Appearance-Motion Memory Consistency Network for Video Anomaly Detection

Ruichu Cai[1,*], Hao Zhang[1,*], Wen Liu[2,3,*], Shenghua Gao[2,†], Zhifeng Hao[1,†]
[1]Guangdong University of Technology, Guangzhou 510006, China.
[2]ShanghaiTech University, Shanghai Engineering Research Center of Intelligent Vision and Imaging, Shanghai 201210, China.
[3]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, China.
{cairuichu,haodotzhang}@gmail.com, {liuwen,gaoshh}@shanghaitech.edu.cn, zfhao@gdut.edu.cn

Abstract

Abnormal event detection in the surveillance video is an essential but the challenging task and many methods have been proposed to deal with this problem. The previous methods either only considers the appearance information or directly integrate the results of appearance and motion information without considering their endogenous consistency semantic explicitly. Inspired by the rule that humans identify the abnormal frames from multi-modality signals, we propose an Appearance-Motion Memory Consistency Network (AMMC-Net). Our method first makes full use of the prior knowledge of appearance and motion signals to capture the correspondence between them in the high-level feature space explicitly. Then, it combines the multi-view features to obtain a more essential and robust feature representation of regular events, which can significantly increase the gap between an abnormal and a regular event. In the anomaly detection phase, we further introduce a commit error in the latent space joint with the prediction error in pixel space to enhance the detection accuracy. Solid experimental results on various standard datasets validate the effectiveness of our approach.

Introduction

Video anomaly detection (VAD) is a critical task in surveillance video. It has been studied for many years but remains unsolved due to the difficulties and challenges of collecting abnormal data (Kiran, Thomas, and Parakkal 2018). Compared to the regular events, the anomalies happen at a low frequency, and the types of them are diverse and even unbounded. Thereby, it seems infeasible to collect balanced normal and abnormal data and tackle this problem using traditional supervised binary-classification methods. Considering that the regular events are abundant in video surveillance, a prevalent setting (Luo, Liu, and Gao 2017a; Kiran, Thomas, and Parakkal 2018; Liu et al. 2018; Ionescu et al. 2019) is only the normal data provided.

Under this setting, modeling the appearance and motion information of regular events is the first principle. In addition to representing the two types of data independently, it is also crucial to model the corresponding between them. The consistency law existing in nature is an important concept and is widely used in computer vision (Wang, Jabri, and Efros 2019). Unlike the above work in the use of time correspondence, the consistency in VAD proposed in this paper considers modeling the correspondence between appearance and motion signals in regular events explicitly. For example, in a supermarket mall setting, the regular events are that people are pushing the shopping cart forward or staying together with the shopping cart. Some anomalies could be detected by the appearance (breaking out of the fire) or the motion (people fighting with each other) separately. In contrast, some anomalies need to be detected by considering the correlation between appearance and motion. For instance, the anomalies happen when people are standing still, and the shopping cart moves forward out of human control. From the perspective of the appearance alone, people and the shopping cart are both regular objects with any unusual appearance changed. People standing still and the shopping cart moving forward are both normal cases from the motion alone. Without Considering the correlation between appearance and movement, the anomaly detector may fail on these anomalies inevitably. Only by modeling the consistency between the appearance (people, the shopping cart)and the motion (people pushing the shopping cart forward)could we detect these anomalies and make the anomaly detector more robust.

However, the consistent correlation between appearance and motion in VAD was ignored by previous methods, including reconstruction (Hasan et al. 2016), prediction (Liu et al. 2018), and motion fusion (Xu et al. 2017; Yan et al. 2018; Vu et al. 2019). The former two methods ignore the motion information, and the latter directly combines the information of the two modalities in the testing phase. It does not jointly model the two types of information in the same space during the training phase, which does not capture the two modalities' consistency.

To explicitly model the consistency between appearance and motion information, we propose an appearance-motion memory-consistency framework for video anomaly detection. 1): We first learn the prior information of appearance and motion signals in regular events and stores them in two memory pools called AppMemPool and MotMemPool. Since there are many background pixels irrelevant to anomaly detection in the pixel space, and the original fea-

[*] Equal contribution.

[†] Corresponding author.

