Slim-DETR: Real-time Tiny Object Detection with Efficient Interaction and Gaussian Query

Tong Wu[1,2], Jiahao Zhang[1,2], Juntao Guan[1,4*], and Rui Lai[1,3*]

[1] Key Laboratory of Analog Integrated Circuits and Systems, Xidian University, Xi'an, China
[2] Zhejiang Key Laboratory of Analog Integrated Circuits, Hangzhou Institute of Technology, Xidian University, Hangzhou, China
[3] Faculty of Integrated Circuit, Xidian University, Xi'an, China
[4] Hangzhou Institute of Technology, Xidian University, Hangzhou, China
guanjuntao@xidian.edu.cn, rlai@mail.xidian.edu.cn

Abstract. Tiny object detection is critical for applications ranging from drone-based scene analysis to remote sensing. However, existing DETR frameworks suffer from inefficient encoder architectures and a severe query-object misalignment for tiny targets, hindering their deployment in real-time scenarios. To overcome these limitations, we propose Slim-DETR, a novel framework tailored for real-time tiny object detection. Central to our approach is an Efficient Interaction (EI) encoder, which employs a gather-and-inject mechanism to aggregate global semantics. Uniquely, the EI encoder restricts semantic attention interactions exclusively to extracted foreground tokens, thereby significantly alleviating the computational overhead of conventional Transformer-based encoders. Furthermore, we introduce a Gaussian Target Guided (GTG) query selection strategy that leverages the Gaussian spatial distribution of targets to dynamically concentrate high-quality queries on regions containing tiny objects. Extensive experiments demonstrate that Slim-DETR achieves state-of-the-art performance, yielding 33.9% AP on VisDrone and 33.6% AP on AI-TOD-V2. Notably, it maintains real-time inference speeds across platforms, achieving a latency of 24.4 ms on an NVIDIA RTX 3090 GPU and 25.1 ms (40 FPS) on the embedded Jetson AGX Orin.

Keywords: Tiny object detection · Detection Transformer · Real-time detection

1 Introduction

Tiny object detection has become pivotal for various fields, ranging from drone-based real-time scene analysis to military reconnaissance, disaster management, and smart city infrastructure [23,31,32,41,44]. This growing demand underscores the critical need for detectors that can balance real-time performance with high

* Corresponding authors.

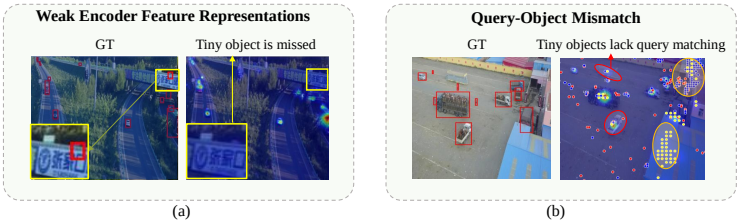

Fig. 1: The previous DETR methods overlooked the problems.

accuracy. However, discriminative features for tiny objects are often confined to extremely small regions (typically less than 32×32), posing significant challenges to achieving this trade-off.

In DETR-based methods, effective incorporation of multi-scale feature interactions, particularly low-level features containing rich local details, is vital for improving the accuracy of tiny object detection. Although deformable attention [45] reduces computational costs per token, the drastically increased sequence length renders the encoder a speed bottleneck.

Recent DETR variants [6, 22, 43] aim to optimize multi-level feature interactions within the encoder. While Sparse-DETR and Salience-DETR [6, 22] focus on extracting sparse salient tokens, RT-DETR [43] adopts a hybrid design combining high-level interaction with FPN-like cross-scale fusion for speed. Nevertheless, both strategies exhibit limitations in detecting tiny objects, resulting in missed detection (Fig. 1(a)). The former suffers from error accumulation across layers and poor generalization due to complex multi-layer supervision [10]. The latter is hindered because its high-level interaction lacks essential local details, forcing the model to rely on indirect pathways to recover low-level information.

To address these issues, we propose Slim-DETR, a novel framework for real-time tiny object detection. A key component of our framework is the Efficient Interaction (EI) encoder, which employs a novel gather-and-inject mechanism. Specifically, this mechanism first aggregates multi-level features to enable global semantic interactions. It then injects the enhanced representations directly into our lightweight cross-level fusion modules, effectively overcoming the limitations of indirect hierarchical information propagation. Complementing this, during attention interaction, the EI encoder restricts semantic computations exclusively to potential foreground tokens extracted from the gathered feature map. This strategy not only circumvents the computational bottleneck imposed by increased feature length but also mitigates the progressive degradation of tiny object features across layers, thereby enabling high-performance real-time inference.

Furthermore, mainstream DETR-like frameworks typically select a fixed number of queries following the encoding stage for decoder prediction. As illustrated in Fig. 1(b), relying primarily on IoU and classification confidence for this selection often leads to a severe query-object mismatch. Such criteria inherently skew the selection towards large-scale objects and background regions, resulting in a significant query allocation imbalance and insufficient query coverage for tiny ob-

jects. To address this, Slim-DETR introduces a Gaussian Target Guided (GTG) query selection strategy. GTG employs a size-decoupled Gaussian distribution to model target likelihood, concentrating the probability mass specifically on tiny object locations. This mechanism amplifies response scores within tiny object regions while suppressing background noise. Consequently, GTG facilitates the dynamic concentration of high-quality queries around tiny objects, effectively resolving the query mismatch problem.

