SEDEIM: SMALL-OBJECT ENHANCEMENT IN GENERAL OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Detecting small objects in general object detection remains challenging in crowded scenes and near larger instances, where weak instance-specific cues can be obscured by dominant surrounding responses. To investigate this issue, we analyze DEIMv2 and find that missed small objects can still contain query-selected feature locations, indicating that spatial coverage alone is insufficient for successful detection. This motivates SEDEIM, a small-object-enhanced detector including the following three key components. Spatial-Enhanced Adapter supplements vision-transformer features with CNN details and spatially modulates their contribution. Context Decoupled Pyramid provides bidirectional cross-scale interaction while preserving a separate local-detail path to retain distinctions from neighboring instances. Local Evidence Refinement reuses features sampled by deformable attention to construct a localization-oriented representation for box prediction. Results on COCO val2017 show that SEDEIM-m achieves 54.2 AP and 35.5 AP_S, outperforming SOTA methods by 1.2 AP and 1.3 AP_S.

1 INTRODUCTION

Recent advances in real-time object detectors, including DETR-based models Zhao et al. (2024); Peng et al. (2025); Huang et al. (2025b;a) and YOLO series Jocher et al. (2023); Jocher & Qiu (2024); Jocher et al. (2026), have substantially improved both accuracy and efficiency. However, small objects remain difficult to detect in general scenes, including in applications such as autonomous driving (Yang et al., 2022) and video surveillance (Xiao et al., 2023). While dedicated small-object detectors Sun et al. (2025); Shi et al. (2025); Liu et al. (2026); Hu et al. (2025) have been developed for specialized scenarios dominated by tiny instances (*e.g.* VisDrone Zhu et al. (2021a) and AITODv2 Xu et al. (2022)), improving small-object detection accuracy in general scenes is more challenging due to the broader range of object categories and larger scale variations.

As illustrated in Fig. 5, small objects are often densely distributed or located near or within larger objects, where stronger surrounding responses can obscure their weak cues, leading to missed detections and inaccurate localization. To investigate their detection failures, we first examine whether query selection covers their locations along the detection pipeline. Figure 1(a) shows that, at an IoU matching threshold of 0.50, 70.2% of missed small objects contain at least one token chosen for query selection. This statistic measuring spatial coverage does not imply that the corresponding query correctly represents or detects the object. Nevertheless, it shows that many failures persist despite spatial coverage, motivating further analyses of the selected representations and their subsequent refinement.

We examine three aspects of the detection process. First, under a feature distinction metric, missed small objects are less distinct from their surroundings than detected objects before the encoder, and their distinction decreases further after encoding (Figure 1(b)). Second, perturbing a nearby region of a small object changes its features almost as much as perturbing the object itself (Figure 1(c)). This suggests that the representation is sensitive to nearby content as well as to the object's own details. Third, for spatially covered objects, we track the same selected candidate from initialization through decoder refinement. Candidates associated with missed objects exhibit only limited improvement in box IoU (Figure 1(d)). Together, these observations motivate preserving instance-specific information and making it more useful for localization, while also ensuring spatial coverage.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

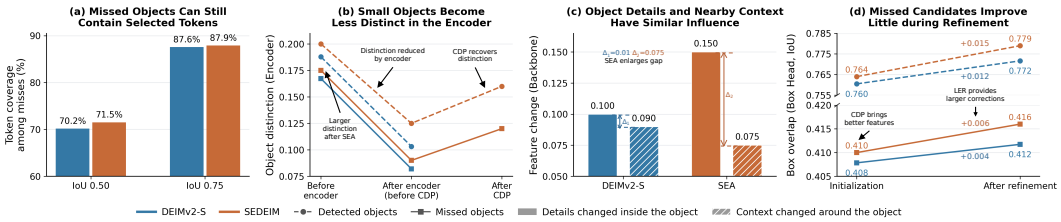


Figure 1: **Spatial coverage alone does not ensure accurate small-object detection.** (a) Many missed small objects contain source-token locations selected for query initialization. (b) Small objects become less distinguishable from their surroundings after the encoder; SEA and CDP improve distinction at their respective stages. (c) Object details and nearby context exhibit similar sensitivity to matched perturbations. SEA increases target contribution and reduces neighborhood sensitivity under this protocol. (d) Tracking the same selected candidate reveals limited IoU improvement for missed objects. CDP improves initialization IoU, while LER increases the IoU gain during refinement in the evaluated configurations. Appendix E provides the metric definitions and evaluation protocols.

Motivated by these observations, we propose **S**mall-object-**E**nhanced DEIM (SEDEIM), which strengthens small-object representations across feature adaptation, cross-scale interaction, and localization refinement. Spatial-Enhanced Adapter (SEA) adaptively regulates the contribution of spatial details during early feature fusion, providing more suitable local information for small-object representation. Context Decoupled Pyramid (CDP) introduces local differences during cross-scale interaction, aiming to preserve instance-level distinctions while incorporating multi-scale context. Localization Enhancement Refinement (LER) module reuses the multi-scale information collected by the decoder and selectively adjusts its contribution to the localization features, helping candidates obtain more suitable information for box regression.

The comparisons in Figure 1 are consistent with these design objectives, showing improved feature separation, greater relative sensitivity to target details, and better candidate localization in the evaluated configurations. On COCO val2017, SEDEIM-M achieves 54.2 AP and 35.5 AP_S, improving upon DEIMv2-M by 1.2 AP and 1.3 AP_S, respectively, while also improving the reported medium- and large-object accuracy.

We summarize our contributions as follows:

- We analyze small-object detection failures beyond spatial coverage in query selection, examining target–neighborhood feature separation, sensitivity to local details, and candidate localization through decoder refinement.
- We introduce SEDEIM, which combines adaptive spatial-detail fusion, cross-scale interaction with an explicit local-contrast path, and localization-specific reuse of existing deformable-attention samples
- SEDEIM achieves state-of-the-art performance on COCO. In particular, SEDEIM-M obtains 54.2 AP and 35.5 AP_S, outperforming DEIMv2-M by 1.2 AP and 1.3 AP_S.

2 Related Work

2.1 DETR-based Real-time Object Detection

Despite the strong performance of DETR-based object detectors Carion et al. (2020); Zhu et al. (2021b); Li et al. (2022); Yao et al. (2021); Zhang et al. (2023), their high computational cost and inference latency limit their applications in real-time detection. RT-DETR Zhao et al. (2024) breaks through this efficiency bottleneck with a hybrid encoder that decouples intra-scale interaction and cross-scale fusion. D-FINE Peng et al. (2025) improves localization by iteratively refining fine-grained bounding-box distributions. DEIM Huang et al. (2025b) accelerates convergence by combining Dense One-to-One matching. DEIMv2 Huang et al. (2025a) further incorporates DINOv3 ViT Siméoni et al. (2025) through a Spatial Tuning Adapter to supplement CNN-derived spatial

