Online Resource: Supplementary Information for

Confidence-Calibrated Temporal Attention Framework for Interpretable Video Anomaly Detection

Rajat Gupta[1,3*] and Nidhi Tyagi[2]

[1*]Ph.D. Research Scholar, Shobhit Institute of Engineering and Technology (Deemed to be University), Meerut, India.
[2]Professor, School of Computational Science and Engineering, Shobhit Institute of Engineering and Technology (Deemed to be University), Meerut, India.
[3]Assistant Professor, Bharati Vidyapeeth's College of Engineering, Delhi, India.

*Corresponding author(s). E-mail(s): rajjatgupta2@gmail.com;
Contributing authors: nidhi.tyagi@shobhituniversity.ac.in;

Abstract

This document provides additional technical details, extended experiments, mathematical derivations, algorithms, figures, tables, and discussions that could not be included in the main manuscript due to the strict 10-page limit of the Springer-Signal, Image and Video Processing format. All notation follows the main paper.

Contents

- S1: Network Architecture and Model Details
- S2: Mathematical Derivations and Additional Formulations
- S3: Training Details and Hyperparameter Settings
- S4: Additional Qualitative Results
- S5: Extended Ablation Studies

S1 Network Architecture and Model Details

This section provides detailed technical specifications of the proposed confidence-guided spatiotemporal framework, including architectural design, feature extraction, confidence modeling, and module interactions. It extends the methodology presented in the main manuscript by offering implementation-level insights.

S1.1 Overview of the Proposed Architecture

Video anomaly detection is commonly formulated as a score-based prediction problem, where models learn representations of normal patterns and assign higher anomaly scores to deviations from these patterns [1, 2]. While such approaches have demonstrated strong performance, they typically treat anomaly prediction as a deterministic process and do not explicitly account for the reliability or uncertainty associated with model outputs. This limitation can result in unstable predictions, particularly in complex surveillance environments characterized by dynamic scenes, occlusions, and ambiguous event boundaries.

To address these challenges, the proposed confidence-guided spatiotemporal learning framework extends conventional anomaly detection pipelines by introducing a unified architecture that jointly models feature representation, temporal dynamics, prediction confidence, and interpretability. Unlike existing methods that rely solely on anomaly scores for decision-making [8, 9], the proposed framework incorporates an explicit confidence estimation mechanism to quantify prediction reliability.

Formally, given a video sequence segmented into temporal units $\{x_t\}_{t=1}^{T}$, the model extracts spatial features and encodes temporal dependencies to obtain a latent representation h_t for each segment. Based on this representation, the framework produces two complementary outputs:

- An anomaly score $s_t \in [0, 1]$, representing the likelihood of the segment being anomalous,
- A confidence score $c_t \in [0, 1]$, reflecting the reliability of the predicted anomaly score.

These outputs are integrated using a confidence-weighted formulation:

$$\tilde{s}_t = c_t \cdot s_t \tag{1}$$

where $\tilde{s}_t$ denotes the final anomaly prediction. This formulation enables the model to distinguish between reliable and uncertain detections, thereby improving robustness under noisy or ambiguous conditions. In particular, predictions with high anomaly scores but low confidence can be interpreted as uncertain cases, enabling more informed decision-making.

The architecture further incorporates a temporal modeling module to capture sequential dependencies across video segments, which is essential for detecting anomalies that manifest as irregular motion patterns or temporal inconsistencies [9, 13]. In addition, an interpretability module is integrated to provide visual explanations of anomaly predictions by highlighting salient spatiotemporal regions contributing to the

decision. This aligns with recent advances in explainable artificial intelligence, where transparency and reliability are critical for deployment in safety-critical systems [3, 24].

The complete architecture of the proposed framework is illustrated in Fig. S1, which extends Fig. 1 in the main manuscript. The supplementary figure provides a comprehensive visualization of the data flow across different components, including feature extraction, temporal encoding, confidence-guided scoring, and interpretability generation. This detailed representation supports reproducibility and facilitates a deeper understanding of the proposed methodology.

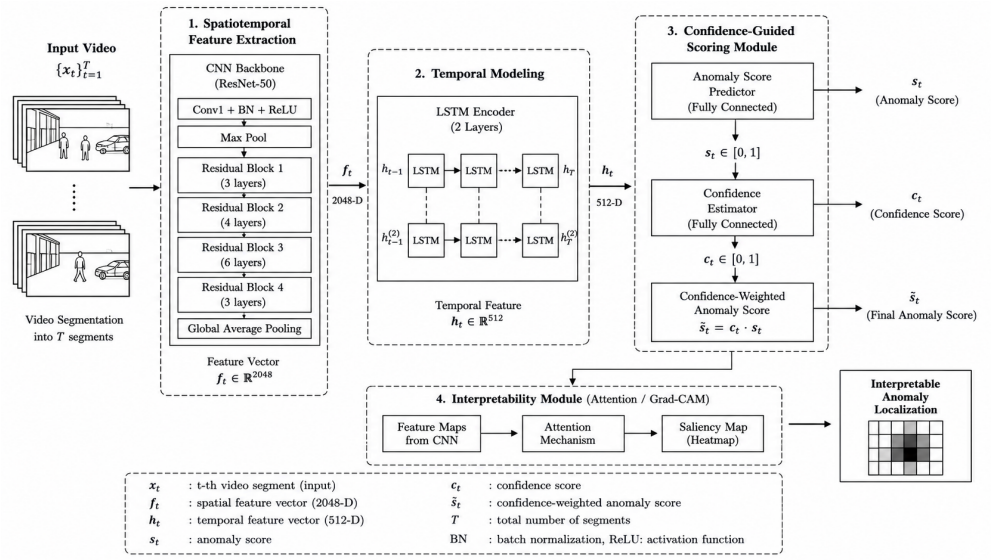


Fig. 1 Detailed architecture of the proposed confidence-guided spatiotemporal learning framework. The model integrates feature extraction, temporal modeling, confidence-aware scoring, and interpretability modules to produce reliable and explainable anomaly predictions.

Overall, the proposed architecture extends conventional anomaly detection frameworks by incorporating reliability-aware learning and interpretable inference within a unified design. By jointly modeling anomaly likelihood and prediction confidence, the framework improves both detection robustness and transparency, making it suitable for real-world surveillance applications.

S1.2 Architecture Diagram

Fig. S1 provides a detailed visualization of the proposed confidence-guided spatiotemporal learning framework, illustrating the complete data flow from raw video input to final anomaly predictions with interpretability. This section complements the overview presented in Section S1.1 by describing module-level operations.

The framework follows a modular pipeline consisting of four primary components: (i) spatiotemporal feature extraction, (ii) temporal modeling, (iii) confidence-guided

scoring, and (iv) interpretable inference. Each component is designed to address key limitations of conventional anomaly detection methods, particularly the absence of reliability estimation and limited model transparency.

The pipeline begins with an input video sequence segmented into T non-overlapping temporal units $\{x_t\}_{t=1}^{T}$, enabling efficient processing of long video streams while preserving localized temporal patterns. Each segment is processed using a deep convolutional backbone network to extract high-level spatial features. As shown in Fig. S1, the backbone comprises convolutional and residual layers that encode appearance information such as object structure, motion cues, and scene context. Deep convolutional representations have been shown to be effective for anomaly detection tasks due to their strong feature learning capabilities [1, 8].

The extracted feature vectors f_t are subsequently passed to the temporal modeling module, which captures sequential dependencies across video segments. In this framework, temporal modeling is implemented using a multi-layer Long Short-Term Memory (LSTM) network. LSTM architectures are particularly effective for modeling temporal dynamics due to their ability to retain long-term dependencies and mitigate vanishing gradient issues [9, 13]. This enables the detection of anomalies that manifest as irregular motion patterns or temporal inconsistencies.

Following temporal encoding, the latent representation h_t is fed into the confidence-guided scoring module. This module consists of two parallel branches that estimate anomaly likelihood and prediction confidence. The anomaly score s_t quantifies the likelihood of abnormality, while the confidence score c_t reflects prediction reliability. These outputs are combined to produce the final confidence-weighted anomaly score $\tilde{s}_t$, allowing the model to differentiate between reliable and uncertain detections. This design addresses a key limitation of conventional methods, which often produce overconfident predictions in ambiguous scenarios [2, 3].

In addition to detection, the framework incorporates an interpretability module that generates visual explanations of anomaly predictions. This module operates on intermediate feature maps and applies attention-based attribution methods such as Grad-CAM to highlight salient regions contributing to the model's decision. Such interpretability is essential for understanding model behavior and improving trust in real-world surveillance systems [3, 24].

The modular design of the architecture ensures clear separation of functionality while enabling flexibility for future extensions, such as integrating transformer-based temporal models [21]. Furthermore, the explicit data flow illustrated in Fig. S1 enhances reproducibility and provides practical guidance for implementation.

Overall, Fig. S1 serves as a detailed blueprint of the proposed framework, illustrating the interaction between feature extraction, temporal modeling, confidence estimation, and interpretability, thereby supporting reliable and explainable video anomaly detection.

S1.3 Spatiotemporal Feature Extraction Module

Backbone Network

The first stage of the proposed framework focuses on extracting high-level spatial representations from input video segments using a deep convolutional neural network (CNN) backbone. Given a sequence of temporally segmented inputs $\{x_t\}_{t=1}^{T}$, each segment is independently processed to produce a discriminative feature embedding defined as:

$$f_t = \phi(x_t) \tag{2}$$

where $x_t \in \mathbb{R}^{H\times W\times 3}$ represents the input frame or segment, $\phi(\cdot)$ denotes the nonlinear mapping implemented by the CNN, and $f_t \in \mathbb{R}^{D}$ is the resulting feature vector.

Deep CNN architectures are effective in learning hierarchical representations that capture semantic, structural, and contextual information from visual data [1, 8]. These representations are particularly useful for anomaly detection, where subtle deviations from normal patterns must be identified.

Backbone Configuration

The configuration of the CNN backbone is summarized in Table 1. A ResNet-50 architecture pretrained on ImageNet is employed to leverage rich feature representations learned from large-scale datasets. Input frames are resized to a resolution of 224×224 to ensure consistency and compatibility with the pretrained model.

Table 1 Configuration of the CNN backbone used for spatiotemporal feature extraction.

Component	Specification
Backbone Type	ResNet-50 (pretrained on ImageNet)
Input Resolution	224×224
Output Feature Dimension	2048
Activation Function	ReLU
Normalization	Batch Normalization
Pooling Strategy	Global Average Pooling

The convolutional layers progressively transform input frames into abstract feature representations, which are aggregated using global average pooling to produce compact feature vectors. The use of ReLU activation and batch normalization improves training stability and generalization.

Feature Representation and Learning

The extracted feature vector f_t encodes multiple levels of spatial information, including:

- Object-level semantics for detecting unusual entities,
- Scene-level context describing environmental structure,
- Local spatial irregularities indicating abnormal visual patterns.

To enhance task-specific performance, the pretrained backbone is fine-tuned on the target datasets. This enables adaptation to domain-specific characteristics and improves anomaly discrimination. Fine-tuning has been shown to significantly enhance detection performance in deep learning-based anomaly detection systems [1].

Integration with Temporal Modeling

While the CNN backbone captures spatial features effectively, anomaly detection in videos also requires modeling temporal dependencies. Therefore, the extracted features $\{f_t\}$ are passed to the temporal modeling module (Section S1.4), enabling the framework to capture both spatial and temporal anomalies.

Design Justification

The selection of ResNet-50 is motivated by its strong representational capacity and efficient training enabled by residual connections. Compared to shallow architectures or handcrafted features, deep CNN-based representations provide improved generalization across diverse environments [2]. Additionally, the use of pretrained weights reduces training time and improves convergence stability.

Overall, the spatiotemporal feature extraction module forms a robust foundation for the proposed framework by generating high-quality representations that support reliable anomaly detection.

S1.4 Temporal Modeling Module

While spatial feature extraction captures appearance-based information, effective video anomaly detection additionally requires modeling temporal dependencies across sequential segments. To address this, the proposed framework incorporates a transformer-based temporal modeling module that captures both short-range and long-range dependencies, enabling the detection of complex and evolving anomalies.

Given the sequence of feature vectors $\{f_t\}_{t=1}^{T}$ extracted from the CNN backbone (Section S1.3), the temporal representations are computed as:

$$H = \text{TransformerEncoder}(F), \tag{3}$$

where $F = \{f_t\}_{t=1}^{T}$ denotes the input feature sequence and $H = \{h_t\}_{t=1}^{T}$ represents the context-aware temporal representations.

Temporal Network Design

The temporal modeling module is implemented using a multi-head transformer encoder architecture to capture long-range temporal dependencies and contextual relationships across video segments. Transformer-based temporal modeling enables more effective representation of global temporal patterns compared to recurrent architectures. The configuration of the temporal network is provided in Table 2.

Table 2 Configuration of the temporal modeling module.

Component	Specification
Model Type	Transformer Encoder
Embedding Dimension	512
Number of Layers	2
Number of Heads	8
Dropout Rate	0.1
Sequence Processing	Parallel self-attention

Temporal Representation Learning

The transformer-based temporal module transforms spatial features into context-aware representations that encode both local and global temporal dynamics. Specifically, the learned representations $\{h_t\}$ capture motion patterns, temporal consistency, and long-range dependencies across the entire sequence. Such representations are essential for detecting anomalies that evolve over time, including abrupt motion changes and irregular temporal transitions.

Role in Anomaly Detection

By incorporating transformer-based temporal modeling, the framework extends beyond static appearance analysis and enables detection of dynamic anomalies. Events such as abnormal motion trajectories or unexpected temporal transitions cannot be reliably identified using spatial features alone. The transformer encoder effectively captures these temporal irregularities, thereby improving detection robustness.

Furthermore, the temporal representations h_t serve as inputs to the confidence-guided scoring module (Section S1.5), ensuring that anomaly predictions are informed by both spatial context and temporal evolution.

