Multi-Task Spatial-Temporal Graph Auto-Encoder for Hand Motion Denoising

Kanglei Zhou, Hubert P. H. Shum, *Senior Member, IEEE*, Frederick W. B. Li, Xiaohui Liang

Abstract—In many human-computer interaction applications, fast and accurate hand tracking is necessary for an immersive experience. However, raw hand motion data can be flawed due to issues such as joint occlusions and high-frequency noise, hindering the interaction. Using only current motion for interaction can lead to lag, so predicting future movement is crucial for a faster response. Our solution is the Multi-task Spatial-Temporal Graph Auto-Encoder (Multi-STGAE), a model that accurately denoises and predicts hand motion by exploiting the inter-dependency of both tasks. The model ensures a stable and accurate prediction through denoising while maintaining motion dynamics to avoid over-smoothed motion and alleviate time delays through prediction. A gate mechanism is integrated to prevent negative transfer between tasks and further boost multi-task performance. Multi-STGAE also includes a spatial-temporal graph autoencoder block, which models hand structures and motion coherence through graph convolutional networks, reducing noise while preserving hand physiology. Additionally, we design a novel hand partition strategy and hand bone loss to improve natural hand motion generation. We validate the effectiveness of our proposed method by contributing two large-scale datasets with a data corruption algorithm based on two benchmark datasets. To evaluate the natural characteristics of the denoised and predicted hand motion, we propose two structural metrics. Experimental results show that our method outperforms the state-of-the-art, showcasing how the multi-task framework enables mutual benefits between denoising and prediction.

Index Terms—Hand motion denoising, Hand motion prediction, Graph convolutional network, Multi-task learning

1 Introduction

WITH the rapid advancement of Human-Computer Interaction (HCI) techniques, human hands play a vital role in performing operations such as grasping and manipulating objects [42], [58]. In many HCI applications [57] such as Virtual/Augmented Reality (VR/AR), providing an immersive experience for users relies on quickly tracking human hands and accurately estimating the corresponding hand poses [27], [40]. That said, on the one hand, the complex articulations, self-occlusion, and self-similarity of hands make immersive interaction challenging [54]. In addition, prolonged operation and movement disorders such as Parkinson's disease may lead to involuntary handshakes [23], resulting in interaction failures. On the other hand, the user experience deteriorates when applications lag due to delays in the processing and rendering pipeline [45]. Predicting future motion would be beneficial for pre-processing and improving response time [38].

The first challenge of inaccurate tracking and handshakes can be tackled by denoising. Existing motion denoising algorithms [8], [17], [18], [23] mainly focus on human body data. Due to the high degree of freedom of the hand model [41], the relative noise amplitude of hand motion data is larger than that of human motion data. Therefore, applying these human motion denoising methods directly to hand motion data is ineffective. Also, the use of Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) to separately model spatial dependence and temporal relationships is insufficient to identify unified spatial-temporal patterns in hand motion data, as these methods [8], [17], [18] do not satisfy structural constraints or provide temporal coherence. Furthermore, denoising alone often leads to over-smoothing problems [5], which results in a loss of temporal dynamics. Thus, it may be difficult to understand user intention accurately.

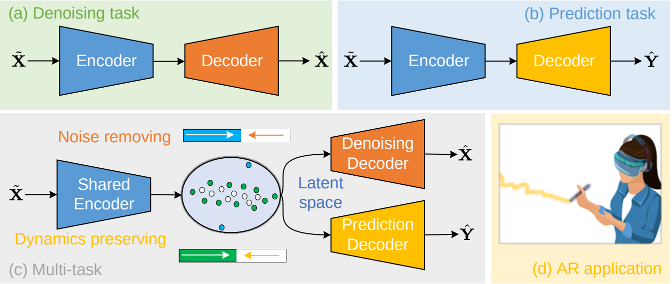

Fig. 1. An illustration of our main idea. We integrate denoising (a) and prediction (b) to propose a multi-task method, which can be used in AR applications (d). Denoising removes noise from the latent space, resulting in more accurate and stable predictions, while prediction maintains motion dynamics, preventing over-smoothed motion.

The second challenge of improving system response speed can be tackled by prediction. Existing motion pre-

- K. Zhou is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China.
  E-mail: zhoukanglei@buaa.edu.cn
- H. Shum and F. Li are with the Department of Computer Science, Durham University, Durham DH1 3LE, United Kingdom.
  E-mail: {hubert.shum, frederick.li}@durham.ac.uk
- X. Liang is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China, and also with Zhongguancun Laboratory, Beijing, China.
  E-mail: liang_xiaohui@buaa.edu.cn

Manuscript received XXX; revised XXX.
(Corresponding author: Xiaohui Liang)

diction algorithms [6], [33], [34] mainly focus on human body data, while a few [6] are designed for hand motion prediction. It is more challenging to realize real-time hand gesture prediction as the hand structure is more complex than the human body [6]. Therefore, simply applying the human motion prediction algorithms [34] to the hand may not guarantee optimal performance. It is also difficult to adapt hand prediction methods [6], [33] to real-world application scenarios as they have poor generalization using traditional statistical learning methods. Moreover, prediction alone ignores the noise that exists in the raw data, making it difficult to accurately predict future dynamics that align with the user's intention.

We observe that hand motion denoising and prediction are highly interdependent and mutually beneficial. Denoised motion allows a more accurate and stable prediction, while prediction helps inform the motion dynamics, thereby preventing denoising from creating over-smoothed motion. Therefore, our core insight is to incorporate denoising and prediction to remove noises from the latent space while maintaining motion dynamics. As hand motion data is a sequence of natural graphs, spatial-temporal graph convolutional networks (STGCN) have achieved great success in human motion analysis [43] for feature representation. This motivates us to leverage STGCN to model the structural priors and the temporal coherence for hand motion denoising and prediction.

In this paper, we propose Multi-STGAE, the Multi-task Spatial-Temporal Graph Auto-Encoder, for jointly denoising hand motions and predicting future movements. As visualized in Fig. 1, Multi-STGAE is essentially a two-branch network with a shared encoder, by exploiting the inter-dependency of the two tasks. Notably, we demonstrate that denoising hand motion and predicting future motion are two interconnected tasks so that the multi-task framework achieves better performance than the single denoising task through domain sharing between complementary tasks. To avoid the possible negative transfer between the two tasks, a simple yet effective gate mechanism has been developed in contrast to the previous work [9]. This allows the effective information to be transferred to the corresponding downstream branch flows, thus enhancing multi-task performance further.

The core of Multi-STGAE is the spatial-temporal graph autoencoder block, which explicitly models the structural priors of hands and the temporal coherence of motion through spatial-temporal graph convolutional networks. It enables the transfer of corrective data from reliable neighboring joints to those that are noisy, reducing the noise while maintaining the physiological constraints of the hand. Different from the hand motion compensation method [23] that only relies on the physic-connected connections, our method utilizes both the hand symmetry structure prior and the temporal correlation to enhance the information compensation. Additionally, we explore several learning strategies for graph structures in order to increase their flexibility and adaptability.

To evaluate the performance of our method, we have created two large-scale datasets by applying a data corruption algorithm on NYU and SHREC, two hand pose estimation and gesture recognition datasets. For the purpose of increasing the diversity of the dataset, the selected two datasets have different topological structures. The experimental results demonstrate that our method outperforms the state-of-the-art, showcasing the benefits of our multi-task framework for denoising and prediction.

Our preliminary results of a denoising-only method have been presented in [59]. This paper has made significant advancements and presents the following technical novelty. (1) We have designed and implemented a new multi-task framework for hand motion denoising and prediction, allowing a stable and fast interaction experience for users. (2) We have included new materials in the paper, including the design and justification to explain the new multi-task framework, with a light touch to explain how individual components of [59] are adapted. (3) We have conducted new experiments to fully evaluate the proposed system with an additional dataset, SHREC, alongside the NYU dataset, to provide a comprehensive and diverse evaluation of our system. We evidence the mutual benefit of the two tasks and the performance advancement over the state-of-the-art, including [59]. (4) We have conducted new qualitative experiments, which are presented in the paper as well as a newly created video.

Our main contributions are summarized as follows:

- We propose a novel multi-task network that considers prediction and denoising at the same time, exploiting their inter-dependency to boost the performance of both tasks. The source code is available at https://github.com/ZhouKanglei/Multi-STGAE
- We propose a simple yet effective gate mechanism to avoid the negative transfer between different tasks, which can further improve the performance of denoising and prediction.
- We propose an improved hand motion denoising method built upon stacked spatial-temporal graph convolutional blocks (STGCBs), which utilize a novel skeleton partition strategy along with a dynamic self-attention mechanism to preserve the structural constraints of hands.
- To facilitate the algorithm performance verification, we propose two new synthesized datasets, which can be used as a benchmark for hand motion denoising and prediction.

The remainder of this paper is organized as below. We first review the related works in Sect. 2. Then, we elaborate on the proposed multi-task framework in Sect. 3. We validate the proposed method with a wide range of experiments in Sect. 4. Finally, we conclude the whole paper and discuss future directions in Sect. 5.

2 Related Work

We first review prior works on denoising and predicting motion data. Then, focus the discussion on graph networks for modeling human motion. Finally, we review related works on multi-task learning.

2.1 Motion Denoising

There are two main categories of methods for denoising motion data: prior knowledge-based and machine learning-based approaches. While most existing works focus on denoising motion data for human bodies, these methods provide valuable insights and inspiration for developing techniques to remove noise from hand motion data.

Motion data is a spatial-temporal signal, and two priors can be used to identify and remove noise: the spatial dependency between joints [25] and the temporal relationship over time [2], [13], [30], [32]. The spatial dependency means that the motion of different joints in the body is highly correlated, aiding in identifying the likelihood of noise at a certain joint based on the motion patterns of adjacent joints. Li *et al.* [25] proposed BoLeRO, which uses hard and soft constraints to preserve bone length constraints. Temporal relationship over time refers to the fact that motion data is a time-series signal, and the motion patterns of adjacent frames are often highly correlated. Lou *et al.* [32] proposed a data-driven method for learning filter bases from motion data and utilizing temporal relationships to estimate underlying motion trajectories and remove noise. In contrast to using motion databases, Feng *et al.* [13] incorporated the low-rank structure and temporal stability properties of motion data to refine motion capture data.

Building upon prior works [7], [14], [35], [53], machine learning-based methods [17], [18], [20], [23] for denoising motion data have been developed. Holden *et al.* [17], [18] proposed a convolutional auto-encoder to learn the motion manifold and reconstruct corrupted motion data. Kim *et al.* [20] introduced a bidirectional recurrent neural network with an attention mechanism to improve denoising accuracy by emphasizing important input poses. Leng *et al.* [23] proposed a method to estimate hand pose during tremors using a WaveNet and a graph neural network. However, non-adjacent joint constraints were not taken into account, which is an important factor for denoising.

The machine learning-based method is preferred over the prior knowledge-based method as it addresses limitations such as poor generalization and manual parameter setting [47]. Despite achieving significant results, machine learning-based methods are limited by the absence of structural relationships between joints. To this end, our proposed method incorporates hand-prior knowledge (in the form of graph design) into the machine learning system, thereby combining the benefits of both approaches.

2.2 Motion Prediction

We focus on reviewing deep learning-based methods for predicting human motion. As hand and body motion prediction share many similarities, while there are fewer existing methods for hand motion prediction [6], [33], we widen our scope to include body motion prediction, which could potentially offer insights into predicting hand motion.

