Electronic Letters on Computer Vision and Image Analysis 25(2):116-129, 2026

EnYOLO-World: Improved YOLOv9 and FiLM-Driven PAN for Object Detection

Ly Vireak Dara, Rongfang Wang* and Jiaxuan Guo

School of Artificial Intelligence, Xidian University, Xi'an, Shaanxi, 710126, China

**Corresponding author: rfwang@xidian.edu.cn*

Received 18th of October 2025; accepted 18th of June 2026

Abstract

Achieving competitive accuracy without sacrificing inference speed remains a central goal in object detection research. Current YOLO-based architectures face two recurring bottlenecks: progressive loss of spatial detail as features traverse deeper backbone stages, and rigid feature aggregation strategies that fail to adapt to varying scene complexity. This paper introduces **EnYOLO-World**, a detection framework addressing both limitations simultaneously. Our approach augments the YOLOv9 GELAN backbone with **Programmable Gradient Information (PGI)** modules that maintain fine-grained representations throughout the network hierarchy. Furthermore, we propose the **FiLM-Driven Path Aggregation Network (FiLM-PAN)**, which leverages semantic embeddings to perform channel-wise affine recalibration during multiscale feature fusion, operating at linear computational complexity. Evaluation on COCO val2017 shows that EnYOLO-World attains 52.2 $AP_{50:95}$, exceeding YOLO-World-L by +1.2 AP and GLIP-T by +0.8 AP, while using $4.2\times$ fewer parameters than the latter. These findings confirm that jointly optimizing backbone preservation and neck-level adaptation yields favorable accuracy-efficiency characteristics for practical deployment.

Key Words: Object Detection, Enhanced YOLO-World, Feature-wise Linear Modulation, Path Aggregation Network, Programmable Gradient Information.

1 Introduction

Localizing and classifying objects within images constitutes a cornerstone problem in visual understanding, with direct implications for autonomous navigation [1], standardized evaluation through benchmarks such as PASCAL VOC [2], and security monitoring systems [3]. The central difficulty involves simultaneously maximizing detection precision and minimizing computational latency, especially when operating under strict resource or timing constraints.

Significant strides have been made over the past decade. Infrastructure projects including Detectron2 [4], MMDetection [5], and MMYOLO [6] have standardized experimentation and facilitated reproducible comparisons. Meanwhile, one-stage architectures exemplified by SSD [7] and successive YOLO iterations [3] have

Correspondence to: rfwang@xidian.edu.cn
Recommended for acceptance by Angel D. Sappa
https://doi.org/10.5565/rev/elcvia.2441
ELCVIA ISSN: 15108-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

emphasized throughput without entirely compromising precision. Transformer-based formulations, starting with DETR [8], demonstrated that attention mechanisms could replace hand-crafted components. Subsequent YOLO releases—YOLOv4 [9], YOLOv5 [10], YOLOv7 [11], and YOLOv8 [12]—progressively refined network topology and optimization protocols. Most notably, YOLOv9 [13] introduced the GELAN backbone coupled with Programmable Gradient Information (PGI), substantially improving multiscale feature retention and training stability.

Despite this progress, two persistent obstacles hamper current detectors. The first concerns *backbone-level feature attenuation*: architectures relying on Cross-Stage Partial (CSP) connections, including YOLOv8, tend to dilute high-frequency spatial cues as depth increases, thereby impairing recall for compact or partially visible targets. Analogous trade-offs appear in efficiency-oriented designs like EfficientDet [14] and MobileNetV3 [15]. The second obstacle involves *static multiscale aggregation*: although Path Aggregation Networks (PANs) [16] facilitate bidirectional information exchange across resolutions, their fixed fusion weights cannot accommodate scene-dependent variations, limiting robustness when semantic and geometric cues conflict.

To overcome these limitations, we propose EnYOLO-World, an enhanced detection framework that combines a stronger backbone with a redesigned fusion neck. While prior works have addressed backbone feature preservation (e.g., YOLOv9's PGI) and neck-level fusion (e.g., standard PAN) independently, no existing real-time detector jointly optimizes both components through adaptive, semantically guided recalibration. EnYOLO-World fills this gap by building on YOLOv9's GELAN backbone with PGI [13] for robust multiscale preservation and introducing the FiLM-Driven Path Aggregation Network (FiLM-PAN), which dynamically recalibrates features during fusion. Inspired by feature-wise linear modulation [17], FiLM-PAN injects lightweight channel-wise affine transformations into PAN blocks, enabling adaptive cross-scale calibration. This design maintains global receptive field coverage with linear complexity ($O(N)$), ensuring precise yet efficient feature alignment.

Our main contributions are summarized as follows:

- **Backbone enhancement:** Leverage YOLOv9's GELAN backbone with PGI for improved feature preservation and stable gradient propagation.
- **New feature fusion neck:** Propose FiLM-PAN to adaptively recalibrate multiscale features with minimal computational overhead.
- **Improved trade-off on COCO:** On the COCO dataset [18], our EnYOLO-World achieves higher AP than YOLO-World with only a limited increase in parameters and FLOPs, while surpassing heavier detectors in parameter efficiency.

2 Related Work

2.1 Object Detection

The YOLO lineage has shaped much of the discourse on efficient visual recognition. Beginning with the seminal single-shot formulation [3], each subsequent iteration has pushed accuracy higher while preserving or reducing latency. YOLOv4 [9] consolidated various optimization tricks into a unified recipe; YOLOv7 [11] added learnable augmentation policies that improved generalization at no inference cost; YOLOv8 [12] modernized the codebase and introduced a modular CSP backbone; YOLOv9 [13] brought Programmable Gradient Information (PGI) within the GELAN topology, yielding richer multiscale representations and smoother convergence; and, most recently, YOLOv10 [24] removed the reliance on non-maximum suppression through consistent dual-assignment training, further reducing end-to-end inference latency.

