DI-MaskDINO: A Joint Object Detection and Instance Segmentation Model

Zhixiong Nan[1], Xianghong Li[1], Tao Xiang *[1], and Jifeng Dai[2]

[1]College of Computer Science, Chongqing University, Chongqing, China.
[2]Department of Electronic Engineering, Tsinghua University, Beijing, China.
nanzx@cqu.edu.cn, lixianghong@stu.cqu.edu.cn, txiang@cqu.edu.cn, daijifeng@tsinghua.edu.cn

Abstract

This paper is motivated by an interesting phenomenon: the performance of object detection lags behind that of instance segmentation (i.e., *performance imbalance*) when investigating the intermediate results from the beginning transformer decoder layer of MaskDINO (i.e., the SOTA model for joint detection and segmentation). This phenomenon inspires us to think about a question: will the *performance imbalance* at the beginning layer of transformer decoder constrain the upper bound of the final performance? With this question in mind, we further conduct qualitative and quantitative pre-experiments, which validate the negative impact of *detection-segmentation imbalance* issue on the model performance. To address this issue, this paper proposes **DI-MaskDINO** model, the core idea of which is to improve the final performance by alleviating the *detection-segmentation imbalance*. **DI-MaskDINO** is implemented by configuring our proposed ***De-Imbalance (DI)*** module and ***Balance-Aware Tokens Optimization (BATO)*** module to MaskDINO. ***DI*** is responsible for generating balance-aware query, and ***BATO*** uses the balance-aware query to guide the optimization of the initial feature tokens. The balance-aware query and optimized feature tokens are respectively taken as the *Query* and *Key&Value* of transformer decoder to perform joint object detection and instance segmentation. **DI-MaskDINO** outperforms existing joint object detection and instance segmentation models on COCO and BDD100K benchmarks, achieving **+1.2** AP^{box} and **+0.9** AP^{mask} improvements compared to SOTA joint detection and segmentation model MaskDINO. In addition, **DI-MaskDINO** also obtains **+1.0** AP^{box} improvement compared to SOTA object detection model DINO and **+3.0** AP^{mask} improvement compared to SOTA segmentation model Mask2Former.

1 Introduction

Object detection and instance segmentation are two fundamental tasks in the computer vision community. Intuitively, the two tasks are closely-related and mutually-beneficial. However, in the current time, specialized detection or segmentation gains more focuses, and the amount of works studying the specialized task is significantly larger than that for joint tasks. One typical explanatory is that the multi-task training even hurts the performance of the individual task.

Confronting current research situations, we think about whether there are some essential cruxes that are ignored in previous works and these cruxes hinder the cooperation of object detection and instance

*Corresponding author.

This work is supported by Chongqing Natural Science Foundation Innovation and Development Joint Fund (CSTB2023NSCQ-LZX0109).

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1930

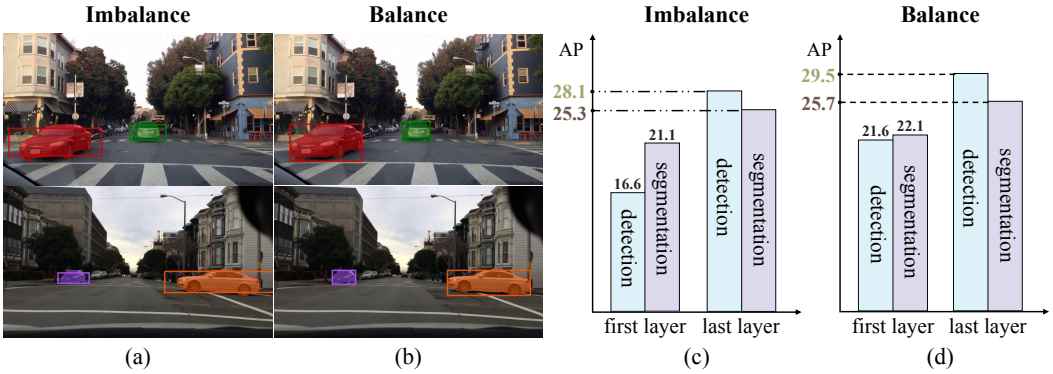

Figure 1: Qualitatively, (a) shows that the detection bounding boxes predicted by the query/feature at the first decoder layer of MaskDINO do not fit well with segmentation masks, and (b) exhibits that the corresponding results of **DI-MaskDINO** are optimized and the detection bounding boxes closely surround segmentation masks. Quantitatively, (c) displays that there exists a significant performance gap between detection and segmentation at the first decoder layer of MaskDINO, and (d) demonstrates **DI-MaskDINO** not only alleviates the performance imbalance at the first layer but also improves the performance upper bound.

segmentation tasks, which further constrains the breakthrough of the performance upper bound. This paper reveals one of cruxes is the imbalance of object detection and instance segmentation. As shown in Fig. 1, when investigating the results at the first layer of transformer decoder of MaskDINO model [25], an interesting phenomenon is found that there exists the performance imbalance between object detection and instance segmentation, as qualitatively illustrated in Fig. 1a and quantitatively illustrated in the first bar of Fig. 1c. After considering the imbalance issue, the performance gap at the first layer is narrowed as illustrated in the first bar of Fig. 1d, and the final performance upper bounds are improved (i.e., 28.1 to 29.5 for detection and 25.3 to 25.7 for segmentation) as illustrated in the second bar of Fig. 1d. The qualitative results are also optimized, the detection bounding boxes closely surround segmentation masks, as illustrated in Fig. 1b.

According to the above analysis, we can find the detection-segmentation imbalance at the beginning layer is one of essential cruxes that hinders the cooperation of object detection and instance segmentation. Therefore, we reviewed the previous works to investigate whether there are works that have been aware of this issue. The idea of many classical and excellent methods [16, 2, 5, 11] is combining two tasks by adding a segmentation branch to an object detector. These detect-then-segment methods make the performance of segmentation task to be limited by the performance of the object detector. Thanks to the thriving of transformer [44] and DETR [3], recent research attention has been geared towards transformer-based methods, which make giant contributions to the community. For example, [10, 50, 25] use the unified query representation for object detection and instance segmentation tasks based on transformer architecture.

However, to our best knowledge, there is no existing work to solve the above mentioned detection-segmentation imbalance issue. Factually, the imbalance issue naturally exists, which is determined by the **individual characteristics** of detection and segmentation tasks and also derived from the **supervision manners** for the two tasks. **Firstly**, segmentation is a pixel-level grouping and classification task [16, 46], thus local detailed information is important for this task. In contrast, detection is a region-level task to locate and regress the object bounding box [13, 38], which requires global information focusing on the complete object. The query at the beginning decoder layer conveys relatively local features, which is more beneficial for the segmentation task, thus the detection task tends to achieve lower performance at the beginning layer. **Secondly**, supervision manners for detection and segmentation are distinctly different. The segmentation is densely supervised by all pixels of the GT mask, while detection is sparsely supervised by a 4D vector (i.e., x, y, w, and h) of GT bounding box. The dense supervision provides richer and stronger information than the sparse supervision during the optimization procedure. Therefore, the optimization speeds of the two tasks are not synchronous, which will lead to the imbalance issue.

