

Lightweight Multimodal Anomaly Detection Based on GhostNet and FiLM

Boyao Yin *, Mingquan Zhang

North China Electric Power University, Baoding, Hebei, China

* Corresponding author

Abstract: To address the trade-off between detection accuracy and deployment efficiency in surveillance-based anomaly detection, we propose a lightweight multimodal abnormal behavior detection method based on GhostNet and FiLM fusion (G-FLM). GhostNet is employed as a lightweight visual backbone to reduce computational redundancy, while audio features are introduced to generate FiLM-based channel-wise affine parameters to generate FiLM-based channel-wise affine parameters for conditional recalibration of video features, thereby achieving effective audio-video information fusion with limited additional overhead. Experimental results on the XD-Violence dataset show that, under the same experimental configuration, G-FLM achieves an mAP of 0.7392. Compared with multimodal baselines such as Two-Stream and I3D, the mAP is improved by 5.6% and 18.0%, respectively. Precision reaches 0.8163 and F1 reaches 0.7260, both outperforming the comparison methods. At the same time, the model has only 3.63M parameters, requires only 1.9054 GFLOPs, and achieves an inference latency of 30.53 ms/batch, achieving higher detection accuracy with substantially lower computational overhead than the compared baselines. The results indicate that G-FLM achieves a good overall balance between detection accuracy and deployment efficiency, and provides a feasible lightweight solution for multimodal abnormal behavior detection in resource-constrained scenarios.

Keywords: Abnormal behavior detection, multimodal fusion, GhostNet, FiLM, lightweight model.

1. Introduction

With the growing demand for intelligent surveillance in scenarios such as public safety governance, rail transit monitoring, and campus or park inspection, abnormal behavior detection based on surveillance video has become an important task in intelligent security systems.

Abnormal events often provide complementary audio and visual cues: fighting, explosions, traffic accidents, and other events are reflected not only in visual motion patterns but also in discriminative audio cues such as impacts, shouts, and collision noises. Visual-only models are vulnerable to occlusion, poor illumination, viewpoint changes, and background clutter. Audio-video fusion helps improve robustness, but common fusion strategies, such as feature concatenation, two-stream networks, and cross-modal attention, often increase memory consumption and inference cost. Therefore, multimodal abnormal detection in resource-constrained scenarios requires a balance between recognition performance and deployment cost.

For video-based abnormal behavior detection, Sultani et al. [1] were among the first to introduce weakly supervised learning into real surveillance video anomaly detection, reducing segment-level annotation costs through video-level labels. Wu et al. [2] further constructed the XD-Violence dataset and introduced audio information, providing an important benchmark for multimodal violence detection. Tian et al. [3] and Zaheer et al. [4] improved weakly supervised anomaly detection from the perspectives of temporal feature magnitude constraints, clustering-assisted learning, and normalcy suppression. In recent Chinese studies, Mao [5] and Wu Peichen et al. [6] improved modality fusion, feature enhancement, and interaction mechanisms, indicating that the use of cross-modal information has become an important direction for complex abnormal behavior detection. However, some models rely on heavy visual backbones or complex

temporal modeling, and their Params, GFLOPs, and Latency are relatively high, which makes it difficult to meet the real-time deployment requirements of edge devices.

In multimodal fusion and lightweight modeling, two-stream convolutional networks [7], three-dimensional convolutional networks [8], and inflated three-dimensional convolutional networks [9] can enhance spatiotemporal feature representation, but they usually incur high computational overhead. GhostNet generates redundant features through computationally inexpensive linear operations and maintains good visual representation capability with a small number of parameters [10, 11]. Its subsequent improvements and applications have also verified the potential of lightweight backbones in real-time tasks [12, 13]. FiLM realizes feature-level modulation through conditional affine transformation [14]. Compared with complex attention interaction, it has a simpler forward path and is well suited to lightweight backbones.

To address the above problems, this paper proposes a lightweight multimodal abnormal behavior detection method based on GhostNet and FiLM fusion, referred to as G-FLM. GhostNet is used to construct a lightweight video branch to reduce the cost of visual feature extraction, while audio features are introduced to generate FiLM parameters for channel-level modulation, achieving audio-video fusion through a streamlined forward path. The effectiveness of the method is verified on the XD-Violence dataset, and evaluations are conducted in terms of fusion strategy, video backbone, and lightweight indicators, with emphasis on verifying the trade-off between multimodal detection performance and deployment efficiency.

2. Related Theories and Technical Foundations

2.1. Research Task Definition and Problem Modeling

Abnormal behavior detection aims to automatically identify potential risk events from surveillance videos. Unlike traditional single-label classification, abnormal behaviors in real scenarios may have multi-label attributes; therefore, this paper adopts a multi-label modeling approach. Suppose that the sample input consists of a video clip and the corresponding audio signal. The model output is a seven-dimensional class prediction vector corresponding to Normal, Fighting, Shooting, Riot, Abuse, Car accident, and Explosion.

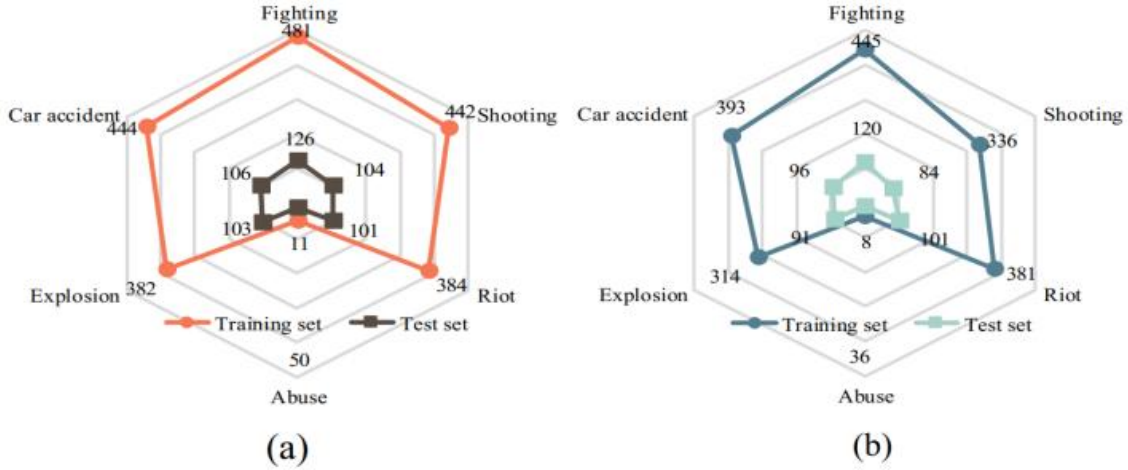


Figure 1. Schematic diagram of the XD-Violence class distribution

This paper uses the XD-Violence dataset [2] for experiments. Table 1 lists the number of training and testing samples in each class. The Normal class contains 2045 training samples and 300 testing samples, which is substantially larger than the number of samples in the abnormal classes. The Abuse class contains only 36 training samples and 8 testing samples, making it the least represented class. In the training set, the number of Normal samples is about 56.8 times that of Abuse; in the testing set, the ratio is also 37.5:1. Therefore, the class distribution shown in Fig. 1

The label vector is represented by multi-hot encoding, and the predicted logits are mapped to class probabilities through the sigmoid function.