These methods can be categorized based on the network design [34]. Spatial-temporal RNN [37] has been proposed for human motion prediction that utilizes skeletal information for feature extraction. Batch prediction addresses the ineffective temporal modeling of motion multi-modality and variances, resulting in accurately predicting long-duration motions [50]. To improve prediction performance, some RNN variants [15], [44] have been proposed. CNNs [9], [10], [26], [56] have also been incorporated due to their ability to capture spatial dependencies. GNN is adaptable for representing the human skeleton, making it widely used in human motion prediction. Li *et al.* [26] used a multi-scale graph to model the internal relations of the human body for feature learning and fused them across scales. They modeled temporal relationships using GRU. Additionally, Zhong *et al.* [56] employed Temporal Convolutional Networks (TCN) to predict future dynamics. GANs have also been used for learning the distribution of motion sequences and generating more diverse and realistic motions than deterministic models [16], [22], [31], producing realistic motions. In this context, Barquero *et al.* [1] delved into behavior prediction during dyadic conversations, emphasizing full-body dynamics, and showcased that transformer-based models, particularly their temporal transformer, achieve state-of-the-art results. Concurrently, Palmero *et al.* [39] underscored challenges in behavior forecasting, hinting at the potential of multi-modal architectures in future research.

Overall, there are still many challenges to overcome in motion prediction, such as dealing with noisy or incomplete historical motion sequences [37]. The objective of our work is to investigate how incomplete observation data can be reconstructed into complete data and to predict the future dynamics of the motion.

2.3 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) [21], [51] aggregate information from neighbors and learn intrinsic features, making them ideal for exploiting the structural constraint relationships between human joints. Due to the large quantity of time-series data available, STGCN and its variants have become very popular for this task.

STGCN is a powerful method that combines GCN with Long Short-Term Memory (LSTM) or TCN. It has been widely used in action recognition [43], [52] and time-series forecasting [50], among other fields [3]. Yan *et al.* [52] first proposed STGCN for skeleton-based action recognition, where GCNs handle spatial modeling and TCN focuses on temporal modeling, allowing both spatial and temporal patterns to be learned simultaneously. This approach has become one of the most commonly used paradigms for processing human motion data. Shi *et al.* [43] further improved the performance of STGCN by introducing a self-attention mechanism to learn an adaptive adjacent matrix. Cai *et al.* [3] extended STGCN to 3D pose estimation by exploiting multi-scale features in graph-based representations.

Motivated by the success of STGCN, we propose a novel approach called STGAE [59] that combines STGCN with a hand skeleton partitioning strategy based on hand symmetry priors to effectively denoise hand motion data. However, there is a risk of over-smoothing the results and losing temporal dynamics with existing STGCN-based methods. To address this, our method incorporates motion prediction and motion denoising into a multi-task architecture to enhance temporal dynamics and improve performance.

2.4 Multi-Task Learning

Multi-task learning is a powerful technique that has been applied to various applications [12], [19], [55], and it has shown promising results in improving the performance of

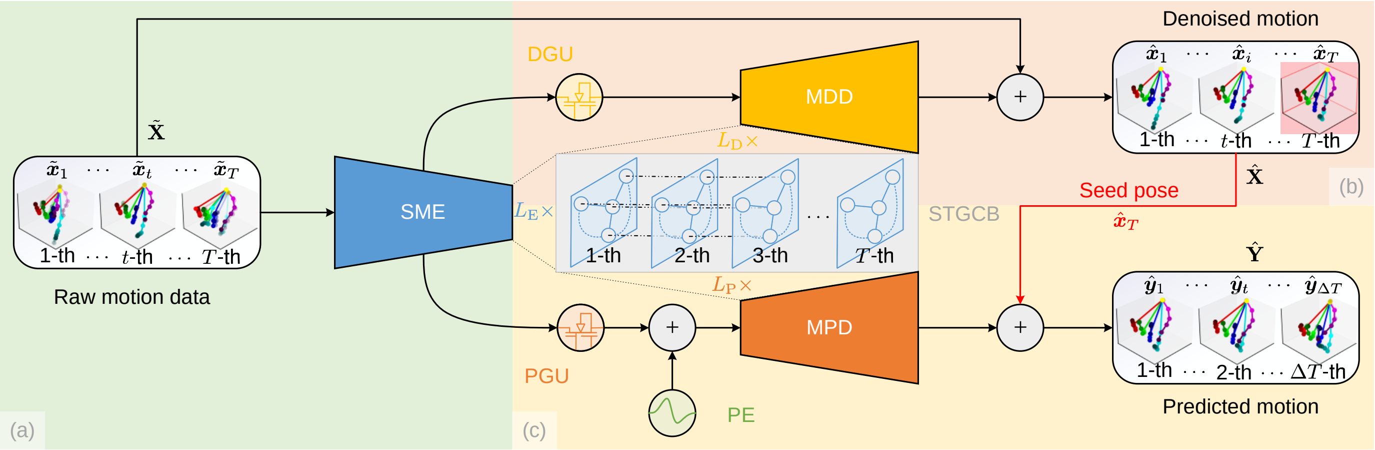
Denoised motion
$\hat{\boldsymbol{x}}_1$ ··· $\hat{\boldsymbol{x}}_i$ ··· $\hat{\boldsymbol{x}}_T$
1-th ··· t-th ··· T-th
DGU
MDD
$L_{\mathrm{D}}\times$
$\tilde{\mathbf{X}}$
$\tilde{\boldsymbol{x}}_1$ ··· $\tilde{\boldsymbol{x}}_t$ ··· $\tilde{\boldsymbol{x}}_T$
1-th ··· t-th ··· T-th
Raw motion data
SME
$L_{\mathrm{E}}\times$
1-th
2-th
3-th
···
T-th
STGCB
$L_{\mathrm{P}}\times$
Seed pose
$\hat{\boldsymbol{x}}_T$
$\hat{\mathbf{X}}$
(b)
$\hat{\mathbf{Y}}$
$\hat{\boldsymbol{y}}_1$ ··· $\hat{\boldsymbol{y}}_t$ ··· $\hat{\boldsymbol{y}}_{\Delta T}$
1-th ··· 2-th ··· ΔT-th
Predicted motion
PGU
MPD
PE
(a)
(c)

Fig. 2. Framework overview of our proposed method Multi-STGAE: we utilize the prediction task to propose a multi-task framework for hand motion denoising. Through this framework, the denoised result is capable of preserving the temporal dynamics and the time delay problem can be greatly alleviated. In this way, it is possible to provide users with a satisfying experience during the interaction.

individual tasks by leveraging shared feature representations. In this section, we primarily review works of multi-task learning to human motion processing [9], [24], [28].

Learning multi-task [55] is a paradigm for tackling related tasks simultaneously while constructing a shared structure to enhance the overall performance of the task, which can be viewed as an inductive approach to knowledge transfer. To improve human motion prediction, a multi-task framework was proposed by Cui *et al.* [9], which focused on accurately forecasting future human actions and repairing incomplete observations. Li *et al.* [24] also proposed a multi-task training paradigm for low-level human skeleton prediction and high-level human action recognition, resulting in improved prediction performance. In skeleton-based action recognition, Lin *et al.* [28] suggested using multiple tasks such as motion prediction and jigsaw puzzle recognition to learn more general representations for better recognition performance.

Motivated by previous research in the human body motion domain [9], [50], we demonstrate that hand motion denoising and prediction are related tasks. By leveraging shared feature representations and learning a joint optimization, the model can better capture complex spatial-temporal patterns in the data, leading to better performance in both denoising and prediction tasks.

3 Multi-task Spatial-Temporal Graph Auto-Encoder (Multi-STGAE)

We propose Multi-STGAE, an efficient multi-task framework that enhances hand motion denoising and prediction by incorporating hand structural priors and temporal motion coherence. First, it removes noises from the input hand motion, facilitating a better experience of hand motion-based human-computer interaction. Second, it addresses the interaction delay problem and prevents dynamic information loss of the denoising result by using the prediction task to forecast future motion. We define the problem and provide an overview of the proposed framework in Sect. 3.1. The entire multi-task architecture is elaborated in Sect. 3.2, followed by a detailed description of the basic spatial-temporal graph convolution block in Sect. 3.3. Lastly, the loss function for training is described in Sect. 3.4.

3.1 Problem Definition and Framework Overview

Considering a historical sequence of raw hand motion data $\tilde{\mathbf{X}} \in \mathbb{R}^{T\times N\times 3}$ that consists of 3D coordinates with N joints within a temporal window of T frames, our multi-task framework Multi-STGAE aims to simultaneously recover the clean motion $\mathbf{X} \in \mathbb{R}^{T\times N\times 3}$ and predict the future motion $\mathbf{Y} \in \mathbb{R}^{\Delta T\times N\times 3}$. As a result, we can obtain the denoised motion $\hat{\mathbf{X}} \in \mathbb{R}^{T\times N\times 3}$ and the predicted motion $\hat{\mathbf{Y}} \in \mathbb{R}^{\Delta T\times N\times 3}$, which are supervised by the ground truth motions during the training phase. Practically, it is difficult to construct the corresponding clean motion from the collected raw motion data with noises for training purposes. In Sect. 3.2, we detail the network architecture of the proposed multi-task framework. It should be noted that both the encoder and the two decoders are stacked with several spatial-temporal graph convolution blocks (STGCB), which is elaborated in Sect. 3.3.

One core reason for the lack of research in hand motion denoising and prediction is the lack of benchmark datasets. To this end, we contribute two new large-scale datasets by corrupting the clean motion $\mathbf{X}$ to simulate the raw motion $\tilde{\mathbf{X}}$ with errors. The details of the data synthesis process will be described in Sect. 4.1.

The overview of the framework is shown in Fig. 2. (a) The corrupted motion data $\tilde{\mathbf{X}} = \{\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \cdots, \tilde{\boldsymbol{x}}_T\}$ is first fed into a shared motion encoder (SME) that projects the input to a compact and versatile latent space. By sharing a latent space, both denoising and prediction branches can remove noise while maintaining temporal dynamics by regularizing the latent representation space. Because the latent representation may contain independent information components between different tasks, a simple yet effective gating mechanism is used to prevent the latent representation from causing harm to others. (b) The denoising branch involves passing the shared latent variable through the denoising gating unit (DGU) and the motion denoising

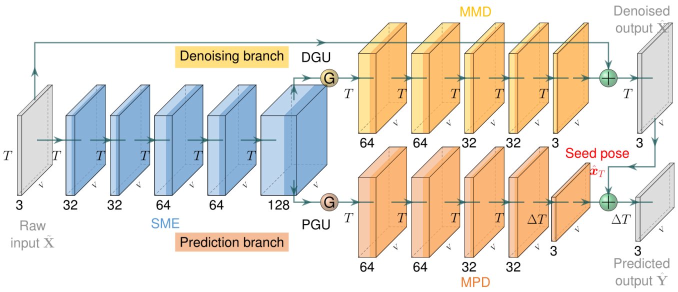

Fig. 3. The network architecture of the proposed multi-task framework.

decoder (MDD) in order to obtain the noise signal, and the denoising result $\hat{\mathbf{X}} = \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_T\}$ is finally obtained using a skip connection. (c) In the prediction branch, the shared latent variable is added to the positional encoding (PE) after passing through the prediction gating unit (PGU). By introducing the PE module, our method is injected with a positional signal that ensures the sequential property of the motion data as a time series.Then, the future motion offset is determined after passing through the motion prediction decoder (MPD). The predicted result $\hat{\mathbf{Y}} = \{\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \cdots, \hat{\boldsymbol{y}}_{\Delta T}\}$ is finally obtained by adding the seed pose $\hat{\boldsymbol{x}}_T$ from the denoised branch.

3.2 The Multi-Task Architecture

Both noises (*e.g.*, jitters) and temporal movement dynamics (*e.g.*, grasping) may correspond to the high-frequency component of the hand movement, making them difficult to be distinguished. Therefore, the key to motion data cleanup is to recognize noises from high-frequency signals. Existing work [8], [17], [59] may not preserve motion dynamics when removing noise as they do not take into account the possibility of losing dynamic information. An insufficient denoising intensity easily results in residual noise, whereas an excessive denoising intensity removes motion dynamics. In both cases, discontinuous denoising results may fail to confirm the users' intention, resulting in interaction failure.