Slim-DETR achieves state-of-the-art performance, striking an optimal accuracy-speed balance for tiny object detection. Our contributions are summarized as follows:

- We pinpoint two critical bottlenecks hindering existing DETR methods in tiny object detection: the computational overhead caused by long sequence lengths in encoder attention interaction, and the severe query-object mismatch inherent in conventional selection strategies.
- We devise a novel Efficient Interaction (EI) encoder with a gather-and-inject mechanism and attention on foreground tokens, eliminating computational bottlenecks for real-time inference while preserving fine-grained details.
- We introduce a Gaussian Target Guided (GTG) query selection strategy, which leverages size-decoupled Gaussian distributions to dynamically concentrate queries on tiny objects, resolving selection mismatch.
- Slim-DETR achieves state-of-the-art results (33.9% AP on VisDrone, 33.6% on AI-TOD-V2) with real-time latency: 24.4 ms on an NVIDIA RTX 3090 GPU and 25.1 ms on the embedded Jetson AGX Orin.

2 Related Work

2.1 Tiny Object Detection

Tiny object detection is hindered by pixel scarcity [1]. Traditional overlap-based matching often fails to provide sufficient positive samples [3, 25], while recent advanced assignment strategies [29, 37] incur heavy computational costs [35]. Although scale-aware architectures [4, 5, 16, 42] and attention mechanisms like CBAM [34] enhance feature representation, mainstream detectors like the YOLO series [27] still depend on Non-Maximum Suppression (NMS). This post-processing step creates a bottleneck, leading to unpredictable inference latency and potential accuracy loss [43], thus limiting their real-time applicability.

2.2 End-to-End Object detectors

DETR frameworks [10,40,45] enable end-to-end object detection via an encoder-decoder architecture, eliminating the need for NMS while leveraging self-attention to capture powerful semantic information [8,43]. However, detecting tiny objects necessitates high-resolution features, imposing substantial computational overhead on the encoder. While recent studies [6, 22] alleviate this burden through sparse attention, they suffer from accumulated miss rates across layers, and their

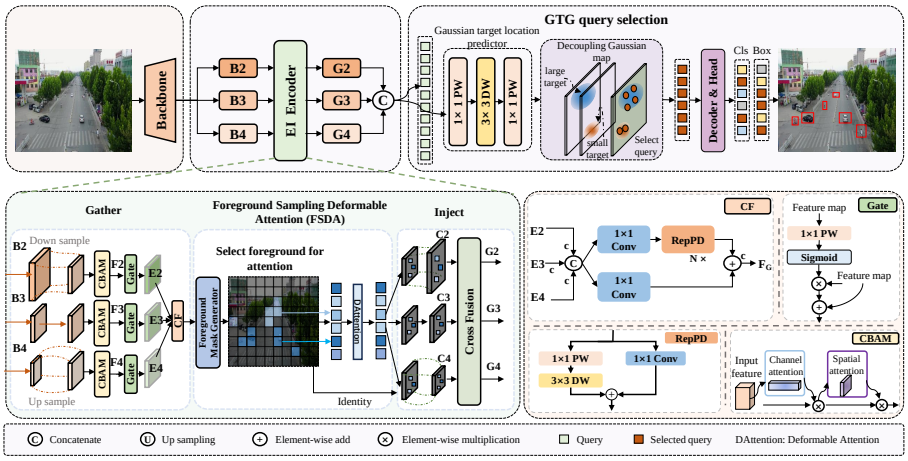

Fig. 2: Overview of the proposed Slim-DETR.

reliance on multi-layer detection losses hinders generalization to other DETR-based models [10]. Alternatively, RT-DETR [43] and its variations [7, 18] remove low-level feature interactions to achieve real-time speeds; however, this design severely hampers tiny object detection, as such objects critically depend on the local details present in low-level features. Furthermore, its FPN-like cross-fusion mechanism still faces the bottleneck of indirect information exchange. Beyond encoder architecture, query selection [39] is pivotal for learning effective representations of tiny objects. Although DQ-DETR [8] employs density maps to dynamically adjust query counts, the fundamental query-object mismatch remains unresolved, making it significantly harder for tiny objects to match with high-quality queries compared to larger ones.

3 Method

3.1 Model Overview

Fig. 2 illustrates the Slim-DETR architecture. Backbone features $\{\mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4\}$ enter the Efficient Interaction (EI) encoder, which employs a gather-and-inject mechanism to unify global semantics and refine representations via attention on foreground tokens only. The enhanced features are then directly injected into a lightweight cross-level fusion module for efficient inter-level exchange. Subsequently, the Gaussian Target Guided (GTG) module selects a fixed number of high-quality queries guided by a predicted Gaussian distribution, initializing the decoder for final detection.

3.2 Gather Multi-level Features in EI Encoder

Given that high-level features are rich in semantics while low-level features retain essential local details for tiny object detection, we aggregate multi-level features

for semantic interaction. To balance efficiency and detail preservation, all feature maps are projected to a unified resolution aligned with $\mathbf{B}_3$. However, low-level features suffer from limited receptive fields, capturing only local regions [33], and often lack high-level context. To mitigate this, we apply CBAM [34] to enhance their semantic representation, enabling low-level features to convey initial conceptual information about tiny objects, thereby reducing the semantic gap between different levels prior to aggregation. Regarding the mechanism of the CBAM module, Woo et al. [34] demonstrated that its channel and spatial branches employ pooling operations to aggregate global context. This enables the model to focus on meaningful "what" (channel-wise) and "where" (spatial-wise) cues, which is instrumental in enhancing the semantic representations of tiny objects, particularly within low-level feature maps. Motivated by these insights, we integrate the CBAM module into features $\mathbf{B}_2$, $\mathbf{B}_3$, and $\mathbf{B}_4$ to endow them with initial semantic cues regarding tiny objects.