Given the nature of the task, this study focuses on both detection accuracy and deployment feasibility under resource-constrained conditions. Therefore, the research problem can be summarized as follows: in a multi-label abnormal behavior detection task, how can a lightweight network structure and a low-overhead fusion mechanism reduce computation and deployment costs while ensuring detection performance?

2.2. Dataset Foundation and Task Challenge Analysis

exhibits a clear long-tailed distribution. The audio-video examples in Fig. 2 further show that abnormal events not only involve disturbances such as visual occlusion, poor illumination, and fast motion, but may also be accompanied by silence, noise, or inconsistent audio-visual cues. This distribution indicates that the proposed model needs to pay attention to minority-class learning and also use complementary audio-video information to improve robustness in complex scenarios.

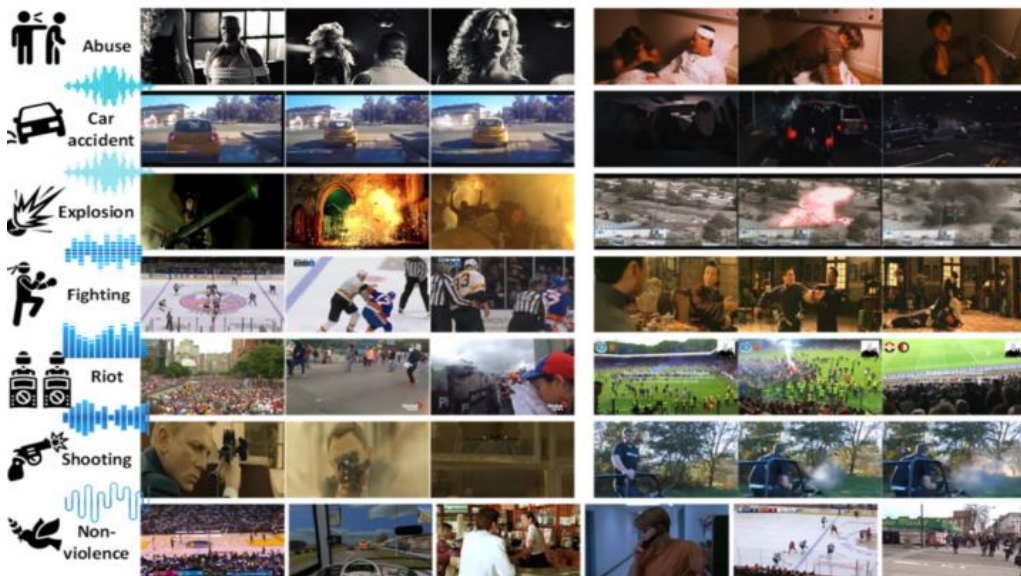


Figure 2. Audio-video examples from the XD-Violence dataset

Table 1. Statistics of class distribution in the XD-Violence dataset

Class	Training samples	Testing samples	Description
Normal	2045	300	Normal samples
Fighting	446	120	Fighting
Shooting	335	84	Shooting
Riot	381	101	Riot
Abuse	36	8	Minority class
Car accident	393	96	Traffic accident
Explosion	314	91	Explosion

2.3. Theory of the Lightweight Visual Backbone Network: GhostNet

The visual branch is usually the most computationally expensive component in abnormal behavior detection models. The core idea of GhostNet is to first generate primary features with a small number of standard convolutions, and then expand redundant feature maps using computationally inexpensive linear operations, thereby maintaining effective representation capability with a small parameter count and low GFLOPs.

GhostNet is selected because it can provide relatively stable visual representations without significantly increasing deployment cost, thereby providing a lightweight feature representation for subsequent audio-conditioned modulation. Compared with heavy two-dimensional backbones or complex temporal networks, GhostNet is more consistent with the objective of balancing accuracy and efficiency in this study for resource-constrained security scenarios.

2.4. Theoretical Basis of Cross-Modal Fusion: FiLM Conditional Modulation

The key to multimodal learning is to exploit complementarity between modalities. Common explicit cross-modal attention mechanisms are highly expressive, but often require additional pairwise feature interactions and high memory overhead. FiLM injects cross-modal information through conditional affine transformations, and its basic form can be expressed as:

$$F' = \gamma \odot F + \beta \quad (1)$$

Where F denotes video features, γ and β denote channel-level scaling and shifting parameters generated by the conditional modality, respectively, and $\odot$ denotes channel-wise multiplication. Furthermore, the parameter generation process can be written as $(\gamma, \beta) = G(z_a)$, where z_a denotes the audio embedding and $G(\cdot)$ denotes the parameter generator.

2.5. Multi-Label Learning and Class-Imbalance Optimization Methods

Because the abnormal class distribution is imbalanced, if only a standard loss function is used, the model tends to favor classes with more samples. Multi-label abnormal detection can be regarded as performing binary classification separately for each class. For class c , the true label is y_k and the predicted probability is p_k . For convenience in explaining the differences among loss functions, the BCE loss term of class c is first defined as:

$$\ell_k = -[y_k \log(p_k) + (1 - y_k) \log(1 - p_k)] \quad (2)$$

BCE directly averages the losses of all classes:

$$LBCE = (1/K) \sum_{k=1}^K \ell_k \quad (3)$$

Weighted BCE introduces class weight w_k on the basis of BCE:

$$L_W BCE = (1/K) \sum_{k=1}^K w_k \ell_k \quad (4)$$

For classes with fewer samples or higher learning difficulty, a larger w_k can be set so that they obtain a higher loss contribution during training, thereby alleviating the learning bias caused by class-count differences.

Focal Loss further adjusts from the perspective of sample difficulty. Let $p_{t,k}$ denote the predicted probability corresponding to the true label, then:

$$LFL = -(1/K) \sum_{k=1}^K \alpha_k (1 - p_{t,k})^\gamma \log(p_{t,k}) \quad (5)$$

Here, α_k is used to balance the importance of different classes, and γ is the focusing factor. When a sample is already easy to classify, a larger $p_{t,k}$ reduces its loss weight, so that the model pays more attention to hard-to-classify samples.

In summary, BCE provides the basic multi-label optimization objective; Weighted BCE mainly compensates minority classes from the perspective of class weights; and Focal Loss mainly reduces the dominant role of easy samples from the perspective of sample difficulty. These three loss functions emphasize different aspects of optimization, corresponding to multi-label optimization, class imbalance, and hard-sample learning, respectively, and provide a theoretical basis for the subsequent training-strategy selection.

2.6. Evaluation Metrics and Lightweight Evaluation Framework

We evaluate the model from two perspectives: detection performance and model efficiency (see Table 2). Detection performance is measured using mAP, AP, precision, recall, and F1 score, where mAP reflects the overall ranking capability of the multi-label task, and F1 is used to observe whether Precision and Recall are balanced under a fixed threshold. Model efficiency is assessed using the number of parameters, GFLOPs, latency, and FPS. The reason for setting the metrics in this way is that a lightweight abnormal detection model should not only pursue higher mAP, but also prove that it has lower resource consumption and higher processing efficiency on actual devices.

Experimental results are presented using tables, ablation studies, and trade-off diagrams, ablation experiments, and trade-off diagrams to show differences among models in detection performance, GFLOPs, Latency, and other metrics, thereby providing a relatively complete evaluation of the accuracy-efficiency trade-off.

