Decoupled DETR: Spatially Disentangling Localization and Classification for Improved End-to-End Object Detection

Manyuan Zhang\textsuperscript{1,2} \quad Guanglu Song\textsuperscript{2} \quad Yu Liu\textsuperscript{2} \quad Hongsheng Li\textsuperscript{1,3,4}

\textsuperscript{1} Multimedia Laboratory, The Chinese University of Hong Kong \quad \textsuperscript{2} SenseTime Research
\quad \textsuperscript{3} Centre for Perceptual and Interactive Intelligence Limited \quad \textsuperscript{4} Shanghai AI Laboratory

\texttt{zhangmanyuan@link.cuhk.edu.hk}

Abstract

The introduction of DETR represents a new paradigm for object detection. However, its decoder conducts classification and box localization using shared queries and cross-attention layers, leading to suboptimal results. We observe that different regions of interest in the visual feature map are suitable for performing query classification and box localization tasks, even for the same object. Salient regions provide vital information for classification, while the boundaries around them are more favorable for box regression. Unfortunately, such spatial misalignment between these two tasks greatly hinders DETR’s training. Therefore, in this work, we focus on decoupling localization and classification tasks in DETR. To achieve this, we introduce a new design scheme called spatially decoupled DETR (SD-DETR), which includes a task-aware query generation module and a disentangled feature learning process. We elaborately design the task-aware query initialization process and divide the cross-attention block in the decoder to allow the task-aware queries to match different visual regions. Meanwhile, we also observe that the prediction misalignment problem for high classification confidence and precise localization exists, so we propose an alignment loss to further guide the spatially decoupled DETR training. Through extensive experiments, we demonstrate that our approach achieves a significant improvement in MSCOCO datasets compared to previous work. For instance, we improve the performance of Conditional DETR by 4.5 AP. By spatially disentangling the two tasks, our method overcomes the misalignment problem and greatly improves the performance of DETR for object detection.

1. Introduction

Object detection is a critical problem in computer vision, with traditional detectors relying on convolution to extract informative representations of the image, including single-stage and multi-stage detectors [29, 24, 28, 1, 18]. In contrast, recent work, such as DETR [3], breaks this paradigm. DETR consists of an encoder and a decoder, with the encoder extracting image features using self-attention and the decoder using these features to estimate object locations and categories in interaction with the object query. It is an end-to-end solution that does not require post-processing, such as non-maximum suppression (NMS).

However, the lack of a good positional prior to matching the queries to the visual feature map has been observed to slow down DETR’s convergence. As a result, subsequent work has focused on improving performance by developing various techniques for initializing object queries [23, 27, 7, 34, 37]. On the other hand, previous RCNN-based object detection work [13, 35, 32] has shown
that sharing the detection head for both classification and localization tasks may result in suboptimal performance, as the classification and localization branches may have conflicting learning targets and their actual regions of interest may not be well-aligned with each other. To address this issue, Double-Head R-CNN [35] disentangles the detection head into two dedicated branches for classification and localization, respectively. While the disentanglement of the detection head can yield satisfactory performance, there persists a conflict between the two tasks due to the fact that the features fed into both branches originate from the same proposal through the use of ROI pooling. Additionally, as DETR relies heavily on attention mechanisms for information extraction, disentanglement methods used in previous RCNN-based detectors that rely on anchors or convolutional features cannot be directly transferred to the DETR-based detector.

Our study highlights the issue of misalignment between classification and localization tasks also exists in DETR, which has been ignored for a long time. To illustrate this problem, we conduct a pilot study to decouple the classification and localization branches entirely by creating two copies of the decoder. We then visualize the neuron activation maps for each of the two branches, as shown in Figure 1. The second and third columns display the cross-attention maps for classification and localization branches, respectively. The highlighted activations of each branch are significantly different, indicating a significant semantic misalignment. Further analysis shows that features from different positions within an object make varying contributions to classification and localization tasks. For instance, salient regions within an object provide vital information for classification, while the boundaries around objects are more favorable for box localization.

