# MLP-DINO: Category Modeling and Query Graphing with Deep MLP for Object Detection

**Guiping Cao**[1,2], **Wenjian Huang**[1], **Xiangyuan Lan**[2*], **Jianguo Zhang**[1,2*], **Dongmei Jiang**[2] and **Yaowei Wang**[2]

[1]Research Institute of Trustworthy Autonomous Systems and Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

[2]Peng Cheng Laboratory, Shenzhen, China

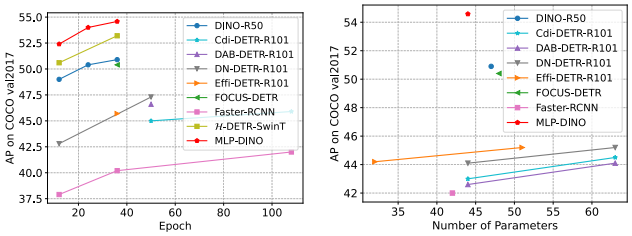
12131099@mail.sustech.edu.cn, {huangwj, zhangjg}@sustech.edu.cn, {lanxy, jiangdm, wangyw}@pcl.ac.cn

## Abstract

Popular *transformer-based* detectors detect objects in a one-to-one manner, where both the bounding box and category of each object are predicted only by the single query, leading to the *box-sensitive* category predictions. Additionally, the initialization of positional queries solely based on the predicted confidence scores or learnable embeddings neglects the significant *spatial interrelation* between different queries. This oversight leads to an *imbalanced* spatial distribution of queries (SDQ). In this paper, we propose a new **MLP-DINO** model to address these issues. Firstly, we present a new **Q**uery-**I**ndependent **C**ategory **S**upervision (QICS) approach for modeling categories information, decoupling the sensitive bounding box prediction process. Additionally, to further improve the category predictions, we introduce a deep MLP model into transformer-based detection framework to capture the *long-range* and *short-range* information simultaneously. Thirdly, to balance the SDQ, we design a novel **G**raph-based **Q**uery **S**election (GQS) method that distributes each query point in a discrete manner by graphing the spatial information of queries to cover a broader range of potential objects, significantly enhancing the hit-rate of queries. Experimental results on COCO indicate that our MLP-DINO achieves **54.6%** AP with only **44M** parameters under 36-epoch setting, greatly *outperforming* the DINO by **+3.7%** AP with *fewer* parameters and FLOPs. The source codes will be available at https://github.com/Med-Process/MLP-DINO.

## 1 Introduction

Object detection (OD) plays a crucial role in computer vision tasks by identifying the bounding boxes and categories of objects within images [Girshick, 2015; He *et al.*, 2017]. Over the years, convolution-based detectors [Girshick, 2015; Liu *et al.*, 2016; Lin *et al.*, 2020; Wang *et al.*, 2023] have made remarkable advancements in this field. However, those methods usually rely on hand-designed components and complex pipelines [Carion *et al.*, 2020], which imposes certain limitations on their performance and scalability.



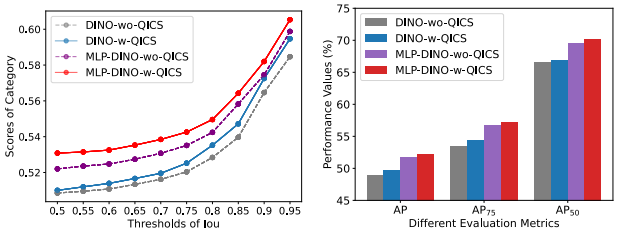

(a) Epoch-AP (b) Params-AP

Figure 1: Comparison of different models of AP w.r.t the training epochs and parameters on `val2017` of COCO. MLP-DINO gets the *best* performance under different training epoch settings.

Recently, transformer-based detectors simplify the pipeline of OD and get more attention in computer vision community. DEtection TRansformer (DETR) [Carion *et al.*, 2020] is a novel end-to-end detection framework that casts OD as a set prediction problem to match the predicted object and ground truth by using a bipartite matching method. DETR achieves comparable results with the classical state-of-the-art detectors while eliminating the need of hand-designed components.

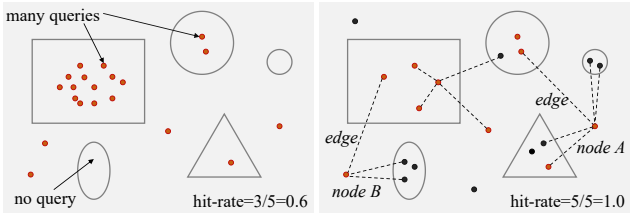
However, we argue that the DETR-like framework suffers from the *box-sensitive* category predictions and *imbalanced-query* distribution issues. On one hand, the bipartite matching process in DETR assigns each decoder query to a single object. The bounding box and category of this object is predicted simultaneously by the assigned query. This paradigm results in the category prediction are mutually influenced by the box prediction. As illustrated in Fig. 2 (a), *the prediction scores of the categories decrease as the accuracy of the box predictions decreases.* Here, we mark this issue as *the Box-Sensitivity in Category Prediction (**BS-ICP**)*. On the other hand, the existing model [Carion *et al.*, 2020; Yao *et al.*, 2021; Li *et al.*, 2022; Zhang *et al.*, 2022] select the positional queries from all tokens solely based on the predicted confidence scores or static learnable embeddings.

*Corresponding author.




(a) Category Score-Iou of Box (b) Performance-OD Metrics

Figure 2: The models with (-w-) QICS achieves both higher scores of categories prediction and boosted performance than that of without (-wo-) utilizing QICS on `val2017` of COCO dataset. The DINO and MLP-DINO model use ResNet50 and Strip-MLP-T-SWMP as the backbone, respectively.




