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

Congqi Cao^{*†}, Yue Lu^{*}, and Yanning Zhang

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 four datasets.

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

I. INTRODUCTION

V IDEO 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 can only obtain the normal data, while the abnormal data is absent. Second, video anomalies are scene-dependent [1]. For example, playing football is normal on the pitch but anomalous on the road. Third, some kinds of normal events happen frequently while some happen occasionally. If an algorithm cannot handle the imbalanced data distribution well, it is easy 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]-[4], KNN [5], [6], clustering [7]-[10], a mixture of Gaussians [11], [12] and other methods to represent normal events with representative features, such as clustering centers and distribution parameters. The anomaly probability of a testing event is determined by the distance between it and the representative features. Although these methods have the advantage of good interpretability, they lack adequate and flexible generalization of normal events. For example, the number of cluster centers needs to be set manually and is usually small. With the development of deep learning, methods based on snippet reconstruction and future frame prediction [13]–[28] have become popular in recent years. These two kinds of 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. Since they both aim to recover the context information of frames, we classify them as the context recovery method. This method is good at distinguishing short-term anomalous movements, but lacks the understanding of normality. For example, a snippet reconstruction model can accurately reconstruct the action of playing football not only on the pitch but also on the road, making it impossible to detect the anomalous event. Although some memory-augmented context recovery methods [14], [17], [20]-[22] try to explicitly utilize the diversity of normal data, they are essentially designed for recovering the context. Therefore, it is still difficult for such models to mitigate that drawback. 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" in the knowledge about normality, we can assume that an anomalous event occurs.

In this work, we propose a novel two-stream framework that can not only discriminates short-term abnormal motions, but also leverages the knowledge from normal events to enhance the understanding of normality. Different from other multistream models that use multiple modalities [18], [26], [29], [30], our proposed two-stream framework consists of a context recovery stream and a knowledge retrieval stream, as shown in Fig. 1. In our context recovery stream, the normal shortterm motion patterns are modeled by predicting the future frame based on the input snippet. The anomaly probability

^{*}Equal contribution. [†]Corresponding author.

Congqi Cao, Yue Lu and Yanning Zhang 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).



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.

of a testing event is obtained based on the prediction error. In the knowledge retrieval stream, a series of normal events are encoded as the knowledge about normality and stored in a knowledge base. The anomaly probability can be 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.

For context recovery, the existing models [13]–[15], [17], [18], [20]–[23], [25], [26] usually concatenate the input frames as an image and feed it into a 2D-CNN-based U-Net [31] to reconstruct the snippet or predict the future frame. To supplement the motion information, the constraint of optical flow is required. Although there are a few works [32]-[34] adopt 3D CNN as the encoder, they do not explore a suitable 3D-CNN-based U-Net structure and can only use a shallow convolutional auto-encoder, which limits the representation capacity of the model. We propose a novel spatiotemporal U-Net (STU-Net) for future frame prediction, which takes the 3D CNN designed for action recognition as the encoder and retains the temporal dimension in the deep layers, so as 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, the existing context recovery method has the problem that the reconstruction or prediction error is proportional to the number of foreground objects [1], which is easy to cause false positives. Some methods [4], [15], [21], [30], [32] employ the object detector to solve this problem, but seriously ignore the scene-dependent characteristic of video anomalies. Moreover, they are incapable of detecting the anomalous objects not included in the training classes of the object detector. We propose a maximum local error (MLE) mechanism to focus on the recovery degree of the local anomalous region. Specifically, based on the assumption that the anomalous region causes larger context recovery error than normal regions, we propose to use the maximum patchlevel recovery error in the frame instead of the frame-level error to reflect the anomaly probability. Due to the lack of validation sets in the existing datasets [35]–[38], 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.

For knowledge retrieval, although the representative features generated by some methods, such as the cluster centers in clustering [8] and the nearest neighbor samples in KNN [6], could be regarded as the knowledge about normality, they can hardly meet the requirements of efficiency, flexibility and comprehensiveness for knowledge extraction. In this work, we propose an improved learnable locality-sensitive hashing (iL²SH) based on [39] to store and retrieve the knowledge about normality, which can find the knowledge representation consistent with a testing event adaptively and efficiently. 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 with the same key as the knowledge representation of such normal events. A testing event obtains the hash code through the same process, and tries to find the knowledge representation sharing the same key. We calculate the anomaly probability according to the distance between the testing hash code and the retrieved knowledge representation. Compared with LLSH [39], we make the following improvements. First, we improve the optimization of the hash encoder. LLSH adopts MoCo [40] contrastive learning framework to train the hash encoder, where the optimization effect 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 can achieve better training results with only positive samples. Second, 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 make the hash layers different from each other. It can enlarge the distances between the hash layers and improve the performance after optimization.

We conduct comprehensive studies across several datasets to verify the effectiveness of the proposed two-stream framework, including ShanghaiTech [37], CUHK Avenue [36], IITB Corridor [38] and UCSD Ped2 [35] datasets. Without using optical flow and object detection, our context recovery stream can exceed 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 nearestneighbor-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. Our method achieves the state-of-theart performance on all the four 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 the finelevel local context to detect anomalies, but also takes 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 the state-of-the-art performance on ShanghaiTech, CUHK Avenue, IITB Corridor and UCSD Ped2 datasets.

