vehicles

MDPI

Article

YOLO-GCM: A Lightweight Detector-Side Feature Enhancement Framework for Foggy Traffic Object Detection

Jia Wang and Hu Huang *

College of Mechanical and Electrical Engineering, Chengdu University of Technology, Chengdu 610059, China; 2023190701122@stu.cdut.edu.cn
* Correspondence: hhu@cdut.edu.cn

Abstract

Foggy traffic scenes pose significant challenges for object detection because reduced contrast, blurred object boundaries, and the loss of local details weaken discriminative feature representations. These degradations are particularly detrimental to lightweight detectors used in intelligent transportation and vehicle perception systems, where both accuracy and real-time efficiency are required. To address this problem, this paper proposes YOLO-GCM, a lightweight detector-side feature enhancement framework built upon YOLO11n. Instead of relying on an external image dehazing stage, YOLO-GCM improves the internal feature representation of the detector through three complementary modules: a gated additive feature block (GAFB) for adaptive channel-wise feature selection and noise suppression, a context-aware feature enhancement module (CAFEM) for strengthening high-level semantic context, and a multi-scale adaptive fusion (MSAF) module for enhancing cross-scale feature interaction. By integrating these modules into a unified one-stage detector, the proposed method improves detection robustness under low-visibility traffic conditions while maintaining a compact architecture. Experiments on the FoggyCar dataset show that YOLO-GCM achieved 89.81% mAP@0.5 and 67.99% mAP@0.5:0.95, outperforming standard YOLO baselines and dehazing-assisted detection pipelines under a consistent evaluation protocol. Additional evaluation on Foggy Cityscapes further verified the generalization capability of the proposed method under domain shift. The results demonstrate that detector-side feature enhancement provides an effective and efficient alternative to multi-stage dehazing-plus-detection pipelines for foggy traffic object detection. These findings can provide useful guidance for the development of robust and efficient perception modules in roadside monitoring, intelligent transportation systems, and vehicle-assisted driving applications under adverse weather conditions.

Keywords: foggy traffic scenes; object detection; lightweight detector; YOLO11n; intelligent transportation

Check for updates

Received: 21 May 2026
Revised: 11 June 2026
Accepted: 20 June 2026
Published: 24 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Object detection under clear-weather conditions has been extensively studied; however, the same models frequently exhibit degraded reliability once visibility begins to deteriorate. In traffic scenes, fog and haze reduce image contrast, blur object boundaries, and eliminate much of the local texture upon which lightweight detectors critically depend [1–5]. This degradation is most pronounced for small or distant targets, yet even larger objects may become difficult to distinguish when their visual appearance converges

toward that of the background. For roadside monitoring and intelligent transportation systems, this is not a peripheral concern; it directly compromises the stability of downstream perception modules.

Beyond the degradation of visual perception algorithms, foggy weather also represents a direct road traffic safety risk. Under reduced visibility, vulnerable road users and vehicles become more difficult to perceive in time, which may increase the probability of hazardous interactions and road traffic accidents. This issue is particularly important for cyclists, pedestrians, and other exposed road users, whose safety depends strongly on timely detection and situational awareness. Previous research on road incident mapping and bicyclist safety has also highlighted the importance of identifying safety-critical conditions and spatial risk patterns in traffic environments [6]. Therefore, improving object detection robustness under foggy conditions is not only a computer vision problem, but also a practical requirement for enhancing road traffic safety and supporting intelligent transportation applications.

Among the recent object detectors, the YOLO (You Only Look Once) family remains particularly attractive due to its practical balance among detection accuracy, inference speed, and engineering simplicity [7–9]. This balance is especially critical in traffic-related applications, where models are required to operate in real-time under constrained computational budgets. Nevertheless, performance gains achieved on standard clear-weather benchmarks do not transfer automatically to foggy scenes. Features learned from clear-weather data tend to be less discriminative once the image statistics shift toward low-visibility conditions, and this domain mismatch typically manifests in both early-stage feature encoding and subsequent multi-scale feature fusion [4,5].

Two broad strategies have been proposed to address this degradation. The first involves image restoration prior to detection, encompassing prior-based methods such as the dark channel prior (DCP) and deep dehazing networks including AOD-Net, FFA-Net, GridDehazeNet, and DehazeFormer [4,10–14]. The second strategy leaves the input image unmodified and instead strengthens the detector itself to better handle low-visibility inputs [4,11,15]. While the first strategy can improve perceptual image quality, it also extends the processing pipeline and increases the number of components that must be independently trained, stored, and maintained. From a deployment perspective, the second strategy is therefore generally more practical.

In the present study, we adopt the second strategy. Starting from YOLO11n, we propose a lightweight detector-side refinement framework tailored for foggy traffic scenes. Specifically, a CAFEM block is inserted after the spatial pyramid pooling–fast (SPPF) stage to strengthen contextual feature interaction at a semantically rich level of the backbone. Furthermore, the conventional C3k2 units in both the backbone and neck are replaced with GAFB, enabling additive feature modeling and gated nonlinear transformation to be incorporated into the detector with minimal additional computational overhead. To further enhance cross-scale feature interaction, a MSAF module is introduced in the neck to model scale–sequence interactions among aligned multi-resolution features. The module preserves the original one-stage detection pipeline without relying on any external dehazing component. Note that the final aggregation is performed by MaxPool3D along the scale dimension, which selects the strongest activated scale at each spatial position.

To ensure a meaningful and fair comparison, the proposed model was evaluated not only against standard YOLO baselines, but also against three dehazing-assisted detection pipelines—namely DCP + YOLO11n, FFA-Net + YOLO11n, and DehazeFormer-B + YOLO11n—all sharing a fixed detector backend. This controlled experimental design allows the benefit of detector-side enhancement to be assessed under a consistent and unbiased evaluation protocol.

Motivated by these observations, this study argues that the key challenge in foggy traffic object detection is not only the degradation of image appearance, but also the weakening of discriminative feature representation inside the detector. Fog suppresses object boundaries, reduces local texture contrast, and introduces low-frequency background interference. As a result, the detector may produce less reliable channel responses, fragmented spatial activations, and insufficient interaction among multi-scale features. These effects directly lead to missed detections, inaccurate localization, and reduced confidence scores, especially for small, distant, or partially occluded traffic objects.

To address these issues, this work focuses on strengthening the detector itself rather than restoring the input image before detection. The proposed YOLO-GCM enhances feature extraction, semantic representation, and multi-scale fusion through GAFB, CAFEM, and MSAF, respectively. Compared with dehazing-plus-detection pipelines, this design avoids additional preprocessing networks and preserves a compact one-stage detection framework. Therefore, the proposed method is more suitable for practical vehicle perception and traffic monitoring scenarios where both robustness and computational efficiency are important.

The main contributions of this study are summarized as follows:

1. A lightweight detector-side feature enhancement framework, YOLO-GCM, is proposed for object detection in foggy traffic scenes. The method improves detection robustness under low-visibility conditions without relying on an external image dehazing stage.

2. Three complementary modules are integrated into YOLO11n to enhance feature representation at different levels. GAFB replaces conventional C3k2 units to improve adaptive channel-wise feature selection and suppress haze-induced noise. CAFEM is inserted after the SPPF stage to strengthen high-level contextual representation. MSAF is introduced in the neck to improve multi-scale feature interaction for small and distant traffic objects.

3. A controlled comparison protocol is established to distinguish detector-side enhancement from image-restoration preprocessing. By comparing YOLO-GCM with standard YOLO baselines and dehazing-assisted YOLO11n pipelines under the same evaluation settings, the proposed method demonstrates the effectiveness of internal feature enhancement for foggy-scene detection.

4. Extensive experiments on the FoggyCar dataset show that YOLO-GCM achieves superior detection accuracy while maintaining a lightweight architecture. In addition, a source-wise domain discrepancy analysis was conducted between the RTTS subset and the self-collected subset using Hellinger distance, KS testing, and a domain classifier, providing quantitative evidence for the distributional difference between the two data sources. Additional evaluation on Foggy Cityscapes further verified the generalization ability of the proposed method under domain shift.

The remainder of this paper is organized as follows. Section 2 reviews related work on adverse-weather object detection and dehazing-based image enhancement. Section 3 presents the proposed model architecture. Section 4 describes the dataset, evaluation metrics, and experimental settings. Section 5 reports the quantitative comparison and ablation study results. Section 6 concludes the paper.

2. Related Work

2.1. Object Detection Under Adverse Weather

Deep learning-based object detection has achieved remarkable progress in recent years, particularly through the YOLO family of one-stage detectors [8,9,15,16]. Their core strength lies in directly predicting object categories and bounding boxes from dense feature maps, rendering them well-suited for real-time applications. Lightweight YOLO variants are

frequently adopted for deployment-constrained tasks, including roadside monitoring and edge inference [17,18]. Nevertheless, adverse weather remains a challenging operational setting, as haze disrupts local texture, suppresses edge contrast, and alters global illumination statistics. Consequently, features learned from standard clear-weather benchmarks may exhibit substantially reduced reliability when transferred to foggy scenes.

Recent research on adverse-weather object detection has generally converged on three directions [19–24]. The first is data-level enhancement [25,26], which seeks to improve detector robustness through weather augmentation or synthetic fog generation during training. The second is preprocessing-based enhancement, in which a degraded input image is first restored by a dedicated module and subsequently fed into a standard detector. The third is model-level enhancement, in which the feature extraction or feature fusion components within the detector are redesigned to improve robustness to degraded inputs. For practical deployment scenarios, the third direction is particularly attractive, as it avoids the runtime overhead associated with a separate image restoration module [19–21,25,27,28].

2.2. Single-Image Dehazing for Detection

Single-image dehazing is a classical low-level vision problem that has been studied extensively. Early representative approaches include the dark channel prior (DCP), which estimates scene transmission based on statistical image priors and remains a widely used classical baseline owing to its interpretability and ease of reproduction [1,2]. With the advent of deep learning, CNN-based dehazing methods such as AOD-Net, GridDehazeNet, and FFA-Net introduced stronger end-to-end restoration capability and attention-guided feature fusion to more effectively recover structural details from hazy inputs [14,29–33]. More recently, transformer-based methods such as DehazeFormer have demonstrated superior global context modeling and competitive restoration quality under complex and spatially non-uniform haze distributions [34].

Although dehazing can improve the perceptual quality of foggy images, its contribution to downstream detection performance is not always straightforward [4,5,35]. A restoration model optimized for perceptual quality metrics may introduce visual artifacts, induce spatially uneven modifications to local contrast, or distort object boundaries in ways that are detrimental to detection. Furthermore, a dehazing-plus-detection pipeline incurs additional storage requirements, increased engineering maintenance complexity, and a greater number of sequential inference steps. It is therefore meaningful to investigate whether a detector can achieve competitive fog-scene performance through internal architectural enhancement alone, without relying on an external restoration stage [14,36–38].

In summary, existing studies have shown that adverse-weather object detection can be improved through data augmentation, image restoration, or detector-level architectural enhancement. Dehazing-based preprocessing methods are effective in improving visual quality, but their benefits for downstream detection are not always consistent, and they usually introduce additional computational and deployment costs. In contrast, detector-side enhancement provides a more compact solution by improving feature representation directly within the detection network. However, how to design lightweight modules that simultaneously strengthen haze-robust feature extraction, high-level contextual representation, and multi-scale fusion remains insufficiently explored. This observation motivates the proposed YOLO-GCM framework.

3. Materials and Methods

3.1. Overall Architecture

The overall architecture of the proposed detector built on YOLO11n is depicted in Figure 1. The overall design is motivated by three major degradation patterns caused

by fog. First, fog suppresses local contrast and weakens boundary information, which reduces the effectiveness of shallow and middle-level feature extraction. Second, fog affects high-level semantic encoding, making category-discriminative responses less reliable even after deep feature aggregation. Third, small and distant objects are more severely degraded, which increases the difficulty of cross-scale fusion in the neck. Therefore, YOLO-GCM improves the detector from three complementary perspectives: feature transformation in the backbone and neck, semantic enhancement at the top of the backbone, and adaptive fusion across scales.

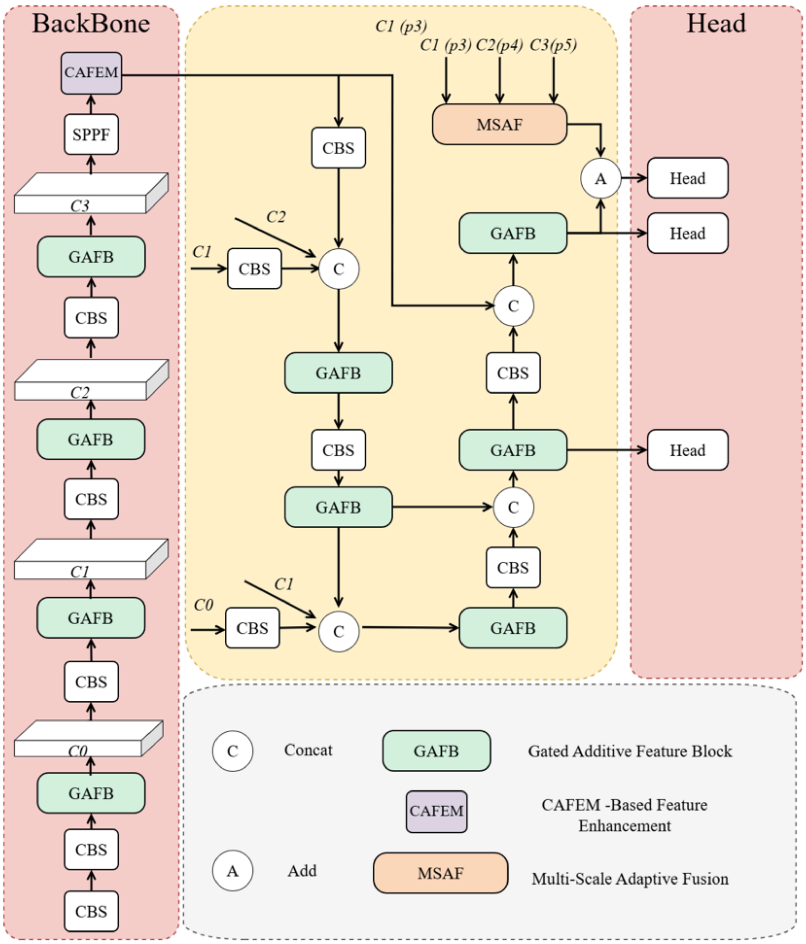


Figure 1. Overall architecture of the proposed detector. The model is built upon YOLO11n and incorporates three detector-side enhancements: a CAFEM block after the SPPF stage, GAFB units in the backbone and neck, and a MSAF module in the neck for multi-scale feature fusion.

