Digital Intelligence PRESS

Journal of Discovery Core

https://doi.org/10.67541/jdc2604

Research Article **Volume 1, Issue 1 (July 2026)** *Open Access*

Hybrid Architecture Edge Object Detector Based on Structured Pruning and Attention Reparameterization

Ruisheng Zhang *

Shenzhen Yunlang Network Technology Co., Ltd., Shenzhen, Guangdong, China

***Corresponding author:** Ruisheng Zhang. Email: zhangruishengZZ@163.com

Abstract

Edge computing scenarios put forward strict requirements on inference delay and computational power consumption of target detectors, and the existing lightweight models are difficult to achieve the best balance between accuracy and speed. In this paper, a hybrid architecture edge object detector based on structured pruning and attention reparameterization is proposed, which is co-optimized from three levels: architecture design, model compression and inference acceleration. Firstly, a CNN-Transformer dual-branch heterogeneous backbone network is constructed, and the basic mAP is increased to 44.2% on the premise of only increasing the number of parameters by 15% by fusing local details and global context features through cross-modal interaction modules. Then, a structured pruning algorithm driven by hierarchical sensitivity is proposed, which combines Hessian trace and batch normalization scaling factor to dynamically determine the redundancy of each layer. The differentiable mask and Group Lasso sparse regularity are introduced to realize training, namely pruning. With the layer-by-layer feature reconstruction compensation mechanism, the accuracy retention rate can reach 97.5% when FLOPs are reduced by 50.4%. The attention reparameterization technique during inference is further proposed, which uses the linear additivity of affine transformations to losslessly fuse the multi-branch attention structure in the training stage into a single convolutional layer, and combines piecewise linear fitting to deal with Softmax nonlinearity, completely eliminating the attention computation cost during inference. Experiments show that the proposed method achieves 43.0% mAP on COCO dataset, 112.3 FPS on Jetson Xavier, and the model size is only 5.8 MB. In terms of accuracy, speed and model size, it is superior to mainstream methods such as YOLOv8n, MobileDet and EfficientDet-D0, and also shows optimal robustness in edge scenes such as low light and motion blur.

Keywords

Edge object detection; Structured pruning; Attention reparameterization; Hybrid architecture; Model compression

Received: 20 Jun 2026; Revised: 18 Jul 2026; Accepted: 28 Jul 2026; Published: 07 Aug 2026

1. INTRODUCTION

With the rapid evolution of edge computing technology, real-time perception applications such as autonomous driving, industrial vision quality inspection and intelligent monitoring put forward increasingly stringent deployment requirements for target detection models [1],[2]. Such scenarios are often constrained by the limited computing resources and battery capacity of embedded devices, requiring accurate object positioning and recognition within millisecond response time while maintaining low power consumption. In order to meet such requirements, a series of lightweight convolutional neural networks such as MobileNet, ShuffleNet and EfficientNet-Lite have been

proposed in academia and industry, which greatly reduce the number of model parameters and floating-point operations by means of deep separated convolution, channel shuffling and neural architecture search [3],[4]. However, these lightweight models still face a significant precision-speed trade-off dilemma on resource-constrained edge devices: When the width or depth of the network is excessively compressed to meet the requirements of very low latency, the feature extraction ability declines sharply, especially in small target detection and positioning accuracy under complex background, which is difficult to meet the high standard requirements of detection reliability in practical engineering tasks [5],[6],[7].

In recent years, model compression technology, especially structured pruning, provides an effective way to alleviate the above contradictions. By removing channels, filters or attention heads in an organized manner, the computational cost can be significantly reduced on the premise of maintaining the regularity of network structure [8],[9]. However, existing pruning methods suffer from an inherent flaw: the degradation of feature representation during pruning. Simple removal of redundant units will lead to irreversible shift of feature distribution received by subsequent layers, and it is difficult to fully restore the representation ability of the original model even after fine-tuning [10],[11]. The accuracy attenuation is especially serious when the pruning rate is high. At the same time, Transformer architecture and its self-attention mechanism have made breakthroughs in the field of object detection with their excellent global modeling ability, and their unique advantage of capturing long-range dependencies just makes up for the limited receptive field of CNN. However, the attention mechanism involves a large number of matrix multiplications and Softmax normalization operations in the inference stage, and its computational complexity and memory access cost are far higher than those of convolution operations of the same scale, often accounting for more than 30% of the total inference time on edge devices. This severely restricts the direct deployment of Transformer in real-time edge detection scenarios.

To solve the above dual challenges, this paper proposes a new solution path: constructing a hybrid architecture edge object detector, which organically integrates the local inductive bias of CNNs with the global context modeling capability of Transformers, and introduces a joint optimization strategy of structured pruning and attention reparameterization to achieve optimal adaptation across the training and inference stages. This core idea consists of three levels: first, the dual-branch heterogeneous backbone network is designed, the shallow CNN branch efficiently extracts texture edge and other detail features, and the deep sparse Transformer branch captures global semantic association with controllable cost, and realizes bidirectional information flow through cross-modal interaction modules, so that the two inductive preferences are mutually enhanced rather than simply superimpose [12],[13],[14]. Secondly, a structured pruning algorithm driven by hierarchical sensitivity is introduced to dynamically decide the pruning granularity of each layer, and the differentiable mask sparse training and layer-by-layer feature reconstruction compensation mechanism are combined to compress the model size and retain the discriminant feature expression to the maximum extent. Third, the attention reparameterization mechanism is proposed during inference, which maintains the multi-branch attention structure in the training stage to fully tap the representation potential. Before inference, multiple attention branches are equivalently fused into a single convolution layer by using the affine nature of convolution and fully connected layer, and the Softmax nonlinearity is processed by piecewise linear fitting to completely eliminate the attention computing cost during inference [15]. The three are interlinked, and pruning provides a compact weight basis for subsequent reparameterization, which further reduces inference computation from another dimension, and the synergistic effect is far better than that of independent optimization methods.

The main contributions of this paper can be summarized as the following three points: First, the CNN-Transformer dual-branch heterogeneous backbone network and the supporting cross-modal interaction module are proposed, which greatly improve the diversity and complementarity of feature expression on the premise of only increasing moderate computing redundancy, and provide an initial architecture with abundant accuracy for edge detection. Second, a layered sensitivity driven structured pruning algorithm for edge deployment is designed, including a redundancy criterion based on the joint measurement of Hessian trace and BN scaling factor, a sparse training strategy based on the co-optimization of differentiable mask and Group Lasso, and a layer-by-layer compensation calibration mechanism based on feature map reconstruction error. Precision and lossless pruning at high compression rate is achieved. Third, the attention reparameterization technology during inference is pioneered. By deducing the linear additivity condition of multi-branch attention and combining with Softmax piecewise linear approximation, the rich attention structure in the training stage is lossless converted into a single convolutional layer before inference, which essentially eliminates the negative impact of attention mechanism on edge inference delay.

2. BASIC SKELETON DESIGN OF HYBRID ARCHITECTURE EDGE DETECTOR

In order to obtain high-resolution details and global semantic information on edge devices at the same time, this work abandons the limitations of single network architecture and constructs a dual-branch heterogeneous backbone network. The skeleton consists of two parallel computing paths: shallow efficient CNN branch and deep sparse Transformer branch. Let the input image be $\mathrm{I} \in \mathbb{R}^{H\times W\times 3}$, where H and W are the image height and width, respectively [16],[17]. The CNN branch adopts lightweight depth separable convolution stack. The first three stages of cnn branch include downsampling layer with step size of 2, which gradually reduce the space size to $1/8$ of the original image, and expand the number of channels to C_{cnn}. The core operation of this branch is the combination of channel-by-channel convolution and point by point convolution, and its calculation amount can be expressed as [18]:

$$\mathrm{FLOPs}_{\mathrm{DS}} = \sum_{l=1}^{L}\left(k_l^2 \cdot C_{\mathrm{in}}^{(l)} \cdot H_l \cdot W_l + C_{\mathrm{in}}^{(l)} \cdot C_{\mathrm{out}}^{(l)} \cdot H_l \cdot W_l\right) \tag{1}$$

Where k_l is the size of the l-th layer convolution kernel, $C_{\mathrm{in}}^{(l)}$ and $C_{\mathrm{out}}^{(l)}$ are the number of input and output channels respectively, and H_l and W_l are the spatial dimensions of feature graphs [19]. L is the total number of convolution layers. This branch aims to capture high-frequency local responses such as edges, corners and textures with few parameters to provide accurate spatial localization clues for subsequent detection [20].

At the same time, the deep Transformer branch receives the low-resolution feature map ($1/16$ of the space size of the original map) after the 1×1 convolution projection and flattens it into a token sequence $\mathrm{T} \in \mathbb{R}^{N\times d}$, Where $N = \frac{H}{16}\cdot\frac{W}{16}$ is the number of tokens and d is the embedding dimension. In order to control the computing load on edge devices, this branch adopts the sparse self-attention mechanism to restrict each token to interact only with s nearest tokens and r globally uniformly sampled proxy tokens in the spatial neighborhood, rather than globally fully connected. For the i-th query token, its attention output is:

$$O_i = \sum_{j \in \mathcal{N}(i) \cup \mathcal{G}} \alpha_{ij} \cdot V_j, \alpha_{ij} = \frac{\exp\left(Q_i \cdot \frac{K_j^T}{\sqrt{d}}\right)}{\sum_{t \in \mathcal{N}(i) \cup \mathcal{G}} \exp\left(Q_i \cdot \frac{K_t^T}{\sqrt{d}}\right)} \quad (2)$$

Where $\mathcal{N}(i)$ represents the local neighborhood index set of the i-th token, and the set size is s; $\mathcal{G}$ is the globally sampled token index set with size r and $s + r \ll N$. Q_i, K_j, V_j query, key and the value vector, respectively, obtained by learnable linear projections of the token sequence the sparser design reduces the self-attention complexity from $\mathcal{O}(N^2 d)$ to $\mathcal{O}(N(s+r)d)$, enabling Transformer branch to model the long-term dependence and global context relationship between objects with acceptable cost, effectively making up for the limitation of CNN receptive field.