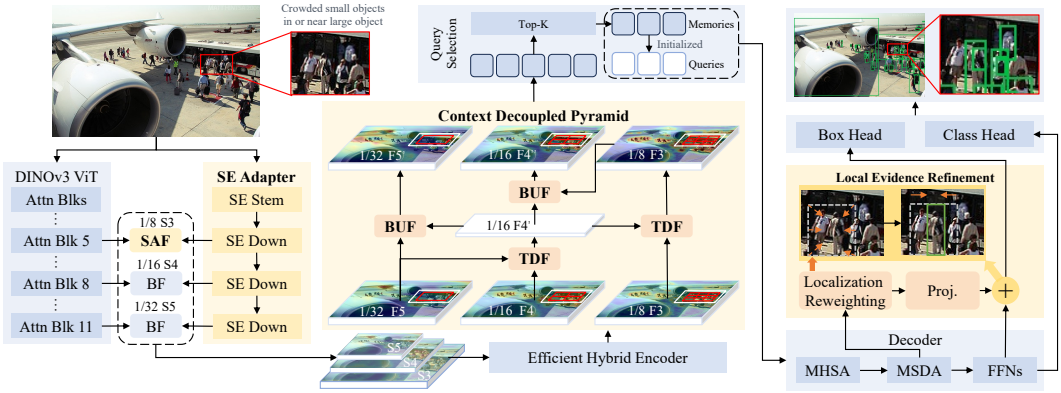


Figure 2: **Overview architecture of SEDEIM.** SEDEIM contains three modules: (1) Spatial-Enhanced Adapter (SEA), which generates spatial-enhanced CNN features and fuses with semantic features from DINOv3 ViT via Spatial-adaptive fusion (SAF); (2) Context Decoupled Pyramid (CDP), which separates cross-scale context from local differences to preserve detail during feature fusion; (3) Local Evidence Refinement (LER), which provides the box regression branch with complementary localization cues derived from reweighted multi-scale attention evidence.

details. Our work focuses on enhancing small-object performance in general object detection by preserving the distinctions between small objects and nearby larger or dense instances.

2.2 Small Object Detection

Small object detection is commonly studied on specialized aerial benchmarks such as VisDrone Zhu et al. (2021a) and AITODv2 Xu et al. (2022), which contain many densely distributed tiny instances. DQ-DETR Huang et al. (2024) uses predicted category counts and density maps to dynamically adjust the number and spatial positions of object queries. Dome-DETR Hu et al. (2025) performs density-oriented feature and query manipulation to focus computation and query allocation on foreground regions. FSDC-DETR Liu et al. (2026) constructs collaborative frequency and spatial representations to preserve fine object information during feature fusion and resolution changes. Although effective, these methods primarily target scenes dominated by tiny objects. TinyFormer Hsieh et al. (2026) extends small-object preservation to general scenes through spatial adaptation and bidirectional fusion, but its gains are less consistent across broader object scales. In contrast, SEDEIM targets small-object limitations in general scenes and improves small-object accuracy while maintaining robust performance across the full range of object scales.

3 Methods

3.1 Pipeline

Figure 2 illustrates the overall architecture of SEDEIM. Specifically, an input image is processed by both the DINOv3 ViT and a lightweight Spatial-Enhanced Adapter (SEA) to extract semantic and spatial features, respectively. The extracted S_3 features are then fused through the Spatial-adaptive Fusion (SAF) module to adaptively regulate spatial-detail contributions, while $\{S_4, S_5\}$ retain the standard Bi-fusion. These multi-scale features are then fed into the Efficient Hybrid Encoder, followed by the Context Decoupled Pyramid (CDP) module to preserve instance-level distinctions. Subsequently, the query selection is implemented to select the most informative tokens, which are then used to initialize the object queries for the decoder. Queries are then refined in the decoder, and the Local Evidence Refinement (LER) module utilizes localization evidence derived from Deformable Attention to enhance the box regression branch, while the class prediction branch uses original refined queries from the decoder.

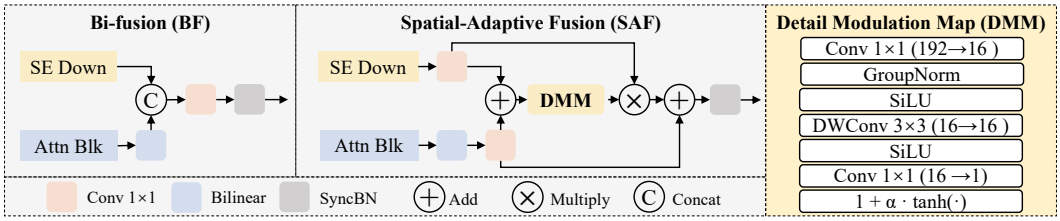


Figure 3: **Feature fusion modules in Spatial-Enhanced Adapter.** Bi-Fusion (BF) directly combines aligned semantic and detail features. Spatial-Adaptive Fusion (SAF) instead adaptively modulates the detail contribution using a Detail Modulation Map (DMM). The DMM generator inside SAF produces a single-channel spatial modulation weight to regulate the detail branch. SAF is applied at level S_3, while S_4 and S_5 retain Bi-Fusion.

3.2 Spatial-enhanced Adapter

Small objects near larger or densely distributed instances require spatial details that distinguish them from their surroundings. To strengthen this evidence when adapting DINOv3 features for detection, we introduce the Spatial-Enhanced Adapter (SAE), which extends the Spatial Tuning Adapter (STA) in DEIMv2 Huang et al. (2025a). SAE retains STA's multi-scale structure and adds a lightweight CNN-based block after each downsampling stage to further extract local spatial patterns, detailed in Appendix C. The CNN detail features are fused with the corresponding DINOv3 semantic features to produce S_3, S_4, and S_5 at strides 8, 16, and 32. We use Spatial-Adaptive Fusion for S_3 and retain the original Bi-Fusion for S_4 and S_5.

Bi-Fusion. For the S_4 and S_5 level features, we adopt the Bi-Fusion module from DEIMv2 Huang et al. (2025a), as illustrated in Figure 3. The semantic feature S from the corresponding DINOv3 block is bilinearly resized to the resolution of the CNN detail feature D, yielding S_{bl}. The aligned features are concatenated along the channel dimension and processed by a 1×1 convolution followed by synchronized batch normalization (SyncBN):

$$S_{\text{BF}} = \text{SyncBN}\left(\text{Conv}_{1\times1}([S_{bl}; D])\right).$$

This design is straightforward and efficient, but its spatially shared projection does not explicitly adjust the contribution of CNN details to the surrounding context at each location. Consequently, stronger responses from neighboring instances can overwhelm weak small-object cues during fusion, making small objects harder to distinguish from their surroundings.

Spatial-Adaptive Fusion To retain useful small-object details in the presence of dominant neighboring responses, Spatial-Adaptive Fusion (SAF) modulates the CNN contribution using the joint semantic and detail context, as illustrated in Figure 3. We apply SAF to S_3 level features, whose higher resolution provides finer spatial support for distinguishing small objects from nearby instances. To expose the two branch contributions while retaining the original Bi-Fusion projection, we decompose its convolution into two projections:

$$S_p = \text{Conv}^{S}_{1\times1}(S_{bl}), \quad D_p = \text{Conv}^{D}_{1\times1}(D).$$

The two kernels correspond to the semantic and detail channel slices of the original convolution. Their sum $F_p = S_p + D_p$ therefore equals the original convolution applied to $[S_{bl}; D]$, as derived in Appendix C. This decomposition allows the CNN contribution to be modulated independently before normalization. We then predict a single-channel Detail Modulation Map (DMM) from F_p:

$$M_{\text{DM}}(F_p) = \tanh(\mathcal{G}(F_p)),$$

and use the bounded learnable scalar α to control its modulation strength. The map generator $\mathcal{G}$ first reduces the channel dimension through a 1×1 convolution followed by GroupNorm and SiLU. A depthwise 3×3 convolution with SiLU then incorporates neighboring responses, and a final 1×1 convolution predicts one modulation value per spatial location. Finally, the modulated detail and semantic projections are added and normalized:

$$S_{\text{SAF}} = \text{SyncBN}\left(\text{Conv}^{S}_{1\times1}(S_{bl}) + (1 + \alpha M_{\text{DM}}) \odot \text{Conv}^{D}_{1\times1}(D)\right).$$

SAF thus provides a spatially adaptive adjustment of detail contributions intended to preserve small-object cues against stronger neighboring responses, and recovers the original Bi-Fusion when $\alpha = 0$.