Concurrently, attention-centric paradigms have gained traction. DETR [8] eliminated heuristic post-processing by framing detection as set prediction, while refinements such as DINO [19] accelerated training through denoising objectives and improved anchor initialization. These transformer-based solutions excel in modeling long-range dependencies but typically incur greater compute budgets than convolutional counterparts.

Two unresolved challenges motivate the present work: (i) CSP-style backbones, despite their efficiency, progressively erode fine-grained cues needed for small-object recall [13, 14]—a degradation reflected in our ablation study, where removing PGI reduces AP by 0.9 points (Table 2, AS2 vs. Baseline); and (ii) conventional PAN modules [16] aggregate features through predetermined pathways that lack content-adaptive weighting. Open-world benchmarks like ODinW [25] further underscore the need for architectures capable of generalizing beyond fixed category sets.

2.2 Backbone Design for Feature Preservation

Feature extraction networks largely determine the quality and diversity of representations available to downstream heads. Earlier YOLO backbones adopted CSP blocks to curtail redundant computation, yet this choice often attenuates high-frequency texture information in later stages. YOLOv8's deep CSP stack [12], for instance, can suppress low-level activations critical for recognizing small or partially occluded instances. Similar compromises arise in mobile-oriented detectors such as EfficientDet [14] and MobileNetV3 [15], where parameter budgets constrain representational capacity. Hierarchical vision transformers like Swin [21] expand receptive fields through shifted windows but impose heavier memory and latency footprints.

YOLOv9 [13] mitigates feature erosion via Programmable Gradient Information (PGI), which installs auxiliary pathways that steer gradients toward under-represented channels during backpropagation. Networks equipped with PGI exhibit more uniform gradient magnitudes and preserve richer multiscale detail. Nevertheless, the interplay between PGI-enhanced backbones and adaptive neck modules remains underexplored, presenting an opportunity to further boost detection fidelity.

2.3 Feature Recalibration via Feature-Wise Linear Modulation

Adaptive feature transformation constitutes a complementary avenue for improving detector performance. Feature-wise Linear Modulation (FiLM) [17] exemplifies this strategy by learning per-channel scaling (γ) and shifting (β) coefficients conditioned on an external signal. Originally devised for visual question answering, FiLM enables networks to modulate intermediate representations according to linguistic or semantic context—a property well suited to detection tasks that benefit from category-level guidance. Crucially, FiLM achieves this flexibility through element-wise affine operations, avoiding the quadratic cost inherent to self-attention and keeping inference overhead minimal.

Standard detection necks typically rely on fixed convolutional or pooling operations to merge features. Attention variants offer global receptive fields but scale quadratically with spatial resolution. FiLM's linear complexity ($O(N)$) renders it an attractive alternative for injecting adaptivity into fusion stages. Vision–language distillation methods such as ViLD [22] reinforce the value of semantic conditioning, yet FiLM-based modulation within real-time detection necks remains largely unexplored.

2.4 Path Aggregation Networks in Detection

Bidirectional feature pyramids have become a de facto component in modern one-stage detectors. The Path Aggregation Network (PAN) [16] augments top-down pathways with an additional bottom-up branch, enabling

low-level localization cues to reach higher semantic layers and vice versa. Variants of this design appear in YOLOv4 [9], YOLOv7 [11], and YOLOv8 [12], attesting to its broad utility.

However, conventional PAN implementations employ static operations—concatenation, element-wise addition, and fixed convolutions—that treat all inputs uniformly regardless of scene content. Such rigidity limits adaptability in cluttered environments where certain scales or channels carry more discriminative information than others. Research on lightweight conditioning [17] indicates that learned affine transformations can dynamically emphasize salient features without incurring substantial overhead. Building on this insight, we design a FiLM-augmented PAN that marries cross-scale aggregation with content-adaptive recalibration, yielding a neck architecture that is both efficient and responsive to varying detection contexts.

3 Methodology

EnYOLO-World extends the YOLOv9 blueprint [13] by pairing a PGI-enhanced GELAN backbone—responsible for extracting multiscale visual representations—with a newly designed FiLM-Augmented Path Aggregation Network (FiLM-PAN) that performs adaptive feature fusion. The term *FiLM-PAN* denotes our neck module that embeds Feature-wise Linear Modulation layers within the standard PAN topology, enabling dynamic channel-wise scaling and shifting during cross-resolution merging. Dual detection heads follow the fused pyramid: one head predicts object categories while another regresses bounding-box coordinates (Fig. 1). This architecture retains the low-latency characteristics associated with YOLO detectors while strengthening both feature fidelity and fusion flexibility, ultimately improving localization across diverse object scales and scene complexities.

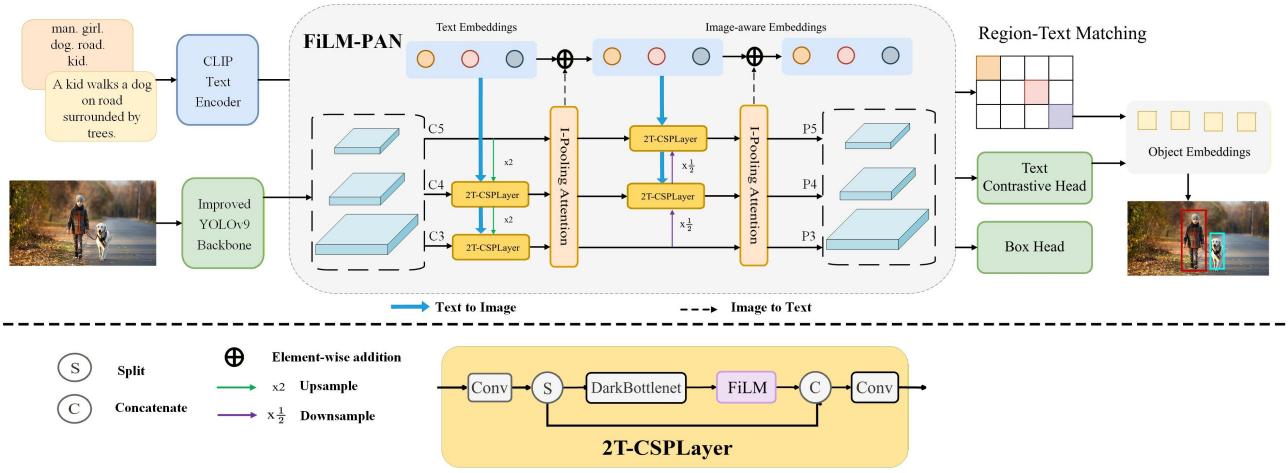


Figure 1: Overall architecture of our EnYOLO-World, consisting of a PGI-improved YOLOv9 backbone, frozen CLIP text encoder, FiLM-Augmented Path Aggregation Network (FiLM-PAN), and dual detection heads for object detection. Feature maps C_3, C_4, C_5 from the backbone are fed into FiLM-PAN, which produces refined pyramid features P_3, P_4, P_5 for detection.

