Graph-Jigsaw Conditioned Diffusion Model for Skeleton-based Video Anomaly Detection

Ali Karami[1,2], Thi Kieu Khanh Ho[1,2], Narges Armanfard[1,2]
[1]Department of Electrical and Computer Engineering, McGill University
[2]Mila - Quebec AI Institute, Montreal, QC, Canada
{ali.karami, thi.k.ho, narges.armanfard}@mail.mcgill.ca

Abstract

*Skeleton-based video anomaly detection (SVAD) is a crucial task in computer vision. Accurately identifying abnormal patterns or events enables operators to promptly detect suspicious activities, thereby enhancing safety. Achieving this demands a comprehensive understanding of human motions, both at body and region levels, while also accounting for the wide variations of performing a single action. However, existing studies fail to simultaneously address these crucial properties. This paper introduces a novel, practical, and lightweight framework, namely **G**raph-**Ji**gsaw **C**onditioned **Di**ffusion Model for **S**keleton-based Video **A**nomaly **D**etection (GiCiSAD) to overcome the challenges associated with SVAD. GiCiSAD consists of three novel modules: the Graph Attention-based Forecasting module to capture the spatio-temporal dependencies inherent in the data, the Graph-level Jigsaw Puzzle Maker module to distinguish subtle region-level discrepancies between normal and abnormal motions, and the Graph-based Conditional Diffusion model to generate a wide spectrum of human motions. Extensive experiments on four widely used skeleton-based video datasets show that GiCiSAD outperforms existing methods with significantly fewer training parameters, establishing it as the new state-of-the-art.*

1. Introduction

Skeleton-based video anomaly detection (SVAD) is an important task in computer vision and video surveillance [7, 8, 27, 29, 31, 37, 50]. It refers to a task of identifying abnormal behaviors or motions that deviate from the typical patterns observed in normal activities. Unlike conventional video anomaly detection (VAD) [33, 38, 45, 49, 51], SVAD involves skeleton-based representations, which focus on the key joints and their connections, effectively capturing more concise and essential information of human activities in video sequences, while reducing the computational complexity compared to the pixel-level analysis. However, SVAD datasets pose several critical challenges for anomaly detection algorithms.

First, skeleton-based video data is inherently a time-series data, which exhibits *spatio-temporal dependencies* [15, 35]. In essence, spatial dependencies signify the relationships among skeleton joints within a frame such as body posture, gestures, and interactions. Meanwhile, temporal dependencies are represented by the temporal evolution of skeletal motions that capture the dynamics of human activities over time. Understanding these spatio-temporal dependencies is crucial for distinguishing normal and abnormal motions. For example, deviations from expected spatial arrangements or sudden changes in joint trajectories over time may indicate potential anomalies. Therefore, by analyzing the spatio-temporal evolution of skeletal joints, anomaly detection algorithms can attain a semantic understanding of human activities. Recently, graph-based approaches [6, 13–15, 17, 20, 39] have gained significant attention in time-series data due to their capabilities of dynamically learning graphs to effectively capture both types of dependencies, making them well-suited for SVAD tasks. Although several graph-based studies have been conducted for SVAD [23, 27, 29], none of them have dynamically learned the evolving relationships between joints, which are essential for capturing the dynamic nature of human activities.

Second, subtle differences between normal and abnormal actions can oftentimes be localized to specific regions of the body rather than affecting the entire body. This is while all existing SVAD methods are based on modeling the human body as a whole and ignore the importance of such local variations when detecting anomalies [7, 18, 41, 44]. In the presence of a localized anomaly, these holistic-based models tend to classify the activity as normal since the majority of the body regions are acting normally except for a small region. For instance, consider a scenario where a person is walking normally, but their arm exhibits abnormal movements due to injury. We refer to this issue as *region-*

specific discrepancies. Recently, self-supervised learning (SSL) [4, 16, 22, 24, 40, 52] has emerged as a promising research direction for VAD. Unlike unsupervised methods, which learn directly from unlabeled data to identify patterns indicative of anomalies, SSL goes a step further by defining pretext tasks that encourage the model to focus on region-level features [49]. While SSL has been widely used in the context of the image domain [5,30,46,47], it remains unanswered how to adapt this approach to the field of SVAD, particularly considering the presence of skeleton data instead of traditional images in this context.

Third, when dealing with skeleton-based video data, it is essential to acknowledge that there are *infinite variations* of performing both normal and abnormal actions [7, 32]. In other words, both normal and abnormal behavior can be complex and multifaceted, encompassing a wide range of actions, gestures, and interactions. While some studies [8,27,29,31] focused on generating a single reconstruction of the input data, these approaches often fail to capture the wide spectrum of human motions. Moreover, while recent research has addressed the diversity of both normal and abnormal activities [7], by considering the body as a whole, they overlook the fact that abnormalities may be localized to only specific regions of the body, potentially leading to misdetection in cases where anomalies occur in isolated regions while the rest of the body remains normal. [1]

Recognizing the critical importance of spatio-temporal dependencies, region-specific discrepancies, and infinite variations inherent in skeleton-based video data, in this paper, we propose **G**raph-**Ji**gsaw **C**onditioned **Di**ffusion Model for **S**keleton-based Video **A**nomaly **D**etection - hereafter GiCiSAD. Essentially, GiCiSAD includes three novel modules. The Graph Attention-based Forecasting module leverages a graph learning strategy to effectively capture the spatio-temporal dependencies. To address the issue of region-specific discrepancies, we propose a novel graph-level SSL with a difficult pretext task called Jigsaw puzzles [28, 49]. We name this module Graph-level Jigsaw Puzzle Maker, which involves various subgraph augmentations applied to the learnable graph, hence providing supervisory signals to help GiCiSAD capture a slight region-level difference between normal and abnormal behaviors. Lastly, to address the infinite variations, GiCiSAD incorporates a newly proposed diffusion-based model called the Graph-level Conditional Diffusion Model, which utilizes the learned graph from past frames as conditional information to generate diverse future samples.