II. RELATED WORK

A. Context Recovery Methods

The context recovery methods [13]-[28], which are based on the assumption that normal events are easy to recover while the abnormal events are hard to recover, become the mainstream in the field of VAD in recent years. This kind of 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 proposed in the field of image segmentation to videos, making it difficult to model the motion of objects. Hence, they have to combine with the optical flow modality [13], [18], [22] or recurrent units [16], [37]. 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 [32]–[34]. Nevertheless, these models usually use shallow networks to avoid gradient vanishing. Compared with the above methods, our proposed STU-Net can model the motion information without additional modality or recurrent operation. There is a 3D-CNN-based U-Net in the field of image summary [41]. It directly applies squeeze-andexcitation blocks [42] for compressing feature dimensionality and thus cannot be applied for future frame prediction, in which 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.

Besides, most of the methods feed the whole frame into the model in both training and testing phases, and others take the detected objects [4], [15], [21], [30], [32] or video patches [6], [33], [43] as the input. As described in the Introduction section, the methods based on object detection only use the foreground objects and thus ignore the scene information. Additionally, they cannot detect the anomalous objects whose categories are not covered in the pre-trained object detector. The patch-based methods are difficult to capture the full movement of the object, since patches of the same spatial grid across times incurs visual misalignment when rigidly dividing a moving object. In our context recovery stream, we still use the whole frame as the input so that complete and long-term motions can be captured. In 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 [44] also train a frame-level model and calculate normalized patch-level errors, they do not exploit an effective solution to solve the large error problem caused by foreground objects. They use a fixed and small patch for different datasets, which is easily affected by the noise in the error map and cannot handle different resolutions of the videos. We propose a novel maximum local error (MLE) mechanism that utilizes the training videos to simulate anomalies and selects an appropriate patch size for the dataset. The proposed MLE can effectively focus on the anomalous region and calculate a more accurate recovery error. Therefore, it alleviates the problem of recovery error proportional to the number of foreground objects.

B. Knowledge Retrieval Methods

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. The commonly used knowledge representations include decision boundaries of OC-SVM [2]-[4], nearest neighbor samples [5], [6], cluster centers [7]-[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, the highest similarity between its Kfeature space representations and corresponding space centers is regarded as its regularity score. However, the value of Kis usually too small to fully exploit the knowledge in normal data. Ramachandra and Jones [6] build an exemplar set, in which only the normal sample whose distance from the stored samples in the set exceeds a threshold will be added. The anomaly score is determined simply based on the distance between the testing sample and its nearest exemplars, which lacks the abstraction of knowledge. The normal patterns in the memory modules of context recovery methods [14], [17], [20]-[22] can also be regarded as the knowledge representations. Limited by the size of the memory, it is difficult to contain adequate 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.



Fig. 2. Overview of our context recovery stream. A clip of previous 8 frames are fed into STU-Net and predict the current frame. The maximum local error (in red box) between the predicted frame and the ground truth frame is used as the anomaly score.

Even if iL^2SH is combined with the memory-augmented context recovery model, it is still able to bring significant improvements. Different from Lu *et al.* [39] who use MoCo [40] to train the 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 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 the sequential processes of training hash encoder, constructing knowledge base and retrieving knowledge. Finally, we introduce 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 τ , the snippet containing the τ -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 error between the recovered frame and the ground truth frame is taken as the anomaly score. In the knowledge retrieval

TABLE I Network Details of Spatiotemporal U-Net

	Spatiotemporal U-Net							
	Eı	ncoder			Decoder	r		
fn1(n):	$[(n,3,1^2)]$	$, (n,1,3^2), ($	$4n,1,1^2)]$	fn3(n).	$[(n,1,3^2), ($	<i>n</i> ,1,3 ²),		
fn2(n):	$[(n,1,1^2)]$, (<i>n</i> ,1,3 ²), ($4n,1,1^2)]$	<i>Jiio</i> (<i>iii</i>):	$(4n,1,3^2), ($	$(2n,1,2^2)^{T}]$		
Level	Kernel	Stride	Shape	Level	Kernel	Shape		
Input	-	-	$(3, 8, 256^2)$	Input	-	$(2048, 1, 2^2)$		
	(64,5,72)	$(1,2^2)$	$(64, 8, 128^2)$		$(512,1,1^2)$	$(512, 1, 4^2)$		
L1	$max(1,3^2)$	(1,2 ²)	(64,8,64 ²)	L5	$(512,\!1,\!2^2)^T$	(512,1,1)		
	$fn1(64) \times 3$	$(1,1^2) \times 9$	(256,8,64 ²)		$(512,1,1^2)$	$(2048, 1, 8^2)$		
	$max(2,1^2)$	(2,1 ²)	(256,4,64 ²)		$(2048, 1, 1^2)^{T}$	(20.0,1,0)		
L2	$[fn1(128), fn2(128)] \times 2$	$(1,1^2)$ $(1,2^2)$	(512,4,32 ²)	L4	fn3(512)	(1024,1,16 ²)		
		$(1,1^2) \times 10$		L3	fn3(256)	$(512.1.32^2)$		
	[fn1(256),	$(1,1^2)$				(===,=,==)		
L3	fn2(256)]×3	$^{(1,2^2)}_{(1,1^2) \times 16}$	(1024,4,16 ²)	L2	fn3(128)	(256,1,64 ²)		
	fn2(512)	(1,1 ²)	2	L1	<i>fn3</i> (64)	(128,1,128 ²)		
L4	fnl(512)	$(1,2^2)$	$(2048,4,8^2)$	21	fn3(32)	(64,1,256 ²)		
	fn2(512)	$(1,1^2) \times 7$		Output	$(64, 1, 3^2)$	$(3.1.256^2)$		
L5	$avg(4,7^2)$	(1,1 ²)	(2048,1,2 ²)	put	$(3,1,3^2) \times 2$	(0,1,200)		

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 given a high value. The anomaly scores of the two streams are added together as the final anomaly score at time τ .

