SAP-DETR: Bridging the Gap between Salient Points and Queries-Based Transformer Detector for Fast Model Convergency

Yang Liu[1,3] * Yao Zhang[1,3] * Yixin Wang[2] * Yang Zhang[4] Jiang Tian[4]
Zhongchao Shi[4] Jianping Fan[4] Zhiqiang He[1,3,5] †
[1]Institute of Computing Technology (ICT), Chinese Academy of Sciences [2]Stanford University
[3]University of Chinese Academy of Sciences [4]AI Lab, Lenovo Research [5]Lenovo Ltd.
{liuyang20c,zhangyao215}@mails.ucas.ac.cn yxinwang@stanford.edu
{zhangyang20,tianjiang1,shizc2,jfan1,hezq}@lenovo.com

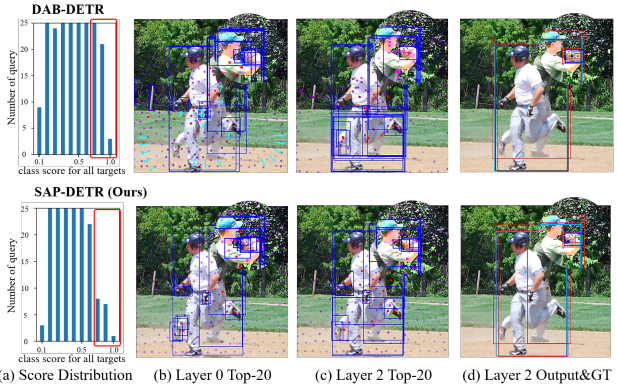
Abstract

Recently, the dominant DETR-based approaches apply central-concept spatial prior to accelerating Transformer detector convergency. These methods gradually refine the reference points to the center of target objects and imbue object queries with the updated central reference information for spatially conditional attention. However, centralizing reference points may severely deteriorate queries' saliency and confuse detectors due to the indiscriminative spatial prior. To bridge the gap between the reference points of salient queries and Transformer detectors, we propose ***SA****lient* ***P****oint-based* ***DETR*** *(****SAP-DETR****) by treating object detection as a transformation from salient points to instance objects. Concretely, we explicitly initialize a query-specific reference point for each object query, gradually aggregate them into an instance object, and then predict the distance from each side of the bounding box to these points. By rapidly attending to query-specific reference regions and the conditional box edges, SAP-DETR can effectively bridge the gap between the salient point and the query-based Transformer detector with a significant convergency speed. Experimentally, SAP-DETR achieves 1.4× convergency speed with competitive performance and stably promotes the SoTA approaches by ∼1.0 AP. Based on ResNet-DC-101, SAP-DETR achieves 46.9 AP. The code will be released at https://github.com/liuyang-ict/SAP-DETR.*

1. Introduction

Object detection is a fundamental task in computer vision, whose target is to recognize and localize each object from input images. In the last decade, various detectors [6, 11, 14, 18, 20, 22] based on Convolutional Neural Networks (CNNs), have received widespread attention and made significant progress. Recently, Carion *et al.* [2] proposed a new end-to-end paradigm for object detection based on the Transformer [24], called DEtection TRansformer (DETR), which treats object detection as a problem of set prediction. In DETR, a set of learnable positional encodings, namely object queries, are employed to aggregate instance features from the context image in Transformer Decoder. The predictions of queries are finally assigned to the ground truth via bipartite matching to achieve end-to-end detection.

Despite the promising results of DETR, its application is largely limited by considerably longer training time compared to conventional CNNs. To address this problem, many



(a) Score Distribution (b) Layer 0 Top-20 (c) Layer 2 Top-20 (d) Layer 2 Output&GT

Figure 1. Comparison of SAP-DETR and DAB-DETR under 3-layer decoder model and 36-epoch training scheme. (a) Statistics of the query count in different classification score intervals. (b) and (c) Visualize the distribution of all reference points (pink) and highlight the top-20 classification score queries with their bounding boxes (blue) and reference points (red) in different decoder layers. (d) Visualize the outputs of positive queries and ground truth (red) during the training process, each query has a representative color for its reference point and bounding box. Best viewed in color.

*This work was done when working as an intern at AI Lab, Lenovo Research, Beijing, China.
†Corresponding author.

variants attempted to take a close look at query paradigm and introduced various spatial priors for model convergency and efficacy. According to the type of spatial prior, they can be categorized into implicit and explicit methods. The implicit ones [5, 16, 31] attempt to decouple a *reference point* from the object query and make use of this spatial prior to attend to the image context features efficiently. The current state-of-the-arts (SoTAs) are dominated by the explicit ones [13, 25], which suggest to instantiate a position with spatial prior for each query, i.e., explicit reference coordinates with a center point or an anchor box. These reference coordinates serve as helpful priors and enable the queries to focus on their expected regions easily. For instance, Anchor DETR [25] introduced an anchor concept (center point with different box size patterns) to formulate the query position and directly regressed the central offsets of the bounding boxes. DAB-DETR [13] further stretched the center point to a 4D anchor box concept $[cx, cy, w, h]$ to refine proposal bonding boxes in a cascaded manner. However, instantiating the query location as a target center may severely degrade the classification accuracy and convergency speed. As illustrated in Fig. 1, there exist many plausible queries [19] with high-quality classification scores (Fig. 1(a) within red box) and box Intersection over Union (IoU, see the redundant blue boxes in Fig. 1(b) and (c)), which only brings a slight improvement on precision rate but inevitably confuses the detector on the positive query assignments when training with bipartite matching strategy. This is because the plausible predictions are considered in negative classification loss, which severely decelerates the model convergency. As shown in Fig. 1(b) and (c), the predefined reference point of the positive query may not be the nearest one to the center of the ground truth bounding box, and the reference points tend to be centralized or marginalized (cyan arrows in Fig. 1(b)), hence losing the spatial specificity. With further insight into the one-to-one label assignment during the training process, we find that the query, whose reference point is closest to the center point, also has a high-quality IoU, but it still exists a disparity with the positive query in the classification confidence. Therefore, we argue that such a centralized spatial prior may cause degeneration of target consistency in both classification and localization tasks, which leads to inconsistent predictions.

