Reviving the Discarded High-Resolution Feature for Transformer-Based Tiny Object Detection

Zehao Wang[1,2] **Si Wu**[1,2] **Guiping Cao**[2*] **Xin Li**[2*] **Yaowei Wang**[3,2]
[1]South China University of Technology, Guangzhou, China
[2]Pengcheng Laboratory, Shenzhen, China
[3]Harbin Institute of Technology, Shenzhen, China
wangzh07@pcl.ac.cn, cswusi@scut.edu.cn, caogp@pcl.ac.cn,
xinlihitsz@gmail.com, wangyaowei@hit.edu.cn

Abstract

Tiny object detection remains challenging because tiny instances occupy only a few pixels, and their fine-grained details are easily lost during feature downsampling. Although the highest-resolution feature preserves rich local details for tiny objects, DETR-like detectors often discard it due to its large computational cost, while directly reintroducing this feature also brings dense background noise. To address this problem, we propose **ReDHF**, a novel framework that **Re**vives the **D**iscarded **H**igh-resolution **F**eature for tiny object detection. ReDHF has three cooperative components. Cross-Scale Deformable Fusion (CSDF) first constructs a semantically enhanced highest-resolution feature and injects its fine-grained details into the shallow feature under a moderate cost. Ellipse-Guided Shallow Supervision (EGSS) then guides the enhanced feature to learn more reliable classification and localization priors by applying geometry-adaptive auxiliary supervision. Full-Scale Query Interaction (FSQI) further enables decoder queries to access both low-level details and high-level semantic information, allowing them to aggregate richer cues for tiny objects. Extensive experiments show that ReDHF consistently improves multiple baselines across different DETR variants and achieves state-of-the-art performance on the evaluated benchmarks. The best ReDHF variants surpass their baselines by **+1.4 AP** on AI-TOD-v2, **+1.6 AP** on VisDrone, and **+1.5 AP** on SODA-D, with especially notable gains on tiny and small objects.

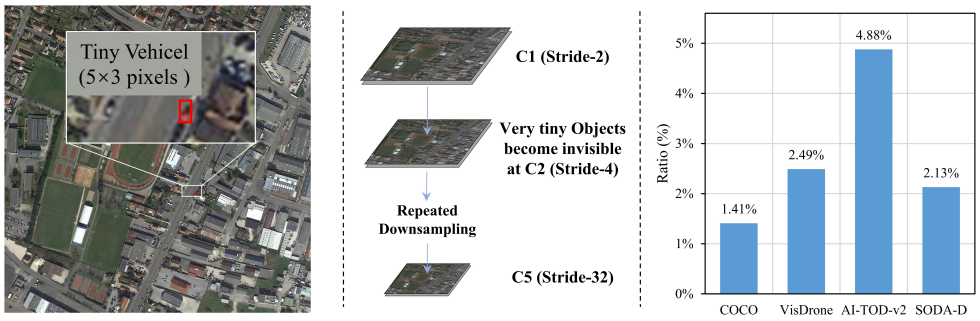
1 Introduction

Object detection has advanced rapidly, with both CNN-based detectors [7, 9, 12, 20, 27, 31, 34, 35, 41, 44] and Transformer-based detectors [4, 5, 38, 25, 29, 30, 52, 56, 59] achieving strong performance. However, detecting tiny objects remains difficult, especially in aerial, remote sensing, and traffic scenarios. Tiny objects occupy only a few pixels, leaving their edges, textures, and geometric cues weak in the image and further weakened by feature downsampling. As a result, high-level features often fail to preserve sufficient discriminative details for tiny objects, as illustrated in Fig. 1(b).

To address the issues of tiny object detection, a natural idea is to preserve these weak cues by using higher-resolution inputs or features. Prior works have shown that high-resolution information is important for tiny objects. For example, Deformable DETR [59] points out that small objects rely on high-resolution feature maps, and DM-EFS [37] shows that shallower high-resolution features can improve tiny-object representation. However, many detectors [26, 52, 59, 25] still discard the highest-resolution backbone feature to control computation. This design saves overhead but sacrifices the local details that are critical for tiny objects. As shown in Fig. 1(a)(c), discarding this feature

* Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Aerial Image with Tiny Object (b) Feature Downsampling Pipeline (c) Ultra-tiny (≤ 4 px) objects ratio across datasets

Figure 1: (a) Tiny objects may occupy only a few pixels, making their visual evidence extremely weak. (b) Very tiny objects can become nearly invisible on stride-4 feature map in the feature downsampling pipeline. (c) The ratio of ultra-tiny objects (short side $\leq$ 4 pixels) is much higher in tiny object datasets such as AI-TOD-v2, VisDrone, and SODA-D than in COCO.

becomes especially detrimental on AI-TOD-v2 [46], VisDrone [58], and SODA-D [6], where ultra-tiny objects with a short side no larger than 4 pixels appear frequently and may be compressed to nearly a single point even on a stride-4 feature map. Meanwhile, directly reintroducing the highest-resolution backbone feature is also impractical, *since it has a much larger spatial size that may lead to unacceptable extra cost, and its shallow responses contain dense background activations that may overwhelm weak tiny-object signals.* These issues highlight the pressing need for detection frameworks that can balance fine-grained information preservation with computational efficiency.

In this work, we propose **ReDHF**, a novel and efficient plug-and-play high-resolution detail reintroduction framework for Deformable-DETR-like detectors, which revives the discarded highest-resolution backbone feature while controlling computation and background disturbance through three cooperative components. First, to address the contradiction between the limited representation of small objects and the high cost of utilizing high-resolution features, we design a lightweight **Cross-Scale Deformable Fusion (CSDF)** module that semantically enhances the highest-resolution feature to provide more useful fine-grained cues with moderate cost. Second, considering that high-resolution features mainly include low-level edges and textures and are more likely to introduce background noise, we apply **Ellipse-Guided Shallow Supervision (EGSS)** to guide the shallow feature to learn more stable classification and localization priors. Moreover, since final predictions in DETR-like detectors are query-driven, we further propose **Full-Scale Query Interaction (FSQI)**, which integrates the semantically enhanced highest-resolution feature together with higher-level features into the encoder memory, so that decoder queries can directly access pixel-level details and improve their ability of detecting tiny objects. Overall, ReDHF makes more effective use of the highest-resolution backbone feature with only a modest extra cost and significantly improves tiny object detection.

Extensive experiments show that ReDHF consistently improves multiple DETR-like [52, 59, 14] baselines and achieves state-of-the-art performance. Specifically, the best ReDHF model achieves 35.7% AP on AI-TOD-v2 [46], improving AP_{vt} to 20.8% (+1.2%) over its corresponding baseline. In addition, the ReDHF variant built upon DINO-4scale [52] reduces GFLOPs by 86.7% compared with directly integrating the highest-resolution feature, as detailed in Appendix D. On VisDrone [58] and SODA-D [6], ReDHF again brings consistent gains. The best ReDHF model achieves 40.6% AP and 38.7% AP, and also improves AP_s to 33.1% (+1.6%) and AP_{eS} to 21.4% (+1.7%), respectively. These results support our claim that the highest-resolution feature, though often discarded for efficiency, can be effectively revived through semantic enhancement. In summary, our main contributions are:

- We revisit the commonly discarded highest-resolution backbone feature in DETR-like detectors and show that its effective reuse requires addressing not only computational cost, but also weak semantics and noisy shallow responses.
- We propose ReDHF, which revives the discarded feature through semantically enhanced highest-resolution feature construction (CSDF), training-time shallow auxiliary supervision (EGSS), and decoder full-scale query interaction (FSQI), thereby enabling efficient high-resolution detail reintroduction for tiny-object detection.

- We validate ReDHF on multiple DETR-like baselines across AI-TOD-v2, VisDrone, and SODA-D. It achieves state-of-the-art performance and consistently improves the baselines with average gains of +1.5 AP_{vt}, +1.7 AP_s, and +1.8 AP_{eS} on the three datasets, respectively.

2 Related work

2.1 Tiny object detection

Identifying tiny objects is inherently difficult because tiny instances provide limited visual evidence and are vulnerable to complex background noise. Early CNN-based detectors, such as Faster R-CNN [9, 35], often struggle with such objects due to inadequate feature representation and limited context modeling. Some works improve tiny-object learning by enriching training samples, such as copy-pasting augmentation [18] and instance-level transformations [60]. Recently, transformer-based models [4, 19, 25, 30, 38, 52, 56, 59] have shown strong potential by bypassing hand-crafted post-processing like NMS and leveraging self-attention mechanisms. Cross-DINO [3] combines deep MLP representations with transformer encoder features through a Cross Coding Twice Module to enhance small-object details. DQ-DETR [15] introduces density maps to adjust query numbers and positions based on instance density, while Dome-DETR [14] further improves tiny object detection by generating density-oriented features. However, these methods often rely heavily on specific density modeling or carefully designed query mechanisms, which may complicate the optimization process and limit their flexibility across different datasets.

2.2 Multi-scale and high-resolution features

Feature pyramids [20] are widely used to handle object scale variation. Advanced architectures like BiFPN [40] and DetectoRS [32], improve cross-level information flow through bidirectional fusion or recursive feedback. For tiny objects, fusion factors [10] further regulate adjacent pyramid levels to reduce the learning burden. Recent works also explore high-resolution information more explicitly. QueryDet [49] refines sparse locations from low to high resolutions. ESOD [24] uses patch slicing and sparse prediction to reduce redundant computation on high-resolution features. DM-EFS [37] expands feature sets to enrich small-object representations. These works show the value of high-resolution cues for tiny objects. However, many detection pipelines still discard the highest-resolution backbone feature to keep computation affordable, leaving the finest local details underused. In contrast, our method revives this discarded feature through semantic enhancement, making its fine-grained details more compatible and usable for tiny object detection.

2.3 Auxiliary supervision and label assignment

Auxiliary supervision has been widely used to strengthen intermediate representations in object detection. In CNN-based detectors, DSFPN [50], LAF-YOLOv10 [8] and Jin et al. [16] attach auxiliary prediction heads to early backbone stages or bottom-up pathways to improve feature learning. A similar idea is adopted in DETR-like models [52, 15, 14, 30, 38], where auxiliary losses on intermediate decoder layers are commonly used to stabilize training and facilitate optimization. Beyond intermediate decoder supervision, MLP-DINO [2] introduces query-independent category supervision to mitigate the sensitivity of category predictions to bounding box prediction.

