Construct Dynamic Graphs for Hand Gesture Recognition via Spatial-Temporal Attention

Yuxiao Chen[1]
yc984@cs.rutgers.edu
Long Zhao[1]
lz311@cs.rutgers.edu
Xi Peng[2]
xipeng@udel.com
Jianbo Yuan[3]
jyuan10@cs.rochester.edu
Dimitris N. Metaxas[1]
dnm@cs.rutgers.edu

[1] Department of Computer Science
Rutgers University
New Jersey, USA

[2] Department of Computer and Information Sciences
University of Delaware
Delaware, USA

[3] Department of Computer Science
University of Rochester
New York, USA

Abstract

We propose a Dynamic Graph-Based Spatial-Temporal Attention (DG-STA) method for hand gesture recognition. The key idea is to first construct a fully-connected graph from a hand skeleton, where the node features and edges are then automatically learned via a self-attention mechanism that performs in both spatial and temporal domains. We further propose to leverage the spatial-temporal cues of joint positions to guarantee robust recognition in challenging conditions. In addition, a novel spatial-temporal mask is applied to significantly cut down the computational cost by 99%. We carry out extensive experiments on benchmarks (DHG-14/28 and SHREC'17) and prove the superior performance of our method compared with the state-of-the-art methods. The source code can be found at https://github.com/yuxiaochen1103/DG-STA.

1 Introduction

Hand gesture recognition has been an active research area due to its wide range of applications such as human computer interaction, gaming and nonverbal communication analysis including sign language recognition [20, 28, 37]. Previous work can be classified into two categories based on the modality of their inputs: image-based [12, 19, 38] and skeleton-based [5, 8, 9, 13, 21] methods. Image-based methods take RGB or RGB-D images as inputs and rely on image-level features for recognition. On the other hand, skeleton-based methods make predictions by a sequence of hand joints with 2D or 3D coordinates. They are more robust to varying lighting conditions and occlusions given the accurate joint coordinates. Thanks to the low-cost depth cameras (*e.g.*, Microsoft Kinect or Intel RealSense) and great progress made on hand pose estimation [22, 23, 24], accurate coordinates of hand joints are easy to be obtained. Therefore, we follow the skeleton-based method in this work.

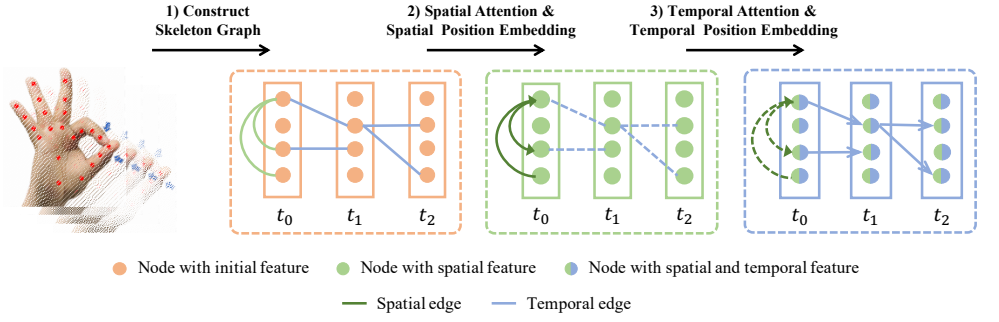

Figure 1: Illustration of our method. The nodes in the graph correspond to hand joints and the dashed lines represent disconnected edges. The proposed DG-STA calculates edge weights and learn node features in both spatial and temporal domains of the hand skeleton graph.

Conventional methods [8, 9, 25] of skeleton-based hand gesture recognition aim to design powerful feature descriptors to model the action of hands. However, these hand-crafted features have limited generalization capability. Recent studies [5, 13, 21] have achieved significant improvement using deep learning. They usually concatenate the joint coordinates into a tensor which is fed into a neural network, and then hand features are directly learned by the network during training. Nevertheless, the spatial structures and temporal dynamics of hand skeletons are not explicitly exploited in these deep learning based approaches.

More recent studies [29, 39, 42] attempt to incorporate structures and dynamics of skeletons based on skeleton graphs. Specifically, given a sequence of skeletons, they define a spatial-temporal graph where structures and dynamics of skeletons are embedded. The feature representation of the graph is then extracted for action recognition. However, a predefined graph with fixed structure lacks the flexibility to capture the variance and dynamics across different actions, yielding sub-optimal performance in practice.

