electronics

MDPI

Article

VW-SC3D: A Sparse 3D CNN-Based Spatial–Temporal Network with View Weighting for Skeleton-Based Action Recognition

Xiaotian Lin [1,*], Leiyang Xu [1], Songlin Zhuang [2] and Qiang Wang [1]

[1] Harbin Institute of Technology, No. 92 Xidazhi Street, Nangang District, Harbin 150001, China
[2] Yongjiang Laboratory, Ningbo 315202, China
* Correspondence: xiaotian.lin@hit.edu.cn; Tel.: +86-1871-460-1897

Abstract: In recent years, human action recognition has received increasing attention as a significant function of human–machine interaction. The human skeleton is one of the most effective representations of human actions because it is highly compact and informative. Many recent skeleton-based action recognition methods are based on graph convolutional networks (GCNs) as they preserve the topology of the human skeleton while extracting features. Although many of these methods give impressive results, there are some limitations in robustness, interoperability, and scalability. Furthermore, most of these methods ignore the underlying information of view direction and rely on the model to learn how to adjust the view from training data. In this work, we propose VW-SC3D, a spatial–temporal model with view weighting for skeleton-based action recognition. In brief, our model uses a sparse 3D CNN to extract spatial features for each frame and uses a transformer encoder to obtain temporal information within the frames. Compared to GCN-based methods, our method performs better in extracting spatial–temporal features and is more adaptive to different types of 3D skeleton data. The sparse 3D CNN makes our model more computationally efficient and more flexible. In addition, a learnable view weighting module enhances the robustness of the proposed model against viewpoint changes. A test on two different types of datasets shows a competitive result with SOTA methods, and the performance is even better in view-changing situations.

Keywords: skeleton-based action recognition; sparse 3D convolutional neural network; transformer; view adaptive

Citation: Lin, X.; Xu, L.; Zhuang, S.; Wang, Q. VW-SC3D: A Sparse 3D CNN-Based Spatial-Temporal Network with View Weighting for Skeleton-Based Action Recognition. *Electronics* **2023**, *12*, 117. https://doi.org/10.3390/electronics12010117

Academic Editors: Zhan Li, Zhang Chen and Yiyong Sun

Received: 24 November 2022
Revised: 20 December 2022
Accepted: 23 December 2022
Published: 27 December 2022

1. Introduction

Human action recognition has become a popular area in current research studies, and has been widely used in industrial [1,2], social [3–5], medical [6], and sports [7] fields. With the development of deep learning and the drop in computational cost, many excellent works [8] for action recognition have been proposed in recent years. According to the type of action data, the existing works are mainly divided into two types: video-based [9] and skeleton-based [10–12] action data. Compared with RGB and optical flow sequences, skeleton modality is robust to illumination changes, body scales, dynamic camera views, and background noise. In addition, both motion capture devices [13,14] and some advanced human pose estimation algorithms [15–17] can make it easier to obtain high-accurate 3D skeletal action data. Based on the above advantages of the skeleton, skeleton-based action recognition has received much attention and become an active topic in computer vision.

Skeleton-based action recognition (SAR) aims to recognize action categories by using skeletal action sequences. The most challenging problem for skeleton-based action recognition is how to extract the spatial and temporal features from skeleton data properly. For spatial features, due to the geometric constraints of the human skeleton and the strong correlation between adjacent joints, rich spatial information of the human skeleton can be extracted in each frame. For temporal features, time domain correlation information can be given by calculating the change in the skeletons between frames. Designing a

framework to combine the temporal and spatial features of skeleton sequences and extract as many effective features as possible is the key to the SAR task. Traditional SAR methods generally focus on modeling the motion characteristics of the skeleton joints and designing hand-crafted features [14,18,19] according to human experience in recognizing actions. However, handcrafted features are suboptimal in most cases and depend on the dataset. For different datasets, we often need to design different features, which is inconvenient. With the successful application of deep learning in the action recognition task, methods based on various neural network architectures have emerged [20]. At present, deep learning methods for SAR can be divided into three types: RNN-based methods, GCN-based methods, and CNN-based methods.

The skeleton sequence records the joint positions at each moment, which can be regarded as a kind of time series. Meanwhile, RNNs are suitable for analyzing time series data due to their specially designed structure [21–23]. Aiming at the problem of the gradient vanishing and exploding of RNNs making them difficult to train, Li et al. [24] presented an independently recurrent neural network (IndRNN), where neurons in the same layer are independent of each other and are connected across layers. However, although the RNN is good at obtaining the temporal features of the skeleton sequence, it lacks the ability to effectively learn the spatial relationship between skeleton joints.

From the skeleton's view, human skeleton data are represented in a natural topological graph, so graph-related neural networks, especially GCNs [25–28], are suitable and frequently used for SAR. Yan et al. [29] first introduced GCNs into the SAR field. They proposed a novel model of dynamic skeletons called spatial–temporal graph convolutional networks (ST-GCNs), which can automatically learn both the spatial and temporal patterns from skeleton sequence. Human skeletons were processed as spatial graphs of edges and nodes in this study. The excellent results of this work have inspired many researchers to investigate GCN-based methods. However, GCN-based methods are difficult to generalize to skeletons with a different number of joints or connections.

CNNs are widely used in image analysis tasks due to their natural and excellent high-level information extraction capabilities. Compared with the above deep learning models, CNNs have better feature extraction capabilities and better flexibility toward different sizes of skeletons. However, it is also a challenge for CNN-based methods to balance and model the spatial and temporal information. Many works [30–34] represent skeleton sequences as 2D pseudo-images, which can be fed into CNNs directly. However, there are still limitations for these methods: (1) converting 3D skeletal data to a 2D pseudo-image may not be an ideal skeleton representation, because they will inevitably lose some information, such as the potential connections between some joints; (2) the coordinates of joints are employed as features, which may ignore the 3D spatial structure relationship of joints under different poses; (3) how to fully utilize the spatial feature extraction capability of CNNs, the architecture of CNNs, and the size and speed of the model [11,35] are still problems.

In this paper, we propose a novel view-invariant spatio-temporal model for SAR, called VW-SC3D, which utilizes sparse three-dimensional convolutional neural networks (3D CNNs) to extract spatial features of a 3D skeleton within each frame to form feature vectors. First, 3D skeletons in each frame were processed to the 3D maps, and we rotated the skeleton in each frame to a uniform angle by weighting the average view of each frame. Second, sparse 3D CNNs took the 3D maps of skeletons as an input to extract the spatial features, and then a temporal transformer was applied to obtain the temporal dependencies between frames and dynamic relationships. Figure 1 shows an overall pipeline of our work. Third, we verified our methods and carried out experiments on a large-scale public dataset, NTU-RGB+D 60, and a small-scale dataset, the Taichi dataset. To the best of our knowledge, we are the first to apply sparse 3D CNNs on the 3D map transformed by the 3D human skeleton. Our main contributions are summarized as follows:

(1) This work presents a novel spatial–temporal architecture for SAR that uses sparse 3D CNNs to obtain spatial features for each skeleton and a temporal transformer to

extract temporal dynamic information between skeletons, which take full advantage of the properties of different network structures.

