Salience DETR: Enhancing Detection Transformer with Hierarchical Salience Filtering Refinement

Xiuquan Hou[1], Meiqin Liu[1,2,*], Senlin Zhang[2], Ping Wei[1], Badong Chen[1]
[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications,
and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
[2]College of Electrical Engineering, Zhejiang University, Hangzhou, China
xiuqhou@stu.xjtu.edu.cn, liumeiqin@zju.edu.cn, slzhang@zju.edu.cn,
pingwei@mail.xjtu.edu.cn, chenbd@mail.xjtu.edu.cn

Abstract

DETR-like methods have significantly increased detection performance in an end-to-end manner. The mainstream two-stage frameworks of them perform dense self-attention and select a fraction of queries for sparse cross-attention, which is proven effective for improving performance but also introduces a heavy computational burden and high dependence on stable query selection. This paper demonstrates that suboptimal two-stage selection strategies result in scale bias and redundancy due to the mismatch between selected queries and objects in two-stage initialization. To address these issues, we propose hierarchical salience filtering refinement, which performs transformer encoding only on filtered discriminative queries, for a better trade-off between computational efficiency and precision. The filtering process overcomes scale bias through a novel scale-independent salience supervision. To compensate for the semantic misalignment among queries, we introduce elaborate query refinement modules for stable two-stage initialization. Based on above improvements, the proposed Salience DETR achieves significant improvements of +4.0% AP, +0.2% AP, +4.4% AP on three challenging task-specific detection datasets, as well as 49.2% AP on COCO 2017 with less FLOPs. The code is available at https://github.com/xiuqhou/Salience-DETR.

1. Introduction

Object detection is a fundamental task in computer vision with numerous practical applications. Despite the significant advancements made by convolutional detectors [1, 21, 23, 24] in recent decades, they are still limited by manually-designed components such as non-maximum suppression [3]. With the recent advent of DEtection TRansformer (DETR) [3], end-to-end transformer-based detectors have shown remarkable performance improvement in the COCO challenge [32, 38].

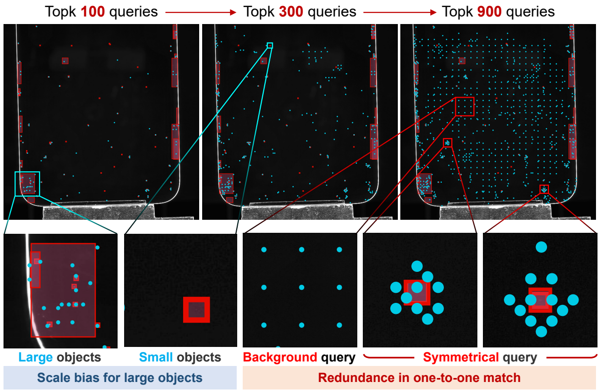

Figure 1. Visualization of *selected queries in two-stage initialization*. Queries and object annotations are denoted in **points** and **bounding boxes** respectively. The selection results illustrate scale bias and redundancy despite one-to-one Hungarian matching.

Among the large number of variants of DETR, the latest high-performance frameworks follow a two-stage pipeline that performs dense self-attention in the encoder and selects sparse queries for cross attention in the decoder [6, 14, 19, 32, 34, 37, 38]. This does improve the detection performance but also results in increased computation and the requirement for stable two-stage query initialization [35]. As shown in Figure 1, we observe that in task-specific detection scenarios involving weak objects (*e.g.* small-scale objects affected by scattering and low contrast [11]), existing two-stage selection results exhibit a significant scale bias towards large objects and redundancy in background

[*] Corresponding author

Table 1. Pre-ablation studies on query number of DINO on MSSD

Two-stage queries	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
300	49.0	78.1	49.0	19.6	28.8	43.3
600	49.4	79.3	48.7	18.9	46.5	43.8
900	51.0	80.0	52.5	20.0	47.4	44.8

and symmetrical queries. This results in unsatisfactory performance due to indiscriminative queries. So, what causes these issues and how can we mitigate them?

We attribute these issues to two types of redundancy in the detection transformer: **encoding redundancy** and **selection redundancy**. It is generally agreed that image foreground contributes more discriminative features for determining object categories and locations than background [15, 26, 35]. Therefore, performing self-attention on background queries may introduce irrelevant and indiscriminative information, which leads to the encoding redundancy. Table 1 shows that DETR-like methods can still benefit from more two-stage queries, even though the number of them has been much larger than that of actual objects. This indicates that the queries selected for two-stage initialization do not exactly match one-to-one with actual objects, *i.e.* selection redundancy. These two redundancies result in a heavy computational burden as well as indiscriminative queries.

Numerous efforts have been made to mitigate redundant calculation and select discriminative queries. For instance, Deformable DETR [37] reduces the complexity from quadratic to linear and explores multi-scale information usage through deformable attention with sparse reference points. Sparse DETR [26] and Focus DETR [35] update only foreground queries for encoding efficiency and achieve comparable precision with much fewer self-attention queries. However, existing query filtering methods apply coarse-grained filtering directly to all tokens, disregarding the multi-scale characteristics where high-level tokens embed more abstract semantics while requiring lower computation compared to low-level tokens [11, 15]. Moreover, scale independence is essential when evaluating query importance for unbiased query selection, while the above methods select queries based on the foreground confidence, which may favor large-scale objects and result in a semantic imbalance. Consequently, query filtering becomes ambiguous and misleading.

To tackle these challenges, this paper proposes a novel detector with hierarchical salience filtering refinement, named Salience DETR. We introduce a salience-guided supervision that is scale-independent to overcome the scale bias during query filtering. With the proposed supervision, a hierarchical query filtering mechanism is proposed to mitigate encoding redundancy by encoding only selected queries. In order to compensate for the semantic misalignment among queries, we propose three elaborate modules to refine queries from the perspectives of multi-scale features, foreground-background differences and selection strategies. Extensive experiments confirm the superior performance and minimal computational cost of Salience DETR.