Label assignment is another key factor affecting detection performance. ATSS [53] shows that positive and negative sample definitions are critical. For generic object detection, AutoAssign [57] and IQDet [28] select high-quality regions according to predicted quality scores, while FCOS [43] formulates detection as dense per-pixel prediction. For transformer-based detectors, DS-Det [11] introduces a unified Single-Query paradigm and disentangles cross-attention-based box localization with one-to-many assignment from self-attention-based deduplication with one-to-one assignment. However, tiny objects make assignment more difficult because small localization shifts can strongly affect matching quality. Recent methods therefore redesign assignment for tiny objects. UGS [39] shows that IoU-based assignment can be unstable for tiny objects and proposes a classification-based objective. DCNet [55] uses dynamic scale-aware assignment, while RFLA [47] matches Gaussian receptive fields with ground truth to alleviate positive-sample scarcity. Unlike these methods that mainly design assignment for final prediction, our work focuses on strengthening intermediate representations. Specifically, we propose a geometry-adaptive assignment strategy and auxiliary

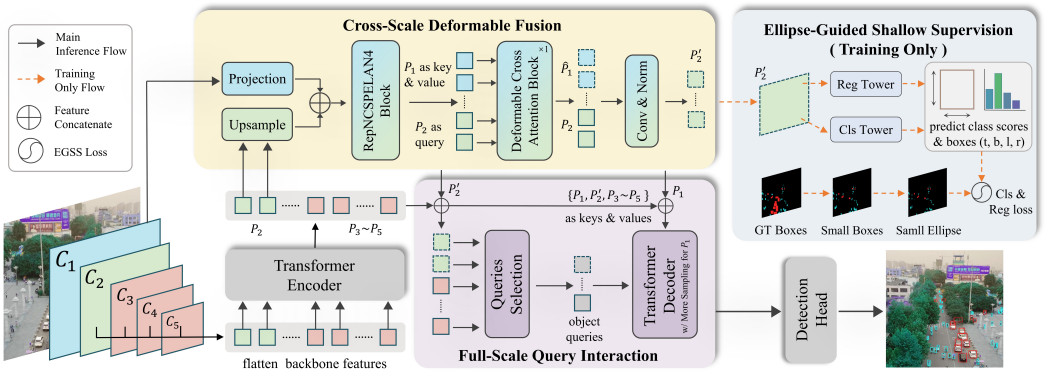


Figure 2: Overview of ReDHF. CSDF revives the discarded highest-resolution feature C_1 as a semantically enhanced feature P_1 and injects its fine-grained details into P_2 to obtain the detail-enhanced shallow feature P_2'. EGSS further supervises P_2' with ellipse-guided auxiliary supervision during training, while FSQI exposes P_1, P_2', and deeper features to decoder queries for final detection.

supervision to build coarse classification and localization priors for tiny objects, while the auxiliary head is removed during inference.

3 Method

3.1 Overall framework

Given an input image, let $\{C_i\}_{i=1}^{l}$ denote the multi-level backbone features, and let $\{P_i\}_{i=1}^{l}$ denote the corresponding multi-scale features after encoder context modeling, where C_1 is the highest-resolution feature. To revive the discarded highest-resolution feature without incurring excessive computation or amplifying background responses, we propose **ReDHF**, a plug-and-play framework for transformer-based tiny object detection. As shown in Fig. 2, ReDHF consists of three components: Cross-Scale Deformable Fusion (CSDF, Sec. 3.2), Ellipse-Guided Shallow Supervision (EGSS, Sec. 3.3), and Full-Scale Query Interaction (FSQI, Sec. 3.4). During training, CSDF and FSQI are optimized end-to-end in the main detection pipeline, while EGSS supervises the shallow feature to learn stable classification and localization priors. During inference, EGSS is completely removed, and only CSDF and FSQI remain for feature construction and decoder interaction.

3.2 Cross-scale deformable fusion

CSDF is designed to exploit the fine-grained details of the highest-resolution backbone feature while reducing the side effect of its noisy background responses. Instead of directly using raw C_1, CSDF first constructs a semantically enhanced highest-resolution feature P_1, and then injects its high-resolution details into the shallow feature P_2.

Semantically enhanced highest-resolution feature construction. Directly using C_1 is suboptimal because it contains rich details but weak semantics [20], making it sensitive to background noises. To address this issue, we inject the semantic information from P_2 into C_1 through a lightweight RepNCSPELAN4 block [30]. Since lower layers preserve local details while higher layers aggregate more global context [33], we only upsample P_2 and fuse it with the projected C_1 as follows:

$$P_1 = \mathcal{F}_{\text{RepNCSPELAN4}}\big(\text{Concat}(\text{Up}(P_2), \phi(C_1))\big), \tag{1}$$

where $\phi(\cdot)$ denotes channel projection and $\mathcal{F}_{\text{RepNCSPELAN4}}$ denotes the fusion block. In this way, P_1 preserves the high-resolution local details of C_1 while inheriting the semantic prior from P_2, alleviating the semantic gaps between different depth features.

High-resolution detail injection for tiny objects. After obtaining P_1, we further use it to compensate for the fine-grained details that are missing in the shallow feature P_2. However, either a

direct dense convolution or a bidirectional cross attention for the fusion of P_1 and P_2 will cause computational bottlenecks due to the large spatial size of P_1. We therefore adopt a deformable cross-attention design, where P_2 serves as query and P_1 serves as key and value:

$$\hat{P}_1 = \mathrm{DeformAttn}(Q = P_2, K = P_1, V = P_1). \tag{2}$$

Here, $\hat{P}_1$ denotes the retrieved detail feature, which is adaptively extracted from P_1 according to the spatial prior of P_2. Since $\hat{P}_1$ is essentially local detail responses from P_1, it cannot replace the original P_2. Instead, we fuse it back into P_2 to form the final detail-enhanced shallow feature P_2':

$$P_2' = \mathcal{F}_{\text{act}}\Big(\mathcal{F}_{\text{norm}}\big(\mathrm{Conv}_{1\times1}(\mathrm{Concat}(P_2, \hat{P}_1))\big)\Big), \tag{3}$$

where $\mathcal{F}_{\text{norm}}$ and $\mathcal{F}_{\text{act}}$ denote normalization and activation, respectively. The resulting P_2' retains the semantic content of P_2 and the local detail supplement from the highest-resolution P_1. Compared with the global or deformable self-attention interaction, the computational cost of CSDF is more acceptable, as analyzed in Appendix D, making the use of the highest-resolution feature more practical in tiny object detection.

3.3 Ellipse-guided shallow supervision

EGSS aims to guide P_2' to learn reliable foreground classification and coarse localization priors, rather than making final predictions on the intermediate representation. Although CSDF enriches P_2' with high-resolution details, it may also transfer dense background responses from P_1 to P_2. As shown in Fig. 4, both P_1 and P_2 can respond strongly to background textures and local patterns. When injecting detail cues from P_1 into P_2, these background responses may be transferred together with object-related details, thereby amplifying non-object responses in complex scenes. To address this side effect, we introduce EGSS, which applies shallow auxiliary supervision to P_2' during training and is completely removed during inference, introducing no additional inference cost.

Ellipse-guided label assignment. Standard FCOS-style dense supervision [43] usually adopts box-shaped assignment or fixed-radius center sampling, which is not ideal for tiny objects. Since tiny objects occupy only a few pixels, box boundaries often include background pixels and introduce noisy supervision. Center sampling can reduce boundary noise, but a fixed region cannot adapt well to different aspect ratios and may even remove all valid positives for extremely tiny objects.

At the same time, the detail-enhanced shallow feature P_2' has a limited receptive field and is more suitable for detecting small objects. For large objects, a feature point on P_2' often perceives only a small part of the instance and cannot understand the complete structure well. Assigning such objects to supervise P_2' may introduce ambiguous signals and distract the shallow feature from tiny-object regions. Motivated by the scale-aware design of FCOS [43], we restrict EGSS to size-matched objects and define positive samples with a scale filtering strategy and an ellipse-guided geometric constraint. Let $\mathcal{G}$ denote the set of ground-truth boxes, $\mathcal{P}$ denote the set of feature locations on P_2', and s_2 denote the stride of P_2'. The positive sample set $\mathcal{A}_{\text{pos}}$ is defined as:

$$\begin{aligned}\mathcal{A}_{pos} = \Big\{(p_i, q_j) \mid\ & p_i = (x_i, y_i) \in \mathcal{P},\ q_j = (c_x^{(j)}, c_y^{(j)}, w_j, h_j) \in \mathcal{G},\\ & r_{\min} \le \frac{\sqrt{w_j h_j}}{s_2} \le r_{\max},\ \frac{(x_i - c_x^{(j)})^2}{(w_j/2)^2} + \frac{(y_i - c_y^{(j)})^2}{(h_j/2)^2} \le 1\Big\}.\end{aligned} \tag{4}$$

Here, $(c_x^{(j)}, c_y^{(j)}, w_j, h_j)$ denote the center, width, and height of the j-th ground-truth box, and $[r_{\min}, r_{\max}]$ filters objects whose sizes do not match the receptive field of P_2'. Compared with rectangular assignment, the elliptical constraint naturally adapts better to object aspect ratios and suppresses boundary noise for tiny objects, while the scale filter focuses supervision on size-matched tiny objects, which helps P_2' pay more attention to learning to perceive tiny objects. Meanwhile, when a feature point matches multiple objects, we assign it to the smallest one to reduce ambiguity and bias the supervision toward smaller instances.

Shallow auxiliary supervision. Based on the above assignment, we attach a lightweight FCOS-style [43] auxiliary head to P_2' during training. The head contains two parallel towers for classification

and regression, respectively. Each tower consists of two 3×3 convolution layers, followed by a prediction layer that outputs class logits or box regression results. The auxiliary loss is:

$$\mathcal{L}_{aux} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{reg}\mathcal{L}_{reg}, \tag{5}$$

where λ_{cls} and λ_{reg} are the loss weights. We use focal loss for $\mathcal{L}_{cls}$ to handle the severe foreground-background imbalance and a combination of ℓ_1 loss and GIoU [36] loss for $\mathcal{L}_{reg}$ to provide more stable regression supervision.

Tiny-object regression is naturally sensitive to small localization shifts [39], especially on noisy shallow features. Since P_2' still contains non-negligible background responses, as shown in Fig. 4, a strong regression loss may force the auxiliary branch to fit unstable box targets and disturb foreground discrimination. We therefore assign a small regression weight. In addition, we do not adopt centerness supervision by default, since tiny objects have only a few valid locations on the feature map and their centerness targets can vary sharply and become unreliable on the noisy shallow feature. EGSS therefore focuses on foreground classification and coarse localization, while final box refinement is left to the decoder. We empirically validate this design choice in Appendix G.6.