Based on two above-analyzed reasons for the detection-segmentation imbalance issue, it is straightforward that the performance of detection task will lag behind at the beginning layer. Considering existing

methods share a unified query for detection and segmentation tasks, the performance of a task will be negatively affected by another poorly-performed task, leading to that the multi-task joint training even hurts the performance of the individual task. Therefore, addressing the detection-segmentation imbalance issue is significant for designing a joint object detection and instance segmentation model. To address the detection-segmentation imbalance issue, we propose **DI-MaskDINO** model, which is implemented by configuring our proposed ***De-Imbalance (DI)*** module and ***Balance-Aware Tokens Optimization (BATO)*** module to MaskDINO. ***DI*** module is responsible for generating balance-aware query. Specifically, ***DI*** module strengthens the detection at the beginning decoder layer to balance the performance of the two tasks, and the core of ***DI*** module is our proposed ***residual double-selection*** mechanism. In this mechanism, the *token interaction* based ***double-selection*** structure learns the global geometric, contextual, and semantic patch-to-patch relations to update initial feature tokens, and high-confidence tokens are selected to benefit the detection task since the selected tokens have learned global semantics during the *token interaction* procedure. In addition, this mechanism makes use of initial feature tokens by the ***residual*** structure, which is the necessary compensation for the information loss occurring during ***double-selection***. Apart from ***DI*** module, we also design ***BATO*** module, which uses the balance-aware query to guide the optimization of initial feature tokens. The balance-aware query and optimized feature tokens are respectively taken as the *Query* and *Key&Value* of transformer decoder to perform joint object detection and instance segmentation.

The contributions of this paper are as follows: ***i***) to our best knowledge, this paper for the first time focuses on the detection-segmentation imbalance issue and proposes ***DI*** module with the ***residual double-selection*** mechanism to alleviate the imbalance; ***ii***) **DI-MaskDINO** outperforms existing SOTA joint object detection and instance segmentation model MaskDINO (**+1.2** AP^{box} and **+0.9** AP^{mask} on COCO, using ResNet50 backbone with 12 training epochs), SOTA object detection model DINO (**+1.0** AP^{box} on COCO), and SOTA segmentation model Mask2Former(**+3.0** AP^{mask} on COCO).

2 Related Work

Object Detection. Classical object detection methods [13, 38, 27, 37, 5, 43, 42] have achieved significant success. In recent years, transformer-based methods such as DETR [3] make a giant contribution to object detection by introducing the concept of object queries and the one-to-one matching mechanism. The success of DETR has sparked a boom in query-based end-to-end detectors, and numerous excellent variants are proposed [56, 33, 52, 28, 23, 6, 47, 51, 24, 32, 21, 54, 18]. Specifically, to enhance the convergence speed of DETR, Deformable DETR [56] proposes deformable attention mechanism that learns sparse feature sampling and aggregates multi-scale features accelerating model convergence and improving performance. From the interpretability of object queries, DAB-DETR [28] formulates the queries as 4D anchor boxes and dynamically updates them in each decoder layer.

Instance Segmentation. CNN-based instance segmentation methods are categorized into top-down and bottom-up paradigms. The top-down paradigm [16, 2, 11, 19, 7, 4, 1] firstly generates bounding boxes by object detectors, and then segments the masks. The bottom-up paradigm [35, 9, 29, 12] treats instance segmentation as a labeling-clustering problem, where pixels are firstly labeled as a class or embedded into a feature space and then clustered into each object. Recently, many transformer-based instance segmentation methods [8, 14, 20, 53, 17, 10, 50, 25] are proposed. The transformer-based methods treat the instance segmentation task as a mask classification problem that associates the instance categories with a set of predicted binary masks.

Joint Object Detection and Instance Segmentation. The goal of joint object detection and instance segmentation is to carry out the two tasks simultaneously [45, 34, 41, 36]. The traditional joint object detection and instance segmentation methods [16, 2, 5, 11] are usually implemented by adding a mask branch to a strong object detector. For example, the classical model Mask RCNN [16] achieves joint object detection and instance segmentation by adding a mask branch to Faster RCNN [39]. Recently, the proposal of the transformer-based framework has promoted the development of joint object detection and instance segmentation. Following DETR [3], SOLQ [10] proposes a unified query representation for simultaneous detection and segmentation tasks. The recent MaskDINO [25] achieves optimal performance with the unified query representation on both detection and segmentation tasks.

https://doi.org/10.52202/079017-1930

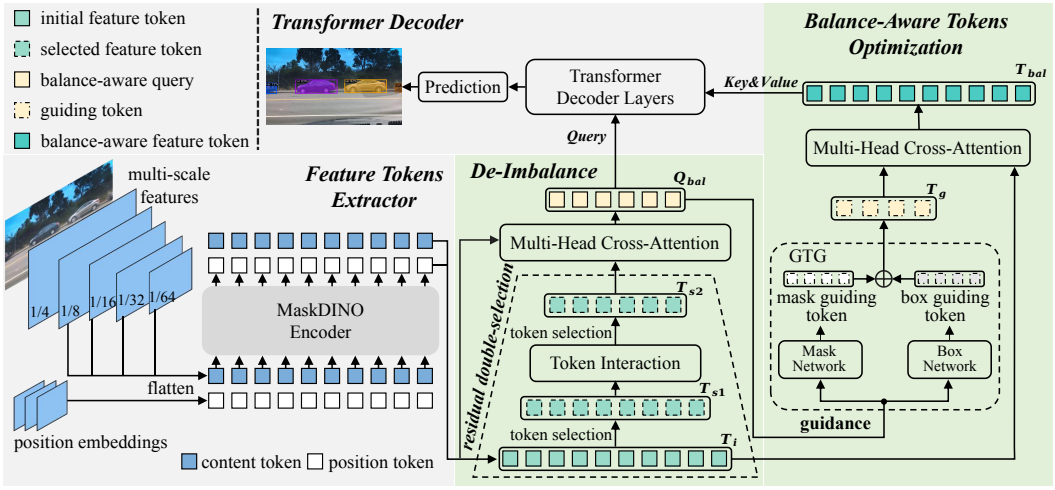

Figure 2: The overview of **DI-MaskDINO** model based on MaskDINO (grey shaded), with the extensions (green shaded) of ***De-Imbalance*** and ***Balance-Aware Tokens Optimization***. For simplicity, content token and position token are merged in ***De-Imbalance*** (i.e., $\boldsymbol{T}_i$, $\boldsymbol{T}_{s1}$, $\boldsymbol{T}_{s2}$, and $\boldsymbol{Q}_{bal}$ contain both content and position token) in presentation. GTG is short for guiding token generation.

3 Proposed Method

In response to the naturally-existing but commonly-ignored imbalance issue between object detection and instance segmentation, we propose **DI-MaskDINO** model, which is based on MaskDINO [25]. To better understand our proposed **DI-MaskDINO**, we firstly review MaskDINO (§ 3.1), and then introduce **DI-MaskDINO** (§ 3.2).

