Video Anomaly Detection by Solving Decoupled Spatio-Temporal Jigsaw Puzzles

Guodong Wang[1,2], Yunhong Wang[2], Jie Qin[3], Dongming Zhang[4],
Xiuguo Bao[4], and Di Huang[1,2]*

[1] SKLSDE, Beihang University, Beijing, China
[2] SCSE, Beihang University, Beijing, China
[3] CCST, NUAA, Nanjing, China
[4] CNCERT/CC, Beijing, China
{wanggd,yhwang,dhuang}@buaa.edu.cn, qinjiebuaa@gmail.com,
zhdm@cert.org.cn, baoxiuguo@139.com

Abstract. Video Anomaly Detection (VAD) is an important topic in computer vision. Motivated by the recent advances in self-supervised learning, this paper addresses VAD by solving an intuitive yet challenging pretext task, *i.e.*, spatio-temporal jigsaw puzzles, which is cast as a multi-label fine-grained classification problem. Our method exhibits several advantages over existing works: 1) the spatio-temporal jigsaw puzzles are decoupled in terms of spatial and temporal dimensions, responsible for capturing highly discriminative appearance and motion features, respectively; 2) full permutations are used to provide abundant jigsaw puzzles covering various difficulty levels, allowing the network to distinguish subtle spatio-temporal differences between normal and abnormal events; and 3) the pretext task is tackled in an end-to-end manner without relying on any pre-trained models. Our method outperforms state-of-the-art counterparts on three public benchmarks. Especially on ShanghaiTech Campus, the result is superior to reconstruction and prediction-based methods by a large margin.

Keywords: video anomaly detection; spatio-temporal jigsaw puzzles; multi-label classification

1 Introduction

Video anomaly detection (VAD) refers to the task of detecting unexpected events that deviate from the normal patterns of familiar ones. Recently, it has become a very important task in the community of computer vision and pattern recognition with the exponential increase of video data captured from various scenarios. VAD is rather challenging as abnormal events are infrequent in real world and unbounded in category, jointly making typical supervised methods inapplicable due to the unavailability of balanced normal and abnormal samples for training. Therefore, VAD is generally performed in a one-class learning manner where only normal data are given [13, 29, 30].

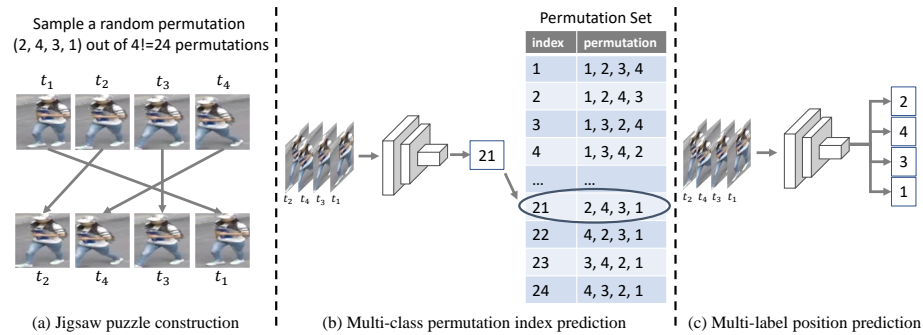* Corresponding author (ORCID: 0000-0002-2412-9330).

Fig. 1: Multi-class index classification *vs.* multi-label position classification. (a) Jigsaw puzzle construction. We permute the original sequence based on a randomly selected permutation from all possible ones. (b) Multi-class permutation index prediction. Traditional methods [24,26,41] take a permutation as one class out of 4!=24 classes. (c) Multi-label permutation position prediction (**ours**). We directly output multiple predictions, indicating the absolute position in the original sequence for each frame.

In this regime, a series of VAD approaches [12, 16, 27, 29, 34, 40, 56, 59] have been proposed, among which reconstruction and prediction based methods are two representative paradigms in the context of deep learning. Reconstruction based methods [16, 40] build models, *e.g.*, autoencoders and generative adversarial networks (GAN), to recover input frames, and examples with high reconstruction errors are identified as anomalies at test time. In consideration of the temporal coherence, prediction based methods render missing frames *e.g.*, middle frames [59] or future frames [29, 56], according to motion continuity. The difference between the predicted frame and its corresponding ground-truth suggests the probability of anomaly occurring.

The two types of methods above report promising performance; however, as stated in [16, 39, 62], they aim at high-quality pixel generation, and even though the networks are only trained to perfectly match normal examples, their inherent generalization abilities still make the anomalies well reconstructed or predicted, especially for static objects, *e.g.*, a stopped car in a pedestrian area. To address this, some follow-up studies attempt to boost the accuracy through incorporating memory modules [16, 43], modeling optical flows [29], redesigning specific architectures [56], *etc.*

More recently, self-supervised learning has opened another avenue for VAD with significantly improved results. Different from the unsupervised generative solutions, self-supervised learning based methods explore supervisory signals for learning representations from unlabeled data [13, 53], and current investigations mainly differ in the design of pretext tasks. Wang *et al.* [53] propose an instance discrimination task to establish subcategories of normality as clusters and examples far away from the cluster centers are determined as anomalies. Georgescu

et al. [13] deliver an advancement by a model jointly considering multiple pretext tasks including discriminating arrow of time and motion regularity, middle frame reconstruction and knowledge distillation. Nevertheless, their pretext tasks are basically defined as binary classification problems, making them not so competent at learning highly discriminative features to distinguish *subtle spatio-temporal differences* between normal and abnormal events. Additionally, these methods [13,53] depend on the networks pre-trained on large-scale datasets, *e.g.*, ImageNet [47] and Kinetics-400 [23].