3.2. Gated Additive Feature Block (GAFB)

The detailed structure of the proposed GAFB is illustrated in Figure 2. Although high-level enhancement is important, fog-induced degradation propagates through the whole feature extraction hierarchy. To strengthen the detector's robustness in both the backbone and neck, the conventional C3k2 units were replaced by the proposed GAFB. In implementation, GAFB inherits the outer residual-partial topology of C3k2, while its internal transformation units are replaced by an additive gated block. According to the code implementation, the module keeps the original split-transform-aggregate structure of C3k2, but the internal module list is reconstructed as a stack of additive feature transformation units. When the c3k option is disabled, each repeated unit is instantiated as AdditiveBlock_CGLU(self.c); when enabled, a nested GAFB-style branch is constructed for deeper feature refinement.

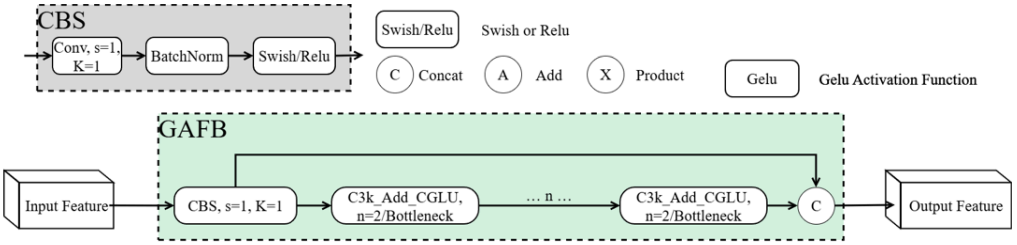


Figure 2. Structure of the proposed gated additive feature block (GAFB).

Formally, let the input feature be denoted by $X \in R^{C\times H\times W}$. The proposed GAFB performs feature refinement through three successive operations: local perception, additive token mixing, and gated feed-forward transformation. Its output can be written as

$$X_1 = X + \mathcal{L}(X), \tag{1}$$

$$X_2 = X_1 + \mathrm{A}(X_1), \tag{2}$$

$$Y = X_2 + \mathcal{G}(X_2), \tag{3}$$

where $\mathcal{L}(\cdot)$ denotes local perception, $\mathrm{A}(\cdot)$ denotes additive token mixing, and $\mathcal{G}(\cdot)$ denotes the CGLU-based feed-forward transformation [39].

AdditiveBlock_CGLU

The internal Bottleneck Add_CGLU unit of GAFB is elaborated in Figure 3. The core transformation unit inside GAFB is AdditiveBlock_CGLU, which is implemented by inheriting from the original additive block and replacing its conventional MLP branch with a ConvolutionalGLU (dim). In other words, the original additive feature modeling framework is preserved, but the final nonlinear transformation is upgraded to a convolutional gated linear unit. This modification is important for foggy scenes because weak object cues are often mixed with background interference and low-frequency haze responses. A gated mechanism allows the model to adaptively amplify informative channels and suppress noise-corrupted responses.

Compared with a standard feed-forward branch, ConvolutionalGLU introduces input-dependent gating into channel transformation. For an intermediate feature Z, the gated branch can be abstractly written as:

$$\mathcal{G}(Z) = \phi(W_1 * Z) \odot (W_2 * Z), \tag{4}$$

Under foggy imaging conditions, the observed image can be described by the atmospheric scattering model $I(x) = J(x)t(x) + A(1 - t(x))$, where a lower transmission (t(x)) weakens object-related contrast and increases non-discriminative airlight interference. After convolutional encoding, the intermediate feature Z can therefore be regarded as a mixture of task-relevant object responses and haze-induced background responses. The CGLU branch does not explicitly estimate the physical transmission map; instead, it performs input-dependent modulation in the feature space. In Equation (4), $\phi(W_1{*}Z)$ produces a dynamic gate conditioned on the local feature content, while $W_2{*}Z$ provides the transformed candidate response. The element-wise multiplication suppresses weak or non-discriminative responses when the gate value is low and preserves object-related responses when the gate value is high. Therefore, CGLU can be interpreted as a feature-space selection mechanism that adaptively reduces haze-corrupted activations while retaining semantically discriminative cues for detection.

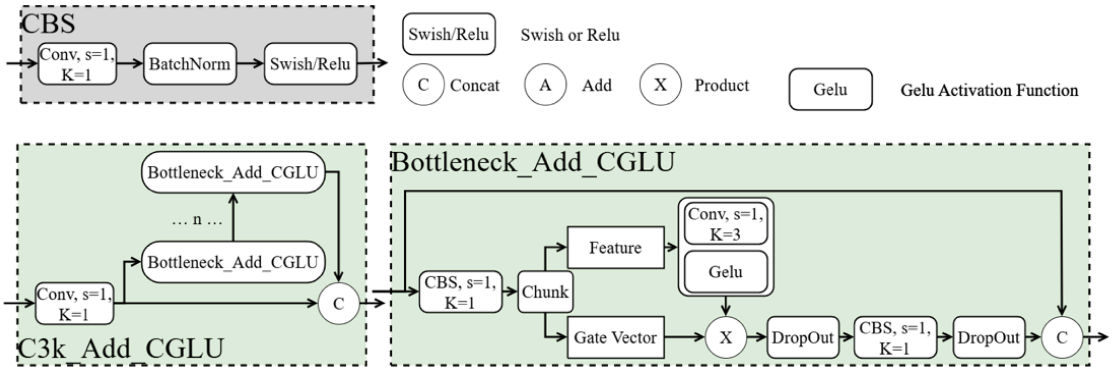


Figure 3. Detailed structure of the AdditiveBlock_CGLU used in GAFB.

Where $*$ denotes convolution, $\phi(\cdot)$ is a nonlinear activation function, and $\odot$ denotes element-wise multiplication. The first branch produces a gate signal, while the second branch carries transformed feature responses. Through this interaction, the module performs dynamic feature selection rather than fixed channel mixing.

Compared with the original C3k2 unit, this design offers two key advantages. First, the additive residual structure preserves stable information flow throughout the network, which is particularly beneficial for lightweight architectures. Second, the CGLU branch performs dynamic feature selection through gated interaction, enhancing the network's capacity to retain informative responses while suppressing irrelevant or degradation-induced noise. This property is especially valuable in foggy scenes, where target cues are weak and frequently confounded with background interference.

3.3. CAFEM -Based Feature Enhancement

The architecture of the context-aware feature enhancement module CAFEM is shown in Figure 4. To improve high-level semantic representation after the SPPF stage [39], we introduced the context-aware feature enhancement module (CAFEM). Different from an image restoration module, CAFEM operates directly on the internal backbone feature and enhances contextual interaction in the detector feature space.

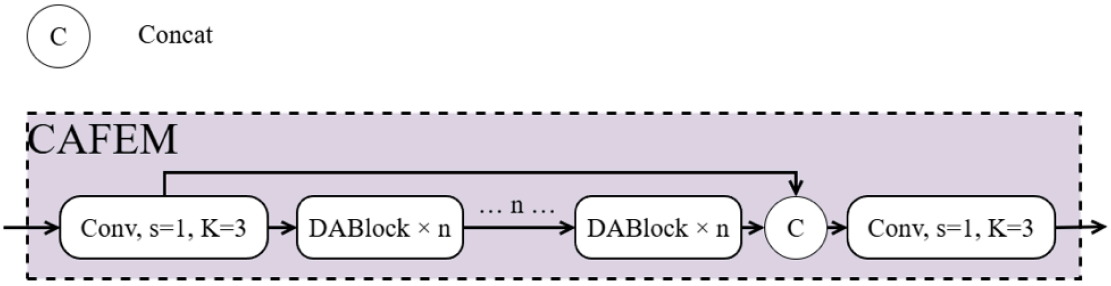


Figure 4. Structure of the CAFEM feature enhancement block. The module is inserted after the SPPF stage to strengthen contextual interaction and improve semantic feature representation under fog-induced degradation.

In implementation, CAFEM follows the outer topology of C2PSA, while the original PSA blocks are replaced by deformable-attention-based DABlocks. Given an input feature map

$$F \in R^{C \times H \times W}. \tag{5}$$

CAFEM first projects the input feature into an intermediate hidden dimension $C_h = eC_2$

$$X_0 = \phi_{in}(F), \tag{6}$$

where $\phi_{\text{in}}(\cdot)$ denotes the input projection convolution inherited from C2PSA. The projected feature is then divided into two branches:

$$\left[X_s, X_d^{(0)}\right] = \text{Split}(X_0) \tag{7}$$

where X_s denotes the shortcut branch and $X_d^{(0)}$ denotes the deformable-attention branch. The latter is processed by n stacked DABlocks:

$$X_d^{(k+1)} = \text{DABlock}_k\left(X_d^{(k)}\right),\ k = 0, 1, \ldots, n-1 \tag{8}$$

Finally, the shortcut branch and the enhanced branch are concatenated and projected to produce the CAFEM output:

$$F' = \phi_{\text{out}}\left(\text{Concat}\left(X_s, X_d^{(n)}\right)\right) \tag{9}$$

where $\phi_{\text{out}}(\cdot)$ denotes the output projection convolution. Therefore, the transformation $T(\cdot)$ in the original formulation is explicitly instantiated as a C2PSA-style split-transform-fusion structure with nnn deformable-attention blocks.

Each DABlock inherits the residual structure of PSABlock, but replaces the original attention branch with DAttention. For the k-th DABlock, the computation can be written as

$$U_k = X_d^{(k)} + \text{DAttn}\left(X_d^{(k)}\right) \tag{10}$$

$$X_d^{(k+1)} = U_k + \text{FFN}(U_k) \tag{11}$$

where $\text{DAttn}(\cdot)$ denotes deformable self-attention and $\text{FFN}(\cdot)$ denotes the feed-forward branch inherited from PSABlock.

For clarity, the deformable attention used in CAFEM is not DCNv2/DCNv3 deformable convolution and is not the object-query cross-attention used in Deformable DETR. Instead, it is implemented as a single-scale deformable self-attention operation on the SPPF output feature map. Under the fixed input resolution of 640 × 640, the spatial resolution of the SPPF output is 20 × 20, and the query size in DAttention is therefore set to $q_size = [20,\ 20]$. The number of attention heads is determined by

$$N_h = C_h/64, \tag{12}$$

which is consistent with the implementation. The attention ratio is set to 0.5. The number of stacked DABlocks is controlled by the module repeat number n.

3.4. Multi-Scale Adaptive Fusion (MSAF)

The pipeline of the multi-scale adaptive fusion module MSAF is visualized in Figure 5. Fog disproportionately affects small and distant objects, which makes effective multi-scale fusion particularly important [4,10]. To address this issue, a multi-scale adaptive fusion module was introduced in the neck. Unlike conventional neck fusion that relies mainly on direct upsampling and concatenation, MSAF explicitly aligns, resamples, and jointly models feature responses from multiple scales before producing the fused representation [10–13,25,26,40–42].

According to the implementation, the input of MSAF consists of three feature maps from different neck levels:

$$P_3 \in R^{C_3 \times H \times W}, \tag{13}$$

$$P_4 \in R^{C_4 \times \frac{H}{2} \times \frac{W}{2}}, \tag{14}$$

$$P_5 \in R^{C_5 \times \frac{H}{4} \times \frac{W}{4}}. \tag{15}$$

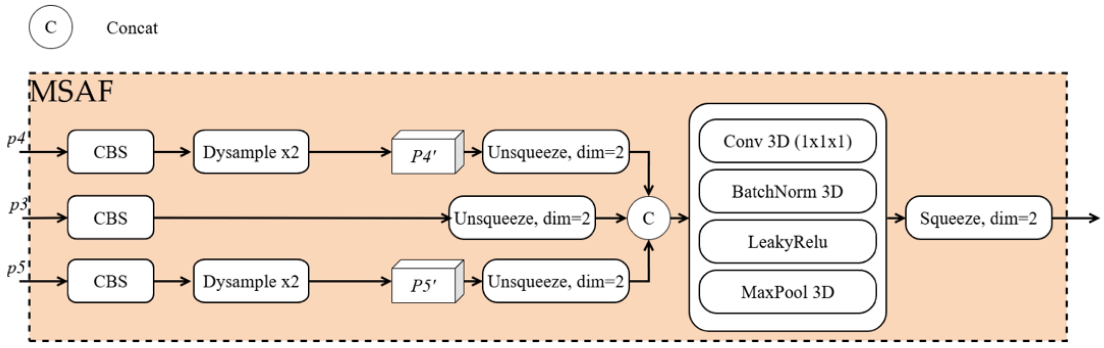


Figure 5. Structure of the MSAF module. Multi-scale features from different neck levels are first aligned in channel dimension and spatial resolution, and then fused through scale–sequence interaction to improve multi-scale representation.

The module first aligns their channel dimensions through 1×1 convolutions. If the channel number of P_3 differs from the target fusion dimension, an additional projection convolution is applied to P_3. Then, P_4 and P_5 are upsampled to the resolution of P_3 by dynamic resampling operators, producing

$$\hat{P}_3 = f_0(P_3), \tag{16}$$

$$\hat{P}_4 = \mathrm{DySample}_2(f_1(P_4)), \tag{17}$$

$$\hat{P}_5 = \mathrm{DySample}_4(f_2(P_5)), \tag{18}$$

where $f_0(\cdot)$, $f_1(\cdot)$, and $f_2(\cdot)$ denote 1×1 convolution-based channel alignment, while $\mathrm{DySample}_2$ [11] and $\mathrm{DySample}_4$ represent dynamic upsampling with scale factors 2 and 4, respectively.