To this end, we propose a *Dynamic Graph-Based Spatial Temporal Attention (DG-STA)* model for hand gesture recognition. The key idea is to perform a self-attention mechanism in both spatial and temporal domains to modify a unified graph dynamically in order to model different actions. Figure 1 gives an overview of our approach. There are three crucial designs that distinguish our approach from previous methods. **First**, instead of a pre-defined graph with fixed structure, we propose to construct a unified graph where the edges and nodes are dynamically optimized according to different actions. This makes it is possible to achieve action-specific graphs with improved expressive power. **Second**, we propose *spatial-temporal position embedding* which improves the temporal position embedding [34]. It encodes the identity and temporal order information of each node in the graph. Combining node features with their position embeddings can further improve the performance of our approach. **Third**, to implement our DG-STA more efficiently, we present a novel *spatial-temporal mask operation* which is directly applied to the matrix of scaled dot-products among all nodes. It significantly improves the computational efficiency of our model and allows easier input data arrangement.

To evaluate the effectiveness of our approach, we conduct comprehensive experiments on two standard benchmarks: DHG-14/28 Dataset [8] and SHREC'17 Track Dataset [9]. The results demonstrate that our method outperforms the state-of-the-art methods. In summary, our main contributions are summarised as follows:

- We propose Dynamic Graph-Based Spatial-Temporal Attention (DG-STA) for skeleton-based hand gesture recognition. The structures and dynamics of hand skeletons can be learned automatically and more efficiently by our approach.
- We propose spatial-temporal position embedding which encodes the identity and temporal order information of nodes to boost the performance of our model, and a spatial-temporal mask operation for efficient implementation of DG-STA.
- We conduct comprehensive experiments to validate our approach on two standard benchmarks. The proposed DG-STA achieves the state-of-the-art performance.

2 Related Work

In this section, we review recent work on self-attention and recent developments in skeleton-based action recognition, which motivated our approach.

Self-Attention. The self-attention mechanism is widely used in computer vision and natural language processing tasks [3, 4, 6, 17, 31, 33, 36, 40]. Vaswani *et al*. [34] proposed to apply the self-attention module to model temporal and semantic relationships among words within a sentence for machine translation. Instead, we study applying the self-attention mechanism to learn spatial-temporal information contained in hand skeletons represented by graphs, which are largely different from sequences. Graph Attention Networks (GATs) [35] employed self-attention to learn node embeddings of graphs. By contrast, our approach is able to capture additional temporal information as well as node identities.

Skeleton-Based Hand Gesture Recognition. Skeleton-based hand gesture is a well-studied but still challenging task. Traditional methods [8, 9, 18, 25] mainly focus on designing powerful hand feature descriptors. Smedt *et al*. [8] proposed the Shape of Connected Joints descriptor to represent the hand shape of hand skeleton. Recent studies [5, 13, 21] apply deep neural networks for this task and achieve significant performance improvement. Convolutional Neural Networks and Long Short-Term Memory are leveraged to learn the spatial and temporal features from the sequence of hand joints for hand gesture classification in [21]. One limitation of these learning-based methods is that they do not explicitly explore the structures and dynamics of human hands.

Skeleton-Based Human Action Recognition. Recent studies in skeleton-base human action recognition [27, 32, 41] started to incorporate structures and dynamics of human bodies by building skeleton graphs [29, 39, 42]. This idea is first introduced by [11] which employs Graph CNNs. Recently, Yan *et al*. [39] built a skeleton graph based on the natural structure of human body, and extract its representation by Graph Convolution Networks [15] for action recognition. Nevertheless, it is difficult to define an optimal skeleton graph which represents all action-specific structures and dynamics information. Instead, our method can automatically learn multiple action-specific graphs with the multi-head attention mechanism [34], which efficiently encode structures and dynamics of hand gestures.

