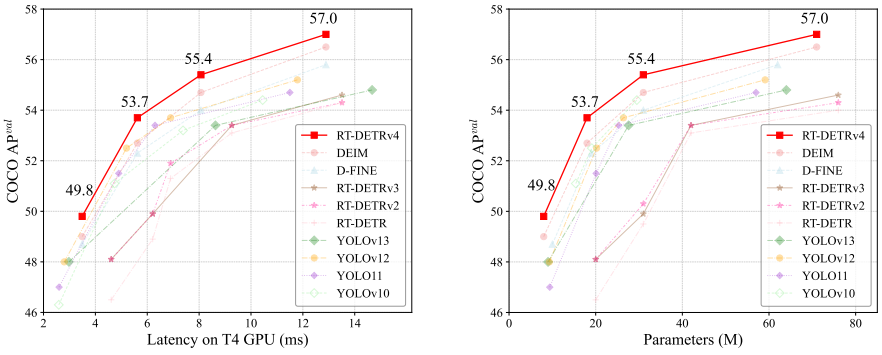
RT-DETRv4: Painlessly Furthering Real-Time Object Detection with Vision Foundation Models

Zijun Liao[1⋆], Yian Zhao[1⋆†], Xin Shan[1], Yu Yan[1], Chang Liu[5✉], Lei Lu[1], Xiangyang Ji[5✉], and Jie Chen[4,2,1,3✉]

[1] School of Electronic and Computer Engineering, Peking University, Shenzhen, China
[2] Pengcheng Laboratory, Shenzhen, China
[3] AI for Science (AI4S)-Preferred Program, Peking University Shenzhen Graduate School, China
[4] School of Intelligence Science and Engineering, Harbin Institute of Technology, Shenzhen, China
[5] Department of Automation and BNRist, Tsinghua University, Beijing, China
jiechen2019@pku.edu.cn



(a) COCO AP^{val} vs. T4 GPU Latency. **(b) COCO AP^{val} vs. Params (M).**

Fig. 1: Compared with advanced real-time object detectors on COCO [21].

Abstract. Real-time object detection has achieved substantial progress through meticulously designed architectures and optimization strategies. However, the pursuit of high-speed inference via lightweight network designs often leads to degraded feature representation, which hinders further performance improvements and practical on-device deployment. In this paper, we propose a cost-effective and highly adaptable distillation framework that harnesses the rapidly evolving capabilities of Vision Foundation Models (VFMs) to enhance lightweight object detectors. Given the significant architectural and learning objective disparities between VFMs and resource-constrained detectors, achieving stable and task-aligned semantic transfer is challenging. To address this, on one hand, we introduce a **Deep Semantic Injector (DSI)** module that facilitates the integration of high-level representations from VFMs into the deep layers of the detector. On the other hand, we devise a **Gradient-guided Adaptive Modulation (GAM)** strategy, which dynamically adjusts the intensity of semantic transfer based on gradient norm ratios. Without

⋆ Equal contribution. †Project leader. ✉ Corresponding authors.

increasing deployment and inference overhead, our approach painlessly delivers striking and consistent performance gains across diverse DETR-based models, underscoring its practical utility for real-time detection. Our new model family, RT-DETRv4, achieves state-of-the-art results on COCO, attaining AP scores of 49.8/53.7/55.4/57.0 at corresponding speeds of 273/169/124/78 FPS. Code is publicly available at `https://github.com/RT-DETRs/RT-DETRv4`.

Keywords: Real-time object detection · Representation learning

1 Introduction

Real-time object detection stands as a fundamental task in computer vision, which underpins numerous interactive and safety-critical applications that demand instant perception and decision making, such as autonomous driving [7], embodied intelligence [22], and human–computer interaction [17, 29, 43]. Over the past decade, remarkable progress has been driven by efficient network architectures and end-to-end learning frameworks. In particular, two representative series, YOLO [30] and DETR [4], have profoundly influenced the evolution of object detection paradigms. The YOLO family emphasizes rapid one-stage detection achieving high inference speed and practical deployment efficiency, and the DETR series has reshaped the detection paradigm through its unified modeling of object queries and set-based prediction. Among its variants, RT-DETR [44] marked a milestone as the first real-time DETR, introducing the DETR family to the real-time community by outperforming YOLO models in both speed and accuracy.

Despite the notable progress, a long-standing challenge remains: the inherent trade-off between designing lightweight models to achieve high inference speed and employing complex architectures to improve feature representation. To meet real-time constraints, detectors typically adopt lightweight backbones and carefully designed computational modules, which inevitably reduce their ability to capture high-level semantics and lead to a semantic bottleneck. This limitation not only hinders further performance improvement, but also increases the difficulty of practical on-device deployment.

In this paper, inspired by the rapid advances in Vision Foundation Models (VFMs) [14, 33], we propose a cost-effective and highly adaptable distillation framework that leverages the powerful representational capacity of VFMs to enhance lightweight object detectors. By transferring the rich semantics of VFMs to real-time detectors during training while keeping the detector architecture unchanged during inference, our method enables significant enhancement without introducing any additional inference or deployment cost. This advantage is particularly important for practical real-time detection applications.

However, achieving stable and task-aligned semantic transfer is challenging due to the large architectural and learning objective disparities between VFMs and resource-constrained detectors. To address this issue, we first introduce a Deep Semantic Injector (DSI) module that enables the integration of high-level representations from VFMs into the deep layers of the detector. To ensure

stable and efficient optimization, we further design a Gradient-guided Adaptive Modulation (GAM) strategy that dynamically adjusts the strength of semantic injection based on gradient norm ratios, thereby harmonizing the learning of semantic transfer and detection objectives.

