Dual-scale Temporal Dependency Learning for Unsupervised Video Anomaly Detection

Xu Li, Xue Wang(✉), Zexing Du, and Qing Wang

School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
xwang@nwpu.edu.cn

Abstract. Video anomaly detection plays an increasingly crucial role in intelligent surveillance systems. Inspired by previous unsupervised methods, this paper focuses on detecting frame-level anomalies with long-term temporal dependencies. To this end, we propose a dual-scale temporal dependency learning method for video anomaly detection model, which consists of two main modules: a single-frame reconstruction module and a multi-frame feature enhancement module, processed end-to-end without relying on any pre-trained models. To validate the proposed approach, we introduce a new Elevator dataset containing various types of remote temporal dependency anomalies. Experimental results on the self-constructed Elevator dataset and two benchmarks demonstrate the effectiveness of our proposed approach.

Keywords: Video anomaly detection · Unsupervised learning · Long temporal dependency · Frame reconstruction.

1 Introduction

Video Anomaly Detection (VAD) is a high-level vision task utilizing computer vision and machine learning technologies to identify frame-level anomalous behaviors or events within video footage. Traditional video surveillance systems usually rely on manual intervention to identify anomalies, a method prone to inefficiency and time-lag. Consequently, the emergence of automatic VAD techniques addresses this gap by autonomously detecting anomalous behavior within video streams, enabling timely detection and response to issues.

Unfortunately, it is extremely challenging to identify and locate anomalies in a long video [3]. The one-class or unspervised learning setting has been widely and successfully adopted for VAD due to the imbalance between normal and anomaly events [28]. Note that, the term 'unsupervised' used here refers to the one-class classification strategy. Researchers rely on reconstructing a single video frame or predicting future frames based on a few nearby frames to detect anomalies within video data. During the process of reconstruction and prediction, these methods learn normal patterns to differentiate anomalies during testing. However, such methods are incapable of detecting anomalies with long temporal

Fig. 1. illustrates an example of remote temporal dependency anomalies. Individual frames seem normal, however, the opposite conclusion can be easily derived by considering long-term temporal dependencies.

dependencies in a long video. An individual frame may seem normal, but the oppsite conclusion can be easily derived considering other frames by extending the temporal range. As shown in Fig. 1, it seems nothing unsual happens in each individual frame. However, if longer temporal context is taken into consideration: in (a), the elevator door opens, and a girl appears and is going to enter the elevator; in (b), the girl stands at the entrance of the elevator, spreading her arms to hold the elevator door; in (c), the little girl stands at the entrance of the elevator. The transition from state (b) to (c) lasts for a long time (such as several hundred frames), which poses a significant risk and indicates an anomaly that can only be detected by considering long-term temporal dependencies.

To address the above limitation, we propose a novel video anomaly detection method leveraging dual-scale temporal dependency learning, which detects both local anomalies and remote dependency anomalies at different temporal scales. The multi-frame feature enhancement module learns features with longer temporal ranges and relates them to detect anomalies with remote temporal dependencies. The proposed method is evaluated on two large-scale benchmarks, CUHK Avenue and ShanghaiTech. Furhtermore, we self-construct a challenging Elevator dataset, comprising various anomalies with remote temporal dependencies. To summarize, our contributions include:

—We propose a **Dual-Scale Temporal Dependency Learning network (DSTDL)** for video anomaly detection, which is capable of detecting both local and remote dependency anomalies at different temporal scales.

—We introduce a multi-frame feature enhancement module employing self-attention and prototype pool to enhance remote temporal dependency learning.

—Experimental results on the self-constructed Elevator dataset, which comprises both short-term and long-term anomalies, show that the proposed method outperforms the state of the arts, especially for long-term anomaly detetion.

2 Related Work

Unsupervised Video Anomaly Detection. Exisiting VAD methods are primarily divided into two categories: unsupervised methods[14, 5, 23, 2, 7, 15] and weakly supervised methods [24, 25]. Weakly supervised methods utilize video-

level annotations or noisy labels. They often employ auxiliary information or prior knowledge to assist anomaly detection. Within the unsupervised framework or the one-class learning branch, methods based on reconstruction and prediction are the two most representative paradigms in the current era of deep neural network-based VAD.

Reconstruction-based methods typically employ deep autoencoders to learn to reconstruct input frames, considering frames with significant reconstruction errors as anomalies. Song et al.[23], Gong et al.[5], and Chen et al.[2] all explore different approaches to enhance autoencoders (AEs) for anomaly detection. The utilization of attention mechanisms in AEs, as highlighted by Song et al.[23] and Li et al.[11], provides significant benefits by selectively focusing on relevant features during encoding and decoding processes. Prediction based methods learn to predict missing frames, such as future frame prediction [4, 12, 17] or middle frame completion [9, 27]. The proposed method in this work belongs to the reconstruction-based unsupervised branch. Unlike previous reconstruction-based methods that struggle with detecting long-term dependency anomalies, our approach conducts frame reconstruction at different temporal scales. This allows for detection of both localized anomalies and temporal dependencies occurring over longer time spans.