(a) Imbalanced distribution (b) Balanced distribution

Figure 3: Example of different spatial distributions of 20 query points on 5 objects (4 classes marked with different shapes) within an image. (a) Queries are selected only by the predicted confidence score, with many query points (in red color) clustering in a single object (*e.g.* indicated by the rectangle). (b) By representing query points as nodes in a *graph*, we incorporate spatial information into the query selection process, getting updated distribution of query points (in black color) and enabling higher hit-rate of queries to objects, as presented in Table 5.

These approaches explicitly neglect the significant *spatial* information of each query, leading to an *imbalanced* spatial distribution of selected queries. Fig. 3 (a) illustrates that a considerable proportion of query points tends to cluster around a single object, resulting in lower query hit-rate for other objects. The *hit-rate* represents the proportion of the number of objects that are successfully hit by the query points, out of the total number of objects within the image.

**Query-Independent Global Categories Modeling.** For the issue of BS-ICP, our observations indicate that modeling the categories information globally would significantly boost the detection performance. Therefore, we propose the QICS approach, which contains an image-level classification layer and classification loss, *to reduce the dependency of category recognition on the box prediction process*. In particular, this approach uses the feature from backbone and encoder to predict all categories of objects to be detected. Thus, this category prediction is query independent and unrelated to each box predictions, enabling the model to identify all potential object categories within the image. This query independence method draws inspiration from the partitioned intelligence observed in brain organoid.

**Long-range and Short-range Information Modeling.** In the original DETR architecture, conventional CNNs, such as ResNet50 and ResNet101 [He *et al.*, 2016], are employed as the backbone to generate lower-resolution feature maps. Then, these feature maps are directly utilized by the subsequent encoder and decoder modules. Here, we make a *hypothesis* that incorporating a backbone model capable of aggregating both long and short range information can significantly enhance the detection performance across objects of varying scales. To verify our hypothesis and further improve the category predictions, we conduct a comprehensive research on CNN-based, Transformer-based, and MLP-based backbone architectures in DETR-like frameworks. Recently, MLP-based models [Liu *et al.*, 2022a] have shown effectiveness in capturing both long-range and short-range information, yielding promising performance in image classification tasks. We introduce a new MLP-based backbone model into the DINO framework, and the experiment results demonstrate their effectiveness. As presented in Fig. 2, the category predictions and detection performance are remarkably improved with MLP-based backbone model than CNN-based model.

**Query Graphing for Distribution Balanced.** Query selection is a significant process in DETR-like models [Yao *et al.*, 2021; Zhang *et al.*, 2022]. In original DETR, the query has no clear meaning and is simply initialized as learnable parameters. The follower-up works [Yao *et al.*, 2021; Liu *et al.*, 2022b] make efforts mainly on optimizing the query formulations, such as specifying the composition of the query [Meng *et al.*, 2021], initializing the query in static [Li *et al.*, 2022], dynamic [Zhu *et al.*, 2020] or mixed [Zhang *et al.*, 2022] ways, etc. However, these works neglect the spatial distribution of queries, leading to an *imbalance* distribution in selected queries. As illustrated in Fig. 3 (a), most selected queries cluster around a large object, leading to many redundant queries and a low query hit-rate on other objects.

We argue that incorporating the spatial information into the query selection process can effectively mitigate the query redundancy and improve the hit-rate of queries on objects. In this study, inspired by the functional connectivity networks observed among biological neurons, we propose a novel GQS method that utilizes the *relative distance* information between paired queries to select decoder queries with higher discreteness. This approach ensures that the selected queries are more representative and informative. Fig. 3 (b) illustrates a sparser distribution with higher hit-rate of queries to objects, balancing the selected queries distribution.

Our new proposed **MLP-DINO** detector is illustrated Fig. 4, which is the *first* attempt to integrate the deep MLP model into *transformer-based* detection framework. The performance comparison between MLP-DINO and other popular models is presented in Fig 1. Our contributions are as follows:

- We design a new MLP-DINO model by introducing the deep MLP model with our **S**haring **W**eight on **M**ini-**P**atch (**SWMP**) approach into DINO framework, modeling the long and short range information simultaneously, and propose a QICS method to decouple the box regression process and boost the category predictions.
- We propose a novel GQS method to balance the spa-

tial distribution of queries, reducing the redundancy of queries and improving the hit-rate of query points to objects, enhancing the model's performance.

- Extensive experiments indicate that MLP-DINO remarkably improves the performances of the DINO model. MLP-DINO achieves **54.6%** AP with only **44M** parameters under 36-epoch setting, higher than original DINO by **+3.7%** AP with *fewer* parameters and FLOPs.

## 2 Related Work

**Classical Detectors.** Over the past decade, convolution-based detectors have made tremendous advancements, primarily including anchor-based and anchor-free methods. Anchor-based methods, like Faster R-CNN [Girshick, 2015] and Mask R-CNN [He *et al.*, 2017], generate anchor boxes in the first stage, and then classify them and regress their coordinates relative to the anchor. Anchor-free methods are designed to reduce the dependency on predefined anchors by directly predicting the locations of bounding boxes or keypoints without the need for explicit anchor boxes, like CenterNet [Duan *et al.*, 2019], YOLOX [Ge *et al.*, 2021]. However, all those methods rely on handcrafted components such as Non-Maximum Suppression and complex pipelines, which significantly limits their performance [Carion *et al.*, 2020].

**Transformer-based Detectors.** DETR [Carion *et al.*, 2020] is the first *transformer-based* detector that directly predicts objects by utilizing a fixed small set of learned object queries. Each query is responsible for the prediction of the class label and bounding box coordinates of a single object. Therefore, how to initialize and select these queries has become crucial research directions in DETR-like models [Zhang *et al.*, 2022].

