STGAE: Spatial-Temporal Graph Auto-Encoder for Hand Motion Denoising

Kanglei Zhou[1] Zhiyuan Cheng[1] Hubert P. H. Shum[2] Frederick W. B. Li[2] Xiaohui Liang[1,*]

[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China
[2] Department of Computer Science, Durham University, Durham, UK

ABSTRACT

Hand object interaction in mixed reality (MR) relies on the accurate tracking and estimation of human hands, which provide users with a sense of immersion. However, raw captured hand motion data always contains errors such as joints occlusion, dislocation, high-frequency noise, and involuntary jitter. Denoising and obtaining the hand motion data consistent with the user's intention are of the utmost importance to enhance the interactive experience in MR. To this end, we propose an end-to-end method for hand motion denoising using the spatial-temporal graph auto-encoder (STGAE). The spatial and temporal patterns are recognized simultaneously by constructing the consecutive hand joint sequence as a spatial-temporal graph. Considering the complexity of the articulated hand structure, a simple yet effective partition strategy is proposed to model the physic-connected and symmetry-connected relationships. Graph convolution is applied to extract structural constraints of the hand, and a self-attention mechanism is to adjust the graph topology dynamically. Combining graph convolution and temporal convolution, a fundamental graph encoder or decoder block is proposed. We finally establish the hourglass residual auto-encoder to learn a manifold projection operation and a corresponding inverse projection through stacking these blocks. In this work, the proposed framework has been successfully used in hand motion data denoising with preserving structural constraints between joints. Extensive quantitative and qualitative experiments show that the proposed method has achieved better performance than the state-of-the-art approaches.

Index Terms: Motion data cleanup—Hand motion denoising—Graph convolutional network—Spatial-temporal graph auto-encoder

1 INTRODUCTION

With the rapid development of human-computer interaction techniques in mixed reality (MR), more and more interactive methods based on different algorithms are provided to enable users to obtain feelings of immersion in 3D virtual scenes [33]. During the interaction, human hands play a key role to perform operations such as grasp, move, and rotate. Accurate hand pose estimation and tracking [16] can effectively improve the user experience. However, the complex articulations, self-occlusion, and self-similarity of hands make this task challenging [35], and prolonged interactions may cause fatigue and involuntary handshaking. Moreover, hand-object interaction in MR is extremely unfriendly for patients with motion disorders such as Parkinson's disease. As a result, hand motion capture data with natural noise, dislocation, and jitters may not conform to the intention of users. It is of the great interest to develop effective hand motion denoising methods for addressing these problems.

Cleaning motion capture data is a key process. Existing methods for motion data denoising mainly include prior knowledge based methods and machine learning based methods. The former is done by modelling human prior knowledge on the feature and the problem by different algorithms such as skeleton hierarchy and filter designs. The latter is done by machine learning frameworks, such as deep learning, etc. Compared with the latter, the former is always dependent on some priors and restricted by personal cognitive level. Although some machine learning based methods have achieved considerable results, the capability of these methods is limited due to the inexplicit spatial relationships among joints. To extract knowledge directly from the data, the proposed method allows us to introduce some human prior knowledge (i.e. the design of the graph) into a machine learning system, such that we get the advantages of both.

Constructing the hand as a graph is effective to learn the spatial constraints between joints. Recently, a hand motion compensation method [13] attempts to leverage the physic-connected connections between joints using the graph convolutional neural network. This method shows the encouraging performance, which indicates the significance of spatial connections between joints. However, this method has two separated stages for spatial domain and temporal domain, making it ineffective to extract correlated spatial-temporal patterns. In addition, roughly applying tremor to the hand as a whole ignores the relative motion between joints.

To effectively model the spatial-temporal patterns for hand motion denoising, in this paper, we propose an end-to-end approach using the spatial-temporal graph auto-encoder (STGAE). Following the existing practice [8, 22], we generate noisy variants of motions by adding noise to each joint independently. Considering the physical laws of hand motion, we optimize them to preserve structural constraints such as bone length. Then, the hand motion data is constructed as a spatial-temporal graph, where a simple yet effective partition strategy is proposed to model the structural constraints of hand. The introduced self-attention mechanism enables the graph topology to adjust dynamically. Combining graph convolution and temporal convolution, a basic spatial-temporal encoder or decoder block is designed. The proposed method adopts an hourglass architecture by stacking these basic blocks. The encoder is responsible for learning a manifold projection and the decoder is for the corresponding inverse projection. The whole network architecture adopts a global residual structure, which can ensure the stabilization of training. The proposed method can keep the structural constriants between the joints well that the previous works [8, 13] suffer from.

The proposed method outperforms the state-of-the-art works in several metrics, such as hand pose similarity error and bone length error. Ablation studies show the power of the introduced self-attention mechanism the proposed novel partition strategy. Extensive quantitative and qualitative results verify the strength of our approach.

The main contributions of this work are summarized as follows:

- A powerful framework for hand motion data denoising is proposed using the spatial temporal graph auto-encoder, which can effectively extract spatial-temporal patterns.
- During the data corruption, the structural constraints between joints are considered to simulate natural abnormal hand motion and preserve the bone length simultaneously.
- A novel hand skeleton partition strategy is presented and a dynamic self-attention mechanism is introduced, ensuring that the structural constraints of the hand can be well maintained.

*e-mail: liang_xiaohui@buaa.edu.cn (corresponding author)

2 Related Works

We first briefly review the previous works aiming at solving motion data cleanup. It is followed by the previous works that use deep learning methods to deal with noisy data and graph-structured data.

2.1 Motion Data Cleanup

To tackle the motion data cleaning problem, one key idea is to introduce prior belief about the behavior of skeletal joints [8]. In the process of motion, the joints should follow the law of physics in the time domain and the physiological structure constraints in the spatial domain. Therefore, two significant kinds of priors can be used, i.e., temporal priors [1, 6, 18] and spatial priors [15].

These prior knowledge based methods [1, 15, 20] mainly include skeleton-based methods, Kalman filter based methods and dimension reduction methods. Burke *et al.* [1] have proposed a marker position estimation algorithm combining temporal smoothing with a Kalman filter and low rank matrix completion. This method is effective for gap filling, while the universal noise scenario is not considered. Lou *et al.* [20] construct a series of filter bases from the clean motion data and determine the filter weights through a non-linear optimization method. Different types of motion require learning separate bases, which is computationally expensive. Therefore, it is not suitable for a large variety of input motions. Li *et al.* [15] have utilized a prior belief about the distances between markers and the joint length constraint to present a principled technique called BoLeRO. Though it is capable of filling gaps when motion capture is occluded, the spatial noise such as marker swaps scenario is ignored.

Since the motion data usually contains spatial noise and temporal noise, only considering any one has limitations. This motivates us to exploit the spatial-temporal patterns for hand motion data cleanup.

2.2 Denoising Neural Network

The deep learning method is highly favored because it overcomes several drawbacks such as manually setting parameters [28]. Denoising auto-encoder is a universally used architecture, which has evolved in many versions [7, 21]. The classical auto-encoder comprises encoder and decoder. The encoder attempts to obtain the robust latent representations by corrupting the clean data, and the decoder reconstructs the original data. Another popular denoising framework is the generative adversarial network [3, 34], generating the denoised output and then inputting it into the discriminator to train the denoiser. These architectures are not robust to the human motion data since they do not consider the temporal relations between adjacent frames and spatial relationships among joints.