In order to fully integrate the features of the above two heterogeneous branches rather than simply adding or splicing, the cross-modal feature interaction module is introduced as the bridging unit in this framework. The module contains two streams of information: First, the CNN branch's high-resolution feature map $F_{cnn} \in \mathbb{R}^{h \times w \times C_c}$ Through point-by-point convolution and adaptive average pooling, it is compressed into a bootstrap vector $g \in \mathbb{R}^d$ of the same dimension as the Transformer token, which is used to modulate the key-value matrix of each token in the Transformer branch to realize the guidance of CNN details on the global attention distribution [21]. Secondly, the context enhancement token sequence output by Transformer branch is reshaped into a two-dimensional feature graph, which is weighted by channel attention mechanism and upsampled to the size of CNN feature graph, and added element-by-element with the original CNN feature. In order to ensure the smooth synchronization of gradient among branches, the interaction module adopts the form of residual connection in forward propagation to ensure that the gradient can be transmitted back through two paths without attenuation in back propagation [22],[23]. The whole interaction process can be formally expressed as:

$$F_{fuse} = F_{cnn} + \mathrm{Up}\big(\mathrm{CA}(\mathrm{Reshape}(T_{out}))\big), T_{out} = \mathrm{SA}(T, g) \quad (3)$$

Where $\mathrm{CA}(\cdot)$ represents channel attention operation, $\mathrm{SA}(\cdot)$ represents guided sparse self-attention, $\mathrm{Up}(\cdot)$ is bilinear upsampling, and F_{fuse} is the joint feature representation after fusion. This design not only retains the local detail transfer path of CNN branch, but also maintains the global modeling independence of Transformer branch [24]. The two are optimized together and do not interfere with each other.

After obtaining the fusion feature F_{fuse}, this framework uses the improved bidirectional feature pyramid network (BiFPN) to perform multi-scale feature fusion, so as to construct a feature pyramid with rich semantic information [25]. Suppose that the detection neck contains M levels (corresponding to down-sampling multiples of 8, 16 and 32), and the feature graph of each level i is denoted as P_i. Traditional BiFPN uses fixed weighted summation of each input feature, ignoring the differentiated contributions of different levels to the final detection task. In this work, learnable weight coefficient λ_i (initialized as 1) is introduced, which is normalized by Softmax and used as hierarchical importance factor to dynamically balance the contribution from top-level semantic information and low-level spatial details [26]. The improved fusion output of layer i is calculated as follows:

$$P_i^{out} = \mathrm{Conv}\left(\sum_{j \in \{i, i-1, i+1\}} \frac{\exp(\lambda_j)}{\sum_k \exp(\lambda_k)} \cdot \mathrm{Resize}(P_j)\right) \quad (4)$$

In the formula, Resize(·) adjusts the features of the adjacent layer to the space size of the i-th layer through upsampling or downsampling, and Conv(·) is the 3×3 depth separable convolution after fusion to eliminate the aliasing effect. The learnable weight coefficient is automatically optimized with the loss function during training, so that the network can adaptively emphasize more informative levels according to the data distribution, so as to maximize the multi-scale representation efficiency within a limited computational budget.

Finally, the preliminary complexity theory evaluation of the basic framework is carried out. Compared with the standard single architecture (such as the pure ResNet-50 backbone), the dual-branch heterogeneous design adds additional paths, but through the combination of depthwise separable convolution and sparse self-attention, the overall floating-point computation and parameter number remain in the same order of magnitude [27]. Let the computation amount of the standard CNN backbone be $\mathrm{FLOPs}_{\mathrm{base}}$, and the total computation amount of this skeleton be:

$$\mathrm{FLOPs}_{\mathrm{total}} = \mathrm{FLOPs}_{\mathrm{DS}} + \mathrm{FLOPs}_{\mathrm{SA}} + \mathrm{FLOPs}_{\mathrm{fuse}} + \mathrm{FLOPs}_{\mathrm{BiFPN}} \tag{5}$$

Where sparse self-attention branch $\mathrm{FLOPs}_{\mathrm{SA}} \approx 4N(s+r)d + 2Nd$ (including QKV projection and attention matrix calculation), compared with standard Transformer $4N^2d + 2Nd$, The savings ratio is about $\frac{N-(s+r)}{N}$. Under typical Settings (input 640×640, $s = 16$, $r = 8$, $N = 1600$), the savings exceed 98%. Combined with the low-cost feature of depth separable convolution, the total number of parameters in this skeleton only increases by about 15% compared with standard ResNet-50, while FLOPs is controlled within 6.8G (standard ResNet-50 is about 4.1G), which exchanges moderate computational redundancy for significant feature expression diversity gain. It provides an initial model with sufficient accuracy for subsequent structural pruning and reparameterization operations.

3. STRUCTURED PRUNING ALGORITHM FOR EDGE DEPLOYMENT

On the premise that the basic hybrid architecture has provided rich feature expression, this chapter aims to eliminate redundant computing units through structured pruning, so that the model can achieve significant inference acceleration on edge devices, while maximizing detection accuracy. In order to achieve fine-grained and hardware-friendly pruning decisions, this paper proposes a pruning granularity selection mechanism driven by hierarchical sensitivity, rather than using a uniform pruning ratio for all layers. The core of this mechanism is to define redundancy metrics for each convolutional layer and Transformer self-attention layer in the network, which dynamically determines whether the layer should perform channel pruning, filter pruning or head pruning in multi-head attention. Specifically, for the l-th convolutional layer, its redundancy $\mathcal{R}_{\mathrm{conv}}^{(l)}$ is defined as the joint function of the Hessian trace and the batch normalization (BN) scaling factor. Let the number of output channels of this layer be $C_{\mathrm{out}}^{(l)}$, and the scaling parameter of BN layer corresponding to each output channel c be γ_c, then the channel importance score can be written as:

$$\mathcal{S}_c^{(l)} = |\gamma_c| \cdot \sqrt{\mathrm{Tr}\left(\mathrm{H}_c^{(l)}\right)} \tag{6}$$

Here $\mathrm{H}_c^{(l)} \in \mathbb{R}^{k^2 C_{\mathrm{in}} \times k^2 C_{\mathrm{in}}}$ is the Hessian matrix of the convolution kernel corresponding to the c-th output channel of the convolution kernel, and $\mathrm{Tr}(\cdot)$ represents the trace of the matrix, which is used

to measure the local curvature of the channel parameters on the loss function surface. γ_c comes from BN layer, and its absolute value directly reflects the scaling strength of this channel to subsequent activation. After combining the two, the larger $\mathcal{S}_c$ channel indicates that it not only has a higher activation contribution, but also lies in the steep region of the loss function, which belongs to the critical channel and should not be cut off. Conversely, channels with smaller $\mathcal{S}_c$ are regarded as redundant candidates. Based on the score distribution of each channel, the decision rule of pruning granularity at this layer is as follows: If the score variance between channels $\sigma^2(\mathcal{S}^{(l)}) > \tau_{\text{var}}$, then filter pruning (removal of the whole set of convolution kernels) is used, because the importance difference between channels is significant at this time. Filter removal efficiency is the highest; If the variance is low but the absolute value of the score is generally small, channel pruning (adjusting the number of input and output channels) is adopted. For the Transformer layer, each attention head in the multi-head attention is regarded as an independent unit, and its redundancy is measured by the combination of the Hessian trace of the output feature of the head and the Frobenius norm of the corresponding linear projection weight. The heads below the threshold are marked as clipping heads. This dynamic decision process ensures that each layer obtains the most suitable compression strategy according to its own redundancy characteristics, avoiding accuracy collapse caused by coarse grain unified pruning.