Extensive experiments show that the proposed method achieves significant and consistent performance improvements over advanced DETR-based detectors without increasing inference or deployment overhead, underscoring its effectiveness. In summary, our main contributions are as follows:

- We propose a cost-effective and highly adaptable distillation framework that leverages the evolving capabilities of VFMs to painlessly enhance real-time detectors, providing a scalable pathway for transferring foundation-level semantics to lightweight architectures.
- We propose the Deep Semantic Injector (DSI) and Gradient-guided Adaptive Modulation (GAM), which enable stable and task-aligned semantic transfer between VFMs and detectors with significantly different architectures and learning objectives.
- We establish a new family of models, RT-DETRv4-S/M/L/X, achieving 49.8/53.7/55.4/57.0 AP scores on COCO [21] at 273/169/124/78 FPS, setting a new SOTA on COCO dataset.

2 Related work

2.1 Real-time object detection

The evolution of real-time object detection has long been driven by the You Only Look Once (YOLO) family [30], which popularized the single-stage paradigm through an efficient and unified detection pipeline. Over the past few years, this lineage has undergone rapid iteration, introducing continuous refinements in backbone design, label assignment, and optimization strategy [3,11,12,19,31,32,37,38]. Recent generations have expanded the design space even further: YOLOv10 [36] eliminated NMS that the YOLO series has long relied on, YOLO11 [13] improved the architectural hierarchy and neck connectivity, YOLOv12 [34] incorporated attention mechanisms for better contextual reasoning, and YOLOv13 [18] explored hypergraph representations to capture higher-order feature dependencies. These advances have pushed the performance–efficiency frontier of convolutional and hybrid architectures, gradually narrowing the gap between real-time and high-accuracy detectors.

In parallel, another line of research has evolved around the DEtection TRansformer (DETR) [4], which redefined object detection as a set prediction problem and eliminated hand-crafted components such as anchor design and NMS. This transformer-based paradigm inspired numerous variants, including Deformable DETR [45], Conditional DETR [25], and DAB-DETR [23], which focus on improving convergence and localization accuracy. Later works such as DN-DETR [20], DINO [42], and Group-DETR [8] introduced denoising objectives and group-wise supervision to further enhance training stability and representational quality.

Building on this foundation, RT-DETR [44] established the first real-time end-to-end transformer detector that achieved parity with, and in some cases surpassed, contemporary YOLO models. Subsequent works have continued to improve its training efficiency and representation learning without incurring inference overhead. For instance, RT-DETRv2 [24] and RT-DETRv3 [40] incorporated auxiliary supervision for enhanced gradient flow, D-FINE [27] employed self-distillation to refine semantic representation, and DEIM [16] introduced dense matching for more precise feature alignment. Collectively, these developments illustrate a clear trend: as architectural efficiency saturates, training supervision and semantic representation become the primary levers for further progress. Our work builds on this insight by strengthening the core representation via deep semantic transfer, achieving higher accuracy at no additional deployment or inference cost.

2.2 Knowledge distillation for object detection

Knowledge distillation has been widely explored to improve object detectors under efficiency constraints. FGD [41] transfers focal local knowledge and global relational information by modeling the uneven importance of regions and channels. DETRDistill [6] adapts distillation to DETRs via Hungarian-matching logits, target-aware feature, and query-prior assignments. CrossKD [39] forwards the student's head features through the teacher head for more task-aligned prediction distillation. These studies show that customized supervision can enhance detector representations without additional inference cost. However, most methods rely on teachers trained for the target detection task, whereas Vision Foundation Models offer more general semantic priors, motivating our use of a frozen VFM as an external teacher.

2.3 Vision Foundation Models

Vision Foundation Models (VFMs) have become a dominant paradigm for learning general-purpose vision representations from large-scale image corpora with minimal or no human supervision. Early progress stemmed from self- and weakly-supervised learning methods, which enabled models to capture high-level semantics from unlabeled or loosely labeled data. Representative approaches include contrastive learning frameworks such as SimCLR [9] and MoCo [15], which learn discriminative features by enforcing consistency across augmented views of the same image while contrasting them with others. CLIP [28] further extended this idea to large-scale image–text contrastive training, aligning visual and linguistic embeddings and demonstrating strong zero-shot transferability across diverse tasks.

Inspired by masked language modeling in NLP, Masked Image Modeling (MIM) approaches were introduced to reconstruct masked image regions, thereby learning context-aware and holistic representations. Notable methods include MAE [14] and BEiT [2]. In parallel, I-JEPA [1] learns semantic image representations with a joint embedding predictive objective, predicting target block representations from

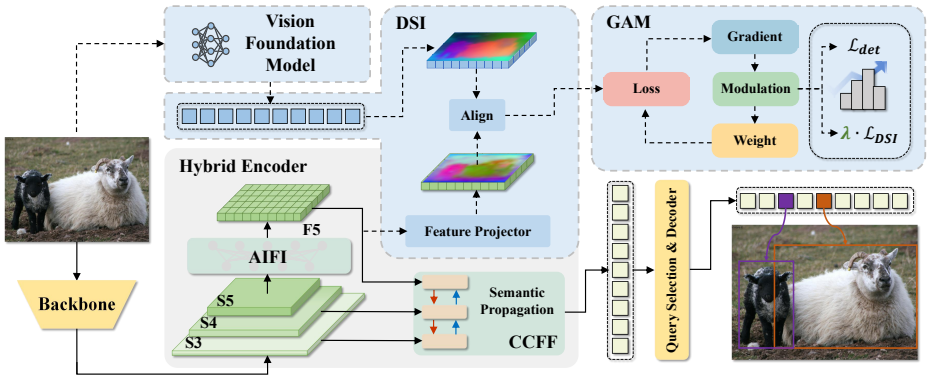


Fig. 2: Overview of RT-DETRv4. We leverage a Vision Foundation Model (VFM) to extract high-quality semantic representations, which are aligned with the deepest feature map (F_5) from the AIFI module via a Feature Projector in the Deep Semantic Injector (DSI). To ensure faster and more stable convergence, a Gradient-guided Adaptive Modulation (GAM) dynamically adjusts the DSI loss during training. The proposed framework operates only during the training phase (dashed arrows and blue blocks) of the real-time detector and keeps the original architecture unchanged during inference and deployment, introducing no additional overhead while improving accuracy.

a context block. The DINO family [5,26,33] integrates contrastive, reconstruction-based, and self-distillation objectives to produce highly semantic and transferable features. In particular, DINOv3 [33] demonstrates the scalability and efficacy of large-scale self-supervised learning, achieving rich and robust representations without human annotations.

