





VideoPatchCore: An Effective Method to Memorize Normality for Video Anomaly Detection

Sunghyun Ahn¹, Youngwan Jo¹, Kijung Lee¹, and Sanghyun Park^{1,*}

Department of Computer Science, Yonsei University, Seoul, Republic of Korea
{skd, jyy1551, rlwjd4177, sanghyun}@yonsei.ac.kr

Abstract. Video anomaly detection (VAD) is a crucial task in video analysis and surveillance within computer vision. Currently, VAD is gaining attention with memory techniques that store the features of normal frames. The stored features are utilized for frame reconstruction, identifying an abnormality when a significant difference exists between the reconstructed and input frames. However, this approach faces several challenges due to the simultaneous optimization required for both the memory and encoder-decoder model. These challenges include increased optimization difficulty, complexity of implementation, and performance variability depending on the memory size. To address these challenges, we propose an effective memory method for VAD, called VideoPatchCore. Inspired by PatchCore, our approach introduces a structure that prioritizes memory optimization and configures three types of memory tailored to the characteristics of video data. This method effectively addresses the limitations of existing memory-based methods, achieving good performance comparable to state-of-the-art methods. Furthermore, our method requires no training and is straightforward to implement, making VAD tasks more accessible. Our code is available online at github.com/SkiddieAhn/Paper-VideoPatchCore.

Keywords: Video Anomaly Detection · Machine learning · Computer vision

1 Introduction

The task of video anomaly detection (VAD) involves identifying anomalies within video sequences. Typically, deep learning-based VAD methods employ one-class classification (OCC) techniques as anomalies are rare and it is challenging to define them precisely. OCC methods train models exclusively with normal data. In VAD, prominent OCC approaches include reconstruction and prediction-based methods [10,26,39,18,37]. These approaches utilize encoder-decoder models such as autoencoders (AEs), generative adversarial networks (GANs) [9], and others.

* Corresponding author: Sanghyun Park (sanghyun@yonsei.ac.kr)

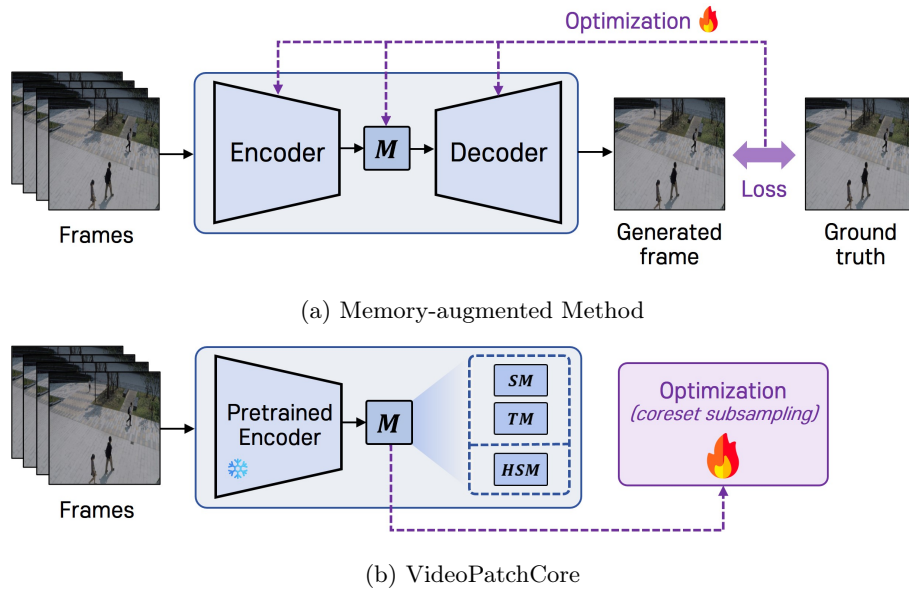


Fig. 1: (a) Conventional memory-augmented methods optimize both the memory and the encoder-decoder model, simultaneously. (b) The proposed VideoPatchCore focuses on memory optimization.

During training, these models train to reconstruct or predict frame using only normal data, operating under the assumption that anomalies will result in high reconstruction or prediction errors during testing. However, these methods face challenges due to the exceptional generalization ability of deep learning models, which can generate plausible outputs for anomalous data [39,13,14,12].

Recently, methods that store the features of normal data in memory during training have shown improved performance in reducing the generalization ability for anomalies. Memory-augmented VAD approaches [8,27,3,24,20,38,32,19,33] store the features of normal frames in memory and retrieve similar features from the memory to generate the input or future frame during testing. In such cases, anomalies result in higher reconstruction or prediction errors because their similar features are not stored in the memory. Figure 1a illustrates these memory utilization approaches. While memory-augmented methods demonstrate good performance, they also present several challenges. First, the simultaneous optimization of the model and memory makes the optimization process difficult and the implementation complex. Additionally, the size of the memory significantly impacts performance [41,2]. Although many recent studies [27,33] have focused on reducing memory size, finding an appropriate memory size for large video datasets remains a challenge.