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 $I_{t-8}, I_{t-7}, \dots, I_{t-1}$ are fed into STU-Net to generate the predicted frame \hat{I}_t . We calculate the proposed maximum local error (MLE) between \hat{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 clip of 8 frames with the spatial resolution of 256×256 are 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 in each level are connected via a shortcut.

Different from the previous works that stack the input frames as an image to use 2D convolutions, we retain the temporal dimension in the encoder, which helps to predict the future frame through the motion in previous frames. Due to the inequality in temporal dimension, the feature map output by the encoder cannot be directly connected to the input of the decoder in the same level. We address this issue by adding 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 in the same level can be concatenated along the channel dimension, to serve as the input of the next level of the decoder.

The network structures of the encoder and decoder are detailedly displayed in TABLE I. We adopt the I3D network [45] with ResNet-50 backbone [46] as the encoder, and design the decoder on our own. Each parenthesis in the table describes the kernel size of the convolution/pooling layer or the shape of the output feature map in dimension (C, T, HW), where C, T, H and W respectively represent the number of channels, the length, height and width of the feature map. For strides and pooling layers, we omit the channel dimension C. We define three functions fn1, fn2 and fn3 for convenience. Each of them represents a block composed of multiple convolution layers. max and avg denote the max pooling and average pooling, respectively. The residual connection in each level is not displayed.

For the layers in the encoder, each convolution is followed by a batch normalization (BN) [47] and a ReLU activation. We remove the last ReLU activation, as it restricts diverse feature representations. In the decoder, the convolution layer with superscript T represents a transposed convolution, which enlarges the spatial resolution of the feature map. The strides in regular convolutions and transposed convolutions are $(1, 1^2)$ and $(1, 2^2)$, 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 kernel size $(C, 4, 1^2)$ and stride $(1, 1^2)$ 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:

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

$$L_{FLE}(I_t, \hat{I}_t) = \|I_t - \hat{I}_t\|_F^2 + \lambda_{L1} \times |I_t - \hat{I}_t|, \quad (2)$$

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

The proposed STU-Net has the ability to utilize 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 is comparable to existing frame prediction methods that combine with optical flow to supplement motion information.

2) Maximum Local Error: For the VAD task, we expect the context recovery model to be accurate in predicting normal regions and inaccurate in abnormal regions. However, because of the nonstatic background, large number of foreground objects, image noise and other possible factors, we cannot **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 $m \leftarrow \text{floor}(N/nseq)$ \triangleright get the length of a segment for each $i \in \{1, 2, \dots, N\}$ do $i \leftarrow i \mod m$ \triangleright get the segment index $start \leftarrow floor(m \times (j + 0.5 - 0.5 \times ratio)) + 1$ $end \leftarrow start + floor(m \times ratio)$ $\tilde{I}_i \leftarrow \text{rotate}(\text{flip}(I_i))$ if $i \ge start$ and i < end then $\tilde{I}_f \leftarrow \text{rotate}(\text{flip}(I_{i+offset}))$ $\tilde{I}_i \leftarrow 0.5 \times (\tilde{I}_i + \tilde{I}_f)$ $L_i \leftarrow 1$ else $L_i \leftarrow 0$ end if end for

make fully accurate predictions for normal regions in the next frame in any case. Inaccurate prediction of normal regions will lead to higher errors and false positives.

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

$$MLE(I_t, \hat{I}_t)_{k,s} = \text{MaxPool}_{k,s}(||I_t - \hat{I}_t||_F^2 + \lambda_{L1} \times |I_t - \hat{I}_t|),$$
(3)

where k is the size of the sliding window, and s is the stride.

The hyper-parameter k is dataset-based and can be determined by a validation set. However, most VAD datasets do not have validation sets. Thus, we propose an alternative solution that uses training videos to simulate anomalies by means of data augmentation, which can be seen in Algorithm 1. Specifically, we spatially flip and rotate all the frames in a video at a random angle to obtain a new video that the model has not seen before. To generate an anomalous frame, we fuse 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 $K = \{k_1, k_2, \dots, k_n\}$ according to the evaluation metric.

The proposed MLE mitigates the interference of recovery errors in normal regions. With MLE, our STU-Net can reach



Fig. 3. The architecture and training process of iL^2SH . The proposed iL^2SH includes an event encoder and a hash encoder which consists of a group of hash layers. We use a Siamese network for training iL^2SH , where the two branches share the same parameters.

similar or better 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 retrieve the knowledge efficiently to detect anomalies. To this end, we propose an improved learnable locality-sensitive hashing (iL²SH). First, we take hash functions as learnable parameters and embed them into a neural network to optimize using the training data. Then, we map all training events into hash codes by the optimized hash functions. The mean vector of similar hash codes are stored in each bucket of the hash table and served as the normal knowledge representations. Finally, in the testing phase, we look up a bucket consistent with the testing event, and calculate the anomaly score based on the distances between the testing hash codes and the retrieved knowledge representation.