Following the spatial resolution alignment of multi-level features, we introduce a gate unit that transforms the semantic knowledge embedded in the CBAM-processed features into a gate signal $\mathbf{G}$. Specifically, this mechanism assigns lower activation values to regions characterized by rich high-frequency details but ambiguous semantics (e.g., complex textured backgrounds). Consequently, working in conjunction with CBAM, the gate mechanism effectively suppresses background noise while selectively amplifying responses corresponding to target concepts. Finally, we apply the gate unit separately to the feature maps $\mathbf{F}_2$, $\mathbf{F}_3$ and $\mathbf{F}_4$ to enhance their prominent semantic representations.

Subsequently, we leverage the CNN-based Fusion (CF) module to aggregate multi-scale features and generate enhanced comprehensive semantic representations. The CF module comprises two parallel branches, as illustrated in the right panel of Fig. 2. The first branch incorporates RepPD blocks, which follow the architecture of RepBlock [43], and replaces the standard 3×3 convolution to a more lightweight 3×3 depthwise convolution and a pointwise convolution. These components work together to smooth the boundaries of targets across different scales and reduce feature conflicts [11,17]. The second branch leverages a parallel 1×1 convolution to reconstruct inter-channel relationships. This design prepares the global feature map $\mathbf{F}_G$ for Transformer-based interaction.

3.3 Foreground Sampling Deformable Attention (FSDA) module

To circumvent the computational bottleneck caused by long sequences, we introduce an attention-based Foreground Semantic Deformable Attention (FSDA) module. Unlike Salience-DETR and Sparse-DETR, which perform layer-wise hierarchical query filtering, our approach samples only foreground tokens on the aggregated global features. This avoids the cumulative loss of tiny object information across layers. Specifically, we generate a foreground mask $\mathbf{M}$ to distinguish foreground features from the global feature map $\mathbf{F}_G$. This process is formulated as:

$$\mathcal{S}_k = \left\{ m \mid \mathbf{M_s}(m) \in \text{top-}k(sort(\mathbf{M_s})) \right\}, \tag{1}$$

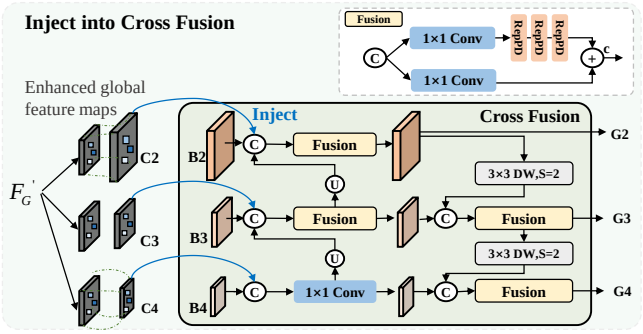

Fig. 3: Structure of the cross fusion in EI encoder with injecting semantic-enhanced global feature map $\mathbf{F}'_G$ into the FPN-like cross fusion module. Fusion acts as the lightweight fusion block followed [43]. The structure of RepPD is shown in Fig. 2

where $\mathbf{M_s} \in [0,1]^{H\times W}$ denotes the probability of pixel m belonging to informative foreground regions. Here, H and W represent the height and width of $\mathbf{F}_G$, respectively, and $m \in [0, H \times W]$. The function top-$k(\cdot)$ selects the indices of the top-k values, with all scores sorted in descending order. $\mathcal{S}_k$ denotes the set of selected foreground query indices, satisfying $num(\mathcal{S}_k) = \rho HW$. The parameter ρ is a sampling ratio correlated with the proportion of target pixels. Subsequently, $\mathbf{F}_G$ is flattened into a matrix $\mathbf{F}_G \in \mathbb{R}^{HW\times C}$, and the features $\mathbf{f}_s = \mathbf{F}_G[\mathcal{S}_k]$ are gathered as focused foreground queries, which are then updated via deformable attention:

$$\mathbf{F}'_G = \begin{cases} \text{DAttention}\big(\mathbf{f}_s + \mathbf{pos}_m, \mathbf{F}_G + \mathbf{pos}, \mathbf{F}_G\big) & m \in \mathcal{S}_k \\ \mathbf{F}_{Gm} & m \notin \mathcal{S}_k \end{cases}, \tag{2}$$

where $\mathbf{pos}_m$ is the positional encoding vector at position m, and $\mathbf{F}_{Gm}$ is the token of $\mathbf{F}_G$ at position m. Our foreground-aware sparse queries replace the dense input queries used in standard deformable attention [45], thereby providing more target-relevant enhanced features for prediction. However, simply removing non-focused background tokens leads to spatial discontinuity and disrupts the structured feature organization [6]. To address this, we propose embedding the updated foreground features back into their original positions within the background map to restore the spatial structure. Consequently, the resulting semantic-enhanced global feature map is denoted as $\mathbf{F}'_G$.

3.4 Inject Semantic-enhanced Features into Cross Fusion

Due to the limitations of indirect interaction, traditional FPN-like fusion approaches often suffer from inefficiency in cross-level information exchange, such as between level 2 and level 4. To overcome this bottleneck, we inject the semantic-enhanced global feature map $\mathbf{F}'_G$ into each level of the cross fusion module, as shown in Fig. 3.

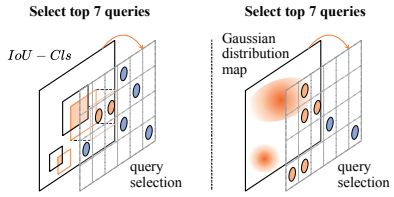

Fig. 4: Qualitative comparison of query selection strategies. The conventional method (left) relies on IoU and classification scores, whereas GTG query selection (right) utilizes a Gaussian distribution map for guidance. Blue dots represent background queries, and orange dots denote effective foreground queries.