Inspired by PatchCore [31], we propose VideoPatchCore (VPC) to address the limitations of the previously mentioned memory utilization methods. As

shown in Figure 1b, VPC leverages the vision encoder of the pre-trained CLIP [28] without additional training and optimizes memory using greedy coreset subsampling [1]. This enables effective VAD with a small memory footprint even for large datasets. Unlike PatchCore, which is used at the image level, VPC performs anomaly detection at the video level by using two streams: local and global. The local stream detects anomalies in individual objects, while the global stream detects anomalies in entire frames. The local stream employs a spatial memory bank to identify the appearance anomalies and a temporal memory bank to detect action anomalies. The global stream utilizes a high-level semantic memory bank to detect anomalies in interactions among multiple objects and abnormalities that can only be identified by analyzing the scene (*e.g.*, wrong direction). By utilizing spatial and temporal memory banks, VPC considers the spatiotemporal characteristics of videos, and with the two-stream approach, it can detect various forms of anomalies.

Through experiments, we demonstrate the effectiveness of using three memory banks in the video domain. Additionally, we introduce two methods for effectively memorizing video characteristics and demonstrate the robustness of performance concerning memory size on large datasets. Our proposed VPC shows good performance comparable to state-of-the-art techniques. We anticipate that VPC will enhance the accessibility of the VAD field and facilitate its use in real-world applications.

Our contributions are as follows:

- We propose VPC, an extension of PatchCore developed for image anomaly detection, to perform effective video anomaly detection.
- VPC employs two streams (local and global) and three memory banks (spatial, temporal, and high-level semantic) to capture the spatiotemporal characteristics of videos and detect various forms of anomalies.
- VPC achieves good performance comparable to state-of-the-art methods.

2 Related Work

2.1 Reconstruction & prediction-based Video Anomaly Detection

The reconstruction-based approach is based on the assumption that a model trained to reconstruct normal video frames fails to accurately reconstruct abnormal frames. Nguyen [26] proposed a model comprising one encoder and two decoders. This model considered the t^{th} frame as the input and predicted both the reconstructed frame and the optical flow between the t^{th} and $t + 1^{th}$ frames. Zaheer [39] used GANs to generate high-quality reconstructed frames. In this approach, the discriminator was trained to classify high-quality frames as normal and low-quality frames as abnormal, utilizing a generator in its previous state that produced low-quality frames.

The prediction-based approach is based on the assumption that when a model is trained to predict future or past frames of normal video sequences, it fails to accurately predict abnormal frames. Liu [18] utilized FlowNet and GANs to

predict the $t + 1^{th}$ frame from t frames. Yang [37] suggested selecting key frames from t frames and predicting all t frames based only on these key frames.

2.2 Video Anomaly Detection using Memory

In VAD, memory is utilized to address the issue of accurately generating abnormal frames in OCC method. Gong [8] proposes a method that stores the features of normal frames in memory and generates normal frames using these stored memory items. Park [27] suggests a method for learning various patterns of normal data by clustering similar features and using the cluster centers as memory items, thus storing numerous normal features in a compact memory. Sun [33] employs contrastive learning to cluster features based on their classes and stores each feature in memory. This technique allows for the creation of a compact memory by leveraging the semantic information of objects.

2.3 Representation-based Image Anomaly Detection

In the image domain, representation-based approaches have gained attention as effective methods for anomaly detection. This approach utilizes pre-trained networks on large-scale datasets such as ImageNet [7] and compares the embeddings of normal and abnormal images to detect anomalies. Cohen [6] proposes feature pyramid matching, leveraging the different information at each layer of the network. Features from the training data are stored in memory, and anomaly scores for the features of test data are computed using k-NN with respect to the features stored in memory. Roth [31] points out that high-level features of pre-trained networks are specialized for classification purposes, indicating the need to extract features at the middle level. These features are stored in memory, and memory optimization is achieved through Coreset Subsampling.

3 Method

3.1 Overview

Figure 2 depicts the proposed VPC model, consisting of two streams: the local stream at the object level and the global stream at the frame level. Each stream extracts features through a vision encoder from CLIP, pre-trained on large-scale datasets, and these features are segmented according to the type of memory and accessed in the memory bank. During this process, the memorization and inference stages are performed. In the memorization stage, features are stored in memory, which is then optimized using greedy coreset subsampling from Patch-Core. In the inference stage, anomaly scores are derived by calculating distances between memory and features. The following provides a detailed description of each component of the proposed VPC.

Vision Encoder. We adopt a CNN-based CLIP model as the Encoder and

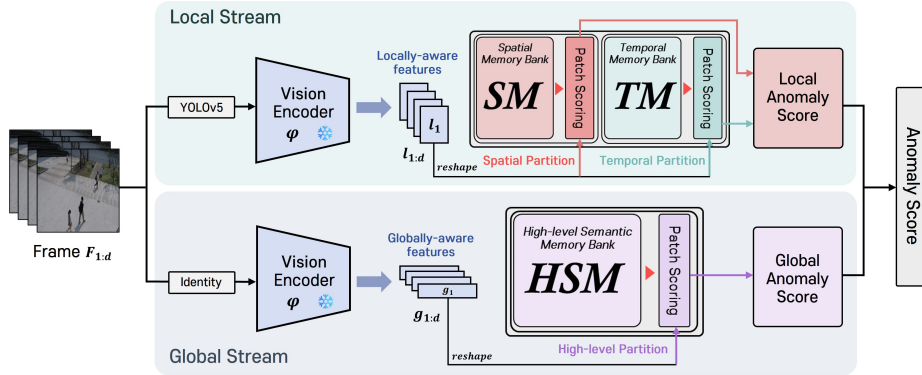


Fig. 2: Architecture of VideoPatchCore (VPC). It consists of two streams (local and global) and three memory banks (spatial, temporal, and high-level semantic). The Spatial Memory Bank deals with the appearance information of objects to identify inappropriate objects. The Temporal Memory Bank handles the motion information of objects to detect inappropriate actions. The High-level Semantic Memory Bank processes the global context of frames to identify anomalies related to multiple objects or scenes.

