

# Group DETR: Fast DETR Training with Group-Wise One-to-Many Assignment

Qiang Chen<sup>1\*</sup>, Xiaokang Chen<sup>2\*</sup>, Jian Wang<sup>1</sup>, Shan Zhang<sup>3</sup>  
Kun Yao<sup>1</sup>, Haocheng Feng<sup>1</sup>, Junyu Han<sup>1</sup>, Errui Ding<sup>1</sup>, Gang Zeng<sup>2</sup>, Jingdong Wang<sup>1†</sup>  
<sup>1</sup>Baidu VIS

<sup>2</sup>Key Lab. of Machine Perception (MoE), School of IST, Peking University

<sup>3</sup>Australian National University

{chenqiang13, wangjian33}@baidu.com

{fenghaocheng, hanjunyu, dingerrui, wangjingdong}@baidu.com

{pkucxk, gang.zeng}@pku.edu.cn, shan.zhang@anu.edu.au

## Abstract

Detection transformer (DETR) relies on one-to-one assignment, assigning one ground-truth object to one prediction, for end-to-end detection without NMS post-processing. It is known that one-to-many assignment, assigning one ground-truth object to multiple predictions, succeeds in detection methods such as Faster R-CNN and FCOS. While the naive one-to-many assignment does not work for DETR, and it remains challenging to apply one-to-many assignment for DETR training. In this paper, we introduce Group DETR, a simple yet efficient DETR training approach that introduces a group-wise way for one-to-many assignment. This approach involves using multiple groups of object queries, conducting one-to-one assignment within each group, and performing decoder self-attention separately. It resembles data augmentation with automatically-learned object query augmentation. It is also equivalent to simultaneously training parameter-sharing networks of the same architecture, introducing more supervision and thus improving DETR training. The inference process is the same as DETR trained normally and only needs one group of queries without any architecture modification. Group DETR is versatile and is applicable to various DETR variants. The experiments show that Group DETR significantly speeds up the training convergence and improves the performance of various DETR-based models. Code will be available at <https://github.com/Atten4Vis/GroupDETR>.

## 1. Introduction

Detection Transformer (DETR) [2] conducts end-to-end object detection without the need of many hand-

\*Equal contribution.

†Corresponding author.

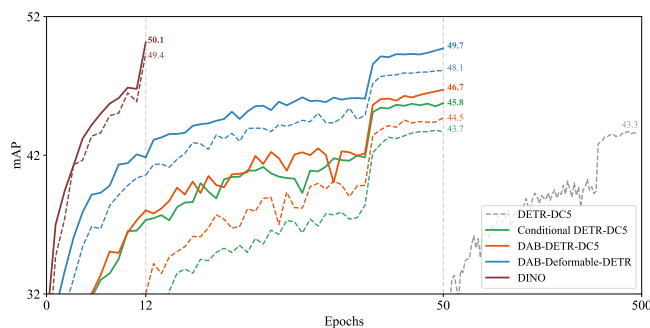


Figure 1. **Group DETR accelerates the training process for DETR variants.** The training convergence curves are obtained on COCO val2017 [24] with ResNet-50 [13]. Dashed and bold curves correspond to the baseline models and the Group DETR counterparts. Best viewed in color.

crafted components, such as non-maximum suppression (NMS) [14] and anchor generation [33, 23, 32]. The architecture consists of a CNN [13] and transformer encoder [37], and a transformer decoder that consists of self-attention, cross-attention and FFNs, followed by class and box prediction FFNs. During training, one-to-one assignment, where one ground-truth object is assigned to one single prediction, is applied for learning to only promote the predictions assigned to ground-truth objects, and demote the duplicate predictions.

This work explores the solutions to accelerate the DETR training process. Previous solutions contain two main lines. The one line is to modify *cross-attention* so that informative image regions are selected for effectively and efficiently collecting the information from image features. Example methods include sparse sampling, through deformable attention [47], and spatial modulations with modifying object queries [8, 30, 4, 40, 43, 25, 9]. The other line is to stabilize *one-to-one assignment* during training, e.g., feeding ground-truth bounding boxes with noises into transformer

decoder [20, 44].

We are interested in the second line. Instead of focusing on stabilizing the assignment like DN-DETR [20], we study the assignment scheme for efficient DETR training from a new perspective: introducing more supervision. It has been proven that assigning one ground-truth object to multiple predictions, i.e., one-to-many assignment, is successful in traditional object detection methods, e.g., Faster R-CNN [33] and FCOS [36] with more anchors and pixels assigned to one ground-truth object. Unfortunately, naive one-to-many assignment does not work for DETR training. It remains a challenge to apply one-to-many assignment to DETR training.

We present a simple yet efficient DETR training approach that uses a group-wise way for one-to-many assignment, called Group DETR. Our approach is based on that end-to-end detection with successful removal of NMS post-processing for DETR comes from the joint effect of two components [2, 30]: decoder self-attention, which collects the information of other predictions, and one-to-one assignment, which expects to learn to score one prediction higher and other duplicate predictions lower for one ground-truth object.

Our approach adopts  $K$  groups of object queries, and introduces *group-wise one-to-many assignment*. This assignment scheme conducts one-to-one assignment within each group of object queries, resulting in that one ground-truth object is assigned to multiple predictions. It is encouraged that the prediction assigned to the ground-truth object gets a high score, and other duplicate predictions from the same group of queries get low scores. In other words, the predictions make competition within each group. Thus, our approach uses *separate self-attention*, i.e., self-attention is done for each group separately, eliminating the influence of predictions from other groups and easing DETR training. Regarding inference, it is the same as DETR trained normally, and only needs a single group of object queries.

The resulting architecture is equivalent to DETR with a group of parallel decoders, illustrated in Figure 2 (a). During training, the parallel decoders boost each other through sharing decoder parameters and using different object queries. On the other hand, using more groups of object queries resembles data augmentation, and behaves as query augmentation. It introduces more supervision and improve the decoder training. In addition, it is empirically observed that the encoder training is also improved, presumably with the help of the improved decoder.

