FSD: BRIDGING FOUNDATION SEMANTICS AND LOCAL DETAILS FOR REAL-TIME OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

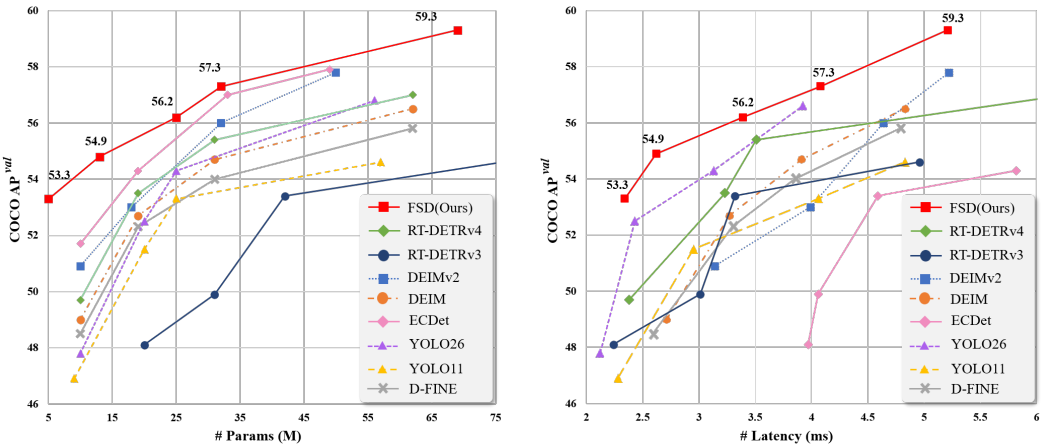
Real-time object detection has advanced rapidly with CNN- and Transformer-based architectures. Recently, Vision Foundation Models (VFMs) have been introduced to further enhance detectors with rich pretrained semantics. Where partially overlapping semantic modeling may lead to redundant representations and limit further improvements. In fact, the structural complementarity between the VFM and detector remains largely overlooked in existing VFM-enhanced detection research. Motivated by this, we propose FSD, a VFM-enhanced real-time detector that exploits complementary semantic and spatial representations. Specifically, FSD couples a frozen EuPE semantic branch with a trainable CNN spatial pathway to combine global semantics with fine-grained spatial details. To bridge their heterogeneous representations, we further propose Multi-level Semantic-Detail Interaction (MSDI) for adaptive multi-level interaction. Finally, we design an Efficient Auxiliary Head (EAH) to promote detection-oriented adaptation of semantic features and better balance semantic and spatial-detail cues. Extensive experiments on COCO demonstrate that FSD achieves state-of-the-art performance with a favorable accuracy and efficiency trade-off. Further analyses highlight the importance of representational complementarity between VFMs and detectors.

1 INTRODUCTION

In recent years, real-time object detection has made remarkable progress along two major architectural paradigms. CNN-based detectors, represented by the YOLO familyJiang et al. (2022); Sapkota et al. (2025), efficiently preserve fine-grained visual structures such as edges, textures, and object boundaries. Meanwhile, their relatively low computational overhead makes them particularly attractive for practical deployment. In parallel, Transformer-based detectors, originating from DETR variantsShehzadi et al. (2025); Zhu et al. (2020); Meng et al. (2021); Li et al. (2022), leverage global context modeling and long-range dependencies to achieve strong detection performance. Recent methods such as RT-DETRZhao et al. (2024), DEIMHuang et al. (2025b), and their variantsHaris et al. (2026) further narrow the efficiency gap between these paradigms through efficient hybrid architectures and improved training strategies. Nevertheless, despite these impressive advances, these conventional detectors are still primarily trained within task-specific frameworks and remain limited in semantic abstraction and generalizationYao et al. (2026).

The emergence of vision foundation models (VFMs) provides a promising opportunity to overcome these limitationsAwais et al. (2025). Benefiting from large-scale pre-training, VFMs encode rich semantic knowledge and exhibit strong transferability across a wide range of downstream tasksLu et al. (2025); Zhang et al. (2024). Accordingly, recent studies have begun to exploit foundation-model representations to enhance object detectors, such as DEIMv2Huang et al. (2025a) and RT-DETRv4Liao et al. (2026). However, these VFM-enhanced detectors largely follow a straightforward paradigm: injecting foundation-model features into state-of-the-art detection frameworks, while most of these strong baselines are built upon Transformer-based architectures. Although this paradigm has led to further performance improvements, existing VFM-enhanced detectors primarily integrate strong foundation representations into already powerful Transformer-based detection frameworks, with limited consideration of architectural complementarity. More importantly, whether such integration truly improves the balance between semantic abstraction and the fine-grained local details required for detection remains insufficiently understood. Since both Transformer-based VFMs and Transformer detectors are effective at modeling global context and high-level semantics, their com-

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.



(a) COCO performance v.s. Trainable Parameters.

(b) COCO performance v.s. Inference Latency.

Figure 1: Compared with real-time object detectors on COCO. All variants of FSD achieve SOTA performance while maintaining a small number of trainable parameters and low inference latency.

bination may introduce partially redundant semantic representations while providing insufficient compensation for the fine-grained spatial cues required for accurate localization. Such representational imbalance may therefore limit further performance gains. Moreover, existing approaches often rely on high-level or single-scale VFM features, leaving the hierarchical representations of foundation models under-explored. These observations motivate a more complementary design that combines strong VFM semantics with spatial representations.

Motivated by these observations, we propose FSD, a VFM-enhanced real-time detector that exploits the complementarity between foundation-model semantics and CNN spatial representations. By coupling the high-level knowledge of a frozen VFM with the localization-sensitive inductive bias of a CNN-based detector, FSD balances global semantic abstraction and fine-grained spatial modeling. To further bridge these heterogeneous representations, we propose Multi-level Semantic-Detail Interaction (MSDI), which adaptively performs soft alignment and interaction between semantic and spatial features across multiple scales. Finally, considering that simply introducing VFM features does not guarantee their effective utilization for detection. To address this, we design an Efficient Auxiliary Head (EAH) for semantic supervision, strengthening the relevance of semantic representations to the detection task. Together, these designs enable FSD to achieve state-of-the-art accuracy while maintaining high inference efficiency. In summary, our contributions are as follows:

- We propose FSD, a VFM-enhanced real-time detector that couples strong VFM semantics with fine-grained CNN spatial representations. FSD employs Multi-level Semantic-Detail Interaction (MSDI) for hierarchical feature fusion and an inference-free auxiliary head to improve the detection awareness of VFM features during training.
- We establish a new family of FSD, with N/S/M/L/X, achieving $53.3/54.9/56.2/57.3/59.3$ AP scores on COCO at $427/382/251/245/192$ FPS, setting a new SOTA on COCO.
- We further systematically analyze the complementarity between VFM semantics and CNN spatial representations, demonstrating the effectiveness of hierarchical semantic-detail interaction for real-time object detection.

