Learning Dynamic Query Combinations for Transformer-based Object Detection and Segmentation

Yiming Cui [1 2] **Linjie Yang** [2] **Haichao Yu** [2]

Abstract

Transformer-based detection and segmentation methods use a list of learned detection queries to retrieve information from the transformer network and learn to predict the location and category of one specific object from each query. We empirically find that random convex combinations of the learned queries are still good for the corresponding models. We then propose to learn a convex combination with dynamic coefficients based on the high-level semantics of the image. The generated dynamic queries, named modulated queries, better capture the prior of object locations and categories in the different images. Equipped with our modulated queries, a wide range of DETR-based models achieve consistent and superior performance across multiple tasks including object detection, instance segmentation, panoptic segmentation, and video instance segmentation.

1. Introduction

Object detection is a fundamental yet challenging task in computer vision, which aims to localize and categorize objects of interest in the images simultaneously. Traditional detection models (Ren et al., 2015; Cai & Vasconcelos, 2019; Duan et al., 2019; Lin et al., 2017b;a) use complicated anchor designs and heavy post-processing steps such as Non-Maximum-Suppression (NMS) to remove duplicated detections. Recently, Transformer-based object detectors such as DETR (Carion et al., 2020) have been introduced to simplify the process. In detail, DETR combines convolutional neural networks (CNNs) with Transformer (Vaswani et al., 2017) by introducing an encoder-decoder framework to generate a series of predictions from a list of object queries. Following works improve the efficiency and convergence speed of DETR with modifications to the attention module (Zhu et al., 2021; Roh et al., 2021), and divide queries into positional and content queries (Liu et al., 2022; Meng et al., 2021; Wang et al., 2022; Zhang et al., 2022). This paradigm is also adopted for instance/panoptic segmentation, where each query is associated with one specific object mask in the decoding stage of the segmentation model (Cheng et al., 2021a; Dong et al., 2021; Cheng et al., 2021b; Hu et al., 2021; Wang et al., 2021d).

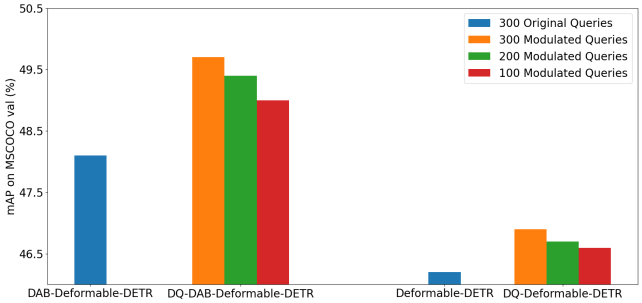


Figure 1. Comparison of DETR-based detection models integrated with and without our methods on MS COCO (Lin et al., 2014) `val` benchmark. ResNet-50 (He et al., 2016) is used as the backbone.

The existing DETR-based detection models always use a list of fixed queries, regardless of the input image. The queries will attend to different objects in the image through a multi-stage attention process. Here, the queries serve as global priors for the location and semantics of target objects in the image. In this paper, we would like to associate the detection queries with the content of the image, i.e., adjusting detection queries based on the high-level semantics of the image in order to capture the distribution of object locations and categories in this specific scene. For example, when the high-level semantics shows the image is a group photo, we know that there will be a group of people (category) inside the image and they are more likely to be close to the center of the image (location).

Since the detection queries are implicit features that do not directly relate to specific locations and object categories in the DETR framework, it is hard to design a mechanism to change the queries while keeping them within a meaningful "query" subspace to the model. Through an empirical study, we notice that convex combinations of learned

[1]Department of Electrical and Computer Engineering, University of Florida, Gainesville, USA [2]ByteDance Inc., San Jose, USA. Correspondence to: Linjie Yang <yljatthu@gmail.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

queries are still good queries to different DETR-based models, achieving similar performance as the originally learned queries (See Section 3.2). Motivated by this, we propose a method to generate dynamic detection queries, named *modulated queries*, based on the high-level semantics of the image in DETR-based methods while constraining the generated queries in a sequence of convex hulls spanned by the static queries. Therefore, the generated detection queries are more related to the target objects in the image and stay in a meaningful subspace. We show the superior performance of our approach combined with a wide range of DETR-based models on MS COCO (Lin et al., 2014), CityScapes (Cordts et al., 2016) and YouTube-VIS (Yang et al., 2019b) benchmarks with multiple tasks, including object detection, instance segmentation, and panoptic segmentation. In Figure 1, we show the performance of our method on object detection combined with two baseline models. When integrated with our proposed method, the mAP of recent detection models DAB-Deformable-DETR (Liu et al., 2022) can be increased by 1.6%. With fewer modulated queries, our method can still achieve better performance than baseline models on both Deformable-DETR and DAB-Deformable-DETR.

2. Related Works

Transformers for object detection. Traditional CNN-based object detectors require manually designed components such as anchors (Ren et al., 2015; Cai & Vasconcelos, 2019; Girshick, 2015; He et al., 2017) or post-processing steps such as NMS(Neubeck & Van Gool, 2006; Hosang et al., 2017; Rothe et al., 2015). Transformer-based detectors directly generate predictions for a list of target objects with a series of learnable queries. Among them, DETR (Carion et al., 2020) first combines the sequence-to-sequence framework with learnable queries and CNN features for object detection.

Following DETR, multiple works (Chen et al., 2022; Zhu et al., 2021; Roh et al., 2021; Jia et al., 2022; Zhang et al., 2022; Liu et al., 2022) were proposed to improve its convergence speed and accuracy. Deformable-DETR (Zhu et al., 2021) and Sparse-DETR (Roh et al., 2021) replace the self-attention modules with more efficient attention operations where only a small set of key-value pairs are used for calculation. Conditional-DETR (Tian et al., 2020) changes the queries in DETR to be conditional spatial queries, which speeds up the convergence process. SMCA-DETR (Gao et al., 2021) introduces pre-defined Gaussian maps around the reference points. Anchor-DETR (Wang et al., 2022) generates the object queries using anchor points rather than a set of learnable embeddings. DAB-DETR (Liu et al., 2022) directly uses learnable box coordinates as queries which can be refined in the Transformer decoder layers. DN-DETR (Li et al., 2022) improves the convergence speed of DETR by introducing noises to the ground truths and forcing the Transformer decoder to reconstruct the bounding boxes. DINO (Zhang et al., 2022) and DN-DETR (Li et al., 2022) introduce a strategy to train models with noisy ground truths to help the model learn the representation of the positive samples more efficiently.

