Title: Final Report

G015 (s1919582, s1935167, s1899254)

Abstract

Dynamic hand gesture recognition is a task in computer vision with applications in human-computer interaction, sign language translation. In this report, we propose an improved multi-branch model based on Dynamic Graph-Based Spatial-Temporal Attention (DG-STA) and investigate the effect of different branches on the experimental performance of our model for hand gesture recognition. Due to a very small test set during leave-one-cross-validation evaluation, the performance of DG-STA is quite unstable. We employs a more stable evaluation method and demonstrates higher experimental accuracy on one hand gesture recognition benchmark, the DHG-14/28 Dataset, while maintaining performance on par with the original model on another hand gesture recognition benchmark, the SHREC'17 Track Dataset. To validate the generalization ability of both the DG-STA and our proposed model, we conducted experiments on the human body dataset HumanAct12 as well. The results show that our model outperforms the DG-STA with an accuracy improvement of approximately 5% on this dataset.

1. Introduction

Dynamic hand gesture recognition is a technology that aims to predict and recognize gesture types from videos or picture sequences, making it an essential tool for computer vision and human-computer interaction. It has a wide range of applications, including behavior detection, sign language translation, and wheelchair control. Existing research on hand gesture recognition can be divided into two main categories based on the type of input: image-based and skeleton-based. Image-based methods rely on full image pixels, while skeleton-based methods use 2D or 3D coordinates of hand nodes. Due to the widespread adoption of low-cost depth cameras, such as the Microsoft Kinect or Intel RealSense, the technology of extracting hand nodes through depth images has become mature (Oberweger & Lepetit, 2017). Skeleton-based methods, which rely on keypoints information, are more reliable and accurate than RGB-based methods, which are prone to uncertainties, such as lighting conditions (Romeo et al., 2021).

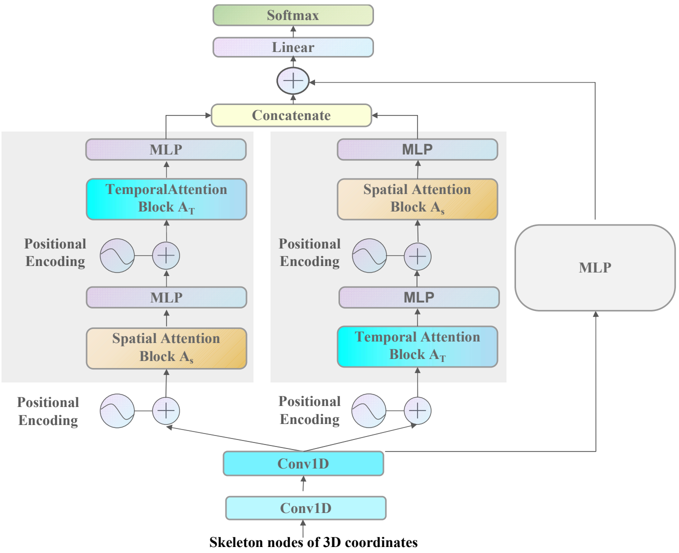
Softmax
Linear
Concatenate
MLP
TemporalAttention Block A_T
Positional Encoding
MLP
Spatial Attention Block A_s
MLP
Spatial Attention Block A_s
Positional Encoding
MLP
Temporal Attention Block A_T
MLP
Positional Encoding
Positional Encoding
Conv1D
Conv1D
Skeleton nodes of 3D coordinates

Figure 1. The illustration of our proposed model, an advanced multi-branch version of DG-STA model.

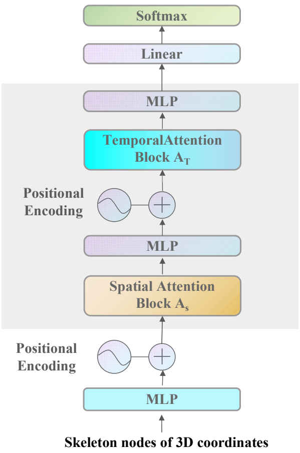
Softmax
Linear
MLP
TemporalAttention Block A_T
Positional Encoding
MLP
Spatial Attention Block A_s
Positional Encoding
MLP
Skeleton nodes of 3D coordinates

Figure 2. The illustration of the DG-STA model (Chen et al., 2019).

Hand node coordinates can be used to extract physical or geometric features, such as distances and keypoint connection angles, to represent nodes. While manually constructed features have shown good results (Zhang et al., 2018; Hu et al., 2022), they have limited generalization ability and are computationally expensive to obtain. To overcome these limitations, researchers have turned to deep neural network architectures, such as LSTM (Hochreiter & Schmidhuber, 1997) and transformer (Vaswani et al., 2017), for automatic feature extraction in action recognition. The self-attention mechanism used in transformer has been particularly effective in finding temporal and spatial information between node sequences, allowing for the extraction of feature representations of the graph spanning time and structure. Recent studies (Chen et al., 2019; Shi et al., 2020) have leveraged graph-based temporal/spatial self-attention mechanisms to improve the accuracy of action recognition.