2. Related Work

2.1. End-to-End detection transformer

DETR (DEtection TRansformer) proposed by Carion *et al.* [3] treats detection as a set prediction task and supervises the training with one-to-one matching through the Hungarian algorithm. Various works have been exploring the transformer-based detectors by accelerating training convergence and improving detection performance [14, 19, 32, 34, 37, 38]. Deformable DETR [37] introduces a framework consisting of two-stage pipeline, deformable attention, and multi-scale tokens that is instructive for subsequent DETR-like methods. Condition-DETR [22] focuses on the extremities of objects through conditional spatial queries to address the slow convergence issue. Since the queries in DETR have no explicit physical meanings, Anchor DETR [30] reintroduces the concepts of anchor query to guide the transformer to focus on specific region modes. DINO [34] integrates dynamic anchor [19] and contrast denoising training [14] to construct a mainstream detection framework and realizes the first state-of-the-art performance among DETR-like methods on COCO. Several recent works focus on improving performance by incorporating training-only designs, such as group queries [6], hybrid query matching [12] and IoU aware BCE loss [2], while leaving the inference process unchanged. Most recently, $\mathcal{C}$o-DETR [38] introduces a versatile label assignment manner that adds parallel auxiliary heads during training and achieves state-of-the-art performance on COCO [16].

2.2. Lightweight Detection Transformer

Despite the promising performance of the transformer, its high calculation complexity and memory cost hinder further applications. As a representative of attention lightweightness, deformable attention [37] attends sparse spatial samplings to reduce the computational and memory complexity. Efficient-DETR [31] optimizes the structure to reduce encoder and decoder layers while maintaining comparable performance. Recent works focus on reducing the number of queries participating in self-attention in the encoder. In particular, Lite DETR [15] prioritizes high-level feature updates to reduce the number of queries. PnP-DETR [29] compresses entire feature maps by abstracting them into a fine-grained vector and a coarse background vector. Sparse DETR [26] refines only the top-ρ% tokens for all encoder layers based on DAM results. Focus DETR [35] further introduces the foreground token selector integrated with a

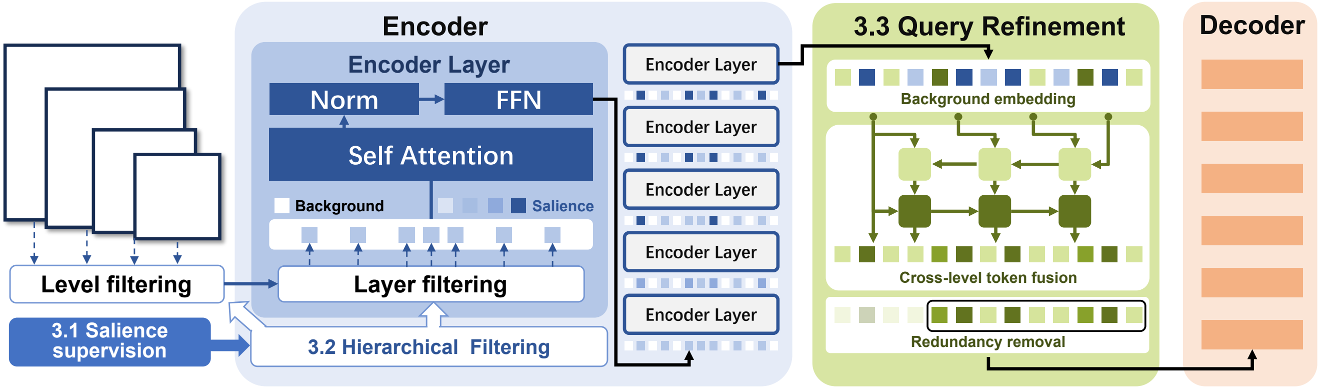

Figure 2. The architecture overview of Salience DETR. We design a hierarchical query filtering for selecting layer-wise and level-wise queries (Section 3.2) under salience-guided supervision (Section 3.1) to mitigate the scale bias in Figure 1. The semantic misalignment among queries is mitigated by query refinement modules (Section 3.3).

cascade set to allocate attention to more informative tokens. However, most of the above methods directly integrate sparsity designs for all tokens while neglecting their multi-scale characteristics.

By contrast, our work performs fine-grained query filtering in both encoding layers and token levels with scale-independent salience supervision and semantic alignment, to address the encoding and selection redundancy.

3. Salience DETR

As depicted in Figure 2, Salience DETR adopts the high-performance two-stage pipeline. The primary architectural difference between Salience DETR and mainstream two-stage DETR-like methods resides in the transformer encoder and query refinement. Given multi-scale features $\{\boldsymbol{f}_l\}_{l=1}^{L}(L=4)$ from the backbone, where $\boldsymbol{f}_l \in \mathbb{R}^{C\times H_l \times W_l}$ denotes the feature map downsampled at scale s_l, the encoder only updates queries selected by hierarchical query filtering (Section 3.2) based on salience-guided supervision (Section 3.1). The semantic misalignment among queries is mitigated through query refinement modules(Section 3.3).