3.1 Preliminaries: MaskDINO

MaskDINO is a unified object detection and segmentation framework, which adds a mask prediction branch on the structure of DINO [52]. MaskDINO (grey shaded part in Fig. 2) is composed of a backbone, a transformer encoder, a transformer decoder, and detection and segmentation heads (i.e., "Prediction" in Fig. 2). Position embeddings and the flattened multi-scale features (extracted by backbone) are inputted to the transformer encoder to generate the initial feature tokens ($\boldsymbol{T}_i$). Note that in MaskDINO, the top-ranked feature tokens selected from $\boldsymbol{T}_i$ directly serve as the *Query* of transformer decoder, while we design ***De-Imbalance*** module with a ***residual double-selection*** mechanism to firstly alleviate the detection-segmentation imbalance and then obtain the balance-aware query $\boldsymbol{Q}_{bal}$ to serve as the *Query* of transformer decoder. In addition, we design ***Balance-Aware Tokens Optimization*** module to optimize $\boldsymbol{T}_i$ and generate the balance-aware feature tokens $\boldsymbol{T}_{bal}$ to serve as the *Key&Value* of transformer decoder. Token and query are specialized terms, and their explanations are provided in Appendix A.

3.2 Our Method: DI-MaskDINO

Fig. 2 illustrates the overview of **DI-MaskDINO**, which consists of four modules, including ***Feature Tokens Extractor*** (***FTE***), ***De-Imbalance*** (***DI***), ***Balance-Aware Tokens Optimization*** (***BATO***), and ***Transformer Decoder*** (***TD***). ***FTE*** extracts the initial feature tokens $\boldsymbol{T}_i$ from the input image using backbone and MaskDINO encoder. The goal of ***DI*** is to generate the balance-aware query $\boldsymbol{Q}_{bal}$, which is implemented by applying our proposed ***residual double-selection*** mechanism on the initial feature tokens $\boldsymbol{T}_i$. ***BATO*** utilizes $\boldsymbol{Q}_{bal}$ to optimize $\boldsymbol{T}_i$, generating the balance-aware feature tokens $\boldsymbol{T}_{bal}$. ***TD*** takes $\boldsymbol{T}_{bal}$ as the *Key&Value* and $\boldsymbol{Q}_{bal}$ as the *Query* to perform joint object detection and instance segmentation.

https://doi.org/10.52202/079017-1930

3.2.1 Feature Tokens Extractor

Given an image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$, the backbone (e.g., ResNet [15]) firstly extracts multi-scale features, which are then flattened and concatenated to serve as the input of transformer encoder comprising six multi-scale deformable attention layers [56], obtaining the initial feature tokens $\boldsymbol{T}_i$ that is composed of $\sum_{i=3}^{6}(\frac{H}{2^i}\times\frac{W}{2^i})$ tokens, where each token expresses the feature of the corresponding patch in $\boldsymbol{I}$.

3.2.2 De-Imbalance

There exists the detection-segmentation imbalance at the beginning layer of transformer decoder, which might constrain the upper bound of model performance. To handle this issue, we design ***DI*** module to alleviate the imbalance, instead of directly providing $\boldsymbol{T}_i$ to the transformer decoder as MaskDINO does. Specifically, detection-segmentation imbalance means that the performance of object detection lags behind that of instance segmentation at the beginning layer of transformer decoder. Therefore, we propose the ***residual double-selection*** mechanism to strengthen the performance of object detection.

The ***double-selection*** consists of the first selection and the second selection. In the first selection, we select top-k_1 ranked feature tokens in $\boldsymbol{T}_i$, based on their category classification scores:

$$\boldsymbol{T}_{s1} = \mathcal{S}(\boldsymbol{T}_i, k_1), \tag{1}$$

where $\boldsymbol{T}_{s1}$ represents the firstly-selected feature tokens, $\mathcal{S}$ denotes the selection operator. The first selection mainly filters out most of the tokens conveying background information, making $\boldsymbol{T}_{s1}$ focus on the objects.

Before the second selection, a *token interaction* network comprising two self-attention layers is applied on $\boldsymbol{T}_{s1}$:

$$\boldsymbol{T}_{s1}^{sa} = \text{MHSA}(\boldsymbol{T}_{s1}), \tag{2}$$

where MHSA is Multi-Head Self-Attention and $\boldsymbol{T}_{s1}^{sa}$ indicates the feature tokens processed by MHSA.

The *token interaction* is the key point to make sure that the secondly-selected tokens are beneficial for detection, we explain its rationality as follows. As we know, each token actually corresponds to a patch (remarkably smaller than an object in most cases) in the image [55]. The bounding box of an object is regressed by integrating the multiple patches (belonging to the same object) that have global patch-to-patch spatial relations, thus it is really needed for the detection task to learn the interaction relation between patches. In contrast, the dense all-pixel supervision for the segmentation task mainly focuses on local pixel-level similarity with GT mask [25], hence the segmentation task is not particularly depend on the patch-to-patch relation as the detection task. By self-attention layers, different tokens representing the patches (belonging to the same object) can interact with each other to learn the global geometric, contextual, and semantic patch-to-patch relations, benefiting the perception of object bounding boxes. Therefore, executing *token interaction* before the second selection makes ***DI*** module to be more beneficial for detection. In addition, verified by some studies (e.g., [32]), the tokens with higher category scores correspond to higher IOU scores of object bounding boxes. Therefore, the second selection further strengthens the object detection and alleviates the detection-segmentation imbalance.

In the second selection, we select the top-k_2 ranked feature tokens in $\boldsymbol{T}_{s1}^{sa}$ to obtain the secondly-selected feature tokens $\boldsymbol{T}_{s2}$:

$$\boldsymbol{T}_{s2} = \mathcal{S}(\boldsymbol{T}_{s1}^{sa}, k_2). \tag{3}$$

The ***residual double-selection*** is inspired by the ***residual*** idea in [15], and the ***residual*** is the necessary compensation for ***double-selection*** since the information loss occurs in the selection procedures. The formulation of this mechanism is combining $\boldsymbol{T}_i$ with $\boldsymbol{T}_{s2}$ by the Multi-Head Cross-Attention network (MHCA, a self-attention layer and a FFN layer are omitted here), generating $\boldsymbol{Q}_{bal}$:

$$\boldsymbol{Q}_{bal} = \text{MHCA}(\boldsymbol{T}_{s2}, \boldsymbol{T}_i). \tag{4}$$

$\boldsymbol{Q}_{bal}$ conveys the balance-aware information, thus it is named as balance-aware query. Subsequently, $\boldsymbol{Q}_{bal}$ is fed to ***BATO*** to guide the optimization of initial feature tokens $\boldsymbol{T}_i$. It is noted that the tokens in $\boldsymbol{Q}_{bal}$ have become significantly different from the tokens in $\boldsymbol{T}_i$. Through Eq. 1-4, the tokens in $\boldsymbol{Q}_{bal}$ have obtained larger receptive field and considered the interaction with other tokens, thus they are better understood as object/instance candidates. Correspondingly, the denotation has been changed from $\boldsymbol{T}$ to $\boldsymbol{Q}$.

https://doi.org/10.52202/079017-1930

3.2.3 Balance-Aware Tokens Optimization

In MaskDINO, initial feature tokens $\boldsymbol{T}_i$ directly serve as the *Key&Value* of ***TD***. Instead, we design ***BATO*** module that makes use of both balance-aware query $\boldsymbol{Q}_{bal}$ and $\boldsymbol{T}_i$ to generate the *Key&Value* of ***TD***. $\boldsymbol{T}_i$ contains a large number ($\approx$ 20k) of tokens conveying detailed local information for both background and objects/instances, while $\boldsymbol{Q}_{bal}$ consists of a small number (=300) of high-confidence tokens mainly focusing on objects/instances. In addition, benefiting from the *token interaction* (i.e., Eq. 2), $\boldsymbol{Q}_{bal}$ has learned rich semantic and contextual interaction relations. Therefore, $\boldsymbol{Q}_{bal}$ is used to guide the optimization of $\boldsymbol{T}_i$. The optimized feature tokens (denoted as $\boldsymbol{T}_{bal}$) is taken as the *Key&Value* of ***TD***.