Our main idea is to incorporate an auxiliary task to improve the denoising performance while preserving the temporal dynamics. Therefore, the choice of the auxiliary task is important for preserving dynamic information and minimizing the negative effects of the denoising process. We choose motion prediction for two reasons: firstly, historical motion data that is free of noise facilitates reliable predictions, and secondly, historical information that maintains dynamics allows producing continuous predictions. Simultaneously learning tasks of denoising noisy observations and predicting future dynamics in motion prediction produces stable and ahead-of-time motion data that better confirms users' intentions. This is in contrast to directly predicting future dynamics, which cannot achieve this goal. As a result, the proposed multi-task framework outperforms any single task. Fig. 3 shows the detailed architecture of our multi-task framework, constructing a shared latent space to enhance the overall performance of the task, in which the three core modules are explained below.

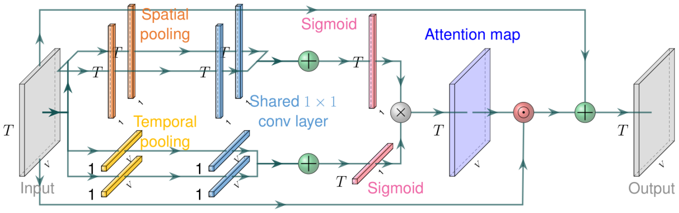

Fig. 4. The network architecture of a gating unit. It allows the network to control what information should be propagated through the layers.

3.2.1 The Shared Motion Encoder

As motion denoising and motion prediction are interconnected tasks, it is possible for them to share information for mutual benefit. By sharing a common encoder for extracting the spatial-temporal patterns, we regularize the latent space with both tasks, enabling the multi-task framework to outperform any individual task.

Through the SME module, we can obtain the latent representation as:

$$\mathbf{H} = \mathrm{enc}(\tilde{\mathbf{X}}), \tag{1}$$

where $\mathrm{enc}(\cdot)$ denotes the encoding operation. To increase the capacity of the network, the SME module stacks five STGCBs, whose structure will be explained in Sect. 3.3.2.

Since the original feature space has limited capacity, it is difficult to distinguish motion from noise. Motivated by previous STGCNs [3], [43], [52], which demonstrate that the hierarchical assimilation of elementary features into intricate structures is facilitated by augmenting channels in deeper layers, we adopt a similar architecture, which offers dual benefits by increasing channel dimensions. Firstly, it allows the allocation of a mere fraction of the new channels to noise, leaving a predominant segment for pertinent data, thereby sharpening the network's focus on significant information and restraining interference. Secondly, the weights of each dimension are learned through back-propagation from the task-related loss, thereby resulting in larger weights for useful information and smaller ones for noise. Consequently, in our design, the channels for the five respective blocks are designated as 64, 64, 128, 128, and 256.

3.2.2 The Denoising Branch

The denoising branch is used to separate the noise in the latent space from the original data so that the original signal can be reconstructed. Additionally, the shared latent space contains independent signals of the two tasks, which inhibits the performance of the other. We achieve this by utilizing a novel gating mechanism that controls the effective flow of information. As shown in Fig. 3, the denoising branch consists of a gate (DGU) and a decoder (MDD).

Denoising Gating Unit. The DGU module aims at providing effective information for motion reconstruction. Through the feature refinement process, noise and motion context information can be separated easily, and irrelevant features will be shielded from denoising. Fig. 4 shows its network architecture. It distinguishes motion context information from the latent space by selectively passing or filtering out information based on the learned gating values. The gating values are learned from the latent representation and allow the network to control the flow of information within the architecture. In both the spatial and temporal dimensions, the input tensor is first pooled with both the average and maximum methods. Following a 1×1 convolutional layer, the two pooling features are combined for two domains, respectively. Next, the attention map is obtained by multiplying the obtained activation features with a size that is consistent with the input tensor. Lastly, the input tensor is multiplied by the attention feature map before a skip connection is used to obtain the output feature. In the way, we can obtain:

$$\mathbf{H}_{\mathrm{d}} = g_{\mathrm{d}}(\mathbf{H}), \tag{2}$$

where $g_{\mathrm{d}}(\cdot)$ denotes the operation of the DGU module, and $\mathbf{H}_{\mathrm{d}}$ represents the refined feature for the denoising branch.

Motion Denoising Decoder. The MDD module aims at reconstructing the clean motion from the refined feature. As an inverse process of encoding, decoding is the process of converting data from the shared latent space into an output space. Rather than reconstructing the original signal directly, we learn an offset between the raw motion and its clean one, instead of the absolute values, to reduce the difficulty of the network. In this way, we obtain the denoised output:

$$\hat{\mathbf{X}} = \mathrm{dec}_{\mathrm{d}}(\mathbf{H}_{\mathrm{d}}) + \tilde{\mathbf{X}}, \tag{3}$$

where $\mathrm{dec}_{\mathrm{d}}(\cdot)$ denotes the decoding operation for the denoising branch. The MDD module stacks five STGCBs with the output channel numbers 128, 128, 64, 64, and 3.

3.2.3 The Prediction Branch

Since hand motion is inherently temporal, denoising hand motion data without considering its temporal characteristics may result in a loss of important information or even the introduction of new artifacts. Motion prediction regularizes the latent space by providing a temporal context for the denoising process, preventing dynamic information loss. Similar to the denoising branch, we also adopt a gating unit (PGU) to refine the latent representation, followed by a decoder (MPD) to predict future motion.

Prediction Gating Unit. The PGU module aims at distinguishing historical dynamics for predicting future motion from the latent space. Through the module, we obtain:

$$\mathbf{H}_{\mathrm{p}} = g_{\mathrm{p}}(\mathbf{H}), \tag{4}$$

where $g_{\mathrm{p}}(\cdot)$ denotes the operation of the PGU module, and $\mathbf{H}_{\mathrm{p}}$ represents the refined feature for the prediction branch.

Motion Prediction Decoder. The MPD module aims at forecasting future dynamics from the refined feature. Different from RNN-based methods [15], [44] that are based on previously predicted poses to forecast the next frame, we use TCN to forecast each frame independently due to its fully parallelized property. Although the core component of MPD integrates graph and temporal convolutions within its spatial-temporal block (detailed in Sect. 3.3), it does not innately comprehend the specific positions of data points. To provide our model with this critical positional information, especially vital when leveraging self-attention mechanisms, we use positional embedding [49] to map each frame number t to a vector and then inject it into each time step of the input features of MPD. Considering two indexes t_1 and t_2, the closer they are, the more similar their respective positional embedded features are. In this way, our non-autoregressive MPD clearly distinguishes the input context at different positions, thus explicitly ensuring the temporal continuity and the sequential relation of the generated sequence. By predicting offsets instead of absolute values, our model is more robust to variations in motion. This is because the predicted offsets capture relative motion between frames, which is often more consistent across different instances of a particular motion. Through the MPD module, we obtain the predicted output:

$$\hat{\mathbf{Y}}_t = \mathrm{dec}_{\mathrm{p}}\left(\mathbf{H}_{\mathrm{p}} + \mathbf{P}\right) + \hat{\boldsymbol{x}}_T, \tag{5}$$

where $\mathbf{P}$ is the positional embedding matrix, $\mathrm{dec}_{\mathrm{p}}(\cdot)$ denotes the decoding operation for the prediction branch, and $\hat{\boldsymbol{x}}_T$ is the last frame of the denoised output. Notably, the incorporation of the last frame into the prediction process can be viewed as a supervisory mechanism for denoising, thereby enhancing the denoising performance. While bypassing the forward step of MDD for prediction might offer efficiency, particularly in real-time contexts, our primary objective remained centered on achieving superior denoising. Finally, the MPD module generates the smooth prediction in parallel, in which each predicted frame is not affected by the previous. The MPD module stacks five STGCBs with the output channel numbers 128, 128, 64, 64, and 3.

3.3 Spatial-Temporal Graph Convolution

As a spatial-temporal time series, the hand skeleton sequence exhibits both spatial correlations among joints and temporal patterns among frames. For both denoising and prediction, it is essential to capture spatial-temporal patterns. The hand motion data is first constructed as a spatial-temporal graph. Then, we model the spatial-temporal relationships by stacking multiple spatial-temporal graph convolution blocks comprised of GCNs and TCNs.

3.3.1 Spatial-Temporal Graph Construction

The key idea behind our method is to combine the structural priors and the temporal coherence of hand motions to propose Multi-STGAE for denoising and predicting hand motions. By pre-defining physic-connected and symmetry-connected links between hand joints, the hand motion data is constructed as a spatial-temporal graph for spatial-temporal graph convolution.

We adapt previous efforts on skeletal action recognition [43], [52] for modelling hand motion data as an undirected spatial-temporal graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Unlike previous works focusing on human body data, we propose a hand-specific skeleton partition strategy based on the symmetry of the hand topology. It aids in denoising by using information from neighboring clean joints to compensate for noisy ones. Simultaneously, joints with less noise contribute to a more accurate prediction. This compensation relationship extends beyond directly connected joints, with an extra connection defined based on the compensation relationship between joints corresponding to different fingers of the hand. For example, a middle finger joint error can be compensated

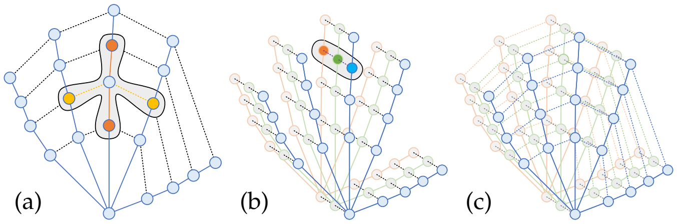

Fig. 5. Illustration of the spatial-temporal graph: (a) spatial connections, (b) temporal connections, (c) spatial and temporal connections.

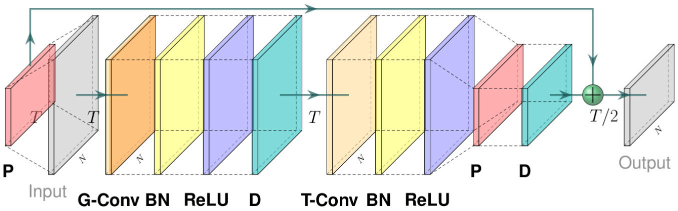

Fig. 6. The network architecture of the STGCB block. 'G-Conv', 'T-Conv', 'BN', 'P', and 'D' denote the graph convolution layer, the temporal convolution layer, the batch normalization layer, the pooling layer, and the dropout layer, respectively.

for by the corresponding counterparts of the index finger and ring finger. This is referred to as a symmetry-connected connection, and alongside the physical-connected connection, they both belong to spatial compensation relationships. Additionally, there is a temporal compensation relationship between corresponding joints over time.

A spatial-temporal graph comprises two basic elements: a spatial graph as shown in Fig. 5(a) and a temporal graph as shown in Fig. 5(b). In the spatial graph, the dependencies between neighboring joints are depicted based on the direct and indirect links between them. In the temporal graph, the continuity between consecutive frames of hand motion is represented. Generally, the initial node feature $\boldsymbol{v}_{t,i} \in \mathbb{R}^3$ with respect to the t-th frame and the i-th joint is comprised of 3D pose coordinates. The node set $\mathcal{V} = \{\boldsymbol{v}_{t,i} | t = 1, 2, \ldots, T, i = 1, 2, \cdots, N\}$ contains all joints in the entire hand motion sequence with T frames and N joints. The edge set $\mathcal{E} = \{\boldsymbol{v}_{t,i} \leftrightarrow \boldsymbol{v}_{t,j}, \boldsymbol{v}_{t-1,i} \leftrightarrow \boldsymbol{v}_{t} | i = 1, 2, \cdots, N, t = 2, 3, \cdots, T\}$ includes both direct and indirect links. Thus, 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 $\boldsymbol{v}_{t,i} \leftrightarrow \boldsymbol{v}_{t+1,i}$. In this way, a clean spatial-temporal neighbor can compensate for noisy joints through the use of this effective hand skeleton partition strategy.

3.3.2 *Spatial-Temporal Graph Convolution Blocks*

We present a two-stage network to implement the spatial-temporal graph convolution block (STGCB) as shown in Fig. 6, with the two stages responsible for the spatial convolution and the temporal convolution respectively. For the l-th spatial-temporal graph convolution block, the input $\mathbf{H}^{(l)} \in \mathbb{R}^{T \times N \times C_{\text{in}}}$ is processed by graph convolution to get the hidden representation $\mathbf{Z}^{(l)} \in \mathbb{R}^{T \times N \times C_{\text{mid}}}$, and then $\mathbf{Z}^{(l)}$ is processed by temporal convolution to get the output $\mathbf{H}^{(l+1)} \in \mathbb{R}^{T \times N \times C_{\text{out}}}$, which is used as the input of the next.