2 Related Work

2.1 Real-time object detection.

Real-time object detection has evolved along two major architectural paradigms. CNN-based one-stage detectors, represented by the YOLO familyTian et al. (2025), have long been favored for their efficient local feature extraction and deployment-friendly design, providing a strong accuracy and latency trade-off for practical applications. Recent variants such as YOLO13Lei et al.

(2025), YOLO26Jocher et al. (2026) further improve feature extraction, model scaling, and inference efficiency, making CNN detectors highly competitive under strict real-time constraints. In parallel, DETR-style detectors have introduced an alternative end-to-end paradigm based on set prediction. RT-DETRZhao et al. (2024) substantially reduces the computational cost of conventional DETRs through an efficient hybrid encoder and establishes real-time end-to-end detection. Subsequent works further improve this paradigm from different perspectives: RT-DETRv2Lv et al. (2024) enhances training and deployment flexibility, RT-DETRv3Wang et al. (2025) introduces hierarchical dense positive supervision, D-FINEPeng et al. (2025) reformulates bounding-box regression as fine-grained distribution refinement, and DEIMHuang et al. (2025b) improves one-to-one matching to accelerate convergence. These advances have significantly narrowed, and in some settings even reversed, the performance gap between real-time CNN and Transformer detectors.

Despite these improvements, existing real-time detectors are still primarily optimized within task-specific detection frameworks. Their representational capacity therefore remains constrained by the detection training data and model architecture.

2.2 Foundation models for detection.

Vision Foundation Models (VFMs) provide rich semantic priors and strong transferable representations, such as DINOv3Siméoni et al. (2025), LingBot-VisionFu et al. (2026), Sapiensv2Khirodkar et al. (2026), offering a promising way to enhance conventional detectors beyond task-specific supervision. Accordingly, recent studies have increasingly explored how foundation representations can be transferred to different object detection scenarios. For example, Frozen-DETR Fu et al. (2024) treats frozen foundation models as plug-and-play feature enhancers for DETR, while FM-FSOD Han & Lim (2024) leverages DINOv2 representations to improve few-shot detection. OW-Rep Lee et al. (2026) further transfers semantic knowledge from VFMs to open-world object detectors for more generalizable representations. Collectively, these studies demonstrate the strong transferability of VFMs representations across diverse detection settings. Most recently, VFMs have also been introduced into real-time detection. RF-DETR Robinson et al. (2026) employs a DINOv2 together with a lightweight DETR decoder to achieve a favorable accuracy-latency trade-off. DEIMv2 Huang et al. (2025a) incorporates DINOv3 representations into the DEIM framework and adapts them to multi-scale detection, while RT-DETRv4 Liao et al. (2026) transfers semantic knowledge from a frozen DINOv3 teacher to lightweight DETR models through semantic injection and adaptive modulation. In addition, ECDet Liu et al. (2026) distills detection-specialized knowledge from a DINOv3-pretrained teacher into compact ViT models for efficient edge deployment.

However, many recent VFM-enhanced approaches combine Transformer-based VFMs with Transformer-based detectors, while largely overlooking their structural complementarity. Since these methods mainly transfer stronger pretrained semantics, this may partially overlap with the detector's own semantic modeling. This motivates a complementary design that better balances global semantics with localization-sensitive spatial details.

3 Method

3.1 Overall Architecture

Existing VFM-assisted detectors largely focus on introducing stronger semantic representations, while overlooking their complementarity with local spatial details. To bridge this gap, we propose FSD, a real-time detector that couples foundation semantics with fine grained spatial representations. As illustrated in Fig. 2, FSD consists of a dual-branch backbone with a frozen VFM semantic branch and a trainable CNN spatial branch, which are further coupled through Multi-level Semantic-Detail Interaction (MSDI) stage, together with a main detection head and an auxiliary head for supervision.

3.2 Backbone

To fully exploit the representational advantages of different visual architectures, FSD adopts a dual-path backbone consisting of a trainable CNN branch and a frozen EuPE-BZhu et al. (2026) branch. Instead of directly replacing the conventional detection backbone with a large-scale pretrained model or as in some recent works, further inject VFM features into transformer detectors that already pos-

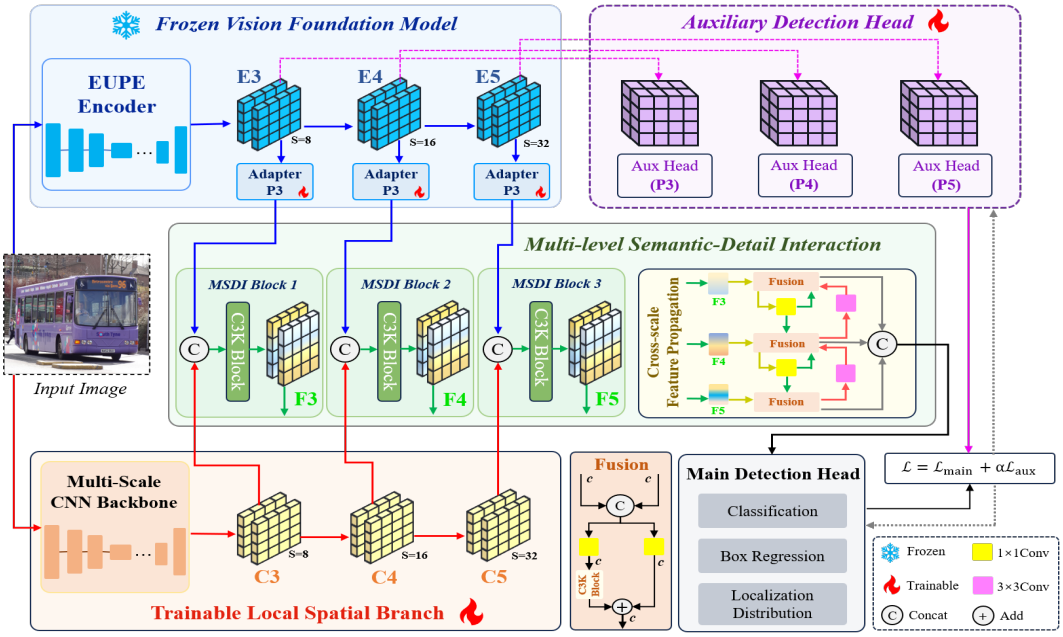


Figure 2: Overall architecture of FSD. FSD consists of a frozen VFM semantic branch and a trainable CNN spatial branch, which are coupled through multi-level semantic-Detail interaction. An auxiliary head is introduced only during training to provide additional semantic supervision.

sess strong high-level semantic modeling capabilities, since such high-semantic-to-high-semantic coupling may introduce redundant information and competing representations.Instead, we employ a CNN structure as the primary spatial feature extraction pathway to preserve the computational efficiency required for real-time detection as well as its sensitivity to local structures. Meanwhile, EuPE is introduced as a frozen semantic expert to provide high-level visual representations learned from large-scale pretraining while maintaining a relatively low training cost.

