Context Recovery and Knowledge Retrieval: A Novel Two-Stream Framework for Video Anomaly Detection

Congqi Cao Member IEEE Yue Lu Graduate Student Member IEEE and Yanning Zhang Senior Member IEEE

Abstract—Video anomaly detection aims to find the events in a video that do not conform to the expected behavior. The prevalent methods mainly detect anomalies by snippet reconstruction or future frame prediction error. However, the error is highly dependent on the local context of the current snippet and lacks the understanding of normality. To address this issue, we propose to detect anomalous events not only by the local context, but also according to the consistency between the testing event and the knowledge about normality from the training data. Concretely, we propose a novel two-stream framework based on context recovery and knowledge retrieval, where the two streams can complement each other. For the context recovery stream, we propose a spatiotemporal U-Net which can fully utilize the motion information to predict the future frame. Furthermore, we propose a maximum local error mechanism to alleviate the problem of large recovery errors caused by complex foreground objects. For the knowledge retrieval stream, we propose an improved learnable locality-sensitive hashing, which optimizes hash functions via a Siamese network and a mutual difference loss. The knowledge about normality is encoded and stored in hash tables, and the distance between the testing event and the knowledge representation is used to reveal the probability of anomaly. Finally, we fuse the anomaly scores from the two streams to detect anomalies. Extensive experiments demonstrate the effectiveness and complementarity of the two streams, whereby the proposed two-stream framework achieves state-of-the-art performance on ShanghaiTech, Avenue and Corridor datasets among the methods without object detection. Even if compared with the methods using object detection, our method reaches competitive or better performance on the ShanghaiTech, Avenue, and Ped2 datasets.

Index Terms—Video anomaly detection, context recovery, knowledge retrieval, two-stream framework.

Manuscript received 15 July 2022; revised 29 December 2023 and 22 February 2024; accepted 25 February 2024. Date of publication 7 March 2024; date of current version 12 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62376217; in part by the Young Elite Scientists Sponsorship Program by the China Association for Science and Technology (CAST) under Grant 2023QNRC001; in part by the Key Research and Development Project of Shaanxi Province under Grant 2023-YBGY-240; and in part by the Young Talent Fund of Association for Science and Technology in Shaanxi, China, under Grant 20220117. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ioannis (Yiannis) Kompatsiaris. (Congqi Cao and Yue Lu contributed equally to this work.) (Corresponding author: Congqi Cao.)

The authors are with the National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, School of Computer Science, Northwestern Polytechnical University, Xi’an 710129, China (e-mail: congqi.cao@nwpu.edu.cn; zugexiaodui@mail.nwpu.edu.cn; ynzhang@nwpu.edu.cn). Digital Object Identifier 10.1109/TIP.2024.3372466

I. INTRODUCTION

VIDEO anomaly detection (VAD) is the task of detecting the events in a video that do not conform to the expected behavior [1], which has wide applications in intelligent surveillance and public security. It is an extremely challenging task for the following reasons. First, anomalous events rarely occur and their categories are agnostic and unbounded. In most practical application scenarios, we only have access to normal data, as the abnormal data is often absent. Second, video anomalies are scene-dependent [1]. For example, playing football is normal on the pitch but abnormal on the road. Third, some kinds of normal events occur frequently while some occur occasionally. If an algorithm cannot handle imbalanced data distributions effectively, it tends to treat infrequent normal events as abnormal events, resulting in false positives.

In the early works, researchers use one-class support vector machine (OC-SVM) [2], [3], [4], [5], [6], clustering [7], [8], [9], [10], a mixture of Gaussians [11], [12] and other methods to represent normal events with representative features, such as clustering centers and distribution parameters. Although these methods have the advantage of good interpretability, they lack adequate and flexible generalization of normal events. For example, the number of cluster requires manual adjustment and is typically small. With the development of deep learning, the methods based on snippet reconstruction and future frame prediction [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] have become popular. These methods train auto-encoders to reconstruct the current snippet or predict the future frame, and calculate the anomaly probability by the reconstruction or prediction error. They are good at distinguishing short-term anomalous movements, but they lack an understanding of normality. For instance, a snippet reconstruction model can accurately reconstruct the action of playing football not only on the pitch but also on the road, which makes it challenging to detect the anomalous event.

In addition, the data-driven nature of deep neural networks makes it hard to reconstruct the normal events that seldom occur. To solve the above problems, it is necessary to make full use of the knowledge from normal events to detect anomalies. For example, if we fail to find “playing football on the road” according to the knowledge about normality, we can assume that an anomalous event has occurred.
In this work, we propose a novel two-stream framework that not only can discriminate short-term abnormal motions, but also leverages the knowledge from normal events to enhance the understanding of normality. As shown in Fig. 1, our proposed two-stream framework consists of a context recovery stream and a knowledge retrieval stream. In the context recovery stream, normal motion patterns are modeled by predicting the future frame based on the input snippet. The anomaly probability of a testing event is obtained from the prediction error. In the knowledge retrieval stream, normal events are encoded as the knowledge about normality and stored in a knowledge base. The anomaly probability is obtained according to the consistency between the testing event and the knowledge. We fuse the anomaly probabilities from the two streams as the final anomaly score of the testing data.

To be specific, in the context recovery stream, we propose a novel spatiotemporal U-Net (STU-Net) for future frame prediction. This model utilizes a 3D CNN designed for action recognition as the encoder, preserving the temporal dimension in the deep layers to extract rich semantic features of the motion. To fuse the motion information, we add a temporal squeezing layer between the encoder and decoder, which also solves the inconsistency of temporal dimensions between the feature maps output by the encoder and those fed into the decoder. In this way, we can take advantage of the motion information in the input frames to predict the future frame. Besides, existing context recovery methods have the problem that the reconstruction or prediction error is proportional to the number of foreground objects [1], which can easily lead to false positives. Based on the assumption that the anomalous region causes larger context recovery error than normal regions, we propose to use the maximum patch-level recovery error in the frame instead of the frame-level error to reflect the anomaly probability, which we term as the maximum local error (MLE) mechanism. Due to the lack of validation sets in the existing datasets [29], [30], [31], [32], we use the training videos to generate pseudo anomalous samples by data augmentation to tune the hyper-parameter (i.e., the size of the patch) in MLE. The proposed MLE can partially ignore the recovery degree of the normal region, so that the recovery error of the anomalous region can be calculated accurately. Meanwhile, it has the advantage of not relying on any object detectors.

In the knowledge retrieval stream, we propose an improved learnable locality-sensitive hashing (iL²SH) based on [33] to store and retrieve knowledge about normality, which can adaptively and efficiently find the knowledge representation consistent with a testing event. Concretely, we first extract the features of the events in training videos and then encode them into hash codes via a trainable hash encoder composed of multiple parallel hash layers. The binary and real-valued hash codes are stored as key-value pairs in multiple hash tables that serve as a knowledge base. We take the mean vector of the hash codes sharing the same key as the knowledge representation of such normal events. For a testing event, the hash code is obtained through the same process, aiming to find the knowledge representation with a matching key. We calculate the anomaly probability according to the distance between the testing hash code and the retrieved knowledge representation. Compared with LLSH [33], we make the following improvements. Firstly, we improve the optimization of hash encoder. LLSH adopts MoCo [34] contrastive learning framework to train the hash encoder, where the optimization is affected by the number of negative samples, and they need to set different numbers of negative samples for different datasets. In contrast, we use a simpler Siamese network and discard negative samples, which achieves better training results with only positive samples. Secondly, the parameters in different hash layers should be as different as possible, which cannot be guaranteed in LLSH since it lacks constraints on the hash layers. We propose a mutual difference loss to ensure distinction among hash layers. This helps increase the gaps between hash layers and improves performance after optimization.

We conduct comprehensive studies across several datasets to verify the effectiveness of the proposed two-stream framework, including the ShanghaiTech [31], CUHK Avenue [30], IITB Corridor [32] and UCSD Ped2 [29] datasets. Without using optical flow and object detection, our context recovery stream surpasses the previous future frame prediction and snippet reconstruction methods [13], [14], [17]. Furthermore, the proposed iL²SH outperforms other knowledge modeling methods, such as the clustering-based CAC [8] and the nearest-neighbor-search-based Exemplar Selection [6]. Through comparing with several existing models [6], [8], [17], [20], [22], we verify that context recovery and knowledge retrieval can complement each other. In a fair comparison without additional preprocessing to extract foreground objects, our two-stream method achieves state-of-the-art performance on the ShanghaiTech, Avenue, and Corridor datasets. Compared to the methods using object detection or skeleton information, our method achieves better or competitive performance on the ShanghaiTech, Avenue, and Ped2 datasets.

We summarize our contributions as follows:

- We propose a novel two-stream framework consisting of a context recovery stream and a knowledge retrieval stream for video anomaly detection. It not only utilizes fine-level local context to detect anomalies, but also takes

Fig. 1. Overview of the proposed two-stream framework. The context recovery stream and the knowledge retrieval stream reflect the anomaly probability by the recovery error of the input and the consistency between the input and the knowledge representations about normality, respectively. The results from the two streams are fused as the final anomaly score.
full advantage of the high-level semantic knowledge of normal events to enhance the understanding of normality.