The STGCB module is designed to extract intrinsic features that are beneficial for motion denoising and prediction. On the one hand, proper identification of clean and noisy joints is crucial for determining the information compensation between joints. Misidentification can cause confusion in the intrinsic features, leading to unfavorable results in denoising and prediction. On the other hand, relying only on the pre-defined topology in Sect. 3.3.1 may disregard other joints that have potential compensation relationships, affecting the denoising and prediction performance. For example, when the thumb and middle fingertip touch during the pinch gesture, they have an information compensation relationship. To address these problems, we learn an adaptive graph topology using a self-attention mechanism.

Graph Convolution. We employ the graph convolution operation to capture the similarity of neighboring joints in space. As neighboring joints tend to have similar characteristics, this operation helps to preserve the natural structure of the hand. In the presence of noise, a corrupted joint can be compensated by information from its clean neighbors. Moreover, the use of graph convolution allows the prediction to be performed in a latent space that is highly representative of the hand's spatial characteristics, such as joint similarity and limits. This encourages the generation of realistic hand poses that are consistent with the hand topology, preventing unrealistic poses like overlapping fingers or unnatural joint angles.

As can be seen in Fig. 5(a), there are two types of spatial relationships between joints: direct (physic-connected) neighbors and indirect (symmetry-connected) ones. We divide the joint i and its direct and indirect neighbors into a subset $\mathcal{S}_i$. If $j \in \mathcal{S}_i$, then set A_{ij} to 1; if the joint j is the direct/indirect neighbor of i, then set $A^{ij}_{\text{direct}}/A^{ij}_{\text{indirect}}$ to 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 $\mathbf{A}_3 = \mathbf{A}_{\text{indirect}}$. In this way, our graph convolution can be represented as:

$$\mathbf{Z}^{(l)} = \sigma\left(\sum_{k=1}^{K} \left(\tilde{\mathbf{\Lambda}}_k^{-\frac{1}{2}} \mathbf{A}_k \tilde{\mathbf{\Lambda}}_k^{-\frac{1}{2}}\right) \mathbf{H}^{(l)} \mathbf{W}_k^{(l)}\right), \tag{6}$$

where $K = 3$, $\tilde{\Lambda}_k^{ii} = \sum_j A_k^{ij} + \epsilon$, and $\sigma(\cdot)$ denotes the RELU activation function. We set ϵ to a little positive number (*e.g.*, 0.001) to avoid the empty row of $\mathbf{A}_k$. The convolution kernel $\mathbf{W}_k^{(l)} \in \mathbb{R}^{C_{\text{in}} \times C_{\text{mid}}}$ is applied for the k-th kind of neighbors.

Human hand motion involves a complex interplay between multiple joints, and the movement between different joints is highly coordinated. Stacking multiple graph convolution layers can construct a latent space that defines what a natural hand pose would be, thus limiting the possible variation of output that a prediction or denoising network can produce. This makes the prediction or denoising task much easier than predicting individual joints without considering the context of neighboring joints. If the network is capable of leveraging the relevant context with the corrupted pose, it is of great benefit to recover the missing information and remove the noise. The main idea of the proposed method is to integrate trustworthy contributions of neighboring joints. This is achieved through the attention mechanism that learns dynamically which neighbors are clean or noisy.

In this work, we employ a self-attention mechanism to learn dynamic weights for joints, rather than using a learnable mask to determine the contribution of joints without defined links. This is because a learnable topology only learns a static weight for each hand pose, which ignores the differences in hand noise that can occur during motion. With self-attention, the network can dynamically learn which neighboring joints are clean or noisy and adjust the contribution of each joint accordingly, leading to more accurate denoising and prediction. Thus, we can obtain: :

$$\mathbf{Z}^{(l)} = \sigma\left(\sum_{k=1}^{K}\left(\tilde{\mathbf{A}}_k + \mathbf{B}_k^{(l)} + \mathbf{C}_k^{(l)}\right)\mathbf{H}^{(l)}\mathbf{W}_k^{(l)}\right), \tag{7}$$

where $\mathbf{B}_k^{(l)} \in \mathbb{R}^{N\times N}$ is a learnable matrix, ensuring compensation for noisy joints from potentially undefined nodes, and $\mathbf{C}_k^{(l)} \in \mathbb{R}^{N\times N}$ is an adaptive matrix, responsible for modulating the connection strength between neighboring nodes. The values in $\mathbf{C}_k^{(l)}$ are determined by leveraging the scaled dot-product attention mechanism as detailed in [49], which can be represented as:

$$\mathbf{C}_k^{(l)} = \mathrm{softmax}\left(\left(\mathbf{Q}^{(l)}\mathbf{K}^{(l)^\top}\right)/\sqrt{d}\right), \tag{8}$$

where $\mathbf{Q}^{(l)}, \mathbf{K}^{(l)} \in \mathbb{R}^{C_{\mathrm{in}}\times d}$ denote the query and the key embeddings respectively, and $\mathrm{softmax}(\cdot)$ indicates the softmax activation function. The scale factor $1/\sqrt{d}$ is used to prevent the dot products from growing too large in magnitude, which can lead to numerical instability during training. Both the query $\mathbf{Q}^{(l)} = \mathbf{H}^{(l)}\mathbf{W}_{\mathrm{query}}^{(l)}$ and the key $\mathbf{K}^{(l)} = \mathbf{H}^{(l)}\mathbf{W}_{\mathrm{key}}^{(l)}$ are the embedding of the latent representation $\mathbf{H}^{(l)}$, where $\mathbf{W}_{\mathrm{query}}^{(l)}, \mathbf{W}_{\mathrm{key}}^{(l)} \in \mathbb{R}^{C_{\mathrm{in}}\times d}$ are the corresponding embedding weights. The matrix $\mathbf{C}_k^{(l)}$ learns a unique connected topology for each hand pose $\mathbf{H}^{(l)}$ and measures the information transfer relationship between any two joints.

Temporal Convolution. We use temporal convolution to capture the temporal pattern of hand motion. To accommodate variable-length inputs due to different sampling rates or system constraints in real-world applications, we pad the sequence start, ensuring consistent output lengths without altering the network structure. We can capture both short-term and long-term motion trends by stacking multiple layers:

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathrm{Conv1D}(\mathbf{Z}^{(l)})\right), \tag{9}$$

where $\mathrm{Conv1D}(\cdot)$ denotes the temporal convolution, which is essentially a 1D convolution.

3.4 Loss Function

During the training phase, the network simultaneously recovers the denoised output $\hat{\mathbf{X}}$ and forecasts the future output $\hat{\mathbf{Y}}$ from the corrupted input $\tilde{\mathbf{X}}$. The entire network is trained end-to-end using a combined loss, which consists of a reconstruction loss and a prediction loss.

The reconstruction loss measures the difference between the denoised output and the clean ground truth, while the prediction loss measures the difference between the predicted future output and the future ground truth. The combined loss is defined as the weighted sum of the reconstruction loss and the prediction loss, which is:

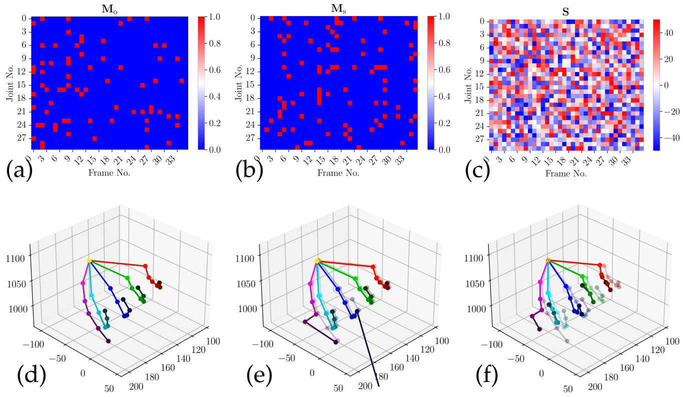

Fig. 7. Example plots of the data synthesis process for a hand motion: (a) $\mathbf{M}_{\mathrm{s}}$, (b) $\mathbf{M}_{\mathrm{o}}$, (c) $\mathbf{S}$, (d) $\mathbf{X}$, (e) $\mathbf{X}'$, and (f) $\tilde{\mathbf{X}}$.

$$\mathcal{L} = \lambda_1\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 + \lambda_2\|\mathbf{Y} - \hat{\mathbf{Y}}\|_2^2, \tag{10}$$

where λ_1 and λ_2 are balance factors, balancing the scale of each loss term, distinguishing the importance of two tasks, and ensuring that the two branches converge synchronously as much as possible to stabilize the training process.

This work introduces a two-step training strategy to better optimize the performance of our network. In the first step, we focus solely on denoising to generate a high-quality denoised output $\hat{\mathbf{X}}$, with the prediction branch disabled. The second step trains the entire two-branch network end-to-end. This approach allows the network to recover the denoised output $\hat{\mathbf{X}}$ and predict the future output $\hat{\mathbf{Y}}$ from the corrupted input $\tilde{\mathbf{X}}$. By first ensuring the denoising branch produces accurate results, the two-step training strategy enhances the overall performance of the prediction branch.

4 Experiments

A description of the experimental setup, including datasets, metrics, and implementation details, is presented first, followed by an analysis of the results.

4.1 Datasets

We have contributed to the field of hand motion denoising and prediction by introducing two new datasets. These datasets were constructed by applying our proposed data synthesis algorithm to existing benchmark datasets. In the following sections, we will describe the proposed data synthesis algorithm and provide an overview of the two synthesized datasets.

Hand Motion Data Synthesis. We use Algorithm 1 to generate synthetic data. The proposed algorithm accepts as input the original hand motion data in the form of a tensor, along with three hyper-parameters: σ_{o}, σ_{s}, and β. The core functionality of the algorithm involves a series of steps, beginning with data normalization, followed by the generation of shifting and occlusion probabilities.

These probabilities are then leveraged to apply Bernoulli distributions for mask generation, with the resultant masks utilized to shift and occlude the original data. The final stage involves the optimization of the corrupted data, with a focus on meeting certain structural constraints. Ultimately, the output of the algorithm is the corrupted hand joint data, presented in tensor format.

Algorithm 1: Motion data corruption algorithm

```
Input: Hand motion 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
$\mathbf{M}_\mathrm{s}^{T\times N} \leftarrow \mathrm{Bernoulli}(\min(|\boldsymbol{\alpha}_\mathrm{s}|, 2\sigma_\mathrm{s}))$ ;  // Shifting 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
$\mathbf{M}_\mathrm{o}^{T\times N} \leftarrow \mathrm{Bernoulli}(\min(|\boldsymbol{\alpha}_\mathrm{o}|, 2\sigma_\mathrm{o}))$ ; // Occlusion mask
$\mathbf{X}'^{T\times N\times 3} \leftarrow \mathbf{X}_\mathrm{s} \odot (1 - \mathbf{M}_\mathrm{o})$ ;          // Data occlusion
$\tilde{\mathbf{X}}^{T\times N\times 3} \leftarrow \mathrm{Optimize}(\mathbf{X}`)$ ; // Structural constraint
```

To visualize the data synthesis process, we provide example plots of a hand motion in Fig. 7. Specifically, Figs. 7(a) to 7(c) depict the heatmaps of matrices $\mathbf{M}$s, $\mathbf{M}$o, and $\mathbf{S}$, respectively. Additionally, Figs. 7(d) to 7(f) illustrate frames of the original motion $\mathbf{X}$, inter-corrupted motion $\mathbf{X}'$, and the corrupted motion $\tilde{\mathbf{X}}$, with the full motion available in our supplementary video. Initially, the hand motion data $\mathbf{X}$ (shown in Fig. 7(d)) is normalized to have a uniform length across all motions using $\mathrm{Normalize}(\cdot)$. To simulate joint deviations, we add shifting noise $\mathbf{S} \in \mathbb{R}^{T\times N\times 3}$ (shown in Fig. 7(c)) following a uniform distribution to the clean data. A binomial distribution is used to sample the shifting mask $\mathbf{M}_\mathrm{s} \in \mathbb{R}^{T\times N}$ (shown in Fig. 7(b)) to control the distribution of shifting noise in data space. To simulate occlusion scenarios, we generate an occlusion mask $\mathbf{M}_\mathrm{o} \in \mathbb{R}^{T\times N}$ (shown in Fig. 7(a)) that sets certain hand joints to zero. Thus, the noisy data $\mathbf{X}'$ (shown in Fig. 7(e)) can be obtained.