utilize layers 2, 3 φ_2, φ_3 similar to PatchCore. Typically, the shallow layers of the network capture the local features, while deeper layers capture the global features. However, the first layer captures overly detailed features such as edges and corners, and the last layer captures features biased towards classification, which are not helpful for anomaly detection. Thus, we process and utilize features from the intermediate layers φ_2, φ_3 .

Local Stream. The local stream is composed of YOLOv5 [15] for object detection, a vision encoder φ for feature extraction, and spatial memory (SM) and temporal memory (TM) for storing the object appearance and motion information. Given a sequence of d consecutive frames $F_1, F_2, \dots, F_d \in \mathbb{R}^{3 \times H \times W}$, YOLOv5 generates a set of n objects for each frame, $O_1, O_2, \dots, O_d \in \mathbb{R}^{n \times 3 \times H \times W}$. Subsequently, the locally-aware features $l_1, l_2, \dots, l_d \in \mathbb{R}^{n \times c \times h \times w}$ are produced that represent the features of the objects.

Specifically, average pooling is performed on the features from φ_2, φ_3 and they are concatenated based on φ_2 . This strategy aims to integrate both fine-grained and coarse-grained information that is necessary for object-level VAD. However, as φ_2, φ_3 have relatively small receptive fields, pooling is used to consolidate information over a wider range. The equation for constructing l_i is as follows:

$$l_i = \langle f_{\text{ap}}(\varphi_2(O_i)), f_{\text{ap}}(\varphi_3(O_i)) \rangle \quad (1)$$

where f_{ap} represents the average pooling and $\langle \cdot, \cdot \rangle$ denotes tensor concatenation.

Subsequently, $l_{1:d}$ are reshaped into $lf \in \mathbb{R}^{n \times c \times d \times h \times w}$. However, the dimensionality of lf is impractically large for video processing. Therefore, we employ a method called split pooling to reduce the number of channels, transforming it into $\mathbb{R}^{n \times c' \times d \times h \times w}$ ($c \gg c'$). The details of this approach can be found in Section 4.6. Subsequently, lf is split into spatial patches and temporal patches through spatial partition and temporal partition, respectively, and stored in SM and TM.

Global Stream. The global stream constructs globally-aware features $g_1, g_2, \dots, g_d \in \mathbb{R}^{c \times 1 \times 1}$ to capture the overall context of frames, thereby differing from the local stream, which focuses on individual objects within frames. We utilize φ_2, φ_3 , applying global pooling to obtain global information. Specifically, average pooling is applied to the features of each layer, and these are concatenated based on φ_3 , which handles more global information. During global pooling, both average and max pooling techniques are employed to achieve a better representation. The equation for constructing g_i is as follows:

$$g_i = \langle f_{\text{ap}}(\varphi_2(F_i)), f_{\text{ap}}(\varphi_3(F_i)) \rangle \quad (2)$$

$$g_i = f_{\text{gap}}(g_i) + f_{\text{gmp}}(g_i) \quad (3)$$

where f_{gap} and f_{gmp} denote global average pooling and global max pooling, respectively.

Subsequently, $g_{1:d}$ are reshaped into $gf \in \mathbb{R}^{c \times d \times 1 \times 1}$. The gf constructed in this manner includes the global information of each frame and is converted into high-level patches through high-level partition. Afterwards, the patch is stored in the High-level Semantic Memory (HSM).

3.2 Patch Partition

Figure 3 illustrates the proposed patch partition method, which consists of spatial partition, temporal partition, and high-level partition. The spatial partition focuses on object appearance information, generating patches while disregarding temporal information. The temporal partition emphasizes object motion information, generating patches while ignoring spatial information. The high-level partition utilizes extensive spatiotemporal information to generate patches for understanding the contextual information across frames. The detailed descriptions of each partition are as follows.

