Deep Anomaly Discovery from Unlabeled Videos via Normality Advantage and Self-Paced Refinement —Supplementary Material—

1. Dataset Details

All benchmark datasets adopted in our paper are publicly available datasets that are frequently-used for VAD in the literature. The official agents of those datasets have guaranteed that all data are collected, released, and used with the consent of subjects. We provide the public download links to these datasets in the footnote[1][2][3].

2. Foreground Localization for LBR

Algorithm 1 Foreground Localization

Input: A video frame I and its optical flow OF, pre-trained object detector M, threshold $T_c, T_a, T_o, T_b, T_{ar}$
Output: Foreground set $\mathcal{F}$

$\mathcal{F}_o \leftarrow$ *ObjDet*(I, M, T_c) # Detect foreground objects
$\mathcal{F}_a = \{\}$ # To get appearance based foreground set
for $f_o \in \mathcal{F}_o$ **do**
  if $Area(f_o) > T_a$ **and** $Overlap(f_o, \mathcal{F}_o) < T_o$ **then**
    $\mathcal{F}_a = \mathcal{F}_a \cup \{f_o\}$
  end if
end for
$OF_b \leftarrow$ *OFBin*(OF, T_b) # Optical flow binarization
$OF_b \leftarrow$ *ForeSub*$(OF_b, \mathcal{F}_a)$ # Subtract appearance based foreground
$C \leftarrow$ *ContourDet*(OF_b) # Contour detection
$\mathcal{F}_m = \{\}$ # To get motion based foreground set
for $c \in C$ **do**
  $f_m = BoundingBox(c)$ # Get bounding box of contour
  if $Area(f_m) > T_a$ **and** $\frac{1}{T_{ar}} < AspectRatio(f_m) < T_{ar}$ **then**
    $\mathcal{F}_m = \mathcal{F}_m \cup \{f_m\}$
  end if
end for
$\mathcal{F} = \mathcal{F}_a \cup \mathcal{F}_m$

Following the localization scheme proposed in [22], we show the whole procedure of foreground localization in Algorithm 1. Unlike [22] that uses temporal gradients as motion cues to localize novel or blurred foreground, we use optical flow (OF) instead, because it is a more accurate motion representation that enjoys better robustness to low-level noises. Concretely, the foreground localization scheme consists of an appearance based localization stage and a motion based localization stage, which are detailed below:

For appearance based localization stage, the goal is to build an appearance based foreground set $\mathcal{F}_a$. To this end, a pre-trained object detector M first performs object detection on a raw video frame I, so as to obtain a preliminary foreground object set $\mathcal{F}_o$. Each object $f_o \in \mathcal{F}_o$ enjoys a confidence score that is greater than T_c in detection. Then, two simple but efficient heuristic rules are used to filter out those object regions that are either too small ($Area(f_o) \leq T_a$) or significantly overlapped with other object regions ($Overlap(f_o, \mathcal{F}_o) \geq T_o$) from $\mathcal{F}_o$, while the rest of foreground objects are put into the appearance based foreground set $\mathcal{F}_a$. Thus, most common objects in daily life can be localized precisely.

When it comes to the motion based localization stage, the goal is to establish a motion based foreground set $\mathcal{F}_m$. Specifically, the frame I's OF is first binarized by a threshold T_b into a binary map OF_b. The highlighted areas in this binary map is then used to indicate regions with intense motion. Afterwards, foreground objects in $\mathcal{F}_a$ are subtracted from the binary map OF_b, which avoids duplication of localization and facilitates localizing novel foreground objects more precisely. Finally, contour detection is performed on the binary map OF_b to obtain the contour set C of novel objects and their corresponding bounding boxes. Similarly, two simple rules are introduced to filter out overly small foreground objects ($Area(f_m) \leq T_a$) or objects with irregular aspect ratio. The remaining foreground objects are collected to form the motion based foreground set $\mathcal{F}_m$. In this way, novel or deformed foreground objects that are not detected by pre-trained object detector can be localized.