Many deep learning methods [11, 32, 36] also focus on human motion data denoising and have achieved state-of-the-art results. In particular, Holden *et al.* [9] try to learn motion manifolds with the convolutional auto-encoder. The corrupted motion data can be reconstructed to clean output by defining a manifold projection and the corresponding inverse one. Wang *et al.* [30] have proposed a spatial-temporal manifold that is capable of denoising motion data with corrupted stepping patterns. Although these two methods can deal with the noise in the spatial-temporal domain, they ignore the structural constraints between the skeletal joints. To consider more denoising scenarios, Holden [8] has trained a deep denoising feed-forward neural network with a residual structure, learning transform mapping from clean data to corrupted data. However, both the temporal constraint of consecutive frames and the spatial constraint of different skeletal joints are not considered.

Kim *et al.* [11] have proposed a novel method of denoising human motion, using a bidirectional recurrent neural network with an attention mechanism. The attention mechanism ensures that a higher weight value is selectively given to the more important input at a specific frame, thus achieving better optimization results than others. Though using a recurrent neural network or long short-term memory architecture can effectively mine the temporal patterns, it is really hard to capture the structural constraints between joints due to the limitations of the general convolution operation. To this end, a WaveNet followed by a graph neural network [13] is proposed to stably estimate the hand pose under tremor. This two-stage neural network is only tackling the hand tremor and ignores the relative motion among joints. Further, the two separated operations are not conductive to extract spatial-temporal patterns.

Considering the structural constraints between skeletal joints, we aim to develop an end-to-end neural network to dig out spatial-temporal patterns rather than restricting it to any abnormal motion.

2.3 Graph Convolutional Neural Network

Different from the conventional convolution neural networks on non-structured data, graph convolutional neural network (GCN) [12] aims at generalizing the convolution operation to the versatile graph-structured data [31]. There are two technique routes to implement GCN, i.e., spectral-based approaches [5, 14, 25] and spatial-based approaches [4, 19, 27]. The former defines graph convolution by introducing filters according to graph spectral analysis, while the latter is based the on information propagation among graph nodes.

GCN has been developed into many variants in the past few years, including graph auto-encoders [10, 24] and spatial-temporal GCNs [2, 17, 32]. They have been widely extended to skeleton-based classification, time-series forecasting, and so on. In particular, Yan *et al.* [32] have designed a spatial-temporal convolution neural network for skeleton-based action recognition, which can automatically learn both the spatial and temporal patterns simultaneously. To improve its recognition accuracy, Shi *et al.* [26] have introduced an adaptive graph convolution block to learn the graph topology. This work combines the essence of the above two GCN variants to construct a novel network architecture for human skeletal motion data cleaning, aiming at learning both spatial and temporal patterns simultaneously.

Comparing with existing works, we aim at devising an end-to-end method to fix the corrupted data using STGAE, without restricting it to the particular noise distribution or skeleton model. The proposed method can exploit the temporal-spatial patterns, ensuring maximum consideration of structural constraints for the faithful performance.

3 Methodology

In this section, we first briefly introduce the overview of the framework of the proposed method. Then, the detailed data corruption algorithm is presented. Finally, the proposed method is elaborated.

3.1 Framework Overview

As shown in Fig. 1, a pipeline for hand motion data denoising is presented using STGAE. This section briefly describes the proposed processing framework. Given a time-series dataset of hand poses $\mathcal{D} = \{\mathbf{X}_i : \mathbf{X}_i \in \mathbb{R}^{N\times 3}, i = 1, 2, \cdots, T\}$ where T is the number of frames and N is the number of hand joints, let $\mathbf{X} \in \mathbb{R}^{T\times N\times 3}$ be the hand pose matrix and suppose that it is clean.

In Fig. 1, we first corrupt the clean hand motion data $\mathbf{X}$ to obtain the training pairs (the corrupted data $\tilde{\mathbf{X}} \in \mathbb{R}^{T\times N\times 3}$ as input and the clean data $\mathbf{X}$ as output) using a novel corruption algorithm. Then, construct the hand motion data as the spatial-temporal graph $\mathcal{G}$. Input the corrupted data $\tilde{\mathbf{X}}$ into the proposed network and output the reconstructed motion data $\mathbf{Y}$. In the end, the denoising results are evaluated through three different metrics. The trained model can accurately estimate the user's intention in MR. In practice, we only need to feed the data into the network to get the faithful motion.

3.2 Corrupted Motion Data Synthesis

In this section, we simulate the abnormal hand motion by corrupting the hand joints independently to produce the corrupted data $\tilde{\mathbf{X}}$.

Inspired by Holden *et al.* [8], we use algorithm 1 to corrupt the clean hand joint data $\mathbf{X}$. Before training, we first scale all the hand

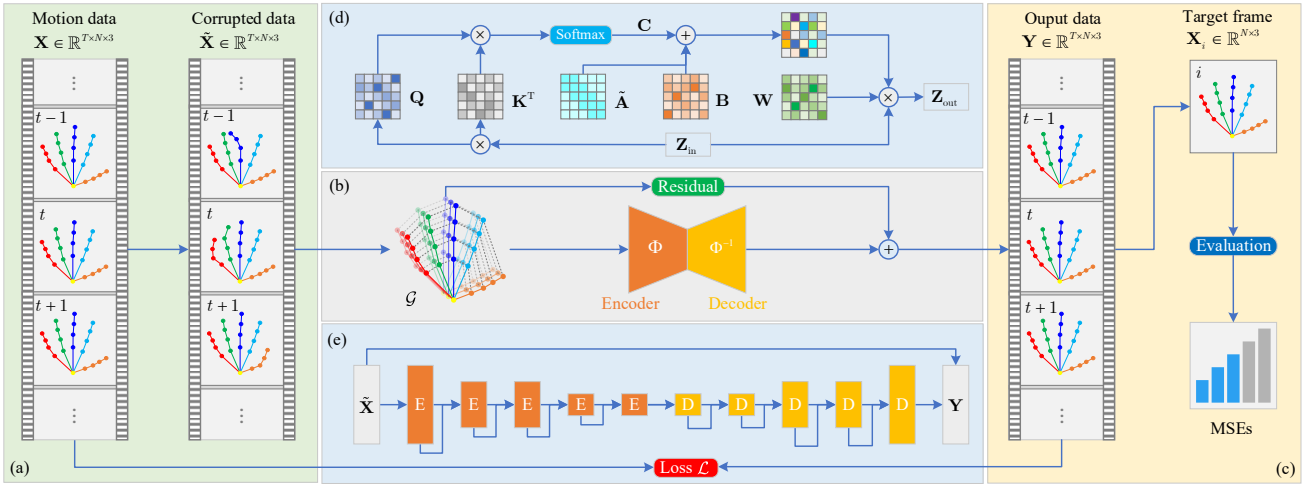


Figure 1: The pipeline of the proposed method for hand motion denoising using STGAE: (a) the data-preprocessing phase introducing data corruption where the clean hand motion data $\mathbf{X} \in \mathbb{R}^{T\times N\times 3}$ is broken into the corrupted data $\tilde{\mathbf{X}} \in \mathbb{R}^{T\times N\times 3}$; (b) the graph construction by modeling the spatial dependencies between joints and the temporal continuity between consecutive frames as well as the proposed STGAE architecture consisting of a projection operator Φ and a corresponding inverse one Φ^{-1}; (c) the evaluation phase where the reconstructed data $\mathbf{Y} \in \mathbb{R}^{T\times N\times 3}$ is evaluated by different metrics; (d) the introduced graph convolution module with the self-attention mechanism in Sect. 3.3.3. (e) the network architecture in Sect. 3.3.4. Note that only inputting the real motion data into our network is required in the inference phase.

joints data so that the hand has a uniform length. This form of normalization $\mathrm{Normalize}(\cdot)$ ensures that we do not have to explicitly deal with hands of different sizes in the proposed framework but only hands of different proportions. We then corrupt the motion data in three steps, considering the involuntary tremor caused by a long period of hand-object interaction and the presence of rhythmic hand tremor in patients with Parkinson's disease.