3 Methodology

The overview of our approach is shown in Figure 1. First, a fully-connected skeleton graph is constructed from the input sequence of hand skeletons as described in Section 3.1. In Section 3.2, we devise DG-STA to learn the edge weights and node embeddings within

the graph. The learned node features by DG-STA are then average-pooled into a vector which captures the structures and dynamics of the input skeleton graph. We use it for hand gesture classification. Section 3.3 presents the spatial-temporal position embedding which is combined with node features to incorporate node identity and temporal order information contained in the hand skeletons. Moreover, a spatial-temporal mask operation is introduced in Section 3.4 which implements our proposed DG-STA more efficiently.

3.1 Skeleton Graph Initialization

Given a video of T frames, N hand joints are extracted from each frame to represent the hand skeleton. Then a fully-connected skeleton graph $G = (V, E)$ is constructed from this sequence of hand skeletons. Let $V = \{v_{(t,i)} | t = 1, \ldots, T, i = 1, \ldots, N\}$ denote the node set where $v_{(t,i)}$ represents the i-th hand joint at the time step t. The node features are represented by $F = \{\mathbf{f}_{(t,i)} | t = 1, \ldots, T, i = 1, \ldots, N\}$, where $\mathbf{f}_{(t,i)}$ indicates the feature vector of the node $v_{(t,i)}$. They are extracted from the 3D coordinates of nodes. Note that each node is connected with all other nodes including itself. For clarity, we define three types of edges on the edge set E as follows.

- A spatial edge $v_{(t,i)} \rightarrow v_{(t,j)} (i \neq j)$ connects two different nodes at the same time step.
- A temporal edge $v_{(t,i)} \rightarrow v_{(k,j)} (t \neq k)$ connects two nodes at different time steps.
- A self-connected edge $v_{(t,i)} \rightarrow v_{(t,i)}$ connects the node with itself.

3.2 Dynamic Graph Construction via Spatial-Temporal Attention

The proposed DG-STA consists of the spatial attention model $\mathbf{A}_S$ and temporal attention model $\mathbf{A}_T$ which are employed to extract spatial and temporal information from the hand skeleton graph respectively. Both $\mathbf{A}_S$ and $\mathbf{A}_T$ are based on multi-head attention [34]. $\mathbf{A}_S$ first takes the initial node features F as the input and updates them to encode spatial information. The updated node features are then fed to $\mathbf{A}_T$ to further learn temporal information. Finally, the results are average-pooled to a vector which is used as the feature representation of the skeleton graph for classification.

Specifically, given the input feature $\mathbf{f}_{(t,i)}$ of the node $v_{(t,i)}$ in the skeleton graph, the h-th spatial attention head first applies three fully-connected layers to map $\mathbf{f}_{(t,i)}$ into the key, query and value vectors respectively, which are formulated as:

$$\mathbf{K}^h_{(t,i)} = W^h_K \mathbf{f}_{(t,i)}, \quad \mathbf{Q}^h_{(t,i)} = W^h_Q \mathbf{f}_{(t,i)}, \quad \mathbf{V}^h_{(t,i)} = W^h_V \mathbf{f}_{(t,i)}, \tag{1}$$

where $\mathbf{K}^h_{(t,i)}$, $\mathbf{Q}^h_{(t,i)}$ and $\mathbf{V}^h_{(t,i)}$ represent the key, query and value vectors of the node; $\mathbf{W}^h_K$, $\mathbf{W}^h_Q$ and $\mathbf{W}^h_V$ are the corresponding weight matrices of the three fully-connected layers of the h-th spatial attention head.

The spatial attention head computes the weights of the spatial and self-connected edges in two steps. First, it calculates the "scaled dot-product" [34] between the query vectors and key vectors of the nodes within the same time step. Then it normalizes the results by a Softmax function. These two steps are formulated as:

$$u^h_{(t,i)\rightarrow(t,j)} = \frac{\langle \mathbf{Q}^h_{(t,i)}, \mathbf{K}^h_{(t,j)} \rangle}{\sqrt{d}}, \quad \alpha^h_{(t,i)\rightarrow(t,j)} = \frac{\exp\left(u^h_{(t,i)\rightarrow(t,j)}\right)}{\sum_{n=1}^{N} \exp\left(u^h_{(t,i)\rightarrow(t,n)}\right)}, \tag{2}$$

where d is the dimension of the key, query and value vectors; $u^h_{(t,i)\to(t,j)}$ is the scaled dot-product of the node $v_{(t,i)}$ and $v_{(t,j)}$; $\langle\cdot,\cdot\rangle$ represents the inner product operation; $\alpha^h_{(t,i)\to(t,j)}$ is the attention weight between the node $v_{(t,i)}$ and $v_{(t,j)}$, which measures the importance of information from node $v_{(t,j)}$ to node $v_{(t,i)}$. Meanwhile, the weights for all temporal edges are set to 0 in order to block the information passing in the temporal domain. As a result, each spatial attention head produces a weighted skeleton graph which represents a specific type of spatial structure of the hand.