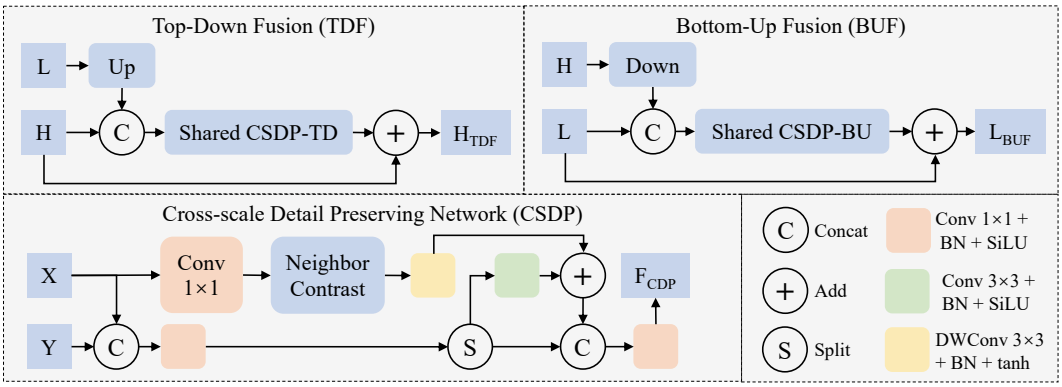


Figure 4: **Bidirectional cross-scale detail preservation in CDP.** CDP performs cross-scale interaction through Top-Down Fusion (TDF) and Bottom-Up Fusion (BUF), where features from adjacent levels are aligned and fused in both directions. The internal Cross-scale Detail Preserving Network (CSDP) retains the main fusion path while extracting neighbor contrast from the updated feature and injecting it as a residual detail cue. The TDF and BUF branches use their corresponding shared CSDP modules to preserve local distinctions during multi-scale context propagation.

3.3 Context Decoupled Pyramid

After Spatial-Adaptive Fusion and Bi-Fusion, the resulting multi-scale features are fed into the Efficient Hybrid Encoder from RT-DETR Zhao et al. (2024). Although the Efficient Hybrid Encoder constructs a unified multi-scale representation, its repeated cross-scale aggregation reduces the local separability of small-object representations, making their local evidence increasingly indistinguishable from dominant responses of nearby large or crowded instances.

To address this limitation, we introduce Context-Decoupled Pyramid (CDP) to preserve the evidence of each small-object instance, as illustrated in Figure 2. Specifically, CDP performs context-decoupled interaction through a four-stage schedule: (1) Top-Down Fusion (TDF) combines F_4 with F_5 to obtain the context-enriched intermediate feature F_4'; (2) Bottom-Up Fusion (BUF) feeds the updated F_4' back to F_5, yielding F_5' with complementary spatial information; (3) TDF then uses F_4' as an intermediate bridge to supply F_3 with semantic support and obtain F_3'; and (4) BUF propagates the fine-grained localization evidence preserved in F_3' back to F_4', producing F_4''.

In each top-down stage, TDF, illustrated in Figure 4, upsamples the low-resolution feature L and concatenates it with the high-resolution feature H, then applies a Cross-scale Detail-Preserving Network (CSDP), supplying fine-scale representations with semantic support from the adjacent low-resolution feature. The two TDF instances share CSDP-TD convolutional weights to reduce parameter overhead, while normalization layers and residual coefficients remain specific to each stage. BUF, also illustrated in Figure 4, performs the reverse operation and sharing strategy, thereby feeding localization-sensitive evidence back to the low-resolution feature.

Cross-scale Detail-Preserving Network. Although the four-stage schedule provides bidirectional contextual support, it does not explicitly preserve the local distinctions that small objects need to remain separable from nearby large or densely distributed instances. CSDP addresses this limitation by retaining local differences from the surrounding region alongside cross-scale context aggregation. As shown in Figure 4, it combines a path that fuses the two aligned inputs with a path that extracts neighborhood contrast from the feature being updated. Let X denote the feature being updated and Y the spatially aligned feature from the adjacent level, both with C channels. In TDF, $X = H$ and $Y = \mathrm{Up}(L)$; while in BUF, $X = L$ and $Y = \mathrm{Down}(H)$. The lower path fuses the aligned inputs:

$$P = \mathrm{SiLU}(\mathrm{BN}(\mathrm{Conv}_{1\times1}([X;Y]))) .$$

The Split operation divides P into two $C/2$-channel features B and V. We retain B unchanged and transform V as:

$$S = \mathrm{SiLU}(\mathrm{BN}(\mathrm{Conv}_{3\times3}(V))) .$$

Processing only half the channels reduces computation while retaining a direct path for part of the fused representation. To preserve local cues from X, we project X alone to $C/2$ channels as $R_X = \mathrm{Conv}^{\mathrm{loc}}_{1\times1}(X)$. The *Neighbor Contrast* operation then subtracts the local mean:

$$C_X = R_X - \mathrm{AvgPool}_{3\times3}(R_X).$$

For small objects surrounded by stronger responses, this operation emphasizes differences from the immediate neighborhood rather than absolute activation strength. Moreover, this contrast is transformed before being added to the spatial fusion feature:

$$L_X = \tanh(\mathrm{BN}(\mathrm{DWConv}_{3\times3}(C_X))).$$

The depthwise convolution refines neighborhood differences independently in each channel, helping retain the boundaries and fine structures that distinguish small objects from nearby instances. The local correction is added to the spatial fusion feature as $\widehat{S} = S + L_X$. The retained feature B is concatenated with the corrected feature $\widehat{S}$ and projected to produce:

$$F_{\mathrm{CDP}} = \mathrm{SiLU}\Big(\mathrm{BN}\Big(\mathrm{Conv}_{1\times1}([B;\widehat{S}])\Big)\Big).$$

Thus, the cross-scale fusion branch supplies contextual information for recognition, while the Neighbor Contrast branch preserves local differences that help distinguish small objects from surrounding instances. Together, these branches allow CDP to propagate multi-scale context while preserving the local distinctions between small objects and their surroundings.

3.4 Local Evidence Refinement

Although CDP preserves local distinctions during feature exchange, accurately localizing a small object also requires the decoder to distinguish its spatial extent from nearby larger or densely distributed instances. To favor a box that fits the small instance rather than encompasses neighboring responses, as illustrated in the Figure 2, we introduce Local Evidence Refinement (LER) to supplement the box-prediction representation with local information from Multi-Scale Deformable Attention (MSDA). LER applies this correction to the localization features while retaining the original query representation for classification.

Evidence Extraction from MSDA. MSDA predicts offsets and attention weights to sample and aggregate features across pyramid levels. Before cross-scale aggregation, LER summarizes the sampled features within each level $k \in \{3,4,5\}$ as Z_k using within-level normalized attention weights. To retain the original reliance on each scale, it also sums the attention weights assigned to that level and averages over heads to obtain m_k. These quantities are derived from existing MSDA samples without additional sampling.