Chen *et al.* (Chen et al., 2019) introduced the Dynamic Graph-Based Spatial-Temporal Attention (DG-STA) method for hand gesture recognition. DG-STA constructs an undirected graph of nodes, including temporal edges, spatial edges, and self-connected edges, and uses a self-attention mechanism to learn the spatial and temporal features of the graph with the spatial-temporal position embeddings of the nodes. To improve computation efficiency, they proposed a spatial-temporal mask matrix to focus on the spatial and temporal information in turn.

However, Chen *et al.* (Chen et al., 2019) did not investigate the impact of the order of spatial and temporal attention blocks on the results of DG-STA, nor did they explore alternative methods for projecting 3D points into position embeddings. As shown in Fig. 2, DG-STA has a fixed order of spatial and temporal attention blocks. If we prioritize extracting the temporal information of the graph, when the spatial correlations between frames are strong enough, these learned temporal relations will include sufficient spatial information. For instance, when the hand moves slowly, such as in Swipe actions in the DHG-14/28 dataset (De Smedt et al., 2016). After this, the learned embeddings are fed into the spatial attention block, which may cause the model to overfit on spatial patterns. Conversely, if we first extract the spatial features of the graph, the new embeddings may focus more on temporal information since the hand structure remains nearly unchanged. Therefore, we hypothesize that different orders of temporal and spatial attention blocks may lead to differences in the extracted feature representations.

In this report, we investigate and justify whether the integration of diverse space and time arrangements for the purpose of extracting additional information serves as an effective improvement strategy. Additionally, while Chen *et al.* directly projected 3D points into 128 dimensions through a linear layer to obtain the skeleton node embeddings, we propose that this step might be a bottleneck and that there may be a better processing method if we increase the network capacity. Thus, we propose an improved model (Fig. 1) based on DG-STA that achieves higher experimental accuracy in hand gesture recognition benchmarks (DHG-14/28 Dataset (De Smedt et al., 2016) and keepSHREC'17 Track Dataset (De Smedt et al., 2017)). Furthermore, we conduct experiments on human action recognition task (HumanAct12 Dataset (Guo et al., 2020)) to examine the generalization ability of our method across different domains.

In a nutshell, our project has the following objectives and attribution:

- The proposition and implementation of an advanced multi-branch model based on the DG-STA. The improvement components of this model encompass the integration of a temporal-spatial fusion module, modifications to the keypoint embedding, and the incorporation of a residual network module;
- Justification of the effectiveness of different branches used in the proposed model, including the order of spatial/temporal attention;
- Modification of the original unstable evaluation methods;
- Exploring the generalization ability of DG-STA and our new modified network in datasets on different action categories.

2. Data set and task

In this project, we mainly use these three data set: DHG-14/28 Dataset (De Smedt et al., 2016), SHREC'17 Track Dataset (De Smedt et al., 2017) and HumanAct12 Dataset (Guo et al., 2020).

The DHG-14/28 Dataset serves as a test target to evaluate the effectiveness of our model on gesture recognition. We chose this dataset because our baseline model also uses this dataset, which makes it easier for us to compare the performance of our model against the baseline. The dataset contains sequences of 14 hand gestures performed in two ways: using one finger and using the whole hand. It provides sequences of hand skeleton in addition to the depth image, which was generated using the geometry of the hand to extract valid descriptors from the skeletally connected nodes of the hand returned by the Intel RealSense depth camera. The original videos were captured from 20 right-handed participants who performed each gesture 5 times in both ways, resulting in 2800 sequences. These sequences are labeled according to the gesture, the number of fingers used, the performer, and trials. As shown in Fig. 3 from (De Smedt et al., 2016), each frame contains a depth image and coordinates of 22 nodes in 2D depth image space and 3D world space, which together form a complete hand skeleton. To train and test our model on this dataset, we apply cross-validation to split the dataset into training and testing sets. We take the data of one participant at a time as the test set to evaluate the accuracy of our model. We repeat this process for all 20 participants and take the average of the results as the performance of our model. To ensure a fair comparison, we apply data augmentation techniques from (Chen et al., 2019), including adding noise, shifting, scaling, and other operations. This helps to increase the variability of the data and improve the generalization ability of our model.

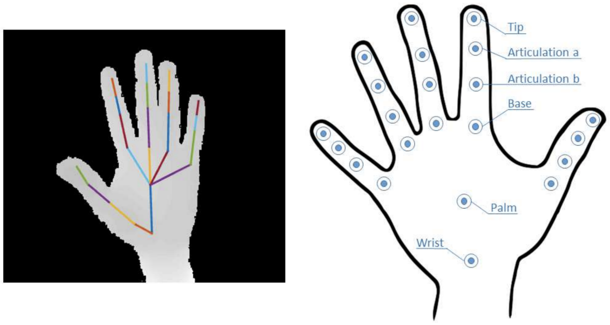

Figure 3. Depth and hand skeleton of the DHG-14/28 Dataset and SHREC'17 Dataset. Taken from (De Smedt et al., 2016; 2017)

The SHREC'17 Dataset serves as a testing target for evaluating the effectiveness of models in gesture recognition, and is similar to the DHG-14/28 Dataset in this regard. The data is extracted using the same methodology as the DHG-14/28 Dataset, as described in (De Smedt et al., 2017). However, unlike the DHG-14/28 Dataset, where each gesture is performed five times in two ways by each participant, the SHREC'17 Dataset involves random performance of each gesture 1-10 times. This results in a dataset of the same size as the DHG-14/28 Dataset, but with a different data distribution, allowing for the evaluation of models on unevenly distributed datasets. While the original paper does not specify the partitioning of the dataset, it provides two files with lists of training and testing set IDs, with the training set containing 1960 3D coordinates sequences and the testing set containing 840 sequences. In our experiments, we use these provided files as partition of training set and testing set.