3.1. Salience-guided supervision

Query filtering updates the most informative queries to achieve comparable performance with less computational burden, according to the predicted confidence. Drawing inspiration from Focus DETR [35], we provide supervision for the queries at each level in the multi-scale features. Instead of discrete labels $\{0,1\}$ that only classify foreground and background, we construct a scale-independent salience as supervision targets to overcome the scale bias. In particular, each query $t_l^{(i,j)}$ at position (i,j) in the l-th feature map corresponds to a coordinate $\boldsymbol{c}=(x,y)$ in the original image, denoted as $\left(\lfloor\frac{s_l}{2}\rfloor+i\cdot s_l, \lfloor\frac{s_l}{2}\rfloor+j\cdot s_l\right)$. The salience confidence $\theta_l^{(i,j)}$ of the query is determined according to the following rules:

$$\theta_l^{(i,j)}=\begin{cases} d(\boldsymbol{c},\mathcal{D}_{Bbox}), & \boldsymbol{c}\in\mathcal{D}_{Bbox} \\ \quad 0 & ,\boldsymbol{c}\notin\mathcal{D}_{Bbox}\end{cases} \tag{1}$$

where $\mathcal{D}_{Bbox}=(x,y,w,h)$ denotes the ground truth boxes. We highlight the difference between our scale-independent supervision and discrete foreground-background supervision in red. The salience confidence is calculated through the relative distance to object centers:

$$d(\boldsymbol{c},\mathcal{D}_{Bbox})=1-\sqrt{2\left(\frac{\Delta x}{w}\right)^2+2\left(\frac{\Delta y}{h}\right)^2} \tag{2}$$

where Δx and Δy denote the distance between queries and the center of the corresponding bounding boxes.

Figure 3 illustrates the comparison between salience supervision and discrete supervision. Rather than indistinguishable labels [26, 35] that favor large-scale objects (see Figure 1), our salience supervision ensures fine-grained confidence for objects of different scales, bringing more stable filtering results.

3.2. Heirachical query filtering

Revisting query filtering in Focus DETR. Focus DETR [35] introduces an extra branch that predicts foreground confidence with top-down score modulations on multi-scale features, as follows:

$$\boldsymbol{s}_{l-1}=\mathbf{MLP_F}(\boldsymbol{f}_{l-1}(1+\mathbf{UP}(\alpha_l*\boldsymbol{s}_l))) \tag{3}$$

where **MLP** is a global score predictor, **UP** is bilinear interpolation, and $\{\alpha_l\}_{l=1}^{L-1}$ are learnable modulation coefficients. Based on this, top $\rho\%$ queries with the highest foreground confidence are gathered for transformer encoding and scattered back to update tokens after each encoder layer.

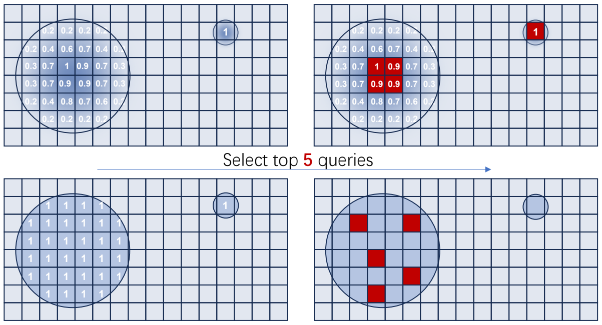

Figure 3. Qualitative illustration of scale-independent supervision (top) and discrete foreground-background supervision (bottom). With salience reducing from the object center to the border, scale-independent supervision balances selected queries even for small-size objects.

Hierarchical query filtering. Typically, high-level tokens bring less calculation burden while preserving more informative semantics than low-level tokens. Therefore, in addition to the traditional layer-wise filtering, a natural motivation is to introduce level-wise filtering for handling multi-scale characteristics [35]. We introduce two sets $\{v_t\}_{t=1}^{T}$ and $\{w_l\}_{l=1}^{L}$ as the corresponding filtering ratios, where T and L denote the number of feature levels and encoder layers. For the t-th encoder layer and l-th feature level, only the top $v_t w_l$ queries are filtered for attention encoding while others are kept unchanged:

$$q_i = \begin{cases} \mathrm{Attention}(q_i + pos_i, \boldsymbol{q} + \boldsymbol{pos}, \boldsymbol{q}), & \text{if } q_i \in \Omega_t \\ \qquad\qquad q_i & \text{, if } q_i \notin \Omega_t \end{cases} \tag{4}$$

where $\boldsymbol{q} = \{q_i\}_{i=1}^{\sum_{l=1}^{L} H_l W_l}$ and $\boldsymbol{pos}$ are queries and the corresponding position encodings, Ω_t is the filtered query set in the t-th encoder layer.

When using deformable attention [37] in the encoder, hierarchical query filtering reduces the encoding computation from $O(\sum_{l=1}^{L} H_l W_l CT(C + KC + 5K + 3MK))$ to $O(\sum_{1\leqslant l\leqslant L}\sum_{1\leqslant t\leqslant T} w_l v_t H_l W_l C(C + KC + 5K + 3MK))$, where M and K denote the number of attention head and sampled keys, and the number of queries becomes only $\frac{\|\boldsymbol{w}\|_1}{T}\frac{\|\boldsymbol{v}*\boldsymbol{s}^2\|_1}{\|\boldsymbol{s}^2\|_1}$ of the original one, with $\boldsymbol{s} = [s_1, \cdots, s_L]$ denoting the downsample scales of the multi-scale features.

3.3. Query refinement

Due to the differences in the process for selected and unselected queries, the hierarchical query filtering may lead to semantic misalignment among queries. Therefore, we propose three refinement modules (*i.e.*, background embedding, cross-level token fusion, and redundancy removal) to bridge the gap from the perspectives of multi-scale features, foreground-background differences and selection strategies, respectively.