Specifically, $\mathbf{F}'_G$ is spatially aligned to match each feature level's resolution and concatenated with level-wise features in our lightweight cross-fusion block. This design injects global semantic context directly into every hierarchical stage, establishing a shortcut for global-local interaction and eliminating the inefficient, indirect propagation typical of traditional architectures.

3.5 Gaussian Target Guided (GTG) query selection

Query selection [39] plays a pivotal role in DETR optimization. Conventional mechanisms relying on IoU and classification scores [8,43] often face issues where tiny object predictions have near-zero prior overlap and where scores are biased toward textural features. These factors collectively bias the selection towards large objects and background regions, thereby marginalizing tiny objects.

As illustrated in Fig. 4, departing from these conventional criteria, we introduce the Gaussian Target Guided (GTG) query selection strategy, characterized by three key designs: (1) the peak probability at object centers is normalized to 1, ensuring independence from target scale; (2) a spatially adaptive Gaussian distribution establishes a continuous confidence decay from the target center to the boundary, while enforcing strict background suppression (zero response in non-target regions); (3) a size-decoupled Gaussian formulation ensures balanced query guidance for both large and tiny objects, mitigating scale bias.

Specifically, for feature maps at levels $l \in \{2, 3, 4\}$, each query $q_l = (i, j)$ corresponds to coordinate p^l. Gaussian distribution map $Y_l^{(i,j)}$ for the ground-truth bounding boxes $\mathcal{D}_{\text{Bbox}} = (x_l, y_l, w_l, h_l)$ is then constructed as:

$$Y_l^{(i,j)} = \begin{cases} G(x_l, y_l, \sigma_l) & q_l \in \mathcal{D}_{\text{Bbox}} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $Y_l^{(i,j)} \in [0, 1]$ and the Gaussian kernel is:

$$G(x_l, y_l, \sigma_l) = \exp\left(-\frac{(p_i^l - x_l)^2 + (p_j^l - y_l)^2}{2\sigma_l^2}\right), \tag{4}$$

where $\sigma_l = r_l(w_l, h_l)$ with $r_l(w_l, h_l)$ denotes the adaptive radius computed from bounding box dimensions [3]. In cases where Gaussian kernels of the same class overlap, we compute their element-wise maximum.

We use a lightweight 3-layer CNN branch to predict the Gaussian map $\hat{Y}_l$ with output channels equal to the number of classes (in Fig. 2). This explicit supervision directly guides query allocation, reducing interference from large objects and background. For scale variance, we adopt decoupled prediction: (1) Targets with an absolute scale of 32×32 or smaller are predicted via $\mathbf{G}_2$; (2) Targets ranging from 32×32 to 64×64 are predicted via $\mathbf{G}_3$; (3) Larger targets are predicted by $\mathbf{G}_4$. This approach effectively prevents mutual interference between objects of disparate sizes. Crucially, the number of queries allocated to each scale is proportional to its predicted Gaussian response.

It is worth noting that the specific thresholds of 32 and 64 are empirically derived but can be adaptively scaled with input resolution or calibrated to target domain statistics, ensuring optimal feature-target alignment across varying scenarios.

3.6 Loss Optimization

To supervise the prediction of foreground mask, we introduce an auxiliary loss function defined as:

$$\mathcal{L}_m(\hat{M}, M) = \mathrm{BCE}(\hat{M}, M) \tag{5}$$

where $\mathcal{L}_m$ denotes the auxiliary supervision loss. BCE represents the Binary Cross-Entropy loss, and $\hat{M} \in [0, 1]$ is the predicted foreground mask. Note that the ground-truth mask M is generated from ground-truth bounding boxes, where pixels inside the boxes are set to 1 and those outside are set to 0. Consequently, optimizing this supervision loss during training ensures that foreground tokens are prioritized for attention interaction. In contrast to [6, 22], which relies on layer-wise detection losses, our method eliminates this dependency, simplifying the training process.

For the Gaussian distribution map optimization, the training objective combines focal loss [12]:

$$\mathcal{L}_G(\hat{Y}, Y) = \sum_{l=2}^{4} (\mathcal{L}_{focal}(\hat{Y}_l, Y_l) + L_{reg}(\hat{Y}_l, Y_l)), \tag{6}$$

where $\hat{Y}$ and Y are Gaussian distribution prediction and ground truth. We set $\alpha = 2$ and $\beta = 4$ in $\mathcal{L}_{focal}$ following [12] and MSE [21] loss compensating for regression shift in L_{reg}.

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

$$\mathcal{L}_{total} = \lambda_m \mathcal{L}_m + \lambda_G \mathcal{L}_G + \lambda_{box} \mathcal{L}_{box} + \lambda_{cls} \mathcal{L}_{cls}, \tag{7}$$

Table 1: Experiments on VisDrone val set. "-" indicates that the result is not reported. "R18" and "R50" are ResNet-18 and ResNet-50 backbone. Input size is 640×640.

