Cluster Attention Contrast for Video Anomaly Detection

Ziming Wang∗
wziming@pku.edu.cn
ADSPLAB, School of ECE, Peking University
ShenZhen, China

Yuexian Zou†
zhouyx@pku.edu.cn
ADSPLAB, School of ECE, Peking University
ShenZhen, China

Zeming Zhang
zhangzm@hit.edu.cn
Harbin Institute of Technology
Harbin, China

ABSTRACT
Anomaly detection in videos is commonly referred to as the discrimination of events that do not conform to expected behaviors. Most existing methods formulate video anomaly detection as an outlier detection task and establish normal concept by minimizing reconstruction loss or prediction loss on training data. However, these methods performances suffer drops when they cannot guarantee either higher reconstruction errors for abnormal events or lower prediction errors for normal events. To avoid these problems, we introduce a novel contrastive representation learning task, Cluster Attention Contrast, to establish subcategories of normality as clusters. Specifically, we employ multi-parallel projection layers to project snippet-level video features into multiple discriminate feature spaces. Each of these feature spaces is corresponding to a cluster which captures distinct subcategory of normality, respectively. To acquire the reliable subcategories, we propose the Cluster Attention Module to draw the cluster attention representation of each snippet, then maximize the agreement of the representations from the same snippet under random data augmentations via momentum contrast. In this manner, we establish a robust normal concept without any prior assumptions on reconstruction errors or prediction errors. Experiments show our approach achieves state-of-the-art performance on benchmark datasets.

CCS CONCEPTS
- Computing methodologies → Computer vision; Scene anomaly detection.

KEYWORDS
video anomaly detection, cluster attention contrast, contrastive learning

ACM Reference Format:

1 INTRODUCTION
Anomaly detection in videos is an important task because of its extensive applications in intelligent surveillance, activity recognition, evidence investigation, etc. Since anomalous events are rarely seen in common environments, anomalies are often defined as behavioral or appearance patterns different from usual patterns in previous work [2, 3]. It is extremely challenging because of the ambiguity and complexity of anomaly definition in real applications. Obviously, it is almost infeasible to collect all examples of abnormal events and perform traditional supervised learning methods. Therefore, for a typical setting for anomaly detection tasks, only normal scenarios are given in the training data. Based on this definition, most recently anomaly detection approaches [3, 6, 27, 30, 38, 53, 60] are based on outlier detection and learn a model of normality from training videos. During inference, events are labeled as abnormal if they deviate from the normality model. These methods for modeling normal concept can be categorized into two types, namely the reconstruction-based framework [12, 13, 35, 53] and the prediction-based framework [37, 61]. They both regard events that do not fit into the normality models as abnormal, but their performances suffer when they cannot guarantee either higher reconstruction
errors for abnormal events or lower prediction errors for normal events. However, these methods do not explicitly distinguish between normal conditions that differ greatly. For example, on the sidewalks on campus, there are both normal when no one to pass by or when a large number of pedestrians pass by after school, but the two conditions have completely different motion patterns.

Different from previous methods, we propose a novel contrastive representation learning task, Cluster Attention Contrast, to establish multiple subcategories of normality. We appreciate the diversity of normal conditions and exploit these gaps to establish a robust normal concept. We propose to employ a contrastive learning paradigm to establish reliable subcategories of normality as clusters. Specifically, we employ multi-parallel projection layers to project snippet-level video features into multiple discriminative feature spaces. Each of these feature spaces is corresponding to a cluster which captures distinct subcategory of normality, respectively. To acquire the reliable subcategories, we propose the Cluster Attention Module and strive to draw the optimal cluster attention representation of each snippet by maximizing the agreement of the representations from the same snippet under random data augmentations. During inference, for each test snippet, the highest similarity between its subcategory-specific representations and corresponding centers is regarded as its regularity score. We detect anomalies based on this intuition: if a sample is far away from the centers in all corresponding feature spaces, we can determine that an anomaly has occurred.