Group DETR is versatile and is applicable to various DETR variants. Extensive experiments demonstrates that our approach is effective in achieving fast training convergence, shown in Figure 1. Group DETR obtains consistent improvements on various DETR-based methods [30, 25, 20, 44]. For instance, Group DETR significantly improves

Conditional DETR-C5 by 5.0 mAP with 12-epoch training on COCO [24]. The non-trivial improvements hold when we adopt longer training schedules (e.g., 36 epochs and 50 epochs). Furthermore, Group DETR outperforms baseline methods for multi-view 3D object detection [26, 27] and instance segmentation [5].

## 2. Background

**DETR Architecture.** DETR [2] is composed of an encoder, a transformer decoder, and object class and box position predictors. The encoder takes an image  $\mathbf{I}$  as input, and outputs the image feature  $\mathbf{X}$ ,

$$\text{Encoder}(\mathbf{I}) \rightarrow \mathbf{X}. \quad (1)$$

The decoder receives the image feature  $\mathbf{X}$  and the *object queries*, denoted by a matrix  $\mathbf{Q} (= [\mathbf{q}_1 \mathbf{q}_2 \dots \mathbf{q}_N])$  as input, and outputs the embeddings  $\tilde{\mathbf{Q}}$ , followed by the predictors with the output denoted by  $\mathbf{Y} (= [y_1 y_2 \dots y_N])$ ,

$$\text{Decoder}(\mathbf{X}, \mathbf{Q}) \rightarrow \tilde{\mathbf{Q}}, \quad \text{Predictor}(\tilde{\mathbf{Q}}) \rightarrow \mathbf{Y}. \quad (2)$$

The decoder is a sequence of multiple layers. Each layer includes: (i) self-attention over object queries, which performs interactions among queries for collecting the information about duplicate detection; (ii) cross-attention between queries and image features, which collects the information from image features that is useful for object detection; (iii) feed-forward network that processes the queries separately to benefit object detection.

**DETR Training.** The predictions during DETR training are in the set form, and have no correspondence to the ground-truth objects. DETR uses one-to-one assignment, i.e., one ground-truth object is assigned to one predictions and vice versa, through building a bipartite matching between the predictions and the ground-truth objects:

$$(\mathbf{y}_{\sigma(1)}, \bar{\mathbf{y}}_1), (\mathbf{y}_{\sigma(2)}, \bar{\mathbf{y}}_2), \dots, (\mathbf{y}_{\sigma(N)}, \bar{\mathbf{y}}_N). \quad (3)$$

Here,  $\sigma(\cdot)$  is the optimal permutation of  $N$  indices, and  $[\bar{\mathbf{y}}_1 \bar{\mathbf{y}}_2, \dots \bar{\mathbf{y}}_N] = \bar{\mathbf{Y}}$  corresponds to ground truth. The loss is then formulated as below:

$$\mathcal{L} = \sum_{n=1}^N \ell(\mathbf{y}_{\sigma(n)}, \bar{\mathbf{y}}_n), \quad (4)$$

where  $\ell(\cdot)$  is a combination of the classification loss and the box regression loss between the ground-truth object  $\bar{\mathbf{y}}$  and the prediction  $\mathbf{y}$  [2, 47, 30].

Optimization with one-to-one assignment aims to score the predictions for promoting one prediction for one ground-truth object, and demoting duplicate predictions. Such scoring needs the comparison of one prediction with other predictions, and the information of other predictions is provided from decoder self-attention over queries. The

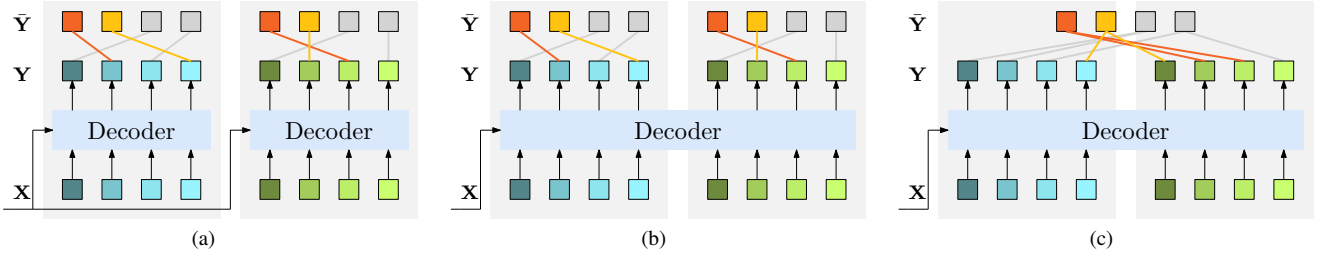


Figure 2. **Architecture illustration.** (a) Our Group DETR: group-wise one-to-many assignment and separate self-attention, architecturally equivalent to parallel decoder. (b) Group-wise one-to-many assignment only. (c) Naive one-to-many assignment. We use two groups of 4 object queries as an example.  $\mathbf{X}$ : image features;  $\mathbf{Y}$ : predictions;  $\bar{\mathbf{Y}}$ : ground-truth objects, where two color boxes mean two objects and two gray boxes mean dummy objects (no objects). The color lines between  $\mathbf{Y}$  and  $\bar{\mathbf{Y}}$  correspond to the assignment for ground-truth objects, and the gray lines for dummy objects. For clarity, the predictors are not explicitly included.

two designs, *one-to-one assignment and self-attention over object queries*, are critical for end-to-end detection without the need of the post-processing NMS.

**One-to-many assignment for non-end-to-end detection.** One-to-many assignment is successfully adopted for introducing more supervision to non-end-to-end detection training, such as Faster R-CNN [33], FCOS [36], and so on [12, 23, 32, 45, 10, 3, 11]. One ground-truth object is assigned to multiple anchors or multiple pixels. During inference, a post-processing NMS is conducted for duplicate detection removal.