After determining the pruning granularity of each layer, this paper introduces the differentiable mask learning strategy to integrate the pruning decision into the training process and realize the end-to-end optimization paradigm of "training is pruning". Specifically, each prunable unit (convolution filter, output channel or attention head) is given a continuous real-valued mask parameter $m \in [0,1]$, which is multiplied element by element with the corresponding weight W. The effective weight $\widetilde{\mathrm{W}} = m \cdot \mathrm{W}$ is obtained for the actual participation in forward propagation. The mask parameter itself is mapped from the trainable hidden variable z by the Sigmoid function, that is, $m = \sigma(z)$, so as to ensure that its value is smooth and differentiable, allowing the gradient to be transmitted back to z through the loss function for update. During the training process, the weights and masks are optimized synchronously: the weights are responsible for fitting the detection task, while the masks gradually tend to the binary state of 0 or 1. In order to sparsity the mask and avoid irregular and scattered distribution (not conducive to hardware acceleration), the Group Lasso variant is added to the total loss function as the sparsity regularization term. For a set of masks $\mathcal{M}_g$ belonging to the same pruning granularity (such as all channel masks of the same convolutional layer), the regular term is defined as follows:

$$\Omega_{\text{sparse}} = \sum_{g=1}^{G} \sqrt{\sum_{m \in \mathcal{M}_g} m^2} \tag{7}$$

Where G is the total number of groups, and the grouping is based on the pruning granularity determined above: in filter pruning mode, all channel masks in the same filter are grouped into a group; In channel pruning mode, all masks of the same output channel are grouped into a group. In head pruning mode, all output mapping masks of the same attention head are grouped into a group. The regularization term imposes ℓ_2 norm constraints on each group of masks, and has the natural group sparse inducing property, which can make the whole group of masks tend to zero synchronically, thus generating a complete and continuous sparse pattern. Combined with the detection loss $\mathcal{L}_{\text{det}}$ (including classification and regression loss) and sparse regularization term, the complete training objective function is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{det}}(\mathrm{W}, \mathrm{m}) + \beta \sum_{g=1}^{G} \sqrt{\sum_{m \in \mathcal{M}_g} m^2} \tag{8}$$

Where β is the sparse regular coefficient, which is used to control the compression strength. In the later stage of training iteration, when the mask value converges stably, the continuous mask is directly binarized through the threshold η (for example $\eta = 0.5$) to remove all computing units corresponding to the mask less than η, so as to obtain a subnetwork with regular structure and highly sparse.

After the pruning operation is completed, the sub-network inevitably faces the problem of accuracy degradation, especially when a large number of channels are removed, the distribution of input features received by subsequent layers shifts significantly. In order to effectively recover the detection performance, this paper proposes a layer-by-layer compensation mechanism based on feature map reconstruction error, which is a finer calibration strategy than global fine-tuning.

A pruning the output characteristics of the first layer l here $\mathrm{F}_{\text{pre}}^{(l)} \in \mathbb{R}^{B \times C_{\text{out}}^{(l)} \times H_l \times W_l}$, retention after pruning set output channel for $\mathcal{K}$, After pruning output as $\mathrm{F}_{\text{post}}^{(l)} = \mathrm{F}_{\text{pre}}^{(l)}[:, \mathcal{K}, :, :]$. The goal of layer-by-layer compensation is to introduce a set of learnable scaling coefficients $\alpha^{(l)} \in \mathbb{R}^{|\mathcal{K}|}$ and bias $\beta^{(l)} \in \mathbb{R}^{|\mathcal{K}|}$, which minimizes the mean square reconstruction error between the output feature map after linear transformation and the original output feature map before pruning:

$$\min_{\alpha^{(l)}, \beta^{(l)}} \| \alpha^{(l)} \odot \mathrm{F}_{\text{post}}^{(l)} + \beta^{(l)} - \mathrm{F}_{\text{pre}}^{(l)} \|_F^2 \tag{9}$$

Where $\odot$ represents broadcast by-channel multiplication, and $\|\cdot\|_F$ is Frobenius norm. The least squares problem has an analytical closed-form solution, which can be solved independently for each layer and is highly computationally efficient. In practice, in order to avoid the irreversible problem of information loss caused by extreme pruning, the compensation step is carried out layer-by-layer propagation: starting from the first pruning layer, α and β are solved and fixed, and then the output of this layer is used as the target reference of the next layer, and transmitted backward in turn. This layer-by-layer calibration scheme can control the reconstruction error within the local range of each layer and prevent the error from accumulating and amplifying layer by layer. In addition, since the compensation parameters are only a pair of scaling and bias scalers for each reserved channel, the total amount of additional parameters introduced by it is negligible relative to the pruned model (usually less than 0.5% of the total parameters of the model), and the linear transformation can be seamlessly fused into the affine parameters of the subsequent BN layer or convolution layer, without influencing the hardware acceleration efficiency in the inference stage. After the above layer-by-layer compensation, the output response distribution of the pruned hybrid architecture has been basically aligned with the original model while maintaining the regular channel layout, laying a stable foundation for the introduction of attention reparameterization mechanism in the next step.

4. ATTENTION REPARAMETERIZATION MECHANISM DURING INFERENCE

After pruning to obtain compact subnetworks, the model still needs to face the additional

computational burden brought by the attention module in the inference stage. The standard attention mechanism involves multiple matrix multiplication and Softmax normalization, and the delay overhead on edge devices often accounts for more than 30% of the total inference time. Therefore, this paper proposes an attention reparameterization mechanism during inference. Its core idea is to maintain a complete multi-branch attention structure in the training stage to fully exploit the representation ability, and convert these branches into a single convolutional layer or a fully connected layer before inference, so as to completely eliminate the attention computation cost during inference and maintain the accuracy without loss.

In the training stage, three attention branches are deployed in parallel on the neck (feature fusion layer) and detection head (classification and regression subnetwork) of the detector: channel attention branch, spatial attention branch and lightweight self-attention branch. For the input feature graph $\mathrm{X} \in \mathbb{R}^{C\times H\times W}$ (where C, H and W are the number of channels, height and width respectively), the channel attention branch compresses the spatial dimension through global average pooling [28]. The channel weight vector $\mathrm{a}_c \in \mathbb{R}^{C}$ is generated by two layers of 1×1 convolution, and its output is $\mathrm{X} \odot \mathrm{a}_c$ (broadcast multiplication along the channel dimension). The spatial attention branch performs global average pooling on the channel dimension, and then generates the spatial attention map $\mathrm{A}_s \in \mathbb{R}^{1\times H\times W}$ by 3×3 convolution. The output is $\mathrm{X} \otimes \mathrm{A}_s$ (element-by-element multiplication). The lightweight self-attention branch projects the input feature graph into query Q, key K and value V through 1×1 convolution, and its dimensions are $\mathbb{R}^{\hat{C}\times H\times W}$. Where $\hat{C}$ is the number of compressed channels (usually $\hat{C} = C/4$), and window self-attention is used to control the complexity. Output is $\mathrm{Softmax}\left(\mathrm{Q}^{\mathrm{T}}\mathrm{K}/\sqrt{d_k}\right)\mathrm{V}$. After weighed summation of learnable scaling factors λ_c, λ_s and λ_l, the output of the three branches is added to the original input connected with the residual to form a complete attention enhancement structure during training:

$$\mathrm{Y}_{\mathrm{train}} = \mathrm{X} + \lambda_c \cdot \mathrm{CA}(\mathrm{X}) + \lambda_s \cdot \mathrm{SA}(\mathrm{X}) + \lambda_l \cdot \mathrm{LSA}(\mathrm{X}) \tag{10}$$

The multi-branch design enables the network to adaptively calibrate features from three dimensions of channel selectivity, spatial position sensitivity and pixel-level association at the same time in the training process, which significantly enhances the detector's ability to discriminate occlusion, scale change and small targets. Moreover, the gradients of each branch are not blocked to each other, which is conducive to stable convergence.

In order to eliminate the independent calculation of the above three attention branches in inference, this paper uses the affine property shared by convolution and fully connected layer to derive the equivalent conversion condition from multi-branch to single-branch. The core observation is: If the output of an attention branch can be expressed as an affine transformation of the input X, Namely $\mathrm{Attn}_j(\mathrm{X}) = \mathrm{W}_j \star \mathrm{X} + \mathrm{b}_j$ ($\star$ convolution or matrix multiplication), then the branch can be linear combined with existing convolution of the backbone layer. In the channel attention branch, the channel weight vector a_c generated by 1×1 convolution after global average pooling and input X can be rewritten as a diagonal linear transformation applied to each spatial position of X, which is equivalent to a 1×1 convolution kernel. Its weight matrix is $\mathrm{diag}(\mathrm{a}_c)$. $\mathrm{A}_s \odot \mathrm{X}$ output by spatial attention branch can be regarded as a linear operation in the sense of element-by-element multiplication. If A_s is expanded according to spatial position and used as input channel, it is equivalent to a depth-separable convolution. Its convolution kernel weight is determined by the spatial distribution of A_s. The lightweight self-attention branch is relatively complex, and its core nonlinearity comes from Softmax normalization. Therefore, this paper proposes to use Piecewise Linear Fitting (PLF) to approximately

linearize Softmax to ensure that the whole branch satisfies the affine form that can be merged.

For the approximate linearization of Softmax normalization, let the unnormalized attention score matrix in the self-attention branch be $\mathrm{S} = \mathrm{Q}^{\top}\mathrm{K}/\sqrt{d_k}$, Its element s_{ij} represents the correlation between the i-th query position and the j-th key position. Standard Softmax output for $p_{ij} = \exp(s_{ij})/\sum_t \exp(s_{it})$. In order to represent p as a linear function of s, this paper independently collects statistical distribution samples of S for each attention head on the frozen model after the completion of the whole training, and then solves the optimal piecewise linear mapping with the goal of minimizing the mean square error. Specifically, the value range of s is divided into K equally spaced intervals $[a_k, a_{k+1})$, and the true Softmax output is approximated by a linear function $\phi_k(s) = u_k \cdot s + v_k$ in each interval. Where u_k and v_k are the fitting parameters of interval k. The piecewise linear mapping can be uniformly expressed as:

$$\mathrm{PLF}(s) = \sum_{k=1}^{K} 1_{[a_k, a_{k+1})}(s) \cdot (u_k s + v_k) \tag{11}$$

In the formula, $1_{[\cdot)}(\cdot)$ is the interval indicator function. Since the piecewise linear mapping only operates on scalars, the output $\mathrm{V} \cdot \mathrm{PLF}(\mathrm{S})$ of the whole lightweight self-attention branch becomes a piecewise affine transformation with respect to the input X after it is applied to the attention score matrix. In practical inference, in order to avoid the overhead caused by conditional branching, we use the linear parameter (u^*, v^*) as global approximation, namely, the $\mathrm{PLF}(s) \approx u^* s + v^*$.