Firstly, to provide guidance for both detection and segmentation, the mask network and box network are separately applied on $\boldsymbol{Q}_{bal}$ to generate mask guiding token $\boldsymbol{T}_g^{mask}$ and box guiding token $\boldsymbol{T}_g^{box}$:

$$\boldsymbol{T}_g^{mask} = \mathcal{N}_{mask}(\boldsymbol{Q}_{bal}), \tag{5}$$

$$\boldsymbol{T}_g^{box} = \mathcal{N}_{box}(\boldsymbol{Q}_{bal}), \tag{6}$$

where $\mathcal{N}_{mask}$ and $\mathcal{N}_{box}$ indicate the mask network and box network, respectively. Both $\mathcal{N}_{mask}$ and $\mathcal{N}_{box}$ consist of a mlp network.

Then, the overall guiding token $\boldsymbol{T}_g$ is obtained by fusing $\boldsymbol{T}_g^{mask}$ and $\boldsymbol{T}_g^{box}$:

$$\boldsymbol{T}_g = \boldsymbol{T}_g^{mask} + \boldsymbol{T}_g^{box}. \tag{7}$$

Finally, $\boldsymbol{T}_g$ guides the optimization of the initial feature tokens $\boldsymbol{T}_i$ through a Multi-Head Cross-Attention. The motivation is straightforward. Same with $\boldsymbol{Q}_{bal}$, each token in $\boldsymbol{T}_g$ corresponds to an object/instance candidate. When $\boldsymbol{T}_i$ interacts with $\boldsymbol{T}_g$, the tokens (in $\boldsymbol{T}_i$) that belong to the same object/instance will be aggregated, enhancing the foreground information. For a better comprehension, a token in $\boldsymbol{T}_g$ could be assumed as the center of a "cluster", and the tokens (in $\boldsymbol{T}_i$) belonging to the same object/instance could be assumed as the points in the "cluster". The points move towards the "cluster" center, realizing the optimization of $\boldsymbol{T}_i$. This procedure is formulated as follows:

$$\boldsymbol{T}_{bal} = \text{MHCA}(\boldsymbol{T}_i, \boldsymbol{T}_g), \tag{8}$$

generating balance-aware feature tokens $\boldsymbol{T}_{bal}$ (also called optimized feature tokens), which serve as the *Key&Value* of ***TD***.

3.2.4 Transformer Decoder

TD is responsible for the predictions of instance mask, object box, and class. ***TD*** consists of decoder layers and each layer contains a self-attention, a cross-attention, and a FFN. The inputs of ***TD*** are $\boldsymbol{T}_{bal}$ (in Eq. 8) and $\boldsymbol{Q}_{bal}$ (in Eq. 4). $\boldsymbol{Q}_{bal}$ interacts with $\boldsymbol{T}_{bal}$ in the decoder layers, continuously refining the query:

$$\boldsymbol{Q}_{ref} = \mathcal{N}_{de}(\boldsymbol{Q}_{bal}, \boldsymbol{T}_{bal}), \tag{9}$$

where $\boldsymbol{Q}_{ref}$ is the refined query, and $\mathcal{N}_{de}$ denotes the transformer decoder network.

Subsequently, we follow the detection head and segmentation head structures of MaskDINO to perform object detection and instance segmentation. For object detection, $\boldsymbol{Q}_{ref}$ is used to predict the categories $\boldsymbol{c}$ and bounding boxes $\boldsymbol{b}$:

$$\{\boldsymbol{c}, \boldsymbol{b}\} = \mathcal{N}_{det}(\boldsymbol{Q}_{ref}), \tag{10}$$

where $\mathcal{N}_{det}$ denotes the detection head network. For instance segmentation, $\boldsymbol{Q}_{ref}$, $\boldsymbol{T}_i$, and the 1/4 resolution CNN feature $\boldsymbol{F}_{cnn}$ are used to predict the instance masks $\boldsymbol{m}$:

$$\boldsymbol{m} = \mathcal{N}_{seg}(\boldsymbol{Q}_{ref}, \boldsymbol{T}_i, \boldsymbol{F}_{cnn}), \tag{11}$$

where $\mathcal{N}_{seg}$ denotes the segmentation head network.

4 Experiments

4.1 Settings

We conduct extensive experiments on COCO [26] and BDD100K [49] datasets using ResNet50 [15] backbone pretrained on ImageNet-1k [40] as well as SwinL [30] backbone pretrained on ImageNet-22k. NVIDIA RTX3090 GPUs are used when the backbone is ResNet50. Due to the large memory

requirement of SwinL, NVIDIA RTX A6000 GPUs are used when the backbone is SwinL. More implementation details are in Appendix B.

Table 1: Comparison with other methods on the COCO validation set.

Methods	Epochs	AP^{box}	AP^{box}_S	AP^{box}_M	AP^{box}_L	AP^{mask}	AP^{mask}_S	AP^{mask}_M	AP^{mask}_L	FPS
ResNet50 backbone										
Mask RCNN [16]	36	41.0	24.9	43.9	53.3	37.2	18.6	39.5	53.3	21.0
HTC [5]	36	44.9	-	-	-	39.7	22.6	42.2	50.6	5.5
SOLQ [10]	50	48.5	30.1	51.6	64.8	40.1	20.9	43.7	59.4	7.0
DINO [52]	36	50.9	34.6	54.1	64.6	-	-	-	-	6.8
Mask2Former [8]	50	-	-	-	-	43.7	23.4	47.2	64.8	5.3
MaskDINO [25]	12	45.7	-	-	-	41.4	21.1	44.2	61.4	8.0
DI-MaskDINO (Ours)	12	**46.9 (+1.2)**	28.8	49.5	62.9	**42.3 (+0.9)**	22	44.8	62.8	7.7
MaskDINO [25]	24	48.4	-	-	-	44.2	23.9	47.0	64.0	8.0
DI-MaskDINO (Ours)	24	**49.6 (+1.2)**	31.7	52.6	65.3	**44.8 (+0.6)**	24.3	47.8	65.0	7.7
MaskDINO [25]	50	51.7	34.2	54.7	67.3	46.3	26.1	49.3	66.1	8.0
DI-MaskDINO (Ours)	50	**51.9 (+0.2)**	36.3	54.7	66.7	**46.7 (+0.4)**	27.5	49.8	66.2	7.7
SwinL backbone										
MaskDINO [25]	12	52.2	34.8	55.6	69.9	47.2	26.3	50.3	69.1	3.4
DI-MaskDINO (Ours)	12	**53.3 (+1.1)**	36.7	56.7	70.4	**47.9 (+0.7)**	27.7	51.1	69.3	3.0
MaskDINO [25]	50	56.8	40.2	60.2	72.3	51.0	31.3	54.1	71.2	3.4
DI-MaskDINO (Ours)	50	**57.8 (+1.0)**	41.5	61.2	73.9	**51.8 (+0.8)**	31.8	55.1	72.2	3.0

Table 2: Comparison with other methods on the BDD100K validation set.