1) Training iL^2SH : The structure of iL^2SH 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 will be used to construct a hash table. A hash layer has three sequential layers including a liner layer, a layer normalization [48] and a sigmoid activation. Each linear layer 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^2SH is embedded as a branch of the Siamese network, where the two branches share the same parameters. We feed a snippet S_t into one of the branches. At the same time, a similar snippet $S_{t+\Delta t}$ which is temporally close to S_t is sampled from the same video and fed into the

Algorithm 2 The Process of Constructing Knowledge Base

Require: $\{S_i\}_{i=1}^N$: N training snippets $Enc(\cdot)$: iL²SH, which maps a snippet to B hash codes $\{\mathcal{H}_b[key] = (cnt, val)\}_{b=1}^{B}$: B empty hash tables Ensure: hash tables $\{\mathcal{H}_b\}_{b=1}^{B}$ that stores the hash codes function BIN(a) ▷ Binary function for all $a^{\langle i \rangle}$ of the *i*-th bit in *a* do $a^{\langle i \rangle} \leftarrow 0$ if $a^{\langle i \rangle} < 0.5$ else 1 end for return a end function for each $i \in \{1, 2, \dots, N\}$ do $\{h_b\}_{b=1}^B \leftarrow Enc(S_i)$ for each $b \in \{1, 2, \cdots, B\}$ do $k \leftarrow BIN(h_b)$ if k exists in \mathcal{H}_b .key then $\mathcal{H}_b[k].val$ $\leftarrow \quad (\mathcal{H}_b[k].val \ \times \ \mathcal{H}_b[k].cnt \ +$ $h_b)$ / $(\mathcal{H}_b[k].cnt+1)$ $\mathcal{H}_b[k].cnt \leftarrow \mathcal{H}_b[k].cnt + 1$ else $\mathcal{H}_b[k].cnt \leftarrow 1$ $\mathcal{H}_b[k].val \leftarrow h_b$ end if end for end for

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:

$$L_{c}(l_{t}, l_{t+\Delta t}) = 1 - \frac{l_{t}}{\|l_{t}\|} \cdot \frac{l_{t+\Delta t}}{\|l_{t+\Delta t}\|}$$
(4)

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

 S_t and $S_{t+\Delta t}$ make up a positive pair and the distance between them is pulled close by Eq. (4). We do not take the snippets from different videos as negative pairs and push away 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 output as different hash codes as possible to construct different hash tables. Therefore, we propose a mutual difference loss that enlarges the difference between the hash codes output by a hash encoder:

$$L_m([h_1, h_2, \cdots, h_B]) = \frac{2}{RB(B-1)} \sum_{j=1}^B \sum_{i=j+1}^B \frac{h_i}{\|h_i\|} \cdot \frac{h_j}{\|h_j\|},$$
(5)

where $h_i \in \mathbb{R}^R$ is 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 total loss for training iL^2SH 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 the mutual difference loss of the *i*-th branch, and λ_m is the weight.



Fig. 4. An example of constructing a hash table. F denotes a feature fed into the hash layer. The hash codes with the same key are averaged and stored in one bucket.

When the training process is finished, the hash layers has an enhanced ability to map similar events to similar hash codes. We can use iL^2SH for constructing knowledge base in the next step.

2) Constructing Knowledge Base: Our purpose is to construct a knowledge base which contains the knowledge representations about normality obtained from the training data. To this end, we first map each training event to a group of hash codes, and then store each hash code into a corresponding 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 of the hash codes, and the values are the mean vectors of those hash codes sharing the same binary key. A detailed process of constructing 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 obtain 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 has the same binary key "1101", we calculate the mean vector of the two hash codes, which is then stored in the second bucket. 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 binary keys. Then, we retrieve a bucket from the corresponding hash table, and calculate the L2 distance between the testing hash code

Algorithm 3 The Process of Retrieving Knowledge

Require: S_t : a testing snippet $Enc(\cdot)$: iL²SH, which maps a snippet to B hash codes ${\mathcal{H}_b[key] = (cnt, val)}_{b=1}^B$: B hash tables obtained by Algorithm 2 P_{max} : a predefined maximum anomaly score **Ensure:** p_t : the anomaly score of S_t ${h_b}_{b=1}^B \leftarrow Enc(S_t)$ $p_t = P_{max}$ for each $b \in \{1, 2, \cdots, B\}$ do $k \leftarrow \text{BIN}(h_b)$ \triangleright BIN is the function in Algorithm 2 if k exists in \mathcal{H}_b .key then $d = \|\mathcal{H}_b[k].val - h_b\|_2$ if $d < p_t$ then $p_t \leftarrow d$ end if end if end for

and the vector in the retrieved bucket. The minimum distance from all the hash tables is taken as the anomaly score of S_t . By using the decision from multiple hash tables, we can find the most relevant knowledge representation and hence reduce false alerts. However, each of the binary keys may not exist in the corresponding hash table. In this case, we treat S_t as an anomalous event which is inconsistent with normal knowledge. S_t is assigned with a predefined high anomaly score P_{max} , which is the maximum L2 distance between any two hash codes:

$$P_{max} = \sqrt{R},\tag{7}$$

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.* mean vector of hash codes) from each hash table, and compare the testing event with the retrieved knowledge representations to detect anomalies.

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 a late fusion:

$$p_{fuse} = \lambda_{cr} p_{cr} + \lambda_{kr} p_{kr},\tag{8}$$

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 anomalous movements, and the knowledge retrieval stream can make use of high-level semantic knowledge about normality to detect anomalous events. The anomaly detection results from the two streams can complement each other and thereby improve the performance for VAD.

Dataset	Year	Training videos / frames	Testing videos / frames	Scenes	Resolution	Abnormal events
UCSD Ped2 [35]	2010	16 / 2.6k	12 / 2.0k	1	360×240	Bikers, carts and skaters
CUHK Avenue [36]	2013	16 / 15k	21 / 15k	1	640×360	Throwing object, wrong direction, running, etc.
ShanghaiTech [37]	2017	330 / 275k	107 / 43k	13	856×480	Bikers, loitering, fighting, vehicles, etc.
IITB Corridor [38]	2020	208 / 302k	150 / 182k	1	1920×1080	Protest, playing with ball, unattended baggage, etc.