(2) To retain more information and generalize to any skeleton, we transformed the 3D skeleton to a 3D point cloud instead of converting a 3D skeleton to a 2D pseudo-image. The entire 3D point cloud was taken as features in place of the coordinates of joints. In addition, sparse 3D CNNs were employed instead of general 3D CNNs to make our model lighter.

(3) A view-weighted transformation mechanism was introduced to address the view variation problem of 3D point cloud for better action recognition.

The overall structure of the study takes the form of five chapters, including this introduction. Section 2 first provides a review of the existing related works. Then, a detailed description of our proposed method is provided in Section 3. After this, extensive experiments on the large-scale public action dataset were conducted, followed by an analysis and comparison of experimental results in Section 4. Finally, Section 5 summarizes our paper and draws conclusions.

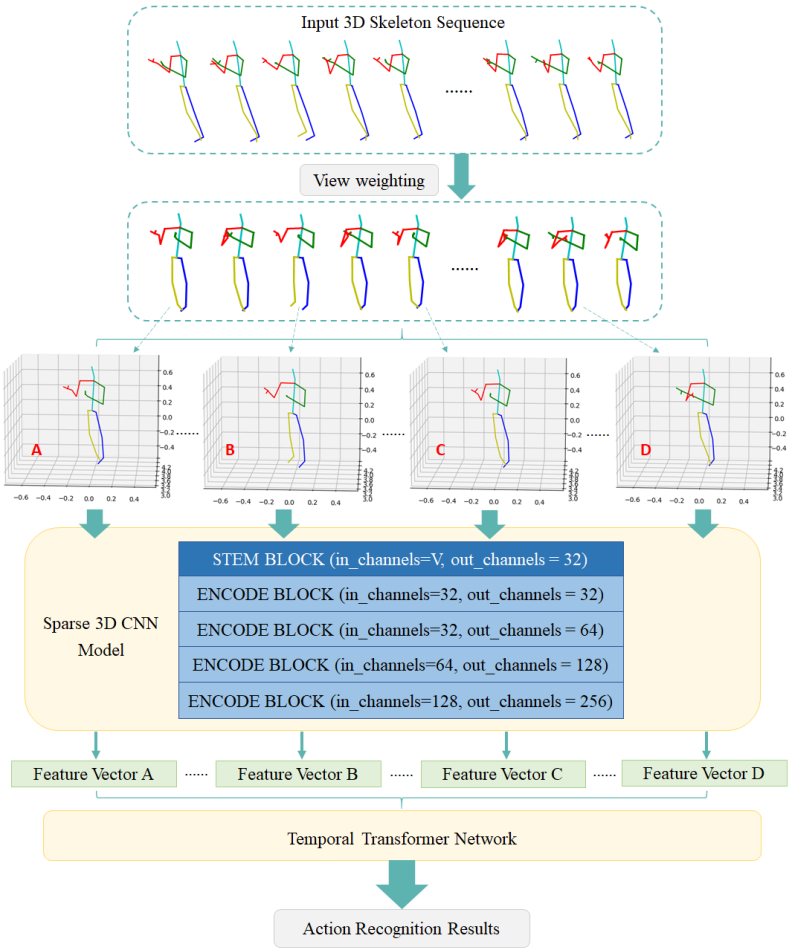

Figure 1. Diagram of the proposed method.

2. Related Work

In this section, we collect and discuss relevant prior work, including CNN-based methods for SAR, temporal transformer networks, and view invariance in SAR.

2.1. CNN-Based Methods for Skeleton-Based Action Recognition

In recent years, CNNs have shown encouraging performances in image and video analysis. However, such models cannot directly act on 3D skeleton sequences due to their limitations on the 2D image input. For 2D CNN-based methods, 3D skeleton sequence

data are converted to a pseudo-image to meet the needs of the CNN input. Wang et al. [36] proposed joint trajectory maps (JTM) to represent 3D skeleton sequences by encoding the the joint trajectories and their dynamics into three 2D images. Caetano et al. [37] introduced a novel skeleton image representation, named SkeleMotion, where the temporal dynamics of the sequence are encoded as variations in columns and the spatial structure of each frame is represented as rows of a matrix. In [35], Yang et al. presented a double-feature double-motion network (DD-Net), which uses a lightweight network structure and can reach a very fast speed. However, this method of processing the skeleton sequence inevitably loses some information. Liu et al. [38] firstly applied 3D CNNs in SAR. They proposed a novel two-stream model using 3D CNNs, and two 3D CNN models extracted the spatial and temporal information, respectively. However, it is difficult for 3D CNNs to mine long-term temporal information, and the two-stream structure model has large parameters and a slow speed. However, this work treats skeleton sequences in the temporal and spatial dimension in the same way, ignoring the difference between the temporal and spatial dimension in skeleton data. Duan et al. [39] represented the human skeleton sequences with a 3D heatmap volume instead of a 2D graph sequence, and then used 3D CNNs for feature extraction and action classification. This method also essentially converts the 3D skeleton into a 2D graph, and then adds the time dimension to form the three-dimensional input data for the 3D CNN. Shi et al. [40] proposed a novel sparse 4D convolutional network (SC4D) by regarding the skeletal sequence as a spatial–temporal point cloud and voxelizing it into a four-dimensional grid. The advantage of this work is that there is no need to manually design hand-crafted transformation rules. It makes better use of the advantages of convolutional networks, and provides a more general and robust framework for skeletal data. Similar to [40], we converted the human skeleton into a 3D map, retaining the information of three dimensions in the skeleton. However, we only used sparse 3D CNNs to extract the spatial information of the skeleton in order to make full use of the spatial extraction ability of CNNs. The temporal features of skeleton sequences were obtained by other methods.

2.2. Transformer for Skeleton-Based Action Recognition

The success of the transformer [41] proposes a new simple network architecture for modeling long temporal sequences through a powerful self-attention mechanism. The transformer has a good performance in the field of natural language processing [42,43]. Since text and action sequences have a high logical similarity, it is a natural idea to introduce the transformer into the action recognition field [44]. Cho et al. [45] first introduced the self-attention mechanism for SAR, and presented three variants of the self-attention network to extract high-level semantics. In addition, Plizzari et al. [46] used a spatial self-attention module to understand intra-frame interactions between different body parts, and a temporal self-attention module to model inter-frame correlations. For 2D pose-based action recognition, Mazzia et al. [47] introduced an action transformer to provide a low-latency solution for an accurate and effective real-time performance. To adopt the skeleton with noise, Zhang et al. [48] proposed a self-supervised learning method, and designed a spatial transformer block and directional temporal transformer block for modeling skeleton sequences in spatial and temporal dimensions, respectively. These works have proved that the transformer has shown an excellent performance on some action datasets. It is worth mentioned that the time series modeling ability of the transformer is outstanding, especially when dealing with long temporal sequences. Therefore, in our work, we use the transformer to extract the temporal features of the 3D skeleton; that is, the dynamic information between frames.

