 44.6  | 42.0  | 48.2  | 42.2  | 46.7  | 47.5  | 49.0  | 49.0  |        |      |      |      |        |      |      |

| UniAP75         |        |      |      |        |      |      |        |      |      |      |      |      |        |      |      |

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.

Figure 8. Box predictions and corresponding scores of Cascade-DN-DETR 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


[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.