Recently, Group-DETR (Chen et al., 2022) and HDETR (Jia et al., 2022) both added auxiliary queries and a one-to-many matching loss to improve the convergence of the DETR-based models. They still use static queries which does not change the general architecture of DETR. All these Transformer-based detection methods use fixed initial detection queries learned on the whole dataset. he queries will attend to different objects in the image through a multi-stage attention process. Without the global context, the queries might attend to regions that do not contain any objects or search for categories that do not exist in the image, which may limit the model's performance. In contrast, we propose to modulate the queries based on the image's content, which generates more effective queries for the current image.

Transformers for object segmentation. Besides object detection, Transformer-based models are also proposed for object segmentation tasks including image instance segmentation (He et al., 2017; Wang et al., 2020a; Bolya et al., 2019; Wang et al., 2020b; Bolya et al., 2020; Cao et al., 2020), panoptic segmentation (Kirillov et al., 2019; Wang et al., 2021a; Zhang et al., 2021; Xiong et al., 2019) and video instance segmentation (VIS) (Yang et al., 2019b; Hwang et al., 2021; Liu et al., 2021a; 2019). In DETR (Carion et al., 2020), a mask head is introduced on top of the decoder outputs to generate the predictions for panoptic segmentation. Following DETR, ISTR (Hu et al., 2021) generates low-dimensional mask embeddings, which are matched with the ground truth mask embeddings using Hungarian Algorithm for instance segmentation. SOLQ (Dong et al., 2021) uses a unified query representation for class, location, and mask.

Besides image object segmentation, researchers have begun to investigate object segmentation in video domains (Wang et al., 2021d; Wu et al., 2022; Thawakar et al., 2022; Yang et al., 2022; Hwang et al., 2021). VisTR (Wang et al., 2021d) extends DETR from the image domain to the video domain by introducing an instance sequence matching and segmentation pipeline for video instance segmentation. SeqFormer (Wu et al., 2022) utilizes video-level instance queries where each query attends to a specific object across frames in the video. MSSTS-VIS (Thawakar et al., 2022) introduces a multi-scale spatial-temporal split attention module for video instance segmentation.

Recently, multiple works (Cheng et al., 2021a; Jain et al., 2022; Cheng et al., 2021b; Liang et al., 2023) pay attention to unified frameworks for object segmentation tasks in

Model	DAB-DETR		Deformable-DETR		Mask2Former
	$r=2$	$r=4$	$r=2$	$r=4$	$r=2$
Convex Combination	37.9($\pm$0.10)	30.4($\pm$0.20)	35.0($\pm$0.20)	24.2($\pm$0.05)	41.2($\pm$0.10)
Non-convex Combination	37.0($\pm$0.10)	29.5($\pm$0.10)	32.6($\pm$0.25)	24.0($\pm$0.10)	40.7($\pm$0.45)
Averaged Combination	37.0	28.4	32.9	22.5	40.9
Queries sampled randomly	39.7($\pm$0.05)	33.9($\pm$0.15)	39.8($\pm$0.30)	28.1($\pm$0.30)	41.7($\pm$0.10)

Table 1. Comparison of pretrained detection models DAB-DETR (Liu et al., 2022) and Deformable-DETR and segmentation model Mask2Former (Cheng et al., 2021a) with different queries. The shown metrics are box mAP for detection and mask mAP for segmentation. ResNet-50 is used as the backbone and models are evaluated on MS COCO `val`.

both image and video domains. Cheng et al. (2021b) present MaskFormer, a straightforward mask classification model. It predicts binary masks linked to global class labels, simplifying semantic and panoptic segmentation tasks and achieving impressive empirical outcomes. By extending MaskFormer, Mask2Former (Cheng et al., 2021a) introduces masked attention to extract localized features and predict output for panoptic, instance, and semantic segmentation in a unified framework. These Transformer-based models follow the general paradigm of DETR and use fixed queries regardless of the input.

Dynamic deep neural networks. Dynamic deep neural network (Han et al., 2021) aims at adjusting the computation procedure of a neural network adaptively in order to reduce the overall computation cost or enhance the model capacity. Slimmable networks (Yu et al., 2018; Yu & Huang, 2019; Li et al., 2021) introduce a strategy to adapt to multiple devices by simply changing channel numbers without the need for retraining. Dynamic Convolution (Chen et al., 2020) proposes a dynamic perceptron that uses dynamic attention weights to aggregate multiple convolution kernels based on the input features. Similar to dynamic convolution, CondConv (Yang et al., 2019a) introduces an operation named conditionally parameterized convolutions, which learns specialized convolutional kernels for each individual input.

On object detection, Dynamic R-CNN (Zhang et al., 2020) proposes a new training strategy to dynamically adjust the label assignment for two-stage object detectors based on the statics of proposals. Dynamic-DETR (Li et al., 2021) introduces a dynamic attention module to DETR that dynamically adjusts attention according to factors such as the importance of scale to improve the performance on small objects and convergence speed. Cui et al. (2022) proposes to train a single detection model which can adjust the number of proposals based on the complexity of the input image. TF-Blender (Cui et al., 2021; Cui, 2022a; Cui & Yang, 2023; Cui, 2022b) simplifies the feature aggregation process for video object detection by using a dynamic number of frames to enhance the object representations. Wang et al. (2021c) introduces a Dynamic Transformer to determine the number of tokens according to the input image for efficient image recognition, by stacking multiple Transformer layers with increasing numbers of tokens. SODAR (Wang et al., 2021b) focuses on instance segmentation based on a one-stage SOLO model (Wang et al., 2020a;b) for better performance. It improves the final segmentation quality by leveraging the rich neighboring information with a learning-based aggregation method. This model cannot be directly applied to other models, such as DETR-based models. GCNet (Cao et al., 2019) is designed for long-range dependency modeling in traditional convolutional networks. It simplifies the Non-Local Network (NLNet) by only considering the global context in the attention block.

Both SODAR and GCNet deal with CNN-based model backbones, which are different from the Transformer encoder-decoder structure in the DETR framework. We believe our method can shed light on dynamic model designing in the Transformer paradigm. In contrast to the existing work, we explore generating dynamic queries for a wide range of DETR-based models using the same framework. Our focus is not to reduce the computation cost of DETR-based models, but to improve the model performances with queries more related to the content of each individual image.

