Graph Sequence Recurrent Neural Network for Vision-Based Freezing of Gait Detection

Kun Hu, Zhiyong Wang, *Member, IEEE*, Wei Wang, *Member, IEEE*, Kaylena A. Ehgoetz Martens, Liang Wang, *Fellow, IEEE*, Tieniu Tan, *Fellow, IEEE*, Simon J. G. Lewis, and David Dagan Feng, *Fellow, IEEE*

***Abstract*— Freezing of gait (FoG) is one of the most common symptoms of Parkinson's disease (PD), a neurodegenerative disorder which impacts millions of people around the world. Accurate assessment of FoG is critical for the management of PD and to evaluate the efficacy of treatments. Currently, the assessment of FoG requires well-trained experts to perform time-consuming annotations via vision-based observations. Thus, automatic FoG detection algorithms are needed. In this study, we formulate vision-based FoG detection, as a fine-grained graph sequence modelling task, by representing the anatomic joints in each temporal segment with a directed graph, since FoG events can be observed through the motion patterns of joints. A novel deep learning method is proposed, namely graph sequence recurrent neural network (GS-RNN), to characterize the FoG patterns by devising graph recurrent cells, which take graph sequences of dynamic structures as inputs. For the cases of which prior edge annotations are not available, a data-driven based adjacency estimation method is further proposed. To the best of our knowledge, this is one of the first studies on vision-based FoG detection using deep neural networks designed for graph sequences of dynamic structures. Experimental results on more than 150 videos collected from 45 patients demonstrated promising performance of the proposed GS-RNN for FoG detection with an AUC value of 0.90.**

***Index Terms*— Parkinson's disease, freezing of gait detection, deep learning, recurrent neural network, graph sequence.**

I. Introduction

PARKINSON'S disease (PD) is a neurodegenerative disorder, characterized by motor symptoms as a result of dopaminergic loss in the substantia nigra [1], [2]. Freezing of gait (FoG) is a debilitating symptom of PD, presenting as a sudden and brief episode where patients feet get stuck to the floor, and a cessation of movement results despite the intention to keep walking [3], [4]. As the disease progresses, FoG becomes more frequent and severe, posing a major risk for falls [5], [6] and eventually affecting the mobility, independence and quality of the life [7]. Early detection and quantification of FoG events are of great importance in clinical practice and could be used for the evaluation of treatment efficacy for FoG [8]. However, current FoG annotations heavily rely on subjective scoring by well-trained experts, which is extremely time-consuming. Therefore, computer-aided intelligent solutions are needed to establish objective and timely FoG detection and quantification.

Since observing PD subjects has been the gold standard of identifying when FoG events happen in clinical assessments [9], we can formulate FoG detection as a task which classifies each short segment of a long assessment video into two classes: FoG and non-FoG. To this end, vision-based FoG detection methods have been rarely studied, although a few vision-based Parkinsonian gait analysis methods have been proposed [10]–[13]. These PD gait analysis methods were mainly devised to characterize Parkinsonian gaits at a coarse level (e.g., categorizing a given gait video as normal or abnormal) and are not intended for accurately reporting individual FoG events in a video. In addition, following a traditional machine learning pipeline, these methods rely on extracting hand-crafted features by assuming that a video contains only a patient walking independently. However, in clinical settings, supporting staff are often involved to ensure the safety of

Manuscript received March 23, 2019; revised July 31, 2019 and September 24, 2019; accepted September 25, 2019. Date of publication October 15, 2019; date of current version November 27, 2019. This work was supported in part by the Australian Research Council (ARC) under Grant DP160103675, in part by the NHMRC-ARC Dementia Fellowship under Grant 1110414, in part by the National Health and Medical Research Council (NHMRC) of Australia Program Grant under Grant Ł1037746, in part by the Dementia Research Team under Grant Ł1095127, in part by the NeuroSleepCentre of Research Excellence under Grant Ł1060992, in part by the ARC Centre of Excellence in Cognition and Its Disorders Memory Program under Grant CE110001021, in part by the Sydney Research Excellence Initiative (SREI) 2020 of the University of Sydney, in part by the Natural Science Foundation of China under Grant 61420106015, and in part by the Parkinson Canada. The ethical approval of this research was obtained from the University of Sydney Human Ethics Board (#2014/255). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tao Mei. *(Corresponding author: Kun Hu.)*

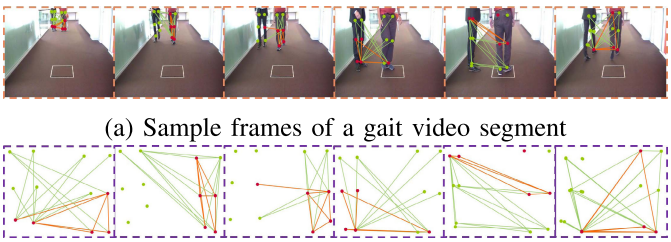
K. Hu, Z. Wang, and D. D. Feng are with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: kuhu6123@uni.sydney.edu.au.com; zhiyong.wang@sydney.edu.au; dagan.feng@sydney.edu.au).

W. Wang is with the National Laboratory of Pattern Recognition (NLPR), Center for Research on Intelligent Perception and Computing (CRIPAC), Institute of Automation Chinese Academy of Sciences (CASIA), University of Chinese Academy of Sciences (UCAS), Beijing 100190, China (e-mail: wangwei@nlpr.ia.ac.cn).

K. A. Ehgoetz Martens and S. J. G. Lewis are with the Parkinson's Disease Research Clinic, Brain and Mind Centre, The University of Sydney, Sydney, NSW 2050, Australia (e-mail: kaylena.ehgoetzmartens@sydney.edu.au; simon.lewis@sydney.edu.au).

L. Wang and T. Tan are with the National Laboratory of Pattern Recognition (NLPR), Center for Research on Intelligent Perception and Computing (CRIPAC), Institute of Automation Chinese Academy of Sciences (CASIA), University of Chinese Academy of Sciences (UCAS), Beijing 100190, China, and also with the Center for Excellence in Brain Science and Intelligence Technology (CEBSIT), Institute of Automation Chinese Academy of Sciences (CASIA), University of Chinese Academy of Sciences (UCAS), Beijing 100190, China (e-mail: wangliang@nlpr.ia.ac.cn; tnt@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TIP.2019.2946469



(a) Sample frames of a gait video segment

(b) Graph representation of each video frame

Fig. 1. Illustration of a graph sequence (b) produced by a gait video segment (a). The graph vertices are associated with the human anatomical joints which are obtained from a human pose estimation algorithm, and the edges among these vertices are further identified by the proposed method.