Specifically, the CNN branch employs multiple convolutional blocks with the C3K structure to facilitate rich multi-level gradient flow, thereby preserving fine-grained information such as object edges, textures, and spatial locations. In parallel, the semantic abstraction branch leverages the high-level visual abstraction capability of EuPE to provide more stable semantic representations and broader contextual information. In this way, hierarchical features with distinct representational properties are constructed. During training, the parameters of EuPE are kept frozen, which not only preserves the general representations learned during pretraining but also substantially reduces the additional training overhead introduced by the vision foundation model.

Given an input image $\mathbf{I}$, the two branches independently construct hierarchical feature:

$$\{\mathbf{C}_i\}_{i=1}^{L_c} = \mathcal{B}_{\text{cnn}}(\mathbf{I}), \qquad \{\mathbf{S}_j\}_{j=1}^{L_s} = \mathcal{B}_{\text{vfm}}(\mathbf{I}). \tag{1}$$

where $\mathbf{C}_i$ and $\mathbf{S}_j$ denote the features produced at different stages of the CNN and EuPE branches, respectively, while L_c and L_s denote the numbers of feature stages in the two branches. These hierarchical representations provide diverse and informative feature bases for subsequent detection.

3.3 Multi-level Semantic-Detail Interaction

Due to significant differences in feature distribution, abstraction level, and spatial response patterns between CNN and VFM, simple element by element addition or channel concatenation is difficult to fully utilize their effectiveness. Therefore, we design the Multi-level Semantic-Detail Interaction (MSDI) module to facilitate effective inter-feature interaction between heterogeneous multi-scale features. Specifically, FSD use three feature levels from the CNN and EuPE branches, denoted as $\{\mathbf{C}_3, \mathbf{C}_4, \mathbf{C}_5\}$ and $\{\mathbf{S}_3, \mathbf{S}_4, \mathbf{S}_5\}$, respectively. These features correspond to spatial strides of 8, 16, and 32 relative to the input image, enabling interactions across different spatial scales. For each

scale, the CNN and EuPE features are first concatenated along the channel dimension and then passed through a C3k blockKhanam & Hussain (2024) for soft adaptive representation alignment. To further exploit the complementary information distributed across feature levels, we design a Cross-scale Complementary Feature Propagation strategy, which establishes staged interactions between adjacent-scale features and progressively incorporates their complementary information into each representation level. In this way, high-level information can be transferred to finer-resolution features, while spatially detailed responses can be further propagated back to deeper representations, yielding more coherent multi-scale features for subsequent detection.

3.4 Efficient Auxiliary Head

Although the dual-branch backbone and MSDI address how to introduce heterogeneous semantic representations and interact them with spatial local features, effective interaction does not necessarily ensure that the semantic representations contain sufficient detection-relevant cues. Since the detection supervision is applied only after semantic-detail interaction, the semantic features are constrained indirectly through the fused prediction. This may make it difficult for the model to explicitly identify and exploit detection-discriminative information from EuPE representations, while the spatial features may dominate the learning process. To provide an explicit detection-oriented constraint for the adapted semantic representations, we introduce an Efficient Auxiliary Head (EAH), which directly guides the model to extract and exploit detection-relevant semantic cues. Specifically, given the hierarchical EuPE features, we select three scales $\{S_3, S_4, S_5\}$ corresponding to strides of 8, 16, and 32, respectively, to provide multi-scale semantic supervision for objects of different sizes:

$$\hat{\mathcal{Y}}_{\text{aux}} = \mathcal{H}_{\text{aux}}(S_3, S_4, S_5). \tag{2}$$

The auxiliary detection objective provides an explicit detection-oriented constraint on the semantic pathway, preventing the optimization from being dominated by the readily trainable spatial branch. Together with the main detection objective, it encourages the model to maintain a better balance between semantic cues and fine-grained spatial details during semantic-detail interaction.

3.5 Loss Function

The overall training loss of FSD consists of the main detection loss and the auxiliary detection loss, formulated as

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}, \tag{3}$$

where $\mathcal{L}_{\text{main}}$ supervises the final detection features after MSDI stage, $\mathcal{L}_{\text{aux}}$ provides additional detection supervision for the semantic branch, and λ_{aux} denotes the weight of the auxiliary loss(default=0.5).

For each detection branch $h \in \{\text{main}, \text{aux}\}$, the loss consists of the bounding-box regression loss, classification loss, and Distribution Focal Loss:

$$\mathcal{L}_h = \lambda_{\text{box}}\mathcal{L}^h_{\text{box}} + \lambda_{\text{cls}}\mathcal{L}^h_{\text{cls}} + \lambda_{\text{dfl}}\mathcal{L}^h_{\text{dfl}}, \tag{4}$$

where $\mathcal{L}_{\text{box}}$ adopts CIoU to constrain the localization error between the predicted and ground-truth boxes, $\mathcal{L}_{\text{cls}}$ adopts binary cross-entropy loss for category prediction, and $\mathcal{L}_{\text{dfl}}$ employs Distribution Focal Loss to further model the bounding-box distribution. The main and auxiliary branches perform target assignment and loss computation independently based on their respective predictions, and jointly optimize FSD through the above objective.

4 Experiments

4.1 Setup

Implementation Details. Our experiments are built upon the Ultralytics YOLO architecture Sapkota & Karkee (2025). For fair comparison, the training settings and hyperparameters are kept consistent with baselines unless otherwise specified. FSD adopts a pre-trained EuPE-Base model as the semantic branch, whose parameters remain frozen throughout training. Inference latency is measured in milliseconds on a single NVIDIA RTX 4090 GPU with TensorRT FP16 precision.

Table 1: Comparison with state-of-the-art real-time object detectors. #Params. denotes trainable parameters. All latency are measured using TensorRT with FP16 on single RTX 4090 GPU.