### 3. Group DETR

#### 3.1. Algorithm

**Naive one-to-many assignment.** We start from a naive way for one-to-many assignment depicted in Figure 2 (c). We replace one-to-one assignment with one-to-many assignment: assign one ground-truth object to multiple predictions. It does not work and the performance is much low. The reason is that the model is trained to output multiple predictions for one ground-truth object, and lacks the scoring mechanism to promote one single prediction and demote duplicate predictions for one ground-truth object.

**Group-wise one-to-many assignment.** We adopt the multi-group object query mechanism: form the initial  $N$  queries as the primary group and introduce more  $(K - 1)$  groups of  $N$  queries, totally  $K$  groups,  $\{\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_K\}$ . Accordingly, we have  $K$  groups of predictions,  $\{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_K\}$ . We perform one-to-one assignment for each group, and find a bipartite matching  $\sigma_k(\cdot)$ , between each group of predictions and the ground-truth objects ( $\mathbf{Y}_k, \bar{\mathbf{Y}}$ ). This results in that only one prediction for one ground-truth object is expected to score higher, and duplicate predictions is expected to score lower within one group other than within all the groups.

**Separate self-attention.** One-to-one assignment in one group means that the prediction assigned to one ground-

#### Algorithm 1 Pseudocode of one Group Decoder Layer

```

# SA: Self-Attention in the decoder layer
# CA: Cross-Attention in the decoder layer
# FFN: FFN in the decoder layer
# X: output image features of the encoder
# Q: object queries, with size (KxN, B, C)
# N, K, B, C: object query number, group number,
    batch size, feature dimension

# group decoder
if training:
    # split object queries to K groups
    Q_list = Q.split(N, dim=0) # a list of K tensors
    parallel_Q = cat(Q_list, dim=1) # (N, KxB, C)

    # parallel self-attention
    out = SA(parallel_Q) # (N, KxB, C)
    # concat all groups: (KxN, B, C)
    out = cat(out.split(B, dim=1), dim=0)

    # cross-attention and ffn
    out = FFN(CA(out, X))
else:
    # in inference, only one group is kept
    Q = Q[:N] # (N, B, C)

    # self-attention, cross-attention, and ffn
    out = SA(Q)
    out = FFN(CA(out, X))

```

truth object is superior to other predictions within the same group. This implies that we only need to collect the information of the predictions only from the same group, rather than from all the groups. Thus we perform self-attention (abbreviated as SA) over queries for each group separately:

$$SA(\mathbf{Q}_1), SA(\mathbf{Q}_2), \dots, SA(\mathbf{Q}_K). \quad (5)$$

**Training architecture.** The resulting architecture for training is very simple: the encoder keeps the same, and the decoder contains  $K$  separate parallel decoders as shown in Figure 2 (a):

$$\begin{aligned}
& \text{Decoder}(\mathbf{X}, \mathbf{Q}_1) \rightarrow \mathbf{Q}_1, \text{ Predictor}(\mathbf{Q}_1) \rightarrow \mathbf{Y}_1, \\
& \text{Decoder}(\mathbf{X}, \mathbf{Q}_2) \rightarrow \mathbf{Q}_2, \text{ Predictor}(\mathbf{Q}_2) \rightarrow \mathbf{Y}_2, \\
& \quad \dots \\
& \text{Decoder}(\mathbf{X}, \mathbf{Q}_K) \rightarrow \mathbf{Q}_K, \text{ Predictor}(\mathbf{Q}_K) \rightarrow \mathbf{Y}_K. \quad (6)
\end{aligned}$$

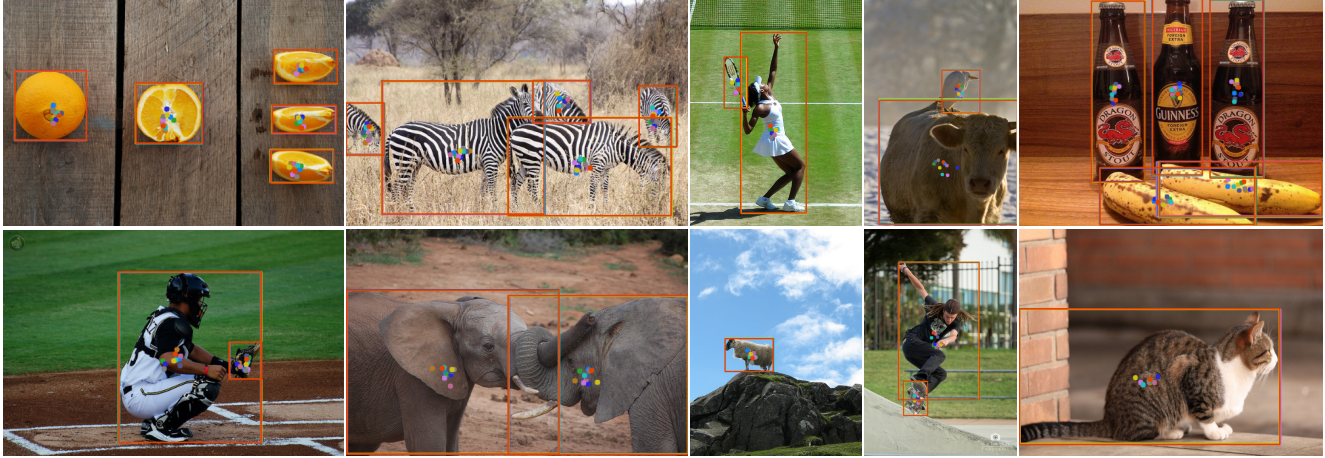


Figure 3. **Illustrating object queries.** The predicted boxes and reference points corresponding to object queries in different groups for the same ground-truth object are plotted in different colors for one group. It can be seen that these queries are spatially close and can be viewed as an augmentation of other queries. The results are from Group DETR over Conditional DETR-R50 [30]. The predicted boxes and reference points may overlap. Best view in color and zoom in.