Methods	Epochs	AP^{box}	AP^{box}_S	AP^{box}_M	AP^{box}_L	AP^{mask}	AP^{mask}_S	AP^{mask}_M	AP^{mask}_L	FPS
ResNet50 backbone										
Mask RCNN [16]	50	25.5	15.7	32.8	56.1	20.7	15.7	26.6	49.1	20.1
HTC [5]	50	26.9	15.7	35.4	55.3	21.1	11.0	26.7	46.4	4.8
SOLQ [10]	50	27.0	16.5	35.3	45.6	19.6	10.1	25.8	37.4	6.3
DINO [52]	36	28.9	18.0	37.0	48.1	-	-	-	-	6.1
Mask2Former [8]	50	-	-	-	-	19.6	8.4	25.9	41.0	4.7
MaskDINO [25]	68	28.1	17.4	36.1	47.9	25.3	14.2	31.8	48.1	6.7
DI-MaskDINO (Ours)	68	**29.5 (+1.4)**	18.0	37.4	50.4	**25.7 (+0.4)**	14.5	32.1	48.1	6.4
SwinL backbone										
MaskDINO	68	30.2	19.0	37.5	48.6	27.0	15.4	32.6	50.5	3.2
DI-MaskDINO (Ours)	68	**31.4 (+1.2)**	19.4	40.4	48.7	**27.9 (+0.9)**	16.6	34.1	51.2	2.8

4.2 Comparison Experiments

MaskDINO is the SOTA model for joint object detection and instance segmentation, thus we mainly compare our model with MaskDINO under different backbones (ResNet50 and SwinL). Additionally, our model is compared with some classical (i.e., Mask RCNN [16]) and recently-proposed (i.e., HTC [5] and SOLQ [10]) joint object detection and instance segmentation models. Furthermore, our model is compared with SOTA model that is specifically designed for object detection (i.e., DINO [52]) and instance segmentation (i.e., Mask2Former [8]). The comparison results on COCO dataset are summarized in Tab. 1. It is noted that the experiments with the Swin-L backbone are conducted on the A6000 GPUs with the batch size of 4 (the maximum bacth size that 4 A6000 GPUs supports). The batch size is smaller than that in MaskDINO paper (i.e., batch size = 16) and the 4 A6000 GPUs present weaker computation power than 4 A100 GPUs, thus the results we reproduced are lower than those in the original MaskDINO paper. We can observe that **1)** our model surpasses MaskDINO under different training conditions (epoch = 12, 24, and 50). Notably, our model presents more significant advantage with the training condition of epoch = 12, which potentially reveals that our model reaches the convergence with a faster speed; **2)** under the Swin-L backbone, **DI-MaskDINO** exhibits significant superiority over MaskDINO, further confirming the effectiveness of our model; **3)** the performance of our model on individual detection and segmentation tasks is simultaneously higher than that of SOTA models specifically designed for detection (i.e., DINO) and segmentation (i.e., Mask2Former) when they are fully trained (epoch = 36 or 50), which is really hard-won since the single-task model usually designs the specific module for the specific task (e.g., DINO uses the tailored query formulation to improve the detection performance and Mask2Former proposes tailored masked attention module to improve the segmentation performance).

https://doi.org/10.52202/079017-1930

Existing joint detection and segmentation models like [5, 10, 50] only conduct the experiments on COCO dataset. In this paper, to further verify the robustness and generalization of our model, additional experiments are conducted on more complex traffic scene dataset BDD100K [49] using ResNet50 and Swin-L backbones, and the results we reproduce are shown in Tab. 2. Due to the complexity of traffic scenes, the overall performance is lower than the performance on COCO dataset, and the model asks for more training epochs (epoch = 68) to reach the convergence. It can be observed that our model still exhibits superiority over MaskDINO, DINO, and Mask2Former, which presents the robustness and generalization of our model. It should be noted that the performance of MaskDINO on detection task is lower than that of the specialized object detection model DINO, indicating that DINO still exhibits the advantage in complex traffic scene datasets. In contrast, our model improves DINO by 0.6 AP^{box}, further demonstrating the effectiveness of our model.

4.3 Diagnostic Experiments

4.3.1 Imbalance Tolerance Test

There exists the natural imbalance between object detection and instance segmentation, and we are interested in how will a model perform if the imbalance condition is worsened. Therefore, we conduct the imbalance tolerance test by designing two severe imbalance conditions: **1)** *loss weight constraint*, which is implemented by constraining the weight of detection loss to 1/10 of the default value while the weight of segmentation loss remains unchanged; **2)** *position token constraint*, position token conveys important cues of object locations, thus constraining position token will generate disturbing location information to confuse detection task. The position token constraint is implemented by randomly initializing position token of $\boldsymbol{Q}_{bal}$ (composed of position token and content token) in Eq. 4. ***DI*** module is mainly responsible for alleviating imbalance issue, thus the imbalance tolerance test on **DI-MaskDINO** only enables ***DI*** module. The experiments are conducted on more complex BDD100K dataset, because the results on the more complex dataset can better reflect the performance of imbalance tolerance. Considering the imbalance issue is severe at the beginning decoder layer, thus the experiments utilize models configured with 3 decoder layers.

Table 3: Imbalance tolerance test comparison of MaskDINO and **DI-MaskDINO**.

Imbalance conditions	MaskDINO		DI-MaskDINO (Ours)	
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
standard	27.5	23.7	27.9	24.9
loss weight constraint	24.7 (**-10.2%**)	23 (**-3.0%**)	27.1 (**-2.9%**)	25.2 (**+1.2%**)
position token constraint	21.5 (**-21.8%**)	21.9 (**-7.6%**)	23.8 (**-14.7%**)	23.7 (**-4.8%**)

The results of imbalance tolerance test are summarized in Tab. 3, and the percentage of performance drops (compared with standard condition) is highlighted in colors. We can observe that **1)** the imbalance between detection and segmentation has remarkable affect on the upper bound of model performance, potentially indicating the significance of our work; **2)** the effects of imbalance conditions on detection task are larger than that on segmentation task, because the two imbalance conditions are implemented to mainly constrain the detection task to simulate the natural detection-segmentation imbalance; **3)** even the performance of SOTA model MaskDINO is largely affected by the imbalance conditions (e.g., **-21.8%** AP^{box} degradation on the condition of position token constraint), which potentially reflects that ***De-Imbalance*** deserves the research focus; **4)** compared with MaskDINO, the performance degradation of our model is slighter (i.e., **-10.2%** v.s. **-2.9%** and **-21.8%** v.s. **-14.7%** on the AP^{box} metric, **-3.0%** v.s. **+1.2%** and **-7.6%** v.s. **-4.8%** on the AP^{mask} metric), which demonstrates the effectiveness of our model; **5)** from a comprehensive perspective, we think the standard condition is still a detection-segmentation imbalance condition (which is commonly treated as a regular condition in previous works), and we claim the imbalance is one of the cruxes that limit the upper bound of model performance, hence it should be further studied.