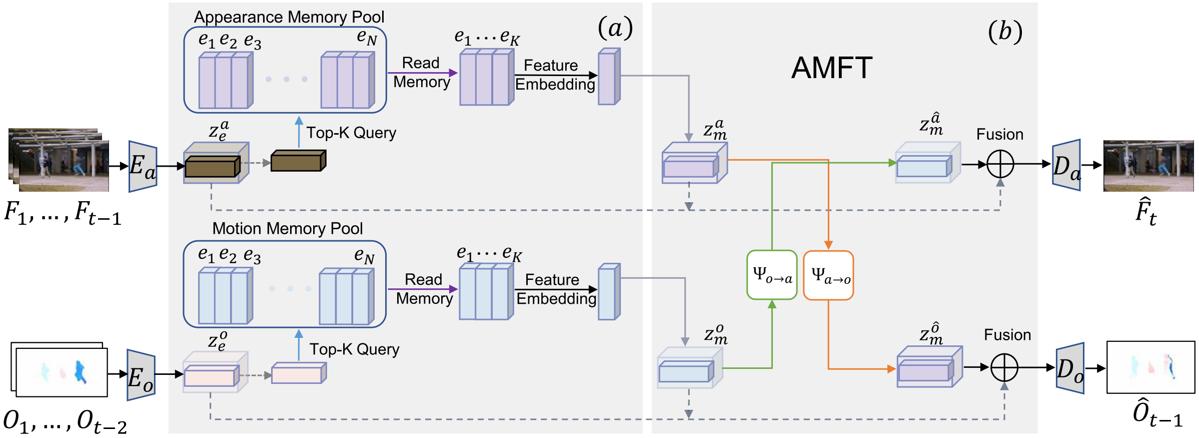

Figure 1: The overview of our proposed Appearance Motion Memory Consistency Network (AMMC-Net). Our model takes a sequence of images and optical flows as inputs. (a) Our memory pool defines a latent embedding space $\mathbf{M} \in \mathbb{R}^{D \times N}$ containing N feature embeddings with dimension D. The memory module takes a query feature z_e^a and z_e^o as the inputs and outputs the pool's memory items. (b) AMFT first transfers the AppMemPool-guided feature to the MotMemPool-guided feature and vice versa by using a network. It outputs the transferred consistent features $z_m^{\hat{a}}$ and $z_m^{\hat{o}}$. Then, the original feature z_e^a and z_e^o, memory-enhanced prototype feature z_m^a and z_m^o, and consistent feature $z_m^{\hat{a}}$ and $z_m^{\hat{o}}$ are combined to produce the final feature representation of the appearance and motion.

ture contains the specific information present in the sample, we choose to model the two features' prior information in the feature space. Considering the insufficient representation capacity of a single memory element, we propose to use the multiple memory units to represent the prototype feature of a query vector. 2): Then, we model the consistent correlation between the appearance and the motion by learning two mapping functions from the AppMemPool-guided feature to the MotMemPool-guided feature and vice versa, called Appearance-Motion Feature Transfer Network (AMFT). 3): Since the memory items only contain the prior information from the training data, and the unique information of each input might be lost, to compensate for the lost information by memory prior, we finally integrate the initial feature from the encoder, the prototype feature generated by the memory module, and the transformed feature from AMFT to form a robust and expressive feature of the regular events. 4): In the testing phase, we combine the prediction error of appearance/motion and the commit error (Den Oord, Vinyals, and Kavukcuoglu 2017) between the original feature and memory items to calculate the abnormal scores and determine whether a frame is anomalous or not.

We summarize our contributions as follows: i) We propose an appearance-motion memory consistent network(AMMC-Net) to model the consistency between regular videos' appearance and motion. ii) We introduce the memory-mechanism to model the prototype information in both appearances (rgb) and motion (optical flow) signals, respectively. iii) We propose an appearance-motion memory-guided feature transfer module to realize the cooperation and fusion of two modalities' information. iv) Extensive experiments demonstrate our methods' effectiveness, and all codes[1] have been released for further research convenience to the community.

Related Work

Recently, a large number of methods have been proposed to solve video abnormal event detection. In (Hasan et al. 2016; Sabokrou, Fathy, and Hoseini 2016), reconstruction-based models are proposed based on the assumption that models trained on regular events cannot reconstruct abnormal events that they have not seen. Conv-AE(Hasan et al. 2016) uses a deep auto-encoder to reconstruct an input sequence of frames from a training video set. Conv3D-AE(Sabokrou, Fathy, and Hoseini 2016) uses a 3D convolutional neural network to encode a video clip's appearance and motion information. A deconvolutional neural network is used to reconstruct the input video clip. A series of prediction-based models (Luo, Liu, and Gao 2017a; Shi et al. 2015; Zhao et al. 2017) was proposed to alleviate identity mapping in reconstructed models. Those methods view video frames as temporal patterns or time series, and the goal is to learn a generative model that can predict the future structure using the past frames. In (Luo, Liu, and Gao 2017a; Shi et al. 2015), a convolutional representation of the input video is input to the convolutional LSTM.. Then a deconvolution layer reconstructs output to the original resolution from the learned feature. (Zhao et al. 2017) proposes a Spatio-Temporal AutoEncoder (STAE), which utilizes deep neural networks to extract features from both spatial and temporal dimensions by performing 3-dimensional convolutions and reconstruct current clips and generating future frames. Besides, some models based on a two-stream network (Xu et al. 2017; Yan et al. 2018) were proposed to solve anomaly detec-

[1] https://github.com/NjuHaoZhang/AMMCNet_AAAI2021

tion. These methods were used initially for action recognition (Simonyan and Zisserman 2014) because it allows explicit modeling of the appearance and motion information, respectively. In another work, (Vu et al. 2019), a framework using multilevel representations of both intensity and motion data was proposed by Hung to encode regular frames. This detector can localize anomaly regions with high accuracy and low false detections by finding unusual objects at high-level representations besides low-level data and combining these detection results. More works can be founded in (Kiran, Thomas, and Parakkal 2018).

The methods most relevant to our work are the following three papers (Gong et al. 2019; Nguyen and Meunier 2019; Xu et al. 2017). In (Gong et al. 2019), a memory-augmented autoencoder called MemAE was proposed to improve the network's performance. Given an input, MemAE firstly obtains the encoding from the encoder and then uses it as a query to retrieve the most relevant memory items for reconstruction. It attempts to reconstruct the appearance utilizing auto-encoder architecture. In (Nguyen and Meunier 2019), an rgb-to-optical flow translation network was proposed to exploits the correspondence between appearances and their motions. It uses a U-Net structure to predict the corresponding motion given an input RGB frame. (Xu et al. 2017) proposes a double fusion framework combining the traditional early fusion and late fusion strategies. It first uses stacked denoising autoencoders to separately learn both appearance and motion features and a joint representation (early fusion). Then, multiple one-class SVM models predict each input's anomaly scores based on the learned feature. Finally, it uses a late fusion strategy to combine the computed scores and detect abnormal events. In contrast to the above methods, our proposed model can explicitly force two modalities to represent their features in a shared space during the training phase, thereby helping anomaly detection.

