DAPE: Harmonizing Content-Position Encoding for Versatile Dense Visual Prediction

Xiuquan Hou[1], **Meiqin Liu**[2,1,*], **Senlin Zhang**[2], **Shaoyi Du**[1]

[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence
Xi'an Jiaotong University, Xi'an, 710049, China
[2]College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China
xiuqhou@stu.xjtu.edu.cn, liumeiqin@zju.edu.cn, slzhang@zju.edu.cn, dushaoyi@mail.xjtu.edu.cn

Abstract

Dense visual prediction tasks, including object detection and segmentation, inherently require precise and discriminative positional information to delineate object boundaries and pixel regions. Recent DETR-based frameworks advance dense prediction tasks through iterative attention applied to content queries, with sampled proposals as position references. However, this paradigm suffers from the misaligned sampling distribution and insufficient interaction between the content and position features, thereby limiting the encoding effectiveness. To overcome these limitations, we investigate the encoding paradigm for content-position harmonization and propose an effective predictor for dense visual tasks, termed DAPE (DETR with hArmonized content-Position Encoding). DAPE introduces explicit position encoding to facilitate content enhancement while maintaining low memory overhead. To achieves this process, DAPE comprises a Shifted Query Sampler (SQS) that enforces strict alignment between the distributions of content and position queries, and a 2D Low-Rank Position Encoder (LRPE) that progressively modulates attention maps based on the aligned representations. DAPE provides a unified solution for various dense prediction tasks. Extensive experiments on object detection, instance segmentation, and few-shot detection benchmarks demonstrate that DAPE achieves state-of-the-art performance while reducing memory consumption.

Code — https://github.com/xiuqhou/DAPE

Introduction

Dense prediction tasks (*e.g.*, object detection (Ren et al. 2016), instance segmentation (He et al. 2017), and few-shot detection (Kang et al. 2019)) pursue fine-grained localization and classification at instance or pixel levels. These tasks critically rely on precise positional representations to accurately delineate object boundaries and pixel regions. Traditionally, convention-based methods incorporate explicit spatial priors (*e.g.* anchor proposals (Ren et al. 2016), center points (Duan et al. 2019, 2023), and corner keypoints (Law and Deng 2018)) to generate potential object candidates.

Recently, DEtection TRansformer (DETR) (Carion et al. 2020) has emerged as an end-to-end framework that eliminates the need for hand-crafted priors and heuristics in vision tasks. DETR generates queries directly from the memory enhanced by a transformer encoder and decodes them into predictions through iterative attention (Carion et al. 2020; Zhang et al. 2022b; Chen et al. 2023; Zhang et al. 2023; Hou et al. 2024a). In this process, positional encodings are employed to guide the attention over content queries. Many works explore position-aware queries, such as anchor-based queries (Liu et al. 2022), dense distinct queries (Zhang et al. 2023), to enable accurate dense visual prediction.

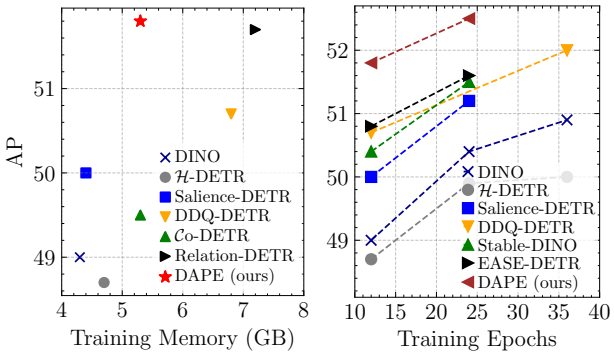

Figure 1: Comparison of training memory and performance on COCO (Lin et al. 2014) with ResNet-50 backbone.

Despite significant advances, the existing content-position encoding paradigm in DETR still suffers from two limitations, namely **distribution misalignment** and **inexplicit position encoding**. Conventionally, DETR represents objects as a set of queries composed of content embeddings $\boldsymbol{Q}_c$ and position embeddings $\boldsymbol{Q}_p$. However, as depicted in Fig. 2, $\boldsymbol{Q}_p$ is initialized from proposals $\tilde{\boldsymbol{b}}$ refined by regression deltas $\boldsymbol{\delta} = (\delta_x, \delta_y, \delta_w, \delta_h)$, while content queries $\boldsymbol{Q}_c$ are directly initialized from the top-N_m pixel features. This independent sampling process treats the two types of information separately, overlooking their inherent distributional shift (Zhu et al. 2021). Such misalignment introduces severely misleading positional references for the content queries, thereby impairing the effectiveness of the encoding and slowing convergence (Zhang et al. 2023).

Second, the lack of explicit spatial priors in the query generation process further exacerbates the difficulty of modeling positional relations between objects (Hou et al. 2024b).

*Corresponding author.

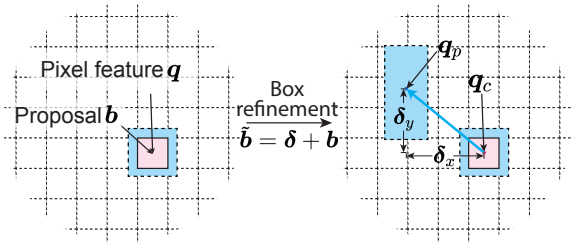

Figure 2: Misalignment between Q_c and Q_p after one-step bounding box refinement.

Prevalent DETR frameworks (Zhu et al. 2021; Zheng et al. 2023) typically employ 1D sinusoidal absolute positional encodings before projecting them to Q and K for attention computation (Vaswani 2017), *i.e.*, $\boldsymbol{Q} = \boldsymbol{Q}_p + \boldsymbol{Q}_c$. However, this approach couples position and content information in a 1D manner, without explicitly capturing relative geometric structures and spatial dependencies required by 2D attention map. Although directly encoding 2D positional relations (Hou et al. 2024b) enhances positional awareness, it incurs quadratic memory overhead, as shown in Fig. 1, which becomes prohibitive for dense prediction tasks with numerous objects.