Many related works have focused on improving the *query selection* method. Instead of randomly initialing the query in original DETR, Conditional DETR [Meng *et al.*, 2021] decouples queries into positional and content queries, and defines the positional query as 2D coordinates to learn a spatial query for better matching with image features. Efficient DETR [Yao *et al.*, 2021] indicates that selecting Top-K dense prior features is better than random initialization of queries. DAB-DETR [Liu *et al.*, 2022b] constructs the positional query with 4D anchor box coordinates to join the object scales information, improving the query-to-feature similarity. DN-DETR [Li *et al.*, 2022] introduces a denoising query part to reconstruct the original boxes, accelerating the training process. Based on DN-DETR, DINO [Zhang *et al.*, 2022] designs a new query part for rejecting "no object" anchors, and proposes a mixed query selection method for initializing query better. Although these works enhance the query selection process and get improved performance, all of them ignores the issues of BS-ICP and the imbalanced spatial distribution of selected queries.

**Detection with Deep MLP Models.** Since the introduction of MLP-Mixer [Tolstikhin *et al.*, 2021] into the field of computer vision, many MLP-based variants models [Liu *et al.*, 2021a; Tang *et al.*, 2022a; Tang *et al.*, 2022b; Cao *et al.*, 2023] have made significant advancements in image classification tasks. As an attention-free model, deep MLP model offers advantages such as simple structure and low computational complexity, achieving comparable accuracy to other approaches. Researchers have also observed improved accuracy and robustness in object detection tasks by incorporating the deep MLP model into CNN-based detection frameworks, such as Mask R-CNN [He *et al.*, 2017] and Cascade Mask R-CNN [Cai and Vasconcelos, 2018].

However, the reliance on *fully-connected* (FC) layers in deep MLP models makes them highly *resolution sensitive* [Liu *et al.*, 2022a], as *the number of neurons in FC layer is related to the input size of the image*, which means the model cannot accept the input image with arbitrary size. This sensitivity poses a significant challenge that these models cannot apply to dense prediction tasks like object detection. Indeed, some deep MLP methods, such as CycleMLP [Chen *et al.*, 2021], Hire-MLP [Guo *et al.*, 2022], and Wave-MLP [Tang *et al.*, 2022b] have addressed this issue by employing techniques such as feature shifting or using smaller kernels within local windows. But, the method of feature-shifting operation relies on specific model designs and is not universally applicable to other MLP models. Using smaller kernels also limits the advantages of long-range information aggregation of MLP models. Moreover, their performance in downstream detection tasks is also limited. Consequently, it is valuable to address the resolution limitation of deep MLP models and extend their application within transformer-based frameworks.

## 3 Method

In this section, we first describe the overall architecture of our new MLP-DINO with deep MLP backbone model. Then, we introduce the QICS approach for the category modeling by decoupling the box regression process. Furthermore, we introduce a **G**raph-based **Q**uery **S**election (GQS) method to balance the SDQ. Finally, we propose the **S**haring **W**eight on **M**ini-**P**atch (**SWMP**) approach to accept input images with arbitrary size for MLP-based models.

### 3.1 Overall Architecture

Our MLP-DINO architecture, depicted in Fig. 4, extends the robust DINO framework and comprises three key components: a backbone for feature extraction, a multi-layer Transformer encoder for feature enhancement, and a multi-layer decoder with two feed-forward networks (FFN) for box prediction. The process begins with feeding an image of arbitrary size into the model. The backbone network extracts features from this image, resulting in *backbone features* represented as $b_1 \sim b_4$. Then, these backbone features are flattened before being fed into the transformer-encoder along with the positional embedding [Vaswani *et al.*, 2017] and get enhanced feature $e_1 \sim e_4$. This step serves to boost these backbone features by leveraging attention mechanisms to disentangle the different objects within the image.

Next, we introduce the QICS approach to address the BS-ICP issue and introduce the Strip-MLP [Cao *et al.*, 2023] model into detection framework to further boost the category predictions. Strip-MLP is a MLP-based model that aggregates *long-range* and *short-range* information simultaneously and efficiently improves the token interaction power,

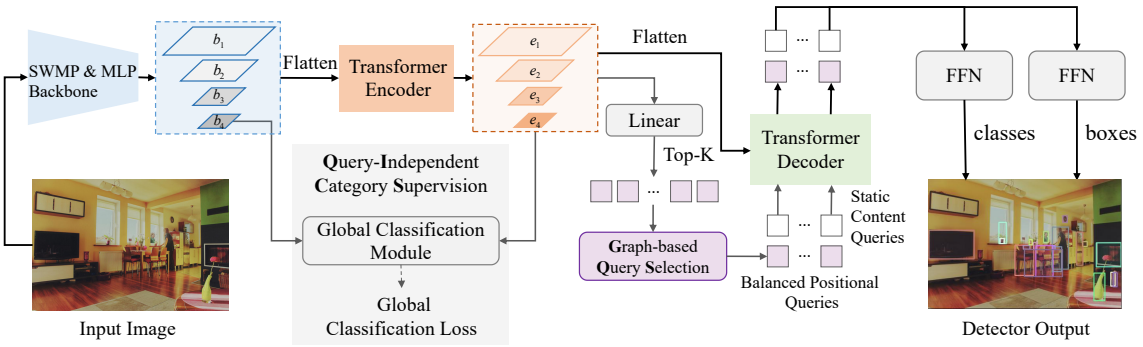



Figure 4: The overall architecture of the proposed MLP-DINO. $b_1 \sim b_4$ and $e_1 \sim e_4$ represent the output features from the deep MLP backbone and transformer-encoder, respectively. These features are at different levels and resolutions.

showing promising performance in classification tasks. However, the main limitation of the Strip-MLP is that it cannot accept images with arbitrary input sizes, as the number of neurons in MLP layers must align with the size of the input feature, which is still the *common challenge* in MLP-based models [Liu *et al.*, 2022a]. To overcome this limitation, we reformulate the Strip-MLP model with our SWMP approach to accept images with arbitrary size.