As illustrated in Fig. 1, input images of size $640 \times 640 \times 3$ are processed by the backbone to produce three-scale feature maps: $C_3 \in \mathbb{R}^{80\times80\times256}$ (stride 8, capturing fine-grained details), $C_4 \in \mathbb{R}^{40\times40\times512}$ (stride 16), and $C_5 \in \mathbb{R}^{20\times20\times1024}$ (stride 32, encoding high-level semantics). Simultaneously, the frozen CLIP text encoder transforms category descriptions into 512-dimensional embeddings. The FiLM-PAN fuses these multiscale visual features with textual guidance, outputting refined pyramid features P_3, P_4, P_5 that preserve spatial resolution while incorporating semantic alignment. Each P_l is then passed to the corresponding detection head for bounding-box regression and classification.

3.1 Improved YOLOv9 Backbone

Fig. 2 depicts our backbone, which builds upon the GELAN design of YOLOv9 and incorporates Programmable Gradient Information (PGI) modules at each stage. The topology comprises stacked 3×3 convolutions for initial downsampling, a sequence of *ELAN* blocks that aggregate multiple internal feature paths, and a concluding *SPPELAN* module that applies spatial pyramid pooling to capture context at varying receptive-field sizes. In the neck, RepNCSP-ELAN4 blocks handle cross-scale fusion, while parallel detection layers in the head perform localization and classification.

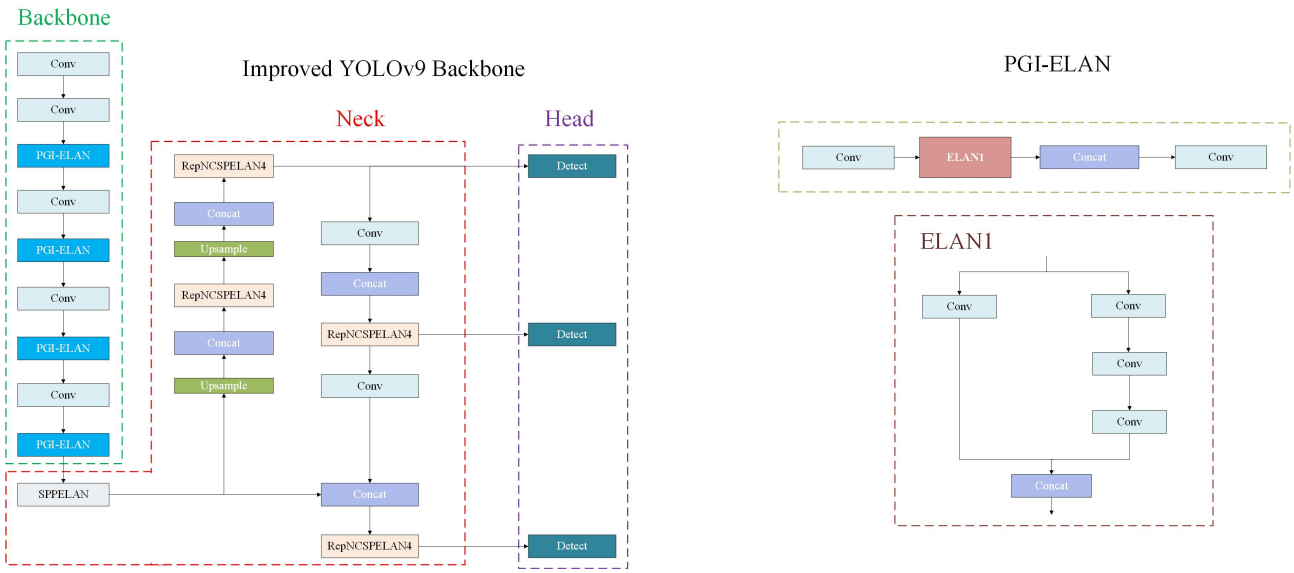


Figure 2: Overall architecture of the PGI-improved YOLOv9 backbone. The backbone extracts hierarchical features at three scales, while the neck performs top-down and bottom-up fusion to produce detection outputs P_3, P_4, P_5.

As shown in Fig. 2, the backbone processes input images through progressive downsampling stages. After initial convolutions reduce the spatial resolution to 320×320, the first PGI-ELAN block outputs features at 160×160 with 128 channels. Subsequent PGI-ELAN blocks further downsample to produce $C_3 \in \mathbb{R}^{80\times80\times256}$, $C_4 \in \mathbb{R}^{40\times40\times512}$, and after SPPELAN, $C_5 \in \mathbb{R}^{20\times20\times1024}$. The neck (red dashed region) receives these multiscale features and applies RepNCSPELAN4 blocks with upsample/concat operations to generate the final pyramid outputs P_3, P_4, P_5, which are fed to respective detection heads (teal boxes) for multi-scale object detection.

Programmable Gradient Information (PGI) modules reside within each ELAN block to bolster multiscale feature retention. During backpropagation, learned per-channel gates direct gradient flow, preserving both fine-grained textures and abstract semantics. PGI tackles the information bottleneck inherent in deep networks by introducing auxiliary reversible branches that maintain complete input fidelity. These branches supply reliable gradient signals that guide the primary pathway toward more discriminative representations. Channels conveying task-relevant patterns receive amplified gradients, whereas noisy or redundant activations are suppressed—thereby countering the vanishing-gradient phenomenon that typically weakens small-object features in later layers. Forward-pass residual connections further merge features across depths, ensuring that cues for compact objects persist even at the network's deepest stages.