Spatial Partition. Appearance information is crucial for evaluating anomalies in objects and can be derived from spatial features. We compress the input feature map lf using temporal global pooling [16] to preserve spatial information while ignoring temporal aspects. Subsequently, average pooling is applied to consider various regions of the object. For instance, when assessing a person’s pose, it is essential to consider multiple joints together. The equation for creating *SpatialPatches* is as follows:

$$\text{SpatialPatches} = f_{\text{ap}}(f_{\text{tgp}}(lf)) \quad (4)$$

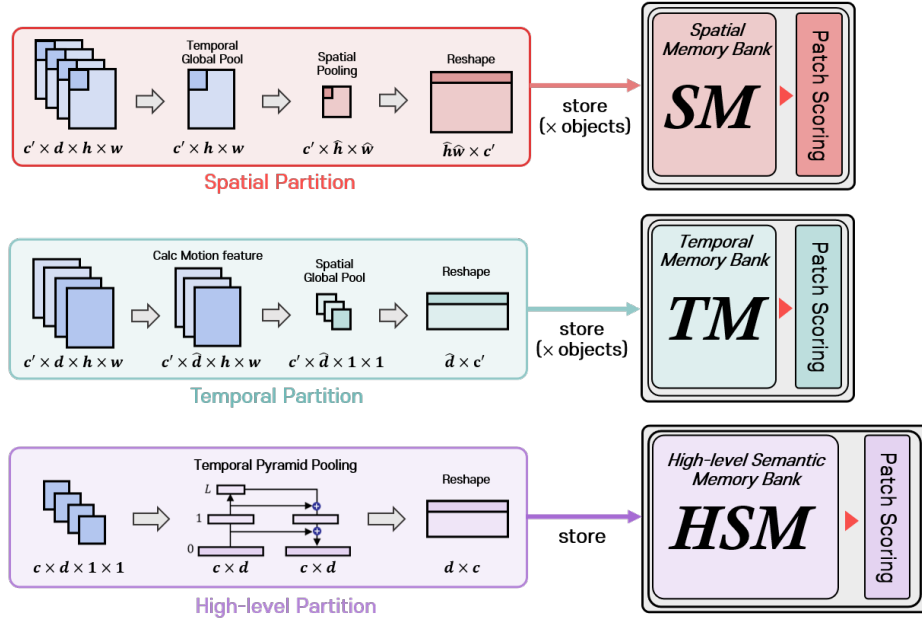


Fig. 3: Patch Partition Method. ‘objects’ denotes the number of objects(n), and objects are processed in parallel considering batch processing.

where $f_{t_{gp}}$ denotes the temporal global pooling. Finally, the results are reshaped into $\mathbb{R}^{(n \cdot \hat{h} \cdot \hat{w}) \times c'}$, where $h \geq \hat{h}$ and $w \geq \hat{w}$.

Temporal Partition. Motion information in objects represents changes over time, making it crucial for VAD. We utilize the adjacent temporal information within lf to generate motion features mf . To achieve this, the feature differences are computed and the representative motion values are determined through global average pooling. The equation for creating *TemporalPatches* is as follows:

$$mf_{(t)} = |lf_{(t+1)} - lf_{(t)}| \quad (5)$$

$$TemporalPatches = f_{gap}(mf) \quad (6)$$

$mf_{(t)} \in \mathbb{R}^{n \times c' \times h \times w}$ represents the difference between the t^{th} and $t + 1^{th}$ time within lf , and it belongs to $mf \in \mathbb{R}^{n \times c' \times \hat{d} \times h \times w}$, where $\hat{d} = d - 1$. Finally, the results are reshaped into $\mathbb{R}^{(n \cdot \hat{d}) \times c'}$.

High-level Partition. In frames, the global context processes the relationship between the objects and the scene, and considers interactions between different objects, thereby enabling more accurate VAD. As gf already encompasses the global context from a spatial perspective, temporal pyramid pooling is utilized

to obtain high-level temporal information. Additionally, this approach secures multi-scale temporal information, addressing the limitation of only using adjacent temporal information in temporal partition methods. The equation for creating *HighlevelPatches* is as follows:

$$HighlevelPatches = \sum_{l=0}^L f_{mp}^l(gf) \quad (7)$$

Temporal pyramid pooling is implemented using f_{mp}^l . In this case, f_{mp}^l represents applying the max pooling operation l times. Finally, the results are reshaped into $\mathbb{R}^{d \times c}$.

3.3 Inference

Mismatched Pooling. During the temporal partition process, global pooling is applied differently in the memorization and inference stages. The memorization stage employs global average pooling (GAP), while global max pooling (GMP) is used during inference. As normal objects typically exhibit more static characteristics compared to abnormal ones, we expected normal motion features to show similar average and maximum values. In contrast, abnormal motion features were anticipated to show significant discrepancies in these values. By adopting this approach, VAD performance improves by preventing the retrieval of nearby temporal patches from memory when abnormal data is input. Experimental validation of the effectiveness of this methodology is detailed in Section 4.6.

Anomaly Scoring. All nearest neighbors are computed between patches X generated from test frames and the memory bank M . Subsequently, the representative patch x^* is selected from the X and its nearest neighbor m^* is selected from the M . The distance s^* between x^* and m^* is assigned as the score representing the frames.

$$x^*, m^* = \arg \max_{x \in X} \arg \min_{m \in M} \|x - m\|_2 \quad (8)$$

$$s^* = \|x^* - m^*\|_2 \quad (9)$$

As three types of memory are used, s^* is computed three times for SM, TM, and HSM, respectively. The $s_{spatial}^*$ calculated from SM and $s_{temporal}^*$ calculated from TM determine the local anomaly score (LAS) in the local stream. Conversely, $s_{high-level}^*$ computed from HSM is utilized as the global anomaly score (GAS). Ultimately, the LAS and GAS combine to derive the anomaly score. The equation for calculating the anomaly score is as follows:

$$LAS = \delta_1 \cdot s_{spatial}^* + \delta_2 \cdot s_{temporal}^* \quad (10)$$

$$GAS = s_{high-level}^* \quad (11)$$

$$AnomalyScore = \gamma_1 \cdot \frac{LAS - \mu(LAS)}{\sigma(LAS)} + \gamma_2 \cdot \frac{GAS - \mu(GAS)}{\sigma(GAS)} \quad (12)$$