TABLE II FOUR DATASETS USED IN OUR EXPERIMENTS

IV. EXPERIMENTS

In this section, we conduct comprehensive experiments to verify the effectiveness of our proposed two-stream framework and compare with other methods. We first introduce the datasets, evaluation metric and implementation details in our experiments. Next, we carry out a detailed ablation study to investigate the effect 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. After that, we visualize and analyze the effect of MLE and fusion of two streams. Finally, we compare our two-stream framework with existing methods, where our method achieves the state-of-the-art performance.

A. Datasets

We evaluate our method on four commonly used datasets, i.e. ShanghaiTech [37], CUHK Avenue [36], IITB Corridor [38] and UCSD Ped2 [35], 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 making it difficult to detect anomalies. Corridor is a newly proposed challenging dataset which has a 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 and with low resolution. We mainly use ShanghaiTech and Avenue for ablation studies, and compare with other methods on the four datasets.

B. Evaluation Metric

We adopt the most widely used area under curve (AUC) to evaluate the performance of anomaly detection. 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 [30]. Micro-AUC is obtained by concatenating all frames in a dataset as a 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 of the 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 fed into STU-Net. To mitigate the serious distortion cause by resizing, we manually crop three fixed square regions along the corridor¹. The weight of L1 loss λ_{L1} in Eq. (2) is set to 1. The predefined sizes of sliding windows (i.e. Ks) are $\{2^n\}_{n=4}^8$, and the final sizes used in our experiments on ShanghaiTech, Avenue, Corridor and Ped2 are 128, 64, 256 and 32, respectively. In the knowledge retrieval stream, each input snippet consists of 8 frames. The temporal sampling rates are 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 λ_m , λ_{cr} and λ_{kr} are set to 0.64, 1 and 1, respectively. The event encoder I3D is pre-trained on Kinetics-400 dataset [45], [49] and freezed during training. Following previous works [4], [18], [20], [30], [32], we normalize the anomaly scores and apply a Gaussian filter to smooth them. All the experiments are performed with two Nvidia Tesla V100 GPUs using PyTorch [50]. More details can be seen in our code: https://github.com/zugexiaodui/TwoStreamUVAD.

D. Ablation Study

We conduct ablation experiments on the proposed twostream framework to study the effect of different components. The results on ShanghaiTech and Avenue datasets are shown 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 [46] network and other layers are the same as those in STU-Net. The encoder is pre-trained on ImageNet [51] and freezed during training. Although the proposed STU-Net has better performance than the 2D U-Net with pre-training, it can use the parameters trained on Kinetics-400 dataset to further improve the performance. In the knowledge retrieval stream, iL²SH without training is adopted as the baseline. "w/ neg." means that iL²SH is trained with negative pairs, where the videos of negative instances differs 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 . "w/ L_m " denotes iL²SH is trained with our proposed mutual difference loss.

¹The left-top corners and widths are $(x, y, w) = \{(320, 184, 896), (576, 96, 412), (672, 0, 256)\}$

	Context recovery stream				Knowledge retrieval stream				Micro-AUC (%)	
index	U-Net	STU-Net	Pre-training	MLE	iL ² SH	Training	w/ neg.	w/ λ_m	ShanghaiTech	Avenue
1	\checkmark		\checkmark						72.5	82.0
2		\checkmark							74.4	84.8
3		\checkmark	\checkmark						75.3	85.1
4	\checkmark		\checkmark	\checkmark					75.8	85.1
5		\checkmark	\checkmark	\checkmark					79.7	87.2
6					\checkmark				71.8	83.2
7					\checkmark	\checkmark	\checkmark		78.9	86.0
8					\checkmark	\checkmark			79.6	86.7
9					\checkmark	\checkmark		\checkmark	81.0	88.1
1+6	~		\checkmark		\checkmark				75.4	86.7
1+9	\checkmark		\checkmark		\checkmark	\checkmark		\checkmark	80.2	88.2
5+6		\checkmark	\checkmark	\checkmark	\checkmark				81.5	88.9
5+9		\checkmark	\checkmark	\checkmark	\checkmark	\checkmark		\checkmark	83.7	90.8

TABLE III EFFECT OF DIFFERENT COMPONENTS

 TABLE IV

 Fusion of Different Methods on ShanghaiTech (Micro-AUC %)

	Context r	methods	Knowledge retrieval methods				
Method	STU-Net	MPN	AMMC	iL ² SH	CAC	Exemplar	
STU-Net	79.7	77.5	78.4	83.7	82.1	80.9	
MPN	77.5	73.1	74.2	79.9	78.7	77.4	
AMMC	78.4	74.2	73.7	80.4	79.3	77.9	
iL ² SH	83.7	79.9	80.4	81.0	80.9	79.1	
CAC	82.1	78.7	79.3	80.9	75.8	77.9	
Exemplar	80.9	77.4	77.9	79.1	77.9	74.2	

 TABLE V

 Fusion of Different Methods on Avenue (Micro-AUC %)

	Context	recovery r	nethods	Knowledge retrieval methods			
Method	STU-Net	MNAD	AMMC	iL ² SH	CAC	Exemplar	
STU-Net	87.2	88.1	86.8	90.8	88.3	88.3	
MNAD	88.1	87.5	88.4	90.6	89.7	89.2	
AMMC	86.8	88.4	86.6	90.3	89.6	88.9	
iL ² SH	90.8	90.6	90.3	88.1	87.0	87.3	
CAC	88.3	89.7	89.6	87.0	83.0	84.3	
Exemplar	88.3	89.2	88.9	87.3	84.3	84.1	