Algorithm 1 The whole pipeline of our AMMC-Net.

```
Require: $\{F_1, ..., F_{t-1}\}$, $\{O_1, ..., O_{t-2}\}$, $M^a$ and $M^o$
    • $\{F_1, ..., F_{t-1}\}$: the inputs sequence of images;
    • $\{O_1, ..., O_{t-2}\}$: the inputs sequence of optical flows;
    • $\mathbf{M^a} \in \mathbb{R}^{D \times N}$: the memory pool of appearance;
    • $\mathbf{M^o} \in \mathbb{R}^{D \times N}$: the memory pool of motion;
Ensure: $\hat{F}_t$: the predicted image and $\hat{O}_{t-1}$: the optical flow.
 1: $z_e^a = E_a(\{F_1, ..., F_{t-1}\})$ # encoding the images;
 2: $z_e^o = E_o(\{O_1, ..., O_{t-2}\})$ # encoding the optical flow;
 3: $z_e^a = \mathbb{R}^{H \times W \times D} \rightarrow \mathbb{R}^{HW \times D}$ # Flatten the feature map;
 4: $dist = \|z_e^a - M^a\|_2^2 \in \mathbb{R}^{HW \times N}$ # the distance of each spatial query feature to each memory item;
 5: $ids$ = topK (dist, dim=1) $\in \mathbb{R}^{HW \times K}$ # query the $K$ most closest memory indexes;
 6: $z_m^a = M^a[ids] \in \mathbb{R}^{HW \times K \times D}$ # taking the $K$ most closet memory items;
 7: $z_m^a = f(z_m^a) \in \mathbb{R}^{HW \times D}$ # appearance memory embeddings by a 1 x 1 convolution layer;
 8: Repeat 3-7 and get the motion memory embeddings $z_m^o$;
 9: $z_m^{\hat{a}} = \Phi_{o \rightarrow a}(z_m^o) \in \mathbb{R}^{HW \times D}$ # motion to appearance;
10: $z_m^{\hat{o}} = \Phi_{a \rightarrow o}(z_m^a) \in \mathbb{R}^{HW \times D}$ # appearance to motion;
11: $z_c^a = \mathcal{F}(z_m^{\hat{a}}, z_m^a, z_e^a)$ # fusing the transferred feature, memory feature, and encoding feature of appearance;
12: $z_c^o = \mathcal{F}(z_m^{\hat{o}}, z_m^o, z_e^o)$ # fusing the transferred feature, memory feature, and encoding feature of motion;
13: $\hat{F}_t = D_a(z_c^a)$ # image by appearance decoder;
14: $\hat{O}_{t-1} = D_o(z_c^o)$ # optical flow by motion decoder;
15: return $\hat{F}_t$ and $\hat{O}_{t-1}$.
```

Method

As is shown in Figure 1, our proposed AMMC-Net can be divided into three parts: Encoder, Decoder, and an appearance-motion memory-augmented feature transfer module (AMMT). We first input an image clip and its optical flow clip into the encoder to obtain the appearance and motion's initial feature representations. Then we feed the initial feature map into the memory module (AppMemPool and MotMemPool) to extract the prototype item of the two modalities. Next, we use two neural networks to transfer information between two input prototype features to obtain two consistent features. After that, we perform feature fusion between the initial features, memory-guided prototype features, and consistent features. Finally, feeding the fused feature into the decoder to predict the future appearance (image) and motion (optical flow), as shown in Algorithm 1.

Encoder and Decoder

The encoder is utilized to extract feature representations from input video frames. The decoder is trained to reconstruct the samples by taking the aggregated feature obtained from the previous step. We adopt the res-block used in Encoder and UNet-like skip connection structure (Ronneberger, Fischer, and Brox 2015) as the backbone network to construct the whole module. First, to enhance the range of network output and improve the representation ability, the original ReLU was modified as Tanh in this paper. Secondly, the 4-scale of the original architecture is reduced to 3-scale to control the model's complexity and reduce the number of parameters and training time.

AMMT

It consists of three components, i.e., memory pool, feature transfer module, and feature aggregation module. The memory pool first extracts the prototype pattern of appearance and motion features. Then we feed these feature maps into the feature transfer module (AMFT) to learn the transferred feature. Finally, we aggregate the encoder feature, memory priors, and transferred feature.

Memory Pool. Compared with the diversity and unboundedness of abnormal event types, regular events available for training can be exhaustive. So it may be feasible to sum up the prior information of a regular pattern in theory. However, the original feature contains the prior information of the normal event and its specific information. Only the prior information has a strong correlation between the two modalities. Therefore, we introduced a memory module with a discrete latent space combined with the traditional reconstruction model to extract prototype features and stored them in the memory pool.

