GUIDE DETECTORS IN PIXEL SPACE WITH GLOBAL POSITIONING AND ABDUCTIVE MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

End-to-End object Detector ensembles prior knowledge in a concise framework. DETR (DEtection TRansformer) contains two steps: Learn object queries in the representation space and match the queries with boxes in the pixel space. The ambiguity of object queries in DETR lead to an uncertain assignment in the Hungarian Matching. The formulation loss in the pixel space will in turn affect the learning representations. Therefore, we propose the Abductive DETR, which learns object queries in the representation space with global positioning in the pixel space and matches object queries in the pixel space with the abductive awareness from the representation space. Experimentally, Abductive DETR can be transferred to other DETR-variants methods and achieves a satisfactory improvement. And it takes only 2 epochs to achieve the 98.7% accuracy of predicting the number of objects. Compared with other state-of-the-art methods on the MS COCO dataset, Abductive DETR also achieves outstanding performance and arrives at convergence much faster. Our code will be made publicly available soon.

1 INTRODUCTION

Object detection that predicts class labels and exact bounding boxes in the image regions of objects plays a significant role in computer vision. The development of detectors (Ren et al., 2017b; Redmon & Farhadi, 2018b; Carion et al., 2020b) makes great differences to artificial intelligence. In the early stage, the development of convolutional networks facilitates the detectors to predict more precisely. Recently, Carion *et al.* (Carion et al., 2020b) propose the DEtection TRansformer (DETR) to enhance the end-to-end detection with transformers. One of the most innovative contributions of DETR is that DETR utilizes the queries to probe the visual features from the Transformer encoders and match box prediction based on bipartite graphs. This effectively relieves the necessity of hand-designed post-processing tricks like non-maximum suppression and inspires more research to explore more elegant end-to-end detection paradigms.

However, DETR usually requires 500 epochs of training on the MSCOCO dataset (Lin et al., 2014b) compared with only 12 epochs of Faster R-CNN. In contrast to the previous CNN-based detectors, the slow training convergence limits the development of end-to-end detection. With the advancement of DETR, many works explore the representations of DETR and enhance the training procedure (Wang et al., 2021; Meng et al., 2021; Zhu et al., 2021; Sun et al., 2020; Liu et al., 2022; Dai et al., 2021a). Their insightful contributions can be roughly derived from two aspects: 1) Enhancement for Attention mechanism. 2) Representation of the query. (Sun et al., 2020) considered that the low efficiency of the cross-attention mechanism results in slow convergence and proposed the DETR with an encoder only. (Dai et al., 2021a) introduced a dynamic decoder based on RoI to optimize the regions of interest. Besides, some works (Wang et al., 2021; Meng et al., 2021; Zhu et al., 2021) introduced positions as priors to query for probing more efficient features. (Zhu et al., 2021) and (Wang et al., 2021) introduced the $2D$ reference points as queries to perform cross-attention. (Meng et al., 2021) disentangled queries considering contextual and positional information to reinforce the model with a specific spatial position.

Nevertheless, the limitations of training a DETR are still suffering from the dilemma: the quality of object queries determines the matching results and loss function, and the loss function in turn affects the optimization gradient for the encoded queries. In other words, initially ambiguous object queries are difficult to match precise box or backward a meaningful loss. In detail, the problem reflects

DETR **DAB-DETR** **Abdutive DETR**

Figure 1: Intuitively, we consider object detection as finding a specific item in pixel space. DETR Carion et al. (2020a) is more likely looking for it in a black room. DAB-DETR Liu et al. (2022) in (b) is likely looking for it with a flashlight (dynamic anchor box). And the proposed Abductive DETR in (c) chooses to turn on the light, where circles indicate the object queries.

on several perspectives: 1) Imbalance between pixel space and representation space: An obvious problem with the learnable query is that the number of queries is quite limited in contrast to the resolution of an image, which results in each query having to cover a large region. 2) The excessive reliance on the pixel distribution: Anchor-based detection method relies too much on the object distribution in the pixel space. Once the discrepancy of distributions between training and validation datasets is overloaded, anchor-based methods weaken the ability of the model for generalization. 3) Misleading from the positional priors: existing methods also utilize the positional priors of 2D references to speed up the convergence. It can assist the model to predict the box around the local regions. But the number of these references is too few in the whole visual region. These priors may bring the risks of neglecting imbalanced and dense-distributed objects. 4) The redundancy and uncertainty of the Hungarian Matching are paid little attention to addressing, which are of significance to arrive at a faster convergence.

Therefore, we conclude that in the early stage, researchers (Erhan et al., 2014; Girshick, 2015; Dai et al., 2016; Huang et al., 2015) exploit a hand-craft algorithm to extract certain visual regions and employ the network to predict the bounding boxes. In this circumstance, the relationship between network representation space and original pixel space is manually balanced. Recently, DETR and its variants develops end-to-end object detection, which can be regarded as utilizing visual features to match the bounding boxes. The relationship between representation and pixel spaces is ignored and paid limited attention. Meanwhile, due to the limitations of Hungarian Matching, DETR-like methods urgently require guidance while learning representations in the pixel space and the higher-efficiency matching process.

To address these limitations, we proposed a novel framework, namely Abductive DEction TRansformer (Abductive DETR). It aims to guide the transformer network to realize the imbalance between pixel space and representation space. Then it utilizes the global positioning queries to locate pixels and objects in specific visual regions. We present the Region Tokens (§ 4.1) to reinforce the regional attributes of learned representations and we propose a new dynamic content-aware query generator (§ 4.2) to lead decode acquiring a global position in the pixel space. In addition, to further improve the efficiency of Abductive DETR, we rethink the intrinsic of Hungarian Matching and reveal the lower efficiency of DETR to the redundancy and uncertainty in the matching process. Then we propose a novel abductive matching scheme (§ 4.3) to replace the Hungarian Loss and optimize the matching process with less redundancy and uncertainty.

