Adversarial 3D Convolutional Auto-Encoder for Abnormal Event Detection in Videos

Che Sun, *Student Member, IEEE*, Yunde Jia, *Member, IEEE*, Hao Song, and Yuwei Wu, *Member, IEEE*

***Abstract*—Abnormal event detection aims to identify the events that deviate from expected normal patterns. Existing methods usually extract normal spatio-temporal patterns of appearance and motion in a separate manner, which ignores low-level correlations between appearance and motion patterns and may fall short of capturing fine-grained spatio-temporal patterns. In this paper, we propose to simultaneously learn appearance and motion to obtain fine-grained spatio-temporal patterns. To this end, we present an adversarial 3D convolutional auto-encoder to learn the normal spatio-temporal patterns and then identify abnormal events by diverging them from the learned normal patterns in videos. The encoder captures the low-level correlations between spatial and temporal dimensions of videos, and generates distinctive features representing visual spatio-temporal information. The decoder reconstruccts the original video from the encoded features representing by 3D de-convolutions and learns the normal spatio-temporal patterns in an unsupervised manner. We introduce the denoising reconstruction error and adversarial learning strategy to train the 3D convolutional auto-encoder to implicitly learn accurate data distributions that are considered normal patterns, which benefits enhancing the reconstruction ability of the auto-encoder to discriminate abnormal events. Both the theoretical analysis and the extensive experiments on four publicly available datasets demonstrate the effectiveness of our method.**

***Index Terms*—Adversarial 3D convolutional auto-encoder, normal patterns, adversarial learning, abnormal event detection.**

I. Introduction

ABNORMAL event detection in videos has received much attention from both academia and industry [1]–[3]. Detecting abnormal events is still a challenging problem due to diverse events, lack of training data, and highly contextual definition of abnormal events in videos. Numerous efforts have devoted to dealing with these issues [4], [5]. A feasible solution is to learn normal patterns from training data and identify abnormal events deviated from the normal patterns [6], [7]. Recently, with the success of deep learning methods on various visual tasks [8]–[10], many researchers pay attention to learning normal spatio-temporal patterns via deep auto-encoders, such as fully connected auto-encoders [11], 2D convolutional auto-encoders [12], [13], convolutional long short-term memory (LSTM) auto-encoders [14], and so on. Most of them learn normal spatio-temporal patterns of appearance and motion separately. These methods, however, fail to obtain the fine-grained spatio-temporal patterns that usually occur at short intervals in local regions, limiting the improvement of detection performance.

In this paper, we propose a novel method that learns appearance and motion simultaneously to obtain fine-grained spatio-temporal patterns by performing information correlations on low-level pixel spaces. We build a 3D convolutional auto-encoder to learn the spatio-temporal patterns and train the auto-encoder by using the denoising reconstruction error and adversarial learning strategy. Specifically, given a video, the encoder of the 3D convolutional auto-encoder that consists of 3D convolutional layers captures the low-level appearance and motion simultaneously. It encodes correlations between spatial and temporal dimensions of videos into distinctive features representing visual spatio-temporal information. The decoder that consists of 3D de-convolution layers reconstructs the video from the learned features directly. Different from auto-encoders with recursive structures (e.g., LSTM [14]) which over-emphasize learning of temporal information [15], our method takes the spatial and temporal information into account simultaneously. The joint modeling method is well-suited for learning the subtle spatio-temporal patterns.

We introduce the denoising reconstruction error and adversarial learning strategy to enhance the reconstruction ability of the auto-encoder for learning better normal patterns. Specifically, we use the denoising reconstruction error to force the 3D convolutional auto-encoder to implicitly learn data distributions of normal data that are considered to be the normal patterns. During the process, some higher-order terms are omitted, which may bring extra errors. Thus, we introduce the adversarial learning strategy based on generative adversarial networks (GANs) [16] to train our auto-encoder using an extra discriminator to learn more accurate data distributions. The discriminator is designed to distinguish the reconstructed video from the original input video. The 3D convolutional auto-encoder is treated as the generator, which aims at reconstructing realistic videos to confuse the discriminator. Compared with the traditional training method of auto-encoders using denoising reconstruction errors [17], the well-trained discriminator using adversarial learning strategy

Manuscript received January 6, 2020; revised May 30, 2020 and August 3, 2020; accepted August 30, 2020. Date of publication September 10, 2020; date of current version September 24, 2021. This work was supported by the Natural Science Foundation of China under Grants 61702037 and 61773062. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianfei Cai. *(Corresponding author: Yuwei Wu.)*

The authors are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 10081, China (e-mail: sunche@bit.edu.cn; jiayunde@bit.edu.cn; songhao@bit.edu.cn; wuyuwei@bit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2020.3023303.

Digital Object Identifier 10.1109/TMM.2020.3023303

without variational lower bound will help the 3D convolutional auto-encoder perfectly recover more accurate distributions.

We give a rigorous theoretical analysis of the effectiveness of our method in abnormal event detection. Experiments on four challenging datasets demonstrate that our method can achieve promising results compared with state-of-the-art methods.

The contributions of this paper are summarized as follows:

- We propose a novel anomaly detection method that learns appearance and motion simultaneously to obtain fine-grained spatio-temporal patterns. Our method performs information correlations on low-level pixel spaces, which can be helpful to distinguish abnormal events in videos.
- We build a 3D convolutional auto-encoder to learn fine-grained spatio-temporal patterns directly from both spatial and temporal dimensions to capture subtle appearance and motion changes. We introduce denoising reconstruction errors and the adversarial learning strategy to help the auto-encoder learn robust normal patterns.
- We conduct a rigorous theoretical analysis of the effectiveness of integrating denoising reconstruction errors with adversarial learning to learn normal patterns from the perspective of data distributions.

The rest of this paper is organized as follows. In section II, we review related work of unsupervised abnormal event detection in videos. Section III presents the framework of simultaneously learning appearance patterns and motion patterns. Section IV presents experimental settings and evaluation results. This paper is concluded in Section V.

II. Related Work

There has been much work on abnormal event detection in surveillance videos [5], [12], [18]. Since the abnormal events are of infrequency and ambiguity [19], [20], most existing methods adopt an unsupervised learning scheme to learn normal patterns. In this paper, we also focus on the unsupervised abnormal event detection task.

Deep learning has been successfully applied to abnormal event detection in videos [21]–[23]. The deep methods of abnormal event detection can be roughly divided into classification-based methods [13], [24], [25] and reconstruction-based methods [12], [26], [27]. Classification-based methods treat abnormal event detection as a classification problem, and the classification task is performed with deep features. Zhou *et al.* [24] built a spatio-temporal convolutional neural network (spatio-temporal CNN) to identify abnormal events. Ionescu *et al.* [13] formulated the abnormal event detection task as a multi-class problem based on object detection results.

Reconstruction-based methods learn normal patterns and distinguish abnormal events through reconstruction errors. Benefiting from the promising representation capabilities of auto-encoders [17], [28], [29], many methods use auto-encoders to reconstruct the input video sequence, and distinguish abnormal events through reconstruction errors. Hasan *et al.* [12] used a fully connected auto-encoder and an end-to-end 2D convolutional auto-encoder to learn regular dynamics respectively and identify irregularity far from the regular dynamics. Their work stacks multiple frames into different channels, without effectively modeling the temporal relationship between sequential frames. Chong and Tay [14] employed a spatio-temporal architecture for abnormal event detection, which consists of two components, i.e., one for spatial feature representation and the other for learning the temporal evolution. Since this method separately models spatio-temporal relationships, it is ineffective to represent the location-variant relationships, e.g., rotation and scaling. Medel and Savakis [30] introduced a composite convolutional LSTM network to predict the evolution of a video sequence, and detected anomalous video segments using a regularity evaluation algorithm. Most of these methods use 2D convolutional layers [12] or 2D convolutional LSTM layers [14], [30] to formulate auto-encoders for abnormal event detection. These methods, however, do not fully exploit low-level appearance and motion cues, which restricts the improvement of abnormal detection performance. Different from these reconstruction-based methods, our method learns the intrinsic normal spatio-temporal patterns by simultaneously taking the spatial and temporal information into account, which is better suitable for learning normal appearance-changing and motion-changing patterns for abnormal event detection in videos.

Recently, due to the good performance of GANs [16], some methods employ adversarial learning in abnormal event detection [11], [31], [32]. Ravanbakhsh *et al.* [11] proposed a reconstruction model as a generator inspired by the conditional GANs [33], and used the reconstruction error to differentiate normal and abnormal events. Similar to the work of [11], Sabokrou *et al.* [31] introduced an adversarial model and used both generator and discriminator to detect and fine-segment abnormal events. Schlegl *et al.* [34] learned a manifold of normal objects and calculated the anomaly scores based on the mapping from image space to a latent space using GANs. We use the adversarial learning to directly guide auto-encoders to learn normal patterns from the perspective of data distributions. We use denoising errors to force auto-encoders to implicitly model data distributions. The adversarial learning without variational lower bound can reduce extra errors caused by omitted higher-order terms of denoising reconstruction errors, thereby helping the auto-encoders learn more accurate data distributions (i.e., better normal patterns) to improve abnormal detection performance.