Spatiotemporal Relationship Modeling. Currently, spatiotemporal relationship modeling has found numerous applications across diverse domains, including target detection [8], behavior recognition [13, 19], and beyond. For instance, Li et al. [10] extend relation modeling from the spatial domain to the spatio-temporal domain by incorporating an existing video temporal relation network, enabling the encoding of spatio-temporal dynamics within the video. Hao et al. [6] present a spatiotemporal consistency enhanced network to generate spatio-temporal consistency predictions. Wang et al. [26] propose to detect anomalies by analyzing the spatio-temporal relationships among objects. Different from these methods, our approach employs a distinct strategy by separately addressing spatio-temporal relationships across two temporal scales, local and remote temporal ranges.

Prototype Pool. Prototypical learning aims to represent each class or category by a prototypical feature vector computed from the features of its constituent instances, providing benefits such as robustness, interpretability, efficiency, and adaptability. For image classification, prototypes can serve as centroids or representatives of their respective classes in the feature space [22]. For action recognition, features of a video at different timestamps can be averaged to represent the video [21]. Previous methods focused on learning spatial prototype representations of single-frame features when using a prototype pool [17]. However, in our approach, each prototype is utilized to represent a class of generic continuous behavioral characteristics within a remote temporal domain. The prototype pool is employed to aggregate a series of generic continuous behavioral characteristics in normal videos, placing greater emphasis on learning spatio-temporal prototype representations within a long-term temporal range.

3 Methodology

In this section, we first present the overall process of the proposed unsupervised video anomaly detection method based on dual-scale learning in Section 3.1. Then, in Section 3.2, we describe the single-frame reconstruction module. Section 3.3 introduces the multi-frame feature enhancement module. Finally, in Section 3.4, we discuss the computation and composition of anomaly scores.

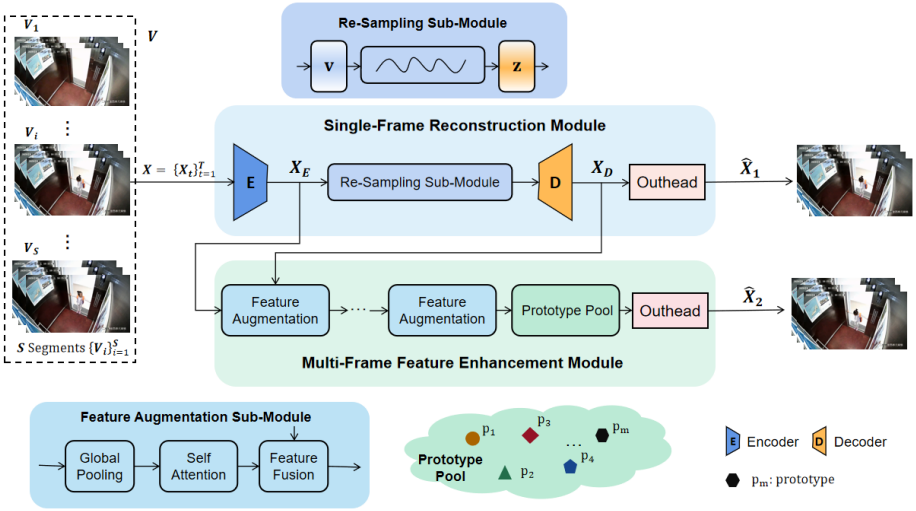

Fig. 2. The overall pipeline of our proposed **DSTDL** network.

3.1 Overview

The overall pipeline of our proposed **DSTDL** network is shown in Fig. 2. Given an input video $\boldsymbol{V}$, we first partition $\boldsymbol{V}$ into a series of non-overlapping video segments $\{\boldsymbol{v}_i\}_{i=1}^{S}$ follow the temporal order, where S represents the number of video segments. Each video segment comprises an equal number of video frames, denoted by $\boldsymbol{X} = \{\boldsymbol{X}_t\}_{t=1}^{T} \in \mathbb{R}^{B\times T\times C\times H\times W}$, where B, T, C, H, W represent the batch size, the number of video frames, the number of channels, the image height and width, respectively. Our **DSTDL** primarily consists of two components: the **single-frame reconstruction module** and the **multi-frame feature enhancement module**.

3.2 Single-frame Reconstruction Module

Inspired by the work [17], we employ autoencoders for single frame reconstruction. Specially, to enhance the discrimination between normal and abnormal frames, we introduce a re-sampling sub-module similar to [6] after the encoder, which maps the spatial distribution of encoded features to another feature space.