In summary, our contributions are as follows:

- The first study in the SVAD field that presents a unified framework for effective tackling of the challenges posed by the spatio-temporal dependencies, region-specific discrepancies, and infinite variations inherent in skeleton-based video data.
- A novel graph attention-based approach dynamically learns dependencies between body joints across spatial and temporal dimensions.
- The first graph self-supervised learning study aimed at achieving more discriminative region-level understanding in SVAD.
- A novel graph-level conditional diffusion model to facilitate generating diverse future motion patterns, leveraging past motions as a guide.
- A thorough validation on four widely used SVAD datasets showcases our superior anomaly detection performance and a remarkable 40% reduction in training parameters compared to state-of-the-art (SOTA).

2. Proposed Method

We define a skeleton-based video dataset that consists of human poses over time as $X = \{\mathbf{x}_{(i)}\}_{i=1}^{N}$, where $\mathbf{x}_{(i)} = (x_{(i)}^1, x_{(i)}^2, \ldots, x_{(i)}^K)$ is the ith observation in the frame sequences of N observations, $\mathbf{x}_{(i)} \in \mathbb{R}^{K \times L}$. K and L, respectively, denote the number of joints and the number of frames in the ith observation. An observation can be conceptualized as a sliding window of size L. We further divide $\mathbf{x}_{(i)}$ into two parts: the first (past) frames $\mathbf{x}_{(i)}^{1:l}$ and the latter (future) frames $\mathbf{x}_{(i)}^{l+1:L}$, where l regulates the size of past and future frames. For simplicity, we present $\mathbf{x}_{(i)}^{1:l}$ and $\mathbf{x}_{(i)}^{l+1:L}$ as $\mathbf{x}^-$ and $\mathbf{x}^+$, respectively. Our task is to detect the frames with abnormal poses in the test data by training the model with only normal data.

The block diagram of the proposed GiCiSAD method, depicting the three novel modules, namely Graph Attention-based Forecasting, Graph-level Jigsaw Puzzle Maker, and Graph-based Conditional Diffusion Model, is shown in Fig. 1. Details of each module are presented below. For the sake of completeness, we provide the pseudocode in Section B of the Supplementary Material.

2.1. Graph Attention-Based Forecasting

In this section, our objective is to capture both the spatial dependencies between joints within each frame and the temporal dependencies across frames. This is to provide a model that is more sensitive to capture the holistic structure of the body. We achieve this by constructing a graph for the past frames, i.e., $\mathbf{x}^-$, and performing the task of forecasting future frames, i.e., $\mathbf{x}^+$, based on the past. We first represent $\mathbf{x}^-$, as a graph denoted as $\mathcal{G} = \{\mathbf{H}, \mathcal{A}\}$, where $\mathbf{H} \in \mathbb{R}^{D \times K}$ is the representation matrix of all K nodes

[1] We present the Related Works section in Section A of the Supplementary Material.

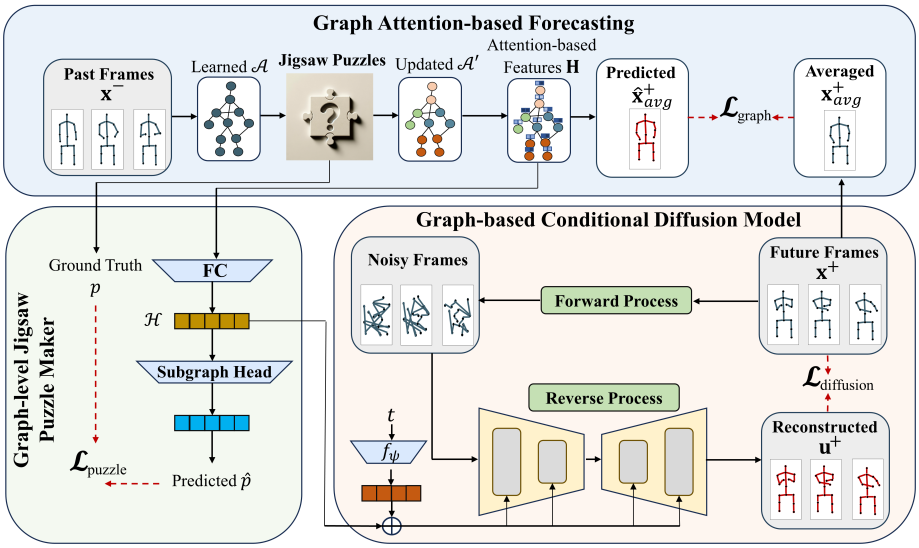

Figure 1. The overall framework of GiCiSAD.

(joints), D is a hyperparameter defining the feature dimension, and $\mathcal{A} \in \mathbb{R}^{K\times K}$ is the adjacency matrix that encodes the relationship between joints. We learn $\mathbf{H}$ and $\mathcal{A}$ as below.

Starting with $\mathcal{A}$, in the context of human motion analysis, different joints have different characteristics, and these characteristics can be related in complex ways. It is crucial to represent each joint in a flexible way that captures the different underlying patterns. We do this by introducing a feature vector for each joint. For instance, the feature vector for the kth joint is denoted as $\mathbf{v}_k \in \mathbb{R}^D$, $k \in \{1, 2, \ldots, K\}$. Note that these feature vectors are initialized randomly and then trained along with the rest of the model. To this end, we construct $\mathcal{A}$, where each element $\mathcal{A}_{kn}$ represents the relationship between the feature vectors $\mathbf{v}_k$ and $\mathbf{v}_n$ of the kth joint and the nth joint, respectively. To learn $\mathcal{A}$, we first compute the cosine similarity [6] between the kth node's feature vector and the feature vectors of its candidates named $\mathcal{C}_k$, i.e., all K nodes, excluding k as below:

$$\text{Sim}_{kn} = \frac{\mathbf{v}_k^\top \cdot \mathbf{v}_n}{\|\mathbf{v}_k\| \cdot \|\mathbf{v}_n\|} \quad \text{for } n \in \mathcal{C}_k \tag{1}$$

Then, for each node k, we select the Topδ highest values of cosine similarities in its candidates and consider that the kth node is connected to those nodes. Hence, each node has δ connections. δ is a hyperparameter defining the sparsity level based on specific applications at hand. This process is shown as below:

$$\mathcal{A}_{kn} = \begin{cases} 1 & \text{if } n \in \text{Top}\delta(\{\text{Sim}_{kn'} : n' \in \mathcal{C}_k\}) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Note that the connection from node k to node n indicates that the feature vector of node k is used for modeling the behavior of node n. It is worth mentioning that we use a *directed* graph since the dependency patterns between nodes are not necessarily symmetric.