Model	#Epochs	#Params.	GFLOPs	Latency(ms)	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
YOLOv8-S	500	11	29	2.53	44.9	61.8	48.6	25.7	49.9	61.0
YOLO11-S	500	9	22	2.28	46.6	63.4	50.3	28.7	51.3	64.1
YOLOv12-S	500	9	19	3.07	48.0	65.0	51.8	29.8	53.2	65.6
YOLO26-S	600	10	23	2.12	47.8	64.6	52.1	29.1	52.5	64.3
D-FINE-S	120	10	25	2.59	48.5	65.6	52.6	29.1	52.2	65.4
DEIM-S	120	10	25	2.71	49.0	65.9	53.1	30.4	52.6	65.7
ECDET-S	74	10	26	3.19	51.7	69.4	55.8	32.3	56.4	70.5
RT-DETRv3-S	120	20	60	2.24	48.1	65.6	52.0	30.2	51.5	63.9
RT-DETRv4-S	120	10	25	2.38	49.7	66.8	54.1	30.2	53.6	66.9
DEIMv2-S	120	10	26	3.14	50.9	68.3	55.1	31.4	55.3	70.3
FSD-N	200	5	108	2.34	**53.3**	**72.1**	**57.8**	**32.6**	**59.8**	**74.4**
FSD-S	120	13	117	2.62	**54.9**	**73.4**	**59.2**	**35.1**	**60.9**	**74.7**
YOLOv8-M	500	26	79	3.02	50.2	67.2	54.6	32.0	55.7	66.4
YOLO11-M	500	20	68	2.95	51.2	67.9	55.3	33.0	56.7	67.5
YOLOv12-M	600	20	60	3.46	52.6	65.9	-	-	-	-
YOLO26-M	500	22	76	2.43	52.5	69.8	57.2	36.2	56.9	68.5
D-FINE-M	120	19	57	3.31	52.3	69.8	56.4	33.2	56.5	70.2
DEIM-M	90	19	57	3.27	52.7	70.0	57.3	35.3	56.7	69.5
ECDET-M	90	19	53	4.87	54.3	72.2	58.7	35.9	59.1	72.7
RT-DETRv3-M	120	31	92	3.01	49.9	67.5	58.6	35.8	58.6	72.1
RT-DETRv4-M	120	31	92	3.23	49.9	67.5	58.6	35.8	58.6	72.1
DEIMv2-M	90	18	52	3.99	53.0	70.2	57.6	34.2	57.4	71.5
FSD-M	90	25	359	3.39	**56.2**	**74.5**	**61.2**	**37.4**	**61.8**	**74.9**
YOLOv8-L	500	43	165	3.96	52.9	68.6	57.5	35.3	58.3	69.8
YOLO11-L	500	25	87	4.06	53.4	70.1	58.2	35.6	59.1	69.2
YOLOv12-L	600	27	82	4.42	54.0	72.6	-	-	-	-
YOLO26-L	500	26	94	3.13	54.3	71.5	59.4	37.8	58.6	70.3
D-FINE-L	72	31	91	3.85	54.0	71.6	58.4	36.5	58.0	71.9
DEIM-L	50	31	91	3.91	54.7	72.4	59.4	36.9	59.6	71.8
ECDet-L	50	49	151	5.03	57.0	76.0	62.9	38.7	63.4	76.1
RT-DETRv3-L	72	42	136	3.62	53.4	71.6	57.4	36.1	57.9	70.8
RT-DETRv4-L	50	31	91	3.81	55.4	73.0	60.3	37.1	60.1	72.9
DEIMv2-L	60	32	96	4.64	56.0	73.4	60.9	37.5	60.8	75.2
FSD-L	60	32	385	4.08	**57.3**	**76.6**	**63.4**	**39.3**	**64.0**	**76.5**
YOLOv8-X	500	68	257	4.68	53.9	71.0	58.7	35.7	59.3	70.7
YOLO11-X	500	57	195	4.83	54.7	71.6	59.5	37.7	59.7	70.2
YOLOv12-X	600	59	185	5.30	55.7	72.2	-	-	-	-
YOLO26-X	500	59	210	3.92	56.8	74.1	62.1	41.4	61.2	72.7
D-FINE-X	72	62	202	4.79	55.8	73.7	60.2	37.3	60.5	73.4
DEIM-X	50	62	202	4.83	56.5	74.0	61.5	38.8	61.4	74.2
ECDet-X	50	49	151	6.68	57.9	76.0	62.9	38.7	63.4	76.1
RT-DETRv3-X	72	76	259	4.96	54.3	72.8	58.8	35.8	58.8	72.1
RT-DETRv4-X	50	62	202	5.86	57.0	74.6	62.1	39.5	61.9	74.8
DEIMv2-X	50	50	152	5.22	57.8	75.4	63.2	39.2	62.9	75.9
FSD-X	50	70	517	5.21	**59.3**	**76.7**	**64.6**	**40.2**	**63.8**	**76.6**

Dataset and Metric. All experiments are conducted on the COCO-2017 datasetLin et al. (2014). The train-2017 split is adopted for model training, and val-2017 is used for evaluation. We report standard COCO evaluation metrics, including AP (averaged over IoU thresholds from 0.50 to 0.95 at 0.05 intervals), AP_{50}, AP_{75}, as well as scale-aware metrics AP_S, AP_M and AP_L.

4.2 Comparison with SOTA

We compare FSD with representative state-of-the-art real-time object detectors, including both CNN-based and Transformer-based methods, with detailed statistics provided in Table 1. As shown in Table 1, although EuPE noticeably increases the nominal FLOPs, the practical increase in inference latency is substantially smaller than what the nominal FLOPs alone would suggest, and FSD still achieves extremely low inference latency. This discrepancy arises because FLOPs do not account for hardware execution efficiency, while most additional computations introduced by EuPE are regular matrix-based operations that can be efficiently parallelized on modern GPUs. Moreover, the auxiliary detection head is completely discarded at inference, introducing no additional runtime cost. Moreover, during training, EuPE remains fully frozen and performs only forward feature extraction without gradient computation, back-propagation, or parameter updates. Consequently,

Table 2: Ablation experiments for each key component of FSD on COCO dataset. DB denotes the EuPE-assisted dual-branch backbone, and EAH denotes the Efficient Auxiliary Head.

DB	MSDI	EAH	AP	AP_{50}	AP_{75}
-	-	-	39.4	55.2	42.9
✓	-	-	47.5	65.3	51.3
✓	✓	-	52.6	71.5	56.7
✓	-	✓	47.6	65.9	51.9
✓	✓	✓	53.3	72.1	57.8

Table 3: Interaction strategy analysis of MSDI on COCO dataset.

Strategy	E_3	E_4	E_5	Fusion	AP	AP_{50}	AP_{75}
Baseline	-	-	-	-	39.4	55.2	42.9
Position	✓	-	-	Ours	46.8	65.9	51.3
	-	✓	-		47.9	67.8	52.6
	-	-	✓		48.1	69.2	53.7
	✓	✓	✓		**53.3**	**72.1**	**57.8**
Strategy	✓	✓	✓	Add	52.8	71.3	57.3
	✓	✓	✓	Concat	52.9	71.5	57.4
	✓	✓	✓	Ours	**53.3**	**72.1**	**57.8**

the additional training-time overhead is also kept minimal. Beyond computational efficiency, FSD consistently gains significant accuracy improvements across different model scales.

For lightweight detectors, FSD-N and FSD-S exhibit particularly strong advantages. Among all compared lightweight models, FSD-N contains only 5M trainable parameters, while simultaneously achieving the highest AP of 53.3 and the fastest inference speed. This demonstrates that FSD-N delivers a substantially better accuracy and efficiency trade-off than existing SOTA detectors. FSD-S further improves the AP to 54.9. Remarkably, it even surpasses all M-scale SOTA detectors, while maintaining a low average inference latency of only 2.62 ms. For larger model variants, FSD continues to achieve the best AP performance. Notably, even the largest FSD-X model requires only 5.21 ms for inference, remaining well suited to stringent real-time requirements.

These results demonstrate that coupling the strong semantic representations of the foundation model with the efficient local spatial perception of CNNs provides consistent benefits across model representational capacity, with particularly pronounced gains for lightweight CNN model, further validating the effectiveness and great potential of the proposed method.