Furthermore, the mentioned central point-based variants also have difficulties in detecting occluded objects. For example, Fig. 1(d) shows that DAB-DETR detects the left baseman twice, while the query point in SAP-DETR is not necessarily the center point, so the query point of the bounding box for the occluded baseman is from a pixel on the occluded baseman on the top right area instead of from the left baseman. One solution for center-based method [25] is to predefine different receptive fields (similar to the scaling anchor box in YOLO [17]) for the position of each query. However, increasing the diversity of the receptive fields for each position query is unsuitable for non-overlapped targets, as it still generates massive indistinguishable predictions for one position as same as other center-based models.

To bridge these gaps, in this paper, we present a novel framework for Transformer detector, called SAlient Point-based DETR (SAP-DETR), which treats object detection as a transformation from salient points to instance objects. Instead of regressing the reference point to the target center, we define *the reference point belonging to one positive query as a salient point*, keep this query-specific spatial prior with a scaling amplitude, and then gradually update them to an instance object by predicting the distance from each side of the bounding box. Specifically, we tile the mesh-grid referenced points and initialize their center/corner as the query-specific reference point. To disentangle the reference sparsity as well as stabilize the training process, a movable strategy with scaling amplitude is applied for reference point adjustment, which prompts queries to consider their reference grid as the salient region to perform image context attention. By localizing each side of the bounding box layer by layer, such query-specific spatial prior enables compensation for the over-smooth/inadequacy problem during center-based detection, thereby vastly promoting model convergency speed. Inspired by [5, 13, 16], we also take advantage of both Gaussian spatial prior and conditional cross-attention mechanism, and then a salient point enhanced cross-attention mechanism is developed to distinguish the salient region and other conditional extreme regions from the context image features.

We bridge the gap between salient points and query-based Transformer detector by speedily attending to the query-specific region and other conditional regions. The extensive experiments have shown that SAP-DETR achieves superior convergency speed and performance. To the best of our knowledge, this is the first work to introduce the salient point based regression into end-to-end query-based Transformer detectors. Our contributions can be summarized as follows.

1) We introduce the salient point concept into query-based Transformer detectors by assigning query-specific reference points to object queries. Unlike center-based methods, we restrict the reference location and *define the point of the positive query as the salient one*, hence enlarging the discrepancy of query as well as reducing the redundant predictions (see Fig. 1). Thanks to the efficacy of the query-specific prior, our SAP-DETR accelerates the convergency speed greatly, achieving competitive performance with 30% fewer training epochs. The proposed movable strategy further boosts SAP-DETR to a new SoTA performance.

2) We devise a point-enhanced cross-attention mechanism to imbue query with spatial prior based on both reference point and box sides for final specific region attention.

3) Evaluation over COCO dataset has demonstrated that SAP-DETR achieves superior convergency speed and detection accuracy. Under the same training settings, SAP-DETR

outperforms the SoTA approaches with a large margin.

2. Related Work

Anchor-Free Object Detectors. Classical anchor-free object detectors can be grouped into center-based and keypoint-based approaches. The center-based approaches aim to localize the target objects based on the central locations [11] or predefined ROI [22]. For example, FCOS [22] treated all points within the bounding box as positive ones to predict their distances from each side ($[\ell, t, r, b]$), and a centerness score was then considered to prohibit the low-quality prediction whose point is located near the border. Compared with FCOS, we also restrict the candidate queries within the bounding box but treat only one as positive to perform end-to-end object detection via an inner matching cost.

The target of keypoint-based approaches is to localize the specific object locations and assign them to the predefined keypoints of the object for box localized training. For instance, diagonal corner points were considered in CornerNet [8], center point was further grouped into CenterNet [29], and ExtremeNet [30] added some conjectural extreme points for object localization. These works showed an impressive performance, but the complicated keypoint matching may limit their upper bound. Our SAP-DETR takes the advantage of salient point regression to focus on the distinct regions without complicated point-based supervision.

Query-Based Transformer Detectors. DETR [2] pioneered a new paradigm of Transformer detector for end-to-end object detection without any post-processing [1]. In DETR, a new representation, namely object query, aggregates the instance features and then yields a detection result for each instance object [15]. Following DETR, many votarists put efforts on the optimization of convergency and accuracy.

Sun *et al.* [21] revealed that the main reason for slow convergency of DETR is attributed to the Transformer decoder, and they considered an encoder-only structure to alleviate such a problem. For in-depth understanding of the object query, one way is to generate a series of implicit spatial priors from queries to guide feature aggregation in cross-attention layers. SMCA [5] applied a Gaussian-like attention map to augment the query-concerned features spatially. The *reference point* concept was first introduced by Deformable DETR [31], where the sampling offsets are predicted by each reference point to perform deformable cross-attention. Following such a concept, Conditional DETR [16] reformulated the attention operation and rebuilt positional queries based on the reference points to facilitate extreme region discrimination. Another way is towards position-instantiation explicitly, where this position information enables to directly conduct positional query generation. Anchor DETR [25] utilized a predefined 2D anchor point $[cx, cy]$ to explicitly capitalize on the spatial prior during cross-attention and box regression. DAB-DETR [13] extended such a 2D concept to a 4D anchor box $[cx, cy, w, h]$ and refined it layer-by-layer.

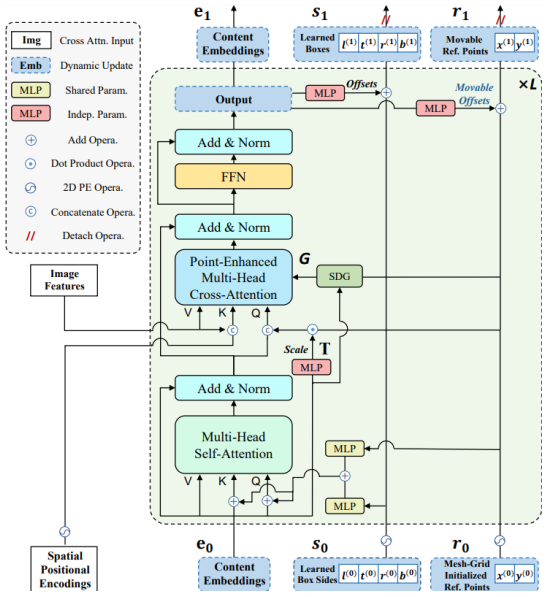


Figure 2. Illustration of SAP-DETR. Each object query in SAP-DETR is assigned to a specific grid region and initialized by the corner/center of the grid as its reference point. A learnable 4D coordinate represents the distance from the four sides of the box to the reference point. Both reference points and box sides are served as positional encodings added/concatenated to content embeddings. All embeddings are refined to predict target objects gradually.