3.4 Full-scale query interaction

In DETR-like detectors [4, 59, 25], final predictions are generated by decoder queries. Instead of attending only to high-level features, FSQI enables these queries to directly access the fine-grained details in P_1, further improving tiny object detection performance. Here, "full-scale" means that the decoder can attend to all feature levels, from the highest-resolution P_1 to deeper semantic features.

Full-scale encoder memory. We flatten and concatenate the semantically enhanced highest-resolution feature P_1, the detail-enhanced shallow feature P_2', and the higher-level features $\{P_3, \ldots, P_l\}$ to construct the full-scale encoder memory:

$$\mathcal{M} = \mathrm{Concat}\big(\mathrm{Flatten}(P_1), \mathrm{Flatten}(P_2'), \ldots, \mathrm{Flatten}(P_l)\big). \tag{6}$$

Here, $\mathcal{M}$ serves as the value tokens for decoder cross-attention. Since P_1 is semantically enhanced from raw C_1, its tokens are more compatible with the encoded features $\{P_3, \ldots, P_l\}$, alleviating the distribution gap between different-level features. To avoid noisy shallow responses disturbing query selection, FSQI excludes P_1 from query initialization and exposes it only during decoder cross-attention, where decoder queries can adaptively retrieve relevant high-resolution details.

Scale-aware sparse sampling. Since the receptive field of P_1 is relatively small, if we keep its sampling setting the same as other feature levels, decoder queries may fail to collect enough information from P_1. We therefore allocate more sampling points on P_1 in decoder deformable cross-attention phase. In practice, FSQI increases the number of sampling points on P_1 from 4 to 8, while keeping it unchanged for the remaining feature levels. This gives queries a larger local search radius on P_1 and improves their access to fine-grained spatial cues for tiny object detection.

4 Experiments

4.1 Datasets and evaluation metrics

Datasets. We evaluate ReDHF on three challenging tiny-object detection datasets: AI-TOD-v2 [46], VisDrone [58], and SODA-D [6]. AI-TOD-v2 and VisDrone cover dense aerial scenes with many tiny instances, while SODA-D provides high-resolution traffic scenes with many small and extremely small objects. These datasets cover both aerial and driving scenarios, making them suitable for evaluating the generality of ReDHF. Detailed dataset statistics are provided in Appendix B.

Evaluation metrics. We use Average Precision (AP) [21] as the main metric, which is calculated as the mean AP over the range of IoU thresholds from 0.50 to 0.95, with an interval of 0.05. To better evaluate performance on tiny objects, we also report dataset-specific scale-aware metrics. For AI-TOD-v2 [46], we follow [45] and report AP_{vt}, AP_t, AP_s, and AP_m. For VisDrone [58] and SODA-D [6], we report AP_s and AP_{eS} as the tiny-object metrics, respectively. Full scale-aware metrics are provided in Appendix C.

Table 1: Experiments on AI-TOD-v2 test split. All models are evaluated with 800×800 input resolution. We compare ReDHF with both CNN-based and Transformer-based detectors. † denotes our re-implementation of the corresponding baseline. ⋆ denotes the average GFLOPs measured over the whole split when the computational cost varies across images.

Method	**#Params.**	**GFLOPs**	**AP**	**AP$_{50}$**	**AP$_{75}$**	**AP$_{vt}$**	**AP$_t$**	**AP$_s$**	**AP$_m$**
CNN-based models									
Cascade R-CNN [1]	68.9M	88.7	15.1	34.2	11.2	0.1	11.5	26.7	38.5
DetectoRS [32]	134.8M	≈ 160	16.1	35.5	12.5	0.1	12.6	28.3	40.0
DetectoRS w/ RFLA [47]	–	–	24.8	55.2	18.5	9.3	24.8	30.3	38.2
YOLOv12-X [42]	59.3M	185.4	16.1	33.5	13.4	5.4	17.0	21.4	28.5
Transformer-based models									
DNTR [23]	128.4M	298.1	26.2	56.7	20.2	12.8	26.4	31.0	37.0
Deformable-DETR[59]	40M	232.5	18.2	59.7	10.2	5.8	16.8	25.2	34.0
DINO-4scale† [52]	47M	354.4	25.1	60.8	16.3	12.1	24.0	31.0	40.5
DQ-DETR† [15]	58.7M	1805.4⋆	30.2	68.6	22.3	15.3	30.5	36.5	44.6
Dome-DETR-S† [14]	13.2M	173.1⋆	32.1	66.3	26.9	17.5	32.1	38.2	45.0
Dome-DETR-M† [14]	23.9M	252.6⋆	34.0	68.8	29.4	18.9	34.1	38.9	47.4
Dome-DETR-L† [14]	36.0M	358.7⋆	34.6	69.2	30.4	19.6	35.6	**39.8**	47.3
Our method									
DINO-4scale w/ ReDHF	49.7M	407.8	26.0+0.9	62.1	17.3	13.8+1.7	25.2	32.0	39.6
DQ-DETR w/ ReDHF	61.9M	1872.3⋆	31.4+1.2	69.8	22.6	16.6+1.3	31.2	36.7	44.3
Dome-DETR-S w/ ReDHF	15.9M	188.2⋆	33.5+1.4	68.1	28.9	18.9+1.4	33.5	38.7	45.8
Dome-DETR-M w/ ReDHF	26.7M	304.7⋆	35.0+1.0	68.9	30.6	20.7+1.8	34.8	39.2	**47.7**
Dome-DETR-L w/ ReDHF	39.0M	415.3⋆	**35.7**+1.1	**70.2**	**31.5**	**20.8**+1.2	**35.9**	**39.8**	47.4

Table 2: Experiments on VisDrone validation split and SODA-D test split. Each entry reports baseline → ReDHF, with the gain over the corresponding baseline shown in parentheses. AP$_s$ is used as the tiny-object metric for VisDrone, and AP$_{eS}$ is used for SODA-D.

Baseline	**VisDrone AP**	**VisDrone AP$_s$**	**SODA-D AP**	**SODA-D AP$_{eS}$**
DINO-4scale [52]	34.3 → 35.4 (+1.1)	25.6 → 27.4 (+1.8)	28.6 → 30.1 (+1.5)	10.5 → 12.6 (+2.1)
DQ-DETR [15]	35.7 → 37.1 (+1.4)	28.2 → 29.6 (+1.4)	34.4 → 35.5 (+1.1)	17.0 → 18.5 (+1.5)
Dome-DETR-S [14]	35.9 → 37.5 (+1.6)	28.3 → 30.3 (+2.0)	35.4 → 36.8 (+1.4)	17.6 → 19.6 (+2.0)
Dome-DETR-M [14]	38.9 → 40.2 (+1.3)	31.4 → **33.1** (+1.7)	37.2 → 38.4 (+1.2)	18.6 → 20.4 (+1.8)
Dome-DETR-L [14]	39.0 → **40.6** (+1.6)	31.5 → **33.1** (+1.6)	37.6 → **38.7** (+1.1)	19.7 → **21.4** (+1.7)
Avg. gain	**+1.4**	**+1.7**	**+1.3**	**+1.8**

4.2 Implementation details

We evaluate ReDHF on three representative baselines: DINO-4scale [52], DQ-DETR [15], and Dome-DETR [14]. Dome-DETR uses HGNetv2 [56] as the backbone, while DINO-4scale and DQ-DETR use ResNet-50 [13]. All experiments are trained end-to-end and evaluated on 4 Tesla V100 GPUs. Unless otherwise stated, we keep all the baseline architectures unchanged and only introduce the three components of our method. For Dome-DETR and DQ-DETR, we follow their original configurations on AI-TOD-v2 and VisDrone. Since SODA-D is not included in their original benchmarks, we reuse the VisDrone training configuration for this dataset. For DINO-4scale, we adopt its default COCO [21] 24-epoch training recipe for all three datasets, changing only dataset-specific settings. For EGSS, we set $r_{\min} = 0$ for all datasets, with $r_{\max} = 8$ for AI-TOD-v2 and $r_{\max} = 16$ for VisDrone and SODA-D. Complete training and implementation details are provided in Appendix A.

4.3 Main results

AI-TOD-v2. Table 1 reports the results on the AI-TOD-v2 [46] test split. ReDHF consistently improves all evaluated DETR-like variants, with average gains of +1.1 AP and +1.5 AP$_{vt}$. Dome-DETR-L w/ ReDHF achieves the best result of 35.7 AP, surpassing previous state-of-the-art methods.

The improvement is more notable on very tiny objects, with an average gain of +1.5 AP_{vt} across the three baseline variants. ReDHF brings consistent gains for most metrics except for AP_s or AP_m, which is expected since medium objects rely less on fine-grained local details and can already be well represented by deeper semantic features.

VisDrone and SODA-D. Table 2 reports results on VisDrone [58] and SODA-D [6]. ReDHF improves all baselines on both datasets, with average gains of +1.4 AP / +1.7 AP_s on VisDrone and +1.3 AP / +1.8 AP_{eS} on SODA-D. The best variants achieve 40.6 AP on VisDrone and 38.7 AP on SODA-D, respectively. Full scale-aware results are provided in Appendix C.

4.4 Efficiency analysis

ReDHF aims to improve tiny object detection with moderate overhead. As shown in Table 1, it consistently improves DETR-like baselines with limited increases in model size and computation. For example, on Dome-DETR-S [14], ReDHF improves AP by +1.4 while just increasing the average GFLOPs from 173.1 to 188.2. Similar trends are observed on DINO-4scale [52], DQ-DETR [15], and other Dome-DETR variants. Appendix D provides a detailed complexity analysis.

4.5 Ablation study

We conduct ablation studies on AI-TOD-v2 [46] to analyze the effectiveness of ReDHF and its key design choices. Unless otherwise stated, Dome-DETR-S [14] is used as the default baseline. Additional ablations on sampling points, and EGSS loss design are provided in Appendix G.

Main ablation experiment. Table 3 reports the contribution of each component. For CSDF, raw C_1 reduces AP from 32.1 to 31.1, while the semantically aligned P_1 improves AP to 32.7, indicating that raw high-resolution details suffer from background responses and semantic mismatch. For FSQI, adding raw C_1 to the encoder memory improves AP to 32.5 since decoder queries can selectively attend to useful high-resolution details and are therefore less sensitive to shallow noise, and replacing it with P_1 further improves AP to 32.8. In contrast, using P_1 for query initialization drops AP to 31.9, suggesting that shallow high-resolution features are unsuitable for initializing decoder queries. When cooperating CSDF and FSQI together, AP can increase to 33.1, and EGSS further improves the full model to 33.5 AP and 18.9 AP_{vt}, confirming the complementary effects of all components.