III. Method

In this paper, we introduce an abnormal event detection method of simultaneously learning normal appearance patterns and motion patterns to learn robust normal spatio-temporal patterns. As shown in Fig. 1, we build an adversarial 3D convolutional auto-encoder to encode normal patterns of video sequences with a small reconstruction error. The adversarial learning with an extra discriminator helps the 3D convolution auto-encoder learn normal patterns better and distinguish normal and abnormal events without any supervised information. The adversarial 3D convolutional auto-encoder is able to implicitly learn the characteristics that reflect the accurate distributions. The intuition is that the implicit distributions are considered the normal patterns, which can be used to identify abnormal events

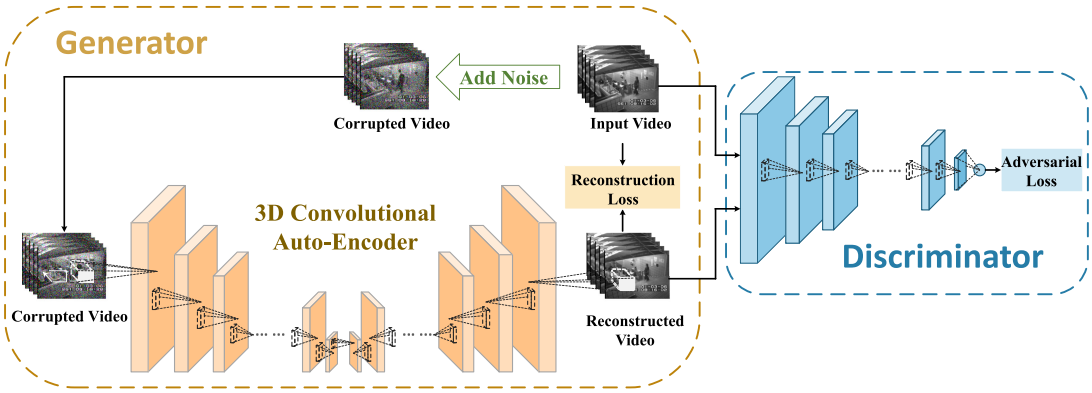


Fig. 1. Overview of our method. A video corrupted with random Gaussian noise is input to the 3D convolutional auto-encoder. The auto-encoder is utilized to learn normal patterns by the reconstruction loss. It is also treated as a generator competing with the discriminator. The adversarial loss with respect to the discriminator is introduced to help the auto-encoder learn the accurate normal patterns.

deviating from the learned normal pattern. Moreover, we add random Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$ to the input video to prevent the auto-encoder from learning an identity function.

A. Problem Statement

Given a video S, conventional reconstruction-based methods use a reconstruction model (e.g., an auto-encoder) by only using normal videos, and then use the reconstruction error e to discriminate abnormal events. e is calculated by

$$e = \|S - R(S)\|^2, \tag{1}$$

where $R(\cdot)$ denotes the reconstruction model. The intuition is that the reconstruction model trained by normal events is able to reconstruct normal events well but fails to reconstruct abnormal events. Therefore, these methods learn the normal patterns and detect abnormal events through the reconstruction error e.

In this paper, we learn the intrinsic normal spatio-temporal patterns by simultaneously learning appearance patterns and motion patterns. To learn the normal patterns, we build a 3D convolutional auto-encoder as the reconstruction model

$$R_\theta(S) = G_{\theta_2}\left(F_{\theta_1}(S)\right), \tag{2}$$

where $F_{\theta_1}(\cdot)$ denotes the 3D convolutional encoder with the parameter θ_1, $G_{\theta_2}(\cdot)$ stands for the 3D convolutional decoder with the parameter θ_2, and $R_\theta(\cdot)$ represents the reconstruction model with the parameter $\theta = \{\theta_1, \theta_2\}$. The 3D convolutional encoder $F_{\theta_1}(\cdot)$ learns the low-level correlations between spatial and temporal dimensions of videos, and generates distinctive features representing visual spatio-temporal information. The 3D de-convolutional decoder $G_{\theta_2}(\cdot)$ reconstructs the video from the learned features directly. The 3D convolutional auto-encoder takes the spatial and temporal information into account simultaneously, which is well-suited for learning the intrinsic normal spatio-temporal patterns.

To solve the optimal parameter θ of $R_\theta(\cdot)$, we use the denoising reconstruction error [17] to train the 3D convolutional auto-encoder. The auto-encoder implicitly learns the normal data distributions that are considered normal patterns for detecting abnormal events. Specifically, we add random Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$ to the input video S to generate the corrupted input video $\tilde{S} = S + \eta$. Then the 3D convolutional auto-encoder reconstructs the corrupted video $\tilde{S}$, and outputs the reconstructed one $R(\tilde{S})$. According to the theory in the work of Alain and Bengio [17], the optimal output of the 3D convolutional auto-encoder with the denoising reconstruction error is calculated in the following theorem.

Theorem 1: Assume that $p(S)$ represents the probability density function of the training (normal) data S. The optimal output of the 3D convolutional auto-encoder trained by the denoising reconstruction error is

$$R^*_\theta(\tilde{S}) = \frac{\mathbb{E}_\eta\left[p(\tilde{S} - \eta)(\tilde{S} - \eta)\right]}{\mathbb{E}_\eta\left[p(\tilde{S} - \eta)\right]},$$
$$\text{s.t. For } \forall x,\ p(x) \neq 0. \tag{3}$$

The proof of Theorem 1 is given in the Appendix. From Eq. (3), the optimal output $R^*_\theta(\tilde{S})$ refers to a kind of weighted average around the data point $\tilde{S}$. If the noise level is high (the standard deviation σ of the Gaussian noise is set to a large value), the output $R^*_\theta(\tilde{S})$ will produce blurry results from its neighbors with Gaussian noise. Thus, the reconstruction error $R_\theta(\tilde{S}) - S$ with high level noise cannot be directly used to discriminate abnormal events. On the contrary, a small value of σ should be selected, and we can calculate the optimal output of the trained 3D convolutional auto-encoder in Theorem 2.

Theorem 2: Assume that $p(S)$ represents the probability density function of the training data S. When the noise level σ asymptotically approaches to 0 ($\sigma \to 0$), the optimal output is

$$R^*_\theta(S) = S + \sigma^2 \frac{\partial \log p(S)}{\partial S} + o(\sigma^2). \tag{4}$$

The proof of Theorem 2 is found in the Appendix. From Eq. (4), we find that the reconstruction error $R^*_\theta(S) - S \propto \frac{\log \partial p(S)}{\partial S}$ defines a local vector field for the trained 3D convolutional auto-encoder, and the vector field points towards the nearest

high-density point on the data manifold. Since we train the 3D convolutional auto-encoder only with normal videos, the normal samples S_n are dense to the distribution $p(S_n)$, and the abnormal ones S_a are sparse (or even zero). Therefore, we will get larger reconstruction errors of abnormal samples than that of normal ones. In other words, the reconstruction model trained by the denoising reconstruction error learns the normal patterns that implicitly capture the characteristics of the data distributions. This proves that abnormal events can be effectively detected through the reconstruction error $R^*_\theta(S) - S$.

Furthermore, to better recover the data distribution $p(S)$, we introduce the adversarial learning strategy [16] with an extra discriminator to train the 3D convolutional auto-encoder. The auto-encoder is treated as the generator. According to the theory in [16], the well-trained discriminator using the adversarial learning strategy without variational lower bound will help the 3D convolutional auto-encoder perfectly recover a more accurate distribution $p(S)$ of normal data.

Theorem 3: Let $p(S)$ be the data distribution of normal events and $p_r(R^*_\theta(\tilde{S}))$ be the distribution of the reconstructed data, where $R^*_\theta(\tilde{S})$ denotes the optimal output of the 3D convolutional auto-encoder in Eq. (4). For simplicity, we assume that $D_\phi(\cdot)$ represents the mapping of the discriminator, where ϕ is the parameter of the discriminator. The global optimal solution of the discriminator D^*_ϕ is

$$D^*_\phi(S) = \frac{p(S)}{p(S) + p_r(R^*_\theta(\tilde{S}))}. \tag{5}$$

With the optimal discriminator D^*_ϕ, the adversarial loss is equivalent to

$$C(R_\theta(\tilde{S})) = -\log 4 + 2 \times JSD\left(p(S) \| p_r(R^*_\theta(\tilde{S}))\right), \tag{6}$$

where JSD represents Jensen-Shannon divergence [35]. Since the JSD is non-negative and $JSD(p\|q) = 0 \Leftrightarrow p = q$, the optimal R^* is obtained when $p_r(R^*_\theta(\tilde{S})) = p(S)$.

We refer the readers to the work of Goodfellow *et al.* [16] for a detailed proof of Theorem 3. Obviously, the adversarial learning forces the reconstructed samples $R^*_\theta(\tilde{S})$ of the auto-encoder to follow the true data distribution $p(S)$. From Theorem 1, we know that the optimal output $R^*_\theta(\tilde{S})$ without the adversarial learning refers to a kind of weighted average around the data point $\tilde{S}$. The adversarial learning in Theorem 3 guarantees the output $R^*_\theta(\tilde{S}) \sim p_r(R^*_\theta(\tilde{S})) = p(S)$, which makes the reconstructed output $R^*_\theta(\tilde{S})$ of normal data more authentic and indistinguishable from the raw data S. For any given abnormal sample S_a that does not follow $p(S)$, the auto-encoder trained by the adversarial learning strategy may not guarantee to map $R^*_\theta(S_a)$ to $p(S)$, which will make its reconstruction error larger.

B. Network Architecture

We build the adversarial 3D convolutional auto-encoder as the reconstruction model R_θ to learn the intrinsic normal spatio-temporal patterns. As depicted in Fig. 1, the adversarial 3D convolutional auto-encoder consists of three subnetworks: the 3D convolutional encoder F_{θ_1}, the 3D convolutional decoder G_{θ_2} and the discriminator D_ϕ.

3D Convolutional Encoder: To reconstruct an input video at the pixel level, motivated by 3D convolutional neural networks [36], several 3D convolutional layers and 3D max-pooling layers are used to encode spatio-temporal structures of the input video. The set of learnable parameters of the encoder is represented as θ_1 that requires to be solved. The 3D convolution operation maintains the spatio-temporal relationships between pixels by learning video features using small cuboids of the input video. The low-level appearance and motion information in the cuboids is extracted by the 3D convolutional encoder simultaneously. With several spatio-temporal convolutional layers, the input video can be effectively encoded into informative feature maps. Moreover, the 3D max-pooling layer is used for translation invariance, rotation invariance, and scale invariance. The 3D encoder composed of both 3D convolutional layers and 3D max-pooling layers produces the informative feature maps with indispensable spatio-tempral information for reconstruction.