3. Methodology

3.1. Preliminary

We first summarize the inference process of the existing Transformer-based models for a series of tasks, including object detection, instance segmentation, and panoptic segmentation, as the following Equation:

$$\boldsymbol{Y} = \mathcal{N}_t\left(\mathcal{N}_{dec}\left(\mathcal{N}_{enc}\left(\boldsymbol{F}\right), \boldsymbol{Q}\right)\right). \tag{1}$$

For the object detection task, given the input image $\boldsymbol{I}$, multi-scale features $\boldsymbol{F}$ are extracted from the backbone network and then fed into a Transformer encoder $\mathcal{N}_{enc}$. After processing the features with multiple encoder layers, the output features are fed into a Transformer decoder $\mathcal{N}_{dec}$ together with n randomly initialized query vectors $\boldsymbol{Q} \in \mathbb{R}^{n \times f}$, where n and f denote the number of queries and length of each query respectively. Each query can be a feature vector (Carion et al., 2020; Zhu et al., 2021), or a learned anchor box (Liu et al., 2022). The outputs of $\mathcal{N}_{dec}$ are then fed into a task head $\mathcal{N}_t$ to generate the final predictions $\boldsymbol{Y} = \{(\boldsymbol{b}_i, \boldsymbol{c}_i), i = 1, 2, \ldots, n\}$, where $\boldsymbol{b}_i, \boldsymbol{c}_i$ represent the bounding boxes and their corresponding categories of the

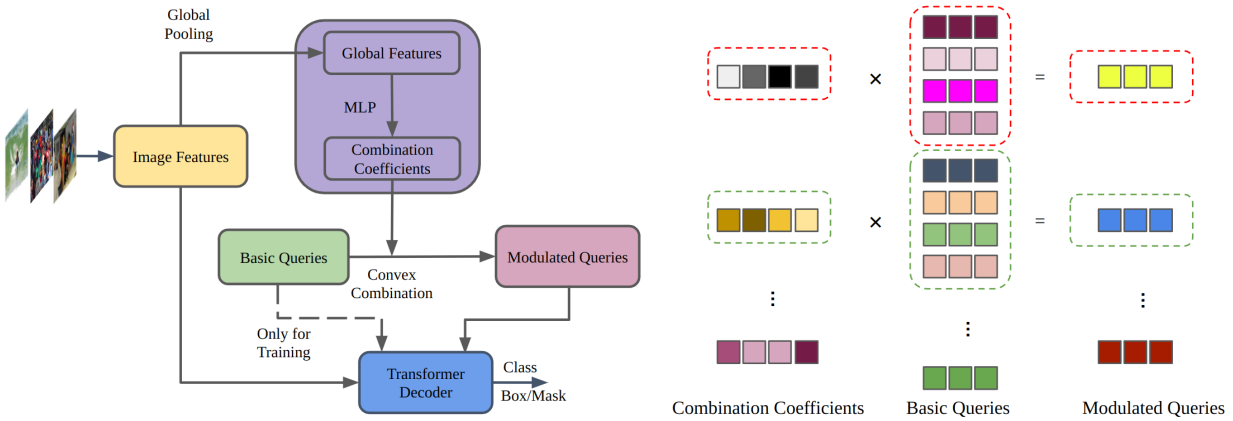


Figure 2. The framework of the proposed method. (a) Model pipeline with dynamic query combinations. The step with the dashed line is only used in training. (b) Illustration of generating modulated queries from basic queries given combination coefficients.

detected objects. Then, the predictions are matched with the ground truths $\boldsymbol{Y}^\star$ using the Hungarian Algorithm (Carion et al., 2020) to generate a bipartite matching. Then, the final loss is computed based on this bipartite matching:

$$\mathcal{L} = \mathcal{L}_{\text{Hungarian}}\left(\boldsymbol{Y}, \boldsymbol{Y}^\star\right). \tag{2}$$

For the segmentation tasks, the final predictions are updated to $\boldsymbol{Y} = \{(\boldsymbol{b}_i, \boldsymbol{c}_i, \boldsymbol{m}_i), i = 1, 2, \ldots, n\}$, where $\boldsymbol{m}_i$ denotes the predicted masks for different object instances. Since there is no direct correspondence of the predictions with the ground truth annotations, a bipartite matching is also computed to find the correspondence of the predictions and the ground truths $\boldsymbol{Y}^\star$. The final loss is then computed based on the matching. In some models such as Mask2Former (Cheng et al., 2021a), there will be no Transformer encoder $\mathcal{N}_{enc}$ to enhance the feature representations, while the other computational components follow the same paradigm.

3.2. Fixed Query Combinations

Though some existing works analyze the contents of the queries for the decoder, such as Conditional-DETR (Tian et al., 2020) and Anchor-DETR (Wang et al., 2022), they always exam each query individually. To the best of our knowledge, there is no work studying the interaction between the queries in $\boldsymbol{Q}$. Here, we would like to explore what kind of transformations conducted between the learned queries still generate "good" queries. If we compute the average of a few queries, is it still an effective query? If we use different types of linear transformations, which would be better to produce good queries?

We conduct experiments to analyze the results of queries generated by different perturbations from the original queries. The procedure of the experiments is as follows: given a well-trained Transformer-based model, the initial queries for the decoder are denoted as $\boldsymbol{Q}^P = \left\{\boldsymbol{q}_1^P, \boldsymbol{q}_2^P, \ldots, \boldsymbol{q}_n^P\right\} \in \mathbb{R}^{n \times f}$. The first type of perturbation uses linear combinations of the original queries. We first separate the n queries into m groups, where each group has $r = \frac{n}{m}$ queries and generates one new query. Then, we initialize the combination coefficients $\boldsymbol{W} \in \mathbb{R}^{m \times r}$, where $w_{ij} \in \boldsymbol{W}$ is the coefficient used for the i-th group, j-th queries, denoted as $\boldsymbol{q}_{ij}^P$, to generate a group of new queries $\boldsymbol{Q}^C = \{\boldsymbol{q}_1^C, \boldsymbol{q}_2^C, \ldots, \boldsymbol{q}_m^C\} \in \mathbb{R}^{m \times f}$. The process can be summarized as:

$$\boldsymbol{q}_i^C = \sum_{j=1}^{r} w_{ij} \boldsymbol{q}_{ij}^P, \tag{3}$$

We use three settings to evaluate the impact of different coefficients in Equation 3, namely Convex Combination, Non-convex Combination, and Averaged Combination:

In Convex Combination, $\boldsymbol{q}_i^C$ is within the convex hull of $\boldsymbol{q}_{ij}^P, j = 1, 2, \ldots, r$. The combination coefficients w_{ij} are randomly initialized using uniform distribution in $[-1, 1]$ and then passed through a softmax function to satisfy the criteria: $w_{ij} \geq 0, \quad \sum_{j=1}^{r} w_{ij} = 1$.

For Non-convex Combinations, w_{ij} are initialized in the same way as those in the convex combination, and the sum of w_{ij} is forced to be 1. However, there is no guarantee on its range and w_{ij} can be negative values. For Averaged Combination, we generate $\boldsymbol{q}_i^C$ by averaging $\boldsymbol{q}_{ij}^P, j = 1, 2, \ldots, r$. As a baseline, we evaluate the model on m queries randomly sampled from $\boldsymbol{Q}^P$. The experiments are conducted on MS COCO benchmark (Lin et al., 2014) for object detection, and instance segmentation, using DAB-DETR (Liu et al., 2022), Deformable-DETR (Zhu et al., 2021) and Mask2Former (Cheng et al., 2021a), with ResNet-50 (He et al., 2016) as the backbone. The results are summarized in Table 1.