The final foreground set $\mathcal{F}$ is a union of appearance based foreground set $\mathcal{F}_a$ and motion based foreground set $\mathcal{F}_m$. The settings of parameters in foreground localization are specified in next section.

[1] http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm
[2] http://www.cse.cuhk.edu.hk/leojia/projects/detectabnormal/dataset.html
[3] https://svip-lab.github.io/dataset/campus_dataset.html, which is subject to MIT License

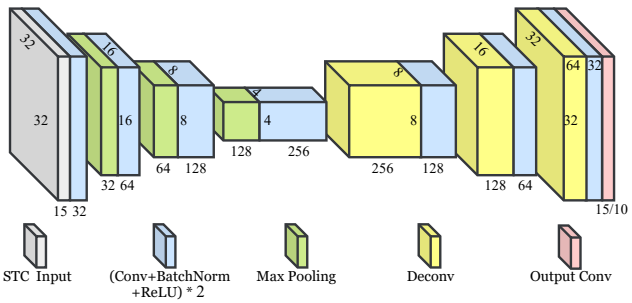

Figure 1. CAE architecture for reconstruction.

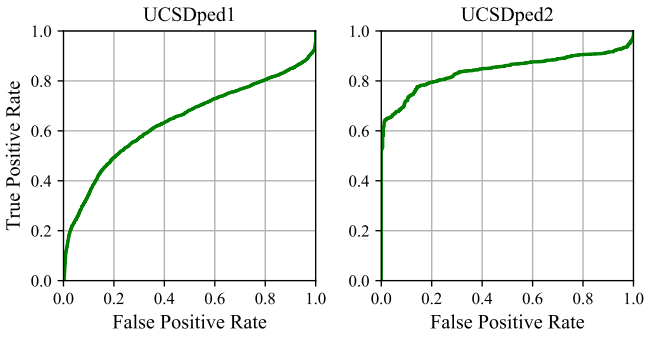

Figure 2. Pixel-level ROC curves of LBR-SPR* with motion enhancement (ME) under partial mode.

3. More Implementation Details

We provide more implementation details of our experiments in this section. For foreground localization, we use mmdetection toolbox[4] [1] for object detection. Specifically, we adopt YOLO v3 [14] pre-trained on Microsoft COCO dataset [9] to localize foreground objects because it achieves a good trade off between detection performance and speed (up to 60 fps). Meanwhile, we use the pre-trained FlowNet v2[5] [6] to estimate optical flow for each video frame. As for thresholds, we set the confidence score threshold $T_c = 0.2$, overlapping ratio threshold $T_o = 0.6$, optical flow binarization threshold $T_b = 1$ and aspect ratio threshold $T_{ar} = 10$ in all experiments. Considering the resolution and foreground object scale of different datasets, we set area threshold T_a to 10×10 for UCSDped1/UCSDped2, 40×40 for avenue and 30×30 for ShanghaiTech. To alleviate the evident foreground depth variations in UCSDped1, which may lead to significant differences in object scales and undermine the performance, we evenly divide the video frame into 4×1 local regions and process foreground objects in each region by a separated DNN. We simply assign each object to the region where it is centered. Hence, we train four DNNs to score objects in four regions respectively. As for the DNN architecture for reconstruction, we adopt a 7-layer fully convolutional autoencoder (CAE), the architecture of which is shown in Fig. 1. For SPR scheme, warm-up epoch is typically set as $T' = 5$, while it is set to 20 for the motion enhanced LBR-SPR on Avenue dataset, so as to enable a stable performance. Following previous works [10, 16], we also apply a sliding window with window size 10 to smooth frame anomaly scores. Our implementation can be accessed at https://github.com/yuguangnudt/LBR_SPR.

[4] https://github.com/open-mmlab/mmdetection, which is subject to Apache License 2.0.

[5] https://github.com/vt-vl-lab/flownet2.pytorch, which is subject to Apache License 2.0.