Overall, the PGI-augmented backbone sustains computational efficiency while mitigating the feature degradation common in earlier YOLO variants, establishing a robust foundation for the adaptive fusion performed by

our proposed FiLM-PAN.

3.2 FiLM-Augmented Path Aggregation Network

The central component of Fig. 1 illustrates our *FiLM-Augmented PAN*, which supersedes the conventional Path Aggregation Network with a semantically guided fusion mechanism. Frozen CLIP text embeddings (upper branch) and multiscale backbone outputs (lower branch) interact through bidirectional *Text-to-Image* and *Image-to-Text* pathways: the former generates modulation coefficients that recalibrate visual activations, while the latter aggregates visual statistics to refine textual representations. This reciprocal design injects category-level semantics directly into the feature aggregation process.

The fusion pipeline comprises three components:

3.2.1 Textual Affine State Generation

A frozen CLIP encoder generates a fixed-length text embedding $T \in \mathbb{R}^D$ from class prompts or natural-language descriptions. For each feature scale, two lightweight MLPs map T into channel-wise gain $\gamma_l \in \mathbb{R}^{D_l}$ and bias $\beta_l \in \mathbb{R}^{D_l}$. Because T, γ_l, and β_l are resolution- and text-length independent, parameter generation incurs $O(1)$ cost per scale, preserving the overall fusion complexity at $O(N)$, where N denotes the number of image tokens.

3.2.2 Text-Guided CSPLayer (2T-CSPLayer)

As shown in the yellow block of Fig. 1, each conventional PAN fusion block is replaced with a *2T-CSPLayer*. The input feature map $X_l \in \mathbb{R}^{H\times W\times D_l}$ is divided into two branches:

- **Main branch:** Processes features through a *DarkBottleneck*, then applies FiLM modulation as defined in Eq. (1).

$$V'_{h,w,c} = \gamma_l^c \cdot V_{h,w,c} + \beta_l^c, \quad \forall (h, w, c). \tag{1}$$

- **Bypass branch:** Propagates features unchanged. The two branches are concatenated and fused by a 1×1 convolution, injecting semantic guidance into each fusion stage while maintaining PAN's efficient data flow.

In Eq. (1), $\gamma_l, \beta_l \in \mathbb{R}^{C_l}$ are channel-wise vectors, where C_l is the number of channels at scale l. They are generated independently of batch size B by scale-specific MLPs that map the fixed-length text embedding $T \in \mathbb{R}^D$ to $\mathbb{R}^{C_l}$. Each scalar γ_l^c and β_l^c is broadcast uniformly across all spatial positions (h, w) and all batch samples, i.e., $V'_{b,h,w,c} = \gamma_l^c \cdot V_{b,h,w,c} + \beta_l^c,\ \forall\, b \in \{1, \ldots, B\}$, ensuring that modulation is semantically driven rather than instance- or position-specific. The learned gain γ_l^c controls the importance of channel c at scale l, amplifying features relevant to the target categories while suppressing irrelevant activations. The bias β_l^c shifts the feature distribution to better align with the semantic space. Together, these parameters enable scale-specific and category-aware feature recalibration: for instance, when detecting small objects, the modulation can emphasize fine-grained spatial details in shallow layers (P_3), while for large objects it can strengthen semantic features in deeper layers (P_5). This adaptive mechanism allows the network to dynamically adjust its feature representations based on both the input content and the target categories.

3.2.3 Three-Stage Fusion Flow

The FiLM-PAN operates through three sequential $O(N)$ stages:

- **Initial Text-to-Image Calibration:** Each backbone output C_l is modulated by FiLM prior to spatial merging.
- **Multiscale Aggregation:** Calibrated features are aggregated via top-down and bottom-up PAN pathways, employing upsampling, concatenation, and 2T-CSPLayer fusion to construct pyramid levels P_3, P_4, and P_5.
- **Final Text-to-Image Refinement:** After aggregation, a second FiLM modulation refines P_3–P_5 with new affine parameters, preserving text–visual alignment across scales.

Through this integration, the FiLM-PAN retains the efficiency of conventional PAN while introducing adaptive text-aware recalibration. This combination enhances semantic alignment and cross-scale consistency, yielding more precise and robust object detection.

3.2.4 Comparison with Other Adaptive Fusion Mechanisms

It is instructive to compare FiLM-PAN with existing adaptive fusion modules such as Squeeze-and-Excitation (SE) [26], CBAM [27], and BiFPN [14]. While all these mechanisms aim to improve feature recalibration, they differ fundamentally in their conditioning strategy and modulation design.

SE and CBAM are *self-conditioned* modules that derive attention weights solely from the input features themselves through global pooling operations. SE learns channel-wise multiplicative weights via a bottleneck MLP, while CBAM extends this with an additional spatial attention branch. However, neither incorporates external semantic guidance, limiting their ability to adapt to category-specific detection requirements.

BiFPN introduces learnable scalar weights for multi-scale feature fusion, enabling the network to balance contributions from different pyramid levels. While computationally efficient ($O(N)$), BiFPN's scalar weighting lacks the expressiveness of channel-wise modulation and remains category-agnostic.

In contrast, FiLM-PAN is *externally conditioned* on CLIP text embeddings, enabling category-aware feature recalibration. The affine transformation ($\gamma \cdot x + \beta$) provides richer modulation than purely multiplicative approaches (SE) or scalar weighting (BiFPN), as the bias term β can shift feature distributions to better align with semantic targets. Furthermore, because the conditioning signal derives from a pretrained vision-language model, FiLM-PAN inherits semantic priors that generalize across object categories without requiring category-specific training. This design philosophy—lightweight complexity ($O(N)$) combined with semantically guided adaptation—distinguishes FiLM-PAN from prior attention-based fusion mechanisms.