From Table 1, we notice that Convex Combination achieves the best results among all the compared settings except the baseline. Convex Combination only degenerates

Backbone	Method	mAP	$AP_{0.5}$	$AP_{0.75}$
ResNet-50	Conditional-DETR (Tian et al., 2020)	40.9	61.7	43.3
	DQ-Conditional-DETR	$42.0_{\uparrow 1.1}$	$63.3_{\uparrow 1.6}$	$44.2_{\uparrow 0.9}$
	SMCA-DETR (Gao et al., 2021)	41.0	61.5	43.5
	DQ-SMCA-DETR	$42.1_{\uparrow 1.1}$	$63.3_{\uparrow 1.8}$	$44.9_{\uparrow 1.4}$
	DAB-DETR (Liu et al., 2022)	42.1	63.1	44.6
	DQ-DAB-DETR	$43.7_{\uparrow 1.6}$	$64.4_{\uparrow 1.3}$	$46.6_{\uparrow 2.0}$
	Deformable-DETR (Zhu et al., 2021)	46.2	65.0	49.9
	DQ-Deformable-DETR	$47.0_{\uparrow 0.8}$	$65.5_{\uparrow 0.5}$	$50.9_{\uparrow 1.0}$
	DAB-Deformable-DETR (Liu et al., 2022)	48.1	66.4	52.0
	DQ-DAB-Deformable-DETR	$49.7_{\uparrow 1.6}$	$68.1_{\uparrow 1.7}$	$54.2_{\uparrow 2.2}$
Swin-Base	Deformable-DETR (Zhu et al., 2021)	50.9	70.5	55.3
	DQ-Deformable-DETR	$53.2_{\uparrow 2.3}$	$72.8_{\uparrow 2.3}$	$57.7_{\uparrow 2.4}$
	DAB-Deformable-DETR (Liu et al., 2022)	52.7	71.8	57.4
	DQ-DAB-Deformable-DETR	$53.8_{\uparrow 1.1}$	$72.8_{\uparrow 1.0}$	$58.6_{\uparrow 1.2}$

Table 2. Comparison of existing DETR-based object detectors with/without our proposed methods integrated on MS COCO `val` split.

slightly compared with learned queries on DAB-DETR and Mask2Former. In addition, the performance of Convex Combination only has very small variances across different models, proving that convex combinations of the group-wise learned queries are naturally high-quality object queries for different Transformer-based models on both detection and segmentation tasks. n is set to 300 for detection models and 100 for Mask2Former. We run each experimental setting 6 times to compute the variance.

3.3. Dynamic Query Combinations

From the previous section, we learn that fixed convex combinations of learned queries can still produce a reasonable accuracy compared to the learned queries. In this section, we propose a strategy to learn dynamic query combinations for the Transformer-based models instead of randomly generating the coefficients w_{ij} for query combinations. Our model predicts their values according to the high-level content of the input. Thus, each input image will have a distinct set of object queries fed into the Transformer decoder.

To generate dynamic queries, a naive idea is to generate the queries directly from the input features $\boldsymbol{F}$. This method will increase the number of parameters dramatically, causing it difficult to optimize and inevitably computationally inefficient. To verify this, we conduct an experiment on Deformable-DETR (Zhu et al., 2021) with ResNet-50 as the backbone. We replace the original randomly initialized queries with those generated by a multi-layer perceptron (MLP), which transforms the image feature $\boldsymbol{F}$ to $\boldsymbol{Q}$. With 50 epochs, the model only achieves 45.1% mAP, which is lower than the original model with 46.2%.

Inspired by the dynamic convolution (Chen et al., 2020; Yang et al., 2019a), which aggregates the features with multiple kernels in each convolutional layer, we propose a query modulation method. We introduce two types of queries: basic queries $\boldsymbol{Q}^B \in \mathbb{R}^{n\times f}$ and modulated queries $\boldsymbol{Q}^M \in \mathbb{R}^{m\times f}$, where n, m are the number of queries and $n = rm$. Equation 3 is updated as:

$$\boldsymbol{q}_i^M = \sum_{j=1}^{r} w_{ij}^D \boldsymbol{q}_{ij}^B, \tag{4}$$

where $\boldsymbol{W}^D \in \mathbb{R}^{m\times r}$ is the combination coefficient matrix and $w_{ij}^D \in \boldsymbol{W}^D$ is the coefficient for the i-th group, j-th query in $\boldsymbol{Q}^B$, denoted as $\boldsymbol{q}_{ij}^B$. To guarantee our query combinations to be convex, we add extra constraints to the coefficients as $w_{ij}^D \geq 0, \quad \sum_{j=1}^{r} w_{ij}^D = 1$.

We use an example here to illustrate how to divide the basic queries into multiple groups. The basic queries are represented as $\boldsymbol{Q}^B = \{\boldsymbol{q}_0^B, \boldsymbol{q}_1^B, \boldsymbol{q}_2^B, \boldsymbol{q}_3^B, \boldsymbol{q}_4^B, \boldsymbol{q}_5^B, \boldsymbol{q}_6^B, \boldsymbol{q}_7^B\}$ and $r = 4$. We divide the basic queries in sequential order. Therefore, $\boldsymbol{q}_0^B, \boldsymbol{q}_1^B, \boldsymbol{q}_2^B, \boldsymbol{q}_3^B$ is used to generate $\boldsymbol{q}_0^M$ and $\boldsymbol{q}_4^B, \boldsymbol{q}_5^B, \boldsymbol{q}_6^B, \boldsymbol{q}_7^B$ is used to generate $\boldsymbol{q}_1^M$. $\boldsymbol{w}_0^D \in \mathbb{R}^4$ is used to weighted average $\boldsymbol{q}_0^B, \boldsymbol{q}_1^B, \boldsymbol{q}_2^B, \boldsymbol{q}_3^B$ to generate $\boldsymbol{q}_0^M \in \mathbb{R}^f$ while $\boldsymbol{w}_1^D \in \mathbb{R}^4$ is used to weighted average $\boldsymbol{q}_4^B, \boldsymbol{q}_5^B, \boldsymbol{q}_6^B, \boldsymbol{q}_7^B$ to generate $\boldsymbol{q}_1^M \in \mathbb{R}^f$. We did not conduct experiments to study the effects of different divisions. Since the basic queries are randomly initialized and are jointly learned with the modulated queries, we believe the results will not change significantly with a different division.