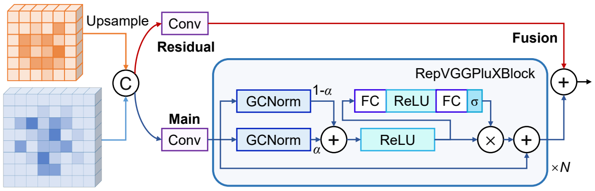

Figure 4. Cross-level token fusion

Background embedding. Given self-learned row embeddings and column embeddings $\boldsymbol{r}, \boldsymbol{c} \in \mathbb{R}^{n\times m}$, where n and m denote the number of embeddings and embedding dimensions respectively, we consider building relative and absolute embeddings to refine queries. The former encodes token $\boldsymbol{f}_l^{(i,j)}$ with relative indexed elements $\boldsymbol{r}^{(in/H_l)}$ and $\boldsymbol{c}^{(jn/W_l)}$ thorugh interpolation.

$$\boldsymbol{b}_l = \underset{(n,n)\to(H_l,W_l)}{\mathrm{Interp}} (\boldsymbol{r} \otimes \boldsymbol{c}) \tag{5}$$

where $\otimes$ denotes outer product, $\boldsymbol{b}_l \in \mathbb{R}^{C\times H_l\times W_l}$. The latter directly encodes $\boldsymbol{f}_l^{(i,j)}$ by concatenating $\boldsymbol{r}^{(i)}$ and $\boldsymbol{c}^{(j)}$.

$$\boldsymbol{b}_l^{(i,j)} = \mathrm{Concat}(\boldsymbol{r}^{(i)}, \boldsymbol{c}^{(j)}) \tag{6}$$

Then the embedding is added to unselected queries for refinement. In our experiments (see Table 8), the absolute embedding achieves a higher detection performance and we choose it as the background embedding of Salience DETR.

Cross-level token fusion. Towards the semantic misalignment of queries at different levels due to the level-specific filtering ratios, we propose a token fusion module that leverages a path aggregation structure [18] to handle cross-level information interaction. In the module, adjacent tokens are fused through a proposed RepVGGPluXBlock, as shown in Figure 4. For adjacent tokens $\boldsymbol{f}_l$ and $\boldsymbol{f}_h$, the calculation is formulated as:

$$\boldsymbol{f}_I^{(0)} = \mathbf{Conv}(\mathbf{Concat}(\boldsymbol{f}_l, \mathbf{UP}(\boldsymbol{f}_h))) \tag{7}$$

$$\boldsymbol{f}_M^{(n)} = \mathcal{R}(\alpha\mathbf{GC}(\boldsymbol{f}_I^{(n)}) + (1-\alpha)\mathbf{GC}(\boldsymbol{f}_I^{(n)})) \tag{8}$$

$$\boldsymbol{f}_I^{(n+1)} = \boldsymbol{f}_M^{(n)} \otimes (\sigma(\mathbf{FC}(\mathcal{R}(\mathbf{FC}(\boldsymbol{f}_M^{(n)}))))) + \boldsymbol{f}_I^{(n)} \tag{9}$$

where $\mathbf{GC}$, $\mathbf{FC}$, $\mathbf{Conv}$, $\mathcal{R}$, σ denote group convolution with batch normalization, dense connection, convolution, ReLU and sigmoid function, respectively. The final refined tokens $\boldsymbol{f}_I^{(N)}$ in the main branch are added with input tokens through a residual branch, as follows.

$$\boldsymbol{f}_O = \boldsymbol{f}_I^{(N)} + \mathbf{Conv}(\mathbf{Concat}(\boldsymbol{f}_l, \mathbf{UP}(\boldsymbol{f}_h))) \tag{10}$$

where N denotes the number of RepVGGPluXBlock.

Redundancy removal for two-stage queries. For those similar objects, especially small-sized objects, the two-stage selection strategy in DETR tends to keep many redundant queries due to their poor discrimination. In addition, one-to-one matching only provides regression supervision on a few positive queries, while massive unsupervised background queries often distribute uniformly like a grid (see Figure 1). Therefore, the transformer decoder suffers from a poor two-stage initialization. Here we simply remove redundancy through non-maximum suppression, and we expect end-to-end solutions to be proposed to deal with the issue. Specifically, we construct a bounding box $\boldsymbol{b}_l^{(i,j)}$ with a distance from the center to the border set to 1 for each selected queries. Then, NMS [24] is applied to the bounding boxes in both image-wise and level-wise manners.

$$Bbox_l^{(i,j)} = [i-1, j-1, i+1, j+1] \tag{11}$$

3.4. Optimization

Similar to other DETR-like detectors, our model is trained with a multi-task loss function, defined as follows:

$$\mathcal{L}_{total} = \lambda_m \mathcal{L}_m + \lambda_{dn} \mathcal{L}_{dn} + \lambda_{enc} \mathcal{L}_{enc} + \lambda_f \mathcal{L}_f \tag{12}$$

$$\mathcal{L}_f = -\alpha_f (1 - p_f)^\gamma \log(p_f) \tag{13}$$

where $\mathcal{L}_f$ denotes the scale-independent supervision loss function, $\alpha_f = 0.25$ and $\gamma = 2$ are focal parameters, and $p_f = \hat{\theta}\theta + (1 - \hat{\theta})(1 - \theta)$, where $\theta_l^{(i,j)}$ is our proposed salience confidence in (1).

4. Experiments and Discussions

This section demonstrates that Salience DETR achieves comparable performance with fewer FLOPs on task-specific and generic detection tasks through quantitative and qualitative analysis, and evaluates the effectiveness of the proposed components through ablation studies.

4.1. Experimental Setup

Dataset and Evaluation Metrics. The evaluation datasets include three task-specific detection datasets (ESD [11], CSD [28], our self-built mobile screen surface dataset (MSSD)), and benchmark COCO 2017 [16]. Details of them are listed in Table 2. We evaluate performance using the standard average precision (AP) and average recall (AR) metrics [16].