The recent accelerating convergency methods are based on auxiliary queries for facilitating detector discrimination. DN-DETR [9] demonstrates the slow model convergency is mainly caused by the instability of bipartite matching, thus providing a denoising training to eliminate this issue. DINO [28] inherits this advance and further introduces negative queries to perform contrastive denoising. Group-DETR [4] proposes a group-wise one-to-many label assignment to match multiple positive object queries with more gradients for fast DETR convergency.

The most relevant approaches to ours are Point-DETR [3] and SAM-DETR [26, 27]. The former applied a point encoder for annotated point label infusion in teacher model, and the latter directly updated content embeddings by extracting salient points from image features for query-image semantic alignment. Unlike these concepts, we redefine the salient point from the perspective of the positive query's position and replace the center-concept prior with the query-specific position, thereby attending extreme regions, differentiating queries' saliency, and alleviating redundant predictions.

3. Method

We propose SAP-DETR to bridge the gap between salient points and query-based detectors. Following DETR, the extracted image features are fed into Transformer encoder

after adding positional encodings, and then re-aggregated by object queries in Transformer decoder. In pursuit of the query-specific prior, we dispense a movable strategy for each query based on a fixed grid region. The query, whose reference region overlaps with ground truth objects, is allowed to predict the relative offsets from four sides of the bounding box to the points. Given the query-specific reference point and the proposal box sides, we propose salient point enhanced cross-attention mechanism to imbue query with spatial prior, thereby attending to extreme regions effectively. Additionally, we discuss two common issues in DETR-like models and address them for further improvements.

3.1. Salient Points-Based Object Detection

Overview. Previous methods [13, 16, 25] normally decompose the object query into both content and position embeddings (queries), and form a center-based anchor point/box prior on the position ones. Unlike the central concept, we tile a fixed mesh-grid region, initialize their left-top corner as the reference points with 2D coordinate $\boldsymbol{r}=\{x, y\}\in[0,1]^2$, and instantiate a learnable 4D offset distance $\boldsymbol{s}=\{\ell, t, r, b\}\in[0,1]^4$ from the reference point to the sides of proposal bounding box for each object query. The object query can be referred as $\boldsymbol{q}=\{\mathbf{e}; \boldsymbol{r}, \boldsymbol{s}\}$, where $\mathbf{e}\in\mathbb{R}^d$ is the content embedding with d dimension. Instead of regressing the center, width, and height of a bounding box, we follow FCOS [22] and directly supervise the 4D offset from the four sides of a bounding box to the reference point. The final box prediction is formulated as $\hat{\boldsymbol{b}}=\{\hat{x}-\hat{\ell}, \hat{y}-\hat{t}, \hat{x}+\hat{r}, \hat{y}+\hat{b}\}$. Ideally, we here fix the reference point $\{\hat{x}, \hat{y}\}=\{x, y\}$ (the movable update strategy is introduced in the next subsection) and only update the 4D box side prediction layer by layer. The prediction for each decoder layer can be calculated by

$$\begin{aligned}\Delta\boldsymbol{s}_l &= \text{BoxHead}_l(\boldsymbol{s}_{l-1}, \mathbf{e}_{l-1}, \boldsymbol{r}_{l-1}),\\ \hat{\boldsymbol{s}}_l &= \sigma(\sigma^{-1}(\boldsymbol{s}_{l-1})+\Delta\boldsymbol{s}_l), \quad \boldsymbol{s}_l=\text{Detach}(\hat{\boldsymbol{s}}_l),\\ \boldsymbol{r}_l &= \hat{\boldsymbol{r}}_l=\boldsymbol{r}_{l-1}, \quad \hat{\boldsymbol{b}}_l=\{\hat{\boldsymbol{r}}_l-\hat{\boldsymbol{s}}_l[:2], \hat{\boldsymbol{r}}_l+\hat{\boldsymbol{s}}_l[2:]\},\end{aligned} \tag{1}$$

where σ and σ^{-1} are the sigmoid and inverse sigmoid operation, respectively. $\Delta\boldsymbol{s}_l$ denotes the side offset prediction. $\hat{\boldsymbol{s}}_l$, $\hat{\boldsymbol{r}}_l$, and $\hat{\boldsymbol{b}}_l$ are the predicted side distance, reference points, and box location from the l decoder layer, respectively. The BoxHead_l is the prediction head following the layer-l decoder, which is independent between different decoder layers in our settings. Detach operation follows DAB-DETR [13].

During the training process, each query is only allowed to predict the bounding boxes that overlap its reference region. We adapt this rule into the one-to-one bipartite matching process via an inner matching cost $\mathcal{L}_{\text{inner}}$. Given N queries $\mathbf{Q}=\{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_N\}$ and M ground truth objects $\mathbf{G}=\{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_M\}$, the $\mathcal{L}_{\text{inner}}(\boldsymbol{g}_i, \boldsymbol{q}_j)$ of each query-box pair is a step function to penalize the reference point $\boldsymbol{r}_j$ of $\boldsymbol{q}_j$ with value of k when $\boldsymbol{r}_j$ is outside the bounding box of $\boldsymbol{g}_i$. We denote $i\in[1, M]$ and $j\in[1, N]$ as the index of query and ground truth, respectively. k can be viewed as a penalty cost, and default to 10^5. The final permutation of the one-to-one label assignment is formulated as

$$\begin{aligned}\mathcal{L}_{\text{inner}}(\boldsymbol{g}_i, \boldsymbol{q}_j) &:= \mathbf{k}_{\boldsymbol{r}_j\notin\boldsymbol{g}_i},\\ \hat{\eta} &= \underset{\eta\in\mathfrak{Y}_N}{\text{argmin}}\sum_{i}^{N}\mathcal{L}_{\text{match}}+\mathcal{L}_{\text{inner}},\end{aligned} \tag{2}$$

where $\mathcal{L}_{\text{match}}$ is the original pair-wise matching cost consist of both classification and localization costs [2]. $\eta\in\mathfrak{Y}_N$ is a permutation of N elements for bipartite matching.

