

S1 Detailed Description of the AFF-ResBlock and MS-CAM

In this section, we provide a comprehensive and detailed description of the Attention Feature Fusion (AFF) mechanism, the Multi-Scale Channel Attention Module (MS-CAM), and their integration into the residual architecture to form the AFF-ResBlock. These details expand upon the brief explanation provided in the main manuscript and illustrate why this attention mechanism is particularly effective for pulmonary nodule detection.

S1.1 Motivation and Background

Pulmonary nodules often share highly similar visual characteristics with vascular structures in two-dimensional CT slices. Both may appear as rounded, high-density regions with smooth boundaries, leading to a significant risk of misclassification. Distinguishing nodules from blood vessels therefore requires a feature extraction mechanism capable of highlighting subtle but critical channel cues.

Convolutional neural networks (CNNs) operate through feature fusion via convolution. Convolution merges information in two primary dimensions:

- **Spatial feature fusion** — enlarging the receptive field to incorporate multi-scale contextual information.
- **Channel feature fusion** — aggregating semantic information across channels.

Spatial fusion enables the extraction of local and global texture patterns, while channel fusion ensures that each semantic channel is considered during representation learning. However, not all channels contribute equally. A simple summation or concatenation of channel features may cause informative responses to be overshadowed by irrelevant ones, particularly when nodules are extremely small.

To address this imbalance, we propose the **AFF-ResBlock**, shown in Figure 3 (a) in the main text. This module adaptively learns channel importance weights through a dual-path attention process, ensuring that channels containing discriminative nodule features are emphasised while those carrying vessel-like noise are suppressed.

S1.2 Attention Feature Fusion (AFF)

The AFF block is shown in Figure 3 (b) in the main text. Given two input tensors X and Y , MS-CAM produces an attention map for the sum $X + Y$. The fusion process is expressed as:

$$Z = M(X + Y) \otimes X + (1 - M(X + Y)) \otimes Y, \quad (1)$$

where:

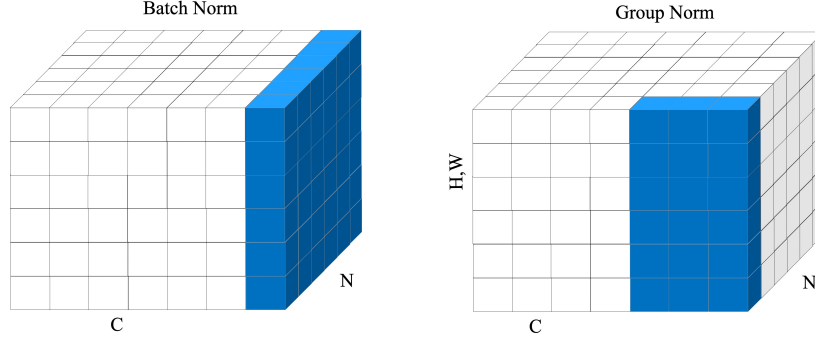


Figure S1: The left image is BN, and the left image is GN.

- \otimes denotes element-wise multiplication,
- $Z \in \mathbb{R}^{C \times D \times H \times W}$ is the fused output.

This equation demonstrates that channels receiving higher attention weights favour features from X , while the complementary channels favour features from Y . As such, AFF enables precise and adaptive feature integration.

S1.3 Multi-Scale Channel Attention Module (MS-CAM)

The MS-CAM block is shown in Figure 3 (c) in the main text. MS-CAM integrates both local and global channel contexts to produce attention weights that reflect fine-grained and holistic semantic information. These two types of context are complementary:

- **Local context** preserves detailed structures and fine variations, which is essential for small nodules that occupy very few voxels.
- **Global context** captures long-range dependencies and overall semantic trends, helping the network suppress background structures resembling nodules.

Unlike the SE module, which relies solely on global context and thus tends to suppress fine-scale features, MS-CAM incorporates local detail, making it especially effective for detecting small and medium-sized nodules.

Let an intermediate feature tensor be defined as:

$$X \in \mathbb{R}^{C \times D \times H \times W}, \quad (2)$$

where C is the number of channels, and D, H, W denote depth, height, and width.

S1.3.1 Local channel context

Local context aims to preserve subtle feature variations and is computed using a bottleneck structure:

$$L(X) = B(PWConv_2(\delta(B(PWConv_1(X))))), \quad (3)$$

where:

- $PWConv_1$ is a pointwise convolution with kernel size $1 \times 1 \times 1$, reducing channels from C to C/r .
- $PWConv_2$ restores channels from C/r back to C .
- $B(\cdot)$ denotes Batch Normalisation.
- $\delta(\cdot)$ is the ReLU activation.

Since both $L(X)$ and X share the same dimensions, the local pathway preserves and highlights fine structural details.