Method	AP	AP_{50}	AP_{75}	Params(M)	GFLOPs	Latency(ms)
CNN-based						
QueryDet [38]	19.6	35.7	19.0	-	-	285.7
YOLOv11-L [9]	26.2	42.9	26.9	25.3	87	15.6
YOLOv12-L [24]	26.4	43.3	27.0	26.3	89	16.4
YOLOv10-L [28]	26.5	43.5	27.0	25.7	126	22.2
YOLOv8-L [27]	26.6	43.5	27.4	43.6	165	23.8
CenterNet [3]	27.8	47.9	27.6	-	1855	100
FBRT-YOLO-X [35]	30.1	48.4	-	22.8	186	19.2
Transformer-based						
RT-DETR-R18 [43]	26.7	44.6	25.9	20.0	60	13.8
Deformable DETR [45]	27.1	42.2	26.1	40.0	173	34.5
Sparse DETR [22]	27.3	42.5	26.4	41.0	121	28.6
DINO-DETR [40]	29.4	46.2	29.1	47.6	279	47.6
Salience-DETR [6]	29.4	46.2	29.1	-	201	82.2
RT-DETR-R50 [43]	30.3	50.8	30.3	42.0	136	19.6
DEIM-R50 [7]	30.4	50.9	30.5	42.0	136	19.1
D-FINE-R50 [18]	32.0	52.1	33.2	40.7	117	18.0
Slim-DETR-R18 (Ours)	32.4	52.5	33.5	**5.9**	**51**	**14.3**
Slim-DETR-R50 (Ours)	**33.9**	**55.2**	**36.9**	15.7	131	24.4

where $\mathcal{L}_{box}$ and $\mathcal{L}_{cls}$ are the localization and classification loss functions [43]. $\lambda_m = 0.01$ and $\lambda_G = 0.5$ are the weighting coefficients used to balance the loss, and $\lambda_{box} = 1$, $\lambda_{cls} = 1$, which are followed the setting of [43].

4 Experiments

4.1 Implementation Details

To evaluate our method, we conduct experiments on the tiny object detection benchmarks, VisDrone [2] and AI-TOD-V2 [30]. All experiments are conducted on 8 NVIDIA GeForce RTX 3090 GPUs, while latency measurements are conducted on an NVIDIA RTX 3090 GPU (in Table. 1 and Table. 2) and a Jetson AGX Orin device (in Table. 5). The training epochs are kept consistent with that of RT-DETR [43], and the models are optimized using the AdamW optimizer with an initial learning rate of 0.0006. We report the COCO [13] evaluation metrics for VisDrone and the AI-TOD-V2 evaluation metrics are followed DQ-DETR [8].

Table 2: Experiments on AI-TOD-V2. All models are evaluated on the test split with 800 × 800 input resolution.

Method	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m	Latency(ms)
CNN-based								
DetectoRS [19]	16.1	35.5	12.5	0.1	12.6	28.3	40.0	-
NWD-RKA [36]	24.7	57.4	17.1	9.7	24.2	29.8	39.3	74.1
DetectoRS w/ RFLA [37]	24.8	55.2	18.5	9.3	24.8	30.3	38.2	-
DNTR [14]	26.2	56.7	20.2	12.8	26.4	31.0	37.0	84.0
Transformer-based								
Deformable-DETR [45]	18.9	50.0	10.5	6.5	17.6	25.3	34.4	66.7
DAB-DETR [15]	22.4	55.6	14.3	9.0	21.7	28.3	38.7	-
DINO-DETR [40]	23.2	56.6	15.4	9.9	23.1	29.3	37.6	92.1
Salience-DETR [6]	23.4	56.4	15.7	9.3	23.0	29.3	36.8	158.9
RT-DETR-R50 [43]	27.6	59.7	17.5	12.7	25.3	32.0	39.7	38.4
DQ-DETR [8]	30.2	68.6	22.3	15.3	30.5	36.5	44.6	-
Slim-DETR-R50 (Ours)	**33.6**	**69.7**	**27.9**	**16.4**	**34.0**	**38.5**	**47.6**	39.4

4.2 Comparison Results on VisDrone

As shown in Table 1, Slim-DETR establishes a new state-of-the-art accuracy-efficiency trade-off. Crucially, under the same ResNet-18 backbone, our Slim-DETR-R18 surpasses RT-DETR-R18 by 5.7% AP with 3.4× fewer parameters at comparable latency, confirming that our gains stem from architectural efficiency rather than model scaling. Importantly, sharing the same RT-DETR baseline as concurrent methods (DEIM, D-FINE), our approach diverges by optimizing the architecture specifically for tiny objects. The lightweight Slim-DETR-R18 outperforms the R50-based DEIM and D-FINE by 2.0% and 0.4% AP respectively. Additionally, Slim-DETR-R50 achieves a remarkable 33.9% AP on VisDrone. These results confirm that our architectural efficiency yields superior gains for tiny object detection.

4.3 Comparison Results on AI-TOD-V2

To further demonstrate the superiority of our method in tiny object detection, we evaluate Slim-DETR-R50 on AI-TOD-V2, a challenging benchmark characterized by a high prevalence of extremely small objects. As shown in Table. 2, Slim-DETR-R50 establishes a new state-of-the-art, achieving 33.6% AP. This surpasses the previous best, DQ-DETR, by 3.4% (including +1.1% AP_{vt} and +3.5% AP_t), while outperforming the real-time RT-DETR by 6.0%. Crucially, we achieve this accuracy with competitive latency (39.4 ms), significantly outpacing other Transformers like Salience-DETR (158.9 ms) and DINO-DETR (92.1 ms).

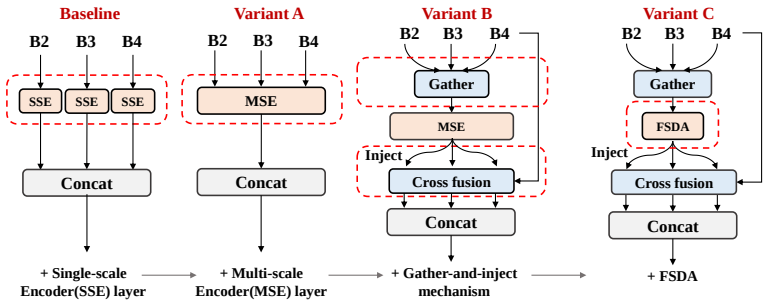

Fig. 5: The encoder structure for each variant. SSE and MSE are indicated as the single-scale Transformer encoder layer and the multi-scale Transformer encoder layer. The red box highlights the changes.