To address the issue of imbalanced query distribution, we propose a new graph-based GQS method on encoder features to select decoder queries in a discrete manner. In general, each object is directly detected by a single query. However, the number of queries is typically much larger than the number of objects within the image to be detected. The key insight behind this approach is that queries with higher hit-rate on objects tend to achieve better detection performance.

Finally, with selected positional queries, we update the positional bias of box predictions layer-by-layer with the deformable attention [Zhu *et al.*, 2020] in transformer-decoder. The detection results of boxes and corresponding classes are finally predicted by two FFN networks, respectively.

### 3.2 Query-Independent Category Supervision

In DETR-like models, a single decoder query predicts the bounding box and category simultaneously, leading the box-sensitive category prediction. That means the category prediction is affected by the box regression process. As illustrated in Fig. 2, as the IoU of the box regression decreases, the category score also decreases. To address this issue, we propose a QICS approach, which contains a Global Classification Module (GCM) and a Global Classification Loss (GCL), aiming to decouple the box regression process and identify all potential object categories within the image.

In DETR, the backbone and encoder features contain more global semantic information rather than query-specific instance information. As shown in Fig. 4, the query is only explicit involved in the decoder part of the network for the box prediction. Accordingly, to enable query-independent global category prediction, we apply QICS solely to the backbone feature and encoder features, which contain a wealth of category information of all objects. Specifically, the GCM in QICS achieves query-independent category prediction by utilizing global average pooling (GAP) [Lin *et al.*, 2013] to pool features from different levels with varying spatial dimensions. This process reduces the spatial dimension of the input features to 1, making them independent of any queries and effectively reducing the spatial complexity of the GCM module. Subsequently, we concatenate the pooled features from different levels along the channel dimension to generate multiple image-level category predictions.

Finally, a linear layer is adopted to predict all classes of the objects within the image. The GCM can be formulated as:

$$GCM(X) = Linear(Cat(GAP(X_1, X_2, X_3, X_4))) \quad (1)$$

where $X$ is the input feature, and $X_1 \sim X_4$ are the input features at four levels with different spatial resolutions. $Linear$ is a head of FC layer for predicting the categories of objects.

With those class prediction results, we achieve the query-independent category supervision by adopting the binary cross-entrophy (BCE) loss, which can be defined as:

$$l_{bce} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\sigma(x_i)) - (1-y_i)\log(1-\sigma(x_i)) \quad (2)$$

where $y_i$ is the binary label for each object category among $N$ images. A label of 1 indicates that the image contains an object belonging to a specific category, while a label of 0 denotes the absence of the target category within the image. $x_i$ is the prediction from the GCM. $\sigma(*)$ is the sigmoid function. With such a *light-weight* module (nearly 0.02M), our QICS approach contributes to decoupling the box regression process and identifying all potential object categories within the image from a global perspective.

### 3.3 Graph-based Query Selection

In DETR-like models, the performance suffers from the spatial distribution of the decoder queries, as each detection result is directly predicted from the corresponding query. Consequently, a series of research works have focused on the query formulations [Meng *et al.*, 2021; Zhu *et al.*, 2020; Liu *et al.*, 2022b; Zhang *et al.*, 2022], including the number and pattern of queries, the composition of queries, the initialization method of queries, and so on. However, these existing works ignores the spatial distribution information of

the queries, which significantly reduces the effectiveness of both the redundant queries and the model.

To deal with this concern, we propose a new GQS method to balance the distribution of positional queries by incorporating the spatial distribution information of queries into the Top-K method. To be specific, we design a *two-step* query selection method for positional queries, where $N_q$ is the number of queries to be selected for the decoder. The output features from the transformer-encoder have different resolutions at 4 levels, denoted as $F_e = \{e_1, e_2, e_3, e_4\}$. In the *first* step, these features are flattened and concatenated in the spatial dimension. Then, this feature is fed into a head of linear layer to obtain the class predictions. We sort these predictions based on their scores and get the Top-K indices of the queries for the second step, where $K = \lambda N_q$ ($\lambda \geqslant 1$). The first step can be formulated as:

$$Q_{N_q} = TopK(Argmax(Linear(Cat(e_1, e_2, e_3, e_4)))) \tag{3}$$

where $Q_{N_q}$ means the selected candidate queries. $Argmax(*)$ gets the highest scores among all class predictions. $Cat(*)$ denotes the concatenation operation.

In our *second* step, we leverage the relative positional information to select $N_q$ queries from the larger candidate pool of $\lambda N_q$ queries, where each query corresponds to a specific location (or point) within the 2D spatial space. Considering that each query point can be located anywhere on the 2D spatial space, we construct a *graph*, denoted by $G$, to describe the relative positional distribution of the query points, with each query's spatial coordinate serving as a node, as illustrated in Fig. 3 (b). The edges between nodes reflect the spatial relative distance between each paired queries. Specifically, we adopt the Euclidean Distance (ED) as the distance metric to measure the relative distance of all paired queries. Based on these distances, we further create an adjacency matrix, referred as $A \in \mathbb{R}^{\lambda N_q \times \lambda N_q}$ for the graph $G$, where each row and column of $A$ corresponds to a query point, and the distance value in each cell represents the weight of the edge between the corresponding pair of query points in $G$.

The objective of our approach is to distribute the query points in a discrete manner in the 2D spatial space of the feature, to cover a larger number of potential objects and increase the query hit-rate. Due to the varying sizes between images, it is challenging to only adopt the absolute distance information to accurately represent the distribution of queries in a uniform manner. To this end, we utilize the Coefficient of Variation (CV) [Abdi, 2010], also known as Normalized Root-Mean-Square Deviation (NRMSD), aiming to measure the discreteness of each query point. The CV denoted $C_V$ is a standardized statistical measure of the dispersion of distance distribution, which is defined as the ratio of the standard deviation $\sigma$ to the mean $\mu$:

$$C_V = \frac{\sigma}{\mu} \tag{4}$$

With the adjacency matrix $A$, we can calculate the standard deviation $A_\sigma$ and mean $A_\mu$ of positional distribution along the second dimension of $A$ for each query. By computing the ratio of $A_\sigma$ to $A_\mu$, we obtain the CV of $A$ denoted as $A_{C_V}$. The values in $A_{C_V}$ represent the level of discreteness in the distance distribution between query points. A *lower* value indicates a more concentrated distribution of *distances* between the current point and other points, implying that the current point is more scattered.