Movable Reference Point. Due to the sparseness of the fixed reference point, some small and slender objects may be indistinguishable when there is no reference point inside these objects. Despite the bipartite matching forcing each object to be assigned to one object query, the positive query, whose reference point is outside the assigned bounding box, is unable to accurately regress the distance from each side by a value between 0 and 1. One straightforward solution is to adjust the locations of reference points inside the ground truth bounding boxes to ensure that each object can be detected by an inner reference point. Similar to the aforementioned box refinement, we first perform a movable reference point design to dynamically update the reference points of each query layer by layer. However, such a full-image point regression inevitably expands the search space as vast variable determinations, causing the final reference point to be trapped in an unexpected corner of the bounding box. To reduce the training search spaces, we scale the offset amplitude of points within their specific grid regions, as illustrated in Fig. 3. Such an operation limits the range of offset values, and hence prevents a large searching space. It is implemented by applying the sigmoid activation σ and multiplying a scale factor $\boldsymbol{s}_{\text{grid}}$ whose value equals to the height and width of one grid. The update process of the reference points is formulated as

$$\begin{aligned}\Delta\boldsymbol{r}'_l &= \text{PointHead}_l(\boldsymbol{s}_{l-1}, \mathbf{e}_{l-1}, \boldsymbol{r}_{l-1}),\\ \Delta\boldsymbol{r}_l &= \sigma(\sigma^{-1}(\boldsymbol{r}_{l-1}-\boldsymbol{r}_0)+\Delta\boldsymbol{r}'_l),\\ \hat{\boldsymbol{r}}_l &= \boldsymbol{r}_0+\Delta\boldsymbol{r}_l\cdot\boldsymbol{s}_{\text{grid}}, \quad \boldsymbol{r}_l=\text{Detach}(\hat{\boldsymbol{r}}_l),\end{aligned} \tag{3}$$

where $\Delta\boldsymbol{r}'_l$ and $\Delta\boldsymbol{r}_l$ are the predicted offsets from $\boldsymbol{r}_l$ to both $\boldsymbol{r}_{l-1}$ and $\boldsymbol{r}_0$ before the sigmoid activation σ, respectively.

3.2. Salient Point Enhanced Cross-Attention

In cross-attention layers, existing center-based methods are limited to the attention on both center and sides of the ground truth bounding box, causing detector confusion among the queries with the same center and side attention. To this end, we expect the queries to focus on their specific regions based on the reference points, four box sides, and other conditional regions in different heads. Accordingly,

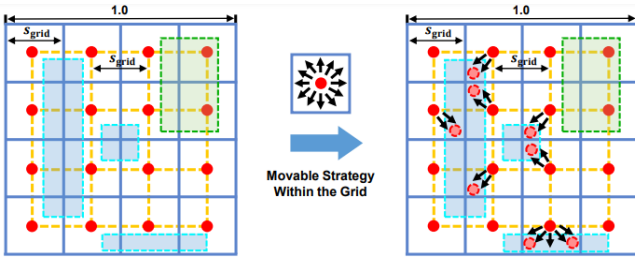


Figure 3. Movable reference point. The reference points are initialized by the center/corner points of the mesh-grid. Based on the inner loss, only the green dashed box can be predicted by the inner points when reference points are fixed. By moving the reference points within their grid, the blue dashed boxes can be detected accurately without extended searching space.

we consider an improved Gaussian [5] $\mathbf{G}$ and conditional attention [16] $\mathbf{A}_{\text{peca}}$ to enhance query specificity and spatially extreme region discrimination. The final attention map $\mathbf{A}_{\text{cross}}$ is the sum of the two attentions $\mathbf{A}_{\text{cross}} = \mathbf{G} + \mathbf{A}_{\text{peca}}$.

Side Directed Gaussian (SDG). Similar to the movable strategy, we enforce the predicted Gaussian attention to be inside the proposal bounding box to reduce the searching space. Given a reference point $\boldsymbol{r}$, the offset scales $\mathbf{o} \in [-1, 1]^2$ for H heads are produced by a simple MLP with a tanh activation, and then multiply to the two sides of the proposal bounding box for head-specific point offset generation, where the direction is guided by the sign of the offset scales. The head-specific points are generated by Algorithm 1. For each head, the Gaussian-like spatial weight map G_i effecting on each pixel (x, y) of context features is then formulated as

$$G_i(x, y) = \exp\left(-\frac{(x - c_{w,i})^2}{v_{w,i}^2} - \frac{(y - c_{h,i})^2}{v_{h,i}^2}\right). \quad (4)$$

Point Enhanced Cross-Attention (PECA). As aforementioned, the semantic class for the query is closely related to its referenced location in our SAP-DETR. To enhance the correlation between queries and their references, we concatenate the locations to the content queries after the sinusoidal positional encoding (PE). Take a close look at the conditional attention [16], we find that the linear positional embedding mostly focuses on one box side in each attention head. So we introduce a more straightforward attention mechanism, where the four side coordinates are concatenated and assigned to the corresponding head for side attention. The process of PECA is formulated as

$$\begin{aligned}\mathbf{A}_{\text{peca}} = &\mathbf{e}_q \mathbf{e}_k^\top + \mathbf{T}\text{PE}(\boldsymbol{r}_q)\text{PE}(\boldsymbol{r}_k)^\top \\ &+ \mathbf{T}g(\text{PE}(\boldsymbol{r}_q - \{\ell, t\}, \boldsymbol{r}_q + \{r, b\}))\text{PE}(\boldsymbol{r}_k)^\top,\end{aligned} \quad (5)$$

where g is a linear layer mapping 4D-PE into 2D to keep channel dimension consistency. $\mathbf{T}$ [16] is a scaling transformation matrix. More details of $\mathbf{T}$ and comparison with conditional cross-attention are available in Appendix C.

Algorithm 1 Side Directed Gaussian

Input: Content embedding $\mathbf{e}$, reference point $\boldsymbol{r}$ and box $\boldsymbol{s}$.
Output: Head-specific points $\mathbf{c} = \{(c_{w,i}, c_{h,i}) | i \in H\}$ and head-specific attention $\mathbf{v} = \{(v_{w,i}, v_{h,i}) | i \in H\}$.