PD subjects. As a result, multiple persons can appear in recorded videos, which may violate the assumptions of those methods where only a patient appears.

Recent years have witnessed the ground-breaking success of deep learning techniques for many vision tasks, such as object recognition, video classification and human action recognition. These techniques provide a unique opportunity to develop deep learning based FoG detection methods to address the limitations of the existing PD gait analysis methods. Although many methods [14] have been proposed for generic video classification problems which involve significant variation between different classes (e.g., *kicking* and *jumping*) and each video frame is generally represented as a whole unit, they may neglect the subtle dynamics of FoG events, considering the variation among different subjects could be higher than that between FoG and non-FoG events. Several recent studies (e.g. Pose-CNN [15], [16] and our recent one [17]) have been conducted to model the subtle variation using region or patch based representations. However, the relationship among patches and the entire temporal sequence have not been adequately explored.

Therefore, for the first time, the current study aims to formulate FoG detection as a fine-grained graph sequence modelling task by representing each temporal video segment collected from a clinical assessment with a graph. As illustrated in Fig. 1, a number of consecutive temporal segments of an assessment video are organized in sequential order: for each segment, the anatomical joints are extracted and characterized as vertices of a directed graph, which is in line with the clinical practice where the joints of the knees and feet are particularly attended to. As a result, a graph sequence is obtained to represent this input video. Note that the spatial structures of the graph sequence are dynamic since the detected joints (vertices) vary over time (i.e. the locations of the subject and the supporting clinical staff could change, and the joints could be occluded from the view in recording procedures).

Traditionally, recurrent neural networks (RNNs) including the long short term memory (LSTM) and the gated recurrent units network (GRU) have been widely used to model sequential vector inputs with promising results [18], [19]. Although several studies have been proposed to address sequential graph inputs [20], [21], it is not trivial to apply them to general graph sequences especially when the structures are dynamic (i.e. vertices and edges can change over time). In this study, we propose a novel RNN architecture, namely graph sequence RNN (GS-RNN), to deal with general sequential graphs of dynamic structures. In particular, to leverage the success of gated mechanisms, which alleviates the gradient vanishing and exploding issues of the original RNN, GS-LSTM and GS-GRU are implemented. Computational operators, gated mechanisms and memory states of GS-RNN cells are devised to track sequential graph patterns while being compatible with dynamic graph structures. Experimental results demonstrate the effectiveness of the proposed GS-RNN architecture for the FoG detection task and the benefits of utilizing graph sequence representation. Moreover, graph sequence representations provide additional localization hints for clinical assessments.

In summary, the major contributions of this paper are threefold:

- We formulate FoG detection as a fine-grained graph sequence modelling task, which is one of the first studies to implement vision-based FoG detection. Instead of characterizing each video temporal segment as a whole unit or the patches of individual joints, we represent each video with a graph sequence where the vertices are associated with the anatomical joints, which enables fine-grained characterization of the dynamic patterns of FoG events.
- A novel recurrent neural network architecture GS-RNN is proposed to learn from graph sequences of dynamic structures. More specifically, GS-LSTM and GS-GRU are implemented to leverage the success of gated mechanisms.
- A large video dataset was created during the clinical assessments of 45 PD subjects to evaluate the effectiveness of our proposed methods.

The paper is organized as follows. Section II reviews the related works including Parkinsonian gait analysis and various deep learning techniques. Section III introduces the details of our proposed methods. Section IV presents comprehensive experimental results to evaluate the effectiveness of our proposed GS-RNN for FoG detection. Lastly, Section V concludes our study with discussions for future work.

II. Related Work

In this section, related studies are reviewed from three aspects: vision-based Parkinsonian gait analysis methods, deep learning based video classification methods, and neural networks for graph data. Note traditional hand-crafted feature based recognition methods are omitted, as deep learning based methods have achieved the state-of-the-art recognition performance. Skeleton based human action recognition methods (e.g., [22]–[24]) are also omitted, as they generally rely on accurate pose information and cannot be directly applied to our FoG detection task where pose information may be incomplete.

A. Vision-Based Parkinsonian Gait Analysis

Several vision-based Parkinsonian gait analysis methods have been proposed [10]–[12]. At first the sagittal view was

applied to record human gait by placing a camera laterally to human subjects. In [10], the stride cycle and the posture lean related features were introduced to characterize gait patterns. The motion cue matching was be computed by the cosine similarity between normal and abnormal gait and the matching percentage was used to predict the label for an entire walking behaviour. However, temporal localization is not available to accurately detect abnormal patterns within a video. In [11], gait patterns were characterized by various motion features including stride length, leg angle, and average cycle time. To identify abnormal gait patterns, traditional binary classifiers were utilized. Besides the sagittal view, frontal view videos have also been explored due to the convenience of the space saving set-up, where a subject is required to walk towards and away from a recording camera [12]. The set-up is similar to clinical assessments and can avoid the issue that one leg is occluded by the other.

Note that these methods were not devised specifically for identifying FoG events and only perform crude gait analysis of PD patients. In addition, the traditional pattern recognition pipeline is followed in these studies, whereby hand-crafted features are extracted and fed into a machine learning model such as support vector machine (SVM), generalized linear model or ensemble learning methods to obtain predictions. However, extracting hand-crafted features usually requires strong assumptions, which may not be feasible in realistic scenarios. For example, these methods often assume that only a patient appears in a video and can walk independently. This ignores the fact that the patient may have mobility difficulties and require external support to prevent possible falls. To address these limitations, deep learning techniques provide a great opportunity for developing real-world applicable FoG detection methods built on the ground-breaking success of many visual understanding tasks.

B. Deep Learning-Based Video Classification

Deep learning techniques have been widely utilized for video classification due to their great success in many visual understanding tasks. At first, single stream [25] and two-stream methods [26] were proposed. The single-stream based method applies pre-trained 2D convolution filters frame by frame and different temporal fusion strategies are investigated. The two-stream based method takes the advantage of the appearance and optical flow features obtained by 2D convolutions to form spatial and temporal representations. Based on these pioneering studies, three major types of deep learning based methods are currently utilized to recognize human actions in video: convolution neural network (CNN), recurrent neural network (RNN) and two-stream based methods. The first type in general extends the 2D CNN architecture to its 3D counterpart by which the convolution filters are extended to filter 3D video data, such as C3D [27], P3D [28] and I3D [14]. By considering an input video as a 2D image sequence, the second type aims to model the temporal structure with recurrent neural networks such as long short term memory (LSTM) or gated recurrent units (GRU) [29], [30]. The last type which is based on the pioneering two-stream approach represents video content with both appearance and motion features [31].