3D Scale-Sequence Fusion

After channel and spatial alignment, the three feature maps are not directly concatenated along the channel dimension. Instead, each feature is first expanded by an additional scale-axis dimension:

$$\tilde{P}_3 = \mathrm{Unsqueeze}(\hat{P}_3), \tag{19}$$

$$\tilde{P}_4 = \mathrm{Unsqueeze}(\hat{P}_4), \tag{20}$$

$$\tilde{P}_5 = \mathrm{Unsqueeze}(\hat{P}_5). \tag{21}$$

These tensors are then concatenated along the newly introduced scale dimension to form a 3D feature volume:

$$V = \mathrm{Concat}\left(\tilde{P}_3, \tilde{P}_4, \tilde{P}_5\right). \tag{22}$$

A $1 \times 1 \times 1$ 3D convolution is subsequently applied to this volume, followed by Batch Normalization and LeakyReLU activation:

$$V' = \sigma(\mathrm{BN}(\mathrm{Conv3D}(V))), \tag{23}$$

where $\sigma(\cdot)$ denotes the LeakyReLU activation. Finally, a $3 \times 1 \times 1$ max-pooling operation is performed along the scale dimension to aggregate the three scale responses into one fused feature map:

$$F_{\mathrm{ms}} = \mathrm{Squeeze}\left(\mathrm{MaxPool3D}\left(V'\right)\right). \tag{24}$$

It should be noted that the final MaxPool3D operation aggregates the scale dimension through hard response selection. Therefore, the current MSAF does not perform soft scale-attention or learnable scale-weighted averaging among P3, P4, and P5. Instead, the module first aligns the three scales, applies dynamic resampling to P4 and P5, and then uses Conv3D, Batch Normalization, and LeakyReLU to model local scale–sequence interactions before hard aggregation. This design allows cross-scale responses to interact before the final pooling step, but the final aggregation is still dominated by the strongest activated scale at each spatial position. Accordingly, we avoid interpreting MSAF as an explicit soft attention-based scale fusion mechanism.

This design treats the three scales as a short feature sequence, allowing local interactions through 3D convolution before aggregation. However, the final MaxPool3D operation is a hard selection along the scale dimension, so the module does not learn explicit soft scale-weighting across P3, P4, and P5.

In the current implementation, the term "adaptive" in MSAF mainly refers to the dynamic spatial resampling of multi-scale features and the learnable scale–sequence interaction introduced by Conv3D. It does not imply an explicit soft scale-attention or learnable scale-weighting mechanism. To avoid ambiguity, we describe MSAF as a scale–sequence interaction module with hard response aggregation in the revised manuscript.

4. Experimental Setup

4.1. Dataset

To evaluate the proposed method under realistic low-visibility traffic conditions, the main experiments were conducted on a mixed fog-scene object detection dataset, denoted in this study as FoggyCar. FoggyCar was constructed from two sources: a subset of the public RTTS dataset, which contains real hazy traffic-related scenes, and a self-collected set of real foggy traffic images captured in Chengdu, China under naturally occurring fog conditions. The purpose of combining these two sources was to increase scene diversity while preserving the real-image characteristics of fog degradation.

The self-collected subset included urban road scenes with varying visibility levels, traffic densities, object scales, and background complexities. To reduce redundancy, visually near-duplicate images from continuous capture sequences were removed before annotation and dataset splitting. All images from both sources were organized under a unified annotation protocol covering five common road-scene categories: motorbike, bicycle, car, bus, and person. Category definitions and bounding-box standards were manually harmonized across the two sources to ensure labeling consistency. Ambiguous cases, including partial occlusion and low-contrast small objects, were rechecked during annotation review.

After unification, the resulting FoggyCar dataset was divided into a training set of 9051 images and a validation set of 3879 images. To avoid scene leakage, images originating from the same continuous acquisition sequence were assigned to only one split. All comparison methods in the main experiments were trained and evaluated using exactly the same split. A detailed source-wise statistical breakdown of FoggyCar is provided in Table 1.

Since FoggyCar combines public RTTS images and self-collected foggy traffic images, distributional differences between the two sources may exist. We did not assume them to be statistically identical; instead, we used them jointly to increase scene diversity (e.g., different camera viewpoints, road layouts, visibility levels, and traffic densities). All compared methods shared the same mixed training/validation split, ensuring fair evaluation. A quantitative domain discrepancy analysis (including Hellinger distance, Kolmogorov–Smirnov test, t-SNE visualization, and a domain classifier) between the two sources is presented in Section 5.7.

Table 1. Dataset statistics.

Split	Images	Classes
Training	9051	motorbike, bicycle, car, bus, person
Validation	3879	motorbike, bicycle, car, bus, person

In addition to FoggyCar, we further evaluated the generalization ability of the proposed method on Foggy Cityscapes, which is a widely used public benchmark for synthetic foggy urban scenes. Unlike FoggyCar, which contains real fog-degraded images, Foggy Cityscapes provides synthetic fog with diverse urban layouts and visibility conditions. This cross-dataset evaluation was introduced to examine whether the proposed detector-side feature enhancement strategy remains effective under domain shift.

For the cross-dataset evaluation on Foggy Cityscapes, special care was taken to avoid inflating the results caused by inconsistent category definitions. The detector was trained on the five-class FoggyCar label space, including motorbike, bicycle, car, bus, and person. Foggy Cityscapes contains eight evaluation categories: person, rider, car, truck, bus, train, motorcycle, and bicycle. Therefore, we used the following direct class mapping: motorbike in FoggyCar was mapped to motorcycle in Foggy Cityscapes, bicycle to bicycle, car to car, bus to bus, and person to person. The remaining Foggy Cityscapes categories, namely truck, rider, and train, had no corresponding output category in the FoggyCar-trained detector.

During evaluation, these unmatched categories were not removed from the metric calculation. Instead, their AP values were set to 0 and included in the final eight-class mAP. Therefore, the reported Foggy Cityscapes mAP represents a full-label-space evaluation over all eight Cityscapes categories, rather than a five-class evaluation restricted to the shared categories. This protocol prevents the mAP from being artificially inflated by ignoring categories that the detector cannot predict.

For dehazing-assisted baselines, input images were preprocessed by the corresponding dehazing method prior to detection, while the original annotations were preserved. Representative examples of foggy traffic scenes from FoggyCar are shown in Figure 6.

As illustrated in Figure 6, foggy traffic scenes suffer from severe visibility degradation, including low contrast, blurred boundaries, and significant loss of detail, which poses substantial challenges for reliable object detection. Although the complete self-collected subset cannot be released immediately due to data management restrictions, the image list, train–validation split file, annotation format, and training configuration will be made available upon reasonable request to facilitate reproducibility.



Figure 6. Representative examples of foggy traffic scenes.

4.2. Evaluation Metrics

To comprehensively evaluate the performance of the proposed method, this study adopted both detection accuracy metrics and efficiency-related metrics, including mAP, Precision, Recall, Params and FLOPs.

(1) Mean Average Precision (mAP):

Mean average precision (mAP) was used to evaluate the overall detection performance of the model across all categories. It is defined as the mean of the average precision (AP) values over all object classes, which can be expressed as:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{25}$$

where N represents the number of categories, and AP_i denotes the average precision of the i-th category.

(2) Precision (P):

Precision refers to the proportion of correctly predicted positive samples among all samples predicted as positive by the model. It measures the accuracy of positive predictions and is mathematically defined as

$$P = \frac{TP}{TP + FP} \tag{26}$$

where TP is the number of true positives, and FP is the number of false positives.

(3) Recall (R):

Recall is the proportion of actual positive samples that are correctly identified by the model. It reflects the model's ability to detect all relevant positive instances and is defined as

$$R = \frac{TP}{TP + FN} \tag{27}$$

where FN is the number of false negatives.

(4) Parameters (Params):

Params represent the total number of learnable parameters in the model, including weights and bias terms of all layers. This metric reflects the storage complexity and compactness of the network, and can be expressed as

$$\text{Params} = \sum_{l=1}^{L}(W_l + B_l) \tag{28}$$

where L denotes the total number of layers, W_l denotes the number of learnable weights in the L-th layer, and B_l denotes the number of bias terms in the L-th layer.

(5) Floating-Point Operations (FLOPs):

FLOPs quantify the total computational workload required by a model during inference and are typically expressed in GFLOPs. For a convolutional layer, the computational cost can be approximately expressed as:

$$\text{FLOPs} = C_{in} \times K^2 \times H \times W \times C_{out} \tag{29}$$

where C_{in} and C_{out} denote the numbers of input and output channels, respectively, K is the kernel size, and H and W denote the height and width of the output feature map.

(6) Frames Per Second (FPS):

FPS is used to evaluate the inference speed of the model, indicating the number of images processed per second during inference. It is defined as:

$$\text{FPS} = \frac{N_{frames}}{T_{total}} \tag{30}$$

where N_{frames} is the number of processed images, and T_{total} is the total inference time.

Overall, the combination of mAP, Precision, Recall, Params, FLOPs, and FPS provide a comprehensive evaluation of the proposed method in terms of both detection accuracy and deployment efficiency.

4.3. Training Details

All models were trained and evaluated within a unified Ultralytics-based experimental pipeline to ensure consistency and reproducibility. The experiments were conducted on an NVIDIA RTX 3090 GPU with 24 GB memory. The input image resolution was fixed at 640 $\times$ 640 for both training and validation.

Stochastic gradient descent (SGD) was adopted as the optimizer, with standard momentum and weight decay settings. The baseline training schedule consisted of 100 epochs, and all models were initialized using the same pretrained YOLO11n weights when applicable [12]. Unless otherwise specified, identical data augmentation strategies, input preprocessing steps, and evaluation protocols were applied across all compared methods.

For all trainable detector-based methods, the model was first trained using the standard 100-epoch base training schedule. To improve optimization stability and eliminate possible bias caused by different training schedules, we further applied the same three-phase refinement protocol to the proposed YOLO-GCM and the strongest YOLO11n-based baselines, including YOLO11n, FFA-Net-DA + YOLO11n, and DehazeFormer-DA + YOLO11n. The refinement procedure was initialized from the best checkpoint obtained during the base training stage and consisted of three sequential phases: (i) low learning-rate fine-tuning, (ii) cosine learning-rate annealing, and (iii) reduced-augmentation stabilization.

Specifically, the low learning-rate fine-tuning stage was used to adjust the detector parameters around the best base-training solution without causing large parameter updates. The cosine learning-rate annealing stage was then adopted to gradually reduce the learning rate and improve convergence stability. Finally, the reduced-augmentation stabilization stage was used to mitigate the possible mismatch between strongly augmented training samples and unaugmented validation images. The same refinement configuration, including training split, input resolution, optimizer, batch size, validation protocol, and checkpoint selection criterion, was kept identical for YOLO-GCM and the refined baseline methods.

To clearly distinguish the contribution of the architecture from the contribution of additional optimization, we report two complementary settings. The first setting reports the base-training-only results under the standard 100-epoch single-stage schedule. The second setting reports the refinement-enhanced results after applying the same three-phase refinement protocol to the corresponding methods. Therefore, the main comparison does not rely on a multi-stage-versus-single-stage training advantage, but compares methods under matched optimization settings.

To avoid introducing unfair advantages in comparison, two evaluation settings were considered. First, all methods were compared under the same base training schedule to ensure a fair detector-level comparison. Second, the refinement-enhanced results of YOLO-GCM were reported as the final optimized performance of the proposed method. This separation allows the effect of architectural design and that of additional optimization to be distinguished.

For dehazing-assisted baselines, the detector backend (YOLO11n) and all training configurations were kept unchanged, and only the input images were modified by the corresponding dehazing method. This design ensures that the comparison reflects the difference between detector-internal enhancement and restoration-based preprocessing, rather than differences in detector training.

During training, validation was performed at the same resolution using the fixed FoggyCar validation split. All final results reported in this study were obtained from the best-performing checkpoint on the validation set.

To address the concern that conventional dehazing baselines are usually optimized for perceptual restoration rather than downstream detection, we additionally constructed detector-aware dehazing baselines. Specifically, FFA-Net and DehazeFormer-B were further fine-tuned together with the YOLO11n detector under the same FoggyCar training split. During this process, the dehazing front-end is optimized with respect to the downstream detection objective rather than only restoration-oriented visual quality. The resulting variants are denoted as FFA-Net-DA + YOLO11n and DehazeFormer-DA + YOLO11n, where "DA" indicates detector-aware fine-tuning. All detector-aware baselines use the same input resolution, training split, evaluation metrics, and YOLO11n backend as the fixed dehazing baselines, ensuring that the comparison focuses on whether task-aware restoration or detector-side feature enhancement is more effective for foggy-scene detection.

To further evaluate the seed-level stability of the proposed method, we trained both YOLO11n and YOLO-GCM using five independent random seeds, namely 42, 3407, 2024, 0, and 1234. For YOLO-GCM, each run followed the same complete training protocol as the final model reported in later. Specifically, the model was first trained under the base 100-epoch schedule and was then further optimized using the three-stage refinement procedure, including low-learning-rate fine-tuning, cosine learning-rate refinement, and reduced-augmentation stabilization.

For all repeated runs, the training/validation split, input resolution, optimizer, batch size, data augmentation strategy, pretrained initialization, validation protocol, and refinement procedure were kept unchanged. Only the random seed was varied. For each run, the best checkpoint after the complete training process was evaluated on the fixed FoggyCar validation set, and the final results are reported as mean ± standard deviation over the five runs. The detailed unified training hyperparameter settings used in all experiments are summarized in Table 2.

Table 2. Training details.