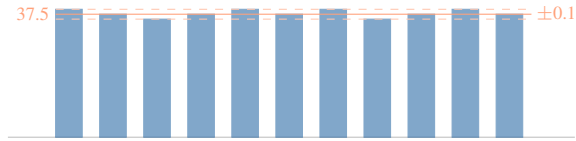


Figure 4. **The performance across groups of queries are similar.** Only a  $\pm 0.1$  mAP is observed over the median (37.5 mAP). The mAP scores over the COCO *val2017* are reported by a 12-epoch trained Conditional DETR-R50 with Group DETR.

Here, the parameters of the decoder and the predictor for the  $K$  groups are shared. Decoder separation and parallelism are feasible in that there is no interaction among queries for the other two operations, cross-attention and FFN. Our approach is called Group Decoder. In model inference, the process is the same as DETR trained normally and only needs one group of queries without any architecture modification. The pseudo-code is shown in Algorithm 1.

**Loss function.** The loss is an aggregation of  $K$  losses, each for one decoder. It is written as follows,

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_k = \frac{1}{K} \sum_{k=1}^K \sum_{n=1}^N \ell(\mathbf{y}_{\sigma_k(n)}, \bar{\mathbf{y}}_{kn}), \quad (7)$$

where  $\sigma_k(\cdot)$  is the optimal permutation of  $N$  indices for the  $k$ th decoder.

### 3.2. Analysis

**Explanation with parameter-shared models.** We discuss Group DETR from the perspective of training multiple models with parameter sharing. Training with Group DETR can be regarded as simultaneously training  $K$  DETR models, which share the parameters of the encoder, the decoder, and the predictor, and only differ in the initialization

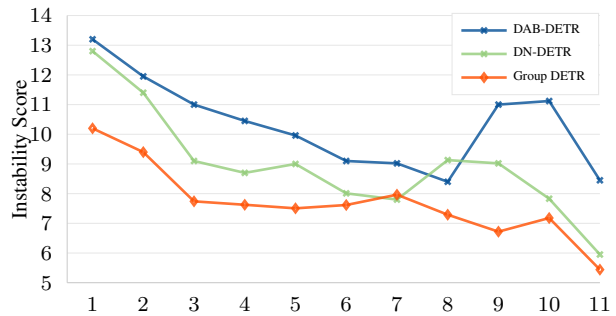


Figure 5. **More stable assignment.** The  $x$ -axis corresponds to #epoch, and the  $y$ -axis corresponds to instability score (the score is introduced by DN-DETR [20], the lower the instability score, the more stable the label assignment) over COCO *val2017*. One can see that the assignment in Group DETR is more stable than DN-DETR and its baseline DAB-DETR.

of object queries. This leads to the shared parameters receive more back-propagated gradients. Thus, these parameters are better trained and accordingly the training process converges faster.

As a side benefit, we observe that Group DETR makes the assignment more stable, as shown in Figure 5. We speculate that the stability is because the improved network leads to more reliable predictions, and thus the assignment quality is better.

**Explanation with object query augmentation.** The multi-group object query mechanism introduces additional  $(K - 1)$  group of queries, which can be regarded as an augmentation of the primary group of queries. This is empirically illustrated in Figure 3. The reference points predicting the same objects are spatially close, and thus the corresponding object queries are similar. This may suggest that the multi-



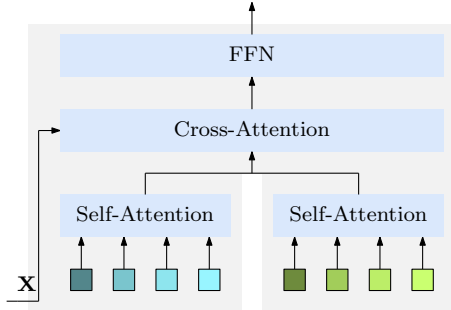


Figure 6. The parallel decoders in Group DETR are efficiently implemented as parallel self-attention, cross attention and FFN.

group object query mechanism resembles data augmentation, and at each iteration, more automatically-learned augmented queries are included, which equivalently introduces more supervision for decoder training. The results in Figure 4 empirically suggest that different groups of augmented queries lead to similar results.

The point about more supervision is also observed from the comparison between Equation 6 (for training with Group DETR) and Equation 2 (for normal DETR training). Group DETR training includes  $K$  pairs of image feature and object query group  $\{(\mathbf{X}, \mathbf{Q}_1), (\mathbf{X}, \mathbf{Q}_2), \dots, (\mathbf{X}, \mathbf{Q}_K)\}$ , and thus the loss contains more components as shown in Equation 7.

Table 1. Illustrating that training with Group DETR improves both encoder and decoder. The encoder, including CNN and transformer encoders, is initialized from a trained Conditional DETR-R50 [30] with 50 epochs and the decoder is random initialized. (a) (Fixed, Single) = the encoder is not retrained, and the decoder is trained normally without using Group DETR. (b) (Fixed, Group) = the encoder is not retrained, and the decoder is with Group DETR. (c) (Group, Group) = the encoder and the decoder are trained with Group DETR. All the results are got through training with 50 epochs. (c) > (b) implies that Group DETR also improves the encoder training.

	Encoder	Decoder	mAP	$AP_s$	$AP_m$	$AP_l$
(a)	Fixed	Single	40.6*	20.2	44.0	59.3
(b)	Fixed	Group	41.5	21.2	45.0	60.2
(c)	Group	Group	42.9*	22.2	46.6	61.6

\*: Training Conditional DETR with a trained encoder gives slightly lower performances than the one trained regularly, even though we train all components. New hyper-parameters may need to get better results.

**Encoder training improvement.** The additional supervision introduces more box regression and classification supervision from more queries assigned to each ground-truth object. The gradients with more supervision are also back-propagated from the decoder to the encoder. It is presumable that the encoder also gets benefit, verified by the empirical results in Table 1.