Table 1. Pixel-level AUROC comparison with UVAD and classic VAD methods. Note that LBR-SPR* indicates the performance of LBR-SPR under partial mode, while LBR-SPR+ indicates the performance under merge mode. ME denotes motion enhancement.

Setup	Method	Ped1	Ped2
Classic VAD	AMDN [19]	67.2%	-
	WTA-CAE [15]	68.7%	89.3%
	AM-GAN [13]	70.3%	-
	Recounting [5]	-	89.1%
	TCP [12]	64.5%	-
	AnomalyNet [24]	45.2%	52.8%
	MLAD [17]	66.6%	97.2%
	DeepOC [18]	63.1%	95.0%
	SIGNet [3]	51.6%	48.4%
UVAD	UM [16]	52.4%	-
	LBR-SPR* (w/o ME)	**62.4%**	**77.5%**
	LBR-SPR* (w/ ME)	**64.0%**	**84.2%**
	LBR-SPR+ (w/o ME)	**62.0%**	**84.7%**
	LBR-SPR+ (w/ ME)	**63.9%**	**87.1%**

4. Other Evaluation Metrics

In VAD, two types of criterion are usually used for evaluation: Frame-level criterion and pixel-level criterion [11]. For frame-level criterion, one abnormal frame is viewed to be correctly detected if any pixel on this frame is detected as abnormal; For pixel-level criterion, an abnormal frame is viewed to be correctly detected only when more than 40% anomaly pixels on this frame are identified. Thus, frame-level criterion focuses on frame-level detection, while pixel-level criterion also emphasizes anomaly localization. Under either criterion, AUROC and equal error rate (EER) can be computed as quantitative measure of performance. Recent VAD works usually report frame-level AUROC only, since many deep VAD methods are performed on a per-frame basis. For a comprehensive evaluation on UCSDped1 and UCSDped2, we report pixel-level AUROC and EER under both types of criterion as the literature does. Since previous VAD works usually report frame-level EER only on Avenue and ShanghaiTech, we simply follow their practice. First, we visualize some pixel-level and frame-level ROC curves

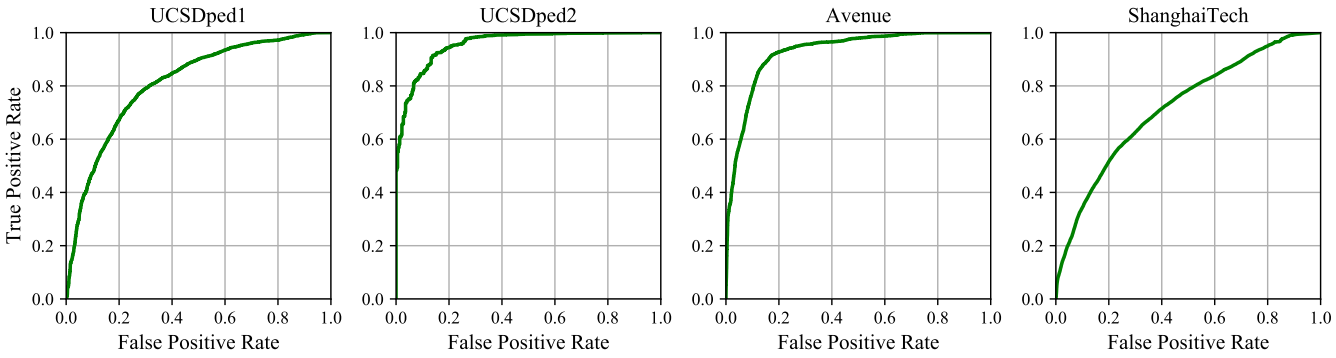

Figure 3. Frame-level ROC curves of LBR-SPR* with motion enhancement (ME) under partial mode.

Table 2. Results of EER. LBR-SPR* indicates the performance of LBR-SPR under partial mode, while LBR-SPR+ indicates the performance under merge mode. EER^F represents frame-level EER, while EER^P represents pixel-level EER. ME denotes motion enhancement.