1: Predict offset scale and attention scale based on content embedding, $\mathbf{o} = \tanh(\text{MLP}(\mathbf{e}))$, $\mathbf{v} = \text{MLP}(\mathbf{e})$;
2: **for** $h \leftarrow 1 \in H$ **do**
3: Select the index of direction guided by the sign of offset scale, $\{a, b\} = \text{sgn}(\mathbf{o}_i) + \{1, 2\}, \quad a, b \in \{0, 1, 2, 3\}$;
4: According to the index of direction, predict head-specific point, $\mathbf{c}_i = \mathbf{o}_i \cdot \boldsymbol{s}[a, b] + \boldsymbol{r}$;
5: **end for**
6: **return** $\mathbf{c}_i, \mathbf{v}_i, \ \forall i = 1, ..., H$

3.3. SAP-DETR with Denoising Strategy

To further explore the capability of our proposed SAP-DETR, we develop SAP-DN-DETR and SAP-DINO-DETR by adding the denoising auxiliary loss [9,28] into the training process. In the denoised SAP-DETR, the main difference from both DN-DETR and DINO lies in the noise design. Instead of the center point, we perform the box jittering and randomly sample a point from the intersection region between the original bounding box and the jittering one as the reference point. As the denoising strategy only serves as an auxiliary training loss increasing the training cost, the variants of denoising models are test-free whose Params and GFLOPs are the same as SAP-DETR models.

4. Experimental Results

4.1. Implementation Details

We conduct the experiments on the COCO 2017 [12] object detection dataset, containing about 118K training images and 5K validation images. All models are evaluated by the standard COCO evaluation metrics. We follow the vanilla DETR [2] structure that consists of a CNN backbone, a stack of Transformer encoder-decoder layers, and two prediction heads for class label and bounding box prediction. We use ImageNet-pretrained ResNet [7] as our backbone, and report results based on the ResNet and its $\times$1/16-resolution extension ResNet-DC. Unlike DAB-DETR [13] sharing box and label head for each layer, we share the class head except the first layer and use an independent box head for the box regression of each layer (for more details please refer to Appendix D). As the mesh-grid initialization for reference points in SAP-DETR, we consider the number of queries N as a perfect square for uniform distribution. Unless otherwise specified, we use $N = 400$ queries in the experiments. Precisely, we also provide a comparison under $N = 300$ in Tab. 2, the standard setting in DETR-like models.

We adopt two different Transformer structures for experiments where a 3-layer encoder-decoder stack is evaluated to

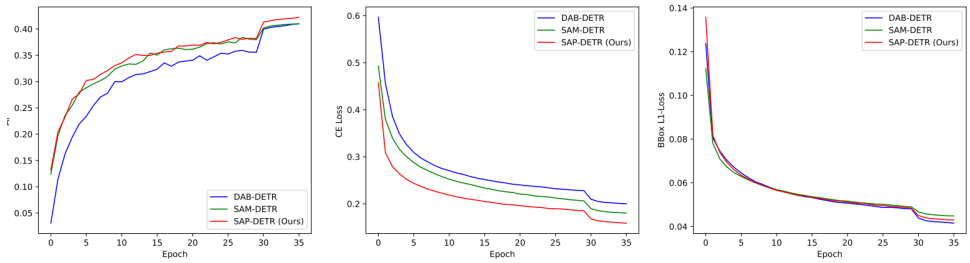


Figure 4. Comparison of performance and training losses curves.

demonstrate our lightweight model efficacy compared with the traditional CNN detectors, and a 6-layer encoder-decoder stack is aligned with previous DETR variants to investigate the performance of large model. Both are trained on two training schemes: the 12-epoch and 36-epoch schemes with a learning rate drop after 11 and 30 epochs, respectively. All models are trained on the Nvidia A100 GPUs with batch size of 16 and 8 for ResNet and ResNet-DC, respectively. For more training details, please refer to Appendix F.

4.2. Main Results

As shown in Tab. 1 and Tab. 2, we comprehensively compare our proposed SAP-DETR with the traditional CNN detectors [18], the original DETR [2], and other DETR-like detectors [5, 13, 16, 25, 26, 31] on COCO 2017 validation dataset. For in-depth analysis, we conduct the comparison in two aspects: model convergency and efficacy.

Model Convergency. Compared with traditional CNN detectors, Transformer detectors are always subject to laborious training time. For example, under the same 12-epoch training scheme, Faster RCNN [18] still achieves good performance, but the mainstream DETR-like models may suffer from inadequate training and perform poorly without the help of auxiliary losses [9]. Under the 12-epoch training scheme, our proposed SAP-DETR can accelerate model convergency significantly, boosting DAB-DETR [13] by 3.9 AP and 2.7 AP on 3-layer and 6-layer encoder-decoder structures, respectively. Compared with the current SoTA, our SAP-DETR also outperforms SAM-DETR [26] by $\sim$1.3 AP, with reducing $\sim$17% parameters and $\sim$10% GFLOPs. Take a close look at the training process, as illustrated in Fig. 4, SAP-DETR conducts with rapid descent curves in both classification and box regression losses. Notably, there is a large margin in classification loss between ours and SoTA methods, which is benefited from the query-specific reference point, hence boosting model performance in early epochs.

Model Efficacy. To analyze model efficacy, we report results on long training epochs and high-resolution features in Tab. 1. Under the 36-epoch training scheme, SAP-DETR achieves superior performance among all single-scale Transformer detectors, *especially on middle and large targets*. For example, SAP-DETR boosts DAB-DETR by 2.0 AP_M and 4.1 AP_L with 3-layer models, 1.0 AP_M and 1.9 AP_L with 6-layer models, which further verifies the effectiveness of our proposed salient point concept for overlapping object detection. Along with layer increase, a deficient upper-bound of SAM-DETR is exposed, with obviously lower 0.5 AP promotion compared to our 1.0 AP improvement. Persuasively, we also report the 50-epoch training results based on the 300-query setting. To align with our mesh-grid initialization strategy, we tile a 17×18 mesh-grid (306 queries) for each reference point initialization. Tab. 2 shows our main results and the most representative approaches with their original reported performance. Notably, SAP-DETR outperforms the current SoTAs with comparable costs based on all backbones. With low-resolution features ($\times 1/32$), it significantly boosts both middle and large object detection accuracy.