3D De-Convolutional Decoder: The decoder consists of 3D de-convolutional layers and 3D un-pooling layers, and has the symmetrical structure with the encoder. θ_2 denotes the set of learnable parameters of the decoder. The 3D de-convolution utilizes 3D convolution-like operations to get the no-sparse cuboid. After de-convolution, the cuboid is larger than the original input video when the filter is multiplied by the input video at the boundary, so we follow the operation in the work of Hasan *et al.* [12], cropping out the boundary of the output to keep the same size of the previous layer. The 3D de-convolutional layers in the decoder directly reconstruct the input video with the learned 3D filters for modeling the normal appearance patterns and motion patterns simultaneously.

In the encoding phase, 3D max-pooling layers are adopted for the purpose of invariance, and they may lose some spatio-temporal information. To reconstruct the input video, the corresponding 3D un-pooling layers are applied. The 3D un-pooling is the reverse operation of the 3D pooling and restores the original size of the features maps. The 3D un-pooling is formulated in a similar way to the 2D un-pooling [37], which records the locations of maximum input selected during a 3D max-pooling operation. The recorded locations are used to place each input back to the original location.

The structure of the 3D convolutional auto-encoder is illustrated in Fig. 2. Following the C3D net [36], all 3D convolution kernels are set to $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions, except that the kernel of the first convolutional layer is $1 \times 2 \times 2$. All pooling kernels are set to $2 \times 2 \times 2$, except that the kernel of the first pooling layer is $1 \times 2 \times 2$. The decoder has a symmetrical structure of the encoder. The dashed lines refer to the locations of recorded maximum input in each sliding window during the max-pooling, which are used for un-pooling.

Discriminator: The adversarial learning strategy is realized through two competing neural networks: a generator and a discriminator. They compete with each other in a two-player game, where the generator is used to produce samples as realistic as possible to confuse the discriminator. The discriminator computes the probability that a sample comes from true observations.

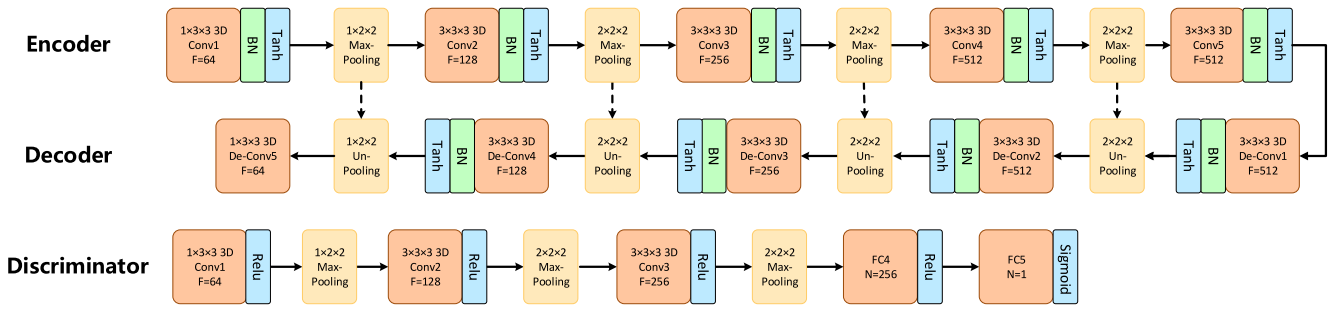


Fig. 2. Illustration of the network structure. The encoder consists of five 3D convolutional layers and four 3D max-pooling layers. Each convolutional layer is followed by a Batch Normalization (BN) layer and a Tanh activation function (Tanh) layer. The decoder has a symmetrical structure of the encoder. The dashed lines refer to the locations of recorded maximum input in each sliding window during the max-pooling, which are used for un-pooling. The discriminator has three 3D convolutional layers and three 3D max-pooling layers, followed by two fully connected (FC) layers.

Specifically, the generator of our network is constructed based on the 3D convolutional auto-encoder $R_\theta(\cdot)$, which takes the corrupted video $\tilde{S}$ as the input. The discriminator $D_\phi(\cdot)$ consisting of several 3D convolutional layers aims to distinguish whether the video is true or generated, where ϕ stands for all learnable parameters of the discriminator. The structure of the discriminator is exhibited in Fig. 2, where "FC" means a fully connected layer. The inputs to the discriminator are the original input video S and the reconstructed video $R_\theta(S)$. The output of the discriminator is the probability that the input video comes from true observations.

C. Optimization Objective

In order to train the adversarial 3D convolutional auto-encoder, a denoising reconstruction loss and an adversarial loss are introduced. The denoising reconstruction loss is based on the Euclidean distance between the input video and the reconstructed video,

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_S \left[\|R_\theta(S+\eta) - S\|^2 \right], \tag{7}$$

where S and $R_\theta(S+\eta)$ indicate the original input video and the corresponding output video, respectively. $\mathbb{E}$ is the empirical estimation of the expected value of the probability, which is realized by sampling from training data. $R_\theta(\cdot)$ denotes the 3D convolutional auto-encoder with the parameter θ, and $\eta \sim \mathcal{N}(0, \sigma^2 I)$ represents the isotropic Gaussian noise to partly corrupt the input video.

The adversarial loss $\mathcal{L}_{\text{adv}}$ is defined as

$$\begin{aligned}\mathcal{L}_{\text{adv}} = \min_\theta \max_\phi \big(& \mathbb{E}_S \left[\log \left(D_\phi(S) \right) \right] \\ & + \mathbb{E}_S \left[\log \left(D_\phi(R_\theta(S+\eta)) \right) \right] \big),\end{aligned} \tag{8}$$

where $D_\phi(\cdot)$ denotes the discriminator with the parameter ϕ.

We combine the denoising reconstruction loss $\mathcal{L}_{\text{rec}}$ and the adversarial loss $\mathcal{L}_{\text{adv}}$ to solve the optimal parameter θ of the 3D convolutional auto-encoder $R_\theta(\cdot)$, and arrive at the following objective function

$$\mathcal{L}_{\text{R}} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{adv}}, \tag{9}$$

where λ is the trade-off parameter. When we train D_ϕ, we use the loss function

$$\mathcal{L}_{\text{D}} = \mathcal{L}_{\text{adv}}. \tag{10}$$

Following [16], [38], [39], we train θ by using the gradient descent method of $-\nabla_\theta(\mathcal{L}_{\text{R}})$, and update ϕ by using the gradient ascent method of $+\nabla_\phi(\mathcal{L}_{\text{D}})$. The gradient of θ is calculated by

$$\begin{aligned}\nabla_\theta(\mathcal{L}_{\text{R}}) &= \frac{\partial \mathcal{L}_{\text{rec}}}{\partial \theta} + \lambda \frac{\partial \mathcal{L}_{\text{adv}}}{\partial \theta} \\ &= \mathbb{E}_S \Bigg[2(R_\theta(S+\eta) - S) \frac{\partial R_\theta(S+\eta)}{\partial \theta} \\ &\quad + \frac{\lambda}{D_\phi(R_\theta(s+\eta))} \frac{\partial D_\phi(R_\theta(s+\eta))}{\partial R_\theta(S+\eta)} \\ &\quad \times \frac{\partial R_\theta(S+\eta)}{\partial \theta} \Bigg],\end{aligned} \tag{11}$$

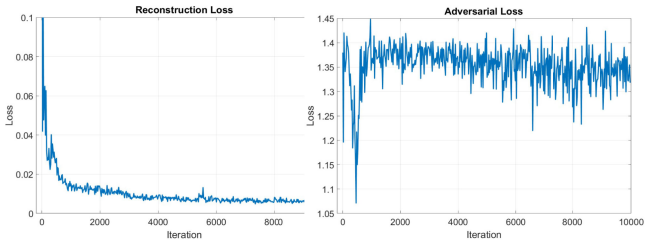
where $\frac{\partial D_\phi(R_\theta(s+\eta))}{\partial R_\theta(S+\eta)}$ denotes the partial derivative of the output of the discriminator $D_\phi(\cdot)$ relative to its input $R_\theta(S+\eta)$, and $\frac{\partial R_\theta(S+\eta)}{\partial \theta}$ represents the partial derivative of the output of the auto-encoder $R_\theta(S+\eta)$ relative to the parameter θ. We refer the readers to [40] for the detailed calculation process of gradients for convolutional networks.

The gradient of ϕ is calculated by

$$\begin{aligned}\nabla_\phi(\mathcal{L}_{\text{D}}) = & \mathbb{E}_S \left[\frac{1}{D_\phi(S)} \frac{\partial D_\phi(S)}{\partial \phi} \right] \\ & + \mathbb{E}_S \left[\frac{1}{D_\phi(R_\theta(S+\eta))} \frac{\partial D_\phi(R_\theta(S+\eta))}{\partial \phi} \right],\end{aligned} \tag{12}$$

where $\frac{\partial D_\phi(S)}{\partial \phi}$ represents the partial derivative of the output of the discriminator $\partial D_\phi(S)$ relative to the parameter ϕ, and $\frac{\partial D_\phi(R_\theta(S+\eta))}{\partial \phi}$ denotes the partial derivative of the output of the discriminator $D_\phi(R_\theta(S+\eta))$ relative to the parameter ϕ.

We update the parameters iteratively using the calculated gradients, and the adversarial learning algorithm of our method is summarized in Algorithm 1.



(a) (b)

Fig. 3. Training loss curves on the ShanghaiTech dataset. (a) Reconstruction loss $\mathcal{L}_{\mathrm{rec}}$; (b) Adversarial loss $\mathcal{L}_{\mathrm{adv}}$.

Algorithm 1: The Adversarial Learning Algorithm of Our Method.

Input: Training videos $\{S_1, S_2, \ldots, S_N\}$;
Output: The parameters $\{\theta, \phi\}$;
1: Initialize the parameters $\{\theta, \phi\}$;
2: **for** iteration number **do**
3: Sample a batch of training data and corrupt each sample S_i with random Gaussian noise $\eta_i \sim \mathcal{N}(0, \sigma^2 I)$ independently: $S_i + \eta_i$;
4: Calculate the output of the 3D convolutional auto-encoder: $\hat{S} = R_\theta(S + \eta)$;
5: Calculate the loss $\mathcal{L}_{\mathrm{rec}}$, $\mathcal{L}_{\mathrm{adv}}$;
6: Update $\{\theta, \phi\}$ using Stochastic Gradient:

$$\theta \leftarrow -\nabla_\theta(\mathcal{L}_{\mathrm{R}}),$$

$$\phi \leftarrow +\nabla_\phi(\mathcal{L}_D);$$

7: **return** The parameters $\{\theta, \phi\}$.