Specifically, we design separate memory modules for appearance and motion information, called AppMemPool and MotMemPool. Our memory module defines a latent embedding space $\mathbf{M} \in \mathbb{R}^{D \times N}$ containing N memory items with dimension D. We denote the appearance and motion memory pool as $\mathbf{M^a}$ and $\mathbf{M^o}$, respectively. The memory pool receives the feature from the encoder, z_e^a and z_e^o as inputs. Then it calculates each spatial feature from the encoder to each memory item and picks the K closest items as the memory prior features z_m^a and z_m^o. We show the whole procedure from line 3 to 6 of Algorithm 1.

Appearance-Motion Feature Transfer Module. We model the appearance and motion correlation in the memory prior space. Specifically, after receiving the memory prior features z_m^a and z_m^o, we firstly use a 1 x 1 convolution layer for feature reduction over the prior features. Then we apply two mapping functions $\Phi_{a \to o}$ and $\Phi_{o \to a}$ to learn the consistent correlation between the appearance and motion priors and obtain the transferred features $z_m^{\hat{o}}$ and $z_m^{\hat{a}}$. The whole procedure is illustrated from line 7 to line 10 in Algorithm 1. Compared with the previous method (Vu et al. 2019), which performs the appearance to motion in feature space, we learn the consistent correlation between the appearance and motion in the prior space. Because in the memory space, it would avoid the side effects of the complex background, and directly learning the transformation from the motion (optical flow) to the appearance (image) is an ill-condition problem, using the prior information would make the problem more feasible.

Feature Aggregation. Since the memory items only contain the prior information, they would lose each input's unique information. To make the features more representative, we aggregate the original feature from the encoder z_e^a (z_e^o), memory priors z_m^a (z_m^o), and the transferred features $z_m^{\hat{o}}$ ($z_m^{\hat{a}}$). Finally, we feed the fused features into the decoder to predict the future frame $\hat{F}_t$ and the optical flow $\hat{O}_{t-1}$. We illustrate these from line 11 to 14 in Algorithm 1.

Loss Function

Let F denote an rgb image sequence, $\hat{F}$ denotes the prediction of F, O denotes the corresponding optical-flow clip of F, $\hat{O}$ denotes the prediction of O. When given $F_{1 \ldots t-1}$ and $O_{1 \ldots t-2}$, the model output $\hat{F}_t$ and $\hat{O}_{t-1}$. To generate a more realistic frame, we leverage the GAN variant (Least Square GAN (Mao et al. 2017)) in our model. We follow the original training procedure of (Mao et al. 2017) with a minmax game. Specifically, we alternatively train the generator and discriminator. The generator tries to produce a realistic-looking result and fool the discriminator. The discriminator tries to classify which image is real or fake (generated).

Training G. To train the generator of AMMC-Net, we construct the following loss functions from the pixel space and feature space of appearance and motion signals, respectively.

$$\mathcal{L}_G = \mathcal{L}_A + \mathcal{L}_M + \mathcal{L}_C \tag{1}$$

For the appearance, we adopt intensity, gradient, flow and adversarial losses ($\mathcal{L}_{int}$, $\mathcal{L}_{gdl}$, $\mathcal{L}_{op}$ and $\mathcal{L}_{adv}^G$, respectively).

$$\begin{aligned} \mathcal{L}_A = \lambda_{int}\mathcal{L}_{int}(\hat{F}_t, F_t) + \lambda_{gdl}\mathcal{L}_{gdl}(\hat{F}_t, F_t) \\ + \lambda_{op}\mathcal{L}_{op}(\hat{F}_t, F_t, F_{t-1}) + \lambda_{adv}\mathcal{L}_{adv}^G(\hat{F}_t) \end{aligned} \tag{2}$$

where the hyper-parameters $\lambda_{int}, \lambda_{gdl}, \lambda_{op}, \lambda_{adv}$ are used to adjust the importance of each part.

We adopt ℓ_2 distance to minimize the loss between predicted frame $\hat{F}$ and its ground true F in intensity space:

$$\mathcal{L}_{int}(\hat{F}_t, F_t) = \|\hat{F}_t - F_t\|_2^2 \tag{3}$$

To sharpen the image prediction, followed (Mathieu, Couprie, and Lecun 2016), a gradient loss is adopted in our loss function. It directly penalizes the differences of image gradient between prediction and its ground truth:

$$\begin{aligned} \mathcal{L}_{gdl}(\hat{F}_t, F_t) = \sum_{i,j} \left\| |\hat{F}_{i,j} - \hat{F}_{i-1,j}| - |F_{i,j} - F_{i-1,j}| \right\|^{\alpha} \\ + \left\| |\hat{F}_{i,j} - \hat{F}_{i,j-1}| - |F_{i,j} - F_{i,j-1}| \right\|^{\alpha} \end{aligned} \tag{4}$$

where i, j denote the spatial index of a video frame, α can adjust the sharpness degree of the predicted image.

To maintain the coherence of motion, which is vital for VAD, we adopt a motion constraint loss (Liu et al. 2018) to enforce the optical flow between the predicted frames to be close to their ground truth.

$$\mathcal{L}_{op}(\hat{F}_t, F_t, F_{t-1}) = \|f(\hat{F}_t, F_{t-1}) - f(F_t, F_{t-1})\|_1^1 \tag{5}$$

where f is a pre-trained flownet (Reda et al. 2017).