After obtaining the discreteness and classification prediction scores of each query, we design the criterion with these two matrix in Eq. 5 for query selection. In particular, we apply the Top-K strategy once again on this new criterion to select $N_q$ queries with higher scores from the pool of $\lambda N_q$ queries. Specifically, query points with higher confidence scores and larger discreteness will be selected as the final queries. This process can be formulated as:

$$Q = TopK(\sigma(Q_{N_q}) \odot (1 - A_{C_V})) \tag{5}$$

where $K = N_q$, and $\sigma$ means the *sigmoid* function. $\odot$ means the Hadamard product of the matrix.

### 3.4 SWMP for Fixed Image Size of Deep MLP

As described in the previous section of `Related Work` about deep MLP Models, their reliance on FC layers for token interaction presents a significant challenge that MLP-based models pre-trained on image classification datasets are not easily transferable to downstream dense prediction tasks, which significantly limits their development and application.

To address this challenge, we propose a new SWMP approach, which involves applying MLP layers with shared weights on mini-patches within cross-regions. Initially, we assume that a pre-trained MLP model is available for image classification with a fixed input size of $H_o \times W_o$. For the downstream task such as object detection, where input images have varying sizes of $H \times W$, we crop (or pad) the image into mini-patches with overlap, ensuring each patch has the *same size* as $H_o \times W_o$. The overlap size of $l_h$ and $l_w$ is determined adaptively by the remainder of $H$ and $W$ divided by $H_o$ and $W_o$, respectively. Afterwards, we utilize the shared MLP layer to process all these mini-patches. To restore those processed patches to their original size, we combine the overlapping regions of the patches with an average weight. The SWMP approach is a *general* method to transfer the MLP-based models into resolution-free models, making them suitable for a wider range of dense prediction tasks.

## 4 Experiments

### 4.1 Experiments Setup

The experiments are conducted on the benchmark dataset of COCO2017 [Lin *et al.*, 2014], which contains 118k training images of `train2017` and 5k validation images of `val2017`. We develop MLP-DINO model by reformulating the Strip-MLP [Cao *et al.*, 2023] with our SWMP method to integrate it into original DINO [Zhang *et al.*, 2022] framework, incorporating with the QICS and GQS to model the category and balance the query distribution. For a fair comparison, we follow the training recipe in DINO and train the model with AdamW [Loshchilov and Hutter, 2017] optimizer with weight decay of $1 \times 10^{-4}$ using Tesla V100 GPUs, with the batch size of 8. The number of decoder queries and denoising queries are 900 and 100 for all of our models, respectively. The default value of $\lambda$ is 2 for GQS. All MLP-DINO