For convenience, we add index in TABLE III to refer to different ablation study settings. Comparing experiment (abbr. exp.) 2 with exp. 1, it can be seen that our proposed STU-Net which can utilize the motion of input snippet outperforms 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, which is an advantage compared with 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 iL^2SH 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 reimplement LLSH [39] 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 iL^2SH by about 2%. From exp. "1+6", we can see that the fusion of two baseline models can bring a significant improvement of $3\% \sim 5\%$ 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, to make a trough fusion study. The results on ShanghaiTech and Avenue datasets are shown in TABLE IV and TABLE V. The micro-AUCs of basic methods which are not fused with others are shown on the diagonal and in gray text. An off-diagonal value denotes the AUC of fusing the methods corresponding to its column and its row. The highest results are shown in bold. Since the weights for fusing two streams are both set to 1, the result matrix in each table 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 reimplemented 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 ShanghaiTech and Avenue datasets, respectively. We take the average distance



Fig. 5. Performance variations of fusing different types of methods (CR: context recovery; KR: knowledge retrieval).

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

	Micro-AUC (%)						
Method	ShanghaiTech	Avenue	Ped2				
STU-Net	77.3	84.6	90.0				
iL ² SH	78.7	88.1	91.3				
Two-Stream	81.1	88.8	93.4				
Improvement	2.4	0.7	2.1				

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 TABLE IV and TABLE V, we can see that the fusion of two context recovery methods or two knowledge retrieval methods cannot bring an obvious improvement. For example, in TABLE IV, by fusing MPN (73.1%) and AMMC (73.7%), the AUC (74.2%) is only increased 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 brings a significant improvement, even though the context recovery methods are equipped with memory modules. For example, fusing CAC and AMMC has an improvement of 3.5%(75.8% + 73.7% \rightarrow 79.3%) on ShanghaiTech. In some cases, the fusion of a context recovery method and a knowledge retrieval method may have a slight decline, which is reasonable since the results of the two methods have a huge gap.

To analyze the effect of fusing different types of methods clearly, we calculate the average improvement of each fusion type based on the results in TABLE IV and TABLE V. The performance variations are shown in Fig. 5, 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 iL²SH are fused, the results are the highest on both ShanghaiTech and Avenue datasets as displayed in TABLE IV and TABLE V.

Furthermore, we conduct an experiment where the STU-Net and iL²SH have the same temporal sampling rate, to verify



Fig. 6. AUCs on pseudo anomalous data and real testing data. K denotes the size of sliding window in MLE.

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 VI, the sampling rates of STU-Net and iL²SH are both set to 4. Although this setting results in lower performance compared with the default setting, the improvements on ShanghaiTech, Avenue and Ped2 datasets are still significant. This experiment proves that fusing the context recovery method and knowledge retrieval method which have the same temporal scale can bring improvement for the fusion. On the contrary, even if two methods of the same type have different temporal scales, the fusion of them cannot boost the performance. For example, the context recovery methods STU-Net and AMMC in TABLE V have different temporal scales (2 and 1), the result of fusion is lower than STU-Net by 0.4%.

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 twostream model consisting of STU-Net and iL^2SH achieves the best performance, which demonstrates 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 iL²SH in this section.

1) Maximum Local Error: Fig. 6 displays the AUCs of different sizes of the sliding window (*i.e.* Ks) 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 on the two types of data almost have the same trend. On ShanghaiTech dataset, the highest AUC on pseudo anomalous data is achieved when K = 128, according to which we set K to 128 on testing data and achieves the best result. The same is true on Avenue dataset when K = 64, which demonstrates the effectiveness of simulating anomalies.



Fig. 7. Visualization of frames from ShanghaiTech (the first row) and Avenue (the second row) datasets. The columns are ground truth frames, predicted frames and error frames. Red boxes represent the anomalous regions, and green boxes denote the patches which have the maximum error.



Fig. 8. Score gaps of frame-level error (FLE) and maximum local error (MLE).

Two examples of frames are shown in Fig. 7, from which we can see MLE finds the anomalous region accurately without using any object detection algorithms. To quantitatively study how MLE affects anomaly scores, we calculate the score gaps of frame-level error (FLE) and MLE, as shown in Fig. 8. The score gap is the difference value between the average scores of anomalous frames and normal frames. It reflects the ability to discriminate normality and anomaly, and is expected to has a higher value. Fig. 8 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 of each stream and fusion of the two streams in Fig. 9 to explain how the two streams complement each other. For example, in video "04_0001" of ShanghaiTech, iL^2SH and STU-Net output relative low anomaly scores in the first and second abnormal periods, respectively. However, the fusion of the two streams can generate high anomaly scores in both periods, hence improving the AUC by 2.7% on this video. The score gaps are displayed in Fig. 10, which shows that the fusion of the two streams increases the score gap. Therefore, our two-stream framework can achieve better performance than a single stream.

G. Comparison with Existing Methods

The comparison of different methods on ShanghaiTech [37], Avenue [36], Corridor [38] and Ped2 [35] datasets is displayed in TABLE VII. We report both micro-AUC and macro-AUC for each method if available, and the best results are in bold text. The results marked with † are implemented by Rodrigues



Fig. 9. Anomaly score curves of STU-Net, iL^2SH and fusion of the two streams on two videos. GT: ground truth.



Fig. 10. Score gaps of STU-Net, iL²SH and fusion of the two streams.

et al. [38] because there are no official results. In LLSH [39] and our two-stream framework, the results on Corridor are under the setting of manually cropped regions aforementioned in Implementation Details, which 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. The best results of the methods without using object detection are underlined.