To address the above issues, we propose a unified predictor for dense visual prediction tasks, termed DAPE (DETR with hArmonized content-Position Encoding). DAPE resolves the distribution misalignment through a Shifted Query Sampler (SQS) and mitigates the limitations of position awareness via a 2D Low-Rank Position Encoder (LRPE). Specifically, SQS initializes content sampling positions that are aligned with the position queries and propagates them across scales via spatially coherent sampling, thereby enabling consistent feature aggregation. Based on the sampled queries, LRPE encodes positional information in a 2D low-rank space to facilitate attention calculation, while preserving linear complexity. In contrast to existing DETR-based methods (Zhu et al. 2021; Zhang et al. 2023; Hou et al. 2024b) that are tailored for single dense prediction task, DAPE provides a unified architecture for *object detection*, *instance segmentation*, and *few-shot detection*, *etc*. Remarkably, DAPE achieves superior performance across diverse tasks without any task-specific tuning, outperforming specialized counterparts and underscoring its effectiveness.

Related Work

Object Representation in Object Detection

Spatial priors provide an initial distribution over the spatial dimensions to guide the detection process to start from high-confidence areas. This paradigm dates back to the sliding window-based approach (Viola and Jones 2001, 2004), which assumes potential objects are uniformly and densely distributed (Gao et al. 2022). It has further developed from region-based search (Girshick 2015) to learning-based proposals (Ren et al. 2016). Meanwhile, extensive convolutional detectors have advanced the research by proposing various forms of priors, such as anchor proposals (Ren et al. 2016) and point-based methods (Duan et al. 2019; Zhou, Wang, and Krähenbühl 2019; Law and Deng 2018). Recently, prevalent DETR-based detectors (Zhang et al. 2022b; Chen et al. 2023; Zhang et al. 2023; Hou et al. 2024a,b) almost exclusively utilize embeddings to represent objects.

Memory-Efficient Attention

The quadratic growth in attention computation with respect to sequence length leads to high computational complexity and memory consumption in transformers (Vaswani 2017). Various memory-efficient methods have been proposed to tackle this issue, primarily focusing on sparse (Zhu et al. 2021; Liu et al. 2021) and low-rank (Wang et al. 2020a) aspects. Sparse methods perform attention on a subset of sequence elements to enhance efficiency, guided by locality-sensitive rules (Zhu et al. 2021; Liu et al. 2021) or self-learned importance metrics for each element (Zheng et al. 2023; Hou et al. 2024a). Low-rank methods factorize the attention weights into low-rank matrices to minimize redundant computations (Wang et al. 2020a). This motivation also informs parameter-efficient fine-tuning techniques exemplified by the LoRA series (Hu et al. 2022). However, limited research in DETR has addressed the gradual increase in memory consumption that accompanies improved performance, particularly from the perspective of low-rank position encoding (Hu et al. 2018; Hou et al. 2024b).

Position-Centric Encoding in DETR

Transformers typically process entire sequences in parallel, leveraging position encoding to capture the order information inherent in the sequence (Vaswani 2017). In the context of object detection, this order information generally represents the position coordinates of pixels or bounding boxes within the spatial (*i.e.* height and width) dimensions of an image. It is vital for adapting to variations in object positions and enhancing the accuracy of bounding box regression (Hou et al. 2024b; Ouyang-Zhang et al. 2022). As a result, massive research has been dedicated to enhance position perception in object detection. Some focus on enhancing query or object representations through position features, such as dense distinct queries (Zhang et al. 2023), anchor queries (Wang et al. 2022), and dynamic anchor queries (Liu et al. 2022). Others aim to foster better interactions among queries, including the use of position relations (Hou et al. 2024b), query ranking or competition mechanisms (Pu et al. 2024; Gao et al. 2024). These methods have shown promise in improving the performance of object detection models. However, the increased memory requirements also pose significant challenges for dense visual tasks.

DAPE

The overall pipeline of DAPE is depicted in Fig. 3, which harmonizes content and position encoding through two key components: (1) aligning the sampling distributions of 1D content and position queries, and (2) incorporating explicit 2D position encoding to guide content interactions in attention. Specifically, DAPE first extracts multi-scale features

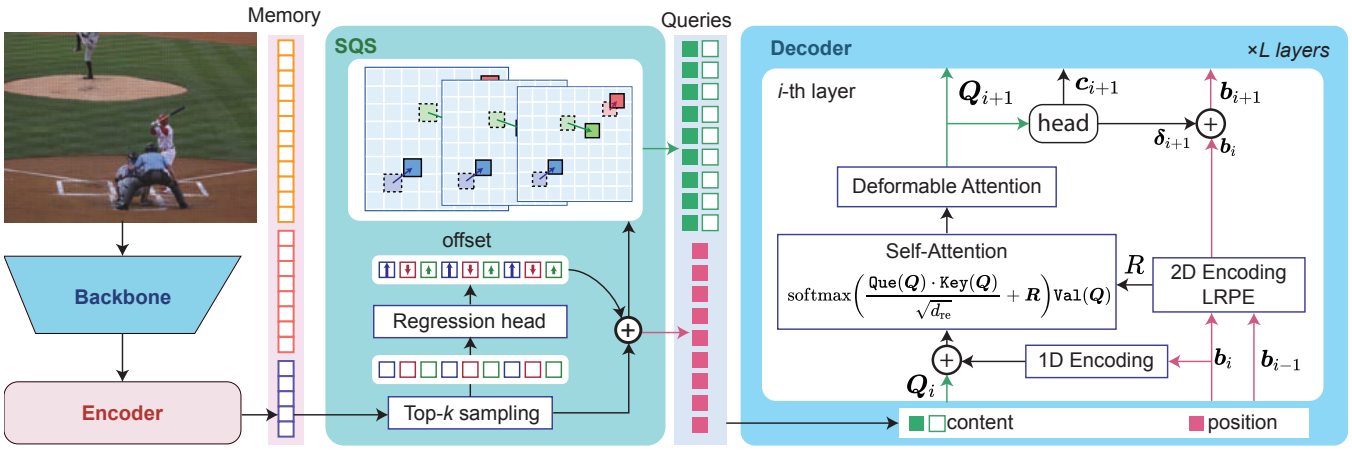

Figure 3: Overview of DAPE. DAPE consists of a backbone, a transformer encoder, and a transformer decoder. The backbone extracts multi-scale features $\{\boldsymbol{f}_l\}_{l=1}^{L}$ from the input image. The transformer encoder refines these features and generates content queries $\boldsymbol{Q}_c$ and position queries $\boldsymbol{Q}_p$. The transformer decoder decodes the queries into final predictions. DAPE employs a Shifted Query Sampler (SQS) to align the sampling of content and position queries, and integrates a Low-Rank Position Encoder (LRPE) to enhance positional awareness in DETR.