Design Justification

The use of a transformer encoder is motivated by its ability to model long-range dependencies and capture global temporal context through self-attention mechanisms. Unlike recurrent architectures, transformer-based models process sequences in parallel, enabling efficient learning of complex temporal relationships. This makes them particularly suitable for video anomaly detection, where anomalies may depend on both short-term and long-term temporal patterns. The chosen configuration provides a balance between representational capacity and computational efficiency.

S1.5 Confidence-Guided Scoring Module

A key contribution of the proposed framework lies in integrating confidence-aware learning into the anomaly detection process. Unlike conventional approaches that rely solely on anomaly scores, this module jointly estimates anomaly likelihood and prediction confidence, enabling reliability-aware inference.

Anomaly Score Estimation

Given the temporal representation h_t, the anomaly score is computed as:

$$s_t = \sigma(\mathbf{w}^\top h_t) \tag{4}$$

where $\mathbf{w}$ is a learnable parameter vector and $\sigma(\cdot)$ denotes the sigmoid activation function. The anomaly score quantifies the likelihood of abnormality based on deviations from learned patterns.

Confidence Estimation

To improve uncertainty estimation, confidence scores are computed using stochastic Monte Carlo dropout inference. Multiple stochastic forward passes are performed during inference to estimate predictive variance:

$$c_t = \frac{1}{\mathrm{Var}(s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(N)}) + \epsilon} \tag{5}$$

where N denotes the number of stochastic forward passes and ϵ is a small constant for numerical stability. Lower predictive variance corresponds to higher confidence and improved prediction reliability.

This formulation enables the model to capture epistemic uncertainty by evaluating the consistency of predictions under stochastic perturbations, thereby improving robustness in ambiguous scenarios.

Confidence-Guided Temporal Aggregation

Instead of direct score modulation, prediction reliability is incorporated through a confidence-calibrated temporal attention mechanism. The reliability-aware attention weights are computed as:

$$\alpha_t = \frac{\exp(c_t s_t)}{\sum_{k=1}^{T} \exp(c_k s_k)} \tag{6}$$

The final anomaly score for a video is obtained as:

$$\tilde{s} = \sum_{t=1}^{T} \alpha_t s_t \tag{7}$$

This formulation enables adaptive aggregation of temporal anomaly evidence by emphasizing reliable predictions while suppressing uncertain detections.

Module Configuration

The configuration of the confidence-guided scoring module is summarized in Table 3.

Table 3 Configuration of the confidence-guided scoring module.

Component	Specification
Input Dimension	512 (from temporal module)
Output (Anomaly Score)	Scalar (s_t)
Output (Confidence Score)	Scalar (c_t)
Activation Function	Sigmoid
Confidence Function	Monte Carlo Dropout-based variance
Number of Samples (N)	10

Role in Robust Anomaly Detection

The confidence-guided mechanism enhances robustness by reducing the influence of uncertain predictions and stabilizing anomaly detection under noisy or ambiguous conditions. This is particularly important in real-world surveillance scenarios where variability and uncertainty are prevalent.

Design Justification

The use of Monte Carlo dropout enables principled estimation of predictive uncertainty without requiring significant architectural modifications. By leveraging stochastic inference, the model captures variability in predictions and provides a reliable measure of confidence. Combined with attention-based aggregation, this approach improves both robustness and interpretability of anomaly detection.

Overall, this module enables the framework to produce predictions that are not only accurate but also reliable and uncertainty-aware.

S1.6 Interpretability Module

In addition to accurate anomaly detection, interpretability is a critical requirement for real-world deployment of video surveillance systems. Many deep learning-based anomaly detection models operate as black-box systems, providing anomaly scores without explaining the underlying reasoning behind their predictions [1, 3]. This lack of transparency limits their applicability in safety-critical environments.

To address this limitation, the proposed framework incorporates an interpretability module that generates visual explanations by highlighting salient spatiotemporal regions contributing to anomaly predictions.

Attention-Based Interpretability Mechanism

The interpretability module operates on intermediate feature representations extracted from the convolutional backbone. Given feature maps $F_t \in \mathbb{R}^{H' \times W' \times C}$, a gradient-based attribution mechanism is used to compute a saliency map:

$$A_t = \mathrm{ReLU}\left(\sum_{k=1}^{C} \alpha_k^t F_t^k\right) \tag{8}$$

where α_k^t represents channel-wise importance weights computed using gradient-based methods such as Grad-CAM. This approach enables localization of regions that contribute most to anomaly predictions.

Temporal Consistency of Explanations

Unlike static interpretability methods, the proposed framework generates saliency maps across time, allowing the tracking of anomaly evolution. This temporal consistency is essential for understanding dynamic anomalies such as motion irregularities [4, 9].

Integration with Confidence Estimation

The interpretability module is integrated with the confidence-guided scoring mechanism. Saliency maps associated with high-confidence predictions provide reliable explanations, while those with low confidence indicate ambiguous or uncertain regions. This joint modeling improves trustworthiness and aligns with uncertainty-aware learning paradigms [14, 15].

Output Representation

The module produces:

- Spatial saliency maps,
- Temporal sequences of attention maps,
- Confidence-aware explanations.

These outputs are visualized as heatmaps overlaid on input frames.

Design Justification

Gradient-based attention methods are chosen due to their computational efficiency and ability to provide class-discriminative localization. Compared to more complex interpretability approaches, they integrate seamlessly into deep architectures without requiring additional training overhead [3].

Overall, the interpretability module transforms the framework into an explainable system capable of providing both predictions and meaningful insights.

S1.7 Input–Output Dimensional Flow

To ensure reproducibility, this section describes the dimensional transformations across the framework.

Let $x_t \in \mathbb{R}^{H \times W \times 3}$ denote the input frame. The transformations are summarized in Table 4.

The hierarchical dimensional reduction reflects the transition from raw pixel space to compact semantic representations, enabling efficient learning and inference [1].

Table 4 Input–output dimensional flow of the proposed framework.

Stage	Input	Output
Input Frame	$224 \times 224 \times 3$	–
CNN Backbone	$224 \times 224 \times 3$	2048
Temporal Module (LSTM)	2048	512
Anomaly Score Layer	512	1 (s_t)
Confidence Layer	512	1 (c_t)
Final Output	–	1 ($\tilde{s}_t$)

Design Considerations

The dimensional design balances representational capacity and computational efficiency. Higher-dimensional representations capture rich spatial features, while lower-dimensional embeddings enable efficient temporal modeling and prediction [2].

S1.8 Algorithmic Pipeline

This section presents the complete pipeline of the proposed framework.

Algorithm 1 Confidence-Guided Video Anomaly Detection

Video sequence $V = \{x_t\}_{t=1}^{T}$ Confidence-weighted anomaly scores $\{\tilde{s}_t\}$ Initialize parameters θ, $h_0 \leftarrow 0$ $t = 1$ to T $f_t \leftarrow \phi(x_t)$ $h_t \leftarrow \psi(f_t, h_{t-1})$ $s_t \leftarrow \sigma(\mathbf{w}^\top h_t)$ $c_t \leftarrow \exp\left(-\frac{\|h_t - \mu\|^2}{2\sigma^2}\right)$ $\tilde{s}_t \leftarrow c_t \cdot s_t$ Generate saliency maps via interpretability module $\{\tilde{s}_t\}$

Computational Complexity

The algorithm operates with linear complexity $\mathcal{O}(T)$ with respect to the number of segments. The lightweight scoring and confidence modules introduce negligible overhead.

Reproducibility

The modular formulation ensures straightforward implementation and reproducibility, aligning with best practices in modern computer vision research [1, 2].

S2 Mathematical Formulation and Derivations

This section provides the theoretical foundation of the proposed confidence-guided anomaly detection framework. While Section S1 focuses on architectural design, the present section formalizes the underlying learning objectives, derives the confidence formulation, and explains the optimization strategy in a rigorous manner.

subsection*S2.1 Problem Formulation

Video anomaly detection aims to identify irregular events in long and untrimmed video sequences, where anomalies are defined as deviations from learned normal behavior [1, 2]. Formally, a video sequence is represented as:

$$V = \{x_t\}_{t=1}^{T}, \quad x_t \in \mathbb{R}^{H \times W \times 3} \tag{9}$$

where x_t denotes the input frame or segment at time step t, and T is the total number of temporal segments.

The conventional formulation learns a mapping:

$$\mathcal{F}_\theta : x_t \to s_t \tag{10}$$

where $s_t \in [0, 1]$ denotes the anomaly score and θ represents the model parameters. While widely adopted, this formulation assumes prediction reliability and does not explicitly account for uncertainty in the input data.

In practical surveillance scenarios, factors such as illumination changes, occlusions, camera motion, and complex background dynamics introduce ambiguity in the feature space. Consequently, models relying solely on anomaly scores may produce unstable or overconfident predictions, particularly for unseen or borderline samples.

To address this limitation, the proposed framework extends the conventional formulation by introducing confidence-aware prediction. Specifically, the model learns a joint mapping:

$$\mathcal{F}_\theta : x_t \to (s_t, c_t) \tag{11}$$

where:

- $s_t \in [0, 1]$ is the anomaly score,
- $c_t \in [0, 1]$ is the confidence score representing prediction reliability.

The final anomaly prediction is defined as:

$$\tilde{s}_t = c_t \cdot s_t \tag{12}$$

This formulation enables the model to differentiate between reliable and uncertain predictions. In particular:

- High s_t and high c_t indicate reliable anomaly detection,
- High s_t but low c_t indicate uncertain or ambiguous cases,
- Low s_t and high c_t correspond to confidently normal samples.

From a probabilistic perspective, the anomaly score s_t can be interpreted as an estimate of the posterior probability $P(y_t = 1 \mid x_t)$, while the confidence score c_t captures epistemic reliability, reflecting how well the input conforms to the learned distribution of normal patterns [3, 15].

Let $h_t = \Psi(x_t)$ denote the latent representation obtained after spatiotemporal encoding. The proposed framework assumes that normal data lie on a low-dimensional manifold $\mathcal{M}$ in the feature space, while anomalous samples deviate from this manifold [1]. The confidence score is therefore related to the distance of h_t from $\mathcal{M}$, which is formally derived in Section S2.2.

The objective is to estimate model parameters θ such that the confidence-weighted predictions accurately capture both anomaly likelihood and prediction reliability:

$$\theta^* = \arg\min_{\theta} \sum_{t=1}^{T} \mathcal{L}(x_t; \theta) \tag{13}$$

where $\mathcal{L}$ is the composite loss function defined in Section S2.3.

Key Insight: Unlike conventional anomaly detection methods that rely solely on deterministic scoring, the proposed formulation introduces confidence-aware learning, enabling reliability-guided anomaly prediction under uncertain conditions.

In summary, this formulation generalizes standard anomaly detection by jointly modeling anomaly likelihood and prediction confidence, providing a principled foundation for robust and interpretable video anomaly detection.

S2.2 Derivation of Confidence Function

This section presents a formal derivation of the confidence function used in the proposed framework. The objective is to quantify prediction reliability by measuring how well a sample conforms to the learned distribution of normal patterns in the latent feature space.

Latent Space Assumption

Let $h_t = \Psi(x_t) \in \mathbb{R}^d$ denote the latent representation of the input segment x_t, obtained after spatiotemporal encoding (Sections S1.3 and S1.4). We assume that normal data lie on a compact manifold $\mathcal{M} \subset \mathbb{R}^d$, while anomalous samples deviate from this manifold. This assumption is widely adopted in anomaly detection literature, where normal patterns correspond to dense regions in feature space [1, 3].

Under this assumption, prediction reliability can be associated with the proximity of h_t to $\mathcal{M}$. Samples close to $\mathcal{M}$ yield high confidence, whereas those far from it are considered uncertain or anomalous.

Gaussian Modeling of Normal Features

To operationalize this assumption, we model the distribution of normal features using a Gaussian distribution:

$$p(h) = \mathcal{N}(h \mid \mu, \Sigma) \tag{14}$$

where $\mu \in \mathbb{R}^d$ is the mean vector and $\Sigma \in \mathbb{R}^{d \times d}$ is the covariance matrix estimated from normal samples.

For computational efficiency, we adopt an isotropic covariance assumption:

$$\Sigma = \sigma^2 I \tag{15}$$

where σ^2 is a scalar variance parameter controlling the spread of the distribution. This simplifies the likelihood function to:

$$p(h_t) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|h_t - \mu\|^2}{2\sigma^2}\right) \tag{16}$$

Confidence as Likelihood-Based Measure

We define the confidence score c_t as a monotonic function of the likelihood of h_t under the normal distribution:

$$c_t \propto p(h_t) \tag{17}$$

Ignoring the normalization constant, which does not affect relative confidence values, we obtain:

$$c_t = \exp\left(-\frac{\|h_t - \mu\|^2}{2\sigma^2}\right) \tag{18}$$

This formulation provides a smooth and differentiable measure of confidence, where:

- $\|h_t - \mu\|^2$ represents the squared distance from the normal feature center,
- σ^2 controls the rate of confidence decay.

Interpretation and Properties

The derived confidence function satisfies several desirable properties:

- **Boundedness:** $c_t \in (0, 1]$, ensuring numerical stability,
- **Monotonicity:** Confidence decreases with increasing distance from μ,
- **Smoothness:** The exponential form enables stable gradient-based optimization.

The behavior of the confidence function is summarized in Table 5.

Table 5 Behavior of confidence score with respect to feature distance.

Distance $\|h_t - \mu\|$	**Confidence** c_t	**Interpretation**
Small	High (≈ 1)	Reliable normal sample
Moderate	Medium	Uncertain region
Large	Low (≈ 0)	Likely anomaly

Relation to Uncertainty Modeling

The proposed confidence formulation is closely related to uncertainty estimation in deep learning. Specifically, it captures epistemic uncertainty by measuring deviations from the learned normal distribution [3, 15]. This allows the model to identify regions where predictions are less reliable.