However, the intra-class variation could be higher than the inter-class variation for FoG detection, while these above-mentioned methods mainly address generic human action recognition problems which involve significant inter-class variation. Therefore, novel fine-grained recognition methods are needed to take the characteristics of FoG videos in clinical assessments into consideration. Several recent studies (e.g. Pose-CNN [15]–[17]) model the subtle variation with region or patch based representations. However, the relations among patches and an entire temporal sequence have not been adequately explored.

C. Neural Networks for Graph Data

Graph neural network (GNN) [32] was proposed as one of the first graph-based neural networks to process the data containing graph structures. Recently, graph convolution neural network (GCNN) has been proposed to exploit the structure context of input data as an extension of the convolution neural network [33]. It helps solve many challenging problems including material design which involves molecular structures [34], [35], social network analysis [36], pose-based applications [37], [38], video analysis [16], [39], and sheds lights on FoG detection by analyzing temporal structure data. The key advantages of GCNNs are implemented by graph convolution layers which address graph inputs of varying structures. By stacking multiple graph convolution layers, it is feasible to construct deep neural networks. Nonetheless, GCNNs are designed for independent graph inputs and not available to formulate sequential or temporal patterns from graph sequences.

Generally, RNNs have been widely used to model sequential data. In particular, the LSTM and GRU methods were proposed to address the gradient vanishing and exploding issues of the original RNNs by introducing gated mechanisms [18], [19]. GRU involves less computation compared with LSTM while keeping similar performance and improving the efficiency of the original RNNs. Moreover, GRU has shown better classification performance on smaller datasets [40]. However, these RNNs are designed for general sequential inputs of which the input vectors are of a fixed length, whilst graph sequences usually describe more complex structures over time and it is more challenging to learn. Although structural graph RNNs [20], [21], [41]–[43] have been proposed to take graph sequences of fixed graph structures as the inputs, dynamic graph sequences, which are more general for a wide range of applications, have not been fully explored in the past. Therefore, advanced RNNs are needed to represent and model these complex patterns conveyed through the graph sequences of dynamic graph structures.

III. Proposed Method

The major components of our novel GS-RNN architecture are illustrated in Fig. 2, including the adjacency estimation layer, the graph RNN layer, the vertex-wise RNN layer and the graph pooling layer: the adjacency estimation layer aims

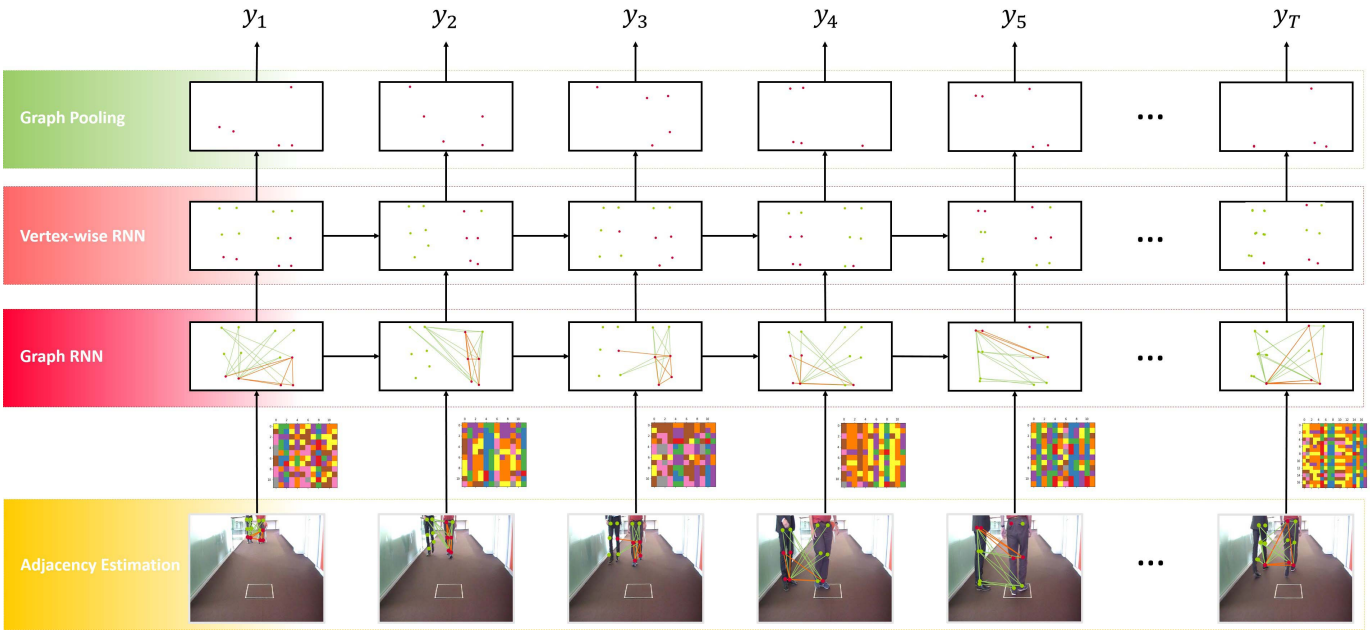


Fig. 2. Illustration of the proposed GS-RNN architecture for modelling gait videos with graph sequences. The adjacency estimation layer estimates the edge weights by utilizing the bilinear transform, the graph RNN layer is designed to track and propagate temporal graph patterns, the vertex-wise RNN layer helps to reduce model complexity by taking fewer vertex relations into account, and the graph pooling layer generates graph-level predictions referring to the vertices with top likelihood which contribute to the FoG patterns.

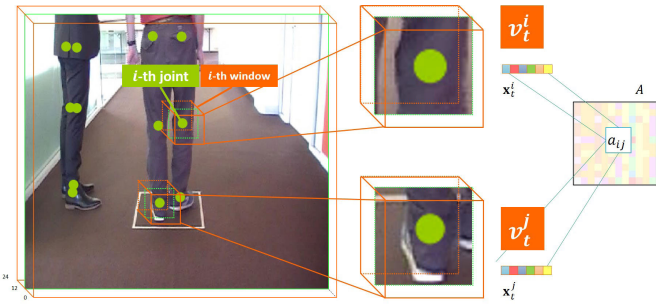


Fig. 3. Illustration of the construction of $\mathbb{G}_t$ from the input temporal video segment V_t. Joints are detected by applying convolution pose machines to the middle frame of V_t and proposals are extracted by the bounding windows centered at their associated joints. By treating the proposals $\{\mathbf{v}_t^i\}$ as the vertices of a graph and computing the pre-trained feature of each proposal $\mathbf{v}_t^i$ as $\mathbf{x}_t^i$, the adjacency matrix $\mathbf{A}$ can be estimated edge-wisely.

to estimate the weights of the edges of each graph; the graph RNN layer is designed to track and propagate the graph patterns of the input graph sequence; the vertex-wise RNN layer helps reduce model complexity by involving less vertex relations; the graph pooling layer generates graph-level predictions, which refers to the vertices that have the highest likelihood of contributing to FoG patterns. Similar to general RNNs, deep representation can be achieved by stacking multiple graph RNN layers. Furthermore, the architecture can be extended as bi-directional GS-RNNs to take both the past and the future graphs for modelling.