- We propose a spatiotemporal U-Net and maximum local error mechanism to respectively enhance the ability of motion modeling of the auto-encoder and the ability of error calculation in anomalous regions, which significantly improve the accuracy of context recovery.

- We propose iL²SH, which improves the optimization process of learnable locality-sensitive hashing. It can efficiently extract, store and retrieve the knowledge about normality, and detect anomalies according to the consistency between testing events and the knowledge.

- We prove that the context recovery stream and the knowledge retrieval stream are complementary for video anomaly detection by experiments. With the fusion of the two streams, our method achieves leading or competitive performance on ShanghaiTech, CUHK Avenue, IIITB Corridor and UCSD Ped2 datasets even compared with the methods using object detection.

II. RELATED WORK

A. Context Recovery Methods

The context recovery methods [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] have become the mainstream in the field of VAD in recent years. They are based on the assumption that normal events are easier to recover than abnormal events. These methods reconstruct the snippet or predict the future frame through an encoder-decoder-style generative model. For example, [13], [17], [18], [20], [22], [26] concatenate the input frames as an image, and feed it into a U-Net model to predict the next frame. However, they directly apply a 2D U-Net originally proposed for image segmentation to videos, posing challenges in effectively modeling object motion. Hence, they need to be combined with the optical flow modality [13], [18], [22] or recurrent units [16], [31]. For example, Lee et al. [16] adopt ConvLSTM to encode the spatiotemporal features in both forward and backward directions. There are some methods that adopt a 3D CNN as the encoder [35], [36], [37]. Nevertheless, these models typically employ shallow networks to avoid gradient vanishing. In contrast to the above methods, our proposed STU-Net can model the motion information without the need for additional modalities or recurrent operations. While a 3D-CNN-based U-Net exists in the field of image summary [38], it directly applies squeeze-and-excitation blocks [39] for compressing feature dimensionality. Therefore, it cannot be applied for future frame prediction, where the encoder and decoder have different numbers of frames. In contrast, we propose temporal squeezing layers in our STU-Net and solve the problem of inconsistent temporal dimensions between the feature maps of the encoder and the decoder.

Moreover, most of the methods use the entire frame as the input during both training and testing phases, while others use detected objects [4], [15], [21], [36], [40] or video patches [6], [35], [41]. As described in the Introduction section, the methods based on object detection exclusively consider foreground objects, overlooking essential scene information. Additionally, they cannot detect the anomalous objects whose categories are not covered by the pre-trained object detector. The patch-based methods face challenges in capturing the full movement of an object, since dividing a moving object into patches of the same spatial grid across time can result in visual misalignment. In our context recovery stream, we still use the entire frame as the input so that complete and long-term motions can be captured. During the testing phase, we adopt the maximum error among the patches of the recovery error map (i.e., a frame) to reveal the anomaly score. Although Nguyen and Meunier [42] also train a frame-level model and calculate normalized patch-level errors, they do not exploit an effective solution to address the problem of large errors caused by foreground objects. Their use of a fixed and small patch size for different datasets makes them susceptible to noise in the error map and incapable of handling variations in video resolutions. We propose a novel maximum local error (MLE) mechanism that utilizes the training videos to simulate anomalies and select an appropriate patch size for the dataset. The proposed MLE can effectively focus on the anomalous region and calculate a more accurate recovery error. Consequently, it alleviates the problem of recovery errors being proportional to the number of foreground objects.

B. Knowledge Retrieval Methods

Large-scale knowledge retrieval typically relies on hashing techniques. Beyond the domain of VAD, hashing is widely employed in retrieval-based tasks, such as image matching, finding similar texts and multimedia retrieval. Researchers have made efforts to improve hashing-based retrieval methods [43], [44], [45], [46]. For instance, Tian et al. [43] address the dilemma of large index sizes and hash boundary problems by organizing projected spaces with multi-dimensional indexes, thereby reducing space costs significantly. Ma et al. [44] leverage causal intervention within a deep learning framework to learn compact and robust descriptions for fine-grained image retrieval, effectively addressing issues related to quantization errors and discriminative feature extraction. A two-stage binary code refinement process aimed at reducing correlations and redundancies between hash bits is proposed in [45] to enhance discriminative power and effectiveness. Qin et al. [46] introduce an adaptive margin quadruplet loss to explore the underlying similarity relationship between image pairs, ensuring similarity preservation and discriminative hash code generation. Nonetheless, the inherent differences between VAD and those tasks such as image retrieval make it hard to directly apply these methods in the VAD domain.

In the filed of VAD, the knowledge retrieval methods explicitly extract the representations of training data as the knowledge about normality, and detect anomalies according to the consistency between the testing event and the knowledge representations. Commonly used knowledge representations include decision boundaries of OC-SVM [2], [3], [4], nearest neighbor samples [5], [6], cluster centers [7], [8], [9], [10] and probability distributions [11], [12]. For example, Ionescu et al. [4] first classify the normal data into $K$ classes by
K-means, and then use the maximum classification score from K one-versus-rest SVMs as the anomaly score. Wang et al. [8] propose a cluster attention module to map the input event into K feature spaces. For a testing sample, its regularity score is determined by the highest similarity between its K feature space representations and corresponding space centers. However, the value of K is usually too small to fully exploit the knowledge in normal data. Ramachandra and Jones [6] build a exemplar set, in which only the normal samples whose distances from stored samples in the set exceed a threshold will be added. The anomaly score is determined simply based on the distance between the testing sample and its nearest exemplar, which lacks the induction of knowledge. The normal patterns in the memory modules of context recovery methods [14], [17], [20], [21], [22] can also be regarded as knowledge representations. Limited by memory size, it is difficult to contain sufficient knowledge in the memory module.

Compared with the above methods, our proposed iL2SH can make use of the knowledge from training data and retrieve it efficiently. Even when combined with a memory-augmented context recovery model, iL2SH still brings significant improvements. Different from Lu et al. [33] who use MoCo [34] to train hash functions in their hash encoder, which is easily affected by the number of negative samples (i.e. the length of queue in MoCo), we use a Siamese network for optimization and discard negative samples. In addition, we propose a new loss which can enlarge the differences between any two hash functions.

III. PROPOSED METHOD

In this section, we first introduce the architecture of the proposed two-stream framework. Then, we illustrate the spatiotemporal U-Net (STU-Net) in our context recovery stream, followed by the introduction of maximum local error (MLE) mechanism. Next, we describe the proposed improved learnable locality-sensitive hashing (iL2SH) in the knowledge retrieval stream, which includes training hash encoder, constructing the knowledge base and retrieving knowledge. Finally, we present the fusion of anomaly scores of the two streams.

A. Two-Stream Framework

The proposed two-stream framework for video anomaly detection is shown in Fig. 1. To detect if an anomalous event occurs in a video sequence at time \( t \), the snippet containing the \( \tau \)-th frame is fed into the context recovery stream and the knowledge retrieval stream, respectively. In the context recovery stream, the input is recovered by an encoder-decoder-style future frame prediction model in our implementation. The anomaly score is calculated base on the error between the recovered frame and the ground truth frame. In the knowledge retrieval stream, the knowledge base contains the knowledge about normality extracted from training data. The normal knowledge representation consistent with the input event is retrieved from the knowledge base, and the anomaly score is determined by the distance between the knowledge representation and the event. If no knowledge representation can be retrieved, the anomaly score is assigned a high value. The anomaly scores from both streams are added together as the final anomaly score at time \( \tau \).

B. Context Recovery Stream

We propose a novel future frame prediction model named spatiotemporal U-Net (STU-Net) for the context recovery stream. STU-Net takes a snippet as the input and predicts the next frame, as illustrated in Fig. 2. In order to detect the anomaly for the current frame \( I_t \), the previous 8 frames \( I_{t-8}, I_{t-7}, \cdots , I_{t-1} \) are fed into STU-Net to generate the predicted frame \( \hat{I}_t \). We calculate the proposed maximum local error (MLE) between \( I_t \) and its ground truth \( I_t \) as the anomaly score.

1) Spatiotemporal U-Net: As shown in Fig. 2, the proposed STU-Net is a 5-level encoder-decoder with U-Net architecture. A snippet of 8 frames with a spatial resolution of 256 \( \times \) 256 is input into it. The encoder gradually reduces the spatial and temporal resolutions of the input to extract high-level semantic features, while the decoder gradually recovers the feature map by increasing the spatial resolution. To avoid the gradient vanishing problem, the feature maps of the encoder and decoder at each level are connected via a shortcut.