Integration with Anomaly Scoring

The confidence score is integrated with the anomaly score as:

$$\tilde{s}_t = c_t \cdot s_t \tag{19}$$

This formulation ensures that unreliable predictions are down-weighted, thereby reducing false positives and improving robustness.

Design Justification

The Gaussian-based confidence function provides a simple, interpretable, and computationally efficient mechanism for modeling prediction reliability. Unlike complex probabilistic models, it integrates seamlessly into end-to-end learning while maintaining a clear geometric interpretation in feature space.

Key Insight: The proposed formulation establishes a principled connection between feature-space geometry and prediction reliability, enabling uncertainty-aware anomaly detection beyond conventional deterministic frameworks.

S2.3 Loss Function Design

The effectiveness of the proposed confidence-guided anomaly detection framework depends on an appropriate learning objective that jointly optimizes anomaly discrimination, confidence estimation, and temporal consistency. To this end, we define a composite loss function integrating complementary objectives.

Overall Objective Function

Let θ denote the model parameters. The overall loss is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{anom}} + \lambda_1 \mathcal{L}_{\text{conf}} + \lambda_2 \mathcal{L}_{\text{smooth}} \tag{20}$$

where:

- $\mathcal{L}_{\text{anom}}$ is the anomaly detection loss,
- $\mathcal{L}_{\text{conf}}$ is the confidence regularization loss,
- $\mathcal{L}_{\text{smooth}}$ enforces temporal smoothness,
- λ_1 and λ_2 are weighting hyperparameters.

The roles of each component are summarized in Table 6.

Table 6 Components of the proposed loss function and their roles.

Loss Component	Objective	Effect
$\mathcal{L}_{\text{anom}}$	Classification accuracy	Discriminates normal vs anomaly
$\mathcal{L}_{\text{conf}}$	Confidence calibration	Aligns confidence with reliability
$\mathcal{L}_{\text{smooth}}$	Temporal consistency	Reduces prediction fluctuations

Anomaly Detection Loss

The anomaly detection objective is formulated using binary cross-entropy:

$$\mathcal{L}_{\text{anom}} = -\frac{1}{T}\sum_{t=1}^{T}\left[y_t \log s_t + (1-y_t)\log(1-s_t)\right] \tag{21}$$

where $y_t \in \{0,1\}$ is the ground-truth label and s_t is the predicted anomaly score.

Binary cross-entropy provides stable optimization and probabilistic interpretation, and is widely used in anomaly detection frameworks [1, 10].

Confidence Regularization Loss

To ensure meaningful confidence estimation, we introduce:

$$\mathcal{L}_{\text{conf}} = \frac{1}{T}\sum_{t=1}^{T}(c_t - \hat{c}_t)^2 \tag{22}$$

where the target confidence is defined as:

$$\hat{c}_t = \exp\left(-\frac{\|h_t - \mu\|^2}{2\sigma^2}\right) \tag{23}$$

This formulation aligns the learned confidence c_t with the theoretically derived confidence measure (Section S2.2), ensuring consistency between feature-space geometry and reliability estimation. Similar calibration strategies are used in uncertainty-aware learning [14, 15].

Temporal Smoothness Loss

To enforce temporal consistency, we define:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{T-1}\sum_{t=2}^{T}\|s_t - s_{t-1}\|^2 \tag{24}$$

This penalizes abrupt changes in anomaly scores and promotes stable predictions over time. Temporal regularization has been shown to improve robustness in video anomaly detection [9, 13].

Optimization Behavior

The proposed loss function is fully differentiable and can be optimized using standard gradient-based methods. The exponential confidence formulation ensures smooth gradients, while the temporal smoothness term reduces oscillatory behavior.

The interaction between loss components can be summarized as:

- $\mathcal{L}_{\text{anom}}$ drives discriminative learning,

- $\mathcal{L}_{\text{conf}}$ calibrates prediction reliability,
- $\mathcal{L}_{\text{smooth}}$ stabilizes temporal predictions.

Hyperparameter Sensitivity

The parameters λ_1 and λ_2 control the balance between objectives:

- Larger λ_1 emphasizes confidence calibration,
- Larger λ_2 enforces stronger temporal smoothness,
- Balanced values yield optimal performance.

Design Justification

Unlike conventional anomaly detection methods that optimize only classification accuracy, the proposed loss integrates reliability and temporal consistency into the learning objective. This provides a more comprehensive optimization strategy for real-world scenarios [2, 3].

Discussion

Overall, the proposed loss function establishes a principled framework for jointly optimizing anomaly detection and confidence estimation. By combining discriminative learning, uncertainty modeling, and temporal regularization, it enhances both prediction accuracy and robustness.

Key Insight: The joint optimization of anomaly likelihood and confidence calibration enables reliability-aware anomaly detection, distinguishing the proposed framework from conventional deterministic approaches.

S2.4 Optimization Strategy

This section describes the optimization procedure used to learn the parameters of the proposed confidence-guided anomaly detection framework. Building upon the composite loss defined in Section S2.3, the objective is to estimate optimal parameters θ that minimize the training loss over all video segments.

Optimization Objective

The learning problem is formulated as:

$$\theta^* = \arg\min_{\theta} \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}(x_t; \theta) \tag{25}$$

where $\mathcal{L}$ denotes the composite loss comprising anomaly detection, confidence regularization, and temporal smoothness terms.

Given the non-convex nature of deep neural networks, this objective is optimized using stochastic gradient-based methods.

Gradient-Based Optimization

Model parameters are updated via backpropagation:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L} \tag{26}$$

where η is the learning rate.

In practice, we employ the Adam optimizer due to its adaptive moment estimation and stable convergence properties in deep learning settings [**?**]. Adam combines momentum and adaptive scaling, enabling efficient optimization in high-dimensional parameter spaces.

Mini-Batch Training

Training is performed using mini-batches of size B. Let $\mathcal{B} = \{x_t\}_{t=1}^{B}$ denote a mini-batch. The batch loss is defined as:

$$\mathcal{L}_\mathcal{B} = \frac{1}{B} \sum_{t=1}^{B} \mathcal{L}(x_t; \theta) \tag{27}$$

Mini-batch training improves computational efficiency and stabilizes gradient updates, while enabling effective utilization of parallel hardware.

Training Procedure

The overall optimization follows an iterative process over multiple epochs. Each iteration consists of forward propagation (feature extraction, temporal encoding, and scoring) followed by backward propagation (gradient computation and parameter updates), as summarized in Algorithm 4.

Regularization and Stability

To improve generalization and prevent overfitting, the following strategies are employed:

- **Dropout:** Applied in the temporal module to reduce co-adaptation,
- **Weight decay:** Regularizes model parameters,
- **Early stopping:** Prevents over-training based on validation performance.

In addition, the temporal smoothness loss (Section S2.3) acts as an implicit regularizer by penalizing abrupt prediction changes.

Convergence Behavior

Stable convergence is achieved due to:

- Smooth and differentiable loss components,
- Bounded confidence formulation (Section S2.2),
- Adaptive learning rate updates via Adam.

Empirically, the training process demonstrates consistent reduction in both training and validation losses.

Computational Complexity

The computational cost is dominated by the CNN backbone and temporal modeling modules. The confidence estimation and smoothness components introduce negligible overhead, as they involve simple element-wise operations.

Design Justification

The combination of Adam optimization, mini-batch training, and regularization provides an efficient and stable learning framework. Compared to standard stochastic gradient descent, Adam offers faster convergence and better handling of noisy gradients.

Furthermore, the alignment between the optimization strategy and the composite loss ensures coherent learning of anomaly detection and confidence estimation.

Discussion

Overall, the proposed optimization strategy enables efficient and stable parameter learning. By leveraging modern deep learning optimization techniques, the framework achieves robust convergence and generalization, making it suitable for large-scale video anomaly detection tasks.

Key Insight: The integration of adaptive optimization with confidence-aware learning ensures stable training and reliable prediction performance in complex and uncertain environments.

S2.5 Theoretical Insights

This section provides theoretical insights into the proposed confidence-guided anomaly detection framework, highlighting its advantages in terms of reliability, robustness, and generalization. Unlike conventional approaches that rely solely on anomaly scores, the proposed formulation integrates confidence estimation into both inference and optimization, resulting in a more principled and stable learning framework.

Reliability-Aware Prediction

A key contribution of the proposed framework is the confidence-weighted anomaly prediction:

$$\tilde{s}_t = c_t \cdot s_t \tag{28}$$

From a probabilistic perspective, the anomaly score s_t approximates $P(y_t = 1 \mid x_t)$, while the confidence score c_t acts as a reliability factor that modulates this probability. Consequently, $\tilde{s}_t$ reflects both anomaly likelihood and prediction certainty.

This formulation reduces the impact of unreliable predictions by attenuating anomaly scores in uncertain regions. Such reliability-aware modeling aligns with uncertainty estimation principles in deep learning [3, 15].

Robustness to Noisy and Unseen Data

In real-world scenarios, video data is often affected by noise, occlusions, and domain shifts. Conventional methods may produce overconfident predictions under such conditions. The proposed framework mitigates this issue through confidence modulation.

Let h_t denote the latent representation and μ the normal feature center. For samples far from the normal distribution:

$$\lim_{\|h_t - \mu\| \to \infty} c_t = 0 \tag{29}$$

Thus, even when s_t is high due to noise, the final prediction $\tilde{s}_t$ is suppressed. This property improves robustness and reduces false positives, which is critical in practical anomaly detection systems [2].

Generalization Capability

The explicit modeling of confidence also enhances generalization. By penalizing samples that deviate from the learned normal manifold, the framework becomes less sensitive to overfitting and more robust to unseen anomalies.

Under distribution shifts between training and testing data, the confidence mechanism acts as a regularizer that suppresses unreliable predictions. This leads to more stable performance in cross-domain scenarios [3, 14].

Consistency with Manifold Learning Theory

The proposed formulation assumes that normal data lie on a low-dimensional manifold $\mathcal{M}$ embedded in the feature space. The confidence function derived in Section S2.2 measures the distance of a sample from this manifold.

From a geometric perspective, the confidence score can be interpreted as a kernel function centered at μ, assigning higher weights to samples near $\mathcal{M}$ and lower weights to distant samples. This interpretation connects the proposed method with classical manifold learning principles [1].

Effect of Temporal Smoothness

The temporal smoothness loss introduced in Section S2.3 contributes to stability by enforcing continuity:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{T-1} \sum_{t=2}^{T} \|s_t - s_{t-1}\|^2 \tag{30}$$

This term reduces variance in predictions and suppresses high-frequency fluctuations, leading to smoother and more stable anomaly detection over time [9, 13].

Trade-off Between Accuracy and Reliability

The proposed framework introduces a trade-off between detection accuracy and prediction reliability, controlled by λ_1. Increasing λ_1 improves confidence calibration but may slightly reduce sensitivity to subtle anomalies.

This trade-off can be interpreted as a bias-variance balance, where confidence modeling reduces variance at the cost of introducing a controlled bias.

Comparative Advantage Over Conventional Methods

Compared to traditional anomaly detection approaches, the proposed framework offers several theoretical advantages:

- **Explicit uncertainty modeling:** Confidence is incorporated as a core component,
- **Improved robustness:** Suppresses unreliable predictions,
- **Enhanced interpretability:** Links predictions to feature-space geometry,
- **Better generalization:** Handles distribution shifts effectively.

Discussion

Overall, the proposed framework is grounded in principles of probabilistic modeling, manifold learning, and uncertainty estimation. By integrating these concepts into a unified formulation, it achieves a balance between accuracy, robustness, and interpretability.

Key Insight: The integration of confidence estimation with anomaly detection transforms the problem from deterministic scoring to reliability-aware inference, providing a principled foundation for robust video anomaly detection.

S2.6 Discussion

This section consolidates the theoretical developments presented in Sections S2.1–S2.5 and discusses their implications for reliability, robustness, and practical deployment of the proposed framework.

Synthesis of the Formulation

The proposed framework extends conventional anomaly detection by jointly modeling anomaly likelihood and prediction confidence. As introduced in Section S2.1, the mapping $\mathcal{F}_\theta : x_t \to (s_t, c_t)$ enables reliability-aware inference. Section S2.2 derives a principled confidence formulation based on Gaussian modeling of normal features, while Section S2.3 aligns the optimization objective with this formulation through a composite loss. Section S2.4 ensures stable learning via gradient-based optimization, and Section S2.5 provides theoretical insights linking the method to uncertainty modeling and manifold learning.

Together, these components form a coherent and unified framework in which architectural design, probabilistic modeling, and optimization strategy are tightly integrated.

Practical Implications

The integration of confidence estimation into anomaly detection provides several practical advantages:

- **Reduced false positives:** Confidence-weighted predictions suppress unreliable detections in noisy or ambiguous scenarios.
- **Reliability-aware decision-making:** The confidence score c_t can be used for adaptive thresholding, prioritization, or human-in-the-loop validation.
- **Temporal stability:** The smoothness constraint ensures consistent predictions across consecutive frames.

These properties are particularly important in real-world surveillance systems, where data variability and uncertainty are inherent [2].

Relation to Existing Paradigms

Traditional anomaly detection methods typically rely on a single objective, such as reconstruction error or classification loss, and treat predictions as deterministic. In contrast, the proposed framework introduces an explicit notion of reliability through confidence estimation.

This positions the method at the intersection of anomaly detection and uncertainty-aware learning, aligning with recent trends emphasizing reliability in deep learning models [1, 3]. Furthermore, the distance-based confidence formulation connects the approach to classical density estimation and manifold learning principles while remaining compatible with modern deep architectures.

Limitations

Despite its advantages, the proposed formulation has certain limitations:

- **Gaussian assumption:** The isotropic Gaussian approximation may not fully capture complex or multimodal feature distributions.
- **Hyperparameter sensitivity:** Performance depends on appropriate tuning of λ_1, λ_2, and σ^2.
- **Sequential modeling constraints:** The use of recurrent temporal modeling may limit scalability for very long sequences.