Due to the error caused by hand pose estimation techniques, the coordinates of some joints deviate. To simulate this position shifts, the shifting noise $\mathbf{S} \in \mathbb{R}^{T\times N\times 3}$ is added to the coordinates of some joints. It is satisfying a certain distribution $\mathrm{Shift}(\cdot)$ controlled by the hyper-parameter β. In this work, we use different levels of corruptions and that includes zero corruptions, which ensures that our method can learn both the damage function and the non-damage function. Similarly, the shifting mask $\mathbf{M}_{\mathrm{s}} \in \mathbb{R}^{T\times N}$ is generated by the hyper-parameter σ_{s} controlling which coordinates are offset.

Due to the complex structure and self-occlusion of the hand, it is unavoidable that there are missing joints through hand pose estimation methods. In this phase, we can sample a norm distribution $\mathcal{N}(0, \sigma_{\mathrm{o}}^2)$, and then generate the occlusion mask $\mathbf{M}_{\mathrm{o}} \in \mathbb{R}^{T\times N}$ by sampling a Bernoulli distribution. $\mathbf{M}_{\mathrm{o}}$ controls which hand joints coordinates are set to zero to simulate joints missing.

To confirm the laws of motion, all the corrupted operation should be subject to the structural constraint. In the implementation, we optimize $\mathbf{X}_{\mathrm{c}} \in \mathbb{R}^{T\times N\times 3}$ to preserve structural constraints such as bone length and joint angle. Note that a reasonable assumption can be made that the trajectory of the root joint is clean.

Finally, the corrupted data $\tilde{\mathbf{X}} \in \mathbb{R}^{T\times N\times 3}$ with respect to its ground truth $\mathbf{X}$ is obtained for the next training phase.

Algorithm 1: Hand joint corruption algorithm

Input: Hand joint data $\mathbf{X} \in \mathbb{R}^{T\times N\times 3}$, hyper-parameters $\sigma_{\mathrm{o}} \in \mathbb{R}$, $\sigma_{\mathrm{s}} \in \mathbb{R}$ and $\beta \in \mathbb{R}$.
Output: Corrupted hand joint data $\tilde{\mathbf{X}} \in \mathbb{R}^{T\times N\times 3}$.

$\mathbf{X}_{\mathrm{n}}^{T\times N\times 3} \leftarrow \mathrm{Normalize}(\tilde{\mathbf{X}})$; // Data normalization
$\boldsymbol{\alpha}_{\mathrm{s}}^{T} \leftarrow \mathcal{N}(0, \sigma_{\mathrm{s}}^2)$; // Sample shifting probability
$\mathbf{M}_{\mathrm{s}}^{T\times N} \leftarrow \mathrm{Bernoulli}(\min(|\boldsymbol{\alpha}_{\mathrm{s}}|, 2\sigma_{\mathrm{s}}))$; // Mask
$\mathbf{S}^{T\times N\times 3} \leftarrow \mathrm{Shift}(-\beta, \beta)$; // Shifting distribution
$\mathbf{X}_{\mathrm{s}}^{T\times N\times 3} \leftarrow \mathbf{X}_{\mathrm{n}} + \mathbf{S} \odot \mathbf{M}_{\mathrm{s}}$; // Data shifting
$\boldsymbol{\alpha}_{\mathrm{o}}^{T} \leftarrow \mathcal{N}(0, \sigma_{\mathrm{o}}^2)$; // Sample occlusion probability
$\mathbf{M}_{\mathrm{o}}^{T\times N} \leftarrow \mathrm{Bernoulli}(\min(|\boldsymbol{\alpha}_{\mathrm{o}}|, 2\sigma_{\mathrm{o}}))$; // Mask
$\mathbf{X}_{\mathrm{c}}^{T\times N\times 3} \leftarrow \mathbf{X}_{\mathrm{s}} \odot (1 - \mathbf{M}_{\mathrm{o}})$; // Data occlusion
$\tilde{\mathbf{X}}^{T\times N\times 3} \leftarrow \mathrm{Optimize}(\mathbf{X}_{\mathrm{c}})$; // Structural constraint

3.3 Spatial-Temporal Graph Auto-Encoder

The general idea of STGAE is to learn a manifold projection operator Φ and a corresponding inverse one Φ^{-1}. We implement the STGAE by the spatial graph convolution and the temporal convolution.

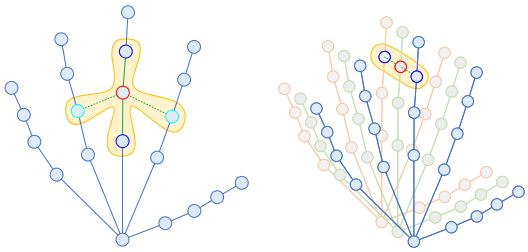
3.3.1 Spatial-Temporal Graph Construction

We construct the hand topology with spatial and temporal connections as an undirected spatial-temporal graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. In this work, a novel hand skeleton partition strategy is proposed.

As depicted in Fig. 2, the spatial-temporal graph consists of two basic parts: the spatial graph in Fig. 2(a) and the temporal graph in Fig. 2(b). The spatial graph shows the direct connected dependencies between neighbor joints and the indirectly linked relationships between symmetric neighbor joints. The temporal graph represents the continuity between consecutive frames of hand motion.

Each joint $\boldsymbol{v}_{t,i}$ has three kinds of neighbors: physic-connected neighbors $\boldsymbol{v}_{t,j}$ with direct intra-hand edges $\boldsymbol{v}_{t,i} \leftrightarrow \boldsymbol{v}_{t,j}$, symmetry-connected neighbors $\boldsymbol{v}_{t,k}$ with indirect intra-hand edges $\boldsymbol{v}_{t,i} \leftrightarrow \boldsymbol{v}_{t,k}$, as well as temporal neighbors $\boldsymbol{v}_{t-1,i}$ and $\boldsymbol{v}_{t+1,i}$ with indirect inter-frame edges $\boldsymbol{v}_{t-1,i} \leftrightarrow \boldsymbol{v}_{t,i}$ and $\boldsymbol{v}_{t,i} \leftrightarrow \boldsymbol{v}_{t+1,i}$. This effective hand skeleton partition strategy is helpful for noisy joints to obtain compensation from their trustworthy spatial-temporal neighbors.

Generally, the node feature $\boldsymbol{v}_{t,i}$ of the t-th frame and the i-th joint consists of the coordinates in the 2D or 3D space. The node set $\mathcal{V} = \{\boldsymbol{v}_{t,i} | t = 1, 2, \ldots, T, i = 1, 2, \cdots, N\}$ includes all the joints of the whole hand motion sequences with T frames and N joints. The edge set $\mathcal{E} = \{\boldsymbol{v}_{t,i} \leftrightarrow \boldsymbol{v}_{t,j}, \boldsymbol{v}_{t-1,i} \leftrightarrow \boldsymbol{v}_{t} | i = 1, 2, \cdots, N, t = 2, 3, \cdots, T\}$ comprises both the direct and indirect intra-hand edges as well as inter-frame edges.



(a) (b)

Figure 2: The illustration of the spatial-temporal graph: (a) is a spatial graph where the red circle denotes the center joint, blue circles represent its physic-connected neighbors (solid line) and cyan circles are the symmetry-connected ones (dashed line); (b) is a temporal graph where the red circle indicates the center joint and blue circles show its temporal neighbors of the before and after frame.