Table 3: Ablation study on the encoder with speed analysis on VisDrone. The configuration $\{\mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4\}$ utilizes higher-resolution backbone features as input.

Method	Latency(ms)	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
$\{\mathbf{B}_3, \mathbf{B}_4, \mathbf{B}_5\}$							
Baseline	28.57	29.0	49.4	28.9	19.6	40.6	53.7
Variant A	33.33	29.1	49.5	28.9	19.6	40.5	54.0
Variant B	20.83	30.0	50.6	29.9	20.7	41.6	55.5
Variant C	**20.40**	**30.3**	**50.7**	**30.0**	**20.7**	**41.7**	**56.2**
$\{\mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4\}$							
Baseline	100	30.7	50.4	30.5	21.7	41.1	53.7
Variant A	166.66	31.3	51.9	31.7	22.8	41.9	56.0
Variant B	28.57	33.1	54.0	33.7	24.6	43.9	54.4
Variant C	**24.39**	**33.3**	**54.3**	**34.2**	**24.9**	**44.0**	**55.9**

These results confirm that Slim-DETR effectively optimizes semantic interactions and query selection for minute targets, delivering robust performance in scenes with densely clustered tiny objects without compromising speed.

4.4 Ablation Study

Speed analysis of the encoder components We evaluate four encoder configurations (Fig. 5) across standard and high-resolution inputs (Table 3). While adding multi-scale layers (Variant A) yields 0.9% AP_s gain, it severely degrades high-resolution latency (100.00 ms to 166.66 ms), exposing the bottleneck of increased sequence lengths. Our gather-and-inject mechanism (Variant B) resolves this by eliminating redundant cross-scale interactions, achieving a 3.5× speedup and a +2.4% AP gain over the high-resolution Baseline. Further integrating

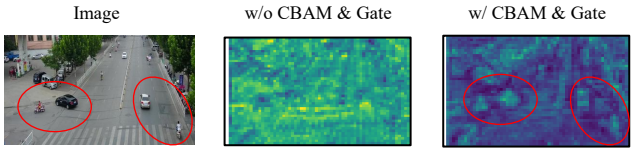

Fig. 6: Feature map visualization for gathered global feature map. CBAM and gate help to suppress the noise and emphasize the target concepts responses.

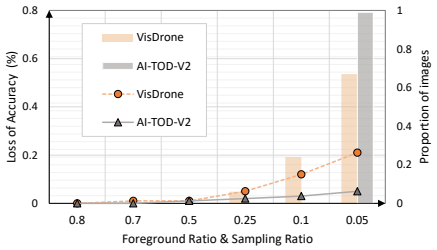

Fig. 7: Loss of accuracy with different sampling ratios and foreground proportions across VisDrone and AI-TOD-V2.

FSDA (Variant C) refines attention to foreground tokens only, minimizing background noise. The final configuration delivers the optimal trade-off: 33.3% AP at 24.39 ms (4.1× overall acceleration), confirming our EI encoder enables real-time tiny object detection despite high-resolution costs.

Effect of CBAM and Gate in EI encoder Fig. 6 visualizes the gathered global feature maps to highlight the role of CBAM and gate modules. Without them, the feature map is cluttered with high-response background noise, obscuring target details. However, the integration of the CBAM and gate acts as an effective enhancement: it suppresses noisy background activations and sharply highlights tiny object regions. This enhances the semantic discriminability between foreground targets and complex backgrounds.

Effect of sampling ratio in FSDA Fig. 7 analyzes the sampling ratio ρ (0.05 to 0.8) on VisDrone and AI-TOD-V2. The results demonstrate that optimal performance is achieved when ρ matches or slightly exceeds the dataset's inherent foreground proportion. Notably, detection accuracy exhibits strong robustness for $\rho > 0.25$ on VisDrone and $\rho > 0.1$ on AI-TOD-V2, where further increases incur unnecessary computational costs. Consequently, we adopt $\rho = 0.25$ and $\rho = 0.1$ for VisDrone and AI-TOD-V2, respectively. However, for extreme scene variations (e.g., transitioning from sparse vehicles to dense crowds), we recommend that ρ can be instantly increased to prevent foreground truncation. Given the negligible overhead of index recalculation, this allows seamless adaptation to drastic density shifts without retraining.

Table 4: Ablation Experiment of the GTG in decoder optimization on the VisDrone val set. N-layer represents the number of decoder layers in Slim-DETR-R50. Uncertainty-minimal [43] is based on IoU and classification confidence.