Firstly, the data is encoded along the channel dimension, which is akin to a downsampling process involving multiple convolution and pooling operations. This encoding process serves as a preliminary feature extraction step:

$$\boldsymbol{X}_E = E(\boldsymbol{X}), \tag{1}$$

where $E(\cdot)$ denotes the encoder, and $\boldsymbol{X}_E$ represents the encoded features. The re-sampling sub-module maps the spatial distribution of $\boldsymbol{X}_E$ from the source feature space $\boldsymbol{v}$ to the destination space $\boldsymbol{z}$ and obtain the re-sampled features:

$$\widehat{\boldsymbol{X}}_E = ReS(\boldsymbol{X}_E) : \{\boldsymbol{v} \rightarrow \boldsymbol{z}\}, \tag{2}$$

where the latent feature space $\boldsymbol{v}$ is assumed to be Gaussian distributed $N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, $\boldsymbol{z} = \boldsymbol{\mu} + \lambda\boldsymbol{\sigma}$ takes a similar distribution to $\boldsymbol{v}$, and $\lambda \sim N(0, 1)$ is an auxiliary noise parameter. Specifically, two fully connected layers are utilized to compute the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ of the Gaussian distribution. To constrain the re-sampling process, the Kullback-Leibler divergence is employed to ensure the similarity between the source and the destination distributions:

$$L_c = KL(N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2), N(0, 1)) = -\frac{1}{2}(1 + log\boldsymbol{\sigma}^2 - \boldsymbol{\mu}^2 - \boldsymbol{\sigma}^2). \tag{3}$$

Then, $\widehat{\boldsymbol{X}}_E$ is passed to the decoding part, which is akin to an upsampling process involving multiple deconvolution operations. This can be represented as:

$$\boldsymbol{X}_D = D(\widehat{\boldsymbol{X}}_E), \tag{4}$$

where $\boldsymbol{X}_D$ represents the decoded features, and $D(\cdot)$ denotes the decoder. The reconstructed video frames are finally obtained by passing the decoded features $\boldsymbol{X}_D$ through an output head, denoted as Θ_1. The output head is composed of multiple layers of convolution and activation functions:

$$\widehat{\boldsymbol{X}}_1 = \Theta_1(\boldsymbol{X}_D), \tag{5}$$

where $\widehat{\boldsymbol{X}}_1$ represents the reconstructed frames by the single-frame reconstruction module. The entire module is constrained by the mean squared error loss to achieve accurate frame reconstruction:

$$L_r = \frac{1}{HW}\sum_{h,w} \|\widehat{\boldsymbol{X}}_1 - \boldsymbol{X}\|_2^2. \tag{6}$$

3.3 Multi-frame Feature Enhancement Module

The intermediate results of the single-frame reconstruction module $\boldsymbol{X}_E$ and $\boldsymbol{X}_D$ are taken as input and fed into the multi-frame feature enhancement module. This module comprises three parts: stacked feature enhancement sub-modules, a prototype pool, and an output head.

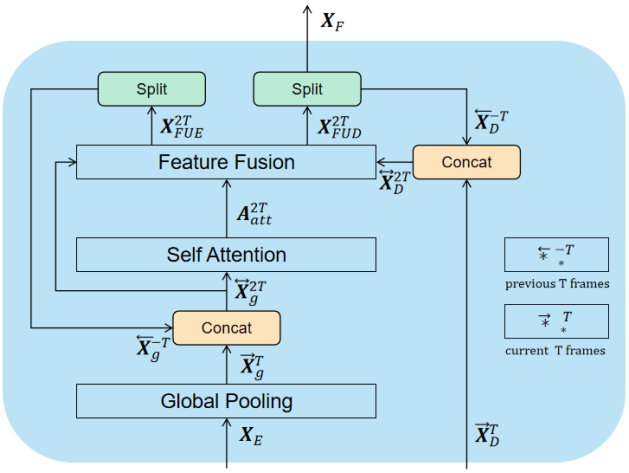

Fig. 3. Detailed structure of the feature enhancement sub-module.