| Model | Backbone | Epochs | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ | Params | GFLOPs |
|---|---|---|---|---|---|---|---|---|---|---|
| Faster-RCNN [Ren *et al.*, 2015] | ResNet50 | 108 | 42.0 | 62.4 | 44.2 | 20.5 | 45.8 | 61.1 | 42M | 180G |
| DETR [Carion *et al.*, 2020] | ResNet50 | 500 | 42.0 | 62.4 | 44.2 | 20.5 | 45.8 | 61.1 | 41M | 86G |
| DETR-DC5 [Carion *et al.*, 2020] | ResNet50 | 500 | 43.3 | 63.1 | 45.9 | 22.5 | 47.3 | 61.1 | 41M | 187G |
| Deformable-DETR [Zhu *et al.*, 2020] | ResNet50 | 50 | 46.2 | 65.2 | 50.0 | 28.8 | 49.2 | 61.7 | 40M | 173G |
| Efficient-DETR [Yao *et al.*, 2021] | ResNet50 | 36 | 44.2 | 62.2 | 48.0 | 28.4 | 47.5 | 56.6 | 32M | 159G |
| Conditional DETR [Meng *et al.*, 2021] | ResNet50 | 108 | 43.0 | 64.0 | 45.7 | 22.7 | 46.7 | 61.5 | 44M | 90G |
| DAB-DETR [Liu *et al.*, 2022b] | ResNet50 | 50 | 42.6 | 63.2 | 45.6 | 21.8 | 46.2 | 61.1 | 44M | 100G |
| DN-DETR [Li *et al.*, 2022] | ResNet50 | 50 | 44.1 | 64.4 | 46.7 | 22.9 | 48.0 | 63.4 | 44M | 94G |
| Focus-DETR [Zheng *et al.*, 2023] | ResNet50 | 36 | 50.4 | 68.5 | 55.0 | 34.0 | 53.5 | 64.4 | 48M | 154G |
| DETR [Carion *et al.*, 2020] | ResNet101 | 500 | 43.5 | 63.8 | 46.4 | 21.9 | 48.0 | 61.8 | 60M | 152G |
| Conditional DETR [Meng *et al.*, 2021] | ResNet101 | 108 | 44.5 | 65.6 | 47.5 | 23.6 | 48.4 | 63.6 | 63M | 156G |
| Efficient-DETR [Yao *et al.*, 2021] | ResNet101 | 36 | 45.2 | 63.7 | 48.8 | 28.8 | 49.1 | 59.0 | 51M | 239G |
| DAB-DETR [Liu *et al.*, 2022b] | ResNet101 | 50 | 44.1 | 64.7 | 47.2 | 24.1 | 48.2 | 62.9 | 63M | 179G |
| DN-DETR [Li *et al.*, 2022] | ResNet101 | 50 | 45.2 | 65.5 | 48.3 | 24.1 | 49.1 | 65.1 | 63M | 174G |
| DINO [Zhang *et al.*, 2022] | ResNet50 | 12 | 49.0 | 66.6 | 53.5 | 32.0 | 52.3 | 63.0 | 47M | 279G |
| Grounding DINO [Liu *et al.*, 2023] | ResNet50 | 12 | 48.1 | 65.8 | 52.3 | 30.4 | 51.3 | 62.3 | - | - |
| Co-DETR-4scale [Zong *et al.*, 2023] | ResNet50 | 12 | 49.5 | 67.6 | 54.3 | 32.4 | 52.7 | 63.7 | - | - |
| Co-DETR-5scale [Zong *et al.*, 2023] | ResNet50 | 12 | 52.1 | 69.4 | 57.1 | 35.4 | 55.4 | 65.9 | - | - |
| $\mathcal{H}$-Deformable-DETR [Jia *et al.*, 2023] | Swin-T | 12 | 50.6 | - | - | 33.4 | 53.7 | 65.9 | - | - |
| DINO [Ren *et al.*, 2023] | Swin-T | 12 | 51.3 | 69.0 | 56.0 | 34.5 | 54.4 | 66.0 | 48M | 278G |
| **DINO (ours)** | Strip-MLP-T-SWMP | 12 | 51.7 | 69.5 | 56.8 | 34.7 | 55.1 | 66.0 | 44M | 263G |
| **MLP-DINO (ours)** | ResNet50 | 12 | 50.1 | 67.5 | 54.6 | 33.2 | 53.1 | 64.9 | 47M | 279G |
| **MLP-DINO (ours)** | Swin-T | 12 | 52.2 | 69.8 | 57.1 | 34.3 | 55.3 | 67.1 | 48M | 280G |
| **MLP-DINO (ours)** | Strip-MLP-T-SWMP | 12 | **52.4** | **70.3** | **57.3** | **36.1** | **56.0** | **66.7** | 44M | 263G |
| DINO [Zhang *et al.*, 2022] | ResNet50 | 24 | 50.4 | 68.3 | 54.8 | 33.3 | 53.7 | 64.8 | 47M | 279G |
| **MLP-DINO (ours)** | ResNet50 | 24 | 51.1 | 68.7 | 55.9 | 34.6 | 54.3 | 65.3 | 47M | 279G |
| **MLP-DINO (ours)** | Swin-T | 24 | 54.0 | 71.7 | **59.0** | **36.8** | 57.1 | 68.7 | 48M | 280G |
| **MLP-DINO (ours)** | Strip-MLP-T-SWMP | 24 | **54.0** | **72.2** | 58.9 | 36.2 | **57.6** | **68.9** | 44M | 263G |
| DINO [Zhang *et al.*, 2022] | ResNet50 | 36 | 50.9 | 69.0 | 55.3 | 34.6 | 54.1 | 64.6 | 47M | 279G |
| $\mathcal{H}$-Deformable-DETR [Jia *et al.*, 2023] | Swin-T | 36 | 53.2 | - | - | 35.9 | 56.4 | 68.2 | - | - |
| **MLP-DINO (ours)** | ResNet50 | 36 | 51.5 | 69.2 | 56.2 | 35.6 | 54.6 | 65.8 | 47M | 279G |
| **MLP-DINO (ours)** | Swin-T | 36 | 54.3 | 72.2 | 59.4 | **38.2** | 57.4 | 69.1 | 48M | 280G |
| **MLP-DINO (ours)** | Strip-MLP-T-SWMP | 36 | **54.6 (+3.7)** | **72.9** | **59.5** | 37.4 | **58.4** | **69.9** | 44M | 263G |

Table 1: Comparison results of MLP-DINO with other popular detection models under different backbones and training epochs on `val2017` of COCO. Models of DINO and MLP-DINO adopt 4 scales of feature maps from the backbone network.

models adopt *4-scale* features from the backbone. All our models are trained for 12 epochs (1× training scheduler) in the ablation study, except for specially marked ones. The detection performance is measured by the standard average precision (AP) under different IoU thresholds and object scales.

## 4.2 Main Results

Our MLP-DINO enhances the DINO model by effectively modeling the category information and balancing the query distribution. Table 1 presents a comprehensive comparison of our MLP-DINO with other DETR-like detectors [Carion *et al.*, 2020; Zhu *et al.*, 2020; Yao *et al.*, 2021; Meng *et al.*, 2021; Liu *et al.*, 2022b; Li *et al.*, 2022; Zheng *et al.*, 2023; Jia *et al.*, 2023; Zhang *et al.*, 2022], as well as Faster R-CNN [Ren *et al.*, 2015]. MLP-DINO achieves the *best* performance on all evaluation metrics. In particular, under the 12-epoch setting, MLP-DINO achieves **52.4%** AP with *fewer* parameters and GFLOPs, *higher* than original DINO [Zhang *et al.*, 2022] by **+3.4%** AP. In addition, in both 24-epoch and 36-epoch settings, our MLP-DINO consistently *outperforms* all other methods, demonstrating superior performance.

## 4.3 Ablation Study

**Ablation on All Components of MLP-DINO.** In Table 2, we present the results of ablation studies, using DINO as the baseline model with the backbone of ResNet50. The incorporation of the Strip-MLP-T-SWMP backbone leads to a significant improvement of **+2.7%** AP compared to the baseline, highlighting the importance of modeling long and short range information in the backbone network. Fig. 2 illustrates that both the category predictions and detection performance are improved with the MLP model and QICS approach. Additionally, the models show an additional performance boost of **+0.6%** AP when QICS or GQS is applied individually.

| Eps | BackB | QICS | GQS | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|---|
| 12 | | | | 49.0 | 66.6 | 53.5 | 32.0 | 52.3 | 63.0 |
| | ✓ | | | 51.7 | 69.5 | 56.8 | 34.7 | 55.1 | 66.0 |
| | ✓ | ✓ | | 52.3 | 70.2 | 57.3 | 35.7 | 55.9 | 66.6 |
| | ✓ | | ✓ | 52.3 | 70.3 | 57.3 | 35.7 | 56.0 | 66.6 |
| | ✓ | ✓ | ✓ | 52.4 | 70.3 | 57.3 | 36.1 | 56.0 | 66.7 |

Table 2: Ablation results on all components of the MLP-DINO.