Next, we learn $\mathbf{H}$ by a graph-based attention mechanism [6], which leverages $\mathcal{A}$. However, $\mathcal{A}$ is first fed into the Graph-level Jigsaw Puzzle Maker module, and the output of this module is a permuted version of the adjacency matrix, namely $\mathcal{A}'$. This modified $\mathcal{A}'$ serves as the input for the graph-based attention mechanism. Further details regarding the Graph-level Jigsaw Puzzle Maker module is provided in Sec. 2.2. Essentially, the graph-based attention mechanism is to integrate a node's information with that of its neighbors, guided by the learned and permuted graph structure, i.e., $\mathcal{A}'$. We define the attention coefficient between two nodes k and n as α_{kn} as below:

$$\alpha_{kn} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{s}^\top(\mathbf{g}_k \oplus \mathbf{g}_n)\right)\right)}{\sum_{j\in\mathcal{N}(k)\cup\{k\}} \exp\left(\text{LeakyReLU}\left(\mathbf{s}^\top(\mathbf{g}_k \oplus \mathbf{g}_j)\right)\right)}, \tag{3}$$

where $\mathbf{g}_k = \mathbf{v}_k \oplus \mathbf{W}\mathbf{x}_k^-$, $\mathbf{W} \in \mathbb{R}^{D\times l}$ is a trainable weight matrix applied to every node, $\mathbf{x}_k^- \in \mathbb{R}^l$ is the kth node's input value over the past frames, $\oplus$ denotes concatenation, $\mathcal{N}(k) = \{n|\mathcal{A}'_{kn} > 0\}$ is the set of neighbors of the node k obtained from $\mathcal{A}'$, and the vector of learned coefficients is denoted by $\mathbf{s}$.

Then, we obtain the representation vector for the kth node as below:

$$\mathbf{h}_k = \text{ReLU}\Big(\alpha_{k,k}\mathbf{W}\mathbf{x}_k^- + \sum_{n\in\mathcal{N}(k)} \alpha_{k,n}\mathbf{W}\mathbf{x}_n^-\Big). \tag{4}$$

We then element-wise multiply (denoted as $\odot$) the representation vector of each node, i.e., $\mathbf{h}_k$, with its corresponding feature vector, i.e., $\mathbf{v}_k$, and the output will be fed into stacked fully-connected layers, i.e., $f(\theta)$, with the output dimension of K to predict an average of future frames, denoted as $\hat{\mathbf{x}}^+_{\text{avg}}$. We observed that computing the graph loss on the average of temporal frames or per-frame yields comparable performance. We chose the average approach as it benefits both speed and smoothness of convergence. Note that the graph loss serves as additional supervision to facilitate the training of the conditioning network, essential as the training of the diffusion model relies on a well-trained conditioning. $\hat{\mathbf{x}}^+_{\text{avg}}$ is calculated as below:

$$\hat{\mathbf{x}}^+_{\text{avg}} = f_\theta\big(\mathbf{v}_1 \odot \mathbf{h}_1, \ldots, \mathbf{v}_K \odot \mathbf{h}_K\big) \tag{5}$$

We aim to capture the holistic structure of the body by optimizing $\mathcal{L}_{\text{graph}}$ denoted in Eq. (6), which involves utilizing the Mean Square Error between the predicted $\hat{\mathbf{x}}^+_{\text{avg}}$ and the actual average of future frames, denoted as $\mathbf{x}^+_{\text{avg}}$.

$$\mathcal{L}_{\text{graph}} = \big\|\hat{\mathbf{x}}^+_{\text{avg}} - \mathbf{x}^+_{\text{avg}}\big\|_2^2 \tag{6}$$

The representation matrix $\mathbf{H}$ of the past frames is the combination of the representation vectors of all nodes as $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K\}$. We then project $\mathbf{H}$ to a fully connected layer (FC) and output $\mathcal{H} \in \mathbb{R}^D$, which is used to solve the task of the Graph-level Jigsaw Puzzle Maker module described in the next section. Simultaneously, $\mathcal{H}$ is also used as the conditioning signal for the Graph-based Conditional Diffusion Model, described in Sec. 2.3.

2.2. Graph-level Jigsaw Puzzle Maker

In self-supervised learning, the quality of pseudo-labeled data plays a pivotal role in the effectiveness of the learning process. It is essential to curate the pseudo-labeled data that is neither ambiguous nor too easy for the model to solve [36, 49]. As mentioned earlier in the Introduction, subtle differences between normal and abnormal actions can oftentimes be localized to specific regions of the body rather than affecting the entire body. Therefore, we introduce a novel graph-based Jigsaw puzzle-solving approach as a self-supervised learning method, shown in Fig. 2. Note that while the Jigsaw puzzle-solving task has been widely used in the context of the image domain [5, 30, 46, 47], our adaptation marks a pioneering step into the realm of graphs.

To represent the notation of body regions, we initially partition the graph $\mathcal{A}$, learned in Sec. 2.1, into subgraphs. Here, the entire graph corresponds to the entire body, while each subgraph corresponds to an individual body region. However, subgraph identification itself is very challenging as we aim to extract subgraphs that are as distinct as possible from each other while maintaining a close relationship among nodes within the same subgraph. To this end, we employ a subgraph extraction algorithm, namely the Girvan-Newman algorithm [11] to extract η subgraphs from the adjacency matrix $\mathcal{A}$. Note that these subgraphs may not have the same size, i.e., they could have different numbers of nodes. Then, we select two of these η subgraphs randomly and swap their nodes and connections, creating a perturbed version of the adjacency matrix called $\mathcal{A}'$. However, unlike traditional image-level puzzle-solving approaches, shuffling graph-level puzzles is very challenging and requires careful consideration.