**Computation and memory complexity.** Group DETR uses more decoders during training. It is expected that

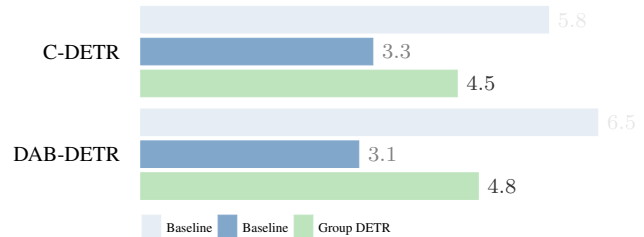


Figure 7. Baseline models vs their Group DETR counterparts w.r.t training memory. The grey baseline represents using the naive implementation of attention modules. With a memory-efficient implementation [6], Group DETR does not bring much memory burden during training, only requires 1.2 G and 1.7 G more GPU memory with Conditional DETR [30] (‘C-DETR’ for short) and DAB-DETR [25].

Table 2. Group DETR outperforms baseline models with a similar training time. Conditional DETR [30] and DAB-DETR [25] serve as baseline models to compare the performances on COCO val2017 [24]. ‘C-DETR’ and ‘w/ Group’ are the abbreviations of ‘Conditional DETR’ and ‘with Group DETR’. The entries noted by grey are the results of baseline models with the same training epochs (12 or 50 epochs) as Group DETR. To match the training times of Group DETR, we adopt longer training schedules for baselines (15 or 60 epochs). The training times are measured on 8 A100 GPUs in hours.

Model	w/ Group	Hours	mAP	$AP_s$	$AP_m$	$AP_l$
C-DETR		4.6	32.6	14.7	35.0	48.3
		5.8	34.4	15.1	37.3	51.3
	✓	5.6	37.6	18.2	40.7	55.9
C-DETR		19.2	40.9	20.5	44.2	59.6
		23.0	41.6	21.4	45.1	60.0
	✓	23.3	43.4	23.0	47.3	62.3
DAB-DETR		5.6	35.2	16.7	38.6	51.6
		7.0	36.3	17.1	39.4	52.5
	✓	6.6	39.1	19.7	42.5	56.8
DAB-DETR		23.3	42.2	21.5	45.7	60.3
		28.0	42.9	22.8	46.4	61.9
	✓	27.5	44.5	24.2	48.5	63.2

Group DETR will bring additional training computation costs (FLOPs) as well as training memory costs. But the parallel decoders can be implemented as a single decoder by replacing normal self-attention with parallel self-attention (depicted in Figure 6) and we can use an efficient attention implementation, FlashAttention [6, 19]. As a result, Group DETR only takes a small increase in training GPU memory and training time. For example, with Conditional DETR [30] and DAB-DETR [25], the memory increases are just 1.2 G and 1.7 G (Figure 7). The training time is increased by 5 minutes per epoch (from 23 minutes to 28 minutes and from 28 minutes to 33 minutes, respectively).

We provide the results by increasing the training time

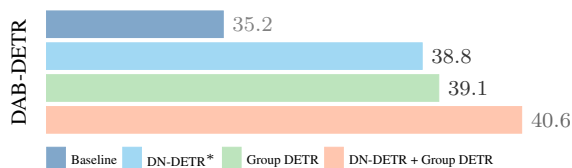


Figure 8. **Comparisons with DN-DETR.** Group DETR outperforms DN-DETR on DAB-DETR [25] (*y*-axis). Combining those two methods give better results, indicating they are complementary to each other. The *x*-axis is the mAP scores with a 12-epoch schedule on COCO *val2017*. \* represents that we report the best results of DN-DETR among different numbers of denoising queries (detailed results are provided in Appendix).

for normal DETR training to see if Group DETR benefits simply from more training time. The results given in Table 2 show that normal training with more training time brings a little benefit and the performance is still much lower than Group DETR, implying that the performance gain from our approach is not from training time increase.

**Connection to DN-DETR.** DN-DETR [20] aims to stabilize one-to-one assignment during DETR training. DN-DETR forms the additional queries by adding the noises to ground-truth objects, which can be regarded as a variant of our multi-group mechanism with clear differences. In DN-DETR [20], on the one hand, the number of queries within each additional group is the same as the number of ground-truth objects. Each one correspond to one ground-truth object, and there is no query corresponding to no-object. In contrast, our approach automatically learns a number of  $N$  (e.g., 300) object queries that correspond to both ground-truth objects and no-object.

On the other hand, DN-DETR performs self-attention over noised queries, mainly for collecting the information from predictions for other objects other than from duplicate predictions. Self-attention in Group DETR instead collects both duplicate predictions and predictions for other objects.

The above two comparisons imply that DN-DETR brings the major help for the box and classification prediction, through the introduction of more positive queries corresponding to ground-truth objects (like FCOS), and no direct help for duplicate prediction removal. Our approach introduces both positive queries and negative queries (no-object), also brings the help for duplicate prediction removal.

Figure 8 shows that the performance of Group DETR is better than DN-DETR. We further investigate if Group DETR still benefits from introducing more positive queries with noised queries. As shown in Figure 8, the performance gain over Group DETR is non-trivial, a 1.5 mAP. This implies that Group DETR and DN-DETR are complementary and their major roles are different, though they have some similarities.

## 4. Experiments

We demonstrate the effectiveness of Group DETR in various DETR variants, and its extension to 3D detection and instance segmentation [30, 25, 20, 47, 44, 26, 27, 5]. The training setting is almost the same as baseline models, for illustrating the effectiveness of our Group DETR. We adopt the same training settings and hyper-parameters as the baseline models, such as learning rate, optimizer, pre-trained model, initialization methods, and data augmentations<sup>1</sup>.

### 4.1. Object Detection

**Setting.** We study various representative DETR-based detectors, such as basic baselines (Conditional DETR [30], DAB-DETR [25], DN-DETR [20]) with dense attentions, and strong baselines (DAB-Deformable-DETR [25, 47] and DINO [44, 47]) with deformable attentions. We report the results on two training schedules, training for 12 epochs and training for more epochs (36 or 50). Unless specified, the models are trained with ResNet-50 [13] as the backbone on the COCO *train2017* and evaluated on the COCO *val2017*. More implementation details are provided in Appendix.