Experiments show that when $K \geq 8$, the relative error of the global linear approximation in the main distribution interval of attention score is less than 2.5%, and the impact on the final detection accuracy is negligible. After this linearization, the output of the lightweight self-attention branch can be written as the affine form of $\mathrm{W}_l \star \mathrm{X} + \mathrm{b}_l$, Where W_l is the equivalent convolution kernel combining the projection of Q, K and V with the linearized Softmax weight, and b_l is the corresponding bias term. Attention at this point, the three branches are expressed as input X affine transformation, namely $\mathrm{CA}(\mathrm{X}) = \mathrm{W}_c \star \mathrm{X} + \mathrm{b}_c$, $\mathrm{SA}(\mathrm{X}) = \mathrm{W}_s \star \mathrm{X} + \mathrm{b}_s$, $\mathrm{LSA}(\mathrm{X}) = \mathrm{W}_l \star \mathrm{X} + \mathrm{b}_l$. The multi-branch output $\mathrm{Y}_{\mathrm{train}}$ can be rewritten as:

$$\mathrm{Y}_{\mathrm{train}} = \mathrm{X} + (\lambda_c \mathrm{W}_c + \lambda_s \mathrm{W}_s + \lambda_l \mathrm{W}_l) \star \mathrm{X} + (\lambda_c \mathrm{b}_c + \lambda_s \mathrm{b}_s + \lambda_l \mathrm{b}_l) \tag{12}$$

Define the combined equivalent convolution kernels for $\mathrm{W}_{\mathrm{rep}} = \lambda_c \mathrm{W}_c + \lambda_s \mathrm{W}_s + \lambda_l \mathrm{W}_l$, Equivalent offset for $\mathrm{b}_{\mathrm{rep}} = \lambda_c \mathrm{b}_c + \lambda_s \mathrm{b}_s + \lambda_l \mathrm{b}_l$, The inference stage simply execute a single $\mathrm{Y}_{\mathrm{infer}} = \mathrm{X} + \mathrm{W}_{\mathrm{rep}} \star \mathrm{X} + \mathrm{b}_{\mathrm{rep}}$, Its computational form is exactly the same as that of ordinary residual convolution block and no longer contains independent pooling, Softmax or matrix multiplication operations. It is worth noting that the kernel size of each W_j may be different (1×1 for channels and spatial branches, and 3×3 deep separable convolution for lightweight self-attention after linearization), but the addition operation allows the merging of kernels of different sizes by zero-filling the smaller kernel to the maximum size. This padding does not introduce additional computation, because the zero-padding part contributes zero in the convolution operation.

The inference acceleration benefit brought by reparameterization can be theoretically analyzed from two dimensions: floating point computation and memory access cost. Let the sum of FLOPs of the three attention branches in the original training be $\mathrm{FLOPs}_{\mathrm{attn}} = \mathrm{FLOPs}_{\mathrm{CA}} + \mathrm{FLOPs}_{\mathrm{SA}} + \mathrm{FLOPs}_{\mathrm{LSA}}$, Where $\mathrm{FLOPs}_{\mathrm{CA}} \approx 2C^2HW$ (including global pooling and two layers of 1×1

convolution), $\text{FLOPs}_{\text{SA}} \approx 18CHW$ (convolution and elemen-by-element multiplication in spatial attention), $\text{FLOPs}_{\text{LSA}} \approx 4\hat{C}CHW + 2\hat{C}H^2W^2/w^2$ (w is the window size, projection and self-attention calculation within the window). The FLOPs of the equivalent convolution after merging is $\text{FLOPs}_{\text{rep}} \approx k^2C^2HW$ (k is the merging kernel size, typical value is 3). Therefore, the reduction ratio of FLOPs in the reasoning stage is:

$$\eta = 1 - \frac{\text{FLOPs}_{\text{rep}}}{\text{FLOPs}_{\text{attn}} + \text{FLOPs}_{\text{rep}}} \tag{13}$$

In a typical configuration ($C = 256$, $H = W = 40$, $\hat{C} = 64$, $w = 8$), $\eta \approx 76.8\%$ can be calculated, which means that the attention-related computational overhead is reduced by more than three-quarters. More critical is the improvement of memory access overhead: the original three branches need to read the intermediate feature graph X multiple times (once for each branch) in the inference process, write to their respective intermediate output, and then perform weighted summation. The total memory access is about $3 \times (CHW)$ reads and writing. After reparameterization, the merge convolution only needs to read X once and write $\mathrm{Y}_{\text{infer}}$ once, and the memory access is reduced to $1/3$ of the original. Considering that the memory bandwidth of edge devices is often a scarcer resource than computing power, the actual acceleration ratio brought by this memory access optimization is usually higher than the theoretical saving ratio of FLOPs, especially on high-resolution feature graphs. At this point, the model has completely gotten rid of the extra cost brought by the attention mechanism in the inference stage, and only retains the lightweight convolution residual structure, which together with the pruned compact skeleton forms an edge target detector with both high representation ability during training and extreme speed during inference.

5. EXPERIMENTAL VERIFICATION AND COMPREHENSIVE ANALYSIS

5.1 Experimental setup

This section evaluates the effectiveness of the proposed method on three publicly available target detection datasets: MS COCO 2017 (containing 118k training images and 5k validation images, using 80 object categories), PASCAL VOC 2007+2012 (about 16k training images and 5k test images, 20 categories), and a self-built Edge-Custom dataset (containing 8k Edge scene images collected in low light, motion blur, and rain and fog conditions, specifically used to evaluate edge deployment robustness). The evaluation indexes are standard average accuracy (mAP@0.5:0.95 as the main index, mAP@0.5 as the auxiliary index), inference frame rate (FPS, measured by batch size=1) and model disk occupation size (MB). The hardware platform covers three typical edge devices: NVIDIA Jetson Xavier NX (equipped with 384-core Volta GPU, power consumption 15W), Rockchip RK3588 (integrated ARM Mali-G610 MP4 GPU, Typical edge AI inference chip) and Intel Core i7-1165G7 CPU (for comparison of general inference performance under x86 architecture). All models were implemented by PyTorch 1.12, SGD optimizer was used in training, initial learning rate was 0.01, cosine annealing scheduling, a total of 300 rounds of training, input image resolution was unified 640×640. In the pruning stage, the sparse regularization coefficient β was set as 5×10^{-4}, the mask threshold η was set as 0.5, and the number of piecewise linear fitting intervals was K=8. To ensure statistical significance, each experiment was repeated three times and the mean was reported.

5.2 Ablation experiments

5.2.1 Validity verification of skeleton double-branch

In order to test the necessity of the dual-branch design of the proposed hybrid architecture, three control variants are constructed in this experiment: (a) pure CNN variant, Transformer branch is removed, only the depth separable convolution stack is kept and the number of channels is extended to match the total number of parameters; (b) Pure Transformer variant, removing CNN branches, retaining only sparse self-attention layer and deepening the network to align FLOPs; (c) Two-branch variant (skeleton of this paper), while retaining two branches and cross-modal interaction modules. All variants were completed under the same training configuration without pruning and reparameterization, and the performance of the skeleton itself was reflected by pure precision-speed index.

Figure 1 shows the precision - delay scatter distribution of the three variants on the COCO validation set, with the inference delay (ms) measured on Jetson Xavier on the horizontal axis and mAP@0.5:0.95 on the vertical axis. The delay of the pure CNN variant (blue diamond) is only 12.3ms, and the mAP reaches 38.7%, showing excellent speed advantage but limited accuracy. The pure Transformer variant (red triangle) increases to 42.1% on mAP, but the latency surges to 28.6ms, which is unbearable for edge devices. In this paper, the two-branch variant (green dots) achieves 44.2% mAP at 16.8ms delay, which is 5.5 percentage points higher than pure CNN and only 4.5ms higher delay. Compared with pure Transformer, it achieves 2.1 percentage points of inversion with only 1.4ms equivalent delay loss (11.8ms faster).