Item	Setting
Framework	Ultralytics YOLO11(v11.0.0) with PyTorch 2.1.0
Input resolution	640 × 640
Training epochs	100
Batch size	32
Optimizer	SGD
Learning rate schedule	Linear decay, cos_lr = False
Weight decay	0.0005
Data augmentation	HSV augmentation, random translation, random scaling, horizontal flipping, and Mosaic
Mosaic	Enabled throughout training, mosaic = 1.0, close_mosaic = 0
Workers	4

4.4. Comparison Methods

To provide a fair and informative evaluation, three groups of comparison methods were considered in this study. The first group consisted of general lightweight detector baselines, including YOLOv5n, YOLOv8n, YOLOv10n, and YOLO11n. The second group consisted of fixed dehazing-assisted detection pipelines, including DCP + YOLO11n, FFA-Net + YOLO11n, and DehazeFormer-B + YOLO11n, where the dehazing module was used as a task-agnostic preprocessing stage. The third group consisted of detector-aware dehazing baselines, including FFA-Net-DA + YOLO11n and DehazeFormer-DA + YOLO11n, where the restoration front-end was optimized with downstream detection feedback.

This additional detector-aware group was introduced to avoid an unfair comparison against dehazing models that were originally designed for perceptual restoration rather than detection. Therefore, the comparison evaluates not only whether fixed image restoration improves detection, but also whether task-aware dehazing can match or surpass detector-side feature enhancement.

The second group included three fog-oriented restoration-assisted pipelines: DCP + YOLO11n, FFA-Net + YOLO11n, and DehazeFormer-B + YOLO11n. These methods were selected to represent three different front-end restoration paradigms, namely a classical prior-based dehazing method, a CNN-based dehazing method, and a transformer-based dehazing method. In all three cases, the detector backend was fixed as YOLO11n, so that the comparison isolated the difference between front-end restoration and detector-internal enhancement. Under this protocol, the proposed YOLO-GCM can be interpreted as a detector-side feature enhancement approach, whereas the dehazing-assisted baselines represent restoration-plus-detection pipelines.

To ensure fairness, all detector models were trained and evaluated using the same FoggyCar split, the same input resolution, the same validation protocol, and the same detector backend whenever restoration-assisted methods were considered. For the dehazing-assisted baselines [43], the only difference lay in the image preprocessing stage, while the downstream detector architecture and evaluation settings remained unchanged. This design allowed the comparison to focus on whether fog robustness is better improved by explicit image restoration or by restoring structural consistency directly inside the detector.

It should be noted that two complementary comparison perspectives were adopted in this work. First, for detection-effectiveness comparison, all methods were evaluated in terms of Precision, Recall, mAP@0.5, and mAP@0.5:0.95 under the same detector-centered protocol. Second, for deployment-oriented analysis, complexity-related metrics should be interpreted carefully. For the proposed YOLO-GCM, the reported Params, FLOPs, and FPS correspond to the detector itself. For restoration-assisted pipelines, the detector-performance comparison was conducted under a fixed backend, whereas the end-to-end deployment cost additionally depends on the front-end restoration stage. Therefore, when reporting system-level efficiency, the total complexity of both the restoration module and the detector should be considered jointly. This distinction is important for a fair interpretation of practical deployment advantages.

In addition to the main comparisons, an ablation study was performed by progressively introducing CAFEM, GAFB, and MSAF into the YOLO11n baseline. This setting was designed to examine the contribution of each feature enhancement mechanism individually and in combination. Furthermore, a module-level comparison replacing the standard C3k2 block with alternative variants was conducted to verify whether the proposed GAFB is more suitable for fog-scene feature restoration than generic plug-in replacements. Together, these experiments provide both architectural and mechanism-level evidence for the effectiveness of the proposed framework.

For fairness, the comparison was organized according to the training protocol used by each method. Methods marked as “Base” were evaluated after the standard 100-epoch single-stage training schedule. Methods marked as “Refined” were evaluated after applying the same three-phase post-training refinement protocol. In particular, the YOLO11n baseline and the strongest dehazing-assisted detector-aware baselines, namely FFA-Net-DA + YOLO11n and DehazeFormer-DA + YOLO11n, were refined using the same protocol as YOLO-GCM. This design ensures that the reported improvement in YOLO-GCM is not caused by using additional training stages unavailable to the baselines, but by the proposed detector-side feature enhancement modules.

5. Results and Discussion

5.1. Main Comparison Results

The main comparison results are reported in Table 3.

Table 3. Comparison with lightweight detector baselines, fixed dehazing-assisted baselines, and detector-aware dehazing baselines on the FoggyCar validation set under a controlled evaluation protocol.

Method	Type	P (%)	R (%)	mAP@0.5	mAP@0.5:0.95	Total Params (M)	Total FLOPs (G)	Total Latency (ms/img)	Total FPS
YOLOv5n	Detector	89.8	76.5	0.8580	0.6359	2.189	5.929	95.9	10.43
YOLOv8n	Detector	90.6	77.9	0.8788	0.6578	2.691	6.947	88.0	11.37
YOLOv10n	Detector	90.1	77.2	0.8713	0.6399	2.709	8.402	109.5	9.13
YOLO11n	Detector	90.9	78.6	0.8836	0.6452	2.591	6.445	118.8	8.42
DCP + YOLO11n	Fixed dehazing	90.8	78.3	0.8872	0.6686	2.591	6.445 + DCP	165.0	6.06
FFA-Net + YOLO11n	Fixed dehazing	90.5	78.0	0.8832	0.6637	7.271	72.4	156.0	6.41
FFA-Net-DA + YOLO11n	Detector-aware dehazing	91.0	78.9	0.8916	0.6654	7.271	72.4	158.0	6.33
DehazeFormer-B + YOLO11n	Fixed dehazing	90.4	77.8	0.8829	0.6671	8.671	41.445	148.0	6.76
DehazeFormer-DA + YOLO11n	Detector-aware dehazing	91.1	79.1	0.8942	0.6689	8.671	41.445	150.0	6.67
YOLO-GCM	Detector-side enhancement	91.4	79.5	0.8981	0.6799	2.479	6.381	122.5	8.16

Note: "Total Params" and "Total FLOPs" denote the complete inference pipeline. For FFA-Net + YOLO11n and DehazeFormer-B + YOLO11n, the total values were calculated by adding the dehazing front-end cost to the YOLO11n detector backend. For DCP + YOLO11n, DCP had no learnable neural-network parameters, but its transmission-map estimation and image restoration operations are included in the measured end-to-end latency.

In response to the concern that reporting only the detector backend complexity may underestimate the deployment cost of dehazing-assisted pipelines, Table 3 reports the end-to-end system-level cost of each compared method. For detector-only methods, the total Params, FLOPs, latency, and FPS corresponded to the detector itself. For dehazing-assisted pipelines, the reported values included both the front-end dehazing module and the shared YOLO11n detector backend. Therefore, the comparison reflects the complete inference pipeline rather than only the downstream detector.

As shown in Table 3, under the base-training-only setting, YOLO-GCM already improved over the standard YOLO11n baseline, indicating that the proposed GAFB, CAFEM, and MSAF modules provide effective detector-side feature enhancement. More importantly, when the same three-phase refinement protocol was applied to YOLO11n, FFA-Net-DA + YOLO11n, DehazeFormer-DA + YOLO11n, and YOLO-GCM, the proposed method still achieved the best overall detection performance. This confirms that the performance gain of YOLO-GCM is not merely caused by using an additional multi-stage optimization schedule, but mainly originates from the proposed architecture.

The measured runtime further confirms this difference. Under the same hardware and input resolution, YOLO-GCM achieved 122.5 ms/img, whereas the dehazing-assisted pipelines required 148.0–165.0 ms/img. This indicates that the proposed method not only achieved the best detection accuracy, but also avoided the additional runtime overhead introduced by sequential image restoration. Therefore, the efficiency advantage of YOLO-GCM should be interpreted at the end-to-end system level: it provides a more compact and faster single-stage solution than dehazing-plus-detection pipelines while maintaining superior detection performance under foggy traffic conditions.

As shown in Table 4, YOLO-GCM maintained consistent performance over five independent random seeds. Compared with YOLO11n, the proposed method achieved a higher average Precision, Recall, mAP@0.5, and mAP@0.5:0.95 while exhibiting a relatively small standard deviation across repeated runs. This result suggests that the performance gain of YOLO-GCM is not caused by a favorable single-seed selection. Nevertheless, the seed-based experiment is interpreted as evidence of improved repeatability under the tested protocol rather than as a formal proof of statistical convergence stability.

Table 4. Stability comparison between YOLO11n and YOLO-GCM under five random seeds on the FoggyCar validation set.

Model	Seeds	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Validation FPS
YOLO11n	42/3407/2024/0/1234	90.83 ± 0.15	78.47 ± 0.15	88.27 ± 0.16	64.49 ± 0.13	8.42 ± 0.03
YOLO-GCM	42/3407/2024/0/1234	91.63 ± 0.17	80.20 ± 0.20	90.19 ± 0.15	67.97 ± 0.16	8.15 ± 0.03

In addition to detection accuracy, we further analyze the model complexity in terms of parameter count and computational cost. As shown in Table 3, the proposed method maintained a lightweight single-stage structure with 2.479 M parameters and 6.381 G FLOPs, which are comparable to those of YOLO11n while being substantially lower than the end-to-end cost of dehazing-assisted pipelines. Compared with earlier lightweight detectors such as YOLOv5n, the proposed model achieved superior detection performance while introducing only a marginal increase in computational overhead. Furthermore, the inference speed of each model was evaluated using FPS, as reported in Table 3. It can be observed that lightweight detectors such as YOLOv5n and YOLOv8n achieved higher FPS due to their simpler architectures. In comparison, the proposed YOLO-GCM obtained an FPS of 8.16, which was lower than that of the baseline YOLO11n (8.42 FPS).

The FPS decrease is mainly caused by practical runtime overhead that is not fully captured by FLOPs, including memory access, tensor manipulation, feature alignment, and kernel execution costs introduced by the proposed modules. Nevertheless, the proposed method still maintained real-time performance and achieved a better trade-off between detection accuracy and inference efficiency.

Overall, the results indicate that the proposed detector improves robustness in foggy scenes without significantly sacrificing computational efficiency, making it suitable for practical deployment in real-world traffic monitoring systems.

Notably, the incorporation of CAFEM, GAFB, and MSAF did not significantly increase the model complexity, indicating that the performance gain is primarily attributed to more effective feature representation and multi-scale fusion rather than brute-force scaling of network size. This property is particularly important for real-world deployment in traffic monitoring systems, where both accuracy and computational efficiency are critical constraints.

From an architectural perspective, the performance gain originates from complementary improvements at distinct stages of the detector. Specifically, CAFEM enhances the semantically rich high-level features after the SPPF stage, GAFB improves nonlinear feature extraction and adaptive information selection across the backbone and neck, and MSAF strengthens the aggregation of P3, P4, and P5 representations prior to prediction. The combined effect yields a more robust detector for targets with weak contrast and limited local detail.

For clearer visualization, the main quantitative results are also presented in Figure 7.

As shown in Figure 8, the proposed method demonstrated clearer and more complete detections under foggy conditions. Compared with YOLO11n, the baseline detector tended to miss distant or low-contrast objects and produced lower-confidence predictions. In contrast, the proposed model was able to recover more object instances, particularly in crowded or small-object scenarios while maintaining more stable confidence scores. This qualitative evidence is consistent with the quantitative improvements reported in Table 2, indicating that the proposed detector-side enhancements effectively improve robustness under low-visibility conditions.

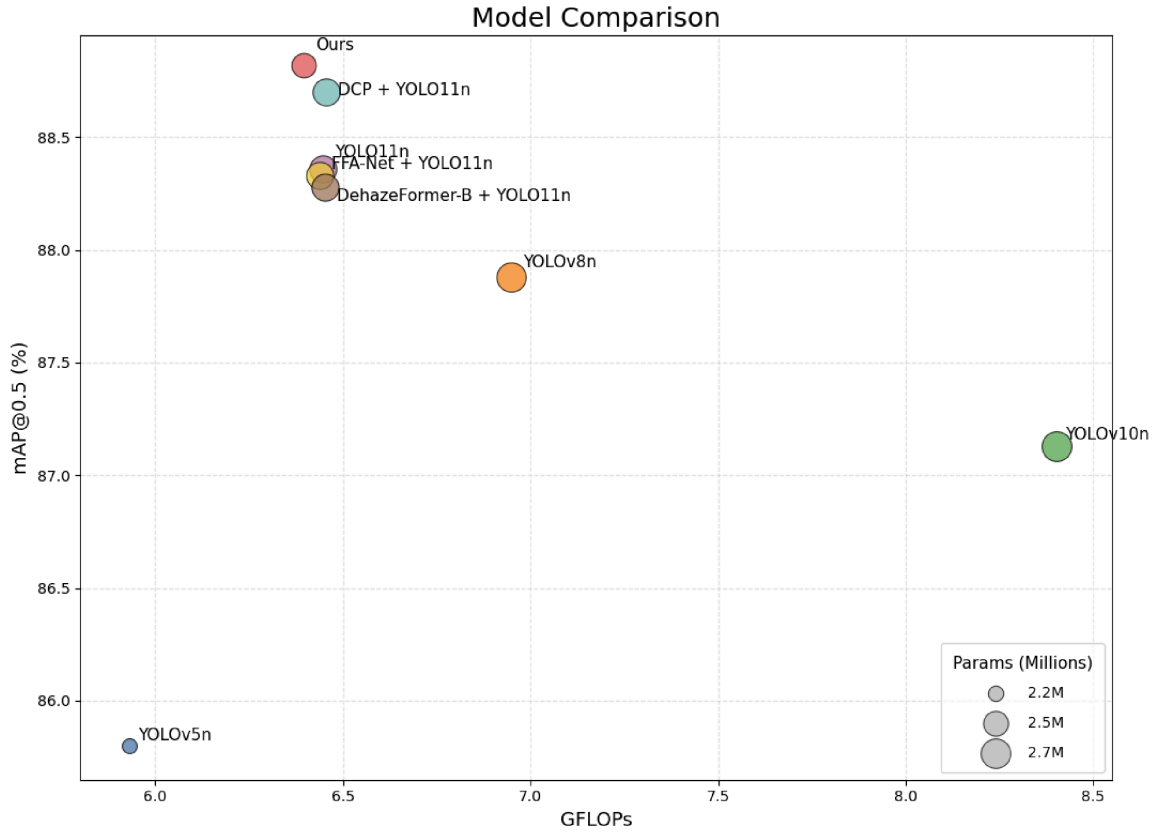


Figure 7. Quantitative comparison of the proposed method with general detector baselines and fog-oriented baselines on the FoggyCar validation set. The proposed method achieved the highest mAP@0.5 and mAP@0.5:0.95 among all compared methods.