First, while in the image-level Jigsaw puzzling approach, the definition of each puzzle remains constant across all images, our task involves learning the adjacency matrix that evolves over time (see Eq. (2)). As a result, the subgraphs that we identify may change dynamically over time. Additionally, the number of nodes within each subgraph may vary, making it challenging to shuffle the positions of subgraphs. Unlike image-level puzzles where shuffling merely permutes the locations of puzzle pieces, in graph-level puzzles, shuffling two subgraphs of different sizes, alters the intra-connections within the bigger subgraph. For example, as illustrated in Fig. 2, the larger teal subgraph and the smaller pink subgraph are selected to be shuffled. Upon shuffling, while the pink subgraph maintains its internal connections, the connections within the teal subgraph undergo significant changes, i.e., nodes #1 and #2 are no longer connected to node #3 after shuffling. This is due to the fact that nodes #9 and #11 were not connected to node #3 before shuffling; hence after shuffling, nodes #1 and #2 would not be connected to node #3 as well. Note that after shuffling, nodes #1, #2, and #3 are still considered part of the same subgraph, which deviates the original definition of subgraphs, as they lack a strong internal connection within the subgraph. This stands as an example of needing an effective subgraph shuffling strategy.

Therefore, we present an effective shuffling mechanism. In this process, upon the random selection of two subgraphs, we initiate the shuffling procedure by swapping the most densely connected node in each subgraph (e.g., swap node #2 in the teal subgraph with node #11 in the pink subgraph). Here, we determine a node's density by counting the number of intra-connections that it has with other nodes in the same subgraph (e.g., node #2 has two connections). As per our graph definition in Sec. 2.1, all nodes possess δ connections; we exclusively consider intra-subgraph connections when computing node density. This avoids the trivial solution where all nodes have the same density, which is equal to δ. The process continues by swapping the next most dense node in each subgraph with each other until the last node is shuffled. The rationale behind prioritizing the shuffling of the most dense nodes is to increase the possibility that they stay connected after shuffling as well. This is due to the understanding that these dense nodes play a crucial role

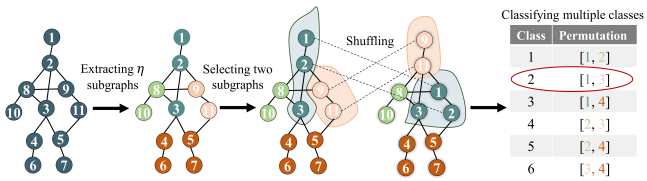

Figure 2. The overview of the graph-level Jigsaw puzzle-solving approach. Nodes with the same color formulate a subgraph. Note that although each node is required to have δ connections, for better visualization, this property is not strictly maintained in the figure.

in preserving the overall structure of subgraphs.

After shuffling, the model is tasked with a multi-class classification problem where each possible permutation is considered as a class. Our objective is to identify which of the two subgraphs have been shuffled. To this end, we use $\mathcal{H}$, obtained from the first module, project it into the subgraph head, which is a fully-connected layer in our implementation, with the output size of the number of classes, and obtain $\hat{p}$. Hence, the loss of the Graph-level Jigsaw Puzzle Maker module is the cross-entropy (CE) described below:

$$\mathcal{L}_{\text{puzzle}} = \sum_{z=1}^{\binom{\eta}{2}} \text{CE}(p_z, \hat{p}_z), \tag{7}$$

where p_z and $\hat{p}_z$ are the ground truth of the selected class and the predicted probability of each class, respectively.

This task forces the model to develop a better understanding of how each body region contributes to the overall normal behavior. Indeed, each class of shuffling can be interpreted as a form of structural augmentation applied to normal data. While these augmentations may be viewed as variations in the normal structure, which our proposed Graph Attention-based Forecasting module can potentially learn, we aim to classify these structural augmentations through $\mathcal{L}_{\text{puzzle}}$ as well, to repel the latent space of each augmentation from others. Thus, by jointly optimizing both $\mathcal{L}_{\text{graph}}$ and $\mathcal{L}_{\text{puzzle}}$, our model not only learns from the additional structural augmentations applied to normal data but also learns the latent space for each augmentation that is compact and well-separated from other latent spaces of augmentations. This allows the reserve of the inter-augmentation spaces for potential abnormal samples that could be observed in the test phase. In essence, during testing, if a sample does not align well with any of the learned augmentation regions, it may be indicative of an anomaly.

2.3. Graph-based Conditional Diffusion Model

As mentioned in the Introduction, there are infinite variations of normal and abnormal behavior in skeleton-based video data. Put differently, there are infinitely many normal and anomalous ways of executing an action, each characterized by subtle variations in movement, timing, and context. Conventional approaches [27, 29, 31] that aim to learn a single reconstruction of the input or a singular representation of normal actions would not be able to capture the wide spectrum of human motions. To address this issue, in this module, we propose to utilize diffusion-based techniques [12, 49], which can generate diverse samples from noise. This provides a more comprehensive exploration of the data space, making anomaly detection algorithms more robust to variations in the normal data during the training phase, as well as both normal and abnormal actions in the inference phase. The background on diffusion models is provided in Section C of the Supplementary Material.

Our objective is to generate a diverse set of reconstructions of the noisy future frames conditioned on $\mathcal{H}$ learned from the past frames (see Sec. 2.1). Essentially, a diffusion model incorporates two Markov chains: a forward process and a reverse process. Given an original clean motion sequence, which is future frames $\mathbf{x}^+$, the forward diffusion process incrementally corrupts the coordinates of the joints over a predefined number of steps T, making them indistinguishable from a pose with spatial coordinates of joints sampled at random. We sample the noise as $\epsilon^{l+1:L} \in \mathbb{R}^{(L-l)\times K}$ from a normal distribution $\mathcal{N}(0, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. For simplicity, we represent $\epsilon^{l+1:L}$ as ϵ. The magnitude of the added noise depends on a variance scheduler $\beta_t \in (0, 1)$, which controls the quantity of noise added at the t-th diffusion step.