**General Effectiveness of QICS and GQS.** To verify the *general* effectiveness of the QICS and GQS approaches, we apply these two approach to different backbone architectures, including both CNN-based and Transformer-based models. Table 3 displays the ablation results, indicating that both QICS and GQS lead to the average improvement of **+0.7%/+0.8%** AP on ResNet50 and SwinT, respectively. These results demonstrate the general effectiveness of these approaches across different DINO models.

**Ablation on Different Backbones in DINO.** To verify the effectiveness of modeling the long-range and short-range information, we conduct experiments with different backbones in DINO, such as CNN-based, Transformer-based, and MLP-based models. Table 4 clearly demonstrates that the models incorporating *long-range* information, such as Swin-T [Liu

| QICS | GQS | BackB | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|
| | | R50 | 49.0 | 66.6 | 53.5 | 32.0 | 52.3 | 63.0 |
| ✓ | | R50 | 49.7 (+0.7) | 67.0 | 54.5 | 33.2 | 52.9 | 63.9 |
| | ✓ | R50 | 49.8 (+0.8) | 67.1 | 54.6 | 32.6 | 53.0 | 64.3 |
| | | Swin-T | 51.3 | 69.0 | 56.0 | 34.5 | 54.4 | 66.0 |
| ✓ | | Swin-T | 52.2 (+0.9) | 69.8 | 56.9 | 35.0 | 55.1 | 67.0 |
| | ✓ | Swin-T | 52.0 (+0.7) | 69.7 | 56.8 | 35.0 | 54.7 | 66.7 |

Table 3: The ablation results of the QICS and GQS on the DINO of CNN-based and Transformer-based backbones.

*et al.*, 2021b] and Strip-MLP-T, outperform ResNet50, which only models *short-range* information. Impressively, we find that Strip-MLP-T gets higher performance than ResNet50 and Swin-T by **+2.0%** and **+0.4%** AP, respectively, showing that the Strip-MLP model is more robust and efficient to aggregate features. The results highlight the advantage of incorporating both the *long-range* and *short-range* information, as it provides more robust feature and enhances the category predictions, as shown in Fig. 2.

| Backbone | Type | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| ResNet50 | $\mathcal{S}$ | 49.0 | 66.6 | 53.5 | 32.0 | 52.3 | 63.0 |
| Swin-T | $\mathcal{L}$ | 51.3 | 69.0 | 56.0 | 34.5 | 54.4 | 66.0 |
| Strip-MLP-T | $\mathcal{S}\&\mathcal{L}$ | **51.7** | **69.5** | **56.8** | **34.7** | **55.1** | **66.0** |

Table 4: The ablation comparison of different backbones in DINO. The type of $\mathcal{S}$ and $\mathcal{L}$ represent short-range and long-range information built by the model, respectively. The number of parameters and FLOPs of Strip-MLP-T model is fewer than ResNet50 and Swin-T.

**Effectiveness of GQS on Hit-Rate.** We conduct a statistical analysis of the hit-rate for each model with different backbones. In Table 5, all models with GQS *consistently* get *better* performance and *higher* hit-rate, indicating that GQS effectively leverages the *spatial distribution* of queries and thus improve the detection performance of the model.

| Model | BackB | GQS | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ | Hit-Rate |
|---|---|---|---|---|---|---|---|---|---|
| DINO | R50 | | 49.0 | 66.6 | 53.5 | 32.0 | 52.3 | 63.0 | 0.8118 |
| DINO | R50 | ✓ | **49.8** | **67.1** | **54.6** | **32.6** | **53.0** | **64.3** | **0.8154** |
| DINO | Swin-T | | 51.3 | 69.0 | 56.0 | 34.5 | 54.4 | 66.0 | 0.8252 |
| DINO | Swin-T | ✓ | **52.0** | **69.7** | **56.8** | **35.0** | **54.7** | **66.7** | **0.8299** |
| DINO | Wave-T | | 49.7 | 67.2 | 54.0 | **33.9** | 53.1 | 63.0 | 0.8005 |
| DINO | Wave-T | ✓ | **50.4** | **68.0** | **54.9** | 33.1 | **53.8** | **64.0** | **0.8117** |
| **MLP-DINO** | Strip-T | | 51.7 | 69.5 | 56.8 | 34.7 | 55.1 | 66.0 | 0.8316 |
| **MLP-DINO** | Strip-T | ✓ | **52.3** | **70.3** | **57.3** | **35.7** | **56.0** | **66.6** | **0.8342** |

Table 5: Ablations of GQS on different backbone models.

**Ablation of QICS on Different Features.** The QICS approach aims to decouple the box regression process and identify all potential object categories within the image. To show the different impacts of QICS on different features, we apply the QICS to the different part of backbone features of $b_1 \sim b_4$ and encoder features of $e_1 \sim e_4$. As presented in Table 6, the ablation study reveals that the *best* results are achieved when applying QICS only to the *high-level* feature $b_4$ and the *enhanced features* of $e_1 \sim e_4$, as the *low-level* features of $b_1 \sim b_3$ lack of necessary *semantic* category information. It would introduce additional noises and decrease accuracy when the QICS is applied to the low-level feature of $b_1 \sim b_3$. Therefore, we apply the QICS on the features of $b_4$ and $e_1 \sim e_4$ for other experiments of this work.

| Different Features | QICS | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| baseline | | 51.7 | 69.5 | 56.8 | 34.7 | 55.1 | 66.0 |
| $b_1 \sim b_4$ | ✓ | 51.7 | 69.6 | 56.7 | 35.0 | 55.5 | 65.8 |
| $b_4$ | ✓ | 51.8 | 69.6 | 56.7 | 34.6 | 55.2 | 66.2 |
| $e_1 \sim e_4$ | ✓ | 52.2 | 70.0 | 57.2 | 35.2 | **56.1** | **66.8** |
| $b_1 \sim b_4$ and $e_1 \sim e_4$ | ✓ | 52.1 | 69.9 | 57.0 | 33.9 | 55.8 | 66.7 |
| $b_4$ and $e_1 \sim e_4$ | ✓ | **52.3** | **70.2** | **57.3** | **35.7** | 55.9 | 66.6 |

Table 6: Ablation results of applying QICS to different part of backbone features and encoder features of MLP-DINO.