S1.3.2 Global channel context

Global statistics are obtained via 3D global average pooling:

$$g(X) = \frac{1}{DHW} \sum_{k=1}^D \sum_{i=1}^H \sum_{j=1}^W X_{[:,k,i,j]}, \quad (4)$$

followed by the same bottleneck structure used in the local branch:

$$G(X) = B(PWConv_2(\delta(B(PWConv_1(g(X)))))). \quad (5)$$

This global descriptor summarises overall semantic tendencies, helping the model suppress irrelevant or misleading structures.

S1.3.3 Combined attention map

Local and global contexts are then merged:

$$M(X) = \delta(L(X) + G(X)). \quad (6)$$

This unified attention reflects both fine-scale details and global semantic information.

S1.4 Comparison with the SE Module

The squeeze-and-excitation (SE) module captures global context through channel-wise pooling but ignores local detail. As SE compresses spatial information into a single global descriptor, it may eliminate vital signals contained in small nodules. In contrast:

- MS-CAM retains local structure,
- MS-CAM maintains sensitivity to small-scale variations,
- AFF combines two feature tensors with learned channel weights.

This makes AFF-ResBlock significantly more effective than SE-based approaches for pulmonary nodule detection.

This residual formulation ensures stable optimisation and efficient gradient flow while significantly enhancing feature discrimination.

S2 Weighted Cluster NMS

In object detection, Non-Maximum Suppression(NMS) is to remove redundant frames in the test process to obtain the final accurate result, which is equivalent to a kind of post-processing. However, the disadvantages of traditional NMS are also obvious, mainly as follows:(1) sequential processing mode, the calculation of IoU drags down the calculation efficiency; and (2) the elimination mechanism is too strict, and it is eliminated based on the NMS threshold. Weighted cluster NMS is a variant of cluster NMS. In order to solve the shortcomings of low computational efficiency of the traditional NMS sequential processing mode, we adopted cluster NMS, which uses the IoU matrix calculation method to improve computational efficiency while maintaining the same performance as traditional NMS. Traditional NMS will violently remove candidate boxes based on thresholds. However, the box of maximum score selected by each iteration of traditional NMS may not be accurately located. Redundant boxes may also be well located. In order to solve the problem of traditional NMS violently removing candidate boxes, We use weighted NMS, which is a weighted average of the candidate boxes coordinates. The objects of the weighted average include the largest score box and adjacent boxes with IOU greater than the NMS threshold. However, weighted NMS will slow down computing efficiency while improving accuracy. Therefore, we have combined the advantages of cluster NMS and weighted NMS and used weighted cluster NMS, which can not only maintain computing efficiency, but also significantly improve accuracy. The detailed procedures of the Weighted cluster NMS algorithm are provided in Algorithm S1 in the Supplementary Materials. The algorithm flow of Weighted cluster NMS is shown in Algorithm S1.

Algorithm S1 Weighted Cluster NMS

Require: N detected boxes sorted by classification scores in non-ascending order: $s_1 \geq s_2 \geq \dots \geq s_N$

Ensure: Updated coordinates and final detection results: 1 for reservation, 0 for suppression

```
1: Initialize  $T \leftarrow N$ ,  $t \leftarrow 1$ ,  $t^* \leftarrow T$ , and  $b_0 \leftarrow 1$ 
2: Compute IoU matrix  $X$ 
3: Transform  $X$  to upper triangular matrix:  $X \leftarrow \text{triu}(X)$ 
4: while  $t \leq T$  do
5:   Compute similarity scores  $g_j$  for each box  $b_j$ 
6:   for each box  $b_j$  do
7:     if  $g_j < \epsilon$  then
8:        $b_j \leftarrow 1$ 
9:     else
10:       $b_j \leftarrow 0$ 
11:    end if
12:  end for
13:  if  $b_t = b_{t-1}$  then
14:     $t^* \leftarrow t$ 
15:    break
16:  end if
17:   $t \leftarrow t + 1$ 
18: end while
19: Update the detection results matrix  $E_{ij}$ 
20: for each pair of boxes  $(i, j)$  do
21:   if IoU condition is satisfied then
22:      $E_{ij} \leftarrow 1$ 
23:   else
24:      $E_{ij} \leftarrow 0$ 
25:   end if
26: end for return Updated coordinates and final detection results  $D$ 
```

S3 Multi-scale Context Information and Model Fusion

In this paper, three scales of 3D examples whose size respectively is $20 \times 20 \times 26$, $30 \times 30 \times 10$, $40 \times 40 \times 26$ are used, the reason that the multi-scale examples is better than the single-scale examples is that the network can learn the information of different receptive fields. The size of the cubic examples, i.e., the surrounding range of a target position, is called the receptive field of a network. If the receptive field is too small, the network will only learn limited context information. When the nodule changes greatly, it is difficult to detect it well. If the receptive field is too large, the network will learn a lot of redundant information and noise, which will reduce the performance of the network. It is difficult for us to find an optimal single receptive field, so 3D examples of three scales are input into three networks at the same time for training and network fusion, which can solve the problem caused by a single receptive field. The $20 \times 20 \times 6$ examples are for smaller pulmonary nodules, which can contain just the small nodules and appropriate background information. The $30 \times 30 \times 10$ examples are aimed at medium-sized pulmonary nodules. This size of nodules is the most common situation in patients. It contains rich context information for small nodules, appropriate background information for medium nodules, and the main part of nodules for large nodules. The $40 \times 40 \times 26$ examples are for the large pulmonary nodules, although it will bring some redundant information to the small nodules, it can obtain rich context information for the medium-sized nodules, and it can obtain appropriate context information for the large nodules. The design of these three sizes of examples covered almost 100% of pulmonary nodules. The three scales of pulmonary nodule samples we selected are shown in Supplement Figure S2. It can be seen from the figure that nodules with different sizes on the same scale contain different contextual information, and the variability of pulmonary nodules is very large, and there are nodules of different shapes.