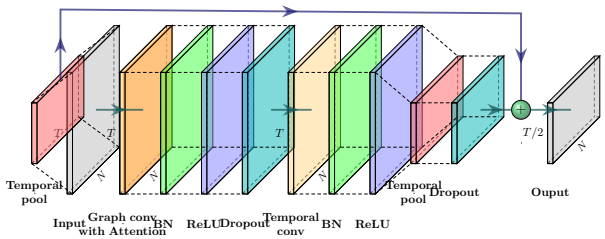


Figure 3: The visualization of the basic spatial-temporal encoder block including two parts: graph convolution sub-block and temporal convolution sub-block. 'conv', 'BN', 'pool' and 'Dropout' denote the convolution layer, the batch normalization layer, the pooling layer and the dropout layer respectively. '+' indicates the residual block. In the graph convolution sub-block, a self-attention mechanism is adopted followed by the graph convolution to learn which neighbors are trustworthy. For the temporal convolution sub-block, a pooling layer is mainly for the top-down process to capture the global features.

3.3.2 Graph Convolution

In a general way, the graph convolution operation can be implemented based on Kipf and Welling [12]. In this section, we introduce how to do convolution operation on the spatial-temporal graph.

Considering the hand joint data of the i-th frame $\tilde{\mathbf{X}}_i$ of size $N \times 3$, $\mathbf{Z}_i \in \mathbb{R}^{N \times F}$ is the result of the graph convolution operation.

$$\mathbf{Z}_i = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{X}}_i \mathbf{W}, \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the normalized adjacency matrix of size $N \times N$ with self-connections. $\tilde{\mathbf{D}}$ is the normalized degree matrix with the diagonal element $D_{ii} = \sum_j \tilde{A}_{ij}$ and $\mathbf{W} \in \mathbb{R}^{N \times F}$ is the filter parameter. Since this uniform operator of Equation 1 is appropriate to the dense graph but not to the sparse such as hand skeleton data, we adopt a nonuniform treatment of neighbor joints similar to [2].

Fig. 2(a) shows that each joint has two spatial relationships: direct (physic-connected) neighbors and indirect (symmetry-connected) neighbors. Therefore, the joint i as well as both its direct and indirect neighbors can be divided into a subset $\mathcal{S}_i$. If $j \in \mathcal{S}_i$, then set $A_{ij} = 1$; if the joint j is the direct/indirect neighbor of i, then set $A^{ij}_{\text{direct}}/A^{ij}_{\text{indirect}} = 1$. According to different relationships, the normalized adjacency matrix $\tilde{\mathbf{A}}$ can be dismantled into several matrices $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ where $\sum_k \mathbf{A}_k = \tilde{\mathbf{A}}$. In this work, we set $\mathbf{A}_1 = \mathbf{I}$, $\mathbf{A}_2 = \mathbf{A}_{\text{direct}}$ and $\mathbf{A}_3 = \mathbf{A}_{\text{indirect}}$. At this point, Equation 1 can be transformed into the following form.

$$\mathbf{Z}_i = \sum_{k=1}^{K} \tilde{\mathbf{\Lambda}}_k^{-\frac{1}{2}} \mathbf{A}_k \tilde{\mathbf{\Lambda}}_k^{-\frac{1}{2}} \tilde{\mathbf{X}}_i \mathbf{W}_k, \tag{2}$$

where $K = 3$ and $\tilde{\Lambda}^{ii}_k = \sum_j A^{ij}_k + \epsilon$. To avoid the empty row of $\mathbf{A}_k$, ϵ can be set to a little positive number, e.g., $\epsilon = 0.001$.

3.3.3 Attention mechanisms

If a joint has noisy neighbors, then it is less trustworthy to obtain the denoising compensation. To this end, we introduce a self-attention mechanism to dynamically learn which neighbors are reliable.

Multiplying Importance Mask To learn the importance of neighbor joints, one possible solution is to implement an attention mechanism by multiplying the mask $\mathbf{M}_k \in \mathbb{R}^{N \times N}$, which indicates the connection strength. Equation 2 is thus represented as:

$$\mathbf{Z}_i = \sum_{k=1}^{K} \left(\tilde{\mathbf{A}}_k \odot \mathbf{M}_k \right) \tilde{\mathbf{X}}_i \mathbf{W}_k, \tag{3}$$

where $\tilde{\mathbf{A}}_k = \tilde{\mathbf{\Lambda}}_k^{-\frac{1}{2}} (\mathbf{A}_k) \tilde{\mathbf{\Lambda}}_k^{-\frac{1}{2}}$ and $\odot$ denotes the element-wise operator. Similarly, Equation 1 can also adopt an attention mechanism. In this work, we initialize $\mathbf{M}_k$ with the all-one strategy.

However, this attention mechanism is so dependent on the prior neighbor connections $\tilde{\mathbf{A}}_k$ and the neighbor importance $\mathbf{M}_k$ that it ignores structural constraints between two non-neighbor joints.

Adding Learnable Terms To determine the contribution of non-neighbor joints, adding learnable terms is a good choice.

$$\mathbf{Z}_i = \sum_{k=1}^{K} \left(\tilde{\mathbf{A}}_k + \mathbf{B}_k \right) \tilde{\mathbf{X}}_i \mathbf{W}_k, \tag{4}$$

where $\mathbf{B}_k \in \mathbb{R}^{N \times N}$ is a learnable matrix, ensuring that noisy joints obtain the compensation from their non-neighbor joints. In this work, we initialize $\mathbf{B}_k$ with all-zero strategy.

$\mathbf{B}_k$ learns a static weight for every hand pose, which ignores the difference of hand noise. In this work, a self-attention mechanism is used to learn dynamic weights for every pose.

$$\mathbf{Z}_i = \sum_{k=1}^{K} \left(\tilde{\mathbf{A}}_k + \mathbf{B}_k + \mathbf{C}_k \right) \tilde{\mathbf{X}}_i \mathbf{W}_k, \tag{5}$$

where $\mathbf{C}_k \in \mathbb{R}^{N \times N}$ can be obtained by the scaled dot-product attention [29] without degrading the performance of Equation 4.

$$\mathbf{C}_k = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right), \tag{6}$$

where $\mathbf{Q} \in \mathbb{R}^{C \times d}$ and $\mathbf{K} \in \mathbb{R}^{C \times d}$ denote the query and the key respectively. The scale factor $1/\sqrt{d}$ is to counteract the dot products growing large in magnitude. Both the query $\mathbf{Q} = \tilde{\mathbf{X}}_i \mathbf{W}_{\text{query}}$ and the key $\mathbf{K} = \tilde{\mathbf{X}}_i \mathbf{W}_{\text{key}}$ are the embedding of the input data $\tilde{\mathbf{X}}_i$, where $\mathbf{W}_{\text{query}} \in \mathbb{R}^{C \times d}$ and $\mathbf{W}_{\text{key}} \in \mathbb{R}^{C \times d}$ are the corresponding embedding weight. $\mathbf{C}_k$ learns a unique connected topology for each hand pose $\tilde{\mathbf{X}}_i$ and measures the relationship between any two joints.

3.3.4 Auto-Encoder Architecture

Due to the success of the stacked hourglass network [23], the local-to-global scheme can effectively extract global features from the graph-based data. Follow this idea, we propose a novel spatial-temporal block with an hourglass structure.