Firstly, $\boldsymbol{X}_E$ and $\boldsymbol{X}_D$ are togethor passed into several sequentially stacked feature enchancement blocks for expanding the temporal range taken into consideration. As shown in Fig. 3, a single feature augmentation block includes three parts: global pooling, self-attention and feature fusion. Global pooling involves pooling the encoded features $\boldsymbol{X}_E$ along the channel, length, and width dimensions, retaining only the temporal dimension unchanged:

$$\overrightarrow{\boldsymbol{X}}_g^T = GP(\boldsymbol{X}_E), \tag{7}$$

where $GP(\cdot)$ represents the global average pooling operation, and $\boldsymbol{X}_g^T \in \mathbb{R}^{B\times T}$ represents the result after pooling over the current T frames. The pooled result is concatenated with the corresponding result from the previous iteration:

$$\begin{cases} \overleftrightarrow{\boldsymbol{X}}_g^{2T} = concat\left(\overleftarrow{\boldsymbol{X}}_g^{-T}, \overrightarrow{\boldsymbol{X}}_g^{T}\right), \\ \overleftrightarrow{\boldsymbol{X}}_D^{2T} = concat\left(\overleftarrow{\boldsymbol{X}}_D^{-T}, \overrightarrow{\boldsymbol{X}}_D^{T}\right), \end{cases} \tag{8}$$

where $\overleftarrow{\boldsymbol{X}}_g^{-T}$ represents the result after pooling over the previous T frames, $\overleftrightarrow{\boldsymbol{X}}_g^{2T} \in \mathbb{R}^{B\times 2T}$ represents the concatenated result of the previous T frames and the current T frames, $\overrightarrow{\boldsymbol{X}}_D^{T}$, $\overleftarrow{\boldsymbol{X}}_D^{-T}$ and $\overleftrightarrow{\boldsymbol{X}}_D^{2T}$ have similar representations for the decoded feature $\boldsymbol{X}_D$.

These concatenated features are subsequently fed into a self-attention block to calculate a correlation matrix capturing context information from the preceding T frames to the succeeding T frames:

$$\boldsymbol{A}_{att}^{2T} = SA(\overleftrightarrow{\boldsymbol{X}}_g^{2T}), \tag{9}$$

where $SA(\cdot)$ represents the self-attention operation, and $\boldsymbol{A}_{att}^{2T} \in \mathbb{R}^{B\times 2T\times 2T}$ represents the correlation matrix obtained for the $2T$ frames. This correlation matrix is further multiplied with the decoded features $\overleftrightarrow{\boldsymbol{X}}_{D}^{2T}$ to obtain the fused features:

$$\boldsymbol{X}_{FUD}^{2T} = \boldsymbol{A}_{att}^{2T} * \overleftrightarrow{\boldsymbol{X}}_{D}^{2T}. \tag{10}$$

Similarly, as shown in Fig. 3, the fused encoded feature $\boldsymbol{X}_{FUE}^{2T}$ can be calculated uisng $\boldsymbol{X}_{FUE}^{2T} = \boldsymbol{A}_{att}^{2T} * \overleftrightarrow{\boldsymbol{X}}_{g}^{2T}$. Throught the integration of the correlation matrix, which encapsulates context information spanning two adjacent video segments, the fused features undergo enhancement. This single-layer enhancement block is stacked multiple times to boost performance:

$$\boldsymbol{X}_F = \{FA_i(\boldsymbol{X}_E, \boldsymbol{X}_D)\}_{i=1}^{Z}, \tag{11}$$

where $FA_i(\cdot,\cdot)$ represents the i-th single-layer feature enhancement operation, Z denotes the iteration of the feature enhancement module for Z repetitions, and $\boldsymbol{X}_F$ is the features obtained by the stacked feature enhancement sub-modules. The feature enhancement operation is propagated continuously along the temporal axis (in the direction of increasing time), enabling the incorporation of a broader range of contextual information in the time domain. This facilitates the accurate detection of long-term temporal dependency anomalies.

Inspired by work [17], we used a prototype pool based on temporal information. After enhancement, the prototypes are obtained from the fused features $\boldsymbol{X}_F$:

$$\boldsymbol{p}_t^m = \sum_{n=1}^{N} \frac{w_t^{n,m}}{\sum_{n'=1}^{N} w_t^{n',m}} \boldsymbol{x}_t^n, \tag{12}$$

where $N = W*H$, $\boldsymbol{X}_F = \{\boldsymbol{x}_t^1, \boldsymbol{x}_t^2, ..., \boldsymbol{x}_t^N\}_{t=1}^{T}$, and $\boldsymbol{x}_t^n \in \mathbb{R}^c$. Within the Attention sub-process, M attention mapping functions $\{\psi_m : \mathbb{R}^c \to \mathbb{R}\}_{m=1}^{M}$ are used to allocate contextual weights to the encoded vectors, here, $w_t^{n,m} \in W_t^m = \psi_m(\boldsymbol{X}_F)$. A collection of N encoded vectors forms a prototype $\boldsymbol{p}_t^m \in \mathbb{R}^c$ and M prototypes constitute the prototype pool $\boldsymbol{P}_t = \{\boldsymbol{p}_t^m\}_{m=1}^{M}$. During training, both the similarity loss L_s and the diversity loss L_d are employed for generating prototypes:

$$L_s = \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{x}_t^n - \boldsymbol{p}_t^*\|_2, \tag{13}$$

$$L_d = \frac{2}{M(M-1)} \sum_{m=1}^{M} \sum_{m'=1}^{M} [-\|\boldsymbol{p}_m - \boldsymbol{p}_{m'}\|_2]. \tag{14}$$