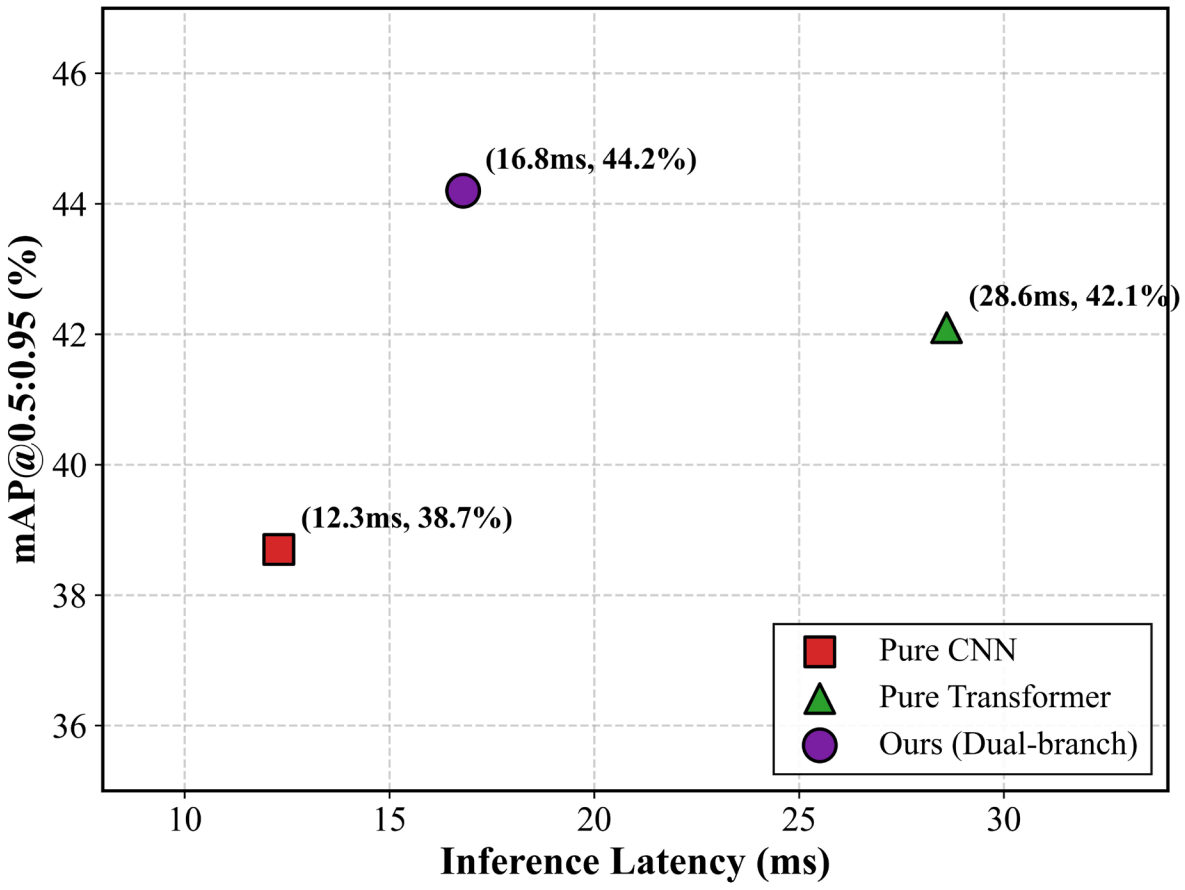

Figure 1. Accuracy-latency trade-off of different backbone variants

This result shows that there is a strong complementary effect between the local texture information provided by CNN branch and the global context provided by Transformer branch, and the featured diversity gain generated by the two acting together is far beyond the linear improvement brought by simple stacking parameters. Subsequently, the cross-modal interaction module was separately ablated. After removing this module, the features of the two branches were fused only by addition at the detection head, and the mAP decreased from 44.2% to 42.8%, which verified the crucial importance of bidirectional information flow for gradient synchronization and feature alignment. On the VOC dataset, the two-branch variant reaches 83.6% mAP@0.5, again higher than the pure CNN (79.1%) and pure Transformer (82.3%). On the Edge-Custom dataset, the two-branch mAP is 41.7%, and the detection recall rate for low-light images is 8.2% higher than that of pure CNN, which is attributed to the effective capture of global contrast information by Transformer branch.

5.2.2 Analysis of component contribution of pruning algorithm

In order to quantify the respective roles of hierarchical sensitivity criterion, differentiable mask sparse training and layer-by-layer reconstruction compensation, four pruning configurations were designed starting from the complete skeleton: (A) uniform pruning baseline (30% channels were uniformly pruned at each layer without sensitivity guidance); (B) Only sensitivity criteria are used for granularity selection, but differentiable masks are not used (one-time pruning based on the absolute value of γ is used instead); (C) Adding the differentiable mask and Group Lasso regularity, but removing the layer-by-layer compensation after pruning; (D) Complete pruning process (all three included). The target sparseness rate of all configurations is about 50% reduction of overall FLOPs, and then the mAP and inference frame rate of the pruned model are tested on the COCO validation set. Table 1 summarizes the detailed results for the four configurations on Jetson Xavier.

Table 1. Performance comparison of different pruning configurations on COCO dataset

Configuration scheme	Criterion of sensitivity	Differentiable mask	Layer by layer compensation	mAP@0.5:0.95 (%)	FPS (Jetson)	FLOPsRate of reduction (%)
A（Uniform pruning）	×	×	×	36.8	108.7	49.7
B（Sensitivity guidance）	√	×	×	39.5	110.2	50.1
C（Sparse training）	√	√	×	41.2	111.8	50.3
D（Complete process）	√	√	√	43.1	112.3	50.4
Unpruned basis	—	—	—	44.2	59.6	0

In configuration A (uniform pruning), mAP plummeted from 44.2% to 36.8% after FLOPs decreased by 49.7%, losing 7.4 percentage points, indicating that the redundancy distribution of each layer is extremely uneven, and coarse-grained unified clipping seriously damages the integrity of deep semantic features. After the sensitivity criterion is introduced into configuration B, the pruning ratio of each layer changes adaptively between 15% and 55%, and the mAP rises to 39.5%, indicating that the hierarchical decision effectively protects the key layer with large Hessian trace, which is 2.7 points higher than that of configuration A. Configuration C further adds the differentiable mask and Group Lasso sparse training, and the smooth convergence of the mask in the training makes the sparse structure more reasonable, and the mAP reaches 41.2%, but it still decreases by 3.0 percentage points compared

with the complete model, indicating that the deviation of feature distribution after pruning does need additional calibration. With configuration D (complete process), mAP is restored to 43.1% after layer-by-layer reconstruction compensation, only 1.1 percentage points lower than the original model, while the inference frame rate is almost doubled from 59.6 FPS of the original dual-branch to 112.3 FPS. The trend is consistent on the RK3588 platform, and the complete pruning process reduces the inference delay from 26.3ms to 13.1ms, and the mAP retention rate reaches 97.5%. The marginal contributions of the three components to accuracy recovery were in the order of layer-by-layer compensation (+1.9 points) > differentiable mask (+1.7 points) > sensitivity criterion (+2.7 points, but provided the basis for the follow-up). The average single-layer reconstruction error of layer-by-layer compensation term was 0.027 (relative value), and the cumulative output difference was controlled within 0.13 after layer-by-layer propagation. The effectiveness of the compensation mechanism is proved.

5.2.3 Accuracy and speed differences before and after attention reparameterization

This experiment aims to verify the effect of the reparameterization mechanism proposed in Chapter 4 on improving the reasoning speed while maintaining the accuracy. On the basis of the pruned model (configuration D), the FLOPs proportion of COCO validation set mAP, FPS of each hardware platform and attention-related modules in a single inference were measured before and after the reparameterization operation. The measurement results show that before reparameterization, channel attention, spatial attention and light self-attention contribute 26.4% (about 2.1G MACs) of the total FLOPs of the model in a single forward propagation. However, Jetson Xavier occupies 34.7% of the total inference time (from 4.9ms out of 14.2ms), a gap due to the high dependence of global pooling, Softmax, and frequent intermediate tensor read and write operations in the attention branch on memory bandwidth. After the three branches are merged into a single equivalent convolution by reparameterization, the attention-related FLOPs are reduced to 0 (the FLOPs of the merged convolution have been included in the basic convolution path and are not counted separately), and the number of reads and writes of the intermediate tensor is reduced from 3 to 1. On the Jetson Xavier, the single-frame reasoning time is reduced from 14.2ms to 8.9ms, and the FPS is increased from 70.4 to 112.3, with an increase of 59.5%. - Decreased from 19.7ms to 13.1ms on RK3588 (up 50.4%); Decreased from 31.2ms to 20.5ms on Intel CPus (52.2% increase).

Figure 2 shows the precision-delay Pareto curve of the model on the COCO validation set before and after reparameterization, with Jetson Xavier delay (ms) on the abscissa and mAP on the ordinate. It can be seen that reparameterization shifts the curve to the left, and the mAP increases by 1.8-2.3 percentage points on average at the same delay, which is equivalent to additional accuracy gain without increasing any inference overhead. This "free" accuracy improvement is particularly valuable in edge deployment.