We use Algorithm 1 to train our network, and show the training loss curves of the reconstruction loss and the adversarial loss on the ShanghaiTech dataset in Fig. 3. We find that the reconstruction loss quickly converges to a relatively small value. The average values of adversarial loss fluctuate at $2 \ln 2$, which means the discriminator is fairly confused between reconstructed videos and input videos.

D. Anomaly Score

In the testing procedure, with one forward pass, the average reconstruction error e_t of all the pixel values in the frame t is computed by the Euclidean distance between the input video $S \in \mathbb{R}^{T \times W \times H}$ and the reconstructed video $R^*_\theta(S) \in \mathbb{R}^{T \times W \times H}$, where T is the number of frames in a video, H and W are the height and width of the frame. The reconstruction error of the t-th frame is calculated by

$$e_t = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \left\| R^*_\theta(S)[t,i,j] - S[t,i,j] \right\|^2, \quad (13)$$

where $S[t,i,j]$ represents the pixel value at the position $[i,j]$ in the t-th frame of the video S. It should be noted that similar to the denoising auto-encoder [41], we do not add Gaussian noise when testing. Then the calculated Euclidean distances of each frame are normalized to the range of [0,1) and the abnormal events are detected with a larger anomaly score. The anomaly score s_t of the frame t is given by

$$s_t = \frac{e_t - \min_t e_t}{\max_t e_t}, \quad (14)$$

where $\min_t e_t$ is calculated by selecting the minimum reconstruction error among all frames in a video and $\max_t e_t$ denotes the maximum frame-level reconstruction error in a video.

To detect abnormal events according to anomaly scores, we select the local maximum in the time series of anomaly scores in a video. Specifically, we use the persistence1D algorithm [42] to identify the meaningful local maximum and span the region with a fixed temporal window. We follow the work of [12] to group nearby overlapped local maximum regions to obtain the final abnormal temporal regions, where abnormal events are localized into the temporal regions.

IV. Experiments

A. Datasets

We evaluate our method on four benchmark datasets: the Subway [43], UCSD [44], Avenue [45], and ShanghaiTech [46] datasets. All the training videos in the experiments consist only of normal events.

The Subway dataset contains two scenarios: the entrance (1 hour 36 minutes with 144249 frames) and exit (43 minutes with 64900 frames) gates. The types of abnormal events are walking in the wrong direction, no payment, loitering, irregular interactions between people, and miscellaneous.

The UCSD dataset consists of two sub-datasets: Ped1 and Ped2, which records the pedestrian walkways. The abnormal events of these two datasets include carts, cars, person skating, bicycling among pedestrians, etc.

The Avenue dataset has 16 training and 21 testing videos with 35240 frames, totally. Each video lasts about 2 minutes long. The abnormal events are running, walking in opposite directions, throwing objects, loitering, etc.

The ShanghaiTech dataset contains 13 scenes with complex light conditions and various viewpoints. This dataset has 130 abnormal events and over 270,000 training frames.

B. Evaluation Metric

We apply the Receiver Operating Characteristic (ROC) by gradually changing the threshold of anomaly scores. Then we calculate the corresponding Area Under Curve (AUC↑) as the evaluation metric, which is commonly used in abnormal event detection. Moreover, the Equal Error Rate (EER↓) [12] is introduced to evaluate the equal probability of miss-classifying a positive or negative sample in the ROC curve. We evaluate our method based on the frame level. Besides, we select the suitable threshold of the anomaly score to detect abnormal events and evaluate our method with the event counts. We count the number of detected abnormal events (True Positives ↑) and the number of detected non-abnormal events (false alarm ↓) for evaluation. Here, ↑ implies that higher scores represent better performance, and ↓ indicates that lower is better.

TABLE I
ABNORMAL EVENT DETECTION RESULTS IN TERMS OF FRAME-LEVEL AUC AND EER ON THE AVENUE DATASET

Method	AUC↑/ EER↓ (%)
Tudor *et. al* [47]	80.6/−
Morais *et al.* [48]	86.3/−
Luo *et. al* [46]	81.7/−
Liu *et. al* [49]	84.9/−
Chong and Tay [14]	80.3/20.7
Hasan *et. al* [12]	70.2/25.1
Wang *et. al* [50]	85.3/23.9
Ours	**88.9/18.2**

TABLE II
ABNORMAL EVENT DETECTION RESULTS IN TERMS OF FRAME-LEVEL AUC AND EER ON THE UCSD DATASET

Method	AUC↑/EER↓ (%)		
	UCSD (Ped1)	UCSD (Ped2)	Average
Mahadevan *et. al* [44]	74.2/32.0	61.3/36.0	67.8/34.0
Hasan *et. al* [12]	81.0/27.9	90.0/21.7	85.5/24.8
Chong and Tay [14]	89.9/12.5	87.4/12.0	88.7/12.3
Adam *et. al* [43]	77.1/38.0	−/42.0	−/40.0
Kim and Grauman [54]	59.0−	69.3/−	64.1/−
Mehran *et. al* [53]	67.5/31.0	55.6/42.0	61.6/36.5
Ravanbakhsh *et al.* [11]	**97.4/8.0**	**93.5**/14.0	**95.5/11.0**
Ours	90.2/11.6	91.0/**10.9**	90.6/11.3

TABLE III
ABNORMAL EVENT DETECTION RESULTS IN TERMS OF FRAME-LEVEL AUC AND EER ON THE SUBWAY DATASET

Method	AUC↑/EER↓ (%)		
	Subway (Entrance)	Subway (Exit)	Average
Mehran *et. al* [53]	67.5/31.0	55.6/42.0	61.6/36.5
Wang *et. al* [55]	81.6/22.8	84.9/17.8	83.3/20.3
Xu *et. al* [56]	−/−	87.9/**6.8**	−/−
Hasan *et. al* [12]	**94.3**/26.0	80.7/9.9	87.5/18.0
Ionescu *et. al* [47]	70.6/−	85.7−	78.2/−
Wang *et. al* [50]	−/−	84.5/21.4	−/−
Chong and Tay [14]	84.7/23.7	94.0/9.5	89.4/16.6
Ours	90.5/**22.77**	**94.8**/9.6	**92.7/16.2**

C. Implementation

We first extract frames of gray images from videos and resize them to 224×224. Then we normalize the images. The pixel values of each image are subtracted from its global mean image calculated by averaging the pixel values of each frame in the training set. We follow the data augmentation algorithm in [12] to generate videos containing 16 frames for training and testing. We set the standard deviation σ of Gaussian noise to 0.005 and 0 during training and testing, respectively.

In the 3D convolutional auto-encoder, a Batch Normalization (BN) [51] layer and a Tanh activation layer follow each convolutional or de-convolutional layer, except for the last layer of the decoder. The discriminator is constructed with several 3D convolutional layers and fully connected layers, without any BN layers. The auto-encoder and the discriminator are both trained with RMSprop optimizer with 0.0002 learning rate and other default parameters, where we use the PyTorch toolkit [52] to implement the proposed network.

D. Comparisons With Existing Methods

Tables I, II, III and IV show the quantitative comparisons of our method with several state-of-the-art methods on four datasets regarding the AUC and the EER. The performance of the compared methods is taken from the original papers. Among these works, some of them are not evaluated on all of these four datasets [49], [53], and several methods do not report the result of the EER [46], [49].

1) Results on the Avenue Dataset: From Table I, we can see that our method outperforms all existing methods on both the AUC and EER evaluations in the Avenue dataset. The performances of existing state-of-the-art methods are shown in Table I, where Liu *et al.* [49], Wang *et al.* [6] and Morais *et al.* [48] achieve the AUC of 84.9%, 85.3%, and 86.4%, respectively. In comparison, our method gains a relatively significant improvement of 2.6% on the AUC evaluation compared with the state-of-the-art method [48] and an improvement of 2.5% on the EER evaluation, which verifies that our method is effective and robust.

2) Results on the UCSD Dataset: Table II reports experiment results on the UCSD dataset. Our method performs better than most existing methods except the method in [11], because Ravanbakhsh *et al.* [11] use both RGB images and optical-flow images as the input, while we only use RGB images as input. It is noted that the work of [11] also reports the AUC score of 84.1% when only using RGB images on the Ped1 dataset, and we achieve the AUC score of 90.2% that is 6.1% higher than their work. This can verify the effectiveness of our method when only using RGB as the input. Furthermore, we investigate the improvement brought by optical-flow images in the later section.

3) Results on the Subway Dataset: Table III compares evaluations of AUC and EER between our method and other state-of-the-art methods on the Subway dataset. We observe that our method obtains the best results on the Subway Exit dataset. Although our method achieves less AUC of 90.2% on the Subway Entrance dataset compared with [12], our method significantly outperforms the work of [12] on the Subway Exit dataset, where the AUC evaluation has increased from 80.7% to 94.6% with a gain of 13.9%. From the overall AUC performance, our method also outperforms the state-of-the-art methods [12], [14], with significant gains of 5.2% and 3.3% respectively.

4) Results on the ShanghaiTech Dataset: Table IV presents the evaluation of our method on the ShanghaiTech Dataset. The ShanghaiTech dataset has complex scenes and various actions, which are recognized challenges of abnormal event detection. Our method performs best with the grains of 1.2% compared with the state-of-the-art method [48], which clearly validates the effectiveness and robustness of our method.