In our dynamic query combination module, the coefficient matrix $\boldsymbol{W}^D$ is learned based on the input feature $\boldsymbol{F}$ through a mini-network, as:

$$\boldsymbol{W^D} = \sigma\left(\theta\left(\mathcal{A}\left(\boldsymbol{F}\right)\right)\right), \tag{5}$$

where $\mathcal{A}$ is a global average pooling to generate a global feature from the feature map $\boldsymbol{F}$, θ is an MLP, σ is a softmax function to guarantee the elements of $\boldsymbol{W}^D$ satisfy the convex constraints. Here we try to make the mini-network

Method	Backbone	Panoptic			Instance	
		PQ	AP^{Th}_{pan}	$mIoU_{pan}$	mAP	$AP_{0.5}$
Mask2Former (Cheng et al., 2021a)	ResNet-50	62.1	37.3	77.5	37.4	61.9
DQ-Mask2Former		$63.2_{\uparrow 1.1}$	$38.2_{\uparrow 0.9}$	$78.7_{\uparrow 1.2}$	$38.5_{\uparrow 1.1}$	$63.2_{\uparrow 1.3}$
Mask2Former (Cheng et al., 2021a)	Swin-Base	66.1	42.8	82.7	42.0	68.8
DQ-Mask2Former		$67.0_{\uparrow 0.9}$	$43.7_{\uparrow 0.9}$	$83.7_{\uparrow 1.0}$	$43.0_{\uparrow 1.0}$	$69.6_{\uparrow 0.8}$

Table 3. Comparison of Mask2Former and DQ-Mask2Former on panoptic and instance segmentation tasks on CityScapes `val` split.

Methods	Backbone	mAP
Mask R-CNN (He et al., 2017)	ResNet-50	35.4
QueryInst (Fang et al., 2021)		39.8
Mask2Former(Cheng et al., 2021a)	ResNet-50	43.7
DQ-Mask2Former		$44.4_{\uparrow 0.7}$
Mask2Former (Cheng et al., 2021a)	Swin-Base	46.7
DQ-Mask2Former		$47.6_{\uparrow 0.9}$

Table 4. Comparison of existing instance segmentation approaches and DQ-Mask2Former on MS COCO `val` split.

as simple as possible to show the potential of using modulated queries. This attention-style structure happens to be a simple and effective design choice.

During the training process, we feed both $\boldsymbol{Q}^M$ and $\boldsymbol{Q}^B$ to the same decoder to generate the corresponding predictions $\boldsymbol{Y}^M$ and $\boldsymbol{Y}^B$ as follows,

$$\begin{aligned} \boldsymbol{Y}^M &= \mathcal{N}_t\left(\mathcal{N}_{dec}\left(\mathcal{N}_{enc}\left(\boldsymbol{F}\right), \boldsymbol{Q}^M\right)\right) \\ \boldsymbol{Y}^B &= \mathcal{N}_t\left(\mathcal{N}_{dec}\left(\mathcal{N}_{enc}\left(\boldsymbol{F}\right), \boldsymbol{Q}^B\right)\right) \end{aligned} \tag{6}$$

The final training loss is then updated to

$$\mathcal{L} = \mathcal{L}_{\text{Hungarian}}\left(\boldsymbol{Y}^M, \boldsymbol{Y}^\star\right) + \beta \mathcal{L}_{\text{Hungarian}}\left(\boldsymbol{Y}^B, \boldsymbol{Y}^\star\right) \tag{7}$$

where β is a hyperparameter. During the inference, only $\boldsymbol{Q}^M$ is used to generate the final predictions $\boldsymbol{Y}^M$ while the basic queries $\boldsymbol{Q}^B$ are not used. Therefore, the computational complexity increases for our models are negligible compared to the original DETR-based models. The only difference in the computation is that we have an additional MLP and a convex combination to generate the modulated queries. Therefore, the role of modulated queries in our model is exactly the same as the fixed object queries in the original models.

4. Experiments

To evaluate the effectiveness of our proposed methods, we first conduct experiments on a series of tasks, including object detection, instance segmentation, panoptic segmentation, and video instance segmentation with different DETR-based models. Then we conduct several ablation studies to investigate the impact of different hyperparameters in our model for a better analysis. Finally, we visualize the dynamic query combinations to show the effectiveness of our model.

4.1. Experiment Setup

Datasets. For the object detection task, we use MS COCO benchmark (Lin et al., 2014) for evaluation, which contains $118,287$ images for training and $5,000$ for validation. For instance and panoptic segmentation, besides the MS COCO benchmark (80 "things" and 53 "stuff" categories), we also conduct experiments on the CityScapes (Cordts et al., 2016) benchmark (8 "things" and 11 "stuff" categories) to validate the effectiveness of our proposed method. For the video instance segmentation task, YouTube-VIS-2019 (Yang et al., 2019b) is used for evaluation. For experiments on video instance segmentation, we pretrain our models on MS COCO and finetune them on the training set of YouTube-VIS-2019.

Evaluation metrics. For panoptic segmentation, the standard **PQ** (panoptic quality) metric (Kirillov et al., 2019) is used for evaluation. For instance segmentation (image or video) and object detection, we use the standard **mAP** (mean average precision) metric for evaluation. For VIS, mAP and AR (average recall) on video instances are the evaluation metrics. We observe around 0.8 mAP fluctuations in performance and we report the results reproduced based on the officially released code in this section.

Implementation details. The query ratio r used to generate the combination coefficients is set to 4 by default. β is set to be 1. θ is implemented as a two-layer MLP with ReLU as nonlinear activations. The output size of its first layer is 512, and that of the second layer is the length of $\boldsymbol{W}^D$ in corresponding models. For detection models, we use 300 modulated queries and 1200 basic queries if not specified otherwise. For the baseline models used for comparison, we use 300 queries as the original implementation by default. For instance segmentation and panoptic segmentation models, we use 100 modulated queries and 400 basic queries for Mask2Former. For video instance segmentation, we use 100 modulated queries and 400 basic queries for Mask2Former and 300 modulated queries, and 1,200 basic queries for SeqFormer. For the baseline models, we use 100 queries for Mask2Former on image segmentation tasks, 100 queries for Mask2Former on video instance segmentation, and 300 queries for SeqFormer on video instance segmentation. The

Methods	Backbone	PQ	PQ_{th}
UPSnet (Xiong et al., 2019)	ResNet-50	42.5	48.6
DETR (Carion et al., 2020)		43.4	48.2
Mask2Former (Cheng et al., 2021a)	ResNet-50	51.9	57.7
DQ-Mask2Former		$52.6_{\uparrow 0.7}$	$58.9_{\uparrow 1.2}$
Mask2Former (Cheng et al., 2021a)	Swin-Base	55.1	61.0
DQ-Mask2Former		$55.7_{\uparrow 0.6}$	$61.8_{\uparrow 0.8}$

Table 5. Comparison of existing panoptic segmentation approaches with DQ-Mask2Former on MS COCO `val` split.

comparison is based on the fairness principle that the same number of queries are used as input to the transformer decoders of our model and the baseline. During inference, the only added computational cost of our method compared to the baselines is the mini-network to produce the modulated queries.