3 Method

3.1 Motivation

Information Bottleneck for detection representations. According to the Information Bottleneck (IB) principle [35], optimal representations should preserve task-relevant semantics while discarding nuisance from the input. Let X denote the input image, Y denote the detection target, and $I(\cdot;\cdot)$ denote mutual information; optimal representations R_ℓ on the ℓ-th scale should minimize

$$\mathcal{L}_{IB} = I(X; R_\ell) - \beta I(R_\ell; Y), \tag{1}$$

where β controls the trade-off between the compression of the input space and the predictive accuracy for the target domain. However, almost all detectors are optimized solely through task-driven supervision, i.e., by maximizing predictive mutual information $I(R_\ell; Y)$ via back-propagation from object detection loss:

$$\mathcal{L}_{det} \sim -I(R_\ell; Y), \tag{2}$$

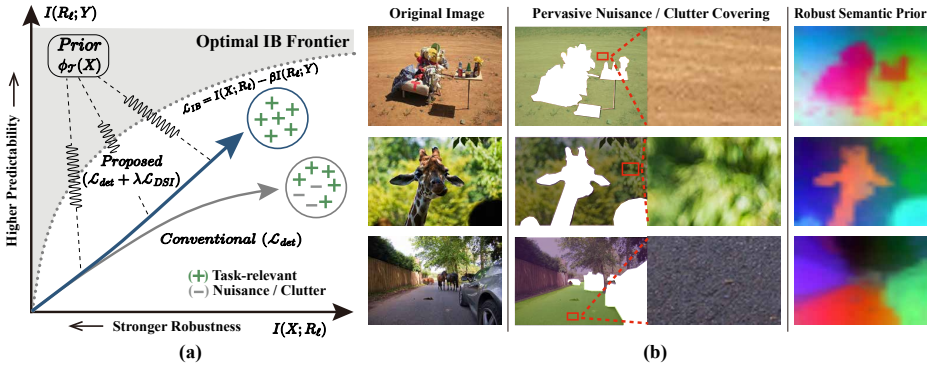


Fig. 3: Semantic injection for object detection. (a) Illustration from the information bottleneck perspective. (b) Visualization of nuisance details in images versus robust semantic representations from a VFM.

as indicated by the gray curve in Fig. 3(a). This training paradigm implicitly encourages representations that are discriminative for Y, but lacking an explicit constraint on $I(X;R_\ell)$. There is no direct mechanism to suppress non-robust components inherited from X, such as background clutter or texture-level patterns that are weakly correlated with object semantics, as illustrated in Fig. 3(b).

Semantic supervision from VFMs. Under the IB principle in Eq. (1), a desirable representation should preserve task-relevant information while discarding nuisance factors. To this end, we introduce semantic supervision from a pretrained VFM, which empirically serves as a strong and robust semantic prior that can be viewed as semantically more compressed, emphasizing high-level object semantics while being less sensitive to low-level nuisance factors:

$$T = \phi_\mathcal{T}(X), \quad I(X;T) \lesssim I(X;R_\ell)\,, \tag{3}$$

Such representations are more robust to appearance changes and background noise. We therefore introduce semantic supervision during detector training:

$$\mathcal{L} = \mathcal{L}_{det} + \lambda \mathcal{L}(R_\ell, T). \tag{4}$$

as indicated by the blue curve in Fig. 3(a). The detection loss encourages task-specific discrimination, while the semantic objective aligns detector features with robust high-level semantics. The coupled objective consequently improves detection performance. Meanwhile, robust features benefit model generalization and help reduce overfitting.

Principle of Effective Semantic Injection. Most modern detectors follow the *Backbone–Neck–Head* architecture, where the feature extraction process can be modeled as

$$X \xrightarrow{\text{Backbone}} \{S_\ell\} \xrightarrow{\text{Neck}} \{F_\ell\} \xrightarrow{\text{Head}} \hat{Y}, \quad R_\ell \in \{S_\ell, F_\ell\}. \tag{5}$$

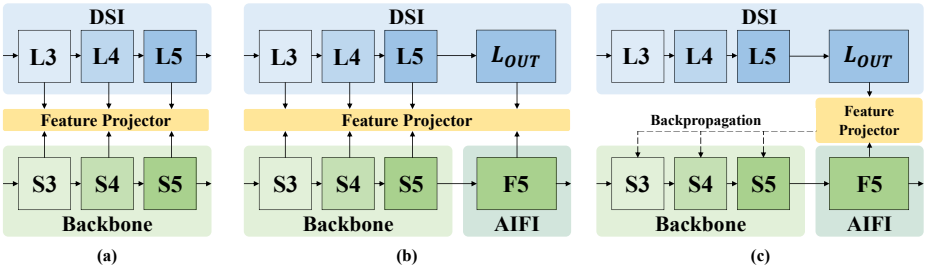


Fig. 4: Illustration of different Deep Semantic Injector (DSI) strategies. (a) Direct alignment of multi-scale backbone features (S_3, S_4, S_5). (b) Hybrid alignment of both backbone and the AIFI features (S_3, S_4, S_5, F_5). (c) Ours: Targeted alignment of only the AIFI feature (F_5), which possesses the highest-level semantics. This design allows gradients to backpropagate, enhancing both the AIFI module and the backbone.

According to the Data Processing Inequality,

$$I(X; S_1) \geq I(X; S_2) \geq \cdots \geq I(X; F_1) \geq I(X; F_2) \geq \cdots . \tag{6}$$

Thus deeper layers naturally perform progressive information compression and should encode increasingly abstract semantics. Consequently, injecting semantic supervision at relatively deeper stages is theoretically preferable.