To produce a predictive image that looks as much like the real image as possible(to fool the discriminator), the adversarial loss of generator (Gauthier 2014) is adopted as follows:

$$\mathcal{L}_{adv}^G(\hat{F}_t) = \sum_{i,j} \frac{1}{2}\|\mathcal{D}(\hat{F_{i,j}}) - 1\|_2^2. \tag{6}$$

For the motion, we apply the smoothed ℓ_1 loss between the predicted optical flow and that of the ground-truth, shown in Equation (7), because it is more suitable for the high sparsity of the optical flow (Girshick 2015).

$$\begin{aligned} \mathcal{L}_M = \lambda_{sth}\mathcal{S}_{L1}(\hat{O}_t - O_t) \\ \mathcal{S}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

To optimize the memory module (AppMemPool and MotMemPool), we push the query feature of both the appearance z_e^a and motion z_e^o to be close to that of the selected memory item, e^a and e^o, shown as follow:

$$\begin{aligned} \mathcal{L}_C = \mathcal{L}_{lat}(z_e^a, e^a) + \mathcal{L}_{lat}(z_e^o, e^o) \\ \mathcal{L}_{lat}(z_e, e) = \|\mathrm{sg}[z_e] - e\|_2^2 + \beta\|z_e - \mathrm{sg}[e]\|_2^2. \end{aligned} \tag{8}$$

Since, there is a non-differentiable argmax operation in our memory network, we follow a stop gradient trick sg (Bengio, Leonard, and Courville 2013; Den Oord, Vinyals, and Kavukcuoglu 2017) to handle the loss backpropagation. Here, β denotes the weight of two types of loss items.

Training D. To force the generator to learn the normal distribution, the discriminator tries to classify the ground-truth frame as real and the predicted frame as fake. Here, we follow the LSGAN (Mao et al. 2017), shown as follow:

$$\mathcal{L}_D = \sum_{i,j} \frac{1}{2}\|\mathcal{D}(F_{i,j}) - 1\|_2^2 + \frac{1}{2}\|\mathcal{D}(\hat{F}_{i,j})\|_2^2. \tag{9}$$

Anomaly Detection in Testing Data

Memory Commit Error. Since we have learned the prior of regular events in memory, the anomalies might have a large distance from the query feature z_e^a to the memory prototype z_m^a, while the normal patterns will result in a small one. We use a memory commit error, shown as follow, to measure the distance:

$$C(z_e^a, z_m^a) = \|z_e^a - z_m^a\|_2^2. \tag{10}$$

Low Commit Error of the t^{th} frame indicates that it is more likely to be normal.

Image Prediction Error. A lot of related work (Liu et al. 2018) has shown that PSNR can increase the gap between normal and abnormal events compared with MSE under the same circumstances. Thus we adopt PSNR in our method.

$$P(F, \hat{F}) = 10 \log_{10} \frac{[\max_{\hat{F}}]^2}{\frac{1}{N}\sum_{i=0}^{N}(F_i - \hat{F}_i)^2} \tag{11}$$

Anomaly Score. Many previous methods (Nguyen and Meunier 2019) only considered the error in the pixel space as an anomaly indicator, ignoring the effect of the error in the feature space on anomaly detection. Our AMMC-Net makes up for this defect. In the testing phase, combining the Memory Commit Error in latent space and the Image Prediction Error in pixel space, we can determine whether a case is an anomaly. The final normal score can be derived as follows:

$$S(t) = (1 - \lambda_c) * \mathcal{H}(P(F, \hat{F})) + \lambda_c * \mathcal{H}(C(z_e^a, z_m^a)) \tag{12}$$

Where λ_c denotes the weight between two types of errors. $\mathcal{H}$ indicates the min-max based normalization operation, and we normalize two types of Error of all frames in the whole testing video to the range [0, 1].

$$\mathcal{H}(t) = \frac{\mathcal{H}(t) - \min_t \mathcal{H}(t)}{\max_t \mathcal{H}(t) - \min_t \mathcal{H}(t)} \tag{13}$$

Therefore, the final $S(t)$ represents the normal degree of a particular frame. The larger, the more normal, the smaller, the more abnormal, and we can determine whether a specific frame is normal or abnormal by selecting a threshold.

Experiments

We conduct the experiments on three challenging video anomaly detection datasets, including UCSD Pedestrian (Ped1 and Ped2) dataset (Li, Mahadevan, and Vasconcelos 2013), CUHK Avenue (Lu, Shi, and Jia 2013) and ShanghaiTech dataset (Luo, Liu, and Gao 2017b). We compare our proposed methods with others and perform solid ablation studies to analyze each system component.

Experiment Setup

Dataset and Evaluation Metric. We follow the previous setting (Liu et al. 2018), and use the frame-level Area Under Curve (AUC) as the evaluation metric. A higher value indicates better performance.

UCSD Pedestrian. It has two different scenes, Ped1 and Ped2. The difference between the two subsets is the walking direction (toward and away from the camera in Ped1, parallel to the camera plane in Ped2). All of these abnormal cases include bicycles, skateboarders, wheelchairs, and vehicles within the regular crowd.

CUHK Avenue. It contains 47 anomalies, including unusual actions (e.g., throwing objects, loitering, and running.), wrong direction, and abnormal objects (e.g., bicycle).

ShanghaiTech. It covers 13 different scenes and 130 abnormal events. Objects except pedestrians (e.g., vehicles) and strenuous motion (e.g., fighting and chasing) are treated as anomalies.