Note that in the retrieval sub-process, the enhanced vector $\boldsymbol{x}_t^n$ is utilized as a query to retrieve relevant items from the prototype pool. The retrieved relevant prototypes are then used to reconstruct the normalcy encoding $\widetilde{\boldsymbol{X}}_T$, of which each normalcy encoding vector $\widetilde{\boldsymbol{x}}_t^n$ is calculated as follows:

$$\widetilde{\boldsymbol{x}}_t^n = \sum_{m=1}^{M} \beta_t^{n,m} \boldsymbol{p}_t^m, \tag{15}$$

where $\beta_t^{n,m} = \frac{\boldsymbol{x}_t^n \boldsymbol{p}_t^m}{\sum_{m'=1}^{M} \boldsymbol{x}_t^n \boldsymbol{p}_t^{m'}}$ represents the correlation score between the n-th vector $\boldsymbol{x}_t^n$ and the m-th prototype item $\boldsymbol{p}_t^m$. The normalcy encoding $\widetilde{\boldsymbol{X}}_T$ is then transformed into reconstructed video frames $\widehat{\boldsymbol{X}}_2$ throught an output head Θ_2, which is composed of multiple layers of convolution and activation functions:

$$\widehat{\boldsymbol{X}}_2 = \Theta_2(\widetilde{\boldsymbol{X}}_T). \tag{16}$$

Similar to the reconstruction loss L_m for the single-frame reconsruction module, defined in Eq. 6, the multi-frame feature enhancement module is also subjected to a mean squared error loss constraint to achieve frame reconstruction, which can be described as follows:

$$L_m = \frac{1}{HW} \sum_{h,w} \|\widehat{\boldsymbol{X}}_2 - \boldsymbol{X}\|_2^2. \tag{17}$$

Therefore, the total loss L_t is defined as follows:

$$L_t = \lambda_1 * L_c + \lambda_2 * L_r + \lambda_3 * L_s + \lambda_4 * L_d + \lambda_5 * L_m, \tag{18}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and λ_5 represent the weighting coefficients for corresponding loss terms, respectively.

3.4 Anomaly Score

The PSNR (Peak Signal-to-Noise Ratio) metric is calculated based on the reconstructed frame $\widehat{X}$ and the ground truth X as follows:

$$PSNR(\boldsymbol{X}, \widehat{\boldsymbol{X}}) = 10log \frac{[max(\widehat{\boldsymbol{X}})]^2}{\frac{1}{N}\sum_{i=0}^{N}(\boldsymbol{X}_i - \widehat{\boldsymbol{X}}_i)^2}. \tag{19}$$

After normalizing the obtained PSNR values across the entire video, the anomaly score s for each video frame could be derived as follows:

$$s = \frac{PSNR(\boldsymbol{X}, \widehat{\boldsymbol{X}}) - \min_t PSNR(\boldsymbol{X}, \widehat{\boldsymbol{X}})}{\max_t PSNR(\boldsymbol{X}, \widehat{\boldsymbol{X}}) - \min_t PSNR(\boldsymbol{X}, \widehat{\boldsymbol{X}})}. \tag{20}$$

The anomaly scores for an individual reconstructed frame from $\widehat{\boldsymbol{X}}_1$ and $\widehat{\boldsymbol{X}}_2$ are calculated according to Eqs. 19 and 20, respectively. Then a weighted sum s_t for the frame at time t is obtained as follows:

$$s_t = \alpha s_1 + (1 - \alpha s_2), \tag{21}$$

where s_1 and s_2 represent the anomaly scores for individual reconstructed frame from $\widehat{\boldsymbol{X}}_1$ and $\widehat{\boldsymbol{X}}_2$ respectively, and α denotes the weighting coefficient.

4 Experiments

4.1 Datasets and Evaluation Metrics

We consider two benchmarks in our analysis, CUHK Avenue [14] and ShanghaiTech [16]. Avenue has 37 videos, including 16 training videos and 21 test videos, respectively. It includes a total number of 47 abnormal events with throwing bag and moving toward/away from the camera being example anomalies. Each video has a resolution of 360×640 RGB pixels. ShanghaiTech has 437 videos, including 307 normal videos and 130 anomaly videos, all collected under 13 different scenes with complex shooting angles. Each video has a resolution of 480×856 RGB pixels.