3.2 Overview

We focus on applying our framework to DETR-based real-time object detectors, *i.e.*, RT-DETR models [44]. The overall framework of our method is shown in Fig. 2, where the proposed modules are highlighted in blue. Given multi-scale backbone features (S_3, S_4, S_5), RT-DETR uses self-attention only on the deepest map S_5 via the Attention-based Intra-scale Feature Interaction (AIFI), producing an enhanced deep feature F_5. Then CNN-based Cross-scale Feature Fusion (CCFF) propagates information from F_5 to obtain multi-scale outputs for the decoder.

The design of the hybrid encoder makes the quality of the feature map F_5 particularly critical. As the only feature subjected to self-attention, F_5 serves as the principal source of high-level, global semantic information for the entire model. Its quality directly affects the subsequent cross-scale fusion in the CCFF module, the initial query selection, and ultimately the performance of the decoder. However, the AIFI module that produces F_5 is trained with only indirect supervision, as the gradients from the final detection losses must backpropagate through the decoder and CCFF before reaching F_5. Such indirect supervision may be insufficient to fully optimize F_5. This leads to the major *Semantic Bottleneck* in RT-DETR. Therefore, we prioritize performing semantic injection on F_5, while also considering alternative designs involving other feature levels and mixed injection strategies, as illustrated in Fig. 4.

Based on the discussion in Sec. 3.1, we propose the **Deep Semantic Injector (DSI)**, a lightweight training-only module that explicitly aligns the deep feature with semantically rich representations from a VFM. This targeted supervision enhances the semantic expressiveness of high-level features and allows the gradient to flow back through AIFI and the backbone, improving both modules synergistically. With DSI incorporated, the total training objective is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{det}} + \lambda \mathcal{L}_{\text{DSI}}, \tag{7}$$

where $\mathcal{L}_{\text{det}}$ denotes the standard detection loss (e.g., classification and bounding box regression), and $\mathcal{L}_{\text{DSI}}$ represents the proposed semantic alignment loss. However, achieving stable and task-aligned semantic transfer is challenging because of the large architectural and learning objective disparities between VFMs and resource-constrained detectors. An inappropriate choice of λ may either provide insufficient semantic supervision in the early stages or excessively dominate the detection objective in later stages, ultimately impeding convergence and degrading performance. To adapt to the evolving optimization dynamics during training, we propose **Gradient-guided Adaptive Modulation (GAM)**, a mechanism that dynamically tunes λ based on gradient statistics, ensuring balanced optimization between detection and semantic supervision.

3.3 Deep Semantic Injector

To address the *Semantic Bottleneck*, we introduce the Deep Semantic Injector (DSI), a training-only module designed to provide explicit and powerful semantic supervision of R_ℓ by aligning it with representations from a high-capacity semantic teacher, denoted as $\mathcal{T}$. Given an input image X, $\mathcal{T}$ produces a high-quality feature representation $\phi_{\mathcal{T}}(X)$.

Feature projector. To align the detector's feature map $R_\ell \in \mathbb{R}^{H_\ell \times W_\ell \times C_k}$ with the teacher's representation, differences in both spatial resolution and channel dimensionality must be reconciled. The teacher, typically a ViT [10], outputs a sequence of patch tokens $T_p \in \mathbb{R}^{N_p \times C_\mathcal{T}}$. To enable spatial comparison, T_p is reshaped into a 2D grid representation $T_{sp} \in \mathbb{R}^{H_\mathcal{T} \times W_\mathcal{T} \times C_\mathcal{T}}$, where $N_p = H_\mathcal{T} \times W_\mathcal{T}$. We then introduce a lightweight feature projector $\mathcal{P}$ to achieve twofold alignment. First, T_{sp} is interpolated to match the spatial resolution of R_ℓ. Meanwhile, $\mathcal{P}$ adjusts the channel dimension of R_ℓ to align with the teacher's semantic space. The complete projection process is summarized as follows:

$$T_{sp} = \text{Reshape}(T_p), \quad T'_{sp} = \text{Interpolate}(T_{sp}), \quad R'_\ell = \mathcal{P}(R_\ell). \tag{8}$$

Alignment loss. To encourage the detector to capture the rich semantics of the teacher's representation, we adopt a cosine similarity loss. We maximize the patch-wise cosine similarity between the detector's projected features R'_ℓ and the teacher's projected features T'_{sp}, formulated as minimizing the negative cosine similarity averaged over all spatial locations (i, j):

$$\mathcal{L}_{DSI}(R'_\ell, T'_{sp}) = -\frac{1}{H_\ell W_\ell} \sum_{i,j} \frac{R'_\ell(i,j) \cdot T'_{sp}(i,j)}{\|R'_\ell(i,j)\| \|T'_{sp}(i,j)\|}. \tag{9}$$

Semantic injection. As illustrated in Fig. 4, we design three progressively enhanced configurations for semantic injection. In configurations (a) and (b), the DSI module aligns features at different hierarchical depths via the Feature Projector, injecting semantic knowledge from the frozen VFM into the detector's feature hierarchy. In configuration (c), given the pivotal role of the AIFI module in the hybrid encoder, alignment is applied to its output F_5, which contains the richest semantics. Without gradient detachment, the DSI loss backpropagates to update the lightweight backbone, while the forward pass leverages the enriched F_5 to guide cross-scale fusion in the CCFF module. This dual-directional supervision unifies semantic enhancement across the detector. The comparative results of different configurations (*cf.* Sec. 4.2) demonstrate that configuration (c) achieves the best performance.

3.4 Gradient-guided Adaptive Modulation

To ensure stable and adaptive semantic supervision, we propose a dynamic Gradient-guided Adaptive Modulation (GAM) mechanism that regulates the DSI loss weight (λ) based on the relative gradient contribution of the target module of semantic injection according to its *gradient norm ratio*. This gradient-based regulation adaptively maintains balanced optimization.