The reverse diffusion process, which is the focus of our model's learning, is defined by its capability to reconstruct the original future frames from their noised versions. To estimate the noise, we employ a U-Net-shaped stack of STS-GCN [43] layers, which is capable of capturing the spatio-temporal dependencies of human joints. We train the network conditioned on the embedded multi-layer perception (MLP) of the diffusion step, i.e., $f_\psi(t)$, and $\mathcal{H}$. Inspired by [7, 42], the loss is derived as:

$$\mathcal{L}_{\text{noise}} = \mathbb{E}_{\mathbf{x}^+,\epsilon,t}\Big[||\epsilon - \epsilon_\psi(\mathbf{x}_t^+, f_\psi(t), \mathcal{H})||\Big], \tag{8}$$

where $\mathbb{E}$ denotes the expectation, and the term $\epsilon_\psi(\mathbf{x}_t^+, f_\psi(t), \mathcal{H})$ denotes the U-Net model's estimation of the added noise at the t-th time step.

Following [10], to have a more stable optimization process, we obtain the loss of the Graph-based Conditional Diffusion Model by smoothing $\mathcal{L}_{\text{noise}}$ as follows:

$$\mathcal{L}_{\text{diffusion}} = \begin{cases} 0.5 \cdot (\mathcal{L}_{\text{noise}})^2 & \text{if } |\mathcal{L}_{\text{noise}}| < 1 \\ |\mathcal{L}_{\text{noise}}| - 0.5 & \text{otherwise} \end{cases} \tag{9}$$

Thus, the total loss in the training phase is:

$$\mathcal{L} = \lambda_1(\mathcal{L}_{\text{graph}} + \lambda_2\mathcal{L}_{\text{puzzle}}) + \mathcal{L}_{\text{diffusion}}, \quad (10)$$

where λ_1 and λ_2 are hyperparameters defined to weigh the importance of each module.

During the inference phase, GiCiSAD predicts future motion frames based on the observed past frames. The process starts from a completely noised state $\mathbf{u}_T^+$, randomly drawn from $\mathcal{N}(0, \mathbf{I})$ and continues by iteratively computing $\mathbf{u}_{t-1}^+$ from $\mathbf{u}_t^+$ for $t = T, T-1, \ldots, 1$. The reverse process is then presented as:

$$\mathbf{u}_{t-1}^+ = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{u}_t^+ - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\psi(\mathbf{u}_t^+, f_\psi(t), \mathcal{H})\right) + \xi\sqrt{\beta_t}, \quad (11)$$

where $\bar{\alpha}_t = \prod_{\gamma=T}^{t}(1-\beta_\gamma)$ and $\xi \sim \mathcal{N}(0, \mathbf{I})$.

Motivated by infinite variations of performing both normal and abnormal actions, we generate M diverse sets of future frames, i.e., $\mathbf{u}_{(1)}^+, \mathbf{u}_{(2)}^+, \ldots, \mathbf{u}_{(M)}^+$. For each generation m, where $m \in M$, we compute the reconstruction error, i.e., $\mathcal{S}_m = \mathcal{L}_{\text{diffusion}}(|\mathbf{x}^+ - \mathbf{u}_{(m)}^+|)$ and consider it as the anomaly score. To aggregate anomaly scores of all M generations, we consider three strategies: the mean, the median, and the minimum distance selector. In the mean and median approaches, we derive either the mean or the median of all M scores and allocate this value to the respective frame to evaluate its anomaly level. Regarding the minimum distance selector approach, the lowest anomaly score among all scores is assigned to the respective frame. Note that in the case of more than one actor performing in the scene, the average anomaly score over all actors is assigned to those frames. Our experiments demonstrate that the minimum distance strategy consistently yields the best results, which is consistent with the findings shown in [7]. Details of the experimental results for other aggregation strategies are shown in Section D of the Supplementary Material.

3. Experiments

3.1. Experimental Settings

Datasets. We use four widely used datasets in SVAD, namely Human-Related (HR) versions of the ShanghaiTech Campus (HR-STC) [26], HR-Avenue [25], UBnormal [1] and HR-UBnormal [8]. HR-STC consists of 13 scenes recorded by different cameras. It contains a total of 303 training videos and 101 test videos with 130 anomalous events. HR-Avenue is comprised of 16 training and 21 test videos with a total of 47 anomalous events. UBnormal comprises 29 scenes generated from 2D natural images using Cinema4D software with 186 normal training and 211 test videos that include 22 categories of anomalies. We also evaluate our model on a subset of UBnormal, called HR-UBnormal, focusing on only human-related anomalies. This excludes 2.32% frames, which are non-human-related anomalies, from the test set.

Implementation Details. To ensure a fair comparison with our latest competitors [7, 44], we employ a window size of six frames (i.e., $L = 6$) for all the experiments, where the first three frames (i.e., $l = 3$) are used as the past frames inputted into the Graph Attention-based Forecasting module, while the subsequent three frames are used as the future frames and are fed into the Graph-based Conditional Diffusion Model. For the Graph Attention-based Forecasting module, hyperparameters D and δ are set to 16 and 5, respectively. The hidden layer in the output prediction of the Graph Attention-based Forecasting module has 128 neurons. For the Graph-based Conditional Diffusion Model, we set $\beta_1 = 1e^{-4}$ and $\beta_T = 0.01, T = 10$ and apply the cosine variance scheduler from [34]. Inspired by [7], our U-Net reduces the number of joints from 17 to 10 and changes the channels from 2 to $(32, 32, 64, 64, 128, 64)$. The diffusion time steps are encoded using the encoding mechanism described in [48]. We assign values of $\lambda_1 = 0.01$ and $\lambda_2 = 1$. Adam optimizer [21] is utilized with a learning rate set at 10^{-4}. The batch size of the HR-Avenue dataset is 1024 and 2024 for the other datasets. Note that for a fair comparison, the above hyperparameters are fixed in all our experiments and are the same across all datasets.