Setup	Method	Ped1		Ped2		Avenue	SHTech
		EER^F	EER^P	EER^F	EER^P	EER^F	EER^F
Classic VAD	CAE [4]	27.9%	-	21.7%	-	25.1%	-
	AMDN [19]	16.0%	40.1%	17.0%	-	-	-
	AM-GAN [13]	8.0%	35.0%	14.0%	-	-	-
	ST-CAE [23]	15.3%	-	16.7%	-	24.4%	-
	WTA-CAE [15]	14.8%	35.7%	8.9%	16.9%	24.2%	-
	R-VAE [20]	32.4%	-	15.5%	-	27.5%	-
	AnomalyNet [24]	25.2%	-	10.3%	-	22.0%	-
	AnoPCN [21]	-	-	10.0%	-	20.2%	33.7%
	DeepOC [18]	23.4%	-	8.8%	-	18.5%	-
	VEC [22]	-	-	7.5%	-	17.9%	31.5%
UVAD	LBR-SPR* (w/o ME)	25.7%	39.1%	14.3%	24.9%	18.5%	36.0%
	LBR-SPR* (w/ ME)	25.5%	37.7%	12.2%	20.4%	13.2%	33.7%
	LBR-SPR+ (w/o ME)	28.8%	40.3%	9.1%	19.2%	18.3%	34.8%
	LBR-SPR+ (w/ ME)	25.8%	38.8%	9.7%	18.5%	17.8%	33.4%

yielded by LBR-SPR in Fig. 2 and Fig. 3 respectively for a intuitive demonstration. Second, we compare pixel-level AUROC of the proposed LBR-SPR and those existing VAD methods that have reported pixel-level results in Table 1: For one thing, LBR-SPR still significantly outperforms the UVAD method UM [16], which is the only UVAD method that reports pixel-level AUROC, by approximately 10% AUROC on UCSDped1 dataset; For another, the pixel-level performance of the proposed LBR-SPR is also highly competitive when compared with classic VAD methods. As to EER, we report the results in Table 2. Since no UVAD methods has reported EER, we simply compare our UVAD methods with classic VAD methods that report EER. As can be seen from the table, results in Table 2 exihibit a very similar trend to previous AUROC comparison, which demonstrates the effectiveness of our LBR-SPR again.

5. Other SP Regularizers for SPR

In addition to the mixture SP regularizer leveraged in this paper, we also explore the customization of other SP regularizers to implement SPR, i.e., hard SP regularizer [8] and linear SP regularizer [7]. Specifically, hard and line SP regularizer have the following forms:

$$\begin{aligned} g^h(\mathbf{v}|\lambda) &= -\lambda \sum_{i=1}^{n} v_i \\ g^l(\mathbf{v}|\lambda) &= \frac{1}{2}\lambda \sum_{i=1}^{n} (v_i^2 - 2v_i) \end{aligned} \tag{1}$$

where $g^h(\mathbf{v}|\lambda)$ and $g^l(\mathbf{v}|\lambda)$ represent the hard and linear SP regularizer respectively. Following the same optimization strategy in the manuscript, when $\boldsymbol{\theta}$ is fixed, the corresponding closed-formed solutions to v_i^* are given as follows:

$$\begin{aligned} v_i^{h*} &= \begin{cases} 0, & L_i(\boldsymbol{\theta}) \geq \lambda \\ 1, & L_i(\boldsymbol{\theta}) < \lambda \end{cases} \\ v_i^{l*} &= \begin{cases} 0, & L_i(\boldsymbol{\theta}) \geq \lambda \\ 1 - \dfrac{L_i(\boldsymbol{\theta})}{\lambda}, & L_i(\boldsymbol{\theta}) < \lambda \end{cases} \end{aligned} \tag{2}$$

where v_i^{h*} and v_i^{l*} represent the optimal v_i^* for hard and linear SP regularizer respectively. To determine λ and enable a progressive removal of anomalies, we adopt the same strategy as the mixture SP regularizer, which gradually lowers λ

Table 3. Different SP regularizers for LBR-SPR with ME.