Specifically, for each training step t within epoch e, we compute the L_1 norm of gradients for each component $\mathcal{C}$:

$$G_t^{(\mathcal{C})} = \|\nabla_{\theta_\mathcal{C}} \mathcal{L}_{total}\|_1. \tag{10}$$

The total gradient magnitude is given by:

$$G_t^{(total)} = \sum_{\mathcal{C}} G_t^{(\mathcal{C})}, \tag{11}$$

and the relative gradient contribution of the target module is defined as:

$$r_t = \frac{G_t^{(\text{target})}}{G_t^{(total)}}. \tag{12}$$

Then average the gradient ratios across all training steps within epoch e to obtain:

$$\bar{r}_e = \frac{1}{T_e} \sum_{t=1}^{T_e} r_t, \tag{13}$$

where T_e denotes the number of steps in epoch e.

At the end of each epoch, GAM checks whether $\bar{r}_e$ lies within the target interval. If $\bar{r}_e \in [\rho-\delta, \rho+\delta]$, the weight λ_e of $\mathcal{L}_{\text{DSI}}$ remains unchanged. Otherwise, λ_e is adjusted such that the next epoch's gradient ratio of the target module of semantic injection is steered toward the *further boundary* of the target range rather than its midpoint, since only a portion of gradients originates from $\mathcal{L}_{\text{DSI}}$, boundary-based adjustment yields more stable convergence near equilibrium.

$$\lambda_{e+1} = \begin{cases} \lambda_e \cdot \dfrac{\rho - \delta}{\bar{r}_e}, & \text{if } \bar{r}_e > \rho + \delta, \\ \lambda_e \cdot \dfrac{\rho + \delta}{\bar{r}_e}, & \text{if } \bar{r}_e < \rho - \delta, \\ \lambda_e, & \text{otherwise.} \end{cases} \tag{14}$$

Two hyperparameters govern the modulation process: the target ratio (ρ) defines the desired supervision intensity, whereas the tolerance interval (δ) regulates the trade-off between responsiveness and stability. This update rule drives the effective gradient contribution of semantic injection to converge within the desired operational range while preventing oscillations. In practice, GAM provides stable convergence and consistently improves semantic alignment without additional tuning overhead.

4 Experiments

4.1 Comparison with SOTA

We compare our proposed RT-DETRv4 with recent state-of-the-art real-time detectors, including the latest YOLO series (YOLOv10 [36], YOLO11 [13], YOLOv12 [34], and YOLOv13 [18]) and DETR-based detectors (RT-DETR [44], RT-DETRv2 [24], RT-DETRv3 [40], D-FINE [27], and DEIM [16]). The results are illustrated in Fig. 1, and detailed statistics are provided in Tab. 1. The results demonstrate that RT-DETRv4 consistently achieves the best performance across all model scales (S, M, L, and X) without introducing any extra inference and deployment overhead.

Specifically, our **RT-DETRv4-L** achieves **55.4 AP** on COCO at **124 FPS**, outperforming YOLOv13-L (53.4 AP) and DEIM-L (54.7 AP) under comparable or even lower computational budgets. The largest variant, **RT-DETRv4-X**, reaches **57.0 AP**, exceeding DEIM-X (56.5 AP) without introducing any inference overhead. At smaller scales, **RT-DETRv4-S** and **RT-DETRv4-M** obtain **49.8** and **53.7 AP**, clearly surpassing their DEIM counterparts (49.0 and 52.7 AP).

4.2 Ablation study

We conduct a series of ablation experiments to verify the effectiveness of the proposed modules. Unless stated otherwise, all ablation experiments are trained for **36 epochs**. Unspecified hyperparameters or configurations follow the best settings for the corresponding experimental setup.

Feature visualization. Fig. 5 visually compares the feature maps between DEIM-L and our RT-DETRv4-L. Notably, our model enriches the semantic content of the AIFI feature F_5, leading to more precise and distinguishable object contours and backgrounds across the subsequent multi-scale features P_3, P_4, and P_5. In particular, P_5 exhibits a markedly stronger and more concentrated response to object regions.

Table 1: Comparison with real-time object detectors on COCO [21] `val2017`. Results are sourced from the official publications. Unreported values evaluated from public weights using the standard protocol are marked with *. R18, R34, R50, and R101 denote ResNet-18, ResNet-34, ResNet-50, and ResNet-101, respectively.