To take advantage of the observation, we propose a decoupled design scheme for the DETR decoder, as depicted in Figure 2. However, we do not naively adopt two totally isolated branches. Instead, we only split the cross-attention block in the decoder into two branches, allowing classification and localization to perform query matching with different regions of the visual feature map. Importantly, the two branches share the self-attention layers, enabling them to cooperate with each other in detecting the same objects. By decoupling cross-attention for classification and localization tasks, our proposed method achieves improved performance over existing DETR detectors.

Furthermore, we highlight the importance of query initialization in DETR’s decoder for achieving good performance and convergence speed. In the original DETR, the input queries consist of a content query and a randomly initialized positional embedding. However, after decoupling the classification and localization branches, initializing the content and positional embeddings becomes critical. To address this, we introduce a task-aware query generation module that learns task-specific queries based on anchor boxes. We first find some discriminative points within the anchors boxes and then the content embedding initialization is sampled from the encoder’s feature map for those discriminative points. While the positional embedding is generated using sinusoidal embedding of the offsets for those points.

We also observe that the misalignment problem between accurate classification and precise localization persists, resulting in high classification confidence with relatively low intersection-over-union (IoU) scores, or vice versa. Inspired by the aligned label assignment mechanism proposed in [6, 8, 16], we further propose an alignment loss to guide the consistency between high classification confidence and precise localization in our Decoupled DETR learning framework.

We summarize our contributions as follows:

- We reveal the feature and prediction misalignment problem of the classification and localization branch in DETR, which significantly limits the performance of DETR-like detectors.
- We disentangle the feature learning process for the classification and localization branches. We split the cross-attention in the decoder to allow the two branches to match different areas. We design task-aware query generation for better query initialization for the two branches. We also propose an alignment loss to guide the consistency between high classification confidence and precise localization.
• We integrate our structure into a wide range of variants of DETR, and a large number of experiments on MSCOCO demonstrate that our approach can lead to significant improvements.

2. Related Work

2.1. Anchor-based Detector

Deep learning have deeply changed the field of computer vision [39, 38, 21, 30, 31, 22, 25, 11], and one widely studied area is object detection. In the past, a significant number of object detection methods have relied on anchors, such as the Faster R-CNN [29, 2] and YOLO [24, 1, 28, 19] series, which predict the offsets between the detected object and predefined anchors. Other approaches, such as Corner-Net [14], CenterNet [5], Scale-aware Detectors [17], and FCOS [40], utilize anchor points to characterize the prior and regress the distance between the object and the points. The anchor mechanism provides a useful assumption for object detection.

2.2. DETR and its Variants

The proposed DETR [3] provides a novel paradigm for object detection that relies on an encoder for further extraction of image content and a decoder for information matching between queries and encoder features to obtain object category and position information. Unlike traditional detectors, DETR eliminates the need for non-maximum suppression (NMS). However, the original DETR suffers from several issues, such as high computational complexity and slow convergence speed.

To tackle the issue of computational complexity, PNP-DETR [33] employs a poll-and-pool strategy to combine redundant tokens while retaining only valuable ones. Deformable-DETR [41] uses the deformable mechanism to dynamically select tokens for multi-head attention calculation, significantly reducing the computational effort and making the complexity independent of the number of tokens.

As for the slow convergence issue, the original DETR typically requires 500 epochs to converge. Many works attribute this to completely random object query initialization in the decoder. SMCA [7] introduces a Gaussian mechanism to constrain global cross-attentions to focus more on specific regions, reducing the difficulty of matching. Conditional DETR [27] decouples content from positional matching, allowing the model to search the extremity. Anchor-DETR [34] introduces the anchor mechanism, where an anchor corresponds to a query, making the query initialization interpretable. DAB-DETR [23] modulates the positional attention map using box width and height information based on the anchor box, making the matching prior more specific. DN-DETR [15], Group DETR [4], and Hybrid DETR [12] focus on the inefficiency of one-to-one matching on the Hungarian loss and transform it into one-to-many matching to accelerate convergence.

A former work by He et al. [10] also addresses the feature misalignment problem of the classification and localization branch in DETR by splitting the cross-attention layer. However, their approach overlooks the importance of proper query embedding initialization and the misalignment problem for high classification confidence and precise localization.