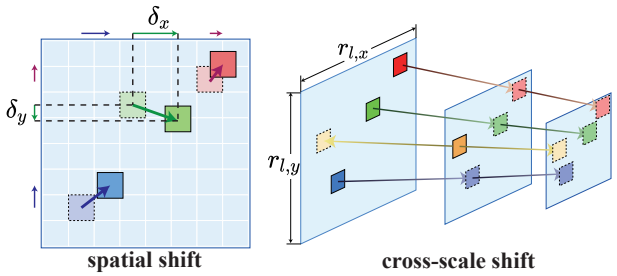

Figure 4: Illustration of spatial shift (left) and cross-scale shift (right) in our SQS.

$\{\boldsymbol{f}_l\}_{l=1}^{L}$ from the input image and encodes them into a flattened feature memory $\boldsymbol{f}$ via a transformer encoder. Subsequently, the Shifted Query Sampler (SQS) samples spatially aligned position queries $\boldsymbol{Q}_p$ and content queries $\boldsymbol{Q}_c$ from the memory. These sampled queries are then fed into a transformer decoder equipped with a Low-Rank Position Encoder (LRPE) for feature enhancement. The transformer encoder and decoder are supervised using one-to-many and one-to-one matching strategies, respectively.

Shifted Query Sampler

We propose Shifted Query Sampler (SQS) to align the sampling distributions of content and position queries. SQS accomplishes this by introducing spatial and cross-scale shifts to the coordinates produced by conventional top-N_m sampling strategies, as illustrated in Fig. 4.

Formally, given the flattened memory $\boldsymbol{f}$, we first select pixel features $\boldsymbol{f}(x, y)$ with the top-N_m classification scores as the initial content queries $\boldsymbol{Q}_c$, following prior works (Zhu et al. 2021; Zhang et al. 2023). Based on $\boldsymbol{f}(x, y)$, we predict the corresponding regression offsets $\boldsymbol{\delta} = (\delta_x, \delta_y, \delta_w, \delta_h)$ using a linear layer, and shift the sampling coordinates of $\boldsymbol{Q}_c$ accordingly. This shift ensures strict alignment between content queries $\boldsymbol{Q}_c$ and positional queries $\boldsymbol{Q}_p$.

$$(\delta_x, \delta_y, \delta_w, \delta_h) = \boldsymbol{W}_\delta \boldsymbol{f}(x, y) \tag{1}$$

$$\boldsymbol{Q}_c^1 = \boldsymbol{f}_l\,(x + \delta_x, y + \delta_y) \tag{2}$$

$$\boldsymbol{Q}_p = \boldsymbol{W}_p \tilde{\boldsymbol{b}}(x + \delta_x, y + \delta_y) \tag{3}$$

We further extend (2) to enable spatially-aligned content sampling across different object sizes by applying a cross-scale shift to the coordinates of $\boldsymbol{Q}c$. Specifically, let r_l denote the scale factor of the l-th feature map, the sampling process of $\boldsymbol{Q}_c$ is updated as:

$$\boldsymbol{Q}_c^2 = \sum_{l=1}^{L} \boldsymbol{W}_{l,c} \boldsymbol{f}_l\,(r_{l,x}(x + \delta_x), r_{l,y}(y + \delta_y)) \tag{4}$$

where $\boldsymbol{W}_{l,c} \in \mathbb{R}^{d \times d}$ is a learnable parameter, and d denotes the embedding dimension of $\boldsymbol{Q}_c$. Given the determined sampling positions, we adopt point-based sampling combined with learnable embeddings, as it outperforms region proposals (Ren et al. 2016) (see Technical Appendix for details). The final content queries $\boldsymbol{Q}_c$ are thus constructed by fusing the sampled features $\boldsymbol{Q}_c^2$ with a learnable embedding $\texttt{Embed}$:

$$\boldsymbol{Q}_c = \boldsymbol{Q}_c^2 + \boldsymbol{W}_e \texttt{Embed} \tag{5}$$

Low-Rank Position Encoding

Unlike the 1D positional encoding used in prior DETR frameworks (Zhu et al. 2021; Zhang et al. 2022b), which fails to adequately capture the geometric structures and spatial dependencies of 2D attention maps, our Low-Rank Position Encoder (LRPE) directly encodes positional information in the 2D feature space. LRPE realizes linear memory complexity through a low-rank scheme.

2D Position Encoding To be specific, given N_m bounding boxes predicted by each decoder layer, the relative coordinates can be encoded into a 2D geometric feature $\boldsymbol{G} \in \mathbb{R}^{N_m \times N_m \times 4}$. The feature is further processed by sin-cos embedding and projected to $\boldsymbol{R}$ through linear transformation, used to modulate the attention map.

$$\boldsymbol{E}(\boldsymbol{G}, 2k) = \sin(s\boldsymbol{G}/T^{2k/d_{\text{re}}}) \tag{6}$$

$$\boldsymbol{E}(\boldsymbol{G}, 2k+1) = \cos(s\boldsymbol{G}/T^{2k/d_{\text{re}}}) \tag{7}$$

$$\boldsymbol{R} = \sum_k \boldsymbol{W}_{ak}\boldsymbol{E}(G, 2k) + \boldsymbol{W}_{bk}\boldsymbol{E}(G, 2k+1) \tag{8}$$

$$\boldsymbol{Q} = \text{softmax}\left(\frac{\texttt{Que}(\boldsymbol{Q}) \cdot \texttt{Key}(\boldsymbol{Q})}{\sqrt{d_{\text{re}}}} + \boldsymbol{R}\right)\texttt{Val}(\boldsymbol{Q}) \tag{9}$$

where s, T, d_{re} are encoding parameters, $\boldsymbol{E} \in \mathbb{R}^{N_m \times N_m \times 4d_{\text{re}}}$, $\boldsymbol{R} \in \mathbb{R}^{N_m \times N_m \times h}$, $\boldsymbol{W}_{ak}, \boldsymbol{W}_{bk} \in \mathbb{R}^{4 \times h}$, h denotes the number of attention head.