Table 3: Component ablation of ReDHF. C_1 and P_1 denote the raw highest-resolution feature and its semantically enhanced one. "q. init." means query initialization with P_1.

CSDF		FSQI			EGSS	AP	AP_{vt}	AP_t
C_1	P_1	C_1	P_1	q. init.				
						32.1	17.5	32.1
✓						31.1	16.8	30.8
	✓					32.7	18.0	32.7
		✓				32.5	17.8	32.4
			✓	✓		31.9	17.6	32.0
			✓			32.8	18.1	32.8
	✓		✓			33.1	18.3	33.0
	✓		✓		✓	**33.5**	**18.9**	**33.5**

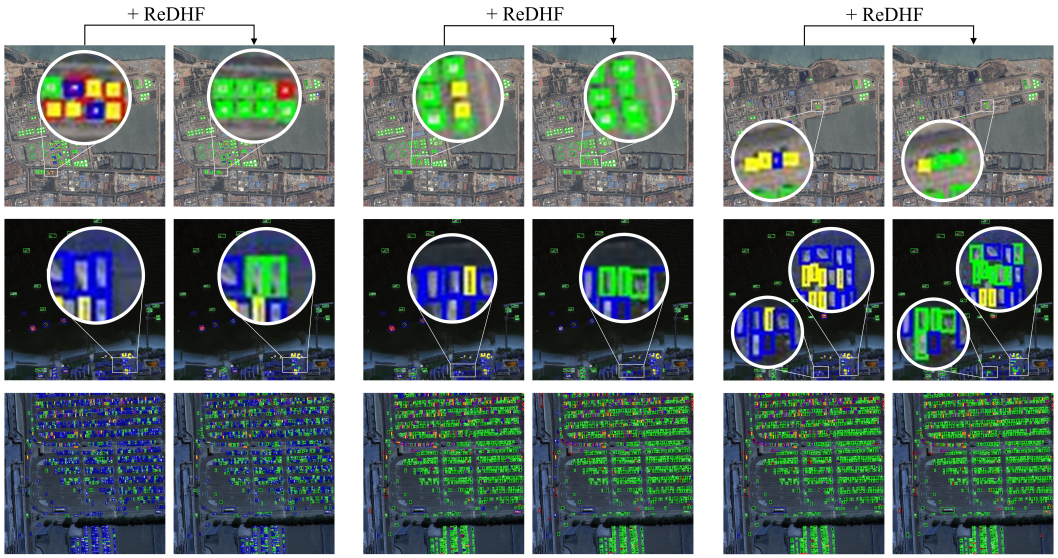
Table 4: Comparison of different label assignment strategies for EGSS. The best result is highlighted in bold.

Assignment	AP	AP_{50}	AP_{vt}
Rectangle	33.2	66.9	17.6
Circle	33.3	67.8	18.1
Ellipse (ours)	**33.5**	**68.1**	**18.9**

Table 5: Ablation on the detail injection level.

Inject layer	AP	AP_{50}	AP_{vt}
$P_2 \sim P_3$	32.8	66.5	18.0
$P_2 \sim P_5$	32.4	66.3	18.3
P_2 only	**33.5**	**68.1**	**18.9**

Design ablations. Tables 4 and 5 study two key design choices of ReDHF. For EGSS, Rectangle, Circle, and Ellipse denote box-based, short-axis circle-based, and ellipse-guided assignment, respectively. Our method achieves the best performance, improving AP_{vt} by +1.3 over Rectangle and +0.8 over Circle. For CSDF, injecting high-resolution details only into P_2 gives the best result, reaching 33.5 AP and 18.9 AP_{vt}. Full results and analysis are provided in Appendix G.1 and Appendix G.2.

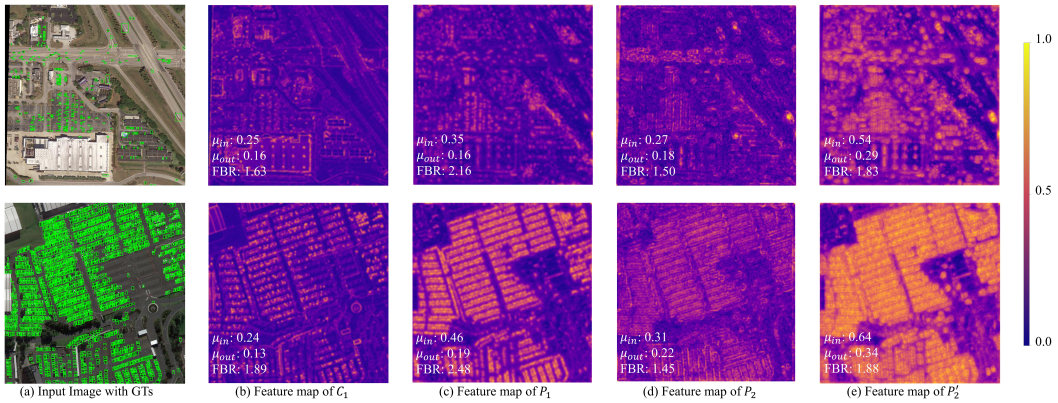


(a) DINO-4scale: baseline (left) vs. ReDHF (right) (b) DQ-DETR: baseline (left) vs. ReDHF (right) (c) Dome-DETR-L: baseline (left) vs. ReDHF (right)

Figure 3: Qualitative comparison on the AI-TOD-v2 test split. Each pair compares a baseline detector with its ReDHF variant on the same scene. The green, red, yellow, and blue boxes represent TP, FP, FN with ground-truth box short side ≤ 4 pixels, and the remaining FN, respectively. Compared with the baselines, ReDHF produces cleaner predictions and helps recover more tiny objects, with more green boxes and fewer red/yellow/blue boxes. Class labels are omitted for clarity.

4.6 Visualization analysis

Detection visualization. Fig. 3 compares ReDHF with corresponding baselines across different scenarios on AI-TOD-v2 [46]. ReDHF makes better predictions and recovers more ultra-tiny objects, showing that our method helps detectors better preserve and use fine-grained details for tiny objects.



(a) Input Image with GTs (b) Feature map of C_1 (c) Feature map of P_1 (d) Feature map of P_2 (e) Feature map of P_2'

Figure 4: Feature visualization and response analysis of CSDF. We compare C_1 vs. P_1 and P_2 vs. P_2'. The bottom-left statistics report μ_{in}, μ_{out}, and FBR defined in Eq. 7. After CSDF enhancement, P_1 and P_2' show stronger response concentration on GT regions.

Feature maps visualization. To further analyze CSDF, we compute channel-wise ℓ_2 response maps and measure their mean responses inside and outside GT regions, denoted as μ_{in} and μ_{out}. The foreground-to-background response ratio (FBR) is then defined as:

$$\mu_{\text{in}} = \frac{1}{|\Omega_{\text{in}}|} \sum_{(x,y)\in\Omega_{\text{in}}} R(x,y), \quad \mu_{\text{out}} = \frac{1}{|\Omega_{\text{out}}|} \sum_{(x,y)\in\Omega_{\text{out}}} R(x,y), \quad \text{FBR} = \frac{\mu_{\text{in}}}{\mu_{\text{out}}}. \tag{7}$$

Here, $R(x, y)$ denotes the normalized response at location (x, y), while Ω_{in} and Ω_{out} denote the set of locations inside and outside GT regions. A larger FBR indicates stronger response concentration on object regions. As shown in Fig. 4, P_1 achieves higher FBR than raw C_1, and P_2' shows the same improvement over P_2. This suggests that CSDF makes the highest-resolution feature more object-focused and transfers more effective fine-grained details to the shallow feature.

5 Conclusion

In this paper, we propose ReDHF, a plug-and-play framework that revives the discarded highest-resolution feature for tiny object detection. Motivated by the contradiction between the limited representation of tiny objects and the high cost of utilizing high-resolution features, we introduce CSDF, EGSS, and FSQI to effectively address these issues, achieving consistent improvements across different DETR-like models on the AI-TOD-v2, VisDrone and SODA-D datasets. Although retaining the highest-resolution representation introduces additional computation, ReDHF yields consistent gains that are most pronounced on tiny objects, and we only observe localized degradation for large objects in dense scenes with smaller-capacity variants. Detailed analyses are provided in Appendix F. Future work will explore more adaptive high-resolution feature utilization under low-resource and crowded scenes. We hope ReDHF can inspire continued progress in this area.

Acknowledgments and Disclosure of Funding

This work was supported by the Pengcheng Laboratory Research Project (PCL2025A14), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011292, Project No. 2024A1515011437, Project No. 2025B1515120012), and the National Natural Science Foundation of China (Grant No. 62606254). The authors declare no competing interests.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[2] Guiping Cao, Wenjian Huang, Xiangyuan Lan, Jianguo Zhang, Dongmei Jiang, and Yaowei Wang. Mlp-dino: category modeling and query graphing with deep mlp for object detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24), Jeju, Republic of Korea*, pages 3–9, 2024.

[3] Guiping Cao, Wenjian Huang, Xiangyuan Lan, Jianguo Zhang, Dongmei Jiang, and Yaowei Wang. Cross-dino: Cross the deep mlp and transformer for small object detection. *IEEE Transactions on Multimedia*, 27:7369–7379, 2025.

[4] 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.

[5] Qiang Chen, Xiangbo Su, Xinyu Zhang, Jian Wang, Jiahui Chen, Yunpeng Shen, Chuchu Han, Ziliang Chen, Weixiang Xu, Fanrong Li, et al. Lw-detr: A transformer replacement to yolo for real-time detection. *arXiv preprint arXiv:2406.03459*, 2024.

[6] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13467–13488, 2023.

[7] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016.

[8] Sohail Ali Farooqui, Zuhair Ahmed Khan Taha, Mohammed Mudassir Uddin, and Shahnawaz Alam. Laf-yolov10 with partial convolution backbone, attention-guided feature pyramid, auxiliary p2 head, and wise-iou loss for small object detection in drone aerial imagery. *arXiv preprint arXiv:2602.13378*, 2026.

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[10] Yuqi Gong, Xuehui Yu, Yao Ding, Xiaoke Peng, Jian Zhao, and Zhenjun Han. Effective fusion factor in fpn for tiny object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1160–1168, 2021.

[11] Cao Guiping, Huang Xiangyuan, Lan Wenjian, Zhang Jianguo, Jiang Dongmei, and Yaowei Wang. Ds-det: Single-query paradigm and attention disentangled learning for flexible object detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*.

[12] 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.

[13] 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.

[14] Zhangchi Hu, Peixi Wu, Jie Chen, Huyue Zhu, Yijun Wang, Yansong Peng, Hebei Li, and Xiaoyan Sun. Dome-detr: Detr with density-oriented feature-query manipulation for efficient tiny object detection, 2025.