4.3.2 Diagnostic Experiments on Main Modules

To test the effects of main modules in our model (i.e., ***DI*** and ***BATO***), we test the performance of our model under four configurations: **#1** exclusion of both ***DI*** and ***BATO***; **#2** exclusion of ***BATO***; **#3** exclusion of ***DI***; **#4** inclusion of both ***DI*** and ***BATO***. To make the results solid, the experiments are conducted on both BDD100K and COCO datasets, and the results are reported in Tab. 4.

Table 4: The results of diagnostic experiments on main modules. The experiments are conducted on the BDD100K dataset with 68 training epochs and on COCO dataset with 12 training epochs. The results in Tab. 5 and Tab. 6 are also obtained under the same experiment settings.

ID	*DI*	*BATO*	BDD100K		COCO	
			AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
#1	-	-	27.8	24.4	45.6	41.2
#2	✓	-	28.8	25.2	46.4	42.1
#3	-	✓	28.3	24.9	46.2	41.8
#4	✓	✓	**29.5**	**25.7**	**46.9**	**42.3**

In comparison with **#1**, the model under the configuration of **#2** or **#3** yields higher performance on both datasets, and the optimum results are achieved when both ***DI*** and ***BATO*** are enabled (**#4**). These results demonstrate the effectiveness of ***DI*** and ***BATO***. The results are explainable. ***DI*** module alleviates the imbalance between detection and segmentation, generating balance-aware query $\boldsymbol{Q}_{bal}$, which is then fed to ***BATO*** to further make use of balance-aware information, contributing to performance improvement.

4.3.3 Diagnostic Experiments on *DI* Module

The core of our model is ***DI***, which improves the model performance by mitigating the imbalance between detection and segmentation. ***DI*** is realized by applying the ***residual double-selection*** mechanism on $\boldsymbol{T}_i$, generating $\boldsymbol{T}_{s1}$ (firstly-selected tokens), $\boldsymbol{T}_{s2}$ (secondly-selected tokens), and $\boldsymbol{Q}_{bal}$ (balance-aware query). To analyze ***DI*** module, we design the fine-grained ablation experiments by respectively using $\boldsymbol{T}_i$, $\boldsymbol{T}_{s1}$, $\boldsymbol{T}_{s2}$, and $\boldsymbol{Q}_{bal}$ as the guidance for ***BATO*** (i.e., $\boldsymbol{gui.} = \boldsymbol{T}_i$, $\boldsymbol{gui.} = \boldsymbol{T}_{s1}$, $\boldsymbol{gui.} = \boldsymbol{T}_{s2}$, and $\boldsymbol{gui.} = \boldsymbol{Q}_{bal}$) and examine the corresponding performance.

Table 5: The results of diagnostic experiments on ***DI*** module. $gui.$ denotes the **guidance** in Fig. 2.

Guidance	BDD100K		COCO	
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
$\boldsymbol{gui.} = \boldsymbol{T}_i$	28.3	24.9	46.2	41.8
$\boldsymbol{gui.} = \boldsymbol{T}_{s1}$	28.5	24.9	46.6	42.0
$\boldsymbol{gui.} = \boldsymbol{T}_{s2}$	28.9	25.6	46.7	42.2
$\boldsymbol{gui.} = \boldsymbol{Q}_{bal}$	**29.5**	**25.7**	**46.9**	**42.3**

The experiment results on BDD100K and COCO datasets are reported in Tab. 5. $\boldsymbol{gui.} = \boldsymbol{T}_i$ actually represents the situation when ***DI*** module is disabled, which serves as the baseline for other situations. Firstly, we can observe $\mathcal{P}(\boldsymbol{gui.} = \boldsymbol{T}_{s2}) > \mathcal{P}(\boldsymbol{gui.} = \boldsymbol{T}_{s1}) > \mathcal{P}(\boldsymbol{gui.} = \boldsymbol{T}_i)$ where $\mathcal{P}(*)$ denotes the performance of the model under the configuration $*$, demonstrating our ***double-selection*** mechanism is effective. The reason is intuitive, by applying ***double-selection*** mechanism, the tokens with high confidence are selected, and high-confidence tokens indicate the location of objects more clearly than other tokens, thus benefiting the object detection task (i.e., mitigating the imbalance between detection and segmentation). Secondly, the highest performance is achieved when $\boldsymbol{gui.} = \boldsymbol{Q}_{bal}$, validating the effectiveness of our ***residual double-selection*** mechanism. In ***DI*** module, apart from the secondly-selected tokens $\boldsymbol{T}_{s2}$, the initial feature tokens $\boldsymbol{T}_i$ is also used to compute $\boldsymbol{Q}_{bal}$, which could be coarsely formulated as $\boldsymbol{Q}_{bal} = \boldsymbol{T}_i + \mathcal{S}(\boldsymbol{T}_i)$. This residual structure enables the model to make use of the information in both the initial feature tokens and the selected feature tokens, hence reaching the optimal performance.

4.3.4 Diagnostic Experiments on *BATO* Module

BATO targets to use the balance-aware query $\boldsymbol{Q}_{bal}$ to guide the optimization of the initial feature tokens $\boldsymbol{T}_i$. The effectiveness of ***BATO*** has been validated in Tab. 4. We further conduct experiments to validate the effect of the proposed guiding token generation (GTG). The GTG is designed to provide guidance for both detection and segmentation, generating mask guiding token and box guiding token that are closely related to mask instances and object boxes through the mask network and box network, respectively. These guiding tokens can provide more global and semantic guiding information for the optimization of the initial feature tokens $\boldsymbol{T}_i$. As shown in Tab. 6, the model with GTG performs better, which demonstrates the effect of GTG.

Table 6: The results of diagnostic experiments on ***BATO*** module.

Configurations	BDD100K		COCO	
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
w/o GTG	28.6	25.4	46.5	42.2
w/ GTG	**29.5**	**25.7**	**46.9**	**42.3**

https://doi.org/10.52202/079017-1930

5 Conclusion

In this paper, we investigate the naturally-existing but commonly-ignore detection-segmentation imbalance issue. The imbalance means that the performance of object detection lags behind that of instance segmentation at the beginning layer of transformer decoder, which is one of cruxes that hurt the cooperation of object detection and instance segmentation tasks and might constrain the breakthrough of the performance upper bound. To address the issue, we propose **DI-MaskDINO** model with the ***residual double-selection*** mechanism to alleviate the imbalance, achieving significant performance improvements compared with SOTA joint object detection and instance segmentation model MaskDINO, SOTA object detection model DINO, and SOTA segmentation model Mask2Former.