Low-Rank Position Encoding Since the memory cost arises from the quadratic spatial complexity with respect to N_m, we propose a low-rank encoding scheme to reduce it to linear complexity by factorizing $\boldsymbol{E}$ into two smaller matrices, $\boldsymbol{E}_A$ and $\boldsymbol{E}_B$, inspired by LoRA (Hu et al. 2022):

$$\boldsymbol{E} = \hat{\boldsymbol{E}} + g(\boldsymbol{E}_A, \boldsymbol{E}_B) \tag{10}$$

The key challenge lies in how to design $\boldsymbol{E}_A$, $\boldsymbol{E}_B$, and the function g.

By combining (6), (7), and (8), we observe that the 2D position encoding performs an element-wise periodic function approximation of the geometric feature $\boldsymbol{G}$ using a Fourier series:

$$\boldsymbol{R} = \sum_k \boldsymbol{W}_{ak}\sin(\omega_k\boldsymbol{G}) + \boldsymbol{W}_{bk}\cos(\omega_k\boldsymbol{G}) \tag{11}$$

where $\omega_k = s/T^{2k/d_{\text{re}}}$ for simplicity. Assuming the relative geometric feature is represented as $\boldsymbol{G}_{i,j} = g_A(\boldsymbol{b}_i) - g_B(\boldsymbol{b}_j)$, then we have the following factorization:

$$\begin{aligned}\cos(\omega_k G_{i,j,n}) &= \cos(\omega_k g_A(\boldsymbol{b}_i)_n - \omega_k g_B(\boldsymbol{b}_j)_n) &(12)\\ &= \langle \boldsymbol{e}_A(\boldsymbol{b}_i)_n, \boldsymbol{e}_B(\boldsymbol{b}_j)_n\rangle &(13)\\ \sin(\omega_k G_{i,j,n}) &= \sin(\omega_k g_A(\boldsymbol{b}_i)_n - \omega_k g_B(\boldsymbol{b}_j)_n) &(14)\\ &= \langle \boldsymbol{e}_A(\boldsymbol{b}_i)_n \boldsymbol{R}_{90^\circ}, \boldsymbol{e}_B(\boldsymbol{b}_j)_n\rangle &(15)\end{aligned}$$

where

$$\boldsymbol{e}_A(\boldsymbol{b}_i)_n = [\sin(\omega_k g_A(\boldsymbol{b}_i)_n), \cos(\omega_k g_A(\boldsymbol{b}_i)_n)] \tag{16}$$

$$\boldsymbol{e}_B(\boldsymbol{b}_j)_n = [\sin(\omega_k g_B(\boldsymbol{b}_j)_n), \cos(\omega_k g_B(\boldsymbol{b}_j)_n)] \tag{17}$$

and $\boldsymbol{R}_{90^\circ}$ is a rotation matrix that rotates vectors by 90° counterclockwise,

$$\boldsymbol{R}_{90^\circ} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{18}$$

It is worth noting that Eq. (16) and Eq. (17) exactly correspond to the sine-cosine embeddings of $g_A(\boldsymbol{b}_i)$ and $g_B(\boldsymbol{b}_j)$, respectively. In other words, the sine-cosine embedding of the relative position $\boldsymbol{G}_{i,j} = g_A(\boldsymbol{b}_i) - g_B(\boldsymbol{b}_j)$ can be formulated as a function of the inner product between the embeddings of $g_A(\boldsymbol{b}_i)$ and $g_B(\boldsymbol{b}_j)$, followed by linear transformations. Based on this observation, the 2D position encoding can be expressed in a low-rank form:

$$\boldsymbol{R} = \boldsymbol{W}_A\boldsymbol{E}_A(\boldsymbol{W}_B\boldsymbol{E}_B)^\top \tag{19}$$

where $\boldsymbol{E}_A, \boldsymbol{E}_B \in \mathbb{R}^{N_m \times 4d_{\text{re}}}$ are the sine-cosine embeddings of $g_A(\boldsymbol{b})$ and $g_B(\boldsymbol{b})$, respectively, and $\boldsymbol{W}_A, \boldsymbol{W}_B$ are learnable projection matrices. When employing a multi-head formulation for the inner product, the number of heads naturally corresponds to the rank r in LoRA (Hu et al. 2022).

For simplicity, we implement g_A and g_B as linear transformations with ReLU activation. The frozen component $\hat{\boldsymbol{E}}$ adopts the original 2D position encoding but with all learnable modules removed, thereby eliminating the need to store intermediate outputs for backpropagation and thus reducing memory consumption. The rank r is empirically set to 16 according to the ablation study in Tab. 6.

Matching and Supervision

We adopt a hybrid matching strategy to supervise DAPE. Specifically, one-to-many matching with non-maximum suppression (NMS) is used in the first stage to provide sufficient candidates, while one-to-one matching is applied in the second-stage decoder to produce non-redundant outputs. The one-to-many matching leverages threshold-based assignment strategy based on classification and IoU scores. For the i-th ground truth $y_i = (c_i^*, \boldsymbol{b}_i^*)$ and the j-th prediction $\hat{y}_j = (\hat{\boldsymbol{p}}_j, \hat{\boldsymbol{b}}_j)$, the matching score is defined as:

$$\mathcal{L}_{\text{o2m}}(y_i, \hat{y}_j) = \alpha\hat{p}_j(c_i^*) + (1-\alpha)\text{IoU}(\boldsymbol{b}_i^*, \hat{\boldsymbol{b}}_j) \tag{20}$$

where $\hat{p}_j(c_i^*)$ denotes the classification confidence of the j-th prediction for class c_i^*, $\boldsymbol{b}^*$ and $\hat{\boldsymbol{b}}$ denote the bounding boxes of ground truths and predictions, respectively. In our implementation, the coefficient α is set to 0.3. For each ground truth, we select top-k from all matched predictions with matching score larger than 0.4 for loss calculation.