Table 2. Statistics of the three task-specific evaluation datasets

Dataset	#ann	#class	#train	#test	Resolution	Obj/img
CSD	4983	3	373	94	1024×1024	10.67
ESD	6075	2	448	49	3620×3700	12.22
MSSD	93343	4	962	106	5120×5120	87.40
COCO	896782	80	118287	5000	-	7.27

Implementation Details. The implementation details of Salience DETR align with other DETR-like detectors. We train our model with NVIDIA RTX 3090 GPU (24GB) using the AdamW [13] optimizer with a weight decay of 1×10^{-4}. The initial learning rate is set to 1×10^{-5} for the backbone and 1×10^{-4} for other parts, which decreases at later stages by 0.1. The batch size per GPU is set to 2. Considering dataset scales, the training epochs on CSD, ESD, and MSSD are 60, 60, 120, respectively. For COCO 2017, we report the results of 12 epochs. The loss coefficients of the salience supervision λ_f is set to 2. Other methods are evaluated on detrex [25] and MMDetection [5] toolbox.

4.2. Comparison with State-of-the-art Methods

Comparison on ESD. Table 3 shows the quantitative comparison on ESD between our Salience DETR and other detectors [4, 9, 12, 14, 17, 19, 24, 32, 34–37]. It can be seen that Salience DETR outperforms the comparison methods under most standard metrics. Notably, considering a strict IoU threshold of 75%, Salience DETR suppresses the second best method with a large margin of 1.8% and becomes the only method with AP$_{75}$ over 40%.

Comparison on CSD. Table 4 shows the results on CSD. It is worth noting that CSD is challenging for DETR-like methods since it lacks large-size objects suitable for DETR detection. Comprehensively speaking, our Salience DETR achieves the highest performance considering average precision (+0.2%) and average recall (+0.2%) and outperforms the latest DETR-like methods [12, 34, 35].

Comparison on MSSD. The MSSD dataset, collected from industrial production lines by ourselves, contains massive small-sized and weak objects with low contrast and indistinguishable contours. Therefore, discriminative query selection is important for detecting this dataset. As can be seen from Table 5, with hierarchical salience filtering refinement, our Salience DETR achieves a superior AP$_{75}$ of 61.9% with a large margin of 9.4% compared to the second best result and improves AP by 4.4%. Moreover, due to the salience-guided supervision that is scale-independent, queries in Salience DETR match small-sized objects better and thus our Salience DETR improves AP$_S$ and AR$_S$ by 8.7% and 9.1% compared to DINO, respectively.

4.3. Ablation Studies

Ablation studies are conducted on CSD [28] using ResNet50 backbone and Table 6 reports the results. As can be seen, by introducing hierarchical query filtering with scale-independent salience supervision, Salience DETR yields a significant improvement of +1.8 AP. Then the background embedding and the redundancy removal steadily increase AP from 52.0% to 52.8% while keeping FLOPs at 168G with no extra computation. Finally, the 6-th row of Table 6 shows that Salience DETR equipped with the cross-

Table 3. Quantitative comparison on ESD. The first and second best results are marked in **Red** and **Blue**, respectively.

Methods	Pub'Year	Backbone	Epochs	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	AR	AR$_S$	AR$_M$	AR$_L$
Faster RCNN [10, 24]	NIPS'2015	ResNet50	60	43.5	84.4	**39.6**	12.5	44.2	52.5	52.0	13.1	53.2	59.0
AutoAssign [36]	Arxiv'2020	ResNet50	60	44.2	86.0	37.5	15.4	44.8	51.8	55.7	24.8	56.2	62.5
Cascade RCNN [32]	CVPR'2018	ResNet50	60	44.5	85.8	39.0	14.8	44.6	55.6	54.4	17.3	55.0	62.3
HTC [4]	CVPR'2019	ResNet50	60	**45.4**	87.1	37.3	15.3	**45.7**	54.7	55.4	26.3	55.7	62.3
RetinaNet [17, 27]	ICML'2019	EfficientNet	60	43.4	87.2	35.7	13.4	43.2	53.2	53.4	23.8	53.6	63.4
YOLOX [9]	Arxiv'2021	CSPDarknet	300	41.6	79.6	36.9	14.9	43.4	50.0	54.3	**36.9**	54.2	60.8
Def-DETR [37]	ICLR'2020	ResNet50	300	42.3	85.8	33.8	**15.5**	42.4	51.2	53.6	24.2	53.9	61.8
DAB-Def-DETR [20]	ICLR'2021	ResNet50	90	39.8	84.6	30.4	7.4	40.0	55.5	52.4	9.4	52.9	66.6
DN-Def-DETR [14]	CVPR'2022	ResNet50	60	40.9	86.2	32.9	10.5	40.9	54.2	56.1	18.1	56.5	65.9
DINO [34]	ICLR'2022	ResNet50	60	42.5	**87.8**	34.1	12.3	42.1	**57.5**	55.0	19.4	54.8	67.2
H-Def-DETR [12]	CVPR'2023	ResNet50	60	41.9	87.7	33.3	12.0	41.8	55.3	54.6	28.1	54.0	66.6
Focus-DETR [35]	ICCV'2023	ResNet50	60	43.3	**88.6**	34.3	11.7	43.3	**57.5**	**57.2**	21.9	**57.0**	**67.4**
Salience DETR	ours	ResNet50	60	**46.5**	**88.6**	**41.4**	**15.4**	**46.8**	**57.7**	**58.4**	**34.0**	**58.2**	**69.5**

Table 4. Quantitative comparison on CSD. The first and second best results are marked in **Red** and **Blue**, respectively.