Our contribution is summarized as follows:

- We introduce a novel contrastive representation learning task, Cluster Attention Contrast, to establish a robust normal concept for video anomaly detection task. As far as we know, this is the first work in the related field.
- We propose an effective contrastive learning framework for video anomaly detection. Our method achieves state-of-the-art results on two benchmark datasets: CUHK Avenue, ShanghaiTech Campus, where it reaches 87.0% and 79.3% frame-level AUC score, respectively.

We organize the paper as follows. We present related work in Section 2 and describe our approach in Section 3. We conduct the video anomaly detection experiments and elaborate on the implementation details in Section 4. We draw our conclusions in Section 5.

2 RELATED WORK

Video Anomaly Detection. As one of the most challenging problems, anomaly detection in videos has been extensively studied for many years. Video Anomaly Detection is commonly formalized as an outlier detection task, most of the research addresses the problem under the assumption that anomalies are rare or unseen, and behaviors deviating from normal patterns are supposed to be anomalous. Researchers attempt to encode regular patterns via a variety of statistic models, e.g., the social force model [38], the mixture of dynamic models on texture [30], Hidden Markov Models on video volumes [16, 28], the Markov Random Field upon spatial-temporal domain [27], Gaussian process modeling [6, 29], and identify anomalies as outliers. With the great success of deep learning, the recent video anomaly detection work can be classified into deep reconstruction methods and deep prediction methods. Hasan et al. [13] employed two autoencoders, one is learned on conventional handcrafted features, while another is learned in an end-to-end fashion with a fully convolutional network. On the other hand, Liu et al. [32] proposed a future frame prediction network for anomaly detection. What more closely related to our work are methods [6, 7, 11, 34, 34, 43] that learn a dictionary of atoms representing normal events during training, they utilize sparse representation to construct a dictionary for normal behavior and detect anomalies as the ones with high reconstruction error. The main difference between these methods and our work is that we do not sparsely encode features to evaluate reconstruction loss but directly judge the occurrence of anomalies by the similarities between its subcategory-specific representations and corresponding centers.

Deep Clustering. Over the years, many methods have been developed for deep clustering. Some earlier deep clustering methods divided representation learning and clustering into two stages, draw on SAE [4, 15] or its variants [36, 39, 49, 58] to extract intermediate features, and then employ k-means [30, 36] or spectral clustering [30]. In contrast to them, some methods [17, 19, 54] directly use the convolutional neural network (CNN) or multi-layer perceptron (MLP) for representation learning by designing specific loss functions instead of the reconstruction contrastive learning benefits from deeper and wider networks such as contrastive learning benefits from larger mini-batch sizes and longer training compared to its supervised counterpart. The main difference is our method directly projects the original space into a lower-dimensional feature space instead of using a subset of the original space dimensions.

Contrastive Representation Learning. Contrastive representation learning makes use of internal patterns of data and devises predictive tasks to train a model. Many pretext tasks have been proposed: counting the objects [40], predicting the context [10], filling in missing parts of an image [42], or recovering colors from grayscale images [58]. As for videos, self-supervision strategies involve: predicting future [47], leveraging temporal continuity via tracking [50, 51], or preserving the equivariance of egomotion [23, 41, 63]. CMC [46] presented a contrastive learning framework that enables the learning of unsupervised representations from multiple views of a dataset. The principle of maximization of mutual information enables the learning of powerful representations. These methods can be thought of as building dynamic dictionaries. SimCLR [5] verified the impact of different types of data augmentation and project heads on contrastive learning and pointed out contrastive learning benefits from larger mini-batch sizes and longer training compared to its supervised counterpart. At the same time, contrastive learning benefits from deeper and wider networks like
supervised learning. MoCo[14] present Momentum Contrast as a way of building large and consistent dictionaries for unsupervised learning with a contrastive loss. Considering that video feature extraction requires a considerable amount of computing resources, we use the momentum contrast to train our network to obtain better performance with a small mini-batch size.