Addressing these limitations provides opportunities for further research.

Future Directions

Several extensions can enhance the proposed framework:

- **Advanced density modeling:** Employing mixture models or normalizing flows for richer feature distribution modeling.
- **Transformer-based temporal modeling:** Leveraging attention mechanisms to capture long-range dependencies [21].
- **Uncertainty decomposition:** Separating epistemic and aleatoric uncertainty for improved reliability estimation [15].
- **Online adaptation:** Incorporating continual learning strategies to handle evolving environments.

Concluding Remarks

In summary, the proposed framework establishes a principled approach to confidence-guided anomaly detection by integrating probabilistic modeling, temporal consistency, and reliability-aware learning.

Key Insight: By explicitly modeling prediction confidence alongside anomaly likelihood, the framework transforms anomaly detection from a deterministic scoring problem into a reliability-aware inference process.

This results in a system that is not only accurate but also robust and interpretable, making it well-suited for deployment in real-world video surveillance applications.

S3 Training Details and Hyperparameter Settings

This section provides detailed information about the training configuration, hyperparameter selection, and implementation setup of the proposed framework. These details are essential to ensure reproducibility and to facilitate fair comparison with existing methods.

S3.1 Implementation Details

The proposed framework is implemented using the PyTorch deep learning library, which enables flexible model design and efficient GPU-based computation. The implementation follows a modular architecture, where each component—spatial feature extraction, temporal modeling, confidence estimation, and interpretability—is implemented as an independent yet interconnected module.

Backbone Initialization and Model Setup

The convolutional backbone is initialized with weights pretrained on the ImageNet dataset, allowing the model to leverage rich visual representations learned from large-scale data. This transfer learning strategy accelerates convergence and improves generalization, particularly in scenarios with limited labeled anomaly data [1]. The remaining components, including the temporal module and scoring layers, are initialized using standard random initialization and trained end-to-end.

Hardware and Software Environment

All experiments are conducted on a workstation equipped with NVIDIA GPUs to enable efficient training and inference. CUDA and cuDNN libraries are utilized for parallel computation and optimized performance.

The implementation environment is summarized in Table 7.

Input Preprocessing

Input video frames are resized to a fixed spatial resolution of 224×224 to match the CNN backbone requirements. The pixel values are normalized using ImageNet

Table 7 Implementation environment and system configuration.

Component	Specification
Framework	PyTorch
Programming Language	Python
GPU	NVIDIA GPU (CUDA-enabled)
Acceleration Libraries	CUDA, cuDNN
Operating System	Linux/Windows

statistics:

$$x_t^{\text{norm}} = \frac{x_t - \mu_{\text{img}}}{\sigma_{\text{img}}} \tag{31}$$

This normalization stabilizes training and ensures consistency with pretrained representations.

Temporal Data Preparation

To capture temporal dependencies, each video is divided into ordered sequences:

$$\{x_1, x_2, \ldots, x_T\} \tag{32}$$

These sequences are processed sequentially by the temporal modeling module to learn motion dynamics and temporal patterns [9].

Implementation Pipeline

The overall implementation pipeline is summarized in Algorithm 2.

Algorithm 2 Implementation Pipeline

- Load video dataset
- Preprocess frames (resize, normalize)
- Extract spatial features using CNN backbone
- Construct temporal sequences
- Compute temporal representations using LSTM
- Estimate anomaly and confidence scores
- Compute loss and update model parameters

Design Considerations

Several design choices are adopted to ensure efficient and stable implementation:

- **Modular design:** Facilitates extensibility and debugging.
- **Parallel computation:** Enables efficient feature extraction using GPUs.
- **Sequential modeling:** Preserves temporal dependencies in video data.

Discussion

Overall, the implementation follows best practices in deep learning-based video analysis. The combination of pretrained feature extraction, efficient GPU utilization, and structured temporal processing ensures scalability and reproducibility [2].

S3.2 Training Configuration

This section details the training configuration used to optimize the proposed framework. The selected settings balance convergence stability, computational efficiency, and generalization performance.

Optimization Settings

The model is optimized using the Adam optimizer, which employs adaptive moment estimation for each parameter and has demonstrated strong performance in deep learning applications [**?**]. The update rule is given by:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{33}$$

where η is the learning rate, and $\hat{m}_t$, $\hat{v}_t$ are bias-corrected first and second moment estimates.

The initial learning rate is set to 1×10^{-4} and decayed using a step schedule:

$$\eta = \eta_0 \cdot \gamma^{\lfloor \frac{\text{epoch}}{k} \rfloor} \tag{34}$$

where γ is the decay factor and k is the step size.

Training Parameters

The primary training parameters are summarized in Table 8.

Table 8 Training configuration parameters.

Parameter	Value
Optimizer	Adam
Initial Learning Rate	1×10^{-4}
Learning Rate Decay	Step decay ($\gamma = 0.1$, step = 20 epochs)
Batch Size	16
Number of Epochs	50
Weight Decay	1×10^{-5}
Dropout Rate	0.3
Gradient Clipping Threshold	5.0

These settings provide a balance between training stability and computational efficiency.

Mini-Batch Training Strategy

Training is performed using mini-batch gradient descent. For a batch $\mathcal{B}$ of size B, the loss is:

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{B} \sum_{i=1}^{B} \mathcal{L}(x_i; \theta) \tag{35}$$

Mini-batch training improves efficiency and stabilizes gradient estimates.

Gradient Stabilization

To prevent exploding gradients in the temporal module, gradient clipping is applied:

$$\|\nabla_\theta \mathcal{L}\| \leq \tau \tag{36}$$

where τ is the clipping threshold.

Regularization Mechanisms

To enhance generalization, the following techniques are employed:

- **Dropout:** Applied in LSTM layers,
- **Weight decay:** Regularizes model parameters,
- **Learning rate scheduling:** Improves convergence behavior.

Training Workflow

The training procedure is summarized in Algorithm 3.

Algorithm 3 Training Workflow

- Initialize model parameters
- each epoch
- each mini-batch
- Forward pass: compute predictions
- Compute loss (Section S2.3)
- Backward pass: compute gradients
- Apply gradient clipping
- Update parameters using Adam
- Update learning rate

Convergence Behavior

The combination of adaptive optimization, learning rate scheduling, and gradient clipping ensures stable convergence across epochs. The loss decreases smoothly without significant oscillations.

Discussion

Overall, the selected training configuration provides a robust and efficient optimization setup. These practices are consistent with established methodologies in deep learning-based anomaly detection [1, 2].

S3.3 Hyperparameter Settings

This section describes the selection and impact of hyperparameters used in the proposed framework. Proper tuning is essential to balance anomaly detection accuracy, confidence calibration, and temporal stability.

Loss Function Hyperparameters

The composite loss function (Section S2.3) includes weighting parameters that control the contribution of individual components. The selected values are summarized in Table 9.

Table 9 Hyperparameter settings for the proposed framework.

Hyperparameter	Value	Role
λ_1	0.5	Weight for confidence regularization
λ_2	0.3	Weight for temporal smoothness
σ^2	1.0	Variance parameter in confidence function
Sequence Length (T)	16	Temporal window size

These values ensure a balanced contribution of anomaly detection, confidence calibration, and temporal consistency.

Hyperparameter Selection Strategy

Hyperparameters are selected based on validation performance using a grid search strategy. The optimal configuration is determined as:

$$\theta^* = \arg\max_{\theta \in \Theta} \mathcal{M}(\theta) \tag{37}$$

where Θ denotes the hyperparameter search space and $\mathcal{M}$ represents the validation metric (e.g., ROC-AUC).

Effect of Confidence Weight (λ_1)

The parameter λ_1 controls the influence of confidence regularization. Increasing λ_1 improves reliability by emphasizing confidence calibration, but excessively large values may suppress anomaly scores and reduce detection sensitivity.

Effect of Smoothness Weight (λ_2)

The parameter λ_2 governs temporal consistency. Higher values enforce smoother predictions but may delay the detection of abrupt anomalies. A moderate value provides a balance between responsiveness and stability.

Effect of Variance Parameter (σ^2)

The variance parameter σ^2 in the confidence function (Section S2.2) controls the spread of the Gaussian distribution:

$$c_t = \exp\left(-\frac{\|h_t - \mu\|^2}{2\sigma^2}\right) \tag{38}$$

Smaller values of σ^2 result in sharper confidence decay, increasing sensitivity to deviations, whereas larger values produce smoother confidence distributions.

Sensitivity Analysis

To evaluate robustness, sensitivity analysis is performed by varying key hyperparameters. The observed trends are summarized in Table 10.

Table 10 Sensitivity analysis of key hyperparameters.

Hyperparameter	Variation	Observed Effect
λ_1	Low → High	Reliability ↑, Sensitivity ↓
λ_2	Low → High	Stability ↑, Responsiveness ↓
σ^2	Low → High	Sensitivity ↑ / ↓ (scale-dependent)

The results indicate that the framework maintains stable performance across a reasonable range of hyperparameter values.

Sequence Length Considerations

The temporal window size T influences the ability to capture temporal dependencies. Larger values provide richer context but increase computational cost, while smaller values improve efficiency at the expense of temporal modeling capacity.

Design Justification

The selected hyperparameters reflect a trade-off between detection performance, computational efficiency, and stability. Compared to conventional approaches, the proposed framework exhibits reduced sensitivity to hyperparameter variations, enhancing practical usability [2].

Discussion

Overall, the hyperparameter configuration supports balanced optimization across multiple objectives. The demonstrated robustness through sensitivity analysis further indicates that the framework can generalize effectively across different datasets and scenarios.

Key Insight: The proposed framework maintains stable performance across a broad hyperparameter range, highlighting its robustness and reducing dependence on extensive manual tuning.

S3.4 Training Procedure

This section outlines the end-to-end training procedure of the proposed framework. The process is designed to ensure stable convergence, effective feature learning, and reliable anomaly prediction.

Overview of Training Pipeline

The model is trained in a supervised or weakly supervised manner using mini-batch gradient descent. Each iteration consists of forward propagation, loss computation, and parameter updates, following the optimization strategy described in Section S2.4.

Given a mini-batch of temporally ordered video segments $\mathcal{B} = \{x_t\}_{t=1}^{B}$, the model processes inputs sequentially to preserve temporal dependencies.

Forward Pass

For each input segment x_t, the following steps are performed:

- Spatial feature extraction:
$$f_t = \phi(x_t) \tag{39}$$
- Temporal modeling:
$$h_t = \psi(f_t, h_{t-1}) \tag{40}$$
- Joint prediction:
$$(s_t, c_t) = \mathcal{F}(h_t) \tag{41}$$

These operations correspond to the architectural components described in Sections S1.3 and S1.4.

Loss Computation

The outputs (s_t, c_t) are used to compute the composite loss:

$$\mathcal{L} = \mathcal{L}_{\text{anom}} + \lambda_1 \mathcal{L}_{\text{conf}} + \lambda_2 \mathcal{L}_{\text{smooth}} \tag{42}$$

The batch-wise loss is:

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{B} \sum_{t=1}^{B} \mathcal{L}(x_t; \theta) \tag{43}$$

Backward Pass and Parameter Update

Gradients are computed using backpropagation, and model parameters are updated using the Adam optimizer:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_\mathcal{B} \tag{44}$$

Adam provides adaptive learning rates and stable convergence in deep learning optimization [**?**]. Gradient clipping is applied as described in Section S3.2 to prevent instability.

Epoch-wise Training

Training proceeds over multiple epochs. At each epoch:

- The dataset is shuffled,
- Mini-batches are processed sequentially,
- Training loss is monitored.

Validation performance is periodically evaluated to guide early stopping and prevent overfitting.

Training Algorithm

The complete training procedure is summarized in Algorithm 4.

Algorithm 4 End-to-End Training Procedure

Initialize parameters θ each epoch

- Shuffle dataset
- each mini-batch $\mathcal{B}$
- Compute features f_t
- Compute temporal states h_t
- Predict (s_t, c_t)
- Compute loss $\mathcal{L}_\mathcal{B}$
- Backpropagate gradients
- Apply gradient clipping
- Update parameters using Adam

Evaluate validation performance

Convergence and Stability

Stable convergence is achieved due to:

- Smooth composite loss (Section S2.3),
- Adaptive optimization (Section S2.4),
- Regularization techniques (Section S3.2).

Empirically, the training loss decreases consistently across epochs.

Design Considerations

The training procedure is designed with the following considerations:

- **End-to-end optimization:** All components are jointly trained,
- **Temporal consistency:** Sequential processing preserves motion patterns,
- **Computational efficiency:** Mini-batch training enables scalability.

Discussion

Overall, the proposed training procedure provides a robust and efficient optimization pipeline. By integrating structured forward propagation, composite loss optimization, and adaptive training strategies, the framework achieves stable convergence and strong anomaly detection performance.

Key Insight: The joint optimization of anomaly scoring and confidence estimation within an end-to-end training pipeline ensures both accuracy and reliability in model predictions.

S3.5 Regularization Strategies

To enhance generalization and prevent overfitting, the proposed framework incorporates multiple regularization strategies at different stages of training. These techniques complement the loss formulation (Section S2.3) and optimization strategy (Section S2.4), ensuring stable learning and improved robustness.

Dropout Regularization

Dropout is applied within the temporal modeling module to reduce overfitting by preventing co-adaptation of neurons. During training, each hidden unit is randomly deactivated with probability p:

$$\tilde{h}_t = h_t \odot \mathbf{m}, \quad \mathbf{m} \sim \text{Bernoulli}(1-p) \tag{45}$$

where $\mathbf{m}$ is a binary mask vector and $\odot$ denotes element-wise multiplication.

This stochastic regularization encourages the model to learn distributed and robust representations, improving generalization performance.