TABLE IV
ABNORMAL EVENT DETECTION RESULTS IN TERMS OF FRAME-LEVEL AUC ON THE SHANGHAITECH DATASET

Method	AUC↑/ EER↓ (%)
Hasan *et. al* [12]	60.85/−
Luo *et. al* [46]	68.00/−
Liu *et al.*[49]	72.8/−
Morais *et al.* [48]	73.4/−
Ours	**74.6/27.6**

TABLE V
THE AUC AND EER RESULTS OF DIFFERENT COMPONENTS OF THE ADVERSARIAL 3D CONVOLUTIONAL AUTO-ENCODER ON THE FOUR PUBLIC DATASETS

Method	AUC↑/EER↓ (%)					
	ShanghaiTech	Avenue	UCSD (Ped1)	UCSD (Ped2)	Subway (Entrance)	Subway (Exit)
w/o 3D convolution	64.2/33.7	82.2/27.5	83.6/16.1	89.7/17.0	89.3/24.6	82.3/20.8
w/o 3D pooling	65.1/34.2	81.1/25.7	84.1/18.0	86.7/16.5	87.5/26.4	84.1/21.9
w/o GAN	70.0/29.2	86.6/20.2	88.7/16.3	90.3/17.8	90.9/23.2	94.6/10.2
w/o recording locations	−/−	87.9/19.9	90.0/11.7	88.3/12.5	−/−	−/−
Ours	**74.6/27.6**	**88.9/18.2**	**90.2/11.6**	**91.0/10.9**	**92.2/21.3**	**95.0/8.8**

E. More Evaluation of the Proposed Method

1) Ablation Study: Table V shows the contributions of different components in our method. "w/o 3D convolution" stands for replacing 3D convolutional layers and 3D pooling layers with 2D convolutional layers and 2D pooling layers, respectively, where temporal frames are stacked into the dimensions of channels. "w/o 3D pooling" represents replacing the 3D pooling operations with 2D pooling operations, where all sliding windows are set to $1 \times 2 \times 2$. "w/o GAN" refers to training the 3D convolutional auto-encoder without the adversarial learning strategy. "w/o recording locations" means that, during un-pooling, we put the input response values into the fixed upper-left positions of the sliding window instead of the locations recorded by max-pooling. From Table V, it is interesting to observe that: (1) The performance is significantly improved by employing the 3D convolutional layers, since the 3D convolutional layers can capture the spatio-temporal structures within videos simultaneously. (2) When discarding the 3D pooling, our method drops more than 5%, probably because our auto-encoder does not maintain the bottleneck structure in the time dimension without 3D pooling, which makes it difficult to effectively model the implicit distribution of sequence data. (3) Adopting the adversarial learning strategy can improve the performance, since the adversarial learning strategy enhances discrimination of our method. (4) The recorded locations of maximum input for un-pooling can improve the reconstruction performance. Due to the small size of the sliding window ($2 \times 2 \times 2$), the contribution of the recorded locations during max-pooling may not be as great as that of other components of our methods, such as "GAN," "3D convolution," etc.

To further verify the effect of the adversarial learning strategy, we calculate the aforementioned gap (Δ_s) proposed by [49] between normal and abnormal scores. The larger gap represents the more separability between normal and abnormal frames. Fig. 4 shows that the adversarial learning strategy ensures the discrimination of our method, which is more suitable for abnormal event detection.

2) Evaluation of Different Noise: We evaluate the contribution of the noise in our method, as shown in Table VI. "w/o noise" denotes that we do not add any noise to the input video during training. "salt-and-pepper noise" represents that we use the salt-and-pepper noise, where the signal to noise ratio (SNR) is set to 95%. "Poisson noise" means that we add Poisson noise to each frame in a video sequence. We observe that the performance of Gaussian noise and salt-and-pepper noise is similar, and is better than other format noise.

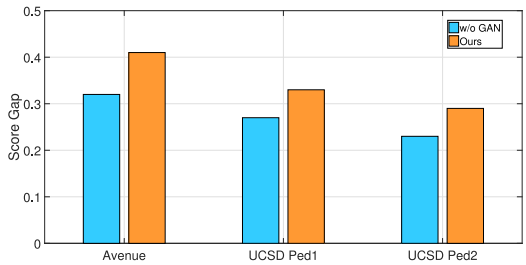


Fig. 4. The gaps of anomaly scores between normal and abnormal frames in the Avenue, UCSD Ped1 and UCSD Ped2 datasets with and without GAN. A discriminative and reliable detector for abnormal event detection often requires a larger score gap.

TABLE VI
THE AUC AND EER RESULTS OF DIFFERENT FORMAT NOISE ON THE AVENUE DATASET

Noise Type	AUC↑/EER↓ (%)
w/o noise	86.8/19.3
salt-and-pepper noise	87.7/20.9
Poisson noise	84.5/22.1
Gaussian noise	**88.9/18.2**

TABLE VII
COMPARISON OF THE SPEED OF ABNORMAL EVENT DETECTION ON THE AVENUE DATASET

Method	Frames Per Second (fps)	AUC↑ (%)
Conv-AE [12]	946.1	70.2
ConvLSTM-AE [14]	602.7	80.3
Ours (10 layers)	467.5	**88.9**
Ours (6 layers)	729.4	84.8

3) Speed Comparison: As shown in Table VII, we compare the detection speed of our method with that of the 2D convolutional auto-encoder (Conv-AE) [12] and convolutional LSTM auto-encoder (ConvLSTM-AE) [14] on the Avenue dataset. Because auto-encoders in both [12] and [14] has 6 layers, we also show the detection speed of our network with 6 layers for fair comparisons, where the last two layers of the encoder and the corresponding decoder layers are removed. The detection speed is measured using a single NVIDIA RTX2080Ti GPU and an Intel i7-7800X CPU.

Among all 6-layer auto-encoders in Table VII, the speed of our method (729.4fps) is slightly slower than that (946.1fps) of [12], and faster than that (467.5fps) of [14]. This shows that 3D operations take more time than 2D convolutions, but are more efficient than ConvLstm operations when processing temporal information. Our network with 6 layers significantly improves

TABLE VIII
THE AUC RESULTS OF DIFFERENT COMPONENTS OF THE ADVERSARIAL 3D CONVOLUTIONAL AUTO-ENCODER

Method	AUC↑(%)			
	Avenue	UCSD (Ped2)	ShanghaiTech	Subway (Exit)
Ionescu *et al.* [13]	90.4	**97.8**	**84.9**	–
Ionescu *et al.* [57]	88.9	–	–	95.1
Ours (w/o object detection)	88.9	96.0	74.6	94.8
Ours (object detection)	**91.2**	96.9	84.0	**98.8**

the performance, with gains of 4.5% and 14..6% compared with the methods in [12] and [14], respectively, which verifies the effectiveness of our method. Besides, due to the deeper structure, our 10-layer auto-encoder takes more time, but meanwhile achieves the best detection results.

4) Pre-Processing Using Object Detection: As our method is based on the pixel-level reconstruction setting, it can achieve high true-positives while usually suffers from high false-positive errors [31]. Several recent methods adopt divided patches [57], [58] or detected object regions [13], [24] to overcome the problem of a high false-positive error, with the price of inference speed. To further investigate the effectiveness of our method, we use the same pre-processing strategy of object detection proposed in [13], and the comparison results are shown in Table VIII. We directly use the object bounding boxes at the t-frame to crop objects at frames from $(t-3)$-th to $(t+3)$-th, and these regions are stacked as the input. We use a shallower network to reconstruct them, where layers of "Conv4," "Conv5," "De-Conv1" and "De-Conv2" are removed. The final frame-level anomaly scores are determined by the maximum reconstruction error of all the object regions in the frame. From Table VIII, we observe that the pre-preprocessing of object detection helps our method achieve significant improvements on the ShanghaiTech dataset, increasing the AUC from 74.6% to 84.0%. Our method outperforms the work of [13] on the Avenue dataset, and achieves comparable results on the UCSD Ped2 and ShanghaiTech datasets. Our method also performs best compared with the state-of-the-art method [57] on the Subway Exit dataset, with the gains of 3.6%.

5) Exploiting Optical-Flow Images: Optical-flow images can bring a great improvement in the crowded scenes for abnormal event detection but their calculation is time-consuming. We exploit optical-flow images on the UCSD dataset to verify the effectiveness of optical-flow images in our method. we follow [11] to extract optical-flow images, and reconstruct the RGB and optical-flow images through two independent auto-encoders. As shown in Table IX, when both RGB images and optical-flow images are used, our method outperforms the state-of-the-art method [11] on the UCSD Ped2 dataset, but the AUC value of our method is 1.7% worse than that of [11] on the UCSD Ped1 dataset. The possible reasoning is that we do not design a special two-branch structure for our 3D convolutional auto-encoder, but simply combine the reconstruction of RGB images and optical-flow images to detect abnormal events. From Table IX, we can also observe that when only using RGB images, we achieve the AUC score of 90.2% that is 6.1% higher than the work of [11]. This can verify the effectiveness of our method.

6) Visualizing Feature Maps: We visualize some feature maps of the ConvLSTM auto-encoder [14] and our 3D convolutional auto-encoder on the UCSD dataset in Fig. 5 to verify that our method is more conducive to capturing subtle spatio-temporal changes. For a fair comparison, we set the kernel size of the ConvLSTM auto-encoder in [14] to 3×3.

Fig. 5(a) shows a gray image of input videos with some subtle behaviors indicated by blue bounding boxes. The behaviors are easily overlooked due to their similarities to the background and their small regions. Fig. 5(b) shows two feature maps of Conv1 and Conv2 layers of the ConvLSTM auto-encoder. Fig. 5(c) shows two corresponding feature maps of our 3D convolutional auto-encoder. We resize the feature maps to the same size as the original gray image, and then select the feature maps with the highest mean response values of the foreground objects in the blue-box regions in all channels. We observe that [14] enhances the patterns of the salient foreground objects outside the blue-box regions, but falls short of response to the subtle behaviors in the blue-box regions, which is particularly obvious in the Conv2 feature maps. In contrast, our method models subtle behaviors in the blue-box regions better, which can verify that our method is more conducive to modeling fine-grained spatio-temporal patterns by performing information correlations on low-level pixel spaces.