A. Anatomic Joint Graph Sequence

As FoG events can be observed from anatomic regions, anatomic joint proposals are extracted to construct a graph as illustrated in Fig. 3 from each temporal segment of an input video by adopting the convolution pose machines [44], [45]. In particular, a clinical assessment video is treated as a sequence $V = \{V_t\}$ of which the element V_t is a temporal segment of a fixed duration and t indicates its temporal index. For each V_t, convolution pose machines take the middle frame to compute anatomic joint locations. Hence, square windows can be identified with their centres located at each joint position of V_t. These windows obtained from the middle frame are extended to the remaining frames of V_t. Hence, the pixels within the i-th window can be extracted as an anatomic joint proposal $\mathbf{v}_t^i$ to characterize the local patterns around the i-th joint.

By treating $\mathbf{v}_t^i$ as the i-th vertex of a graph $\mathbb{G}_t$, and thus $\mathbb{V}_t = \{\mathbf{v}_t^i\}$ is the set of vertices of $\mathbb{G}_t$. In addition, $\mathbb{E}_t$ denotes a set of ordered vertex pairs (i.e., edges or arrows) to represent the relations between any two vertices. Hence, a directed graph $\mathbb{G}_t = (\mathbb{V}_t, \mathbb{E}_t)$ is derived to represent the video segment V_t. To characterize the edges in $\mathbb{E}_t$, a weighted adjacency matrix $\mathbf{A}_t = (a_t^{ij}) \in \mathbb{R}^{n \times n}$ is introduced. Note that $\mathbf{A}_t$ is possible to be asymmetrical as the interactions among joints can be of either action or reaction. To further characterize the graph $\mathbb{G}_t$, let $\mathbf{X}_t = (\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^n)^T$, where $\mathbf{x}_t^i \in \mathbb{R}^d$ denotes the vertex feature vector computed using $\mathbf{v}_t^i$ via pre-trained neural networks to represent joint appearance and motion; let y_t^i be a binary response to indicate whether FoG occurs within the i-th anatomic joint proposal at the temporal index t or not (i.e., 1 for FoG and 0 for non-FoG). In particular, denote $y_t = \max_i y_t^i$ as the graph-level response. Note that at least one joint contributes to an FoG event if a graph-level response is annotated as FoG.

According to the above discussions, an FoG assessment video V can be represented as a dynamic graph sequence $\{\mathbb{G}_t\}$. In terms of the dynamic characteristics, there are two

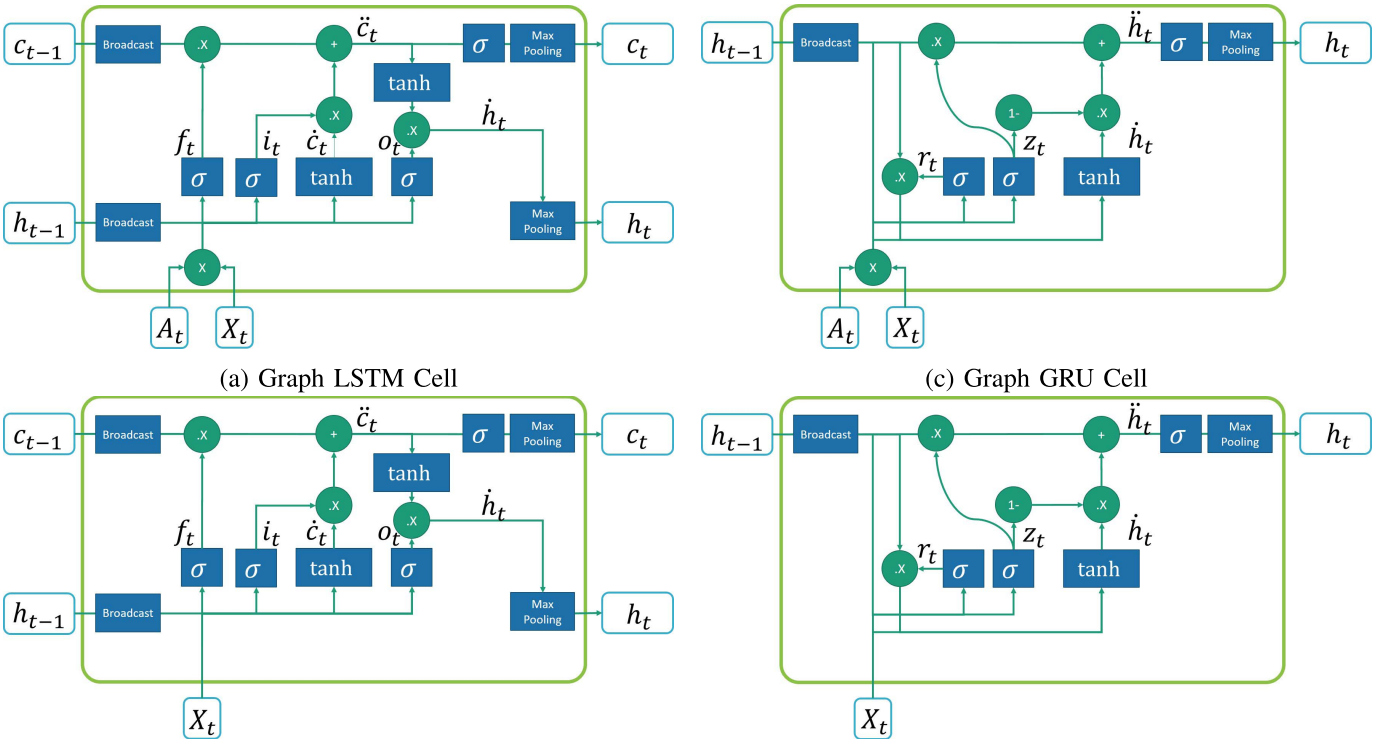


Fig. 4. Illustration of the gate mechanism of the proposed Graph RNN cells based on LSTM and GRU for graph sequence of dynamic structures. Two kinds of cells are designed: the graph RNN cell is applied to represent the vertex patterns with their adjacency relations; the vertex-wise RNN cell is devised to reduce the model complexity, which helps build deep GS-RNNs.

unique attributes. Firstly, the joints can be occluded during a trial, and it is not ensured that all the joints could be accurately tracked across all temporal indices. Secondly, as a computer-aided joint estimation method, some joints could be incorrectly identified by the convolution pose machine. As a result, the vertices and the edges of $\mathbb{G}_t$ could change along the temporal indices. Therefore, advanced sequential modelling methods are required to process dynamic graph sequences by effectively characterizing the dynamic structural patterns for FoG detection.