4.2. Main Results

Object detection. We evaluate our proposed methods with the DETR-based models Deformable-DETR (Zhu et al., 2021), SMCA-DETR (Gao et al., 2021), Conditional-DETR (Tian et al., 2020), DAB-DETR and DAB-Deformable-DETR (Liu et al., 2022) with ResNet50 (He et al., 2016) for object detection on the MS COCO benchmark. We also experiment with Deformable-DETR and DAB-Deformable-DETR on Swin-B(Liu et al., 2021b) to further evaluate the performance of our method on more powerful backbones. For a fair comparison, we run the original model integrated with and without our proposed modulated queries using the same experimental settings. The models equipped with our dynamic query combinations are denoted as DQ-Deformable-DETR, DQ-SMCA-DETR, DQ-Conditional-DETR, DQ-DAB-DETR, and DQ-DAB-Deformable-DETR, respectively. The results are shown in Table 2. From Table 2, when integrated with our proposed method, mAP can be improved consistently by at least 0.8% for all the models listed in the table. For DAB-Deformable-DETR, the mAP can be improved by 1.6% with ResNet50 backbone and 1.1% with Swin-Base backbone. For Deformable-DETR, the mAP can be improved significantly by 2.3% with the Swin-Base backbone. This proves the benefit of our method with different backbones. Note that models with modulated queries only have negligible increased computation cost compared to the original models.

Instance/panoptic segmentation. Mask2Former(Cheng et al., 2021a) is a recent state-of-the-art model that can be used for different segmentation tasks with a unified model architecture. We compare Mask2Former with/without our modulated queries for image instance and panoptic segmentation tasks on the MS COCO (Lin et al., 2014) and CityScapes benchmarks. The model plugged with modulated queries is named DQ-Mask2Former. The results are shown in Table 3, Table 4, and Table 5, respectively.

For instance segmentation (Table 4), our model DQ-Mask2Former achieves consistent improvement across different metrics compared to the original Mask2Former. For example, the performance on mAP is improved by around 0.8% on both MS COCO. For panoptic segmentation, as shown in Table 5, DQ-Mask2Former again significantly outperforms Mask2Former (Cheng et al., 2021a) across all the evaluation metrics on both MS COCO and CityScapes. Since panoptic segmentation is more challenging compared with instance segmentation and object detection where both semantic and instance segmentation tasks are required to generate the final predictions, our model works less effectively for panoptic segmentation compared to other tasks.

Video instance segmentation. Besides image tasks, we also evaluate our method on the video instance segmentation task. We evaluated our method based on two state-of-the-art video instance segmentation methods Mask2Former (Cheng et al., 2021a) and SeqFormer (Wu et al., 2022). Results are shown in Table 6. It can be seen from Table 6 that when integrated with our modulated queries, mAP, and AR of Mask2Former are improved by at around 1.0% and the mAP of SeqFormer is significantly boosted by 1.5%. Note the additional computation cost is negligible with our modulated queries.

4.3. Model Analysis

In this section, we conduct extensive experiments to analyze the designs of our proposed method. By default, for the object detection task, the number of modulated queries is set to 300. For the segmentation tasks, the number of modulated queries is set to 50 for a faster training pipeline, and r is set to 4 for all the tasks.

Analysis of the number of queries. We use Deformable-DETR and DAB-Deformable-DETR as baseline models to study the effects of the number of queries on the performance of object detection. We compare the baseline models with DQ-Deformable-DETR and DQ-DAB-Deformable-DETR integrated with different numbers of queries as in Figure 1. Note that we include the additional components of our models in the FLOPs computation of the decoder. When

Method	Backbone	mAP	$AP_{0.5}$	$AP_{0.75}$
MaskTrack R-CNN (Yang et al., 2019b)	ResNet-50	30.3	51.1	32.6
IFC (Hwang et al., 2021)		42.8	65.8	46.8
Mask2Former (Cheng et al., 2021a)	ResNet-50	46.4	68.0	50.0
DQ-Mask2Former		$47.4_{\uparrow 1.0}$	$69.2_{\uparrow 1.2}$	$51.0_{\uparrow 1.0}$
SeqFormer (Wu et al., 2022)		47.4	69.8	51.8
DQ-SeqFormer		$49.0_{\uparrow 1.6}$	$71.5_{\uparrow 1.7}$	$53.0_{\uparrow 1.2}$
Mask2Former (Cheng et al., 2021a)	Swin-Base	59.5	84.3	67.2
DQ-Mask2Former		$61.3_{\uparrow 1.8}$	$86.1_{\uparrow 1.8}$	$68.6_{\uparrow 1.4}$
SeqFormer (Wu et al., 2022)	Swin-Large	59.3	82.1	66.4
DQ-SeqFormer		$61.2_{\uparrow 1.9}$	$84.1_{\uparrow 2.0}$	$68.0_{\uparrow 1.6}$

Table 6. Comparison of existing video instance segmentation approaches with DQ-Mask2Former and DQ-SeqFormer on YouTube-VIS-2019 `val` split.

β	mAP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
0.0	45.6	64.1	49.4	27.2	49.1	60.5
0.5	46.4	65.0	50.3	28.1	49.2	62.6
1.0	47.0	65.5	50.9	28.8	50.1	62.2

Table 7. Analysis of β using DQ-Deformable-DETR (ResNet-50 as the backbone) on the MS COCO benchmark with different settings.

integrated with our method, even by reducing the number of modulated queries from 300 to 100, the mAPs of DQ-Deformable-DETR and DQ-DAB-Deformable-DETR are still better than the baseline models with 300 queries. We are also able to reduce the computation cost of the decoders of Deformable-DETR and DAB-Deformable-DETR by about 14% and 24% by using our method with 100 queries, respectively. However, we do not observe significant speedup using our method with fewer queries mainly because the main computation costs are from the backbones and the transformer encoders.