For all of the aforementioned datasets, we evaluate models' performance based on the accuracy of their predictions, which is calculated as the number of correct predictions divided by the total number of predictions.

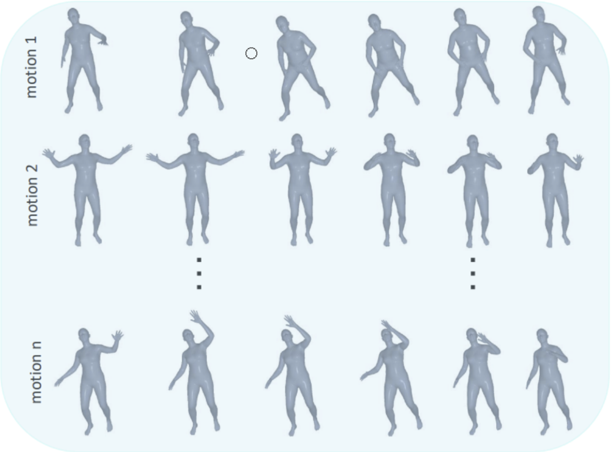

Figure 4. Some "Warm up" action classes and their sub-classes in the HumanAct12 Dataset. Taken from (Guo et al., 2020)

The HumanAct12 Dataset is commonly used as a test target for evaluating the effectiveness of models in human motion recognition. In our project, we use this dataset to examine the generalization ability of our model in recognizing actions in addition to hand gestures. HumanAct12 Dataset comprises 1191 3D motions composed of a total of 90099 poses, classified into 12 action classes and 34 fine-grained sub-classes. The 12 action classes include our daily motions, such as walking, running, and sitting down, while the 34 fine-grained sub-classes are subdivisions and extensions of the 12 basic movements. For example, Fig. 4 shows the "Warm up" action class has several sub-classes, such as "Warm up by bowing left side." To create our training and testing sets, we split the data according to the 12 classes, randomly selecting 20% of each class's data as the testing set and using the remainder for the training set. This resulted in a training set of 935 motions and a testing set of 256 motions.

3. Methodology

Our proposed model is a general multiple-branch network architecture designed for feature extraction of skeleton nodes. It builds upon the DG-STA model, which is a specific architecture for skeleton-based hand gesture recognition. Fig. 1 and Fig. 2 illustrate a comparison between our model and DG-STA.

Section 3.2 outlines our approach for extracting feature representations from the skeleton nodes, which are represented by 3D coordinates. In Section 3.2, we describe how we construct a dynamic fully-connected graph using the extracted feature representations. To capture the temporal and spatial information in the graph, we employ two distinct self-attention blocks (Vaswani et al., 2017), which we describe in Section 3.3. Special position embeddings are introduced in Section 3.4, while mask operations are detailed in Section 3.5.

In Section 3.6, we demonstrate how we utilize these techniques to design multiple branches to learn feature representations of nodes. The final feature representation is obtained by incorporating the outputs of these branches through an average pooling layer, which is then used for action classification.

3.1. Conv1D-based Feature Extraction

Section 3.2 discusses our approach for feature extraction using Conv1D-based architecture. Previous studies have achieved impressive results in recognizing hand gestures through manually built geometric and physical features computed from skeleton node coordinates (Zhang et al., 2018; Hu et al., 2022). However, we believe that automatic feature extraction can capture more complex patterns and relationships within the data with much fewer labor cost.

In addition, we chose Conv1D over linear projection, as it is commonly more effective in capturing complex patterns and relationships in sequential data (Yang et al., 2019;

Do et al., 2020). Therefore, we employed two sequential Conv1D layers to automatically learn feature representations of skeleton nodes. The whole process results in a feature representation of each node, converting 3D coordinates to a 128-dimensional vector $\boldsymbol{F}$. The overview of this step is show in Fig. 5.

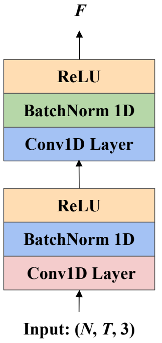

Figure 5. The architecture of the Conv1D-based feature extraction step. The whole process extracts a feature embedding of each node, converting 3D coordinates to a 128-dimensional vector $\boldsymbol{F}$. The first Conv1D layer, with input channels as 3 and output channels as 128, automatically learns geometrical and physical features from the 3D coordinates of each node, resulting in a 128-dimensional feature representation of each keypoint. The second Conv1D layer, holding both input and output channels as 128, learns the connections between these extracted features to obtain an advanced 128-dimensional feature representation for each node.