Micro Designs for Dense Visual Prediction

One of the key advantages of DAPE is its strong representational adaptability and memory efficiency, enabling a unified framework for diverse dense visual prediction tasks. Rather than relying on task-specific architectures, DAPE supports object detection, instance segmentation, and few-shot detection within a shared backbone and representation space, augmented with several micro designs to support each task.

Object Detection Similar to prior works (Carion et al. 2020; Zhang et al. 2022b; Hu et al. 2018), DAPE adopts linear projection layers to predict class labels $\boldsymbol{P} \in \mathbb{R}^{N_m \times C}$ and bounding boxes $\boldsymbol{B} \in \mathbb{R}^{N_m \times 4}$ at each transformer decoder layer. The bounding box prediction is performed in an iterative refinement manner, as in (Zhu et al. 2021):

$$\boldsymbol{P} = \sigma(\boldsymbol{W}_P\boldsymbol{Q}) \tag{21}$$

$$\boldsymbol{B} = \sigma\left(\boldsymbol{W}_B\boldsymbol{Q} + \sigma^{-1}\left(\boldsymbol{B}^{(k-1)}\right)\right) \tag{22}$$

where C is the number of classes, k denotes the index of the decoder layer, and σ is the sigmoid function.

Instance Segmentation Consistent with previous works (Zhang et al. 2022b; Li et al. 2023), we introduce an additional branch to predict instance masks. Specifically, we fuse the $1/4$ feature map from the backbone with the upsampled $1/8$ feature map from the transformer encoder to construct the mask embedding $\boldsymbol{F}_{\text{mask}}$. The mask predictions $\boldsymbol{M} \in \mathbb{R}^{N_m \times h \times w}$ are obtained by computing the dot product between the projected queries $\boldsymbol{Q}$ and the mask feature $\boldsymbol{F}_{\text{mask}}$:

$$\boldsymbol{M} = \boldsymbol{W}_M \boldsymbol{Q} \otimes \boldsymbol{F}_{\text{mask}} \tag{23}$$

The masks are supervised using both pixel-wise binary cross-entropy loss and Dice loss, each with a weight of 5. For clarity, we refer to the DAPE for instance segmentation as Mask-DAPE.

Few-Shot Object Detection Few-Shot Object Detection (FSOD) (Kang et al. 2019; Fan et al. 2020) aims to equip object detectors with the ability to learn from only a few labeled examples. In meta-learning based FSOD methods (Zhang et al. 2022a; Antonelli et al. 2022), this is typically achieved by conditioning predictions on a set of support images. Specifically, for each $\boldsymbol{Q}$, we sample N_{way} support features $\boldsymbol{Q}_{\text{way}}$ along with their corresponding classes $\Omega_c = \{c_i\}_{i=1}^{N_{\text{way}}}$. Then, we introduce an additional cross-attention module into the transformer decoder to draw relevant context from the support features during prediction.

$$\hat{\boldsymbol{Q}} = \text{Attention}(\texttt{Que}(\boldsymbol{Q}), \texttt{Key/Val}(\boldsymbol{Q}_{\text{way}})) \tag{24}$$

For training, $\boldsymbol{Q}_{\text{way}}$ are randomly sampled from features extracted from images in the training set. During inference, they are initialized as the class-wise average of all support features, also known as class prototypes.

For each image with ground truths $\{y_i = (c_i^*, \boldsymbol{b}_i^*)\}$, only objects whose classes belong to Ω_c are considered positive for training supervision.

$$y_i^* = \{(c_i^*, \boldsymbol{b}_i^*) | c_i^* \in \Omega_c\} \tag{25}$$

The model for few-shot object detection is denoted as Meta-DAPE.

Experiments

Implementation Details

The training settings largely follow previous DETR methods for fair comparison. We use the AdamW optimizer with weight decay 1×10^{-4}. The learning rate varies with batch size but is consistent with DINO (Zhang et al. 2022b) (*lr=2e-4, bs=16*) and Relation-DETR (Hou et al. 2024b) (*lr=1e-4, bs=10*), decayed by 0.1 at later stages. Data augmentation follows the resize and crop strategy from DETR (Carion et al. 2020). For few-shot object detection, we adopt the base/novel split from Meta-DETR (Zhang et al. 2022a) on MS-COCO (Lin et al. 2014), with 60 base and 20 novel classes. The model is trained on base classes, then fine-tuned and evaluated on both base and novel classes (generalized FSOD, gFSOD (Wang et al. 2020b)). Following Meta-DETR (Zhang et al. 2022a), we train Meta-DAPE for 25 epochs on the base dataset, followed by fine-tuning until convergence on the balanced few-shot dataset. Consistent with baselines (Zhang et al. 2022b; Hou et al. 2024b; Li et al. 2023), we use 900 queries and 6 decoder layers for (few-shot) object detection, and 300 queries with 9 decoder layers for instance segmentation. The encoder has 6 fixed layers for all tasks. Average Precision (AP) is used as the evaluation metric. AP for base classes, novel classes, bounding boxes, and segmentation are denoted bAP, nAP, AP, and AP^{m}, respectively. More implementation details can be found in the Technical Appendix.

Figure 5: Segmentation visualization of Mask-DAPE.