**Results.** We first report the results of training with 12 epochs in Table 3. Group DETR brings consistent improvements over the baselines with dense attentions that already are superior to the original DETR [2]. It boosts Conditional DETR (-DC5) [30] by 5.0 (4.8) mAP, improves DAB-DETR (-DC5) [25] by 3.9 (4.4) mAP, and brings a 2.0 (2.6) mAP gain to DN-DETR (-DC5) [20].

Group DETR also works well on those strong baselines with deformable attentions that are equipped with two or more accelerating techniques. It gives a 1.5 mAP improvements over DAB-Deformable-DETR [25, 47]. When applying to DINO [44, 20, 47], Group DETR also exceeds it by 0.7 mAP. The gain is non-trivial over such a stronger baseline, considering that DINO is a well-tuned model<sup>2</sup> based on DAB-Deformable-DETR that combines improved hyper-parameters, improved two-stage design, improved query denoising task, and other tricks.

Furthermore, we report the results with 50 training epochs that is commonly adopted in many acceleration methods [47, 30, 4, 25]. Table 4 presents that Group DETR outperforms baseline models by large margins. For the stronger backbone, Swin-Large [29], our approach achieves 58.4 mAP (still a 0.4 mAP higher than its baseline DINO [44] (58.0 mAP with Swin-Large)). This verifies the generalization ability of our Group DETR.

<sup>1</sup>We may adjust the batch size due to the limitation of the GPU memory size for both the baseline model and our approach so that the batch size is the same.

<sup>2</sup>In fact, our approach is compatible with query denoising and two-stage. In Table 3, for example, DN-DETR [20] utilizes query denoising, and our method improves it by 2.0. Similarly, DAB-DETR [25] adopts a two-stage structure, and our method achieves a 1.5 improvement.

Table 3. **Effectiveness of Group DETR with 12 epochs.** Group DETR gives consistent gains over various DETR-based baselines on COCO *val2017* [24], highlighted with brackets. All experiments adopt ResNet-50 [13] and do not use multiple patterns [40]. For DN-DETR, an improved version of DN, dynamic DN groups [44] with 100 DN queries, is used, making the results slightly different from the ones (with 3 patterns) reported in the original paper [20] (more results about the number of DN queries can be found in Appendix. ‘C-DETR’, ‘DAB-D-DETR’, and ‘w/ Group’ are ‘Conditional DETR’ [30], ‘DAB-Deformable DETR’ [25, 47], and ‘with Group DETR’, respectively, for neat representation.

Model	w/ Group	mAP	AP <sub>s</sub>	AP <sub>m</sub>	AP <sub>l</sub>
C-DETR	✓	32.6	14.7	35.0	48.3
		37.6 (+5.0)	18.2	40.7	55.9
C-DETR-DC5	✓	36.4	18.0	39.6	52.5
		41.2 (+4.8)	21.4	45.0	58.7
DAB-DETR	✓	35.2	16.7	38.6	51.6
		39.1 (+3.9)	19.7	42.5	56.8
DAB-DETR-DC5	✓	37.5	19.4	40.6	53.2
		41.9 (+4.4)	23.3	45.6	58.4
DN-DETR	✓	38.6	17.9	41.6	57.7
		40.6 (+2.0)	19.8	43.9	59.4
DN-DETR-DC5	✓	41.9	22.2	45.1	59.8
		44.5 (+2.6)	25.9	48.2	62.2
DAB-D-DETR	✓	44.2	27.5	47.1	58.6
		45.7 (+1.5)	28.1	49.0	60.6
DINO-4scale	✓	49.4	32.3	52.5	63.2
		50.1 (+0.7)	32.4	53.2	64.7

Last, we compare the training convergence curves of the baseline models and their Group DETR counterparts. The results, as shown in Figure 1, provide more evidence that Group DETR speeds DETR training convergence on various DETR variants.

**System-level Results on COCO *test-dev* with ViT-Huge.** We also have the system-level performance on COCO *test-dev* [24] with ViT-Huge [7]. We apply Group DETR to DINO [44] and follow its training pipeline and settings: pre-train the encoder with a self-supervised method, then pre-train the whole model on Object365 [34], and last fine-tune the whole model on COCO [24]. Our model is the first to achieve 64.5 mAP on COCO *test-dev*, which is still superior to other methods with larger encoder and more pre-training data [28, 39, 41, 42]. The details and comparisons with other methods are provided in Appendix.

## 4.2. More Applications

Group DETR is applicable to DETR-style techniques to other vision problems. We report the results for two ad-

Table 4. **Effectiveness of Group DETR with more epochs.** Group DETR still outperforms baselines by non-trivial margins with more training epochs (36 or 50 epochs). Settings and notations are consistent with Table 3, except for the training epochs (36 epochs for DINO-4scale by following the original paper [44] and 50 epochs for other models). ‘DINO-4scale-Swin-L’ means it adopts Swin-Large [29] as the backbone.

Model	w/ Group	mAP	AP <sub>s</sub>	AP <sub>m</sub>	AP <sub>l</sub>
C-DETR	✓	40.9	20.5	44.2	59.6
		43.4 (+2.5)	23.0	47.3	62.3
C-DETR-DC5	✓	43.7	23.9	47.6	60.1
		45.8 (+2.1)	26.8	49.7	63.1
DAB-DETR	✓	42.2	21.5	45.7	60.3
		44.5 (+2.3)	24.2	48.5	63.2
DAB-DETR-DC5	✓	44.5	25.3	48.2	62.3
		46.7 (+2.2)	27.6	50.9	64.0
DN-DETR	✓	44.0	23.9	47.7	62.9
		45.4 (+1.4)	25.1	49.3	63.8
DN-DETR-DC5	✓	47.5	27.9	50.7	65.9
		48.0 (+0.5)	29.3	52.1	65.4
DAB-D-DETR	✓	48.1	31.4	51.4	63.4
		49.7 (+1.6)	31.4	52.5	65.6
DINO-4scale	✓	50.9	34.6	54.1	64.6
		51.3 (+0.4)	34.7	54.5	65.3
DINO-4scale-Swin-L	✓	58.0	41.3	61.9	74.0
		58.4 (+0.4)	41.0	62.5	73.9

ditional problems: multi-view 3D object detection [15, 22, 26, 27] and instance segmentation [5, 21], to further demonstrate the effectiveness.