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:

$$\bar{\mathbf{f}}^h_{(t,i)} = \sum_{j=1}^{N} \left(\alpha^h_{(t,i)\to(t,j)} \cdot \mathbf{V}^h_{(t,j)} \right), \tag{3}$$

where $\bar{\mathbf{f}}^h_{(t,i)}$ denotes the spatial attention feature of the node $v_{(t,i)}$. Intuitively, the computation mechanism of the spatial attention features is essentially the process that each node in the graph sends some information to the others within the same time step and then aggregates the received information based on the learned edge weights.

The spatial attention model $\mathbf{A}_S$ finally concatenates the spatial attention features learned by all spatial attention heads into $\tilde{\mathbf{f}}_{(t,i)}$ which is employed as the spatial feature of the node $v_{(t,i)}$:

$$\tilde{\mathbf{f}}_{(t,i)} = \text{Concate}\left[\bar{\mathbf{f}}^1_{(t,i)}, \bar{\mathbf{f}}^2_{(t,i)}, ..., \bar{\mathbf{f}}^H_{(t,i)} \right], \tag{4}$$

where H is the number of spatial attention heads. The obtained node features encode multiple types of structural information represented by the weighted skeleton graphs which are learned by different spatial attention heads.

The temporal attention model $\mathbf{A}_T$ takes the output node features from the spatial attention model as the input, and then applies the above multi-head attention mechanism in the temporal domain. The node feature which is the output from the temporal attention model encodes both spatial and temporal information carried by the input sequence of the hand skeletons. We average-pool these node features to a vector as the feature representation of the input sequence for hand gesture recognition.

3.3 Spatial-Temporal Position Embedding

The original node features F extracted from the coordinates of the input hand skeletons do not contain spatial identity information describing which hand joint a node corresponds to, and temporal information indicating which time step a node is at. To incorporate these messages, we propose the spatial-temporal position embedding.

Specifically, our spatial-temporal position embedding is made up of the spatial position embedding and the temporal position embedding. The spatial one consists of N vectors and each represents a hand joint. Meanwhile, the temporal one is composed of $N \times T$ distinct vectors and each of them corresponds to a node in the hand skeleton graph. The feature vector of a specific node is added with the corresponding spatial and temporal position embedding vectors before fed into the DG-STA. Therefore, we have:

$$\hat{\mathbf{f}}_{(t,i)} = \mathbf{A}_T \left(\mathbf{p}^T_{(t,i)} + \mathbf{A}_S \left(\mathbf{f}_{(t,i)} + \mathbf{p}^S_{(i)} \right) \right), \tag{5}$$

where $\hat{\mathbf{f}}_{(t,i)}$ denotes the final output feature of node $v_{(t,i)}$, $\mathbf{p}^S_{(i)}$ is the spatial position embedding of the i-th hand joint, and $\mathbf{p}^T_{(t,i)}$ denotes the temporal position embedding of the i-th hand joint at t-th time step. These embeddings are with the same dimension as $\mathbf{f}_{(t,i)}$, and their values are set using the sine and cosine functions of different frequencies following [34].

3.4 Efficient Implementation

It is not straightforward to implement the proposed DG-STA because the input data have to be arranged in a complex format. However, we find that the computation of attention weights and features without domain constraints is straightforward, which can be implemented efficiently using matrix multiplication operations. Therefore, we propose a novel scheme to facilitate the implementation of the DG-STA. The main idea is to first compute the matrix of the scaled dot-products among all nodes and then apply the proposed spatial-temporal mask operation to the matrix in order to let the model focus on the spatial or temporal domain.