In contrast to prior work [17], we impose specific structural constraints on the corrupted operation, as illustrated in Fig. 8. We assume that the root joint and palm joints are relatively stable, and thus we start to optimize from a position two hops away from the root node. As shown in Fig. 8(a), adding noise to each joint separately can cause the third and fourth joints ($\boldsymbol{x}_3'$ and $\boldsymbol{x}_4'$) to deviate from their original position ($\boldsymbol{x}_3$ and $\boldsymbol{x}_4$) and disregard the hinge structure of the hand, resulting in poor generalization. Therefore, we constrain the distance between the child node and the parent node to generate more realistic noisy motions, as shown in Fig. 8(b). We optimize the corrupted data by starting from a node close to the root and moving toward nodes farther away. The child nodes inherit the shift correction from their parent node and then correct themselves, which avoids generating extremely unreasonable noise and facilitates the learning of complex spatial-temporal patterns by the network. In this way, this optimization process leads to better generalization. Finally, we synthesize the corrupted data $\tilde{\mathbf{X}} \in \mathbb{R}^{T\times N\times 3}$ with respect to its ground truth $\mathbf{X} \in \mathbb{R}^{T\times N\times 3}$ (in Fig. 7(f)) using the optimization function $\mathrm{Optimize}(\cdot)$. The paired data is used to train our network.

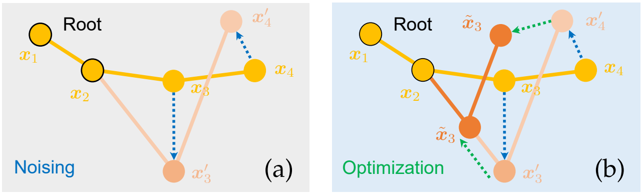

Fig. 8. Illustrations of (a) noising and (b) optimization.

The NYU Dataset. The first dataset, NYU, is derived from a hand pose estimation dataset [48]. We sample T frames at $\lceil T/2 \rceil$ frame intervals from each sequence, which are then organized into clips and serve as the ground truth. Our data synthesis algorithm is applied to corrupt the ground truth and generate the corresponding input data. We set $T = 36$ frames for each clip, consistent with our previous work [59]. The resulting clips form the basis of our training and testing datasets, which we further divide into separate sets with a test rate of $\delta = 0.15$. We obtained a total of 10,299 clips for the training set and 1,818 clips for the testing set.

The SHREC Dataset. The second dataset, SHREC, is constructed from a challenging hand gesture recognition dataset [11]. This dataset contains 14 gestures, including five types of fine gestures that involve hand shape changes and coarse gestures that involve hand movement. We follow the same pre-processing steps as for the NYU dataset, resulting in a total of 6,482 clips for the training set and 1,144 clips for the testing set. This dataset is different from NYU in that it contains occlusions, dislocations, and high-frequency noise, and has a relatively small amount of data, which provides a more comprehensive evaluation of the effectiveness and robustness of our proposed algorithm.

4.2 Metrics

Previous studies [3], [17] have employed pose or rotation errors to assess the algorithm's performance. However, these metrics are inadequate to evaluate the algorithm's ability to preserve the structural constraint of the data. To address this issue, we propose two additional metrics: the mean bone length error and the mean symmetry error, in addition to the mean pose error.

Mean Pose Error. It measures the hand pose difference between the output $\hat{\mathbf{X}}$ and the ground truth $\mathbf{X}$, which is:

$$E_\mathrm{pos} = \mathrm{MSE}(\mathbf{X}, \hat{\mathbf{X}}). \tag{11}$$

Mean Bone Length Error. It measures the error between the bone length of the output $\hat{\mathbf{X}}$ and that of the ground truth $\mathbf{X}$, which is:

$$E_\mathrm{bon} = \mathrm{MSE}(\phi(\mathbf{X}), \phi(\hat{\mathbf{X}})), \tag{12}$$

where $\phi(\cdot)$ calculates the bone length of any two physically connected joints.

Mean Symmetry Error. It measures the symmetry error of two symmetry-connected joints between the relative distance of the output $\hat{\mathbf{X}}$ and that of the ground truth $\mathbf{X}$, which can be represented as:

$$E_{\mathrm{sys}} = \mathrm{MSE}(\psi(\mathbf{X}), \psi(\hat{\mathbf{X}})), \tag{13}$$

where $\psi(\cdot)$ calculates the relative distance of any two symmetry-connected joints.

4.3 Implementation Details

In this work, we have implemented our method and the state-of-the-art methods using the Tensorflow 2 framework. All experiments have been conducted on a single GeForce RTX 3090 GPU with CUDA 11.3.

During the data generation phase, the parameters σ_{o}, σ_{s}, and β are used to regulate joint occlusions, swaps, and noise, respectively, with values of 0.1, 0.1, and 50mm, respectively. A training and testing batch size of 64 is employed to boost the training and inference. To mitigate over-fitting during model training, we employ a piecewise learning rate decay strategy. We use the Adam optimizer with an initial learning rate of 0.1. This learning rate is then reduced by a factor of 0.1 during both the 75th and 90th epochs. The training process is capped at a maximum of 100 epochs.

4.4 Results and Analysis

Firstly, we compare our method with the state-of-the-art. Then, we present the ablation study. Finally, a large amount of quantitative and qualitative experiments are also shown to verify the effectiveness of our method.

4.4.1 Comparison with the State-of-the-Art

To evaluate the performance of the proposed method, the authors compared it with several state-of-the-art methods on two large-scale datasets. The comparison includes joint-space encoder-bidirectional-filter network (EBF) [36], joint-space convolution neural network (CNN) [18], optical motion residual neural network (ResNet) [17], hand tremor compensation module based on graph neural network (CAM-GNN) [23] and our previous work [59], respectively. To ensure a fair comparison, all methods were implemented with the same number of layers and roughly the same memory allowance, with the number of hidden units adjusted accordingly. The performance of each method is evaluated using various metrics.

TABLE 1
Comparisons on the NYU dataset with state-of-the-arts (mm^2): Multi-STGAE$^+$ represents a version of the model with fewer channels compared to Multi-STGAE.

Method	E_{pos}	E_{bon}	E_{sym}
EBF [36]	70.7740	13.5822	69.2183
CNN [18]	170.3657	23.5246	390.9772
ResNet [17]	59.8223	6.5408	89.1416
CAM-GNN [23]	17.6803	3.5570	32.7914
STGAE	2.1741	0.5640	3.8091
Multi-STGAE$^+$	0.9565	0.1291	1.1442
Multi-STGAE	**0.8043**	**0.0904**	**0.8736**

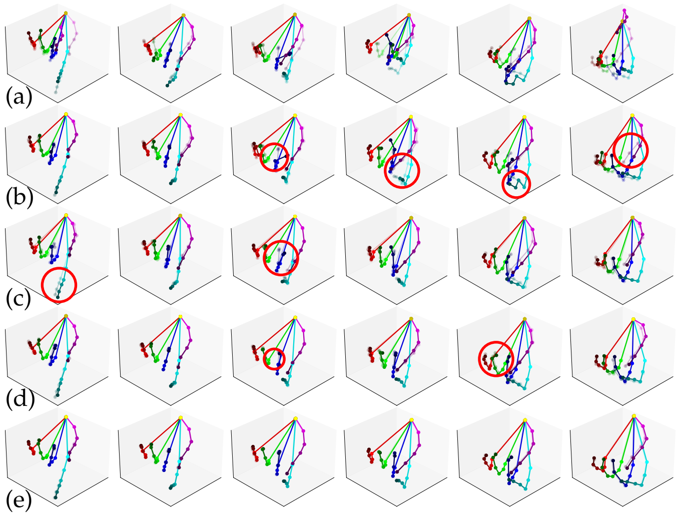

Fig. 9. Qualitative comparisons on the NYU dataset with state-of-the-arts: (a) input, (b) ResNet [17], (c) CAM-GNN [23], (d) STGAE [59], and (e) Multi-STGAE (Ours).

Quantitative Results. Tables 1 and 2 report the comparison results on the NYU dataset and the SHREC dataset, respectively. Compared with these state-of-the-art methods, our model performs best under all three metrics on both datasets. Some methods [17], [18], [36] are designed for denoising human body motion data, and migrating these methods directly to hand motion data performs poorly. This is because the hand has a more complex hinge structure than the human body and the relative magnitude of the noise is greater, making it more susceptible to noise. The proposed method is superior to the method [23] which is designed for hand motion denoising. The reason behind this is that the proposed method can capture both spatial and temporal patterns simultaneously, which helps it to achieve better results. Additionally, our method considers the structural constraints of the hand motion, resulting in significantly better performance in terms of bone length and symmetry errors. By introducing the auxiliary task, the denoising task has been further enhanced. It can simultaneously predict future hand dynamics, which is crucial in AR scenes with real-time hand-object manipulation. To a certain extent, performing motion prediction also alleviates the impact of processing and transmission delays.

TABLE 2
Comparisons on the SHREC dataset with state-of-the-arts (mm^2).

Method	E_{pos}	E_{bon}	E_{sym}
EBF [36]	38.7842	74.2793	66.9073
CNN [18]	6.2059	7.8206	11.1006
ResNet [17]	24.0110	35.6497	44.3853
CAM-GNN [23]	12.7578	10.9209	24.6467
STGAE	7.1960	10.3245	12.1954
Multi-STGAE	**1.9973**	**0.5989**	**1.5023**

Qualitative Results. We visualize several frames of two hand motions on the NYU dataset and the SHREC dataset. Figs. 9 and 10 show the visualization plots of the corresponding denoising results. Figs. 9(a) to 9(e) show results of four methods: ResNet [17], CAM-GNN [23], STGAE [59], and ours. In each subfigure, the ground truth is indicated

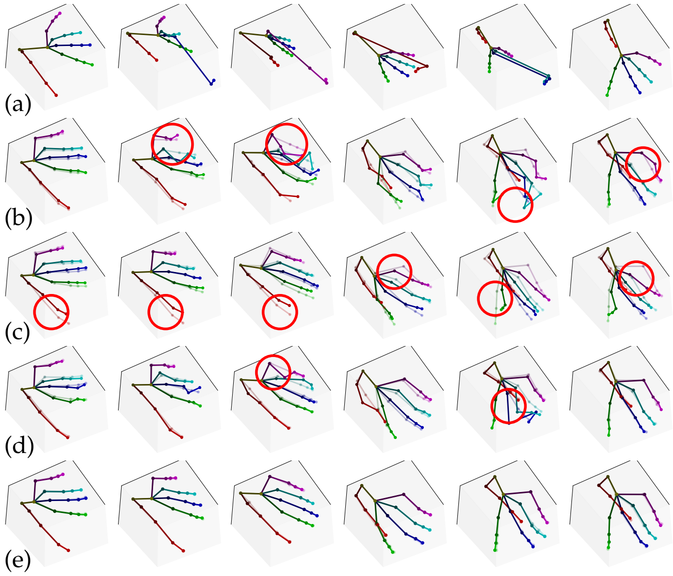

Fig. 10. Qualitative comparisons on the SHREC dataset with state-of-the-arts: (a) input, (b) ResNet [17], (c) CAM-GNN [23], (d) STGAE [59], and (e) Multi-STGAE (Ours).

by light shading whereas the normal shading represents the denoised output. For complete motion, we have provided a video for readers in the supplementary material. We mark the parts with large errors with red circles to make it easier to visualize the visual difference. As can be seen from Fig. 9, the input raw data in Fig. 9(a) contains severe noise, but these methods have shown good performance. Nevertheless, our method is more effective in some areas of detail. For example, CAM-GNN produces poor estimation results for the tip of the index finger in the first frame due to the high degree of freedom of the fingertip. Similarly, Fig. 10 shows that other algorithms exhibit varying degrees of denoising failure on the SHREC dataset. Remarkably, our proposed algorithm demonstrates exceptional performance on this challenging dataset, which contains severe occlusions, dislocations, high-frequency noise, and limited data, as described in Sect. 4.1. This finding highlights the robustness of our algorithm in handling such severe drawbacks.