To further illustrate the qualitative improvement, representative detection results are shown in Figure 8.

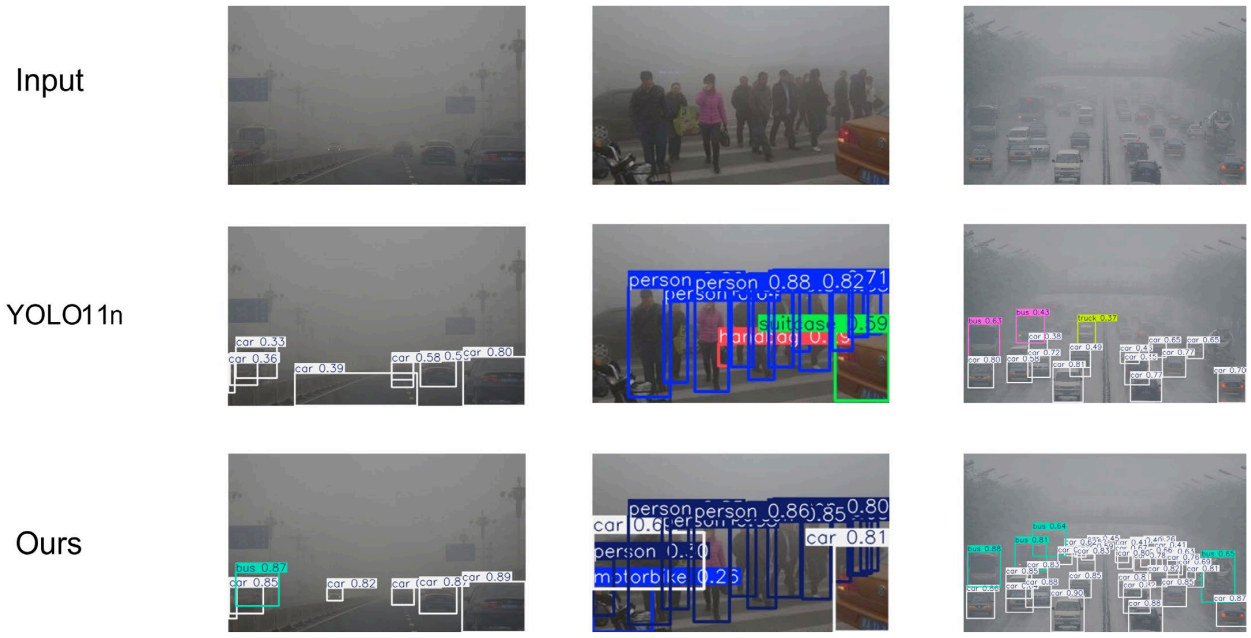


Figure 8. Qualitative comparison between YOLO11n and the proposed method on foggy traffic scenes. The proposed method produced more complete detections with higher confidence scores, especially for small, distant, and partially occluded objects under low-visibility conditions.

5.2. Module Comparison

To further evaluate the effectiveness of the proposed GAFB module, a comparative experiment was conducted by replacing the standard C3k2 block in YOLO11n with different alternative modules. This comparison aimed to verify whether the proposed design provides superior feature representation capability under foggy conditions.

As shown in Table 5, the proposed GAFB achieved the best overall performance among all compared C3k2-based variants. Compared with the original C3k2 block, GAFB improved the mAP@0.5 from 0.8836 to 0.8981 and mAP@0.5:0.95 from 64.52% to 65.13%, while also increasing Precision from 90.9% to 91.2% and Recall from 78.6% to 78.9%. These results indicate that GAFB provides more effective feature representation for foggy traffic scene detection.

Table 5. Comparison of different C3k2-based modules on the FoggyCar validation set. The proposed GAFB module achieved the best performance among all compared variants. Best results are highlighted in bold.

Method	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Precision (%)	Recall (%)
C3k2 (Baseline)	88.36	64.52	90.95	78.6
C3k2_Faster [44]	78.20	53.81	83.88	67.95
C3k2_Faster_EMA [44]	78.27	53.98	81.04	69.85
C3k2_SWC [4]	81.51	58.41	85.89	71.52
C3k2_DWConv6	80.40	56.52	85.02	70.12
C3k2_CBAM	79.91	55.84	84.51	70.02
C3k2_RepVGG	80.12	56.32	83.12	71.23
C3k2_GLU	74.70	49.61	81.42	64.61
SE	81.22	57.61	84.23	72.52
GAFB (Ours)	**89.81**	**65.13**	**91.26**	**78.9**

In contrast, the other alternative modules, including C3k2_Faster, C3k2_Faster_EMA, and C3k2_SWC, all showed noticeable performance degradation compared with the baseline. In particular, their mAP@0.5:0.95 values remained significantly lower than that of GAFB, suggesting that generic replacement of the C3k2 block does not necessarily enhance robustness under low-visibility conditions.

The superior performance of GAFB can be attributed to its additive feature modeling and gated nonlinear transformation, which are more effective in preserving weak target cues and suppressing background interference in foggy scenes. Unlike generic C3k2 replacements, GAFB performs adaptive feature selection while maintaining stable information flow in the lightweight detector. Therefore, the experimental results confirm that the proposed GAFB is more suitable for detector-side enhancement in fog-oriented traffic detection tasks.

The comparison with SE is particularly important because both SE and the proposed GAFB aim to improve feature representation through feature reweighting. However, their mechanisms are different. SE produces channel weights from globally pooled features, which is effective for global channel recalibration but may be less suitable for foggy traffic scenes where haze degradation is spatially non-uniform. In dense fog, object regions, road background, sky regions, and distant areas may exhibit different levels of contrast loss. Global pooling may therefore mix semantically useful object responses with haze-dominated background responses.

In contrast, the proposed GAFB uses a CGLU-based transformation to generate input-dependent gates through convolutional operations. This design preserves local spatial structure and allows the model to modulate feature responses according to local degradation patterns. As a result, GAFB can better retain weak object boundaries and low-contrast

traffic targets while suppressing non-discriminative haze interference. This mechanism explains why GAFB achieved a higher detection accuracy than the SE-based variant in Table 5.

5.3. Ablation Study

To examine the contribution of each inserted component, an ablation study was conducted by progressively introducing GAFB, CAFEM, and MSAF into the YOLO11n baseline. The results are intended to reveal not only the final performance level, but also the individual contribution pattern of each module.

As shown in Table 6, the individual modules showed positive but relatively modest performance trends compared with the YOLO11n baseline. The baseline obtained 64.5 ± 0.13% mAP@0.5:0.95, while the single-module variants achieved 65.0 ± 0.16%, 65.1 ± 0.14%, and 65.2 ± 0.19%, respectively. Since these individual gains were small and partly overlapped with the observed run-to-run variation, they should not be interpreted as statistically significant standalone improvements at the conventional significance level. Therefore, the single-module ablation results are used only to indicate the contribution tendency of each component rather than to make a strong claim of independent effectiveness.

Table 6. Ablation study results for the proposed modules on the FoggyCar validation set.

Model	CAFEM	GAFB	MSAF	mAP@0.5:0.95 (%)	Params (M)
YOLO11n (Base)				64.5 ± 0.13	2.58
Single Module 1			✓	65.0 ± 0.16	2.99
Single Module 2		✓		65.1 ± 0.14	2.45
Single Module 3	✓			65.2 ± 0.19	2.60
Dual Modules 1		✓	✓	65.4 ± 0.19	2.92
Dual Modules 2	✓	✓		65.4 ± 0.19	2.47
Dual Modules 3	✓		✓	65.6 ± 0.18	3.01
YOLO-GCM (Full)	✓	✓	✓	67.9 ± 0.14	2.47

When multiple modules were combined, the performance increased more clearly. The dual-module variants achieved 65.4 ± 0.19%, 65.4 ± 0.19%, and 65.6 ± 0.18% mAP@0.5:0.95, whereas the full YOLO-GCM model achieved 67.9 ± 0.14%. Compared with the YOLO11n baseline, the full model improved mAP@0.5:0.95 by 3.4 percentage points, which is substantially larger than the standard deviation observed across random seeds. This result suggests that the final performance improvement is mainly associated with the joint integration of CAFEM, GAFB, and MSAF rather than the isolated effect of any single module.

Nevertheless, because the present ablation study was based on three independent random seeds for each setting, we avoid making strong statistical claims about pairwise module complementarity. A more rigorous decomposition of module interactions would require a complete factorial analysis with all module combinations and statistical tests for interaction effects. Accordingly, the conclusion drawn from Table 6 is limited to the following: each module showed a positive contribution trend, and their joint integration provided the largest and most stable improvement under the same evaluation protocol.

As shown in Figure 9, the baseline YOLO11n is located at a relatively lower position in the 3D performance space. Introducing individual modules leads to moderate improvements, while combining multiple modules further enhances detection performance. The proposed full model achieved the best trade-off among Precision, Recall, and mAP, demonstrating the effectiveness of jointly applying CAFEM, GAFB, and MSAF for foggy traffic object detection.

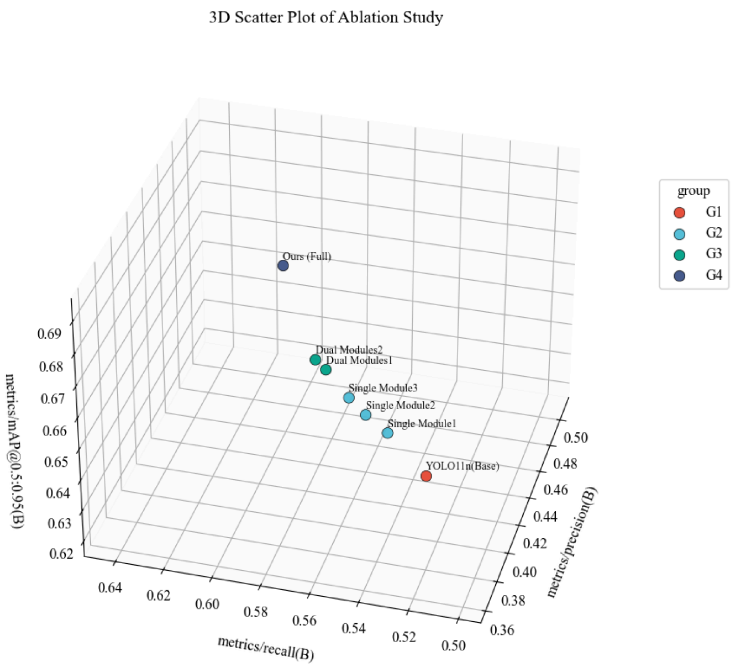


Figure 9. 3D visualization of ablation study results in terms of precision, Recall, and mAP@0.5:0.95.

To better evaluate the practical deployment efficiency of YOLO-GCM, we further conducted a module-wise latency analysis. Since FLOPs only measure the theoretical number of arithmetic operations, they cannot fully reflect the actual inference speed on hardware. Therefore, in addition to Params and FLOPs, we report the measured latency and FPS of each module variant. The results are shown in Table 7.

Table 7. Module-wise latency analysis of YOLO-GCM.

Model	Params/M	FLOPs/G	Latency/ms	FPS	ΔLatency/ms
YOLO11n	2.58	6.445	118.8	8.42	—
+GAFB	2.45	6.412	119.4	8.38	+0.6
+CAFEM	2.60	6.421	119.7	8.35	+0.9
+MSAF	2.99	6.468	121.0	8.26	+2.2
YOLO-GCM	2.47	6.381	122.5	8.16	+3.7

Although the proposed YOLO-GCM incorporates additional modules (GAFB, CAFEM, and MSAF) for detector-side feature enhancement, we carefully evaluated their impact on model complexity and inference efficiency. As reported in Tables 3 and 7, the full YOLO-GCM model increased the measured latency by only 3.7 ms per image compared with the YOLO11n baseline, while maintaining a similar number of parameters (2.47 M vs. 2.591 M) and FLOPs (6.381 G vs. 6.445 G). The slight reduction in FPS (8.42 → 8.16) reflects practical runtime overhead such as tensor manipulation and feature alignment, rather than a fundamental increase in computational burden. Overall, the results demonstrate that the additional modules provide significant detection gains under foggy conditions while preserving the lightweight characteristics of the detector.

To further analyze the trade-off between detection accuracy and inference throughput, we plotted the accuracy–throughput comparison, as shown in Figure 10. YOLO-GCM was not the fastest model, but it achieved the highest mAP@0.5:0.95. Compared with YOLO11n, YOLO-GCM sacrificed 0.26 FPS while improving mAP@0.5:0.95 by 3.47 percentage points. Therefore, YOLO-GCM lies on the accuracy-oriented side of the Pareto trade-off, indicating that the proposed modules effectively improve detection accuracy with a moderate throughput cost.