Implementation Details. Our proposed method contains the training and testing phases. The training phase can be split into three parts: data processing, pre-training, and joint training. The testing phase pipeline can be divided into two stage: future frame prediction and anomaly detection based on the combination of commit error and prediction error. Considering that it is not easy to optimize the whole model directly, and it is easy to get a trivial solution, this paper proposes a two-stage optimization method consisting of pre-training and joint training. Firstly, we optimized the encoder, decoder, and memory pool based on image prediction loss and feature commit loss. Based on the per-trained model, we use the total loss function to focus on training the appearance-motion feature Transfer module(AMFT) and fine-tune the previous module. More details can be founded in the supplementary materials.

Comparison with Existing Methods

Quantitative Comparison. In this section, we use frame-level ground truth to perform a quantitative evaluation. The frame-level ground truth indicates whether one or more anomalies occur in a test frame. To show the effectiveness of our AMMC-Net, we compare our method with different prediction-based method (Liu et al. 2018), memory-based method (Gong et al. 2019; Park, Noh, and Ham 2020) and two-stream-based method (Prawiro et al. 2020). Table 1 shows a quantitative comparison of the different methods in terms of Area Under Curve (AUC). The evaluation results on the above benchmark datasets demonstrate our model's potential with competitive performance compared with the state-of-the-art methods.

Qualitative Comparison and Analysis. To understand our method's advantages and show how it detects abnormal events, we also qualitatively compare our proposed AMMC-Net with the twostream-based baseline method(more details can be founded in the supplementary file). We further analyze that the abnormal events mainly come from three aspects: 1)appearance anomalies, such as the invalid object or

	Ped2	Avenue	ShanghaiTech
ConvLSTM-AE[1]	88.1%	77.0%	-
AE-Conv3D[2]	91.2%	77.1%	-
AMDN(TwoStream)[3]	90.8%	-	-
Stacked RNN[4]	92.2%	81.7%	68.0%
AbnormalGAN[5]	93.5%	-	-
FFP[6]	95.4%	85.1%	72.8%
AnomalyNet[7]	94.9%	86.1%	-
Autoregressive-ConvAE[8]	95.4%	-	72.5%
MemAE[9]	94.1%	83.3%	71.2%
TwoStreamDec[10]	96.1%	86.4%	-
DDGAN[11]	94.9%	85.6%	73.7%
MGAD [12]	97.0%	88.5%	70.5%
IPRAD[13]	96.2%	83.7%	71.5%
Our Method	**96.6%**	**86.6%**	**73.7%**

Table 1: The frame-level AUC performance of different comparison methods, including [1](Luo, Liu, and Gao 2017a), [2](Zhao et al. 2017), [3](Xu et al. 2017), [4](Luo, Liu, and Gao 2017b), [5](Ravanbakhsh et al. 2017), [6](Liu et al. 2018), [7](Zhou et al. 2019), [8](Abati et al. 2019), [9](Gong et al. 2019), [10](Prawiro et al. 2020), [11] (Tang et al. 2020), [12](Park, Noh, and Ham 2020), [13](Dong, Zhang, and Nie 2020).

camera intruding; 2)irregular fast motions of human, such as fighting, chasing, or running; 3)unusual slow motions of human, such as loitering. To demonstrate that our proposed AMMC-Net is more capable of detecting the abnormal events, we illustrate our method's anomaly score map and the baseline method on Avenue testing set in Figure 3. Our approach has a higher response to the abnormal area and a lower response to the normal area than the twostream-based baseline. This result also reflects that our method could capture the consistency between the normal appearance and the normal motion.

We also calculate the score gap between the normal and abnormal scores and illustrate it in Figure 2. It demonstrates that our method achieves a higher score gap and can discriminate the anomaly from the normality.

Ablation Study

In the previous section, we perform numerous quantitative comparison experiments and visualization to prove our method's effectiveness compared with the previous state-of-the-art methods. In this section, we perform some further experiments to analyze each component's importance in our approach.

Effectiveness of Memory Consistency. To verify the effectiveness of the memory consistent module in our method, we evaluate our approach with a naive baseline model(two-stream-based model without AMMT, more details can be founded in supplementary file) on Ped2, Avenue, and ShanghaiTech datasets. From Table 2, we can see that our method with memory consistency achieves a higher AUC than that without memory consistency.

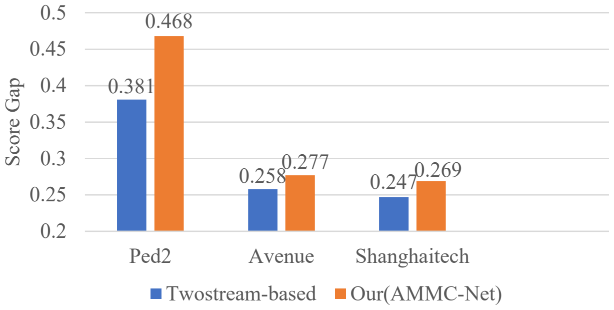

Figure 2: We perform the following experiments on the testing sets of data Ped2, Avenue, and Shanghaitech. We first calculate the score of all normal frames and average them, then perform the same operation on abnormal frames, and finally find the difference between the two averages and call it the score gap. Intuitively, the larger the difference, the more significant the difference between normal and abnormal, indicating better system performance.

Memory Consistency	Ped2	Avenue	Shanghaitech
Without	95.1%	84.9%	71.5%
With	**96.6%**	**86.6%**	**73.7%**

Table 2: Comparison of our proposed AMMC-Net with a twostream-based baseline without memory consistency(more details can be founded in supplementary file). We measure the average AUC on Ped2, Avenue, and ShanghaiTech datasets.