4 Experiments

4.1 Datasets

The proposed model was validated on three datasets: CUHK Avenue (Avenue)[21], ShanghaiTech (SHTech)[23], and IITB Corridor (Corridor)[30]. As each dataset was constructed for OCC method, the training dataset consisted only of normal videos, while the test dataset contained both normal and abnormal videos.

Avenue. The dataset consisted of 16 training and 21 testing videos filmed with a single camera at the same angle. It included anomaly events such as throwing paper, running on the road, and walking in the wrong direction.

SHTech. The dataset consisted of 330 training videos and 107 testing videos, capturing 13 scenes from different angles. It included various anomaly events such as the appearance of anomaly objects (*e.g.*, cars, bicycles) and action anomalies (*e.g.*, fighting, jumping).

Corridor. The dataset consisted of 208 training and 150 testing videos filmed with a single camera at the same angle. It included previously unseen anomaly events such as protesting, hiding, and playing with the ball.

4.2 Implementation details

Object Detection. The pretrained YOLOv5 was utilized for object detection in the videos. The bounding boxes of the objects were generated based on the last or middle frames among the d input frames provided, and these bounding boxes were shared across other frames. Specifically, for the Avenue, SHTech, and Corridor datasets, we used 10, 4, 4 input frames, respectively. For the Avenue dataset, objects were detected based on the middle frame. Furthermore, to minimize the loss of temporal information, margins were added to the width and height of the bounding boxes, ensuring that they remained equal in size. Finally, the object images were resized to (224, 224) before being input into the pre-trained network.

Memorizing details. We utilized CLIP’s ResNet-101 as the vision encoder. The value c' for split pooling in the local stream was determined by the dataset size. Specifically, it was set to 32 for the smaller Avenue dataset and 64 for the larger SHTech and Corridor datasets. For lf and gf generation, average pooling with a kernel size of (3, 3) and stride of 1 was used. In the spatial partition, average pooling was performed with a kernel size of (4, 4). In the high-level partition, L in TPP was set to 2, and during max pooling, a kernel size of (2) with a stride of 1 was used. During memory optimization, coreset subsampling rates were set to 1, 25, and 10% for the Avenue, SHTech, and Corridor datasets, respectively.

Inference details. The anomaly score was smoothed using a 1D Gaussian filter during testing. The parameters δ_1 and δ_2 used in LAS computation were set to (0.7, 0.3) for Avenue and (0.5, 0.5) for the SHTech and Corridor datasets. The parameters γ_1 and γ_2 used in the final anomaly score computation were set to

Table 1: Comparison of VPC against SOTA methods in terms of AUROC. Avenue: CUHK Avenue, SHTech: ShanghaiTech, Corridor: IITB Corridor. The best results are **bolded**. The second-best results are underlined. *experimented by us.

Method	Venue	Memory	Avenue	SHTech	Corridor
FFP[18]	CVPR 18		84.9%	72.8%	64.7%
MPED-RNN[25]	CVPR 19		-	73.4%	64.3%
MemAE[8]	ICCV 19	✓	83.3%	71.2%	-
AMC[26]	ICCV 19		86.9%	-	-
MTP[30]	WACV 20		82.9%	76.0%	67.1%
CDDA[5]	ECCV 20		86.0%	73.3%	-
MNAD[27]	CVPR 20	✓	88.5%	70.5%	-
ROADMAP[35]	TNNLS 21		88.3%	76.6%	-
AMMC-Net[3]	AAAI 21	✓	86.6%	73.7%	-
MPN[24]	CVPR 21	✓	89.5%	73.8%	-
HF ² -VAD[20]	ICCV 21	✓	91.1%	76.2%	-
LLSH[22]	TCSVT 22		87.4%	77.6%	73.5%
VABD[17]	TIP 22		86.6%	78.2%	72.2%
DLAN-AC[38]	ECCV 22	✓	89.9%	74.7%	-
Jigsaw[34]	ECCV 22		92.2%	84.3%	-
Sun et al.[32]	AAAI 23	✓	91.5%	78.6%	-
Cao et al.[4]	CVPR 23		86.8%	79.2%	<u>73.6%</u>
USTN-DSC[37]	CVPR 23		89.9%	73.8%	-
DMAD[19]	CVPR 23	✓	92.8%	78.8%	-
FPDM[36]	ICCV 23		90.1%	78.6%	-
STG-NF[11]	ICCV 23		61.8%*	85.9%	61.4%*
HSC[33]	ICCV 23	✓	92.4%	83.0%	-
Ristea et al.[29]	CVPR 24		91.3%	79.1%	-
Zhang et al.[40]	CVPR 24		92.4%	<u>85.1%</u>	-
VPC (Ours)	-	✓	92.8%	<u>85.1%</u>	76.4%

(0.7, 0.3) for Avenue and (0.9, 0.1) for the SHTech and Corridor datasets. The experiments were conducted using an NVIDIA GeForce RTX 3090 graphics card.