Basic Spatial-Temporal Encoder Block Fig. 3 illustrates the architecture of a basic spatial-temporal encoder block, where both the graph convolution layer and temporal convolution layer are followed by a batch normalization layer, a ReLU layer, and a dropout layer. To determine reliable neighbors, a self-attention mechanism is applied in the graph convolution sub-block. Using 2D convolution operation with $1 \times \Gamma$ kernel and multiplying the attention term, the first hidden feature map $\mathbf{H}_1^{(l)}$ of l-th block can be obtained.

$$\mathbf{H}_1^{(l)} = \mathrm{ReLU}(\mathrm{Conv}_{\mathrm{graph}}(\mathbf{Z}^{(l)}, \tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C})), \tag{7}$$

where $\mathbf{Z}_i^{(l)}$ is the input of the l-th block. Different from the graph convolution sub-block, we introduce a top-down process in temporal domain. Specifically, an average pooling layer is added between the ReLU layer and the dropout layer. Similarly, the temporal convolution can be implemented as the 2D convolution with $\Omega \times 1$ kernel.

$$\mathbf{H}_2^{(l)} = \mathrm{AveragePool}(\mathrm{ReLU}(\mathrm{Conv}_{\mathrm{time}}(\tilde{\mathbf{X}}^{(l)})) \tag{8}$$

To ensure the same dimension, an additional temporal pooling layer is added to the whole residual block.

$$\mathbf{Z}^{(l+1)} = \mathrm{AveragePool}(\mathbf{Z}^{(l)}) + \mathbf{H}_2^{(l)}, \tag{9}$$

where $\mathbf{Z}^{(l+1)}$ is the output of the l-th block, as well as the input of the $(l+1)$-th block. It is noted that all the 2D convolution is equal to 1D convolution with the corresponding kernel size. For other blocks that do not require the top-down process in the temporal dimension, the average pooling layer in Fig. 3 can be omitted.

Hourglass Network Structure The encoder stacks 5 basic blocks. The numbers of output channels for each block are 3, 32, 32, 32, and 64. A data batch normalization layer is added at the beginning. For the second and fourth blocks, the average pooling layer is preserved but not for the others. The encoder learns a manifold projection mapping the input data into a hidden space.

$$\mathbf{H} = \Phi(\tilde{\mathbf{X}}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3), \tag{10}$$

where $\mathbf{H}$ is the manifold feature map. The decoder is also the stack of 5 basic blocks. The numbers of output channels for each block are 64, 32, 32, 32, and 3. In the second and forth blocks, the average pooling layer is replaced by the up-sampling layer. For the others, the average pooling layer is removed. The overall architecture is stabilized by a global residual structure as shown in Fig. 1. The decoder learns a corresponding inverse projection operator.

$$\mathbf{Y} = \tilde{\mathbf{X}} - \Phi^{-1}(\mathbf{H}, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3), \tag{11}$$

where $\mathbf{Y}$ is the denoised output of the same size as $\tilde{\mathbf{X}}$.

3.4 Loss Functions

During the training phase, the network reproduces the original input $\mathbf{X}$ from the input $\tilde{\mathbf{X}}$ following both the forward and backward operations. It is seen as the optimization problem by minimizing the loss function. In this work, the joint losses are applied in training.

Hand Pose Similarity Loss The pose similarity loss measures the mean squared error (MSE) of hand joints position between the ground truth $\mathbf{X}$ and the reproduction $\mathbf{Y}$.

$$\mathcal{L}_{\mathrm{pose}} = \mathbb{E}\left(\|\mathbf{Y} - \mathbf{X}\|_2^2\right), \tag{12}$$

where $\mathbb{E}(\cdot)$ denotes the mathematical expectation.

Hand Bone Length Loss The bone length loss weighs the MSE of bones length between two corresponding bones, which represents the physiological structural constraints of the hand joint.

$$\mathcal{L}_{\mathrm{bone}} = \mathbb{E}\left(\|\psi(\mathbf{Y}) - \psi(\mathbf{X})\|_2^2 | \mathbf{A}_1\right), \tag{13}$$

where $\psi(\cdot)$ represents the Euclidean distance function between two physic-connected joints.

Symmetric Neighbor Loss The symmetric neighbor loss evaluate the MSE of the Euclidean distance to indirect neighbors.

$$\mathcal{L}_{\mathrm{sym}} = \mathbb{E}\left(\|\psi(\mathbf{Y}) - \psi(\mathbf{X})\|_2^2 | \mathbf{A}_2\right), \tag{14}$$

Training Strategy Finally, we train the entire network in an end-to-end manner with the combined loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\mathrm{pose}} + \lambda_2 \mathcal{L}_{\mathrm{bone}} + \lambda_3 \mathcal{L}_{\mathrm{sym}}, \tag{15}$$

where $\lambda_1 = 1$, $\lambda_2 = 1$ and $\lambda_2 = 0.1$. To accelerate training, we only use the latter two losses during the fine-tuning phase.

4 Experiments

Based on a benchmark dataset, the synthetic dataset is first generated. The proposed model is compared with state-of-the-art approaches. Ablation studies examine the contribution of model components. Other quantitative and qualitative results are presented.

4.1 Dataset

In this work, we use NYU hand joint data [1] as ground truth. Both the training set and the test set are corrupted by algorithm 1.

The synthesized dataset contains 8252 consecutive test frames and 72757 consecutive training frames. Each frame consists of a frontal view and two side views. The training set contains samples from a single user while the test set is from two ones.

4.2 Evaluation Metrics

In this work, we evaluate models with three different metrics: hand pose similarity error, hand bone length error and hand symmetric neighbor error. The first one shows the similarity of two hand poses while the others indicate structural constraints of the hand.

Hand Pose Similarity Metric To evaluate the quality of denoising output $\mathbf{Y}$ compared with ground truth $\mathbf{X}$, the most intuitive method is to measure the MSE value between their corresponding hand joint positions. In such a way, the smaller the MSE of two corresponding hand poses, the better the denoising performance.

$$\mathrm{MSE}_{\mathrm{pose}}^i = \frac{1}{N} \sum_{j=1}^{N} \sum_{k=1}^{3} \left(X_{jk}^i - Y_{jk}^i\right)^2 \tag{16}$$

Hand Bone Length Metric Hand bone length error is a metric that represents a kind of structural constraint of the hand skeleton. The smaller the MSE of bone length between the two corresponding bones, the better the denoising performance.

$$\mathrm{MSE}_{\mathrm{bone}}^i = \frac{1}{|\mathcal{E}_{\mathrm{direct}}|} \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j} \left(\psi_{jk}(\mathbf{X}_i) - \psi_{jk}(\mathbf{Y}_i)\right)^2, \tag{17}$$

where $|\mathcal{E}_{\mathrm{direct}}|$ is the size of the direct edges set, $\mathcal{N}_j$ denotes the direct neighbors set of the joint j and ψ_{jk} measures the Euclidean distance between the joint j and the joint k.

Symmetric Neighbor Metric Similar to Equation 17, symmetric neighbor error measures the structural constraint between two symmetry-connected joints. The smaller the MSE of symmetric neighbor error between the two corresponding distances of symmetric neighbors, the better the denoising performance.

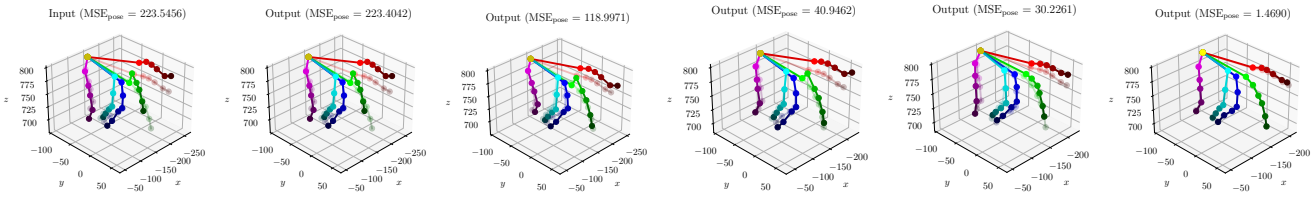
$$\mathrm{MSE}_{\mathrm{sys}}^i = \frac{1}{|\mathcal{E}_{\mathrm{indirect}}|} \sum_{j=1}^{N} \sum_{k \in \mathcal{N}_j^\star} \left(\psi_{jk}(\mathbf{X}_i) - \psi_{jk}(\mathbf{Y}_i)\right)^2, \tag{18}$$

where $|\mathcal{E}_{\mathrm{indirect}}|$ is the size of the indirect edges set, $\mathcal{N}_j^\star$ denotes the indirect neighbors set of the joint j and ψ_{jk} measures the Euclidean distance between the joint j and the joint k.