7) Event Count: We set the appropriate threshold of the anomaly scores to detect abnormal events and evaluate our method based on the event counts. Following the settings in [12], we assume that the local minima within 50 frames belong to the same abnormal event to reduce the noise in the anomaly scores. It is reasonable as an abnormal event should be at least 2-3 seconds long to be meaningful. Table X shows the number of detected abnormal events and a false alarm on the three datasets. For both the Ped1 and Ped2 of the UCSD dataset, we achieve better results than [12], [30]. For the Avenue dataset, our method can detect the abnormal event more precisely, despite it generates more false alarms. For the subway dataset, we achieve better performance than other methods. The results demonstrate that our method can determine the temporal region of the abnormal events more accurately, which makes it more practical in real scenes.

8) Qualitative Results: Fig. 6, Fig. 7 and Fig. 8 list several examples of the detected abnormal events using our method on the Avenue, Subway Entrance and Subway Exit datasets. The detected abnormal events are "throwing papers" on the Avenue dataset, "running" and "wrong direction" on the Subway Entrance video, and "clean the wall" on the Subway Exit video. Fig. 6, From Fig. 7 and Fig. 8, we can clearly see that there is a large score gap between normal and abnormal events, which validates the effectiveness of our method.

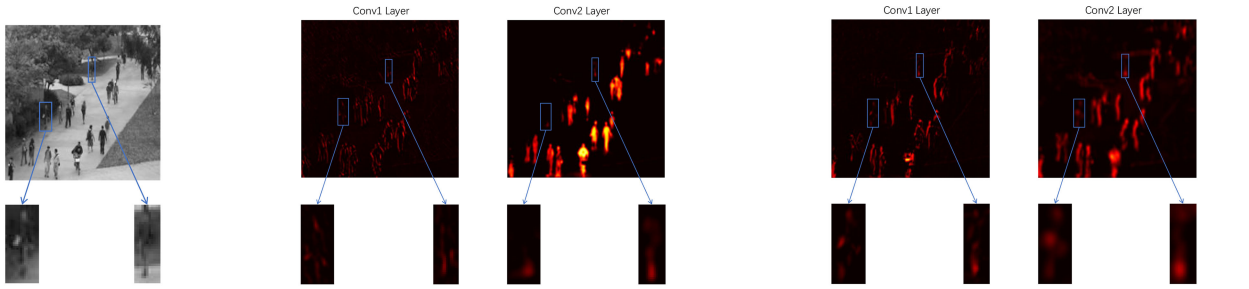
F. Time Complexity Analysis

We analyze the time and space complexity of a 2D convolutional layer, a 3D convolutional layer and a convolutional LSTM layer.

1) Time Complexity: A standard 2D convolutional layer performs one 2D convolution operation with a kernel W, one 2D addition operation with a bias b, and one tanh activation operation. Its time complexity is $\mathcal{O}((K_x \cdot K_y \cdot C_i + n) \cdot D_x \cdot D_y \cdot C_o)$,

TABLE IX
ABNORMAL EVENT DETECTION RESULTS IN TERMS OF FRAME-LEVEL AUC AND EER ON THE UCSD DATASET

Method	AUC↑/EER↓ (%)	
	UCSD (Ped1)	UCSD (Ped2)
Adam *et. al* (RGB+optical-flow) [43]	77.1/38.0	−/42.0
Kim and Grauman (RGB+optical-flow) [54]	59.0/−	69.3/−
Mehran *et. al* (RGB+optical-flow) [53]	67.5/31.0	55.6/42.0
Ravanbakhsh *et al.* (RGB only) [11]	84.1/−	−/−
Ravanbakhsh *et al.* (RGB+optical-flow) [11]	**97.4/8.0**	93.5/14.0
Ours (RGB only)	90.2/11.6	91.0/10.9
Ours (RGB+optical-flow)	95.7/8.8	**96.0/9.2**



(a) A frame of input videos (b) Feature maps of the ConvLSTM auto-encoder (c) Feature maps of our 3D convolutional auto-encoder

Fig. 5. Feature maps of the ConvLSTM auto-encoder and our 3D convolutional auto-encoders on the UCSD dataset.

TABLE X
THE NUMBER OF DETECTED ABNORMAL EVENTS AND FALSE ALARM ON THE THREE PUBLIC DATASETS. GT STANDS FOR GROUDTRUTH VALUES OF EVENT COUNT

Method	True Positives↑/False Alarm↓				
	UCSD Ped1 GT:40	UCSD Ped1 GT:12	Subway Entrance GT:66	Subway Exit GT:19	Avenue GT:47
Lu *et al.* [45]	N/A	N/A	19/2	N/A	N/A
Kim *et. al* [54]	N/A	N/A	56/3	19/2	N/A
Dutta *et. al* [59]	N/A	N/A	60/5	19/2	N/A
Zhao *et al.* [60]	N/A	N/A	60/5	19/2	N/A
Medel and Savakis [30]	40/7	12/1	62/14	19/37	40/2
Hasan *et al.* [12]	38/6	12/1	61/15	17/5	45/4
Chong and Tay [14]	N/A	N/A	61/9	18/10	44/12
Ours	**40**/7	**12/1**	**62**/8	**19**/9	**45**/11

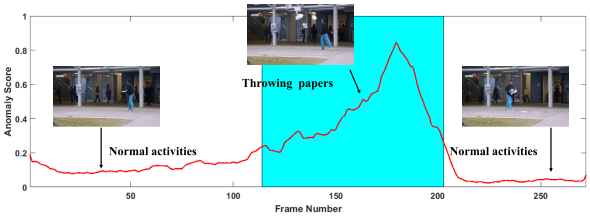


Fig. 6. Qualitative results on the Avenue dataset.

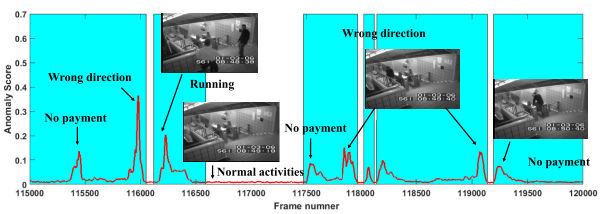


Fig. 7. Qualitative results on the Subway Entrance dataset.

where the kernel size is $K_x \times K_y$ and the feature map size is $D_x \times D_y$. We assume that activation functions take n FLOPs.

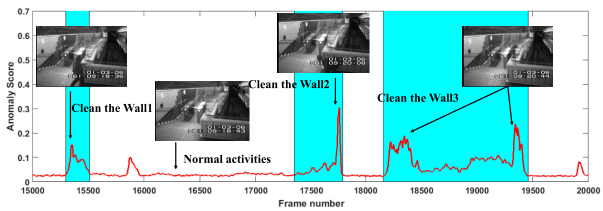


Fig. 8. Qualitative results on the Subway Exit dataset.

The number of input channels is C_i, and the number of output channels is C_o. A standard 3D convolutional layer takes one 3D convolution operation, one 3D addition operation with a bias as well as a tanh activation operation. The time complexity is $\mathcal{O}((K_t \cdot K_x \cdot K_y \cdot C_i + n) \cdot D_t \cdot D_x \cdot D_y \cdot C_o)$, where K_t refers to the temporal kernel size and D_t represents the temporal size of the feature map. The time complexity of a convolutional LSTM layer proposed by Chong and Tay [14] is given by $\mathcal{O}((4 \cdot K_x \cdot K_y \cdot (C_i + C_o) + 5 \cdot n + 12) \cdot D_x \cdot D_y \cdot C_o)$, where a convolutional LSTM layer performs four convolutions on the input layer, four convolutions on the hidden layer,

five 2D activations of sigmoid and tanh, three 2D Hadamard Products, and nine 2D additions.

We observe that the temporal dimensions of convolutional kernels and feature maps increase the time complexity of 3D convolutions compared with 2D convolutions. However, experiment results show that the processing of temporal information in videos can significantly improve detection performance. We need to repeat the convolutional LSTM operation T times to process a video with T frames ($T \geq D_t$ in general), which means that the time complexity of the convolutional LSTM should be $\mathcal{O}(T \cdot (4 \cdot K_x \cdot K_y \cdot (C_i + C_o) + 5 \cdot n + 12) \cdot D_x \cdot D_y \cdot C_o)$. When other parameters are the same and a small value of K_t (3 in this paper) is selected, the time complexity of 3D convolutional layers is less than convolutional LSTM layers.

2) Space Complexity: The parameter size of a 2D convolutional layer is calculated as $(K_x \cdot K_y \cdot C_i + D_x \cdot D_y) \cdot C_o$. A convolutional LSTM layer has the parameter size of $4 \cdot (K_x \cdot K_y \cdot (C_i + C_o) + D_x \cdot D_y) \cdot C_o$. A 3D convolutional layer contains the parameter size of $(K_t \cdot K_x \cdot K_y \cdot C_i + D_t \cdot D_x \cdot D_y) \cdot C_o$. In contrast to convolutional LSTM layers that share parameters across time, 3D convolutions have more parameters, which is helpful to simultaneously learn appearance patterns and motion patterns to capture robust normal spatio-temporal patterns.

V. Conclusion

In this paper, we have presented an effective abnormal event detection method of simultaneously learning normal appearance patterns and motion patterns, which can capture fine-grained spatio-temporal patterns. We built an adversarial 3D convolutional auto-encoder that can capture low-level correlations between appearance and motion patterns. The 3D convolutional encoder can capture appearance and motion information as well as their correlations into encoded features, and the 3D deconvolutional decoder can reconstruct original videos from the encoded features directly. The 3D convolutional auto-encoder trained with the denoising reconstruction error and adversarial learning strategy can implicitly learn more accurate data distributions of normal data that are considered normal patterns, which can better distinguish normal and abnormal events without any supervised information. Theoretical analysis and experiments on four public datasets have demonstrated the effectiveness and superiority of our method on abnormal event detection in videos.

Appendix A
Proof of Theorem 1

Theorem 1: The optimal output of the 3D convolutional auto-encoder calculated by the loss function $\mathcal{L}_{\text{rec}}$ in Eq.(7) is

$$R_\theta^*(\tilde{S}) = \frac{\mathbb{E}_\eta\left[p(\tilde{S}-\eta)(\tilde{S}-\eta)\right]}{\mathbb{E}_\eta\left[p(\tilde{S}-\eta)\right]}, \quad \text{s.t. For } \forall x,\ p(x) \neq 0, \tag{15}$$

where $\tilde{S}$ denotes the corrupted input sample with Gaussian noise and η represents the Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$.