3 PROPOSED METHOD

3.1 Overview

Our method proposes to establish a robust normal concept without any prior assumptions on reconstruction errors or prediction errors. Unlike previous works, we introduce an instance discrimination pretext task, Cluster Attention Contrast, to obtain reliable subcategories of normality, which is the core contribution of this work. As illustrated in Figure 2, our model consists of two fully symmetric encoders. Each encoder mainly consists of three components: a pretrained feature extractor, a parallel projection layers group, and a Cluster Attention Module. We employ multi-parallel projection layers to project snippet-level video features into multiple discriminative feature spaces for capturing diverse normal patterns. We introduce the Cluster Attention Module to connect the Cluster Attention Contrast task with the Video Anomaly Detection task.

Specifically, we first construct the input of encoders by a temporal cuboid using a sliding window technique. We stack $T = 8$ frames together as a snippet. It should be noted, all training and inference phases are performed on such snippets. During Training, for each snippet as illustrated in Figure 2, we draw two random data augmentations on the snippet and send them to the two encoders, respectively. In each encoder, we first use a pretrained action classifier to extract snippet-level features and perform a layer normalization to improve regularization. Then we employ $K$ parallel projection layers to project snippet-level video features into $K$ discriminate feature spaces. In this work, $K$ is a manually pre-fixed parameter. Each of these $K$ feature spaces is corresponding to a cluster which captures distinct subcategory of normality. In Cluster Attention Module, we redraw the original snippet-level feature through the linear combination of its $K$ subcategory-specific representations. The linear combination weight is obtained by the similarity between the feature’s subcategory-specific representations and the corresponding cluster centers. After this redrawn snippet-level feature passes through a project head $H$, we can obtain the output of our encoder, i.e., the cluster attention representation.

To estimate the cluster centers while training, we modify an EM-like[8] optimization mechanism: alternately updating the cluster center and the model parameters, which makes a soft assignment based on the posterior probabilities and converges to a local optimum. It is worth noting that our method only calculates the centroid of features in determining feature space as the cluster center instead of introducing an additional clustering task.

Figure 2: Overview of the Cluster Attention Contrast task. We perform two random data augmentation transformations on the input snippet, and send them to the two encoders, respectively. We regard the output of encoder $q$ and the output of encoder $k$ as a positive pair if they originate from the same snippet, and otherwise as a negative sample pair. Finally, we employ a contrastive loss to maximize agreement the cluster attention representation of the positive pairs. Best viewed in color.
Algorithm 1: Cluster Center Estimating

Input:
- temperature $\tau_c$, number of clusters $K$, feature queue $\{u_i\}_{i=1}^M$
- structure of project layers $\{p^q_1, p^q_2, \ldots, p^q_K\}$
- $\{p^k_1, p^k_2, \ldots, p^K_K\}$

Initialize $s^q_{ij} = k^q_{ij} = 0$.

for all $i = 1$ to $M$, $j = 1$ to $K$ do
  $h^q_{ij} = p^q_i(u_i)$
  $w^q_{ij} = \text{softmax}(s^q_{ij})$
  $c^q_j = \sum_{i=1}^M w^q_{ij} h^q_{ij}$ # update centers
end for

for all $q \in \{q, k\}$ do
  for all $i = 1$ to $M$ do
    $h^q_{ij} = p^q_i(u_i)$
    $w^q_{ij} = \text{softmax}(s^q_{ij})$
    $c^q_j = \sum_{i=1}^M w^q_{ij} h^q_{ij}$ # update centers
  end for
end for

return cluster centers for each encoder,

$$\{c^q_1, c^q_2, \ldots, c^q_K \mid q \in \{q, k\}\}$$  

$$\{c^q_1, c^q_2, \ldots, c^q_k \mid q \in \{q, k\}\}$$

We elaborate in section 3.2 on estimating the center over the features through a differentiable approach, and introduce the Cluster Attention Module and Cluster Attention Contrast task in section 3.3.