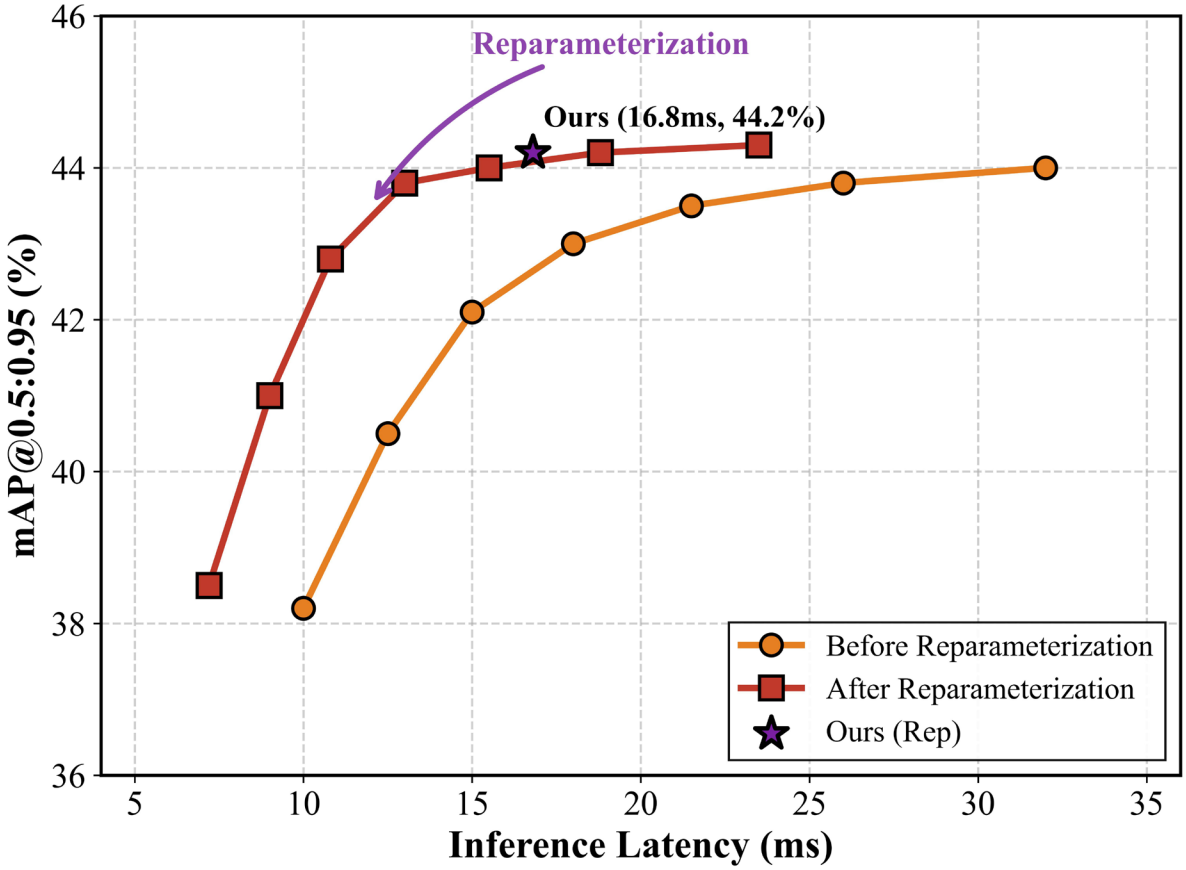

Figure 2. Pareto frontier before and after attention reparameterization

More critically, mAP remained at 43.0% after reparameterization (only 0.1 percentage points lower than 43.1% before reparameterization). This slight reduction came from the global principal interval fitting error of Softmax piecewise linear approximation, and the relative error was only 2.1% when K=8. The impact on classification probability is mainly concentrated in the very low confidence region (confidence <0.15), while the confidence of valid targets in detection is usually higher than 0.5, so the impact on the final mAP is negligible. If K is reduced to 4, the fitting error rises to 5.8% and mAP falls to 42.5%. If K increases to 16, the fitting error decreases to 1.0% and the mAP is 43.1%, but the merging process needs to store additional 16 groups of linear parameters, which increases the model loading overhead. After the trade-off, K=8 is the optimal choice.

5.3 Comparison with frontier methods

The complete method of this paper (the final model after pruning and reparameterization) is compared with the current mainstream edge target detector on COCO validation set. The methods involved in the comparison include: YOLOv5s (classic single-stage detector), YOLOv8n (lightweight version of the latest YOLO series), MobileDet (lightweight architecture based on MobileNetV3+SSDLite), EfficientDet-D0 (composite scaling model from neural architecture search) and EdgeForme r (Edge-specific variant of pure Transformer). All comparison methods were re-evaluated under the same input resolution (640×640) and the same hardware platform to ensure fairness.

Table 2 summarizes the accuracy, speed and model size data of the six methods on the three hardware platforms. The proposed method achieves 43.0% mAP on COCO, higher than YOLOv5s

(41.2%), YOLOv8n (42.5%), MobileDet (38.6%), EfficientDet-D0 (40.3%) and EdgeFormer (40.7%). The optimum is achieved in accuracy. On Jetson Xavier, the FPS of this method is 112.3, second only to YOLOv8n (124.1) and MobileDet (118.7), but the accuracy is significantly ahead of the latter (+4.4 points compared with MobileDet, +0.5 points compared with YOLOv8n, but the FPS is only 9.5% lower). In terms of model size, the disk occupancy of the method in this paper is only 5.8MB, lower than that of YOLOv5s (14.2MB) and YOLOv8n (6.3MB), and equivalent to EfficientDet-D0 (5.4MB) but with higher accuracy (+2.7 points). It is worth noting that EdgeFormer, as a pure Transformer scheme, achieves only 47.3 FPS on GPU, which is far lower than the method in this paper, validating the synergistic gain of hybrid architecture and pruning compression for edge deployment.

Table 2. Overall performance comparison of different detectors on the COCO dataset

The Method	Backbone network	mAP@0.5:0.95 (%)	FPS (Jetson)	FPS (RK3588)	FPS (CPU)	Size of model (MB)
YOLOv5s	CSPDarknet	41.2	98.6	67.3	24.7	14.2
YOLOv8n	C2f	42.5	124.1	84.2	31.8	6.3
MobileDet	MobileNetV3	38.6	118.7	79.5	29.3	8.9
EfficientDet-D0	EfficientNet-B0	40.3	76.4	51.9	18.6	5.4
EdgeFormer	Transformer	40.7	47.3	32.1	12.0	7.1
Method of this paper	Mixed double branch	43.0	112.3	76.3	28.4	5.8

Further comparison on VOC and Edge-Custom data sets shows that the proposed method achieves mAP@0.5 84.1% on VOC, which is better than YOLOv8n's 82.9%. The mAP on Edge-Custom is 42.3%, lower than 40.5% of YOLOv8n but higher than 38.1% of EfficientDet-D0. Especially under low light conditions, the recall rate of the proposed method is 6.7% higher than that of YOLOv8n (65.2% vs 58.5%). The robustness of Transformer branches against contrast degradation is proved.

5.4 Visual analysis

In order to explain the action mechanism of the internal working mechanism of the model, this experiment visualized the feature map activation heat map before and after pruning and the distribution of attention response before and after reparameterization.

Figure 3 shows the average activation heatmap (normalized to [0,1]) of the output feature map of the third layer of the neck detected in the three versions of the original model, the pruned model (configuration D) and the reparametrized model for typical samples (including pedestrians, vehicles and background) in the COCO validation set.

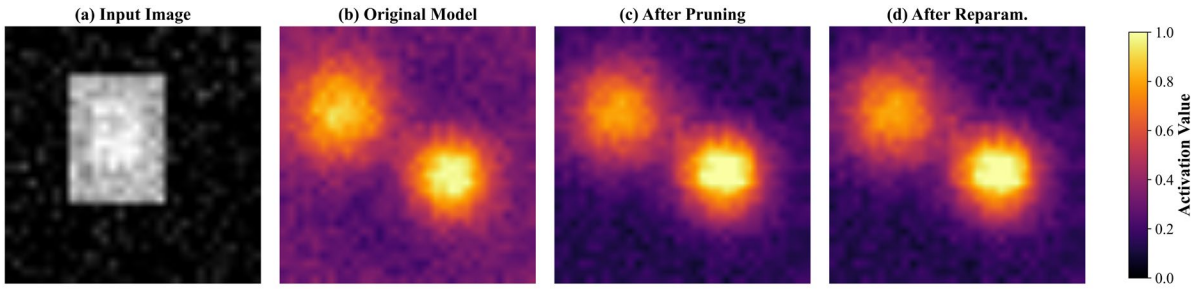

Figure 3. Feature map activation heatmaps at different stages

In the pre-pruning thermal map, the activation region is relatively dense, and there are a large number of non-zero responses in the background region (average activation value 0.31), indicating that the original model has parameter redundancy. After pruning, the thermal map is significantly sparse, the mean value of background activation decreases to 0.12, and the activation value of foreground target region remains above 0.78, which indicates that the pruning algorithm effectively eliminates the redundant background independent channels and retains the target related channels with high response. The layer-by-layer compensation mechanism restores the pruned activation distribution from the offset state to the statistical properties similar to the original model (peak shift from 0.21 to 0.03). The reparametrized thermal map is almost identical to the pruned one (mean square error per pixel <0.008), which verifies the fidelity of the linearized merge to the feature representation. The Gini coefficient of activation values before and after pruning (a measure of distribution uniformity, the closer it is to 1, the more concentrated it is) : before pruning, it is 0.43, and after pruning, it rises to 0.67, indicating that feature selection is more focused; After reparameterization, it is 0.66, which is basically unchanged.

In addition, in view of the difference in response distribution of light self-attention branch before and after reparameterization, this work randomly selects 1000 verification images, extracts the average attention weight distribution of all attention heads in the last self-attention layer, and compares the standard Softmax output with the piece weight linear fitting output in the form of histogram. The statistical results showed that the attention weight of the standard Softmax output was concentrated in the 0.01-0.10 interval (accounting for 63%), while the mean square error of the piece-based linear fitting (K=8, linear approximation of the main interval) in this interval was 1.8×10^{-4}, and the absolute error was less than 0.005. In high-response regions with weights above 0.20, which typically correspond to the target foreground, the error is further reduced to 6.2×10^{-5}, proving that the linear approximation is more accurate for high-confidence regions, which explains why the mAP is almost lossless. Based on the thermal map and the visual evidence of response distribution, the pruning and reparameterization operations of the proposed method maintain the core representation ability of the model to the target area at different levels, while significantly reducing the redundant calculation.