Baselines. We compare GiCiSAD against the most recent SOTA methods from the literature, including GEPC [29], PoseCVAE [18], STGCN-LSTM [23], COSKAD [8], MocoDAD [7], and TrajREC [44]. Details of the baselines are given in Section E of the Supplementary Material.

Evaluation Metric. Following the common practice in the VAD field [1, 2, 7, 18, 19, 23], we report the Receiver Operating Characteristic Area Under the Curve (AUROC) to assess the performance of our proposed GiCiSAD method.

3.2. Comparison with State-of-The-Art

The performances of GiCiSAD and existing methods are summarized in Tab. 1. GiCiSAD demonstrates superior performance with the AUROC scores of 78.0, 89.6, 68.8, and 68.6 on HR-STC, HR-Avenue, HR-UBnormal, and UBnormal datasets, respectively. GiCiSAD outperforms all existing methods, including the most recent competitors, MoCoDAD and TrajREC. This can be attributed to GiCiSAD's comprehensive approach in addressing critical challenges prevalent in SVAD. As outlined in the Introduction, skeleton-based video data poses several critical challenges, including spatio-temporal dependencies, region-specific discrepancies, and infinite variations. While previous methods have attempted to tackle individual aspects of these challenges, none have provided a comprehensive solution. For instance, prior methods such as [8, 23, 27, 29] have leveraged graph-based models to capture spatio-temporal dependencies, yet they failed to take into account other chal-

Method	Venue	HR-STC	HR-Avenue	HR-UBnormal	UBnormal
GEPC [29]	CVPR 2020	74.8	58.1	55.2	53.4
PoseCVAE [18]	ICPR 2021	75.7	87.8	-	-
STGCN-LSTM [23]	Neurocomputing 2022	77.2	86.3	-	-
COSKAD [8]	arXiv 2023	77.1	87.8	65.5	65.0
MoCoDAD [7]	ICCV 2023	77.6	89.0	68.4†	68.3†
TrajREC [44]	WACV 2024	77.9†	89.4†	68.2	68.0
Ours	-	**78.0**	**89.6**	**68.8**	**68.6**

Table 1. Comparison between existing methods and GiCiSAD. The best and second-best AUROC scores are denoted in bold and †.

Method	Params	AUROC
AED-SSMTL◇ [9]	>80M	61.3
TimeSformer◇ [3]	121M	**68.6**
TrajREC	4.9M	68.0
MoCoDAD	142K†	68.3†
GiCiSAD	**82.6K**	**68.6**

Table 2. Comparison between GiCiSAD and existing methods in terms of AUROC on the UBnormal dataset and the number of training parameters (Params). ◇ denotes the supervised methods. The best and second-best results are denoted in bold and †.

lenges, such as region-specific discrepancies, and infinite variations. Similarly, while [7] effectively tackled the issue of infinite variations through a conditional diffusion-based model, it neglected others. Moreover, none of the existing methods have explicitly addressed the challenge of region-specific discrepancies. In contrast, GiCiSAD stands out as it systematically addresses all these challenges through three novel proposed modules, each specifically designed for a specific issue, leading to improved detection performance.

3.3. Parameter Efficiency

Tab. 2 presents a comparative analysis of GiCiSAD and the most recent unsupervised competitors, i.e., MoCoDAD and TrajREC, as well as two recently developed supervised methods, i.e., AED-SSMTL and TimeSformer. Note that supervised methods have access to both normal and abnormal data during the training phase, while unsupervised methods have only access to the normal data in their training phase. The results show that our method outperforms the unsupervised methods from the anomaly detection accuracy point of view. Also, our model shows a significant reduction of the number of parameters — up to 40% less than the most parameter-efficient unsupervised method, MoCoDAD. Moreover, compared with the supervised methods, we achieve comparable performance in terms of AUROC, with only a fraction of the parameters of their models.

3.4. Ablation Study

We assess the effectiveness of GiCiSAD through ablation studies on HR-Avenue and HR-STC datasets considering four key aspects of GiCiSAD: (1) effectiveness of individual components, (2) effectiveness of conditioning mechanism, (3) types of graph-based Jigsaw puzzles, and (4) the number of subgraphs. For simplicity, we refer to Graph Attention-based Forecasting, Graph-based Jigsaw Puzzle Maker, and Graph-based Conditional Diffusion Model as `Graph`, `Puzzle`, and `Diffusion`, respectively.

Effectiveness of Individual Components. As described in our inference phase, GiCiSAD utilizes anomaly scores derived from the reverse process of `Diffusion`; hence, its inclusion is essential for anomaly detection. Therefore, in this experiment, we conduct ablation studies specifically focusing on the exclusion of either `Graph` or `Puzzle`. The results are shown in Tab. 3. The sign "+" denotes the inclusion of a component. Results show that each component in our model plays a crucial role in improving its performance. For instance, in the scenario where we only include `Graph` and `Diffusion`, our model achieves a 77.4% AUROC on HR-STC, which proves our hypothesis that despite the capability of capturing the overall spatio-temporal dependencies inherent in skeleton-based data, `Graph` itself lacks a deeper understanding of the impact of each region of the human body. Conversely, if we include `Puzzle` and `Diffusion`, our model can capture the region-specific discrepancies, yet it is not able to understand the overall nature of human actions and thus, adding another objective that stands for capturing the overall normal structure of the body is needed. Finally, when we include all components together, our model achieves the best result with an AUROC of 78.0% as it addresses all the challenges comprehensively.

Effectiveness of Conditioning Mechanism. In this study, we experiment with three different conditioning strategies to be used in `Diffusion`, including our proposed *Graph*-based, introduced in Sec. 2.1, *Encoder*-based and *AutoEncoder*-based conditioning mechanisms. The architecture of the two latter ones is borrowed from our main competitor, MoCoDAD [7]. The objective is to evaluate

GiCiSAD	HR-Avenue	HR-STC
`Graph` + `Diffusion`	88.2	77.4
`Puzzle` + `Diffusion`	87.9	77.2
`Graph` + `Puzzle` + `Diffusion`	**89.6**	**78.0**

Table 3. The performance of individual components and their combination in GiCiSAD.