To circumvent the shortcomings aforementioned, in this paper, we propose a simple yet effective self-supervised learning method for VAD, through tackling an intuitive but challenging pretext task, *i.e.*, spatio-temporal jigsaw puzzles. We hypothesize that successfully solving such puzzles requires the network to understand the very detailed spatial and temporal coherence of video frames by learning powerful spatio-temporal representations, which are critical to VAD. To this end, we take into account full possible permutations, rather than a subset produced by a heuristic permutation selection algorithm [41], to increase the difficulty of jigsaw puzzles with the aim of offering fine-grained supervisory signals for discriminative features. Based on the observation that anomalous events usually involve abnormal appearances and abnormal motions, we decouple spatio-temporal jigsaw puzzles in terms of spatial and temporal dimensions, responsible for modeling appearance and motion patterns, respectively, which technically facilitates optimization compared to solving 3D jigsaw puzzles [2]. To be specific, we first randomly select a permutation from $n!$ possible permutations, where n is the number of elements in the sequence. With this permutation, we then spatially shuffle patches within frames to construct spatial jigsaw puzzles or temporally shuffle a sequence of consecutive frames to build temporal ones. The training objective is to recover an original sequence from its spatially or temporally permuted version. Unlike existing methods for learning general visual representations [24,26,41] which treat jigsaw puzzle solving as a multi-class classification problem where each permutation corresponds to a class (Figure 1 (b)), we cast it as a multi-label learning problem (Figure 1 (c)), allowing the method to be extendable to more advanced jigsaw puzzles with more pieces and free from significantly increased memory consumption. During inference, the confidence of the prediction with respect to unshuffled frames or images serves as the regularity score for anomaly detection. Abnormal events are expected to have lower confidence scores because they are unseen in training.

Compared to prior work on self-supervised VAD [13,53], the advantages of our method are three-fold. **First**, we dramatically simplify the self-supervised learning framework by solving only a single pretext task, which is decoupled into the spatial and temporal jigsaw puzzles, corresponding to modeling normal appearance and motion patterns, respectively. **Second**, full possible permutations are employed to produce large-scale learning samples of a high diversity, allowing the network to capture subtle spatio-temporal anomalies from the pretext task. To ensure computational efficiency, we formulate puzzle solving as a multi-label learning problem, accommodating a factorial of number of variations.

Third, our method is free from any pre-trained networks, because solving the challenging pretext itself helps to learn rich and discriminative spatio-temporal representations. It achieves state-of-the-art results on three public benchmarks, especially on the ShanghaiTech Campus dataset [35].

2   Related Work

Video anomaly detection. While early studies [1, 3, 9] advocate manually-designed appearance and motion features for VAD, recent methods leverage the powerful representation capabilities of deep neural networks to automatically learn features from video events and deliver better performance. Most VAD methods follow the way of per-pixel generation, with reconstruction based and prediction based ones being the two important lines. Reconstruction based methods [11,18,34] learn to recover input frames or clips, while prediction based methods learn to predict missing frames, such as future frame prediction [12, 29, 33] or middle frame completion [27,59]. The combination of reconstruction and prediction as a hybrid solution is also explored in [38, 58, 61]. These methods aim at high-quality pixel generation during training and examples with large reconstruction or prediction errors are identified as anomalies. However, these networks often exhibit strong generalization abilities on anomalies (even though they are unseen in training), leading to decent reconstruction or prediction quality. The use of memory mechanisms [16, 43] or multi-modal data (*e.g.*, optical flows [30, 40] and RGB differences [6]) suppresses the generalization ability to some extent but the improvement is far from perfect given the additional computation and memory consumption.

Self-supervised learning. Self-supervised learning (SSL) is a generic learning framework which seeks supervisory signals from data only. It can be broadly categorized into constructing pretext tasks and conducting constrastive learning. **Pretext tasks.** For images, pretext tasks typically include solving jigsaw puzzles [41], coloring images [60], and predicting relative patches [10] or image rotations [15], *etc.* For videos, a series of methods additionally exploit temporal information exclusive to videos based on verifying correct frame order [37], sorting frame order [26] or clip order [57], predicting playback speed [5] or arrow of time [45, 54], *etc.* Among the pretext tasks, jigsaw puzzle is widely explored and proves effective in learning visual representation, but related methods fail to leverage all possible permutations, which scales factorially with the input length. Cruz *et al.* [48] avoid such a factorial complexity by directly predicting the permutation matrix that shuffles the original data. **Contrastive learning.** It is another prevalent self-supervised learning paradigm in which each instance is regarded as a category. Motivated by the success of self-supervised image representation learning such as SimCLR [8] and MoCo [19], many extensions [31,42] of constrastive learning are proposed to adapt image-based methods to the video domain.