Localization Reweighting. To supplement the spatial support for a small instance without discarding its existing context, LER computes alternative scale weights $\boldsymbol{\pi}$ from reference-box scale compatibility and native attention support, with query similarity and sampling concentration further guiding the allocation. Rather than injecting another full context vector, LER subtracts the native-weighted aggregation from the localization-weighted aggregation of the same features:

$$E_{\mathrm{loc}} = \sum_k \pi_k Z_k - \sum_k m_k Z_k = \sum_k (\pi_k - m_k) Z_k.$$

Thus, E_{loc} captures a change in feature aggregation rather than a box-coordinate displacement, and vanishes when the two scale allocations coincide. Using this difference avoids adding a second complete context vector when supplementing the spatial support for a small instance. A linear-layer-based residual projector transforms E_{loc}, followed by magnitude normalization and confidence- and size-dependent scaling to obtain Δq_{loc}. The Box Head receives $q_{\mathrm{loc}} = q + \Delta q_{\mathrm{loc}}$ from the FFN output q, while the Class Head retains q. Detailed definitions of the scale statistics, localization weights, and bounded residual are provided in Appendix D.

Table 1: **State-of-the-art comparisons with real-time object detectors on COCO val2017.** SEDEIM substantially improves small-object accuracy while maintaining robust performance on medium and large objects.

Method	Params (M)	FLOPs (G)	AP	AP_S	AP_{50}	AP_{75}	AP_M	AP_L
YOLO26-S Jocher et al. (2026)	9.5	20.9	48.1	22.1	65.0	52.3	46.1	65.1
RT-DETRv2-S Lv et al. (2024)	21.2	61.1	48.1	30.2	65.1	52.1	51.2	64.2
DFINE-S Peng et al. (2025)	10.2	25.2	48.5	29.3	65.4	52.6	52.2	65.4
DEIM-S Huang et al. (2025b)	10.2	25.2	49.0	30.4	65.9	53.1	52.6	65.6
DEIMv2-S Huang et al. (2025a)	9.7	25.7	50.9	31.4	68.3	55.1	55.3	70.3
SEDEIM-S (Ours)	10.3	30.9	51.8	33.1	68.7	56.2	56.0	70.4
YOLO26-M Jocher et al. (2026)	20.4	68.4	52.6	28.8	69.9	57.1	51.2	68.5
RT-DETRv2-M Lv et al. (2024)	36.6	100.4	51.9	33.5	69.9	56.5	56.8	69.2
DFINE-M Peng et al. (2025)	19.3	56.7	52.3	33.2	69.9	56.4	56.5	70.1
DEIM-M Huang et al. (2025b)	19.3	56.7	52.7	35.2	70.0	57.3	56.7	69.4
DEIMv2-M Huang et al. (2025a)	18.1	52.3	53.0	34.2	70.2	57.6	57.4	71.5
SEDEIM-M (Ours)	20.2	67.9	54.2	35.5	71.1	59.1	58.2	72.4
YOLO26-L Jocher et al. (2026)	24.8	86.8	54.5	30.7	71.8	59.5	53.2	70.3
RT-DETRv2-L Lv et al. (2024)	42.9	137.3	53.4	36.0	71.6	57.5	57.8	70.8
DFINE-L Peng et al. (2025)	30.7	91.0	54.0	36.5	71.6	58.4	58.0	71.9
DEIM-L Huang et al. (2025b)	30.8	91.1	54.7	36.9	72.4	59.4	59.7	71.8
DEIMv2-L Huang et al. (2025a)	32.2	96.8	56.0	37.6	73.5	61.1	60.9	74.9
SEDEIM-L (Ours)	34.8	113.9	56.9	39.0	74.3	62.0	61.6	75.2
YOLO26-X Jocher et al. (2026)	55.7	194.4	56.9	34.1	74.2	62.2	55.9	72.8
RT-DETRv2-X Lv et al. (2024)	76.5	260.2	54.3	35.8	72.8	58.8	58.8	72.1
DFINE-X Peng et al. (2025)	61.6	202.5	55.8	37.3	73.7	60.2	60.5	73.5
DEIM-X Huang et al. (2025b)	61.7	202.7	56.5	38.8	74.0	61.6	61.4	74.2
DEIMv2-X Huang et al. (2025a)	50.3	151.7	57.8	39.1	75.3	63.2	62.9	75.9
SEDEIM-X (Ours)	55.5	183.2	58.4	39.9	75.7	64.0	63.4	76.3

4 Experiments

4.1 Implementation Details

We train SEDEIM in multiple scales (S, M, L, X) on COCO 2017 Lin et al. (2014). All experiments are conducted on 4 NVIDIA Tesla V100 GPUs. For each model scale, we follow the corresponding DEIMv2 Huang et al. (2025a) configuration for all training hyperparameters and base-model settings, with architectural changes restricted to our proposed modules. We provide detailed training settings and model configurations in Appendix B. We evaluate SEDEIM on COCO Val 2017 and report the standard COCO bounding-box AP averaged over IoU thresholds from 0.50 to 0.95, together with AP_{50}, AP_{75}, and scale-specific AP_S/AP_M/AP_L for small, medium, and large objects, respectively.

4.2 State-of-the-art Comparison with the Real-Time Detectors

We compare the SEDEIM series with recent real-time object detectors in Table 1. SEDEIM-S achieves 51.8 AP with 10.3M parameters and 30.9G FLOPs. Compared with DEIMv2-S, it improves AP by 0.9 and AP_S by 1.7, while AP_M and AP_L also increase by 0.7 and 0.1, respectively. SEDEIM-M achieves 54.2 AP and 35.5 AP_S, outperforming DEIMv2-M by 1.2 AP and 1.3 AP_S. Its AP_M and AP_L further improve by 0.8 and 0.9, respectively. SEDEIM-L and SEDEIM-X achieve similar gains in detection accuracy. All the results show that the substantial gain on small objects is achieved while maintaining robust performance on larger objects.

4.3 Ablation Studies

We conduct ablation studies to evaluate the contributions of each module in SEDEIM. All ablation experiments are performed on the SEDEIM-S model.

Table 2: **Ablations on SEDEIM components.** SEDEIM improves small object performance in general object detection through three complementary modules: SEA, CDP, and LER. Numbers in parentheses denote the gain compared to the baseline (first row).

SEA	CDP	LER	Params (M)	FLOPs (G)	AP	AP_S	AP_M	AP_L
–	–	–	9.7	25.7	50.9	31.3	55.3	70.1
✓	–	–	9.8	26.0	51.4 (+0.5)	32.7 (+1.4)	55.3	70.3
–	✓	–	10.2	30.5	51.4 (+0.5)	32.5 (+1.2)	55.6	70.4
–	–	✓	9.8	25.7	51.0 (+0.1)	32.0 (+0.7)	55.1	70.4
✓	✓	–	10.3	30.9	51.7 (+0.8)	32.0 (+0.7)	56.0	70.1
✓	–	✓	9.8	26.0	51.5 (+0.6)	32.4 (+1.1)	55.9	70.1
–	✓	✓	10.3	30.5	51.6 (+0.7)	32.6 (+1.3)	55.7	70.0
✓	✓	✓	10.3	30.9	**51.8** (+0.9)	**33.1** (+1.8)	**56.0**	**70.4**

Table 3: **Ablations for SEA and SAF Scope.** Numbers in parentheses denote gains over the baseline (first row). Blue-shaded row indicate our proposed configurations.