2.3. View Invariance in Skeleton-Based Action Recognition

In addition to feature extraction, how to deal with the view change problem [49,50] is also one of the challenges for SAR. The commonly used methods are to preprocess the skeleton [51–53] before feeding it into the model. In [23], Du et al. centralized the joints'

positions to the human center for each frame and smoothed the positions. Shahroudy et al. [54] translated the skeletons to the body coordinate system and scaled all of the 3D points based on a fixed distance. However, these preprocessing methods are inflexible and may result in a loss of dynamic information within actions. Ji et al. [55] proposed a view-guided skeleton CNN (VS-CNN) to solve the arbitrary-view action recognition by emphasizing crucial joints and their relationships. Zhang et al. [33] designed two view adaptive networks, VA-RNN and VA-CNN, which can determine the most suitable observation viewpoints and transform the skeletons to those viewpoints. In addition, Gao et al. [56] presented a view transformation network that transforms arbitrary-view action samples to a base view to seek a view-invariant representation. In our paper, we introduced a simple view transformation mechanism by per-frame view weighting to address the view variation problem.

3. Method

In this section, we will propose the framework of our model, VW-SC3D, a spatial–temporal action recognizer based on sparse 3D convolutional neural network and transformers. Our method is a competitive alternative to GCN-based frameworks and highly adaptable to view variance with the help of a view weighting module. The details of our model will be covered in the following subsections. In brief, our model takes raw 3D skeleton data of an action as the input and outputs the probabilities of possible actions. We will dive into the data format later.

3.1. 3D Point Cloud Generating and View Weighting

The input data contain a 3D skeleton for each frame of an action. The skeleton data consist of 3D coordinates of human joints. There are several different definitions of the joints layout according to different public datasets or the data acquisition device. In different layout definitions, the number of joints and the connectivity between joints may vary, which makes it hard to design a unified model that can handle different layout definitions. Therefore, we first developed a mechanism that can adapt our model to different types of layout definitions.

Given a layout definition and raw 3D coordinates of joint as we described, we set the input data as a tensor X with shape $(N, T, V, 3)$, where N is the batch size, T is the number of frames, V is the number of joints, and 3 indicates the 3 real numbers of a Cartesian 3D coordinate. The layout is described as a list of joint pairs, which indicates which pairs of joints are linked in this particular definition of layout; for example, $[(1,2),(2,21),\ldots,(25,12)]$, where each number is a joint index that corresponds to a human joint such as the elbow, wrist, or neck. With this index information, we can transform the raw input into a tensor X' with shape $(N, T, V', 2, 3)$, where V' is the number of pairs. X' stores the 3D coordinates of two points for each of N samples, T frames, and V' pairs of joints.

Given the tensor X', we linked the joints with discrete 3D points to generate a point cloud. Specifically, we first set a hyperparameter s, which is an integer indicating the density of link points. Then, we constructed a matrix M with shape $(1+s, 2)$ as follows:

$$\begin{bmatrix} 1 & 0 \\ 1-\frac{1}{s} & \frac{1}{s} \\ 1-\frac{2}{s} & \frac{2}{s} \\ \vdots & \vdots \\ 0 & 1 \end{bmatrix} \tag{1}$$

After that, we calculated the link points with the formula

$$Y_{ijklm} = \sum_{n=1}^{N} M_{ln} X'_{ijknm} \tag{2}$$

where Y is a tensor with shape $(N, T, V', 1+s, 3)$, which stores the 3D coordinates of the skeleton point cloud. This operation can be carried out in parallel with the help of an Einstein sum or batch matrix multiplication supported by tools such as Pytorch or Numpy. In Pytorch, this operation can even be built as a differentiable module, which can be integrated into our end-to-end trained model. Figure 2a is an illustration of the generated 3D skeleton point cloud.

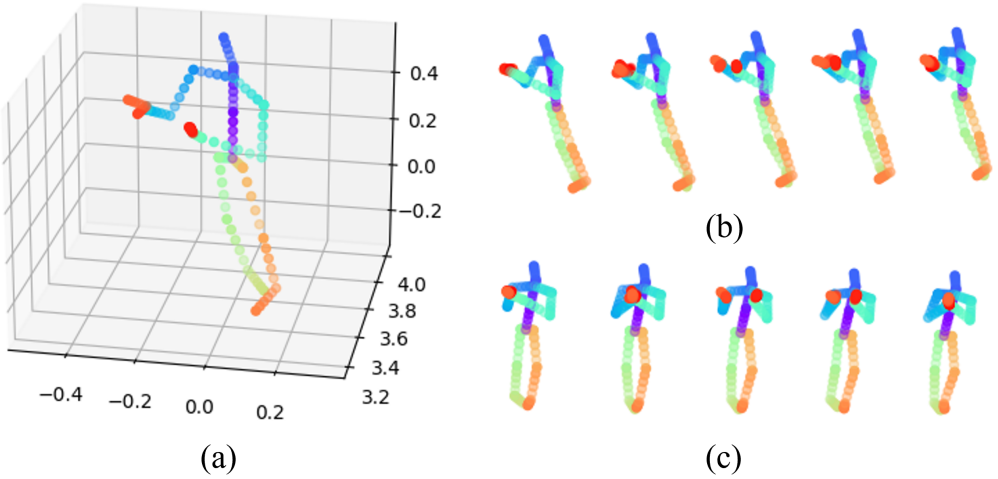

Figure 2. (**a**) Three-dimensional skeleton point cloud in a single frame, (**b**) frames of 3D skeleton point cloud before view adjusting, (**c**) frames of 3D skeleton point cloud after view adjusting.

Finally, for each of N samples, we rotated the skeleton according to the view direction of each frame. This operation is designed to cope with variant view points of the same action. In brief, if the whole action is viewed from a different point, the algorithm should still know it is the same action. Most researchers calculate the view direction of each frame and rotate each frame separately so that all of the frames are viewed along the x axis. The view direction of each frame is usually defined as the normal vector of the plane formed by the left shoulder, right shoulder, and the middle of the spine, which is a stable structure that exists in all different layout definitions. However, rotating the frames separately will lose the global information between frames. Thus, in our work, we calculated a global view direction by a weighted averaging of all of the view directions. We initialized the weights equally, but we set the weights as trainable parameters so that they can be adjusted according to the training data. Some frames such as the beginning and ending frames are not as important as the middle ones, so trainable weights can make the model more flexible and effective. As Figure 2b,c show, the 3D skeleton point cloud was adjusted to a normalized direction.

3.2. Sparse 3D Convolutional Neural Networks

For each frame of each action sample, a 3D point cloud skeleton with a normalized view was generated by the above method. To extract features from 3D data, 3D CNNs are a powerful tool.

Three-dimensional CNNs are usually used for extracting feature from data with a 3D structure, such as videos, MRI, and HSI. Similar to 2D CNNs, 3D CNNs convolves three-dimensional kernels to the input. A single kernel move along three spatial directions in the 3D feature maps to calculate the output at each coordinate.