Comparison with State-of-the-Art Methods

Object Detection Tab. 1 presents a quantitative comparison on the COCO `val2017`. Compared to other methods, DAPE achieves impressive performance (51.8% AP for $1\times$ schedule and 52.5% AP for $2\times$) while consuming less memory (26.4% and 61.0% less than Relation-DETR (Hou et al. 2024b) and Co-DINO (Zong, Song, and Liu 2023)), demonstrating a superior trade-off between memory usage and accuracy. When further integrated with a Swin-L backbone (See Technical Appendix for more details), DAPE surpasses all competing detectors, achieving a leading 58.1% AP, indicating promising scalability with larger model capacity. More importantly, DAPE trained for only 12 epochs (58.1% AP) outperforms DINO (Zhang et al. 2022b) (58.0% AP) trained for 36 epochs, demonstrating its substantially faster convergence.

Instance Segmentation Tab. 2 presents the instance segmentation results. Compared to previous state-of-the-art methods, our Mask-DAPE achieves superior performance with 43.7% Mask AP, surpassing DI-MaskDINO and MaskDINO (Li et al. 2023) by 1.4% and 2.3%, respectively. The qualitative visualizations in Fig. 5 demonstrate that Mask-DAPE produces impressive segmentation results, even for heavily occluded objects (*e.g.* the person inside a bus) and extremely small objects. These results highlight the effectiveness of DAPE for instance segmentation.

Few-Shot Object Detection The FSOD experiment uses the dataset split from (Zhang et al. 2022a). In addition to Meta-DAPE, we include results of DAPE fine-tuned with LoRA (Hu et al. 2022). As shown in Tab. 3, $\text{DAPE}^{\text{LoRA}}$ is more effective at preserving base class performance, whereas Meta-DAPE excels at detecting novel classes in low-shot scenarios. Notably, Meta-DAPE and $\text{DAPE}^{\text{LoRA}}$

Method	Backbone	#epochs	AP↑	AP_{50} ↑	AP_{75} ↑	AP_S ↑	AP_M ↑	AP_L ↑	Train Mem↓
Deformable-DETR (Zhu et al. 2021)	ResNet-50	12	45.4	65.0	49.1	27.2	49.6	61.0	3.8G
DINO (Zhang et al. 2022b)	ResNet-50	36	50.9	69.0	55.3	34.6	54.1	64.6	4.3G
H-DETR (Jia et al. 2023)	ResNet-50	36	50.0	68.3	54.4	32.9	52.7	65.3	4.7G
Stable-DINO (Liu et al. 2023)	ResNet-50	12	50.4	67.4	55.0	32.9	54.0	65.5	4.4G
Stable-DINO (Liu et al. 2023)	ResNet-50	24	51.5	68.5	56.3	35.2	54.7	66.5	4.4G
Salience-DETR (Hou et al. 2024a)	ResNet-50	12	50.0	67.7	54.2	33.3	54.4	64.4	4.4G
DDQ-DETR (Zhang et al. 2023)	ResNet-50	12	50.7	68.1	55.7	—	—	—	6.8G
EASE-DETR (Gao et al. 2024)	ResNet-50	12	50.8	48.9	55.3	34.1	54.2	65.1	
Relation-DETR (Hou et al. 2024b)	ResNet-50	12	51.7	69.1	56.3	36.1	55.6	66.1	7.2G
Relation-DETR (Hou et al. 2024b)	ResNet-50	24	52.1	69.7	56.6	36.1	56.0	66.5	7.2G
DAPE (ours)	ResNet-50	12	51.8	69.4	56.2	**36.2**	55.7	66.2	5.3G
DAPE (ours)	ResNet-50	24	**52.5**	**70.1**	**57.1**	35.9	**56.5**	**67.1**	5.3G

The memory cost is tested with 800×1333 input size, no gradient checkpointing. † denotes the 5-scale setting.

Table 1: Object Detection comparison with ResNet50 (IN-1K) backbone on COCO `val2017`.

Methods	Backbone	#epoch	AP^m ↑	AP^m_{50} ↑	AP^m_{75} ↑	AP^m_S ↑	AP^m_M ↑	AP^m_L ↑
MaskRCNN (He et al. 2017)	ResNet-50	36	37.1	58.3	39.9	18.4	39.8	52.9
HTC (Chen et al. 2019)	ResNet-50	36	39.7	61.4	43.1	22.6	42.2	50.6
Mask2Former (Cheng et al. 2022)	ResNet-50	50	38.7	59.8	41.2	18.2	41.5	59.8
MaskDINO (Li et al. 2023)	ResNet-50	12	41.4	62.9	44.6	21.1	44.2	61.4
DI-MaskDINO (Nan et al. 2024)	ResNet-50	12	42.3	-	-	22.0	44.8	62.8
Mask-DAPE	ResNet-50	12	**43.7**	**66.0**	**47.1**	**23.8**	**46.9**	**63.2**

Table 2: Instance Segmentation comparison with ResNet50 (IN-1K) backbone on COCO `val2017`.

shot	Method+Backbone		bAP	bAP_{50}	bAP_{75}	nAP	nAP_{50}	nAP_{75}
5	Def-DETR	r101				7.4	12.3	7.7
	Meta-DETR	r101				**15.4**	**25.0**	**15.8**
	FS-DETR	r50				10.9	20.7	10.8
	Meta-DAPE	r50	31.5	44.1	33.7	12.9	19.3	13.0
	$DAPE^{LoRA}$	r50	**39.8**	**56.6**	**43.9**	5.7	8.7	6.0
10	Def-DETR	r101				11.7	19.6	12.1
	Meta-DETR	r101				**19.0**	**30.5**	**19.7**
	Meta-DAPE	r50	32.8	45.6	35.2	18.0	26.0	18.9
	$DAPE^{LoRA}$	r50	**40.2**	**57.1**	**44.0**	15.1	22.5	16.2
30	Def-DETR	r101				16.3	27.2	16.7
	Meta-DETR	r101				22.2	35.0	22.8
	Meta-DAPE	r50	31.6	43.9	34.0	22.2	32.4	23.2
	$DAPE^{LoRA}$	r50	**42.3**	**58.8**	**46.1**	**24.7**	**36.5**	**25.7**

Table 3: Few-Shot Object Detection on COCO `val2017`.

with ResNet-50 even outperform some counterparts using ResNet-101 in 30-shot detection, demonstrating their effectiveness and generalization ability.