Methods	Pub'Year	Backbone	Epochs	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	AR	AR$_S$	AR$_M$	AR$_L$
Faster RCNN [10, 24]	NIPS'2015	ResNet50	60	52.4	**93.2**	51.9	48.5	39.7	**70.0**	59.9	57.4	44.0	**70.0**
AutoAssign [36]	Arxiv'2020	ResNet50	60	51.1	92.5	49.5	47.2	40.1	**60.0**	60.1	56.7	58.1	60.0
RetinaNet [17, 27]	ICML'2019	EfficientNet	60	50.5	90.4	54.1	44.0	**40.9**	5.0	59.8	54.3	58.6	20.0
YOLOX [9]	Arxiv'2021	CSPDarknet	300	47.5	90.0	42.4	47.0	36.5	0	56.1	56.6	40.6	0
TOOD [8]	ICCV'2021	ResNet50	60	52.9	**92.9**	52.7	47.7	**41.8**	14.5	60.6	57.0	45.6	60.0
Def-DETR [37]	ICLR'2020	ResNet50	300	43.7	86.2	36.6	40.5	34.9	10.0	56.0	55.3	62.1	10.0
DAB-Def-DETR [20]	ICLR'2021	ResNet50	90	52.9	91.2	55.0	50.3	39.4	0	62.5	60.6	58.1	0
DN-Def-DETR [14]	CVPR'2022	ResNet50	60	49.9	88.0	51.2	47.6	37.7	0	63.7	61.0	**76.3**	0
DINO [34]	ICLR'2022	ResNet50	60	**53.0**	90.8	55.5	50.9	39.6	0	64.0	63.1	59.1	0
H-Def-DETR [12]	CVPR'2023	ResNet50	60	**53.0**	90.6	**55.7**	**51.2**	39.2	6.7	63.2	62.2	45.3	30.0
Focus-DETR [35]	ICCV'2023	ResNet50	60	52.3	91.2	**55.9**	50.3	39.2	0.9	**65.3**	**64.1**	71.2	60.0
Salience DETR	ours	ResNet50	60	**53.2**	92.5	55.1	**51.0**	**40.9**	0	**66.5**	**65.7**	**74.3**	0

level token fusion achieves +0.4 AP. These results demonstrate the effectiveness of our proposed components.

Effect of scale-independent supervision. Unlike assigning queries to feature levels according to their corresponding object scales [35], one of our contributions is introducing a scale-independent supervision that is totally determined by salience. Table 7 compares the effect of them, in which we can see that scale-independent supervi-

Table 5. Quantitative comparison on MSSD. The first and second best results are marked in **Red** and **Blue**, respectively.

Methods	Pub'Year	Backbone	Epochs	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$	AR	AR$_S$	AR$_M$	AR$_L$
Faster RCNN [10, 24]	NIPS'2015	ResNet50	120	44.5	65.3	48.6	18.0	36.7	40.3	51.1	21.3	44.5	46.3
AutoAssign [36]	Arxiv'2020	ResNet50	120	38.4	56.2	40.5	6.3	21.6	40.9	45.7	9.9	37.0	46.6
Cascade RCNN [32]	CVPR'2018	ResNet50	120	47.5	69.5	52.1	**23.5**	42.0	42.0	54.1	**26.1**	51.3	48.3
YOLOX [9]	Arxiv'2021	CSPDarknet	500	41.4	67.5	39.2	13.6	35.5	38.7	53.7	21.5	46.8	48.6
HTC [4]	CVPR'2019	ResNet50	120	47.5	68.1	52.5	19.9	33.2	41.6	53.7	22.7	49.6	46.0
Def-DETR [37]	ICLR'2020	ResNet50	300	33.0	54.3	32.0	9.8	11.1	33.3	39.8	13.6	18.9	38.5
DAB-Def-DETR [20]	ICLR'2021	ResNet50	120	33.7	60.0	31.0	15.9	26.7	29.0	46.2	19.9	38.9	39.8
DN-Def-DETR [14]	CVPR'2022	ResNet50	120	45.6	74.1	44.9	18.6	31.9	41.0	53.9	21.7	50.1	47.7
DINO [34]	ICLR'2022	ResNet50	120	**51.0**	**80.0**	**52.5**	20.0	**47.4**	**44.8**	**61.3**	23.9	**60.1**	**52.4**
H-Def-DETR [12]	CVPR'2023	ResNet50	120	46.9	76.8	47.1	20.3	45.3	40.7	57.1	25.8	57.5	49.0
Focus-DETR [35]	ICCV'2023	ResNet50	120	49.2	**79.3**	47.9	18.3	40.7	43.5	59.4	22.1	53.3	51.1
Salience DETR	ours	ResNet50	120	**55.4**	78.2	**61.9**	**28.7**	**47.5**	**44.5**	**65.4**	**33.0**	**59.8**	**52.1**

Table 6. Ablation results on CSD. HQF: hierarchical query filtering (Section 3.2), BE: background embedding, RR: redundancy removal, CTF: cross-level token fusion

HQF.	BE.	RR.	CTF.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AR	FLOPs
✗	✗	✗	✗	50.2	89.8	49.4	47.8	37.9	64.1	132
✓	✗	✗	✗	52.0	89.7	54.2	49.4	39.8	65.6	168
✓	✓	✗	✗	52.0	90.3	49.9	50.3	39.1	64.5	168
✓	✗	✓	✗	52.6	90.8	52.7	50.9	39.2	64.4	168
✓	✓	✓	✗	52.8	91.0	55.7	51.5	39.3	64.7	168
✓	✓	✓	✓	53.2	92.5	55.1	51.0	40.9	66.5	201

sion brings consistent precision improvements (+1.8% AP, +0.4% AP_{50}, +1.6% AP_{75}) compared to supervision determined according to object scales, confirming its effectiveness to address scale bias in query filtering and benefit final performance.