[15] Yi-Xin Huang, Hou-I Liu, Hong-Han Shuai, and Wen-Huang Cheng. Dq-detr: Detr with dynamic query for tiny object detection. In *European Conference on Computer Vision*, pages 290–305. Springer, 2025.

[16] Zhenchao Jin, Dongdong Yu, Luchuan Song, Zehuan Yuan, and Lequan Yu. You should look at all objects. In *European conference on computer vision*, pages 332–349. Springer, 2022.

[17] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements (2024). *arXiv preprint arXiv:2410.17725*, 2024.

[18] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.

[19] 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.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[21] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[22] Chenguang Liu, Guangshuai Gao, Ziyue Huang, Zhenghui Hu, Qingjie Liu, and Yunhong Wang. Yolc: You only look clusters for tiny object detection in aerial images. *IEEE Transactions on Intelligent Transportation Systems*, 25(10):13863–13875, 2024.

[23] Hou-I Liu, Yu-Wen Tseng, Kai-Cheng Chang, Pin-Jyun Wang, Hong-Han Shuai, and Wen-Huang Cheng. A denoising fpn with transformer r-cnn for tiny object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024.

[24] Kai Liu, Zhihang Fu, Sheng Jin, Ze Chen, Fan Zhou, Rongxin Jiang, Yaowu Chen, and Jieping Ye. Esod: Efficient small object detection on high-resolution images. *IEEE Transactions on Image Processing*, 34:183–195, 2024.

[25] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*, 2022.

[26] 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.

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[28] Yuchen Ma, Songtao Liu, Zeming Li, and Jian Sun. Iqdet: Instance-wise quality distribution sampling for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1717–1725, 2021.

[29] 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.

[30] Yansong Peng, Hebei Li, Peixi Wu, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. D-fine: Redefine regression task in detrs as fine-grained distribution refinement, 2024.

[31] Yansong Peng, Hebei Li, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. Scene adaptive sparse transformer for event-based object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16794–16804, 2024.

[32] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10213–10224, 2021.

[33] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.

[34] 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.

[35] 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.

[36] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[37] Aashish Sharma. Dm-efs: Dynamically multiplexed expanded features set form for robust and efficient small object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24569–24579, 2025.

[38] Huang Shihua, Lu Zhichao, Cun Xiaodong, Yu Yongjun, Zhou Xiao, and Shen Xi. Deim: Detr with improved matching for fast convergence, 2025.

[39] H. Sun, Y. Li, L. Yang, X. Cao, and B. Zhang. Uncertainty-aware gradient stabilization for small object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8407–8417, 2025.

[40] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[41] Shiyi Tang, Shu Zhang, and Yini Fang. Hic-yolov5: Improved yolov5 for small object detection. In *2024 IEEE international conference on robotics and automation (ICRA)*, pages 6614–6619. IEEE, 2024.

[42] Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

[43] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *Proc. Int. Conf. Computer Vision (ICCV)*, 2019.

[44] Chengcheng Wang, Wei He, Ying Nie, Jianyuan Guo, Chuanjian Liu, Yunhe Wang, and Kai Han. Gold-yolo: Efficient object detector via gather-and-distribute mechanism. *Advances in neural information processing systems*, 36:51094–51112, 2023.

[45] Jinwang Wang, Wen Yang, Haowen Guo, Ruixiang Zhang, and Gui-Song Xia. Tiny object detection in aerial images. pages 3791–3798, 2021.

[46] Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. Detecting tiny objects in aerial images: A normalized wasserstein distance and a new benchmark. In *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 190, pages 79–93, 2022.

[47] Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. Rfla: Gaussian receptive field based label assignment for tiny object detection. In *European conference on computer vision*, pages 526–543. Springer, 2022.

[48] Haozhi Xu, Xiaofang Yuan, and Yaonan Wang. Dpaks: Reliable detr guided by prior auxiliary knowledge for small object detection. *IEEE Transactions on Multimedia*, 2026.

[49] Chenhongyi Yang, Zehao Huang, and Naiyan Wang. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 13668–13677, 2022.

[50] Fan Yang, Cheng Lu, Yandong Guo, Longin Jan Latecki, and Haibin Ling. Dually supervised feature pyramid for object detection and segmentation. *arXiv preprint arXiv:1912.03730*, 2019.

[51] Xiang Yuan, Gong Cheng, Kebing Yan, Qinghua Zeng, and Junwei Han. Small object detection via coarse-to-fine proposal generation and imitation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6317–6327, October 2023.

[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. *arXiv preprint arXiv:2203.03605*, 2022.

[53] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9759–9768, 2020.

[54] Shilong Zhang, Xinjiang Wang, Jiaqi Wang, Jiangmiao Pang, Chengqi Lyu, Wenwei Zhang, Ping Luo, and Kai Chen. Dense distinct query for end-to-end object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7329–7338, June 2023.

[55] Tianyang Zhang, Xiangrong Zhang, Chaozhuo Hua, Guanchun Wang, Xiao Han, and Licheng Jiao. Tiny object detection based on dynamic scale-awareness label assignment and contextual enhancement. *Pattern Recognition*, page 112449, 2025.

[56] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16965–16974, 2024.

[57] 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.

[58] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2021.

[59] 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.

[60] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *European conference on computer vision*, pages 566–583. Springer, 2020.

Appendix

This appendix provides additional implementation details, complete experimental results, and supplementary analyses for ReDHF. Section A describes the training and method-specific configurations. Section B provides detailed statistics of the three benchmarks, while Section C reports the complete results on VisDrone and SODA-D. Section D analyzes the practical efficiency and cost-accuracy trade-off of high-resolution feature reuse. Section E examines performance across object scales and generalization to a general-purpose detection benchmark. Section F discusses the practical limitations and failure cases of ReDHF. Finally, Section G provides additional ablations and robustness analyses of the key design choices.

A Additional Implementation Details

Baselines and backbones. We evaluate ReDHF on three representative DETR-based detectors: DINO-4scale [52], DQ-DETR [15], and Dome-DETR [14]. Dome-DETR adopts HGNetv2 [56] as its backbone, while DINO-4scale and DQ-DETR use ResNet-50 [13]. All experiments are trained end-to-end and evaluated on 4 Tesla V100 GPUs. Unless otherwise stated, we keep all the baseline architectures unchanged and only introduce the three components of our method.

Training configurations. For Dome-DETR and DQ-DETR, we follow their original training configurations on AI-TOD-v2 and VisDrone. Since SODA-D is not included in their original benchmarks, we use the corresponding VisDrone training configuration for SODA-D. For DINO-4scale, we follow its default COCO [21] 24-epoch training recipe on all three datasets, while adapting only the dataset-specific settings.

ReDHF settings. For EGSS, we set $r_{\min} = 0$ on all datasets. We use $r_{\max} = 8$ on AI-TOD-v2 and $r_{\max} = 16$ on VisDrone and SODA-D. The default auxiliary-loss weights are $\lambda_{\text{cls}} = 1.0$ and $\lambda_{\text{reg}} = 0.10$, and centerness supervision is not used. For the deformable interactions, we use 8 sampling points by default in both CSDF and FSQI. The sensitivity analyses of the sampling-point number and the EGSS loss configuration are provided in Sec. G.4 and Sec. G.6, respectively.

B Dataset details

AI-TOD-v2. AI-TOD-v2 [46] contains 28,036 aerial images with 752,745 annotated object instances. The dataset is split into 11,214 training images, 2,804 validation images, and 14,018 test images. It is characterized by extremely small object sizes, with an average object size of only 12.7 pixels. In particular, 86% of the objects are smaller than 16 pixels, and the largest object does not exceed 64 pixels. The object density also varies greatly across images, ranging from 1 to 2,667 objects per image, with an average of 24.64 and a standard deviation of 63.94.

VisDrone. VisDrone [58] contains 14,018 images across 10 distinct categories, with 6,471 images for training, 548 for validation, and 3,190 for testing. It covers diverse objects such as pedestrians, bicycles, and vehicles, and presents severe challenges in terms of diverse crowd densities. Specifically, the average number of objects per image is 40.7, with a standard deviation of 46.41.

SODA-D. SODA-D [6] contains 24,828 high-quality traffic-scene images with 278,433 annotated instances across 9 categories. The training, validation, and test subsets are split with an approximate ratio of 50% : 20% : 30%. The dataset includes high-resolution driving scenes with many small and extremely small objects under diverse real-world conditions, including different viewpoints, weather, and illumination. These properties make SODA-D a challenging benchmark for small object detection in driving scenarios.

C Full results on VisDrone and SODA-D

VisDrone. Table 6 reports the results on the VisDrone [58] validation split. Our ReDHF on DQ-DETR [15] improves AP by +1.4%. On Dome-DETR-S and Dome-DETR-M, it raises AP from 35.9

Table 6: Experiments on VisDrone val split. We compare ReDHF with both CNN-based and Transformer-based detectors. † denotes our re-implementation of the corresponding baseline. ⋆ denotes the average GFLOPs measured over the whole split when the computational cost varies across images.

Method	#Params.	GFLOPs	Image Scale	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
CNN-based models									
YOLOv11 [17]	25.3M	86.6	640×640	27.9	–	–	18.1	40.9	53.1
YOLOv12 [42]	26.3M	88.6	640×640	29.2	–	–	18.7	41.8	51.7
RFLA [47]	69.2M	162.0	800×800	24.3	–	–	14.4	35.9	45.8
YOLC [22]	–	–	1024×640	39.6	–	–	32.9	49.3	56.2
CFINet [51]	41.4M	291.9	1200×1200	23.9	–	–	16.3	34.7	57.4
ESOD [24]	–	232.9	1920×1920	39.4	62.9	–	31.1	–	–
Transformer-based models									
DDQ DETR [54]	–	270	1333×800	31.6	–	–	23.2	42.1	51.1
DPAKS [48]	218.3M	1457.6	1333×800	39.1	–	–	32.1	49.5	**63.4**
UGS [39]	–	–	1333×800	38.1	61.9	–	24.2	47.3	56.6
DINO-4scale† [52]	47M	544.3	1333×800	34.3	57.3	34.8	25.6	45.6	59.5
DQ-DETR† [15]	58.7M	2826.6⋆	1333×800	35.7	59.4	36.4	28.2	45.8	55.1
Dome-DETR-S† [14]	13.2M	167.3⋆	800×800	35.9	58.0	37.2	28.3	45.9	58.9
Dome-DETR-M† [14]	23.9M	296.2⋆	800×800	38.9	62.1	40.7	31.4	49.3	62.3
Dome-DETR-L† [14]	36.0M	385.1⋆	800×800	39.0	62.3	40.6	31.5	49.3	62.8
Our method									
DINO-4scale w/ ReDHF	49.7M	627.6	1333×800	35.4+1.1	58.1	36.2	27.4+1.8	45.6	59.4
DQ-DETR w/ ReDHF	61.9M	2928.3⋆	1333×800	37.1+1.4	61.0	37.9	29.6+1.4	46.5	57.1
Dome-DETR-S w/ ReDHF	15.9M	180.8⋆	800×800	37.5+1.6	59.4	39.5	30.3+2.0	47.1	58.6
Dome-DETR-M w/ ReDHF	26.7M	338.3⋆	800×800	40.2+1.3	63.3	42.6	33.1+1.7	50.0	62.0
Dome-DETR-L w/ ReDHF	39.0M	449.3⋆	800×800	**40.6**+1.6	**64.0**	**42.8**	**33.1**+1.6	**51.0**	63.1

to 37.5 and from 38.9 to 40.2, respectively. When applied to Dome-DETR-L, ReDHF achieves the best AP(40.6%), improving AP, AP_{50}, and AP_{75} by +1.6%, +1.7%, and +2.2%, respectively.