4.3 Ablation Study

Comparative experiments demonstrate that FSD achieves state-of-the-art performance; however, the individual contributions of its key components remain unclear. To further investigate their effectiveness, we conducted systematic ablation experiments on COCO. All experiments are performed using the N-scale variant, while the training strategies and settings are kept consistent for a fair comparison. The results are shown in Table 2. introducing the EuPE-assisted dual-branch backbone (DB) brings a substantial improvement over the baseline, increasing AP from 39.4 to 47.5 (+8.1 AP). This result indicates that the high-level semantic information provided by the pretrained VFM can effectively enhance the representation capability of the baseline CNN detector. Building on DB, MSDI further improves AP to 52.6. Moreover, compared with DB alone, MSDI produces disproportionately larger improvements under stricter localization criteria, as reflected by the stronger gain in AP_{75}. This further suggests that MSDI helps translate high-level semantic cues into more precise spatial representations.

Notably, integrating EAH on top of DB alone brings only a relatively modest improvement, suggesting that directly imposing detection supervision on the frozen VFM branch is not sufficient to fully exploit its semantic representations. Without explicit semantic-detail interaction, the detector may still tend to rely more heavily on the readily optimized CNN features, leaving the VFM semantics partially under-utilized. In contrast, when EAH is further combined with MSDI, the gain becomes more pronounced. This indicates that MSDI provides an effective pathway for the detection-oriented VFM semantics encouraged by EAH to interact with fine-grained CNN features, allowing the auxiliary supervision to be better translated into the final detection representation.

4.4 Further Analysis

Comparison of Dense Features. To intuitively compare the feature extraction ability of our method with existing SOTA approaches, we visualize multi-level features from FSD, the CNN-based model YOLO26Jocher et al. (2026), and the Transformer-based model RT-DETRv4Liao et al. (2026), DEIMv2Huang et al. (2025a), as shown in Fig. 3. As observed from the P3 features, CNN-based

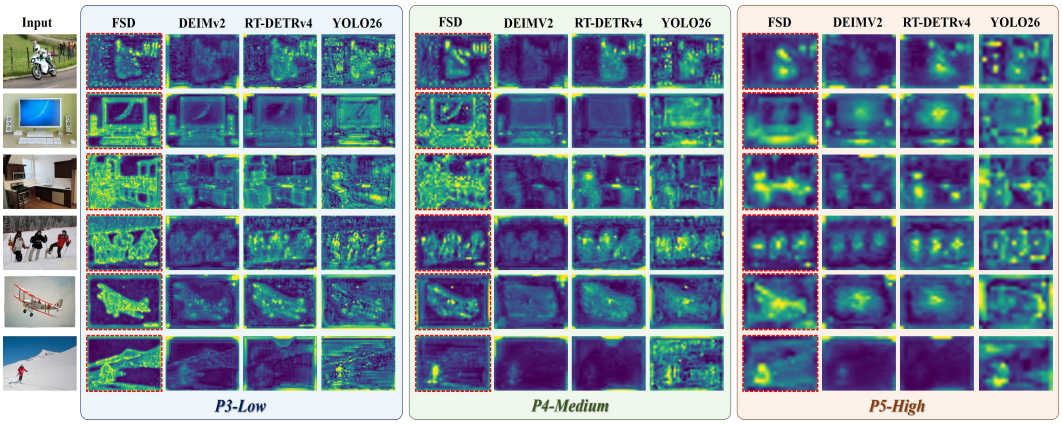


Figure 3: Visualization of multi-level feature from DEIMv2, RT-DETRv4, YOLO26 and FSD. Compared with CNN- and Transformer-based models, FSD better preserves fine-grained spatial details in shallow features while maintaining stronger object-level semantic responses in deeper layers.

detectors YOLO26 shows stronger local responses to textures and object boundaries. However, as the hierarchy deepens, its features become more affected by similar backgrounds, while fine-grained spatial information gradually weakens, leading to less precise semantic responses. As for RT-DETRv4 and DEIMv2, they show stronger global semantic responses in deeper P4 and P5 features, but their shallow representations are relatively less sensitive to local structures and fine boundaries. Meanwhile, excessive feature smoothing can be observed across deeper layers, which may result in incomplete or imprecise object-level semantic responses. In contrast, FSD better suppresses background interference and excessive smoothing, preserves richer local textures and sharper boundary responses in lower-level features, maintains more complete object-level activations and greater sensitivity to spatial structures in deeper layers. Overall, FSD achieves a more favorable balance between fine-grained spatial preservation and semantic aggregation across different feature levels.

Comparison of MSDI Interaction Strategy. Comparative and ablation experiments demonstrate the effectiveness of MSDI. However, features from different levels vary in semantic abstraction and spatial detail; therefore, the selection and fusion of appropriate feature levels require further investigation. To this end, we first conduct a interaction strategy analysis by varying the different fusion layers and method to examine the effects of semantic representations from different depths on detection performance. Specifically, we compare the single-level features E_3, E_4, and E_5 with the complete multi-level representation $\{E_3, E_4, E_5\}$. The results are reported in Table 3. It can be seen that using single-level either shallow or deep EuPE semantic features alone brings only limited improvements, whereas jointly exploiting hierarchical dense representations from different levels yields substantially greater gains. This indicates that features at different depths in EuPE retain complementary spatial and semantic information, validating the necessity of using hierarchical dense VFM features rather than single-level semantic injection.

Furthermore, the comparison of different feature fusion strategies shows that concatenation achieves better performance than element-wise addition, indicating the benefit of preserving complementary information from the semantic and spatial branches. However, direct concatenation lacks explicit alignment between heterogeneous representations, which may lead to representational conflicts between semantic and spatial features and thus limit performance. In contrast, our MSDI block further introduces a C3k module with multi-level gradient information after concatenation to perform soft alignment of heterogeneous features, enabling adaptive adjustment and interaction between the two representations, and consequently achieves the best performance.

VFM Generality Analysis To further investigate the generalization of FSD across different VFM architectures, we evaluate both CNN-based and Transformer-based semantic branches, including SEERGoyal et al. (2021) with RegNetYRadosavovic et al. (2020), DINOv3Siméoni et al. (2025) with ViT and ConvNeXtV2Woo et al. (2023), and EuPEZhu et al. (2026) with ViTDosovitskiy (2020), while keeping YOLO11 as the base detector and all other settings unchanged. As shown in Table 4, both CNN-based and Transformer-based VFMs substantially improve the performance

Table 4: Generality and representation complementarity analysis of VFMs.

Model	Detector	VFM	VFM Arch.	AP(%)	Latency(ms)
FSD-s	YOLO11-s	SEER	RegNetY	51.6	2.34
		DINOv3	ConvNeXtV2	53.5	2.53
		EuPE	**ViT**	**54.9**	2.62
		DINOv3	ViT	54.9	2.97
DEIMv2-s	DEIM-s	None	–	49.0	2.71
		DINOv3	ViT	50.9	3.14

Table 5: Effect of the static weight of EAH.