[1] NYU hand motion dataset: `https://jonathantompson.github.io/NYU_Hand_Pose_Dataset.htm`

Table 1: The comparison result of different methods on the test data set. We measure the mean squared pose error, mean squared bone length error and mean squared symmetric neighbor error for all joints across all frames in the test set, along with the inference time, the training time, and the size of model weights. The methods from top to bottom are joint-space encoder-bidirectional-filter network [22] (EBF), joint-space convolution neural network [9] (CNN), optical motion residual neural network [8] (ResNet), hand tremor compensation module based on graph neural network [13] (CAM-GNN) and ours, respectively.

#	Method	$\mathrm{MSE}_{\mathrm{pose}}$ (mm^2)	$\mathrm{MSE}_{\mathrm{bone}}$ (mm^2)	$\mathrm{MSE}_{\mathrm{sym}}$ (mm^2)	Inference (fps)	Model size (MB)	Training (h)
1	EBF	70.7740	13.5822	69.2183	143	4.3	8
2	CNN	170.3657	23.5246	390.9772	680	30.4	10
3	ResNet	59.8223	6.5408	89.1416	686	15.8	7
4	CAM-GNN	17.6803	3.5570	32.7914	152	10.6	9
5	Ours	**2.1741**	**0.5640**	**3.8091**	**2146**	**2.1**	**6**



(a) Input (b) EBF (c) CNN (d) ResNet (e) CAM-GNN (f) Ours

Figure 4: The comparison of denoising results for the state-of-the-art works. Though these methods work well in many cases, they are often unable to reconstruct the original motion when the input joints contain too many bad errors. From left to right: (a) raw uncleaned data, (b) EBF [22] , (c) CNN [9], (d) ResNet [8], (e) CAM-GNN [13] and (f) ours. Note that the shaded part in each sub-figure is the corresponding ground truth.

4.3 Implementation Details

Based on Tensorflow 2 framework, all experiments are implemented and conducted on one GeForce RTX 2080 Ti GPU with CUDA 10.1.

In the pre-processing phase, the corruption operation is elaborated in algorithm 1. The parameters σ_{o}, σ_{s} and β control the joints occlusions, joints swaps and joints noise respectively. We set $\sigma_{\mathrm{o}} = 0.1$, $\sigma_{\mathrm{s}} = 0.1$ and $\beta = 50\mathrm{mm}$. For the dataset split, the training set is first divided into batches of 36 frames and then 15% of them are randomly sampled as the validation set. In the training phase, we adopt an early stopping mechanism to avoid overfitting with a mini-batch size of 32 using the Adam optimizer. The learning rate is reduced by 80% from 0.01 while the validation loss is increasing. The maximum patience of the early stopping mechanism and the minimum learning rate are set to 10 and 10^{-7} respectively.

4.4 Results and Analysis

This section shows the comparison between our method and the state-of-the-art methods. We also present ablation studies and denoising results. Through a series of experiments, we verify the effectiveness of our proposed method. The results are analyzed in detail.

4.4.1 Comparisons with the State-of-the-art

We compare our method on the test set with the state-of-the-art motion capture data denoising methods [8,9,13,22] and measure the above three metrics for joints across all frames on the test set. Table 1 reports the detailed comparison results along with the inference time, training time, and size of model weights. Note that for simply evaluate the memory usage of models, we only statistic the space taken up by the number of saved model parameters. Fig. 4 displays the denoising results of these state-of-the-art works.

For a fair comparison, we use the same number of layers in all baselines and adjust the number of hidden units such that they have roughly the same memory allowance. Our comparison finds that although all the state-of-the-art methods achieve a good level of performance, the proposed method achieves the best results on the test data in terms of all the metrics used in this work.

Robust Denoising Performance Although all the methods perform well in most cases, Fig. 4 shows that our method achieves the most robust performance when dealing with extremely abnormal motion poses. As shown in Table 1, our model performs best under all three metrics: mean squared pose error, mean squared bone length error, and mean squared symmetric neighbor error. Compared with these methods, our method can capture spatial-temporal patterns at the same time, so it performs the best in dealing with motion data. In addition, due to the consideration of more structural constraints, our proposed method achieves significantly better performance in terms of bone length and symmetric neighborhood errors.

Dynamic Inference As elaborated in Table 1, our method has a high frame rate and can process up to 2146 frames per second in the inference stage. Compared with the EBF method, our method adopts a dynamic inference decoder without waiting for 15 frames after the current frame to start generating outputs. Compared with the others, although they have the same number of layers, they take significantly more time for inference due to the large number of parameters of their networks. It is worth mentioning that all the methods can meet the real-time processing requirements in the inference phase.

Low Memory Usage and Short Training Time Among these methods in Table 1, the space occupied by model parameters of our method is the smallest and the training time is the shortest. Through the effective parameter sharing mechanism, our model is much lightweight for practice. Compared with CAM-GNN, ours fuses spatial and temporal features with less trainable parameters, accelerating model convergence and achieving better performance.

4.4.2 Ablation Studies

We perform ablation studies to investigate the contribution of different components of our proposed method on the synthesized dataset. All the ablation studies are evaluated on the above three metrics.

Attention Mechanisms We compare the performance of different attention mechanisms and no attention module scenario, i.e., only using $\tilde{\mathbf{A}}$ like Equation 2. In this work, we study two different kinds of attention mechanisms. One is similar to Equation 3,

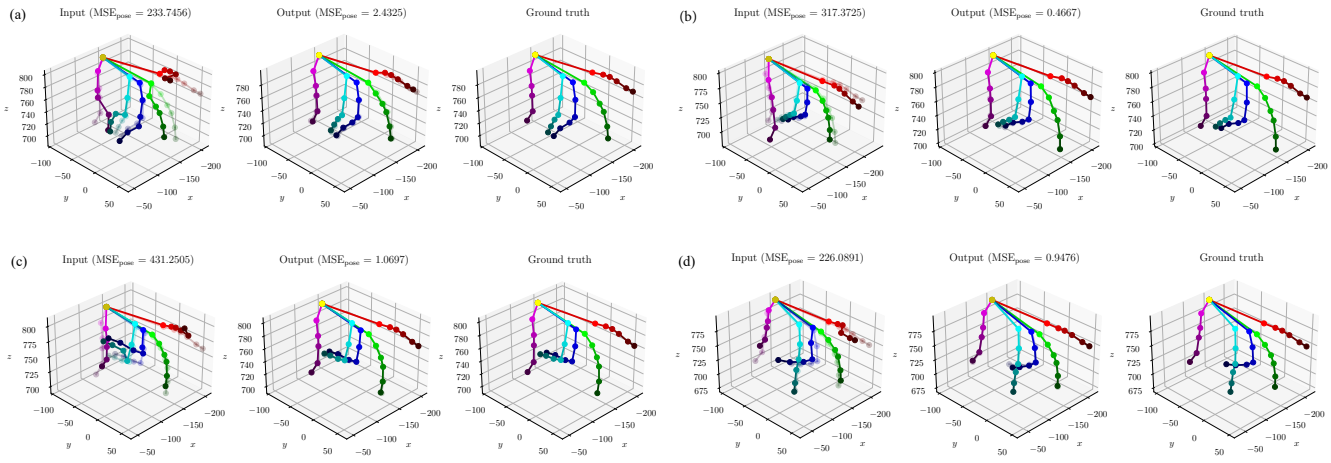


Figure 5: The visualization of hand motion denoising using STGAE: (a) to (d) are the results of four different frames. In each sub-figure, from left to right are the input pose, the denoising output and the ground truth. To distinguish the difference between the input/output and the ground truth, the shaded part represents the ground truth on the corresponding figures. Note that the top 30 joints are displayed for the best view.

multiplying a mask to learn edge importance weights. The other is adding learnable parts to $\tilde{\mathbf{A}}$ analogous to Equation 5. Furthermore, we delete learnable components one by one for the latter to verify the effectiveness of the proposed dynamic attention module.