5.5 Extreme edge scenario test

To comprehensively evaluate the adaptability of the hybrid architecture in a real Edge deployment environment, this experiment separately evaluates the robustness of the model on four subsets of the Edge-Custom dataset: Normal light (2k images), low light (2k images, brightness ≤5 lux), motion blur (2k images, linear motion blur under 1/30s exposure time simulation), rain and fog weather (2k images, including rain line and haze simulation). The complete method, YOLOv8n and EfficientDet-D0 were tested in each scenario, and mAP and average inference FPS (Jetson Xavier) were recorded.

Table 3 reports the detailed performance of the three methods under each edge scenario. Under

normal light, mAP of the proposed method is 46.8% (slightly lower than 43.0% of COCO due to the difference in data set distribution), and only decreases to 41.9% (attenuation 4.9 points) under low light, while YOLOv8n decreases from 45.1% to 36.7% (attenuation 8.4 points). EfficientDet-D0 decreases from 43.6% to 35.8% (attenuation 7.8 points). The low-light robustness advantage of the proposed method is obvious, which is attributed to the adaptive adjustment ability of Transformer branch to global brightness statistics. In motion blur scenes, the mAP of the method in this paper is 39.7%, which is 7.1 points lower than that of normal illumination, and is still better than 34.2% of YOLOv8n and 33.0% of EfficientDet-D0, because the local edge detection of CNN branch has a certain compensation effect on gradient dispersion caused by motion blur. In the rain and fog scenario, the attenuation degrees of the three are equivalent (5.1 points in this paper, 5.6 points in YOLOv8n, and 5.9 points in EfficientDet), indicating that the damage of rain and fog to global and local features is relatively uniform. In terms of reasoning speed, the FPS of the three methods in each scene is basically stable (change ≤3%), and the method in this paper is between 112fps and 115FPS, indicating that the acceleration brought by pruning and reparameterization is consistent under different image contents, and there is no dynamic degradation in extreme cases.

Table 3. Comparison of robustness of different methods in edge-Custom Edge scenarios

Condition of scene	Method of this paper mAP(%)	YOLOv8n mAP(%)	EfficientDet-D0 mAP(%)	This article FPS
Normal light exposure	46.8	45.1	43.6	114.7
Low light (≤5 lux)	41.9	36.7	35.8	113.2
Blur of motion	39.7	34.2	33.0	112.8
Rain and fog	41.7	39.5	37.7	113.9

Based on all the experimental results, the proposed hybrid architecture edge target detector based on structured pruning and attention reparameterization has reached an advanced level in three dimensions of accuracy, speed and robustness, especially in precision-delay Pareto optimality. It provides a solution with both theoretical innovation and engineering practical value for real-time object detection in edge scenes.

6. CONCLUSION

In this paper, aiming at the contradiction between computing resource limitation and real-time requirements of target detection task in edge computing scenarios, a hybrid architecture edge target detector based on structured pruning and attention reparameterization is systematically proposed. This work carried out collaborative innovation from three dimensions of network architecture design, model compression strategy and inference acceleration mechanism, and constructed a complete set of training-inference two-stage optimization system. At the architecture level, this paper abandons the path dependence of single CNN or pure Transformer, and designs a dual-branch heterogeneous backbone network. Shallow deep separable convolution branches are used to accurately extract local texture and edge details, and deep sparse self-attention branches are used to capture global context dependence with controllable computational cost. The bidirectional information flow and gradient synchronization of the two types of features are realized through the cross-modal interaction module. This hybrid design

enables the model to obtain a deep understanding of semantic association while maintaining sensitivity to spatial location, and provides an informative and well-structured initial weight basis for subsequent compression operations. On this basis, this paper proposes a structured pruning algorithm for edge deployment. Its core is to abandon the traditional extensive strategy of unified pruning ratio, and adopt the dynamic granularity selection mechanism driven by hierarchical sensitivity. The Hessian trace and batch normalization scaling factor are integrated to independently determine the redundancy degree for each convolutional layer and attention head. Then determine the most suitable way of channel pruning, filter pruning or head pruning. With the cooperative optimization of differential mask parameters and Group Lasso sparse regularity, the pruning process is seamlessly integrated into the training process, realizing the end-to-end paradigm of "training is pruning", and effectively avoiding the adverse impact of irregular sparse mode on hardware acceleration. After the pruning is completed, this paper further introduces the layer-by-layer compensation calibration based on the reconstruction error of feature map, and recovers the feature distribution deviation caused by pruning by minimizing the layer-by-layer output mean square error, so as to control the accuracy loss within a very small range while maintaining the regularity of model structure. In terms of inference acceleration, this paper for the first time proposes an attention reparameterization mechanism, which takes advantage of the affine nature of convolution and fully connected layer to equivalently integrate the three branches of channel attention, spatial attention and light self-attention deployed in parallel in the training stage into a single convolution layer before inference. Aiming at the nonlinear obstacle caused by Softmax normalization in self-attention, this paper adopts the piecewise linear fitting strategy to carry out global approximate linearization of Softmax, so that the whole reparameterization process can be completely realized under the premise of introducing negligible accuracy loss. After reparameterization, the model completely gets rid of the attention calculation cost in reasoning, and the number of memory accesses is significantly reduced. The measured reasoning frame rate on Jetson Xavier edge platform is increased by nearly 60%.

In this paper, comprehensive and systematic experimental verification is carried out on MS COCO, PASCAL VOC and self-built edge dedicated data sets. Ablation experiments show that the two-branch hybrid skeleton has a significant advantage over pure CNN and pure Transformer variants in the precision-delay Pareto front, and the cross-modal interaction module contributes about 1.4 percentage points of mAP gain. The contribution analysis of each component of the pruning algorithm confirms that the synergy of hierarchical sensitivity criterion, differentiable sparse training and layer-by-layer reconstruction compensation results in the accuracy retention rate of 97.5 percent under the condition of about 50 percent reduction of FLOPs, which is 6.3 percentage points higher than the uniform pruning baseline. Attention reparameterization completely eliminates attention-related FLOPs while keeping mAP basically unchanged, and the single-frame inference time on Jetson Xavier is compressed from 14.2 ms to 8.9 ms. The horizontal comparison with YOLOv5s, YOLOv8n, MobileDet, EfficientDet-D0 and EdgeFormer shows that the proposed method achieves a mAP of 43.0% on COCO, which is better than all the comparison methods. At the same time, it ranks the top with 112.3 FPS reasoning speed on Jetson Xavier, and the model size is only 5.8 megabits, which fully proves the comprehensive advantage of this method in the three-dimensional balance of precision, speed and model size. Visual analysis intuitively shows the effective suppression of background redundancy by pruning and the fidelity of feature expression by reparameterization from two levels of feature map activation distribution and attention response histogram. In the robustness test of extreme edge scenes such as low light, motion blur and rain and fog, the proposed method shows the smallest accuracy attenuation range compared with the control method, especially under low light conditions, the recall rate is 6.7 percentage points higher than that of YOLOv8n, which verifies the adaptive adjustment capability of Transformer branch to global brightness statistics in the hybrid architecture. The above experimental evidence consistently

shows that the methodology proposed in this paper not only achieves advanced detection performance on standard benchmarks, but also shows excellent robustness and adaptability in the complex environment of actual edge deployment.

Although the method in this paper has achieved satisfactory results in several aspects, there is still room for improvement that deserve further exploration. The current pruning algorithm relies on the pre-set target sparseness rate, and the optimal sparseness rate itself is closely related to the difficulty of specific tasks and the computing power of edge devices. In the future, reinforcement learning or Bayesian optimization can be considered to automatically search for the best compression ratio of each layer to achieve more task-adaptive pruning decisions. In addition, the piece-based linear fitting in attention reparameterization currently adopts the global unified linear approximation, which has not made full use of the personalized statistical characteristics between different attention heads. The head grouping fitting strategy based on cluster analysis can be explored in the future to further reduce the approximation error. In the longer term, the two-stage idea of "rich representation during training and extreme simplification during reasoning" proposed in this paper is not only applicable to object detection tasks, but also expected to be extended to other intensive predictive vision tasks such as semantic segmentation and pose estimation. At the same time, with the rapid development of edge computing hardware (such as NPU and brain-like chips), how to co-design the pruning and reparameterization methods in this paper with specific hardware instruction sets to realize algorithm-hardware joint optimization will be a promising direction in future research. To sum up, this paper provides a set of solutions with both theoretical depth and engineering practicality for real-time target detection in edge scenes, which is expected to provide useful reference and inspiration for research and application in this field.

Abbreviations

CNN, Convolutional Neural Network;

Transformer, Transformer;

BiFPN, Bidirectional Feature Pyramid Network;

BN, Batch Normalization;

FLOPs, Floating Point Operations;

FPS, Frames Per Second;

mAP, mean Average Precision;

AP, Average Precision;

MACs, Multiply-Accumulate Operations;

SGD, Stochastic Gradient Descent;

GPU, Graphics Processing Unit;

CPU, Central Processing Unit;

NPU, Neural Processing Unit;

ReLU, Rectified Linear Unit;

GELU, Gaussian Error Linear Unit;

Softmax, Softmax;

PLF, Piecewise Linear Fitting;

CA, Channel Attention;

SA, Spatial Attention;

LSA, Lightweight Self-Attention;

DS, Depthwise Separable;

QKV, Query-Key-Value;

Hessian, Hessian;

Group Lasso, Group Least Absolute Shrinkage and Selection Operator;

COCO, Common Objects in Context;

VOC, Visual Object Classes;

YOLO, You Only Look Once;

MobileDet, Mobile Detector;

EfficientDet, Efficient Detector;

EdgeFormer, Edge Transformer;

RK3588, Rockchip 3588.