The output value of the jth convolutional kernel at position (x, y, z) in the ith layer is given by

$$v_{ij}'^{xyz} = b_{ij} + \sum_{m=1}^{M_i} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} \tag{3}$$

where v_{im}^{xyz} is the mth feature map of the ith layer at position (x, y, z), and there are M_i feature maps in the ith layer. w_{ijm}^{pqr} is the (p,q,r)th value of the jth kernel in the ith layer connected to the mth feature map of the previous layer. Note that, for each combination of i, j, and m, there exists a 3D cubic kernel of shape (P_i, Q_i, R_i), where those kernels are independent to each other. This convolutional operation outputs a tensor v', which has J ($j \in [1, J]$ and $j \in \mathbb{N}$). Figure 3 is an illustration of 3D CNNs.

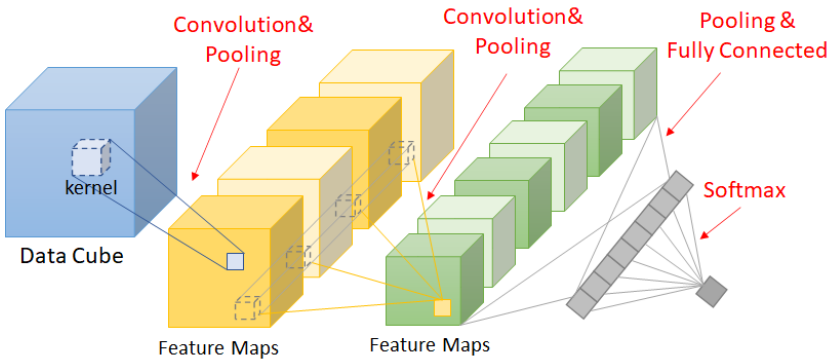

Figure 3. Three-dimensional convolutional neural networks.

We can normally reshape tensor Y to $(N * T, V', s + 1, 3)$, quantize the coordinates to a 3D grid, fill an empty tensor with shape $(N * T, V', D, H, W)$, and use 3D convolution to extract deep features. However, in our case, most of the space is unoccupied, which means normal 3D convolutional modules will waste many computing resources and are extremely slower.

Sparse 3D CNNs (SC3Ds) were developed for reducing the computational consumption of 3D CNNs, especially when processing the point cloud, which is always spatially sparse. SC3Ds formulate the point cloud with features as an unordered set of points paired with features $\{(p_j, x_(j)\}$, where $x_j \in \mathbb{R}^C$ is a C-dimensional feature vector for point $p_j \in \mathbb{Z}^D$. Instead of computing all of the convolutions at each 3D position, the SC3D maps the possible output position to the sparse input points and only performs the necessary computations. There are some slight differences between normal 3D CNNs and SC3Ds because SC3Ds omit some of the possible output positions that are far from the input points. This difference will not significantly influence the performance, especially when the input point cloud is sparse. The details can be found in [57].

The design of the SC3D is different from normal 3D CNNs because we do not obtain the shape of the output in each layer. For each layer in the SC3D, the input is a point cloud with features as mentioned above and the output is also a featured point cloud. In our work, we designed a SC3D with a stem block and several encoder blocks as shown in Figure 4. The stem block extracts the shallowest feature from the input and adjust the channel dimension to 32, but without any down-sampling.

The encoder blocks each extract deeper features by extending the channel dimension and performing down-sampling. After a few encoder blocks, the output point cloud is much smaller but with larger feature channels. Finally, the model performs an average pooling for each channel and outputs a feature vector for each of N samples and T frames. Figure 5 is an illustration of our SC3D layers. Note that we processed the point cloud without it being limited in a cubic boundary in comparison to the 3D CNN in Figure 3, and still extracted more channels of deep features in each layer. Beyond computational efficiency, SC3Ds can help focus on the relevant points without having to set a limit beforehand, which is highly feasible in action data processing.

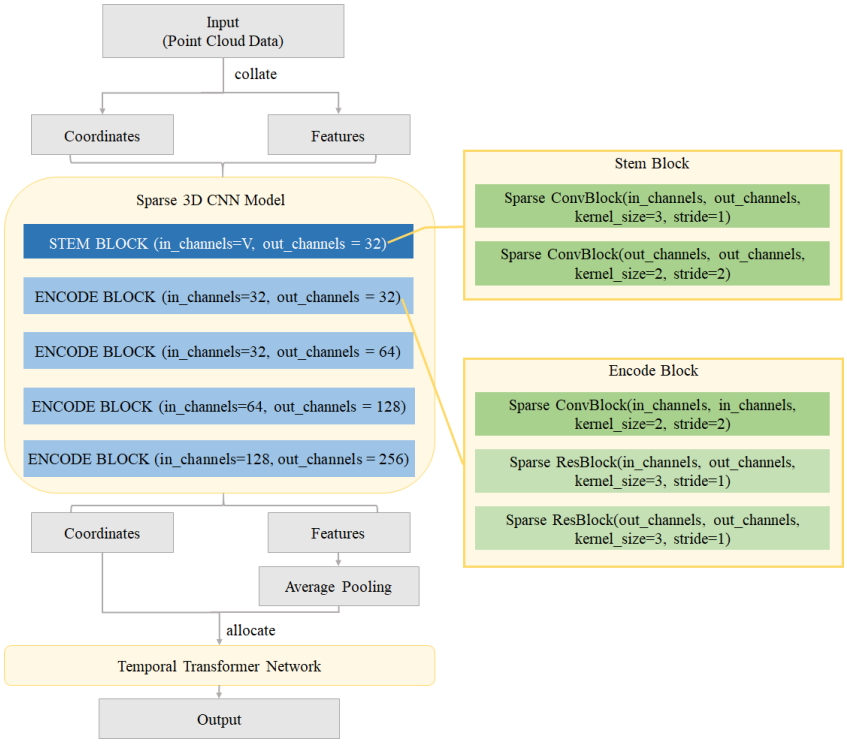

Figure 4. Details of the proposed SC3D-based model.

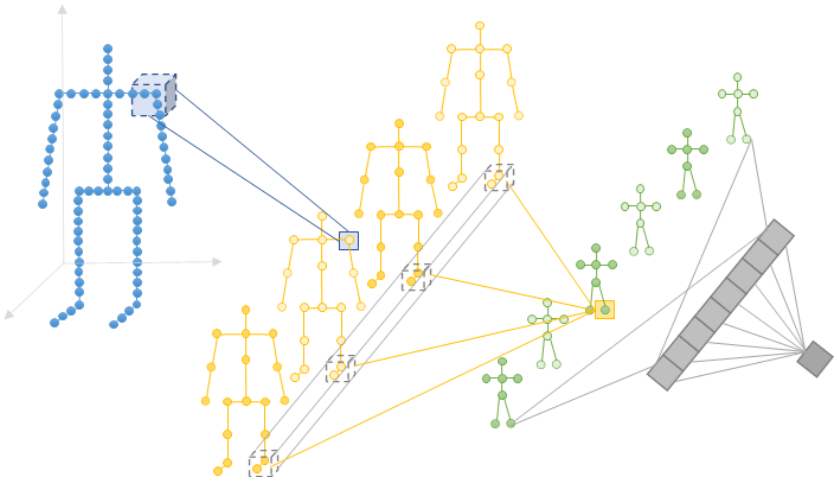

Figure 5. SC3D for skeleton feature extraction.