Under a fair setting that does not use object detection, our two-stream method achieves the best performance in both micro-AUC and macro-AUC metrics on three datasets, i.e. ShanghaiTech, Avenue and Corridor. Especially, it is worthy noting that the micro-AUC of our method on 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 those methods using object detection [4], [15], [21], [29], [30], [32], pose estimation [29] and extra datasets [30] (served as pseudo anomalous data), our method is still the best on the two large-scale datasets (i.e. ShanghaiTech and Corridor) in both metrics, and the best on Avenue in macro-AUC. We also experiment on Corridor without manually selecting regions as shown in TABLE VIII, and our two-stream framework still achieves the best performance compared with other methods [13], [26], [29], [38], [52]. Although the performance of our model is not the best on Ped2 dataset, it can be improved by combining with object detection and optical flow in the future since all the anomalies in Ped2 are related to objects.

V. 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

TABLE VII

COMPARISON OF DIFFERENT METHODS ON FOUR DATASETS. TEXT IN BOLD: THE BEST RESULT; [†]: RESULTS IMPLEMENTED BY OTHERS; [‡]: USING MANUALLY CROPPED REGIONS; W/ OBJ.: USING OBJECT DETECTION; UNDERLINE: THE BEST RESULT WO/ USING OBJECT DETECTION

		Shangl	haiTech	Ave	enue	Cor	ridor	Peo	12
Method	w/ obj.	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
FFP [13]		72.8	-	84.9	-	64.7 [†]	-	95.4	-
MemAE [14]		71.2	-	83.3	-	-	-	94.1	-
OADA [4]	\checkmark	-	84.9		90.4	-	-	-	97.8
MPED-RNN [52]		73.4	-	-	-	64.3†	-	-	-
AMC [44]		-	-	86.9	-	-	-	96.2	-
BMAN [16]		76.2	-	90.0	-	-	-	96.6	-
r-GAN [53]		77.9	-	85.8	-	-	-	96.2	-
GEPC [9]		76.1	-	-	-	-	-	-	-
MNAD [17]		70.5	-	88.5	-	-	-	97.0	-
MTP [38]		76.0	-	82.9	-	67.1	-	-	-
Ada-Net [54]		70.0	-	89.2	-	-	-	90.7	-
CAC [8]		79.3	-	87.0	-	-	-	-	-
DeepOC [34]		-	-	86.6	-	-	-	96.9	-
VEC-AM [15]	\checkmark	74.8	-	89.6	-	-	-	97.3	-
Multispace [55]		73.6	-	86.8	-	-	-	95.4	-
AMMC-Net [22]		73.7	-	86.6	-	-	-	96.6	-
MESDnet [24]		73.2	-	86.3	-	-	-	95.6	-
BAF [30]	\checkmark	82.7	89.3	92.3	90.4	-	-	98.7	99.7
SSMTL [32]	\checkmark	-	90.2	-	92.8	-	-	-	99.8
HF ² -VAD [21]	\checkmark	76.2	-	91.1	-	-	-	99.3	-
F ² PN [18]		73.0		85.7	-	-		96.2	-
LLSH [39]		77.6	85.9	87.4	88.6	73.5 [‡]	74.2 [‡]	-	-
sRNN-AE [56]		69.6	-	83.5	-	-	-	92.2	-
MPN [20]		73.8	-	89.5	-	-	-	96.9	-
SmithNet [19]		73.8	-	89.4	-	-	-	98.4	-
ROADMAP [23]		76.6	-	88.3	-	-	-	96.3	-
HSTGCNN [29]	\checkmark	81.8	-	87.5	-	70.5	-	97.7	-
SIGnet [25]		-	-	86.8	-	-	-	96.2	-
SSAGAN [27]		74.3	-	88.8	-	-	-	97.6	-
VABD [26]		78.2	-	86.6	-	72.2	-	97.1	-
STU-Net (ours)		79.7	87.6	87.2	88.2	74.9 [‡]	77.2 [‡]	95.9	97.4
iL ² SH (ours)		81.0	87.2	88.1	90.6	68.8 [‡]	65.6 [‡]	91.3	99.2
Two-stream (ours)		<u>83.7</u>	<u>90.8</u>	<u>90.8</u>	<u>93.0</u>	<u>78.3</u> ‡	<u>77.9</u> ‡	97.1	<u>99.3</u>

 TABLE VIII

 Results on Corridor without manually cropping

	Corr	idor
Method	Micro	Macro
STU-Net	69.5	63.6
iL ² SH	70.7	59.7
Two-stream	73.1	64.0

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 the recovery error in anomalous region and hence generate more accurate anomaly score. 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 achieves the state-of-the-art performance on ShanghaiTech, Avenue, Corridor and Ped2 datasets.

REFERENCES

- B. Ramachandra, M. J. Jones, and R. R. Vatsavai, "A Survey of Single-Scene Video Anomaly Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2293–2312, 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 *British Machine Vision Conference*, 2015, pp. 8.1–8.12.
- [3] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alexe, "Deep Appearance Features for Abnormal Behavior Detection in Video," in *Image Analysis and Processing*, 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. Comput. Vis. Pattern Recognit.*, 2019, pp. 7842–7851.
- [5] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Proc. Comput. Vis. Pattern Recognit.*, 2012, pp. 2112–2119.
- [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. Alexe, "Detecting Abnormal Events in Video Using Narrowed Normality Clusters," in *Proc. Winter Conf. Appl. Comput. Vis.*, 2019, pp. 1951–1960.
- [8] Z. Wang, Y. Zou, and Z. Zhang, "Cluster Attention Contrast for Video Anomaly Detection," in *Proc. ACM 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. Comput. Vis. Pattern Recognit.*, 2020, pp. 10536–10544.