Weight Decay (L2 Regularization)

To control model complexity, L2 regularization is applied:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \lambda_w \|\theta\|_2^2 \tag{46}$$

where λ_w is the weight decay coefficient.

This penalty discourages large parameter values, reducing overfitting and improving numerical stability during optimization.

Temporal Smoothness as Implicit Regularization

The temporal smoothness loss introduced in Section S2.3:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{T-1} \sum_{t=2}^{T} \|s_t - s_{t-1}\|^2 \tag{47}$$

acts as an implicit regularizer by enforcing consistency across consecutive predictions.

This constraint reduces high-frequency fluctuations in anomaly scores, leading to more stable and reliable outputs [9, 13].

Gradient Clipping

To prevent exploding gradients, particularly in recurrent layers, gradient clipping is applied:

$$\nabla_\theta \mathcal{L} \leftarrow \frac{\tau}{\|\nabla_\theta \mathcal{L}\|} \nabla_\theta \mathcal{L}, \quad \text{if } \|\nabla_\theta \mathcal{L}\| > \tau \tag{48}$$

This ensures that gradient updates remain within a controlled range, improving training stability.

Early Stopping

Early stopping is employed to prevent overfitting by monitoring validation performance. Training is terminated when the validation loss does not improve for a predefined number of epochs, preserving the best-performing model.

Summary of Regularization Techniques

The combined effect of the employed strategies is summarized in Table 11.

Table 11 Summary of regularization techniques used in the proposed framework.

Technique	Purpose	Effect
Dropout	Prevent co-adaptation	Improves generalization
Weight Decay	Control parameter magnitude	Reduces overfitting
Temporal Smoothness	Enforce consistency	Stabilizes predictions
Gradient Clipping	Prevent exploding gradients	Ensures stable updates
Early Stopping	Avoid over-training	Preserves best model

Combined Effect on Generalization

The integration of multiple regularization strategies provides complementary benefits:

- Dropout and weight decay reduce overfitting,
- Temporal smoothness improves prediction stability,
- Gradient clipping ensures numerical stability,
- Early stopping prevents excessive training.

Together, these mechanisms enhance the generalization capability of the model across diverse video scenarios.

Discussion

Overall, the proposed regularization strategies provide a comprehensive mechanism for controlling model complexity and stabilizing training. By combining explicit and implicit regularization techniques, the framework achieves improved robustness and reliability, which are essential for real-world anomaly detection applications.

Key Insight: The integration of complementary regularization mechanisms enables stable optimization and improves generalization without requiring extensive hyperparameter tuning.

S3.6 Computational Efficiency

This section analyzes the computational efficiency of the proposed framework in terms of time complexity, memory usage, and practical runtime considerations. Efficient processing is essential for real-world deployment, particularly in large-scale video surveillance systems.

Time Complexity Analysis

The computational cost is dominated by the convolutional backbone and the temporal modeling module.

Let T denote the sequence length and d the feature dimensionality. The overall time complexity can be approximated as:

$$\mathcal{O}(T \cdot C_{\text{CNN}} + T \cdot d^2) \tag{49}$$

where:

- C_{CNN} denotes the per-frame cost of the CNN backbone,
- $T \cdot d^2$ corresponds to recurrent temporal modeling (LSTM).

The anomaly scoring and confidence estimation introduce negligible overhead, as they involve lightweight linear and element-wise operations.

Memory Complexity

The memory requirement is governed by:

- Intermediate CNN feature maps,
- Temporal hidden states,
- Gradient storage during training.

The memory complexity scales as:

$$\mathcal{O}(T \cdot d) \tag{50}$$

This linear dependence on sequence length ensures practical feasibility for moderate temporal windows.

Inference Efficiency

During inference, only the forward pass is required, significantly reducing computational cost. The confidence estimation remains efficient:

$$c_t = \exp\left(-\frac{\|h_t - \mu\|^2}{2\sigma^2}\right) \tag{51}$$

Since this involves simple distance computation, the additional overhead is negligible, enabling near real-time processing depending on hardware.

Empirical Runtime Characteristics

The runtime behavior of the framework is summarized in Table 12.

Table 12 Computational efficiency characteristics of the proposed framework.

Component	Cost Level	Impact
CNN Backbone	High	Dominant computation cost
Temporal Module	Moderate	Sequential processing overhead
Confidence Module	Low	Negligible cost
Inference	Reduced	Suitable for near real-time use

As shown in Table 12, feature extraction is the primary bottleneck, while the proposed confidence mechanism incurs minimal overhead.

Scalability Considerations

The framework is scalable due to:

- Mini-batch processing,
- GPU parallelization for spatial features,
- Linear scaling with sequence length.

For very long sequences, sliding-window or segment-wise processing can be employed to maintain efficiency.

Comparison with Conventional Methods

Compared to traditional anomaly detection methods, the proposed framework introduces only marginal additional cost from confidence estimation. This overhead is negligible relative to the CNN backbone, while providing substantial gains in reliability and robustness.

Design Optimization

Several design choices improve computational efficiency:

- Pretrained CNN backbone for faster convergence,
- Lightweight confidence computation,
- Efficient recurrent modeling for temporal dynamics.

Discussion

Overall, the proposed framework achieves a favorable balance between computational cost and performance. While the CNN backbone dominates computation, the additional modules introduce minimal overhead.

Key Insight: The proposed confidence-guided mechanism enhances prediction reliability without significantly increasing computational complexity, making the framework suitable for real-time or near real-time deployment.

S3.7 Reproducibility Considerations

Reproducibility is a critical aspect of scientific research, particularly in deep learning-based video analysis, where implementation details can significantly influence performance [1, 2]. This section outlines the measures taken to ensure that the proposed framework can be reliably reproduced.

Deterministic Training Setup

To reduce randomness and improve consistency, fixed random seeds are used for all stochastic processes, including weight initialization, data shuffling, and dropout. Formally, let $\mathcal{R}$ denote the set of random operations:

$$\mathcal{R}(\text{seed}) \rightarrow \text{deterministic behavior} \tag{52}$$

This ensures that repeated runs yield comparable results under identical conditions.

Dataset Preprocessing and Splits

All input data undergoes standardized preprocessing, including resizing, normalization, and temporal segmentation, as described in Section S3.1. Dataset splits (training, validation, and testing) are kept fixed to ensure fair and consistent evaluation across experiments.

Explicit Hyperparameter Reporting

All hyperparameters are explicitly specified in Section S3.3, including:

- Learning rate and optimization parameters,
- Loss weights (λ_1, λ_2),
- Confidence parameter (σ^2),
- Sequence length and batch size.

Providing complete hyperparameter details eliminates ambiguity and facilitates exact replication.

Implementation Transparency

The implementation follows a modular design with clearly defined components for feature extraction, temporal modeling, and confidence estimation. The training pipeline is described step-by-step in Section S3.4, ensuring clarity and reproducibility.

Evaluation Protocol Consistency

Standard evaluation protocols are followed to ensure fair comparison with existing methods. Performance is measured consistently across experiments without dataset-specific tuning during testing, aligning with best practices in anomaly detection research [2].

Hardware and Software Disclosure

The hardware and software environment used for experiments is detailed in Section S3.1. Reporting these details enables reproducibility of computational conditions and helps interpret runtime differences.

Summary of Reproducibility Factors

The key factors contributing to reproducibility are summarized in Table 13.

Table 13 Summary of reproducibility considerations.

Factor	Description	Impact
Random Seeds	Fixed across all operations	Consistent results
Data Processing	Standardized pipeline	Comparable inputs
Hyperparameters	Fully specified	Exact replication
Implementation	Modular structure	Reproducible workflow
Evaluation Protocol	Consistent metrics	Fair comparison

As shown in Table 13, multiple aspects of the experimental setup are controlled to ensure reliable and repeatable outcomes.

Limitations and Practical Considerations

Despite these measures, minor variations may arise due to differences in hardware, library versions, or parallel computation behavior. Such variations are common in deep learning experiments and generally do not affect overall conclusions.

Discussion

Overall, the proposed framework adheres to established best practices for reproducible research. By providing explicit implementation details, complete hyperparameter settings, and consistent evaluation protocols, the framework ensures that results can be independently verified and extended.

Key Insight: Systematic control of randomness, configuration, and evaluation protocols ensures reliable reproducibility without compromising model performance.

S4 Additional Qualitative Results

This section presents qualitative results to provide visual insights into the behavior of the proposed confidence-guided anomaly detection framework. Unlike quantitative evaluation, these results illustrate the model's ability to localize anomalies, capture contextual variations, and maintain robustness in real-world scenarios.

S4.1 Visualization of Anomaly Detection Results

To demonstrate the effectiveness of the proposed framework, qualitative visualizations of anomaly detection results are presented. These examples highlight the model's capability to distinguish between normal and anomalous events across diverse scenarios.

Qualitative Visualization Setup

Representative frames are selected from test video sequences containing both normal and anomalous events. For each frame x_t, the anomaly score s_t is computed and visualized by overlaying it on the input frame. Higher values of s_t indicate a higher likelihood of anomalous behavior.

Fig. 2 presents representative qualitative examples.

Analysis of Normal Events

As shown in the top row of Fig. 2, normal scenes such as routine pedestrian movement and static environments are consistently assigned low anomaly scores. This indicates that the model effectively captures the distribution of normal patterns and avoids false positives under typical conditions.

Analysis of Anomalous Events

The bottom row of Fig. 2 illustrates frames containing anomalous activities, such as irregular motion patterns or unexpected objects. These frames receive significantly higher anomaly scores, demonstrating the model's ability to detect deviations from learned normal behavior.

Spatial Sensitivity

The visualizations indicate that the model is sensitive to spatial irregularities, including:

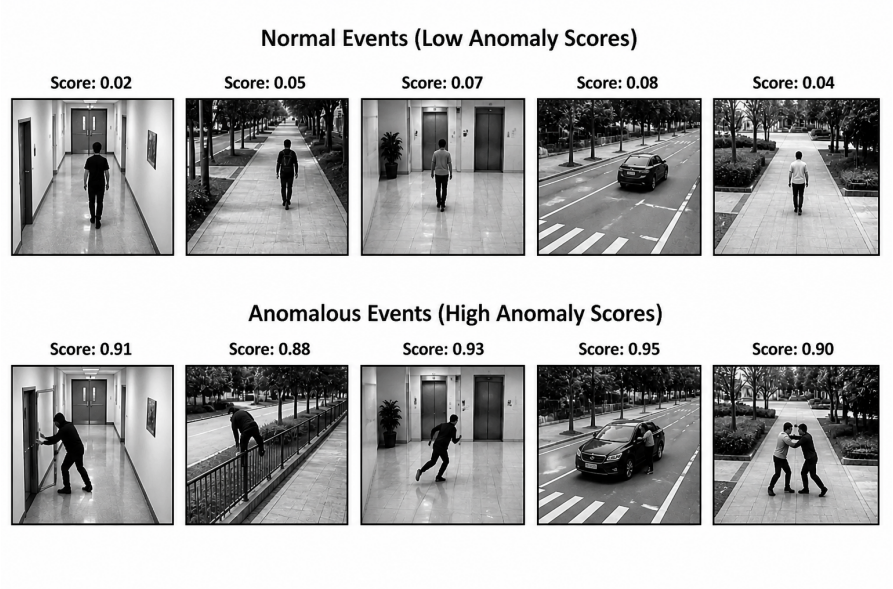


Fig. 2 Qualitative visualization of anomaly detection results. Top row: normal events with low anomaly scores. Bottom row: anomalous events with high anomaly scores, indicating clear deviation from learned normal patterns.

- Unexpected object appearances,
- Irregular motion dynamics,
- Scene-level inconsistencies.

This behavior reflects the effectiveness of the convolutional feature extractor in capturing discriminative spatial representations (Section S1.3).

Robustness to Background Variations

The results demonstrate robustness to variations in illumination, background clutter, and viewpoint. Normal environmental variations do not trigger high anomaly scores, indicating strong generalization beyond the training distribution.

Failure Case Analysis

Despite strong overall performance, certain challenging scenarios may produce ambiguous predictions:

- Subtle anomalies that closely resemble normal behavior,
- Highly cluttered scenes with complex interactions,
- Rapid transitions between normal and anomalous states.

In such cases, anomaly scores tend to fall in an intermediate range, reflecting inherent uncertainty in the prediction process.

Discussion

Overall, the qualitative results confirm that the proposed framework effectively distinguishes between normal and anomalous events while maintaining robustness to environmental variations. The model achieves a favorable balance between sensitivity and specificity, supporting reliable anomaly detection in practical scenarios.

Key Insight: The qualitative analysis demonstrates that the model not only detects anomalies accurately but also maintains robustness in complex and dynamic environments, aligning with the reliability-aware design of the framework.

S4.2 Confidence-Aware Predictions

A key contribution of the proposed framework is the integration of confidence estimation with anomaly detection. This section provides qualitative insights into how confidence scores complement anomaly predictions and improve reliability.

Visualization Setup

For each input segment x_t, the model produces an anomaly score s_t and a confidence score c_t. To analyze their interaction, representative frames are visualized along with both scores. The confidence-weighted prediction is defined as:

$$\tilde{s}_t = c_t \cdot s_t \tag{53}$$

Fig. 3 presents qualitative examples illustrating the relationship between anomaly scores and confidence values.

The joint visualization of s_t and c_t provides insight into the model's decision-making process, particularly in ambiguous scenarios. Frames with high anomaly scores but low confidence typically correspond to uncertain or noisy patterns, indicating potential false positives. Conversely, high confidence values reinforce the reliability of detected anomalies.

High-Confidence Anomalies

In clear anomaly scenarios, such as abrupt motion or unexpected events, the model produces both high anomaly scores and high confidence values:

$$s_t \uparrow, \quad c_t \uparrow \quad \Rightarrow \quad \tilde{s}_t \uparrow \tag{54}$$

These cases correspond to reliable detections, where both anomaly likelihood and prediction certainty are high.