SSL in VAD. While these self-supervised methods prove very effective in generic representation learning, benefiting bundles of downstream recognition
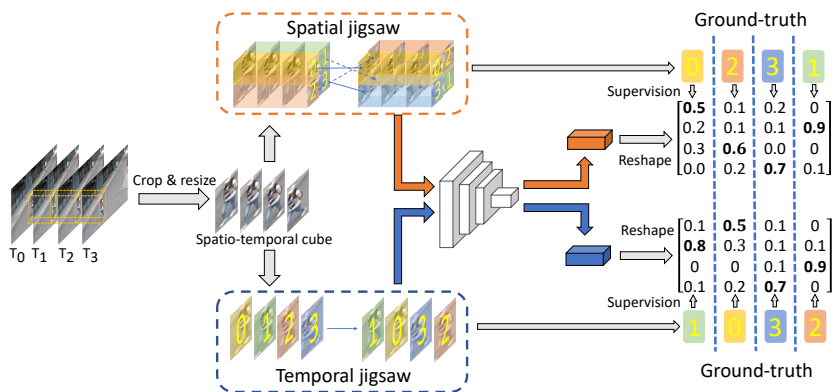
Fig. 2: Method overview. We devise a pretext task including temporal and spatial jigsaw puzzles, for self-supervised learning spatio-temporal representations. Based on the object-centric spatio-temporal cubes, we create jigsaw puzzles by performing temporal and spatial shuffling. The network comprises a shared 3D convolution backbone followed by two disjoint heads to predict the permutation used for shuffling frames in time and patches in space, respectively. Each column of a matrix denotes the prediction of an entry in the permutation.

and detection tasks [8, 19, 31, 42], the efforts that exploit SSL for VAD are very few. SSL based VAD methods capture spatio-temporal representations by either conducting constrastive learning or solving pretext tasks, diverging from per-pixel reconstruction or prediction based methods. Wang *et al.* [53] learn spatio-temporal representations via a contrastive learning framework with a cluster attention mechanism. However, it requires a large number of training samples and customized data augmentation strategies. Georgescu *et al.* [13] train a 3D convolutional neural network jointly on three self-supervised proxy tasks and knowledge distillation for VAD. These proxy tasks are easy to solve, preventing the network from learning highly discriminative representations for VAD.

In this work, we design a more challenging pretext task, *i.e.*, solving spatio-temporal jigsaw puzzles. Though solving jigsaw puzzles as a pretext task has been investigated for SSL by shuffling spatial layout [41], temporal order [26] or their combination [24], they mostly formulate it as a multi-class classification task, therein each type of permutation corresponds to one class. The representations learned from the pretext tasks prove very effective evaluated in a series of downstream tasks, *e.g.*, image retrieval and action recognition; however, its potential on VAD remains unexplored. A straightforward solution is directly applying these methods [26, 41] to learn representations of normal events for VAD. Nevertheless, this simple adaptation is sub-optimal since they only focus on modeling either appearances or motions while abnormal appearances and abnormal motions intertwine with each other in anomalous events in videos. Kim *et al.* [24] manage to solve space-time cubic puzzles, however, the multi-class formulation

restricts itself to more advanced jigsaw puzzles (in fact they only leverage four-piece jigsaw puzzles), leading to inferior performance, shown in Table 2. Though Ahsan *et al.* [2] consider directly solving 3D spatio-temporal jigsaw puzzles, we empirically find that its performance is not as good as expected due to the extreme difficulty of solving 3D jigsaw puzzles. For example, a cube compressing 7 frames with 3×3 grid results in $7! \times (3 \times 3)! = 1,828,915,200$ possibles permutations. Therefore, we decompose the 3D spatio-temporal jigsaw puzzles into spatial and temporal jigsaw puzzles, corresponding to learning appearance and motion patterns, respectively.

3   Method

3.1   Overview

Figure 2 shows the pipeline of the proposed method, in which a sequence of four frames is used as an example for easy illustration. The method contains three steps: object-centric cube extraction, puzzle construction, and puzzle solving. We first employ an off-the-shelf object detector [46] to extract all objects in the frames and stack the objects along the time dimension to construct object-centric cubes. For each cube, we further apply spatial or temporal shuffling to construct the corresponding spatial or temporal jigsaw puzzle. Finally, a convolutional neural network, acting as a jigsaw solver, attempts to recover the original sequence from its spatially or temporally permuted version. The proposed method is equivalent to solving a multi-label classification problem and is trained in an end-to-end manner.

It is noteworthy that we **do not** use any optical flows or pre-trained models (except for the object detector). The spatial and temporal permutations are only applied in training, allowing fast inference with a single forward pass.

3.2   Fine-grained Decoupled Jigsaw Puzzles

In self-supervised learning, it is crucial to prepare neither ambiguous nor easy self-labeled data [41]. Based on the observation [26, 41] that networks can learn richer spatio-temporal representations from a more difficult pretext, we introduce full permutations for fine-grained jigsaw puzzle construction with the aim of capturing subtle spatial and temporal differences.

We first extract a large number of objects of interest by applying a YOLOv3 detector [46] pre-trained on MS-COCO [28] frame by frame, therein we only keep the localization information and discard the classification labels. For each object detected in the frame i, we construct an object-centric spatio-temporal cube by simply stacking patches cropped from its temporally adjacent frames $\{i - t, ..., i - 1, i, i + 1, ...i + t\}$ using the same bounding box and location. We rescale all the extracted patches into a fixed size, *e.g.*, 64×64. Based on the extracted object-centric spatio-temporal cubes, we prepare training samples by constructing spatial or temporal jigsaw puzzles.