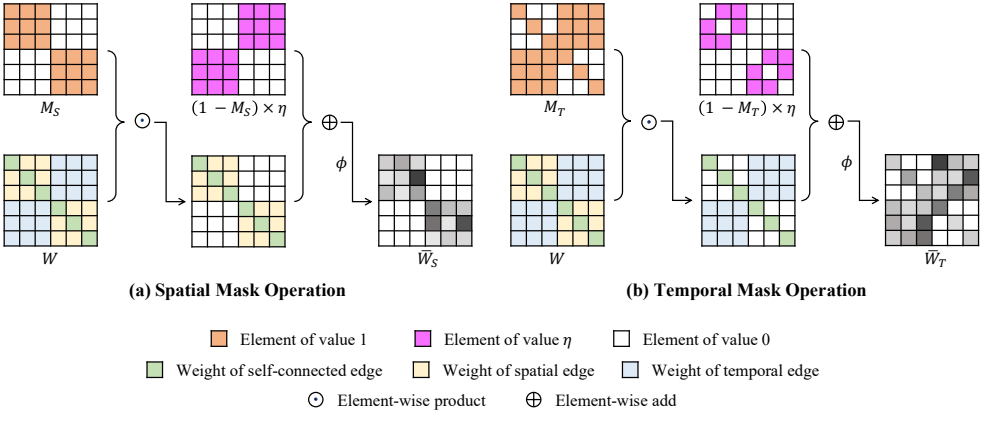

Figure 2: Illustration of the proposed spatial and temporal mask operations.

An illustration of our mask operation is shown in Figure 2. For a specific attention head, we compute a query matrix $\mathbf{Q}$ where each row represents the query vector of each node, and a key matrix $\mathbf{K}$ where each row corresponds to the key vector of each node. The matrix of the scaled dot-products $\mathbf{W}$ (*i.e.*, the edge weights before normalization) can be obtained by:

$$\mathbf{W} = \mathbf{Q} \otimes \mathbf{K}^\top, \tag{6}$$

where $\otimes$ is the matrix multiplication, and $\top$ denotes the matrix transpose operation. Then the proposed spatial mask operation sets the value of each element in $\mathbf{W}$ which represents the temporal edge to η (*i.e.*, a number close to negative infinity) and keeps the values of other elements unchanged. Therefore, the resulting matrix after the spatial mask operation $\bar{\mathbf{W}}_S$ is calculated:

$$\bar{\mathbf{W}}_S = \phi\left(\mathbf{W} \odot \mathbf{M}_S + (1 - \mathbf{M}_S) \times \eta\right), \tag{7}$$

where $\odot$ denotes the element-wise dot operation, $\mathbf{M}_S$ is the spatial mask where the elements are 1 if they represent the spatial or self-connect edges and 0 otherwise. We set η to -9×10^{15} in our implementation. The Softmax activation ϕ essentially normalizes weights across spatial edges, because the exponential value of the η is close to 0. As a result, all

weights of the temporal edges in $\bar{\mathbf{W}}_S$ are set to 0. Equations (6) and (7) efficiently implement the calculation of edge weights in the spatial domain formulated in Equation (2). Moreover, the matrix $\bar{\mathbf{W}}_S$ can be directly employed to implement the computation of node features represented by Equation (3) by performing the matrix multiplication operation with the matrix of value vectors.

We define the temporal mask operation following the same way as Equation (7). The difference is that we use the temporal mask $\mathbf{M}_T$ instead of $\mathbf{M}_S$ to compute the matrix after the temporal mask operation $\bar{\mathbf{W}}_T$. The elements of $\mathbf{M}_T$ are 1 if they represent the temporal or self-connect edges and 0 otherwise. With the help of the proposed spatial-temporal mask operation, we experimentally find that the computation time is reduced by 99%.

4 Experiments

In this section, we first describe our network structure in Section 4.1. In Section 4.2, we introduce the datasets and settings employed in the experiments. Then we conduct ablation studies in Section 4.3 to evaluate the effectiveness of each component proposed in our method. Finally, we report our results and comparisons with the state of the art in Section 4.4.