3.2. Construct A Dynamic Graph

In Section 3.2, we describe how we construct a dynamic fully-connected graph based on the feature representations obtained in Section 3.2. Specifically, we extract N nodes from each of the T frames of a video to represent a hand skeleton sequence, and then build a fully-connected skeleton graph $G = (V, E)$ from this sequence. The node set V is denoted by $V = v_{(t,i)}|t = 1, ..., T, i = 1, ..., N$, where $v_{(t,i)}$ denotes the i-th hand keypoint at time step t. The nodes are represented by $F = \{f_{(t,i)}|t = 1, ..., T, i = 1, ..., N\}$, where $f(t, i)$ is the feature vector of the node $v_{(t,i)}$. This vector is what we obtained from 1 as a 128-dimentional vector $\boldsymbol{F}$.

Following the approach of Chen *et al.* (Chen et al., 2019), we define three types of edges on the edge set E as follows:

- A spatial edge $v_{(t,i)} \to v_{(t,j)}$ $(i \neq j)$ connecting two distinct nodes from the same frame.
- A temporal edge $v_{(t,i)} \to v_{(k,j)}$ $(t \neq k)$ connecting two nodes from the different frames.
- A self-connected edge $v_{(t,i)} \to v_{(t,i)}$ as a self-loop for each node.

In the context of the attention mechanism, the weights of the edges in this graph represent the attention weights among different nodes, and these weights are obtained at different frames, namely a dynamic graph, according to the distinct feature vectors $f_{(t,i)}$.

3.3. Spatial and Temporal Attention Blocks

In this subsection, we describe the construction of the dynamic graph (attention weights) in DG-STA, which is designed to incorporate spatial and temporal information for node embeddings. DG-STA consists of two self-attention blocks: the spatial attention block (A_S) and the temporal attention block (A_T), both of which are based on multi-head attention (Vaswani et al., 2017) for processing multiple patterns of features simultaneously.

In the spatial self-attention blocks $\mathbf{A}_S$, for each node $v_{(t,i)}$ in the constructed graph, the h-th spatial attention head maps the feature embedding $f_{(t,i)}$ into the key, query, and value vectors using three linear layers. The corresponding weight matrices of these layers are W^h_K, W^h_Q, and W^h_V, respectively. The key, query, and value vectors are obtained by:

$$K^h_{(t,i)} = W^h_K f_{(t,i)}, Q^h_{(t,i)} = W^h_Q f_{(t,i)}, V^h_{(t,i)} = W^h_V f_{(t,i)}. \quad (1)$$

Each spatial self-attention head in DG-STA is responsible for finding the spatial relationships among the nodes and computing the weights of the spatial and self-connected edges in two steps. Firstly, the "scaled dot-product" is implemented to calculate the attention weights between the query vectors and key vectors of the nodes within the same time step. Then, it normalizes the results using the Softmax function. The equations for these two steps are:

$$u^h_{(t,i)\to(t,j)} = \frac{\langle Q^h_{(t,i)} \cdot K^h_{(t,j)} \rangle}{\sqrt{d}} \quad (2)$$

,

$$\alpha^h_{(t,i)\to(t,j)} = \frac{\exp(u^h_{(t,i)\to(t,j)})}{\sum_{n=1}^{N} \exp(u^h_{(t,i)\to(t,n)})} \quad (3)$$

,

where $u^h_{(t,i)\to(t,j)}$ is the scaled dot product of the node $v_{(t,i)}$ and $v_{(t,j)}$. $\langle \cdot, \cdot \rangle$ represents the inner product operation. d is the dimension of the key, query, and value vectors, which is 128 in our model; $\alpha^h_{(t,i)\to(t,j)}$ is the attention weight between the node $v_{(t,i)}$ and $v_{(t,j)}$, which measures the spatial information from node $v_{(t,i)}$ to node $v_{(t,i)}$ in this frame t. To focus on the spatial information of the hand, we set the weights for all temporal edges to 0, effectively ignoring the information passing in the temporal domain. Therefore, each attention head in DG-STA learns a weighted skeleton graph (attention weights) that represents a specific pattern of spatial structure of the hand, with the temporal information ignored. The attention head calculates the spatial attention feature of the node $v_{(t,i)}$ as the weighted sum of the value vectors within the same time step, which is defined as:

$$\bar{f}^h_{(t,i)} = \sum_{j=1}^{N} \alpha^h_{(t,i)\to(t,j)} V^h_{(t,j)} \quad (4)$$

,

where N is the number of nodes in the graph. $\bar{f}^h_{(t,i)}$ is feature representation of the node $v_{(t,i)}$ incorporating a type of spatial information obtained from one head (h).

The computation mechanism for spatial attention features is an intuitive process in which each node in the graph collects information from other nodes within the same time step and assigns edge weights. This mechanism allows the model to selectively attend to important information within the spatial structure of the hand, capturing multiple types of structural information represented by the weighted skeleton graphs learned by different spatial attention heads. Finally, we concatenate the spatial attention features learned by all spatial attention heads into a single feature vector, denoted as $\widetilde{f_{(t,i)}}$. This vector serves as the spatial feature of the node $v_{(t,i)}$, and is computed as follows:

$$\widetilde{f_{(t,i)}} = \text{Concatenate}[\bar{f}^{1}_{(t,i)}), \bar{f}^{2}_{(t,i)}, ..., \bar{f}^{H}_{(t,i)}] \tag{5}$$

where H is the number of spatial attention heads, which is 8 in our model. As a result, the resulting node features capture a rich set of spatial characteristics of the hand.