Table 2: The ablation study on attention mechanisms (mm^2)

#	Methods	MSE$_{\text{pose}}$	MSE$_{\text{bone}}$	MSE$_{\text{sym}}$
1	$\tilde{\mathbf{A}}$	15.8325	4.6069	26.2338
2	$\tilde{\mathbf{A}} \odot \mathbf{M}$	13.7035	4.0235	22.7004
3	$\mathbf{B} + \mathbf{C}$	13.6936	4.7644	23.8084
4	$\tilde{\mathbf{A}} + \mathbf{C}$	13.1479	4.3135	20.6892
5	$\tilde{\mathbf{A}} + \mathbf{B}$	14.4268	3.4743	30.0313
6	$\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C}$	2.3879	0.6945	4.3162

Table 2 reports the results of ablation studies on attention mechanisms. From top to bottom, Table 2 lists non-attention mechanism $\tilde{\mathbf{A}}$, masked attention mechanism $\tilde{\mathbf{A}} \odot \mathbf{M}$, as well as four additive types of attention mechanism $(\mathbf{B} + \mathbf{C})$, $(\tilde{\mathbf{A}} + \mathbf{C})$, $(\tilde{\mathbf{A}} + \mathbf{B})$ and $(\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C})$. Without any attention mechanism, the MSE of the hand pose is up to 15.8325. Both the two kinds of attention mechanisms outperform it, indicating the effectiveness of attention mechanisms. The attention mechanism makes the GCN get rid of the dependence of graph structure, leading to stronger generalization performance. Overall, the second kind of attention mechanism works better than the first. The reason is that learning neighbor importance by multiplying masks doesn't change graph topology while adding some learnable components ensure that the graph structure can be completely adjusted. For the second kind of attention mechanism, $(\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C})$ performs the best, indicating the effectiveness of all the learnable components. Deleting $\tilde{\mathbf{A}}$ causes the graph losing the topological prior and makes the network hard to converge compared with $(\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C})$. Both deleting $\mathbf{B}$ and $\mathbf{C}$ are not conducive to dynamically learn which neighbors are trustworthy.

Partition Strategies In this work, we propose a simple yet effective partition strategy where the indirect symmetric connections are also served as the edges of the spatial-temporal graph. To explore the influence of different types of connections on the performance of the proposed model, we conduct ablation studies by removing one type of connection each time, i.e., self-connections $\mathbf{A}_1$, physic-connections $\mathbf{A}_2$, and symmetry-connections $\mathbf{A}_3$.

Table 3: The ablation study on partition strategies (mm^2)

#	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	MSE$_{\text{pose}}$	MSE$_{\text{bone}}$	MSE$_{\text{sym}}$
1	✓	✓	✗	9.1182	2.3199	16.9761
2	✓	✗	✓	11.1967	2.5976	21.0983
3	✗	✓	✓	14.0478	6.4166	26.0971

Table 3 elaborates the detail results of ablation study on partition strategies. Form top to bottom, Table 3 shows the results of deleting $\mathbf{A}_3$, $\mathbf{A}_2$ and $\mathbf{A}_1$ scenarios, respectively. Compared with the performance of adaptive attention mechanism $(\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C})$ in Table 2, it is obvious that removing any of these connections greatly affects the error of the proposed model. Among them, deleting the self-connection item $\mathbf{A}_1$ has the greatest impact, indicating that the self-connection in the graph is the most important. To some extent, self-connection represents that the motion of each joint is continuous in the temporal domain. Second, the influence of immediate neighbors $\mathbf{A}_1$ is also very large. It is clear that there are strong structural constraints between physic-connected joints. The effect on indirect connections $\mathbf{A}_2$ is minimal, indicating that symmetrical neighbors are the least important compared with self-connections and direct connections. Nevertheless, the removal of symmetry-connected relationships still results in significant performance degradation, illustrating the effectiveness of the proposed partition strategies.

4.4.3 Denoising Performance

To better display the experimental results, we select four frames from a piece of motion data containing 36 frames, as shown in Fig. 5. For the entire continuous animation, see the supplementary video. Fig. 6 shows the motion trajectory of the hand joint at the end of the index finger in this continuous motion data and the error result curve. Fig. 7 is an example of the original and learned adjacent matrix.

Qualitative Results In each of the selected frames in Fig. 5, the input, output, and ground truth are shown from left to right where the shaded part denotes the corresponding ground truth. Note that our method can not only learn the non-corrupted function but also the corrupted function, Fig. 5 only shows the corrupted results for visualization. It can be seen that all the left inputs are distorted,

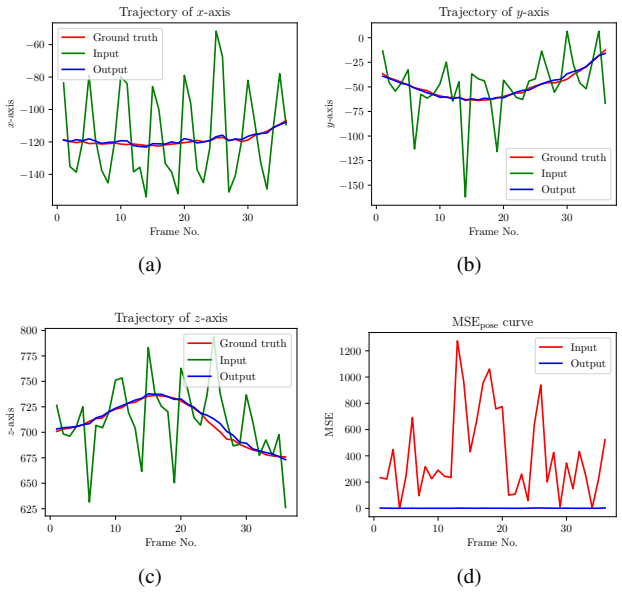


Figure 6: Motion trajectory and pose error curve plots. (a), (b) and (c) are the motion trajectory of the index finger tip on x-axis, y-axis and z-axis, respectively. The trajectory of the input, output and ground truth are colored in green, blue and red respectively. (d) is the corresponding MSE of the pose similarity curve where the red line and the blue are the pose error changes of noisy input and denoised output.

occluded, or warped. Obviously, the middle denoising results are almost exactly the same as the ground truth. From a qualitative point of view, the proposed model achieves amazing denoising performance. In Fig. 6(a), Fig. 6(b) and Fig. 6(c), the red line, green line and blue line show the index finger-tip trajectory of x-axis, y-axis and z-axis respectively. It can be seen that the vibration of the input data after the corruption is very obvious. At the same time, the output trajectory after denoising by our model is very close to the ground truth indicating that the proposed method is powerful.

Quantitative Results Further, the quantitative measurement between the input/output and its corresponding ground truth is indicated at the top of each sub-figures in Fig. 5. The quantitative results are so encouraging that the proposed method can reduce the error from very high values down to about 1, e.g., the input hand pose similarity error in Fig. 5(a) is reduced from 233.7456 to 2.4325. Note that the above measurements are in millimeters. Fig. 6(d) depicts the pose error change curve of these consecutive frames, where the red line and blue line are the MSE curves of the input and the output respectively. Compared with the input, the MSE curve of the output is almost flat and close to zero, indicating the huge power of ours.