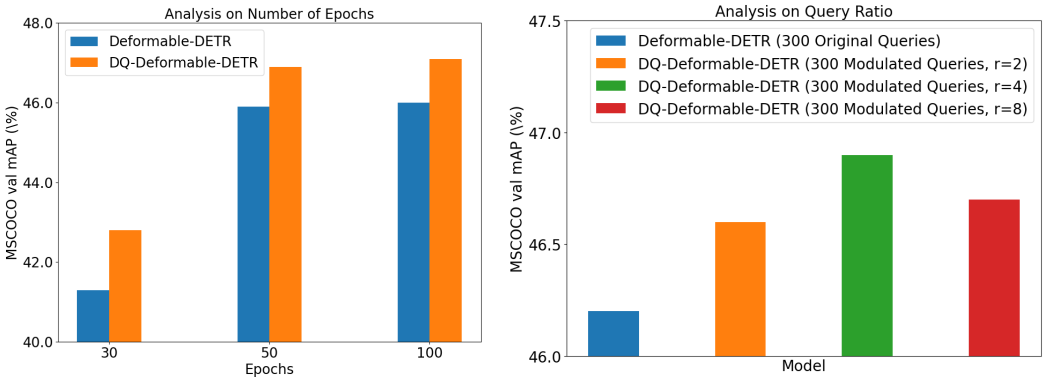
Analysis of the number of training epochs. In Figure 3 (a), we show the impact of the number of training epochs on a sample model DQ-Deformable-DETR together with the original Deformable-DETR. From the figure, the mAP of DQ-Deformable-DETR is always better than that of the original Deformable-DETR at different epochs on the MS COCO benchmark. At early 30 epochs, DQ-Deformable-DETR achieves a more significant performance gain compared to Deformable-DETR compared with later epochs.

Analysis of β. We analyze the impact of the scale of β on models equipped with our modulated queries. We conduct experiments using Deformable-DETR with ResNet-50 as the backbone of the MS COCO benchmark with different values of β. Results are shown in Table 7. As shown in the table, when β is set to be 0, where no loss is directly computed with the prediction from the basic queries, the performance drops by 2.4% compared to the original setting. In this case, the basic queries are not necessarily proper queries for the detection model, which will affect the quality of the modulated queries produced by them. The performance can be improved by increasing the value of β. Empirically we find $\beta = 1$ is a good choice to balance the scale of losses between the basic and modulated queries.

Analysis of query ratio. We use DQ-Deformable-DETR (Zhu et al., 2021) to analyze the performance of our proposed methods with different query ratios 2, 4, and 8, as in Figure 3 (b). From the figure, using 4 as the query ratio achieves the best performance for DQ-Deformable-DETR with 300 modulated queries. However, other query ratio choices still generate better accuracies than the original Deformable-DETR, which validates the effectiveness and robustness of our method.

Non-modulated ablation. Our model uses additional training queries (the basic queries) compared to the baselines. Here we conduct an ablative study on the factor of additional training queries. We train the model with two unrelated groups of queries with Deformable-DETR (1200/300 queries) and Mask2Former (400/100 queries), as Table 8. The first group has the same number as our basic queries while the second group has the same number as our modulated queries. Only the second group is used in inference. From the table, models with the two unrelated groups produce similar results as the baselines. In contrast, our proposed method with modulated queries achieves significant improvement in the two models. This proves that the improvement of our model is not simply due to more queries in training on DETR-based models.

Analysis of speed. The proposed method needs to apply forward passes in two branches for the basic queries and the modulated queries in the training phase of the model, which increases the computation cost. Here, we report the training time and inference speed of our model compared to the baseline with Deformable-DETR (ResNet-50) and Mask2Former (ResNet-50) in Table 9. The training time is based on 8 NVIDIA A100 GPUs and the inference FPS is



(a) (b)

Figure 3. Analysis of the impact of the number of epochs and query ratios on the performance.

Method	mAP	AP_{50}	AP_{75}
Deformable-DETR	46.2	65.0	49.9
Deformable-DETR†	46.5	64.9	50.5
DQ-Deformable-DETR	47.0	65.5	50.9
Mask2Former	43.7	65.5	46.9
Mask2Former†	43.9	65.8	47.2
DQ-Mask2Former	44.4	66.3	47.6

Table 8. Comparison with two groups of unrelated queries. † denotes two groups of unrelated queries.

Model	Training Time	Inference FPS
Deformable-DETR	∼ 61 GPU hours	13.3
DQ-Deformable-DETR	∼ 69 GPU hours	13.1
Mask2Former	∼ 71 GPU hours	5.4
DQ-Mask2Former	∼ 80 GPU hours	5.2

Table 9. Comparison of the training/inference time with and without our proposed methods integrated.

tested on a single TITAN RTX GPU. From the table, our method only increases the training time slightly compared to the baselines. The reason is that the major computation cost of DETR-based models comes from the backbones and transformer encoders that only need to be forwarded once for the two branches. During inference, the FPS of DQ-Deformable-DETR and DQ-Mask2Former are slightly reduced by less than 4% due to the extra computation of the mini-network to produce the modulated queries.

Visualization of $\boldsymbol{W}^D$. Since $\boldsymbol{W}^D$ is conditioned on the high-level content of the image, we conjecture that images with similar scenes or object categories may have similar $\boldsymbol{W}^D$ parameters. We choose 200 images from the validation set of MS COCO and compute their $\boldsymbol{W}^D$ from DQ-DAB-DETR with 300 queries. The resulting $\boldsymbol{W}^D$ are first flattened into vectors and then projected onto a two-dimensional space using t-SNE (Van der Maaten & Hinton, 2008). We visualize the projected $\boldsymbol{W}^D$ parameters along with their corresponding input images as Figure 4. We can see that some object categories tend to be clustered. For example, we can see a lot of transportation vehicles in the top right corner of the figure, and wild animals tend to be in the lower part of the figure, which indicates that the model uses some high-level semantics of the image to produce the combination coefficients.

Figure 4. t-SNE visualization of $\boldsymbol{W}^D$ on 200 images from MS COCO `val`. Zoom in to see details.

5. Conclusion

In this paper, we propose to use dynamic queries depending on the input image to enhance DETR-based detection and segmentation models. We find that convex combinations of learned queries are naturally high-quality object queries for the corresponding models. Based on this observation, we design a pipeline to learn dynamic convex combinations of the basic queries, adapting object queries according to the high-level semantics of the input images. This approach consistently improves the performance of a wide range of DETR-based models on object detection and segmentation tasks. The gain of our model is agnostic to the different designs of the Transformer decoders and different types of object queries. We believe this approach opens the door to designing dynamic queries and creates a new perspective for Transformer-based models.

References

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. Yolact: Real-time instance segmentation. In *ICCV*, 2019.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. Yolact++: Better real-time instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Cai, Z. and Vasconcelos, N. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019. ISSN 1939-3539. doi: 10.1109/tpami.2019.2956516. URL http://dx.doi.org/10.1109/tpami.2019.2956516.

Cao, J., Anwer, R. M., Cholakkal, H., Khan, F. S., Pang, Y., and Shao, L. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020.

Cao, Y., Xu, J., Lin, S., Wei, F., and Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCVW*, 2019.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *ECCV*, 2020.