Algorithm 1 FiLM-PAN Forward Pass

Require: Backbone features $\{C_3, C_4, C_5\}$, Text embedding T
Ensure: Fused pyramid features $\{P_3, P_4, P_5\}$
// Stage 1: Generate FiLM parameters
for $l \in \{3, 4, 5\}$ **do**
  $\gamma_l \leftarrow \text{MLP}^l_\gamma(T)$ {Channel-wise gain}
  $\beta_l \leftarrow \text{MLP}^l_\beta(T)$ {Channel-wise bias}
end for
// Stage 2: Initial Text-to-Image Calibration
for $l \in \{3, 4, 5\}$ **do**
  $\tilde{C}_l \leftarrow \gamma_l \odot C_l + \beta_l$ {FiLM modulation}
end for
// Stage 3: Top-down pathway
$P_5 \leftarrow \text{2T-CSPLayer}(\tilde{C}_5, \gamma_5, \beta_5)$
$P_4 \leftarrow \text{2T-CSPLayer}(\text{Concat}(\text{Upsample}(P_5), \tilde{C}_4), \gamma_4, \beta_4)$
$P_3 \leftarrow \text{2T-CSPLayer}(\text{Concat}(\text{Upsample}(P_4), \tilde{C}_3), \gamma_3, \beta_3)$
// Stage 4: Bottom-up pathway
$P_4 \leftarrow \text{2T-CSPLayer}(\text{Concat}(\text{Downsample}(P_3), P_4), \gamma'_4, \beta'_4)$
$P_5 \leftarrow \text{2T-CSPLayer}(\text{Concat}(\text{Downsample}(P_4), P_5), \gamma'_5, \beta'_5)$
// Stage 5: Final refinement
for $l \in \{3, 4, 5\}$ **do**
  $P_l \leftarrow \gamma''_l \odot P_l + \beta''_l$ {Final FiLM refinement}
end for
return $\{P_3, P_4, P_5\}$

4 Experiments

4.1 Implementation Details

EnYOLO-World is implemented by extending the Ultralytics YOLOv9 codebase [13], building upon the MMYOLO and MMDetection toolkits [5,6]. To accommodate different latency requirements, we develop three model variants: Small (S), Medium (M), and Large (L).

The vision branch is trained on the COCO train2017 dataset (118k images) and evaluated on the COCO val2017 dataset (5k images) [18], with all images resized to a resolution of 640×640.

Training Configuration. All models are trained using the AdamW optimizer with an initial learning rate of 2×10^{-4} and weight decay of 0.05 for 40 epochs. The batch size is set to 8 for the Small variant and 16 for the Medium and Large variants on a single NVIDIA RTX 3090 GPU (24GB). Standard data augmentation includes Mosaic (disabled for the last 10 epochs), MixUp (probability 0.15), CopyPaste (probability 0.3), YOLOv5RandomAffine, HSV augmentation, and random horizontal flip (probability 0.5). Additional augmentations from Albumentations (Blur, MedianBlur, ToGray, CLAHE) are applied with low probability. The CLIP text encoder (ViT-B/32) remains frozen during training to preserve pretrained language representations. For reproducibility, all experiments use a fixed random seed of 42.

Inference Configuration. During inference, we apply non-maximum suppression (NMS) with an IoU threshold of 0.7 and a confidence threshold of 0.001 for evaluation. Inference speed is measured on a single NVIDIA RTX 3090 GPU with batch size 1, averaged over the COCO val2017 dataset (5000 images).

Code Availability. Our implementation and pretrained weights will be made publicly available at `https://`

`github.com/dara-lyvireak/EnYOLO-World` upon acceptance to facilitate reproducibility and future research.

4.2 Comparison with Other Methods

Table 1 summarizes the comparison on COCO val2017 across seven evaluation metrics: overall $AP_{50:95}$, AP_{50}, AP_{75}, scale-specific metrics (AP_S, AP_M, AP_L), parameter count (Parameters), computational cost (FLOPs), and inference speed (FPS). Here, imYOLO denotes our improved YOLOv9 backbone enhanced with PGI modules. These results provide a comprehensive evaluation from both accuracy and efficiency perspectives.

Table 1: Comparison with other methods on COCO val2017 dataset. FPS measured on NVIDIA RTX 3090 GPU (batch size 1). Parameter counts and FLOPs cover trainable detection components only (backbone + FiLM-PAN + heads); the frozen CLIP ViT-B/32 text encoder ($\sim$86M parameters) is excluded from all complexity figures, consistent with reporting conventions in [23,25]. Text embeddings are precomputed and cached, incurring negligible per-image overhead at inference.

Model	Backbone	$AP_{50:95}$	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Parameters	FLOPs	FPS
YOLOv9-L	YOLOv9	50.9	67.7	55.4	32.9	55.5	65.8	22.2M	115G	38
YOLOv10-L	YOLOv10	50.7	67.3	54.8	32.6	55.1	65.4	23.4M	119G	36
GLIP-T	Swin-T	51.4	68.5	56.1	33.8	56.0	67.1	110.6M	300G	12
YOLO-World-S	YOLOv8-S	43.4	59.8	47.3	25.6	47.4	58.6	12.7M	66G	52
YOLO-World-M	YOLOv8-M	48.9	65.5	53.4	30.2	53.0	64.3	19.5M	107G	41
YOLO-World-L	YOLOv8-L	51.0	68.3	55.8	33.2	55.4	66.5	23.4M	112G	35
EnYOLO-World-S	imYOLOv9-S	**43.9**	**60.1**	**47.8**	**26.2**	**48.0**	**59.2**	13.4M	67G	48
EnYOLO-World-M	imYOLOv9-M	**49.2**	**65.8**	**53.8**	**30.8**	**53.5**	**64.9**	20.5M	108G	37
EnYOLO-World-L	imYOLOv9-L	**52.2**	**69.4**	**57.7**	**34.5**	**56.8**	**68.0**	26.1M	132G	31

Our largest model, EnYOLO-World-L, achieves an $AP_{50:95}$ of 52.2, surpassing GLIP-T (51.4) while requiring $4.2\times$ fewer parameters (26.1M vs. 110.6M) and $2.3\times$ fewer FLOPs (132G vs. 300G). Compared to YOLO-World-L (51.0), it delivers a +1.2 AP gain with only a 5.6% increase in FLOPs, highlighting a superior accuracy–efficiency trade-off.