**Multi-view 3D object detection.** We report the results over PETR [26] and PETR v2 [27] on the nuScenes *val* dataset [1]. Table 5 shows that Group DETR brings significant gains to PETR and PETR v2 with 24 training epochs in terms of both the nuScenes Detection Score (NDS) and mAP scores.

Table 5. **Results on multi-view 3D object detection.** All experiments are evaluated on the nuScenes *val* set [1]. We train these experiments for 24 epochs with VoVNetV2 [18] as the backbone and with the image size of 800 × 320. We follow all the settings and hyper-parameters of PETR [26] and PETR v2 [27].

Model	w/ Group	NDS	mAP
PETR	✓	42.0	37.4
		45.0 (+3.0)	38.8 (+1.4)
PETR v2	✓	50.3	40.7
		51.3 (+1.0)	41.9 (+1.2)

**Instance segmentation.** We demonstrate the effectiveness of the representative method, Mask2Former [5]. The results are given in Table 6. Group DETR achieves a 1.2 (0.3) mAP<sup>m</sup> gain with 12 (50) epochs.

Table 6. **Results on instance segmentation.** The mask mAP ( $mAP^m$ ) is used for instance segmentation on COCO *val2017*. We adopt Mask2Former [5] as the baseline. The experiments are conducted with ResNet-50 [13] as the backbone, following all the settings of Mask2Former.

Epochs	w/ Group	$mAP^m$	$AP_s^m$	$AP_m^m$	$AP_l^m$
12		38.5	17.6	41.4	60.4
12	✓	39.7 (+1.2)	18.7	42.8	60.8
50		43.7	23.4	47.2	64.8
50	✓	44.0 (+0.3)	23.8	47.1	65.1

Table 7. **Effects of group-wise one-to-many assignment and separate self-attention.** (a) baseline: one-to-one assignment with 300 object queries. (b) naive one-to-many assignment with 3300 object queries for training and inference. (c) group-wise one-to-many assignment and no separate self-attention with 11 groups of 300 queries, inference with a group of 300 queries. (d) group-wise one-to-many assignment and separate self-attention with 11 groups of 300 queries, inference with a group of 300 queries. o2m = one-to-many, Sep. SA = separate self-attention.

	o2m	Sep. SA	mAP	$AP_s$	$AP_m$	$AP_l$
(a)	×	×	32.6	14.4	34.9	48.6
(b)	Naive	×	8.4	8.0	13.2	13.3
(c)	Group	×	34.8	16.4	37.7	51.4
(d)	Group	✓	<b>37.6</b>	<b>18.2</b>	<b>40.7</b>	<b>55.9</b>

### 4.3. Ablation Study

We conduct the ablation study by using Conditional DETR [30] as the baseline. The CNN backbone is ResNet-50 [13], and the training epoch number is 12. The performances are evaluated on COCO *val2017* [24]. We mainly study the effects of the key design: group-wise one-to-many assignment, separate self-attention, and group number.

**Group-wise one-to-many assignment and separate self-attention.** Table 7 shows how group-wise one-to-many (o2m) assignment and separate self-attention make contributions. In comparison to the baseline (a), group-wise o2m assignment improves the mAP score from 32.6 mAP to 34.8 mAP: with the gain 2.2. The separate self attention (Sep. SA) further gets a 2.8 mAP gain. In addition, we report naive one-to-many assignment. The results are very poor, which is reasonable in that there are duplicate predictions and there is a lack of scoring mechanisms for demoting them. The results suggest that both group-wise o2m assignment and separate self-attention are effective.

**Group number.** Figure 9 shows the influence of the number of groups  $K$  in Group DETR. The detection performance improves when increasing the number of groups, and becomes stable when the group number reaches 11. Thus, we adopt  $K = 11$  by default in Group DETR in our experiments.

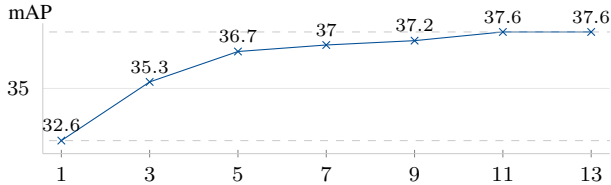


Figure 9. **Influence of group number.** The  $x$ -axis is the number of groups. It can be seen that the performance becomes stable when the number of groups reaches 11.

## 5. Related Works

There are two main lines for accelerating DETR training: modify *cross-attention* and stabilize *one-to-one assignment*. The two are complementary and can be combined to further boost the performance.

**Modifying cross-attention.** Cross-attention module aims to collect the information from the image features useful to classification and localization. Various methods are proposed to select the informative image regions more efficiently and effectively [8, 4, 40, 43, 25, 9]. For example, Deformable attention [47] selects the highly informative positions dynamically according to the previous decoder embedding. Conditional DETR [4] instead continues to use the normal global attention, and dynamically computes the spatial attention to softly select the informative regions. SMCA [8] uses the Gaussian-like weight for spatial modulation.

**Stabilizing one-to-one assignment.** DETR [2] relies on one-to-one assignment, where each ground-truth object is assigned to a single prediction through building a bipartite matching between the predictions and the ground-truth objects. DN-DETR [20] finds the assignment process is unstable and attributes the slow convergence issue to the instabilities. Thus, DN-DETR [20] introduces groups of noisy queries by adding noises to ground-truth objects, to stabilize the assignment, leading to faster convergence. DINO [44] makes further improvement through contrastive denoising training to generate both positive and negative noise queries with different noise levels. Our approach studies the assignment mechanism instead for introducing more supervision.