Table 2. Definitions and calculation descriptions of metrics

Metric category	Metric name	Calculation or meaning
Detection performance	mAP	Mean of AP values for all classes; measures the overall performance of multi-label detection
Detection performance	AP	Area under the single-class Precision-Recall curve
Detection performance	Precision	Proportion of true positives among samples predicted as positive
Detection performance	Recall	Proportion of correctly detected positives among all true positive samples
Detection performance	F1	Harmonic mean of Precision and Recall
Model efficiency	Params	Total number of learnable parameters; reflects storage overhead
Model efficiency	GFLOPs	Floating-point operations for a single forward inference
Model efficiency	Latency	Single-batch inference time; reflects response speed
Model efficiency	FPS	Number of video clips processed per second; reflects real-time capability

Note: GFLOPs are counted using a single-sample, 16-frame, 224 x 224 input; the MACs-as-FLOPs convention is adopted, and only the computation of neural network modules is counted. Latency and FPS are obtained under the same benchmark testing protocol.

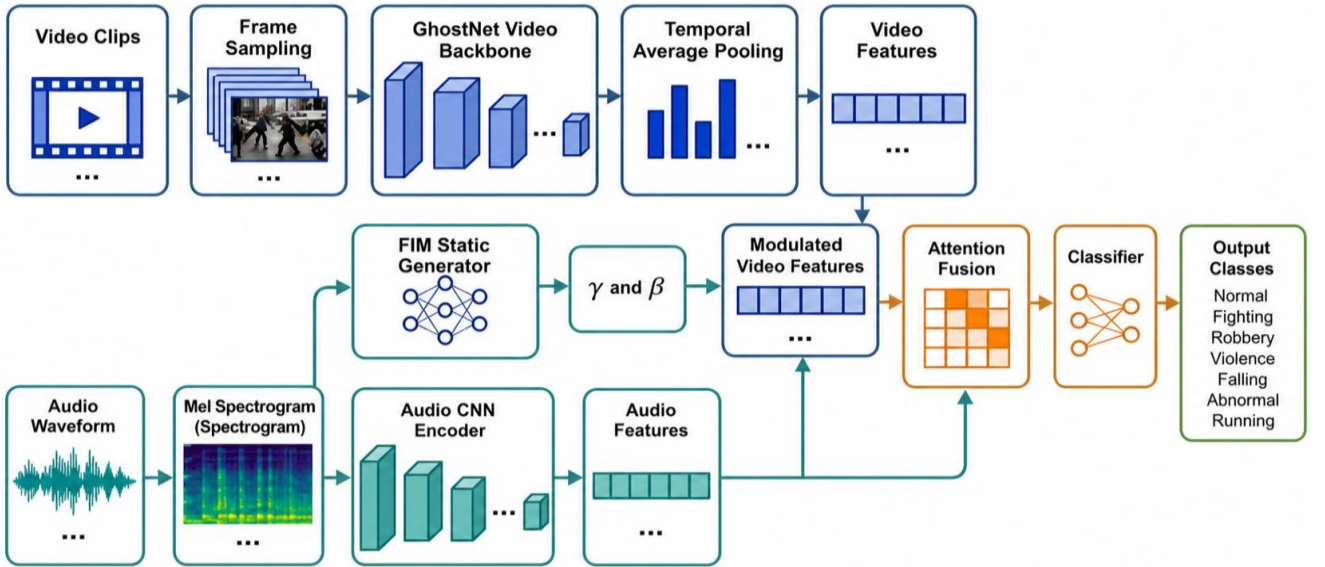
In summary, the task definition, data challenges, lightweight visual modeling, cross-modal fusion mechanism, loss optimization strategy, and evaluation system together form the basis for the proposed method and evaluation.

3. Lightweight Multimodal Abnormal Behavior Detection Method

3.1. Overall Model Architecture

This paper adopts a mainstream dual-branch multimodal architecture. Fig. 3 shows the overall structure of G-FLM, whose core is a single-backbone multimodal fusion framework of audio-conditioned guidance for video feature recalibration: the inputs are a video clip and synchronized audio, which are not directly concatenated at the raw input layer; the upper video path is responsible for extracting primary visual features, the lower audio path is responsible for extracting conditioning information from audio for subsequent modulation; the middle FiLM module converts the audio embedding into channel-level modulation parameters and applies them to video features; the attention reweighting module and classification head on the right complete abnormal-class prediction.

In this design, the video branch serves as the main discriminative path because abnormal behaviors are primarily reflected in actions, object interactions, and scene changes in video frames. The audio branch is used as the conditional path because cues such as impacts, shouts, and collision noises can often indicate the occurrence or class likelihood of abnormal events. If the two modalities are directly concatenated, the model must learn only in the final classifier which audio features are useful, which easily introduces redundancy. If only two outputs are averaged at the decision layer, audio information cannot influence intermediate video representations. FiLM is positioned between the two: it maps a 512-dimensional audio embedding into scaling and shifting parameters for 160-channel video features, thereby using audio information to modulate visual features at the channel level before classification.

**Figure 3.** Overall architecture of the proposed model

3.2. Data Processing and Input Construction

For the data construction method in this paper, the model does not take the entire long video as input. Instead, 16 frames are uniformly sampled from each sample, resized to 224 x 224 resolution, and normalized to the [0, 1] range. When the number of valid frames is insufficient, the last frame is repeated or zero frames are padded to ensure consistent batch tensor dimensions. This preprocessing serves two purposes:

first, to avoid training instability caused by different video lengths; second, to control the input scale so that GFLOPs, Latency, and FPS in subsequent complexity statistics are measured under a unified comparison protocol.

The audio branch is temporally aligned with the sampled video clip. The input is a 2 s single-channel waveform with a sampling rate of 22050 Hz. In the main experiments, a unified data pipeline reads audio waveforms and converts them into log-Mel representations during model forward propagation.

Since XD-Violence contains silent, noisy, and samples with inconsistent audio and visual cues, audio validity detection, filtering, normalization, and pre-emphasis are added in the data processing stage to stabilize the input to the audio branch.

Labels are mapped to seven-dimensional multi-hot vectors, consistent with the multi-label abnormal detection task.

3.3. Audio Branch Design

The audio branch in Fig. 3 serves as a conditional generation path. Its output does not directly produce the final class predictions, but determines how the subsequent FiLM module adjusts video channels. In the implementation, the input waveform is first converted into a log-Mel spectrogram, so that the one-dimensional time-domain signal becomes a two-dimensional time-frequency representation describing frequency distribution and short-time energy changes. The spectrogram then passes through a three-layer 2D convolutional encoder with channel numbers of 64, 128, and 256. Each layer contains Conv2d, GroupNorm, ReLU, MaxPool2d, and Dropout2d, and is used to capture anomaly-related acoustic patterns such as explosion impact sounds, fighting shouts, and traffic collision sounds.

After convolutional encoding, audio features are compressed into a global acoustic description through adaptive global pooling and projected into a 512-dimensional audio embedding by a linear layer. No separate audio classification head is used in order to avoid forming a two-stream classification framework with higher computational

overhead. More importantly, audio is expected to guide the channel-wise emphasis of visual features. For example, when strong impacts or abrupt acoustic changes are present in the audio, the model can enhance video channels related to explosions, collisions, or violent motion through FiLM; when audio cues are weak, the video branch still retains the main discriminative capability.