Variant	Params (M)	FLOPs (G)	AP	AP_S	AP_M	AP_L
Baseline	9.7	25.7	50.9	31.3	55.3	70.1
w/ SEA, w/o SAF	9.8	26.0	51.2 (+0.3)	32.4 (+1.1)	55.3	70.3
SAF Scope						
F_3	9.8	26.0	**51.4** (+0.5)	**32.7** (+1.4)	55.3	70.3
$F_3 + F_4$	9.8	26.0	51.2 (+0.3)	31.8 (+0.5)	55.3	69.6
$F_3 + F_4 + F_5$	9.8	26.1	51.4 (+0.5)	32.0 (+0.7)	**55.5**	**70.4**

Ablations on SEDEIM components. Table 2 shows the contributions of each component in SEDEIM. Individually, SEA and CDP both improve AP by 0.5, while increasing AP_S by 1.4 and 1.2, respectively. LER provides gains of 0.1 AP and 0.7 AP_S, showing that refining decoder localization provides a complementary benefit to feature enhancement and cross-scale interaction. Pairwise combinations further raise AP by 0.6–0.8 and AP_S by 0.7–1.3, demonstrating that the three modules improve different stages of the detection pipeline rather than acting as redundant alternatives. Combining SEA, CDP, and LER achieves the best overall performance of 51.8 AP and 33.1 AP_S, outperforming DEIMv2-S by 0.9 AP and 1.8 AP_S, while reaching 56.0 AP_M and 70.4 AP_L with 10.3M parameters and 30.9G FLOPs.

Ablations for SEA and SAF Scope. Table 3 analyzes the effect of SEA and different scopes for SAF. Introducing SEA without SAF already improves the baseline by 0.3 AP and 1.6 AP_S, confirming the benefit of supplementing the ViT representation with convolutional spatial details. Applying SAF only to F_3 further increases AP from 51.2 to 51.4 while retaining strong small-object performance. Notably, the F_3-only and $F_3 + F_4 + F_5$ variants achieve the same 51.4 AP, but the former obtains 0.7 higher AP_S, while remaining within 0.2 AP_M and 0.1 AP_L of the all-level variant and requiring slightly fewer FLOPs. We therefore apply SAF only to F_3, where its spatially adaptive detail modulation is most beneficial to small-object representation.

Ablations on component and interaction for CDP. Table 4 evaluates the impact on the components and interactions in CDP. Introducing TDF and BUF without CSDP improves the baseline by 0.4 AP and 0.5 AP_S, showing that bidirectional exchange between adjacent pyramid levels already benefits multi-scale representation. Equipping these interactions with CSDP further increases the gains to 0.5 AP and 1.2 AP_S, while also improving AP_M and AP_L, which verifies the value of preserving local instance distinctions during cross-scale propagation. The interaction ablation further shows that either P_3–P_4 or P_4–P_5 alone provides an incomplete scale pathway: the former favors small objects but reduces overall and large-object accuracy, whereas the latter improves medium-object accuracy without benefiting small objects. Combining both interactions alleviates these scale-specific limitations. In particular, performing P_4–P_5 interaction before P_3–P_4 achieves the best result of 51.4 AP and 32.5 AP_S, supporting our deep-to-shallow design that first establishes contextual support at coarser levels and then propagates it to the high-resolution representation.

Table 4: **Ablations on component and interaction for CDP.** Numbers in parentheses denote gains over the baseline (first row). Blue-shaded rows indicate our proposed configurations.

Variant	Params (M)	FLOPs (G)	AP	AP_S	AP_M	AP_L
Component Ablation						
Baseline	9.7	25.7	50.9	31.3	55.3	70.1
w/ TDF/BUF, w/o CSDP	10.2	30.5	51.3 (+0.4)	31.8 (+0.5)	55.4	69.9
w/ TDF/BUF, w/ CSDP	10.2	30.5	**51.4** (+0.5)	**32.5** (+1.2)	**55.6**	**70.4**
Interaction Ablation						
P_3–P_4 only	10.2	29.6	50.8 (-0.1)	32.3 (+1.0)	55.1	69.3
P_4–P_5 only	10.2	26.6	50.9 (+0.0)	31.1 (-0.2)	55.7	69.8
P_3–P_4 then P_4–P_5	10.2	30.5	51.3 (+0.4)	31.9 (+0.6)	55.2	70.2
P_4–P_5 then P_3–P_4	10.2	30.5	**51.4** (+0.5)	**32.5** (+1.2)	**55.6**	**70.4**

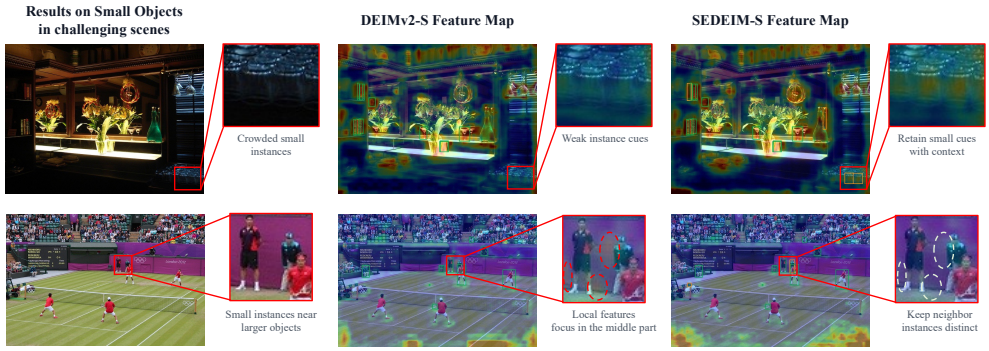


Figure 5: **Comparison of small-object detections and feature responses between DEIMv2-S and SEDEIM-S.** Green boxes indicate small objects detected by both models, while orange boxes indicate those additionally detected by SEDEIM-S. The middle and right columns visualize the respective F_3 features before query selection, with enlarged views of the highlighted regions.

4.4 Visualization Results of SEDEIM

Figure 5 presents qualitative comparisons between SEDEIM-S and DEIMv2-S. We consider two challenging spatial arrangements of small objects. In the crowded scene shown in the top row, the baseline provides weak and less localized responses for the clustered small instances, whereas SEDEIM-S retains clearer responses at their individual locations while preserving the surrounding context. In the bottom row, small instances appear close to larger objects, and the baseline response is concentrated around the dominant central region. SEDEIM-S instead maintains more distinct local responses for the neighboring small instances, which is consistent with the additional detections marked in orange. These results show that SEDEIM preserves small-object cues under both crowded context and interference from nearby larger objects, providing more discriminative F_3 features for subsequent query selection.

5 Conclusion

In this work, we investigate the factors that prevent selected feature locations from producing accurate small-object detections. Our analysis identifies weak instance distinctions and insufficient localization refinement in crowded regions and near larger instances, which motivates SEDEIM. Specifically, the Spatial-Enhanced Adapter regulates spatial detail contributions during early feature fusion, the Context Decoupled Pyramid preserves instance-level distinctions during cross-scale interaction, and Local Evidence Refinement reuses information sampled by deformable attention to improve box refinement. Qualitative and quantitative evaluations on COCO val2017 demonstrate that our approach performs favorably against recent real-time object detectors.

6 AI USE STATEMENT

Generative AI tools were used to improve the readability and linguistic clarity of the manuscript and, to a limited extent, to assist with code development, including method implementation. All AI-assisted text was reviewed and revised by the authors, and AI-assisted code was manually inspected and tested for correctness. Generative AI tools were not used to generate research hypotheses, design the research methodology or experiments, formulate mathematical claims, or interpret experimental results. The authors take full responsibility for the final content of this work, including all text, code, claims, and results.

REFERENCES

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*, pp. 213–229. Springer, 2020.

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

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

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*, pp. 770–778, 2016.

Jun-Wei Hsieh, Meng-Yu Kao, Ghufron Wahyu Kurniawan, and Kuan-Chuan Peng. Tinyformer: Preserving tiny objects in yolo-detr hybrid real-time detectors. *arXiv preprint arXiv:2605.25046*, 2026.