2.3. Feature Decoupling Methods

The misalignment between the classification and localization branches has been extensively investigated in object detection during the convolution era. For example, IoU-Net [13] found that the feature generating a high classification score typically predicts a coarse bounding box. To address this, they introduced an additional head to predict the Intersection over Union (IoU) as the localization confidence and then combined this with the classification confidence to obtain the final classification score. Double-Head R-CNN [35] disentangles the sibling head into two separate branches for classification and localization. Another approach, TSD [32], spatially disentangles the gradient flows for classification and localization. While some label assignment methods such as TOOD [6] and MuSu [8] propose a new anchor alignment metric integrated into the sample assignment and loss functions to dynamically guide the consistency between high classification confidence and precise localization. While those methods have been successful in the convolution era, their efficacy in DETR requires further validation.

3. Method

In this section, we first revisit the original design scheme of DEtection TRansformer (DETR) [3] and then describe our proposed spatially decoupled DETR (SD-DETR). The sub-modules will be introduced in Section 3.2.1 and Section 3.2.2. Then a novel alignment loss that further guides the consistency of high classification confidence and precise localization will be proposed in Section 3.2.3. Finally, we delve into the inherent problem in the original query and decoder fully shared by classification and localization and demonstrate the advantage of our spatially decoupled DETR.

3.1. Revisit the General DETR Pipeline

DETR is a flexible end-to-end detector that views object detection as a set prediction problem. It pre-defines multiple queries and introduces a one-to-one set matching scheme based on a transformer encoder-decoder architecture. Each ground truth is assigned to a specific query as the supervised target for classification and localization. Specifically, given
an input image $I$, its visual feature $F$ can be generated by the backbone and transformer encoder. Let $Q = \{q_0, \ldots, q_n\}$ denotes the pre-defined data-independent queries. The output query embeddings can be generated via:

$$
\hat{Q} = \{\text{self-att}(Q), \text{cross-att}(Q, F), \text{FFN}(Q)\}_{1:L}, \tag{1}
$$

where $\text{self-att}(\cdot)$ are the self-attention that propagates information between queries $q_i$, $\text{cross-att}(\cdot)$ is the cross-attention that absorbs knowledge from $F$ to support classification and localization, and $\text{FFN}(\cdot)$ is the feedforward network. These operations are serially stacked for $L$ times to establish DETR's decoder. $\hat{Q}$ is updated after being forwarded in each operation. Then, DETR applies task-specific prediction heads on $\hat{Q}$ to generate a set of predictions $\hat{P} = \{p_0, \ldots, p_n\}$, which can be formulated as:

$$
p_i = (p_i^{cls}, p_i^{loc}) = (F_{cls}(\hat{q}_i), F_{loc}(\hat{q}_i)), \tag{2}
$$

where $\hat{q}_i \in \hat{Q}$, $p_i^*$ is the predictions for object classification or localization, and $F_\cdot$ indicates the two heads. Finally, DETR adopts the one-to-one bipartite matching to assign ground truths to $\hat{P}$. In the original DETR training, all operations in Eq. (1) use a shared-weight decoder.

### 3.2. Spatially Decoupled DETR

As we analyzed above, the inherent conflict caused by the shared queries in different tasks, i.e., classification and localization, and the shared cross-attention operation in different queries greatly limits the performance of DETR-based detectors. For one instance, the features in some salient areas may have rich information for classification, while these around the boundary may be good at bounding. The fully shared paradigm in DETR impedes it from learning better task-specific features to further improve its performance.

For this potential problem, we introduce the spatially decoupled DETR to alleviate this conflict by disentangling the tasks from two aspects, disentangled feature learning (DFL) and task-aware query generation. In DFL, Eq.(1) is adjusted by:

$$
\hat{Q} = \{\text{self-att}(\text{cat}(Q_{cls}, Q_{loc})), \text{cross-att}_{cls}(Q_{cls}), \text{FFN}_{cls}(Q_{cls}), \text{cross-att}_{loc}(Q_{loc}), \text{FFN}_{loc}(Q_{loc})\}_{1:L}, \tag{3}
$$

where $Q_{cls}$ and $Q_{loc}$ are the classification-aware and localization-aware queries generated by the task-aware query generation module. The cross-attention module and FFN module are not shared between $Q_{cls}$ and $Q_{loc}$. This generation process is formulated as follows:

$$
Q_{cls} = G_{cls}(F, R_{box}), \tag{4}
$$

where $R_{box}$ is a series of anchor boxes. We use the minidetector module proposed in [10, 41] to initialize those anchor boxes. $G_{cls}$ is the task-aware query generation process which will be introduced in 3.2.2