Computational Overhead. Fig. 11 shows the bubble plot with respect to the model size and the computational overhead of our model and baselines. The horizontal axis represents the number of training parameters, and the vertical axis represents the amount of calculation. The smaller the bubble, the smaller the error.

The result in Fig. 11 shows that our method not only achieves the best performance but also requires a smaller model size and lower computational complexity. Multi-STGAE with 0.64M parameters and 392.55 GFLOPs outperforms STGAE with 0.20M parameters and 97.41 GFLOPs while accumulating relatively low computational overhead, as shown in Table 1. Moreover, by reducing the number of channels, Multi-STGAE$^+$ with 0.22M parameters and 126.25 GFLOPs can save a significant number of parameters and calculations with a little denoising performance impact. Unlike the EBF method, our approach, with its network designed to meet real-time requirements, can immediately process input without the necessity to wait for 15 frames, leading to enhanced performance with minimal computational overhead. Additionally, our model is much more lightweight for practical use due to the effective parameter-sharing mechanism. Compared with CAM-GNN, our method fuses spatial and temporal features with fewer trainable parameters, which accelerates model convergence and achieves better performance. It is worth noting that our method satisfies real-time processing demands during inference, achieving a frame rate of 25.7 fps for both denoising and prediction tasks. For specific real-world denoising applications where prediction is unnecessary, our method can reach a higher frame rate of 32.0 fps.

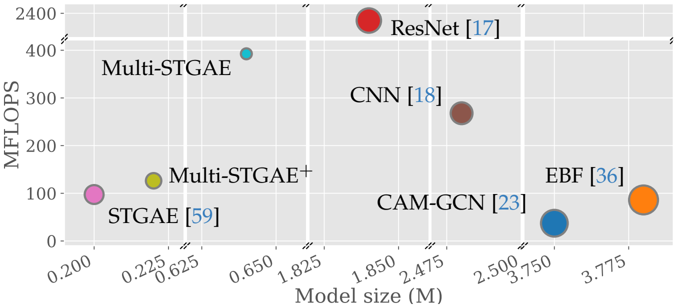

Fig. 11. The computation bubble plot on the NYU dataset.

4.4.2 Ablation Study

Through our ablation study, we investigate the effectiveness of the key components in our proposed method. To ensure a comprehensive evaluation, we conduct experiments on both the NYU and SHREC datasets.

Effectiveness of Different Attention Mechanisms. We first investigate the effectiveness of different attention mechanisms in our proposed method. These experiments are conducted solely within our denoising framework. Specifically, we compare the baseline model with two attention variants: one that multiplies a mask and another that adds learnable parts in Eq. (7). For additive attention, we separately explore the effectiveness of different parts. Table 3 reports the corresponding results on both the NYU and SHREC datasets, where $\mathcal{D}$ denotes the symbol of datasets.

TABLE 3
Results on the effectiveness of different attention mechanisms (mm^2).

$\mathcal{D}$	Setting		E_{pos}	E_{bon}	E_{sym}
NYU	Base	$\tilde{\mathbf{A}}$	15.83	4.60	26.23
	Mask	$\tilde{\mathbf{A}} \odot \mathbf{M}$	13.70 $^{\downarrow 2.13}$	4.02 $^{\downarrow 0.58}$	22.70 $^{\downarrow 3.53}$
	Add	$\mathbf{B} + \mathbf{C}$	13.69 $^{\downarrow 2.14}$	4.76 $^{\uparrow 0.16}$	23.81 $^{\downarrow 2.42}$
		$\tilde{\mathbf{A}} + \mathbf{C}$	13.15 $^{\downarrow 2.68}$	4.31 $^{\downarrow 0.29}$	20.69 $^{\downarrow 5.54}$
		$\tilde{\mathbf{A}} + \mathbf{B}$	14.43 $^{\downarrow 1.40}$	3.47 $^{\downarrow 1.13}$	30.03 $^{\uparrow 3.80}$
		$\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C}$	2.17 $^{\downarrow 13.66}$	0.56 $^{\downarrow 4.04}$	3.81 $^{\downarrow 22.42}$
SHREC	Base	$\tilde{\mathbf{A}}$	18.96	39.98	44.82
	Mask	$\tilde{\mathbf{A}} \odot \mathbf{M}$	16.94 $^{\downarrow 2.02}$	25.57 $^{\downarrow 14.41}$	28.41 $^{\downarrow 16.41}$
	Add	$\mathbf{B} + \mathbf{C}$	16.83 $^{\downarrow 2.13}$	30.13 $^{\downarrow 9.85}$	32.16 $^{\downarrow 12.66}$
		$\tilde{\mathbf{A}} + \mathbf{C}$	17.44 $^{\downarrow 1.52}$	32.26 $^{\downarrow 7.72}$	35.55 $^{\downarrow 9.27}$
		$\tilde{\mathbf{A}} + \mathbf{B}$	15.52 $^{\downarrow 3.44}$	21.94 $^{\downarrow 18.04}$	23.42 $^{\downarrow 21.40}$
		$\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C}$	7.20 $^{\downarrow 11.76}$	10.32 $^{\downarrow 29.66}$	12.20 $^{\downarrow 32.62}$

In Table 3, the results showcase the effectiveness of attention mechanisms in our proposed method across both two datasets. For example, the baseline model without any attention mechanism achieves a pose error of 15.83 on the NYU dataset. In contrast, the two different kinds of attention mechanisms outperform this baseline, indicating that using attention mechanisms is beneficial for denoising. Specifically, attention mechanisms help the GCN become less dependent on the pre-defined graph structure, leading to stronger generalization performance.

In terms of the two attention mechanisms, it can be seen in Table 3 that the additive attention mechanism performs better than the mask attention one. Learning neighbor importance by multiplying masks ($\tilde{\mathbf{A}} \odot \mathbf{M}$) maintains the prior graph topology $\tilde{\mathbf{A}}$, whereas adding the learnable component $\mathbf{B}$ and the adaptive component $\mathbf{C}$ ensures that the graph structure can be fully and properly adjusted. This indicates that a fixed topology lacks adaptability to dynamic systems, potentially leading to challenges in handling noisy data or evolving relationships. Among the different variants of additive attention, ($\tilde{\mathbf{A}}+\mathbf{B}+\mathbf{C}$) performs the best, indicating the effectiveness of all the learnable components.

To further explore the effectiveness of different parts of ($\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C}$), we conduct ablation studies by separately deleting each part, as shown in Table 3. Deleting the pre-defined adjacent matrix $\tilde{\mathbf{A}}$ causes the graph to lose the topological prior, making it difficult for the network to converge compared with ($\tilde{\mathbf{A}} + \mathbf{B} + \mathbf{C}$). Both deleting $\mathbf{B}$ and $\mathbf{C}$ are not conducive to dynamically learning which neighbors are trustworthy. This can be primarily attributed to the adaptive matrix $\mathbf{B}$, which allows the network to modulate the degree of denoising contribution. Concurrently, the learnable matrix $\mathbf{C}$ endows the network with enhanced flexibility, enabling it to discern potential dependencies without being constrained by prior limitations. Based on the results of our experiments, we conclude that both the prior topology and the learned topology are essential for improving denoising performance, which is a general conclusion that holds true in other fields as well [43].

In comparing the results across different datasets from Table 3, the pose error on the SHREC dataset is consistently larger than on the NYU dataset. This discrepancy can be attributed to the distinct joint representations in each dataset. Specifically, the NYU dataset employs a 36-joint representation (in Fig. 9), whereas the SHREC dataset utilizes a 22-joint configuration (in Fig. 10). For the same hand poses, the model with a greater number of joints tends to produce more refined and thus less erroneous results. Moreover, in the NYU dataset, the bone length error is significantly lower than the symmetry error, while in the SHREC dataset, these errors are comparable. This is due to the NYU hand model's denser joint arrangement, which results in physically connected joints being closer together compared to their symmetrical counterparts.

Effectiveness of Different Partition Strategies. Different from other methods [3], [43], this work proposes a simple yet effective partition strategy where the indirect symmetric connections also serve as the edges of the spatial-temporal graph. To examine the effectiveness of different types of connections on the performance of the proposed model, we conduct experiments on our denoising framework by removing one type of connection at a time, *i.e.*, self-connections $\mathbf{A}_1$, physic-connections $\mathbf{A}_2$, and symmetry-connections $\mathbf{A}_3$. Table 4 presents the corresponding results on both the NYU and SHREC datasets.

TABLE 4
Results on the effectiveness of different partition strategies (mm^2).

$\mathcal{D}$	Setting			E_{pos}	E_{bon}	E_{sym}
	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$			
NYU	✓	✓	✓	2.17	0.56	3.81
	✓	✓	✗	9.12 $^{\uparrow 6.95}$	2.32 $^{\uparrow 1.76}$	16.98 $^{\uparrow 13.17}$
	✓	✗	✓	11.20 $^{\uparrow 9.03}$	2.60 $^{\uparrow 2.04}$	21.10 $^{\uparrow 17.29}$
	✗	✓	✓	14.05 $^{\uparrow 11.88}$	6.42 $^{\uparrow 5.86}$	26.10 $^{\uparrow 22.29}$
SHREC	✓	✓	✓	7.20	10.32	12.20
	✓	✓	✗	16.13 $^{\uparrow 8.93}$	29.83 $^{\uparrow 19.51}$	32.84 $^{\uparrow 20.64}$
	✓	✗	✓	15.32 $^{\uparrow 8.12}$	27.13 $^{\uparrow 16.81}$	30.35 $^{\uparrow 18.15}$
	✗	✓	✓	16.95 $^{\uparrow 9.75}$	31.27 $^{\uparrow 20.95}$	34.58 $^{\uparrow 22.38}$

From Table 4, it is evident that removing any of these connections greatly reduces the error of the proposed model compared to the baseline that consists of all three connections. 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 $\mathbf{A}_2$ is also very large. A solid structural constraint is evident between physic-connected joints. The effect on indirect connections $\mathbf{A}_3$ is minimal, indicating that symmetrical neighbors are the least important compared with self-connections and direct connections. Despite this, removing symmetry-connected relationships results in significant performance degradation, indicating the effectiveness of the proposed partition strategy. It is important to note that these conclusions might be applied to other fields, such as action recognition, while the previous work has not been explored.

Effectiveness of the Multi-task Framework. To avoid the over-smoothing problem caused by the denoising process, we introduce a prediction task to improve denoising performance. To verify the relevance of the two tasks, we separately analyze the results of motion denoising and prediction when one of them is removed from our multi-task framework. Also, we evaluate the effectiveness of the key components. Table 5 shows the corresponding results on both the NYU dataset and SHREC datasets, where E^{D}, E^{P}, and E^{avg} denote the evaluation metric of the motion denoising, the motion prediction, and the multi-task framework, respectively. For the multi-task framework, we calculate the individual errors (both E^{D} and E^{P}) for each frame and then average these errors across all frames to obtain E^{avg}. This approach provides a more comprehensive understanding of the error distribution over the entire sequence, rather than just a direct sum of two error metrics.

In our single-task configurations, for the denoising network (without prediction), the model is simplified by omitting the prediction branch, concentrating solely on denoising. Conversely, the prediction network (without denoising) aligns with the multi-task setting by predicting the relative offsets, thus allowing for a direct comparison to estimate the effect of denoising on prediction performance. It can be seen from Table 5 that considering these two branches jointly

TABLE 5
Results on the effectiveness of the multi-task framework (mm^2).