**Ablation of the Queries Pool Size for GQS.** In the first step of GQS method, the hyper-parameter of $\lambda$ determines the size of queries candidate pool. A smaller value, such as $\lambda = 1$, would result in the method degrading to the original Top-K approach. On the other hand, a larger value of $\lambda$ indicating more candidate queries would introduce additional noise and complicate the training and optimization process. Consequently, choosing an appropriate value for $\lambda$ is crucial to enhance the model's performance. We conduct additional experiments using various $\lambda$ to identify the optimal values. Table 7 displays the results, indicating that $\lambda = 2$ achieves the best performance for the GQS method. Therefore, we set $\lambda = 2$ for the GQS method in our MLP-DINO models.

| $\lambda$ | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|
| 1 | 51.7 | 69.5 | 56.8 | 34.7 | 55.1 | 66.0 |
| 2 | **52.3** | **70.3** | **57.3** | **35.7** | **56.0** | **66.6** |
| 3 | 51.9 | 69.4 | 56.7 | 34.6 | 55.7 | 66.2 |

Table 7: The ablation results of different queries candidate pool size of $\lambda$ for GQS. The backbone model is Strip-MLP-SWMP.

## 5 Conclusion

This paper introduces a novel MLP-DINO model, which is the *first* attempt to integrate deep MLP models into *transformer-based* detection framework, leveraging the strengths of both models. We emphasize the crucial roles of both category modeling and balanced query distribution in DETR-like models. To address this, we decouple the category prediction process by a novel QICS approach and further enhance the category predictions by introducing the deep MLP models into DINO with our general SWMP approach, enabling the MLP models to accept the input image of any size and greatly extending their application to downstream dense prediction tasks. In addition, we design a new GQS method to balance the query distribution, leveraging the spatial information of queries into query selection process. Our extensive experimental results demonstrate the effectiveness and potential of MLP-DINO in detection task. We hope that our results can spark further research based on MLP-DINO in computer vision community, such as the segmentation and other vision tasks.

## Acknowledgments

This work is supported by National Key Research and Development Program of China (2021YFF1200800); Peng Cheng Laboratory Research Project (PCL2023A08); Shenzhen International Research Cooperation Project (Grant No. GJHZ20220913142611021).

## References

[Abdi, 2010] Hervé Abdi. Coefficient of variation. *Encyclopedia of research design*, 1(5), 2010.

[Cai and Vasconcelos, 2018] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[Cao *et al.*, 2023] Guiping Cao, Shengda Luo, Wenjian Huang, Xiangyuan Lan, Dongmei Jiang, Yaowei Wang, and Jianguo Zhang. Strip-mlp: Efficient token interaction for vision mlp. *arXiv preprint arXiv:2307.11458*, 2023.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[Chen *et al.*, 2021] Shoufa Chen, Enze Xie, Chongjian Ge, Runjian Chen, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. *arXiv preprint arXiv:2107.10224*, 2021.

[Duan *et al.*, 2019] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019.

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

[Girshick, 2015] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[Guo *et al.*, 2022] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 826–836, 2022.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[Jia *et al.*, 2023] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19702–19712, 2023.

[Li *et al.*, 2022] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022.

[Lin *et al.*, 2013] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[Lin *et al.*, 2014] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[Lin *et al.*, 2020] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 42(02):318–327, 2020.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[Liu *et al.*, 2021a] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021.

[Liu *et al.*, 2021b] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[Liu *et al.*, 2022a] Ruiyang Liu, Yinghui Li, Linmi Tao, Dun Liang, and Hai-Tao Zheng. Are we ready for a new paradigm shift? a survey on visual deep mlp. *Patterns*, 3(7), 2022.

[Liu *et al.*, 2022b] 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. *arXiv preprint arXiv:2201.12329*, 2022.

[Liu *et al.*, 2023] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[Meng *et al.*, 2021] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[Ren *et al.*, 2023] Tianhe Ren, Shilong Liu, Feng Li, Hao Zhang, Ailing Zeng, Jie Yang, Xingyu Liao, Ding Jia, Hongyang Li, He Cao, et al. detrex: Benchmarking detection transformers. *arXiv preprint arXiv:2306.07265*, 2023.

[Tang *et al.*, 2022a] Chuanxin Tang, Yucheng Zhao, Guangting Wang, Chong Luo, Wenxuan Xie, and Wenjun Zeng. Sparse mlp for image recognition: Is self-attention really necessary? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2344–2351, 2022.

[Tang *et al.*, 2022b] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10935–10944, 2022.

[Tolstikhin *et al.*, 2021] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2023] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[Yao *et al.*, 2021] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021.

[Zhang *et al.*, 2022] 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. *arXiv preprint arXiv:2203.03605*, 2022.

[Zheng *et al.*, 2023] Dehua Zheng, Wenhui Dong, Hailin Hu, Xinghao Chen, and Yunhe Wang. Less is more: Focus attention for efficient detr. *arXiv preprint arXiv:2307.12612*, 2023.

[Zhu *et al.*, 2020] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[Zong *et al.*, 2023] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023.