S4 Dataset

S4.1 Luna16 Datasets

The dataset used in this study is the open-source LUNA16 benchmark. LUNA16 (Lung Nodule Analysis 2016) was introduced in 2016 as a standardized evaluation platform for computer-aided detection (CAD) systems targeting pulmonary nodules. It is derived from the largest publicly available lung nodule dataset, LIDC-IDRI, which originally contains 1,018 low-dose chest CT scans. In constructing LUNA16, CT volumes with slice thickness greater than 2.5 mm and nodules smaller than 3 mm were excluded, resulting in a curated set of 888 high-quality CT scans.

Each CT volume in the LUNA16 dataset consists of multiple axial slices of the thoracic cavity, and the number of slices varies depending on scanner type,

acquisition protocol, and patient characteristics. The data are inherently three-dimensional, formed by stacking a series of two-dimensional slices. A nodule is considered valid in LUNA16 when at least three of the four participating radiologists annotate it with a diameter greater than 3 mm. Consequently, non-nodules, nodules smaller than 3 mm, and nodules marked by only one or two radiologists are treated as irrelevant findings and excluded from the official ground truth.

Figure S2 illustrates representative nodules of different sizes, demonstrating the contextual information and patch dimensions used for analysis. Small nodules (diameter < 7 mm), medium nodules (9–16 mm), and large nodules (> 24 mm) exhibit distinct visual characteristics, and the corresponding patch sizes are adaptively set to $20 \times 20 \times 6$, $30 \times 30 \times 10$, and $40 \times 40 \times 26$, respectively.

In this work, we utilized the lung CT volumes, lung parenchyma segmentation masks, nodule location annotations (`annotation.csv`), and candidate nodule annotations (`candidates_V2.csv`) provided by the dataset. Both the CT data and the segmentation masks are supplied in RAW and MHD formats, where the RAW files store the voxel intensity information and the corresponding MHD files define the header metadata required to correctly interpret each volume.

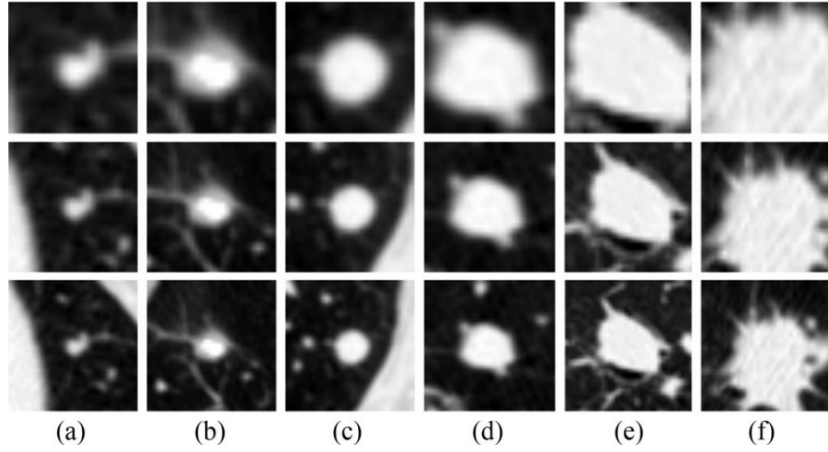


Figure S2: Explanation of contextual information about nodules. (a) and (b) are small nodules less than 7 mm in diameter. (c-e) are medium-sized nodules with a diameter between 9 and 16 mm. (f) is a large nodule with a diameter of more than 24 mm. The patch sizes are $20 \times 20 \times 6$, $30 \times 30 \times 10$, and $40 \times 40 \times 26$ for the first, second, and third row, respectively.

S4.2 Training Processing

Candidate detection is fine-tuned with pre-trained weights. The batchsize is 4, the initial learning rate is 0.001, and 300 epochs are trained. It is implemented

based on python and uses the pytorch framework. the weight training of 3D CNNs false positive reduction uses Adam optimizer, weights are initialized to Gaussian distribution $N(0, 0.01^2)$, using mini-batch training method back propagation training 300 epochs, batchsize size is 64, the initial learning rate is 0.001. This network is based on Python and uses the tensorflow framework. Both stages are trained on NVIDIA 4090 GPU with 24G memory.