3.4. Video Branch Design

Fig. 4 further expands the video branch in Fig. 3. The input tensor of this branch can be represented as $B \times T \times 3 \times H \times W$, where B is batch size, T is the number of sampled frames, 3 is the RGB channel number, and H and W are frame height and width. To reuse a two-dimensional lightweight backbone, the model first rearranges the input as $(B \times T) \times 3 \times H \times W$, that is, merging the temporal dimension into the batch dimension, so that different sampled frames share the same GhostNet backbone for frame-level spatial feature extraction. Then the last feature map of GhostNet is taken as the frame-level feature. If its channel number is consistent with the 160 channels required by the subsequent FiLM module, the channel alignment module adopts an identity mapping; otherwise, a 1×1 Conv, BatchNorm, and ReLU are used to project the feature to 160 channels. Finally, the model restores it to the clip structure and performs average aggregation along the temporal dimension to form clip-level video features for subsequent FiLM conditional modulation.

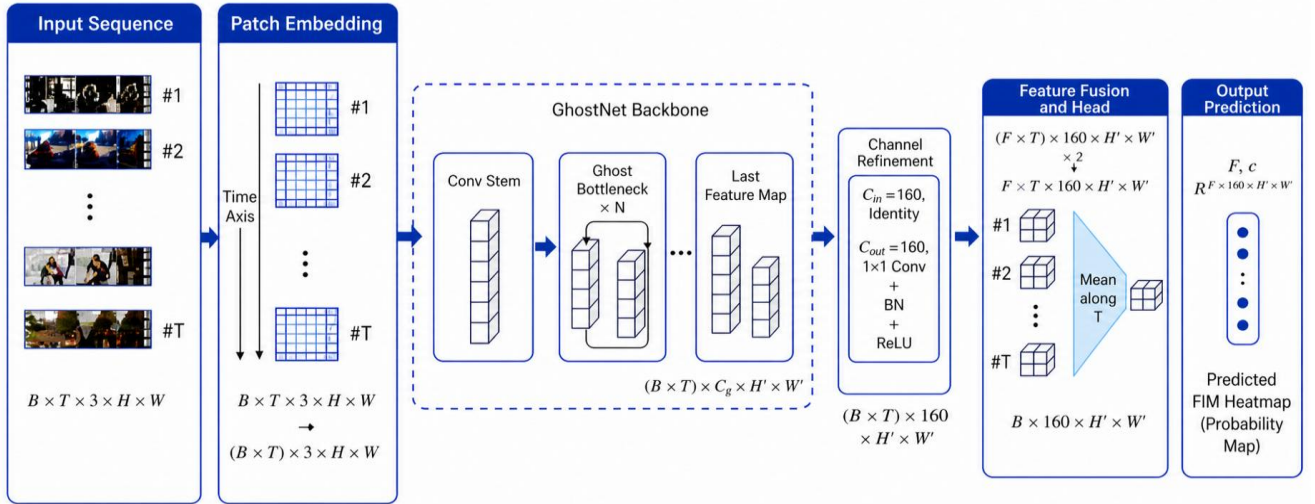


Figure 4. Schematic diagram of feature extraction in the GhostNet video branch

The key to this design is to keep the cost of temporal modeling low. The frame-level two-dimensional feature extraction - channel alignment - temporal average aggregation shown in Fig. 4 provides a lightweight design: GhostNet extracts spatial semantics from each frame, the channel alignment module ensures that the video feature dimension is consistent with the FiLM modulation module, and temporal averaging obtains a clip-level representation at low computational cost. This design avoids the additional overhead introduced by heavy temporal modules and provides stable video-feature input for subsequent audio-conditioned modulation.

3.5. FiLM Cross-Modal Fusion Mechanism

Fig. 5 illustrates the key mechanism of the proposed method. After processing, the audio embedding and video features are not simply concatenated; instead, both enter the

FiLM parameter generator. The generator outputs two sets of parameters corresponding one-to-one with video channels: γ controls the scaling of channel responses, and β controls the shifting of channel responses. For 160-channel video features, each channel obtains an audio-conditioned γ and β . Therefore, FiLM determines, at the channel level, which visual features should be enhanced and which should be suppressed and which should be suppressed. Its calculation form is as follows:

$$(\gamma, \beta) = G(z_a) \quad (6)$$

$$F' = \gamma \odot F + \beta \quad (7)$$

Combined with Fig. 5, the roles of γ and β can be further understood: the audio branch first outputs a 512-dimensional audio representation, and the FiLM generator produces γ and β with the same number of channels as the video features. For video feature map $F \in \mathbb{R}^{B \times 160 \times H' \times W'}$, γ and β are broadcast in $\mathbb{R}^{B \times 160}$ and act on each channel, can be expressed as

$F' = \gamma \odot F + \beta$. γ controls the amplification or suppression of channel responses, while β adjusts the offset of channel responses. In the implementation, γ is constrained by $\tanh(\cdot) + 1$, avoiding feature instability caused by excessive scaling; β is constrained by $\tanh(\cdot)$ to limit the shift

amplitude and avoid destroying the original video semantics through overly strong bias. Thus, audio cues can regulate video features in the form of conditional parameters, enhancing channels related to abnormal events while relatively suppressing channels related to background noise or irrelevant motion.

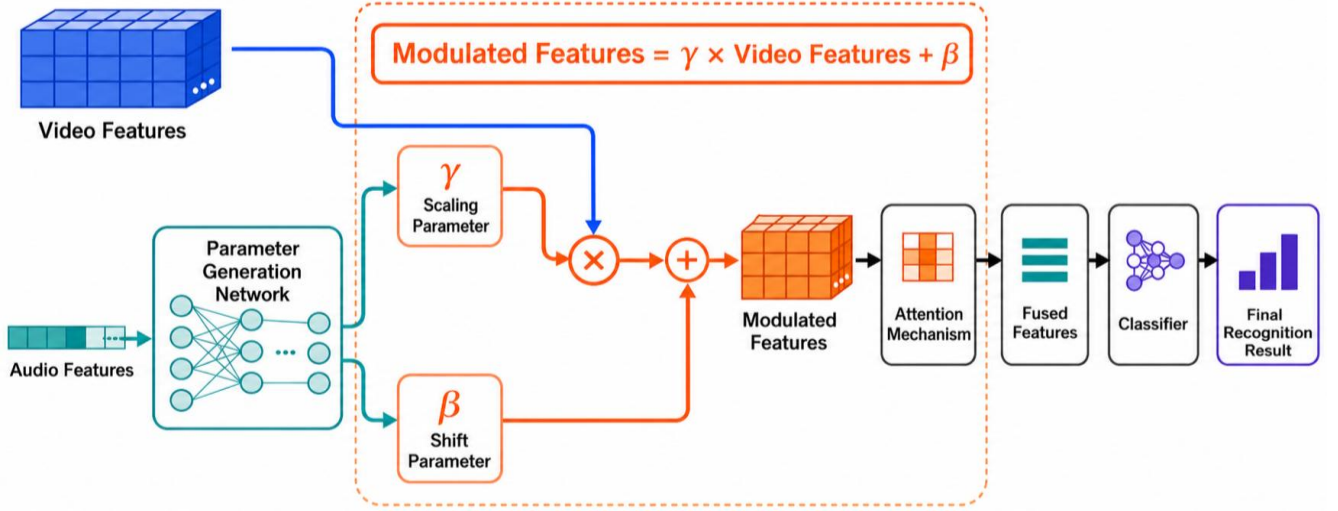


Figure 5. FiLM cross-modal modulation mechanism

In forward propagation, the model performs FiLM modulation in multiple stages and adopts residual modulation. The audio embedding is input into four FiLM parameter-generator blocks to obtain multiple groups of γ and β . The first group completes basic modulation, and the subsequent three groups gradually correct video-channel responses in a

weighted residual form, where the first two residual weights are 0.7/0.3 and the last one is 0.8/0.2. The purpose is to preserve the main discriminative ability of the original video features while allowing audio conditions to gradually correct the channel distribution, avoiding effective visual information being overwritten by one-time strong modulation.