**TABLE I**

| Spatiotemporal U-Net | Encoder | Decoder |
|----------------------|---------|---------|
| fn1(n): \( [(n,1,3^2), (n,1,3^2), (n,1,1^2)] \) | \( f(n): \) \( [(n,1,3^2), (n,1,3^2), (4n,1,1^2)] \) | \( f(n): \) \( [(n,1,3^2), (n,1,3^2), (4n,1,1^2)] \) |
| L1 | max(1,3^2) | \( \text{fn1(64)} \times 3 \) \( (1,1^2) \times 9 \) | \( \text{L5} \) | \( \text{fn(3,1,2^T)} \) |
| \( \text{fn(3,1,2^T)} \) \( \times 2 \) \( (1,2^T) \times 9 \) | \( \text{max(2,1^2)} \times 2 \) \( (2,1^2) \times 9 \) | \( \text{L5} \) | \( \text{fn(3,1,2^T)} \) |
| \( \text{fn(3,1,2^T)} \) \( \times 2 \) \( (1,2^T) \times 9 \) | \( \text{max(2,1^2)} \times 2 \) \( (2,1^2) \times 9 \) | \( \text{L5} \) | \( \text{fn(3,1,2^T)} \) |
| \( \text{fn1(256)} \) \( \times 3 \) \( (1,2^T) \times 10 \) | \( \text{fn1(256)} \) \( \times 3 \) \( (1,2^T) \times 10 \) | \( \text{L2} \) | \( \text{fn1(256)} \) |
| \( \text{fn1(256)} \) \( \times 3 \) \( (1,2^T) \times 10 \) | \( \text{fn1(256)} \) \( \times 3 \) \( (1,2^T) \times 10 \) | \( \text{L2} \) | \( \text{fn1(256)} \) |
| \( \text{fn1(256)} \) \( \times 2 \) \( (1,2^T) \times 10 \) | \( \text{fn1(256)} \) \( \times 2 \) \( (1,2^T) \times 10 \) | \( \text{L2} \) | \( \text{fn1(256)} \) |
| \( \text{fn1(256)} \) \( \times 2 \) \( (1,2^T) \times 10 \) | \( \text{fn1(256)} \) \( \times 2 \) \( (1,2^T) \times 10 \) | \( \text{L2} \) | \( \text{fn1(256)} \) |
| \( \text{fn1(256)} \) \( \times 2 \) \( (1,2^T) \times 10 \) | \( \text{fn1(256)} \) \( \times 2 \) \( (1,2^T) \times 10 \) | \( \text{L2} \) | \( \text{fn1(256)} \) |
Different from previous works that stack input frames as an image to use 2D convolutions, we preserve the temporal dimension in the encoder, which helps to predict the future frame through the motion in previous frames. Due to the discrepancy in temporal dimension, the feature map output by the encoder cannot be directly connected to the input of the decoder at the same level. We address this issue by introducing a temporal squeezing layer (TSL) on each shortcut. The TSL fuses the features from different time steps and squeezes the temporal dimension to 1. In this way, the feature maps from the encoder and the decoder at the same level can be concatenated along the channel dimension, to serve as the input for the next level of the decoder.

The network structures of the encoder and decoder are detailed in TABLE I. We adopt the I3D network [47] with ResNet-50 backbone [48] as the encoder, and we design the decoder on our own. For the layers in the encoder, each convolution is followed by a batch normalization (BN) [49] and a ReLU activation. We remove the last ReLU activation to allow for diverse feature representations. In the decoder, the transposed convolutional layers enlarge the spatial resolution of the feature maps. The strides in regular convolutions and transposed convolutions are (1, 12) and (1, 22), respectively. Except for the last layer, each convolution is followed by a BN and a Leaky ReLU. The temporal squeezing layers are not shown in the table. There is only a convolution with a kernel size (C, 4, 12) and a stride (1, 12) in each TSL.

We minimize the mean square error (MSE) and L1 loss between the predicted frame \( \hat{I}_t \) and the ground truth \( I_t \) for training STU-Net:

\[
\hat{I}_t = \text{STUNet}(I_{t-8}, I_{t-7}, \ldots, I_{t-1}),
\]

\[
L_{FLE}(I_t, \hat{I}_t) = \|I_t - \hat{I}_t\|_F + \lambda_{L1} \times |I_t - \hat{I}_t|,
\]

where \( \lambda_{L1} \) is the weight of the L1 loss, and \( L_{FLE} \) denotes the loss of frame-level error (FLE).

The proposed STU-Net has the ability to leverage the temporal information of the input snippet to predict the future frame. By using the frame-level error (i.e., Eq. (2)) as anomaly score, it achieves comparability to existing frame prediction methods that integrate optical flow to complement motion information.

2) Maximum Local Error: For the VAD task, we expect the context recovery model to accurately predict normal regions while exhibit inaccuracies in abnormal regions. However, due to factors such as a nonstatic background, large number of foreground objects, image noise and other variables, achieving fully accurate predictions for normal regions in the subsequent frame is challenging in any scenario. The inaccurate prediction of normal regions will lead to higher errors and false positives.

To pay more attention to the anomalous region and disregard normal regions, we propose a maximum local error (MLE) mechanism in the anomaly detection process, as shown in the red box in Fig. 2. We use a square sliding window with a fixed size to calculate a number of local errors on the frame-level error map. Based on the hypothesis that the error in anomalous regions is greater than that in normal regions, we choose the maximum local error as the anomaly score. The proposed MLE can be implemented by average pooling. Mathematically, MLE is denoted as:

\[
MLE(I_t, \hat{I}_t)_{k,s} = \text{Max}(\text{Avgpool}_{k,s}(\|I_t - \hat{I}_t\|_F^2 + \lambda_{L1} \times |I_t - \hat{I}_t|)),
\]

where \( k \) is the size of the sliding window, and \( s \) is the stride. The hyper-parameter \( k \) is dataset-dependent and can be determined by a validation set. However, as most VAD datasets do not have validation sets, we propose an alternative solution outlined in Algorithm 1. Specifically, we utilize training videos to simulate anomalies through data augmentation. We first spatially flip and rotate all the frames in a video at a random angle to create a new and unseen video for the model.

To generate an anomalous frame, we then blend the current frame with its future frame by averaging. We use the simulated abnormal videos for video anomaly detection, and select an appropriate \( k \) from the predefined set \( K = \{k_1, k_2, \ldots, k_n\} \) according to the evaluation metric.

The proposed MLE mitigates the interference from recovery errors in normal regions. With MLE, our STU-Net can reach similar or superior performance compared with the methods using object detectors.

C. Knowledge Retrieval Stream

The knowledge retrieval stream is proposed for enhancing the understanding of normality. We aim to adaptively construct a knowledge base, store the knowledge about normality from training data, and efficiently retrieve the knowledge to detect anomalies. To this end, we propose an improved learnable locality-sensitive hashing (\( \text{IL}^2\text{SH} \)). First, we take hash functions as learnable parameters and embed them into a neural network for optimization using the training data.
Algorithm 1 The Maximum Local Error Mechanism

Require: \( V = \{ I_i \}_{i=1}^{N} \): a training video of \( N \) frames
\( nseg \): the number of anomalous segments in a video (default: 1)
\( ratio \in (0, 1) \): the ratio of anomaly frames in each segment (default: 0.5)
\( offset \): the offset index of the frame to be averaged with current frame (default: 2)

Ensure: an anomalous video \( \tilde{V} = \{ \tilde{I}_i \}_{i=1}^{N} \), and a list of labels \( \{ L_i \}_{i=1}^{N} \) indicating if the frame is normal (0) or not (1)

function rotate(I): rotate I with a random angle \( \alpha \in [2^\circ, 5^\circ] \)
function flip(I): horizontally flip I

1) Training iL\(^2\)SH: The structure of iL\(^2\)SH is illustrated in Fig. 3, which mainly consists of an event encoder and a hash encoder. The event encoder outputs a feature vector to represent the event in the input snippet. In this work, it is the I3D network as introduced in the context recovery stream. The hash encoder contains a group of parallel hash layers, each of which maps the feature to a real-valued hash code and serves as a hash function and we aim to optimize it to generate similar hash codes for similar features.

As shown in Fig. 3, iL\(^2\)SH is embedded as a branch of the Siamese network, where the two branches share the same parameters. We feed a snippet \( S \) into one of the branches. At the same time, a similar snippet \( S_{t+\Delta t} \) which is temporally close to \( S \) is sampled from the same video and fed into the other branch. Each branch outputs a group of short hash codes, which are concatenated as a compact long hash code. We minimize the cosine distance between the concatenated hash codes of the two branches:

\[
L_c(l_t, l_{t+\Delta t}) = 1 - \frac{l_t}{\|l_t\|} \cdot \frac{l_{t+\Delta t}}{\|l_{t+\Delta t}\|} \tag{4}
\]

where \( l_t \) and \( l_{t+\Delta t} \) is the concatenated hash codes corresponding to the input snippets \( S_t \) and \( S_{t+\Delta t} \).

Snippets \( S_t \) and \( S_{t+\Delta t} \) form a positive pair and the distance between them is minimized by Eq. (4). We do not take the snippets from different videos as negative pairs and maximize the distances between them, since they provide little improvement for training and the number of negative pairs is sensitive to the scale of dataset. Instead, we expect different hash layers to produce as different hash codes as possible to capture diverse event patterns and construct distinct hash tables. Therefore, we propose a mutual difference loss that amplifies the difference between the hash codes output by a hash encoder:

\[
L_m([h_1, h_2, \cdots, h_B]) = \frac{2}{RB(B-1)} \sum_{j=1}^{B} \sum_{j'=j+1}^{B} h_j \cdot h_{j'} \tag{5}
\]

where \( h_j \in \mathbb{R}^R \) represents a hash code of length \( R \), and \( B \) denotes the number of hash codes.