Model	#Epochs	#Params.	GFLOPs	Latency (ms)	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
YOLOv10-S [36]	500	7	22	2.52	46.3	63.0	50.4	26.8	51.0	63.8
YOLO11-S [13]	500	9	22	2.60	47.0	63.4*	50.5*	-	-	-
YOLOv12-S [34]	600	9	21	2.78	48.0	65.0	51.8	29.8	53.2	65.6
YOLOv13-S [18]	600	9	21	2.98	48.0	65.2	52.0	-	-	-
RT-DETR-R18 [44]	120	20	60	4.61	46.5	63.8	50.4	28.4	49.8	63.0
RT-DETRv2-S [24]	120	20	60	4.61	48.1	65.1	52.1*	30.2*	51.2*	64.2*
RT-DETRv3-R18 [40]	120	20	60	4.61	48.1	65.6	52.0*	30.2*	51.5*	63.9*
D-FINE-S [27]	120	10	25	3.66	48.5	65.6	52.6	29.1	52.2	65.4
DEIM-S [16]	120	10	25	3.66	49.0	65.9	53.1	30.4	52.6	65.7
RT-DETRv4-S (ours)	120	10	25	3.66	**49.8**	**67.1**	**54.0**	**30.6**	**53.4**	**67.3**
YOLOv10-M [36]	500	15	59	4.74	51.1	68.1	55.8	33.8	56.5	67.0
YOLO11-M [13]	500	20	68	4.85	51.5	68.1*	55.8*	-	-	-
YOLOv12-M [34]	600	20	68	4.96	52.5	69.6	57.1	**35.7**	**58.2**	68.8
RT-DETR-R34 [44]	120	31	92	6.91	48.9	66.8	52.9	30.6	52.4	66.3
RT-DETRv2-M [24]	120	31	92	6.91	49.9	67.5	54.1*	32.0*	53.2*	66.5*
RT-DETRv3-R34 [40]	120	31	92	6.91	49.9	67.7	53.9*	31.7*	54.0*	66.2*
D-FINE-M [27]	120	19	57	5.91	52.3	69.8	56.4	33.2	56.5	70.2
DEIM-M [16]	90	19	57	5.91	52.7	70.0	57.3	35.3	56.7	69.5
RT-DETRv4-M (ours)	90	19	57	5.91	**53.7**	**71.0**	**58.4**	35.6	57.4	**71.2**
YOLOv10-L [36]	500	24	120	7.38	53.2	70.1	58.1	35.8	58.5	69.4
YOLO11-L [13]	500	25	87	6.33	53.4	69.7*	58.3*	-	-	-
YOLOv12-L [34]	600	27	89	6.85	53.7	70.7	58.5	36.9	59.5	69.9
YOLOv13-L [18]	600	28	88	8.63	53.4	70.9	58.1	-	-	-
RT-DETR-R50 [44]	72	42	136	9.29	53.1	71.3	57.7	34.8	58.0	70.0
RT-DETRv2-L [24]	72	42	136	9.29	53.4	71.6	57.4*	36.1*	57.9*	70.8*
RT-DETRv3-R50 [40]	120	42	136	9.29	53.4	71.7	57.3*	35.4*	57.4*	69.8*
D-FINE-L [27]	72	31	91	8.07	54.0	71.6	58.4	36.5	58.0	71.9
DEIM-L [16]	50	31	91	8.07	54.7	72.4	59.4	36.9	59.6	71.8
RT-DETRv4-L (ours)	50	31	91	8.07	**55.4**	**73.0**	**60.3**	**37.1**	**60.1**	**72.9**
YOLOv10-X [36]	500	30	160	10.45	54.4	71.3	59.3	37.0	59.8	70.9
YOLO11-X [13]	500	57	195	11.50	54.7	71.3*	59.7*	-	-	-
YOLOv12-X [34]	600	59	199	11.80	55.2	72.0	60.2	**39.6**	60.7	70.9
YOLOv13-X [18]	600	64	199	14.67	54.8	72.0	59.8	-	-	-
RT-DETR-R101 [44]	72	76	259	13.88	54.3	72.7	58.6	36.0	58.8	72.1
RT-DETRv2-X [24]	72	76	259	13.88	54.3	72.8	58.8*	35.8*	58.8*	72.1*
RT-DETRv3-R101 [40]	120	76	259	13.88	54.6	73.1	-	-	-	-
D-FINE-X [27]	72	62	202	12.90	55.8	73.7	60.2	37.3	60.5	73.4
DEIM-X [16]	50	62	202	12.90	56.5	74.0	61.5	38.8	61.4	74.2
RT-DETRv4-X (ours)	50	62	202	12.90	**57.0**	**74.6**	**62.1**	39.5	**61.9**	**74.8**

Ablation on DSI and GAM. We first assess the effectiveness of DSI and GAM. As shown in Tab. 2, applying DSI can bring a slight performance improvement, while further applying GAM can significantly improve the performance gain (0.5 AP), which fully proves the effectiveness of both. To verify the general applicability of our method, we also conduct experiments on RT-DETRv2 and D-FINE, and the results show that our method can bring consistent performance gains to different detectors.

Ablation on semantic injection position. To validate the choice of injection position, we compare the strategies shown in Fig. 4. Results in Tab. 3 indicate that directly applying semantic supervision to backbone features (S_3, S_4, or S_5)

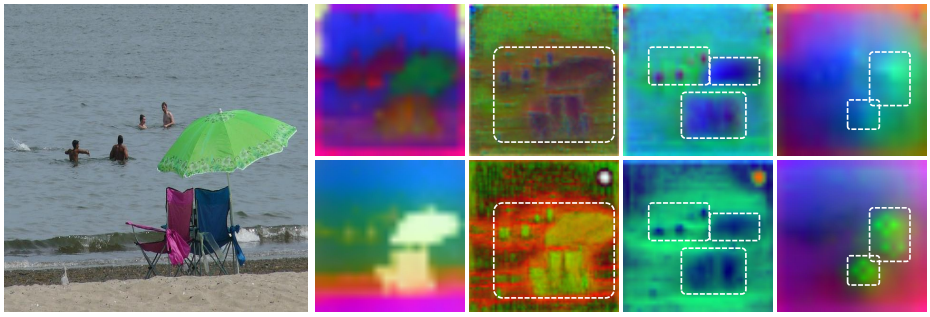


Fig. 5: Comparison of dense features. We compare the feature map quality of DEIM-L (top) and RT-DETRv4-L (bottom) by projecting dense outputs to RGB space using PCA. The visualization reveals that our DSI module substantially enhances the semantic representation of deep features. From left to right: input image, AIFI feature map F_5, and CCFF features P_3, P_4, P_5.

Table 2: Results of ablation on DSI and GAM across multiple detectors. Our method brings consistent and significant gains with zero additional inference cost.

Method	$\mathbf{AP}^{val}$	$\mathbf{AP}^{val}_{50}$	$\mathbf{AP}^{val}_{75}$
RT-DETRv2-L	52.1	70.2	56.7
w/ DSI	52.3 (+0.2)	70.4 (+0.2)	56.4 (-0.3)
w/ DSI+GAM	**52.6 (+0.5)**	**70.7 (+0.5)**	**56.8 (+0.1)**
D-FINE-L	53.1	70.8	57.4
w/ DSI	53.2 (+0.1)	70.8 (+0)	57.7 (+0.3)
w/ DSI+GAM	**53.4 (+0.3)**	**71.1 (+0.3)**	**58.0 (+0.6)**
DEIM-L	53.8	71.4	58.5
w/ DSI	53.9 (+0.1)	71.3 (-0.1)	58.8 (+0.3)
w/ DSI+GAM	**54.3 (+0.5)**	**71.8 (+0.4)**	**59.0 (+0.5)**

Table 3: Results of ablation on semantic injection position. We compare the effectiveness of applying DSI at different positions of **DEIM-L**, corresponding to the strategies in Figure 4. Aligning only the AIFI output (F_5) yields the best performance.