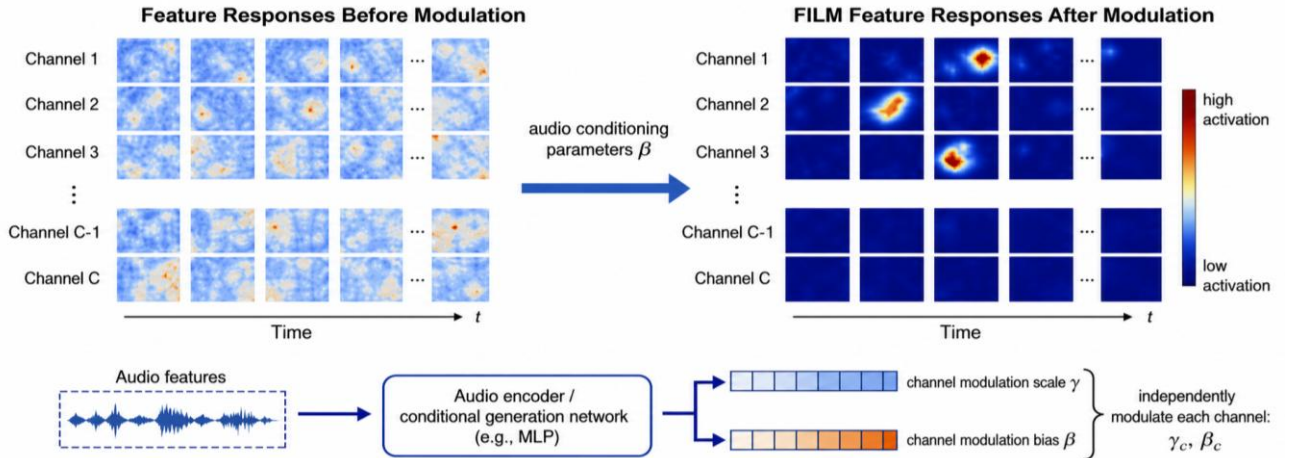


Figure 6. Schematic diagram of video feature responses before and after FiLM modulation

Fig. 6 illustrates changes in feature responses before and after FiLM modulation. Before modulation, video features are mainly determined by visual texture, motion patterns, and background structure, and their responses may be scattered. After modulation, audio-conditioned parameters adjust the importance of different channels through γ and β , making abnormal-related regions or channels more prominent while suppressing background or weakly related responses. In other words, FiLM does not modify the input frames; instead, it conditionally recalibrates the video representation at the feature level.

3.6. Attention Reweighting and Classification Output

After FiLM modulation, the model further introduces lightweight attention reweighting as a secondary feature-

selection mechanism. Specifically, the model first performs global average pooling on the modulated video features to obtain a 160-dimensional clip-level visual description; then it concatenates this with the 512-dimensional audio embedding into a 672-dimensional joint representation, and generates channel weights through a lightweight MLP of $672 \rightarrow 256 \rightarrow 160$ and a Sigmoid function. The weights are expanded to $B \times 160 \times 1 \times 1$ and applied to the video feature map, after which pooling is performed again and the result is input into the classification head. The classification head consists of Dropout, Linear (160, 512), ReLU, Dropout, and Linear (512, 7), and finally obtains seven-class probabilities through sigmoid. This module only performs reweighting along the channel dimension and does not involve complex spatial interactions, so the additional overhead is small.

3.7. Loss Function and Training Optimization Strategy

Section 2.5 introduced the related loss functions. This section focuses on the consistency between the training protocol and the implementation. The unified experimental configuration in this paper uses Focal Loss to reduce the dominant role of easy-to-classify samples in the total loss. To ensure consistency between the experimental protocol and the training script, the experimental results in this paper are reported under the Focal Loss configuration, and the same data split, input settings, and optimization parameters are maintained in all ablation groups.

4. Experimental Design and Result Analysis

To verify the effectiveness and deployment applicability of the proposed method, this paper evaluates the proposed method from four perspectives: detection performance, module contribution, lightweight performance, and class-level error. Detection performance is used to observe whether the model can effectively identify abnormal events; module contribution is used to judge whether FiLM, fusion method, and backbone selection contribute to performance improvements; lightweight performance is used to explain whether the model is suitable for resource-constrained scenarios; and class-level error is used to analyze weak classes and subsequent improvement directions.

4.1. Experimental Environment and Parameter Settings

For the input protocol, the video branch uses a fixed number of frames, unified resolution, and normalization, while the audio branch takes fixed-length waveforms and converts them into log-Mel representations during model forward propagation. All comparison models use consistent data splits, input settings, and training configurations to reduce the influence of irrelevant variables on results.

The loss function adopts Focal Loss [15], which is used to reduce the dominant role of easy-to-classify samples in the total loss. During training, data augmentation, Dropout [16], learning rate scheduler, and gradient clipping are used to improve generalization ability and training stability, and AdamW is used as the optimizer [17].

Experiments are conducted on Windows 11 Pro with an Intel Core i5-14600KF CPU, an NVIDIA GeForce RTX 5060 GPU (8151 MiB memory), and about 32 GB system memory. The software environment is Python 3.10.19, PyTorch 2.9.0+cu128, and CUDA 12.8.

In terms of evaluation protocol, detection performance is mainly measured by mAP, AP, Precision, Recall, and F1, which are used to judge the overall ranking ability, class-level recognition ability, and prediction reliability under a fixed threshold. Deployment applicability is mainly measured by Params, GFLOPs, Latency, and FPS, which characterize model size, computational cost, inference speed, and throughput. The above protocol ensures comparability for subsequent fusion-strategy, video-backbone, and lightweight analyses.

4.2. Model Complexity Statistics and Lightweight Analysis

The complexity analysis addresses whether the proposed

G-FLM remains lightweight after the audio branch and FiLM modulation are introduced. The analysis considers Params, GFLOPs, Latency, and FPS. The parameter distribution in Table 3 shows that the complete model has 3.6298 million parameters, among which the FiLM parameter generator has 1.8800M, accounting for 51.79%, and has the largest share of parameters; the video backbone has 0.9484M, accounting for 26.13%; the audio branch has 0.5021M, accounting for 13.83%; and the attention reweighting module plus classification head account for less than 10% in total. This shows that model complexity mainly comes from cross-modal modulation and visual feature extraction, rather than the final classification layer.