3.2 Clustering

Establish Subcategories of Normality as Clusters. For simplicity, consider the task of obtaining the center of the subcategory $j$ specific feature space of the encoder $q$, which can be expressed as $c^q_j$. The simplest idea is selecting the centroid data point of all features in current subcategory specific feature space as its center. Consider all the features in subcategory $j$ specific feature space: $\{s^q_{1j}, h^q_{2j}, \ldots, h^q_{Mj} \mid h^q_{1j} \in R^{|q|} \}$ and an alternative center $\hat{c}^q_j$, hence, the objective function can be formulated as,

$$\min \mathcal{F}(c^q_j) = \sum_{i=1}^M d(h^q_{ij}, c^q_j)$$

where $d(h^q_{ij}, c^q_j)$ means the distance between current point and the alternative center. However, simply introducing Eq.(1) into the deep networks will make it difficult to optimize by gradient descent, as the minimization function in Eq.(1) is a hard assignment of whether a data point is the center or not, which is not differentiable. Inspired by DMC[18], we transform the hard assignment in Eq.(1) to a soft assignment by adjusting clustering temperature hyperparameter $\tau_c$. In our early experiments, we found simply introducing a cosine proximity as similarity measure may make the cluster centers tend to a simple mean average of related features. Therefore, we introduce the temperature $\tau_c$ to control the softness of the assignment. Experimental results show setting a slightly harder assignment than DMC[18] can achieve better performance. Then, the minimization operation in Eq.(1) is approximatively computed as,

$$c^q_j = \sum_{i=1}^M w^q_{ij} h^q_{ij}$$

However, there remains another problem that the computation of $s^q_{ij}$ depends on the current center $c$ which makes it difficult to get the direct update rules for the centers. We choose the same way as DMC[18] to alternatively update the coefficient $s^q_{ij}$ and center $c^q_{j+1}$, Eq.(3) is modified into,

$$c^q_{j+1} = \sum_{i=1}^M w^q_{ij} h^q_{ij}$$

**Consistent Centers Use a Queue.** Considering that mini-batch gradient descent is usually used for optimization during training, which result in incoherent center representation due to the limited mini-batch size. To address this problem, we propose to use a feature queue to maintain the features from several previous batches. The introduction of the feature queue decouples the number of subcategory-specific representations involved in clustering from the mini-batch size. The new centers can benefit from the feature queue by performing clustering on more projected features. Our queue size can be much larger than a typical mini-batch size and be flexibly and independently set as a hyperparameter. We show the complete cluster center estimating process in Algorithm 1.

Avoiding Trivial Solutions. Furthermore, trivial solutions may appear when the cluster centers are not constrained. For example, in an extreme case, all cluster centers are highly similar, which is not guaranteed to obtain distinguished subcategories. This will result in all the feature spaces focus on a few patterns and lose the remaining essential information for the normal concept. To overcome this problem, we recommend attaching constraints to align the centers obtained by the two encoders. In practice, we regard the cluster centers in the encoder $q$ as the query, while the cluster centers in the encoder $k$ as the key. The encoder $q$ and encoder $k$ are designed to have the same structure, and the cluster centers obtained from the corresponding project layers in the two encoders are considered as positive pairs. Then we employ the InfoNCE[48] loss to force the $c_q$ is similar to its positive key $c_k$, and dissimilar to the other keys:

$$L_D \equiv \frac{1}{K} \sum_{i=0}^{K} -\log \frac{\exp(c^q_i \cdot c^q_j / \tau)}{\sum_{j=0}^{K} \exp(c^q_i \cdot c^q_j / \tau)}$$

Another typical trivial solution may appear in other clustering method is that the vast majority of samples are assigned to a few clusters, but did not appear in our experiments. This is possible because this type of trivial solution does not contribute to minimize the loss of the contrastive representation learning task we proposed in the next section.
Algorithm 2 Cluster Attention Contrast

Input:
- mini-batch size $N$, structure of $T$,
- structure of encoder $q$, encoder $k$.