Conditioning Mechanism	HR-Avenue	HR-STC	Params
Encoder-based	83.9	74	111.1K
AutoEncoder-based	86.8	76.6	142.3K
Graph-based	**88.2**	**77.4**	**82.6K**

Table 4. The performance of different conditioning mechanisms for `Diffusion`.

	HR-Avenue	HR-STC
Inter-Community	**89.6**	**78.0**
Intra-Community	88.5	77.5

Table 5. Different graph-based Jigsaw puzzling strategies.

η	HR-Avenue	HR-STC
2	88.6	77.5
3	88.9	77.7
4	**89.6**	77.9
5	89.5	**78.0**
6	89.5	77.9
7	89.3	77.8

Table 6. Variability on the number of subgraphs.

which of these approaches yields the most effective latent representation from past frames to guide `Diffusion`. In the *Encoder*-based approach, the encoder architecture constructs the conditioning latent space for `Diffusion` without introducing an additional loss to the network. In the *AutoEncoder*-based approach, the reconstruction loss of the autoencoder is added to the loss of `Diffusion`. More details of their architectures are provided in Section F of the Supplementary Material. For a fair comparison, since in *Encoder*-based and *AutoEncoder*-based approaches, no graphs are constructed to build the Jigsaw puzzles upon, in our proposed *Graph*-based approach, we exclude `Puzzle`, and keep only `Graph` and `Diffusion`. Tab. 4 indicates that in general, the *Autoencoder*-based approach outperforms *Encoder*-based approach. This discrepancy can be attributed to the supervision of the reconstruction loss in the *Autoencoder*-based method, which aids in obtaining a better representation of past frames. In contrast, our proposed *Graph*-based method achieves significantly higher AUROC scores. This is due to `Graph`'s ability to effectively capture spatio-temporal dependencies in the data. Parameter-wise, in our model, each joint is only connected to a few other joins, while the *Encoder*-based and *Autoencoder*-based methods are fully connected neural networks, where all joints contribute to the output. This results in a much lower number of parameters in our proposed method compared to those other methods.

Types of Graph-based Jigsaw Puzzles. In this experiment, we explore the efficacy of various graph-based Jigsaw puzzling strategies. Specifically, we compare our proposed Jigsaw puzzling method described in Sec. 2.2, so-called *Inter-community*, which selects two subgraphs and interchanges them, with a new strategy, called *Intra*-community. This new technique chooses a single subgraph and randomly rearranges the nodes inside, where the objective is to determine which of these subgraphs has been changed. A visualization of this technique is provided in Section G of the Supplementary Material. As shown in Tab. 5, *Inter*-community yields superior results compared to *Intra*-community. This could be attributed to our method of subgraph identification, which aims to select subgraphs that are as distinct from each other as possible while preserving the tight connections among nodes within the same subgraph. Consequently, shuffling within the same subgraph is less effective because the nodes are highly interdependent, and detecting the shuffled subgraph is a simpler pretext task compared to the *Inter*-community shuffling, hence, less guidance is provided through the self-supervision process.

Number of Subgraphs. This experiment varies the number of subgraphs, denoted as η, to be extracted from the graph. Results are shown in Tab. 6. We range the value of η from 2, where the constructed subgraphs are relatively small, up to 7, where the constructed subgraphs become too large. Our results suggest that slight variations do not adversely affect the model's overall performance as long as the size of subgraphs remains reasonable to adequately represent body regions, with η values between 4 and 6.

4. Conclusion

We introduce GiCiSAD, a lightweight framework addressing three key challenges in SVAD datasets. Our method emphasizes a challenging pretext task to learn region-specific discrepancies in human body motions, requiring dynamic graph-based modeling. Additionally, we propose a conditional diffusion model to generate diverse future human motions guided by encoded past-frame representations. Experiments demonstrate our approach's efficacy, achieving SOTA performance on four benchmarks with notably fewer training parameters.

References

[1] Andra Acsintoae, Andrei Florescu, Mariana-Iuliana Georgescu, Tudor Mare, Paul Sumedrea, Radu Tudor Ionescu, Fahad Shahbaz Khan, and Mubarak Shah. Ubnormal: New benchmark for supervised open-set video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20143–20153, 2022. 6

[2] Antonio Barbalau, Radu Tudor Ionescu, Mariana-Iuliana Georgescu, Jacob Dueholm, Bharathkumar Ramachandra, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Ssmtl++: Revisiting self-supervised multi-task learning for video anomaly detection. *Computer Vision and Image Understanding*, 229:103656, 2023. 6

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 7

[4] Behzad Bozorgtabar and Dwarikanath Mahapatra. Attention-conditioned augmentations for self-supervised anomaly detection and localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14720–14728, 2023. 2

[5] Silvia Bucci, Antonio D'Innocente, Yujun Liao, Fabio M Carlucci, Barbara Caputo, and Tatiana Tommasi. Self-supervised learning across domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5516–5528, 2021. 2, 4

[6] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4027–4035, 2021. 1, 3

[7] Alessandro Flaborea, Luca Collorone, Guido Maria D'Amely Di Melendugno, Stefano D'Arrigo, Bardh Prenkaj, and Fabio Galasso. Multimodal motion conditioned diffusion model for skeleton-based video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10318–10329, 2023. 1, 2, 5, 6, 7

[8] Alessandro Flaborea, Guido Maria D'Amely di Melendugno, Stefano D'arrigo, Marco Aurelio Sterpa, Alessio Sampieri, and Fabio Galasso. Contracting skeletal kinematic embeddings for anomaly detection. *arXiv preprint arXiv:2301.09489*, 2023. 1, 2, 6, 7

[9] Mariana Iuliana Georgescu, Radu Tudor Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4505–4523, 2021. 7