4.3 Evaluation Criteria

Consistent with prior research, the frame-level area under the curve (AUC) of Receiver Operation Characteristic (ROC) was used to evaluate the performance of the proposed model. This evaluation was calculated as the area under the ROC curve by varying the threshold. Specifically, we aggregated anomaly scores across all frames in the dataset to calculate the micro-averaged AUROC.

4.4 Comparison with SOTA models

Table 1 presents the results of the proposed model’s performance on three benchmark datasets. It demonstrates the competitive performance compared

to other state-of-the-art (SOTA) methods. Specifically, when compared to HSC [33], which utilizes appearance and motion memory, our approach outperforms by 0.4% and 2.1% on the Avenue and SHTech datasets, respectively, leveraging three memory components effectively. And, compared to FPDM [36], which also employs the CLIP, our model achieves better performance by accurately detecting anomaly events at the local stream. Furthermore, compared to the other methods using memory, a superior performance is achieved. This demonstrates that a method focused on memory optimization is more effective than traditional methods using memory. In addition, individuals without deep learning expertise can easily use the proposed approach because it does not require training.

4.5 Qualitative Evaluation

To intuitively understand the use of multiple memories, for the Avenue dataset, we compared PatchCore (PC), which utilizes appearance information, with VideoPatchCore (VPC), which incorporates not only appearance information, but also motion and high-level information. Here, PC refers to the use of spatial memory only in VPC. Figure 4a depicts an anomalous frame showing the action of throwing papers, necessitating consideration of motion information. Therefore, temporal memory plays a crucial role in this scenario. Figure 4b depicts an anomalous frame showing the action of walking in the incorrect direction, necessitating consideration of both object and scene contexts. Therefore, high-level semantic memory plays a crucial role in this scenario. It is known that VPC, which utilizes temporal and high-level semantic memory, detects anomalies well, whereas PC, which does not utilize them, does not detect anomaly.

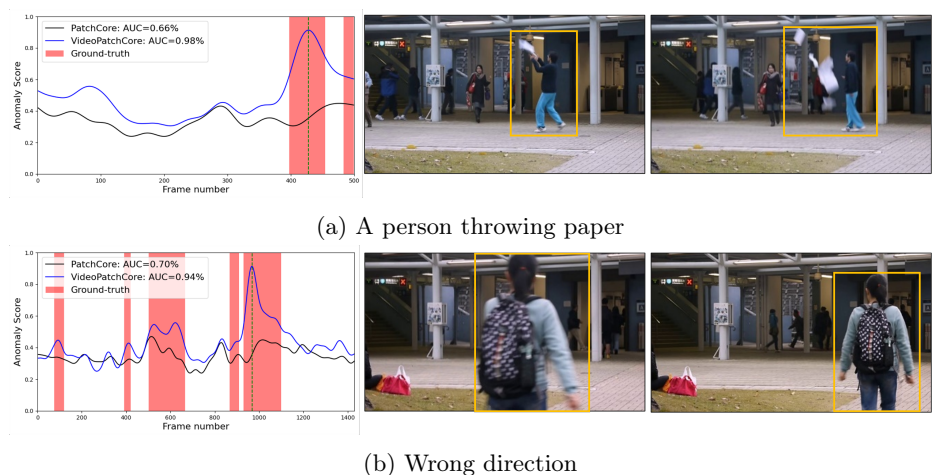


Fig. 4: Comparison of the anomaly scores between VideoPatchCore and PatchCore in the Avenue dataset.

Table 2: Comparison of the AUROC scores for the spatial and temporal memory on the Avenue, SHTech and Corridor datasets.

Memory	Avenue	SHTech	Corridor
Spatial	84.8%	74.7%	70.5%
Temporal	66.9%	78.8%	73.5%
Spatial+Temporal	90.3%	84.8%	76.3%

Table 3: Comparison of the AUROC scores for compression methods on the Avenue, SHTech and Corridor datasets.

Compression	Avenue	SHTech	Corridor
Head	82.7%	76.9%	74.9%
Random	85.9%	83.8%	74.8%
Split Pool	90.3%	84.8%	76.3%

Additional experiments on the SHTech and Corridor dataset are presented in the supplementary materials.

4.6 Ablation study

Spatial and Temporal memory. Table 2 presents a comparison of the VAD performance affected by the spatial and temporal memory. The spatial memory effectively detects the appearance of the anomaly object, while temporal memory effectively detects the action anomaly event. As it is important to consider both the motion and appearance information for accurate anomaly detection, using both types of memory helps achieve high performance on the Avenue, SHTech, and Corridor datasets.

Split pooling. Table 3 presents the AUROC scores based on information compression techniques in the local stream. Head compresses lf by selecting c' channels from the front. Random compresses lf by selecting random c' channels. Split pool divides lf into c' groups and compresses by averaging the channels of each group. The common compression methods lose performance as they fail to preserve the original information, whereas the proposed compression method achieves high performance by effectively compressing the original information.