In summary, Abductive DETR enhances learned representation with global positioning to reinforce the network in pixel space. And it proposes a novel target for optimization to reduce the original redundancy and uncertainty derived from Hungarian Matching. The improvements in our method can be easily plugged into other advanced DETR-like methods. We summarize our contribution as follows:

- We design a novel framework Abductive DETR for end-to-end object detection, which successfully establishes a reliable learning representations in the pixel space with region tokens from representation space to concentrate on the pre-definite attributes.

- After analyzing ambiguity and low efficiency of Hungarian Matching, we are the first to propose abductive costs to compensate these disadvantages, which reveals that abductive-aware maximum-weights matching is better for training a detector.
- Accordingly, Region Tokens (§ 4.1) concentrate on learning specific attributes of different visual regions, and explicitly reinforces the representations with region intrinsic. They takes only 2 epochs to achieve 98.7% accuracy for precisely predicting the number of objects. It proves that
- We conduct a series of ablation and comparison studies to analyze the effectiveness of different components of our model. Experimental results can validate that global positioning and abductive matching are significant to develop a more precise detector with less time.

2 Related Work

Anchor-based and Anchor-free detectors Anchor-based detectors can be roughly categorized into 2 types: one-stage and two-stage methods. Technically, two-stage methods (Girshick et al., 2014; Gao et al., 2021) firstly generate region proposals and analyze proposals to regress bounding boxes. (Ren et al., 2017b) designed a Region Proposal Network to end-to-end detect objects. One-stage methods (Redmon & Farhadi, 2016; 2018b) predict the bounding boxes related to anchors directly. Other methods like SSD (Liu et al., 2016), YOLOv2 (Redmon & Farhadi, 2017), YOLOv3 (Redmon & Farhadi, 2018a), Cascade R-CNN (Cai & Vasconcelos, 2018), Libra R-CNN (Pang et al., 2019), RetinaNet (Lin et al., 2020), TSD (Song et al., 2020), and YOLOv4 (Bochkovskiy et al., 2020) all predict boxes with the assistance of anchors. The other anchor-free detectors predict boxes according to the points nearby the centers of objects. (Redmon et al., 2016) firstly proposed the YOLOv1, and the methods like CornerNet (Law & Deng, 2018), ExtremeNet (Zhou et al., 2019b), CenterNet (Zhou et al., 2019a; Duan et al., 2019), FCOS (Tian et al., 2019) and others (Li et al., 2019; Lu et al., 2019; Zhu et al., 2019; Kong et al., 2019; Zhu et al., 2020; Law et al., 2020; Huang et al., 2015; Yu et al., 2016; Zhang et al., 2020) develop the anchor-free technical approach.

DETR and other end-to-end detectors. (Carion et al., 2020b) designed the first transformer-based (Vaswani et al., 2017) framework for end-to-end detection, namely DEtection TRansformer (DETR). DETR is also an anchor-free method without any other post-processing tricks. Compared with Faster-RCNN (Ren et al., 2017b), DETR takes much more time for training (500 epochs). Due to the slow convergence of DETR, a series of methods attribute the issue to different aspects of DETR. (Sun et al., 2020) enhanced DETR with traditional encoder-decoder architecture and proposed a new DETR with an encoder only. (Dai et al., 2021a) introduced a dynamic decoder based on RoI to optimize the regions of interest, which designed a coarse-to-fine manner and reduced the difficulties of learning. There are also some works to improve the decoder queries. (Zhu et al., 2021) boosted the attention mechanism of DETR to get more focused on sampling points around the reference points. (Meng et al., 2021) disentangled the queries to the contextual and positional aspects and optimize the queries with more specific spatial positions in the cross-attention formulation. (Yao et al., 2021) applied the top-K recommendation to this field, which utilized a Region Proposal Network to generate these more likely anchor points via ranking. (Li et al., 2022a) proposed a query-denoising method. (Li et al., 2022b) explored plain backbone for object detection.

However, neither Anchor-based nor Anchor-free detectors can not get rid of human-craft algorithms. The end-to-end detectors utilized the learned features derived from the representation space to match the bounding boxes in the pixel space. These DETR-like methods all suffer from two aspects: 1) The number of queries in the representation space is so far from the number of image pixels. 2) The uncertainty and redundancy of the Hungarian Matching limit the convergence of end-to-end detectors. Therefore, it is an urgent need to guide the end-to-end detector learning representation in the pixel space and break through the bottlenecks from the even widely-adopted Hungarian Matching.

3 How Hungarian Matching limits the Detectors?

3.1 A Brief Review of DETR

Given the visual features $\boldsymbol{x} \in \mathbb{R}^{C' \times H' \times W'}$ extracted by a CNN backbone (He et al., 2016) from an input image $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$, DETR (Carion et al., 2020b) employs the Transformer encoder-decoder network to turn the visual features to the object-relevant queries. The decoded object

queries are fed into the feed-ford network (FFN) as the detection head to regress the normalized center coordinates $\boldsymbol{b} \in [0,1]^4$ of the bounding box $\boldsymbol{b} = \{b_x, b_y, b_w, b_h\}$ and projected by the linear layer for classification. For each DETR-like detector, query and key are both derived from representation space. More specifically, the self-attention mechanism in the encoder of DETR requires the complexity $\mathcal{O}(H'^2W'^2C')$. And the decoder attention mechanism also takes the complexity of $\mathcal{O}(2NC'^2 + N^2C\prime)$, where N denotes the number of object queries and keys.