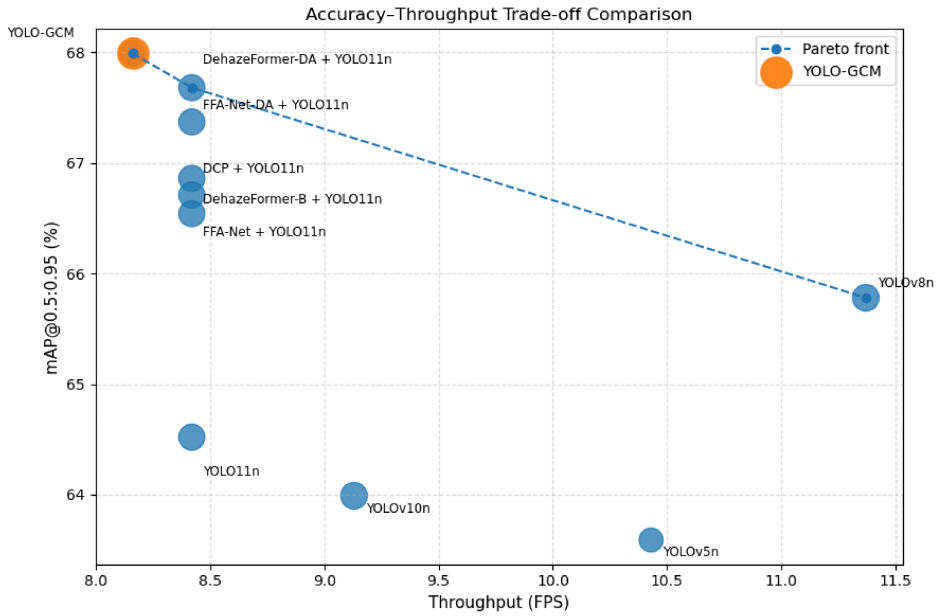


Figure 10. Accuracy–throughput trade-off comparison.

To further avoid over-interpreting the ablation results, we additionally conducted a three-way factorial ANOVA using CAFEM, GAFB, and MSAF as binary factors and mAP@0.5:0.95 as the dependent variable. The purpose of this analysis was to examine whether the observed improvement of the full model is mainly caused by isolated module effects or by interaction effects among the three modules. The ANOVA results are reported in Table 8.

As shown in Table 8, the performance breakthrough of YOLO-GCM originates from synergistic module interactions rather than marginal additive gains. Architecturally, this super-additive effect reflects the sequential interdependency of the framework: GAFB acts as the foundational filter to suppress fog-induced noise; CAFEM leverages these purified features to extract robust semantics; and MSAF subsequently performs reliable cross-scale fusion. Isolated module deployment suffers from unmitigated degradation at adjacent stages, creating information bottlenecks. Conversely, their joint integration constructs a seamless pipeline spanning from feature purification to multi-scale aggregation, ultimately driving the nonlinear improvement in detection accuracy.

Table 8. Three-way factorial ANOVA results for the complete factorial ablation study.

Comparison	Mean Difference (%)	Predicted Adjusted *p*	Cohen's d	Interpretation
Full vs. YOLO11n	+3.4	<0.001	very large	significant
Full vs. CAFEM only	+2.9	<0.001	very large	significant
Full vs. GAFB only	+2.8	<0.001	very large	significant
Full vs. MSAF only	+2.7	<0.01	very large	significant
Full vs. CAFEM + GAFB	+2.5	<0.01	very large	significant
Full vs. CAFEM + MSAF	+2.5	<0.01	very large	significant
Full vs. GAFB + MSAF	+2.3	<0.01	very large	significant

5.4. Parameter Sensitivity Analysis

To further evaluate the robustness of YOLO-GCM with respect to key hyperparameters, we conducted a parameter sensitivity analysis on the FoggyCar validation set. Three representative parameters were considered, including input image size, learning rate, and NMS IoU threshold. For each experiment, only one parameter was changed while all other training and evaluation settings were kept identical to the main experiments. The input

size and learning rate experiments were conducted by retraining YOLO-GCM under the same training protocol, whereas the NMS IoU threshold experiment was performed using the same trained checkpoint during validation.

As shown in Table 9, YOLO-GCM maintained stable performance under different parameter settings. For input image size, increasing the resolution from 512 to 768 slightly improved the detection accuracy, especially mAP@0.5:0.95, but also reduced the inference speed. The default input size of 640 × 640 achieved a favorable trade-off between accuracy and efficiency. For learning rate, the model obtained the best performance around the default value, while smaller or larger learning rates only led to limited performance degradation, indicating stable optimization behavior. For the NMS IoU threshold, the changes in Precision, Recall, and mAP were relatively small, suggesting that the proposed detector is not overly sensitive to post-processing parameter selection. Overall, these results demonstrate the robustness of YOLO-GCM to reasonable hyperparameter variations.

Table 9. Parameter sensitivity analysis of YOLO-GCM on the FoggyCar dataset.

Parameter	Value	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	FPS
Input size	512	90.7	78.4	89.03	66.72	9.45
Input size	640	91.4	79.5	89.81	67.99	8.16
Input size	768	91.5	79.7	89.96	68.08	6.70
Learning rate	0.005	91.0	79.1	89.38	67.45	8.16
Learning rate	0.010	91.4	79.5	89.81	67.99	8.16
Learning rate	0.020	90.8	78.7	89.16	67.05	8.16
NMS IoU	0.50	91.8	78.8	89.60	67.70	8.18
NMS IoU	0.60	91.4	79.5	89.81	67.99	8.16
NMS IoU	0.70	91.1	79.7	89.69	67.82	8.15

5.5. Discussion of Fog-Oriented Baselines

The fog-oriented baseline design warrants separate discussion, especially because restoration models are often optimized for perceptual image quality rather than downstream detection accuracy. Therefore, using only fixed DCP, FFA-Net, or DehazeFormer-B as preprocessing baselines may underestimate the potential of restoration-assisted detection. To provide a more balanced comparison, this study includes both task-agnostic fixed dehazing baselines and detector-aware dehazing baselines. The comparison between FFA-Net and FFA-Net-DA, as well as between DehazeFormer-B and DehazeFormer-DA, shows that detector-aware optimization improves detection performance. This observation confirms that dehazing modules can become more suitable for object detection when optimized with downstream detection feedback. Therefore, the added detector-aware baselines provide a stronger and fairer reference than fixed perceptual dehazing alone [27,36,45–50].

Nevertheless, YOLO-GCM still achieved the highest mAP@0.5 and mAP@0.5:0.95 among all compared methods, including detector-aware dehazing baselines. This suggests that for foggy traffic detection, explicitly restoring image appearance is not the only effective solution. Directly enhancing detector-side features can provide comparable or better detection robustness while avoiding the additional restoration stage.

The results indicate that dehazing preprocessing can indeed help sustain strong detection performance in foggy scenes, particularly when paired with a capable detector such as YOLO11n. However, the performance advantage of dehazing-assisted pipelines over the proposed end-to-end detector is not substantial. This finding suggests that detector-side feature enhancement constitutes a viable and competitive alternative to multi-stage restoration pipelines.

From a deployment perspective, this distinction carries practical significance. A dehazing-plus-detection system requires additional model loading, extra preprocessing

latency, and separate maintenance of restoration weights. In contrast, the proposed method performs detection directly on the original foggy input image. Consequently, the proposed method combines the highest mAP@0.5:0.95 among all compared methods in Table 3 with a more compact and streamlined deployment pipeline, thereby strengthening its practical applicability.

Representative qualitative detection examples are presented in Figure 11. The predictions demonstrate that the final proposed model can preserve dense object responses in low-visibility traffic scenes while maintaining reasonable localization quality across multiple object categories.

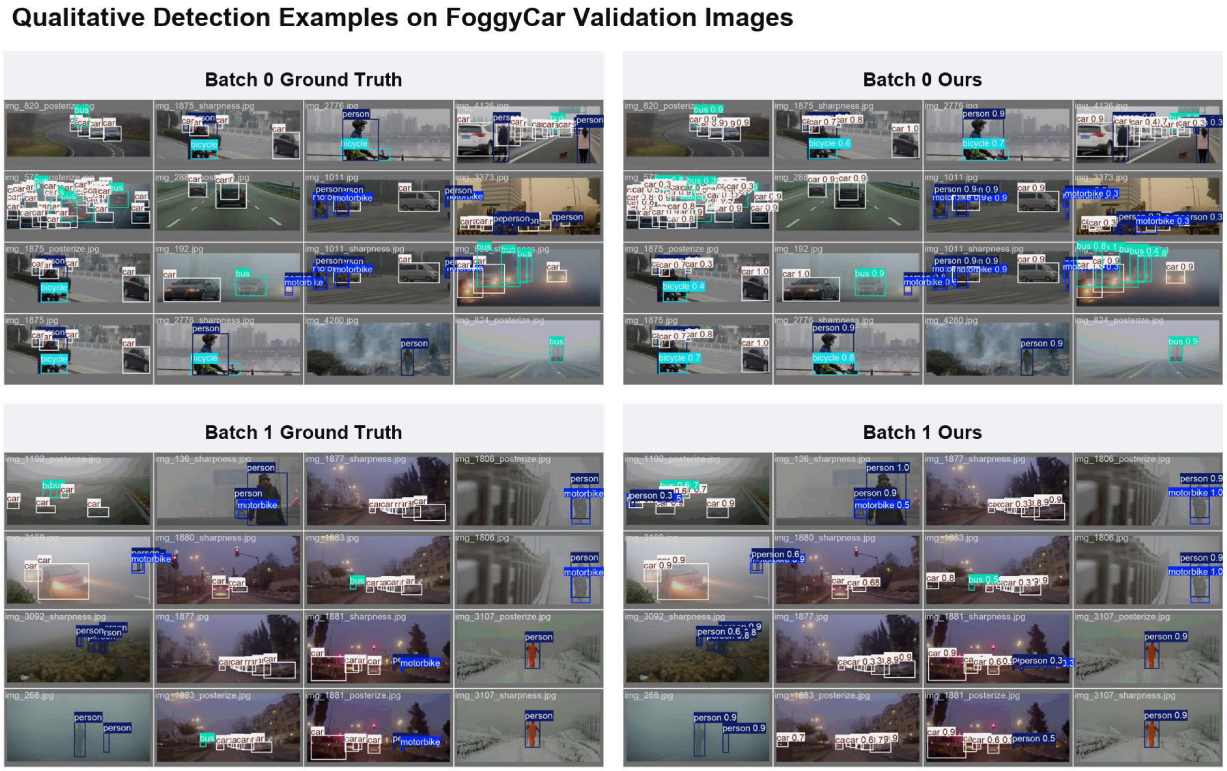


Figure 11. Representative qualitative examples from the FoggyCar validation set. Ground-truth annotations and predictions of the final proposed model are shown for two representative validation batches.

5.6. Feature Visualization and Analysis

To further examine whether the improvement of the proposed module is related to fog-relevant feature selection rather than only black-box detection accuracy, we visualized the feature responses of YOLO11n, the SE-based variant, and the proposed method, as shown in Figure 12.

As shown in Figure 12, the baseline YOLO11n produced relatively weak, sparse, and fragmented activations under foggy conditions. This indicates that the standard detector has difficulty maintaining stable responses to low-contrast traffic objects when object boundaries and local textures are weakened by haze. After introducing SE, the feature responses become stronger than those of YOLO11n, which suggests that channel recalibration can enhance certain discriminative responses. However, the SE-based variant still showed relatively scattered activations and tended to respond to broader haze-dominated background regions. This is because SE mainly relies on global channel statistics, and the global pooling operation may mix object-related regions with foggy background regions when the degradation is spatially non-uniform.

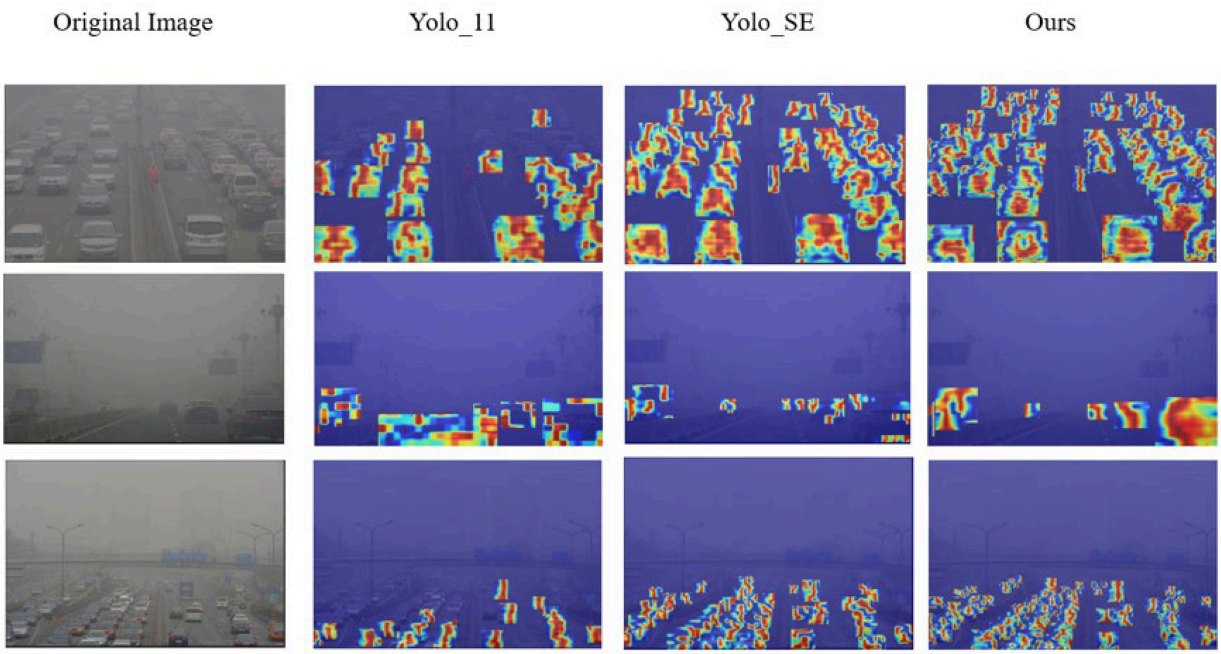


Figure 12. Feature response comparison among the original foggy image, baseline YOLO11n, the SE-based variant, and the proposed method. Compared with YOLO11n, the SE-based variant enhanced more feature responses but still showed scattered activations in haze-dominated background regions. In contrast, the proposed method produced stronger, more continuous, and more object-focused responses, indicating that the CGLU-based gated transformation improves feature selectivity under foggy conditions.