The temporal attention block $\mathbf{A}_T$ employs the same multi-head self-attention mechanism as the spatial attention block $\mathbf{A}_S$, but operates in the temporal domain. It takes the node embeddings $f_{(t,i)}$ as input, and allows nodes to selectively focus on finding information from other frames while dismissing information from the same frame.

3.4. Position Embeddings for Spatial and Temporal attention

To incorporate spatial and temporal identity information into the original node features $\boldsymbol{F}$ extracted from the hand skeleton 3D coordinates, Chen *et al.* (Chen et al., 2019) introduced the spatial and temporal position embeddings. The spatial position embedding consists of N vectors, with each vector representing a skeleton node, while the temporal position embedding is composed of a $N \times T$ matrix, each row vector of which corresponds to a node in T frames. Prior to feeding the node embedding into the temporal/spatial attention blocks, the feature vector of a specific node is added with the corresponding spatial and temporal position embedding vectors.

The spatial and temporal position embeddings can be set using the sine and cosine functions of different frequencies, as proposed in (Vaswani et al., 2017; Devlin et al., 2019). By adding the spatial-temporal position embedding to the original node features, the model is able to capture both spatial and temporal identity information for each node in the hand skeleton graph. In analogy to Natural Language Processing, where N nodes can be treated as tokens and T frames as sentences, the use of spatial and temporal position embeddings helps the model to learn more robust representations of the hand movements.

3.5. Mask for Spatial and Temporal Attention

In DG-STA, unique masking techniques were developed to more efficiently and separately extract spatial and temporal information from the node feature embeddings in $\mathbf{A}_S$ and $\mathbf{A}_T$ (Fig. 6). These mask operations also significantly reduce computation time by 99

For the spatial domain, only the edges between nodes in the same frame are used to extract spatial information. To prevent one node from attending to nodes from other frames, those nodes are masked. The spatial mask, denoted by M_S, is a binary matrix with elements equal to 1 for spatial connections and 0 otherwise. The equation for the spatial domain operation is:

$$W_S = \phi(W \odot M_S + (1 - M_S) \times \eta), \tag{6}$$

where ϕ represents the Softmax function, and $\odot$ denotes element-wise multiplication. W denotes the result after scaled dot-products. The term $(1 - M_S)$ masks the temporal connections, which times with η, a big negative number like -9×10^{15}. As a result, after applying the Softmax function, the weights of temporal connections in W_S will be close to 0, effectively normalizing weights only along the spatial domain..

Similarly, in the temporal domain, the nodes in one frame only attend to nodes from other time steps. Therefore, the nodes from one time step are masked out with logits set as a big negative number like -9×10^{15}. After applying the Softmax function, the weights of spatial edges inside one frame will be 0.

3.6. Multi-branches and Feature Fusion

Considering that the order of the attention blocks might have an influence on the feature embeddings learned, we design two attention-based branches to incorporate the temporal and spatial information in the sequence. The first branch enters the spatial attention block after we obtain the feature embeddings from Conv1D adding with the position information. One attention-based branch will firstly enter the spatial attention block $\mathbf{A}_S$, followed by a point-wise multilayer perceptron (MLP) bringing in non-linearity, and then temporal attention block $\mathbf{A}_T$, and we denote the feature representations learned from this branch as $\boldsymbol{F}_{ST}$. The other one will change this order and obtain the feature embeddings denoted as $\boldsymbol{F}_{TS}$. The MLP employed between attention blocks is shown in Fig. 7.

The third branch consists of a residual MLP that is added to the concatenated feature embeddings from the attention-based branches. The feature representation learned from this branch is denoted as $\boldsymbol{F}_{RES}$. Fig. 8 illustrates this residual MLP.

The multi-branch feature representation F_{multi} is obtained by concatenating the feature embeddings obtained from the attention-based branches and adding them to the feature embeddings learned from the residual layer, as follows:

$$F_{multi} = \text{Concatenate}[\boldsymbol{F}_{ST}, \boldsymbol{F}_{TS}] + \boldsymbol{F}_{RES}. \tag{7}$$

The resulting F_{multi} is then fed into an average-pool layer to obtain a vectorized feature embedding that can be fed

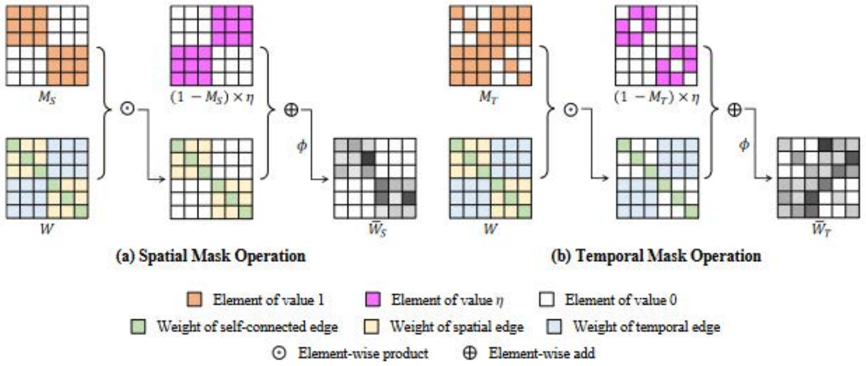

Figure 6. The illustration of mask techniques used in our model. In these matrices, each block denotes a edge weight, namely attention weights from one node in a frame to other nodes in/out the same frame. For instance, in M_s on the top-left corner, the value in (0, 3) block means the attention weight between the skeleton node in the first frame to the next time step. Since this is a spatial attention block, this edge weight will be masked out as 0. Proposed and taken from (Chen et al., 2019)