Ablation Study

We conduct ablation studies to verify the effectiveness of modules in DAPE based on ResNet-50 and $1\times$ schedule.

Effectiveness of Model Designs We select DINO (Zhang et al. 2022b) as baseline to evaluate the effectiveness of our proposed modules. As shown in Tab. 4, each component consistently improves performance, increasing AP from 49.9% to 51.8%. Notably, integrating SQS, LRPE, and one-to-many matching individually contributes gains of 0.7%, 1.0%, and 1.0% AP, respectively. The combination of all components achieves the best performance of 51.8% AP, demonstrating the effectiveness of the proposed modules.

SQS	LRPE	Matching	AP↑	AP_{50} ↑	AP_{75} ↑
			49.9	67.4	54.5
✓			50.6	68.6	55.1
	✓		50.9	68.5	55.7
		✓	50.9	68.2	55.1
	✓	✓	51.4	68.6	55.7
✓		✓	51.3	68.8	55.7
✓	✓		51.5	68.8	55.8
✓	✓	✓	51.8	69.4	56.2

Table 4: Ablation study on key components of DAPE.

Sampling Distribution Evaluation Figure 6, Figure 7 and Figure 8 provide qualitative comparisons of the sampling distributions, illustrating that SQS closely matches the spatial distribution of ground-truth boxes, while top-k sampling introduces a nonnegligible distribution shift.

Low-Rank Position Encoding Our proposed LRPE provides a memory-efficient solution for modeling interactions between box coordinates. Table 5 compares the performance of our LRPE with the 2D position relation module proposed in Relation-DETR (Hou et al. 2024b). With both Relation-

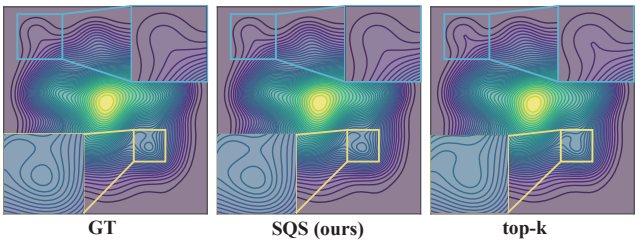

Figure 6: Comparison of 2D sampling distributions with and without our Shifted Query Sampler (SQS).

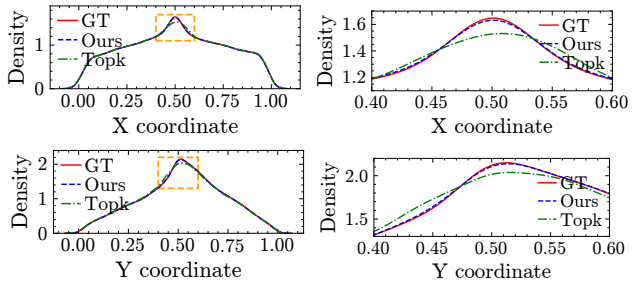

Figure 7: Comparison of coordinate distributions with and without our Shifted Query Sampler (SQS). The right column shows a magnified view of the orange rectangular region highlighted in the left column.

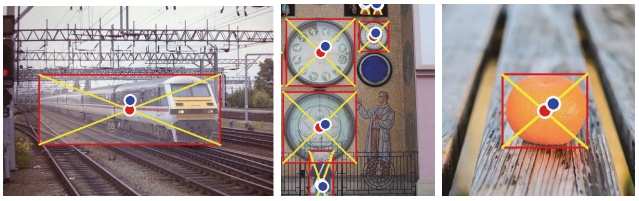

Figure 8: Qualitative comparison of sampling positions generated by our SQS (red) and top-k sampling (blue).

DETR and our DAPE as baselines, our LRPE achieves comparable performance to the 2D position relation while significantly reducing memory consumption, demonstrating its effectiveness and efficiency. Moreover, with the precise guidance provided by our SQS, LRPE achieves even better performance on DAPE than on Relation-DETR.

Effect of the Encoding Rank Table 6 investigates the effect of rank in LRPE for performance and memory usage. Due to linear complexity of LRPE, the memory usage increases by only about 1.8 MB for each increment in rank, which remains nearly negligible across different ranks. As can be seen, the performance of DAPE steadily improves as the rank r increases from 4 to 16 and then decreases slightly. The setting $r = 16$ achieves the best performance. Therefore, we adopt $r = 16$ as the default setting of DAPE.

Efficiency Evaluation Table 7 compares the efficiency of DAPE with several representative DETR methods. The FPS is measured on a single NVIDIA RTX 4090 GPU with 800×1333 input size. The training time was recorded for

Model	Encoding	AP	AP_{50}	AP_{75}	Mem
Relation-DETR	Relation	51.7	69.1	**56.3**	7.2G
Relation-DETR	LRPE	51.6	69.2	56.0	5.5G
DAPE	Relation	51.7	69.3	56.1	7.1G
DAPE	LRPE	**51.8**	**69.4**	56.2	**5.3G**

Table 5: Comparison of 2D position encoding.

rank	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Mem	FLOPs
4	51.5	69.0	56.0	36.1	55.1	65.8	5.3G	303G
8	51.6	69.6	56.0	35.1	55.4	65.9	5.3G	303G
16	51.8	69.7	56.2	35.8	55.5	66.0	5.3G	303G
32	51.7	69.3	56.4	35.8	55.6	66.3	5.3G	304G
64	51.9	69.5	56.6	35.9	55.7	66.4	5.3G	308G
128	52.0	69.7	56.8	36.3	55.6	65.9	5.4G	315G

Table 6: Ablation study on the rank of LRPE.

Models	GFLOPs	FPS(img/s)	Training time	AP
DAPE	303	13.7±0.09	38h	51.8
Relation-DETR	302	75.8±0.8	42h	51.7
DDQ-DETR	291	86.7±0.4	58h	50.7
DINO	290	69.6±0.8	35h	49.9

Table 7: Efficiency comparison for DAPE.

a single training schedule (1x) on the COCO 2017 dataset, using 2 NVIDIA RTX 4090 GPUs and batch size 10. From the results, DAPE demonstrates a superior balance between computational cost and accuracy. It achieves the highest AP (51.8%) while maintaining competitive latency (72.6 ms), outperforming Relation-DETR (Hou et al. 2024b) and DDQ-DETR (Zhang et al. 2023) in both speed and accuracy. Moreover, DAPE requires significantly less training time (38h) compared to Relation-DETR (42h) and DDQ-DETR (58h), highlighting its efficiency in training.