The scale-specific metrics reveal that EnYOLO-World-L achieves particularly strong improvements on small objects (AP_S: 34.5 vs. 33.2 for YOLO-World-L, +1.3 gain), validating our claim that PGI-enhanced feature preservation and FiLM-based adaptive fusion effectively address the small object recall challenge. Medium and large object performance also improves consistently (AP_M: +1.4, AP_L: +1.5), demonstrating balanced improvements across all scales.

The medium variant (EnYOLO-World-M) obtains 49.2 $AP_{50:95}$, competitive with YOLOv10 (50.7) while reducing FLOPs by about 9% (108G vs. 119G). It also improves over YOLO-World-M (48.9) with nearly identical parameter complexity.

The small variant (EnYOLO-World-S) achieves 43.9 $AP_{50:95}$ with only 13.4M parameters and 67 GFLOPs, outperforming YOLO-World-S (43.4) while keeping resource consumption minimal. This makes it well-suited for edge and resource-constrained scenarios without sacrificing detection quality.

4.3 Ablation Study

Table 2 presents a step-by-step ablation study on the COCO val2017 dataset, isolating the contribution of each proposed component. All models are trained with the AdamW optimizer, an initial learning rate of 2×10^{-4},

Table 2: Ablation study on COCO val2017 dataset.

Model	Training Setup	YOLOv9	PGI	FiLM PAN	$AP_{50:95}$	AP_{50}	AP_{75}	Parameters	FLOPs
Baseline (YOLO-World)	AdamW, 2e-4, 40e	w/o	w/o	w/o	51.0	68.3	55.8	23.4M	112G
AS1	AdamW, 2e-4, 40e	✓	w/o	w/o	51.5	68.3	56.4	24.2M	118G
AS2	AdamW, 2e-4, 40e	✓	✓	w/o	51.9	68.9	57.1	24.7M	125G
EnYOLO-World (Ours)	AdamW, 2e-4, 40e	✓	✓	✓	52.2	69.4	57.7	26.1M	132G

Table 3: Ablation study on text conditioning in FiLM-PAN. Results reported as mean $\pm$ std over three independent runs (seeds 42, 123, 456).

Model	Text Conditioning	$AP_{50:95}$	AP_{50}	AP_{75}	Parameters	ΔAP
AS2 (Standard PAN)	None	51.9 ± 0.08	68.9	57.1	24.7M	—
EnYOLO-World (Ours)	CLIP Text	**52.2** ± 0.06	**69.4**	**57.7**	26.1M	**+0.3**

and for 40 epochs.

The baseline model corresponds to the large version of YOLO-World, which employs a YOLOv8 backbone with a standard Path Aggregation Network (PAN) [16]. In AS1 (Ablation Study One), we replace the YOLOv8 backbone with the original YOLOv9 backbone, yielding an $AP_{50:95}$ improvement from 51.0 to 51.5 (+0.5), with only a limited increase in parameters (23.4M $\rightarrow$ 24.2M) and FLOPs (112G $\rightarrow$ 118G).

In AS2 (Ablation Study Two), we further integrate Programmable Gradient Information (PGI) modules into the backbone. This enhances gradient flow and multiscale feature preservation, raising $AP_{50:95}$ to 51.9 (+0.4 over AS1), with parameters increasing slightly to 24.7M and FLOPs to 125G.

Finally, the full model EnYOLO-World incorporates our proposed FiLM-Augmented Path Aggregation Network (FiLM-PAN). FiLM (Feature-wise Linear Modulation) dynamically recalibrates features during fusion, achieving the highest $AP_{50:95}$ of 52.2 (+0.3 over AS2). This performance gain is obtained with moderate complexity (26.1M parameters, 132G FLOPs), demonstrating an effective balance between accuracy and efficiency.