SODA-D. Table 7 reports the results on the SODA-D [6] test split. ReDHF brings +1.5% AP on DINO-4scale, +1.1% AP on DQ-DETR, and +1.4%, +1.2%, +1.1% AP on Dome-DETR-S/M/L, respectively. The gains are more pronounced on extremely small objects. Dome-DETR-L w/ ReDHF achieves the best result, with 38.7 AP and 21.4 AP_{eS}.

D Detailed efficiency analysis

D.1 Computational cost of high-resolution feature reuse

Let N_i denote the number of tokens in P_i, and K denote the number of sampling points. Standard global self attention for P_i and P_{i+1} has a complexity of $\mathcal{O}(N_i^2 \cdot N_{i+1}^2)$, while that in deformable self attention is $\mathcal{O}(K \cdot N_i + K \cdot N_{i+1})$, which becomes expensive when the spatial resolution is large. In contrast, the complexity of our deformable interactions in CSDF is $\mathcal{O}(K \cdot N_{i+1})$, where $K \ll N_i$, making the use of the highest-resolution feature P_1 much more practical. Table 8 compares the computational cost of different ways to use the highest-resolution feature on DINO-4scale [52]. Directly adding raw C_1 into the encoder and decoder causes a large cost increase, raising the whole-model GFLOPs from 354.4 to 3057.7. The main cost comes from the encoder, whose GFLOPs increase from 203.9 to 2709.5 due to dense interaction on the highest-resolution feature. The decoder cost also increases from 34.9 to 161.0. This is because each decoder layer performs value projection on the multi-scale memory, and the very large spatial size of C_1 makes repeated projection expensive.

In contrast, ReDHF increases the whole-model GFLOPs only to 407.8. Compared with direct C_1 usage, ReDHF reduces the total computation by about 86.7%, the encoder computation by about 90.7%, and the decoder computation by about 74.7%. This reduction comes from two designs. First, CSDF retrieves high-resolution details through sparse deformable interaction instead of dense global interaction. Second, FSQI exposes P_1 to decoder cross-attention in a lightweight manner. In our

Table 7: Experiments on SODA-D test split. We compare ReDHF with both CNN-based and Transformer-based detectors. † denotes our re-implementation of the corresponding baseline. ⋆ denotes the average GFLOPs measured over the whole split when the computational cost varies across images.

Method	**#Params.**	**GFLOPs**	**Image Scale**	**AP**	**AP$_{50}$**	**AP$_{75}$**	**AP$_{eS}$**	**AP$_{rS}$**	**AP$_{gs}$**	**AP$_{N}$**
CNN-based models										
Faster RCNN [35]	41.2M	292.3	1200×1200	28.9	59.7	24.2	13.9	25.6	34.3	43.2
Cascade RCNN [1]	68.9M	320.1	1200×1200	31.2	59.9	27.8	14.1	27.5	37.1	46.9
RFLA [47]	41.2M	292.1	1200×1200	29.7	60.2	25.2	13.2	26.9	35.4	44.6
Transformer-based models										
Deformable-DETR[59]	35.2M	739.1	1200×1200	19.2	44.8	13.7	6.3	15.4	24.9	34.2
DINO-4scale† [52]	47M	354.4	800×800	28.6	59.3	24.2	10.5	24.2	36.2	47.4
DQ-DETR† [15]	58.7M	1802.6*	800×800	34.4	62.6	31.8	17.0	30.9	41.2	51.4
Dome-DETR-S† [14]	13.2M	160.12	800×800	35.4	63.9	33.0	17.6	31.8	42.1	52.2
Dome-DETR-M† [14]	23.9M	291.78	800×800	37.2	66.6	34.7	18.6	34.1	44.8	53.6
Dome-DETR-L† [14]	36.0M	373.12	800×800	37.6	67.5	35.2	19.7	34.1	44.5	54.8
Our method										
DINO-4scale w/ ReDHF	49.7M	407.8	800×800	30.1**+1.5**	59.6	26.3	12.6**+2.1**	25.6	37.3	47.6
DQ-DETR w/ ReDHF	61.9M	1867.5	800×800	35.5**+1.1**	63.6	33.3	18.5**+1.5**	32.2	41.9	51.4
Dome-DETR-S w/ ReDHF	15.9M	184.08	800×800	36.8**+1.4**	65.7	34.5	19.6**+2.0**	33.6	43.2	52.8
Dome-DETR-M w/ ReDHF	26.7M	332.34	800×800	38.4**+1.2**	67.9	35.8	20.4**+1.8**	34.9	45.1	54.6
Dome-DETR-L w/ ReDHF	39.0M	435.78	800×800	38.7**+1.1**	68.4	36.3	21.4**+1.7**	35.1	45.4	55.0

Table 8: Computational cost comparison on DINO-4scale. We report GFLOPs of the backbone, encoder, decoder, other parts, and the whole model. "Direct C_1" denotes directly adding the raw highest-resolution feature into the encoder and decoder. "Others" includes components not counted in the backbone, encoder, or decoder.

Model	Backbone	Encoder	Decoder	Others	Whole
DINO-4scale [52]	104.5	203.9	34.9	11.1	354.4
DINO-4scale w/ direct C_1	104.5(–)	2709.5(+2505.6)	161.0(+126.1)	82.7(+71.6)	3057.7(+2703.3)
DINO-4scale w/ ReDHF	104.5(–)	251.2(+47.3)	40.7(+5.8)	11.4(+0.3)	407.8(+53.4)

implementation, the value projection for the highest-resolution feature is applied only once, while the remaining feature levels follow the original per-layer projection. Although FSQI allocates more sampling points to P_1, the total GFLOPs increase remains moderate, as shown in Fig. 5. Moreover, EGSS is only used during training and removed during inference, so it does not add test-time computation. Overall, ReDHF is an efficient high-resolution detail reintroduction framework for tiny object detection. It avoids the expensive computational cost of utilizing raw C_1, while still effectively releasing the potential of the highest-resolution feature for tiny object detection.

D.2 Compute-matched comparison with input resolution scaling

One misconception for the improvement of ReDHF is that it introduces additional computation for high-resolution feature processing. To distinguish the contribution of the proposed design from simply increasing the computational budget, we compare ReDHF with resolution-scaled baselines while keeping the remaining detector and training settings unchanged.

As shown in Table 9, increasing the input resolution of DINO-4scale from 800 to 864 raises the computation from 354.4 to 412.4 GFLOPs, but improves AP and AP$_{vt}$ by only 0.3 points. In contrast, ReDHF achieves 26.0 AP and 13.8 AP$_{vt}$ at 407.8 GFLOPs, outperforming the resolution-scaled baseline by 0.6 AP and 1.4 AP$_{vt}$ under a comparable computational budget. A similar trend is observed for Dome-DETR-L. These results indicate that the gains of ReDHF cannot be reproduced by uniformly increasing the input resolution, supporting the effectiveness of explicitly recovering and exploiting the discarded high-resolution representation.

Table 9: Compute-matched comparison with input-resolution scaling on AI-TOD-v2. We increase only the input resolution of the baselines while keeping the detector and training settings unchanged. The best result within each baseline family is highlighted in bold.

Model	Input	GFLOPs	AP	AP_{vt}
DINO-4scale	800	354.4	25.1	12.1
DINO-4scale	864	412.4	25.4	12.4
DINO-4scale w/ ReDHF	800	407.8	**26.0**	**13.8**
Dome-DETR-L	800	358.7	34.6	19.6
Dome-DETR-L	880	481.1	35.0	20.0
Dome-DETR-L w/ ReDHF	800	415.3	**35.7**	**20.8**

Table 10: Practical inference cost of ReDHF on AI-TOD-v2 and VisDrone. Wall-clock latency and GPU memory are measured on a single Tesla V100 with batch size 1 under identical baseline and ReDHF settings. Each entry reports baseline → ReDHF. Tiny AP denotes AP_{vt} on AI-TOD-v2 and AP_s on VisDrone.

Dataset	Variant	Tiny AP ↑	Latency (ms) ↓	Avg. memory (GiB) ↓	Peak memory (GiB) ↓
AI-TOD-v2	Dome-DETR-S	17.5 → **18.9**	409.6 → 507.6	0.30 → 0.56	0.63 → 1.72
	Dome-DETR-M	18.9 → **20.7**	450.0 → 578.0	0.39 → 0.65	0.77 → 1.86
	Dome-DETR-L	19.6 → **20.8**	487.8 → 655.2	0.47 → 0.76	0.86 → 1.99
VisDrone	Dome-DETR-S	28.3 → **30.3**	454.5 → 584.1	0.27 → 0.46	0.63 → 1.72
	Dome-DETR-M	31.4 → **33.1**	512.8 → 680.1	0.36 → 0.57	0.77 → 1.86
	Dome-DETR-L	31.5 → **33.1**	543.5 → 751.8	0.47 → 0.69	0.88 → 2.01

D.3 Inference latency and GPU memory

GFLOPs do not fully characterize the practical cost of retaining the highest-resolution feature. We therefore measure wall-clock inference latency and GPU memory under identical baseline and ReDHF settings. All measurements are conducted on a Tesla V100 with batch size 1.