Mode	Method	Ped1	Ped2	Avenue	SHTech
Partial	LBR	79.7%	90.9%	90.4%	71.7%
	LBR-SPR (Hard)	80.5%	95.8%	91.1%	72.1%
	LBR-SPR (Linear)	81.6%	96.4%	92.5%	71.8%
	LBR-SPR (Mixture)	81.1%	95.7%	92.8%	72.1%
Merge	LBR	80.1%	91.8%	89.5%	71.7%
	LBR-SPR (Hard)	81.2%	97.3%	89.8%	72.6%
	LBR-SPR (Linear)	81.3%	97.4%	90.6%	72.5%
	LBR-SPR (Mixture)	80.9%	97.2%	90.7%	72.6%

from $\mu(t) + 4\sigma(t)$ to $\mu(t) + \sigma(t)$ as the number of training iteration t increases:

$$\lambda = \max\{\mu(t) + (4 - t \cdot r) \cdot \sigma(t), \mu(t) + \sigma(t)\} \quad (3)$$

Likewise, we can also illustrate how hard/linear SP regularizer exclude anomalies by Eq. (2): When the RL of a STC $L_i(\boldsymbol{\theta})$ is larger than the threshold λ, the STC's weight v_i will be directly set as 0 and it will be dropped at the current training iteration for both hard and linear regularizer, which is similar to the mechanism of mixture SP regularizer. However, when the RL of a STC is smaller than the threshold λ, hard SP regularizer will directly set its sample weight to be 1, which is an optimistic strategy. On the contrary, linear SP regularizer adopts a pessimistic strategy that simply lowers the weights of all remaining samples according to their RL, while only few samples with very small RL are assigned with a weight close to 1. As a comparison, mixture SP regularizer used in the manuscript adopts an intermediate strategy that determines sample weights by dividing samples into certain normality/anomalies and uncertain samples. In Table 3, we evaluate the performance of hard/linear SP regularizer and compare them with original LBR and mixture SP regularizer: It is observed that all three regularizers are able to produce evident performance improvement when compared with LBR. Among three SP regularizers, we note that mixture SP regularizer tends to be the better performer on Avenue and ShanghaiTech, while linear and hard SP regularizer is generally better on UCSDped1 and UCSDped2 dataset. In our paper, we simply choose mixture SP regularizer as a relatively moderate strategy to exclude anomalies, while the results in Table 3 suggest that other forms of SP regularizer are also readily applicable.

6. Full mode for UVAD Evaluation

In addition to the two UVAD setups (i.e., partial mode and merge mode) reported in the manuscript, it is also interesting to explore another natural setup named ***full mode***: The original training and testing set of the original VAD dataset are merged into one unlabeled set, which is used for both training and evaluation. In other words, partial mode and merge mode perform evaluation on the original testing set, while the full mode performs evaluation on the merged unlabeled set. As the partial and merge mode do, all labels are strictly not used in learning. We show the VAD performance of our solution under the full mode in Table 4, and compare the UVAD performance under the partial and merge mode. As the results suggest in the table, evaluations under a full mode constantly report a more optimistic performance than the partial and merge mode. To facilitate comparison with existing methods, we only report results of the partial and merge mode in the manuscript.

Table 4. Frame-level AUROC of LBR-SPR under different modes. LBR-SPR* indicates the performance of LBR-SPR under partial mode, and LBR-SPR$^+$ indicates the performance under merge mode, while LBR-SPR$^\#$ indicates the performance under full mode. ME denotes motion enhancement.