3.3. Transformer Encoder

In recent research, the transformer has proved to be quite powerful in processing sequential data, which is compatible with our method. The original transformer model solves the problem of machine translation, where an encoder and a decoder are developed to encode a sentence in one language to a feature and decode it to a sentence in another language. The details of the whole structure contains multi-head attention, masked multi-head attention, a feed-forward network, positional encoding, and word embeddings. Many modules are designed specifically for the language model, but the multi-head attention mechanism is adaptive to multiple different tasks, including extracting features from sequential data such as human actions.

An attention function maps a query and a set of key–value pairs to an output, where the query, keys, and values are all vectors. The output is computed as a weighted sum of the values, where the weights are computed by a dot-product of the query and keys. In

naive attention, queries, keys, and values are all identical to the input vectors, which makes the output linear combinations of the input vectors; that is, given a vector v_t for each time step t, the attention mechanism outputs a vector v'_t, respectively, and each output vector is a linear combination of all of the input vectors $v_{t'}$, where the weight for $v_{t'}$ is determined by calculating the dot-product of v_t and $v_{t'}$. This mechanism helps the information exchange between feature vectors along the time dimension; thus, it can extract temporal features from the input data. Compared with methods based on recurrent neural networks, the attention mechanism exchanges information in both forward and backward directions, and the distance between frames will not influence the strength of the exchange. This is basically an advantage, but with a small drawback. Regardless of the order of input vectors, the output will be the same because of this mechanism. Thus, in the transformer encoder, there is a positional encoding module to introduce position information to the model. Multi-head attention is an upgraded version of attention, and feeds the original input to different linear layers to produce different queries, keys, and values, which makes the model more flexible.

In our work, after we fed our data to SC3D networks and obtained a feature vector for each frame, we used a stacked multi-head attention module with a feed-forward sub-network to extract temporal features from the spatial features over time, and obtained a set of feature vectors of the same length as the input. Finally, we used global average pooling to compress the whole spatial–temporal information into a single feature vector. Instead of directly processing the spatial feature vectors with a fully connected neural network, our method takes full advantage of the prior knowledge that the spatial feature vectors are correlated along the axis of time. Thus, our method is more powerful in extracting spatial–temporal features than naive neural networks or GCN-based networks, which focus more on the spatial topology of the human body. In the next section, we will show some experimental results to prove our thoughts.

4. Experiments

In this section, we conducted our experiments on one large-scale and one small-scale skeleton-based action recognition dataset and compared our results with state-of-the-art methods under different evaluations. In addition, we designed multiple ablation experiments to verify the effectiveness of each part of our architecture.

4.1. Datasets

The NTU-RGB+D 60 dataset [54] is a large-scale public action dataset for human action recognition. It contains 56,880 action samples in 60 labelled action classes, including daily actions, mutual actions, and medical conditions. This dataset is captured by three Kinect V2 cameras, and has four different modalities—RGB videos, depth map sequences, 3D skeletal data, and infrared (IR) videos—for each sample. We used the 3D skeletal data as our experiment data. Each skeleton has 25 body joints, as shown in Figure 6a, and 3D skeletal data contain the 3D coordinates of joints at each frame. Forty distinct subjects, various camera settings, changed captured views, and different orientations of objects form a large and diverse sample. There are two evaluations proposed in [54]. Cross-subject (CS) evaluation splits the 40 subjects into two groups: 20 subjects for training and the other 20 subjects for testing. For cross-view (CV) evaluation, all of the samples of camera 1 are for testing and the samples of cameras 2 and 3 are for training. Both evaluation settings were still followed in our experiments.

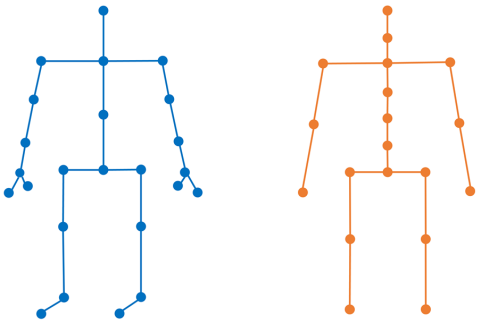

Figure 6. Skeletons of NTU-RGB+D 60 and Tai Chi dataset.

The Taichi dataset [14] is a self-established fine-grained small-scale dataset. We used the actions that come from Wu Style Tai Chi, which is one of the most popular genres of Tai Chi. Motion data were collected by an inertial sensor system, and the skeletons were obtained by processing the motion quaternion based on the chain rule. This dataset contains twenty sets of Tai Chi actions performed by two subjects, including one female and one male. Each subject collects ten sets, and we used the first ten action categories in each set for the next experiments. The skeleton has 21 joints, as shown in Figure 6b. Note that the inertial sensor system is not capable of collecting different views of the same action simultaneously like the camera-based acquisition method. In our work, we augmented the data by rotating the skeletons to several new directions in 3D space to simulate view changing. The directions were aligned to the same axes as the setup of NTU RGB+D dataset. This augmentation makes the dataset a good evaluation standard of the cross-view performance because, in camera-based data, the depth camera is less sensitive in the depth direction than other directions, which brings additional disturbance to pure view changing.

4.2. Experiment Settings

Our experiments were conducted using Pytorch framework. We trained our model on eight NVIDIA GTX 3090 GPUs in parallel with the pytorch distributed package. We chose SGD as our optimizer, with a learning rate of 0.1, momentum of 0.9, and weight decay of 5×10^{-4}. A cosine annealing strategy of learning rate adjustment was used. We performed a few instances of data pre-processing on both the training dataset and testing dataset, such as uniform sampling and 3D normalization, before feeding the raw data into our model to make sure that the spatial and temporal scale of different samples were comparable. In our final settings, there were approximately 2.4 M FLOPS in the SC3D module and 9.1 M FLOPS in the transformer module, which cost approximately 100 ms on our platform. Note that the cost of the view weighting module can be ignored compared to the SC3D and transformer. The computational complexity of the transformer is proportional to the square of input dimensions and frames, which can be adjusted for different situations.

4.3. Experiment Results

Table 1 shows the performance of our method on the NTU-RGB+D 60 dataset and the Taichi dataset under CS and CV evaluations. For each sample, our model will predict a score for each category. There are two different types of accuracy measurements in our experiment: Top-1 accuracy and Top-5 accuracy. Top-1 accuracy measures the proportion of examples for which the label with the highest predicted score matches the single target label; Top-5 accuracy means any of our model's top five highest scores matching with the target label. We recorded both the Top-1 and Top-5 action recognition accuracy to better represent the capabilities of our model.

Table 1. Action recognition accuracy (%) on the NTU-RGB+D 60 and Taichi datasets.