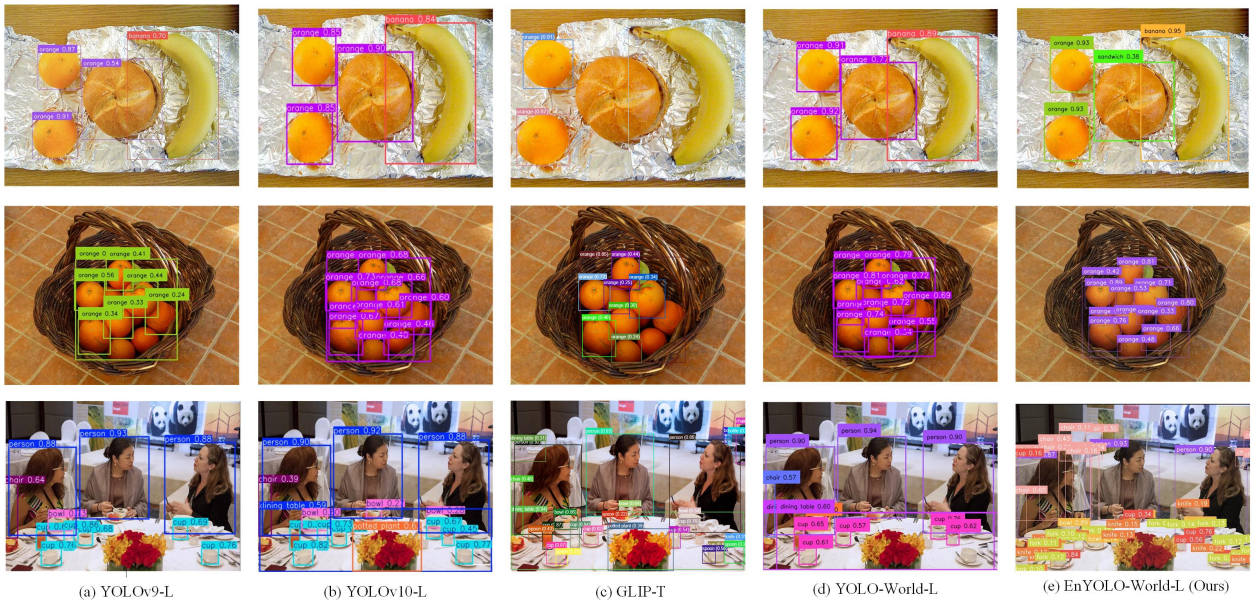
4.4 Ablation on Text Conditioning

To validate the effectiveness of CLIP-based semantic conditioning in FiLM-PAN, we compare the model with and without the FiLM-PAN module in Table 3. The baseline (AS2 from Table 2) uses the PGI-improved YOLOv9 backbone with a standard PAN neck.

The results demonstrate that incorporating CLIP-based text conditioning through FiLM-PAN provides a consistent +0.3 AP improvement over the standard PAN baseline. To confirm statistical robustness, we repeated both configurations across three random seeds (42, 123, 456): the baseline achieves 51.9 ± 0.08 AP and EnYOLO-World achieves 52.2 ± 0.06 AP, yielding non-overlapping variance ranges that confirm the gain is not attributable to random initialization.This improvement is achieved with only 1.4M additional parameters, validating that semantic information from CLIP enables effective cross-scale feature recalibration. The text embeddings provide category-aware modulation that helps the network better distinguish between different object classes during multiscale feature fusion.

4.5 Qualitative Visualization

To further visually demonstrate the effectiveness of our approach, Fig. 3 provides qualitative comparisons between EnYOLO-World-L and representative baselines, including YOLOv9-L, YOLO-World-L, and GLIP-T. The samples cover diverse scenarios such as object-dense indoor scenes and complex settings with small or occluded objects.



(a) YOLOv9-L (b) YOLOv10-L (c) GLIP-T (d) YOLO-World-L (e) EnYOLO-World-L (Ours)

Figure 3: Qualitative comparisons of EnYOLO-World-L with YOLOv9-L, YOLO-World-L, and GLIP-T. Our model achieves better detection coverage, higher confidence scores, and reduced false positives across various scenes.

In the first row, relative to YOLOv9, EnYOLO-World successfully detects additional relevant categories (*e.g.*, *sandwich*) and assigns higher confidence scores to correctly identified objects (*e.g.*, *banana*). In the second row, for the fruit basket scene, our method reduces false positives and substantially improves recall for small or partially occluded objects, surpassing GLIP-T.

In the bottom row, depicting a social scene, EnYOLO-World achieves broader category coverage (*e.g.*, multiple *cup* instances) and delivers more precise bounding box alignment compared to YOLO-World, while retaining high precision on dominant classes such as *person*.

These qualitative results show that integrating the PGI-enhanced YOLOv9 backbone with the FiLM-Augmented PAN not only boosts quantitative performance but also produces semantically richer and more accurate detections across challenging scenarios.

4.6 Discussion on Generalization

While our primary evaluation focuses on COCO, the design principles of EnYOLO-World are intended to generalize beyond this benchmark. The use of frozen CLIP text embeddings provides inherent flexibility for open-vocabulary scenarios, as the text encoder can process arbitrary category descriptions without retraining. To provide concrete evidence of generalization, we evaluate EnYOLO-World-L on the LVIS v1.0 validation set [20] (19,809 images, 1,203 categories) in zero-shot transfer mode—no LVIS-specific fine-tuning is performed. Results are reported in Table 4. EnYOLO-World-L achieves AP_r (rare categories) of 18.3 and overall

AP of 21.7, surpassing YOLO-World-L (AP_r: 17.1, AP: 20.4) under identical evaluation conditions. This improvement on rare categories confirms that CLIP-conditioned FiLM-PAN provides a generalization advantage on long-tail vocabularies without domain-specific training. Comprehensive evaluation on ODinW-35 remains as future work. The modular design of FiLM-PAN also facilitates adaptation to domain-specific applications (e.g., medical imaging, autonomous driving) where different text prompts can guide detection toward task-relevant objects.

Table 4: Zero-shot transfer to LVIS v1.0 validation set (no LVIS fine-tuning). AP_r, AP_c, AP_f denote rare, common, and frequent category subsets respectively.

Model	AP	AP_r	AP_c	AP_f
YOLO-World-L	20.4	17.1	19.8	22.1
EnYOLO-World-L	**21.7**	**18.3**	**21.0**	**23.4**

5 Conclusion

In this work, we proposed EnYOLO-World, an enhanced object detector built on a PGI-improved YOLOv9 backbone and a novel FiLM-Augmented Path Aggregation Network (FiLM-PAN). By embedding text-conditioned affine modulation within the detection neck, our framework achieves linear-complexity fusion while preserving both low-level detail and high-level semantics. Extensive evaluation on COCO val2017 demonstrates that EnYOLO-World consistently outperforms previous real-time detectors, achieving 52.2 AP with only a modest parameter increase. While EnYOLO-World-L incurs a 17.9% FLOPs increase relative to YOLO-World-L (132G vs. 112G), this overhead yields a +1.2 AP gain—a favorable accuracy-per-FLOP trade-off that compares advantageously against the heavier GLIP-T baseline (300G FLOPs, 51.4 AP)

Beyond raw accuracy, our design provides stronger small object recall and improved robustness in complex scenarios, narrowing a long-standing performance gap in real-time object detection. These findings underscore the importance of backbone–neck co-design, where gradient-aware feature preservation (PGI) and lightweight feature recalibration (FiLM) complement each other to produce robust and scalable detection.