Method	Ped1	Ped2	Avenue
LBR-SPR* (w/o ME)	81.1%	93.3%	88.5%
LBR-SPR* (w/ ME)	81.1%	95.7%	92.8%
LBR-SPR$^+$ (w/o ME)	79.4%	97.0%	89.7%
LBR-SPR$^+$ (w/ ME)	80.9%	97.2%	90.7%
LBR-SPR$^\#$ (w/o ME)	82.3%	97.9%	92.7%
LBR-SPR$^\#$ (w/ ME)	83.8%	98.4%	93.8%

7. More Discussion

Computational Cost. By a python implementation on a PC with Intel i9-10900X CPU and NVIDIA 2080Ti GPUs, LBR-SPR takes an average 0.033/0.041/0.067/0.11s per frame to extract STCs and infer anomaly scores on UCSDped1/UCSDped2/Avenue/ShanghaiTech respectively. It should be noted that our UVAD solution is an offline transductive approach, so the computational cost is only shown as a reference.

Limitation. The proposed solution is merely tested as a transductive approach so far, which detects anomalies from all given unlabeled videos. It is unclear if it can work as an inductive solution that deals with newly incoming videos. Then, formation of normality advantage requires that normal events outnumber anomalies in videos, which usually but not always holds. Besides, non-generative paradigms like contrastive learning [2] are not studied in this paper.

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 2

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning

of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 4

[3] Zhiwen Fang, Jiafei Liang, Joey Tianyi Zhou, Yang Xiao, and F. Yang. Anomaly detection with bidirectional consistency in videos. *IEEE transactions on neural networks and learning systems*, PP, 2020. 2

[4] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016. 3

[5] Ryota Hinami, Tao Mei, and Shin'ichi Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3619–3627, 2017. 2

[6] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 2

[7] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. *Proceedings of the 22nd ACM international conference on Multimedia*, 2014. 3

[8] M. Pawan Kumar, Ben Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NIPS*, 2010. 3

[9] Tsung Yi Lin. *Microsoft COCO: Common objects in context*. Springer International Publishing, 2014. 2

[10] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13588–13597, October 2021. 2

[11] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981. IEEE, 2010. 2

[12] Mahdyar Ravanbakhsh, Moin Nabi, Hossein Mousavi, E. Sangineto, and N. Sebe. Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1689–1698, 2018. 2

[13] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1577–1581. IEEE, 2017. 2, 3

[14] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018. 2

[15] Hanh TM Tran and David Hogg. Anomaly detection using a convolutional winner-take-all autoencoder. In *Proceedings of the British Machine Vision Conference 2017*. British Machine Vision Association, 2017. 2, 3

[16] Radu Tudor Ionescu, Sorina Smeureanu, Bogdan Alexe, and Marius Popescu. Unmasking the abnormal events in video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2895–2903, 2017. 2, 3

[17] Hung Thanh Vu, Tu Dinh Nguyen, Trung Le, Wei Luo, and Dinh Q. Phung. Robust anomaly detection in videos using multilevel representations. In *AAAI*, 2019. 2

[18] P. Wu, Jing Liu, and Fang Shen. A deep one-class neural network for anomalous event detection in complex scenes. *IEEE Transactions on Neural Networks and Learning Systems*, 31:2609–2622, 2020. 2, 3

[19] Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117–127, 2017. 2, 3

[20] Shiyang Yan, Jeremy S Smith, Wenjin Lu, and Bailing Zhang. Abnormal event detection from videos using a two-stream recurrent variational autoencoder. *IEEE Transactions on Cognitive and Developmental Systems*, 2018. 3

[21] Muchao Ye, Xiaojiang Peng, Weihao Gan, Wei Wu, and Yu Qiao. Anopcn: Video anomaly detection via deep predictive coding network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1805–1813. ACM, 2019. 3

[22] Guang Yu, Siqi Wang, Zhiping Cai, En Zhu, Chuanfu Xu, Jianping Yin, and Marius Kloft. Cloze test helps: Effective video anomaly detection via learning to complete video events. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 583–591, 2020. 1, 3

[23] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941. ACM, 2017. 3

[24] Joey Tianyi Zhou, Jiawei Du, Hongyuan Zhu, Xi Peng, Yong Liu, and Rick Siow Mong Goh. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, 2019. 2, 3