By disentangling the queries and feature learning in cross-attention and FFN, our spatially decoupled DETR can learn the better task-aware feature representation adaptively. It’s applicable to most existing DETR-based detectors whilst introducing few overheads. The overall pipeline of our spatially decoupled DETR can be seen in Figure 3.
3.2.1 Disentangled Feature Learning

To fully utilize the network’s capability, we need to design a disentangled feature learning architecture for these two sub-tasks. The simplest approach is to directly split the transformer decoder, but this results in a suboptimal design due to a lack of information propagation between the two branches and a significant increase in model parameters.

In this work, we keep the self-attention block and split the cross-attention block in the decoder. The two branches share the self-attention block to enhance the information propagated between them. Obtained the task-aware object query initialization, we concatenate the classification query and localization query for the self-attention block. So the output for the self-attention block will be \( O_{self} \in \mathbb{R}^{2C} \). Then we split the \( O_{self} \) into two parts for the two branches, each for \( C \)-dimensional vector. Those two features will conduct cross-attention separately with the image feature output from the encoder. Each branch will search its matching interest area and distill relevant features, avoiding feature misalignment between those two branches.

When conducting the cross-attention, we concatenate the content query output from the split self-attention and positional embedding initialized based on the task-aware anchor prior to forming the object query for conditional matching [27, 23]. The cross-attention output \( O_{cross}^{cls} \in \mathbb{R}^{C} \) and \( O_{cross}^{loc} \in \mathbb{R}^{C} \) are passed to normalization, FFN, and residual unit layer and then to the next decoder stage. Then the classification head and box regression head will take the final stage disentangled feature output as input, respectively, and finally obtain the object’s category and location.

3.2.2 Task-Aware Query Generation

In the previous section, we divided the cross-attention block in the decoder for disentangled feature learning. In this section, we focus on generating task-aware queries to enable each branch’s cross-attention to concentrate on their respective regions of interest and eliminate feature conflicts. We introduce the task-aware query generation module. Building upon the mini-detector proposed in [41, 10], we begin by obtaining a set of anchor boxes \( R_{box} \). Next, we modify the semantic-aligned matching module proposed in [36]. Specifically, we employ ROIAAlign to extract region-level features \( F_{R} \in \mathbb{R}^{N \times T \times T \times d} \) from the encoded features \( F \) for the region of the anchor boxes \( R_{box} \).

To effectively capture object features within the anchor boxes, we select the most discriminative points to generate object content embeddings. As illustrated in Figure 3, we utilize a ConvNet and an MLP to obtain the coordinates \( R_{SP} \in \mathbb{R}^{N \times M \times 2} \) for these points within each region after obtaining region features \( F_{R} \) through RoIAlign.

\[
R_{SP} = \text{MLP} \left( \text{ConvNet} \left( F_{R} \right) \right) \quad (5)
\]

These discriminative points are crucial for object recognition and localization. We employ bilinear interpolation to obtain their features. We then calculate the average feature and offsets from these discriminative points for each branch. Subsequently, we use these average features to update the query content embedding. Meanwhile, we employ the positional encoding function \( \text{PE} \) to generate positional embeddings for the average offsets, which are used to update the positional embedding for the learnable query.