Mismatched pooling. Table 4 presents the AUROC scores based on feature pooling methods in the temporal partition. The avg and max pool perform GAP and GMP during the memorization and inference stage, respectively. The common pooling methods fail to effectively detect action anomalies, whereas the proposed method achieves effective detection. This demonstrates that differentiating between normal and abnormal temporal patches by varying the pooling

Table 4: Comparison of AUROC scores for feature pooling methods in the temporal partition across the Avenue, SHTech and Corridor datasets.

Pooling	Avenue	SHTech	Corridor
Avg Pool	84.9%	76.1%	74.7%
Max Pool	88.2%	83.8%	75.7%
Mismatched Pool	90.3%	84.8%	76.3%

Table 5: Comparison of the AUROC scores for the local and global stream on the Avenue, SHTech and Corridor datasets.

Stream	Avenue	SHTech	Corridor
Local	90.3%	84.8%	76.3%
Global	84.4%	68.4%	67.2%
Local+Global	92.8%	85.1%	76.4%

methods at each stage is effective for VAD.

Local and Global stream. Table 5 presents a comparison of the anomaly detection performance affected by the local and global stream. When using only the local stream, it is difficult to detect anomalies based on scenes and abnormal behaviors among multiple objects. To address this issue, we incorporated the global stream, which utilizes a wide range of spatial-temporal information. When adding the global stream to Avenue, there is a significant performance improvement, whereas adding the global stream to SHTech and Corridor results in relatively slight performance improvements. In the latter datasets, there are many situations where objects are adjacent to each other, allowing the local stream alone to partially fulfill the role of the global stream.

4.7 Further Analysis

Coreset subsampling ratio. Table 6 presents the AUROC scores based on the subsampling ratio. Generally, increasing the sampling ratio in large datasets stores a diverse range of normal features in memory, leading to improved performance, but it can also result in slower processing speed. However, using Coreset subsampling, the performance difference in memory usage between 1% and 99% is very minimal, less than 0.5%, and the frames per second (FPS) can be up to 7.3 times faster. This demonstrates the effectiveness of coreset subsampling in large datasets such as SHTech and Corridor.

Memorizing Technique. Table 7 presents the AUROC scores based on the subsampling order. The former method constructs memory from each video and concatenates them, whereas the latter method constructs memory from all videos collectively. We expected that the former method, which optimizes memory for

Table 6: Comparison of AUROC scores for the subsampling ratio on the SHTech and Corridor datasets.

Subsampling ratio	SHTech		Corridor	
	AUC	FPS	AUC	FPS
1%	84.6% (-0.5%)	170.9	76.0% (-0.4%)	143.4
10%	85.0% (-0.1%)	154.8	76.4% (-0.0%)	113.5
25%	85.1% (-0.0%)	96.1	76.3% (-0.1%)	60.0
50%	85.1% (-0.0%)	54.8	76.3% (-0.1%)	40.0
75%	85.1% (-0.0%)	39.6	76.3% (-0.1%)	25.2
99%	85.1%	31.0	76.4%	19.5

Table 7: Comparison of AUROC scores for the memorizing technique at memory usage levels of both 10% and 99% on the SHTech and Corridor datasets.

Memorizing technique	Ratio	SHTech	Corridor
Subsampling	10%	85.0%	76.4%
→ concat	99%	85.1%	76.4%
Concat	10%	84.9% (-0.1%)	76.3% (-0.1%)
→ subsampling	99%	85.0% (-0.1%)	76.4% (-0.0%)

each video, would be less constrained by hardware limitations but might suffer from a lower performance due to the uneven distribution of features. However, in fact, the performance difference between the two methods is almost nonexistent. This suggests that it is possible to sufficiently store a diverse range of normal features regardless of the order of subsampling.

5 Conclusion

We present a VPC that extends PatchCore to the video level. The proposed model consists of three types of memory considering the characteristics of the video, and maintains high AUC and FPS through memory optimization using coreset subsampling. This approach can be used as an effective memory in VAD by solving three problems of traditional memory methods (increased optimization difficulty, complexity of implementation, and performance variability depending on the memory size). In addition, VPC shows excellent performance in all benchmark datasets and does not require training, increasing accessibility in the VAD field. We expect VPC to make a significant contribution to the VAD field.

Acknowledgements. This research was supported by the National Research Foundation (NRF) funded by the Korean government (MSIT) (No. RS-2023-00229822).