Spatial jigsaw. Following [41], for each frame, we start by decomposing it into $n \times n$ equal-sized patches which are then randomly shuffled. We make all the frames in the cube share the same permutation meanwhile keep them in the chronological order. Different from [41] that separately passes each patch into the network, we directly take as input the frames after spatial shuffling, which are of the same size with the original frame, *i.e.*, 64×64 in our setting.

Temporal jigsaw. To construct temporal jigsaw puzzles, we shuffle a sequence of l frames without disorganizing the spatial content. Jenni *et al.* [22] reveal that the most effective pretext tasks for powerful video representation learning are those that can be solved by observing the largest number of frames. For instance, motion irregularity [13] can be easily detected by just comparing the first two frames, in contrast to observing the total number of frames to solve our temporal jigsaw puzzles, which is crucial for learning more discriminative representations of motion patterns. Note that we do not temporally shuffle the frame sequence containing only static contents since it is impossible to infer its temporal order by simply observing visual cues.

3.3   Multi-label Supervision

Our jigsaw solving task is essentially a permutation prediction problem. Recall that, to make the task more challenging for learning discriminative representations, we employ the full permutations to produce fine-grained jigsaw puzzles. Different from typical methods [24, 26, 41] that formulate jigsaw puzzle solving as multi-class classification, therein each permutation is a class, we cast jigsaw puzzle solving as a multi-label classification problem and attempt to directly predict the absolute position of each frame or the location of each patch. For each frame in the temporally shuffled sequence, we predict the correct position in the original sequence, while for each patch in the spatially shuffled frame, we predict the correct location in the original splitting grid. The strategy reduces the complexity $\mathcal{O}(l!)$ to $\mathcal{O}(l^2)$, and it can thus be easily extended to input frames of longer sequences or finer grid-splits with negligible memory consumption.

We adopt the mixed training strategy where a training mini-batch consists of two disjoint sets: Q_s and Q_t, denoting the sets of spatial and temporal jigsaw puzzles, respectively. Thus, the mini-batch has a total of $|Q_t| + |Q_s|$ samples. It is worth noting that the two solvers (heads) are only responsible for their own puzzle types, *i.e.*, we do not rely on the temporal solver to deal with spatial jigsaw puzzles to avoid ambiguity and vice versa. Algorithm 1 provides more details for constructing puzzles in mini-batches.

We optimize the network using the cross-entropy (CE) loss. For a jigsaw puzzle p, its loss is computed as Eq. (1).

$$L_p = \begin{cases} \frac{1}{l} \sum_{i=1}^{l} CE(t_i, \hat{t}_i), & p \in Q_t \\ \frac{1}{n^2} \sum_{j=1}^{n^2} CE(s_j, \hat{s}_j), & p \in Q_s \end{cases}, \tag{1}$$

where t_i and $\hat{t}_i$ are the ground-truth and predicted positions of a frame in the original sequence, respectively, and s_j and $\hat{s}_i$ are the ground-truth and predicted locations of a patch in the original splitting grid, respectively.

Algorithm 1: Puzzle construction in mini-batches

    Input: object-centric spatio-temporal cubes C, ratio r , frame length l,
            number of patches n^2, threshold ζ.
    Output: sets of jigsaw puzzles: Q_t, Q_s.

1 $Q_t \leftarrow \varnothing, Q_s \leftarrow \varnothing$
2 $P^t \leftarrow$ all permutations $[P_1^t, P_2^t, ..., P_{l!}^t]$
3 $P^s \leftarrow$ all permutations $[P_1^s, P_2^s, ..., P_{(n^2)!}^s]$
4   **for** c in C **do**
5 $p \leftarrow \mathcal{U}_{float}[0, 1]$ `// uniform sampling`
6   **if** $p \leq r$ **then**
7   **if** $p \leq \zeta$ **then**
8 $i \leftarrow 1$
9   **else**
10 $i \leftarrow \mathcal{U}_{int}[1, (n^2)!]$
11   **end**
12 $q \leftarrow$ SpatiallyShuffle(c, P_i^s)
13 $Q_s \leftarrow Q_s \cup \{q\}$
14   **else**
15 $j \leftarrow \mathcal{U}_{int}[1, l!]$
16 $q \leftarrow$ TemporallyShuffle(c, P_j^t)
17 $Q_t \leftarrow Q_t \cup \{q\}$
18   **end**
19   **end**

3.4   VAD Inference

Following the same protocol of object detection in training, for each object in frame i, we construct the corresponding object-centric cube by cropping the bounding boxes from its temporally adjacent frames $\{i-t, ..., i-1, i, i+1, ...i+t\}$. During inference, we reuse the built-in jigsaw solvers to obtain the regularity scores. We pass the object-centric cubes without performing spatial or temporal shuffling and obtain two matrices, M_s and M_t, corresponding to spatial and temporal permutation predictions, respectively.