Conclusion

This paper presents a novel and unified framework for dense visual prediction, termed DAPE, which effectively breaks through the bottlenecks of content and position encoding in DETR frameworks. DAPE consists of a Shifted Query Sampler (SQS) to align the sampling distributions of content and position queries, and a Low-Rank Position Encoder (LRPE) to model the interactions between position embeddings in a memory-efficient manner. DAPE offers a general and extensible solution for different dense prediction tasks, including object detection, instance segmentation, and few-shot object detection. Extensive experiments validate that DAPE significantly outperforms existing approaches in terms of performance, convergence, and memory.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62327808 and the Fundamental Research Funds for Xi'an Jiaotong University under Grant xzy022024009.

References

Antonelli, S.; Avola, D.; Cinque, L.; Crisostomi, D.; Foresti, G. L.; Galasso, F.; Marini, M. R.; Mecca, A.; and Pannone, D. 2022. Few-shot object detection: A survey. *ACM Computing Surveys (CSUR)*, 54(11s): 1–37.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. 2019. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4974–4983.

Chen, Q.; Chen, X.; Wang, J.; Zhang, S.; Yao, K.; Feng, H.; Han, J.; Ding, E.; Zeng, G.; and Wang, J. 2023. Group detr: Fast detr training with group-wise one-to-many assignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6633–6642.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1290–1299.

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6569–6578.

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2023. CenterNet++ for object detection. *IEEE transactions on pattern analysis and machine intelligence*.

Fan, Q.; Zhuo, W.; Tang, C.-K.; and Tai, Y.-W. 2020. Few-shot object detection with attention-RPN and multi-relation detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4013–4022.

Gao, Y.; Sun, Y.; Ding, X.; Zhao, C.; and Liu, S. 2024. EASE-DETR: Easing the Competition among Object Queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17282–17291.

Gao, Z.; Wang, L.; Han, B.; and Guo, S. 2022. Adamixer: A fast-converging query-based object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5364–5373.

Girshick, R. 2015. Fast r-cnn. *arXiv preprint arXiv:1504.08083*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Hou, X.; Liu, M.; Zhang, S.; Wei, P.; and Chen, B. 2024a. Salience DETR: Enhancing Detection Transformer with Hierarchical Salience Filtering Refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17574–17583.

Hou, X.; Liu, M.; Zhang, S.; Wei, P.; Chen, B.; and Lan, X. 2024b. Relation DETR: Exploring Explicit Position Relation Prior for Object Detection. In *European conference on computer vision*. Springer.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; and Wei, Y. 2018. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3588–3597.

Jia, D.; Yuan, Y.; He, H.; Wu, X.; Yu, H.; Lin, W.; Sun, L.; Zhang, C.; and Hu, H. 2023. Detrs with hybrid matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19702–19712.

Kang, B.; Liu, Z.; Wang, X.; Yu, F.; Feng, J.; and Darrell, T. 2019. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8420–8429.

Law, H.; and Deng, J. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, 734–750.

Li, F.; Zhang, H.; Xu, H.; Liu, S.; Zhang, L.; Ni, L. M.; and Shum, H.-Y. 2023. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3041–3050.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; and Zhang, L. 2022. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. In *International Conference on Learning Representations*.

Liu, S.; Ren, T.; Chen, J.; Zeng, Z.; Zhang, H.; Li, F.; Li, H.; Huang, J.; Su, H.; Zhu, J.; et al. 2023. Detection transformer with stable matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6491–6500.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Nan, Z.; Xianghong, L.; Xiang, T.; and Dai, J. 2024. DI-MaskDINO: A Joint Object Detection and Instance Segmentation Model. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds.,

Advances in Neural Information Processing Systems, volume 37, 60359–60381. Curran Associates, Inc.

Ouyang-Zhang, J.; Cho, J. H.; Zhou, X.; and Krähenbühl, P. 2022. Nms strikes back. *arXiv preprint arXiv:2212.06137*.

Pu, Y.; Liang, W.; Hao, Y.; Yuan, Y.; Yang, Y.; Zhang, C.; Hu, H.; and Huang, G. 2024. Rank-DETR for high quality object detection. *Advances in Neural Information Processing Systems*, 36.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6): 1137–1149.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Viola, P.; and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, I–I. Ieee.

Viola, P.; and Jones, M. J. 2004. Robust real-time face detection. *International journal of computer vision*, 57: 137–154.

Wang, S.; Li, B. Z.; Khabsa, M.; Fang, H.; and Ma, H. 2020a. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Wang, X.; Huang, T.; Gonzalez, J.; Darrell, T.; and Yu, F. 2020b. Frustratingly Simple Few-Shot Object Detection. In *International Conference on Machine Learning*, 9919–9928. PMLR.

Wang, Y.; Zhang, X.; Yang, T.; and Sun, J. 2022. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 2567–2575.

Zhang, G.; Luo, Z.; Cui, K.; Lu, S.; and Xing, E. P. 2022a. Meta-detr: Image-level few-shot detection with inter-class correlation exploitation. *IEEE transactions on pattern analysis and machine intelligence*, 45(11): 12832–12843.

Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L. M.; and Shum, H.-Y. 2022b. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. arXiv:2203.03605.

Zhang, S.; Wang, X.; Wang, J.; Pang, J.; Lyu, C.; Zhang, W.; Luo, P.; and Chen, K. 2023. Dense distinct query for end-to-end object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7329–7338.

Zheng, D.; Dong, W.; Hu, H.; Chen, X.; and Wang, Y. 2023. Less is more: Focus attention for efficient detr. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6674–6683.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*.

Zong, Z.; Song, G.; and Liu, Y. 2023. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6748–6758.