Hungarian Matching. Given a bipartite graph $\mathcal{G}(V, E)$, we denote a match as $\mathcal{M} = (V_M, E_M)$. V can be split as $V = \{V_A, V_{\bar{A}}\}$. The edge $e_{ij} \in E$ connects the vertexes v_i, v_j. The Hungarian matching algorithm builds the maximum-weight matchings. And its complexity is $\mathcal{O}(|V|^3)$.

3.2 THE LIMITATIONS BETWEEN MATCHING AND OPTIMIZATION

Let y denote the objects of ground truth and predictions of the bounding box and class can be formulated as $\hat{y} = \{\hat{y}_i = (c_i, \boldsymbol{b}_i)\}_{i=1}^N$. The objective function for optimization can be formulated as $\mathcal{L}_{\text{match}} = \sum_{i=1}^N [-\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}\left(\boldsymbol{b}_i, \hat{\boldsymbol{b}}_{\sigma(i)}\right)]$, where $\hat{p}_{\sigma(i)}(c_i)$ denotes the probability of class c_i, $\varnothing$ denotes "non-object", and $\sigma(i)$ is the optimal assignment $\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_i^N \mathcal{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right)$. Accordingly, the designed architecture of DETR seems sound in each training iteration step. However, the comprehensive optimization process is not well-considered. We found that, for each end-to-end detector, the objective function can not be easily decoupled from the perspectives of matching and box regression. More specifically:

1) **Uncertainty**. Different assignments $\sigma(i), \sigma(j)$ can correspond to the similar box loss $\mathcal{L}_{\text{box}}(\boldsymbol{b}_i, \hat{\boldsymbol{b}_{\sigma(i)}}) = \mathcal{L}_{\text{box}}(\boldsymbol{b}_i, \hat{\boldsymbol{b}_{\sigma(j)}})$, which brings uncertainty to the matching process. In other words, if the statistical variable $q = \sum_{i=1}^N [\sum_{\sigma \in \mathfrak{S}_N} \mathbb{1}_{\{\sigma(i)=\sigma(j) | \mathcal{L}_{\text{box}}(\boldsymbol{b}_i, \hat{\boldsymbol{b}_{\sigma(i)}}) = \mathcal{L}_{\text{box}}(\boldsymbol{b}_i, \hat{\boldsymbol{b}_{\sigma(j)}})\}}]$ denotes the number of different assignments $\sigma(i)$, and the upper bound of the expectation $\mathbb{E}_q$ should be continuously lowered to reduce the uncertainty.

2) **Redundancy**. The matching costs just formulate the single-round objective, and the comprehensive matching costs are neither scheduled nor optimized in the whole training process, which brings redundancy. In other words, the low-efficiency assignments repeatedly consume the computational resources and the repetition should be also formulated as the abductive cost in the training process $\sum_{i=1}^N \sum_{n=1,\sigma \in \mathfrak{S}_N}^t - \mathbb{1}_{\left\{\boldsymbol{b}_{\sigma(i)}^{(\hat{n})} \neq \boldsymbol{b}_{\sigma(i)}^{(\hat{n+1})}\right\}} \mathcal{L}_{\text{box}}(\boldsymbol{b}_i, \hat{\boldsymbol{b}_{\sigma(i)}}) + \mathbb{1}_{\left\{\boldsymbol{b}_{\sigma(i)}^{(\hat{n})} = \boldsymbol{b}_{\sigma(i)}^{(\hat{n+1})}\right\}} \mathcal{L}_{\text{box}}(\boldsymbol{b}_i, \hat{\boldsymbol{b}_{\sigma(i)}})$, where t denotes the t-th training epoch.

Therefore, the improvements of the DETR-like method should focus on the two aspects. One is to enhance the representations derived from the pixel space, which will optimize the object queries and make the network tend to learn a synergistic representation with 2D positional priors and generalization. The other is to find comprehensively better assignments to break the low-efficiency repetition in the matching process, which will lead the detectors to a faster convergence with less redundancy and uncertainty. The balance between pixel and representation space can optimize the stability of the assignment. And better matching assignment can also guide the visual features with a more precise backward gradient. These two points mutually promote, develop, and contribute to building a robust end-to-end detector.

4 ABDUCTIVE DETR

4.1 REGION TOKENS

As analyzed above, the suffering problems can be attributed to that, the ambiguity of object queries leading to an uncertain assignment in the Hungarian Matching. And formulation loss in the pixel space according to the matching results in turn affect the learning representations. Therefore, unsatisfactory assignments result in a non-appropriate gradient to the optimization in parameter space. To alleviate this problem, we introduce the region tokens. Its basic idea is to concentrate on the global attributes of regions, such as the number and class of objects in a certain region. The labels of regions are explicitly definite and will not influence by prediction fluctuation.