into a fully-connected layer for classification. The overall architecture is summarized in Fig. 1.

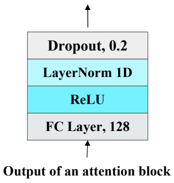

Figure 7. The architecture of the MLP after each attention block. Hyperparameters follow DG-STA.

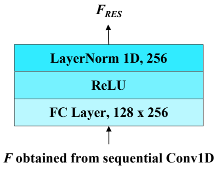

Figure 8. The architecture of the residual MLP. This network extracts a feature embedding of each node, which is denoted as $\boldsymbol{F}_{RES}$.

4. Experiments

4.1. Implementation Details

Based on the DG-STA model (baseline model), we changed the order of the spatial and temporal attention blocks and examine if the order of the spatial and temporal attention can significantly affect the performance. Additionally, we applied the improvement techniques mentioned in the methodology to our model and will compare the differences between our proposed model and the baseline model. As for the generalization ability, although the DG-STA model is designed for the hand gesture recognition task, we believe that both DG-STA and our model have the ability to recognize actions for other objects with skeleton representations. Therefore, we used the HumanAct12 Dataset to examine the baseline and our modified model's ability to recognize human actions, thus verifying these models' generalization ability and the possibility of transfer learning in the future work.

DG-STA is our primary comparative baseline model and we reproduced it on both the and DHG-14/28 Dataset and the SHREC'17 Tracking Dataset using the same structure, hyperparameters, and dataset splitting introduced in DG-STA. All experiments were conducted using the Pytorch framework (Paszke et al., 2017). We followed the hyperparameters settings from the original paper of DG-STA, as this model has shown high performance across several benchmarks. Specifically, we employed the Adam optimizer (Kingma & Ba, 2017) with a learning rate of 0.001 and a batch size of 32. We also applied a Dropout rate (Srivastava et al., 2014) of 0.2 during training.

4.2. Proposed Evaluation Method

For the evaluation metrics, we used the same ones employed in DT-STA. However, we found that there is a discrepancy in the empirical results obtained from our experiments compared to the original paper's on the DHG-14/28 Dataset. We suppose that the difference in GPUs used in the experiments probably cause this variation. Furthermore, in the original paper of DG-STA, the authors used a leave-one-subject-out cross-validation strategy, conducting a single experiment per subject in this dataset, where each subject is chosen for testing while the other 19 subjects are utilized for training. However, we found that the dataset only contains 2800 training samples, which are evenly distributed over 20 objects, resulting in a small validation set size of 140 samples. A small validation set size can lead to unreliable performance evaluation of the model due to the impact of chance. As we use accuracy as the evaluation criterion, every accidental correct classification in a validation set of size 140 increases the performance by nearly 1%. These results can be affected by the initialization of random parameters and training time. Furthermore, in the original paper of DG-STA, an early stopping mechanism is employed with a patience value set to 50. Thus, during the later epochs of training, when convergence has already been achieved, oscillations in the performance curve may generate unintended performance improvements, which should be considered unreliable.

To address these issues, we performed each experiment four times on the DHG-14/28 Dataset and took the average of the four values as the final experimental result. The choice of four is due to the fact of limited time for this project. We believe that our evaluation approach provides a more accurate reflection of the model's performance compared to that of the original paper's. In the original paper, we suppose that the reported results may be derived from the highest values obtained through multiple trials.

4.3. Performance of models on Hand Recognition Datasets

For the DHG-14/28 Dataset, the four results obtained from the original model are showed in Table 1. The result of our modified model is showed in Table 2. Although the performance or our proposed method varied as well in the four experiments, our model achieved a better average accuracy, which is our proposed new evaluation metric on the DHG Dataset for both 14 and 28 gestures recognition. Our model's averaged accuracy is 91.7% and 88.2%, respectively, while DG-STA obtained 91.2% and 88.1%.

EXPERIMENT	14 GESTURES	28 GESTURES
TEST1	91.2	88.0
TEST2	91.8	88.1
TEST3	91.0	88.0
TEST4	90.7	88.2
AVERAGE	91.2	88.1

Table 1. The experiment results of the DG-STA model's on the DHG-14/28 Dataset.

EXPERIMENT	14 GESTURES	28 GESTURES
TEST1	91.9	88.0
TEST2	92.3	88.4
TEST3	91.2	88.1
TEST4	91.3	88.2
AVERAGE	**91.7**	**88.2**

Table 2. The experiment results of our proposed model's on the DHG-14/28 Dataset.