Table 3. Proportion of Params in each model module

Module	Params (M)	Proportion (%)	Description
Audio branch	0.5021	13.83	Mel-spectrogram/audio convolutional encoding
Video backbone	0.9484	26.13	GhostNet visual feature extraction
Video projection layer	0.0000	0.00	GhostNet output dimension is already matched
FiLM parameter generator	1.8800	51.79	Four groups of γ/β parameter generation modules in total
Attention reweighting module	0.2134	5.88	Audio-video joint channel-weight generation
Classification head	0.0860	2.37	Seven-class multi-label prediction
Total	3.6298	100.00	Complete G-FLM model

Structurally, the FiLM parameter generator accounts for more than half, indicating that audio-conditioned modulation is the main learnable part of the model; the video backbone accounts for about one quarter, which is consistent with its role as a lightweight visual branch; the video projection layer introduces no additional parameters, indicating that the GhostNet output dimension already matches the subsequent modules and does not require additional projection overhead. Therefore, the lightweight property of G-FLM is not achieved by removing multimodal components, but by controlling the overall scale through a lightweight backbone and concise modulation structure.

4.3. Comparison Experiments with Existing Methods

To further verify the effectiveness of the proposed method in the abnormal behavior detection task, this paper compares C3D, Two-Stream, I3D approximate baselines, and the proposed G-FLM under a unified training protocol.

Table 4. Detection performance comparison of different methods on the XD-Violence dataset

Method	mAP	Precision	Recall	F1
C3D	0.5434	0.6087	0.3924	0.4772
Two-Stream	0.7002	0.6914	0.5543	0.6153
I3D	0.6266	0.6818	0.4414	0.5359
G-FLM	0.7392	0.8163	0.6537	0.7260

Table 4 gives the detection performance comparison of different methods. The C3D structure is relatively compact, but its mAP and F1 are 0.5434 and 0.4772, respectively, indicating that a simple three-dimensional convolutional baseline has limited representation capability under the current data scale and class imbalance. Two-Stream introduces dynamic cues through spatial and motion-difference branches, improving mAP to 0.7002, but F1 is still lower than that of G-FLM. I3D has strong spatiotemporal modeling capability, but under the unified training protocol in this paper, its mAP is 0.6266, lower than Two-Stream and G-FLM. In contrast, G-FLM achieves an mAP of 0.7392, Precision of 0.8163, and F1 of 0.7260, indicating that audio-conditioned modulation can enhance abnormal-class discrimination without relying on heavy temporal structures.

Accuracy metrics alone are insufficient to demonstrate the deployment value of a lightweight model; therefore, model efficiency is further compared from four aspects: parameter count, theoretical computation, inference latency, and throughput. Table 5 shows that Two-Stream has 22.88M parameters, GFLOPs of 56.4309, and inference latency of 66.20 ms/batch. Although its mAP is close to 0.70, its deployment overhead remains high. I3D has GFLOPs of 18.2370, also significantly higher than the main model in this paper. G-FLM has 3.63M parameters and GFLOPs of 1.9054, significantly reducing theoretical computation while maintaining the highest mAP. Its latency is 30.53 ms/batch, comparable to that of I3D, but its accuracy is significantly higher. Compared with Two-Stream, G-FLM has both lower latency and higher detection performance. This result shows that the advantage of the proposed method mainly lies in the comprehensive balance between accuracy and efficiency, rather than superiority on only one metric.

Table 5. Deployment efficiency comparison of different methods

Method	Params	GFLOPs	Latency	FPS
C3D	3.51	9.5715	11.68	342.43
Two-Stream	22.88	56.4309	66.20	60.42
I3D	7.92	18.2370	30.51	131.10
G-FLM	3.63	1.9054	30.53	131.01

4.4. Ablation Experiments

To further analyze the contribution of each module to model performance, this paper conducts ablation experiments from the perspectives of fusion strategy and video backbone.

All ablation experiments are conducted under the same training settings and data-processing pipeline whenever applicable to reduce the influence of irrelevant variables on experimental conclusions.

4.4.1. Ablation of Fusion Strategies

The fusion-strategy ablation focuses on how audio information can be incorporated effectively. The pure-video baseline does not use audio and can therefore serve as the starting point for judging audio gains. Concat and Late Fusion represent simple feature-level fusion and decision-level late fusion, respectively. FiLM uses audio as a conditional signal acting on video channels. Table 6 gives the mAP and F1 of the four settings, and Fig. 7 visualizes these differences. These results are used to evaluate the effect of different fusion strategies on detection performance and threshold stability.

Specifically, the pure-video baseline has an mAP of 0.7165 and an F1 of 0.6394, indicating that relying only on visual cues already provides a certain level of detection capability, but the comprehensive performance under a fixed threshold is limited. After adding audio, the mAP of Concat increases to 0.7321, and the F1 of Late Fusion increases to 0.6527, indicating that audio information contributes to performance improvement; however, the improvement is relatively limited, showing that simple concatenation or decision-level averaging cannot fully exploit audio-video complementarity. FiLM reaches 0.7392 in mAP and 0.7260 in F1, improving by 0.0227 and 0.0866 over the pure-video baseline, respectively, with the F1 improvement being especially obvious. In Fig. 7, FiLM achieves the highest values for both metrics, indicating that conditional modulation improves not only overall ranking capability but also the balance between Precision and Recall.

Table 6. Ablation results of fusion strategies

Setting	Audio input	Fusion method	mAP	F1
Pure-video baseline (GhostNet)	No	None	0.7165	0.6394
Concat	Yes	Feature concatenation	0.7321	0.6491
Late Fusion	Yes	Decision-level fusion	0.7296	0.6527
FiLM	Yes	Conditional modulation	0.7392	0.7260

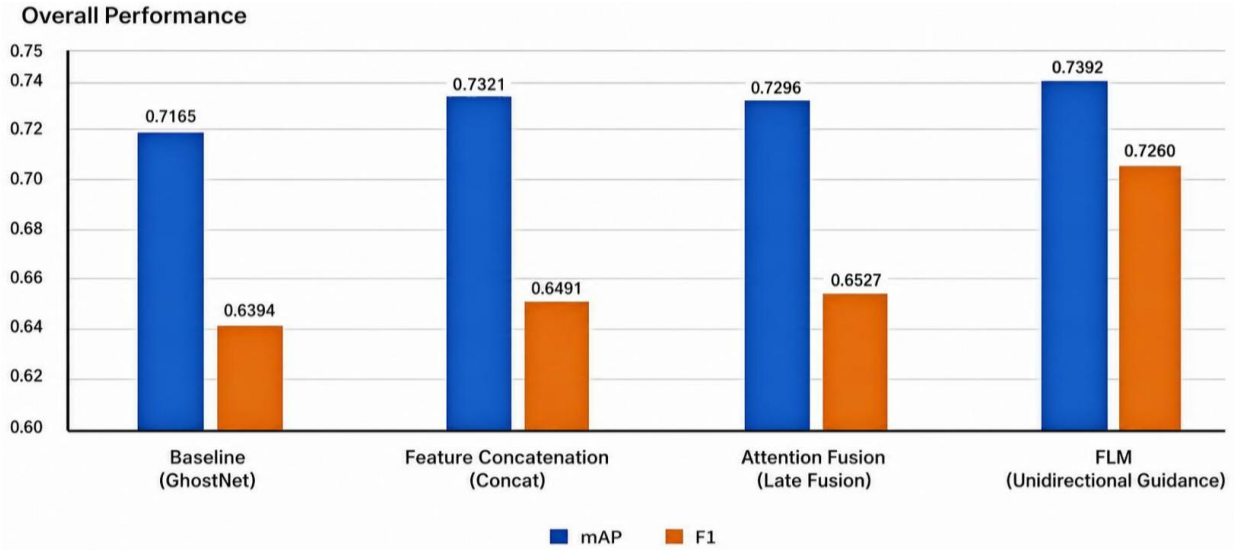


Figure 7. Comparison of mAP and F1 under different fusion strategies

4.4.2. Ablation of Video Backbones

With the FiLM fusion method fixed, different video backbones such as GhostNet, MobileNetV3 [18], and ResNet18 [19] are compared in terms of performance and complexity. This experiment is used to verify the suitability of GhostNet for lightweight modeling and analyze the trade-off between detection accuracy and deployment overhead of different backbones.