Initialize the queues and cluster centers.

for sampled mini-batch $(x_i)_{i=1}^N$ do
  for all $i \in \{1, \ldots, N\}$ do
    draw two augmentation functions $t \sim T$, $t' \sim T$
    $x_i^t = t(x_i)$, $x_i^{t'} = t'(x_i)$
    $h_i^q = f_q(z_i^q)$, $h_i^k = f_k(z_i^k)$
    $z_i^q = g_q(h_i^q, c_q)$, $z_i^k = g_k(h_i^k, c_k)$
  end for
  update encoder network $q$ to minimize $L(W)$ in Eq. (9).
  momentum update encoder network $k$ parameters.
  update key queue and feature queues.
  update cluster centers $\{(q_j^q)_{j=1}^K, (k_j^k)_{j=1}^K\}$ use algorithm 1.
end for
return encoder network $q$

Once we have obtained the current clustering center in each feature space, we can perform Cluster Attention Contrast according to the similarity between the subcategory-specific representations and their corresponding centers.

3.3 Cluster Attention Contrast

Cluster Attention Module. In section 3.2, we have obtained the cluster centers in $K$ feature spaces, the next problem to be solved is the approach to redraw the original feature using its $K$ projected features and corresponding centers. A naive idea is to directly formalize the $K$ similarity scores between the subcategory-specific representations and corresponding centers as the final representation of each original snippet. However, this scheme is not ideal in our early experiments. We speculate that this manner may result in losing the abundant information in features and confuse our instance discrimination pretext task.

In this work, we propose to represent the original snippet-level features in a similar way as estimating the soft-assignment center in the $K$ feature spaces. We redraw the snippet-level video features through a linear combination of its $K$ projected features. The linear combination weight is evaluated by the similarity between the subcategory-specific representations and their corresponding centers. The higher the similarity score measures the larger the proportion occupies in the final representation. The above process can be formulated as,

$$w_{ij} = \frac{e^{-d_{ij} / r_z}}{\sum_{j=1}^K e^{-d_{ij} / r_z}} = \text{softmax} (s_{ij})$$

where $s_{ij} = h_i^j \cdot c_j / (r_z \| h_i^j \| \| c_j \|)$ measures the similarity between current point and the corresponding center. $r_z$ is a manually set hyperparameter. Then the linear combination of the subcategory-specific representations is regarded as our redrawn feature,

$$r_i = \sum_{j=1}^K w_{ij} h_{ij} \quad (7)$$

Finally, previous work[5] have empirically shown that using an additional projection before the contrastive losses can improve performance by suppressing the loss of information and our early experiment concluded consistent with them. Hence, we use an additional projection head $z_i = H(r_i)$ at the end of in our later experimental settings. After the redrawn features pass through the project head, we can obtain the output of the Cluster Attention Module, i.e., the cluster attention representation. We show the complete Cluster Attention Module design in Figure 3.

**Momentum Contrast Training.** Since Cluster Attention Module is the final component of our encoders, the cluster attention representation is the final output of our model. Next, follow the classic contrastive learning paradigm[5], we consider the output of encoder $q$ and the output of encoder $k$ as a positive pair if they originate from the same snippet, and otherwise as a negative sample pair. Then we employ the InfoNCE[48] loss to force the positive pairs are similar and the negative pairs are dissimilar,

$$L_C = \frac{1}{N} \sum_{i=0}^N \log \frac{\exp\left(\frac{s_{i}^q \cdot s_{ij}^k}{\tau}\right)}{\sum_{j=0}^P \exp\left(\frac{s_{i}^q \cdot s_{ij}^k}{\tau}\right)} \quad (8)$$