We average the mutual difference losses of the two branches. The overall loss for training iL\(^2\)SH is denoted as:

\[
L_{total} = L_c + \frac{\lambda_m}{2} (L_m^{(1)} + L_m^{(2)}) \tag{6}
\]

where \( L_m^{(i)} \) \((i \in \{1, 2\})\) denotes mutual difference loss of the \( i \)-th branch, and \( \lambda_m \) is the weight.

When the training process is finished, the hash layers have an enhanced ability to map similar events to similar hash codes. Subsequently, we can use iL\(^2\)SH for constructing a knowledge base in the next step.

2) Constructing Knowledge Base: Our purpose is to construct a knowledge base which encompasses the knowledge representations about normality obtained from the training
Algorithm 2 The Process of Constructing Knowledge Base

Require: \( \{S_i\}_{i=1}^N \): \( N \) training snippets

\( Enc(\cdot): \text{iL}^2\text{SH}, \) which maps a snippet to \( B \) hash codes

\( \{H_b[\text{key}] = (\text{cnt, val})\}_b^B \): \( B \) empty hash tables

Ensure: hash tables \( \{H_b\}_b^B \) that stores the hash codes

function \( \text{BIN}(a) \) \( \triangleright \) Binary function

for all \( a_{<\cdots<} \) of the \( i \)-th bit in \( a \) do

\( a_{<\cdots<} \leftarrow 0 \) if \( a_{<\cdots<} < 0.5 \) else 1

end for

return \( a \)

end function

for each \( i \in \{1, 2, \cdots, N\} \) do

\( \{h_b\}_b^B \leftarrow Enc(S_i) \)

for each \( b \in \{1, 2, \cdots, B\} \) do

\( k \leftarrow \text{BIN}(h_b) \)

if \( k \) exists in \( H_b.\text{key} \) then

\( h_b[\text{key}].\text{val} \leftarrow (H_b[\text{key}].\text{val} \times H_b[\text{key}].\text{cnt} + h_b) / (H_b[\text{key}].\text{cnt} + 1) \)

\( H_b[\text{key}].\text{cnt} \leftarrow H_b[\text{key}].\text{cnt} + 1 \)

else

\( H_b[\text{key}].\text{cnt} \leftarrow 1 \)

\( H_b[\text{key}].\text{val} \leftarrow h_b \)

end if

end for

end for

data. To this end, we first map each training event to a group of hash codes, and then store each hash code in its respective hash table. Each hash table is composed of a number of buckets in the form of key-value pairs. The keys are the binary vectors corresponding to the hash codes, and the values are the mean vectors of those hash codes sharing the same binary key. A detailed process of constructing the knowledge base is summarized in Algorithm 2.

We take one hash layer as an example to explain the process of constructing a hash table, which is shown in Fig. 4. The normal events \( S_1, S_2 \) and \( S_3 \) generate three real-valued hash codes via the same hash layer. Then we use the binary function in Algorithm 2 to derive a binary key for each hash code. The hash code of \( S_1 \) is stored in the first bucket with its binary key “0110”. Since \( S_2 \) and \( S_3 \) share the same binary key “1101”, we calculate the mean vector of the two hash codes, which is then stored in the second bucket. It indicates that the two similar events \( S_2 \) and \( S_3 \) are implicitly grouped together. They belong to the same type of event, and this type of event can be represented by the corresponding value (i.e., the vector [0.8, 0.7, 0.8, 0.6]) in the hash table. In this way, similar events are abstracted into a knowledge representation and stored as a vector in a bucket. Meanwhile, each knowledge representation can be retrieved efficiently via the binary key, which will be introduced in the following step.

3) Retrieving Knowledge: Given a testing snippet, we aim to discriminate whether it is consistent with the normal knowledge and estimate an anomaly probability. Therefore, we try to retrieve a knowledge representation from the knowledge base and use the distance between it and the retrieved knowledge representation as the anomaly score. Algorithm 3 describes the process of retrieving knowledge. For a testing snippet \( S_t \), we first obtain a group of hash codes and their corresponding binary keys. Then, we retrieve a bucket from the corresponding hash table, and calculate the L2 distance between the testing hash code and the only vector in the retrieved bucket. The minimum distance from all the hash tables is taken as the anomaly score for \( S_t \). By using the decision from multiple hash tables, we can find the most relevant knowledge representation and hence reduce false alerts.

Nevertheless, it cannot be guaranteed that a binary key will necessarily be found in the corresponding hash table. In this case, it is intuitive that if a test event \( S_t \) fails to find a matching normal knowledge in the knowledge base, it is highly likely that the test event is anomalous. Therefore, we treat \( S_t \) as an anomalous event and assign it with a pre-defined high anomaly score \( P_{max} \), which is the maximum L2 distance between any
two hash codes:

\[ p_{\text{max}} = \sqrt{R}, \]  

where \( R \) is the length of a hash code.

In summary, we construct a group of hash tables as the knowledge base, retrieve a knowledge representation (i.e., the mean vector of hash codes) from each hash table, and compare the testing event with the retrieved knowledge representations to detect anomalies. Our strategy for consistency measure can be regarded as selecting a number of normal events that are close to the test event, since a nearest neighbor (i.e., a matched bucket) contains a number of training events. However, the following mechanisms and characteristics of our approach make it more effective and efficient than a simple k-nearest neighbors (KNN). Firstly, our method consolidates event representations within the same bucket by average, thereby enhancing robustness. This can be seen as an adaptive strategy for adjusting the \( K \) value in KNN. Secondly, we employ learnable feature mappings, i.e., hash layers, which can optimize the representations of events according to the data distribution. Thirdly, we utilize multiple hash layers and hash tables, and propose a mutual difference loss to enhance the diversity of event representations. Lastly, the computational cost of hash mapping is minimal, and our strategy of storing hash codes rather than feature vectors reduces storage costs significantly, thus enhancing the feasibility of our method in practical scenarios.

D. Fusion of Two Streams

Now, we can obtain the anomaly scores from the context recovery stream and the knowledge retrieval stream. We fuse the results from the two streams by late fusion:

\[ p_{\text{fuse}} = \lambda_{cr} p_{cr} + \lambda_{kr} p_{kr}, \]  

where \( p_{cr} \) and \( p_{kr} \) respectively denote the anomaly score obtained from the context recovery stream and the knowledge retrieval stream, and \( \lambda_{cr} > 0 \) and \( \lambda_{kr} > 0 \) are the corresponding weights.

The context recovery stream has a good ability to detect short-term rapid motion anomalies, and the knowledge retrieval stream can leverages high-level semantic knowledge about normality to detect anomalous events. The anomaly detection results from the two streams can complement each other, thereby improving the performance for VAD.

IV. EXPERIMENTS

In this section, we conduct comprehensive experiments to verify the effectiveness of our proposed two-stream framework and compare it with other methods. We begin by introducing the datasets, evaluation metric and implementation details in our experiments. Subsequently, we carry out a detailed ablation study to investigate the effects of different components proposed in our method. Then, we study the complementarity of context recovery and knowledge retrieval by fusing different types of existing methods. Afterward, we visualize and analyze the effect of MLE and fusion of two streams. Next, we compare our two-stream framework with existing methods. After that, we report the running time of our method, followed by two representative failure cases. Finally, we provide an analysis of the robustness of our proposed approach.

A. Datasets

We evaluate our method on four commonly used datasets, i.e., ShanghaiTech [31], CUHK Avenue [30], IITB Corridor [32] and UCSD Ped2 [29], which are shown in TABLE II. ShanghaiTech is an extremely challenging dataset, since there are 13 separate scenes and the anomalous events vary widely. We need to train only one model to detect the abnormal events in all scenes. Although Avenue is a single-scene dataset, the complex pedestrian movements in the background make it challenging to detect anomalies. Corridor is a newly proposed challenging dataset with high resolution. It contains group-level anomalies, e.g., protest, which are absent in other datasets. Ped2 is a small-scale dataset and all of the abnormal events are related to objects. All the frames in Ped2 are grayscale with low resolution. We mainly use ShanghaiTech and Avenue for ablation studies, and compare our method with other methods on all four datasets.

B. Evaluation Metric

We adopt the most widely used area under curve (AUC) as the evaluation metric for anomaly detection performance. AUC is computed by the area under the receiver operating characteristic (ROC) curve, which is drawn by false positive rates and true positive rates with changing the threshold of anomaly scores. In order to compare with existing methods fairly, we adopt both micro-AUC and macro-AUC following [40]. Micro-AUC is obtained by concatenating all frames in a dataset as a single video and calculating the AUC. Macro-AUC is the average AUC of all videos. In ablation studies, we only report micro-AUC, which is adopted in most previous works.