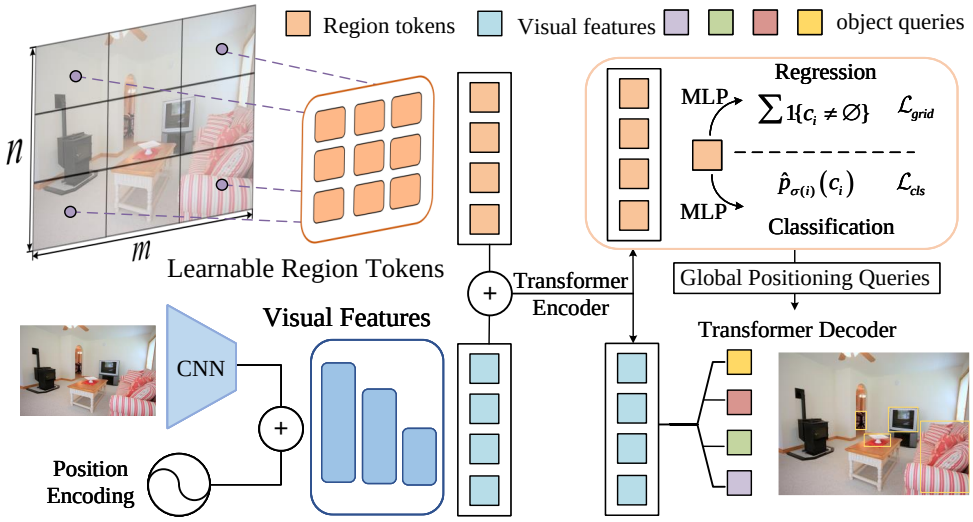

Figure 2: Abductive DETR: 1) We employ extra learnable embeddings and visual features as inputs of the transformer encoder. 2) Region tokens are fed into MLPs for classification and regression. Therefore, region attributes can be learned as prior knowledge to generate queries of the transformer decoder. Finally, decoded queries are higher-efficiently matched with boxes via abductive matching.

At the beginning of Vision Transformer development, the success of ViT (Dosovitskiy et al., 2020) proves that extra learnable embeddings can aggregate global class information from visual tokens. Therefore, we think that extra learnable embeddings encoded by the transformer network can benefit class prediction and visual token recognition. Furthermore, introduced embedding can efficiently focus on the target and extract useful semantics from the corresponding tokens. So, we think that several learnable region embeddings to aggregate the regional information can benefit predicting region attributes *i.e. class, number, etc*, which formulates intrinsically definite prior representations while independent from the matching assignments.

More specifically, we first split the image into $m \times n$ regions and then initialize $m \times n$ learnable embeddings for the regions. Then all the embeddings, as well as image tokens extracted from the backbone network (He et al., 2016), are sent into the Transformer Encoder to encode rich semantics. At the same time, region embeddings are made to learn regional semantics from regions in an explicit manner. Then, several Feed-Forward Networks (FFN) are employed for converting the regional information into region attributes, as shown in Fig. 2.

$$\begin{aligned} \mathbf{z}_0 &= \left[\mathbf{x}_{\mathrm{r}1}, \mathbf{x}_{\mathrm{r}2}, \cdots, \mathbf{x}_{\mathrm{r(mn)}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^{N_p}\mathbf{E}\right] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{\left(P^2 \cdot C\right) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \\ \mathbf{z}'_\ell &= \mathrm{MSA}\left(\mathrm{LN}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \ldots L = 6 \end{aligned} \tag{1}$$

Where $\mathbf{x}$ denotes visual features, $\mathbf{E}$ denotes embeddings, MSA denotes Multi-head Self-Attention (Vaswani et al., 2017), $(P \times P), C, N_p$ denote the resolution, channel and number of patches.

In this paper, we mainly think about two kinds of region attributes: number and class of objects. The number of objects will lead the network to perceive the spatial distribution of objects. With guidance, it is easier to focus on the occurrence of objects in a specific region. In addition, the other attribute is the multi-classification attribute for regions. ViT (Dosovitskiy et al., 2020) employs a class embedding to predict classification logits for the whole image. Regional classification based on the region tokens can also benefit the network concentrate on the existence of specific types of objects in the regions and facilitating the classification of targets. Due to regions may contain more than one type of object, therefore the classification task used in object detection should be a multi-classification.

Regression $\sum \mathbb{1}_{c_i \neq \varnothing}$. For each region token $\boldsymbol{T_r}$, we employ a simple MLP to regress the number of objects. The regression process is an approach under supervised training. The corresponding labels can be calculated with the center points for each object in the image. And we utilize the mean square

error to supervise the regression process, where $\mathbf{R}_i$ denotes the set of coordinates in the i-th region.

$$\hat{y_{\mathrm{n}}} = \mathrm{MLP}(\boldsymbol{T_r}), \quad y_n = \sum \mathbb{1}_{\{(p_x,p_y)\in\mathbf{R}_i\}}, \quad \mathcal{L}_{grid} = (y_n - \hat{y_{\mathrm{n}}})^2, \tag{2}$$

Classification $\hat{p}_{\sigma_i}(c_i)$. Similarly, we perform a multi-classification with the region tokens. We take the multi-classification as N_c independent binary classification tasks, where N_c is the number of classes. And we employ another MLP to predict the logits for the N_c binary classification tasks. It is also a supervised process, and the ground truth labels can be calculated by one-hot vectors according to the class labels of the object in the region. We utilize the Foal Loss (Lin et al., 2017) to supervise the classification process.

$$\hat{\boldsymbol{p}}_{\mathrm{c}} = \mathrm{MLP}(\boldsymbol{T_r}), \quad \boldsymbol{y_c} = [\mathbb{1}_{\{(p_x,p_y)\in\mathbf{R}_i\}}; \cdots;] \in \mathbb{R}^{1\times N_c}, \quad \mathcal{L}_{cls} = -\alpha_{\mathrm{t}}\left(1-\hat{p}_{\mathrm{t}}\right)^{\gamma}\log\left(\hat{p}_{\mathrm{t}}\right), \tag{3}$$

where t indicates the t-th class of N_c, and α, γ are two hyper-parameter which are set to be 0.25 and 2 in our experiments.