By incorporating the constraint on the clusters, the total loss function is:

$$L(W) = \alpha_1 L_C + \alpha_2 L_D + \|W\|_F$$ \quad (9)

where $W$ represents model weights, $\alpha_i (i = 1, 2)$ are hyperparameters to weight the losses. However, the contrastive learning paradigm commonly consumes lots of resources particularly with respect to video feature extraction. It brings great challenges to model
training. Inspired by MoCo[14], we recommend performing momentum contrast training to address our proposed Cluster Attention Contrast task. In practice, we maintain a queue during training, store the keys of the latest several optimization steps, and continuously update the queue as the training process. In addition, to maintain the key representations’ consistency in the key queue, we follow MoCo[14] to employ momentum update on the parameters of encoder k during training, which is an exponential moving average of the encoder q parameters\(^1\).

Formally, denoting the parameters of encoder q as \(\theta_q\) and parameters of encoder k as \(\theta_k\), we update \(\theta_k\) as:

\[
\theta_k \leftarrow m\theta_k + (1 - m)\theta_q
\]

where \(m \in [0, 1)\) is a momentum coefficient.

We show the complete Cluster Attention Contrast algorithm in Algorithm 2.

4 EXPERIMENTS

In this section, we evaluate our proposed method as well as the functionalities of different components on two publicly available anomaly detection datasets, including the CUHK Avenue dataset[34], and the ShanghaiTech Campus dataset[35].

4.1 Datasets and Evaluation Metric

Here we briefly introduce the datasets used in our experiments.

CUHK Avenue dataset. It contains 16 training videos and 21 testing ones with a total of 47 abnormal events, including throwing objects, loitering, and running. The size of people may change because of the camera position and angle.

The ShanghaiTech Campus dataset. ShanghaiTech Campus dataset is a challenging anomaly detection dataset. It contains 330 training videos and 107 testing ones, which is taken in 13 different scenes with various camera angles and illumination. Abnormal activities are diverse and realistic, which include appearance anomalies like bicycles, skateboards, motorbikes, cars, strollers, and motion anomalies such as fighting, chasing, pushing, jumping, etc.

Evaluation Metric. In most previous works[13, 32, 34, 35, 61], a popular evaluation metric is to calculate the Receiver Operation Characteristic (ROC) by gradually changing the threshold of regular scores. Then the Area Under Curve (AUC) is cumulated to a scalar for performance evaluation. A higher value indicates better anomaly detection performance. In this paper, following the work[13, 34, 35], we leverage frame-level AUC for performance evaluation.

4.2 Implementation Details

Data Augmentation for Contrastive Representation Learning. For data augmentation, we use: (1) random multiscale crop and resize, (2) random horizontal flip, (3) random color distortions, (4) random rotation, and (5) random grayscale for contrastive representation learning. For each training snippet which stacking 8 frames, we first uniformly oversample each video frame with the “8-crop” augment\(^2\) and resize the crops to size 256 \(\times\) 256. Then we randomly perform the data augmentations on these 8-frame-group. The data augmentations used in encoder k and encoder q are sampled from the same family of augmentations. Some examples are shown in Figure 4.

Pretrained Feature Extractor. Considering the contrastive learning paradigm commonly requires a larger mini-batch size and more training steps than supervised learning, we chose a light and fast action classifier, PAN[56], as our feature extractor. PAN[56] designed a concise motion cue called Persistence of Appearance (PA), which focuses on modeling the small displacements of motion boundaries by calculating the pixel-wise differences between two adjacent frames in the feature space. We choose BN-Inception[20] pretrained on Kinetics-400[26] as the backbone, and extract features from its global_pool layer. We freeze the weights of PA layer in all training phase. As mentioned above, all training and inference phases are performed on the snippets temporally stacking 8 frames, which means our snippet consists of two typical video segments length in PAN[56]. We strongly recommend performing layer normalization on the extracted features to improve regularization.