λ_{aux}	0.0	0.1	0.2	0.3	0.5	0.6	0.8	1.0
AP	52.8	53.0	53.1	**53.3**	**53.3**	53.2	53.0	52.8
AP_{50}	71.5	71.8	71.9	72.0	**72.1**	71.8	71.8	71.6
AP_{75}	57.2	57.2	57.5	57.6	**57.8**	57.7	57.4	57.3

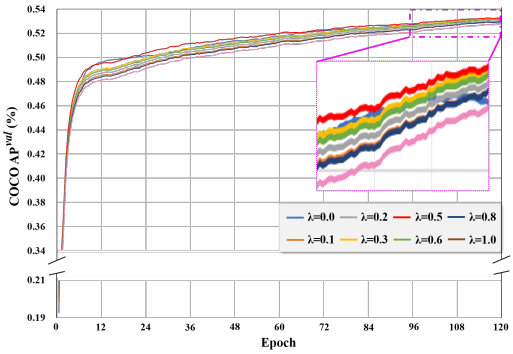


Figure 4: Effect of the auxiliary loss weight λ_{aux} on COCO during training.

of the CNN detector, indicating that VFM representations can effectively enhance its representation and generalization capabilities. Further comparison shows that, under the same CNN detector, Transformer-based VFMs provide larger gains than CNN-based counterparts, suggesting that their stronger global semantic modeling ability more effectively complements the local structures and spatial details preserved by CNNs. To further examine this observation, we apply the same VFM, DINOv3, to both CNN- and Transformer-based detectors. As shown in Table 4, although the original DEIM-S outperforms YOLO11-S, introducing DINOv3 improves DEIM-S by only 1.9 AP. In contrast, the CNN detector gains 8.3 AP and ultimately surpasses the VFM-enhanced Transformer detector by 4.0 AP. These results suggest that stronger VFM representations do not necessarily lead to larger gains for downstream detectors; instead, the observed gains appear to be closely associated with the representational complementarity between the VFM and detector. This observation is further supported by the representation similarity analysis in Appendix B.

Static weights of EAH. The weight λ_{aux} determines the contribution of EAH during joint optimization and therefore affects the balance between the main detection objective and auxiliary semantic supervision. To investigate its influence, we vary λ_{aux} while keeping all other training settings unchanged. As shown in Table 5, introducing auxiliary supervision consistently improves the performance over the setting without EAH. However, an excessively small weight provides insufficient guidance for detection-relevant semantic representation learning, whereas an overly large weight may place excessive emphasis on the auxiliary branch and interfere with the optimization of the fused detection representation. A moderate weight of $\lambda_{\text{aux}} = 0.5$ achieves the best overall performance and is therefore adopted in all experiments. The training AP curves in Fig. 4 further reveal the overall effect of different auxiliary weights on the training process. It can be seen that when the auxiliary semantic loss weight is too small, the semantic guidance provided by EAH is insufficient, resulting in limited performance improvement. In contrast, an excessively large weight may suppress the optimization of spatial-detail representations and lead to inferior convergence. A moderate λ_{aux} achieves a better balance between semantic guidance and spatial-detail learning.

5 Conclusion

In this work, we revisit VFM-enhanced real-time object detection from the perspective of complementary representation learning and propose FSD, a real-time detector that bridges strong foundation semantics with fine-grained CNN spatial details. FSD combines a frozen EuPE semantic branch with a trainable CNN spatial pathway to balance semantic abstraction and fine-grained localization. To further exploit this complementarity, we design MSDI and an auxiliary head to enhance semantic-spatial interaction and detection oriented feature learning. Extensive experiments on COCO demonstrate that FSD consistently improves detection accuracy across different model scales, with particularly pronounced gains for lightweight models. Further analyses suggest that effectively integrating VFMs into object detectors benefits not merely from stronger pretrained representations, but also from exploiting complementary representational characteristics between the VFM and the detector. We believe this perspective provides a promising direction for designing future VFM-enhanced detectors with better accuracy-efficiency trade offs.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing, manuscript organization, and improving the clarity of technical descriptions. They were not used to generate experimental data or replace model training, evaluation, or empirical analysis. All AI-assisted content was carefully reviewed and verified by the authors. The authors take full responsibility for the final content of the paper, including all text, results, and conclusions.

ETHICS STATEMENT

This work focuses on real-time object detection and does not involve the collection of private data, interaction with human subjects, or personally identifiable information. All experiments are conducted on publicly available benchmark datasets following their established usage protocols. The proposed method is intended for general-purpose visual recognition research, and we do not target any harmful or privacy-invasive applications.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the FSD architecture, MSDI module, Efficient Auxiliary Head (EAH), loss functions, and training settings in the main paper. Experimental configurations, and additional implementation details are provided in the corresponding experimental sections and supplementary materials. We also provide the core implementation code and model configurations in the supplementary materials to facilitate reproduction of the reported results.

REFERENCES

Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundation models defining a new era in vision: a survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2245–2264, 2025.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Shenghao Fu, Junkai Yan, Qize Yang, Xihan Wei, Xiaohua Xie, and Wei-Shi Zheng. Frozen-detr: Enhancing detr with image understanding from frozen foundation models. *Advances in Neural Information Processing Systems*, 37:105949–105971, 2024.

Zelin Fu, Bin Tan, Changjiang Sun, Shaohui Liu, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Vision pretraining for dense spatial perception. *arXiv preprint arXiv:2607.05247*, 2026.

Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021.

Guangxing Han and Ser-Nam Lim. Few-shot object detection with foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28608–28618. IEEE, 2024.

Malik Haris, Guoqiang Zhang, Liming Li, Muhammad Shahid Mastoi, Jibran Hussain, and Zhenqing Li. Single-stage object detection in the deep learning era: a critical survey of cnn-, transformer-, and hybrid architectures. *Artificial Intelligence Review*, 2026.

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.

Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia computer science*, 199:1066–1073, 2022.

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.

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

Rawal Khirodkar, He Wen, Julieta Martinez, Yuan Dong, Zhaoen Su, and Shunsuke Saito. Sapiens2. In *International Conference on Learning Representations*, volume 2026, pp. 58484–58507, 2026.

Sunoh Lee, Minsik Jeon, Jihong Min, and Junwon Seo. Ow-rep: Open world object detection with instance representation learning. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 339–349. IEEE, 2026.

Mengqi Lei, Siqi Li, Yihong Wu, Han Hu, You Zhou, Xinhu Zheng, Guiguang Ding, Shaoyi Du, Zongze Wu, and Yue Gao. Yolov13: Real-time object detection with hypergraph-enhanced adaptive visual perception. *arXiv preprint arXiv:2506.17733*, 2025.

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.

Zijun Liao, Yian Zhao, Xin Shan, Yu Yan, Chang Liu, Lei Lu, Xiangyang Ji, and Jie Chen. Rt-detrv4: Painlessly furthering real-time object detection with vision foundation models. In *European Conference on Computer Vision*, pp. 95–113. Springer, 2026.

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.

Longfei Liu, Yongjie Hou, Yang Li, Qirui Wang, Youyang Sha, Yongjun Yu, Yinzhi Wang, Peizhe Ru, Xuanlong Yu, and Xi Shen. Edgecrafter: Compact vits for edge dense prediction via task-specialized distillation. *arXiv preprint arXiv:2603.18739*, 2026.