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. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 101–110, 2025.

Shihua Huang, Yongjie Hou, Longfei Liu, Xuanlong Yu, and Xi Shen. Real-time object detection meets dinov3. *arXiv preprint arXiv:2509.20787*, 2025a.

Shihua Huang, Zhichao Lu, Xiaodong Cun, Yongjun Yu, Xiao Zhou, and Xi Shen. Deim: Detr with improved matching for fast convergence. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15162–15171. IEEE, 2025b.

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*, pp. 290–305. Springer, 2024.

Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. URL `https://github.com/ultralytics/ultralytics`.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL `https://github.com/ultralytics/ultralytics`.

Glenn Jocher, Jing Qiu, Mengyu Liu, Shuai Lyu, Fatih Cagatay Akyon, and Muhammet Esat Kalfaoglu. Ultralytics yolo26: unified real-time end-to-end vision models. *arXiv preprint arXiv:2606.03748*, 2026.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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*, pp. 13619–13627, 2022.

Xin Li, Yuqing Huang, Zhenyu He, Yaowei Wang, Huchuan Lu, and Ming-Hsuan Yang. Citetracker: Correlating image and text for visual tracking. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9940–9949. IEEE, 2023.

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*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 936–944. Ieee, 2017.

Aiwen Liu, Chengguang Zhu, Gang Wang, Dandan Zhu, Haodong Lin, Yan Wang, Huiyu Zhou, and Zhengyi Pan. Fsdc-detr: A frequency-spatial domain collaborative detr for small object detection. In *European Conference on Computer Vision*, 2026.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 8759–8768. Ieee, 2018.

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

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer, 2024. URL https://arxiv.org/abs/2407.17140.

Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 8834–8844. IEEE, 2022.

Yansong Peng, Hebei Li, Peixi Wu, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. D-FINE: Redefine regression task of DETRs as fine-grained distribution refinement. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=MFZjrTFE7h.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Zican Shi, Jing Hu, Jie Ren, Hengkang Ye, Xuyang Yuan, Yan Ouyang, Jia He, Bo Ji, and Junyu Guo. Hs-fpn: High frequency and spatial perception fpn for tiny object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 6896–6904, 2025.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Huixin Sun, Runqi Wang, Yanjing Li, Linlin Yang, Shaohui Lin, Xianbin Cao, and Baochang Zhang. Set: Spectral enhancement for tiny object detection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4713–4723. IEEE, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jinsheng Xiao, Yuanxu Wu, Yunhua Chen, Shurui Wang, Zhongyuan Wang, and Jiayi Ma. Lstfe-net: Long short-term feature enhancement network for video small object detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14613–14622. IEEE, 2023.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

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. *ISPRS Journal of Photogrammetry and Remote Sensing*, 190:79–93, 2022.

Jinrong Yang, Songtao Liu, Zeming Li, Xiaoping Li, and Jian Sun. Real-time object detection for streaming perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5385–5395, 2022.

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

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3mRwyG5one.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pp. 1–21. Springer, 2022.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16965–16974. IEEE, 2024.

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, 2021a.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=gZ9hCDWe6ke.

A Deep Learning-based Object Detection

Computer vision has witnessed remarkable progress across a wide range of tasks, including image classification Krizhevsky et al. (2012); Liu et al. (2021; 2022), object tracking Li et al. (2023); Zhang et al. (2022); Meinhardt et al. (2022), semantic segmentation Ronneberger et al. (2015); Xie et al. (2021); Kirillov et al. (2023), and object detection. Among them, object detection serves as a fundamental task that aims to simultaneously recognize and localize objects in complex visual scenes.

Early deep learning-based detectors were dominated by the R-CNN family. R-CNN Girshick et al. (2014) introduced convolutional neural networks into region-based object detection, while Fast R-CNN Girshick (2015) improved computational efficiency by sharing convolutional features among region proposals. Faster R-CNN Ren et al. (2015) further incorporated a Region Proposal Network (RPN), establishing the representative two-stage detection paradigm. To improve the detection of objects at different scales, FPN Lin et al. (2017) introduced a top-down feature pyramid with lateral connections to aggregate multi-level features, while PAN Liu et al. (2018) further enhanced feature propagation through additional bottom-up pathways. In parallel, one-stage detectors, particularly the YOLO series Jocher et al. (2023); Jocher & Qiu (2024); Jocher et al. (2026), progressively improved the trade-off between detection accuracy and inference efficiency, making real-time object detection increasingly practical.

More recently, DETR Carion et al. (2020) reformulated object detection as a set prediction problem and introduced Transformers Vaswani et al. (2017) into the detection framework, eliminating many hand-crafted components such as anchor generation and non-maximum suppression. Subsequent DETR-based Zhu et al. (2021b); Li et al. (2022); Yao et al. (2021); Zhang et al. (2023) methods have continuously improved its convergence, computational efficiency, and multi-scale representation capability, establishing Transformer-based detectors as another major paradigm for modern object detection.

Table 5: Inherited DEIMv2 settings for the S, M, L, and X variants. The three detection features have strides 8, 16, and 32.

Setting	S	M	L	X
Training Settings				
Input resolution	640^2	640^2	640^2	640^2
Total batch size	32	32	32	32
Optimizer	AdamW	AdamW	AdamW	AdamW
Base LR	5e−4	5e−4	5e−4	5e−4
Backbone LR	2.5e−5	2.5e−5	1.25e−5	1e−5
Weight decay	1e−4	1e−4	1.25e−4	1.25e−4
AdamW β	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$
Warm-up iterations	2000	2000	2000	2000
Training epochs	132	102	68	58
Flat-LR epoch	64	49	34	29
No-augmentation epochs	12	12	8	8
Augmentation stop epoch	120	90	60	50
Matcher-switch epoch	100	80	50	45
Gradient clipping	0.1	0.1	0.1	0.1
Architecture Settings				
Backbone	`vit_tiny`	`vit_tinyplus`	`vits16`	`vits16plus`
Interaction blocks	[3, 7, 11]	[3, 7, 11]	[5, 8, 11]	[5, 8, 11]
Spatial stem width	16	16	32	64
Detection hidden dim.	192	256	224	256
Encoder FFN dim.	512	512	896	1024
Encoder depth mult.	0.67	1.00	1.00	1.37
Encoder expansion	0.34	0.67	1.00	1.25
Decoder layers	4	4	4	6
Decoder FFN dim.	512	512	1792	2048

B Implementation Details

All variants inherit their optimization, data processing, and detector settings from the corresponding DEIMv2 Huang et al. (2025a) model unless otherwise stated. Table 5 summarizes the inherited training and architecture settings for all model scales.

The base learning rate is used for parameters outside the DINOv3 Siméoni et al. (2025) backbone, while the smaller backbone learning rate is applied to all DINOv3 parameters. Normalization parameters and biases are excluded from weight decay. All models use automatic mixed precision, an exponential moving average with decay 0.9999, and the flat-cosine learning-rate schedule adopted by

DEIMv2. The training pipeline follows Dense O2O augmentation, including Mosaic, MixUp Zhang et al. (2018), and CopyBlend, together with photometric distortion, random zoom-out, random IoU crop, horizontal flipping, and multi-scale collation. Validation resizes each image to 640×640 and reports COCO bounding-box metrics.

The interaction indices specify the DINOv3 blocks used to construct the three detection levels. The encoder depth multiplier and expansion ratio control the convolutional cross-scale fusion inherited from DEIMv2.