Position	S_3	S_4	S_5	F_5	$\mathbf{AP}^{val}$	$\mathbf{AP}^{val}_{50}$	$\mathbf{AP}^{val}_{75}$
Baseline	-	-	-	-	53.8	71.4	58.5
Backbone	✓	-	-	-	53.7	71.2	58.4
	-	✓	-	-	53.7	71.3	58.4
	-	-	✓	-	53.8	71.3	58.5
	✓	✓	✓	-	53.7	71.3	58.5
Hybrid	✓	✓	✓	✓	53.8	71.4	58.4
AIFI	-	-	-	✓	**54.3**	**71.8**	**59.0**

Table 4: Results of ablation on the feature projector.

Projector Arch.	AP^{val}	AP^{val}_{50}	AP^{val}_{75}
DEIM-L	53.8	71.4	58.5
w/ 1x1 Conv	53.8	71.5	58.5
w/ MLP	54.2	71.7	59.0
w/ Linear	**54.3** (+0.5)	**71.8** (+0.4)	**59.0** (+0.5)

Table 5: Results of ablation on the semantic teacher.

Semantic Teacher	AP^{val}	AP^{val}_{50}	AP^{val}_{75}
DEIM-L	53.8	71.4	58.5
w/ MAE-ViT-B	53.9	71.4	58.6
w/ DINOv2-ViT-Base	54.1	71.6	58.8
w/ DINOv3-ViT-Base	**54.3** (+0.5)	**71.9** (+0.5)	**59.0** (+0.5)

Table 6: Results of ablation on the alignment loss. The cosine similarity loss demonstrates superior performance. DEIM-M is adopted as the baseline, with all models trained for 90 epochs and 12 EMA epochs.

Loss Function	AP^{val}	AP^{val}_{50}	AP^{val}_{75}
DEIM-M	52.5	69.9	57.2
w/ MSE Loss	52.7	70.0	57.4
w/ Cosine Similarity	**53.7** (+1.2)	**71.0** (+1.1)	**58.4** (+1.2)

Table 7: Ablation on the loss weighting strategy. GAM consistently outperforms the static weighting. DEIM-L is adopted as the baseline, with all models trained for 50 epochs and 8 EMA epochs.

λ	0.1	0.2	0.4	1	2	4	10	**20**	30	50	100	**GAM**
AP^{val}	54.7	54.7	54.8	54.9	54.9	55.0	55.0	**55.1**	55.0	54.9	54.6	**55.4**

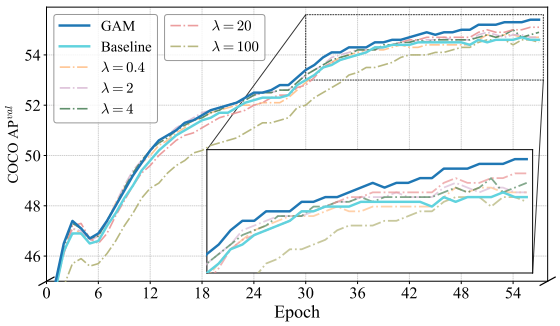


Fig. 6: AP^{val} evolution on COCO during training.

individually or jointly yields no improvement. Similarly, the hybrid approach (strategy (b)) that aligns both backbone and F_5 features provides no gain.

In contrast, our design (strategy (c)), which aligns only the AIFI output F_5, achieves a clear 0.5 AP improvement (54.3 AP). This demonstrates that maintaining richer semantics in F_5 is crucial for enhancing detection performance, as it plays a key role in propagating high-level semantic information to subsequent feature hierarchies. Furthermore, the ineffectiveness of the hybrid approach suggests that simultaneously aligning features from the CNN-based backbone and the Transformer-based AIFI may introduce optimization conflicts or semantic misalignment. Our chosen strategy is not only more effective but also more efficient. It avoids the complexity of multiple intermediate projections and interpolations, and the gradient from the single F_5 alignment loss naturally flows back to synergistically update both AIFI and the backbone, achieving consistent enhancement with a single, targeted objective.

Design of the feature projector. The feature projector is a crucial component for bridging the student and teacher feature spaces. We explore several architectural choices, as detailed in Tab. 4. A linear-based projector yields the best results, striking an optimal balance between expressive power and parameter efficiency.

Choice of semantic teacher. Tab. 5 shows that DINOv3-ViT-Base outperforms DINOv2-ViT-Base and MAE-ViT-B as the semantic teacher, confirming that models with stronger semantic priors acquired from large-scale self-supervised learning serve better in this role. Moreover, our method remains agnostic to the type of VFMs, allowing detectors to continuously benefit from future advances in pretrained VFMs, providing a flexible and training-friendly solution.

Ablation on loss function. We further study the alignment loss of DSI. We compare the Mean Squared Error (MSE) loss and the cosine similarity loss in Tab. 6, the cosine similarity loss surpasses MSE.

Comparison of GAM and static weights. Finally, we validate the proposed GAM against static weights. The superiority of GAM in navigating the training dynamics is illustrated in Fig. 6, showing AP^{val} over training epochs. As shown, GAM consistently surpasses the baseline and all static weight configurations, demonstrating a clear and stable performance advantage throughout training.

Tab. 7 details the results for static weight and GAM. For static weighting, performance peaks at $\lambda = 20$, but degrades with either smaller or larger values. However, observing the training curve in Fig. 6, we find that even this optimal static weight ($\lambda = 20$) leads to slower convergence in the early-to-mid stages, highlighting the inherent limitations of a fixed hyperparameter. Our experiments indicate that GAM achieves the best performance (55.4 AP).

5 Discussion

To further highlight the advantages of our framework over existing methods [16, 18, 27, 34], we discuss it from three perspectives: deployment efficiency, scalability, and training efficiency.