Impact of each component in AMMC-Net. In this section, we will study the impact of each component in AMMC-Net, including Memory Pool, AMFT, original feature z_e obtained from the encoder, feature z_m obtained from Memory Pool, as well as the commit error C. The first row in the Table 3 is taken from the previous Table 2 as the baseline of this section. All details can be founded in the Method section. We verify its impact on the evaluation indicator AUC by removing each component one by one. We can see that our model with the memory module gives a slightly improved but not significant result from the second row. The third row shows that the AUC performance is significantly enhanced because AMFT realizes the cooperation of two kinds of information vital for video anomaly detection. The next two rows respectively indicate that our proposed multi-stage feature fusion strategy can improve performance. The last row shows that combining the information of feature space and pixel space can significantly improve anomaly detection performance since we make full use of the error information in feature space by using the memory pool.

Impacts of the number of Selected Memory Items K. Our method uses multiple memory items in the Memory Pool to represent a query vector to deal with complicated real events. More number of the selected memory items might improve the network's representation capability, making the anomalies predictable and decreasing the anomaly

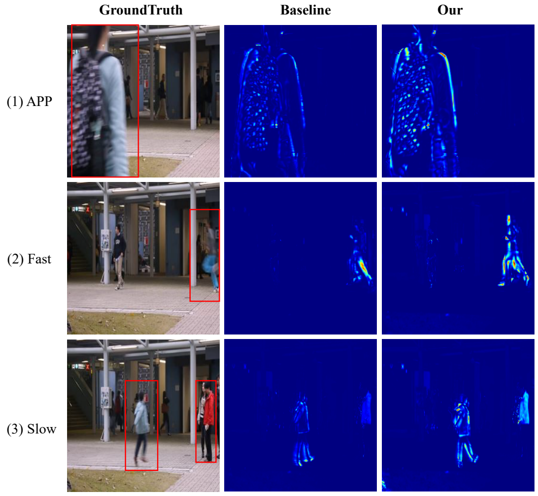

Figure 3: Anomaly score map of twostream-based baseline and our proposed AMMC-Net on three abnormal frames of Avenue dataset. The first row shows the abnormality caused by appearance. The second-row shows anomaly caused by a rapid movement. The last row shows an abnormal event caused by a slow-moving child and a loitering man. We can see that our model localizes the regions of abnormal events (in the red bounding box) significantly.

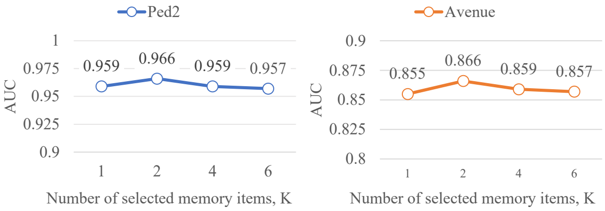

Figure 4: The AUC results of our methods on Ped2 and Avenue datasets with a different number of selected memory items K, which ranges from 1 to 6.

detection performance. In this section, we perform our methods with a different number of the selected memory items, ranging from 1 to 6 on the Ped2 and Avenue datasets, respectively. As shown in Figure 4, with K increases, the AUC first increases and then decreases, indicating that $K = 2$ achieves the best performance.

Impacts of the Memory Size N and Dimension D. Our proposed memory module defines a latent embedding space $\mathrm{M} \in \mathbb{R}^{N \times D}$, where N denotes the number of memory item and D denotes the embedding dimension of the memory item. We use the Ped2 to study the robustness of N and D. We conduct the experiments using different memory size settings and show the AUC values in Figure 5. As shown in Figure 5, D and N are two important hyper-parameters of the memory module; With the increase of N and D, the performance will gradually rise first and then slowly fall. This finding is intuitive. If N or D is too large, it will be challenging to optimize the model. If N or D is too small, it will limit the model's capability. We can precisely adjust the model's representation ability for different datasets by revising these two parameters.

Task	Memory	AMFT	z_e	z_m	C	Ped2
A	-	-	-	-	-	95.1%
B	✓					95.3%
C	✓	✓				96.1%
D	✓	✓	✓			96.2%
E	✓	✓	✓	✓		96.4%
F	✓	✓	✓	✓	✓	96.6%

Table 3: Ablation studies of each component in our AMMC-Net. We report the results on the Ped2 dataset of our method. The 1^{st} row is the baseline method, the 2^{nd} row means that with memory pool. The 3^{rd} row means that with additional Appearance-Motion feature translation module. The 4^{th} and 5^{th} rows represent that our method fuses the extra features from the encoder z_e and the features from memory pool z_m. The last row means whether using the commit error C in testing phase or not.

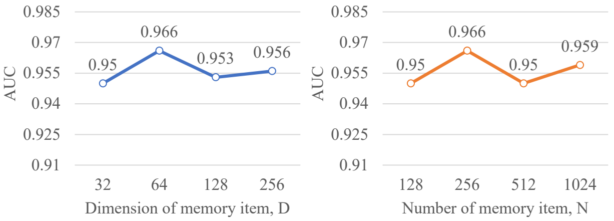

Figure 5: Ablation studies of memory size and dimension. We measure the average AUC on UCSD Ped2. N denotes the number of memory items, and D represents the memory item's embedding dimension.