In contrast, the proposed method generated more continuous and object-focused activations. The high-response regions were more consistently distributed around traffic objects such as vehicles and buses, while activations in low-informative foggy background areas were relatively suppressed. This difference indicates that the proposed CGLU-based GAFB does not simply amplify all channel responses. Instead, the multiplicative gate performs input-dependent modulation of local feature responses, allowing the detector to retain semantically discriminative object cues while reducing haze-corrupted or background-dominated activations.

Therefore, the visualization results provide mechanism-level evidence for the advantage of the proposed gated feature transformation over SE-style channel recalibration. Although the proposed module does not explicitly estimate the atmospheric transmission map, its feature-space gating behavior is consistent with the objective of suppressing non-discriminative haze interference and preserving object-related responses under low-visibility conditions.

5.7. Source-Wise Domain Discrepancy Analysis

To further address the source-wise difference between RTTS and the self-collected subset, we conducted a dataset-level domain discrepancy analysis. Unlike the detection-performance comparison, this experiment does not aim to evaluate model accuracy. Instead, it aims to verify whether the two image sources exhibit statistically different visual distributions. The t-SNE visualization and domain-classifier results are shown in Figure 13.

For each image, we extracted a set of low-level image statistics, including brightness, saturation, contrast, Laplacian sharpness, edge density, dark-channel response, and entropy. These features describe illumination, color distribution, texture complexity, image clarity, and haze-related appearance. Hellinger distance was used to measure the magnitude

of distribution discrepancy, while the Kolmogorov–Smirnov test was used to evaluate whether the two feature distributions were statistically different.

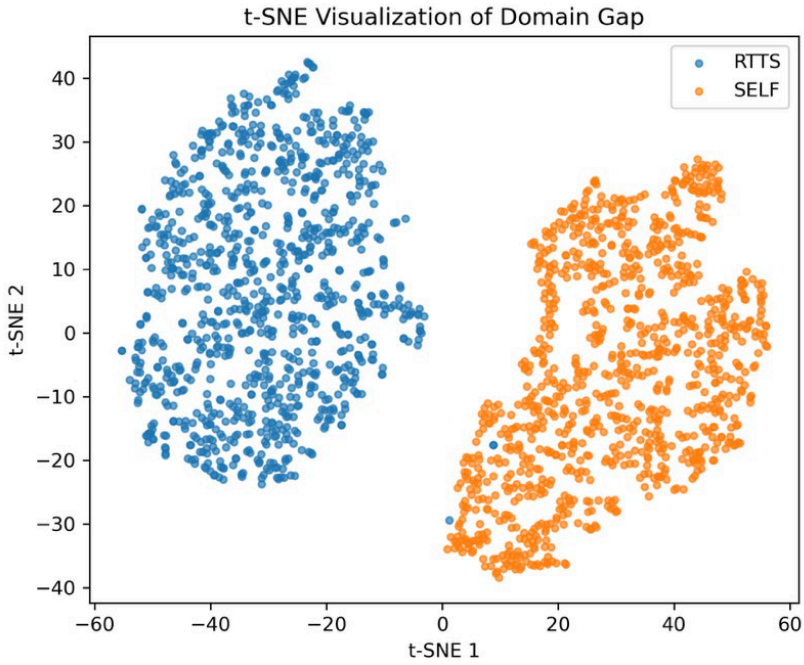


Figure 13. t-SNE visualization of the feature distributions of RTTS and the self-collected subset.

As shown in Table 10, the RTTS subset and the self-collected dataset exhibited significant differences in multiple image-level statistical features. Specifically, the KS test results show that all selected features had p-values smaller than 0.05, indicating statistically significant distribution discrepancies between the two data sources. In addition, the Hellinger distances were non-zero for all features, further confirming that RTTS and the self-collected dataset had different feature distributions. These differences are reflected in the brightness, saturation, contrast, sharpness, edge density, dark-channel response, and entropy, suggesting that the two datasets differ in illumination, color distribution, image clarity, haze-related appearance, and texture complexity.

Table 10. Source-wise feature distribution discrepancy between RTTS and the self-collected subset.

Feature	RTTS Mean	Self-Collected Mean	Hellinger Distance	KS Statistic
Brightness	0.4520	0.4789	0.1664	0.1170
Saturation	0.1204	0.1098	0.4313	0.2705
Contrast	0.1542	0.1871	0.2722	0.2845
Sharpness	197.9320	105.8477	0.2921	0.2020
Edge density	0.0236	0.0252	0.2894	0.2200
Dark channel	0.3499	0.3753	0.1875	0.0980
Entropy	6.2012	7.0109	0.4876	0.4885

As shown in Figure 13, the RTTS subset and the self-collected dataset formed two clearly separated clusters in the t-SNE feature space. This visualization result indicates an obvious source-wise distribution shift between the two datasets.

5.8. Feature-Level Cross-Domain Discrepancy Analysis

To further investigate whether the improvement on Foggy Cityscapes originated from more domain-invariant feature representation rather than overfitting to fog-specific artifacts, we conducted a feature-level cross-domain discrepancy analysis between FoggyCar

and Foggy Cityscapes. Specifically, intermediate features were extracted from the early backbone stage, the SPPF output, and the neck-fusion stage of YOLO11n and YOLO-GCM. We then computed the CORAL distance and MMD between source-domain and target-domain features. In addition, a lightweight domain classifier was trained to distinguish whether a feature came from FoggyCar or Foggy Cityscapes. Lower CORAL/MMD values and lower domain classification accuracy indicate stronger domain invariance. The feature-level domain discrepancy results are summarized in Table 11.

Table 11. Feature-level domain discrepancy between FoggyCar and Foggy Cityscapes.

Model	Feature Layer	CORAL	MMD	Domain Acc.
YOLO11n	Early backbone	0.043	0.097	83.1%
YOLO-GCM	Early backbone	0.039	0.090	80.4%
YOLO11n	SPPF/high-level	0.035	0.084	80.2%
YOLO-GCM	CAFEM/high-level	0.024	0.061	72.6%
YOLO11n	Neck fusion	0.038	0.089	81.5%
YOLO-GCM	MSAF/neck fusion	0.026	0.064	71.3%

As shown in Table 11, theresults show that YOLO-GCM consistently reduced feature-level discrepancy compared with YOLO11n, especially at the high-level semantic and multi-scale fusion stages. This suggests that the proposed GAFB, CAFEM, and MSAF modules do not merely fit fog-specific low-level artifacts, but help the detector learn more stable representations under domain shift. Combined with the improved detection performance on Foggy Cityscapes, this analysis provides additional evidence for the cross-domain robustness of the proposed detector-side enhancement strategy.

5.9. Generalization Evaluation on Foggy Cityscapes

As described in Section 4.1, Foggy Cityscapes was evaluated under the full eight-class label space. The five shared categories were evaluated directly through class mapping, while truck, rider, and train were treated as unmatched categories with AP = 0 and included in the final mAP calculation.

To further evaluate the generalization capability of the proposed method, additional experiments were conducted on the Foggy Cityscapes dataset, which is a widely used public benchmark for foggy scene understanding. Different from the FoggyCar dataset used in the main experiments, Foggy Cityscapes provides synthetic fog with varying density levels and diverse urban scenarios, making it suitable for cross-dataset evaluation. This experiment examined whether the proposed detector-side feature enhancement strategy remained effective under domain shift.

Without any additional fine-tuning or domain adaptation, both the baseline YOLO11n and the proposed YOLO-GCM were directly evaluated on the Foggy Cityscapes validation set. The cross-dataset detection results are summarized in Table 12.

Table 12. Generalization performance on the Foggy Cityscapes dataset.

Method	P (%)	R (%)	mAP@0.5	mAP@0.5:0.95	Params (M)	FLOPs (G)
YOLO11n	51.6	36.7	37.5	20.9	2.5	7.2
Ours	59.2	40.7	43.8	24.4	2.3	7.1

As shown in Table 12, the proposed method consistently outperformed the baseline YOLO11n across all evaluation metrics. Specifically, YOLO-GCM improved mAP@0.5 from 37.5% to 43.8% and mAP@0.5:0.95 from 20.9% to 24.4%, demonstrating a clear advantage under domain shift conditions. Meanwhile, Precision and Recall were also improved by 7.6% and 4.0%, respectively.

It is worth noting that these improvements were achieved without increasing the computational complexity. On the contrary, the proposed method slightly reduced the number of parameters and FLOPs compared with the baseline, indicating that the performance gain mainly originates from more effective feature representation rather than model scaling.

For clearer visualization of the cross-dataset generalization performance, the results on Foggy Cityscapes are further illustrated in Figure 14 using a radar chart. Since the compared metrics have different numerical scales, all values were normalized before plotting. In addition, for Params and FLOPs, inverse normalization was adopted so that better efficiency corresponds to a larger radar value.

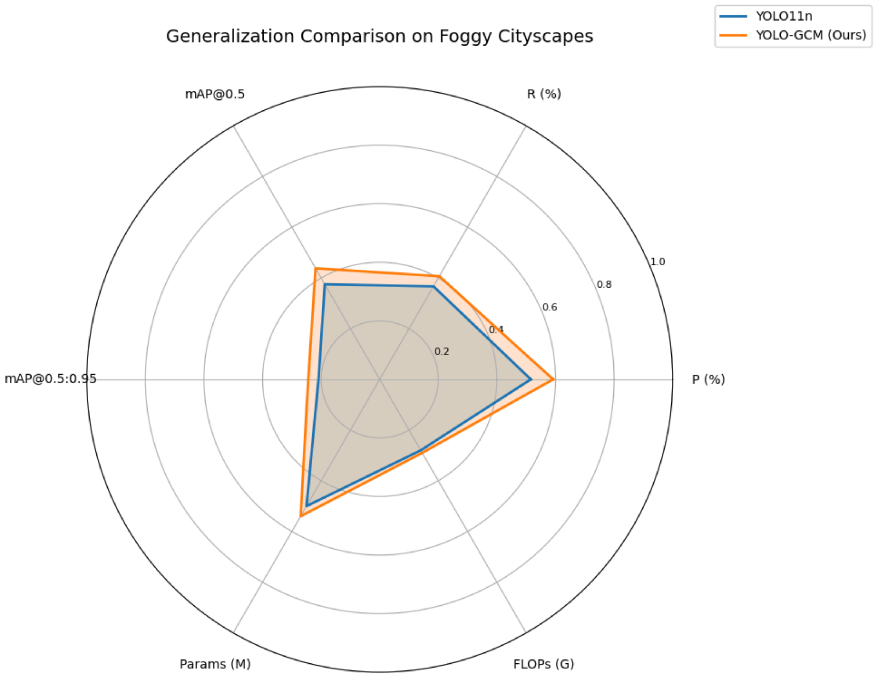


Figure 14. Radar-chart comparison of YOLO11n and the proposed YOLO-GCM on the Foggy Cityscapes dataset. All metrics were normalized for visualization. For Precision, Recall, mAP@0.5, and mAP@0.5:0.95, higher values indicate better performance. For Params and FLOPs, inverse normalization was applied so that a larger radius consistently represents a more favorable result.

As shown in Figure 14, the proposed YOLO-GCM achieved a consistently larger enclosed area than the baseline YOLO11n, indicating a better overall balance between detection accuracy and computational efficiency. In particular, the proposed method exhibited clear advantages in Precision, Recall, mAP@0.5, and mAP@0.5:0.95, while also maintaining a slightly lower parameter count and FLOPs. This result further demonstrates that the proposed detector-side enhancement strategy not only provides stronger detection performance, but also favorable model compactness under cross-dataset evaluation.

Overall, the results confirm that the proposed detector-side enhancement strategy not only improves performance on the target dataset but also exhibits strong generalization ability to unseen foggy environments. This property is particularly important for real-world deployment, where environmental conditions may vary significantly from the training data.

5.10. Failure Analysis and Limitations

Despite the promising results, several limitations of this study should be acknowledged.

First, although the proposed method consistently outperformed the baseline and dehazing-assisted approaches, the performance margin over the strongest YOLO11n-based baseline remained moderate. This suggests that while detector-side enhancement is effec-

tive, further exploration of more expressive feature modeling strategies may be required to achieve more substantial gains under severe visibility degradation.

Second, the current study focused on a single lightweight detector family. Although the proposed GAFB, CAFEM, and MSAF modules demonstrated effectiveness within the YOLO11n framework, their general applicability to other detection architectures, such as transformer-based detectors or larger-scale models, remains to be systematically investigated.

Third, the present work primarily evaluated robustness under general foggy conditions, without explicitly modeling different levels of fog density or spatially non-uniform degradation. A more fine-grained analysis of how feature representations degrade under varying atmospheric conditions could provide deeper insights into the design of adaptive detection mechanisms.

A further limitation lies in the scale aggregation strategy of MSAF. Although MSAF aligns P3, P4, and P5 features and models their scale–sequence interaction through Conv3D before aggregation, the final $3 \times 1 \times 1$ MaxPool3D operation still performs hard selection along the scale dimension. This may discard non-maximum scale responses that could contain complementary information for small or low-contrast objects. In future work, we will investigate learnable soft scale-attention or confidence-guided scale routing to replace hard max pooling. Such a design would allow P3/P4/P5 responses to be continuously weighted and would make it possible to quantitatively analyze whether the learned scale weights are distributed across multiple scales or collapse to a single dominant scale.

To further complement the aggregate mAP results, we provide a detection reliability analysis of the final proposed model, as shown in Figure 15. The precision–recall curves report the class-wise detection performance, while the confusion matrix presents the major misclassification and background-confusion patterns on the FoggyCar validation set.

As shown in Figure 15, the proposed model achieved stable precision–recall behavior across all five traffic-related categories. The car and person categories obtained relatively higher AP values, whereas motorbike and bicycle showed slightly lower AP, mainly because these objects are often smaller, partially occluded, or more visually ambiguous under foggy conditions. The confusion matrix further indicates that most correct predictions were concentrated along the diagonal, demonstrating reliable category-level discrimination. The remaining errors were mainly associated with background confusion, especially for small or low-contrast objects, which is consistent with the visual degradation caused by fog.