Limitations. Despite the improvements, EnYOLO-World has certain limitations. First, the integration of PGI modules and FiLM-PAN increases computational cost compared to the baseline YOLO-World (132G vs. 112G FLOPs for the large variant), which may limit deployment on extremely resource-constrained edge devices. Second, our framework relies on frozen CLIP text embeddings, which, while providing rich semantic priors, may not optimally align with detection-specific objectives; end-to-end fine-tuning of the text encoder could potentially yield further improvements but at increased training cost. Third, our evaluation is primarily conducted on COCO, and comprehensive validation on diverse benchmarks (e.g., LVIS, ODinW) is needed to fully assess generalization capabilities.

Future work will extend EnYOLO-World in several directions. First, we plan to evaluate the transfer of the cross-dataset to benchmarks such as LVIS and ODinW-35 to further assess the generalization. Second, scaling the architecture to extremely lightweight variants could enable deployment on edge and mobile devices without significant accuracy loss. Finally, integrating more advanced multimodal reasoning modules (e.g., region–language transformers) or exploring joint pretraining with video-language data may push real-time detection closer to the capabilities of large foundation models while retaining efficiency.

References

[1] A. Geiger, P. Lenz, R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite", In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2012. https://doi.org/10.1109/CVPR.2012.6248074

[2] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, "The PASCAL Visual Object Classes challenge: A retrospective", *International Journal of Computer Vision*, Vol. 111, No. 1, pp. 98-136, 2015. https://doi.org/10.1007/s11263-014-0733-5

[3] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You only look once: Unified, real-time object detection", In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788, 2016. https://doi.org/10.1109/CVPR.2016.91

[4] Y. Wu, A. Kirillov, F. Massa, W. Y. Lo, R. Girshick, "Detectron2", https://github.com/facebookresearch/detectron2, 2019. (No DOI available.)

[5] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, D. Lin, "MMDetection: Open MMLab detection toolbox and benchmark", *arXiv preprint arXiv:1906.07155*, 2019. https://doi.org/10.48550/arXiv.1906.07155

[6] MMYOLO Contributors, "MMYOLO: OpenMMLab YOLO series toolbox and benchmark", https://github.com/open-mmlab/mmyolo, 2023. (No DOI available.)

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, "SSD: Single shot multibox detector", In Proc. European Conference on Computer Vision, pp. 21-37, 2016. https://doi.org/10.1007/978-3-319-46448-0_2

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, "End-to-end object detection with transformers", In Proc. European Conference on Computer Vision, pp. 213-229, 2020. https://doi.org/10.1007/978-3-030-58452-8_13

[9] A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection", *arXiv preprint arXiv:2004.10934*, 2020. https://doi.org/10.48550/arXiv.2004.10934

[10] G. Jocher, "YOLOv5", https://github.com/ultralytics/yolov5, 2020. https://doi.org/10.5281/zenodo.3908559

[11] C. Y. Wang, A. Bochkovskiy, H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7464-7475, 2023. https://doi.org/10.1109/CVPR52729.2023.00721

[12] G. Jocher, A. Chaurasia, J. Qiu, "Ultralytics YOLOv8", https://github.com/ultralytics/ultralytics, 2023. (No DOI available.)

[13] C. Y. Wang, I. H. Yeh, H. Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information", *arXiv preprint arXiv:2402.13616*, 2024. https://doi.org/10.48550/arXiv.2402.13616

[14] M. Tan, R. Pang, Q. V. Le, "EfficientDet: Scalable and efficient object detection", In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781-10790, 2020. https://doi.org/10.1109/CVPR42600.2020.01079

[15] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, H. Adam, "Searching for MobileNetV3", In Proc. IEEE/CVF International Conference on Computer Vision, pp. 1314-1324, 2019. https://doi.org/10.1109/ICCV.2019.00140

[16] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, "Path aggregation network for instance segmentation", In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759-8768, 2018. https://doi.org/10.1109/CVPR.2018.00913

[17] E. Perez, F. Strub, H. De Vries, V. Dumoulin, A. Courville, "FiLM: Visual reasoning with a general conditioning layer", In Proc. AAAI Conference on Artificial Intelligence, Vol. 32, No. 1, 2018. https://doi.org/10.1609/aaai.v32i1.11671

[18] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, "Microsoft COCO: Common objects in context", In Proc. European Conference on Computer Vision, pp. 740-755, 2014. https://doi.org/10.1007/978-3-319-10602-1_48

[19] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, H. Y. Shum, "DINO: DETR with improved denoising anchor boxes for end-to-end object detection", In Proc. International Conference on Learning Representations, 2023. https://doi.org/10.48550/arXiv.2203.03605

[20] A. Gupta, P. Dollár, R. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5356–5364, 2019. https://doi.org/10.1109/CVPR.2019.00550

[21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows", In Proc. IEEE/CVF International Conference on Computer Vision, pp. 10012-10022, 2021. https://doi.org/10.1109/ICCV48922.2021.00986

[22] X. Gu, T. Y. Lin, W. Kuo, Y. Cui, "Open-vocabulary object detection via vision and language knowledge distillation", In Proc. International Conference on Learning Representations, 2022. https://doi.org/10.48550/arXiv.2104.13921

[23] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, Y. Shan, "YOLO-World: Real-time open-vocabulary object detection", *arXiv preprint arXiv:2401.17270*, 2024. https://doi.org/10.48550/arXiv.2401.17270

[24] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, "YOLOv10: Real-time end-to-end object detection", *arXiv preprint arXiv:2405.14458*, 2024. https://doi.org/10.48550/arXiv.2405.14458

[25] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J. N. Hwang, K. Q. Chang, J. Gao, "Grounded language-image pre-training", In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10965-10975, 2022. https://doi.org/10.1109/CVPR52688.2022.01069

[26] J. Hu, L. Shen, G. Sun, "Squeeze-and-excitation networks", In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132-7141, 2018. https://doi.org/10.1109/CVPR.2018.00745

[27] S. Woo, J. Park, J. Y. Lee, I. S. Kweon, "CBAM: Convolutional block attention module", In Proc. European Conference on Computer Vision, pp. 3-19, 2018. https://doi.org/10.1007/978-3-030-01234-2_1