Combine with Other Fast Convergency Methods. As shown in Tab. 3, we compare our SAP-DETR variants with the current fast convergency methods [4, 9, 28]. With such a subtle modification, our SAP-DETR (grey rows) results in a significant performance improvement compared with the original methods (white rows). Under the 12-epoch training scheme, there exist 0.5-1.9 AP improvements on DN-DETR [9] and 0.7-1.6 AP improvements on Group-DETR [4], but the promotions are slightly reduced when implemented on DINO [28]. We hypothesise that there exists the same effect between the negative query of contrastive denoising [28] and our query-salient reference point. Moreover, we observe that the performance improvements largely originate from the large object detection, especially based on ResNet-DC5 family backbones. We speculate that DETR may prefer the high-resolution features ($\times$1/16) rather than the low-resolution ones ($\times$1/32), and our SAP-DETR can distinguish the salient points accurately on the high-resolution, thereby taking full advantage of the large object detection.

4.3. Ablation Study

Effectiveness of Each Component. To offer an intuitionistic comparison of model convergency for each component, Tab. 4 reports the effectiveness of them based on the 3-layer encoder-decoder structure and 12-epoch training scheme. ***1).*** The proposed salient point concept based on content embeddings improves the performance from 32.3 AP to 33.5

[1]https:// github.com/facebookresearch/detectron2

Method	#Epochs	#Params(M)	GFLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
3-Layer Encoder-Decoder Transformer Neck with ResNet-50 Backbone									
DETR-R50 [2]	36	33	82	15.8	28.0	15.4	5.3	16.7	24.6
Deformable DETR-R50 [31]	36	30	77	37.1	57.6	39.4	18.3	40.8	51.6
SMCA-DETR-R50 [5]	12 / 36	-	-	28.8 / 37.7	48.1 / 58.7	29.9 / 40.1	13.8 / 19.4	31.3 / 40.5	41.3 / 54.8
Conditional DETR-R50 [16]	12 / 36	40	82	29.6 / 37.1	48.7 / 57.9	30.7 / 39.0	13.0 / 17.6	32.3 / 40.3	43.1 / 55.0
Anchor DETR-R50 [25]	12 / 36	31	79	30.8 / 37.6	51.1 / 58.7	31.8 / 39.7	14.3 / 18.8	34.1 / 41.5	44.3 / 53.5
DAB-DETR-R50 [13]	12 / 36	34	83	32.3 / 39.0	51.3 / 58.6	34.0 / 41.8	15.7 / 20.0	35.2 / 42.5	45.7 / 56.0
SAM-DETR-w/SMCA-R50 [26]	12 / 36	41	89	35.1 / 40.4	54.7 / 60.7	36.7 / 42.7	16.0 / 20.2	38.4 / 44.4	52.1 / 58.3
SAP-DETR-R50 (Ours)	12 / 36	36	84	**36.2 / 41.2**	**56.2 / 61.6**	**37.9 / 43.4**	**16.4 / 21.0**	**39.5 / 44.5**	**53.8 / 60.1**
6-Layer Encoder-Decoder Transformer Neck with ResNet-50 Backbone									
DETR-R50 [2]	36	42	89	14.0	24.4	14.0	4.2	13.7	22.5
Deformable DETR-R50 [31]	36	34	81	38.0	58.2	40.4	18.5	41.7	54.2
SMCA-DETR-R50 [5]	12 / 36	-	-	32.4 / 40.1	52.3 / 61.4	34.0 / 42.8	15.5 / 20.3	34.9 / 43.3	47.7 / 57.1
Conditional DETR-R50 [16]	12 / 36	44	90	33.1 / 40.2	53.0 / 61.0	34.8 / 42.4	14.5 / 19.9	35.9 / 43.5	49.2 / 58.8
Anchor DETR-R50 [25]	12 / 36	37	85	33.7 / 39.7	54.5 / 60.5	35.1 / 41.9	15.6 / 19.9	37.3 / 43.5	49.8 / 57.3
DAB-DETR-R50 [13]	12 / 36	44	92	34.9 / 41.0	55.5 / 61.7	36.4 / 43.4	16.2 / 21.3	38.4 / 44.7	51.5 / 58.9
SAM-DETR-w/SMCA-R50 [26]	12 / 36	59	105	36.2 / 40.9	57.2 / 62.2	37.4 / 43.1	16.1 / 20.1	39.8 / 44.7	55.3 / 60.7
SAP-DETR-R50 (Ours)	12 / 36	47	94	**37.5 / 42.2**	**58.5 / 62.7**	**39.2 / 44.6**	**17.3 / 22.6**	**40.6 / 45.7**	**55.4 / 60.8**

Table 1. Comparison between Transformer necks. Based on ResNet-50 backbone, all models are trained by the official source codes with their original settings and evaluated on COCO `val2017`. All models uses 400 queries except Anchor DETR, while Anchor DETR uses 200 queries with 2 pattern embeddings. GFLOPs and Params are measured by Detectron2[1].