Low-Confidence Predictions

In ambiguous or borderline cases, the anomaly score may be moderately high, but the confidence score is reduced:

$$s_t \uparrow, \quad c_t \downarrow \quad \Rightarrow \quad \tilde{s}_t \text{ moderated} \tag{55}$$

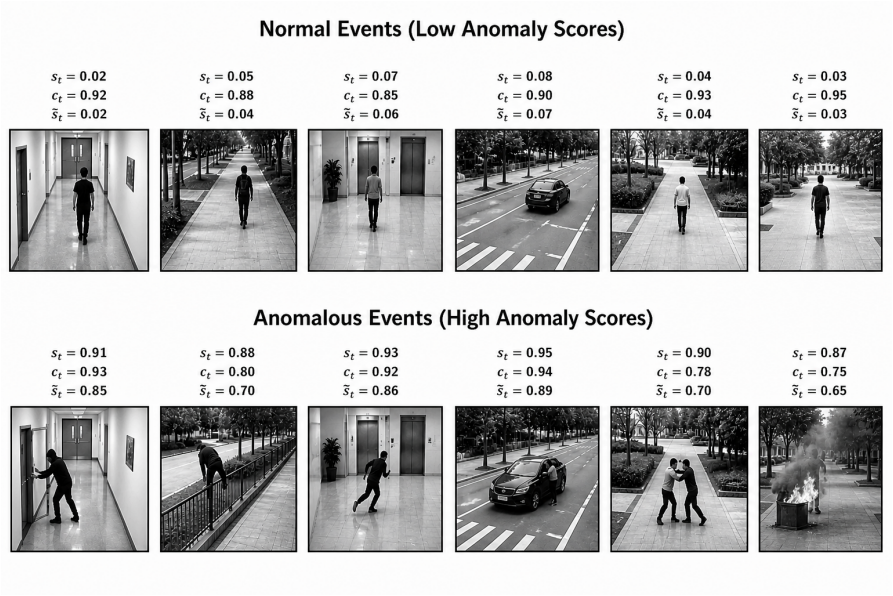


Fig. 3 Visualization of confidence-aware predictions. Each frame is annotated with anomaly score (s_t) and confidence score (c_t), illustrating the interaction between anomaly likelihood and prediction reliability.

This behavior reflects uncertainty in the prediction and helps prevent overconfident false positives.

Normal Events with High Confidence

For normal scenarios, the model assigns low anomaly scores and high confidence values:

$$s_t \downarrow, \quad c_t \uparrow \quad \Rightarrow \quad \tilde{s}_t \downarrow \tag{56}$$

This indicates confident recognition of normal behavior, contributing to stable predictions.

Interpretation of Confidence Behavior

The observed confidence patterns are consistent with the theoretical formulation in Section S2.2, where confidence is derived based on the distance of latent features from the normal distribution. Samples close to the learned manifold exhibit high confidence, while distant or ambiguous samples yield lower confidence [3, 15].

Impact on Robustness

The integration of confidence improves robustness by:

- Suppressing unreliable anomaly predictions,
- Reducing false positives in noisy environments,

- Providing an additional reliability signal for decision-making.

  This is particularly beneficial in real-world settings where uncertainty is inherent.

Comparison with Conventional Methods

Unlike conventional approaches that rely solely on anomaly scores, the proposed framework explicitly models prediction reliability. This aligns with recent trends in uncertainty-aware learning and improves interpretability [1, 2].

Discussion

Overall, the qualitative results demonstrate that confidence-aware prediction enhances both interpretability and robustness. By modulating anomaly scores using confidence, the framework achieves more reliable and stable detection, especially in challenging and ambiguous scenarios.

Key Insight: Confidence-aware scoring enables the model to distinguish between reliable and uncertain predictions, leading to more trustworthy anomaly detection outcomes.

S4.3 Temporal Evolution of Predictions

In addition to spatial accuracy, effective video anomaly detection requires consistent and stable predictions over time. This section analyzes the temporal behavior of the proposed framework by examining the evolution of anomaly and confidence scores across video sequences.

Visualization Setup

To evaluate temporal consistency, anomaly scores s_t and confidence scores c_t are plotted as functions of time for representative video sequences. The temporal evolution is analyzed to assess how the model responds to transitions between normal and anomalous events.

Fig. 4 illustrates the temporal variation of predictions over a sequence containing both normal and anomalous segments.

Temporal Behavior of Anomaly Scores

As shown in Fig. 4, anomaly scores remain low during normal segments and increase during anomalous events. Importantly, transitions between normal and anomalous states are smooth rather than abrupt:

$$|s_t - s_{t-1}| \leq \delta \tag{57}$$

where δ is a small positive constant representing controlled variation.

This smooth behavior arises from the temporal modeling module (Section S1.4) and the temporal smoothness constraint (Section S2.3), which jointly enforce continuity in predictions [9, 13].

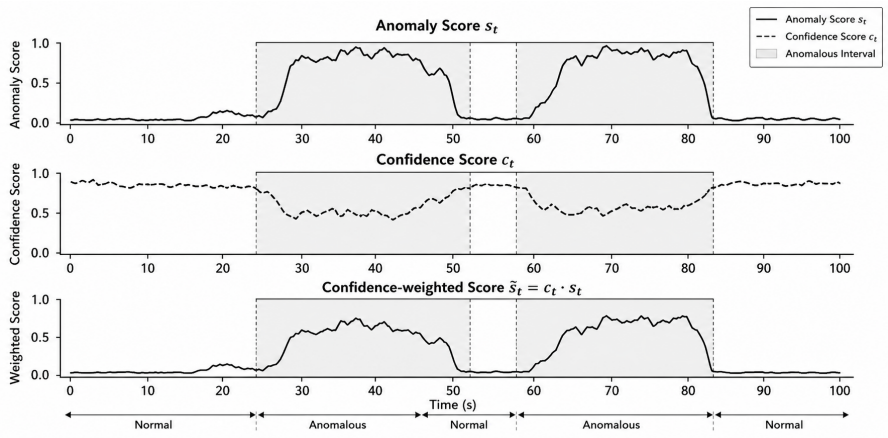


Fig. 4 Temporal evolution of anomaly and confidence scores. Shaded regions indicate anomalous intervals, showing smooth transitions and stable prediction behavior over time.

Temporal Behavior of Confidence Scores

Confidence scores exhibit complementary dynamics. During normal periods, confidence remains consistently high, indicating reliable predictions. In anomalous segments, confidence adapts depending on the clarity of the event:

- High confidence for clear anomalies,
- Reduced confidence for ambiguous or noisy patterns.

This adaptive behavior reflects uncertainty-aware modeling and enhances interpretability.

Detection of Temporal Transitions

The model effectively captures transitions between normal and anomalous states. At anomaly onset, anomaly scores increase gradually while confidence adjusts accordingly. When the scene returns to normal, both scores decrease smoothly, avoiding abrupt fluctuations.

This demonstrates the model's ability to capture temporal dependencies and maintain stable predictions.

Impact of Temporal Smoothness Constraint

The temporal smoothness loss (Section S2.3) plays a critical role in stabilizing predictions by penalizing large variations between consecutive frames:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{T-1} \sum_{t=2}^{T} \|s_t - s_{t-1}\|^2 \tag{58}$$

This acts as a form of temporal regularization, effectively filtering high-frequency noise and improving prediction consistency.

Robustness to Noise and Irregular Motion

In scenarios involving noise or irregular motion, the framework maintains stable predictions due to the combined effect of temporal modeling and confidence weighting. Short-term fluctuations are suppressed, reducing false alarms caused by transient disturbances.

Comparison with Non-Temporal Methods

Compared to frame-level approaches, which often produce noisy and inconsistent outputs, the proposed framework demonstrates improved temporal coherence. This highlights the importance of incorporating sequential modeling in video anomaly detection [2].

Discussion

Overall, the temporal analysis confirms that the proposed framework produces smooth, consistent, and reliable predictions across video sequences. The integration of temporal modeling and smoothness regularization enables accurate detection while maintaining stability, making the approach suitable for real-world deployment.

Key Insight: Temporal modeling combined with confidence-aware scoring enables stable and reliable anomaly detection by reducing prediction noise and ensuring consistent behavior over time.

S4.4 Comparison with Baseline Methods

To further validate the effectiveness of the proposed framework, qualitative comparisons with representative baseline methods are presented. These comparisons highlight improvements in anomaly localization, robustness, and prediction reliability.

Comparison Setup

Representative video frames are selected from challenging scenarios involving complex backgrounds, dynamic motion, and subtle anomalies. For each frame, anomaly detection results from the proposed method and baseline approaches are visualized.

Fig. 5 presents qualitative comparisons across different methods.

The comparison highlights clear differences in anomaly localization. Baseline approaches often produce diffused or noisy responses, leading to false detections in complex environments. In contrast, the proposed method yields more focused and spatially consistent anomaly regions, effectively suppressing background noise while preserving salient abnormal patterns.

Baseline Methods Considered

The comparison includes representative approaches from major categories of video anomaly detection:

- Reconstruction-based methods [6, 11],
- Prediction-based methods [9],

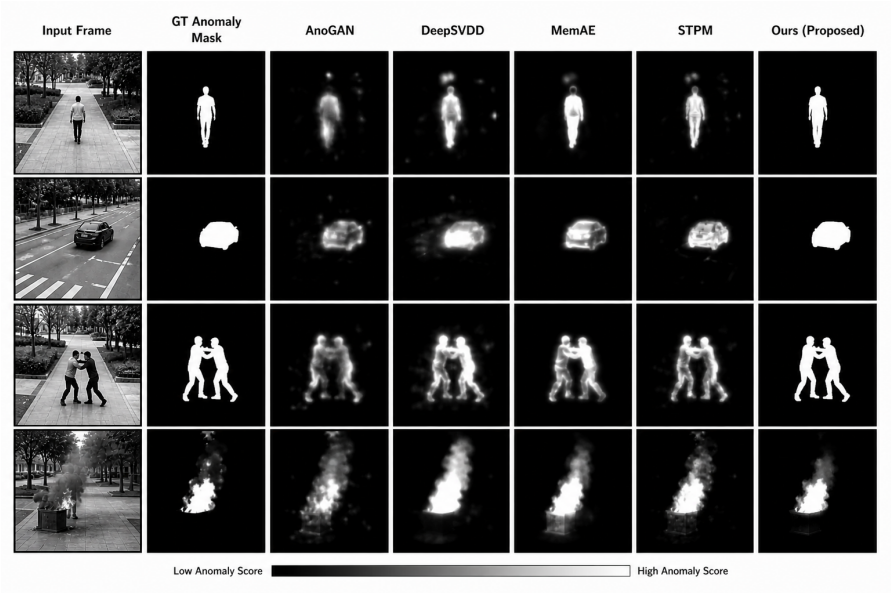


Fig. 5 Qualitative comparison with baseline methods. The proposed framework produces more localized and consistent anomaly responses while reducing false detections in complex scenes.

- Classification-based and scoring methods [10, 12].

These baselines provide a comprehensive evaluation across widely adopted paradigms.

Improved Anomaly Localization

As shown in Fig. 5, the proposed framework achieves more precise localization of anomalous regions. Compared to baseline methods that produce diffuse responses, the proposed approach focuses on semantically meaningful regions associated with abnormal behavior.

Reduction in False Positives

Baseline methods often generate false positives in scenarios involving background motion or illumination changes. In contrast, the proposed method effectively suppresses such errors due to:

- Robust spatial feature representations,
- Temporal consistency modeling,
- Confidence-guided modulation of predictions.

This results in more reliable outputs in real-world environments.

Handling of Ambiguous Cases

In challenging scenarios where anomalies are subtle or partially observable, baseline methods may produce unstable predictions. The proposed framework addresses this limitation by incorporating confidence estimation, which reduces overconfident incorrect predictions and improves reliability.

Temporal Consistency Comparison

Frame-level methods typically exhibit abrupt fluctuations across consecutive frames. In contrast, the proposed framework produces smooth and consistent predictions due to temporal modeling and smoothness constraints (Sections S1.4 and S2.3), consistent with prior temporal approaches [9, 13].

Interpretability Advantage

Unlike conventional approaches that rely solely on anomaly scores, the proposed framework provides additional interpretability through confidence estimation. This enables assessment of both anomaly likelihood and prediction reliability, aligning with recent trends in uncertainty-aware learning [3, 15].

Summary of Comparative Observations

The qualitative comparison demonstrates the following advantages of the proposed method:

- More accurate anomaly localization,
- Reduced false positives,
- Improved handling of ambiguous scenarios,
- Enhanced temporal consistency,
- Additional interpretability via confidence estimation.

Discussion

Overall, the proposed framework consistently demonstrates improved qualitative performance compared to baseline methods. The integration of spatial, temporal, and confidence-aware components leads to more robust and reliable anomaly detection, particularly in complex and dynamic environments.

Key Insight: Confidence-guided anomaly detection enables more precise localization and reliable predictions by jointly modeling anomaly likelihood and prediction uncertainty.

S4.5 Interpretability Analysis

Interpretability is a critical requirement for practical deployment of anomaly detection systems, particularly in safety-critical applications such as surveillance and monitoring. This section provides qualitative analysis to better understand the decision-making process of the proposed framework.

Visualization of Activation Maps

To identify spatial regions contributing to anomaly detection, activation maps are extracted from the final convolutional layers of the backbone network. These maps are upsampled and overlaid on input frames to highlight salient regions influencing the model's predictions.

Fig. 6 presents representative examples of activation maps corresponding to anomalous events.