Limitations. This paper focuses on the task of joint object detection and instance segmentation, thus the model is not applicable for other segmentation tasks such as semantic segmentation and panoptic segmentation.

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1483–1498, 2019.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020.

[4] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8573–8581, 2020.

[5] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019.

[6] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group detr: Fast training convergence with decoupled one-to-many label assignment. *arXiv preprint arXiv:2207.13085*, 2022.

[7] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2061–2069, 2019.

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022.

[9] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.

[10] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Solq: Segmenting objects by learning queries. In *Proceedings of the Neural Information Processing Systems*, volume 34, pages 21898–21909, 2021.

[11] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6910–6919, 2021.

[12] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 642–651, 2019.

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[14] Junjie He, Pengyu Li, Yifeng Geng, and Xuansong Xie. Fastinst: A simple query-based model for real-time instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23663–23672, 2023.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[17] Jie Hu, Liujuan Cao, Yao Lu, ShengChuan Zhang, Yan Wang, Ke Li, Feiyue Huang, Ling Shao, and Rongrong Ji. Istr: End-to-end instance segmentation with transformers. *arXiv preprint arXiv:2105.00637*, 2021.

[18] Zhengdong Hu, Yifan Sun, Jingdong Wang, and Yi Yang. Dac-detr: Divide the attention layers and conquer. *Proceedings of the Neural Information Processing Systems*, 36, 2024.

[19] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6409–6418, 2019.

[20] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023.

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

[22] Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask transfiner for high-quality instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4412–4421, 2022.

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

[24] Feng Li, Ailing Zeng, Shilong Liu, Hao Zhang, Hongyang Li, Lei Zhang, and Lionel M Ni. Lite detr: An interleaved multi-scale encoder for efficient detr. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18558–18567, 2023.

[25] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3041–3050, 2023.

[26] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[28] 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. In *Proceedings of the International Conference on Learning Representations*, 2022.

 https://doi.org/10.52202/079017-1930

[29] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3496–3504, 2017.

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, 2019.

[32] Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jinman Wei, Cheng Cui, Yuning Du, Qingqing Dang, and Yi Liu. Detrs beat yolos on real-time object detection. *arXiv preprint arXiv:2304.08069*, 2023.

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

[34] Zhixiong Nan, Jizhi Peng, Jingjing Jiang, Hui Chen, Ben Yang, Jingmin Xin, and Nanning Zheng. A joint object detection and semantic segmentation model with cross-attention and inner-attention mechanisms. *Neurocomputing*, 463:212–225, 2021.

[35] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Proceedings of the Neural Information Processing Systems*, volume 30, 2017.

[36] Jizhi Peng, Zhixiong Nan, Linhai Xu, Jingmin Xin, and Nanning Zheng. A deep model for joint object detection and semantic segmentation in traffic scenes. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8, 2020.

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

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the Neural Information Processing Systems*, volume 28, 2015.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[41] Ganesh Sistu, Isabelle Leang, and Senthil Yogamani. Real-time joint object detection and semantic segmentation network for automated driving. *arXiv preprint arXiv:1901.03912*, 2019.

[42] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14454–14463, 2021.

[43] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Neural Information Processing Systems*, volume 30, 2017.

[45] Shaoru Wang, Yongchao Gong, Junliang Xing, Lichao Huang, Chang Huang, and Weiming Hu. Rdsnet: A new deep architecture forreciprocal object detection and instance segmentation. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, volume 34, pages 12208–12215, 2020.

[46] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 649–665, 2020.

[47] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, volume 36, pages 2567–2575, 2022.

[48] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[49] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020.

[50] Xiaodong Yu, Dahu Shi, Xing Wei, Ye Ren, Tingqun Ye, and Wenming Tan. Soit: Segmenting objects with instance-aware transformers. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, volume 36, pages 3188–3196, 2022.

[51] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 949–958, 2022.

[52] 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. In *Proceedings of the International Conference on Learning Representations*, 2023.

[53] Hao Zhang, Feng Li, Huaizhe Xu, Shijia Huang, Shilong Liu, Lionel M Ni, and Lei Zhang. Mp-former: Mask-piloted transformer for image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18074–18083, 2023.

[54] Dehua Zheng, Wenhui Dong, Hailin Hu, Xinghao Chen, and Yunhe Wang. Less is more: Focus attention for efficient detr. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6674–6683, 2023.

[55] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021.

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

 https://doi.org/10.52202/079017-1930

Appendix

Due to the space limitation of the main text, we provide more results and discussions in the appendix, which are organized as follows:

- Section A: Prior Knowledge.
- Section B: Experiment Settings.
  - Section B.1: Datasets and Metrics.
  - Section B.2: Implementation Details.
- Section C: Additional Diagnostic Experiments.
  - Section C.1: Diagnostic Experiments on Token Selection.
  - Section C.2: Diagnostic Experiments on the Number of Decoder Layer.
- Section D: Visualization Analysis.

A Prior Knowledge

Encoder and decoder of DETR-like models. MaskDINO is a type of DETR-like models [24, 54, 23, 47, 56, 51, 18]. A DETR-like model usually contains a backbone, an encoder, a decoder, and multiple prediction heads. The encoder is composed of multiple transformer encoder layers, and each encoder layer contains a multi-head self-attention and a FFN. The decoder consists of multiple transformer decoder layers, and each decoder layer has an extra multi-head cross-attention compared to the encoder layer.

Token and query. Both token and query are used to represent features. The concept of query comes from the original DETR [3]. Commonly, it denotes the feature of the object/instance in the decoder of DETR-like models. The token is a concept in the field of NLP (Natural Language Processing). In the computer vision domain, a token corresponds to a patch in an image. Feature tokens in this paper represent the features of image patches.

How to obtain the intermediate results from the beginning transformer decoder layer. The transformer decoder in MaskDINO is composed of multiple decoder layers, and MaskDINO attaches prediction heads after each decoder layer. Therefore, we can obtain prediction results from any decoder layer, which are called intermediate results. The intermediate results from the beginning transformer decoder layer are obtained by applying prediction heads on the 0-th decoder layer.

B Experiment Settings

B.1 Datasets and Metrics

COCO [26] is the most widely used dataset for the object detection and instance segmentation tasks, and many well-known models such as [16, 5, 3, 52, 8, 25] are evaluated on the COCO dataset. Following the common practice, we use the COCO train2017 split (118k images) for training and the val2017 split (5k images) for validation. In addition, considering autonomous driving is a typical and practical application of object detection and instance segmentation, the experiments are also conducted on BDD100K [49] dataset, which is composed of 10k high-quality instance masks and bounding boxes annotations for 8 classes. The training set and validation set are divided following the standard in [49]. Consistent with previous researches [5, 10, 11, 50, 45, 25], we report the metrics of AP^{box} and AP^{mask} for performance evaluation.