3.2.3 Task Alignment Learning

In the previous two sections, we discussed the decoupling of the classification and localization branches in DETR. However, the misalignment between accurate classification and precise localization can significantly hinder the effectiveness of learning when generating predictions from object queries. This misalignment refers to situations where a query yields a high confidence classification but relatively low intersection-over-union (IoU) scores, or vice versa. Inspired by previous label assignment work [6, 8, 16], we made modifications to the loss function of DETR. Our goal is to ensure that both high classification scores and precise localization are achieved simultaneously. To accomplish this, we measure the task alignment based on a high-order combination of the classification score and the IoU. Specifically, we have designed the following metric to calculate the alignment for each query:

\[
t = s^{\alpha} \times u^{\beta} \quad (6)
\]

Here, \( s \) represents the classification score, and \( u \) denotes the IoU value. The parameters \( \alpha \) and \( \beta \) are utilized to control the relative impact of the two tasks in the alignment metric. We then use \( t \) to replace the binary label of positive samples during training, which encourages the learning process to dynamically prioritize high-quality queries. The Binary Cross Entropy (BCE) for the classification task can be rewritten as:

\[
\mathcal{L}_{cls} = \sum_{i=1}^{N_{pos}} |t_{i} - s_{i}| \gamma \text{BCE}(s_{i}, t_{i}) + \sum_{j=1}^{N_{neg}} s_{j}^{\gamma} \text{BCE}(s_{j}, 0), \quad (7)
\]

Here, \( N_{pos} \) and \( N_{neg} \) represent the number of positive and negative samples, respectively, and \( \gamma \) is the focusing parameter. To further increase the matching efficiency, we repeat the positive label for several times to provide a richer positive supervised signal.

3.3. Discussion

Figure 4 presents the cross-attention maps for the classification and localization branches. These maps are obtained by applying softmax normalization over the dot product.
4. Experiments

4.1. Experimental Setup

**Dataset** All experiments are conducted on the COCO 2017 dataset [20], which consists of 117k training examples and 5k validation images.

**Training** We follow the standard training protocol for DETR [12]. We use ResNet [9] as backbones from the TORCHVISION ImageNet-pretrained model zoo. The batch norm layers are fixed, and the transformer parameters are initialized with the Xavier initialization scheme.

We use the AdamW optimizer [26] and train for 50 epochs. The weight decay is set to $10^{-4}$. The learning rate for the backbone and transformer is $10^{-5}$ and $10^{-4}$, respectively. The learning rate is dropped by ten after 40 epochs. We use a dropout rate of 0.1 for the transformer. We keep the number of multi-head to 8 and the attention channel to 256. The default query number is 300. To ensure a fair comparison with other works, we may change the training settings. For the architecture, we use six encoder layers and six decoder layers. We use bipartite matching via the Hungarian algorithm when calculating the loss function. We repeat the positive samples for twice. For the task alignment loss, the $\alpha$ is set to 0.25 and $\beta$ 0.75 respectively. The focusing parameter is 2, following [19]. For the box regression loss, we apply the L1 and generalized IoU loss. We use the same data augmentation as DETR. We resize the input image to the short side within the range [480, 800] and the long side to 1333 pixels. We also conduct a random crop with a probability of 0.5.

**Evaluation** We follow the standard COCO evaluation and report the average precision (AP) at 0.50, 0.75 for small, medium, and large objects.

4.2. Results

Table 1 presents our main results on the COCO 2017 [20] validation set. We compared our proposed spatially decoupled DETR with various state-of-the-art methods including DETR [3], Faster RCNN [29], Anchor DETR [34], SMCA [7], Deformable DETR [41], Conditional DETR [27], DAB-DETR [23], DESTR [10]. Our SD-DETR outperformed all previous methods by a significant margin, achieving a 4.5 AP improvement compared to Conditional DETR. In comparison to DESTR, our method achieved a performance gain of 1.9 AP.

We also tested a stronger backbone, R50-DC5, and our proposed spatially decoupled DETR consistently improved the performance of the original DETR and its variants under various settings. We outperform Conditional DETR for 3.2 AP and DAB-DETR for 2.5 AP. These results demonstrate that decoupling the classification and localization branches in the DETR decoder eliminates both feature and prediction misalignment, enabling accurate classification and localization at different locations. In Figure 5, we show some detection results of our SD-DETR, which show remarkable performance even in highly complex scenarios.