Deployment Efficiency. Our method introduces zero modification to detector architectures and does not alter the inference pipeline, ensuring that no additional computational cost or latency is incurred. This deployment-friendly property is crucial for industrial applications, where real-time detectors are tightly integrated into existing systems and hardware-constrained environments.
Scalability. The framework is general and can be applied seamlessly to detectors with diverse architectures, including CNN-based and transformer-based detectors. It enables all types of real-time detectors to quickly benefit from the rapid progress of Vision Foundation Models (VFMs). Moreover, the framework is not restricted to any specific type or scale of VFM. It can flexibly incorporate different foundation models, such as DINOv3 [33], MAE [14], or CLIP [28], and even benefit from arbitrarily large models for distilling semantics into real-time detectors. This flexibility also opens up promising directions for multi-VFM semantic integration, further demonstrating the framework's generality and scalability.
Training Efficiency. Our approach is lightweight and easy to implement. Since neither the detector structure nor the optimization pipeline is modified, the additional training cost remains minimal. Meanwhile, as VFMs continue to advance, the semantic teacher can be updated with stronger pretrained models while maintaining the same training cost. This efficiency highlights the practicality of our method for large-scale applications and real-time industrial deployment.

6 Conclusion

In this work, we present a cost-effective and adaptable semantic distillation framework that enhances real-time DETR-based object detectors without increasing inference or deployment overhead. Through the proposed Deep Semantic Injector (DSI) and Gradient-guided Adaptive Modulation (GAM), our method effectively transfers high-level semantics from Vision Foundation Models to lightweight detectors in a stable and task-aligned manner. Extensive experiments on COCO demonstrate consistent and significant performance gains across multiple model scales, culminating in the state-of-the-art RT-DETRv4 series. These results highlight the effectiveness of explicit semantic supervision in bridging the gap between large-scale foundation models and resource-efficient detection architectures. Overall, our work provides a practical pathway toward unlocking the potential of foundation models for efficient visual perception.

Acknowledgments

This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0122702), the Shenzhen Medical Research Funds in China (No. B2302037), Natural Science Foundation of China (No. 61972217, 32071459, 62176249, 62006133, 62271465, 62406167), and AI for Science (AI4S)-Preferred Program, Peking University Shenzhen Graduate School, China, and the Shenzhen Loop Area Institute under grant FPF10120250014.

References

1. Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15619–15629 (2023)
2. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=p-BhZSz59o4
3. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
6. Chang, J., Wang, S., Xu, H.M., Chen, Z., Yang, C., Zhao, F.: Detrdistill: A universal knowledge distillation framework for detr-families. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6898–6908 (2023)
7. Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(12), 10164–10183 (2024)
8. Chen, Q., Chen, X., Wang, J., Zhang, S., Yao, K., Feng, H., Han, J., Ding, E., Zeng, G., Wang, J.: Group detr: Fast detr training with group-wise one-to-many assignment. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6633–6642 (2023)
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PmLR (2020)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=YicbFdNTTy
11. Glenn., J.: Yolov5 release v7.0. https://github.com/ultralytics/yolov5/tree/v7.0 (2022), accessed: 30 June 2026
12. Glenn., J.: Yolov8. https://docs.ultralytics.com/models/yolov8/ (2023), accessed: 30 June 2026
13. Glenn., J.: Yolo11. https://docs.ultralytics.com/models/yolo11/ (2024), accessed: 30 June 2026
14. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
15. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
16. Huang, S., Lu, Z., Cun, X., Yu, Y., Zhou, X., Shen, X.: Deim: Detr with improved matching for fast convergence. In: Proceedings of the computer vision and pattern recognition conference. pp. 15162–15171 (2025)

17. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
18. Lei, M., Li, S., Wu, Y., Hu, H., Zhou, Y., Zheng, X., Ding, G., Du, S., Wu, Z., Gao, Y.: Yolov13: Real-time object detection with hypergraph-enhanced adaptive visual perception. arXiv preprint arXiv:2506.17733 (2025)
19. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., et al.: Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976 (2022)
20. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: 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)
21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)
22. Liu, H., Liu, X., Huang, K., Guo, D.: Embodied intelligence. In: Embodied Multi-Agent Systems: Perception, Action, and Learning, pp. 3–48. Springer (2025)
23. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: DAB-DETR: Dynamic anchor boxes are better queries for DETR. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=oMI9PjOb9Jl
24. Lv, W., Zhao, Y., Chang, Q., Huang, K., Wang, G., Liu, Y.: Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer. arXiv preprint arXiv:2407.17140 (2024)
25. Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J.: Conditional detr for fast training convergence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3651–3660 (2021)
26. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
27. Peng, Y., Li, H., Wu, P., Zhang, Y., Sun, X., Wu, F.: D-FINE: Redefine regression task of DETRs as fine-grained distribution refinement. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=MFZjrTFE7h
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
29. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollar, P., Feichtenhofer, C.: SAM 2: Segment anything in images and videos. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=Ha6RTeWMd0
30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
31. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7263–7271 (2017)
32. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)

33. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025)
34. Tian, Y., Ye, Q., Doermann, D.: Yolov12: Attention-centric real-time object detectors. Advances in neural information processing systems **38**, 78433–78457 (2025)
35. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv preprint physics/0004057 (2000)
36. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: Yolov10: Real-time end-to-end object detection. Advances in neural information processing systems **37**, 107984–108011 (2024)
37. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7464–7475 (2023)
38. Wang, C.Y., Yeh, I.H., Mark Liao, H.Y.: Yolov9: Learning what you want to learn using programmable gradient information. In: European conference on computer vision. pp. 1–21. Springer (2024)
39. Wang, J., Chen, Y., Zheng, Z., Li, X., Cheng, M.M., Hou, Q.: Crosskd: Cross-head knowledge distillation for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16520–16530 (2024)
40. Wang, S., Xia, C., Lv, F., Shi, Y.: 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)
41. Yang, Z., Li, Z., Jiang, X., Gong, Y., Yuan, Z., Zhao, D., Yuan, C.: Focal and global knowledge distillation for detectors. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4643–4652 (2022)
42. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., Shum, H.Y.: DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=3mRwyG5one
43. Zhao, Y., Li, K., Cheng, Z., Qiao, P., Zheng, X., Ji, R., Liu, C., Yuan, L., Chen, J.: Graco: Granularity-controllable interactive segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3501–3510 (2024)
44. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J.: Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16965–16974 (2024)
45. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=gZ9hCDWe6ke