4.2 Dynamic Content-Aware Query Generator

Recently, DAB-DETR (Liu et al., 2022) proposes a dynamic-anchor approach for learning efficient representations in the transformer decoder. Nevertheless, the number of dynamic anchors is limited to the whole resolution of an input image. Meanwhile, the pre-defined dynamic anchor in the training set may not also be generic in the validation, these pre-defined anchors are sensitive to the position distribution in the dataset. And some corner cases like densely-distribution objects in a small region are challenging to these anchor pre-defined methods.

To address the limitations, we propose a dynamic content-aware query generator. It dynamically generates region-unique queries by assimilating region-intrinsic semantics derived from region tokens, which provide the number and class of objects as prior knowledge. To be specific, region attributes (§ 4.1) are fed into the query generator as input, and based on the region tokens, a query can concentrate on a certain region rather than glob the targets in the whole image. In addition, we can utilize the number of objects to manage the spatial distribution and density of queries. As we introduced, it is essential to efficiently assign queries for improving the generality of the model. The query generator can highly-efficiency assign queries according to the object density in the region. Finally, the classification results can also provide semantic prior, which leads the generated queries more focused on a certain type of object and reduces the uncertainty.

Formally, the generator takes the refined region attributes $\hat{y_{\mathrm{n}}}, \hat{p_{\mathrm{c}}}$ as input and outputs the global positioning queries for the decoder, one of which contain intrinsically definite attributes of a specific region. For the number of objects, the generator converts the regression results via integration. And position embeddings are generated according to the position of regions and the number of objects $\hat{y_{\mathrm{n}}}$ in them, where $\varnothing$ corresponds to no embedding. Position Embeddings $\mathbf{E}_p$ are generated and uniformly distributed in the region. Therefore, all the position embeddings in a region are concatenated with the multi-classification logits and sent into a feed-forward network FFN to generate the final queries:

$$\boldsymbol{q} = \mathrm{FFN}([\mathbf{E}_p; \hat{p_{\mathrm{c}}}]), \tag{4}$$

All the queries are sent into the decoder, as well as the memory produced by the encoder to generate the final predictions. The query generated will be optimized by the objective function (Eq. 7), without extra guidance.

4.3 Abductive Matching

As we introduced in the § 3.2, limitations of Hungarian Matching are about matching uncertainty and redundancy. Therefore, to further enhance the matching process, we propose abductive matching, which takes the comprehensive matching costs into consideration and couples the matching results with corresponding box IoU factors. So, in contrast to Hungarian Matching, abductive matching compensate for the two aspects as follows: 1) For each object query assigned with different boxes, abductive matching takes the upper bound of the expectation for different assignments, which further constrains the object queries over-disturbance. 2) For the history matching assignments, abductive matching considers the comprehensive matching costs. It helps to avoid over-repetition happening in the matching process.

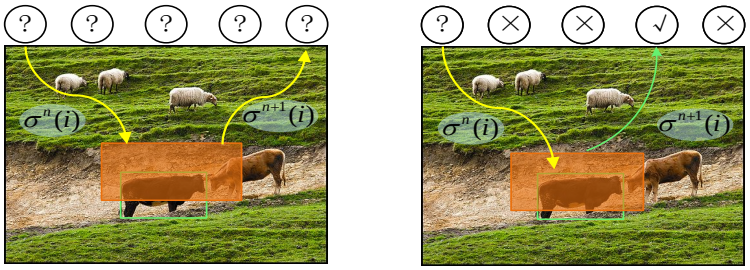

Figure 3: **Abductive Matching**. We denote the object queries as circles, and $\sigma^{(n)}(i)$ denotes the n-th assignment of the i-th object query. Intuitively, abductive matching enhances the matching process by introducing the box uncertainty to the matching cost and reducing the query redundancy.

Uncertainty Alleviation. We utilize uncertainty to describe the variations of assignments. Considering that in the bipartite graph, the better assignments it is, the fewer variations of the assignments would be. And the optimization for the stability of the matching can be easily transferred to lower the upper bound of the variations. Thus, we formulate the uncertainty by mean variances of the whole assignment. Formally, for each assignment σ_i, matching provides matched boxes $\boldsymbol{b}$. We utilize the center coordinates of the box to discriminate between different boxes. In this way, we can easily formulate the variances of each object query in each forward step as $\sum_{\sigma\in\mathfrak{S}_N} \mathbb{1}_{\{\hat{\boldsymbol{b}}_{\sigma(i)}=\hat{\boldsymbol{b}}_{\sigma(j)}|\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(i)}})=\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(j)}})\}}$. And the uncertainty can be formulated as the mean-variance of all object queries $\frac{1}{N}\sum_{i=1}^{N}[\sum_{\sigma\in\mathfrak{S}_N} \mathbb{1}_{\{\hat{\boldsymbol{b}}_{\sigma(i)}=\hat{\boldsymbol{b}}_{\sigma(j)}|\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(i)}})=\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(j)}})\}}]$. To alleviate the uncertainty, we can lower the expectation $\mathbb{E}_q$ for the upper bound of the mean-variance as:

$$\mathcal{L}_{\text{uncertainty}}(\hat{\boldsymbol{b}}^{(n)},\hat{\boldsymbol{b}}^{(n-1)}) = \mathbb{E}(\frac{1}{N}\sum_{i=1}^{N}[\sum_{\sigma\in\mathfrak{S}_N} \mathbb{1}_{\{\hat{\boldsymbol{b}}^{(n)}_{\sigma(i)}=\hat{\boldsymbol{b}}^{(n-1)}_{\sigma(i)}|\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(i)}})=\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(j)}})\}}]), n\geq 2 \quad (5)$$