Conclusions

In this paper, we model the consistency between the appearance and the motion to tackle the video anomaly detection. We first optimize an appearance and motion prediction network to construct two memory pools. Then, we use an appearance-motion feature transfer (AMFT) network to implement the communication and fusion operation between appearance and motion patterns. In the test phase, given an input sequence consisting of images and their optical flow, appearance-motion features are extracted from the AppMemPool and MotMemPool with AMFT. Finally, we combine the predicting error in pixel space with the commit error in feature space to calculate the score of a testing frame. Many experiments prove our method's validity compared with the existing state-of-the-art. Solid ablation studies demonstrate the effectiveness of our proposed AMMC-Net in capturing the consistency between appearance and motion for video anomaly detection.

Acknowledgments

The work was supported by National Key R&D Program of China (2018AAA0100704), NSFC (61932020, 61876043, 61976052), Science and Technology Commission of Shanghai Municipality (Grant No. 20ZR1436000), Science and Technology Planning Project of Guangzhou (201902010058), and ShanghaiTech-Megvii Joint Lab.

References

Abati, D.; Porrello, A.; Calderara, S.; and Cucchiara, R. 2019. Latent space autoregression for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 481–490.

Bengio, Y.; Leonard, N.; and Courville, A. 2013. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv: Learning* .

Den Oord, A. V.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural Discrete Representation Learning 6306–6315.

Dong, F.; Zhang, Y.; and Nie, X. 2020. Dual Discriminator Generative Adversarial Network for Video Anomaly Detection. *IEEE Access* .

Gauthier, J. 2014. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester* 2014(5): 2.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Gong, D.; Liu, L.; Le, V.; Saha, B.; Mansour, M. R.; Venkatesh, S.; and Den Hengel, A. V. 2019. Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection. *arXiv: Computer Vision and Pattern Recognition* .

Hasan, M.; Choi, J.; Neumann, J.; Roychowdhury, A. K.; and Davis, L. S. 2016. Learning Temporal Regularity in Video Sequences 733–742.

Ionescu, R. T.; Khan, F. S.; Georgescu, M.; and Shao, L. 2019. Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video 7842–7851.

Kiran, B. R.; Thomas, D. M.; and Parakkal, R. 2018. An Overview of Deep Learning Based Methods for Unsupervised and Semi-Supervised Anomaly Detection in Videos. *Journal of Imaging* 4(2): 36.

Li, W.; Mahadevan, V.; and Vasconcelos, N. 2013. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence* 36(1): 18–32.

Liu, W.; Luo, W.; Lian, D.; and Gao, S. 2018. Future Frame Prediction for Anomaly Detection - A New Baseline 6536–6545.

Lu, C.; Shi, J.; and Jia, J. 2013. Abnormal Event Detection at 150 FPS in MATLAB 2720–2727.

Luo, W.; Liu, W.; and Gao, S. 2017a. Remembering history with convolutional LSTM for anomaly detection 439–444.

Luo, W.; Liu, W.; and Gao, S. 2017b. A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework 341–349.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z.; and Paul Smolley, S. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2794–2802.

Mathieu, M.; Couprie, C.; and Lecun, Y. 2016. Deep multiscale video prediction beyond mean square error .

Nguyen, T. N.; and Meunier, J. 2019. Anomaly Detection in Video Sequence with Appearance-Motion Correspondence. 1273–1283.

Park, H.; Noh, J.; and Ham, B. 2020. Learning Memory-guided Normality for Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14372–14381.

Prawiro, H.; Peng, J.-W.; Pan, T.-Y.; and Hu, M.-C. 2020. Abnormal Event Detection in Surveillance Videos Using Two-Stream Decoder. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 1–6. IEEE.

Ravanbakhsh, M.; Nabi, M.; Sangineto, E.; Marcenaro, L.; Regazzoni, C. S.; and Sebe, N. 2017. Abnormal event detection in videos using generative adversarial nets 1–5.

Reda, F.; Pottorff, R.; Barker, J.; and Catanzaro, B. 2017. flownet2-pytorch: Pytorch implementation of FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. https://github.com/NVIDIA/flownet2-pytorch.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Sabokrou, M.; Fathy, M.; and Hoseini, M. 2016. Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder. *Electronics Letters* 52(13): 1122–1124.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; and Woo, W. 2015. Convolutional LSTM Network: a machine learning approach for precipitation nowcasting 802–810.

Simonyan, K.; and Zisserman, A. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos 568–576.

Tang, Y.; Zhao, L.; Zhang, S.; Gong, C.; Li, G.; and Yang, J. 2020. Integrating prediction and reconstruction for anomaly detection. *Pattern Recognition Letters* 129: 123–130.

Vu, H.; Nguyen, T.; Le, T.; Luo, W.; and Phung, D. 2019. Robust Anomaly Detection in Videos using Multilevel Representations 33(01): 5216–5223.

Wang, X.; Jabri, A.; and Efros, A. A. 2019. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2566–2576.

Xu, D.; Yan, Y.; Ricci, E.; and Sebe, N. 2017. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding* 156: 117–127.

Yan, S.; Smith, J. S.; Lu, W.; and Zhang, B. 2018. Abnormal Event Detection from Videos using a Two-stream Recurrent Variational Autoencoder. *IEEE Transactions on Cognitive and Developmental Systems* 1–1.

Zhao, Y.; Deng, B.; Shen, C.; Liu, Y.; Lu, H.; and Hua, X. 2017. Spatio-Temporal AutoEncoder for Video Anomaly Detection 1933–1941.

Zhou, J. T.; Du, J.; Zhu, H.; Peng, X.; Liu, Y.; and Goh, R. S. M. 2019. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security* 14(10): 2537–2550.