Method	N-layer	AP	AP_{50}	AP_s	Params(M)	GFLOPs	Latency(ms)
Uncertainty-minimal	6	33.3	54.3	24.7	16.3	128	24.39
GTG	6	33.5	54.5	24.9	17.2	137	27.03
	5	33.7	54.7	25.0	16.4	134	26.31
	4	**33.9**	**55.2**	**25.3**	15.7	131	24.41
	3	33.2	53.9	24.2	14.9	128	23.25

Effect of GTG Query Selection Method To validate the effectiveness of GTG in decoder optimization, Table 4 compares the inference speed and accuracy of Slim-DETR across varying decoder depths. Compared to the conventional Uncertainty-minimal method [43] with 6 layers, GTG achieves a 0.2% AP gain. More importantly, GTG enables a shallower architecture without performance loss: reducing the decoder depth from 6 to 4 layers further boosts accuracy to a peak of 33.9 AP, while maintaining latency identical to the 6-layer baseline (24.41 ms vs. 24.39 ms). Specifically, expanding from 3 to 4 layers yields a substantial 0.7 AP improvement, whereas deeper stacks ($N > 4$) offer diminishing returns. Consequently, we adopt a 4-layer decoder as the optimal configuration. Regarding the computational cost, while the GTG module introduces a lightweight CNN branch for Gaussian map prediction, incurring an initial overhead (+0.9 M parameters and +9 GFLOPs as seen in the 6-layer configuration), this cost is effectively neutralized and even surpassed by the reduction in decoder depth. These results confirm that GTG effectively aggregates high-quality initial queries around targets, reducing the reliance on deep iterative refinement.

4.5 Real-time inference on Jetson AGX Orin Edge Platform

The NVIDIA Jetson AGX Orin, characterized by its low power envelope (15–60W) and compact form factor, is an ideal platform for onboard embedded computing in drone applications [26]. Following the experimental protocol established in [26], we configured the device environment to ensure a fair and reproducible evaluation under resource-constrained conditions. Comparative results on this edge platform are detailed in Table 5. Compared to state-of-the-art detectors, Slim-DETR-416 achieves a superior balance of speed and accuracy. It outperforms YOLOv12-M and RT-DETR-416 by 0.6% and 1.9% in AP, respectively. More critically, Slim-DETR-416 attains a real-time inference speed of 40 FPS (25.12 ms latency), significantly surpassing YOLOv12-M (11 FPS) and RT-DETR-416 (27 FPS). These results demonstrate that Slim-DETR not only enhances small object detection accuracy but also delivers the high throughput required for real-time autonomous drone operations on edge hardware.

Table 5: Experiments on VisDrone val set. Slim-DETR-416 is Slim-DETR-R18 with 416×416 input size.

Method	Metric/Format	Latency(ms)	FPS	AP
YOLOv3-416 [20]	TensorRT FP16	28.57	35	23.1
YOLOv8-M [27]	TensorRT FP16	68.68	15	24.6
YOLOv10-M [28]	TensorRT FP16	68.17	15	24.5
YOLOv12-M [24]	TensorRT FP16	90.12	11	24.8
RT-DETR-416 [43]	TensorRT FP16	35.98	27	23.5
Slim-DETR-416	TensorRT FP16	**25.12**	**40**	**25.4**

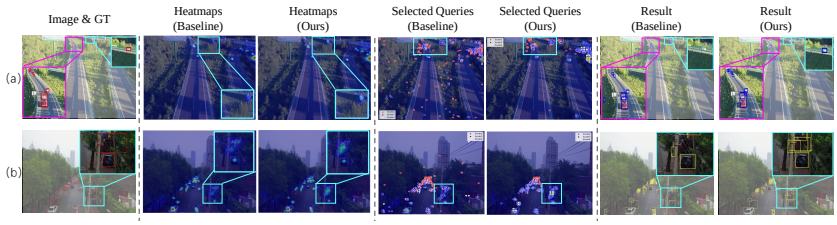

Fig. 8: Visualization of feature heatmaps, selected queries, and results on the VisDrone. In the query selection views, selected queries are represented by points.

4.6 Visualization of Detection Results

Fig. 8 qualitatively validates Slim-DETR's superiority in tiny object detection. Unlike the scattered background noise in RT-DETR, our EI encoder effectively suppresses distractions while preserving semantic integrity, enabling sharp localization of tiny (Fig. 8(a)) and occluded targets (Fig. 8(b)) with high confidence. Furthermore, GTG query selection strategically concentrates effective queries on tiny objects, significantly minimizing background redundancy.

5 Conclusion

In this paper, we propose Slim-DETR, a novel real-time end-to-end Transformer detector featuring an Efficient Interaction (EI) encoder and a Gaussian Target Guided (GTG) query selection strategy. The EI encoder captures and propagates global semantic information via an efficient gather-and-inject mechanism, performing semantic interaction exclusively on potential foreground tokens, thereby overcoming the efficiency bottlenecks of conventional architectures. Moreover, GTG leverages the Gaussian spatial distribution of targets to concentrate high-quality queries around tiny objects. Slim-DETR establishes a new benchmark for real-time DETR-based models in tiny object detection, achieving 33.9% mAP on VisDrone and 33.6% on AI-TOD-V2, sustaining real-time speeds on both the NVIDIA RTX 3090 GPU and embedded Jetson AGX Orin platforms.

Acknowledgements

This work was supported in part by the Key Project of National Natural Science Foundation of China (No.62534006), in part by the Young Scientists Fund of the National Natural Science Foundation of China(No.62304162), in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China(No.JYB2025XDXM121), and in part by Shaanxi Provincial Natural Science Foundation for Basic Research Program (2024JC-YBMS-794).