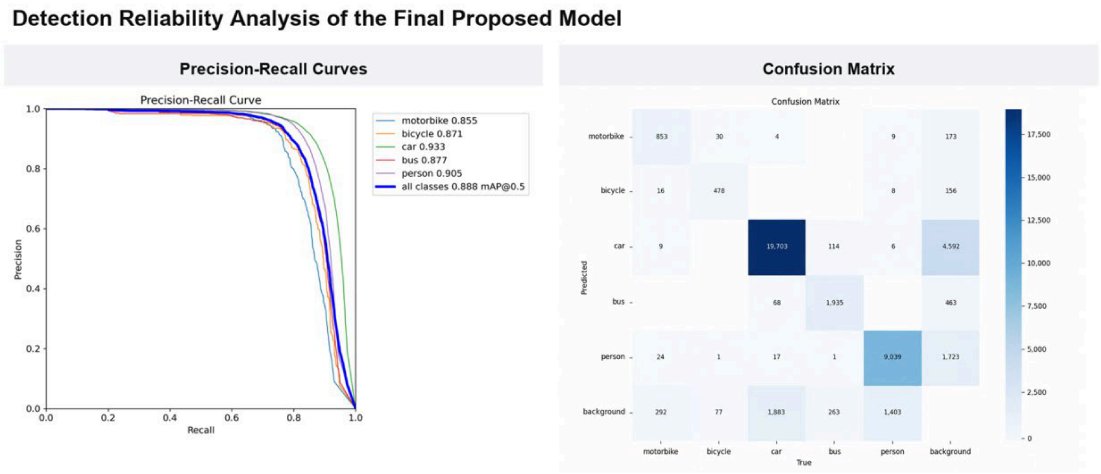


Figure 15. Detection reliability analysis of the final proposed model. The left panel shows the class-wise precision–recall curves, and the right panel shows the confusion matrix on the FoggyCar validation set.

Finally, although this work provides an architectural explanation for the effectiveness of GAFB, CAFEM, and MSAF, the analysis remains mainly based on detection performance and feature visualization. Future work could introduce more quantitative feature-level metrics to evaluate how different modules affect channel responses, contextual coherence, and multi-scale representation under varying fog densities.

6. Conclusions

This study investigated whether a lightweight detector can be effectively strengthened for foggy traffic object detection without relying on an external dehazing stage. Building upon YOLO11n, we proposed YOLO-GCM, a detector-side feature enhancement framework comprising a CAFEM block inserted after the spatial pyramid pooling–fast stage, GAFB units for nonlinear feature extraction and adaptive information transformation, and an MSAF module for adaptive multi-scale feature fusion. The resulting detector preserves the original one-stage detection pipeline while improving robustness to low contrast, blurred object boundaries, and suppressed local details under foggy conditions.

Extensive experiments on the FoggyCar dataset demonstrated that the proposed method achieved 89.81% mAP@0.5 and 67.99% mAP@0.5:0.95, surpassing standard lightweight YOLO baselines and dehazing-assisted comparison methods under a consistent detector backend. Additional evaluation on Foggy Cityscapes further confirmed the generalization ability of YOLO-GCM under domain shift. These findings suggest that detector-side feature enhancement can provide an effective and efficient alternative to multi-stage dehazing-plus-detection pipelines for foggy traffic perception.

The experimental results demonstrate that the proposed YOLO-GCM effectively improves foggy traffic object detection while preserving a lightweight one-stage architecture. On the FoggyCar validation set, YOLO-GCM achieved 89.81% mAP@0.5 and 67.99% mAP@0.5:0.95, outperforming the standard YOLO11n baseline and the compared dehazing-assisted detection pipelines under the same evaluation protocol. The ablation results further indicate that GAFB, CAFEM, and MSAF contribute complementary improvements by enhancing adaptive feature selection, high-level contextual representation, and multi-scale feature interaction, respectively. These findings suggest that detector-side feature enhancement is an effective alternative to sequential dehazing-plus-detection pipelines for low-visibility traffic scenarios. In practical applications, the proposed method can support more reliable roadside monitoring and intelligent transportation perception under foggy conditions. Future work will focus on evaluating the framework under more complex adverse-weather scenarios, such as night fog, mixed rain-fog conditions, and real edge-device deployment.

Future work will further investigate class-wise detection behavior, robustness under different fog density levels, and the transferability of the proposed modules to other lightweight object detection frameworks.

Author Contributions: Conceptualization, H.H. and J.W.; Methodology, J.W.; Software, J.W.; Validation, J.W.; Data Curation, J.W.; Writing—Original Draft Preparation, J.W.; Writing—Review & Editing, J.W. and H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors thank the editor and reviewers for their valuable suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. He, K.; Sun, J.; Tang, X. Single Image Haze Removal Using Dark Channel Prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009*; IEEE: New York, NY, USA, 2009; pp. 1956–1963.
2. He, K.; Sun, J.; Tang, X. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2341–2353. [PubMed]
3. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2016; pp. 3213–3223.
4. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2012; pp. 3354–3361.
5. Li, B.; Ren, W.; Fu, D.; Tao, D.; Feng, D.; Zeng, W.; Wang, Z. Benchmarking Single-Image Dehazing and Beyond. *IEEE Trans. Image Process.* **2019**, *28*, 492–505.
6. Macioszek, E.; Wyderka, A.; Jurdana, I. The Bicyclist Safety Analysis Based on Road Incidents Maps. *Sci. J. Silesian Univ. Technol. Transp.* **2025**, *126*, 129–147. [CrossRef]
7. Meng, X.; Liu, Y.; Fan, L.; Fan, J. YOLOv5s-Fog: An Improved Model Based on YOLOv5s for Object Detection in Foggy Weather Scenarios. *Sensors* **2023**, *23*, 5321. [CrossRef] [PubMed]
8. 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, Las Vegas, NV, USA, 27–30 June 2016*; IEEE: New York, NY, USA, 2016; pp. 779–788.
9. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
10. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision, Munich, Germany*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–19.
11. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*; IEEE: New York, NY, USA, 2018; pp. 7132–7141.
12. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language Modeling with Gated Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia*; PMLR: Cambridge, MA, USA, 2017; pp. 933–941.
13. Zhong, F.; Shen, W.; Yu, H.; Wang, G.; Hu, J. Dehazing & Reasoning YOLO for Object Detection in Foggy Weather. *Pattern Recognit.* **2024**, *156*, 110756.
14. Zhang, K.; Yan, X.; Wang, Y.; Qi, J. Adaptive Dehazing YOLO (DH-YOLO). In Proceedings of the International Conference on Artificial Neural Networks, Crete, Greece, 26–29 September 2023.
15. Ultralytics. Ultralytics YOLO Repository and Documentation. Available online: https://github.com/ultralytics/ultralytics (accessed on 10 April 2026).
16. Michael, C.; Wang, H. YOLOv11-TWCS: Enhancing Object Detection for Autonomous Vehicles in Adverse Weather Conditions Using YOLOv11 with TransWeather Attention. *Vehicles* **2026**, *8*, 16. [CrossRef]
17. Zhou, K.; Jiang, S. Forest Fire Detection Algorithm Based on Improved YOLOv11n. *Sensors* **2025**, *25*, 2989. [CrossRef] [PubMed]
18. Wang, W.; Liu, L.; Dai, L.; An, M.; Khadka, A. Small Traffic Sign Detection Based on Improved YOLOv11n. In *Proceedings of the International Conference on Optoelectronics, Computer Science, and Algorithms*; SPIE: Bellingham, WA, USA, 2026; Volume 14008, pp. 82–87.
19. Li, X.; Chen, J.; Sun, Y.; Lin, N.; Hawbani, A.; Zhao, L. YOLO-Vehicle-Pro: A Cloud-Edge Collaborative Framework for Object Detection in Autonomous Driving under Adverse Weather Conditions. *arXiv* **2024**, arXiv:2410.17734.
20. Zhang, Y.; Jia, N. HR-YOLO: A Target Detection Model for Foggy Driving Scenes. *Sci. Rep.* **2025**, *15*, 1–19. [CrossRef]
21. Wang, H.; Shi, Z.; Zhu, C. Enhanced Multi-Scale Object Detection Algorithm for Foggy Traffic Scenarios. *Comput. Mater. Contin.* **2024**, *82*, 2451–2474.
22. Yang, Y.; You, S.; Yu, J.; Lu, B. FogGate-YOLO: Traffic Object Detection in Foggy Environments Using Channel Selection Mechanisms. *Sensors* **2026**, *26*, 1811. [CrossRef] [PubMed]
23. Wang, Q.; Zhou, Z.; Zhang, Z. Fog-YOLO: A Real-Time Object Detector for Foggy Autonomous Driving Scenarios. *Symmetry* **2026**, *18*, 322. [CrossRef]
24. Fang, W.; Zhang, G.; Zheng, Y.; Chen, Y. Multi-Task Learning for UAV Aerial Object Detection in Foggy Weather Condition. *Remote Sens.* **2023**, *15*, 4617. [CrossRef]
25. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. CARAFE: Content-Aware ReAssembly of FEatures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea*; IEEE: New York, NY, USA, 2019; pp. 3007–3016.

26. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA*; IEEE: New York, NY, USA, 2020; pp. 10781–10790.
27. Chu, Z. D-YOLO: A Robust Framework for Object Detection in Adverse Weather Conditions. *arXiv* **2024**, arXiv:2403.09233v1.
28. Shi, X.; Song, A. Defog YOLO for Road Object Detection in Foggy Weather. *Comput. J.* **2024**, *67*, 3115–3127. [CrossRef]
29. Qin, X.; Wang, Z.; Bai, Y.; Xie, X.; Jia, H. FFA-Net: Feature Fusion Attention Network for Single Image Dehazing. In *Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020*; AAAI Press: Palo Alto, CA, USA, 2020; Volume 34, pp. 11908–11915.
30. Li, B.; Peng, X.; Wang, Z.; Xu, J.; Feng, D. AOD-Net: All-In-One Dehazing Network. In *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017*; IEEE: New York, NY, USA, 2017; pp. 4770–4778.
31. Liu, X.; Ma, Y.; Shi, Z.; Chen, J. GridDehazeNet: Attention-Based Multi-Scale Network for Image Dehazing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019*; IEEE: New York, NY, USA, 2019; pp. 7314–7323.
32. Ren, W.; Liu, S.; Zhang, H.; Pan, J.; Cao, X.; Yang, M.-H. Single Image Dehazing via Multi-Scale Convolutional Neural Networks. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 154–169.
33. Qiu, Y.; Lu, Y.; Wang, Y.; Jiang, H. IDOD-YOLOV7: Image-Dehazing YOLOV7 for Object Detection in Low-Light Foggy Traffic Environments. *Sensors* **2023**, *23*, 1347. [CrossRef] [PubMed]
34. Song, Y.; He, Z.; Qian, H.; Du, X. Vision Transformers for Single Image Dehazing. *IEEE Trans. Image Process.* **2023**, *32*, 1927–1941. [CrossRef] [PubMed]
35. Aryashad, A.; Razmara, P.; Mahjoub, A.; Azizi, S.; Salmani, M.; Firouzkouhi, A. From Filters to VLMs: Benchmarking Defogging Methods through Object Detection and Segmentation Performance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*; IEEE: New York, NY, USA, 2026.
36. Zhao, X.; Chen, Y. YOLO-DroneMS: Multi-Scale Object Detection Network for Unmanned Aerial Vehicle Images. *Drones* **2024**, *8*, 609. [CrossRef]
37. Gharatappeh, S.; Neshatfar, S.; Sekeh, S.Y.; Dhiman, V. FogGuard: Guarding YOLO Against Fog Using Perceptual Loss. *arXiv* **2024**, arXiv:2403.08939.
38. Liang, X.; Liang, Z.; Li, L.; Chen, J. AODs-CLYOLO: An Object Detection Method Integrating Fog Removal and Detection in Haze Environments. *Appl. Sci.* **2024**, *14*, 7357. [CrossRef]
39. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2020**, arXiv:2010.04159.
40. Sakaridis, C.; Dai, D.; Van Gool, L. Semantic Foggy Scene Understanding with Synthetic Data. *Int. J. Comput. Vis.* **2018**, *126*, 973–992. [CrossRef]
41. Wang, Z.; Zhou, W.; Li, Y. MSAF-YOLO: An Efficient Multi-Scale Attention Fusion Network for High-Precision Steel Surface Defect Detection. *Measurement* **2025**, *257*, 118640.
42. Li, R.; Chen, Y.; Wang, Y.; Sun, C. YOLO-TSF: A Small Traffic Sign Detection Algorithm for Foggy Road Scenes. *Electronics* **2024**, *13*, 3744. [CrossRef]
43. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2009; pp. 248–255.
44. Jiang, L.; Ma, G.; Guo, W.; Sun, Y. YOLO-DH: Robust Object Detection for Autonomous Vehicles in Adverse Weather. *Electronics* **2025**, *14*, 4476. [CrossRef]
45. Guo, Z.; Zhong, M.; Duan, S.; Wang, L. ESGN-YOLO: Enhancing Multi-Scale Small Object Detection via Efficient Feature Fusion and Adaptive Spatial Modeling. *IEEE Signal Process. Lett.* **2025**, *33*, 426–430.
46. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to Upsample by Learning to Sample. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2023; pp. 6027–6037.
47. Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J.; Zhang, L. Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions. *arXiv* **2021**, arXiv:2112.08088.
48. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725.
49. Liu, Z.; Zhang, J.; Zhang, X.; Song, H. Robust Object Detection in Adverse Weather Conditions: ECL-YOLOv11 for Automotive Vision Systems. *Sensors* **2026**, *26*, 304. [CrossRef] [PubMed]
50. Wang, L.; Zhang, B.; Tan, Q.; Duan, D.; Wang, Y. Hazy Aware-YOLO: An Enhanced UAV Object Detection Model for Foggy Weather via Wavelet Convolution and Attention-Based Optimization. *Automation* **2026**, *7*, 3. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.