Method	#Epochs	#Params(M)	GFLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Infer. Time(s/img)†
ResNet-50 Backbone										
Faster RCNN-FPN-R50 [10, 18]	108	42	180	42.0	62.1	45.5	26.6	45.5	53.4	0.039
DETR-R50 [2]	500	41	86	42.0	62.4	44.2	20.5	45.8	61.1	0.040
Deformable DETR-R50 [31]	50	34	78	39.4	59.6	42.3	20.6	43.0	55.5	0.043
SMCA-DETR-R50 [5]	50	42	86	41.0	-	-	21.9	44.3	59.1	0.045
Conditional DETR-R50 [16]	50	44	90	40.9	61.8	43.3	20.8	44.6	59.2	0.057
Anchor DETR-R50 [25]	50	39	85	42.1	63.1	44.9	22.3	46.2	60.0	0.050
DAB-DETR-R50 [13]	50	44	90†	42.2	63.1	44.7	21.5	45.7	60.3	0.059
SAM-DETR-w/SMCA-R50 [26]	50	58	100	41.8	63.2	43.9	22.1	45.9	60.9	0.065
SAP-DETR-R50 (**Ours**)	50	47	92	**43.1**	**63.8**	**45.4**	**22.9**	**47.1**	**62.1**	0.063
ResNet-101 Backbone										
Faster RCNN-FPN-R101 [10, 18]	108	60	246	44.0	63.9	47.8	27.2	48.1	56.0	0.050
DETR-R101 [2]	500	60	152	43.5	63.8	46.4	21.9	48.0	61.8	0.066
Conditional DETR-R101 [16]	50	63	156	42.8	63.7	46.0	21.7	46.6	60.9	0.070
Anchor DETR-R101 [25]	50	58	150	43.5	64.3	46.6	23.2	47.7	61.4	0.068
DAB-DETR-R101 [13]	50	63	157†	43.5	63.9	46.6	23.6	47.3	61.5	0.072
SAP-DETR-R101 (**Ours**)	50	67	158	**44.4**	**64.9**	**47.1**	**24.1**	**48.7**	**63.1**	0.078
DC5-ResNet-50 Backbone										
DETR-DC5-R50 [2]	500	41	187	43.3	63.1	45.9	22.5	47.3	61.1	0.087
Conditional DETR-DC5-R50 [16]	50	44	195	43.8	64.4	46.7	24.0	47.6	60.7	0.093
Anchor DETR-DC5-R50 [25]	50	39	151	44.2	64.7	47.5	24.7	48.2	60.6	0.069
DAB-DETR-DC5-R50 [13]	50	44	194†	44.5	65.1	47.7	25.3	48.2	62.3	0.094
SAM-DETR-w/SMCA-DC5-R50 [26]	50	58	210	45.0	65.4	47.9	26.2	49.0	**63.3**	0.126
SAP-DETR-DC5-R50 (**Ours**)	50	47	197	**46.0**	**65.5**	**48.9**	**26.4**	**50.2**	62.6	0.116
DC5-ResNet-101 Backbone										
DETR-DC5-R101 [2]	500	60	253	44.9	64.7	47.7	23.7	49.5	62.3	0.101
Conditional DETR-DC5-R101 [16]	50	63	262	45.0	65.6	48.4	26.1	48.9	62.8	0.105
Anchor DETR-DC5-R101 [25]	50	58	227	45.1	65.7	48.8	25.8	49.4	61.6	0.083
DAB-DETR-DC5-R101 [13]	50	63	263†	45.8	65.9	49.3	27.0	49.8	63.8	0.110
SAP-DETR-DC5-R101 (**Ours**)	50	67	266	**46.9**	**66.7**	**50.5**	**27.9**	**51.3**	**64.3**	0.130

Table 2. Comparison of Transformer necks with 300 queries on COCO `val2017`. All results are reported from their original paper. All models uses 300 queries except Anchor DETR, while Anchor DETR uses 100 queries with 3 pattern embeddings. All inference speeds are measured by a single Nvidia A100 GPU. † denotes the results are measured by ourselves.

AP compared to baseline DAB-DETR (row 8-9). Such a query-specific spatial prior enables queries to attend to their expected region from content features (see Figures in Appendix G) and reduces the false detection rate on occluded and partial objects (see Fig. 1(d)), hence boosting detection performance on middle and large objects. However, there exists a drop in small object detection (15.7 AP_S *vs.* 14.3 AP_S), for which we consider that the failure is mainly caused by the query sparsity. ***2).*** Therefore, the movable strategy is applied to alleviate the constraint, improving the final model by 1.0 AP and 0.6 AP_S (row 1 and 6). ***3).*** Compared with the final model, the inner loss greatly improves the performance on high-quality AP_{75} and large objects AP_L detection (row 1 and 7), with just a slight drop on low-quality object AP_{50}. That is reasonable because the outside reference points are unable to localize objects accurately, and this phenomenon

Backbone	Epoch	w/ SAP	DN-DETR [9]		DINO (Single-Scale) [28]		Group DETR [4]	
			AP / AP_{50} / AP_{75}	AP_S / AP_M / AP_L	AP / AP_{50} / AP_{75}	AP_S / AP_M / AP_L	AP / AP_{50} / AP_{75}	AP_S / AP_M / AP_L
R50	12		38.3 / 58.6 / 40.5	18.4 / 41.6 / 57.1	39.7 / 58.3 / 42.4	19.1 / 43.7 / 57.1	39.1 / - / -	19.7 / 42.5 / 56.8
		✓(**Ours**)	**39.5 / 59.7 / 41.5**	**18.7 / 42.8 / 59.0**	**40.0 / 60.1** / 42.1	**20.2** / 43.4 / **58.5**	**39.8 / 60.2 / 42.0**	**20.2 / 43.5 / 58.6**
R101	12		40.5 / 60.8 / 43.0	19.3 / 44.3 / 59.6	41.9 / 60.8 / 44.4	22.5 / 46.3 / 59.5	- / - / -	- / - / -
		✓(**Ours**)	**41.0 / 61.2 / 43.4**	**19.8 / 45.3 / 60.0**	41.5 / **61.4** / 43.6	20.3 / 45.2 / **60.0**	**41.1 / 61.5 / 43.4**	**20.5 / 45.5 / 59.4**
R50-DC	12		41.7 / 61.4 / 44.1	21.2 / 45.0 / 60.2	43.6 / 61.4 / 47.0	24.8 / 47.3 / 59.5	41.9 / - / -	23.3 / 45.6 / 58.4
		✓(**Ours**)	**43.6 / 62.5 / 46.2**	**23.3 / 47.3 / 61.0**	**44.0 / 63.1** / 46.5	**24.8 / 47.3 / 61.1**	**43.9 / 63.2 / 46.8**	**24.5 / 47.6 / 61.3**
R101-DC	12		43.4 / 61.9 / 47.2	24.8 / 46.8 / 59.4	45.4 / 63.5 / 49.2	26.4 / 49.5 / 61.1	- / - / -	- / - / -
		✓(**Ours**)	**44.6 / 63.9 / 48.0**	**25.5 / 48.9 / 62.5**	**45.6 / 64.5** / 48.7	25.0 / **49.7 / 62.5**	**44.4 / 63.9 /47.4**	**25.9 / 48.5 / 61.4**

Table 3. Comparison with denoised methods on COCO dataset based on the 12-epoch training schedule and 300 object queries.