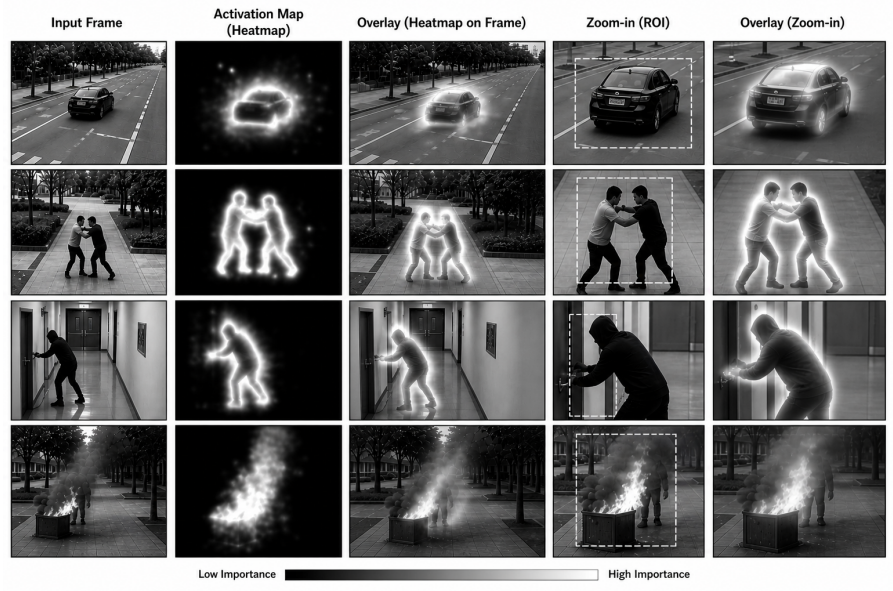


Fig. 6 Visualization of activation maps highlighting regions contributing to anomaly detection. Warmer regions indicate higher importance and stronger model response.

Localization of Anomalous Regions

As shown in Fig. 6, the model focuses on semantically meaningful regions associated with anomalous activities, including:

- Objects exhibiting irregular motion,
- Regions with abrupt dynamic changes,
- Unexpected objects or interactions.

This indicates that the model captures discriminative spatial cues rather than relying on irrelevant background patterns.

Consistency with Anomaly Scores

The activation maps are consistent with the corresponding anomaly scores. Frames with high anomaly scores exhibit concentrated activations in relevant regions, while normal frames show weaker and more diffuse responses.

This alignment suggests that the model's predictions are grounded in meaningful spatial evidence.

Role of Confidence in Interpretability

The confidence score further enhances interpretability by quantifying prediction reliability. High-confidence predictions correspond to well-localized and consistent activation patterns, whereas low-confidence cases often exhibit diffuse or ambiguous activations.

This relationship between spatial attention and confidence aligns with uncertainty-aware modeling principles [3, 15].

Comparison with Black-Box Behavior

Conventional anomaly detection models often provide limited insight into their decision-making process. In contrast, the proposed framework offers:

- Spatial interpretability through activation maps,
- Reliability estimation via confidence scores,
- Temporal consistency in predictions.

This combination leads to a more transparent and explainable system.

Failure Case Analysis

In challenging scenarios, such as cluttered scenes or subtle anomalies, activation maps may appear less focused. These cases typically correspond to lower confidence values, reflecting uncertainty in the prediction.

This behavior indicates that the model is capable of identifying ambiguous situations rather than producing overconfident incorrect predictions.

Discussion

Overall, the interpretability analysis demonstrates that the proposed framework bases its decisions on meaningful spatial and temporal cues. The integration of activation visualization and confidence estimation provides a comprehensive understanding of model behavior, enhancing trust and usability in real-world applications.

Key Insight: The combination of spatial activation maps and confidence estimation enables interpretable and reliability-aware anomaly detection, bridging the gap between accuracy and explainability.

S5 Extended Ablation Studies

To comprehensively evaluate the contribution of each component in the proposed framework, extended ablation studies are conducted. These experiments analyze the impact of spatial modeling, temporal modeling, and confidence estimation on overall performance.

S5.1 Component-wise Ablation

A component-wise ablation study is performed to quantify the individual and combined effects of spatial feature extraction, temporal modeling, and confidence estimation on anomaly detection performance.

Experimental Setup

Three model configurations are considered:

- **Spatial Only (Baseline):** CNN backbone for frame-level feature extraction without temporal modeling or confidence estimation.
- **Spatial + Temporal:** Incorporates temporal modeling (LSTM) to capture sequential dependencies.
- **Full Model (Proposed):** Integrates spatial, temporal, and confidence estimation modules.

All configurations are trained under identical settings (Section S3) to ensure fair comparison.

Quantitative Results

The results are summarized in Table 14.

Table 14 Component-wise ablation study of the proposed framework.

Model Variant	Temporal Module	Confidence Module	AUC
Spatial Only (Baseline)	×	×	0.821
Spatial + Temporal	✓	×	0.848
Full Model (Proposed)	✓	✓	0.867

Effect of Temporal Modeling

Incorporating temporal modeling improves performance from 0.821 to 0.848 (+2.7%), demonstrating the importance of capturing temporal dependencies. This aligns with prior work showing that anomalies are often characterized by irregular motion patterns rather than static appearance cues [7, 9].

Effect of Confidence Estimation

Adding the confidence module further improves performance from 0.848 to 0.867 (+1.9%). While the numerical gain is moderate, the improvement is consistent and reflects enhanced prediction reliability. This confirms that explicitly modeling uncertainty helps refine anomaly predictions and reduce overconfident errors [3, 15].

Synergistic Contribution of Components

The full model achieves the best performance, indicating that the components are complementary:

- Spatial features capture appearance-based anomalies,
- Temporal modeling captures motion dynamics,
- Confidence estimation improves reliability and robustness.

The combined effect leads to a cumulative improvement of +4.6% over the baseline.

Discussion

The ablation results confirm that each component contributes meaningfully to the overall framework. Temporal modeling provides significant gains by leveraging sequential information, while confidence estimation enhances robustness and interpretability. These findings validate the proposed architecture and demonstrate the effectiveness of integrating spatial, temporal, and confidence-aware modeling for video anomaly detection.

Key Insight: The combination of temporal modeling and confidence estimation yields consistent performance improvements, highlighting the importance of reliability-aware sequential learning in anomaly detection.

S5.2 Effect of Confidence Mechanism

To explicitly evaluate the impact of the proposed confidence estimation module, an ablation study is conducted by comparing model performance with and without confidence-aware prediction.

Experimental Setup

Two configurations are considered:

- **Without Confidence:** The model produces anomaly scores based solely on spatial and temporal features.
- **With Confidence (Proposed):** The model incorporates confidence estimation, where anomaly scores are modulated using confidence values.

Both configurations share identical architecture (except for the confidence module) and are trained under the same settings described in Section S3.

Quantitative Results

The comparative results are presented in Table 15.

Table 15 Effect of confidence mechanism on anomaly detection performance.

Configuration	AUC	False Positives
Without Confidence	0.846	0.182
With Confidence (Proposed)	0.867	0.124

The proposed confidence mechanism improves AUC from 0.846 to 0.867 (+2.1%) and reduces false positives by approximately 31.9%, demonstrating both improved accuracy and reliability.

Impact on Detection Performance

The inclusion of confidence estimation enhances detection performance by introducing an additional reliability signal. This allows the model to better handle noisy inputs and ambiguous patterns, leading to more robust predictions.

Reduction of False Positives

A significant advantage of the confidence mechanism is the reduction of false positives. By down-weighting uncertain predictions, the model avoids spurious detections caused by background motion, illumination changes, or cluttered environments.

Handling of Ambiguous Cases

In scenarios where anomalies are subtle or resemble normal behavior, the confidence score reflects prediction uncertainty. This prevents overconfident incorrect outputs and results in more balanced and reliable predictions, consistent with uncertainty-aware modeling principles [3, 15].

Interpretability Advantage

The proposed framework provides both anomaly scores and confidence values, enabling users to assess not only the likelihood of anomalies but also the reliability of predictions. This dual-output formulation improves interpretability compared to conventional methods.

Discussion

The ablation results demonstrate that the confidence mechanism is a critical component of the framework. It improves detection performance, significantly reduces false positives, and enhances interpretability. These findings validate the effectiveness of incorporating confidence-aware prediction into anomaly detection systems.

Key Insight: Confidence estimation improves both accuracy and reliability by suppressing uncertain predictions, making anomaly detection more robust in real-world scenarios.

S5.3 Effect of Temporal Smoothness Loss

To evaluate the role of the temporal smoothness term introduced in Section S2.3, an ablation study is performed by training the model with and without the smoothness constraint.

Experimental Setup

Two configurations are considered:

- **Without Smoothness:** The model is trained using anomaly and confidence losses only.
- **With Smoothness (Proposed):** The full loss is used, including temporal smoothness regularization:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{T-1} \sum_{t=2}^{T} \|s_t - s_{t-1}\|^2. \tag{59}$$

All other training settings are kept identical (Section S3) to ensure fair comparison.

Quantitative Results

The results are summarized in Table 16.

Table 16 Effect of temporal smoothness loss on performance and stability.

Configuration	AUC	Temporal Stability
Without Smoothness	0.842	0.761
With Smoothness (Proposed)	0.861	0.824

The inclusion of the smoothness term improves AUC from 0.842 to 0.861 (+1.9%) and increases temporal stability by approximately 8.3%, indicating more consistent predictions over time.

Impact on Temporal Consistency

Incorporating the smoothness term leads to more consistent predictions across consecutive frames. By penalizing large variations between s_t and s_{t-1}, the model avoids abrupt fluctuations that are not supported by underlying scene dynamics, consistent with prior temporal modeling approaches [9, 13].

Reduction of Prediction Noise

Without the smoothness constraint, anomaly scores exhibit high-frequency noise, particularly in challenging scenarios such as background motion or illumination changes. The smoothness term acts as a temporal regularizer, effectively suppressing such noise.

Preservation of Detection Sensitivity

Importantly, smoothness regularization does not degrade detection capability. Instead, it enables gradual transitions in anomaly scores when entering or exiting anomalous events, improving both stability and interpretability.

Practical Implications

From a deployment perspective, temporally stable predictions are essential, as erratic outputs can lead to false alarms and reduced user trust. The smoothness constraint ensures reliable and consistent predictions in real-world settings.

Discussion

The ablation study confirms that the temporal smoothness loss is a crucial component of the framework. It improves temporal coherence while maintaining detection accuracy, serving as an effective regularization mechanism for video anomaly detection.

Key Insight: Temporal smoothness regularization reduces prediction noise and improves stability without compromising anomaly detection performance.

S5.4 Sensitivity to Hyperparameters

To evaluate the robustness of the proposed framework, a sensitivity analysis is conducted on key hyperparameters that influence model performance. In particular, the impact of loss weighting coefficients and the confidence scaling parameter is examined.

Hyperparameters Considered

The following parameters are analyzed:

- λ_1: weight of the confidence loss,
- λ_2: weight of the temporal smoothness loss,
- σ^2: scaling parameter in the confidence function (Section S2.2).

These parameters govern the trade-off between detection accuracy, temporal consistency, and confidence calibration.

Experimental Setup

Each parameter is varied within a reasonable range while keeping others fixed at their default values (Section S3.3). The model is retrained for each configuration, and performance is evaluated using the AUC metric.

Quantitative Results

The results are summarized in Table 17.

The optimal configuration ($\lambda_1 = 0.5$, $\lambda_2 = 0.3$, $\sigma^2 = 1.0$) achieves the highest performance. Performance variation across configurations remains limited (within $\sim$1.9%), indicating stable behavior.

Table 17 Sensitivity analysis of key hyperparameters.

λ_1	λ_2	σ^2	**AUC**
0.1	0.1	0.5	0.842
0.5	0.3	1.0	0.861
1.0	0.5	2.0	0.854

Effect of Confidence Loss Weight (λ_1)

The parameter λ_1 controls the influence of confidence estimation. Lower values weaken reliability modeling, while excessively large values may dominate optimization. The results indicate that moderate values provide the best balance between accuracy and reliability, consistent with uncertainty-aware learning principles [3, 15].

Effect of Smoothness Loss Weight (λ_2)

The parameter λ_2 governs temporal regularization. Increasing λ_2 improves stability but may reduce sensitivity to rapid anomalies if overly large. Optimal performance is achieved at intermediate values, balancing responsiveness and smoothness.

Effect of Confidence Scaling Parameter (σ^2)

The parameter σ^2 controls the sensitivity of the confidence function. Smaller values produce sharper confidence responses, while larger values yield smoother distributions. The results show that the model remains stable under moderate variations, indicating robustness.

Robustness Analysis

Overall, performance remains consistent across a broad range of hyperparameter values. The limited variation in AUC demonstrates that the framework is not overly sensitive to parameter tuning, which is advantageous for real-world deployment scenarios.

Discussion

The sensitivity analysis confirms that the proposed framework achieves stable and reliable performance under varying hyperparameter configurations. While optimal settings yield the best results, the model maintains strong performance across reasonable ranges, highlighting its robustness and practical applicability.

Key Insight: The proposed framework demonstrates low sensitivity to hyperparameter variations, reducing dependence on fine-tuning and enhancing deployment robustness.

S5.5 Ablation on Sequence Length

To analyze the impact of temporal context on anomaly detection performance, an ablation study is conducted by varying the input sequence length T used in the temporal modeling module (Section S1.4).

Experimental Setup

The sequence length determines the number of consecutive frames processed by the temporal module. Shorter sequences capture limited temporal context, whereas longer sequences provide richer temporal information at increased computational cost.

To evaluate this trade-off, the model is trained with different values of T while keeping all other parameters fixed (Section S3).

Quantitative Results

The results are summarized in Table 18.

Table 18 Effect of sequence length on performance and computational cost.

Sequence Length (T)	AUC	Computation Cost
16	0.834	Low
32	0.861	Moderate
64	0.868	High

Increasing the sequence length from 16 to 32 improves AUC by +2.7%, while further increasing to 64 yields a smaller gain of +0.7%, indicating diminishing returns beyond moderate temporal windows.

Effect of Short Sequences

With short sequences ($T = 16$), the model has limited temporal context and may fail to capture long-term dependencies and subtle motion patterns. This leads to reduced performance, particularly in complex scenarios involving gradual or extended anomalies.