Proof: The reconstruction loss function $\mathcal{L}_{\text{rec}}$ is defined as

$$\mathcal{L}_{\text{rec}} = \mathbb{E}\left[\left\|R_\theta(S+\eta) - S\right\|^2\right], \tag{16}$$

where $R_\theta(\cdot)$ is the mapping of the 3D convolutional auto-encoder. We collect samples S that only contain normal events for training. The loss function can be rewritten as

$$\mathcal{L}_{\text{rec}} = \int_S \mathbb{E}_{\eta\sim\mathcal{N}(0,\sigma^2 I)} \times \left[p(S) \times \left\|R_\theta(S+\eta) - S\right\|^2\right] dS, \tag{17}$$

where p is the density function of the training data. Then we use the auxiliary variable $\tilde{S} = S + \eta$ to replace S and get

$$\mathcal{L}_{\text{rec}} = \int_{\tilde{S}} \mathbb{E}_{\eta\sim\mathcal{N}(0,\sigma^2 I)} \times \left[p(\tilde{S}-\eta) \times \left\|R_\theta(\tilde{S}) - \tilde{S} + \eta\right\|^2\right] d\tilde{S}. \tag{18}$$

Obviously, the minimum value of the loss function $\mathcal{L}_{\text{rec}}$ is 0, that is

$$\mathbb{E}_{\eta\sim\mathcal{N}(0,\sigma^2 I)}\left[p(\tilde{S}-\eta)\left(R_\theta^*(\tilde{S}) - \tilde{S} + \eta\right)\right] = 0, \tag{19}$$

where $R_\theta^*(\tilde{S})$ denotes the optimal solution. The Eq. (19) can be rewritten as

$$\mathbb{E}_{\eta\sim\mathcal{N}(0,\sigma^2 I)}\left[p(\tilde{S}-\eta)R_\theta^*(\tilde{S})\right] = \mathbb{E}_{\eta\sim\mathcal{N}(0,\sigma^2 I)}\left[p(\tilde{S}-\eta)(\tilde{S}-\eta)\right]. \tag{20}$$

The optimal output of the 3D convolutional auto-encoder is

$$R_\theta^*(\tilde{S}) = \frac{\mathbb{E}_\eta\left[p(\tilde{S}-\eta)(\tilde{S}-\eta)\right]}{\mathbb{E}_\eta\left[p(\tilde{S}-\eta)\right]}, \quad \text{s.t. For } \forall x.\ p(x) \neq 0. \tag{21}$$

Appendix B
Proof of Theorem 2

Theorem 2: When the noise level σ asymptotically approaches to 0 ($\sigma \to 0$), the loss function $\mathcal{L}_{\text{rec}}$ can be rewritten as

$$\mathcal{L}_{\text{rec}} = \mathbb{E}\left[\left\|R_\theta(S) - S\right\|^2 + \sigma^2 \left\|\frac{\partial\left(R_\theta(S)\right)}{\partial S}\right\|^2\right] + o(\sigma^2). \tag{22}$$

The optimal output calculated by Eq. (22) during testing is

$$R_\theta^*(S) = S + \sigma^2 \frac{\partial \log p(S)}{\partial S} + o(\sigma^2). \tag{23}$$

Proof: The reconstruction loss function $\mathcal{L}_{\text{rec}}$ is defined as

$$\mathcal{L}_{\text{rec}} = \mathbb{E}\left[\left\|R_\theta(S+\eta) - S\right\|^2\right], \tag{24}$$

The Taylor expansion around S is

$$R_\theta(S+\eta) = R_\theta(S) + \frac{\partial R_\theta(S)}{\partial S}\eta + o(\sigma^2). \tag{25}$$

Substituting it into $\mathcal{L}_{\text{rec}}$, we get

$$\mathcal{L}_{\text{rec}} = \mathbb{E}\left[\left\|S - \left(R_\theta(S) + \frac{\partial R_\theta(S)}{\partial S}\eta + o(\sigma^2)\right)\right\|^2\right]$$

$$= \left(\mathbb{E}\left[\left\| R_\theta(S) - S \right\|^2 \right] - 2\mathbb{E}^T[\eta]\mathbb{E}\left[\frac{\partial R_\theta(S)^T}{\partial S} \times \left(R_\theta(S) - S \right) \right] \right) + Tr\left(\mathbb{E}\left[\eta\eta^T \right] \times \mathbb{E}\left[\frac{\partial R_\theta(S)^T}{\partial S} \frac{\partial R_\theta(S)}{\partial S} \right] \right) + o(\sigma)^2$$
$$= \left(\mathbb{E}\left[\left\| R_\theta(S) - S \right\|^2 \right] + \sigma^2 \mathbb{E}\left[\left\| \frac{\partial R_\theta(S)}{\partial S} \right\|^2 \right] \right) + o(\sigma)^2. \tag{26}$$

The premise of the Eq. (26) are: (1) the noise η is independent from S; (2) $\mathbb{E}[\eta^2\eta] = \sigma^2 I$; (3) $\mathbb{E}[\eta] = 0$.

The input $S \in \mathbb{R}^{T\times H\times W}$ is a video matrix containing normal events and the output $R_\theta(S) \in \mathbb{R}^{T\times H\times W}$ is its reconstruction, where T is the number of frames, H and W are the height and width of the frame, respectively. For simple representation, we assume that the matrix $S \in \mathbb{R}^{T\times H\times W}$ is converted into a vector $S \in \mathbb{R}^d$, where $d = T \times H \times W$ and the F-norm of the matrix transfers into the 2-norm of the vector. In the following proof process, we derive the optimal solution based on the work of Alain and Bengio [17] by operating on each element of S.

We rewrite the loss function $\mathcal{L}_{\text{rec}}$ into the integral form as

$$\mathcal{L}_{\text{rec}} = \int_{\mathbb{R}^d} p(S) \left[\left\| R_\theta(S) - S \right\|_2^2 + \sigma^2 \left\| \frac{\partial R_\theta(S)}{\partial S} \right\|^2 \right]. \tag{27}$$

The extremum of Eq. (27) can be solved by constructing Euler-Lagrange equation. Before constructing, we expand Eq. (27) with each element as

$$\mathcal{L}_{\text{rec}} = \int_{\mathbb{R}^d} p(S) \left[\sum_{i=1}^{d} \left(R_\theta^i(S) - S^i \right)^2 + \sigma^2 \sum_{i=1}^{d}\sum_{j=1}^{d} \left(\frac{\partial R_\theta^i(S)}{\partial S^j} \right)^2 \right] dS$$
$$= \sum_{i=1}^{d} \int_{\mathbb{R}^d} p(S) \left[\left(R_\theta^i(S) - S^i \right)^2 + \sigma^2 \sum_{j=1}^{d} \left(\frac{\partial R_\theta^i(S)}{\partial S^j} \right)^2 \right] dS, \tag{28}$$

where $R_\theta^i(S)$ represents the i-th dimension of $R_\theta(S) \in \mathbb{R}^d$. The Eq. (28) indicates that each dimension $R_\theta^i(S)$ can be optimized separately.

Then we construct the Euler-Lagrange equation based on the work of Dacorogna [61] as

$$f\left(S, R_\theta(S), R_\theta(S)' \right) = p(S) \left[\left\| R_\theta(S) - S \right\|_2^2 + \sigma^2 \left\| \frac{\partial R_\theta(S)}{\partial S} \right\|^2 \right], \tag{29}$$

where $R_\theta(S)'$ is $\frac{\partial R_\theta(S)}{\partial S}$. The Euler-Lagrange equation satisfied at the optimal 3D convolutional auto-encoder $R_\theta(\cdot)$ is given by

$$\frac{\partial f}{\partial R_\theta} = \sum_{j=1}^{d} \frac{\partial}{\partial S^j} \left(\frac{\partial f}{\partial R_\theta^j(S)'} \right)$$
$$= \sum_{j=1}^{d} \frac{\partial}{\partial S^j} \left(2\sigma^2 p(S) \left[\frac{\partial R_\theta^1}{\partial S^j} \; \frac{\partial R_\theta^2}{\partial S^j} \cdots \frac{\partial R_\theta^d}{\partial S^j} \right]^T \right)$$
$$= 2\sigma^2 \sum_{j=1}^{d} \left(\frac{\partial p(S)}{\partial S^j} \left[\frac{\partial R_\theta^1}{\partial S^j} \; \frac{\partial R_\theta^2}{\partial S^j} \cdots \frac{\partial R_\theta^d}{\partial S^j} \right]^T + p(S) \left[\frac{\partial^2 R_\theta^1}{\partial (S^j)^2} \; \frac{\partial^2 R_\theta^2}{\partial (S^j)^2} \cdots \frac{\partial^2 R_\theta^d}{\partial (S^j)^d} \right]^T \right)$$
$$= 2\sigma^2 \sum_{j=1}^{d} \begin{bmatrix} \frac{\partial p(S)}{\partial S^j} \frac{\partial R_\theta^1}{\partial S^j} + p(S) \frac{\partial^2 R_\theta^1}{\partial (S^j)^2} \\ \vdots \\ \frac{\partial p(S)}{\partial S^j} \frac{\partial R_\theta^d}{\partial S^j} + p(S) \frac{\partial^2 R_\theta^d}{\partial (S^j)^2} \end{bmatrix}. \tag{30}$$

At the same time, from Euler-Lagrange Eq. (29), we get

$$\frac{\partial f}{\partial R_\theta} = 2\left(R_\theta(S) - S \right) p(S). \tag{31}$$

Substituting Eq. (31) into Eq. (30) is

$$\left(R_\theta(S) - S \right) p(S) = \sigma^2 \sum_{j=1}^{d} \begin{bmatrix} \frac{\partial p(S)}{\partial S^j} \frac{\partial R_\theta^1}{\partial S^j} + p(S) \frac{\partial^2 R_\theta^1}{\partial (S^j)^2} \\ \vdots \\ \frac{\partial p(S)}{\partial S^j} \frac{\partial R_\theta^d}{\partial S^j} + p(S) \frac{\partial^2 R_\theta^d}{\partial (S^j)^2} \end{bmatrix}. \tag{32}$$