Redundancy Alleviation. Different from the uncertainty, we utilize redundancy to describe the fault repetition in the comprehensive process. Thus, redundancy alleviation aims to reduce the circumstances of over-repeated bad assignments. Formally, for each bounding box $\boldsymbol{b}$, the IoU loss function can precisely describe the quality of object queries. To reduce the repetition of bad assignments, we deduplicate the bad object queries by the IoU description $\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(i)}})$. So that, the better object queries are encouraged to meet the matching box and the worse object queries are got deduplicated by $\mathbb{1}_{\{\text{IoU}(\boldsymbol{b}_{\sigma(i)})\geq\text{IoU}(\boldsymbol{b}_{\sigma(j)})\}}\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(i)}}) - \mathbb{1}_{\{\text{IoU}(\boldsymbol{b}_{\sigma(i)})\geq\text{IoU}(\boldsymbol{b}_{\sigma(j)})\}}\mathcal{L}_{\text{box}}(\boldsymbol{b}_j,\hat{\boldsymbol{b}_{\sigma(j)}})$. For the whole assignments in the n-th step, the redundancy can be optimized by $\sum_{\sigma_i,\sigma_j\in\mathfrak{S}} \mathbb{1}_{\{\text{IoU}(\boldsymbol{b}_{\sigma(i)})\geq\text{IoU}(\boldsymbol{b}_{\sigma(j)})\}}\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(i)}}) - \mathbb{1}_{\{\text{IoU}(\boldsymbol{b}_{\sigma(i)})\geq\text{IoU}(\boldsymbol{b}_{\sigma(j)})\}}\mathcal{L}_{\text{box}}(\boldsymbol{b}_j,\hat{\boldsymbol{b}_{\sigma(j)}})$. Therefore, we can alleviate the redundancy as:

$$\begin{cases} \mathcal{L}'_{\text{box}}(\boldsymbol{b}_{\sigma(i)},\boldsymbol{b}_{\sigma(j)}) = \sum_{i=1}^{N}\sum_{\sigma_i,\sigma_j\in\mathfrak{S}} \mathbb{1}_{\{\text{IoU}(\boldsymbol{b}_{\sigma(i)})\geq\text{IoU}(\boldsymbol{b}_{\sigma(j)})\}}\mathcal{L}_{\text{box}}(\boldsymbol{b}_i,\hat{\boldsymbol{b}_{\sigma(i)}}) \\ \qquad\qquad - \mathbb{1}_{\{\text{IoU}(\boldsymbol{b}_{\sigma(i)})\geq\text{IoU}(\boldsymbol{b}_{\sigma(j)})\}}\mathcal{L}_{\text{box}}(\boldsymbol{b}_j,\hat{\boldsymbol{b}_{\sigma(j)}}) \\ \mathcal{L}_{\text{redundancy}}(\hat{\boldsymbol{b}}^{(n)}) = \sum_{n=1}^{t}\mathcal{L}'_{\text{box}}(\boldsymbol{b}^{(n)}_{\sigma(i)},\boldsymbol{b}^{(n-1)}_{\sigma(j)}), n\geq 2 \end{cases} \quad (6)$$

Objective Function. Therefore, we can easily obtain the objective function of abductive matching. It further develops the Hungarian matching on the matching costs and history-cases tracking, which effectively reduces the uncertainty and redundancy of the DETR-like methods.

$$\begin{cases} \mathcal{L}_{\text{match}} = \sum_{i=1}^{N}[-\mathbb{1}_{\{c_i\neq\varnothing\}}\hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i\neq\varnothing\}}\mathcal{L}_{\text{box}}\left(\boldsymbol{b}_i,\hat{\boldsymbol{b}}_{\sigma(i)}\right)] & n=1 \\ \mathcal{L}_{\text{match}} = \underbrace{\sum_{i=1}^{N}[-\mathbb{1}_{\{c_i\neq\varnothing\}}\hat{p}_{\sigma(i)}(c_i)]}_{\text{Class prediction}} + \underbrace{\mathcal{L}_{\text{redundancy}}}_{\text{Abductive Costs}} + \underbrace{\mathcal{L}_{\text{uncertainty}}}_{\text{Box Regression}} & n\geq 2 \\ \hat{\sigma} = \arg\min_{\sigma\in\mathfrak{S}_N}\mathcal{L}_{\text{match}}\left(y_i,\hat{y}_{\sigma(i)}\right) \end{cases} \quad (7)$$

5 EXPERIMENT

Dataset. MSCOCO 2017 dataset (Lin et al., 2014a) is widely-adopted in object detection. Abductive-DETR is trained on the train set and evaluated on the val set and test-dev set.

Implementation Details. We implement Abductive DETR with 8 NVIDIA Tesla V100 GPUs. We utilize ResNet-50, ResNet-101 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009) dataset as backbones for extracting visual feature. We also report the experimental results with DCN (Dai et al., 2017) for better visual feature extraction, and we label the experiments with a suffix of "DC". In terms of the training process, we follow the strategy of DETR (Carion et al., 2020b), and the settings of hyper-parameters in our experiments are the same. Different from existing methods employing quantities of queries, the number of queries are dynamically set (§ 4.2) according to Region Tokens $N \in [0, 100)$. The initial learning rate is $2 \times 10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$, and the weight decay is 1×10^{-4}. And we provide a Pytorch-like pseudo code (Algorithm 1) for a better understanding of Abductive DETR.