$\mathcal{D}$	Setting	$E^{\mathrm{D}}_{\mathrm{pos}}$	$E^{\mathrm{D}}_{\mathrm{bon}}$	$E^{\mathrm{D}}_{\mathrm{sym}}$	$E^{\mathrm{P}}_{\mathrm{pos}}$	$E^{\mathrm{P}}_{\mathrm{bon}}$	$E^{\mathrm{P}}_{\mathrm{sym}}$	$E^{\mathrm{avg}}_{\mathrm{pos}}$	$E^{\mathrm{avg}}_{\mathrm{bon}}$	$E^{\mathrm{avg}}_{\mathrm{sym}}$
NYU	Multi-STGAE	0.80	0.11	1.10	26.13	0.90	17.27	3.89	0.19	2.87
	w/o gate	1.20 $^{\uparrow 0.40}$	0.18 $^{\uparrow 0.07}$	1.60 $^{\uparrow 0.50}$	35.92 $^{\uparrow 9.79}$	2.55 $^{\uparrow 1.65}$	18.35 $^{\uparrow 1.08}$	4.44 $^{\uparrow 0.55}$	0.36 $^{\uparrow 0.17}$	3.56 $^{\uparrow 0.69}$
	w/o denoising	-	-	-	76.67 $^{\uparrow 50.54}$	3.96 $^{\uparrow 3.06}$	21.12 $^{\uparrow 3.85}$	-	-	-
	w/o prediction	1.47 $^{\uparrow 0.67}$	0.22 $^{\uparrow 0.11}$	1.83 $^{\uparrow 0.73}$	-	-	-	-	-	-
	Multi-STGAE*	0.92 $^{\uparrow 0.12}$	0.11 $^{\uparrow 0.00}$	1.05 $^{\downarrow 0.05}$	27.53 $^{\uparrow 1.40}$	0.95 $^{\uparrow 0.05}$	17.98 $^{\uparrow 0.71}$	4.16 $^{\uparrow 0.27}$	0.21 $^{\uparrow 0.02}$	3.12 $^{\uparrow 0.25}$
SHREC	Multi-STGAE	2.00	0.60	1.50	80.01	10.49	51.92	11.52	1.81	7.65
	w/o gate	2.54 $^{\uparrow 0.54}$	0.82 $^{\uparrow 0.22}$	1.90 $^{\uparrow 0.40}$	92.48 $^{\uparrow 12.47}$	13.21 $^{\uparrow 2.72}$	53.23 $^{\uparrow 1.31}$	13.51 $^{\uparrow 1.99}$	2.33 $^{\uparrow 0.52}$	9.04 $^{\uparrow 1.39}$
	w/o denoising	-	-	-	102.47 $^{\uparrow 22.46}$	21.08 $^{\uparrow 10.59}$	75.61 $^{\uparrow 23.69}$	-	-	-
	w/o prediction	3.42 $^{\uparrow 1.42}$	0.64 $^{\uparrow 0.04}$	1.52 $^{\uparrow 0.02}$	-	-	-	-	-	-
	Multi-STGAE*	2.23 $^{\uparrow 0.23}$	0.71 $^{\uparrow 0.11}$	1.74 $^{\uparrow 0.24}$	93.99 $^{\uparrow 13.98}$	11.93 $^{\uparrow 1.44}$	57.60 $^{\uparrow 5.68}$	13.42 $^{\uparrow 1.90}$	2.98 $^{\uparrow 1.17}$	8.55 $^{\uparrow 0.90}$

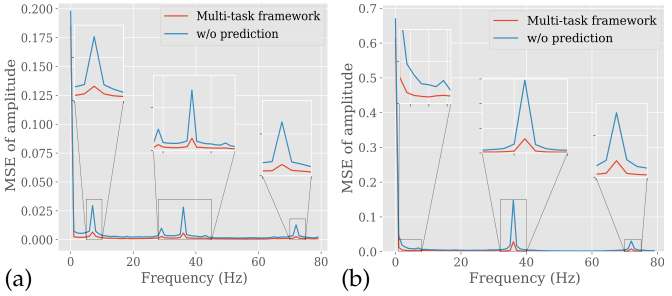

Fig. 12. The amplitude MSE plots of spectrum analysis across the entire test set for (a) the NYU dataset and (b) the SHREC dataset.

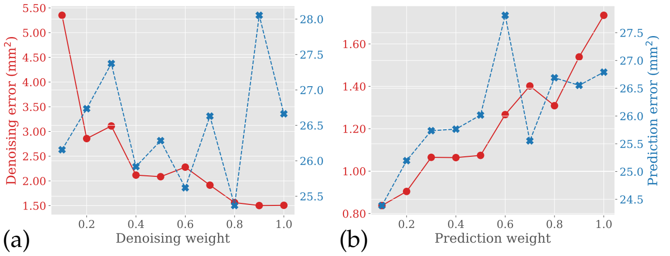

Fig. 13. Plots of the pose error curves with respect to (a) the denoising and (b) the prediction loss weights.

achieves better results than the single one. To determine whether incorporating a prediction task can mitigate the tendencies towards over-smoothing, we employ the Fourier transformation [60] to transform both the denoising output and the ground truth of the entire test set into the frequency domain. Subsequently, we compute the MSE between their frequency spectrums. An analysis comparing the error in a multi-task framework to that in a single-task framework (w/o prediction) is presented in Fig. 12. The results suggest that the introduction of a prediction task not only enhances performance in the high-frequency domain (e.g., about 70 to 75 Hz in Fig. 12(a)) but also manifests promising results in the low-frequency domain (e.g., about 5 to 10 Hz in Fig. 12(a)). This underscores the potential of incorporating a prediction task to preserve intricate details and effectively alleviate over-smoothing tendencies. The results in Table 5 also indicate that motion denoising can significantly enhance the prediction process. Therefore, when forecasting future motion, it is important to denoise historical dynamic information to capture the information accurately.

Previous works [29], [46] have shown that the multi-task framework may suffer from negative transfer due to task conflicts as parameters are shared between tasks. To address this issue, we utilize the gate mechanism. Different from the PLE model [46], we propose a more efficient method that only requires two gates. On the one hand, as can be seen in Table 5, eliminating the gate mechanism reduces the performance of the multi-task framework. This demonstrates that the gate mechanism can mitigate conflicting information between tasks to a certain extent and improve the performance of a single task to a certain extent. On the other hand, we also implement the PLE model (Multi-STGAE*). It can be seen from the last row of Table 5 that our model has achieved better performance. Furthermore, our model encoder requires 2.28M parameters, whereas the PLE model requires 8.19M parameters. The efficiency and effectiveness of our model can therefore be verified.

Furthermore, we further explore the effectiveness of different loss weights between the two tasks in terms of influence. Specifically, we adjust the loss weights of motion denoising and motion prediction, respectively, whereas the other weight is set to 1. The corresponding results on the NYU dataset are shown in Fig. 13. Based on the error magnitudes, it can be concluded that the training difficulty of the two tasks is different. This is in accordance with our common sense that prediction tasks are more difficult than denoising tasks. Consequently, it is evident from Fig. 13 that as the prediction weight increases or the denoising weight decreases, the total error will increase. For example, when the emphasis on training shifts toward denoising due to a reduced prediction weight in Fig. 13(b), it enhances the prediction performance. This further confirms that the denoising task may serve as an auxiliary facilitator for the prediction process. Furthermore, a notable observation in our experiment is that when the denoising weight is set at a magnitude 20 times that of the prediction, the model delivers its optimal performance. Interestingly, this ratio corresponds closely to the observed difference in error magnitude. Therefore, we set this ratio for other experiments.

Others. We also explore the effect of different parameter settings on the performance of our method. There are two factors involved, including the strength of the noise and the length of the prediction. Fig. 14 shows the corresponding results on the NYU dataset.

As shown in Fig. 14(a), the overall performance of the

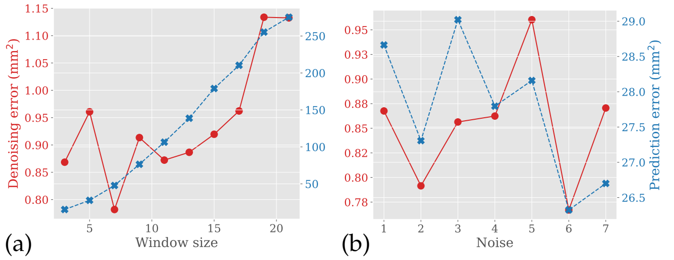

Fig. 14. Plots of the pose error curves with respect to (a) the window size and (b) the noise magnitude.

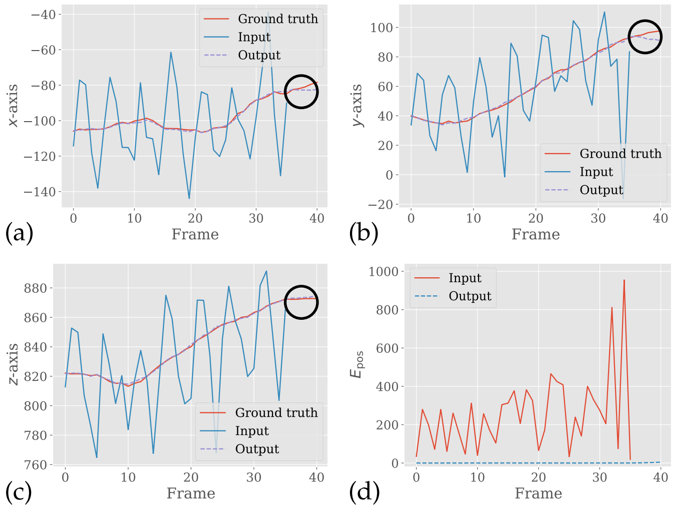

Fig. 15. Plots of a motion trajectory and the corresponding pose error curve using our method: (a), (b), and (c) are the motion trajectory of the index fingertip on the x-axis, y-axis, and z-axis (mm), respectively. (d) is the pose error (mm^2) curve of the complete motion between the noisy input and the denoised output.

method decreases as the window size length increases. It should be noted that the performance of denoising is not very variable, This is primarily due to the difference in error levels between the two tasks. As can be seen from Fig. 14(b), the overall performance of the model tends to be stable with increasing noise amplitude, which supports the robustness.

4.4.3 Visualization

For a more comprehensive understanding of our method's performance, we visualize more quantitative and qualitative results using the NYU dataset.

Motion Trajectory. Motion trajectories provide an intuitive representation of the performance in denoising and predicting. We depict the trajectory of the fingertip joint of the index finger to provide a better understanding of the trajectory. Since the fingertip joint has a greater degree of freedom than the root joint, it is more difficult to predict and recover its posture. Fig. 15 shows the results.

Figs. 15(a) to 15(c) show its trajectory of one hand with respect to the x-axis, y-axis, and z-axis. After the corruption, it is very evident that the input data is vibrating. The output trajectory after denoising by our model is very close to the ground truth, showing the effectiveness of the proposed method. In Figs. 15(a) and 15(b), the predicted trajectory circled in black differs significantly from the actual trajectory, with the difference increasing with the length of the prediction. On the one hand, it is evident that the prediction task is more challenging than the denoising task. On the other hand, it is difficult to predict long sequences. We also visualize the corresponding predicted results in Fig. 16. Visually, there is almost no difference between the predicted result and the ground truth, which supports the effectiveness of our prediction method. In addition, Fig. 15(d) shows the pose error curve of the complete motion. Compared with the input, the pose error curve of the output is almost flat and close to zero in general, indicating the effectiveness of our proposed method in motion denoising and prediction.