For the SHREC'17 Dataset, because origin paper did not use leave-one-subject-out cross-validation and the validation size is much bigger than that of the DHG-14/28 Dataset, we keep the final accuracy as our evaluation metric. The experimental results on the SHREC'17 Dataset is showed in Table. 3. Our proposed model achieved a higher accuracy on the 28 hand gesture recognition task while held a little lower accuracy on the 14 hand gesture recognition task than that of the DG-STA's.

EXPERIMENT	DG-STA	OUR MODEL
14 GESTURES	**93.1**	92.4
28 GESTURES	87.9	**89.0**

Table 3. The experimental results of the DG-STA model and our proposed model on the SHREC'17 Dataset.

4.4. Ablation Study

Although we achieved better and more stable experimental results for the DHG-14/28 and SHREC‘17 datasets in the 28 hand gesture recognition task, it is difficult to determine which part of our multi-branch spatial-temporal model contributed to the improved performance. Therefore, we conducted ablation experiments to isolate the impact of each branch on the model's performance. By gradually incorporating the features extracted from the extra branches into the baseline model, which uses spatial-temporal attention (STA) as its attention flow, we compared the performance with that of the DG-STA model. Due to the lengthy experimental process and limited time (one experiment takes over 20 hours), we only conducted experiments on the DHG-14 dataset. However, we suppose that the results obtained from this task demonstrate the influence in a certain extend. The experimental results are presented in Table 4.

Models	STA	STA+TSA	CONV-STA+TSA+RES
test1	91.2	90.7	91.9
test2	91.8	91.0	92.3
test3	91.0	91.2	91.2
test4	90.7	90.7	91.3
average	91.2	90.1	**91.7**

Table 4. Ablation study (accuracy %) on the DHG-14 Dataset. In this table, the STA model denotes the DG-STA model with a Multilayer Perceptron (MLP) to project the 3D coordinates of nodes. STA and TSA represent attention-based branches with the attention flow as spatial-temporal attention (STA) and temporal-spatial attention (TSA), respectively. CONV denotes the use of Conv1D to project 3D coordinates, and RES represents the residual MLP branch. Our multi-branch model achieved the highest accuracy compared to the other models.

Table 4 shows that using only the proposed STA and TSA branches does not significantly improve the model performance. However, the new model, which combines the two attention branches with residual layers, achieves better stability and performance. Based on these results, we confirm that the order of temporal attention and spatial attention may not significantly affect feature extraction because they extract mostly similar features. Therefore, combining the two orders of attention flow is not significantly different from the DG-STA model. The improvement in the new model's performance may be attributed to the combination of deeper network (Conv1D) for keypoint embeddings and residual layers, or merely one of them. Furthermore, the two attention based branches may need to be combined with the residual layers to prevent gradient vanishing and missing patterns in the mask, enabling the effective extraction of new features. However, we did not have sufficient time to verify the validity and accuracy of these two conjectures. Further exploration will be conducted in future work.

4.5. Generalization Ability

One of our primary objectives in this project is to enhance the generalization ability of DG-STA. Spatial and temporal attention is advantageous in that it does not require information closely related to the hand, potentially making it useful for dynamic action recognition of other objects represented by keypoints. Dynamic action recognition based on keypoints includes human dynamic action recognition and dynamic facial expression recognition, in addition to hand gestures. To demonstrate the broad applicability of our model, we focused on human action recognition and used the HumanAct12 dataset, which contains dynamic human actions expressed by 3D keypoint sequences. We compared the performance of DG-STA and our model with the state-of-the-art method MDM (Tevet et al., 2022), and the results are presented in Table 5.

DG-STA	OUR MODEL	MDM
83.98	89.06	99.00

Table 5. experiment on HumanAct12 Dataset.

Our new model's performance on the HumanAct12 Dataset exhibits a significant improvement compared to the baseline model. Even though our model is not as good as the state-of-the-art (SOTA) results, 5% increase from the baseline model is a considerable improvement, especially considering the size of the test set, which is 240. The results demonstrate that our model is capable of handling other action recognition tasks beyond hand gesture recognition.

5. Related Work

In this section, we review related work about the skeleton-based hand gesture recognition and explain how they motivated us. We also provide some idea for future work since we did not have sufficient time to finish them.

Initial research on skeleton-based hand recognition involved manually extracting hand features, with much of the focus on how to obtain unique characteristics of the hand. (De Smedt et al., 2016) proposed exploiting the geometric shape of the hand to obtain descriptors of the hand skeleton. Researchers have manually construct physical or geometric features of nodes in the real world, such as building features based on distance and keypoint connection angles (Zhang et al., 2018; Hu et al., 2022). Previous studies have shifted experimental emphasis towards deep neural networks composed of Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). (Chen et al., 2019) proposed a new motion feature augmented recurrent neural network which is also evaluated in the DHG-14/28 Dataset. (Nunez et al., 2018) used CNN and LSTM to extract the spatial and temporal information in the DHG-14/28 Dataset. Some Researchers have begun to employ graph convolution techniques to process the spatial-temporal information contained within hand skeletons (Yan et al., 2018).