B. Adjacency Matrix Estimation

When the prior knowledge of a dataset is not available, an adjacency estimation layer is adopted in a GS-RNN to learn a weighted adjacency matrix $\mathbf{A}_t$ of which the element a_t^{ij} represents the relationship between the joint proposals $\mathbf{v}_t^i$ and $\mathbf{v}_t^j$. This layer introduces a bilinear transformation to obtain edge weight estimation a_t^{ij}, which explores the vertex features $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$. In addition, a bilinear operation is able to address the inconsistent dimensions of the two inputs, thus $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$ can be the features of different modalities. By adding superscripts to $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$ to denote different feature modalities, $\mathbf{x}_t^{i(c)} \in \mathbf{R}^p$ is the feature vector extracted from the pre-trained C3D and $\mathbf{x}_t^{j(r)} \in \mathbf{R}^q$ is extracted from the pre-trained ResNet-50. In particular, a_t^{ij} is estimated as:

$$a_t^{ij} = g(\mathbf{x}_t^{i(c)}\mathbf{M} \odot \mathbf{x}_t^{j(r)}), \tag{1}$$

where $\mathbf{M} \in \mathbf{R}^{p\times q}$ is a matrix of parameters which can be updated during the backward propagation, $\odot$ is an element-wise vector multiplication operator, and $g(\cdot)$ represents a function of $\mathbf{R}^q \rightarrow \mathbf{R}$ which is chosen as a linear function with trainable weights and bias in this study. Note that the estimated adjacency matrix $\mathbf{A}_t$ is asymmetrical, i.e., $a_t^{ij} \neq a_t^{ji}$, due to the asymmetry of $\mathbf{M}$ and the different feature modalities of vertex $\mathbf{v}_i^t$ and $\mathbf{v}_j^t$. Such asymmetry is helpful to represent different interactions (i.e., action and reaction) between vertices. Note that the computation of $\mathbf{A}_t$ is differentiable and $\mathbf{A}_t$ can be learned through both forward and backward propagations.

C. Graph RNN Cell

Graph RNN cells in this study introduce gate mechanisms similar to LSTM and GRU. Based on these two types of mechanisms, we introduce the graph LSTM cell and the graph GRU cell as shown in Fig. 4. The computations of the Graph LSTM cell shown in Fig. 4 (a) are as follows:

$$\mathbf{i}_t := \sigma(\mathbf{W}_{xi}\mathbf{A}\mathbf{X}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{1}_t b_i), \tag{2}$$
$$\mathbf{f}_t := \sigma(\mathbf{W}_{xf}\mathbf{A}\mathbf{X}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{1}_t b_f), \tag{3}$$
$$\mathbf{o}_t := \sigma(\mathbf{W}_{xo}\mathbf{A}\mathbf{X}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{1}_t b_o), \tag{4}$$
$$\dot{\mathbf{c}}_t := tanh(\mathbf{W}_{xc}\mathbf{A}\mathbf{X}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{1}_t b_c), \tag{5}$$
$$\ddot{\mathbf{c}}_t := \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \dot{\mathbf{c}}_t, \tag{6}$$
$$\dot{\mathbf{h}}_t := \mathbf{o}_t \odot tanh(\ddot{\mathbf{c}}_t), \tag{7}$$
$$\mathbf{c}_t := \mathbf{1}_{t+1} \max_{v\in V}(\ddot{\mathbf{c}}_t), \quad \mathbf{h}_t := \mathbf{1}_{t+1} \max_{v\in V}(\dot{\mathbf{h}}_t). \tag{8}$$

Similar to general LSTM cells, Eq. (2) computes the input gate $\mathbf{i}_t \in \mathbf{R}^{|\mathbb{V}_t| \times p}$ controlling the extent to which the new patterns are introduced to the cell, where p is the hidden size. All the vertices and the hidden state from the last temporal step are involved, where σ is an vertex-wise sigmoid function and $\mathbf{1}_t$ is an all-one $|\mathbb{V}_t|$ dimensional vector for broadcast purpose. First, vertex features $\mathbf{x}_t$ is multiplied with the adjacency matrix $\mathbf{A}$, which can be interpreted as the information exchange among the vertices in line with the edge weights. Let $\tilde{\mathbf{X}}_t = \mathbf{A}_t \mathbf{X}_t$ denote the exchanged vertex features of the input graph. The original feature and the exchanged feature of the i-th vertex can be written as $\mathbf{x}_t^i = (x_t^{i1}, x_t^{i2}, \ldots, x_t^{id})$ and $\tilde{\mathbf{x}}_t^i = (\tilde{x}_t^{i1}, \tilde{x}_t^{i2}, \ldots, \tilde{x}_t^{id})$ which correspond to the i-th row of $\mathbf{X}_t$ and $\tilde{\mathbf{X}}_t$, respectively.

$$\tilde{x}_t^{ij} = \sum_k a_t^{ik} x_t^{kj}. \tag{9}$$

In detail, Eq. (9) implies that the j-th exchanged feature of vertex i in $\tilde{\mathbf{x}}_t$ acquires information from the j-th original features of its direct predecessors (including itself) x_t^{kj} with the weight a_t^{ik} of the corresponding edges. Exchanging information between the vertices helps capture useful patterns such as simultaneously occurring abnormalities as references to enhance representation learning. Next, linear transforms $\mathbf{W}_{xi}$ and $\mathbf{W}_{hi}$ are applied to the exchanged features of the vertices and the hidden state from the last step which is broadcast to each vertex.