Intuitively, the diagonal entries of the matrices of normal events are larger than those of abnormal ones, as the network is only trained to recover the original normal sequences. We thus simply take the minimum prediction score of a sequence as its regularity, as in Eq. (3).

$$\begin{cases} r_s = \min(\mathrm{diag}(M_s)) \\ r_t = \min(\mathrm{diag}(M_t)) \end{cases}, \tag{2}$$

where $\mathrm{diag}(\cdot)$ extracts the matrix diagonal, M_s and M_t are predicted by the spatial or temporal jigsaw solver, and r_s and r_t indicate the object-level regularity scores, respectively. We select the minimum score along the diagonal of the matrix as the resulting object-level regularity score, since an example is likely anomalous as long as one frame or patch is wrongly predicted, in accordance with fine-grained multi-label supervision in training. Similarly, we obtain the frame-level regularity score R_s (R_d) by simply selecting the minimum object-level regularity score in the frame. Similar to [13], we also apply a 3D mean filter to create a smooth anomaly score map. Following [29, 56, 58], we normalize the irregularity scores of all frames in each video:

$$\begin{cases} R_s = \dfrac{R_s - \min(R_s)}{\max(R_s) - \min(R_s)} \\ R_t = \dfrac{R_t - \min(R_t)}{\max(R_t) - \min(R_t)} \end{cases}. \tag{3}$$

The final frame-level regularity score R (Eq. (4)) is the weighted average of R_s and R_t, followed by a temporal 1-D Gaussian filter.

$$R = w * R_s + (1 - w) * R_t. \tag{4}$$

4 Experiments

4.1 Datasets

We present the experimental results on three popular benchmarks, namely UCSD Ped2 [36], CUHK Avenue [32], ShanghaiTech Campus [35].

UCSD Ped2 [36]. Ped2 contains 16 training videos and 12 test videos captured by a fixed camera. Example objects are pedestrians, bikes, and vehicles. Each video has a resolution of 240×360 pixels in gray scale.

CUHK Avenue [32]. Avenue consists of 16 training videos and 21 test videos, respectively. It includes a total number of 47 abnormal events with throwing bag and moving toward/away from the camera being example anomalies. Each video has a resolution of 360×640 RGB pixels.

ShanghaiTech Campus (STC) [35]. It contains 330 training videos and 107 test videos covering 13 different scenes, making it more challenging than the other two datasets. Example anomalous events include car invading and person chasing. Each video has a resolution of 480×856 RGB pixels.

4.2   Implementation Details

Since we train our network only on the object-centric cubes, the first stage of our method is object detection. For fair comparison, we follow [13] to adopt the same implementation[5] of YOLOv3 [46] and use the same configurations to filter out the detected objects with low confidence. We set the confidence thresholds to 0.5, 0.8, and 0.8 for Ped2, Avenue and STC, respectively. The confidence thresholds are shared during training/test for each dataset. The input to the network is a tensor of $l \times 64 \times 64 \times 3$ where l is the length of the sequence. The difficulty levels of spatial and temporal jigsaw puzzles are adjusted by varying n and l, respectively. We obtain the optimal results with $l = 9$ on STC and $l = 7$ on Ped2 and Avenue, and $n = 3$ for all the three datasets. We empirically set $r = 0.5$ ($|Q_t| = |Q_s|$) and $w = 0.5$ throughout all the experiments, indicating the equivalent importance of spatial and temporal branches. Considering that Avenue and Ped2 are relatively small compared to the scale of spatial jigsaw puzzles (9!=362,800), we do not perform spatial shuffling with the probability $\zeta = 1e - 4$ (Line 7 in Algorithm 1).

   Our framework is implemented using the PyTorch library [44] and trained in an end-to-end manner. We adopt the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$. The learning rate is 1e-4. We train the network for 100 epochs on Avenue and STC and 50 epochs on UCSD Ped2, and set the batch size as 192.

4.3   Evaluation Metric

Following the widely-adopted evaluation metric used in VAD community [12,16, 29,30,53,59], we report the frame-level area under the curve (AUC) of Receiver Operation Characteristic (ROC) with respect to the ground-truth annotations by varying the threshold. Specifically, we concatenate all the frames in dataset and then compute the overall frame-level AUC, *i.e.*, micro-averaged AUROC [14].

4.4   Experimental Results

Table 1 shows the comparison results with different types of state-of-the-art approaches, where we can observe that our method delivers very impressive performance on all the three benchmarks. On the challenging STC, our method outperforms reconstruction-based, prediction-based, and hybrid methods by significant margins. For example, our method achieves 84.3% while the best accuracy of previous generative methods is 77.7% by CT-D2GAN [56]. This suggests the superiority of self-supervised learning which captures discriminative representations of normal events by solving pretext tasks, bypassing the requirement for per-pixel generation. Additionally, compared to the VAD methods [13, 53] leveraging self-supervised learning, we still achieve the best performance, boosting the second-best method [13] by 1.9%. We attribute it to the design of our challenging pretext task, *i.e.*, solving fine-grained spatio-temporal jigsaw puzzles

---

[5] https://github.com/wizyoung/YOLOv3_TensorFlow

Table 1: Comparison with state-of-the-art methods in terms of micro-AUROC (%). The best and second-best results are bold and underlined, respectively. * denotes that micro-AUROC is reported. SSL is short for self-supervised learning.