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 5

[11] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002. 4

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 5

[13] Thi Kieu Khanh Ho and Narges Armanfard. Multivariate time-series anomaly detection with contaminated data. *arXiv preprint arXiv:2308.12563*, 2023. 1

[14] Thi Kieu Khanh Ho and Narges Armanfard. Self-supervised learning for anomalous channel detection in eeg graphs: Application to seizure analysis. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7866–7874, 2023. 1

[15] Thi Kieu Khanh Ho, Ali Karami, and Narges Armanfard. Graph-based time-series anomaly detection: A survey and outlook. *arXiv preprint arXiv:2302.00058*, 2024. 1

[16] Hadi Hojjati, Thi Kieu Khanh Ho, and Naregs Armanfard. Self-supervised anomaly detection in computer vision and beyond: A survey and outlook. *Neural Networks*, page 106106, 2024. 2

[17] Hadi Hojjati, Mohammadreza Sadeghi, and Narges Armanfard. Multivariate time-series anomaly detection with temporal self-supervision and graphs: Application to vehicle failure prediction. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 242–259. Springer, 2023. 1

[18] Yashswi Jain, Ashvini Kumar Sharma, Rajbabu Velmurugan, and Biplab Banerjee. Posecvae: Anomalous human activity detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2927–2934. IEEE, 2021. 1, 6, 7

[19] Asiegbu Miracle Kanu-Asiegbu, Ram Vasudevan, and Xiaoxiao Du. Bipoco: Bi-directional trajectory prediction with pose constraints for pedestrian anomaly detection. *arXiv preprint arXiv:2207.02281*, 2022. 6

[20] Hwan Kim, Byung Suk Lee, Won-Yong Shin, and Sungsu Lim. Graph anomaly detection with graph neural networks: Current status and challenges. *IEEE Access*, 2022. 1

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[22] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9664–9674, 2021. 2

[23] Nanjun Li, Faliang Chang, and Chunsheng Liu. Human-related anomalous event detection via spatial-temporal graph convolutional autoencoder with embedded long short-term memory network. *Neurocomputing*, 490:482–494, 2022. 1, 6, 7

[24] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1):857–876, 2021. 2

[25] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013. 6

[26] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework.

In *Proceedings of the IEEE international conference on computer vision*, pages 341–349, 2017. 6

[27] Weixin Luo, Wen Liu, and Shenghua Gao. Normal graph: Spatial temporal graph convolutional networks based prediction network for skeleton based video anomaly detection. *Neurocomputing*, 444:332–337, 2021. 1, 2, 5, 6

[28] Smaragda Markaki and Costas Panagiotakis. Jigsaw puzzle solving techniques and applications: a survey. *The Visual Computer*, 39(10):4405–4421, 2023. 2

[29] Amir Markovitz, Gilad Sharir, Itamar Friedman, Lihi Zelnik-Manor, and Shai Avidan. Graph embedded pose clustering for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10539–10547, 2020. 1, 2, 5, 6, 7

[30] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717, 2020. 2, 4

[31] Romero Morais, Vuong Le, Truyen Tran, Budhaditya Saha, Moussa Mansour, and Svetha Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11996–12004, 2019. 1, 2, 5

[32] Rashmiranjan Nayak, Umesh Chandra Pati, and Santos Kumar Das. A comprehensive review on deep learning-based methods for video anomaly detection. *Image and Vision Computing*, 106:104078, 2021. 2

[33] Trong-Nguyen Nguyen and Jean Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1273–1283, 2019. 1

[34] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 6

[35] Bahareh Nikpour and Narges Armanfard. Spatio-temporal hard attention learning for skeleton-based activity recognition. *Pattern Recognition*, 139:109428, 2023. 1

[36] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 4

[37] Wenfeng Pang, Qianhua He, Yanxiong Li, and Noman Ahmed. Detecting video anomalies by jointly utilizing appearance and skeleton information. *Expert Systems with Applications*, 246:123135, 2024. 1

[38] Bharathkumar Ramachandra, Michael J Jones, and Ranga Raju Vatsavai. A survey of single-scene video anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2293–2312, 2020. 1

[39] Jing Ren, Feng Xia, Ivan Lee, Azadeh Noori Hoshyar, and Charu Aggarwal. Graph learning for anomaly analytics: Algorithms, applications, and challenges. *ACM Transactions on Intelligent Systems and Technology*, 14(2):1–29, 2023. 1

[40] Nicolae-Cătălin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13576–13586, 2022. 2

[41] Royston Rodrigues, Neha Bhargava, Rajbabu Velmurugan, and Subhasis Chaudhuri. Multi-timescale trajectory prediction for abnormal human activity detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2626–2634, 2020. 1

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5

[43] Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. Space-time-separable graph convolutional network for pose forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11209–11218, 2021. 5

[44] Alexandros Stergiou, Brent De Weerdt, and Nikos Deligiannis. Holistic representation learning for multitask trajectory anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6729–6739, 2024. 1, 6, 7

[45] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018. 1

[46] Aiham Taleb, Christoph Lippert, Tassilo Klein, and Moin Nabi. Multimodal self-supervised learning for medical image analysis. In *International conference on information processing in medical imaging*, pages 661–673. Springer, 2021. 2, 4

[47] Aiham Taleb, Winfried Loetzsch, Noel Danz, Julius Severin, Thomas Gaertner, Benjamin Bergner, and Christoph Lippert. 3d self-supervised methods for medical imaging. *Advances in neural information processing systems*, 33:18158–18172, 2020. 2, 4

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6

[49] Guodong Wang, Yunhong Wang, Jie Qin, Dongming Zhang, Xiuguo Bao, and Di Huang. Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles. In *European Conference on Computer Vision*, pages 494–511. Springer, 2022. 1, 2, 4, 5

[50] Jian Xiao, Tianyuan Liu, and Genlin Ji. Human kinematics-inspired skeleton-based video anomaly detection. *arXiv preprint arXiv:2309.15662*, 2023. 1

[51] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941, 2017. 1

[52] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-

training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022. 2