The forget gate $\mathbf{f}_t \in \mathbf{R}^{|\mathbb{V}_t| \times p}$ controls the extent to which the existing patterns should be kept, the output gate $\mathbf{o}_t \in \mathbf{R}^{|\mathbb{V}_t| \times p}$ controls the extent to which the cell state is involved into computing the output, and the candidate cell state $\dot{\mathbf{c}}_t \in \mathbf{R}^{|\mathbb{V}_t| \times p}$ are computed in the similar manner. Note that the graph-level cell state $\ddot{\mathbf{c}}_t \in \mathbf{R}^{|\mathbb{V}_t| \times p}$ and the graph-level hidden state $\dot{\mathbf{h}}_t \in \mathbf{R}^{|\mathbb{V}_t| \times p}$ are obtained vertex-wisely. The graph-level states are matrices of which each row contains the state of a particular vertex. Given the potential for a temporally changing graph structure, it is not always feasible to find the corresponding vertices of the next graph. Thus maximum pooling operators are introduced for $\ddot{\mathbf{c}}_t$ and $\dot{\mathbf{h}}_t$ on vertices and the pooling states are further broadcast to the vertices of the next graph. Hence, the graph LSTM cell is able to keep track of the dependencies among the graphs of the dynamic structures in the sequence.

Similar strategies are also applied to the computation of graph GRU cells as shown in Fig. 4 (c). Graph GRU cells involve less computation than graph LSTM cells, which are expected to be more efficient for training and prediction.

$$\mathbf{z}_t := \sigma(\mathbf{W}_{xz}\mathbf{A}\mathbf{X}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{1}_t b_z), \tag{10}$$

$$\mathbf{r}_t := \sigma(\mathbf{W}_{xr}\mathbf{A}\mathbf{X}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{1}_t b_r), \tag{11}$$

$$\dot{\mathbf{h}}_t := tanh(\mathbf{W}_{xh}\mathbf{A}\mathbf{X}_t + \mathbf{r}_t \odot \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{1}_t b_h), \tag{12}$$

$$\ddot{\mathbf{h}}_t := (1 - \mathbf{z}_t)\dot{\mathbf{h}}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1}, \tag{13}$$

$$\mathbf{h}_t := \mathbf{1}_{t+1} \max_{v \in V} \sigma(\ddot{\mathbf{h}}_t). \tag{14}$$

D. Vertex-Wise RNN Cell

In general, stacking multiple Graph RNN layers in a GS-RNN involves expensive non-linear computation, which significantly increases the model complexity. Therefore, it tends to result in over-fitting issues. To alleviate such issues and to build deep GS-RNNs, vertex-wise RNN cells are proposed, which apply shared linear transformations on each vertex separately. As a result, no pattern exchange is conducted in the vertex-wise RNN cells as in graph RNN cells. Vertex-wise RNN cells are implemented as vertex-wise LSTM cells and vertex-wise GRU cells. The computation of the Vertex-wise LSTM cell shown in Fig. 4 (b) are formulated from Eq. (15) to Eq. (21).

$$\mathbf{i}_t := \sigma(\mathbf{W}_{xi}\mathbf{X}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{1}_t b_i), \tag{15}$$

$$\mathbf{f}_t := \sigma(\mathbf{W}_{xf}\mathbf{X}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{1}_t b_f), \tag{16}$$

$$\mathbf{o}_t := \sigma(\mathbf{W}_{xo}\mathbf{X}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{1}_t b_o), \tag{17}$$

$$\dot{\mathbf{c}}_t := tanh(\mathbf{W}_{xc}\mathbf{X}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{1}_t b_c), \tag{18}$$

$$\ddot{\mathbf{c}}_t := \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \dot{\mathbf{c}}_t, \tag{19}$$

$$\dot{\mathbf{h}}_t := \mathbf{o}_t \odot tanh(\ddot{\mathbf{c}}_t), \tag{20}$$

$$\mathbf{c}_t := \mathbf{1}_{t+1} \max_{v \in V}(\ddot{\mathbf{c}}_t), \quad \mathbf{h}_t := \mathbf{1}_{t+1} \max_{v \in V}(\dot{\mathbf{h}}_t). \tag{21}$$

Note that the vertex-wise LSTM cell can be applied to a graph with an arbitrary number of vertices. Indeed, the proposed vertex-wise dense layer can be viewed as a special case of the graph RNN cell with $\mathbf{A}_t = \mathbf{I}$.

Similarly, the vertex-wise GRU cell shown in Fig. 4 (d) can be formulated from Eq. (22) to Eq. (26).

$$\mathbf{z}_t := \sigma(\mathbf{W}_{xz}\mathbf{X}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{1}_t b_z), \tag{22}$$

$$\mathbf{r}_t := \sigma(\mathbf{W}_{xr}\mathbf{X}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{1}_t b_r), \tag{23}$$

$$\dot{\mathbf{h}}_t := tanh(\mathbf{W}_{xh}\mathbf{X}_t + \mathbf{r}_t \odot \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{1}_t b_h), \tag{24}$$

$$\ddot{\mathbf{h}}_t := (1 - \mathbf{z}_t)\dot{\mathbf{h}}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1}, \tag{25}$$

$$\mathbf{h}_t := \mathbf{1}_{t+1} \max_{v \in V} \sigma(\ddot{\mathbf{h}}_t). \tag{26}$$

E. Graph Pooling

The state $\dot{\mathbf{h}}_t$ of the graph RNN cell and the Vertex-wise RNN cell can be viewed as hidden vertex features of the corresponding layer. Vertex-wise fully connected layers and activation functions can be applied to compute the FoG probability of each vertex, which indicates whether FoG happens in regard to the vertex or not. In general, for a graph $\mathbb{G}_t = (\mathbb{V}_t, \mathbb{E}_t)$, let $\hat{\mathbf{y}} \in \mathbf{R}^{|\mathbb{V}_p|}$ denote the outputs of the last fully connected layer, in which the i-th component $\hat{y}_i$ of $\hat{\mathbf{y}}$ is the estimation of the response of the i-th vertex y_i and $|\mathbb{V}_p|$ is the number of vertices. Since the vertex-level annotation is not available, i.e. no prior knowledge of y_t^i, a pooling strategy is applied to produce the response y_t of an entire temporal segment. Note that the pooling operation also eliminates the impact of the inconsistent structures of the entire graph sequence. The basic assumption of graph pooling is that at least one vertex contributes to the FoG event when FoG is annotated on the entire graph (the video segment). Hence, the maximum elements of $\hat{\mathbf{y}}_t$ can be viewed as an estimation of the graph-level response y_t, which is estimated as:

$$\hat{y}_t^g = max_i(\hat{y}_t^i), \tag{27}$$

where $\hat{y}_t^g$ represents the response of a graph.