4.1 Implementation Details

Our network structure is shown in Figure 3. We set the head number of the spatial and temporal attention models to 8. The dimension d of the query, key and value vector is set to 32. Layer Normalization [16] is utilized to normalize the intermediate outputs of our network. The input 3D coordinate of a hand joint is projected into an initial node feature of 128 dimension. It is then added with the corresponding spatial position embedding and fed into the spatial attention model, which produces a node feature of 256 dimension. This node feature is projected into a vector of 128 dimension which is added with the corresponding temporal position embedding. The temporal attention model takes it as the input and generates the final node feature. Finally, we average-pool the features of all nodes into a vector and feed it into a fully-connected layer for classification.

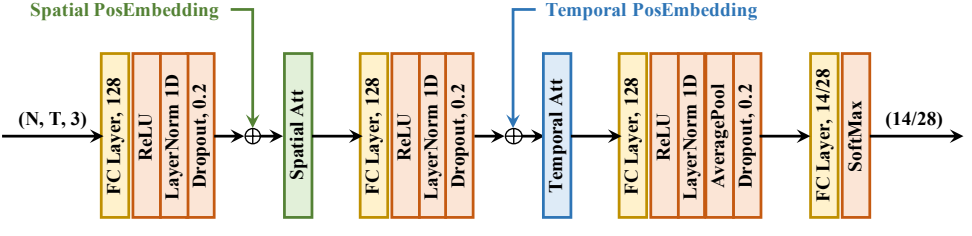

Figure 3: The network architecture of the proposed DG-STA.

4.2 Datasets and Settings

We evaluate our method on the DHG-14/28 Dataset [8] and the SHREC'17 Track Dataset [9]. Both datasets contain 2800 video sequences of 14 hand gestures which are performed in two configurations: using one single finger or the whole hand. The videos of the two datasets are

captured by the Intel Realsense camera. The 3D coordinates of 22 hand joints in real-world space are provided per frame for network training and evaluation.

Network Training. The proposed DG-STA is implemented based on the PyTorch platform. The Adam [14] optimizer with a learning rate of 0.001 is employed to train our model. The batch size is set to 32 and the dropout rate [30] is set to 0.2. We uniformly sample 8 frames from each video as the input. For fair comparison, we perform data augmentation by applying the same operations as proposed in [9, 21] including scaling, shifting, time interpolation and adding noise. We also subtract every skeleton sequence by the palm position of the first frame following Smedt *et al*. [9] for alignment.

Evaluation Protocols. On the DHG-14/28 Dataset, models are evaluated by using the *leave-one-subject-out cross-validation* strategy [8]. Specifically, we perform one experiment for each subject in this dataset. In each experiment, one subject is selected for testing and the remaining 19 subjects are used for training. The average accuracy of 14 gestures (without the single-finger configuration) or 28 gestures (with the single-finger configuration) over the 20 cross-validation folds are reported. For the SHREC'17 Track Dataset, we use the same data split as provided by [9] and report the accuracy of both 14 and 28 gestures.

4.3 Ablation Study

Our proposed approach consists of three major components, including the fully-connected skeleton graph structure (FSG), the spatial-temporal attention model (STA) and the spatial-temporal position embedding (STE). We validate the effectiveness of these components in this section. The results are shown in Table 1.

Setting	FSG+STA	FSG+GAT+STE	SSG+STA+STE	**DG-STA**
14 Gestures (D)	84.3	90.8	89.8	**91.9**
28 Gestures (D)	77.3	87.8	86.6	**88.0**
14 Gestures (S)	88.9	92.7	91.5	**94.4**
28 Gestures (S)	80.1	86.2	87.7	**90.7**

Table 1: Ablation study of accuracy (%) on the DHG-14/28 Dataset (D) and SHREC'17 Track Dataset (S). Our full model (DG-STA) achieves the best performance.

Evaluation of Fully-Connected Graph Structure. We compare the proposed FSG with the sparse skeleton graph structure (SSG) introduced by Yan *et al*. [39], where spatial edges are defined based on the natural connections of the hand joints and temporal edges connect the same joints between consecutive frames. We can see that our model significantly outperforms the one trained on SSG. This is because SSG may be sub-optimal for some hand gestures, while FSG has little constrains on the model so that it is able to learn action-specific graph structures.