Project Layers. For each feature space, we employ two linear layers to project the PAN[56] features to a 128-dimensional representation. One layer directly projects the PAN[56] features to 128-dimensional by a fully-connected layer, while the other one firstly performs a timescale max pooling, then projects the feature to 128-dimensional through another fully-connected layer. Finally, we weight sum the two 128-dimensional vector through a manually set parameter as the projected representation.

Training. We train our models using an SGD[44] optimizer with the default mini-batch size of 128. The SGD weight decay is \(1e^{-4}\) and the SGD momentum is 0.9. We employ Shuffling Batch Normalization reported in [14] to avoid the information leakage caused by Batch Normalization[20]. We train models for 100 epochs with the initial learning rate of 0.015. We freeze the weights of pretrained feature extractor in early 10 epochs for warmup. Hyperparameters \(\alpha_1, \alpha_2\) were empirically set to 1, \(1e^{-3}\), respectively. The default clustering temperature \(\tau_c\) and cluster attention temperature \(\tau_x\) are both empirically set to 0.5. The best empirical setting for m in Eq.10 is 0.999. The temperature \(\tau\) of InfoNCE losses are set to 0.2 (in

\[m \to \min (0.5, m)\]
### Table 1: AUC(%) of various methods on the CUHK Avenue and ShanghaiTech Campus datasets. All methods are listed by the published year. Missing results due to the lack of reports.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CUHK Avenue</th>
<th>ShanghaiTech Campus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lu et al.[34]</td>
<td>80.9</td>
<td>-</td>
</tr>
<tr>
<td>ConvAE [13]</td>
<td>80.0</td>
<td>60.9</td>
</tr>
<tr>
<td>s-RNN [35]</td>
<td>81.7</td>
<td>68.0</td>
</tr>
<tr>
<td>STAE [29]</td>
<td>80.9</td>
<td>-</td>
</tr>
<tr>
<td>Liu et al.[32]</td>
<td>85.1</td>
<td>72.8</td>
</tr>
<tr>
<td>Liu et al.[33]</td>
<td>84.4</td>
<td>-</td>
</tr>
<tr>
<td>Ionescu et al. [22]</td>
<td>88.9</td>
<td>-</td>
</tr>
<tr>
<td>Abati et al.[1]</td>
<td>-</td>
<td>72.5</td>
</tr>
<tr>
<td>AnoPCN [55]</td>
<td>86.2</td>
<td>73.6</td>
</tr>
<tr>
<td>our method</td>
<td>87.0</td>
<td>79.3</td>
</tr>
</tbody>
</table>

### Table 2: Ablation Studies on the CUHK Avenue Dataset.

<table>
<thead>
<tr>
<th>MoCo Training</th>
<th>Cluster Attention</th>
<th>Feature Queue</th>
<th>AUC(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td>68.4</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>75.5</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>86.1</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>87.0</td>
</tr>
</tbody>
</table>

$L_C$, 1.0 (in $L_D$), respectively. Our method introduces two types of queues to help accomplish the contrastive representation learning task. The default size of the feature queue is 16384 and those of the key queue is 8192.

**Inference.** During inference, for each test samples, the highest similarity between its $K$ feature space representations and corresponding centers will be regarded as its regularity score. To maintain comparability with training data, we also oversample each video snippet with the “8-crop” augment and resize the crops to $256 \times 256$-pixel. We take the lowest regularity score evaluated on the cropped snippets as the regularity score of the original snippet. The intuitive idea is that once a position in video detects the irregularity, we believe an abnormality has occurred in the current snippet. Finally, we employ a $1-D$ Gaussian filter to temporally smooth the frame-level regularity scores.

#### 4.3 Comparison with Existing Methods

Table 1 compares our method with a series of state-of-the-art methods [1, 13, 22, 29, 32–35, 55] on the CUHK Avenue and the ShanghaiTech Campus Datasets. Since ShanghaiTech is published in 2017, some early methods have not been evaluated (or reproduced) on it. Besides, some methods, example as Ionescu et al.[22], did not report the other missing results in their paper.