Comment	Movable	Inner Loss	PECA	SDG	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
SAP-DETR (**Ours**)	✓	✓	✓	✓	**36.2**	56.2	**37.9**	**16.4**	**39.5**	**53.8**
−SDG	✓	✓	✓		35.6	56.2	36.9	16.3	38.9	52.7
−PECA	✓	✓		✓	34.8	55.5	36.0	15.7	37.3	52.0
−PECA & SDG	✓	✓			34.0	54.9	35.3	15.0	36.7	51.5
−Movable		✓	✓	✓	35.2	55.4	36.8	15.8	38.5	**53.8**
−Inner Loss	✓		✓	✓	35.9	**56.3**	37.4	16.2	39.3	52.5
DAB-DETR (Baseline)	-	-	-	-	32.3	51.3	34.0	15.7	35.2	45.7
+Salient Point Concept	-	-	-	-	33.5	54.3	35.1	14.3	36.5	51.0

Table 4. Ablation on each components

Inner Cost ($\mathcal{L}_{\text{inner}}$)	Movable within Grid (s_{grid})	AP	AP_S	AP_M	AP_L
		35.9	**17.0**	38.8	52.7
✓		26.3	11.3	28.0	39.5
✓	✓	**36.2**	16.4	**39.5**	**53.8**

Table 5. Ablation on scaling factor of grid

PECA	Scaling Factor of SDG	AP	AP_S	AP_M	AP_L
		33.6	14.7	36.0	50.7
✓		35.7	**17.5**	38.8	52.6
✓	✓	**36.2**	16.4	**39.5**	**53.8**

Table 6. Ablation on scaling factor of SDG

always exists in large objects. ***4).*** For salient point enhanced cross-attention, both SDG and PECA serve as the essential components, independently emerging 0.8 AP and 1.6 AP improvements compared to the standard model (rows 2-4). Interestingly, there exists an effectiveness overlap on small objects, with only 0.1 AP_S improvement when adding SDG to the equipped PECA model (row 1 and 2). We argue that the Gaussian-like map of SDG might be easily overlapped with PECA on small objects.

Scaling Factor of Movable Strategy. We perform an ablation study on the scaling factor of the movable strategy and further investigate the effectiveness of the inner cost in Tab. 5. Notably, it is observed that there exists a conflict between the inner loss and the global search strategy, behaving a sharp drop when only reserving the inner loss. Furthermore, searching within the grid enables the detector to more attend to small objects and avoid a drastic deterioration in normal object detection. See Appendix E for more detailed analyses.

Scaling Factor of SDG. We also compare our side-directed manner with the standard offset prediction method in Tab. 6. Based on PECA, the side-directed scaling factor may limit the detector on small object detection but significantly promote the performance on other objects. This phenomenon would be broken without the help of PECA in which a precipitous decline is emerged on all-scale object detection (row 5 in Tab. 4 *vs.* row 1 in Tab. 6). We hypothesise that it because the predicted reference points may be outside of the proposal boxes, or even the region of the image.

5. Conclusion

In this paper, we propose SAP-DETR for promoting model convergency by treating object detection as a transformation from the salient points to the instance objects. Our SAP-DETR explicitly initializes a query-specific reference point for each object query, gradually aggregates them into an instance object, and predicts the distance from each side of the bounding box. By speedily attending to the query-specific region and other extreme regions from contextual image features, it thus can effectively bridge the gap between the salient points and the query-based Transformer detector. Our extensive experiments have demonstrated that SAP-DETR achieves superior model convergency speed. With the same training settings, our proposed SAP-DETR outperforms SoTA approaches with large margins.

6. Future Work

This point-based design for DETR-like models also comes with remaining issues, in particular regarding training with deformable attention, multi-scale features, and negative query design. Following current center-based methods working for similar issues, we expect future work to successfully address them for point-based design of SAP-DETR.

Acknowledgements

We thank the Lenovo Research AI Master platform for computing GPU supports without which this work would not be possible.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, pages 5561–5569, 2017. 3

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 1, 3, 4, 5, 6, 7, 10

[3] Liangyu Chen, Tong Yang, Xiangyu Zhang, Wei Zhang, and Jian Sun. Points as queries: Weakly semi-supervised object detection by points. In *CVPR*, pages 8823–8832, 2021. 3

[4] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group detr:fast training convergence with deco-upled one-to-many label assignment. *arXiv preprint arXiv:2207.13085*, 2022. 3, 6, 8

[5] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *ICCV*, pages 3621–3630, 2021. 2, 3, 5, 6, 7, 10

[6] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[8] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018. 3

[9] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, pages 13619–13627, 2022. 3, 5, 6, 8, 13

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 7

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 1, 3

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 5

[13] 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. *ICLR*, 2022. 2, 3, 4, 5, 6, 7, 10

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 1

[15] Yang Liu, Yao Zhang, Yixin Wang, Feng Hou, Jin Yuan, Jiang Tian, Yang Zhang, Zhongchao Shi, Jianping Fan, and Zhiqiang He. A survey of visual transformers. *arXiv preprint arXiv:2111.06091*, 2021. 3

[16] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, pages 3651–3660, 2021. 2, 3, 4, 5, 6, 7, 10, 11

[17] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 1, 6, 7

[19] Peize Sun, Yi Jiang, Enze Xie, Wenqi Shao, Zehuan Yuan, Changhu Wang, and Ping Luo. What makes for end-to-end object detection? In *ICML*, pages 9934–9944, 2021. 2

[20] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*, 2020. 1

[21] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *ICCV*, pages 3611–3620, 2021. 3

[22] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 1, 3, 4

[23] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 12

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 1, 10

[25] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, volume 36, pages 2567–2575, 2022. 2, 3, 4, 6, 7, 10, 12

[26] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. In *CVPR*, pages 949–958, 2022. 3, 6, 7, 10

[27] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Jiaxing Huang, Kaiwen Cui, Shijian Lu, and Eric P Xing. Semantic-aligned matching for enhanced detr convergence and multi-scale feature fusion. *arXiv preprint arXiv:2207.14172*, 2022. 3

[28] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *ICLR*, 2022. 3, 5, 6, 8

[29] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 3

[30] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, pages 850–859, 2019. 3

[31] 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*, 2020. 2, 3, 6, 7, 10