Let f_g denote the computation of the forward propagation of the proposed GS-RNN. As a binary classification problem, a cross-entropy loss function is used to optimize the model f_g. A superscript n is used for the variables mentioned above to indicate the n-th training sample. Therefore, the loss function of the proposed GS-RNN is defined as:

$$J = -\sum_{n}\sum_{t}[y^{(n)}log(\hat{y}_t^{g(n)}) + (1 - y^{(n)})log(1 - \hat{y}_t^{g(n)})]. \tag{28}$$

F. Context Fusion

GS-RNN f_g computes the prediction $\hat{y}_t^g$ based on the input graph sequence. Nonetheless, the global context is absent by involving the joint proposals only. To further help accurately characterize FoG patterns, a context model f_c is applied to take the entire video segment sequence $\{V_t\}$ as the input. The prediction $\hat{y}_t^c = f_t(V_t)$ is derived and further fused with $\hat{y}_t^g$. The context model f_c is chosen as a pre-trained C3D network which is adopted for each video segment V_t and an RNN network which is further applied to formulate temporal relations between the C3D features of these temporal segments.

Without increasing the model complexity, the graph sequence based model and the context model are trained independently. Fusing $\hat{y}_t^g$ and $\hat{y}_t^c$ helps to better predict FoG events. Three fusion strategies, including the product fusion, the maximum pooling fusion and the linear fusion, are listed from Eq. (29) to Eq. (31).

$$\hat{y}_t = \hat{y}_t^c \hat{y}_t^g, \tag{29}$$

$$\hat{y}_t = max(\hat{y}_t^c, \hat{y}_t^g). \tag{30}$$

$$\hat{y}_t = \gamma\, \hat{y}_t^c + (1 - \gamma)\hat{y}_t^g, \quad \gamma \in (0, 1). \tag{31}$$

The outputs of these three fusing functions are in $[0, 1]$, representing the probability of an FoG event occurring in V_t.

IV. Experimental Results and Discussions

A. FoG Dataset & Evaluation Metrics

The dataset in this study consists of videos collected from 45 subjects who underwent clinical assessment. During the clinical assessment, each subject completed a Timed Up and Go (TUG) test used for functional mobility assessment [46]. The TUG tests were recorded by frontal view videos at 25 frames per second with a frame resolution of 720×576. FoG events within these videos were annotated by well-trained experts on a per-frame basis. Note that clinical staff were involved in the video recording processes to ensure the safety of the subjects during the TUG tests. Furthermore, the angle of the camera was set to capture the body parts from chest to feet to meet ethical requirements.

In summary, 167 videos totalling 25.5 hours in duration were acquired, where 8.7% of the total hours collected contained FoG patterns. This indicates a highly imbalanced dataset. For FoG detection, these videos were divided into 91,559 one-second long non-overlapped video segments. If any frame of a segment was annotated as FoG referring to the ground truth, this segment was labelled as FoG; otherwise it was labelled as non-FoG. The demographics of the dataset are listed in Table I including age, year since diagnosis, education, cognitive function (evaluated by MMSE), and Hoehn and Yahr stages [47], which is a common metric to describe the symptoms of PD progress. This dataset has the largest number of subjects in the literature of vision-based Parkinsonian gait analysis. For example, 11 subjects were involved in the work of [10] and 30 subjects in [11]. It is also the first one for FoG detection.

TABLE I
Demographics of the FoG Video Dataset

Age, years	Years since diagnosis	Education, years	MMSE
68.49 (7.7)	9.94 (5.8)	13.80 (3.2)	27.94 (1.9)
Hoehn and Yahr stages		1	2%
		2	38%
		3	18%
		4	33%
		5	10%

To evaluate the performance of the proposed method over this dataset comprehensively, a 5-fold cross validation was introduced. The 45 subjects were randomly and evenly partitioned into 5 groups. For each fold, the video segments of 4 groups were chosen for the training and validation purposes, and the video segments of the remaining one group were used for test purposes. Therefore, the videos of each subject only appeared in either the training or test partition, which helps to statistically estimate the performance for unseen subjects.

A number of metrics were adopted to comprehensively evaluate the FoG detection performance. Firstly, as the prediction $\hat{y}_t$ is continuous in $[0, 1]$, which indicates the FoG probability, a threshold θ should be identified in line with a specified use case. If $\hat{y}_t > \theta$, the corresponding video segment was marked as an FoG event; otherwise the segment was marked as a non-FoG event. Next, accuracy (the percentage of the samples correctly classified over the total sample size), sensitivity (true positive rate) and specificity (true negative rate) associated with this threshold were computed. By varying the threshold and plotting the corresponding sensitivity against 1-specificity, receiver operating characteristic (ROC) curve, and area under curve (AUC) were further utilized to evaluate the effectiveness of the proposed GS-RNNs.

B. Implementation Details

In this study, three types of pre-trained features were applied including Res-Net 50 vertex features, C3D vertex features and C3D context features. The details of obtaining these features are introduced as follows:

- **Res-Net 50 vertex feature** Setting the bounding window as 50×50 pixels, the size of each anatomic joint proposal was $25 \times 50 \times 50 \times 3$, where 25 was the frame rate and 3 represented the RGB channels. Pre-trained ResNet-50

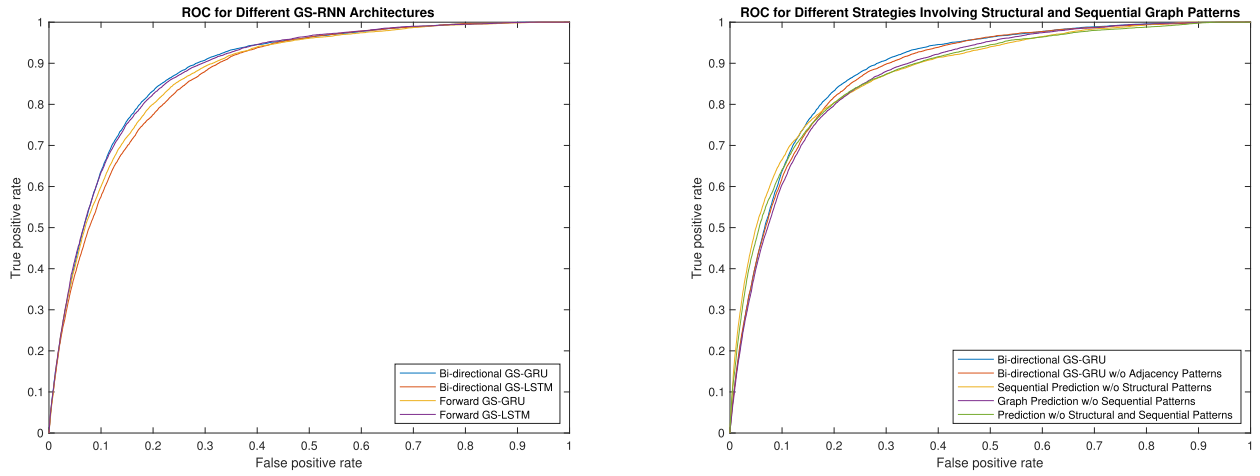


Fig. 5. ROC curves of GS-RNNs. (a) Comparison among different GS-RNN architectures. (b) Comparison among different strategies involving structural and sequential patterns.

with a size $(2, 2)$ at last pooling layer was applied frame by frame. The maximum pooling operation was applied along the temporal axis to reduce the model complexity and the computational cost. The dimension of Res-Net 50 vertex feature vector was 1×2048.