On the ShanghaiTech Campus Dataset, our method establishes a new state-of-the-art performance at 79.3% frame-level AUC. Comparison with the existing deep reconstruction methods and deep prediction methods, our work provides an absolute gain of +5.7% in terms of frame-level AUC, which demonstrating the improvements in our work is appreciable. Notably, Ionescu et al.[21] applying a pretrained SSD detector[31] on each frame, and implement their method on detected objects. For fairness, our method did not compare with their approach.

On the CUHK Avenue dataset, our method achieves comparable performance with the state-of-the-art methods. Our method has a significant improvement over the existing methods based on establishing normal behavior dictionaries, which indicates the effectiveness of our introduced contrastive learning pretext task. Our method is slightly inferior to the work of Ionescu et al.[22]. They employ a two-stage algorithm based on k-means clustering and one-class Support Vector Machines (SVM) [45] to eliminate outliers. In contrast, our method does not have additional SVM classifiers and achieves comparable performance to their reported results.

#### 4.4 Discussion

**Ablation Study.** We conduct ablation study of each component of our method on the CUHK Avenue Dataset. Table 2 present the ablation results. These experimental results demonstrate the method we proposed can significantly improve the performance. Firstly,

![Figure 5: Regularity score curves evolved along time on four test samples. Light red shaded regions represent ground truth temporal segments of the abnormal events.](image-url)
without any components present in Table 2, the anomaly detection performance fails at a frame-level AUC round to 0.68. Figure 6 and the second column in Table 2 indicate the effect of the number of $K$ feature spaces on the performance. The number of $K = 1$ (or without Cluster Attention) means we do not perform the Cluster Attention process but directly use a single projected representation for instance discrimination pretext task. These experimental results support our motivation for employing multi-paral lel projection layers to capture distinct subcategories of normality. Besides, when the number of $K$ feature spaces gradually increases with an additional constraint on the clusters, the improvement of performance validates the effectiveness of our proposed cluster constraint. Finally, it is obvious that increasing the size of the feature queue can improve the quality of clustering, since more samples lead to estimating a more reliable centroid. The ablation result involved in the feature queue concluded consistent with this proposition.

Visualization and Analysis, Figure 5 depicts the frame-level regularity score curves of our method on four test videos. The orange curve represents the regularity score obtained by our model, while the light red shaded regions represent ground truth temporal segments of the abnormal events. Figure 5 indicates that the regularity scores obtained by our model are strongly correlated with the ground truth on these samples, which demonstrates the effectiveness of our approach. Figure 7 shows some of the failed cases of our method on the ShanghaiTech Campus Dataset. We noticed that a considerable number of our failed cases contain abnormal objects away from the camera, which reveals that our method requires to explore improving the performance of detecting small abnormal objects.

5 CONCLUSION

In this paper, we explore addressing video anomaly detection task by a contrastive representation learning pretext task. We introduce a novel instance discrimination pretext task, Cluster Attention Contrast, to obtain reliable subcategories of normality as clusters. Experiments on benchmark datasets demonstrate our approach is effective and interpretable. The Cluster Attention Contrast task can help to establish reliable and discriminate feature spaces for video anomaly detection. By utilizing the similarity between subcategory-specific representations and corresponding centers, the proposed method achieves state-of-the-art performance on benchmark datasets. In this manner, we established a robust normal concept without any prior assumptions on reconstruction errors or prediction errors. In the future, we would like to explore the following directions to improve this work. Firstly, we will explore to improve the performance of detecting small abnormal objects. Furthermore, we will design a more reliable method for determining the number of $K$ feature spaces in encoders automatically. Finally, we will explore different distance measures to obtain a more suitable one for Cluster Attention Contrast task.

ACKNOWLEDGMENTS

This paper was partially supported by the IER foundation (No. HT-JD-CXY-201904) and Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing). Special acknowledgments are given to Aoto-PKUSZ Joint Lab for its support.

REFERENCES