- [10] Y. Chang, Z. Tu, W. Xie, and J. Yuan, "Clustering Driven Deep Autoencoder for Video Anomaly Detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, vol. 12360, pp. 329–345.
- [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, 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 Underst.*, vol. 195, p. 102920, 2020.
- [13] W. Liu, W. Luo, D. Lian, and S. Gao, "Future Frame Prediction for Anomaly Detection - A New Baseline," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 6536–6545.
- [14] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. van den Hengel, "Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 1705–1714.
- [15] G. Yu, S. Wang, Z. Cai, E. Zhu, C. Xu, J. Yin, and M. Kloft, "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. Comput. Vis. Pattern Recognit.*, 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.*, pp. 1–1, 2021.
- [19] T.-N. Nguyen, S. Roy, and J. Meunier, "SmithNet: Strictness on Motion-Texture Coherence for Anomaly Detection," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–14, 2021.
- [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. Int. Conf. Comput. Vis.*, 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.*, 2021, pp. 938–946.
- [23] X. Wang, Z. Che, B. Jiang, N. Xiao, K. Yang, J. Tang, J. Ye, J. Wang, and Q. Qi, "Robust Unsupervised Video Anomaly Detection by Multipath Frame Prediction," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–12, 2021.
- [24] Z. Fang, J. T. Zhou, Y. Xiao, Y. Li, and F. Yang, "Multi-Encoder Towards Effective Anomaly Detection in Videos," *IEEE Trans. Multimedia*, vol. 23, pp. 4106–4116, 2021.
- [25] Z. Fang, J. Liang, J. T. Zhou, Y. Xiao, and F. Yang, "Anomaly Detection With Bidirectional Consistency in Videos," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 3, pp. 1079–1092, 2022.
- [26] J. Li, Q. Huang, Y.-J. Du, X. Zhen, S. Chen, and L. Shao, "Variational Abnormal Behavior Detection With Motion Consistency," *IEEE Trans. Image Process.*, vol. 31, pp. 275–286, 2022.
- [27] C. Huang, J. Wen, Y. Xu, Q. Jiang, J. Yang, Y. Wang, and D. Zhang, "Self-Supervised Attentive Generative Adversarial Networks for Video Anomaly Detection," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 1–15, 2022.
- [28] S. Zhang, M. Gong, Y. Xie, A. K. Qin, H. Li, Y. Gao, and Y.-S. Ong, "Influence-aware Attention Networks for Anomaly Detection in Surveillance Videos," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1– 1, 2022.
- [29] 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.*, pp. 1–1, 2021.
- [30] M. I. Georgescu, R. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "A Background-Agnostic Framework with Adversarial Training for Abnormal Event Detection in Video," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021.
- [31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.
- [32] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "Anomaly Detection in Video via Self-Supervised and

Multi-Task Learning," in Proc. Comput. Vis. Pattern Recognit., 2021, pp. 12742–12752.

- [33] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-Cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1992–2004, 2017.
- [34] P. Wu, J. Liu, and F. Shen, "A Deep One-Class Neural Network for Anomalous Event Detection in Complex Scenes," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 7, pp. 2609–2622, 2020.
- [35] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. Comput. Vis. Pattern Recognit.*, 2010, pp. 1975–1981.
- [36] C. Lu, J. Shi, and J. Jia, "Abnormal Event Detection at 150 FPS in MATLAB," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 2720–2727.
- [37] W. Luo, W. Liu, and S. Gao, "A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 341–349.
- [38] R. Rodrigues, N. Bhargava, R. Velmurugan, and S. Chaudhuri, "Multitimescale Trajectory Prediction for Abnormal Human Activity Detection," in *Proc. Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2615–2623.
- [39] Y. Lu, C. Cao, and Y. Zhang, "Learnable Locality-Sensitive Hashing for Video Anomaly Detection," *CoRR*, vol. abs/2111.07839, 2021.
- [40] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *Proc. Comput. Vis. Pattern Recognit.*, 2020, pp. 9726–9735.
- [41] T.-R. Liu, Q. Meng, J. Huang, A. Vlontzos, D. Rueckert, and B. Kainz, "Video Summarization Through Reinforcement Learning With a 3D Spatio-Temporal U-Net," *IEEE Trans. Image Process.*, vol. 31, pp. 1573– 1586, 2022.
- [42] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, 2020.
- [43] B. Ramachandra, M. J. Jones, and R. R. Vatsavai, "Learning a distance function with a Siamese network to localize anomalies in videos," in *Proc. Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2587–2596.
- [44] T.-N. Nguyen and J. Meunier, "Anomaly Detection in Video Sequence With Appearance-Motion Correspondence," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 1273–1283.
- [45] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 4724–4733.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2016.
- [47] 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.
- [48] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *CoRR*, vol. abs/1607.06450, 2016.
- [49] H. Fan, Y. Li, B. Xiong, W.-Y. Lo, and C. Feichtenhofer, "Pyslowfast," https://github.com/facebookresearch/slowfast, 2020.
- [50] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," p. 4, 2017.
- [51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [52] R. Morais, V. Le, T. Tran, B. Saha, M. R. Mansour, and S. Venkatesh, "Learning Regularity in Skeleton Trajectories for Anomaly Detection in Videos," in *Proc. Comput. Vis. Pattern Recognit.*, 2019, pp. 11996– 12004.
- [53] Y. Lu, 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.
- [54] 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, 2020.
- [55] 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. 31, no. 9, pp. 3694–3706, 2021.
- [56] W. Luo, W. Liu, D. Lian, J. Tang, L. Duan, X. Peng, and S. Gao, "Video Anomaly Detection with Sparse Coding Inspired Deep Neural Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 1070–1084, 2021.