Table 7. Quantitative comparison of supervision methods on MSSD. The intervals $[[-1, 128], [64, 256], [128, 512], [256, \infty]]$ are used in overlap limit range, following Focus-DETR.

Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Overlap limit [35]	53.2	77.4	60.3	**29.0**	39.4	43.7
Scale-independent	**55.0**	**77.8**	**61.9**	28.1	**40.7**	**44.1**

Effect of background embedding. We analyze the effect of the proposed two variants of background embeddings that compensate for semantic misalignments in query filtering. As shown in Table 8, the absolute embedding could boost AP slightly better than the relative embedding. This may be because the query filtering is performed on pixels and absolute embedding provides direct position information and benefits position-related features.

Table 8. Comparison of embedding strategies on MSSD

Embed Strategies	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Relative embedding	55.0	77.6	62.1	29.4	46.9	44.0
Absolute embedding	55.2	77.6	61.3	29.6	51.4	44.1

Effect of redundancy removal. As mentioned in Section 3.3, the redundancy removal is critical to stabilizing two-stage initialization. From the evaluation metrics AP and AP_{50} in Figure 5, we can see that the proposed redundancy removal could speed up convergence with a significant margin, especially at early stages. Salience DETR with redundancy removal achieves better performances of +0.7% AP and +0.4% AP_{50}, which is mainly attributed to the fact that redundancy removal allows the decoder to focus on unique and relevant features provided by more discriminative queries.

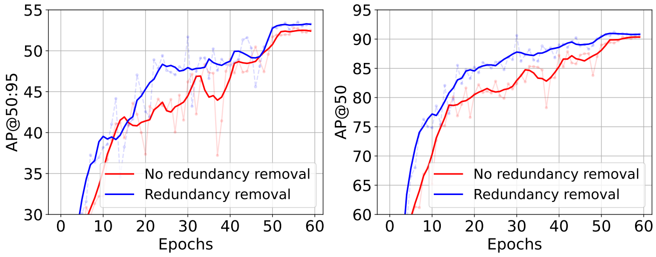

Figure 5. Convergence of Salience DETR

4.4. Scalability of Salience DETR

With designed salience filtering refinement, our Salience DETR can effectively provide well-matched focus for objects in defect detection tasks. Here we illustrate its great scalability to generic large-scale datasets of COCO 2017. As shown in Table 9, our Salience DETR achieves better performance of 49.2% AP compared to other methods with less than 70% FLOPs of DINO [34] under the same setting, demonstrating its superior trade-off between computational complexity and performance.

4.5. Visualization

Salience Confidence. Figure 6 visualizes the salience confidence on MSSD, CSD, and COCO 2017 datasets. The visualization demonstrates that salience guides defect regions to achieve high confidence even for those with small sizes. Additionally, the confidence of large-sized objects decreases from the center to the border, benefiting fine-grained supervision. Interestingly, the salience labels are constructed solely based on bounding box annotations; however, the predicted confidence can further match rough object contours. This suggests the possibility that salience supervision may also benefit pixel-level tasks, such as instance segmentation. Therefore, transitioning from instance-level annotations to pixel-level predictions based on salience is a promising direction for future research.

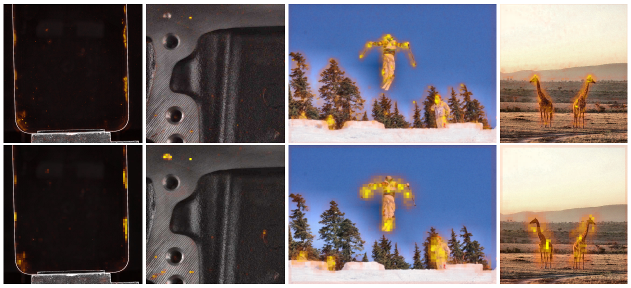

Figure 6. Visualization of salience confidence at multi-scale feature maps on MSSD, CSD and COCO.

Detection results. As shown in Figure 7, the detection results of Salience DETR illustrate a strong adaptability to various appearances of objects, including small size (*e.g.*

Table 9. Quantitative comparison on COCO val2017. Since the FLOPs may differ according to the calculation script, we reimplement DINO and report its FLOPs results using the same script with our Salience DETR.

Method	Pub'Yead	epochs	backbone	AP ↑	AP_{50} ↑	AP_{75} ↑	AP_S ↑	AP_M ↑	AP_L ↑	FLOPs ↓
Conditional-DETR [22]	CVPR'21	108	R50	43.0	64.0	45.7	22.7	46.7	61.5	-
SAM-DETR [33]	CVPR'22	50	R50	41.8	63.2	43.9	22.1	45.9	60.9	-
Anchor-DETR [30]	AAAI'22	50	R50	42.1	63.1	44.9	22.3	46.2	60.0	-
Dynamic-DETR [7]	CVPR'21	12	R50	42.9	61.0	46.3	24.6	44.9	54.4	-
Sparse-DETR [26]	ICLR'21	50	R50	46.3	66.0	50.1	29.0	49.5	60.8	-
Efficient-DETR [31]	Arxiv'21	36	R50	45.1	63.1	49.1	28.3	48.4	59.0	-
Def-DETR [37]	ICLR'20	50	R50	46.9	65.6	51.0	29.6	50.1	61.6	-
DAB-Def-DETR [19]	ICLR'21	50	R50	46.8	66.0	50.4	29.1	49.8	62.3	-
DN-Def-DETR [14]	CVPR'22	50	R50	48.6	**67.4**	52.7	31.0	52.0	**63.7**	-
Focus-DETR [35]	CVPR'23	12	R50	48.8	66.8	52.8	31.7	52.1	63.0	-
H-DETR [12]	CVPR'23	12	R50	48.7	66.4	52.9	31.2	51.5	**63.5**	-
DINO [34]	ICLR'22	12	R50	**49.0**	66.6	**53.5**	**32.0**	**52.3**	63.0	291G
Salience DETR	ours	12	R50	**49.2**	**67.1**	**53.8**	**32.7**	**53.0**	63.1	**201G**

pinhole), indiscriminative contours (*e.g.* tin_ash), intra-class differences (*e.g.* scratch), and clear objects (*e.g.* bubble). Moreover, Salience DETR shows a high level of confidence in distinguishing between categories, which enables it to achieve stable detection for confusable objects.