Dataset	Evaluation	Top-1	Top-5
NTU-RGB+D 60	CS	83.7	94.5
	CV	89.4	97.3
Taichi	CS	90.4	98.5
	CV	95.0	99.7

Our experimental results support that our model can handle skeleton data with different structures and different numbers of joint points. Since our model does not need to obtain the skeleton structure in advance, when dealing with small-scale datasets, we can first pre-train our model on a large-scale dataset and then fine-tune it. For experiments on the Taichi dataset, we fine-tuned the model pre-trained on the NTU RGB+D 60 dataset to solve the insufficient training data problem. In addition, action data from different views in the Taichi dataset were acquired by manually rotating the skeleton randomly. Therefore, the high recognition accuracy under the CV setting on the Taichi dataset proves that our view weighting module is robust to simple view changes.

Figure 7 presents the confusion matrix on the NTU-RGB+D 60 dataset, using the CS benchmark as an example, to reflect the stability of our model. In this figure, we can see that some cases in NTU-RGB+D 60 are more likely to be misclassified. This is because human actions are not only described by the shape and trajectory of body parts. Without the environmental information such as the objects appearing in the scene, some actions are very similar to each other. For example, reading and sitting are basically in the same posture most of the time. The skeleton data do not show that there is a book in the scene. Ideally, the skeleton data and the visual data should be analyzed together to deeply understand human actions. The performance on the Taichi dataset is much better because the Taichi actions are purely martial arts that can be recognized without any environmental information.

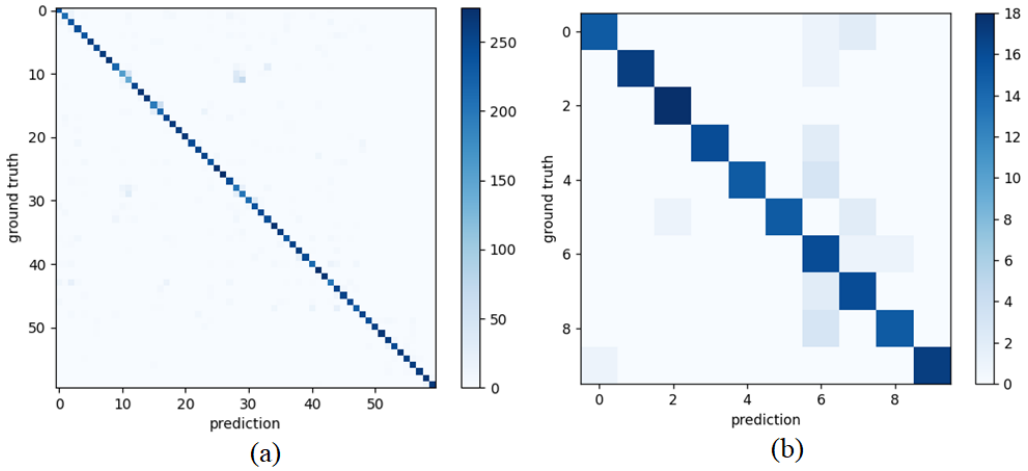

Figure 7. Confusion matrix of the (**a**) NTU-RGB+D 60 dataset and (**b**) Tai Chi dataset under CS setting.

4.4. Ablation Study

In this section, we verify the validity of modules in our framework, including view weighting and transformer modules, and analyze the influence of 3D sparse CNN layers on the experimental results.

4.4.1. Impact of View Weighting and Transformer Modules

View weighting aims to adjust the skeleton sequence to the same view, minimize the impact of view changing for the network to recognize actions, and further help to improve the recognition accuracy, especially under CV evaluation.

In addition, thanks to the powerful modeling capability and unique self-attention mechanism, the transformer has shown an excellent performance in many vision tasks. In our model, the transformer was used to find the correlation between frames and extract temporal features of the skeleton sequences. If we remove the transformer encoder and replace it with a single fully connected layer, the structure still works. In this subsection, we analyze the effectiveness of these two modules.

Table 2 compares the experimental results under different module combinations. From this table, we can see that view weighting in the presence of the transformer can improve the recognition accuracy by around 3 and 8 percent for the CS and CV settings, respectively. Moreover, the transformer is also essential for achieving a better result.

Table 2. Accuracy (%) comparisons on different modules on the NTU-RGB+D 60 dataset.

Module		NTU-RGB+D 60		Tai Chi	
VW	**Transformer**	**CS**	**CV**	**CS**	**CV**
×	×	78.1	79.2	87.3	89.8
✓	×	78.5	82.2	88.2	93.3
×	✓	80.3	81.1	89.1	91.1
✓	✓	83.7	89.4	90.4	95.0

4.4.2. Influence of 3D Sparse CNN Layers

In our design, the backbone of our model consisted of two kinds of layers: a sparse stem layer and sparse encoder layer. The stem layer is meant for data dimension transformation and extracting shallow features that cannot be removed. The sparse encoder layers are serialized one above another with trailing global average pooling, so the number of layers is negotiable. The deeper model abstracts more semantic features but is more complicated and difficult to train. We set different configurations of layers to test the influence.

Table 3 shows a comparison between different numbers of layers. To some extent, adding more encoders will increase the accuracy. Note that, at some point, adding more layers no longer boost the performance; that is how we choose the final configuration of layers.

Table 3. Accuracy (%) comparisons on different numbers of layers for the main network on the NTU-RGB+D 60 dataset.

Main Network	Structure	CS	CV
3D Sparse CNN	1 sparse encoder	81.2	83.5
	2 sparse encoders	82.1	84.4
	3 sparse encoders	83.5	87.2
	4 sparse encoders	83.7	89.4

4.5. Comparisons with the State-of-the-Art Approaches

Table 4 shows the results of SOTA skeleton-based action recognition methods. In this table, it is obvious that our work surpasses the RNN-based methods because these methods use LSTM for temporal feature extraction but ignore the underlying spatial structure of the 3D skeleton in every frame and treat the coordinates as features directly. In contrast, our work combines spatial and temporal features. Further, the transformer encoder does not limit the temporal information exchange to a single direction.

Our work is also better than most CNN-based methods because of two main reasons. First, the input of our model is a pure 3D skeleton, with the 3D spatial structure preserved, whereas some CNN-based methods compress the 3D skeleton into a 2D image, where it is difficult for the networks to directly learn 3D features. Second, our method extracts the spatial and temporal spatial features, respectively, thus being easier to learn for both parts,

whereas some CNN-based methods treat spatial and temporal features in the same way by naively stacking the spatial–temporal data as one big tensor.

Table 4. Accuracy (%) comparisons with SOTA methods on NTU-RGB+D 60 dataset.