**One-to-many assignment.** One-to-many assignment is widely adopted in deep detectors [33, 12, 23, 36], and has attracted a lot of interest [45, 17, 46, 10, 3, 38, 35]. For example, Faster R-CNN [33] and FCOS [36] produce multiple positive anchors and pixels for each ground-truth object. In this paper, we investigate one-to-many assignment in a feasible manner for the end-to-end detector DETR.

Concurrent with our work,  $\mathcal{H}$ -DETR [16] also uses one-to-many assignment to speed up DETR training convergence. Our Group DETR and  $\mathcal{H}$ -DETR are related, but different: (1) Group DETR introduces group-wise one-to-



many assignment with separate self-attention with the same number of object queries in each group.  $\mathcal{H}$ -DETR adopts hybrid assignments in two different groups: One group uses one-to-one assignment and another uses one-to-many assignment with more object queries. (2) All the decoders in Group DETR can be used for inference. But the additional decoder in  $\mathcal{H}$ -DETR is not directly used and requires NMS for inference. (3) During training, our architecture introduces one parameter: the number of groups. In contrast,  $\mathcal{H}$ -DETR introduces the number of additional queries and the number of additional positive queries.

DETA [31] is another concurrent work with our Group DETR. DETA directly uses one-to-many assignment and brings NMS back to DETR frameworks. While our method provides group-wise one-to-many assignment and maintains end-to-end detection.

## 6. Conclusion

The key points in Group DETR include group-wise one-to-many assignment and parallel self-attention. The success stems from involving more groups of object queries as an addition to the primary group of object queries, and thus introducing more supervision. Group-wise assignment mechanism makes sure that the competition among predictions happens within each group separately, and separate self-attention eases the training. Thus, the NMS pose-processing is not necessary, and the inference process is kept the same as normally trained DETR and not dependent on the group design. Our approach is simple, easily implemented, and general.

**Acknowledgements.** This work is supported by the Sichuan Science and Technology Program (2023YFSY0008), National Natural Science Foundation of China (61632003, 61375022, 61403005), Grant SCITLAB-20017 of Intelligent Terminal Key Laboratory of SiChuan Province, Beijing Advanced Innovation Center for Intelligent Robots and Systems (2018IRS11), and PEK-SenseTime Joint Laboratory of Machine Vision.

## References

- [1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 7
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 6, 8
- [3] Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, and Jian Sun. You only look one-level feature. In *CVPR*, pages 13039–13048, 2021. 3, 8
- [4] Xiaokang Chen, Fangyun Wei, Gang Zeng, and Jingdong Wang. Conditional detr v2: Efficient detection transformer with box queries. *arXiv preprint arXiv:2207.08914*, 2022. 1, 6, 8
- [5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint arXiv:2112.01527*, 2021. 2, 6, 7, 8
- [6] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv preprint arXiv:2205.14135*, 2022. 5
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 7
- [8] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *ICCV*, pages 3621–3630, 2021. 1, 8
- [9] Ziteng Gao, Limin Wang, Bing Han, and Sheng Guo. Adamixer: A fast-converging query-based object detector. In *CVPR*, pages 5364–5373, 2022. 1, 8
- [10] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *CVPR*, pages 303–312, 2021. 3, 8
- [11] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3
- [12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 3, 8
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 6, 7, 8
- [14] J Hosang, R Benenson, and B Schiele. Learning non-maximum suppression. *PAMI*, 2017. 1
- [15] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 7
- [16] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detsr with hybrid matching. *arXiv preprint arXiv:2207.13080*, 2022. 8
- [17] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. In *ECCV*, pages 355–371. Springer, 2020. 8
- [18] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, pages 13906–13915, 2020. 7
- [19] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. <https://github.com/facebookresearch/xformers>, 2022. 5
- [20] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, 2022. 2, 4, 6, 7, 8

- [21] Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, Lionel M Ni, Heung-Yeung Shum, et al. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv preprint arXiv:2206.02777*, 2022. 7
- [22] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 7
- [23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1, 3, 8
- [24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 2, 5, 7, 8
- [25] 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. 1, 2, 5, 6, 7, 8
- [26] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 2, 6, 7
- [27] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr v2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 2, 6, 7
- [28] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022. 7
- [29] 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. *arXiv:2103.14030*, 2021. 6, 7
- [30] Depu Meng, Xiaokang Chen, Zejie Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, pages 3651–3660, 2021. 1, 2, 4, 5, 6, 7, 8
- [31] Jeffrey Ouyang-Zhang, Jang Hyun Cho, Xingyi Zhou, and Philipp Krähenbühl. Nms strikes back. *arXiv preprint arXiv:2212.06137*, 2022. 9
- [32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 3
- [33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 1, 2, 3, 8
- [34] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 7
- [35] Peize Sun, Yi Jiang, Enze Xie, Wenqi Shao, Zehuan Yuan, Changhu Wang, and Ping Luo. What makes for end-to-end object detection? In *ICML*, pages 9934–9944. PMLR, 2021. 8
- [36] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 2, 3, 8
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1
- [38] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. End-to-end object detection with fully convolutional network. In *CVPR*, pages 15849–15858, 2021. 8
- [39] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 7
- [40] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022. 1, 7, 8
- [41] Yixuan Wei, Han Hu, Zhenda Xie, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Contrastive learning rivals masked image modeling in fine-tuning via feature distillation. *arXiv preprint arXiv:2205.14141*, 2022. 7
- [42] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks, 2022. 7
- [43] 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. 1, 8
- [44] 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. 2, 6, 7, 8
- [45] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, pages 9759–9768, 2020. 3, 8
- [46] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv preprint arXiv:2007.03496*, 2020. 8
- [47] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. *CoRR*, abs/2010.04159, 2020. 1, 2, 6, 7, 8