References

1. Agarwal, P.K., Har-Peled, S., Varadarajan, K.R.: Geometric approximation via coresets (2007)
2. Astrid, M., Zaheer, M., Lee, J.Y., Lee, S.I.: Learning not to reconstruct anomalies. In: British Machine Vision Conference (2021)
3. Cai, R., Zhang, H., Liu, W., Gao, S., Hao, Z.: Appearance-motion memory consistency network for video anomaly detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 938–946 (2021)
4. Cao, C., Lu, Y., Wang, P., Zhang, Y.: A new comprehensive benchmark for semi-supervised video anomaly detection and anticipation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20392–20401 (2023)
5. Chang, Y., Tu, Z., Xie, W., Yuan, J.: Clustering driven deep autoencoder for video anomaly detection. In: Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. pp. 329–345. Springer (2020)
6. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. ArXiv [abs/2005.02357](https://arxiv.org/abs/2005.02357) (2020)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
8. 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: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1705–1714 (2019)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Advances in neural information processing systems* **27** (2014)
10. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 733–742 (2016)
11. Hirschorn, O., Avidan, S.: Normalizing flows for human pose anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13545–13554 (2023)
12. Hong, S., Ahn, S., Jo, Y., Park, S.: Making anomalies more anomalous: Video anomaly detection using a novel generator and destroyer. *IEEE Access* (2024)
13. Huang, C., Wen, J., Xu, Y., Jiang, Q., Yang, J., Wang, Y., Zhang, D.: Self-supervised attentive generative adversarial networks for video anomaly detection. *IEEE transactions on neural networks and learning systems* (2022)
14. Huang, C., Wu, Z., Wen, J., Xu, Y., Jiang, Q., Wang, Y.: Abnormal event detection using deep contrastive learning for intelligent video surveillance system. *IEEE Transactions on Industrial Informatics* **18**(8), 5171–5179 (2021)
15. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., Fang, J., Yifu, Z., Wong, C., Montes, D., et al.: ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation. Zenodo (2022)
16. Kwon, H., Kwak, S., Cho, M.: Video understanding via convolutional temporal pooling network and multimodal feature fusion. In: Proceedings of the 1st Workshop and Challenge on Comprehensive Video Understanding in the Wild. pp. 35–39 (2018)

17. Li, J., Huang, Q., Du, Y., Zhen, X., Chen, S., Shao, L.: Variational abnormal behavior detection with motion consistency. *IEEE Transactions on Image Processing* **31**, 275–286 (2021)
18. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection—a new baseline. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 6536–6545 (2018)
19. Liu, W., Chang, H., Ma, B., Shan, S., Chen, X.: Diversity-measurable anomaly detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 12147–12156 (2023)
20. 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: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 13588–13597 (2021)
21. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: *Proceedings of the IEEE international conference on computer vision*. pp. 2720–2727 (2013)
22. Lu, Y., Cao, C., Zhang, Y., Zhang, Y.: Learnable locality-sensitive hashing for video anomaly detection. *IEEE Transactions on Circuits and Systems for Video Technology* **33**(2), 963–976 (2022)
23. Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked rnn framework. In: *Proceedings of the IEEE international conference on computer vision*. pp. 341–349 (2017)
24. Lv, H., Chen, C., Cui, Z., Xu, C., Li, Y., Yang, J.: Learning normal dynamics in videos with meta prototype network. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 15425–15434 (2021)
25. Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., Venkatesh, S.: Learning regularity in skeleton trajectories for anomaly detection in videos. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 11996–12004 (2019)
26. Nguyen, T.N., Meunier, J.: Anomaly detection in video sequence with appearance-motion correspondence. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 1273–1283 (2019)
27. Park, H., Noh, J., Ham, B.: Learning memory-guided normality for anomaly detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 14372–14381 (2020)
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning* (2021)
29. Ristea, N.C., Croitoru, F.A., Ionescu, R.T., Popescu, M., Khan, F.S., Shah, M., et al.: Self-distilled masked auto-encoders are efficient video anomaly detectors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 15984–15995 (2024)
30. Rodrigues, R., Bhargava, N., Velmurugan, R., Chaudhuri, S.: Multi-timescale trajectory prediction for abnormal human activity detection. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. pp. 2626–2634 (2020)
31. Roth, K., Pemula, L., Zepeda, J., Scholkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 14298–14308 (2021)

32. Sun, C., Shi, C., Jia, Y., Wu, Y.: Learning event-relevant factors for video anomaly detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2384–2392 (2023)
33. Sun, S., Gong, X.: Hierarchical semantic contrast for scene-aware video anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 22846–22856 (2023)
34. Wang, G., Wang, Y., Qin, J., Zhang, D., Bao, X., Huang, D.: Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles. In: European Conference on Computer Vision. pp. 494–511. Springer (2022)
35. 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 transactions on neural networks and learning systems* **33**(6), 2301–2312 (2021)
36. Yan, C., Zhang, S., Liu, Y., Pang, G., Wang, W.: Feature prediction diffusion model for video anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5527–5537 (2023)
37. Yang, Z., Liu, J., Wu, Z., Wu, P., Liu, X.: Video event restoration based on keyframes for video anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14592–14601 (2023)
38. Yang, Z., Wu, P., Liu, J., Liu, X.: Dynamic local aggregation network with adaptive clusterer for anomaly detection. In: European Conference on Computer Vision. pp. 404–421. Springer (2022)
39. Zaheer, M.Z., Lee, J.h., Astrid, M., Lee, S.I.: Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14183–14193 (2020)
40. Zhang, M., Wang, J., Qi, Q., Sun, H., Zhuang, Z., Ren, P., Ma, R., Liao, J.: Multi-scale video anomaly detection by multi-grained spatio-temporal representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17385–17394 (2024)
41. Zhong, Y., Chen, X., Jiang, J., Ren, F.: A cascade reconstruction model with generalization ability evaluation for anomaly detection in videos. *Pattern Recognition* **122**, 108336 (2022)