Evaluation of Spatial-Temporal Attention. The proposed STA downgrades to Graph Attention (GAT) [35] if only one attention model is applied to the whole graph without distinguishing the spatial and temporal domains. We implement GAT by replacing the spatial and temporal attention models in our network with one attention model, and train it under the same setting of our model. We can observe that the STA-based model achieves better performance than the GAT-based model, which demonstrates the effectiveness of STA.

Evaluation of Spatial-Temporal Position Embedding. We validate the effectiveness of the proposed STE by training a variant of our method where STE is removed. We can see

that our model outperforms the model without STE, which demonstrates the importance of the identity and temporal order information encoded by STE.

4.4 Comparison with Previous Methods

We compare our method with the state-of-the-art methods on the DHG-14/28 Dataset [8] and the SHREC'17 Track Dataset [9], respectively. The compared state-of-the-art methods include traditional hand-crafted feature approaches [1, 2, 5, 7, 8, 10, 25, 26], deep learning based approaches [9, 13, 21] and a graph-based method [39]. The results are shown in Tables 2 and 3. Note that for ST-GCN [39], we implement it following the *distance partitioning* setting and use a three-layer ST-GCN with 128 channels for fair comparison. We collect the results of other baseline methods from [13].

Method	14 Gestures	28 Gestures
SoCJ+HoHD+HoWR [8]	83.1	80.0
Chen *et al*. [5]	84.7	80.3
CNN+LSTM [21]	85.6	81.1
Res-TCN [13]	86.9	83.6
STA-Res-TCN [13]	89.2	85.0
ST-GCN [39]	91.2	87.1
DG-STA (Ours)	**91.9**	**88.0**

Table 2: Comparisons of accuracy (%) on DHG-14/28 Dataset.

Results on DHG-14/28 Dataset. From Table 2, we can see that our method achieves the state-of-the-arts performance under both 14-gesture and 28-gesture setting. Moreover, both our method and ST-GCN [39] outperform other methods which do not explicitly exploit structures and dynamics of hands, which demonstrates that these messages are important for skeleton-based hand gesture recognition.

Method	14 Gestures	28 Gestures
Oreifej *et al*. [26]	78.5	74.0
Devanne *et al*. [10]	79.4	62.0
Classify Sequence by Key Frames [9]	82.9	71.9
Ohn-Bar *et al*. [25]	83.9	76.5
SoCJ+Direction+Rotation [7]	86.9	84.2
SoCJ+HoHD+HoWR [8]	88.2	81.9
Caputo *et al*. [2]	89.5	-
Boulahia *et al*. [1]	90.5	80.5
Res-TCN [13]	91.1	87.3
STA-Res-TCN [13]	93.6	**90.7**
ST-GCN [39]	92.7	87.7
DG-STA (Ours)	**94.4**	**90.7**

Table 3: Comparisons of accuracy (%) on SHREC'17 Track Dataset.

Results on SHREC'17 Track Dataset. Different from the DHG-14/28 Dataset where videos are cropped by human-labeled beginnings and ends of the gestures [8], the SHREC'17

Track Dataset provides raw captured video sequences with noisy frames, and hence is more challenging. We can see that our method achieves the state-of-the-arts performance under the 14-gesture setting, and obtains comparable performance with STA-Res-TCN [13] under the 28-gesture setting. In addition, we can observe that our method and ST-GCN [39] outperform all other methods which do not explicitly exploit structures and dynamics of hands.

5 Conclusions

In this paper, we proposed a graph-based spatial-temporal attention method for skeleton-based hand-gesture recognition. It utilizes two attention models in the spatial and temporal domains of the fully-connected hand skeleton graph to learn edge weights and extract spatial and temporal information for hand gesture recognition. Extensive experiments demonstrate the effectiveness of our framework. Our proposed method provides a general framework that can be further used for other tasks aiming to learn spatial and temporal information from graph-based data, *e.g.*, skeleton-based human action recognition.

6 Acknowledgments

This work was funded partly by ARO-MURI-68985NSMUR and NSF 1763523, 1747778, 1733843, 1703883 to Dimitris N. Metaxas.

References

[1] Said Yacine Boulahia, Eric Anquetil, Franck Multon, and Richard Kulpa. Dynamic hand gesture recognition based on 3D pattern assembled trajectories. In *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2017.