Algorithm 1 Abductive DETR, PyTorch-like

```
def train(img_feat):
#Step 1: Region Tokens
      inputs = torch.cat([img_feat,
          global_token])
      output = self.encoder(inputs)
      memory = output[:, :-global_token_num]
      global_memory = output[:, -
          global_token_num: ]
      grid_reg = self.mlp_grid(global_memory)
      global_cls = self.mlp_global(
          global_memory)

#Step 2: Query Generation
      query = self.Query_Gen(grid_reg,
          global_cls)
      query_token = self.decoder(memory,
          query)
      query_cls = self.cls(query_token)
      query_box = self.mlp_box(query_token)

#Step 3: Loss Computation
      L_grid = MSE_Loss(grid_reg, target_grid)

      L_cls = FocalLoss(global_cls,
          target_cls)
      L_abd = Abd_Loss(query_cls, query_box,
          target)
      Loss = alpha * L_grid + beta * L_cls+
          gamma * L_abd

return Loss
```

5.1 ABLATION STUDY

To further elaborate on the effectiveness of components of Abductive DETR, we conduct the ablation study. The 2nd, 3rd, and 4th rows are in the Table. 1 validate the effectiveness of Region Tokens, Query Generator, and Abductive Match, respectively. Meanwhile, these components can cooperate to optimize different aspects of the detectors, which are shown in the 5th, 6th, and 7th rows. Finally, the best experimental results (AP 44.5%) can be achieved by employing all components. Compared with DETR, Abductive DETR can outperform DETR by +2.5% AP improvement.

Method			AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Region Token 4.1	Query Generator 4.2	Abductive Match 4.3	(%)	(%)	(%)	(%)	(%)	(%)
-	-	-	42.0	62.4	44.2	20.5	45.8	61.1
✓	-	-	43.4	63.5	44.8	23.9	46.6	61.6
-	✓	-	42.2	63.0	44.4	21.4	45.3	60.8
-	-	✓	43.2	63.2	44.6	21.7	46.0	62.1
✓	✓	-	43.5	63.7	45.0	21.8	46.3	62.3
✓	-	✓	43.8	63.9	45.2	22.2	46.9	62.6
-	✓	✓	43.9	64.8	46.3	23.2	47.9	63.7
✓	✓	✓	44.5	65.2	46.9	23.6	48.6	63.9

Table 1: Ablation Study on Abductive DETR with a ResNet-50 backbone.

5.2 MAIN RESULTS

Abductive DETR obtains an obvious improvement among the DETR-like methods. First of all, compared with other variants employing 300 queries, our method employs less than 100. As we analyzed, the complexity of Hungarian is $\mathcal{O}(|V|^3)$. So that Abductive DETR is at least **27× faster** than other variants (Dai et al., 2021b; Liu et al., 2022; Li et al., 2022a; Meng et al., 2021) in the matching process even the performance is better.

Model	Epochs	Queries	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
DETR-R50 (Carion et al., 2020b)	500	100	42.0	62.4	44.2	20.5	45.8	61.1	86	41M
Deformable DETR-R50 (Zhu et al., 2021)	50	300	43.8	62.6	47.7	26.4	47.1	58.0	173	40M
SMCA-R50 (Gao et al., 2021)	50	300	43.7	63.6	47.2	24.2	47.0	60.4	152	40M
DAB-DETR-R50 (Liu et al., 2022)	50	300	42.2	63.1	44.7	21.5	45.7	60.3	94	44M
DN-DETR-R50 (Li et al., 2022a)	50	300	44.1	64.4	46.7	22.9	48.0	63.4	94	44M
SAM-DETR-R50 (Zhang et al., 2022)	50	300	41.8	63.2	43.9	22.1	45.9	60.9	100	58M
Abductive DETR-R50 (Ours)	50	**<100**	**44.5**	65.2	46.9	23.6	48.6	63.9	83	42M
DETR-R101 Carion et al. (2020b)	500	100	43.5	63.8	46.4	21.9	48.0	61.8	152	60M
Faster RCNN-FPN-R101 Ren et al. (2017b)	108	–	44.0	63.9	47.8	27.2	48.1	56.0	246	60M
Anchor DETR-R101 Wang et al. (2021)	50	300	43.5	64.3	46.6	23.2	47.7	61.4	–	58M
Conditional DETR-R101 Meng et al. (2021)	50	300	42.8	63.7	46.0	21.7	46.6	60.9	156	63M
DAB-DETR-R101 Liu et al. (2022)	50	300	43.5	63.9	46.6	23.6	47.3	61.5	174	63M
DN-DETR-R101 (Li et al., 2022a)	50	300	45.2	65.5	48.3	24.1	49.1	65.1	174	63M
Abductive DETR-R101 (Ours)	50	**<100**	**45.6**	66.1	48.9	24.5	49.8	65.4	159	61M
DETR-DC5-R50 Carion et al. (2020b)	500	100	43.3	63.1	45.9	22.5	47.3	61.1	187	41M
Anchor DETR-DC5-R50 Wang et al. (2021)	50	300	44.2	64.7	47.5	24.7	48.2	60.6	151	39M
Conditional DETR-DC5-R50 Meng et al. (2021)	50	300	43.8	64.4	46.7	24.0	47.6	60.7	195	44M
DAB-DETR-DC5-R50 Liu et al. (2022)	50	300	44.5	65.1	47.7	25.3	48.2	62.3	202	44M
DN-DETR-DC5-R50 (Li et al., 2022a)	50	300	46.3	66.4	49.7	26.7	50.0	64.3	202	44M
Abductive DETR-DC5-R50 (Ours)	50	**<100**	**46.8**	67.1	50.3	27.5	50.3	64.6	194	42M
DETR-DC5-R101 Carion et al. (2020b)	500	100	44.9	64.7	47.7	23.7	49.5	62.3	253	60M
Anchor DETR-R101 Wang et al. (2021)	50	300	45.1	65.7	48.8	25.8	49.4	61.6	–	58M
Conditional DETR-DC5-R101 Meng et al. (2021)	50	300	45.0	65.5	48.4	26.1	48.9	62.8	262	63M
DAB-DETR-DC5-R101 Liu et al. (2022)	50	300	45.8	65.9	49.3	27.0	49.8	63.8	282	63M
DN-DETR-DC5-R101 (Li et al., 2022a)	50	300	47.3	67.5	50.8	28.6	51.5	65.0	282	63M
Abductive DETR-DC5-R101 (Ours)	50	**<100**	**47.9**	68.0	52.9	31.9	53.1	65.4	269	61M