Type	Method	Ped2	Avenue	STC
reconstruction	Conv-AE [18]	90.0	70.2	-
	StackRNN [35]	92.2	81.7	68.0
	Mem-AE [16]	94.1	83.3	71.2
	AM-Corr [40]	96.2	86.9	-
	MNAD-Recon. [43]	90.2	82.8	69.8
	ClusterAE [6]	96.5	86.0	73.3
	VEC [59]	97.3	90.2	74.8
	LNRA (Patch based) [4]	94.8	84.9	72.5
	LNRA (Skip frame based) [4]	96.5	84.7	76.0
	I3D-Recons.	-	69.3 -	69.4
prediction	Frame-Pred. [29]	95.4	85.1	72.8
	BMAN [27]	96.6	90.0	76.2
	Multipath-Pred. [52]	96.3	88.3	76.6
	MNAD-Pred. [43]	97.0	88.5	70.5
	CT-D2GAN [56]	97.2	85.9	77.7
	Bi-Prediction [7]	96.6	87.8	-
hybrid	ST-CAE [61]	91.2	80.9	-
	MPED-RNN [38]	-	-	73.4
	AnoPCN [58]	96.8	86.2	73.6
	IntegradAE [50]	96.3	85.1	73.0
	HF^2-VAD [30]	**99.3**	91.1	76.2
others	SCL [32]	-	80.9	-
	DeepOC [55]	96.9	86.6	-
	CAE-SVM* [21]	94.3	87.4	78.7
	Scene-Aware [49]	-	89.6	74.7
SSL	CAC [53]	-	87.0	79.3
	SS-MTL* [13]	97.5	<u>91.5</u>	<u>82.4</u>
	Ours	<u>99.0</u>	**92.2**	**84.3**

with full permutations, which helps to learn discriminative representations. Note that we only use the VAD training set to train our network and do not use an extra model for either knowledge distillation [13] or transfer learning [53]. On Avenue and UCSD Ped2, we also deliver very competitive performance, indicating that our method is robust to datasets of different scales.

5   Ablation Study

To understand the factors that contribute to the anomaly detection performance, we conduct ablation studies on STC and Avenue considering four key factors that control the puzzle difficulty: a) number of permutations; b) number of

frames/patches; c) types of puzzles; and d) other pretexts beyond jigsaw solving. We also discuss the reliance on the object detector.

Table 2: Results of various numbers of permutations on STC and Avenue in terms of AUROC (%). T and S represent the number of permutations for temporal and spatial jigsaw puzzle construction, respectively. Here, $l = 7$ and $n^2 = 9$.

Exp. ID	Method	T	S	Avenue	STC
A1	Baseline (Multi-class)	504	504	84.6	76.8
A2	Baseline (Multi-class)	5040	504	85.3	77.6
A3	Baseline (Multi-class)	504	5040	87.0	78.1
A4	Baseline (Multi-class)	5040	5040	87.5	78.5
A5	Ours (Multi-label)	504	504	87.1	79.7
A6	Ours (Multi-label)	5040	504	87.5	81.2
A7	Ours (Multi-label)	504	5040	88.6	79.8
A8	Ours (Multi-label)	5040	5040	89.5	82.0
A9	Ours (Multi-label)	5040	362880	**92.2**	**83.2**

Number of permutations. We constrain the number of permutations used to construct puzzles by selecting the subsets of full permutations using a Hamming distance based selection algorithm [41]. We first build a baseline that follows the typical solution for solving puzzles, which considers a permutation as a class. During inference, the regularity scores are the probabilities of the spatio-temporal cubes not being spatially or temporally permuted. From Table 2, both the baseline method and our method achieve improved results with a large number of permutations for both spatial and temporal puzzles, since the networks need to capture more discriminative representations to perceive subtle differences among jigsaw puzzles. Moreover, our method consistently outperforms the baseline for the same number of permutations. One possible reason is that the baseline attempts to discriminate jigsaw puzzles by different permutations, while we aim to predict the correct position of each frame/patch in the permutation in a more detailed way. Moreover, for advanced puzzles with more pieces, the baseline model fails due to memory limitation. In contrast, thanks to the multi-label classification formulation, our method can handle finer-grained puzzles with full permutations in a memory-friendly way, achieving the best (A9).

Number of frames/patches. With full permutations considered for puzzle construction, we next examine the effects of the number of frames (l) in the temporal dimension and the number of patches (n^2) in the spatial dimension. We do not try a larger l ($l > 9$) as the object would go beyond the boundary of the spatio-temporal cube. From Table 3, we can observe a trend of performance improvement when we increase l and n^2 in a certain range. The observation is consistent with human beings who need more efforts to solve puzzles with more pieces. However, when we increase l and n^2 further, the performance deteriorates. The reason lies in that the network is difficult to optimize especially on the spatial

Table 3: Results of different numbers of frames/patches on STC and Avenue in terms of AUROC (%). l and n^2 denote the number of frames in an object-centric cube and the number of patches in the frames, respectively.

Exp. ID	l	n^2	Avenue	STC
B1	5	4	89.7	79.3
B2	7	4	90.2	80.2
B3	7	9	**92.2**	83.2
B4	9	4	88.6	81.1
B5	9	9	89.2	**84.3**
B6	9	16	87.9	80.4

jigsaw puzzles, as each patch is very small (16×16 pixels for $n^2 = 16$ in our setting) and thus causes ambiguity.