The Visualization of the Learned Adjacent Matrix In Fig. 7, a learned adjacent matrix heatmap and its original normalized adjacent matrix are shown where the colorful scale of each element in the matrix represents the strength of the connection. Fig. 7(a) is the original normalized adjacent matrix heatmap where self-connections, physic-connected connections and symmetry-connected connections are considered. Fig. 7(b) is an example of its corresponding learned adjacency matrix by the proposed model. Note that both the original normalized adjacent matrix and the learned have 3 channels, Fig. 7 shows that the effect of all channels is overlaid. It is clear that the learned structure of the graph is more adaptive and not constrained to the physical and physiological constraints, which can give full play to the advantages of graph neural network.

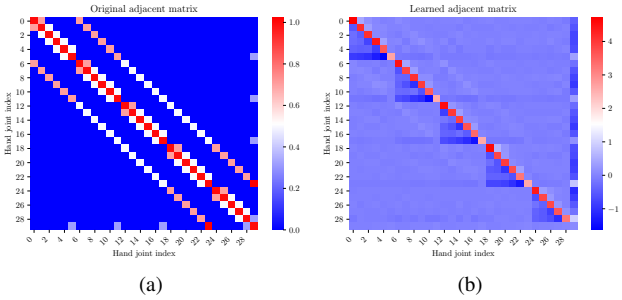


Figure 7: The visualization of the original and learned adjacent matrix: (a) is the original one for the synthetic dataset including self-connections, physic-connections and symmetry-connections; (b) is an example of the corresponding learned matrix. Note that we only illustrate the first 30 joints of the NYU hand model for the best display.

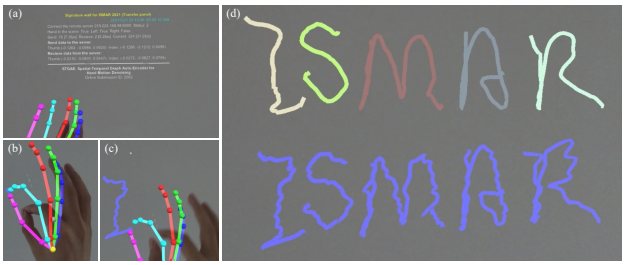


Figure 8: A demo of the handwriting application in AR scene: (a) the control panel; (b) the pose of beginning to write; (c) the pose of ending to write; (d) the handwriting of characters 'ISMAR' under tremor (bottom) and the corresponding denoised output (top).

4.4.4 AR Application

The proposed method is integrated into the Universal Windows Platform of Microsoft HoloLens 2. We can achieve realistic effects with loyal user intent, which is illustrated in Fig. 8. Please refer to the supplementary materials of the complete demo video.

As shown in Fig. 8(d), the bottom row is the handwriting of characters 'ISMAR' under hand tremor after fatigue manipulation. The output shown in the top row is more fluent than the input. In the AR application, we take the intermediate frame in a window of size T as the final output for the best denoising effect. Thus, the output is always delayed by $T/2$ frames. Furthermore, the data transmission between the server and the client is also a reason for the time delay.

5 Conclusion

Raw hand motion data with errors does not meet the intention of users and brings a serious challenge to immersive interaction in MR. In this work, we have proposed an end-to-end method for hand motion denoising called STGAE. We first develop a joint corruption algorithm to ensure that the bone length constraint is preserved. The time-continuous articulated structure of the hand forms a natural spatial-temporal graph topology, bringing inherent advantages to dig out the spatial-temporal patterns using spatial-temporal graph neural network. By introducing a self-attention mechanism, the graph topology can be dynamically adjusted along with the propagation. Experiments show that STGAE outperforms state-of-the-art works.

Although the results are encouraging, there are also some tricky problems, such as dramatic changes of motion. In the future, the muscle-skeletal model will be introduced to synthesize more realistic abnormal motion and the kinematic features will be considered to achieve a more faithful motion intention estimation.

ACKNOWLEDGMENTS

The authors would like to thank all reviewers for their thoughtful comments. This work was supported by the National Key R&D Program of China (Project Number: 2017YFB1002702) and the National Natural Science Foundation of China (Project Number: 61572058).

REFERENCES

[1] M. Burke and J. Lasenby. Estimating missing marker positions using low dimensional kalman smoothing. *Journal of biomechanics*, 49(9):1854–1858, 2016.

[2] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2272–2281, 2019.

[3] J. Chen, J. Chen, H. Chao, and M. Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2018.

[4] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh. Clustergcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 257–266, 2019.

[5] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3837–3845, 2016.

[6] Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J. J. Zhang, and R. Song. Exploiting temporal stability and low-rank structure for motion capture data refinement. *Information Sciences*, 277:777–793, 2014.

[7] C. Ferles, Y. Papanikolaou, and K. J. Naidoo. Denoising autoencoder self-organizing map (dasom). *Neural Networks*, 105:112–131, 2018.

[8] D. Holden. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[9] D. Holden, J. Saito, T. Komura, and T. Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, pages 1–4, 2015.

[10] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang, and G. Liu. Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting. *IEEE Transactions on Sustainable Energy*, 11(2):571–583, 2019.

[11] S. U. Kim, H. Jang, and J. Kim. Human motion denoising using attention-based bidirectional recurrent neural network. In *SIGGRAPH Asia 2019 Posters*, pages 1–2, 2019.

[12] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[13] Z. Leng, C. Jiaying, H. Shum, F. Li, and X. Liang. Stable hand pose estimation under tremor via graph neural network. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces*, 2021.

[14] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[15] L. Li, J. McCann, N. Pollard, and C. Faloutsos. Bolero: A principled technique for including bone length constraints in motion capture occlusion filling. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10)*, pages 179–188, 2010.

[16] S. Li, H. Wang, and D. Lee. Hand pose estimation for hand-object interaction cases using augmented autoencoder. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 993–999, 2020.

[17] Y. Li, Z. He, X. Ye, Z. He, and K. Han. Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition. *EURASIP Journal on Image and Video Processing*, 2019(1):1–7, 2019.

[18] X. Liu, Y.-m. Cheung, S.-J. Peng, Z. Cui, B. Zhong, and J.-X. Du. Automatic motion capture data denoising via filtered subspace clustering and low rank matrix approximation. *Signal Processing*, 105:350–362, 2014.

[19] Z. Liu, C. Chen, L. Li, J. Zhou, X. Li, L. Song, and Y. Qi. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4424–4431, 2019.

[20] H. Lou and J. Chai. Example-based human motion denoising. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):870–879, 2010.

[21] A. Majumdar. Blind denoising autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, 30(1):312–317, 2018.

[22] U. Mall, G. R. Lal, S. Chaudhuri, and P. Chaudhuri. A deep recurrent framework for cleaning motion capture data. *arXiv preprint arXiv:1712.03380*, 2017.

[23] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499, 2016.

[24] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6519–6528, 2019.

[25] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang. Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 16(2):241–245, 2018.

[26] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019.

[27] F. P. Such, S. Sah, M. A. Dominguez, S. Pillai, C. Zhang, A. Michael, N. D. Cahill, and R. Ptucha. Robust spatial filtering with graph convolutional neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):884–896, 2017.

[28] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, 2020.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008, 2017.

[30] H. Wang, E. S. L. Ho, H. P. H. Shum, and Z. Zhu. Spatio-temporal manifold learning for human motions via long-horizon modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):216–227, 2021.

[31] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2020.

[32] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7444–7452, 2018.

[33] L. Yang, J. Huang, T. Feng, W. Hong-An, and D. Guo-Zhong. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware*, 1(1):84–112, 2019.

[34] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang. Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE Transactions on Medical Imaging*, 37(6):1348–1357, 2018.

[35] Q. Ye and T.-K. Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–817, 2018.

[36] Y. Zhu. Denoising method of motion capture data based on neural network. In *Journal of Physics: Conference Series*, page 032068. IOP Publishing, 2020.