Table 10 shows that ReDHF introduces additional latency and memory consumption due to retaining and querying P_1. Across the evaluated Dome-DETR variants on AI-TOD-v2 and VisDrone, the additional latency ranges from 98.0 to 208.3 ms, while the increase in average GPU memory ranges from 0.19 to 0.29 GiB. Meanwhile, ReDHF consistently improves the corresponding tiny-object AP by 1.2-2.0 points. These results quantify the practical cost associated with recovering high-resolution evidence and complement the GFLOPs analysis above. Overall, ReDHF achieves a favorable accuracy-efficiency trade-off, where the additional runtime and memory are accompanied by consistent improvements in tiny-object detection.

E Detection Quality and Generalization

E.1 Scale-wise analysis

Because ReDHF is designed primarily for tiny objects, we further examine whether recovering high-resolution evidence introduces a systematic trade-off at larger object scales. Table 11 summarizes the average gains over the five evaluated DETR variants, with the scale categories ordered from the smallest to larger objects for each dataset.

The improvements are consistently largest on the smallest object categories and gradually decrease with object size. Nevertheless, the averaged gains on the remaining scales are non-negative across all three datasets. This indicates that ReDHF concentrates its benefits on objects most affected by feature downsampling without systematically sacrificing the detection quality of larger objects.

Table 11: Scale-wise performance gains of ReDHF averaged over the five evaluated DETR variants. The dataset-specific metrics are ordered from smaller to larger object scales.

Dataset	Smallest objects	Intermediate objects	Larger objects
AI-TOD-v2	AP_{vt} +1.48	AP_t +0.86	AP_s +0.40, AP_m +0.00
VisDrone	AP_s +1.70	AP_m +0.86	AP_l +0.32
SODA-D	AP_{eS}+1.82	AP_{rS} +1.26	AP_{gS} +0.82, AP_N +0.40

Table 12: Generalization of ReDHF on COCO using DINO-4scale. Both models use the same 12-epoch training recipe.

Model	AP	AP_s	AP_m	AP_l
DINO-4scale	49.0	32.0	52.3	63.0
DINO-4scale w/ ReDHF	**49.6 (+0.6)**	**32.6 (+0.6)**	**53.1 (+0.8)**	**63.5 (+0.5)**

E.2 Generalization on COCO

To further examine whether ReDHF remains effective beyond tiny-object-focused benchmarks, we evaluate DINO-4scale with ReDHF on COCO using the same training recipe as the baseline. As shown in Table 12, ReDHF improves the overall AP from 49.0 to 49.6. Improvements are also observed across all object scales, with AP_s, AP_m, and AP_l increasing by 0.6, 0.8, and 0.5 points, respectively. These results suggest that recovering high-resolution evidence does not require sacrificing medium- or large-object performance and that ReDHF can also provide complementary information on a general-purpose detection benchmark.

F Limitations Analysis

Retaining and querying an additional highest-resolution feature P_1 inevitably introduces extra computation and memory compared with the original detectors. As quantified in Table 8, however, this overhead is substantially smaller than that of directly incorporating the raw highest-resolution feature. More importantly, the additional cost is accompanied by consistent improvements in tiny-object detection, indicating a favorable accuracy-efficiency trade-off for exploiting high-resolution evidence.

As analysed in Sec. E, the benefits of ReDHF are concentrated on tiny objects, and the average improvement gradually decreases as object size increases, although no systematic degradation is observed at larger scales. This behavior is consistent with the motivation of ReDHF: high-resolution evidence is most useful for objects whose visual information is severely affected by feature downsampling. To identify where the occasional degradation on larger objects occurs, we divide the VisDrone validation set into two equally sized subsets according to the number of annotated instances per image and evaluate the same trained checkpoints on both subsets.

Table 13: Density-wise analysis on the VisDrone validation set. Images are ranked by the number of annotated instances and divided into two equally sized sparse and dense subsets. Each entry reports baseline → ReDHF, with the absolute change in parentheses.

Variant	Sparse AP_s	Sparse AP_m	Sparse AP_l	Dense AP_s	Dense AP_m	Dense AP_l
Dome-DETR-S	28.24 → 30.31 (**+2.07**)	47.09 → 48.91 (**+1.82**)	60.60 → 63.12 (**+2.52**)	28.11 → 30.58 (**+2.47**)	44.76 → 45.96 (**+1.20**)	57.15 → 55.60 (-1.55)
Dome-DETR-M	30.86 → 33.02 (**+2.16**)	50.80 → 51.10 (**+0.30**)	64.86 → 65.13 (**+0.27**)	32.02 → 33.46 (**+1.44**)	48.46 → 49.41 (**+0.95**)	60.76 → 59.80 (-0.96)
Dome-DETR-L	31.54 → 32.90 (**+1.36**)	50.12 → 52.09 (**+1.97**)	68.34 → 68.86 (**+0.52**)	31.71 → 33.37 (**+1.66**)	49.15 → 50.40 (**+1.25**)	55.60 → 56.62 (**+1.02**)

As shown in Table 13, across all three variants, ReDHF improves all object scales in sparse scenes. In dense scenes, AP_s and AP_m remain consistently improved, whereas AP_l decreases for Dome-DETR-S and Dome-DETR-M but increases for Dome-DETR-L. Dense scenes contain many spatially adjacent high-resolution responses, which may increase background interference and hinder smaller-capacity detectors from preserving the broader semantic context required for large-object recognition.

Table 14: Comparison of different label assignment strategies for EGSS. The best result in each column is highlighted in bold.

Assignment strategy	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
Rectangle-based	33.2	66.9	28.7	17.6	33.0	38.2	45.0
Short-axis circle-based	33.3	67.8	28.4	18.1	32.9	38.4	**46.1**
Ellipse-guided (ours)	**33.5**	**68.1**	**28.9**	**18.9**	**33.5**	**38.7**	45.8

Table 15: Ablation on the detail injection level. We keep other components unchanged and only vary the target feature levels. The best result in each column is highlighted in bold.

Inject layer	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
$P_2 \sim P_3$	32.8	66.5	28.1	18.0	32.7	38.5	45.0
$P_2 \sim P_5$	32.4	66.3	27.3	18.3	32.2	38.2	44.6
P_2 only	**33.5**	**68.1**	**28.9**	**18.9**	**33.5**	**38.7**	**45.8**

Therefore, the degradation is specific to large objects in dense scenes for smaller-capacity detectors, rather than a systematic trade-off across object scales.

G Additional ablations and Robustness analyses

G.1 Label assignment strategy in EGSS

Table 14 compares different label assignment strategies for EGSS. The results show that the positive sample definition is important for shallow auxiliary supervision. Rectangle-based assignment reaches 33.2 AP, but its AP_{vt} is 17.6. This suggests that using the whole box region as positives may introduce boundary background pixels, which is harmful for very tiny objects.

We further compare a short-axis circle-based strategy, where the positive region is a circle centered at the object center and its radius is set to the short semi-axis of the corresponding ground-truth box. This strategy reduces boundary noise and improves AP_{vt} from 17.6 to 18.1, but it still ignores object aspect ratios and may discard valid positive regions along the long axis.

In contrast, ellipse-guided assignment adapts to the object geometry by using both the long and short axes. It achieves the best AP, AP_{50}, AP_{75}, AP_{vt}, AP_t, and AP_s. Compared with rectangle-based and short-axis circle-based assignment, it improves AP_{vt} by +1.3 and +0.8, respectively. These results show that the elliptical constraint better suppresses boundary noise while preserving sufficient valid positives for tiny objects. Although the circle-based strategy obtains the best AP_m, EGSS mainly targets tiny and small objects, where ellipse-guided assignment gives more consistent gains. Therefore, we adopt ellipse-guided assignment as the default strategy.

G.2 Detail injection level analysis

Table 15 studies where the highest-resolution details from P_1 should be injected. All settings keep the other components unchanged and only vary which feature levels receive the highest-resolution details. Injecting details only into P_2 gives the best result, reaching 33.5 AP and 18.9 AP_{vt}. This suggests that P_2 is the most suitable level to receive highest-resolution details, because it is spatially closest to P_1 while still containing more semantic information than raw shallow features. In contrast, injecting details into more levels does not bring further gains. When the injection range is extended to $P_2 \sim P_3$ or $P_2 \sim P_5$, AP decreases to 32.8 and 32.4, respectively. This is likely because deeper feature levels are more responsible for medium and large objects, while excessive high-resolution details may become redundant or even introduce noise into semantic features. These results support our design choice of injecting high-resolution details only into P_2.

Table 16: Comparison of different strategies for downsampling P_1 to the P_2 resolution in CSDF. We keep the semantically enhanced P_1 and other ReDHF components unchanged, and only vary the strategies. The best result in each column is highlighted in bold.

Aggregation mode	Params	GFLOPs	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
Bilinear	**15.88**	**182.3**	29.4	61.9	24.0	14.9	29.4	36.6	41.5
AvgPool	**15.88**	**182.3**	33.1	66.9	28.7	18.4	32.9	**38.8**	45.0
Strided Conv	15.95	187.5	32.8	66.7	27.9	18.1	32.8	38.0	44.7
Deform. Cross-Attn	15.93	188.2	**33.5**	**68.1**	**28.9**	**18.9**	**33.5**	38.7	**45.8**

Table 17: Foreground-to-background response analysis for different downsampling strategies in CSDF. ΔFBR denotes the increase from P_2 to P_2'. The best result in each column is highlighted in bold.

Aggregation	$\mu_{\text{in}}(P_2')$	$\mu_{\text{out}}(P_2')$	FBR(P_2')	FBR(P_2)	ΔFBR	AP_{vt}
Bilinear	0.49	0.33	1.48	1.44	0.04	14.9
AvgPool	0.45	**0.27**	1.67	1.45	0.22	18.4
Strided Conv	0.52	0.32	1.63	1.45	0.18	18.1
Deformable cross-attn	**0.57**	0.31	**1.84**	1.46	**0.38**	**18.9**

G.3 Downsampling strategy for semantically enhanced highest-resolution feature

Table 16 compares different strategies for adapting P_1 to the resolution of P_2 in CSDF. The results show that simple fixed-rule downsampling is not always effective. Bilinear interpolation severely degrades performance, suggesting that directly resizing high-resolution responses may blur tiny-object details and introduce background interference. AvgPool and strided convolution perform better, but they still remain inferior to deformable cross-attention on the main metrics.