Currently, the transformer architecture and self-attention mechanisms have become a hot topic, DG-STA is the first to introduce self-attention mechanisms into hand gesture recognition. The DG-STA researchers proposed the basic framework used in our model, including the spatial-temporal graph architecture and spatial-temporal masking mechanisms mentioned in our approach. Inspired by (Miah et al., 2023), we believe that the spatial-temporal order might be important. However, unlike them, we use Conv1D layers to project the 3D coordinates instead of using linear layers. Additionally, we implemented a residual MLP branch to further fuse features, improving the performance.

6. Conclusions

In summary, our proposed model has achieved higher accuracy on some hand gesture recognition benchmarks compared to the baseline model (DG-STA). We have also demonstrated that our advanced model improves the generalization ability of the baseline model. Our ablation study has revealed that the order of the attention flows does not significantly affect the model's performance, but the Conv1D layers for feature extraction and residual networks play crucial roles in achieving stable and effective performance.

We learned that: transformer architecture is a powerful and generative architecture for sequence-to-sequence tasks. The use of residual layers in these attention blocks probably is essential for achieving stable and effective performance; the choice of evaluation metrics should be based on the specific goals and experiments to obtain a fair comparison with other models. For example, when evaluating on a small test set, the results may contain considerable uncertainty, and we need to find ways to mitigate this uncertainty. This also suggests that we should pay much attention to the dataset itself as well when conducting an experiment.

In the future, we plan to examine our model's generalization ability on more classification tasks and conduct further ablation experiments to determine which single approach is responsible for the improvement on the empirical results.

References

Chen, Yuxiao, Zhao, Long, Peng, Xi, Yuan, Jianbo, and Metaxas, Dimitris N. Construct dynamic graphs for hand gesture recognition via spatial-temporal attention, 2019. URL https://arxiv.org/abs/1907.08871.

De Smedt, Quentin, Wannous, Hazem, and Vandeborre, Jean-Philippe. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference* *on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

De Smedt, Quentin, Wannous, Hazem, Vandeborre, Jean-Philippe, Guerry, Joris, Le Saux, Bertrand, and Filliat, David. Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *3DOR-10th Eurographics Workshop on 3D Object Retrieval*, pp. 1–6, 2017.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Do, Nhu-Tai, Kim, Soo-Hyung, Yang, Hyung-Jeong, and Lee, Guee-Sang. Robust hand shape features for dynamic hand gesture recognition using multi-level feature lstm. *Applied Sciences*, 10(18):6293, 2020.

Guo, Chuan, Zuo, Xinxin, Wang, Sen, Zou, Shihao, Sun, Qingyao, Deng, Annan, Gong, Minglun, and Cheng, Li. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2021–2029, 2020.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Hu, Qihao, Gao, Qing, Gao, Hongwei, and Ju, Zhaojie. Skeleton-based hand gesture recognition by using multi-input fusion lightweight network. In *Intelligent Robotics and Applications: 15th International Conference, ICIRA 2022, Harbin, China, August 1–3, 2022, Proceedings, Part I*, pp. 24–34. Springer, 2022.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization, 2017.

Miah, Abu Saleh Musa, Hasan, Md Al Mehedi, and Shin, Jungpil. Dynamic hand gesture recognition using multi-branch attention based graph and general deep learning model. *IEEE Access*, 2023.

Nunez, Juan C, Cabido, Raul, Pantrigo, Juan J, Montemayor, Antonio S, and Velez, Jose F. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, 76:80–94, 2018.

Oberweger, Markus and Lepetit, Vincent. Deepprior++: Improving fast and accurate 3d hand pose estimation. *CoRR*, abs/1708.08325, 2017. URL http://arxiv.org/abs/1708.08325.

Paszke, Adam, Gross, Sam, Chintala, Soumith, Chanan, Gregory, Yang, Edward, DeVito, Zachary, Lin, Zeming, Desmaison, Alban, Antiga, Luca, and Lerer, Adam. Automatic differentiation in pytorch. 2017.

Romeo, Laura, Marani, Roberto, Malosio, Matteo, Perri, Anna G, and D'Orazio, Tiziana. Performance analysis of body tracking with the microsoft azure kinect. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, pp. 572–577. IEEE, 2021.

Shi, Lei, Zhang, Yifan, Cheng, Jian, and Lu, Hanqing. Decoupled spatial-temporal attention network for skeleton-based action recognition, 2020. URL https://arxiv.org/abs/2007.03263.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15 (56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Tevet, Guy, Raab, Sigal, Gordon, Brian, Shafir, Yonatan, Cohen-Or, Daniel, and Bermano, Amit H. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

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

Yan, Sijie, Xiong, Yuanjun, and Lin, Dahua. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yang, Fan, Wu, Yang, Sakti, Sakriani, and Nakamura, Satoshi. Make skeleton-based action recognition model smaller, faster and better. In *Proceedings of the ACM multimedia asia*, pp. 1–6. 2019.

Zhang, Songyang, Yang, Yang, Xiao, Jun, Liu, Xiaoming, Yang, Yi, Xie, Di, and Zhuang, Yueting. Fusing geometric features for skeleton-based action recognition using multilayer lstm networks. *IEEE Transactions on Multimedia*, 20(9):2330–2343, 2018.