The video-backbone ablation further examines whether a lightweight backbone is sufficiently effective. Table 7 shows that G-FLM has 3.63M Params, which is lower than the 5.81M parameters of MobileNetV3 + FiLM and the 13.94M parameters of ResNet18 + FiLM; its GFLOPs are 1.905, only

about 48.8% of MobileNetV3 + FiLM and about 6.5% of ResNet18 + FiLM. Meanwhile, the latency of G-FLM is 30.53 ms and FPS is 131.01, its latency is lower and FPS is higher than those of the other two backbone configurations. More importantly, the mAP of G-FLM is 0.7392, higher than those of MobileNetV3 + FiLM (0.7035) and ResNet18 + FiLM (0.7008). Fig. 8 places mAP, GFLOPs, and Latency in the same trade-off diagram, showing more clearly that G-FLM is located in the high-accuracy, low-computation, and low-latency region. This shows that the proposed method does not exchange increased model complexity for accuracy, but achieves a more reasonable accuracy-efficiency balance under the current task conditions.

Table 7. Comparison of detection performance and lightweight efficiency of different video backbones

Model configuration	Params	GFLOPs	Latency	FPS	mAP
G-FLM	3.63	1.905	30.53	131.01	0.7392
MobileNetV3 + FiLM	5.81	3.901	33.50	119.41	0.7035
ResNet18 + FiLM	13.94	29.404	42.91	93.22	0.7008

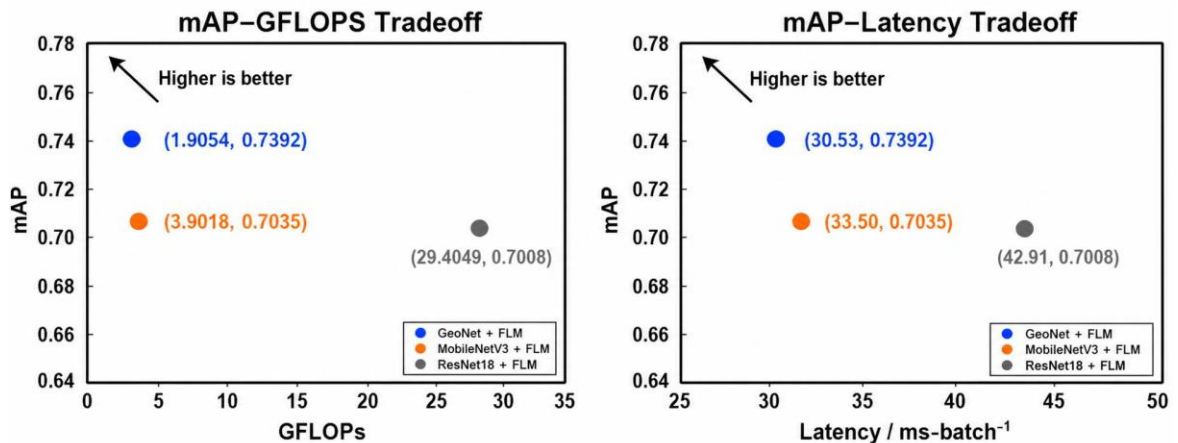


Figure 8. mAP-GFLOPs and mAP-Latency trade-off of different video backbones

4.5. Lightweight Performance Analysis of the Model

The lightweight performance analysis aims to determine not only which model is faster, but also whether a model is suitable for resource-constrained security scenarios by combining multiple deployment indicators. Params affect

model storage and loading, GFLOPs affect hardware computational requirements, Latency determines inference response time, and FPS determines processing throughput. Therefore, only when a model performs well across these indicators simultaneously can it be considered to have good deployment applicability.

Fig. 9 comprehensively presents the lightweight

performance of different backbone models. The figure shows normalized scores (the larger the score, the better), rather than raw values, and the ranking is consistent with the specific values in Table 7. G-FLM achieves the lowest Params, lowest GFLOPs, lowest Latency, and highest FPS, and its mAP is also the highest; therefore, it obtains a higher overall score. Compared with ResNet18 + FiLM, G-FLM does not simply

sacrifice accuracy for speed, but maintains and even improves detection performance while reducing computational overhead. Compared with MobileNetV3 + FiLM, G-FLM has lower computation and fewer parameters, and its detection results are also better. This shows that the combination of GhostNet and FiLM is more consistent with the design goal of lightweight multimodal detection in this task.

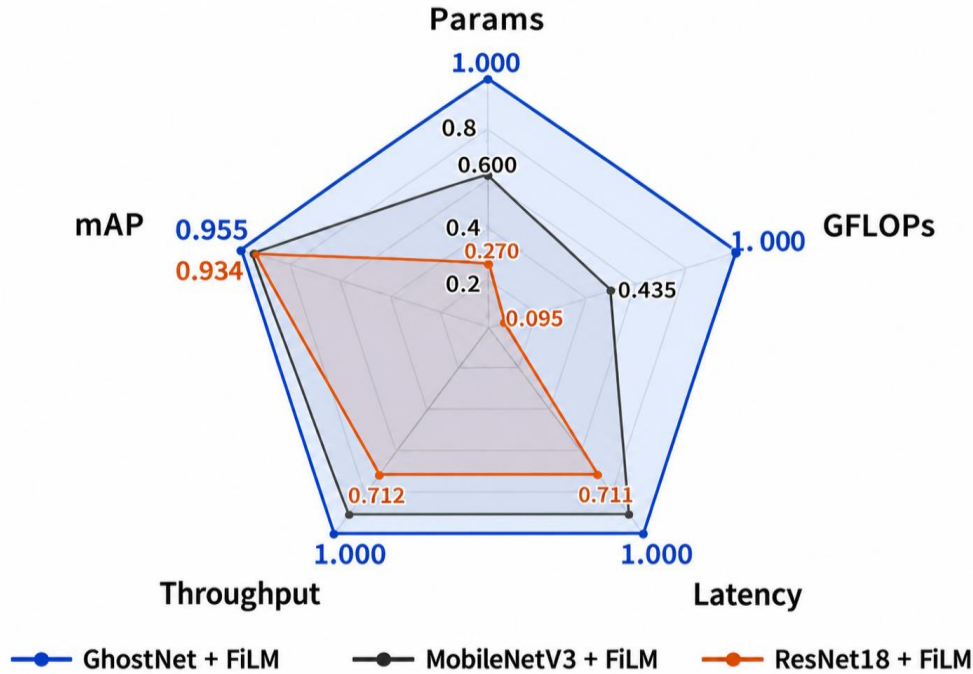


Figure 9. Comparison of lightweight performance scores of different backbone models