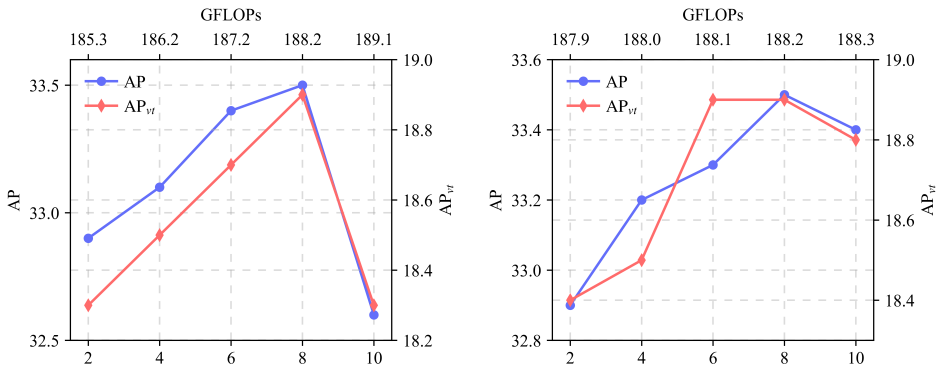
Deformable cross-attention achieves the best AP and AP_{vt}. This is because it allows P_2 to sparsely retrieve more relevant local responses from P_1. Rather than uniformly aggregating all nearby high-resolution signals, it can preserve more useful details for tiny object while reducing the redundant interferences of P_1 more effectively. Although AvgPool obtains a slightly higher AP_s, deformable cross-attention provides the best overall trade-off between accuracy and complexity, especially on very tiny objects. We therefore use deformable cross-attention as the default downsampling strategy for P_1 in CSDF.

To better understand the large performance gap between different downsampling strategies, we further measure their foreground-to-background response ratios. As shown in Table 17, bilinear interpolation increases FBR by only 0.04 over the original P_2, whereas deformable cross-attention increases it by 0.38 and achieves the highest AP_{vt}. AvgPool and strided convolution show intermediate improvements. These results indicate that the degradation of bilinear interpolation is associated with indiscriminate mixing of shallow high-resolution responses, while selective aggregation better concentrates the injected information on object regions.

G.4 Sampling points in CSDF and FSQI

We further study the influence of sampling-point numbers in CSDF and FSQI, as shown in Fig. 5. Both modules achieve the best performance with 8 sampling points, suggesting that sufficient sparse sampling is important for capturing fine-grained cues from the highest-resolution feature. However, further increasing the number of sampling points leads to noticeable drops in both AP and AP_{vt}, indicating that overly dense sampling may introduce redundant or noisy responses from high-resolution features.

The two modules also show different sensitivities to the sampling-point number. CSDF performs encoder-side detail injection from P_1 to P_2, so the sampled responses are directly fused into the shallow feature. As a result, inappropriate sampling may more easily introduce background responses and disturb the feature representation. In contrast, FSQI exposes P_1 to decoder queries through cross-attention, where each query can adaptively retrieve useful high-resolution responses together



(a) Sampling points in CSDF (b) Sampling points in FSQI

Figure 5: Sensitivity analysis of the sampling-point numbers in CSDF and FSQI. The bottom x-axis shows the number of sampling points, the top x-axis shows the corresponding GFLOPs. The left y-axis reports AP, while the right y-axis reports AP_{vt}. In the left subplot, we vary the CSDF points while fixing the FSQI points to 8. In the right subplot, we vary the FSQI points while fixing the CSDF points to 8.

Table 18: Sensitivity to the scale threshold $r_{\max}$ in EGSS. GT coverage denotes the percentage of valid ground-truth instances selected for supervising P_2' under the corresponding threshold. Tiny AP denotes AP_{vt}, AP_s, and AP_{eS} on AI-TOD-v2, VisDrone, and SODA-D, respectively. The adopted setting for each dataset is highlighted in bold.

Dataset	Baseline AP / Tiny AP	GT coverage @8 / @16 (%)	AP / Tiny AP @8	AP / Tiny AP @16	Default $r_{\max}$
AI-TOD-v2	32.1 / 17.5	98.17 / 100.00	**33.5 / 18.9**	33.4 / 18.7	**8**
VisDrone	35.9 / 28.3	64.14 / 90.28	37.1 / 29.8	**37.5 / 30.3**	**16**
SODA-D	35.4 / 17.6	73.69 / 100.00	36.6 / 19.3	**36.8 / 19.6**	**16**

with information from other feature levels. This query-driven selection makes FSQI relatively more stable under different sampling settings, although excessive sampling is still harmful. Therefore, we set the number of sampling points to 8 for both CSDF and FSQI by default.

G.5 Scale threshold in EGSS

We further analyze the scale threshold $r_{\max}$ used by EGSS. Since $r_{\max}$ determines which ground-truth objects are considered size-matched to the shallow feature P_2', its appropriate value depends on the object-scale distribution of each dataset.

Table 18 reports both the percentage of valid ground-truth instances covered by each threshold and the resulting detection performance. On AI-TOD-v2, $r_{\max} = 8$ already covers 98.17% of valid ground-truth instances and performs slightly better than $r_{\max} = 16$. In contrast, VisDrone and SODA-D contain broader object-scale distributions, for which increasing $r_{\max}$ from 8 to 16 raises the coverage from 64.14% to 90.28% and from 73.69% to 100%, respectively, with small corresponding performance improvements. Importantly, both settings consistently outperform their baselines on all three datasets. These results show that $r_{\max}$ mainly controls scale coverage rather than requiring sensitive model-specific tuning.

G.6 Loss design in EGSS

Table 19 studies the loss design of EGSS. The classification-only setting already achieves 33.2 AP, suggesting that foreground discrimination is essential for supervising the shallow feature. Adding a lightweight regression loss further improves AP to 33.5 and AP_{vt} from 18.1 to 18.9, indicating that coarse localization guidance is helpful for very tiny objects. However, adding centerness supervision causes a clear drop to 32.1 AP and 15.8 AP_{vt}. This suggests that centerness targets are unreliable on noisy shallow features, especially when tiny objects have only a few valid locations.

Table 19: Ablation study on the EGSS loss design. The centerness loss weight is set to 0.1 when used. The best result is highlighted in bold.

Setting	λ_{cls}	λ_{reg}	AP	AP_{50}	AP_{vt}	AP_t	AP_s
cls only	1.0	–	33.2	**68.2**	18.1	33.2	38.5
cls + reg	1.0	0.10	**33.5**	68.1	**18.9**	**33.5**	38.7
cls + reg + ctr	1.0	0.10	32.1	66.5	15.8	32.2	38.2
cls + reg	1.0	0.05	33.3	67.1	18.7	33.3	**39.2**
	1.0	0.20	33.1	67.2	17.6	33.7	38.6
	1.0	0.50	32.8	67.7	17.3	32.3	38.6
	0.5	0.10	32.6	67.0	16.0	32.4	38.2

Table 20: Sensitivity to the centerness loss weight in EGSS. All other settings are kept unchanged.

λ_{ctr}	AP	AP_{vt}
0	**33.5**	**18.9**
0.05	32.9	18.0
0.10	32.1	15.8

Table 21: Robustness of the full ReDHF model across three random seeds on AI-TOD-v2.

Run	AP	AP_{vt}
Original	33.52	18.88
Seed 2	33.44	18.74
Seed 3	33.50	18.82
Mean $\pm$ Std.	**33.49** $\pm$ **0.04**	**18.81** $\pm$ **0.07**

We further vary the classification-regression weights. A small regression weight works better than a strong one: increasing λ_{reg} from 0.10 to 0.20 or 0.50 reduces AP_{vt} from 18.9 to 17.6 and 17.3, respectively. Reducing the classification weight to 0.5 also degrades performance, showing that foreground classification should remain dominant in EGSS. Therefore, we use $\lambda_{\text{cls}} = 1.0$ and $\lambda_{\text{reg}} = 0.10$ by default, and omit centerness supervision.

To examine whether the degradation caused by centerness results from a particular loss weight, we vary λ_{ctr} from 0 to 0.10. As shown in Table 20, the performance decreases monotonically as the centerness weight increases, with AP_{vt} dropping from 18.9 without centerness to 18.0 and 15.8 at weights 0.05 and 0.10, respectively. We then analyze the sparsity of centerness targets across object scales as shown in Table 22. Very tiny objects have only 1.90 positive locations on average, and 93.90% of them contain at most one location with a centerness target no smaller than 0.5. In comparison, the number of valid locations increases rapidly with object size. This analysis explains why explicit centerness supervision becomes excessively sparse and unstable for very tiny objects, supporting its exclusion from EGSS.

G.7 Robustness across random seeds

To examine whether the reported gains depend on a favorable random initialization, we repeat the full ReDHF configuration using three random seeds. As shown in Table 21, ReDHF achieves 33.49 ± 0.04 AP and 18.81 ± 0.07 AP_{vt}. The small standard deviations indicate that the improvements of ReDHF are stable across different random seeds.

Table 22: Statistics of centerness targets across object scales on AI-TOD-v2. N_{pos} denotes the number of positive feature locations assigned to each ground-truth instance, and $N_{c\geq 0.5}$ denotes the number of positive locations whose centerness target is at least 0.5.

Scale	Num. GT	Avg. N_{pos}	$N_{\text{pos}} \leq 2$ (%)	Avg. $N_{c\geq 0.5}$	$N_{c\geq 0.5} \leq 1$ (%)
Very tiny	47317	1.90	78.26	0.54	93.90
Tiny	276865	6.44	23.69	1.79	33.76
Small	45612	19.19	3.18	5.21	1.57
Medium	6782	81.05	0.13	21.59	0.93

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: We provide a detailed and accurate description of the paper's contributions and scope in both the abstract and introduction.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: We discuss the limitations of ReDHF in Sec. 5 and Appendix F, including additional computational overhead, scale-dependent gains, and localized degradation for large objects in dense scenes.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] 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: The paper provide the network architectures and hyperparameters required for the experiments in Sec. 3 and Sec. 4.2.

   Guidelines:

   - The answer [N/A] 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: The code are provided in the supplemental material. The data is public available and have been widely used.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed descriptions of all the training and test details, including backbones, hyperparameters, and training recipes in Sec. 4.2.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: We report the mean and standard deviation across three random seeds for the full ReDHF model on AI-TOD-v2, as shown in Table 21.

   Guidelines:

   - The answer [N/A] 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).

- 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: We provide the hardware specifications required for the experiments.

   Guidelines:

   - The answer [N/A] 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 conform with the NeurIPS Code of Ethics

   Guidelines:

   - The answer [N/A] 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: [Yes]

    Justification: Our method significantly improves tiny object detection and can benefit applications such as remote sensing, traffic monitoring, and safety inspection.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our method exclusively utilizes open-source models and datasets, ensuring that there is no risk of misuse.

    Guidelines:

    - The answer [N/A] 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: Our method exclusively utilizes open-source models and datasets.

    Guidelines:

    - The answer [N/A] 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.

- 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: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This study does not involve any crowdsourced experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.