	Method	CS	CV
RNN-Based	STA-LSTM [53]	73.4	81.2
	VA-LSTM [58]	79.4	87.6
	DS-LSTM [59]	77.8	87.3
GCN-based	ST-GCN [29]	81.5	88.3
	AS-GCN [25]	86.8	94.2
	Shift-GCN [26]	90.7	96.5
CNN-Based	JTM [36]	73.4	75.2
	SkeletonNet [60]	75.9	81.2
	Clips+CNN+MTLN [61]	79.6	84.8
	SkeleMotion [37]	76.5	84.7
	Skepxel [62]	81.3	89.2
	Banerjee et al. [63]	84.2	89.7
	VW-SC3D (Our)	83.7	89.4

The GCN-based method is slightly better than our work because the model makes full use of the topology information by processing the skeleton graph. This is an advantage but sometimes a limitation. As we mentioned above, the graph-based method relies on the definition of the skeleton layout, which makes it difficult for the model to adapt to new tasks where the layout is different. Our work is more flexible in the situation where the layout is a little different from training data because our model transforms the raw skeleton to a point cloud. For example, in our experiments, we can apply the model pre-trained on the NTU-RGB+D 60 to the small-scale Tai Chi dataset. As long as the skeleton belongs to a human, the generated point cloud will vary little, which makes our model reusable.

5. Conclusions

In this paper, we present VW-SC3D, a spatial–temporal model for skeleton-based human action recognition. Our model is basically based on sparse 3D CNNs and transformers. First, the model transforms the raw data into a point cloud with a linking module and a view weighting module. Then, a sparse 3D CNN extracts spatial features from the point cloud. Finally, transformer encoders extract temporal information from the spatial features. We trained our model on two different types of human action datasets. The results show that our proposed method is competitive with SOTA methods and performs better against view changing. We also conducted many ablation studies to show the effectiveness of different modules in this method. Specifically, we compared the performances with different setups, such as the existence of different modules and the number of sparse encoders. The results show that the view weighting module mainly enhances the cross-view performance and that the transformer encoder enhances the overall performance. Furthermore, the number of sparse encoders should be limited according to our study.

Author Contributions: X.L.: Conceptualization, methodology, and writing original draft; L.X., S.Z. and Q.W.: writing assistance. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China, grant number 61876054.

Data Availability Statement: All data, models, or code supporting the results of this study are available from the authors upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
STA-LSTM	Spatio-Temporal Attention Long Short-Term Memory
VA-RNN	View Adaptive Recurrent Neural Network
DS-LSTM	Denoising Sparse Long Short-Term Memory
GCN	Graph Convolutional Network
ST-GCN	Spatio-Temporal Graph Convolutional Network
AS-GCN	Actional–Structural Graph Convolution Network
JTM	Joint Trajectory Maps
SC4D	Sparse 4D Convolutional Network
3D-CNN	3D Convolutional Neural Network
S3D-CNN	Sparse 3D Convolutional Neural Network
VW-SC3D	View Weighting Sparse 3D Convolutional Neural Network
VA-CNN	View Adaptation Convolutional Neural Network
TCN	Temporal Convolutional Network
SAR	Skeleton-based Action Recognition

References

1. Yang, L.; Shan, X.; Lv, C.; Brighton, J.; Zhao, Y. Learning Spatio-Temporal Representations with a Dual-Stream 3-D Residual Network for Nondriving Activity Recognition. *IEEE Trans. Ind. Electron.* **2021**, *69*, 7405–7414. [CrossRef]
2. Dallel, M.; Havard, V.; Baudry, D.; Savatier, X. Inhard-industrial human action recognition dataset in the context of industrial collaborative robotics. In Proceedings of the 2020 IEEE International Conference on Human-Machine Systems (ICHMS), Rome, Italy, 7–9 September 2020; pp. 1–6.
3. Xian, Y.; Rong, X.; Yang, X.; Tian, Y. Evaluation of low-level features for real-world surveillance event detection. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *27*, 624–634. [CrossRef]
4. Deepak, K.; Vignesh, L.; Srivathsan, G.; Roshan, S.; Chandrakala, S. Statistical Features-Based Violence Detection in Surveillance Videos. In *Cognitive Informatics and Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2020; pp.197–203.
5. Karbalaie, A.; Abtahi, F.; Sjöström, M. Event detection in surveillance videos: A review. *Multimed. Tools Appl.* **2022**, *81*, 35463–35501. [CrossRef]
6. Yin, J.; Han, J.; Wang, C.; Zhang, B.; Zeng, X. A skeleton-based action recognition system for medical condition detection. In Proceedings of the 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), Nara, Japan, 17–19 October 2019; pp. 1–4.
7. Wang, P. Research on sports training action recognition based on deep learning. *Sci. Program.* **2021**, *2021*, 3396878. [CrossRef]
8. Sun, Z.; Ke, Q.; Rahmani, H.; Bennamoun, M.; Wang, G.; Liu, J. Human action recognition from various data modalities: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 1–20. [CrossRef] [PubMed]
9. Pareek, P.; Thakkar, A. A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artif. Intell. Rev.* **2021**, *54*, 2259–2322. [CrossRef]
10. Xing, Y.; Zhu, J. *Deep Learning-Based Action Recognition with 3D Skeleton: A Survey*; Wiley Online Library: Hoboken, NJ, USA, 2021.
11. Ren, B.; Liu, M.; Ding, R.; Liu, H. A survey on 3d skeleton-based action recognition using learning method. *arXiv* **2020**, arXiv:2002.05907.
12. Gu, X.; Xue, X.; Wang, F. Fine-grained action recognition on a novel basketball dataset. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2563–2567.
13. Zhang, Z. Microsoft kinect sensor and its effect. *IEEE Multimed.* **2012**, *19*, 4–10. [CrossRef]
14. Xu, L.; Wang, Q.; Yuan, L.; Ma, X. Using trajectory features for tai chi action recognition. In Proceedings of the 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Dubrovnik, Croatia, 25–28 May 2020; pp. 1–6.
15. Cao, Z.; Hidalgo Martinez, G.; Simon, T.; Wei, S.; Sheikh, Y.A. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *30*, 7291–7299. [CrossRef]
16. Song, L.; Yu, G.; Yuan, J.; Liu, Z. Human pose estimation and its application to action recognition: A survey. *J. Vis. Commun. Image Represent.* **2021**, *76*, 103055. [CrossRef]
17. Zheng, C.; Zhu, S.; Mendieta, M.; Yang, T.; Chen, C.; Ding, Z. 3d human pose estimation with spatial and temporal transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 11656–11665.
18. Kumar, N.; Sukavanam, N. Motion trajectory for human action recognition using fourier temporal features of skeleton joints. *J. Image Graph.* **2018**, *6*, 174–180. [CrossRef]
19. Cheng, G.; Wan, Y.; Saudagar, A.N.; Namuduri, K.; Buckles, B.P. Advances in human action recognition: A survey. *arXiv* **2015**, arXiv:1501.05964.