C. Implementation Details

The default settings in our experiments are introduced as follows. In the context recovery stream, the temporal sampling rate is set to 2, and the frames are resized to 256 × 256 pixels before being fed into STU-Net. To mitigate the serious distortion caused by resizing, we manually crop three fixed square regions along the corridor.\(^1\) The weight of L1 loss \( \lambda_{L1} \) in Eq. (2) is set to 1. The predefined sizes of sliding windows (i.e., \( Ks \)) are \( \{2^n\}_{n=4}^{8} \), with the final sizes used in our experiments on ShanghaiTech, Avenue, Corridor and Ped2 being 128, 64, 256 and 32, respectively. In the knowledge retrieval stream, each input snippet consists of 8 frames. The temporal sampling rates are set to 8 for ShanghaiTech and Corridor, and 4 for Avenue and Ped2. We use \( B = 8 \) hash layers in the hash encoder, and the length of each hash code \( R \) is 32. The weights \( \lambda_m, \lambda_{cr} \) and \( \lambda_{kr} \) are set to 0.64, 1 and 1, respectively. The event encoder I3D is pre-trained on Kinetics-400 dataset [47], [51] and frozen during training. Following

\(^1\)The left-top corners and widths are \((x, y, w) = \{(320, 184, 896), (576, 96, 412), (672, 0, 256)\}. 
previous works [4], [18], [20], [36], [40], we normalize the anomaly scores and apply a Gaussian filter to smooth them. All experiments are conducted using two Nvidia Tesla V100 GPUs with PyTorch [52]. Further details can be found in our code repository: https://github.com/zugexiaodui/TwoStreamUVAD.

D. Ablation Study

We conduct ablation experiments on the proposed two-stream framework to study the effect of different components. The results on the ShanghaiTech and Avenue datasets are presented in TABLE III. In the context recovery stream, we adopt a 2D U-Net as the baseline, in which the encoder is replaced with a ResNet-50 [48] network and other layers keep the same as those in STU-Net. The encoder is pre-trained on ImageNet [53] and frozen during training. Despite the proposed STU-Net has better performance than the 2D U-Net with pre-training, it can use the parameters trained on the Kinetics-400 dataset to further improve the performance. In the knowledge retrieval stream, iL2SH without training is adopted as the baseline. In the experiment where iL2SH is trained with negative samples (i.e. “w/ neg.”), the videos of negative instances differ from those of positive instances. We take the negative cosine distance (i.e. negative value of Eq. (4)) between negative pairs as the loss function and set its weight to 0.5 to be added to the loss of positive pairs $L_c$.

Comparing experiment (abbr. exp.) 2 with exp. 1, it can be seen that our proposed STU-Net, which can leverage the motion of input snippet, outperforms the 2D U-Net by about 2% and 3% on ShanghaiTech and Avenue, even if the encoder of STU-Net is not pre-trained. Exp.3 indicates the I3D encoder in our STU-Net can also benefit from the learned representation in action recognition, providing an advantage compared to other models. Equipped with MLE, the performances of U-Net and STU-Net improve by 2%–4% in exp. 4 and exp. 5, demonstrating the effectiveness of the proposed MLE. In exp. 8, our iL2SH trained without negative pairs boosts the performance of the basic iL2SH in exp. 6 by 7.8% and 3.5% on ShanghaiTech and Avenue, respectively. It also surpasses the iL2SH trained with negative samples in exp. 7 by 0.7%. With the proposed mutual difference loss in exp. 9, the results of iL2SH on ShanghaiTech and Avenue increase by 1.4% compared with exp. 8 which does not have a constraint on the difference of hash layers. Moreover, we re-implement LLSH [33] on ShanghaiTech and Avenue datasets. The micro-AUCs of LLSH on the two datasets are 78.7% and 86.3%, which are inferior to our iL2SH by about 2%. From exp. “1+6”, we can see that the fusion of two baseline models can bring a significant improvement of 3%–5% compared with a single model. Through fusing the two streams proposed in this work, our two-stream framework achieves the best results on both datasets in exp. “5+9”.

E. Complementarity Study

To verify the complementarity of context recovery and knowledge retrieval, we re-implement several recent methods, each of which can be generalized as a kind of context recovery or knowledge retrieval method, for a thorough fusion study. The results on ShanghaiTech and Avenue datasets are shown in Fig. 5 in the form of heat maps. Since the weights for fusing any two methods are both set to 1, which means the same pair of methods produces the same fused anomaly score, the result matrix in each heat map is symmetric. In addition to our STU-Net, the context recovery methods include MPN [20], MNAD [17] and AMMC [22], which are state-of-the-art memory-augmented context recovery models. We follow their official codes for re-implementation. The re-implemented knowledge retrieval methods are CAC [8] and Exemplar Selection [6] (abbr. Exemplar), which have been introduced in Related Work. For these two methods, we use the same pre-trained I3D encoder as in iL2SH for feature extraction. In CAC, we freeze the pre-trained feature extractor and only train the cluster attention module instead of training the whole network. We report the result under the setting of 16 clusters since it achieves the best performance. As to Exemplar, we adopt MSE to measure the distance between two samples, and the distance thresholds for constructing exemplar sets are set to 150 and 60 for the ShanghaiTech and Avenue datasets, respectively. We take the average distance between the testing sample and its 8 / 64 nearest exemplars for ShanghaiTech / Avenue as the anomaly score, which achieves the best performance compared with other settings.

From the heat maps in Fig. 5, we can see that the fusion of two context recovery methods or two knowledge retrieval methods does not yield a significant improvement. For example, in the result matrix on the ShanghaiTech dataset, by fusing MPN (73.1%) and AMMC (73.7%), the AUC (74.2%) only increases by 0.5% compared with the higher performance between MPN and AMMC (i.e. 73.7%). However, fusing two different types of methods can generally bring a noteworthy improvement, even though the context recovery methods are equipped with memory modules. For example, fusing CAC and AMMC leads to an improvement of 3.5% (75.8% + 73.7% → 79.3%) on ShanghaiTech. In some cases, the fusion of a context recovery method and a knowledge retrieval

| Dataset         | Year | Training videos / frames | Testing videos / frames | Scenes | Resolution | Abnormal events                  |
|-----------------|------|--------------------------|-------------------------|--------|------------|----------------------------------|
| UCSD Ped2 [29]  | 2010 | 16 / 2.6k                | 12 / 2.0k               | 1      | 360×240    | Bikers, carts and skaters       |
| CUHK Avenue [30] | 2013 | 16 / 15k                 | 21 / 15k                | 1      | 640×360    | Throwing object, wrong direction, running, etc. |
| ShanghaiTech [31]| 2017 | 330 / 275k               | 107 / 43k               | 13     | 856×480    | Bikers, loitering, fighting, vehicles, etc. |
| ITB Corridor [32]| 2020 | 208 / 302k               | 150 / 182k              | 1      | 1920×1080  | Protest, playing with ball, unattended baggage, etc. |
TABLE III
Effect of Different Components. For convenience, we add indices to refer to different ablation study settings. The addition of two indices represents the combination of two corresponding experiments. “w/ neg.” means that IL^2SH is trained with negative pairs. “w/ Lcon” denotes IL^2SH is trained with our proposed mutual difference loss.

| Index | Context recovery stream | Knowledge retrieval stream | Micro-AUC (%) |
|-------|-------------------------|---------------------------|---------------|
|       | U-Net | STU-Net | Pre-training | MLE | IL^2SH | Training | w/ neg. | w/ Lcon | ShanghaiTech | Avenue |
| 1     | ✓     | ✓       | ✓            |      |        | 72.5     | 82.0     |          |          |
| 2     | ✓     | ✓       | ✓            |      |        | 74.4     | 84.8     |          |          |
| 3     | ✓     | ✓       | ✓            |      |        | 75.3     | 85.1     |          |          |
| 4     | ✓     | ✓       | ✓            |      |        | 75.8     | 85.1     |          |          |
| 5     | ✓     | ✓       | ✓            |      |        | 79.7     | 87.2     |          |          |
| 6     | ✓     | ✓       | ✓            |      |        | 71.8     | 83.2     |          |          |
| 7     | ✓     | ✓       | ✓            | ✓    | ✓      | 78.9     | 86.0     |          |          |
| 8     | ✓     | ✓       | ✓            |      | ✓      | 79.6     | 86.7     |          |          |
| 9     | ✓     | ✓       | ✓            |      | ✓      | 81.0     | 88.1     |          |          |

Fig. 5. Heat maps of fusing different methods on the ShanghaiTech and Avenue datasets. The micro-AUCs of the basic methods which are not fused with others are shown on the diagonal. An off-diagonal value denotes the AUC obtained by fusing the methods corresponding to its column and row. The best results are highlighted in bold. Both matrices are symmetric and we mark each result only once for conciseness.

TABLE IV
Fusion of Two Streams Which Have the Same Temporal Sampling Rate (=4)

| Method       | ShanghaiTech | Avenue | Ped2 |
|--------------|--------------|--------|------|
| STU-Net      | 77.3         | 84.6   | 90.0 |
| IL^2SH       | 78.7         | 88.1   | 91.3 |
| Two-Stream   | 81.1         | 88.8   | 93.4 |
| Improvement  | 2.4          | 0.7    | 2.1  |

To clearly analyze the effect of fusing different types of methods, we calculate the average improvement of each fusion type based on the results in Fig. 5. The performance variations are shown in Fig. 6, where the fusion types include two context recovery methods (CR-CR), two knowledge retrieval methods (KR-KR) and a context recovery method with a knowledge retrieval method (CR-KR). It can be seen that the fusion of context recovery methods and knowledge retrieval methods can bring the highest improvement on both datasets, which significantly exceeds the fusion of two identical types of methods by more than 1.7%, demonstrating the complementarity of context recovery and knowledge retrieval. Particularly, when STU-Net and IL^2SH are fused, the results are the highest on both ShanghaiTech and Avenue datasets as displayed in Fig. 5.