4.3. Ablations

We conducted a comprehensive analysis of each component in our spatially decoupled DETR and assessed its impact on the final results, as shown in Table 2. The table
<table>
<thead>
<tr>
<th>Method</th>
<th>multi-scale</th>
<th>Epochs</th>
<th>AP</th>
<th>AP0.5</th>
<th>AP0.75</th>
<th>AP5</th>
<th>AP25</th>
<th>AP50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster-RCNN-R50-DC5 [29]</td>
<td>✓</td>
<td>108</td>
<td>41.1</td>
<td>61.4</td>
<td>44.3</td>
<td>22.9</td>
<td>45.9</td>
<td>55.0</td>
</tr>
<tr>
<td>Faster-RCNN-FPN-R50 [29, 18]</td>
<td>✓</td>
<td>108</td>
<td>42.0</td>
<td>62.1</td>
<td>45.5</td>
<td>26.6</td>
<td>45.4</td>
<td>53.4</td>
</tr>
<tr>
<td>DETR-R50 [3]</td>
<td></td>
<td>150</td>
<td>42.0</td>
<td>62.4</td>
<td>44.2</td>
<td>20.5</td>
<td>45.8</td>
<td>61.1</td>
</tr>
<tr>
<td>DETR-R50-DC5 [3]</td>
<td></td>
<td>150</td>
<td>43.3</td>
<td>63.1</td>
<td>45.9</td>
<td>22.5</td>
<td>47.3</td>
<td>61.1</td>
</tr>
</tbody>
</table>

Baseline methods trained for long to address:

| Faster-RCNN-R50 [29]         | 12      | 35.7 | 56.1 | 38.0  | 19.2   | 40.9 | 48.7 |
| DETR-R50 [3]                 | 12      | 23.3 | 39.5 | 22.2  | 6.6    | 22.8 | 36.6 |
| Deformable-DETR-R50 [41]     | 12      | 31.8 | 51.4 | 33.5  | 15.0   | 35.7 | 44.7 |
| Conditional-DETR-R50 [27]    | 12      | 32.2 | 52.1 | 33.4  | 13.9   | 34.5 | 48.7 |
| SMCA-DETR-R50 [7]            | 12      | 31.6 | 51.7 | 33.1  | 14.1   | 34.4 | 46.5 |
| SAM-DETR-R50 [36]            | 12      | 33.1 | 54.2 | 33.7  | 13.9   | 36.5 | 51.7 |
| DESTR-R50 [10]               | 12      | 35.9 | 56.8 | 39.7  | 20.1   | 41.7 | 50.0 |
| DAB-DETR-R50 [23]            | 12      | 34.9 | 55.3 | 36.5  | 16.4   | 38.1 | 51.3 |

Comparison of with other detectors under 12 epochs training schemes:

| Faster-RCNN-R50 [29]         | 36      | 38.4 | 58.7 | 41.3  | 20.7   | 42.7 | 53.1 |
| DETR-R50-DC5 [3]             | 12      | 25.9 | 44.4 | 26.0  | 7.9    | 27.1 | 41.4 |
| Deformable-DETR-R50-DC5 [41] | 12      | 34.9 | 54.3 | 37.6  | 19.0   | 38.9 | 47.5 |
| Conditional-DETR-R50-DC5 [27]| 12      | 35.9 | 55.8 | 38.2  | 17.8   | 38.8 | 52.0 |
| SMCA-DETR-R50-DC5 [7]        | 12      | 32.5 | 52.8 | 33.9  | 14.2   | 35.4 | 48.1 |
| SAM-DETR-R50-DC5 [36]        | 12      | 38.3 | 59.1 | 40.1  | 21.0   | 41.8 | 55.2 |
| DESTR-R50-DC5 [10]           | 12      | 37.2 | 57.5 | 39.2  | 18.9   | 40.5 | 53.2 |
| DAB-DETR-R50-DC5 [23]        | 12      | 37.1 | 58.0 | 40.1  | 19.5   | 41.5 | 52.1 |