Furthermore, we self-construct a dataset comprising 192 surveillance videos (≈190 minutes) recorded inside elevators within residential complexes. The Elevator dataset consists of 169 normal training videos and 23 anomaly testing videos, of which eash video has a resolution of 1920×1080 RGB pixels. A total of 8 types of abnormal events are considered, including *animal in the elevator, electrics vehicle in the elevator, suspected weapon, large cargo, smoking, absence of individual when the elevator door opens, passenger holding the elevator door for an extended period, and passenger continuously moving around.*

Following [17], we compute the Area Under the Curve (AUC) of the frame-level receiver operating characteristics (ROC) as the main metric to evaluate the performance of our proposed method and comparison methods, where a larger AUC value implies better distinguishing ability.

4.2 Implementation Details

For training, all the videos are divided into 20 segments ($S = 20$), with each segment containing 10 frames ($T = 10$). For Elevator, each video comprises 200 frames. While for Avenue and ShanghaiTech, each training video undergoes average downsampling to maintain 200 frames. Input frames are resized to the resolution of 224×224 and normalized to the range of [-1, 1] following [17]. The encoder and decoder in the single-frame reconstruction module use the same settings as in [17]. The size of the prototype pool for the multi-frame feature enhancement module is set to 10 ($M = 10$). The weighting coefficient α for calculating the final anomaly score is set to 0.5. The coefficients $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and λ_5 for the loss function are set to 1, 1, 1, 0.0001 and 1, respectively (where the values for coefficients λ_2, λ_3 and λ_4 are referenced from [17]). All hyper-parameters remain consistent across three datasets.

We train our network on two NVIDIA RTX-2080Ti GPUs using PyTorch and trained in an end-to-end manner. The learning rates are set to 1e-5, 1e-4, 1e-4 for Elevator, Avenue and ShanghaiTech, respectively.

4.3 Comparisons with State-Of-The-Art (SOTA)

We implement a full convolutional encoder-decoder video anomaly detection method with a prototype pool as the baseline, similar to the work presented

Table 1. Performance comparison with existing SOTA methods on Avenue, ShanghaiTech and Elevator datasets. The best performance is **bold**, while the 2nd and 3rd performances are underlined.

Methods	Avenue	ShanghaiTech	Elevator
MemAE[5]	83.3	71.2	-
Stacked RNN[16]	81.7	68.0	-
Conv-AE[7]	70.2	60.9	-
ConvLSTM-AE[15]	77.0	-	-
AMC[18]	**86.9**	-	-
CDDA[1]	86.0	**73.3**	-
MNAD[20]	82.8	69.8	60.4
MPN[17]	84.0	66.7	63.4
Ours	85.7	70.5	**76.5**

Table 2. Ablation analysis of loss terms on Elevator dataset.

L_c	L_r	L_s	L_d	L_m	AUC(%)
✓	✓				67.82
	✓	✓	✓	✓	71.28
✓		✓	✓	✓	57.75
✓	✓		✓	✓	72.35
✓	✓	✓		✓	70.28
✓	✓	✓	✓		68.92
✓	✓	✓	✓	✓	**76.51**

in [17]. Moreover, our method is compared with several reconstruction-based unsupervised works [5, 16, 7, 15, 18, 1, 20]. Some methods are not compared on Elevator dataset due to the lack of publicly released codes from their authors. As shown in Table 1, although our method cannot achieve the optimal performance on Avenue and ShanghaiTech datasets, our method achieves improvements of 1.7% and 3.8% over the baseline, respectively. Furthermore, our method achieves a 13.1% improvement compared to the baseline on Elevator dataset. Additional details can be found in the supplementary material. Compared to the proposed method, without the multi-frame feature enhancement and reconstruction branch, the baseline shows a significant decline in long-term dependency anomaly detection.

4.4 Ablation Study

We conduct ablation studies to validate the effectiveness of the loss terms in our **DSTDL** on Elevator dataset. The total loss function comprises the resampling loss, the single-frame reconstruction loss, the similarity loss, the diversity loss and the multi-frame reconstruction loss. As shown in Table 2, the multi-frame reconstruction loss enhances the model's awareness of long-term temporal depen-

dencies, thus introduces a nearly 7.59% AUC improvement on Elevator dataset. From the table, we can also see that the single-frame reconstruction loss plays an important role. The single-frame reconstruction module has a good detection of abnormal pedestrian movement. At the same time, the input of the multi-frame module depends on the output of the encoder and decoder of the single-frame module. In addition, combing all the loss terms together can further improve the performance, indicating their compatibility in video anomaly detection.