Jianglin Lu, Hailing Wang, Yi Xu, Yizhou Wang, Kuo Yang, and Yun Fu. Representation potentials of foundation models for multimodal alignment: A survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16680–16695, 2025.

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. *arXiv preprint arXiv:2407.17140*, 2024.

Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3631–3640. Ieee, 2021.

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 *International Conference on Learning Representations*, volume 2025, pp. 44015–44031, 2025.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. *arXiv preprint arXiv:2003.13678*, 2020.

Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. Rf-detr: Neural architecture search for real-time detection transformers. In *International Conference on Learning Representations*, volume 2026, pp. 56087–56100, 2026.

Ranjan Sapkota and Manoj Karkee. Ultralytics yolo evolution: An overview of yolo26, yolo11, yolov8 and yolov5 object detectors for computer vision and pattern recognition. *arXiv preprint arXiv:2510.09653*, 2025.

Ranjan Sapkota, Marco Flores-Calero, Rizwan Qureshi, Chetan Badgujar, Upesh Nepal, Alwin Poulose, Peter Zeno, Uday Bhanu Prakash Vaddevolu, Sheheryar Khan, Maged Shoman, et al. Yolo advances to its genesis: A decadal and comprehensive review of the you only look once (yolo) series. *Artificial Intelligence Review*, 58(9):274, 2025.

Tahira Shehzadi, Khurram Azeem Hashmi, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Object detection with transformers: A review. *Sensors*, 25(19):6025, 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.

Ying Tian, Wenbo Xu, Bo Yang, Xinglong Yang, Hongliang Guo, Gaige Wang, and Helong Yu. Development and evolution of yolo in object detection: A survey. *Neurocomputing*, pp. 132436, 2025.

Shuo Wang, Chunlong Xia, Feng Lv, and Yifeng Shi. Rt-detrv3: Real-time end-to-end object detection with hierarchical dense positive supervision. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1628–1636. IEEE, 2025.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16133–16142. Ieee, 2023.

Huizai Yao, Sicheng Zhao, Pengteng Li, Yi Cui, Shuo Lu, Weiyu Guo, Yunfan Lu, Yijie Xu, and Hui Xiong. Beyond boundaries: Leveraging vision foundation models for source-free object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11866–11874, 2026.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

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.

Chenchen Zhu, Saksham Suri, Cijo Jose, Maxime Oquab, Marc Szafraniec, Wei Wen, Yunyang Xiong, Patrick Labatut, Piotr Bojanowski, Raghuraman Krishnamoorthi, et al. Efficient universal perception encoder. *arXiv preprint arXiv:2603.22387*, 2026.

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.

A Implementation Details

Basic Environment Settings

All experiments were conducted on a computer with Ubuntu 20.04.4 LTS operating system also having an Intel Xeon Gold 6430 CPU, 64G of RAM and an NVIDIA RTX 4090 GPU. The equipped software runtime environment was also set up with Pycharm2024, python 3.8.20, PyTorch 2.1.1, CUDA 12.4 and cuDNN 9.5.0. Since the VFM branch remains frozen throughout training, we report the number of trainable parameters in the main comparison to reflect optimization complexity. Note that the frozen semantic encoder remains part of the inference graph and therefore contributes to the overall model footprint.

Model Hyperparameter Settings

To facilitate reproducibility, we summarize the main training hyperparameters of different FSD variants in Table 6. Following the corresponding ultralytics training protocol, all models are trained with an input resolution of 640×640. The training schedule is adjusted according to model scale, while the remaining optimization settings follow the same configuration unless otherwise specified. We additionally report the major data augmentation settings, including Mosaic, MixUp, CutMix, and random scaling, which have a direct influence on detector optimization.

Table 6: Detailed training hyperparameters of different FSD variants.

HyperParams	**N**	**S**	**M**	**L**	**X**
Resolution	640	640	640	640	640
Batch Size	128	128	96	64	64
Epochs	200	120	90	60	50
Base LR	0.01	0.01	0.01	0.01	0.01
Final LR Ratio	0.01	0.01	0.01	0.01	0.01
Weight Decay	5e-4	5e-4	5e-4	5e-4	5e-4
Momentum	0.937	0.937	0.937	0.937	0.937
Warmup Epochs	5	5	3	3	3
Scale	0.5	0.5	0.5	0.7	0.9
Mosaic$_{prob}$	1.0	1.0	1.0	1.0	1.0
MixUp$_{prob}$	0.0	0.0	0.1	0.2	0.35
CutMix$_{prob}$	0.0	0.0	0.1	0.2	0.35
λ_{aux}	0.5	0.5	0.5	0.5	0.5
AMP	✓	✓	✓	✓	✓

B Representation Similarity Analysis

To further verify that the effectiveness of VFM-enhanced detection is closely related to the representational complementarity between the VFM and downstream detector, we quantitatively analyze the representation similarity across different detector architectures and examine its relationship with the performance gains brought by VFM integration. Specifically, we extract three corresponding feature levels with strides of 8, 16, and 32 from the VFM and detector, and measure their similarity using linear Centered Kernel Alignment (CKA). Given two feature matrices X and Y, linear CKA is defined as

$$\mathrm{CKA}(X, Y) = \frac{\left\|X^\top Y\right\|_F^2}{\left\|X^\top X\right\|_F \left\|Y^\top Y\right\|_F}, \tag{5}$$

where a higher CKA value indicates greater similarity between the two representations.

As shown in Table 7, Transformer-based detectors exhibit substantially higher similarity to the VFM representations, whereas CNN-based detectors maintain lower similarity and obtain larger gains after VFM integration. This consistent relationship provides further empirical evidence that the effectiveness of VFM enhancement is closely associated with representational complementarity. This

Table 7: Representation similarity between the VFM and different detector architectures. ΔAP denotes the gain brought by VFM integration.

Detector	Architecture	CKA$_{P3}$	CKA$_{P4}$	CKA$_{P5}$	Mean CKA	ΔAP
YOLO11-S	CNN	0.25	0.35	0.45	0.35	+8.3
YOLO26-S	CNN	0.24	0.34	0.41	0.33	+5.7
DEIM-S	Transformer	0.62	0.72	0.82	0.72	+1.9
RT-DETRv4-S	Transformer	0.55	0.65	0.75	0.65	+1.6

Table 8: Ablation study of different adapter designs on COCO.

Adapter	AP	AP$_{50}$	AP$_{75}$
Baseline	53.0	71.7	57.5
MLP	53.0	71.8	57.6
1×1 Conv	53.1	72.0	57.6
Linear	**53.3**	**72.1**	**57.8**

relationship is consistent with the hypothesis that lower representational overlap enables pretrained semantic features to provide information that is less redundant with the detector's native representations, thereby facilitating more effective semantic transfer.

C Design of the Adapter.