4.6. Class-Level Results and Error Analysis

Class-level analysis is used to observe whether the model performs consistently across abnormal categories. The AP results in Fig. 10 show that Riot, Car accident, Normal, and Explosion reach 0.9536, 0.9519, 0.9378, and 0.8621, respectively, indicating that the model detects classes with obvious visual motion patterns or prominent acoustic cues more reliably. The AP of Fighting is 0.7578, ranking in the middle, which may be related to the large variations in fighting behaviors. The AP values of Shooting and Abuse are 0.4288 and 0.2827, respectively, significantly lower than other classes. The difference between the highest class Riot and the lowest class Abuse is 0.6709. This gap corresponds to the extremely small number of Abuse samples in Table 1, indicating that class imbalance directly affects the model's

ability to learn long-tailed classes. The error examples in Fig. 10 further explain the causes of low-AP classes. Missed detections are usually characterized by short abnormal duration, severe visual occlusion, or weak audio signals, making it difficult for the model to obtain stable discriminative cues. False positives often arise from violent motion or sudden noise in normal scenes, indicating that strong motion and abrupt or intense acoustic signals can still interfere with model judgment. Inter-class confusion mostly occurs between semantically similar classes such as Riot and Fighting, indicating ambiguous boundaries between these categories. Combining class AP and error examples, subsequent improvements should prioritize low-AP classes such as Abuse and Shooting, and strategies such as resampling, threshold calibration, or few-shot optimization can be considered.

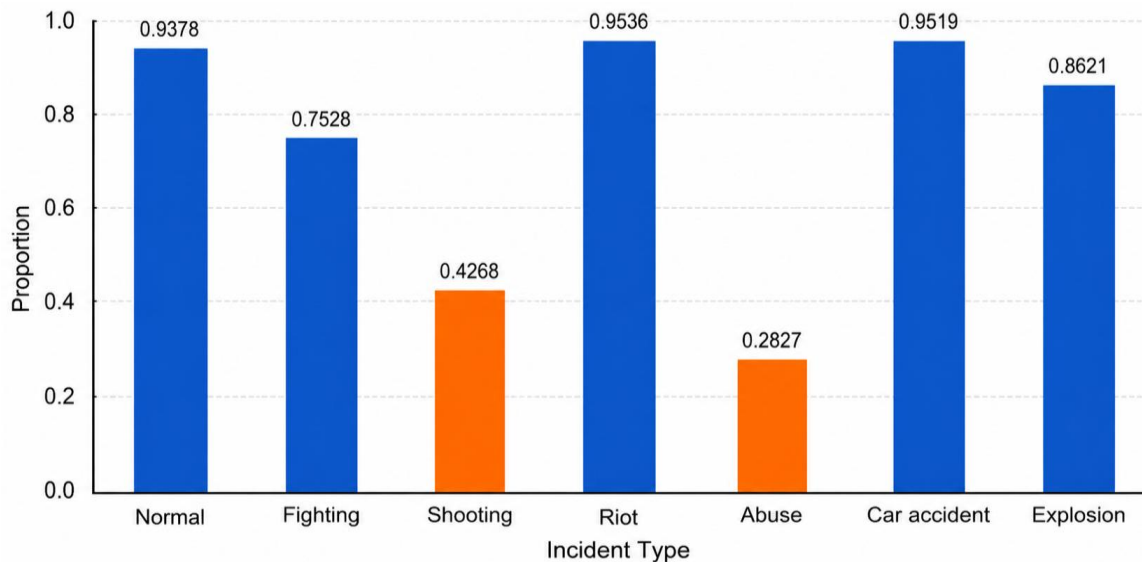


Figure 10. Class-wise AP comparison and error sample analysis

5. Conclusion

To meet the need for abnormal behavior detection in resource-constrained scenarios, this paper proposes a lightweight multimodal detection method based on GhostNet and FiLM fusion. The method reduces feature extraction cost through a lightweight video backbone and enhances video feature representation through audio-conditioned modulation, achieving audio-video fusion with controllable computational overhead.

Experimental results show that, under a unified training protocol, the proposed method achieves better overall performance than the pure-video baseline and common multimodal fusion methods, and improves abnormal behavior detection performance while maintaining low model complexity and inference overhead.

However, the proposed method still has room for improvement in recognizing minority classes and fine-grained temporal information. In the future, few-shot optimization, lightweight temporal modeling, and edge-device deployment validation will be further explored to facilitate the deployment of lightweight abnormal behavior detection methods on edge devices in practical scenarios such as security surveillance.

References

- [1] Sultani, W., Chen, C., & Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6479-6488). Salt Lake City, USA.
- [2] Wu, P., Liu, J., Shi, Y., et al. (2020). Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *European Conference on Computer Vision* (pp. 322-339). Glasgow, UK.
- [3] Tian, Y., Pang, G., Chen, Y., et al. (2021). Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4975-4986). Montreal, Canada.
- [4] Zaheer, M. Z., Mahmood, A., Astrid, M., & Lee, S. I. (2020). CLAWS: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection. In *European Conference on Computer Vision*. Glasgow, UK.
- [5] Mao, J. (2024). Research on weakly supervised violence detection technology based on modality fusion (Master's thesis). Xidian University, China. (In Chinese)
- [6] Wu, P., Li, W., Guo, F., et al. (2025). Video anomaly detection based on feature enhancement and modal interaction. *Journal of Computer-Aided Design & Computer Graphics*, 37(3), 407-413. (In Chinese)
- [7] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems* (pp. 568-576). Montreal, Canada.
- [8] Tran, D., Bourdev, L., Fergus, R., et al. (2015). Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4489-4497). Santiago, Chile.
- [9] Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? A new model and the Kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6299-6308). Honolulu, USA.
- [10] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). GhostNet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1580-1589). Seattle, USA. arXiv:1911.11907.
- [11] Liu, C., Duan, X., & Xie, B. (2022). Improvement of ECO target tracking algorithm based on GhostNet convolution features. *Laser Technology*, 46(2), 239-247. (In Chinese)
- [12] Tang, Y., Han, K., Guo, J., Xu, C., Xu, C., & Wang, Y. (2022). GhostNetV2: Enhance cheap operation with long-range attention. In *Advances in Neural Information Processing Systems*. New Orleans, USA. arXiv:2211.12905.
- [13] Putra, A. K., Santika, S. P., Peranginangin, E., & Girsang, A. S. (2025). Lightweight YOLO11 with GhostNet and CBAM for real-time fire detection. In *2025 IEEE International Symposium on Future Telecommunication Technologies* (pp. 316-321). Lombok, Indonesia. <https://doi.org/10.1109/SOFTT67007.2025.11213346>
- [14] Perez, E., Strub, F., de Vries, H., Dumoulin, V., & Courville, A. (2018). FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 3942-3951). New Orleans, USA. arXiv:1709.07871.
- [15] Lin, T. Y., Goyal, P., Girshick, R., et al. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE*

- International Conference on Computer Vision (pp. 2980-2988). Venice, Italy.
- [16] Srivastava, N., Hinton, G., Krizhevsky, A., et al. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929-1958.
- [17] Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*. New Orleans, USA.
- [18] Howard, A., Sandler, M., Chen, B., et al. (2019). Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1314-1324). Seoul, South Korea.
- [19] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778). Las Vegas, USA.
- [20] Chen, X. (2021). Research on fast object finding technology in video (Master's thesis). Xidian University, China. (In Chinese)