SD-DETR-R50-DC5

Comparison of with other detectors under 50 epochs training schemes:

| Faster-RCNN-R50 [29]         | 36      | 38.4 | 58.7 | 41.3  | 20.7   | 42.7 | 53.1 |
| DETR-R50 [3]                 | 50      | 34.9 | 55.5 | 36.0  | 14.4   | 37.2 | 54.5 |
| Deformable-DETR-R50-DC5 [41] | 50      | 39.4 | 59.6 | 42.3  | 20.6   | 43.0 | 55.5 |
| Conditional-DETR-R50-DC5 [27]| 50      | 41.0 | 61.8 | 43.3  | 20.8   | 44.6 | 59.2 |
| SMCA-DETR-R50-DC5 [7]        | 50      | 39.8 | 61.8 | 41.6  | 20.5   | 43.4 | 59.6 |
| SAM-DETR-R50-DC5 [36]        | 50      | 43.6 | 64.7 | 46.5  | 23.6   | 47.5 | 62.1 |
| DESTR-R50 [23]               | 50      | 42.2 | 63.1 | 44.7  | 21.5   | 45.7 | 60.3 |
| DAB-DETR-R50-DC5 [23]        | 50      | 46.8 | 66.0 | 50.4  | 29.1   | 49.8 | 62.3 |

| Deformable-DETR-R50 [41]     | ✓       | 50    | 43.8 | 62.6 | 47.7   | 26.4 | 47.1 | 58.0 |
| SMCA-DETR-R50 [7]            | ✓       | 50    | 43.7 | 63.6 | 47.2   | 24.2 | 47.0 | 60.4 |
| DAB-Deformable-DETR-R50 [23] | ✓       | 50    | 46.8 | 66.0 | 50.4   | 29.1 | 49.8 | 62.3 |

SD-DETR-R50-DC5

Table 1. Comparison of the proposed SD-DETR, other DETR-like detectors, and Faster R-CNN on MSCOCO validation set. We report the results with multiple backbones. Some method results are reported by [36].

demonstrates that the performance of our spatially decoupled DETR improves gradually with the incorporation of different modules. In the top row of the table, we show the results of our baseline, Conditional DETR [27] with minidetector [10, 41]. By employing disentangled feature learning, which decouples the cross-attention layer while maintaining shared self-attention, we achieve a 1.4 AP performance gain, highlighting the necessity of decoupling. Furthermore, through task-aware query generation, which generates more informative content and positional embeddings initialization for each branch, we further enhance the performance to 43.6. Lastly, by improving the loss function of the original DETR to address prediction misalignment issues in high-confidence classification and precise localiza-
Figure 5. Visualization of detection results for our SD-DETR. By addressing the issue of misalignment between the classification and localization branches, we have achieved a more robust detection performance, particularly in complex scenarios.

<table>
<thead>
<tr>
<th>DFL</th>
<th>TAQ</th>
<th>TAL</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>45.5</td>
</tr>
</tbody>
</table>

Table 2. Ablation study for different components in our spatially decoupled DETR. The results are reported on the MSCOCO validation set. We gradually add new components. DFL refers to disentangled feature learning, which decouples the cross-attention layer. While TQG means task-aware query generation. TAL means task alignment learning, which modifies the loss function.

<table>
<thead>
<tr>
<th>Model</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>41.7</td>
</tr>
<tr>
<td>Split decoder</td>
<td>42.2</td>
</tr>
<tr>
<td>Split cross-attention</td>
<td>43.1</td>
</tr>
</tbody>
</table>

Table 3. Performance comparison of different decouple structures on the MSCOCO validation set. We compare the performance of the decoder fully decoupled and split cross attention block only.

5. Conclusion

Our work proposes a spatially decoupled DETR model that effectively addresses feature and prediction misalignment, we significantly boost the performance to 45.5. These three contributions of our work are propagated and all are aimed at resolving the misalignment problem in DETR’s classification and localization.

Comparison of fully split decoder In this section, we investigate the impact of different decoupling structures in the decoder. The most straightforward decoupling structure is a direct copy of the decoder, with the classification branch entirely decoupled from the localization branch. However, as shown in Table 3, this structure only results in a 0.5 performance gain. Full decoupling ignores the information propagation between the two branches. Therefore, in our design, we split the cross-attention and share self-attention, which preserves information propagated between queries of different branches, resulting in a 0.9 gain and introducing fewer extra parameters.

6. Acknowledgment

This project is funded in part by the National Key R&D Program of China Project 2022ZD0161100, by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)’s InnoHK, by General Research Fund of Hong Kong RGC Project 14204021. Hongsheng Li is a PI of CPII under the InnoHK.
References