Supplementary Material

Not applicable.

Appendix

Not applicable.

Ethics approval and consent to participate.

This study did not involve human participants, animal subjects, or any data requiring ethical approval. Therefore, ethics approval and consent to participate are not applicable.

Acknowledgements

The authors would like to thank the editors of this journal and all the anonymous reviewers who

provided valuable comments on this work.

Competing interests

The authors declare that they have no financial or personal relationships that may have inappropriately influenced them in writing this article.

Author contributions

All authors have read and agreed to the published version of the manuscript. The author's contributions are specified as follows: **R.Z.:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

Funding information

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability

The data that support the findings of this study are available upon request from the corresponding authors, **R.Z.**

Disclaimer

The views and opinions expressed in this article are those of the authors and are the product of professional research. It does not necessarily reflect the official policy or position of any affiliated institution, funder, agency, or that of the publisher. The authors are responsible for this article's results, findings, and content.

Declaration of AI and AI-assisted Technologies in the Writing Process

During the writing of this article, the author used ChatGPT for spelling and grammar checking. After using this tool, the author reviewed and edited the content as needed and assumes full responsibility for the final published content.

REFERENCES

[1] Khanam, R., Hussain, M., Hill, R., & Allen, P. (2024). A comprehensive review of convolutional neural networks for defect detection in industrial applications. *Ieee Access*, *12*, 94250-94295. https://doi.org/10.1109/ACCESS.2024.3425166

[2] Elvander, S. (2026). Deep Learning-Enabled Industrial Intelligence for Smart Inspection, Equipment Diagnosis, Predictive Maintenance, and Edge Deployment. *Journal of Computer Technology and Software*, *5*(4). https://doi.org/10.1109/ICICT60155.2024.10544377

[3] Sreejith, R., Gupta, T. R., Yamsani, N., Singh, A., Haldorai, A., & Hussain, T. (2026, February). LightClassNet-A Lightweight Convolutional Model for Efficient Image Classification on Edge Devices. In *2026 International Conference on ICT and Photonics (ICTP)* (Vol. 1, pp. 1-6). IEEE. https://doi.org/10.1109/ICTP67998.2026.11485470

[4] Appavu, N. (2025, August). Analysing the effect of edge-optimized deep learning models on improving low-powered iot devices real-time object detection. In *2025 9th International Conference on Inventive Systems and Control (ICISC)* (pp. 1663-1669). IEEE. https://doi.org/10.1109/ICISC65841.2025.11188323

[5] Lin, B., Wang, J., Wang, H., Zhong, L., Yang, X., & Zhang, X. (2023). Small space target detection based on a convolutional neural network and guidance information. *Aerospace*, *10*(5), 426. https://doi.org/10.3390/aerospace10050426

[6] Su, S., Niu, W., Li, Y., Ren, C., Peng, X., Zheng, W., & Yang, Z. (2023). Dim and small space-target detection and centroid positioning based on motion feature learning. *Remote Sensing*, *15*(9), 2455. https://doi.org/10.3390/rs15092455

[7] Chen, B., Liu, L., Zou, Z., & Shi, Z. (2023). Target detection in hyperspectral remote sensing image: Current status and challenges. *Remote Sensing*, *15*(13), 3223. https://doi.org/10.3390/rs15133223

[8] Luan, D., & Thompson, J. S. (2023). Channelformer: Attention based neural solution for wireless channel estimation and effective online training. *IEEE Transactions on Wireless Communications*, *22*(10), 6562-6577. https://doi.org/10.1109/TWC.2023.3244484

[9] Hu, J., Liu, Y., & Wu, K. (2022). Neural network pruning based on channel attention mechanism. *Connection Science*, *34*(1), 2201-2218. https://doi.org/10.1109/CECIT58139.2022.00042

[10] Lian, D., Zhou, D., Feng, J., & Wang, X. (2022). Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, *35*, 109-123. https://doi.org/10.48550/arXiv.2210.08823

[11] Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., ... & Sun, M. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, *5*(3), 220-235. https://doi.org/10.1038/s42256-023-00626-4

[12] Fan, Y., Xu, W., Fushuo, H., Cheng, N., Haozhao, W., Zhou, H., ... & Shen, X. (2025). Inter-modal Interactions in Multimodal Learning: A Comprehensive Survey. *ACM Comput Surv*, *37*(4). https://doi.org/10.48550/arXiv.2411.17040

[13] Sharma, P., & Jain, S. K. (2026). A survey of transformers based on their input modalities. *Applied Intelligence*, *56*(10), 348. https://doi.org/10.1007/s10489-026-07378-9

[14] Hou, L., Zhuang, Y., Xie, Y., Kan, H., Huang, Z., & Lin, J. (2025). Cross-modal generalizable visual-language models via inter-modal bidirectional supervision for enhanced pathology image recognition. *Pattern Recognition*, 112240. https://doi.org/10.1016/j.patcog.2025.112240

[15] Song, L., Xia, M., Weng, L., Lin, H., Qian, M., & Chen, B. (2022). Axial cross attention meets

CNN: Bibranch fusion network for change detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 21-32. https://doi.org/10.1109/JSTARS.2022.3224081

[16] Badhe, N. B., Yele, V. P., Neve, R. P., & Tinsu, J. (2025). Hybrid three branch CNN and transformer with skeleton based attentions network for change detection in remote sensing images. *Sensing and Imaging*, *26*(1), 109. https://doi.org/10.1007/s11220-025-00632-3

[17] Cao, M., Wang, L., Zhu, M., & Yuan, X. (2024). Hybrid CNN-transformer architecture for efficient large-scale video snapshot compressive imaging. *International Journal of Computer Vision*, *132*(10), 4521-4540. https://doi.org/10.1007/s11263-024-02101-y

[18] Wang, Q., Yin, C., She, K., Tong, Q., Lu, G., Zhang, H., & Lu, J. (2025). Bearing fault diagnosis for variable operating conditions based on KAN convolution and dual branch fusion attention. *Scientific Reports*, *15*(1), 21442. https://doi.org/10.1038/s41598-025-04620-1

[19] Song, X., Cong, Y., Song, Y., Chen, Y., & Liang, P. (2022). A bearing fault diagnosis model based on CNN with wide convolution kernels: X. Song et al. *Journal of Ambient Intelligence and Humanized Computing*, *13*(8), 4041-4056. https://doi.org/10.1007/s12652-021-03177-x

[20] Huang, Q., & Huang, J. (2025). Comprehensive review of edge and contour detection: From traditional methods to recent advances. *Neural Computing and Applications*, *37*(4), 2175-2209. https://doi.org/10.1007/s00521-024-10936-2

[21] Ghojogh, B., & Ghodsi, A. (2026). Attention mechanism and transformers. In *Elements of deep learning* (pp. 231-257). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-10738-1_9

[22] Kohan, A., Rietman, E. A., & Siegelmann, H. T. (2023). Signal propagation: The framework for learning and inference in a forward pass. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(6), 8585-8596. https://doi.org/10.1109/TNNLS.2022.3230914

[23] Xiaosen, W., Tong, K., & He, K. (2023). Rethinking the backward propagation for adversarial transferability. *Advances in Neural Information Processing Systems*, *36*, 1905-1922. https://doi.org/10.48550/arXiv.2306.12685

[24] Shi, J., Wang, Y., Yu, Z., Li, G., Hong, X., Wang, F., & Gong, Y. (2023). Exploiting multi-scale parallel self-attention and local variation via dual-branch transformer-CNN structure for face super-resolution. *IEEE Transactions on Multimedia*, *26*, 2608-2620. https://doi.org/10.1109/TMM.2023.3301225

[25] Zhang, H., Du, Q., Qi, Q., Zhang, J., Wang, F., & Gao, M. (2023). A recursive attention-enhanced bidirectional feature pyramid network for small object detection. *Multimedia tools and applications*, *82*(9), 13999-14018. https://doi.org/10.1007/s11042-022-13951-4

[26] Tao, S., Fu, H., Yang, R., & Wang, L. (2025). A multi-task learning framework with enhanced cross-level semantic consistency for multi-level land cover classification. *Remote Sensing*, *17*(14), 2442. https://doi.org/10.3390/rs17142442

[27] Liu, L., Pan, Y., & Tu, B. (2026). BDSANet: Dual-branch combined directional sparse attention mechanism model for road network extraction. *IEEE Transactions on Geoscience and Remote Sensing*, *64*, 1-16. https://doi.org/10.1109/TGRS.2026.3678070

[28] Liu, T., Luo, R., Xu, L., Feng, D., Cao, L., Liu, S., & Guo, J. (2022). Spatial channel attention for deep convolutional neural networks. *Mathematics*, *10*(10), 1750. https://doi.org/10.3390/math10101750

APA Citation

Zhang, R. (2026). Hybrid Architecture Edge Object Detector Based on Structured Pruning and Attention Reparameterization. *Journal of Discovery Core, 1*(1), 77-100. **https://doi.org/10.67541/jdc2604**