The remaining detector settings are shared across scales: three feature levels, 300 object queries, 100 denoising queries, a label-noise ratio of 0.5, a box-noise scale of 1.0, three/six/three MSDA sampling points on the three levels, and $\mathrm{reg_max} = 32$ with regression scale 4. The Hungarian matcher uses class, box, and GIoU costs of 2, 5, and 2, respectively. The DEIM loss weights for MAL, box regression, GIoU, FGL, and DDF are 1, 5, 2, 0.15, and 1.5.

C Additional Details of Spatial-Enhanced Adapter

C.1 STA-based Spatial Trunk

SEA retains the semantic path of DINOv3 and follows STA by constructing a parallel convolutional trunk from the input image. The original STA spatial prior forms its stride-4 stem with a 3×3 stride-2 convolution followed by a 3×3 stride-2 max-pooling layer, and then uses three successive stride-2 convolutions to produce detail features at strides 8, 16, and 32, with channel dimensions $(2c, 4c, 4c)$. SEA preserves this output hierarchy but replaces the max-pooling operation with a learnable stride-2 convolution and augments the resulting convolutional trunk with C3k2 blocks. Specifically, two 3×3 stride-2 Conv–SyncBN–SiLU layers form the stride-4 feature, and each subsequent stage begins with another 3×3 stride-2 layer. A C3k2 block is applied after every stage, including the stride-4 stem stage:

$$X_2 = \mathcal{B}_2\big(\mathcal{C}^c_{3\times3,2}\left(\mathcal{C}^c_{3\times3,2}(I)\right)\big), \qquad X_l = \mathcal{B}_l\Big(\mathcal{C}^{C_l}_{3\times3,2}(X_{l-1})\Big), \quad l \in \{3,4,5\},$$

Here, I is the input image and $\mathcal{C}_{3\times3,2}$ denotes Conv–SyncBN–SiLU with stride 2. The stage widths are $(C_2, C_3, C_4, C_5) = (c, 2c, 4c, 4c)$, where the stem width c for each model scale is given in Table 5. The three spatial outputs are

$$(D_3, D_4, D_5) = \sigma(g)(X_3, X_4, X_5),$$

where the shared output logit g is initialized to -3. This initialization introduces the enhanced spatial trunk gradually while retaining the original three-level STA interface.

C.2 C3k2 Block

For an input X and output width C_o, the C3k2 block first projects X to $2h$ channels and splits the result into two equal parts:

$$[U_0, V_0] = \mathrm{Split}\big(\mathcal{C}^{2h}_{1\times1}(X)\big), \qquad h = \lfloor eC_o \rfloor.$$

Only the second part is successively transformed by n bottlenecks, $V_j = \mathcal{R}_j(V_{j-1})$, while U_0 bypasses these spatial transformations. The block output is

$$\mathcal{B}(X) = \mathcal{C}^{C_o}_{1\times1}\left([U_0; V_0; V_1; \ldots; V_n]\right).$$

Each bottleneck contains a 1×1 projection followed by a 3×3 convolution, both with SyncBN and SiLU, and uses a residual connection He et al. (2016) when its input and output dimensions agree:

$$\mathcal{R}(V) = V + \mathcal{C}_{3\times3}\left(\mathcal{C}_{1\times1}(V)\right).$$

We use $n = 1$ and expansion ratio $e = 0.5$ for the standard SEA variants. The final activation of the C3k2 projection is omitted at the three returned detail levels to preserve the output convention of the original STA branch.

D Additional Details of Local Evidence Refinement

D.1 Review of Multi-Scale Deformable Attention

At decoder layer l, the query q_i after self-attention Vaswani et al. (2017) and its reference box r_i are passed to Multi-Scale Deformable Attention (MSDA) Zhu et al. (2021b). For attention head a, feature level $k \in \{3, 4, 5\}$, and sampling point j, MSDA predicts a sampling offset Δp_{iakj} and an attention logit e_{iakj} from the query. The logits are normalized across all levels and sampling points within each head,

$$\omega_{iakj} = \mathrm{Softmax}_{k,j}(e_{iakj}), \qquad \sum_k \sum_j \omega_{iakj} = 1.$$

The reference box and predicted offset determine the sampling location $p_{iakj} = \mathcal{R}_k(r_i, \Delta p_{iakj})$ on level k, where $\mathcal{R}_k$ denotes the level-aware reference-point transformation. Bilinear interpolation on the projected feature map F_k then produces $v_{iakj} = \mathcal{B}(F_k, p_{iakj})$. The standard MSDA output aggregates these sampled values as

$$o_i = W_o \operatorname*{Concat}_a \left(\sum_k \sum_j \omega_{iakj} v_{iakj} \right).$$

LER does not introduce new sampling locations. Instead, it retains the sampled values v_{iakj} and their native attention weights ω_{iakj} before the cross-scale summation, from which the scale-specific statistics below are constructed.

D.2 Scale-specific Statistics from MSDA

Using the MSDA quantities defined above, LER retains the attention mass assigned to each level and renormalizes the point weights inside that level:

$$M_{iak} = \sum_j \omega_{iakj}, \qquad p_{iakj} = \frac{\omega_{iakj}}{M_{iak} + \epsilon}.$$

The scale-specific feature, native scale mass, and normalized within-level entropy are

$$Z_{ik} = \operatorname*{Concat}_a \left(\sum_j p_{iakj} v_{iakj} \right), \qquad m_{ik} = \frac{1}{A} \sum_a M_{iak},$$

$$h_{ik} = -\frac{1}{A} \sum_a \frac{\sum_j p_{iakj} \log(p_{iakj} + \epsilon)}{\log N_k}.$$

Here, A is the number of attention heads and N_k is the number of sampling points at level k. For a level with one sampling point, its entropy is set to zero. The masses are clamped for numerical stability and normalized across levels before use. These quantities reuse the original MSDA locations and sampled values; LER performs no additional feature sampling.

D.3 Localization Reweighting

Let $d_i = \sqrt{w_i h_i}$ denote the normalized extent of the current reference box, and let $d_k = t_k / \sqrt{H_k W_k}$ denote the learned characteristic extent of level k. The positive cell extent t_k is initialized to 4 and learned through a softplus parameterization. LER measures scale compatibility by

$$g_{ik} = -\frac{|\log d_i - \log d_k|}{\tau}.$$

It further measures whether the scale-specific feature agrees with the query entering MSDA through $c_{ik} = \cos(Z_{ik}, q_i)$. The localization score is then

$$r_{ik} = \beta_g g_{ik} + \beta_m \log(m_{ik} + \epsilon) + \beta_c c_{ik} - \beta_h h_{ik} + b_k,$$

followed by

$$\pi_{ik} = \mathrm{Softmax}_k \left(r_{ik} - \beta_{\mathrm{loc}} h_{ik} \right).$$

Table 6: **Ablations for LER.** Numbers in parentheses denote gains over the baseline (first row). Loc. Reweight., Native Sub., and Size Scale denote localization reweighting, native context subtraction, and size-dependent scaling, respectively. Blue-shaded row indicates our proposed configuration.

Loc. Reweight.	Native Sub.	Size Scale	Params (M)	FLOPs (G)	AP	AP_S	AP_M	AP_L
–	–	–	9.7	25.7	50.9	31.3	55.3	70.1
✓	✓	–	9.76	25.7	50.8 (-0.1)	31.5 (+0.2)	55.0	69.8
✓	–	✓	9.76	25.7	50.8 (-0.1)	31.9 (+0.6)	54.9	70.5
✓	✓	✓	9.76	25.7	**51.0** (+0.1)	**32.0** (+0.7)	54.8	70.4

All β values are constrained to be positive through softplus. The geometry, mass, and query-consistency coefficients are initialized to 1, the two entropy coefficients to 0.5, and the level biases b_k to zero. The temperature is $\tau = 0.7$.