The adapter serves as a lightweight bridge between the frozen VFM and the detector, projecting multi-level semantic features into a representation space compatible with the CNN features. Since the adapter architecture may affect the preservation and transfer of pretrained semantics, we investigate three common designs with MLP, 1×1 convolution, and linear projection, while keeping all other settings unchanged. As shown in Table 8, the linear projector achieves the best detection performance. Although MLP provides stronger nonlinear transformation capacity, the additional transformation may unnecessarily alter the pretrained semantic representations. The 1×1 convolution performs spatially shared channel projection, but brings no further benefit over the simpler linear mapping. In contrast, the linear adapter provides sufficient channel alignment while better preserving the original VFM semantics. Therefore, we adopt the linear projector as the adapter for all FSD variants.

Cross-Device Inference Latency

To further assess the practical deployment efficiency and hardware robustness of FSD, we benchmark its inference latency on three GPU platforms with substantially different computational capabilities, including NVIDIA RTX 4090, RTX A6000, and RTX 2080 Ti. All models are evaluated at an input resolution of 640×640 with a batch size of 1 using TensorRT FP16 under identical inference settings. In addition to nominal model complexity, we report both trainable and total parameters as well as measured memory consumption, providing a more comprehensive view of the practical cost of FSD. The results are summarized in Table 9.

As shown in Table 9, FSD maintains consistently low inference latency across different GPU architectures. In particular, FSD-N and FSD-S require only 2.34 ms and 2.62 ms on RTX 4090, respectively, while remaining highly efficient even on the substantially older RTX 2080 Ti, with latencies of 3.96 ms and 4.50 ms. The larger variants exhibit a similarly favorable runtime profile: even FSD-X requires only 9.64 ms on RTX 2080 Ti and 5.21 ms on RTX 4090. These results demonstrate that the strong accuracy–latency trade-off of FSD is not restricted to a particular high-end GPU platform.

More importantly, Table 9 reveals a noticeable discrepancy between nominal model complexity and empirical deployment cost. Although the frozen VFM branch substantially increases the total parameter count and GFLOPs, this increase does not translate proportionally into either inference latency or memory consumption. For example, FSD-N contains 91M total parameters and requires 108 GFLOPs, yet achieves 2.34 ms latency on RTX 4090 with only 0.32 GB peak inference memory.

Table 9: Cross-device inference efficiency of FSD. All models are evaluated at an input resolution of 640×640 with a batch size of 1 using TensorRT FP16.

	Model Complexity			Latency (ms)			Peak Training Memory (GB)	Peak Inference Memory (GB)
Model	GFLOPs	Total Params. (M)	Trainable Params. (M)	RTX 2080 Ti	RTX A6000	RTX 4090		
FSD-N	108	91	5	7.96	5.57	2.34	1.9	0.32
FSD-S	117	99	13	8.50	5.94	2.62	2.2	0.32
FSD-M	359	111	25	9.07	7.91	3.39	3.3	0.38
FSD-L	385	119	32	12.98	9.25	4.08	3.7	0.41
FSD-X	517	224	70	13.64	10.33	5.21	8.6	0.49

Even FSD-X, with 517 GFLOPs and 224M total parameters, maintains a latency of 5.21 ms and a peak inference memory of only 0.49 GB.

This discrepancy highlights that FLOPs and parameter count alone are insufficient to characterize the practical efficiency of modern detectors. The additional computation introduced by the frozen semantic branch is dominated by regular dense operations, which can be efficiently parallelized and accelerated by modern GPU hardware. Consequently, despite its increased nominal complexity, FSD retains a low empirical runtime and memory footprint across different devices. These results suggest that FSD trades additional model capacity for substantially stronger semantic representations without proportionally increasing the actual deployment cost.

D Discussion

The central observation of this work is that the effectiveness of VFM-enhanced detection is associated not only with the strength of pretrained representations, but also with their complementarity to the downstream detector. CNN detectors naturally preserve local textures, boundaries, and spatial structures, while semantic-rich VFMs provide stronger global abstraction. Our empirical results show that combining representations with such distinct characteristics can yield larger gains than coupling representations with more similar properties, suggesting that VFM-enhanced detection can be viewed as complementary representation learning rather than simple feature injection. This perspective may also help explain the effectiveness of multi-level semantic-detail interaction. Semantic abstraction and spatial precision vary across feature hierarchies, and effective interaction should preserve the strengths of each representation while selectively exchanging complementary information. Excessive alignment may reduce representation diversity, whereas insufficient interaction may leave useful complementarity under-exploited. More broadly, these observations provide a general perspective for integrating heterogeneous representations. Rather than simply pursuing increasingly similar or uniformly stronger features, effective model design may benefit from preserving and exploiting differences across architectures. Representations with distinct inductive biases, abstraction levels, and spatial sensitivities can provide mutually compensatory information, and their utility may depend on how effectively these differences are aligned and utilized. This suggests that complementary representation learning may offer a broader design principle for combining foundation models with task-specific architectures beyond object detection.

E Recommendations for the Application of the propose FSD

Based on the above experiments and analyses, we provide the following recommendations for applying FSD in different scenarios.

- **Standard model and dataset;** When using the same detector, VFM, and dataset configurations as those in this paper, we recommend directly adopting the hyperparameter settings described in above Section, including the multi-level feature interaction, adapter configuration, and auxiliary loss weight. This setting provides a stable balance between semantic guidance, spatial-detail preservation, and training efficiency, while avoiding unnecessary hyperparameter tuning.
- **New model or dataset;** For a new detector or dataset, the representational complementarity between the VFM and detector should be considered first. In particular, semantic-rich VFMs are more suitable for detectors that preserve strong local spatial details, rather than simply selecting the strongest pretrained model. The interaction levels and auxiliary loss

weight may also be adjusted according to object scale and scene complexity. For datasets dominated by small or cluttered objects, finer-scale semantic-detail interaction should be retained to preserve localization cues, whereas scenes dominated by large objects may rely more on higher-level semantic representations. When a substantial domain gap exists, the adapter should also be re-evaluated to ensure effective alignment between pretrained semantic features and detector-specific representations.

F Limitations and Future Learning

Limitations. Despite the strong accuracy–efficiency trade-off achieved by FSD, several limitations remain. First, although the frozen VFM introduces only moderate practical latency on modern GPUs, it still substantially increases the nominal computation and memory footprint. This may limit deployment on resource-constrained devices where memory bandwidth and parallel computing capability are more restricted. Second, our main experiments are conducted on COCO, and the semantic branch relies on representations learned from large-scale pretraining. Although the VFM generality analysis demonstrates that FSD benefits from different semantic encoders, its robustness under substantial domain shifts, specialized visual domains, and long-tailed or open-world scenarios remains to be further investigated.

Future Learning. Future work will therefore focus on three directions. First, we aim to develop more efficient semantic transfer strategies, such as feature distillation or semantic compression, to progressively transfer VFM knowledge into the CNN branch and potentially remove part or all of the VFM during inference. Second, we will explore adaptive semantic-detail interaction, including dynamic feature-level selection, content-aware fusion, and adaptive auxiliary supervision, allowing the model to adjust the semantic-spatial balance according to object scale and scene complexity. Finally, we plan to extend FSD to broader detection domains and deployment platforms, with particular emphasis on domain adaptation, edge-device efficiency, and robustness under distribution shifts. These directions may further improve the scalability and generalization of complementary foundation-semantic and CNN-spatial representation learning.