Chen, Q., Chen, X., Zeng, G., and Wang, J. Group detr: Fast training convergence with decoupled one-to-many label assignment. *arXiv preprint arXiv:2207.13085*, 2022.

Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., and Liu, Z. Dynamic convolution: Attention over convolution kernels. In *CVPR*, 2020.

Cheng, B., Choudhuri, A., Misra, I., Kirillov, A., Girdhar, R., and Schwing, A. G. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021a.

Cheng, B., Schwing, A., and Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. *NIPS*, 2021b.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Cui, Y. Dfa: Dynamic feature aggregation for efficient video object detection. *arXiv preprint arXiv:2210.00588*, 2022a.

Cui, Y. Dynamic feature aggregation for efficient video object detection. In *ACCV*, 2022b.

Cui, Y. and Yang, L. Faq: Feature aggregated queries for transformer-based video object detectors, 2023.

Cui, Y., Yan, L., Cao, Z., and Liu, D. Tf-blender: Temporal feature blender for video object detection. In *ICCV*, 2021.

Cui, Y., Yang, L., and Liu, D. Dynamic proposals for efficient object detection. *arXiv preprint arXiv:2207.05252*, 2022.

Dong, B., Zeng, F., Wang, T., Zhang, X., and Wei, Y. Solq: Segmenting objects by learning queries. *NIPS*, 2021.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019.

Fang, Y., Yang, S., Wang, X., Li, Y., Fang, C., Shan, Y., Feng, B., and Liu, W. Instances as queries. In *ICCV*, 2021.

Gao, P., Zheng, M., Wang, X., Dai, J., and Li, H. Fast convergence of detr with spatially modulated co-attention. In *ICCV*, 2021.

Girshick, R. Fast r-cnn. In *ICCV*, 2015.

Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *ICCV*, 2017.

Hosang, J., Benenson, R., and Schiele, B. Learning non-maximum suppression. In *CVPR*, 2017.

Hu, J., Cao, L., Lu, Y., Zhang, S., Wang, Y., Li, K., Huang, F., Shao, L., and Ji, R. Istr: End-to-end instance segmentation with transformers. *arXiv preprint arXiv:2105.00637*, 2021.

Hwang, S., Heo, M., Oh, S. W., and Kim, S. J. Video instance segmentation using inter-frame communication transformers. *NIPS*, 2021.

Jain, J., Li, J., Chiu, M., Hassani, A., Orlov, N., and Shi, H. Oneformer: One transformer to rule universal image segmentation. *arXiv preprint arXiv:2211.06220*, 2022.

Jia, D., Yuan, Y., He, H., Wu, X., Yu, H., Lin, W., Sun, L., Zhang, C., and Hu, H. Detrs with hybrid matching. *arXiv preprint arXiv:2207.13080*, 2022.

Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. Panoptic segmentation. In *CVPR*, 2019.

Li, C., Wang, G., Wang, B., Liang, X., Li, Z., and Chang, X. Dynamic slimmable network. In *CVPR*, 2021.

Li, F., Zhang, H., Liu, S., Guo, J., Ni, L. M., and Zhang, L. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, 2022.

Liang, J., Zhou, T., Liu, D., and Wang, W. Clustseg: Clustering for universal segmentation. *arXiv preprint arXiv:2305.02187*, 2023.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *CVPR*, 2017a.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *ICCV*, 2017b.

Liu, D., Cui, Y., Tan, W., and Chen, Y. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*, 2021a.

Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., and Zhang, L. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *ICLR*, 2022.

Liu, X., Ren, H., and Ye, T. Spatio-temporal attention network for video instance segmentation. In *ICCVW*, 2019.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021b.

Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., and Wang, J. Conditional detr for fast training convergence. In *ICCV*, 2021.

Neubeck, A. and Van Gool, L. Efficient non-maximum suppression. In *ICPR*, 2006.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

Roh, B., Shin, J., Shin, W., and Kim, S. Sparse detr: Efficient end-to-end object detection with learnable sparsity. *arXiv preprint arXiv:2111.14330*, 2021.

Rothe, R., Guillaumin, M., and Van Gool, L. Non-maximum suppression for object detection by passing messages between windows. In *ACCV*, 2015.

Thawakar, O., Narayan, S., Cao, J., Cholakkal, H., Anwer, R. M., Khan, M. H., Khan, S., Felsberg, M., and Khan, F. S. Video instance segmentation via multi-scale spatio-temporal split attention transformer. In *ECCV*, 2022.

Tian, Z., Shen, C., and Chen, H. Conditional convolutions for instance segmentation. In *ECCV*, 2020.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *NIPS*, 2017.

Wang, H., Zhu, Y., Adam, H., Yuille, A., and Chen, L.-C. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021a.

Wang, T., Liew, J. H., Li, Y., Chen, Y., and Feng, J. Sodar: Exploring locally aggregated learning of mask representations for instance segmentation. *IEEE Transactions on Image Processing*, 31:839–851, 2021b.

Wang, X., Kong, T., Shen, C., Jiang, Y., and Li, L. SOLO: Segmenting objects by locations. In *ECCV*, 2020a.

Wang, X., Zhang, R., Kong, T., Li, L., and Shen, C. Solov2: Dynamic and fast instance segmentation. *NIPS*, 2020b.

Wang, Y., Huang, R., Song, S., Huang, Z., and Huang, G. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *NIPS*, 2021c.

Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., and Xia, H. End-to-end video instance segmentation with transformers. In *CVPR*, 2021d.

Wang, Y., Zhang, X., Yang, T., and Sun, J. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022.

Wu, J., Jiang, Y., Bai, S., Zhang, W., and Bai, X. Seqformer: Sequential transformer for video instance segmentation. In *ECCV*, 2022.

Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., and Urtasun, R. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019.

Yang, B., Bender, G., Le, Q. V., and Ngiam, J. Condconv: Conditionally parameterized convolutions for efficient inference. *NIPS*, 2019a.

Yang, L., Fan, Y., and Xu, N. Video instance segmentation. In *ICCV*, 2019b.

Yang, S., Wang, X., Li, Y., Fang, Y., Fang, J., Liu, W., Zhao, X., and Shan, Y. Temporally efficient vision transformer for video instance segmentation. In *CVPR*, 2022.

Yu, J. and Huang, T. Autoslim: Towards one-shot architecture search for channel numbers. *arXiv preprint arXiv:1903.11728*, 2019.

Yu, J., Yang, L., Xu, N., Yang, J., and Huang, T. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018.

Zhang, H., Chang, H., Ma, B., Wang, N., and Chen, X. Dynamic r-cnn: Towards high quality object detection via dynamic training. In *ECCV*, 2020.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L. M., and Shum, H.-Y. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

Zhang, W., Pang, J., Chen, K., and Loy, C. C. K-net: Towards unified image segmentation. *NIPS*, 2021.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.