[2] Fabio M Caputo, Pietro Prebianca, Alessandro Carcangiu, Lucio D Spano, and Andrea Giachetti. Comparing 3D trajectories for simple mid-air gesture recognition. *Computers & Graphics*, 73:17–25, 2018.

[3] Tianlang Chen, Yuxiao Chen, Han Guo, and Jiebo Luo. When e-commerce meets social media: Identifying business on wechat moment using bilateral-attention lstm. In *Proceedings of the World Wide Web Conference (WWW)*, pages 343–350, 2018.

[4] Tianlang Chen, Zhongping Zhang, Quanzeng You, Chen Fang, Zhaowen Wang, Hailin Jin, and Jiebo Luo. "factual"or"emotional": Stylized image captioning with adaptive learning and attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 519–535, 2018.

[5] Xinghao Chen, Hengkai Guo, Guijin Wang, and Li Zhang. Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 2881–2885, 2017.

[6] Yuxiao Chen, Jianbo Yuan, Quanzeng You, and Jiebo Luo. Twitter sentiment analysis via bi-sense emoji embedding and attention-based lstm. In *Proceedings of the ACM Multimedia Conference on Multimedia Conference (MM)*, pages 117–125, 2018.

[7] Quentin De Smedt. *Dynamic hand gesture recognition-From traditional handcrafted to recent deep learning approaches*. PhD thesis, Université de Lille 1, Sciences et Technologies; CRIStAL UMR 9189, 2017.

[8] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–9, 2016.

[9] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. SHREC'17 Track: 3D hand gesture recognition using a depth and skeletal dataset. In *Eurographics Workshop on 3D Object Retrieval*, 2017.

[10] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 3-D human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE Transactions on Cybernetics*, 45(7):1340–1352, 2015.

[11] M. Edwards and X. Xie. Graph-based CNN for human action recognition from 3D pose. In *British Machine Vision Conference Workshop: Deep Learning on Irregular Domains*, pages 1.1–1.10, 2017.

[12] William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, volume 12, pages 296–301, 1995.

[13] Jingxuan Hou, Guijin Wang, Xinghao Chen, Jing-Hao Xue, Rui Zhu, and Huazhong Yang. Spatial-temporal attention Res-TCN for skeleton-based dynamic hand gesture recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 273–286, 2018.

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

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[17] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[18] Shan Lu, Dimitris Metaxas, Dimitris Samaras, and John Oliensis. Using multiple cues for hand tracking and model refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 443–450, 2003.

[19] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3D convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–7, 2015.

[20] Ashish S Nikam and Aarti G Ambekar. Sign language recognition using image based hand gesture recognition techniques. In *Proceedings of the International Conference on Green Engineering and Technologies (IC-GET)*, pages 1–5, 2016.

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

[22] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3D hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 585–594, 2017.

[23] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[24] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3316–3324, 2015.

[25] Eshed Ohn-Bar and Mohan Trivedi. Joint angles similarities and HOG2 for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 465–470, 2013.

[26] Omar Oreifej and Zicheng Liu. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 716–723, 2013.

[27] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2226–2234, 2018.

[28] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2015.

[29] Chenyang Si, Ya Jing, Wei Wang, Liang Wang, and Tieniu Tan. Skeleton-based action recognition with spatial reasoning and temporal stack learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[31] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. Deep semantic role labeling with self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[32] Zhiqiang Tang, Xi Peng, Shijie Geng, Lingfei Wu, Shaoting Zhang, and Dimitris Metaxas. Quantized densely connected U-Nets for efficient landmark localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 339–354, 2018.

[33] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N Metaxas. CR-GAN: Learning complete representations for multi-view generation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 942–948, 2018.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, 2017.

[35] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[36] Patrick Verga, Emma Strubell, and Andrew McCallum. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. *arXiv preprint arXiv:1802.10569*, 2018.

[37] Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Yael Edan. Vision-based hand-gesture applications. *Communications of the ACM*, 54(2):60–71, 2011.

[38] Chong Wang, Zhong Liu, and Shing-Chow Chan. Superpixel-based hand gesture recognition with kinect depth camera. *IEEE Transactions on Multimedia*, 17(1):29–39, 2015.

[39] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[40] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 7354–7363, 2019.

[41] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 387–403, 2018.

[42] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3D human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3425–3435, 2019.