The native and localization-oriented summaries are

$$Z_i^{\text{nat}} = \sum_k m_{ik} Z_{ik}, \qquad Z_i^{\text{loc}} = \sum_k \pi_{ik} Z_{ik}.$$

LER retains only their difference,

$$E_i^{\text{loc}} = Z_i^{\text{loc}} - Z_i^{\text{nat}} = \sum_k (\pi_{ik} - m_{ik}) Z_{ik},$$

so that it represents the change induced by localization reweighting rather than a second complete context vector.

D.4 Localization Residual

The difference is transformed by a bias-free two-layer projector

$$\phi(E) = W_2 \,\text{SiLU}(W_1 E),$$

whose bottleneck width is 32. To bound the feature correction, we use

$$\mathcal{B}(v) = \frac{v}{1 + \text{RMS}(v)}.$$

LER continuously prioritizes small candidates using their predicted reference-box area rather than ground-truth size:

$$s_i = \sigma\left(\frac{\log\sqrt{a_0} - \log d_i}{\tau}\right), \qquad a_0 = 0.0025.$$

This is a soft weighting rather than a hard small/medium/large partition. The confidence of the localization allocation is

$$c_i^{\text{route}} = 1 - \frac{\mathcal{H}(\boldsymbol{\pi}_i)}{\log 3}.$$

The final residual is

$$\Delta q_i^{\text{loc}} = \rho \, s_i \, c_i^{\text{route}} \mathcal{B}\big(\phi(E_i^{\text{loc}})\big), \qquad \rho = 0.1 \tanh(\gamma),$$

where ρ is initialized to 0.05. This residual is computed at every decoder layer and added only to the feature passed to the box-regression head. The unmodified decoder query is retained for classification and for propagation to the next decoder layer. Thus, LER changes the localization readout without directly adding an offset to the predicted box coordinates or altering query selection.

D.5 Ablation Study of LER

Table 6 evaluates the main design choices of LER. Starting from the baseline, all LER variants add localization reweighting. Adding localization reweighting and native context subtraction without the size-dependent scaling s_i lowers AP from 51.0 to 50.8 and AP_S from 32.0 to 31.5. The combination of localization reweighting and s_i without native context subtraction directly uses Z_i^{loc} instead of $E_i^{\text{loc}} = Z_i^{\text{loc}} - Z_i^{\text{nat}}$, reaching 31.9 AP_S, but reduces overall AP to 50.8 and AP_M to 54.9 relative to the baseline. In contrast, full LER adds both native context subtraction and size-dependent scaling, yielding 51.0 AP and 32.0 AP_S with negligible computational overhead. The inference-stage analysis further shows that retaining the correction during decoder refinement is important for improving small-object localization.

E Analysis Protocol

The analysis in Figure 1 is conducted on all 5,000 images of COCO 2017 validation. We use category-aware COCO bounding-box matching with the standard area ranges and a maximum of 100 detections per image. Crowd and ignored annotations are excluded from the valid object cohort. Unless stated otherwise, small objects follow the COCO annotation-area definition, $a \leq 32^2$, and the hit/miss grouping in panels (b)–(d) uses matching at IoU 0.50.

E.1 Spatial Coverage of Missed Objects

The encoder proposal stage selects 300 source tokens from the flattened stride-8, stride-16, and stride-32 feature grids. Each selected token is mapped back to the center of its native grid cell in normalized image coordinates. A small ground-truth object is considered spatially covered when at least one selected grid center lies inside its bounding box:

$$\mathrm{cov}(G) = \mathbf{1}[\exists n : \; p_n \in G] \, .$$

The reported quantity is

$$R_{\mathrm{cov}}(t) = \frac{\sum_{G \in \mathcal{M}_t} \mathrm{cov}(G)}{|\mathcal{M}_t|},$$

where $\mathcal{M}_t$ is the set of unmatched small GT objects at COCO matching threshold t. This statistic describes spatial coverage only: it does not require the selected token's initialized or final box to match the object.

E.2 Target–Neighborhood Feature Separation

For each feature level, we use native-grid cell centers without upsampling the feature map. The target set T_G contains grid centers inside ground-truth box G. The context set R_G contains centers inside a box with the same center and 2.5 times the width and height of G, excluding the target region itself. Objects with fewer than two target tokens or fewer than two context tokens are excluded for that level.

Let $\bar{f}_p = f_p / \|f_p\|_2$ be the channel-normalized feature at grid position p. We compute the off-diagonal target similarity

$$C_{\mathrm{in}}(G) = \frac{1}{|T_G|(|T_G|-1)} \sum_{p \neq q \in T_G} \bar{f}_p^{\top} \bar{f}_q,$$

and target–ring similarity

$$C_{\mathrm{ring}}(G) = \left(\frac{1}{|T_G|} \sum_{p \in T_G} \bar{f}_p \right)^{\top} \left(\frac{1}{|R_G|} \sum_{q \in R_G} \bar{f}_q \right).$$

The separation score is

$$\mathrm{Sep}(G) = C_{\mathrm{in}}(G) - C_{\mathrm{ring}}(G).$$

A larger value indicates that target tokens are more mutually consistent than they are similar to the surrounding ring under this metric. The pre-encoder feature is the output of STA or SEA, the post-encoder/pre-CDP feature is the output of the original HybridEncoder before CDP, and the post-CDP feature is the final encoder output. Panel (b) reports the object-wise mean of the P3 score for the COCO hit and miss cohorts.

E.3 Sensitivity to Local Details

For each valid small object, we construct two matched evaluations using the same perturbation type and magnitude: one modifies only the detail content inside the target box, and the other modifies an equal-area region sampled from its surrounding neighborhood. Let I be the original image, I_T and I_R the target- and neighborhood-perturbed images, and $\mu_T(F_3(I))$ the mean F3 feature over target-grid locations. The relative changes are

$$\Delta_T = \frac{\|\mu_T(F_3(I_T)) - \mu_T(F_3(I))\|_2}{\|\mu_T(F_3(I))\|_2 + \epsilon},$$

$$\Delta_R = \frac{\|\mu_T(F_3(I_R)) - \mu_T(F_3(I))\|_2}{\|\mu_T(F_3(I))\|_2 + \epsilon}.$$

The same target annotations and perturbation regions are used when comparing DEIMv2 and SEA. Panel (c) reports the object-wise means of Δ_T and Δ_R. A larger target response together with a smaller neighborhood response indicates that the target representation depends more selectively on the object's own local details rather than on surrounding content.

E.4 Localization through Refinement

This analysis is restricted to objects containing at least one query-selected source location. For each such ground-truth box G, let $\mathcal{Q}_G$ be the corresponding selected queries whose source-grid centers lie inside G. At initialization, we choose

$$q_G^* = \arg\max_{q \in \mathcal{Q}_G} \mathrm{IoU}(B_q^{\mathrm{init}}, G).$$

The same query index is then tracked through all decoder layers; the best query is not reselected at later stages. Panel (d) reports

$$\frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} \mathrm{IoU}(B_{q_G^*}^s, G), \qquad s \in \{\mathrm{init}, \mathrm{final}\},$$

separately for the COCO hit and miss cohorts. The resulting curve measures how much an already spatially covered candidate improves during decoder refinement; it is not obtained by independently selecting the best box at each decoder stage.