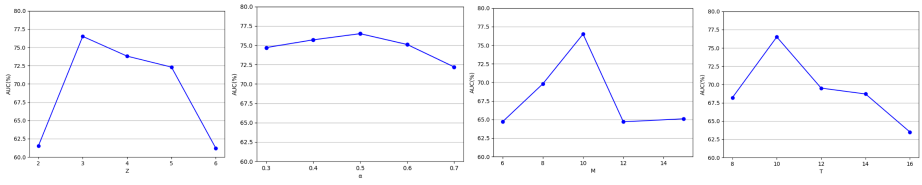

Fig. 4. Analysis of the effects of the key hyperparameters of DSTDL on Elevator dataset. **From left to right**: the number of stacked blocks of the single-frame feature enhancement module (Z), the weighted sum parameters of the anomaly score (α), the prototype pool size (M), and the number of frames per video segment (T).

4.5 Robustness Analysis

To enhance the features extracted from a larger temporal scale, the single-layer enhancement block is stacked multiple times to boost performance. To investigate the effect of the number of stacked blocks, denoted as Z in Eq. 11 , we carry out experiments on different numbers of stacked blocks on Elevator dataset. The results are showed in Fig. 4. Based on the results, $Z = 3$ is an appropriate number of stacked blocks. We set α to 0.5 in the experiment.

The anomaly score for an individual frame is a weighted sum of the anomaly scores obtained from the reconstructed frames by two branches of our proposed DSTDL network. To analyze the influence of the weighting coefficient α on the VAD performance, we also carry out experiments on different values of α on Elevator dataset. From the results showed in Fig. 4, when $\alpha = 0.5$ the proposed method achieves a highest AUC value. We set Z to 3 in the experiment.

In addition, we conduct experiments to verify the influence of the prototype pool size (M) on the multi-frame feature enhancement module, as well as the effect of the number of frames per video segment (T) on anomaly detection performance. As shown in Fig. 4, our proposed DSTDL achieves the hightest AUC values when $M = 10$ and $T = 10$, respectively. Note that in each experiment, we vary only the parameter under study, keeping all other variables unchanged to isolate its specific impact.

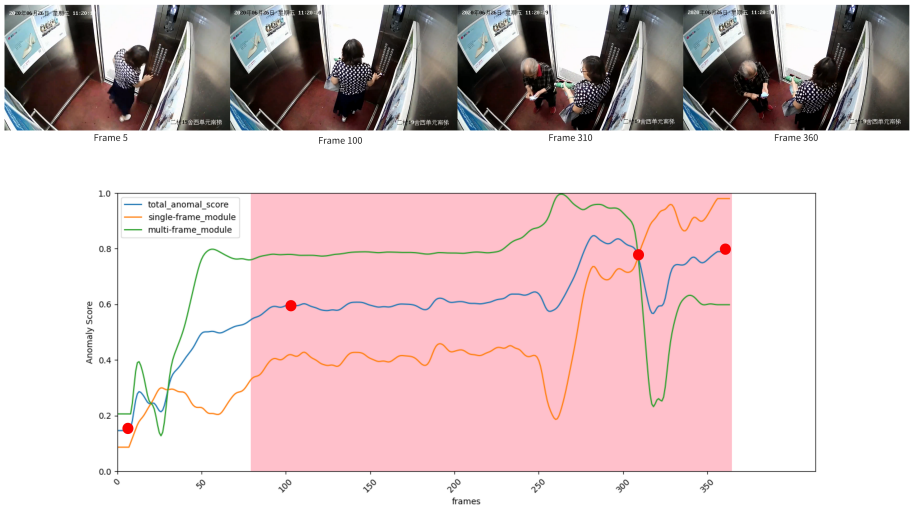

Fig. 5. Anomaly Score Curves for Video_008 on Elevator dataset. Frame 5 indicates a person entering the elevator and preparing to press the floor button, Frame 100 indicates the passanger continuously pressing the elevator button, Frame 310 indicates another person entering the elevator, and Frame 360 indicates the elevator preparing to close. The red dots on the curves indicate the sampled video frames. Pink background represents the ground truth.

4.6 Visualization Analysis

In Fig. 5, we illustrate three anomaly score curves for *Video*_008 from the Elevator dataset: single-frame reconstruction module (orange), multi-frame feature enhancement module (green), and the overall anomaly score curve (blue). *Video*_008 *presents the passenger holding the elevator door for an extended period* anomaly. While there are no apparent anomalous behaviors when observing each individual video frame separately, expanding the time window to encompass the period from someone entering the elevator until the door eventually closes reveals anomalies. From the graph, it is evident that the single-frame reconstruction module detects passanger movement effectively, yet its performance diminishes when faced with prolonged static anomalies. Conversely, our multi-frame feature enhancement module demonstrates excellent detection capabilities for such anomalies. Additional details can be found in the supplementary material.

From the visualization analysis, it is evident that the multi-frame feature enhancement module effectively detects frame-level anomalies with distant temporal dependencies. By combining both modules, anomalies spaning various temporal ranges can be identified proficiently with a single framework. Currently, we stick to the strategy of computing a weighted average to keep the overall appraoch as simple as possible. More advanced fusion strategies could be explored in furure work.