20. Han, T.; Yao, H.; Xie, W.; Sun, X.; Zhao, S.; Yu, J. TVENet: Temporal variance embedding network for fine-grained action representation. *Pattern Recognit.* **2020**, *103*, 107267. [CrossRef]
21. Liu, J.; Wang, G.; Duan, L.Y.; Abdiyeva, K.; Kot, A.C. Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Trans. Image Process.* **2017**, *27*, 1586–1599. [CrossRef] [PubMed]
22. Li, W.; Wen, L.; Chang, M.C.; Nam Lim, S.; Lyu, S. Adaptive RNN tree for large-scale human action recognition. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1444–1452.
23. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1110–1118.
24. Li, S.; Li, W.; Cook, C.; Zhu, C.; Gao, Y. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5457–5466.
25. Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; Tian, Q. Actional-structural graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3595–3603.
26. Cheng, K.; Zhang, Y.; He, X.; Chen, W.; Cheng, J.; Lu, H. Skeleton-based action recognition with shift graph convolutional network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 183–192.
27. Zhu, Q.; Deng, H.; Wang, K. Skeleton Action Recognition Based on Temporal Gated Unit and Adaptive Graph Convolution. *Electronics* **2022**, *11*, 2973. [CrossRef]
28. Panagiotakis, C.; Papoutsakis, K.; Argyros, A. A graph-based approach for detecting common actions in motion capture data and videos. *Pattern Recognit.* **2018**, *79*, 1–11. [CrossRef]
29. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
30. Hou, Y.; Li, Z.; Wang, P.; Li, W. Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *28*, 807–811. [CrossRef]
31. Li, C.; Zhong, Q.; Xie, D.; Pu, S. Skeleton-based action recognition with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 597–600.
32. Ding, Z.; Wang, P.; Ogunbona, P.O.; Li, W. Investigation of different skeleton features for cnn-based 3d action recognition. In Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Hong Kong, China, 10–14 July 2017; pp. 617–622.
33. Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; Zheng, N. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1963–1978. [CrossRef]
34. Caetano, C.; Brémond, F.; Schwartz, W.R. Skeleton image representation for 3D action recognition based on tree structure and reference joints. In Proceedings of the 2019 32nd SIBGRAPI IEEE Conference on Graphics, Patterns and Images (SIBGRAPI), Rio de Janeiro, Brazil, 28–31 October 2019; pp. 16–23.
35. Yang, F.; Wu, Y.; Sakti, S.; Nakamura, S. Make skeleton-based action recognition model smaller, faster and better. In Proceedings of the ACM Multimedia Asia, Beijing, China, 15–18 December 2019; pp. 1–6.
36. Wang, P.; Li, W.; Li, C.; Hou, Y. Action recognition based on joint trajectory maps with convolutional neural networks. *Knowl.-Based Syst.* **2018**, *158*, 43–53. [CrossRef]
37. Caetano, C.; Sena, J.; Brémond, F.; Dos Santos, J.A.; Schwartz, W.R. Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–8.
38. Liu, H.; Tu, J.; Liu, M. Two-stream 3d convolutional neural network for skeleton-based action recognition. *arXiv* **2017**, arXiv:1705.08106.
39. Duan, H.; Zhao, Y.; Chen, K.; Lin, D.; Dai, B. Revisiting skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 2969–2978.
40. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Sc4d: A sparse 4d convolutional network for skeleton-based action recognition. *arXiv* **2020**, arXiv:2004.03259.
41. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
42. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; pp. 38–45.
43. Kalyan, K.S.; Rajasekharan, A.; Sangeetha, S. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv* **2021**, arXiv:2108.05542.
44. Bertasius, G.; Wang, H.; Torresani, L. Is space-time attention all you need for video understanding? In Proceedings of the ICML, Online, 18–24 July 2021; Volume 2, p. 4.

45. Cho, S.; Maqbool, M.; Liu, F.; Foroosh, H. Self-attention network for skeleton-based human action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 635–644.
46. Plizzari, C.; Cannici, M.; Matteucci, M. Skeleton-based action recognition via spatial and temporal transformer networks. *Comput. Vis. Image Underst.* **2021**, *208*, 103219. [CrossRef]
47. Mazzia, V.; Angarano, S.; Salvetti, F.; Angelini, F.; Chiaberge, M. Action Transformer: A self-attention model for short-time pose-based human action recognition. *Pattern Recognit.* **2022**, *124*, 108487. [CrossRef]
48. Zhang, Y.; Wu, B.; Li, W.; Duan, L.; Gan, C. STST: Spatial-temporal specialized transformer for skeleton-based action recognition. In Proceedings of the 29th ACM International Conference on Multimedia, Chengdu, China, 20–24 October 2021; pp. 3229–3237.
49. Ji, X.; Liu, H. Advances in view-invariant human motion analysis: A review. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2009**, *40*, 13–24.
50. Trong, N.P.; Minh, A.T.; Nguyen, H.; Kazunori, K.; Le Hoai, B. A survey about view-invariant human action recognition. In Proceedings of the 2017 56th IEEE Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Kanazawa, Japan, 19–22 September 2017; pp. 699–704.
51. Zhu, W.; Lan, C.; Xing, J.; Zeng, W.; Li, Y.; Shen, L.; Xie, X. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
52. Liu, J.; Shahroudy, A.; Xu, D.; Wang, G. Spatio-temporal lstm with trust gates for 3d human action recognition. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 816–833.
53. Song, S.; Lan, C.; Xing, J.; Zeng, W.; Liu, J. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
54. Shahroudy, A.; Liu, J.; Ng, T.T.; Wang, G. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1010–1019.
55. Ji, Y.; Xu, F.; Yang, Y.; Shen, F.; Shen, H.T.; Zheng, W.S. A large-scale RGB-D database for arbitrary-view human action recognition. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Republic of Korea, 22–26 October 2018; pp. 1510–1518.
56. Gao, L.; Ji, Y.; Kumie, G.A.; Xu, X.; Zhu, X.; Shen, H.T. View-invariant Human Action Recognition via View Transformation Network. *IEEE Trans. Multimed.* **2021**, *24*, 4493–4503. [CrossRef]
57. Tang, H.; Liu, Z.; Li, X.; Lin, Y.; Han, S. TorchSparse: Efficient Point Cloud Inference Engine. *Proc. Mach. Learn. Syst.* **2022**, *4*, 302–315.
58. Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; Zheng, N. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2117–2126.
59. Jiang, X.; Xu, K.; Sun, T. Action Recognition Scheme Based on Skeleton Representation with DS-LSTM Network. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 2129–2140. [CrossRef]
60. Ke, Q.; An, S.; Bennamoun, M.; Sohel, F.; Boussaid, F. Skeletonnet: Mining deep part features for 3-d action recognition. *IEEE Signal Process. Lett.* **2017**, *24*, 731–735. [CrossRef]
61. Ke, Q.; Bennamoun, M.; An, S.; Sohel, F.; Boussaid, F. A new representation of skeleton sequences for 3d action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3288–3297.
62. Liu, J.; Akhtar, N.; Mian, A. Skepxels: Spatio-temporal Image Representation of Human Skeleton Joints for Action Recognition. In Proceedings of the CVPR Workshops, Long Beach, CA, USA, 16–20 June 2019.
63. Banerjee, A.; Singh, P.K.; Sarkar, R. Fuzzy integral-based CNN classifier fusion for 3D skeleton action recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 2206–2216. [CrossRef]