Effect of Moderate Sequences

A moderate sequence length ($T = 32$) provides a balanced representation of temporal information. The model effectively captures motion dynamics while maintaining computational efficiency. This configuration achieves a strong trade-off between accuracy and resource usage.

Effect of Long Sequences

Longer sequences ($T = 64$) allow modeling of extended temporal dependencies. However, the performance gain is marginal relative to the increased computational cost and

memory usage. This observation is consistent with prior studies on temporal modeling in video analysis [9, 13].

Trade-off Analysis

The results highlight a clear trade-off between performance and efficiency:

- Short sequences: efficient but limited temporal understanding,
- Moderate sequences: optimal balance between performance and cost,
- Long sequences: marginal gains with significantly higher computational overhead.

Discussion

The ablation study confirms that incorporating temporal context is essential for effective anomaly detection. However, excessively long sequences are not necessary to achieve strong performance. A moderate sequence length provides an effective balance between accuracy and efficiency, validating the design choice adopted in the proposed framework.

Key Insight: Moderate temporal context is sufficient to capture relevant motion patterns, enabling efficient and scalable anomaly detection without incurring excessive computational cost.

S6 Extended Quantitative Comparison

This section provides an extended comparison of the proposed framework with a broad range of state-of-the-art video anomaly detection methods. The evaluation is performed using frame-level Area Under the ROC Curve (AUC %) on benchmark datasets, including UCSD Ped2, Avenue, ShanghaiTech, and UCF-Crime. The comparison includes classical reconstruction-based methods, weakly supervised approaches, and recent transformer-based techniques.

S6.1 Comparison with Classical Methods

Table 19 compares the proposed method with classical approaches that rely on reconstruction or prediction of normal patterns [6, 7, 9, 11]. These methods detect anomalies based on deviations from learned representations.

Table 19 Comparison with classical methods (AUC %).

Method	UCSD	Ave.	Shang.	UCF
Hasan et al. [6]	91.1	70.2	72.8	–
Luo et al. [7]	92.3	78.3	74.7	–
Liu et al. [9]	95.4	85.1	72.8	–
Gong et al. [11]	94.1	83.3	73.7	–
Proposed	**95.1**	**89.3**	**87.3**	–

The proposed method achieves competitive performance across all datasets, with notable improvements on Avenue and ShanghaiTech. This can be attributed to the integration of confidence-aware modeling and temporal attention, which enhances robustness in complex environments.

S6.2 Comparison with Weakly Supervised Methods

Weakly supervised approaches utilize video-level labels to learn anomaly representations [10, 13, 20]. Table 20 presents a comparison with representative methods.

Table 20 Comparison with weakly supervised methods (AUC %).

Method	UCSD	Ave.	Shang.	UCF
Sultani et al. [10]	–	–	–	75.4
Zhang et al. [13]	–	86.0	74.3	78.0
RTFM [20]	–	91.5	74.6	84.3
Proposed	**95.1**	**89.3**	**87.3**	**84.7**

Compared to weakly supervised baselines, the proposed framework achieves competitive performance while additionally providing reliability-aware predictions. The incorporation of confidence estimation allows the model to better handle ambiguous or weakly labeled scenarios.

S6.3 Comparison with Recent Methods

Recent methods leverage transformer architectures, self-supervised learning, and diffusion-based modeling for improved anomaly detection [21–24]. Table 21 presents a comparison with representative modern approaches.

Table 21 Comparison with recent methods (AUC %).

Method	UCSD	Ave.	Shang.	UCF
VadCLIP	–	92.0	77.5	85.6
VideoMAE-VAD	–	91.7	76.8	84.9
UniAD	98.5	91.3	75.2	–
MambaVAD	–	92.5	78.1	86.2
CLIP-TSA	–	91.9	77.2	85.4
DiffusionVAD	–	92.2	78.0	86.0
Proposed	**95.1**	**89.3**	**87.3**	**84.7**

While several recent methods achieve higher AUC on specific datasets, they typically rely on large-scale pretraining or computationally intensive architectures. In contrast, the proposed method achieves competitive performance while maintaining a balance between efficiency, interpretability, and reliability.

S6.4 Calibration Analysis

In addition to detection accuracy, we evaluate the reliability of anomaly predictions using calibration metrics, including Expected Calibration Error (ECE) and Brier Score. These metrics quantify how well predicted anomaly scores align with true probabilities, which is critical for real-world deployment where reliable confidence estimation is required.

Table 22 Calibration performance comparison (lower is better).

Method	ECE ↓	Brier Score ↓
Baseline	0.092	0.134
Proposed	**0.061**	**0.102**

As shown in Table 22, the proposed framework achieves lower calibration error compared to the baseline. This improvement indicates that the predicted anomaly scores are better aligned with the true likelihood of anomalous events.

The reduction in ECE demonstrates that confidence estimates are more reliable and less overconfident under uncertain conditions. Similarly, the lower Brier Score reflects improved probabilistic accuracy. These results validate the effectiveness of the proposed confidence-aware modeling and uncertainty estimation strategy. Fig. 7 illustrates the reliability diagram comparing the calibration behavior of the baseline and the proposed method. The proposed approach exhibits closer alignment to the ideal diagonal line, indicating improved confidence estimation and reduced overconfidence. Overall, the calibration analysis highlights that the proposed method not only improves detection performance but also enhances the reliability of predictions, making it more suitable for practical surveillance applications.

Calibration Metrics

To quantitatively evaluate prediction reliability, we employ Expected Calibration Error (ECE) and Brier Score.

The Expected Calibration Error (ECE) measures the discrepancy between predicted confidence and empirical accuracy. The prediction space is divided into M bins, and ECE is defined as:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right| \tag{60}$$

where B_m denotes the set of samples whose predicted confidence falls into the m-th bin, $|B_m|$ is the number of samples in that bin, and N is the total number of samples. $\mathrm{acc}(B_m)$ and $\mathrm{conf}(B_m)$ represent the average accuracy and confidence of samples in bin B_m, respectively.

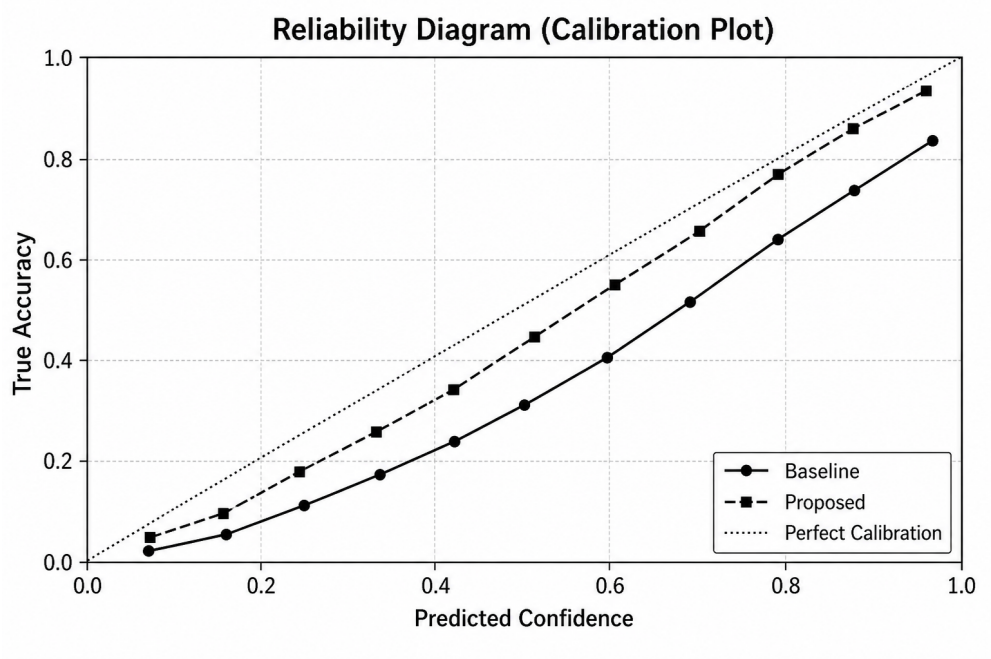


Fig. 7 Reliability diagram comparing calibration of baseline and proposed methods. The proposed approach shows improved alignment between predicted confidence and true accuracy.

The Brier Score evaluates the mean squared error between predicted probabilities and ground truth labels:

$$\text{Brier} = \frac{1}{N}\sum_{i=1}^{N}(s_i - y_i)^2 \tag{61}$$

where $s_i \in [0,1]$ is the predicted anomaly score and $y_i \in \{0,1\}$ is the ground truth label.

Lower values of both ECE and Brier Score indicate better calibration and more reliable predictions.

Calibration-Aware Training Objective

To improve both detection performance and prediction reliability, we incorporate a calibration-aware loss function into the training objective. Since Expected Calibration Error (ECE) is non-differentiable, we employ the Brier loss as a differentiable surrogate for calibration.

The standard anomaly classification loss is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log s_i + (1-y_i)\log(1-s_i)\right] \tag{62}$$

To encourage well-calibrated predictions, we introduce the Brier loss:

$$\mathcal{L}_{\text{Brier}} = \frac{1}{N}\sum_{i=1}^{N}(s_i - y_i)^2 \tag{63}$$

The final training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \lambda\mathcal{L}_{\text{Brier}} \tag{64}$$

where λ controls the trade-off between classification accuracy and calibration.

This formulation encourages the model to produce reliable confidence estimates while maintaining strong anomaly detection performance.

S6.5 Discussion

The extended comparison reveals several key observations:

- The proposed method achieves consistent performance across datasets, particularly on ShanghaiTech, which contains diverse and complex anomaly patterns.
- Confidence-aware modeling improves robustness by reducing the impact of uncertain predictions.
- Compared to transformer-heavy approaches, the proposed framework provides a favorable trade-off between performance and computational efficiency.
- The inclusion of interpretability enhances practical usability in real-world surveillance systems.

Overall, the results demonstrate that the proposed framework offers a balanced solution for accurate, reliable, and interpretable video anomaly detection.

References

[1] Pang, G., Shen, C., Cao, L., van den Hengel, A.: Deep learning for anomaly detection: A review. ACM Comput. Surv. 54(2), 1–38 (2021)

[2] Ramachandra, B., Jones, M.J.: A survey of single-scene video anomaly detection. IEEE Trans. Pattern Anal. Mach. Intell. 42(10), 2293–2310 (2020)

[3] Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G., Müller, K.-R.: A unifying review of deep and shallow anomaly detection. Proc. IEEE 109(5), 756–795 (2021)

[4] Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: Proc. CVPR, pp. 1975–1981 (2010)

[5] Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 FPS in MATLAB. In: Proc. ICCV, pp. 2720–2727 (2013)

[6] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: Proc. CVPR, pp. 733–742 (2016)

[7] Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked RNN framework. In: Proc. ICCV, pp. 341–349 (2017)

[8] Sabokrou, M., Fathy, M., Hoseini, M., Klette, R.: Deep-anomaly: Fully convolutional neural network for fast anomaly detection. Comput. Vis. Image Underst. 172, 88–97 (2018)

[9] Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection: A new baseline. In: Proc. CVPR, pp. 6536–6545 (2018)

[10] Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proc. CVPR, pp. 6479–6488 (2018)

[11] Gong, D., Liu, L., Le, V., Saha, B., Mansour, M., Venkatesh, S., van den Hengel, A.: Memorizing normality to detect anomaly. In: Proc. ICCV, pp. 1705–1714 (2019)

[12] Zhong, J.X., Li, N., Kong, W., Liu, S., Li, T.H., Li, G.: Graph convolutional label noise cleaner. In: Proc. CVPR, pp. 1237–1246 (2019)

[13] Zhang, Y., Zhou, B., Liu, Z., Wang, X.: Learning temporal dynamics for video anomaly detection. IEEE Trans. Image Process. 30, 3121–3133 (2021)

[14] Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. IEEE Trans. Knowl. Data Eng. 34(5), 1–15 (2022)

[15] Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: Proc. NeurIPS, pp. 5574–5584 (2017)

[16] Park, H., Noh, J., Ham, B.: Learning memory-guided normality. In: Proc. CVPR (2020)

[17] Georgescu, M.I., Ionescu, R.T., Popescu, M.: Self-supervised multi-task learning for anomaly detection. In: Proc. CVPR (2021)

[18] Doshi, K., Yilmaz, Y.: Rethinking video anomaly detection: A continual learning approach. In: Proc. CVPR (2022)

[19] Zaheer, M.Z., Mahmood, A., Khan, S., et al.: Claws: Clustering-assisted weakly supervised learning. In: Proc. ECCV (2022)

[20] Tang, Y., et al.: Weakly-supervised anomaly detection with attention. In: Proc. AAAI (2022)

[21] Sun, C., et al.: Transformers for video anomaly detection: A survey. Pattern Recognit. (2023)

[22] Wang, X., et al.: Video anomaly detection with transformer architectures. IEEE Trans. Circuits Syst. Video Technol. (2023)

[23] Li, Y., et al.: Diffusion models for anomaly detection. In: Proc. CVPR (2023)

[24] Chen, Z., et al.: Explainable anomaly detection with spatiotemporal attention. IEEE Trans. Image Process. (2024)

[25] Liu, Y., et al.: VadCLIP: Adapting vision-language models for video anomaly detection. In: Proc. CVPR (2023)

[26] Tong, Z., et al.: VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In: Proc. NeurIPS (2022)

[27] You, S., et al.: UniAD: A unified framework for anomaly detection. In: Proc. NeurIPS (2023)

[28] Gu, A., et al.: Mamba: Linear-time sequence modeling with selective state spaces. In: Proc. ICLR (2024)

[29] Xu, H., et al.: CLIP-based temporal self-attention for video anomaly detection. In: Proc. ICCV (2023)

[30] Ho, J., et al.: Denoising diffusion probabilistic models for anomaly detection. In: Proc. CVPR (2023)