Furthermore, we conduct an experiment where both STU-Net and IL^2SH have the same temporal sampling rate, to verify that the improvement of fusing a context recovery stream and a knowledge retrieval stream is not caused by different temporal scales. As shown in TABLE IV, the sampling rates of STU-Net and IL^2SH are both set to 4. Although this setting results in lower performance compared with the default setting, the improvements on the ShanghaiTech, Avenue and Ped2 datasets...
are still significant. This experiment confirms that fusing the context recovery method and knowledge retrieval method which have the same temporal scale can bring improvement to the fusion. Conversely, even if two methods of the same type have different temporal scales, their fusion does not enhance performance. For example, although the context recovery methods STU-Net and AMMC on the Avenue dataset in Fig. 5 have different temporal sampling rates (2 and 1), the result of fusion is lower than STU-Net by 0.4%. Fusing streams with different temporal sampling rates typically yields improvements in action recognition. However, this is not the case in our VAD experiments. The reason is that the temporal scales of these methods are quite close (temporal sampling rates of 1 and 2), making it difficult to capture features at different temporal scales. This phenomenon demonstrates that the key factor contributing to the improvement in the fusion of two streams is not the temporal scale but rather the “functionality” (i.e., context recovery and knowledge retrieval) of the two streams.

To sum up, context recovery methods and knowledge retrieval methods can complement each other and bring significant improvement, while the same type of methods cannot. Among the fusions of different methods, our proposed two-stream model consisting of STU-Net and iL^2SH achieves the best performance, demonstrating the superiority of the proposed method.

F. Visualization and Analysis

We visualize several testing samples and anomaly scores to analyze the effects of MLE and fusion of STU-Net and iL^2SH in this section.

1) Maximum Local Error: Fig. 7 displays the AUCs of different sizes of the sliding window (i.e., $K$) in MLE on testing data and pseudo anomalous data simulated by training videos. On both ShanghaiTech and Avenue datasets, it can be seen the AUC curves exhibit similar trends for the two types of data. On ShanghaiTech dataset, the highest AUC on the pseudo anomalous data is achieved when $K = 128$, according to which we set $K$ to 128 for the testing data and achieves the best performance. Similarly, on the Avenue dataset, the optimal performance is obtained when $K = 64$, which demonstrates the effectiveness of simulating anomalies.

Two examples of frames are shown in Fig. 8, from which we can see that MLE accurately finds the anomalous region without the use of any object detection algorithms. In the first row of Fig. 8, the red skirt is predicted to be green. We attribute this to a bias in the frame prediction model towards green induced by the prevalence of green elements such as trees, grass, and ponds in the natural environment. To quantitatively study how MLE affects anomaly scores, we calculate the score gaps of frame-level error (FLE) and MLE, as shown in Fig. 9. The score gap represents the difference value between the average scores of anomalous frames and normal frames. It reflects the ability to discriminate between normality and anomaly, and is expected to have a higher value. Fig. 9 shows that MLE can increase the score gap, thus improving the ability to detect anomalies in videos.

2) Fusion of Two Streams: We illustrate the anomaly score curves for each stream and the fusion of the two streams in Fig. 10 to explain how the two streams complement each other. For example, in video “04_0001” from ShanghaiTech, iL^2SH and STU-Net output relatively low anomaly scores in the first and second abnormal periods, respectively. However, the fusion of the two streams produces high anomaly scores in both periods, hence improving the AUC by 2.7% in this video. The score gaps are displayed in Fig. 11, which shows that the fusion of the two streams increases the score gap. Therefore,
G. Comparison With Existing Methods

The comparison of different methods on ShanghaiTech [31], Avenue [30], Corridor [32] and Ped2 [29] datasets is presented in TABLE V. We report both micro-AUC and macro-AUC for each method if available, and the best results are highlighted in bold text. The results marked with † are implemented by Rodrigues et al. [32] because there are no official results. In LLSH [33] and our two-stream framework, the results on Corridor are under the setting of manually cropped regions, as mentioned in Implementation Details, and are marked with ‡. We mark the methods using object detection (w/ obj.) since anomalies in most datasets are closely related to objects in the current stage of video anomaly detection. Although DMAD [60] does not directly employ an object detector to detect objects, it utilizes additional preprocessing to extract and remove the static background, thereby equivalent to performing object detection. The best results of the methods without using object detection are underlined.

In a fair comparison without additional preprocessing to extract foreground objects, our two-stream method achieves state-of-the-art performance on the ShanghaiTech, Avenue, and Corridor datasets in both micro-AUC and macro-AUC metrics. Especially, it is worthy noting that the micro-AUC of our method on the ShanghaiTech dataset is higher than other two outstanding methods without using object detection, i.e. CAC [8] and VADB [26], by 4.4% and 5.5%. Even though compared with the methods which utilize object detection [4], [15], [21], [36], [40], [59], [60], [61], [63], [64], [65] or skeleton information [32], [59], [62], our method achieves competitive or better performance on the ShanghaiTech, Avenue, and Ped2 datasets. Among which, the macro-AUC metric of our method is almost consistently the highest, while the micro-AUC metric is approximately 2% to 3% lower than STG-NF [62], AMSCR [61] and DMAD [60] methods. We also experiment on Corridor without manually selecting regions, as shown in TABLE VI. Our two-stream framework consistently achieves the best performance compared with other methods [13], [26], [32], [54], [59]. While the performance of our model on the Ped2 dataset may not be the top, there is potential for improvement by incorporating object detection and optical flow in the future, since all the anomalies in Ped2 are related to objects.

To qualitatively assess the effectiveness of different methods in detecting various types of anomalies, we conduct observations of the methods presented in Fig. 5 using test videos from the ShanghaiTech dataset. Furthermore, we reviewed the analysis of the other existing methods presented in their papers. A qualitative analysis of different types of methods is summarized as follows. (1) Context recovery methods (e.g., STU-Net, MPN [20], AMMC [22]) excel in detecting rapid anomalous movements, such as biking and sudden sprints. However, their ability to detect anomalies with long-distance trajectories or appearance anomalies is weak. (2) Knowledge retrieval methods (e.g., iL2SH, CAC [8], Exemplar [61]) exhibit a good capability for detecting appearance anomalies (e.g., baby strollers, slow-moving cars) and a reasonable capability for short-term rapid motion anomalies. However, their performance in detecting anomalies with long-distance trajectories is suboptimal. (3) Skeleton-based methods (e.g., MTP [32], HSTGCNN [59], STG-NF [62]) constitute a distinct category of methods. They perform well in detecting anomalies related to specific behavior categories, such as jumping and falling. Notably, MTP specifically models long-distance trajectories and exhibits commendable capability in detecting long-distance trajectory anomalies. Nonetheless, such methods cannot directly detect appearance anomalies like cars or baby strollers. Our proposed two-stream framework integrates the advantages of both context recovery and knowledge retrieval methods. The complementarity of the two streams enables the detection of diverse types of anomalies, thereby leading to superior performance in our approach.

H. Running Time

We analyze the inference time of our method on the ShanghaiTech [31] dataset using two Nvidia Tesla V100 GPUs and 12 Intel Xeon Silver 4214R cores. As shown in TABLE VII, we report the running time for inferring one frame and the corresponding FPS. As the two streams are independent, we first run each stream independently using one GPU. To study the running speed of our proposed techniques, we separate the context recovery stream into STU-Net and maximum local error components, and separate the knowledge retrieval stream into I3D feature extractor and iL2SH components. Finally, we run the two streams simultaneously on two GPUs and report the entire running time in the last row. It can be seen that the proposed STU-Net in the context recovery stream and the feature extractor in the knowledge retrieval stream occupy the majority of the running time, but each of the components still achieves over 20 FPS. The proposed
TABLE V