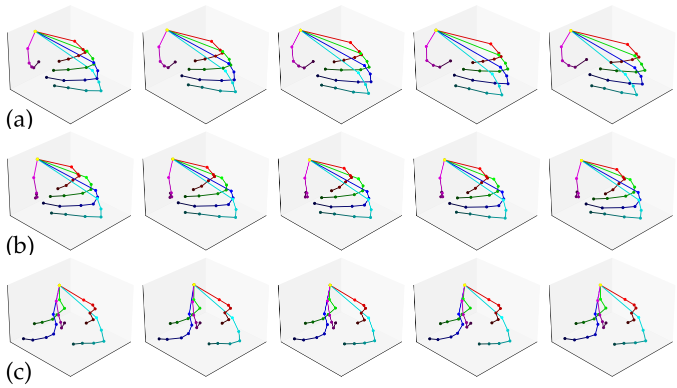

Fig. 16. Plots of prediction results on the NYU dataset: (a) ground truth, (b) the predicted result, (c) the predicted result without denoising.

Learned Adjacent Matrix. Since human motion is extremely complex and diverse, it is difficult to model spatial-temporal relationships. This work shows GCNs are instrumental in solving the motion denoising and prediction problem to some extent. The learned graph is also shown to demonstrate its adaptability in a more intuitive manner, which is shown in Fig. 17.

In Figure Fig. 17, we display a learned adjacent matrix heatmap and its corresponding normalized adjacent matrix, where the intensity of each element in the matrix is represented by a colorful scale, indicating the strength of the connection. Fig. 17(a) shows the original normalized adjacent matrix heatmap, where self-connections, physics-connected connections, and symmetry-connected connections are considered. Fig. 17(b) displays an example of its corresponding learned adjacency matrix generated by our proposed model. Note that both the original normalized adjacent matrix and the learned adjacency matrix have 3 channels, and in Fig. 17, the effect of all channels is overlaid. As can be seen in Fig. 17, the learned structure of the graph is more adaptive and not restricted by physical or physiological constraints, allowing GNNs to fully leverage their advantages.

5 Conclusion and Discussions

In conclusion, our work presents a multi-task framework called Multi-STGAE that addresses the challenge of raw hand motion data with errors in HCI applications. We achieve this by explicitly modeling the structural priors of hands and the temporal coherence of motion through the spatial-temporal graph block. Our proposed novel hand

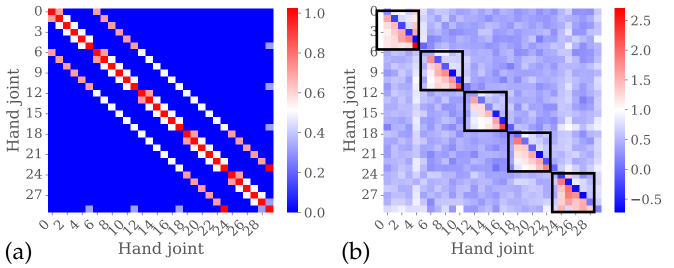

Fig. 17. Heatmaps of the original (a) and learned (b) adjacent matrices.

skeleton partition strategy enhances the information compensation from clean joints to noisy joints, and we introduce a loss function to meet the bone length constraint. The motion prediction task is incorporated to maintain temporal dynamics, and the gate mechanism is designed to avoid negative transfer between different tasks. We show that motion denoising and motion prediction are related tasks and that a multi-tasking framework can outperform any single-task framework. Additionally, our method can alleviate the delay problem, thereby improving user experience in interactive applications. Experimental results demonstrate that Multi-STGAE outperforms state-of-the-art methods, confirming the efficacy of our approach.

The main focus of our research is on the use of motion data for pose estimation, although our work has potential applications in denoising data acquired through other methods, such as motion capture. Additionally, our Multi-STGAE model can be extended to the processing of human motion data. However, despite the promising results of our work, there are limitations that require further exploration. (1) Our pre-training process involves synthetic motion data, which may not generalize well to real-world scenarios due to domain gap issues. To address this limitation, collecting a small amount of real motion data and incorporating unsupervised learning techniques could help overcome domain shifts and improve generalization performance in real-world applications. (2) Currently, our system adeptly handles sequences at a particular frame rate. However, it remains paramount to evaluate its efficacy across diverse temporal granularities. Conducting a comprehensive examination could provide insights into the balance between computational efficiency, motion detail capture, and noise sensitivity. (3) While our model mainly leverages input supervision, the integration of a discriminator loss, similar to the principles of Generative Adversarial Networks (GANs) [4], to distinguish the noise level in a joint could provide further refinement. This strategy could guarantee that the produced motions are not just precise but also maintain a semblance of realism. These avenues of research present exciting possibilities for future work in the field.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Project Number: 62272019), and by the EPSRC NorthFutures project (Ref: EP/X031012/1).

References

[1] G. Barquero, J. Núñez, Z. Xu, S. Escalera, W.-W. Tu, I. Guyon, and C. Palmero. Comparison of spatio-temporal models for human motion and pose forecasting in face-to-face interaction scenarios. In *Understanding Social Behavior in Dyadic and Small Group Interactions*, pages 107–138. PMLR, 2022.

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

[3] 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.

[4] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan. Generative adversarial networks: A survey toward private and secure applications. *ACM Computing Surveys (CSUR)*, 54(6):1–38, 2021.

[5] M. Centin and A. Signoroni. Mesh denoising with (geo) metric fidelity. *IEEE Transactions on Visualization and Computer Graphics*, 24(8):2380–2396, 2017.

[6] A. Chan, R. Lau, and L. Li. Hand motion prediction for distributed virtual environments. *IEEE transactions on visualization and computer graphics*, 14(1):146–159, 2007.

[7] 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.

[8] K. Chen, Y. Wang, S.-H. Zhang, S.-Z. Xu, W. Zhang, and S.-M. Hu. Mocap-solver: A neural solver for optical motion capture data. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021.

[9] Q. Cui and H. Sun. Towards accurate 3d human motion prediction from incomplete observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4801–4810, 2021.

[10] Q. Cui, H. Sun, and F. Yang. Learning dynamic relationships for 3d human motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6519–6527, 2020.

[11] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat. Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In *3DOR-10th Eurographics Workshop on 3D Object Retrieval*, pages 1–6, 2017.

[12] K. Doshi and Y. Yilmaz. Multi-task learning for video surveillance with limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3889–3899, 2022.

[13] 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.

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

[15] X. Guo and J. Choi. Human motion prediction via learning local structure representations and temporal dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2580–2587, 2019.

[16] A. Hernandez, J. Gall, and F. Moreno-Noguer. Human motion prediction via spatio-temporal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7134–7143, 2019.

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

[18] 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.

[19] H. Jung, Y. Kim, H. Jang, N. Ha, and K. Sohn. Multi-task learning framework for motion estimation and dynamic scene deblurring. *IEEE Transactions on Image Processing*, 30:8170–8183, 2021.

[20] 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.

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

[22] J. N. Kundu, M. Gor, and R. V. Babu. Bihmp-gan: Bidirectional 3d human motion prediction gan. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8553–8560, 2019.

[23] 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.

[24] B. Li, J. Tian, Z. Zhang, H. Feng, and X. Li. Multitask non-autoregressive model for human motion prediction. *IEEE Transactions on Image Processing*, 30:2562–2574, 2020.

[25] 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.

[26] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 214–223, 2020.

[27] 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.

[28] L. Lin, S. Song, W. Yang, and J. Liu. Ms2l: Multi-task self-supervised learning for skeleton based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2490–2498, 2020.

[29] S. Liu, Y. Liang, and A. Gitter. Loss-balanced task weighting to reduce negative transfer in multi-task learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9977–9978, 2019.

[30] 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.

[31] Z. Liu, K. Lyu, S. Wu, H. Chen, Y. Hao, and S. Ji. Aggregated multi-gans for controlled 3d human motion prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 2225–2232, 2021.

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

[33] R. C. Luo and L. Mai. Human intention inference and on-line human hand motion prediction for human-robot collaboration. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5958–5964. IEEE, 2019.

[34] K. Lyu, H. Chen, Z. Liu, B. Zhang, and R. Wang. 3d human motion prediction: A survey. *Neurocomputing*, 489:345–365, 2022.

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

[36] 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.

[37] Q. Men, E. S. L. Ho, H. P. H. Shum, and H. Leung. A quadruple diffusion convolutional recurrent network for human motion prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(9):3417–3432, 2021.

[38] E. Ng, S. Ginosar, T. Darrell, and H. Joo. Body2hands: Learning to infer 3d hands from conversational gesture body dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11865–11874, June 2021.

[39] C. Palmero, G. Barquero, J. C. J. Junior, A. Clapés, J. Núnez, D. Curto, S. Smeureanu, J. Selva, Z. Zhang, D. Saeteros, et al. Chalearn lap challenges on self-reported personality recognition and non-verbal behavior forecasting during social dyadic interactions: Dataset, design, and results. In *Understanding Social Behavior in Dyadic and Small Group Interactions*, pages 4–52. PMLR, 2022.

[40] G. Park, A. Argyros, J. Lee, and W. Woo. 3d hand tracking in the presence of excessive motion blur. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1891–1901, 2020.

[41] J. M. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European conference on computer vision*, pages 35–46. Springer, 1994.

[42] J. Shen, J. Dudley, G. Mo, and P. O. Kristensson. Gesture spotter: A rapid prototyping tool for key gesture spotting in virtual and augmented reality applications. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3618–3628, 2022.

[43] 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.

[44] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang. Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3300–3315, 2021.

[45] H. P. H. Shum, T. Komura, and S. Takagi. Fast accelerometer-based motion recognition with a dual buffer framework. *The International Journal of Virtual Reality*, 10(3):17–24, Sep 2011.

[46] H. Tang, J. Liu, M. Zhao, and X. Gong. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Fourteenth ACM Conference on Recommender Systems*, pages 269–278, 2020.

[47] 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.

[48] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):1–10, 2014.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] Y. Zhang and Q. Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[56] C. Zhong, L. Hu, Z. Zhang, Y. Ye, and S. Xia. Spatio-temporal gating-adjacency gcn for human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6447–6456, 2022.

[57] K. Zhou, R. Cai, Y. Ma, Q. Tan, X. Wang, J. Li, H. P. Shum, F. W. Li, S. Jin, and X. Liang. A video-based augmented reality system for human-in-the-loop muscle strength assessment of juvenile dermatomyositis. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[58] K. Zhou, C. Chen, Y. Ma, Z. Leng, H. P. Shum, F. W. Li, and X. Liang. A mixed reality training system for hand-object interaction in simulated microgravity environments. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2023.

[59] K. Zhou, Z. Cheng, H. P. Shum, F. W. Li, and X. Liang. Stgae: Spatial-temporal graph auto-encoder for hand motion denoising. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 41–49. IEEE, 2021.

[60] K. Zhou, J. Fan, H. Fan, and M. Li. Secure image encryption scheme using double random-phase encoding and compressed sensing. *Optics &amp; Laser Technology*, 121:105769, 2020.

Kanglei Zhou received a BSc degree from the College of Computer and Information Engineering at Henan Normal University in 2020. Now, he is pursuing a Ph.D. degree at the School of Computer Science and Engineering, Beihang University. His research interests include human motion analysis and action quality assessment.

Hubert P. H. Shum (Senior Member, IEEE) is an Associate Professor and the Deputy Director of Research in Computer Science at Durham University, researching on spatio-temporal visual computing. He is also a Co-Founder of the Responsible Space Innovation Centre. Before this, he was an Associate Professor at Northumbria University, and a Postdoctoral Researcher at RIKEN Japan. He received his PhD degree from the University of Edinburgh. He chaired conferences such as Pacific Graphics, BMVC and SCA, and has authored over 150 research publications.

Frederick W. B. Li received a B.A. and an M.Phil. degree from Hong Kong Polytechnic University, and a Ph.D. degree from the City University of Hong Kong. He is currently an Associate Professor at Durham University, researching computer graphics, deep learning, collaborative virtual environments, and educational technologies. He is also an Associate Editor of Frontiers in Education and an Editorial Board Member of Virtual Reality & Intelligent Hardware. He chaired conferences such as ISVC and ICWL.

Xiaohui Liang received his Ph.D. degree in computer science and engineering from Beihang University, China. He is currently a Professor, working in the School of Computer Science and Engineering at Beihang University. His main research interests include computer graphics and animation, visualization, and virtual reality.