References

1. Cheng, G., Yuan, X., Han, X.J.: Towards large-scale small object detection: Survey and benchmarks. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(11), 13467–13488 (2023)
2. Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., et al.: Visdrone-det2019: The vision meets drone object detection in image challenge results. In: Proceedings of the IEEE/CVF international conference on computer vision workshops. pp. 0–0 (2019)
3. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6568–6577 (2019)
4. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7029–7038 (2019)
5. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Object instance segmentation and fine-grained localization using hypercolumns. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4), 627–639 (2017)
6. Hou, X., Liu, M., Zhang, S., Wei, P., Chen, B.: Salience detr: Enhancing detection transformer with hierarchical salience filtering refinement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17574–17583 (2024)
7. Huang, S., Lu, Z., Cun, X., Yu, Y., Zhou, X., Shen, X.: Deim: Detr with improved matching for fast convergence. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15162–15171 (2025)
8. Huang, Y.X., Liu, H.I., Shuai, H.H., Cheng, W.H.: Dq-detr: Detr with dynamic query for tiny object detection. In: European Conference on Computer Vision. pp. 290–305. Springer (2024)
9. Khanam, R., Hussain, M.: Yolov11: An overview of the key architectural enhancements. arXiv preprint arXiv:2410.17725 (2024)
10. Li, F., Zeng, A., Liu, S., Zhang, H., Li, H., Zhang, L., Ni, L.M.: Lite detr: An interleaved multi-scale encoder for efficient detr. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18558–18567 (2023)
11. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 936–944 (2017)
12. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)

13. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13. pp. 740–755. Springer (2014)
14. Liu, H.I., Tseng, Y.W., Chang, K.C., Wang, P.J., Shuai, H.H., Cheng, W.H.: A denoising fpn with transformer r-cnn for tiny object detection. IEEE transactions on geoscience and remote sensing **62**, 1–15 (2024)
15. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: DAB-DETR: Dynamic anchor boxes are better queries for DETR. In: International Conference on Learning Representations (2022)
16. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8759–8768 (2018)
17. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill **1**(10) (2016)
18. Peng, Y., Li, H., Wu, P., Zhang, Y., Sun, X., Wu, F.: D-fine: Redefine regression task in detrs as fine-grained distribution refinement. arXiv preprint arXiv:2410.13842 (2024)
19. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10213–10224 (2021)
20. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
21. Ren, J., Zhang, M., Yu, C., Liu, Z.: Balanced mse for imbalanced visual regression. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7916–7925 (2022)
22. Roh, B., Shin, J., Shin, W., Kim, S.: Sparse detr: Efficient end-to-end object detection with learnable sparsity. arXiv preprint arXiv:2111.14330 (2021)
23. Shi, T., Gong, J., Jiang, S., Zhi, X., Bao, G., Sun, Y., Zhang, W.: Complex optical remote-sensing aircraft detection dataset and benchmark. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–9 (2023)
24. Tian, Y., Ye, Q., Doermann, D.: Yolov12: Attention-centric real-time object detectors. arXiv preprint arXiv:2502.12524 (2025)
25. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: A simple and strong anchor-free object detector. IEEE transactions on pattern analysis and machine intelligence **44**(4), 1922–1933 (2020)
26. Ulmămei, A.A., D'Adamo, T., Vasile, C.E., Hobincu, R.: Human detection in uav thermal imagery: Dataset extension and comparative evaluation on embedded platforms. Journal of Imaging **11**(12), 436 (2025)
27. Varghese, R., M., S.: Yolov8: A novel object detection algorithm with enhanced performance and robustness. In: 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). pp. 1–6 (2024)
28. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., et al.: Yolov10: Real-time end-to-end object detection. Advances in Neural Information Processing Systems **37**, 107984–108011 (2024)
29. Wang, J., Xu, C., Yang, W., Yu, L.: A normalized gaussian wasserstein distance for tiny object detection. arXiv preprint arXiv:2110.13389 (2021)
30. Wang, J., Yang, W., Guo, H., Zhang, R., Xia, G.S.: Tiny object detection in aerial images. In: 2020 25th international conference on pattern recognition (ICPR). pp. 3791–3798. IEEE (2021)

31. Wang, J., Yao, Z., Chen, L., Yang, R., Zhang, X.: Fcdnet: A multiscale attention network for forest change detection using dual-temporal very-high-resolution remote sensing images. IEEE Transactions on Geoscience and Remote Sensing **63**, 1–15 (2025)
32. Wang, Y., Yan, X., Bao, H., Chen, Y., Gong, L., Wei, M., Li, J.: Detecting occluded and dense trees in urban terrestrial views with a high-quality tree detection dataset. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–12 (2022)
33. Woo, S., Hwang, S., Kweon, I.S.: Stairnet: Top-down semantic aggregation for accurate one shot detection. In: 2018 IEEE winter conference on applications of computer vision (WACV). pp. 1093–1102. IEEE (2018)
34. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
35. Xiao, Y., Xu, T., Xin, Y., Li, J.: Fbrt-yolo: Faster and better for real-time aerial image detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 8673–8681 (2025)
36. Xu, C., Wang, J., Yang, W., Yu, H., Yu, L., Xia, G.S.: Detecting tiny objects in aerial images: A normalized wasserstein distance and a new benchmark. ISPRS Journal of Photogrammetry and Remote Sensing **190**, 79–93 (2022)
37. Xu, C., Wang, J., Yang, W., Yu, H., Yu, L., Xia, G.S.: Rfla: Gaussian receptive field based label assignment for tiny object detection. In: European conference on computer vision. pp. 526–543. Springer (2022)
38. Yang, C., Huang, Z., Wang, N.: Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13658–13667 (2022)
39. Yao, Z., Ai, J., Li, B., Zhang, C.: Efficient detr: improving end-to-end object detector with dense prior. arXiv preprint arXiv:2104.01318 (2021)
40. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
41. Zhao, H., Chen, J., Wang, L., Lu, H.: Arkittrack: a new diverse dataset for tracking using mobile rgb-d data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5126–5135 (2023)
42. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
43. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J.: Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16965–16974 (2024)
44. Zheng, X., Wang, B., Du, X., Lu, X.: Mutual attention inception network for remote sensing visual question answering. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–14 (2021)
45. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)