B.2 Implementation Details

We implement **DI-MaskDINO** based on Detectron2 [48], using AdamW [31] optimizer with a step learning rate schedule. The initial learning rate is set as 0.0001. Following MaskDINO, **DI-MaskDINO** is trained for 50 epochs on COCO with the batch size of 16, decaying the initial learning rate at fractions 0.9 and 0.95 of the total training iterations by a factor of 0.1. For BDD100K, following the setting in [22], we train our model for 68 epochs with the batch size of 8 and the learning rate drops at the 50-th epoch. The number of transformer encoder and decoder layers is 6.

https://doi.org/10.52202/079017-1930

The token numbers of $\boldsymbol{T}_{s1}$ and $\boldsymbol{T}_{s2}$ are 600 and 300, respectively. Unless otherwise specified, the feature channels in both encoder and decoder are set to 256, and the hidden dimension of FFN is set to 2048. The mask network and box network in ***BATO*** are both three-layer mlp networks. We use the same loss function as MaskDINO (i.e., L1 loss and GIOU loss for box loss, focal loss for classification loss, and cross-entropy loss and dice loss for mask loss). Under ResNet50 pretrained on ImageNet [40], our model is trained on NVIDIA RTX3090 GPUs. For Swin-L backbone, NVIDIA RTX A6000 GPUs are used for training and validating.

C Additional Diagnostic Experiments

C.1 Diagnostic Experiments on Token Selection

Token selection is a crucial step in our proposed ***residual double-selection*** mechanism. The number of token selection and the amount of selected tokens may affect the performance. We conduct experiments to verify their impacts. The experimental results are summarized in Tab. 7 and Tab. 8.

The number of token selection. Single-selection actually represents the situation when disabling ***DI*** module. ***Double-selection*** corresponds to our proposed method. Additionally, we add a token selection on $\boldsymbol{T}_{\boldsymbol{s2}}$ for triple-selection. For fair comparison, we set the amount of lastly-selected tokens to the same value (i.e., 300) for single-, double-, and triple-selection. From Tab. 7, we draw two observations: **1)** the results of single-selection are significantly lower than those in other situations, indicating the crucial role of ***DI*** module; **2)** our method achieves the optimal performance with ***double-selection***. The results are explainable. There exists information loss in each selection procedure. Therefore, triple-selection introduces more information loss, leading to a lower performance than ***double-selection***.

The amount of selected tokens. Three k_1 and k_2 settings in the ***double-selection*** mechanism are tested, and their maximum performance gap of AP^{box} on BDD100K dataset is 0.4 (i.e., 29.5-29.1). Similar results are exhibited on the metric of AP^{mask}. These results demonstrate that our method is not sensitive to the hyper-parameters k_1 and k_2. Furthermore, the experiments on COCO dataset also exhibit the similar results, indicating that our method is robust. At last, we explain that the settings of k_1 and k_2 are infinite since they can be set as any value from 1 to 20k. SOTA models such as [18, 25, 24, 14] take 300 as the query number of transformer decoder. In the experiments, k_1 and k_2 are set as 300 or the multiple of 300 to align with the settings of the SOTA models.

Table 7: The results of diagnostic experiments on the number of token selection. Single-, double-, and triple-selection are represented as sing., doub., and trip., respectively. $k_i, i \in [1, 2, 3]$ denotes the amount of selected tokens.

Datasets	Configurations		AP^{box}	AP^{mask}
BDD100K	sing.	k_1=300	28.3	24.9
	doub.	k_1=600,k_2=300	**29.5**	**25.7**
	trip.	k_1=600,k_2=450,k_3=300	29.3	25.5
COCO	sing.	k_1=300	46.2	41.8
	doub.	k_1=600,k_2=300	**46.9**	**42.3**
	trip.	k_1=600,k_2=450,k_3=300	46.6	42.2

Table 8: The results of diagnostic experiments on the amount of selected tokens with our proposed ***double-selection*** mechanism.

Datasets	Configurations	AP^{box}	AP^{mask}
BDD100K	k_1=300,k_2=300	29.1	25.5
	k_1=600,k_2=600	29.3	25.3
	k_1=600,k_2=300	29.5	25.7
COCO	k_1=300,k_2=300	46.6	42.3
	k_1=600,k_2=600	46.8	42.4
	k_1=600,k_2=300	46.9	42.3

C.2 Diagnostic Experiments on the Number of Decoder Layer

We study the effect of different decoder layer numbers for the model performance, and the results are summarized in Tab. 9. We can observe the following: **1)** increasing the number of decoder layers

https://doi.org/10.52202/079017-1930

from 6 to 9 on BDD100K dataset results in the performance degradation, which can be explained by the inconsistency between the complexity of the model and the dataset. The BDD100K dataset only contains 7k training sets and 1k validation sets. The size of BDD100K is small and the model with 9 decoder layers is relatively more complex, leading to the overfitting of the training set; **2)** increasing the number of decoder layers will contribute to both detection and segmentation on COCO. However, the model configured with 9 decoder layers only achieves a slight improvement and introduces more computation cost. Therefore, we use 6 decoder layers in our model; **3)** our model has achieved comparable performance in the configuration with 3 decoder layers compared to MaskDINO with 9 decoder layers (e.g., 45.8 v.s. 45.7 on the AP^{box} metric and 41.3 v.s. 41.4 on the AP^{mask} metric on COCO), demonstrating that our model greatly improves the efficiency of the decoder.

Table 9: The results of diagnostic experiments on the number of decoder layer.

Decoder layer	BDD100K		COCO	
	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
3	28.7	25.3	45.8	41.3
6	29.5	25.7	46.9	42.3
9	28.9	25.6	46.9	42.5

D Visualization Analysis

We visualize the predictions of MaskDINO and **DI-MaskDINO** to show qualitative comparison on BDD100K dataset. As shown in Fig. 3, MaskDINO produces boxes that do not tightly encompass the objects (i.e., Fig. 3a) or do not fully surround the objects (i.e., Fig. 3b). Compared to MaskDINO, our model produces perfectly-fitting boxes, demonstrating the effectiveness of **DI-MaskDINO**. In addition, our model focuses attention on the foreground objects with high category scores through the ***residual double-selection*** mechanism that avoids mispredicting the background as a foreground object. Fig. 3c suggests that our proposed ***residual double-selection*** mechanism is effective.

Figure 3: Qualitative comparison between MaskDINO and **DI-MaskDINO** on BDD100K dataset. Suggest zooming in to view this figure for a clearer view of details.

https://doi.org/10.52202/079017-1930

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are accurately made in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This paper discusses the limitations of the work in § 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/079017-1930

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper provides a detailed description of our proposed **DI- MaskDINO** model in § 3 and comprehensive implementation details in § 4.1 and Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: This paper presents a detailed description of our proposed **DI-MaskDINO** model in § 3, along with comprehensive implementation details provided in Appendix B. We will release the source code of our model at the following URL: https://github.com/CQU-ADHRI-Lab/DI-MaskDINO.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper specifies all the training and test details in § 4 and Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because it would be too computationally expensive and the previous researches in the field of object detection and instance segmentation did not report error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

https://doi.org/10.52202/079017-1930

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources needed to reproduce the experiments in § 4 and Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our research focuses only on improving the overall performance of joint object detection and instance segmentation models. As such, it does not have direct societal impacts beyond the scope of enhancing these technical capabilities.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original paper that produced the code package or dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

https://doi.org/10.52202/079017-1930

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release the source code of our model at the following URL: `https://anonymous.4open.science/r/DI-MaskDINO-12E4`.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-1930