Types of puzzles. Finally, we investigate the effects of solving spatial and temporal jigsaw puzzles for VAD. To this end, we design four alternative configurations based on when and which jigsaw puzzles are activated. Here, we set $l = 7$ and $l = 9$ for Avenue and STC, respectively; $n^2 = 9$ for both. Our method benefits more from solving spatial and temporal jigsaw puzzles in the training phase. For example, C3 indicates simultaneously solving spatial and temporal jigsaw puzzles during training, while C1 represents solving temporal jigsaw puzzles only. We activate the temporal solver only during testing for C1 and C3. In other words, the only difference between C1 and C3 is the training goal, *i.e.*, multi-task *vs.* single-task. Our method clearly benefits from multi-task learning and achieves better performance 82.7% *vs.* 78.6% on STC. However, when only one type of puzzle is activated either during training or testing, the results are always worse than our complete version, namely C5. The observations are intuitive that anomalous events are caused by abnormal appearances and/or motions. Therefore, it is beneficial to include both types of jigsaw puzzles to detect both types of anomalies.

Other pretexts beyond jigsaw solving. We design other alternative pretext tasks considering the spatial dimension (*e.g.*, rotation prediction [25] and translation prediction [20]) and the temporal dimension (*e.g.*, arrow of time prediction [54] and temporal order verification [37]). Our method achieves the best performance in Table 4. It gives evidence that a proper design of the pretext task enabling fine-grained discrimination is essential for VAD. Compared to other pretexts, ours sets a more challenging task which requires the model to perceive every patch within a frame and every frame within a clip.

Object detector. Although we mainly focus on object-level anomaly detection, our method can also be applied at frame level. To this end, we remove the object detector and report the results on the RetroTrucks dataset [17]. For fair comparison with [17], we train an I3D [51] network to predict the absolute position of the original sequence, since we observe that activating the temporal branch only is sufficient. Our method achieves the best performance in Table 6, even outperforming [17] that incorporates object interaction reasoning.

Table 4: Results of different pretexts for VAD on STC in terms of AUROC (%).

Exp. ID	Spatial	Temporal	STC
D1	Rotation	Arrow of time	72.9
D2	Rotation	Temporal order verification	74.8
D3	Translation	Arrow of time	73.0
D4	Translation	Temporal order verification	75.6
D5	Translation	Jigsaw (ours)	81.1
D6	Jigsaw (ours)	Temporal order verification	78.3
D7	Jigsaw (ours)	Jigsaw (ours)	**84.3**

Table 5: Results of different jigsaw puzzles. T and S are short for "temporal" and "spatial", respectively.

Exp. ID	Train		Test		Avenue	STC
	T	S	T	S		
C1	✓	-	✓	-	78.9	78.6
C2	-	✓	-	✓	86.7	76.0
C3	✓	✓	✓		86.9	82.7
C4	✓	✓		✓	89.0	79.8
C5	✓	✓	✓	✓	**92.2**	**84.3**

Table 6: Results on RetroTrucks in terms of AUROC (%).

Method	RetroTrucks
Frame-Pred. [29]	60.6
Mem-AE [16]	63.6
I3D [17]	71.2
I3D + GCN [17]	71.5
Ours	**72.8**

6   Conclusion

In this work, we present a simple yet effective self-supervised learning framework for VAD through solving a challenging pretext task, *i.e.*, spatio-temporal jigsaw puzzles, which are decoupled into spatial and temporal jigsaw puzzles for easy optimization. We emphasize that a challenging pretext task is key to learning discriminative spatio-temporal representations. To this end, we perform full permutations to generate a rich set of spatial and temporal jigsaw puzzles with varying degrees of difficulty, which allows the network to discriminate subtle spatio-temporal differences between normal and abnormal events. We reformulate the pretext task as a multi-label fine-grained classification problem, which is addressed in an efficient and end-to-end manner. Experiments show that our method achieves state-of-the-art on three popular benchmarks.

Acknowledgment

References

1. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. IEEE TPAMI **30**(3), 555–560 (2008)
2. Ahsan, U., Madhok, R., Essa, I.: Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In: WACV (2019)
3. Antić, B., Ommer, B.: Video parsing for abnormality detection. In: ICCV (2011)
4. Astrid, M., Zaheer, M.Z., Lee, J.Y., Lee, S.I.: Learning not to reconstruct anomalies. In: BMVC (2021)
5. Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W.T., Rubinstein, M., Irani, M., Dekel, T.: Speednet: Learning the speediness in videos. In: CVPR (2020)
6. Chang, Y., Tu, Z., Xie, W., Yuan, J.: Clustering driven deep autoencoder for video anomaly detection. In: ECCV (2020)
7. Chen, D., Wang, P., Yue, L., Zhang, Y., Jia, T.: Anomaly detection in surveillance video based on bidirectional prediction. IVC **98**, 103915 (2020)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
9. Cong, Y., Yuan, J., Liu, J.: Abnormal event detection in crowded scenes using sparse representation. PR **46**(7), 1851–1864 (2013)
10. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015)
11. Fan, Y., Wen, G., Li, D., Qiu, S., Levine, M.D., Xiao, F.: Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. CVIU **195**, 102920 (2020)
12. Feng, X., Song, D., Chen, Y., Chen, Z., Ni, J., Chen, H.: Convolutional transformer based dual discriminator general adversarial networks for video anomaly detection. In: ACM MM (2021)
13. Georgescu, M.I., Barbalau, A., Ionescu, R.T., Khan, F.S., Popescu, M., Shah, M.: Anomaly detection in video via self-supervised and multi-task learning. In: CVPR (2021)
14. Georgescu, M.I., Ionescu, R.T., Khan, F.S., Popescu, M., Shah, M.: A background-agnostic framework with adversarial training for abnormal event detection in video. arXiv preprint arXiv:2008.12328 (2020)
15. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018)
16. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: ICCV (2019)
17. Haresh, S., Kumar, S., Zia, M.Z., Tran, Q.H.: Towards anomaly detection in dashcam videos. In: IV (2020)
18. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: CVPR (2016)
19. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
20. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. In: NeurIPS (2019)
21. Ionescu, R.T., Khan, F.S., Georgescu, M.I., Shao, L.: Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In: CVPR (2019)