| Method                        | ShanghaiTech Micro | ShanghaiTech Macro | Avenue Micro | Avenue Macro | Corridor Micro | Corridor Macro | Ped2 Micro | Ped2 Macro |
|-------------------------------|--------------------|--------------------|--------------|--------------|----------------|----------------|------------|------------|
| FFP [13]                      | 72.8               | 84.9               | -            | 64.7†        | -              | 95.4           | -          | -          |
| MemAE [14]                    | 71.2               | 83.3               | -            | -            | -              | 94.1           | -          | -          |
| OADA [4]                      | ✓                  | -                  | 84.9         | 90.4         | -              | -              | 97.8       | -          |
| MPED-RNN [54]                 | 73.4               | -                  | -            | 64.3†        | -              | -              | -          | -          |
| AMC [42]                      | -                  | -                  | 86.9         | -            | -              | 96.2           | -          | -          |
| BMAN [16]                     | 76.2               | -                  | 90.0         | -            | -              | 96.6           | -          | -          |
| r-GAN [55]                    | 77.9               | -                  | 85.8         | -            | -              | 96.2           | -          | -          |
| GEPIC [9]                     | 76.1               | -                  | -            | -            | -              | -              | -          | -          |
| MNAD [17]                     | 70.5               | -                  | 88.5         | -            | -              | -              | -          | -          |
| MTP [32]                      | ✓                  | 76.0               | 82.9         | 67.1         | -              | -              | -          | -          |
| Ada-Net [56]                  | 70.0               | -                  | 89.2         | -            | -              | 90.7           | -          | -          |
| CAC [8]                       | 79.3               | -                  | 87.0         | -            | -              | -              | -          | -          |
| DeepOC [37]                   | ✓                  | 74.8               | 89.6         | -            | -              | 96.9           | -          | -          |
| VEC-AM [15]                   | ✓                  | 73.6               | 86.8         | -            | -              | 95.4           | -          | -          |
| Multispace [57]               | 73.7               | -                  | 86.6         | -            | -              | 96.6           | -          | -          |
| AMMC-Net [22]                 | ✓                  | 73.2               | -            | 86.3         | -              | 95.6           | -          | -          |
| MSBNet [24]                   | ✓                  | 82.7               | 93.2         | 90.4         | -              | 97.8           | 99.7       | -          |
| SSMTL++ [65]                  | ✓                  | 83.7               | -            | 90.5         | 91.6           | 92.5           | -          | -          |

maximum local error and IL2SH are extremely lightweight and their running speeds reach over 250 FPS. When we run the two streams simultaneously, the running speed achieves 18.5 FPS, which is applicable for some scenarios where the speed requirement is not strict. When only one GPU is used for simultaneously running two streams, the FPS decreases to 10.7.

By contrast, the inference speed of four representative methods HF2-VAD [21], VABD [26], Jigsaw [63] and SSMTL++v2 [65] are 10 FPS, 63 FPS, 28 FPS and 18.8 FPS. Compared with the frame-level method VABD, our proposed method is much slower but achieves significantly better performance (+5.5%). Compared with the other three object-detection-based methods, our approach is faster than HF2-VAD and slightly slower than the other two. The primary reason is

TABLE VI

| Method          | Micro | Macro |
|-----------------|-------|-------|
| STU-Net (ours)  | 79.7  | 90.8  |
| IL2SH (ours)    | 81.0  | 90.8  |
| Two-stream (ours) | 83.7  | 90.8  |

TABLE VII

| Method          | Components | Time (s) | FPS  |
|-----------------|------------|----------|------|
| Only context    | STU-Net    | 0.0489   | 20.5 |
| recovery stream | Maximum Local Error | 0.0039 | 256 |
| Total           |            | 0.0527   | 20.0 |
| Only knowledge  | Feature extractor (3D) | 0.0303 | 33.0 |
| retrieval stream| IL2SH      | 0.0034   | 294  |
| Total           |            | 0.0337   | 29.7 |
| Simultaneous two streams | - | 0.0542 | 18.3 |
that the I3D model serving as the backbone for both streams is heavyweight. However, we argue that our method does not rely on object detectors, and maintains a stable inference speed even in the presence of dense objects in the frame, whereas Jigsaw and SSMTL++ will require significantly more inference time. For our model, we can further boost the inference speed by incorporating a more modern and lightweight backbone.

I. Failure cases

We analyze the anomaly detection performance of our method on the ShanghaiTech [31] and Avenue [30] datasets, and showcase two representative failure cases as illustrated in Fig. 12. The first row displays a false negative example. In a video from the ShanghaiTech dataset exhibiting trajectory anomalies, a person is walking in a circular path at a regular pace. Our method fails to generate higher anomaly scores for the corresponding period of this trajectory. It results in an AUC of only 74% for this video, which is significantly below the average AUC for all videos (i.e., macro-AUC) of 90.8%. We believe that the poor performance of trajectory-related anomaly detection can be attributed to insufficient modeling in capturing long-distance movements. The second row demonstrates a false positive case on the Avenue dataset. When the persons close to the camera walk by at a regular pace, their speeds appear to be fast. As a result, the anomaly scores generated by the context recovery stream are higher than normal events. Even after fusion with the anomaly scores from the knowledge retrieval stream which serves as a corrective measure, the final anomaly scores remain high and lead to a false positive. Once these pedestrians in the yellow boxes disappear from the frame, the anomaly scores from the context recovery stream return to a normal low level.

To address the issue of weak capability of detecting trajectory anomalies, we propose to capture long-distance trajectory by incorporating more frames as input. Alternatively, we can obtain accurate pedestrian trajectories using object tracking methods and model normal trajectories appeared in the training set. For the problem of large prediction errors in the context recovery stream caused by moving objects close to the camera, we propose to supplement optical flow information to train the model. It can enhance the modeling capability for fast normal motions, thereby reducing prediction errors for nearby moving objects and decreasing false positives.

J. Robustness experiments

We conduct experiments on the ShanghaiTech [31] and Avenue [30] datasets by introducing noise and frame occlusions to evaluate the robustness of our method to variations in data quality. The results are depicted in Fig. 13. In the noise experiment, we introduce Gaussian noise to the input test snippets, where the mean of the noise is zero, and the standard deviation gradually increases, reaching a maximum of 80. In the occlusion experiment, we randomly select a portion of input frames and set their pixel values to zero, simulating the scenario where some frames are occluded or lost entirely. The maximum proportion of occluded frames is 4/8 = 50%. It can be seen from the figure that even under the influence of noise with a standard deviation of 80, or when half of the frames are occluded, our method still maintains a certain level of anomaly detection capability. On the ShanghaiTech and Avenue datasets, the micro-AUCs remain around 75% and 80%, respectively, demonstrating the robustness of our approach to noise and occlusions.

V. DISCUSSION

A. Selecting Window Size in MLE

The sliding window size of the maximum local error (MLE) is dataset dependent. In real-world applications, the source of a test video may be unavailable. In this case, there is no guidance on how to select a proper sliding window size for the MLE mechanism. To address this issue, we propose to use image-level template matching (with storing backgrounds during training) or a lightweight 2D CNN (being trained for classifying datasets during training) to identify the dataset of the test video. An additional experiment conducted by us demonstrates that a shallow CNN, comprising three convolutional layers, one global average pooling layer and one linear layer, can achieve 99.8% dataset classification accuracy for the videos in ShanghaiTech, Avenue, Corridor and Ped2 datasets. Therefore, obtaining the source of the test video is not difficult,
enabling us to select a suitable window size for the MLE mechanism.

**B. Scaling to Large Amount of Video Data**

As the amount of video data increases, we would like to discuss the accuracy and computational efficiency of our method on large-scale datasets. In terms of accuracy, the performance of our method on four datasets of varying scales indicates that our approach maintains good performance as the dataset scale increases. The frame counts of Ped2 [29], Avenue [30], ShangHaiTech [31], and Corridor [32] are 4.5k, 30.6k, 317k, and 483k, respectively, with the scale of the first three datasets increasing exponentially. Our method achieves state-of-the-art performance among the frame-level methods not only on Avenue, but also on two large-scale datasets, ShangHaiTech and Corridor, demonstrating good scalability with a large amount of video data.

In terms of computational efficiency, the context recovery stream is a fixed network, and thereby the inference speed remains constant. During the inference of the knowledge retrieval stream, the feature extractor, hash layers, and hash tables are fixed. Additionally, each bucket contains only one event pattern vector. Therefore, the inference speed also remains constant. In summary, the inference speed of our method does not vary with an increase in data scale.

It is notable that as the scale of data increases, the size of the hash table constructed during the training phase will grow logarithmically. However, we find that even on the ShangHaiTech dataset which features complex scenarios and a high variety of anomalies, the disk usage of the hash tables stored with “float32” type is only 36 MB. As a contrast, the training videos in the ShangHaiTech dataset occupy 2384 MB, and a typical model ResNet-50 occupies 98 MB. Under conditions of limited storage space, we propose two approaches to reduce the storage space of the hash tables. (1) Storing hash tables with “float16” type reduces the disk usage by half, with almost no impact on accuracy. (2) Pruning hash tables by removing the buckets with low event frequencies. Although it may lead to false positives for infrequent normal events, it maintains effective detection capabilities for anomalous events and the majority of normal events.

**VI. CONCLUSION**

In this paper, we propose a novel two-stream framework composed of a context recovery stream and a knowledge retrieval stream for video anomaly detection. In the context recovery stream, a spatiotemporal U-Net (STU-Net) is proposed to utilize the motion in the current snippet to predict the future frame. Additionally, we propose a maximum local error (MLE) mechanism which can focus on recovery errors in anomalous regions and hence generate more accurate anomaly scores. In the knowledge retrieval stream, we propose an improved learnable locality-sensitive hashing (iL2SH) to store the knowledge about normality and retrieve it to determine whether a testing event is consistent with the normal knowledge. By fusing the context recovery stream and the knowledge retrieval stream, our two-stream framework can use both short-term motion and knowledge about normality to detect anomalies. Extensive experiments verify the effectiveness and complementarity of the two streams, which achieve the state-of-the-art performance on the ShangHaiTech, Avenue and Corridor datasets among the frame-level approaches. In the future work, we will explore more knowledge representations (e.g., structural knowledge and natural language description) for knowledge retrieval.