Table 2: Comparison results between Abductive DETR and other advanced models. We report the experimental results when Abductive DETR with ResNet-50(R50), and ResNet-101(R101), respectively. Where "DC" denotes the DCN Dai et al. (2017).

Under the same setting with ResNet-50 backbones, Abductive DETR can outperform Deformable DETR (Zhu et al., 2021) by $+0.7$ AP, DN-DETR (Li et al., 2022a) by $+0.4$ AP, DAB-DETR (Liu et al., 2022) by $+2.3$ AP, and SAM-DETR (Zhang et al., 2022) by $+2.7$ AP.

Under the same setting with the DC5-R50 backbone, Abductive DETR can outperform DN-DETR (Li et al., 2022a) by **+0.5** AP, Anchor-DETR (Liu et al., 2022) by $+2.6$ AP, Conditional DETR (Liu et al., 2022) by $+3.0$ AP.

When a larger visual backbone of ResNet-101, Abductive DETR can still outperform DN-DETR (Li et al., 2022a) by **+0.4** AP, Anchor-DETR (Liu et al., 2022) by $+2.1$ AP, Conditional DETR (Liu et al., 2022) by $+2.8$ AP. In the meantime, Abductive DETR also outperforms Faster RCNN-FPN (Ren et al., 2017a) by $+1.6$ AP within training 50 epochs rather than 108 epochs. Meanwhile, Abductive DETR can achieve the better performance with DC-ResNet-101 backbone, which is better than Anchor DETR (Wang et al., 2021) by $+2.8$ AP, DN-DETR (Li et al., 2022a) by $+0.6$ AP, Conditional DETR (Meng et al., 2021) by $+2.9$ AP.

Therefore, we conclude that Abductive DETR is suitable to work as a new strong baseline for object detection.

6 CONCLUSION

In this paper, we have analyzed the impact of Hungarian matching on DETR training and proposed a new framework to address the limitations. It learns object queries in the representation space with global positioning in the pixel space and matches object queries in the pixel space with the abductive awareness from the representation space. We compare our framework with other advanced detectors, and experimental results are supportive to prove that Abductive DETR is more advantageous in learning object-relevant representations from pixel space, which takes fewer object queries, less training time, and achieves a better performance with different backbone networks. This study attempts to prove that the relationship between representation and pixel spaces can not be easily linked by Hungarian Matching. And the abductive approach seems important to build a more stable connection between representation and pixel spaces.

REFERENCES

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, 2018.

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

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020b.

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *NeurIPS*, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2988–2997, October 2021a.

Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2988–2997, 2021b.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019.

Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *CVPR*, 2014.

Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. *arXiv preprint arXiv:2101.07448*, 2021.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

Ross B. Girshick. Fast R-CNN. In *ICCV*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *CoRR*, abs/1509.04874, 2015. URL `http://arxiv.org/abs/1509.04874`.

Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Foveabox: Beyond anchor-based object detector. *CoRR*, abs/1904.03797, 2019.

Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018.

Hei Law, Yun Teng, Olga Russakovsky, and Jia Deng. Cornernet-lite: Efficient keypoint based object detection. In *BMVC*. BMVA Press, 2020. URL https://www.bmvc2020-conference.com/assets/papers/0016.pdf.

Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M. Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13619–13627, June 2022a.

Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *ICCV*, pp. 6054–6063, 2019.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022b.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014a.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014b.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *TPAMI*, 2020.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=oMI9PjOb9Jl.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016.

Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid R-CNN. In *CVPR*, 2019.

Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. *arXiv preprint arXiv:2108.06152*, 2021.

Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: towards balanced learning for object detection. In *CVPR*, 2019.

Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016.

Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, 2017.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018a.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018b.

Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017a.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017b.

Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *CVPR*, 2020.

Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. *arXiv preprint arXiv:2011.10881*, 2020.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. In *ICCV*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021.

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.

Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas S. Huang. Unitbox: An advanced object detection network. In *MM*, 2016. doi: 10.1145/2964284.2967274. URL https://doi.org/10.1145/2964284.2967274.

Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating DETR convergence via semantic-aligned matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 949–958, 2022.

Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019a. URL http://arxiv.org/abs/1904.07850.

Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019b.

Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, 2019.

Chenchen Zhu, Fangyi Chen, Zhiqiang Shen, and Marios Savvides. Soft anchor-point object detection. In *ECCV*, 2020. doi: 10.1007/978-3-030-58545-7_6. URL https://doi.org/10.1007/978-3-030-58545-7_6.

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: The Ninth International Conference on Learning Representations*, 2021.