22. Jenni, S., Meishvili, G., Favaro, P.: Video representation learning by recognizing temporal transformations. In: ECCV (2020)
23. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
24. Kim, D., Cho, D., Kweon, I.S.: Self-supervised video representation learning with space-time cubic puzzles. In: AAAI (2019)
25. Komodakis, N., Gidaris, S.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018)
26. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV (2017)
27. Lee, S., Kim, H.G., Ro, Y.M.: Bman: bidirectional multi-scale aggregation networks for abnormal event detection. IEEE TIP **29**, 2395–2408 (2019)
28. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
29. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection– a new baseline. In: CVPR (2018)
30. Liu, Z., Nie, Y., Long, C., Zhang, Q., Li, G.: A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In: ICCV (2021)
31. Lorre, G., Rabarisoa, J., Orcesi, A., Ainouz, S., Canu, S.: Temporal contrastive pretraining for video action recognition. In: WACV (2020)
32. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: ICCV (2013)
33. Lu, Y., Kumar, K.M., shahabeddin Nabavi, S., Wang, Y.: Future frame prediction using convolutional vrnn for anomaly detection. In: AVSS (2019)
34. Luo, W., Liu, W., Gao, S.: Remembering history with convolutional lstm for anomaly detection. In: ICME (2017)
35. Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked rnn framework. In: ICCV (2017)
36. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: CVPR (2010)
37. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: ECCV (2016)
38. Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., Venkatesh, S.: Learning regularity in skeleton trajectories for anomaly detection in videos. In: CVPR (2019)
39. Munawar, A., Vinayavekhin, P., De Magistris, G.: Limiting the reconstruction capability of generative neural network using negative learning. In: MLSP (2017)
40. Nguyen, T.N., Meunier, J.: Anomaly detection in video sequence with appearance-motion correspondence. In: ICCV (2019)
41. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016)
42. Pan, T., Song, Y., Yang, T., Jiang, W., Liu, W.: Videomoco: Contrastive video representation learning with temporally adversarial examples. In: CVPR (2021)
43. Park, H., Noh, J., Ham, B.: Learning memory-guided normality for anomaly detection. In: CVPR (2020)
44. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
45. Pickup, L.C., Pan, Z., Wei, D., Shih, Y., Zhang, C., Zisserman, A., Scholkopf, B., Freeman, W.T.: Seeing the arrow of time. In: CVPR (2014)

46. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
47. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
48. Santa Cruz, R., Fernando, B., Cherian, A., Gould, S.: Visual permutation learning. IEEE TPAMI **41**(12), 3100–3114 (2018)
49. Sun, C., Jia, Y., Hu, Y., Wu, Y.: Scene-aware context reasoning for unsupervised abnormal event detection in videos. In: ACM MM (2020)
50. Tang, Y., Zhao, L., Zhang, S., Gong, C., Li, G., Yang, J.: Integrating prediction and reconstruction for anomaly detection. PRL **129**, 123–130 (2020)
51. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
52. Wang, X., Che, Z., Jiang, B., Xiao, N., Yang, K., Tang, J., Ye, J., Wang, J., Qi, Q.: Robust unsupervised video anomaly detection by multipath frame prediction. IEEE TNNLS **33**, 2301–2312 (2021)
53. Wang, Z., Zou, Y., Zhang, Z.: Cluster attention contrast for video anomaly detection. In: ACM MM (2020)
54. Wei, D., Lim, J.J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: CVPR (2018)
55. Wu, P., Liu, J., Shen, F.: A deep one-class neural network for anomalous event detection in complex scenes. IEEE TNNLS **31**(7), 2609–2622 (2019)
56. Xinyang Feng, Dongjin Song, Y.C.Z.C.J.N.H.C.: Convolutional transformer based dual discriminator generative adversarial networks for video anomaly detection. In: ACM MM (2021)
57. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: CVPR (2019)
58. Ye, M., Peng, X., Gan, W., Wu, W., Qiao, Y.: Anopcn: Video anomaly detection via deep predictive coding network. In: ACM MM (2019)
59. Yu, G., Wang, S., Cai, Z., Zhu, E., Xu, C., Yin, J., Kloft, M.: Cloze test helps: Effective video anomaly detection via learning to complete video events. In: ACM MM (2020)
60. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)
61. Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., Hua, X.S.: Spatio-temporal autoencoder for video anomaly detection. In: ACM MM (2017)
62. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: ICLR (2018)