**REFERENCES**

[1] B. Ramachandra, M. J. Jones, and B. R. Vatsavai, “A survey of single-scene video anomaly detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2293–2312, May 2022.

[2] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, “Learning deep representations of appearance and motion for anomalous event detection,” in *Proc. Brit. Mach. Vis. Conf.*, 2015, p. 8.1–8.12.

[3] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alecte, “Deep appearance features for abnormal behavior detection in video,” in *Proc. Image Anal. Process.*, vol. 10485, 2017, pp. 779–789.

[4] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, “Object-centric auto-encoders and dummy anomalies for abnormal event detection in video,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7842–7851.

[5] V. Saligrama and Z. Chen, “Video anomaly detection based on local statistical aggregates,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2112–2120.

[6] B. Ramachandra and M. J. Jones, “Street Scene: A new dataset and evaluation protocol for video anomaly detection,” in *Proc. Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2558–2567.

[7] R. T. Ionescu, S. Smeureanu, M. Popescu, and B. Alecte, “Detecting abnormal events in video using narrowed normality clusters,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2019, pp. 1951–1960.

[8] Z. Wang, Y. Zou, and Z. Zhang, “Cluster attention contrast for video anomaly detection,” in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 2463–2471.

[9] A. Markovitz, G. Sharir, I. Friedman, L. Zelnik-Manor, and S. Avidan, “Graph embedded pose clustering for anomaly detection,” in *Proc. IEEE Trans. Pattern Recognit.*, 2020, pp. 10536–10544.

[10] Y. Chang, Z. Tu, W. Xie, and J. Yuan, “Clustering driven deep autoencoder for video anomaly discrimination,” in *Proc. Eur. Conf. Comput. Vis.*, vol. 12560, 2020, pp. 338–345.

[11] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, “Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5288–5301, Dec. 2015.

[12] Y. Fan, G. Wen, D. Li, S. Qiu, M. D. Levine, and F. Xiao, “Video anomaly detection and localization via Gaussian mixture fully convolutional variational autoencoder,” *Comput. Vis. Image Understand.*, vol. 195, Jun. 2020, Art. no. 102920.

[13] W. Liu, W. Luo, D. Lian, and S. Gao, “Future frame prediction for anomaly detection—A new baseline,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6536–6545.

[14] D. Gong et al., “Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1705–1714.

[15] G. Yu et al., “Cloze test helps: Effective video anomaly detection via learning to complete video events,” in *Proc. ACM Multimedia*, 2020, pp. 583–591.

[16] S. Lee, H. G. Kim, and Y. M. Ro, “BMAN: Bidirectional multi-scale aggregation networks for abnormal event detection,” *IEEE Trans. Image Process.*, vol. 29, pp. 2395–2408, 2020.

[17] H. Park, J. Noh, and B. Ham, “Learning memory-guided normality for anomaly detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 14360–14369.

[18] W. Luo, W. Liu, D. Lian, and S. Gao, “Future frame prediction network for video anomaly detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7505–7520, Nov. 2022.

[19] T.-N. Nguyen, S. Roy, and J. Meunier, “SmithNet: Strictness on motion-texture coherence for anomaly detection,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2287–2300, Jun. 2022.
[20] H. Lv, C. Chen, Z. Cui, C. Xu, Y. Li, and J. Yang, “Learning normal dynamics in videos with meta prototype network,” in *Proc. Comput. Vis. Pattern Recognit.*, 2021, pp. 15425–15434.

[21] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, “A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13568–13577.

[22] R. Cai, H. Zhang, W. Liu, S. Gao, and Z. Hao, “Appearance-motion memory consistency network for video anomaly detection,” in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2021, pp. 938–946.

[23] X. Wang et al., “Robust unsupervised video anomaly detection by multipath frame prediction,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 1–12, Jun. 2021.

[24] J. Li, Q. Huang, Y. Du, X. Zhen, S. Chen, and L. Shao, “Variational abnormal behavior detection with motion consistency,” *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4106–4116, Oct. 2021.

[25] Z. Fang, J. Liang, J. T. Zhou, Y. Xiao, and F. Yang, “Anomaly detection with bidirectional consistency in videos,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1079–1092, Mar. 2022.

[26] J. Li, Q. Huang, Y. Du, X. Zhen, S. Chen, and L. Shao, “Variational abnormal behavior detection with motion consistency,” *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4106–4116, Oct. 2021.

[27] O. Hirschorn and S. Avidan, “Normalizing flows for human pose block for anomaly detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Jun. 2020, pp. 770–778.

[28] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new benchmark and dataset,” in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 29, no. 2, pp. 2072–2080, 2017.

[29] M. Fathi, Z. Jiang, Y. Li, Z. Wang, L. Yang, and B. Zhang, “Soft-Bayesian self-supervised representation learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 770–778.

[30] X. Tian, X. Zhao, and X. Zhou, “DB-LSH 2.0: Locality-sensitive hashing with query-based dynamic bucketing,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 3, pp. 1000–1015, Mar. 2024.

[31] L. Ma, H. Hong, F. Meng, Q. Wu, and J. Wu, “Deep progressive asymmetric quantization based on causal intervention for fine-grained image retrieval,” *IEEE Trans. Multimedia*, vol. 26, pp. 1306–1318, 2023.

[32] H. Liu, W. Zhou, Z. Wu, S. Zhang, G. Li, and X. Li, “Refining codes for locality sensitive hashing,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 3, pp. 1274–1284, Mar. 2024.

[33] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, “A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13568–13577.

[34] H. Liu, W. Zhou, Z. Wu, S. Zhang, G. Li, and X. Li, “Refining codes for locality sensitive hashing,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 3, pp. 1274–1284, Mar. 2024.

[35] Q. Qin, K. Xie, W. Zhang, C. Wang, and L. Huang, “Deep neighborhood structure-preserving hashing for large-scale image retrieval,” *IEEE Trans. Multimedia*, vol. 26, pp. 1881–1893, 2023.

[36] J. Caraman and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4724–4733.

[37] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[38] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 448–456.

[39] L. J. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv:1607.06450*, 2016.

[40] H. Fan, Y. Li, B. Xiong, W.-Y. Lo, and C. Feichtenhofer. (2020). *PySlowfast* [Online]. Available: https://github.com/facebookresearch/PySlowfast

[41] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[42] O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[43] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh, “Learning regularity in skeleton trajectories for anomaly detection in videos,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11996–12005.

[44] X. Luo, Q. Miao, Y. Zhang, T. Hasegawa, and X. Zhang, “Spatio-temporal graph convolutional neural network for anomaly detection,” in *Proc. Eur. Conf. Comput. Vis.* (ECCV), vol. 115, no. 3, pp. 211–252, Sep. 2021.

[45] R. Yang, X. Zhang, X. Nie, R. He, M. Chen, and Y. Yin, “Normality learning in multispace for video anomaly detection,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 396–376, Sep. 2021.

[46] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh, “Learning regularity in skeleton trajectories for anomaly detection in videos,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11996–12005.

[47] X. Luo, F. Yu, M. K. K. Reddy, and Y. Wang, “Few-shot scene-adaptive anomaly detection,” in *Proc. Eur. Conf. Comput. Vis.*, vol. 12350, 2020, pp. 125–141.

[48] H. Song, C. Sun, X. Wu, M. Chen, and Y. Jia, “Learning normal patterns via adversarial attention-based autoencoder for abnormal event detection in videos,” *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 2138–2148, Aug. 2020.

[49] Y. Zhang, X. Nie, R. He, M. Chen, and Y. Yin, “Normality learning in multispace for video anomaly detection,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 396–376, Sep. 2021.

[50] X. Luo et al., “Video anomaly detection with sparse coding inspired deep neural networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 1070–1084, Mar. 2021.

[51] X. Zeng, Y. Jiang, W. Ding, H. Li, Y. Hao, and Z. Qiu, “A hierarchical spatio-temporal graph convolutional neural network for anomaly detection in videos,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 200–212, Jan. 2023.

[52] W. Liu, H. Chang, B. Ma, S. Shan, and X. Chen, “Diversity-measurable anomaly detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12147–12156.

[53] X. Huang, C. Zhao, and Z. Wu, “A video anomaly detection framework based on appearance-motion semantics representation consistency,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[54] O. Hirschhorn and S. Avidan, “Normalizing flows for human pose anomaly detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13545–13554.

[55] J. Wang, Y. Wang, J. Jia, J. Chen, D. Zhang, X. Bao, and D. Huang, “Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles,” in *Proc. ECCV*, vol. 13670, Springer, 2022, pp. 494–511, doi: 10.1007/978-3-031-20808-9_29.

[56] N. Madan et al., “Self-supervised masked convolutional transformer block for anomaly detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 1, pp. 525–542, Jan. 2024.

[57] A. Barbahai et al., “SSMTL++: Revisiting self-supervised multi-task learning for video anomaly detection,” *Comput. Vis. Comput. Image Understand.*, vol. 229, Mar. 2023, Art. no. 103656.