5 Conclusion

In this work, we propose a Dual-Scale Temporal Dependency Learning network for video anomaly detection, which is capable of detecting both local and remote dependency anomalies at different temporal scales. We introduce a new Elevator dataset containing various types of remote temporal dependency anomalies to validate the proposed approach. Experimental results on the self-constructed Elevator dataset and two benchmarks demonstrate the effectiveness of our proposed approach.

Acknowledge

This work was supported by NSFC under Grant 61801396 and Grant 62031023.

References

1. Chang, Y., Tu, Z., Xie, W., Yuan, J.: Clustering driven deep autoencoder for video anomaly detection. In: European Conference on Computer Vision. pp. 329–345. Springer (2020)
2. Chen, D., Yue, L., Chang, X., Xu, M., Jia, T.: Nm-gan: Noise-modulated generative adversarial network for video anomaly detection. Pattern Recognition **116**, 107969 (2021)
3. Chen, Y., Liu, Z., Zhang, B., Fok, W., Qi, X., Wu, Y.C.: Mgfn: Magnitude-contrastive glance-and-focus network for weakly-supervised video anomaly detection. In: Proceedings of the AAAI Conference on Artifical Intelligence. pp. 387–395. No. 1 (2023)
4. Feng, X., Song, D., Chen, Y., Chen, Z., Ni, J., Chen, H.: Convolutional transformer based dual discriminator generative adversarial networks for video anomaly detection. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 5546–5554 (2021)
5. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1705–1714 (2019)
6. Hao, Y., Li, J., Wang, N., Wang, X., Gao, X.: Spatiotemporal consistency-enhanced network for video anomaly detection. Pattern Recognition **121**, 108232 (2022)
7. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 733–742 (2016)
8. He, L., Zhou, Q., Li, X., Niu, L., Cheng, G., Li, X., Liu, W., Tong, Y., Ma, L., Zhang, L.: End-to-end video object detection with spatial-temporal transformers. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 1507–1516 (2021)
9. Lee, S., Kim, H.G., Ro, Y.M.: Bman: Bidirectional multi-scale aggregation networks for abnormal event detection. IEEE Transactions on Image Processing **29**, 2395–2408 (2019)

10. Li, G., Cai, G., Zeng, X., Zhao, R.: Scale-aware spatio-temporal relation learning for video anomaly detection. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
11. Li, Q., Yang, R., Xiao, F., Bhanu, B., Zhang, F.: Attention-based anomaly detection in multi-view surveillance videos. Knowledge-Based Systems **252**, 109348 (2022)
12. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection–a new baseline. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6536–6545 (2018)
13. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 3202–3211 (2022)
14. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2720–2727 (2013)
15. Luo, W., Liu, W., Gao, S.: Remembering history with convolutional lstm for anomaly detection. In: 2017 IEEE International Conference on Multimedia and Expo. pp. 439–444. IEEE (2017)
16. Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked rnn framework. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 341–349 (2017)
17. Lv, H., Chen, C., Cui, Z., Xu, C., Li, Y., Yang, J.: Learning normal dynamics in videos with meta prototype network. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 15425–15434 (2021)
18. Nguyen, T.N., Meunier, J.: Anomaly detection in video sequence with appearance-motion correspondence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1273–1283 (2019)
19. Pan, J., Chen, S., Shou, M.Z., Liu, Y., Shao, J., Li, H.: Actor-context-actor relation network for spatio-temporal action localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 464–474 (2021)
20. Park, H., Noh, J., Ham, B.: Learning memory-guided normality for anomaly detection. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 14372–14381 (2020)
21. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems **27** (2014)
22. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
23. Song, H., Sun, C., Wu, X., Chen, M., Jia, Y.: Learning normal patterns via adversarial attention-based autoencoder for abnormal event detection in videos. IEEE Transactions on Multimedia **22**(8), 2138–2148 (2019)
24. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6479–6488 (2018)
25. Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J.W., Carneiro, G.: Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4975–4986 (2021)
26. Wang, Y., Liu, T., Zhou, J., Guan, J.: Video anomaly detection based on spatio-temporal relationships among objects. Neurocomputing **532**, 141–151 (2023)

27. Yu, G., Wang, S., Cai, Z., Zhu, E., Xu, C., Yin, J., Kloft, M.: Cloze test helps: Effective video anomaly detection via learning to complete video events. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 583–591 (2020)
28. Zaheer, M.Z., Mahmood, A., Khan, M.H., Segu, M., Yu, F., Lee, S.I.: Generative cooperative learning for unsupervised video anomaly detection. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 14724–14734 (2022)