Each dimension $R_\theta^i(S)$ can be optimized separately, that is

$$\left(R_\theta^i(S) - S^i \right) p(S) = \sigma^2 \sum_{j=1}^{d} \left(\frac{\partial p(S)}{\partial S^j} \frac{\partial R_\theta^i(S)}{\partial S^j} + p(S) \frac{\partial^2 R_\theta^i(S)}{\partial (S^j)^2} \right). \tag{33}$$

Due to $for\ \forall S,\ p(S) \neq 0$, we can divide the Eq. (33) by $p(S)$. According to $\frac{\partial p(S)}{\partial S^i} \big/ p(S) = \frac{\partial \log p(S)}{\partial S^i}$, we get

$$R_\theta^i(S) - S^i = \sigma^2 \sum_{j=1}^{d} \left(\frac{\partial \log p(S)}{\partial S^j} \frac{\partial R_\theta^i(S)}{\partial S^j} + \frac{\partial^2 R_\theta^i(S)}{\partial (S^j)^2} \right). \tag{34}$$

Obviously, the Eq. (34) is recursive about a $R_\theta^i(S)$. Since the noise level σ asymptotically approaches to 0 ($\sigma \to 0$), we can get rid of the items that contain the coefficient of high-order (*e.g.* $(\sigma^2)^2$ or higher). Thus, we get the recursive equation (some expansion steps here are omitted)

$$R_\theta^i(S) = S^i + \sigma^2 \frac{\log \partial p(S)}{\partial S^i} + o(\sigma^2). \tag{35}$$

The optimal output is given by

$$R_\theta^*(S) = S + \sigma^2 \frac{\log \partial p(S)}{\partial S} + o(\sigma^2),\ as\ \sigma \to 0. \tag{36}$$

■

REFERENCES

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 555–560, Mar. 2008.

[2] K. Xu, X. Jiang, and T. Sun, "Anomaly detection based on stacked sparse coding with intraframe classification strategy," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1062–1074, May 2018.

[3] D. Gong *et al.*, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct., 2019, pp. 1705–1714.

[4] W. Chu, H. Xue, C. Yao, and D. Cai, "Sparse coding guided spatiotemporal feature learning for abnormal event detection in large videos," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 246-255, Jan. 2019.

[5] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, 2018, pp. 6479–6488.

[6] C. Wang, H. Yao, and X. Sun, "Anomaly detection based on spatio-temporal sparse representation and visual attention analysis," *Multimedia Tools APP.*, vol. 76, no. 5, pp. 6263–6279, 2017.

[7] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Semi-supervised adapted HMMs for unusual event detection," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2005, pp. 611–618.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1106–1114.

[9] J. Hou, X. Wu, Y. Sun, and Y. Jia, "Content-attention representation by factorized action-scene network for action recognition," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1537–1547, Jun. 2018.

[10] D. Guo, W. Li, and X. Fang, "Fully convolutional network for multi-scale temporal action proposals," *IEEE Trans. Multimedia*, vol. 20, no. 12, pp. 3428–3438, Dec. 2018.

[11] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. S. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *Proc. Int. Conf. Image Process.*, Sep. 2017, pp. 1577–1581.

[12] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2016, pp. 733–742.

[13] R. T. Ionescu, F. S. Khan, M. Georgescu, and L. Shao, "Object-centric auto-encoders and dummy anomalies for abnormal event detection in video," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2019, pp. 7842–7851.

[14] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *Proc. Adv. Neural Net.*, Jun. 2017, pp. 189–196.

[15] Y. Li, *et al*, "Flow-grounded spatial-temporal video prediction from still images," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 609–625.

[16] I. J. Goodfellow *et al.* "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2014, pp. 2672–2680.

[17] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *Jour. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3563–3593, 2014.

[18] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2011, pp. 3449–3456.

[19] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[20] W. Liu, W. Luo, Z. Li, P. Zhao, and S. Gao, "Margin learning embedded prediction for video anomaly detection with a few anomalies," in *Proc. Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3023–3030.

[21] Y. Feng, Y. Yuan, and X. Lu, "Deep representation for abnormal event detection in crowded scenes," in *Proc. ACM Conf. Multimedia*, Oct. 2016, pp. 591–595.

[22] D. Hou, Y. Cong, G. Sun, J. Liu, and X. Xu, "Anomaly detection via adaptive greedy model," *Neurocomputing*, vol. 330, pp. 369–379, 2019.

[23] 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.

[24] S. Zhou, *et al*, "Spatial-temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Sig. Proc.: Image Comm.*, vol. 47, 2016, pp. 358–368.

[25] K. Xu, T. Sun, and X. Jiang, "Video anomaly detection and localization based on an adaptive intra-frame classification network," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 394–406, Feb. 2020.

[26] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, 2020, pp. 14360–14369.

[27] Y. Tang, *et al*, "Integrating prediction and reconstruction for anomaly detection," *Pattern Recognit. Lett.*, vol. 129, pp. 123–130, 2020.

[28] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 843–852.

[29] L. Zhu, Z. Xu, and Y. Yang, "Bidirectional multirate reconstruction for temporal modeling in videos," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, 2017, pp. 1339–1348.

[30] J. R. Medel and A. Savakis, "Anomaly detection in video using predictive convolutional long short-term memory networks," *arXiv:1612.00390*, 2016. [Online]. Available: http://arxiv.org/abs/1612.00390

[31] M. Sabokrou, *et al*, "AVID: Adversarial visual irregularity detection," in *Proc. Asian Conf. Comput. Vis.*, Dec. 2018, pp. 488–505.

[32] T.-N. Nguyen and J. Meunier, "Anomaly detection in video sequence with appearance-motion correspondence," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1273–1283.

[33] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jul. 2017, pp. 5967–5976.

[34] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Inf. Proc. Med. Imag.*, Jun. 2017, pp. 146–157.

[35] B. Fuglede and F. Topsoe, "Jensen-shannon divergence and hilbert space embedding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2004, p. 31.

[36] Z. Jiang, V. Rozgic, and S. Adali, "Learning spatiotemporal features for infrared action recognition with 3D convolutional neural networks," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 309–317.

[37] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 818–833.

[38] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2006, pp. 153–160.

[39] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2018, pp. 3379–3388.

[40] J. Bouvrie, "Notes on convolutional neural networks," Technical report, 2006, pp. 38–44.

[41] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2008, pp. 1096–1103.

[42] Y. Kozlov and T. Weinkauf, "Persistence1D: Extracting and filtering minima and maxima of 1D functions," Nov. 2015. [Online]. Available: http://people.mpi-inf.mpg.de/weinkauf/notes/persistence1d.html

[43] A. Adam, E. Rivlin, I. Shimshoni, and D. id Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 555–560, Mar. 2008.

[44] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2009, pp. 1975–1981.

[45] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2720–2727.

[46] W. Luo, W. Liu, and S. Gao, "A revisit of sparse coding based anomaly detection in stacked RNN framework," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 341–349.

[47] R. Tudor Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Unmasking the abnormal events in video," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jul. 2017, pp. 2895–2903.

[48] R. Morais, *et al*, "Learning regularity in skeleton trajectories for anomaly detection in videos," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2019, pp. 11 996–12 004.

[49] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection–a new baseline," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, 2018, pp. 6536–6545.

[50] S. Wang, Y. Zeng, Q. Liu, C. Zhu, E. Zhu, and J. Yin, "Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection," in *Proc. ACM Conf. Multimedia*, Oct. 2018, pp. 636–644.

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

[52] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. Adv. Neural Inf. Process. Syst. Workshops*, Dec. 2017, pp. 8024–8035.
[53] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2009, pp. 935–942.
[54] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2009, pp. 2921–2928.
[55] T. Wang and H. Snoussi, "Histograms of optical flow orientation for visual abnormal events detection," in *Proc. IEEE Int. Conf. Adv. Video Signal. Based Surveill.*, Sep. 2012, pp. 13–18.
[56] D. Xu, Y. Yan, E. Ricci, and N. Sebe, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Comput. Vis. Image Underst.*, vol. 156, pp. 117–127, 2016.
[57] 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.
[58] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, 2018, pp. 3379–3388.
[59] J. K. Dutta and B. Banerjee, "Online detection of abnormal events using incremental coding length," in *Proc. Conf. Artif. Intell.*, Jan. 2015, pp. 3755–3761.
[60] B. Zhao, F. Li, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proc. IEEE Int. Conf. Vis. Pattern Recognit.*, Jun. 2011, pp. 3313–3320.
[61] B. Dacorogna, *Introduction Calculus Variations*. World Scientific Publishing Company, 2014.

Che Sun (Student Member, IEEE) received the B.S. degree from the Beijing Institute of Technology, Beijing, China, in 2017. He is working toward the Ph.D. degree with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology. His research interests include computer vision, machine learning.

Yunde Jia (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Beijing Institute of Technology (BIT), in 1983, 1986, and 2000, respectively. He was a Visiting Scientist with the Robotics Institute, Carnegie Mellon University, from 1995 to 1997. He is currently a Professor with the School of Computer Science, BIT, and the Team Head of BIT innovation on vision and media computing. He serves as the Director of Beijing Laboratory of Intelligent Information Technology. His interests include computer vision, computational perception and cognition, intelligent robots, and HCI.

Hao Song received the B.S. degree from North China Electric Power University, Baoding, China, in 2012 and the Ph.D. degree from the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, in 2018. His research interests include computer vision, machine learning and video retrieval.

Yuwei Wu (Member, IEEE) received the Ph.D. degree in computer science from the Beijing Institute of Technology (BIT), Beijing, China, in 2014. He is currently an Assistant Professor with School of Computer Science, BIT. From 2014 to 2016, he was a Postdoctoral Research fellow with Rapid-Rich Object Search Laboratory, School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore. He received outstanding Ph.D. Thesis Award from BIT, and Distinguished Dissertation Award Nominee from China Association for Artificial Intelligence .