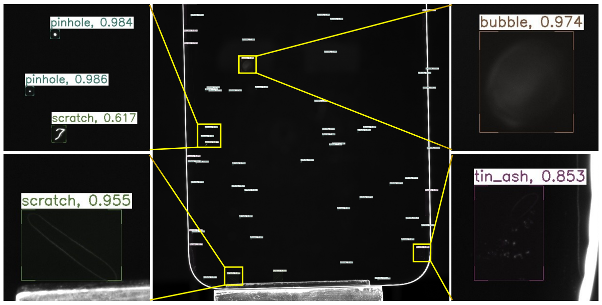

Figure 7. Visualization of detection results on the MSSD dataset.

4.6. Inference cost

Table 10 shows that Salience DETR without cross-level token fusion achieves comparable performance with other approaches at a lower computational cost. Further, introducing the module leads to better performance, with reasonable parameter and memory increase. These results show that Salience DETR obtains a good trade-off between performance and inference cost.

5. Conclusion

The paper develops a transformer-based detection framework, *i.e.* Salience DETR, to mitigate the encoding and selection redundancies in two-stage DETR-like detectors. The key component of Salience DETR, namely hierarchical salience filtering refinement, selectively encodes a fraction of discriminative queries under the supervision of scale-independent salience to overcome scale bias. By conducting background embedding and cross-level token fusion, our model effectively tackles the problem of semantic misalignment among queries at different levels and layers. Furthermore, it leverages an elaborate redundancy removal module to stabilize two-stage initialization. We demonstrate that Salience DETR achieves state-of-the-art performance on three task-specific and one generic object detection datasets, as well as a superior balance between computation and precision. We believe that our work could facilitate future researches on query redundancy in DETR-like methods.

Table 10. Inference time and memory

Methods	AP(CSD)↑	infer time↓	#parameter↓	memory↓
H-Def-DETR	53.0	0.0834	44.4M	201MB
DINO	53.0	0.0860	46M	207MB
Focus DETR	52.3	0.0822	44.4M	201MB
Salience DETR*	52.8	0.0769	46M	209MB
Salience DETR	53.2	0.0819	56.1M	249MB

* denotes a faster variant without the cross-level token fusion.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2020AAA0108302, the National Natural Science Foundation of China under Grant 62327808, and the Fundamental Research Funds for Xi'an Jiaotong University under Grant xtr072022001.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. 1

[2] Zhi Cai, Songtao Liu, Guodong Wang, Zheng Ge, Xiangyu Zhang, and Di Huang. Align-detr: Improving detr with simple iou-aware bce loss. *arXiv preprint arXiv:2304.07527*, 2023. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1, 2

[4] 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. 5, 6

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5

[6] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group detr: Fast detr training with group-wise one-to-many assignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6633–6642, 2023. 1, 2

[7] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2988–2997, 2021. 8

[8] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang. Tood: Task-aligned one-stage object detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3490–3499. IEEE Computer Society, 2021. 6

[9] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 5, 6

[10] 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. 6

[11] Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, and Badong Chen. Canet: Contextual information and spatial attention based network for detecting small defects in manufacturing industry. *Pattern Recognition*, 140:109558, 2023. 1, 2, 5

[12] 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. 2, 5, 6, 8

[13] D Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, page 6. San Diego, California;, 2015. 5

[14] 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. 1, 2, 5, 6, 8

[15] 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. 2

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 5

[17] 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. 5, 6

[18] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. 4

[19] 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 *International Conference on Learning Representations*, 2021. 1, 2, 5, 8

[20] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 6

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 1

[22] 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. 2, 8

[23] 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. 1

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. 1, 5, 6

[25] Tianhe Ren, Shilong Liu, Feng Li, Hao Zhang, Ailing Zeng, Jie Yang, Xingyu Liao, Ding Jia, Hongyang Li, He Cao, Jianan Wang, Zhaoyang Zeng, Xianbiao Qi, Yuhui Yuan, Jianwei Yang, and Lei Zhang. detrex: Benchmarking detection transformers, 2023. 5

[26] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse detr: Efficient end-to-end object detection with learnable sparsity. In *International Conference on Learning Representations*, 2021. 2, 3, 8

[27] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 6

[28] Qishan Wang, Qing Zhao, Weifeng Ge, Xuan Tong, Kingdong Jiang, Chungang Du, and Wenqiang Zhang. Surface defect detection of casting with machined surfaces based on natural artificial defects. *Available at SSRN 4352006*, 2023. 5

[29] Tao Wang, Li Yuan, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Pnp-detr: Towards efficient visual analysis with transformers. In *Proceedings of the IEEE/CVF international Conference on Computer Vision*, pages 4661–4670, 2021. 2

[30] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2567–2575, 2022. 2, 8

[31] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 2, 8

[32] Mingqiao Ye, Lei Ke, Siyuan Li, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Cascade-detr: Delving into high-quality universal object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6704–6714, 2023. 1, 2, 5, 6

[33] 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. 8

[34] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 5, 6, 7, 8

[35] 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. 1, 2, 3, 4, 5, 6, 7, 8

[36] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv preprint arXiv:2007.03496*, 2020. 6

[37] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020. 1, 2, 4, 5, 6, 8

[38] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6748–6758, 2023. 1, 2