- **C3D vertex feature** The purpose of C3D vertex feature differ from the Res-Net 50 feature. The Res-Net 50 feature was only applied in the adjacency matrix estimation whilst the C3D vertex feature was used for modelling motion patterns as well. A higher precision of C3D feature was expected, and the proposal size was set to 100×100. The dimension of C3D vertex feature vector was 1×8192.
- **C3D context feature** The video frames were first re-sized with fixed aspect ratio and cropped into temporal clips with size of $25 \times 224 \times 224 \times 3$. These temporal clips were then fed to a pre-trained C3D network to compute spatial-temporal features. The dimension of the C3D context feature was 1×32768.

Given the imbalanced dataset, all the positive samples were used, with an equivalent number of negative samples randomly selected from the training set in each epoch for model training. The initial learning rate was set to 0.001, utilizing the stochastic gradient decent optimizer. With an Nvidia GTX 1080Ti GPU card, training of the bi-directional GS-GRU for each epoch was completed in 50 minutes (containing videos from 40 subjects); for the testing, each one-second video segment was computed within 0.5 sec.

C. FoG Detection Performance of GS-RNNs

GS-RNNs were evaluated by applying different types of GS-RNN cells including GS-LSTM cells or GS-GRU cells, and the directions were set as forward or bi-directional. Similar to general RNNs, the forward direction involved only the past graphs and can be utilized for online predictions. The bi-directional network utilized both the previous and the future graphs for accurate characterization and prediction of the graph FoG patterns. In detail, we implemented the forward GS-LSTM, the forward GS-GRU, the bi-directional GS-LSTM and the bi-directional GS-GRU. Each model contained an adjacency matrix estimation layer, a graph RNN layer, a vertex-wise layer and a graph pooling layer. Fig. 5 (a) shows the ROC curves of the proposed architectures and Table II lists the AUC of these curves. The bi-directional GS-GRU achieved the highest AUC value 0.884 compared with the other GS-RNN architectures. As expected, the bi-directional GS-GRU outperformed the forward GS-GRU. For GS-LSTM, the gated mechanisms are much more complex than GS-GRU, thus the model complexity of the bi-directional GS-LSTM increases, which negatively impacted the model performance.

GS-RNNs take both the structural and temporal graph patterns into account simultaneously. To evaluate how these

TABLE II
COMPARISON OF DIFFERENT GS-RNN ARCHITECTURES FOR FoG DETECTION

GS-RNN Architectures	AUC
Bi-directional GS-GRU	**0.884**
Bi-directional GS-LSTM	0.869
Forward GS-GRU	0.875
Forward GS-LSTM	0.883

TABLE III
COMPARISON OF DIFFERENT STRATEGIES TO REPRESENT STRUCTURAL AND SEQUENTIAL GRAPH PATTERNS

Methods	AUC
Bi-directional GS-GRU	**0.884**
Bi-directional GS-GRU w/o Adjacency Patterns	0.879
Sequential Prediction w/o Structural Patterns (C3D-GRU)	0.878
Graph Prediction w/o Sequential Patterns (GCNN)	0.878
Prediction w/o Structural and Sequential Patterns (C3D)	0.874

TABLE IV

DeLong's Test for Different Strategies to Represent Structural and Sequential Graph Patterns

Methods	AUC Diff.	Z	p
Sequential Prediction w/o Structural Patterns	0.007	4.04	5.33E-05
Bi-directional GS-GRU w/o Adjacency Patterns	0.005	4.54	5.51E-06
Prediction w/o Structural and Sequential Patterns	0.012	7.23	4.68E-13
Graph Prediction w/o Sequential Patterns	0.015	11.34	$< 2.2E-16$

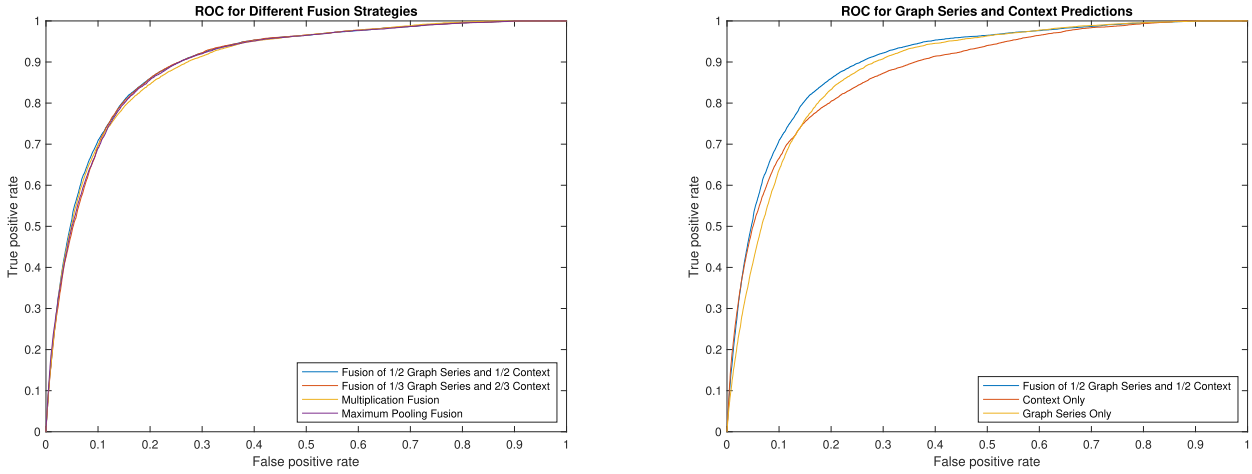


Fig. 6. Comparison of the detection results in terms of ROC Curves. (a) Comparison among different fusion strategies. (b) Comparison among the best fusion strategy and the cases that graph representation and context model are used independently.

patterns contributed to FoG detection, we further compared the bi-directional GS-GRU with different methods, which either partially or do not utilize these patterns. The first one is the bi-directional GS-GRU without the adjacency matrix, which only utilizes two vertex-wise GRU layers. The second (C3D-GRU) utilized the entire video segment with C3D by ignoring the anatomic graph as input, representing temporal patterns with GRU. The third (GCNN) ignored sequential patterns and applied GCNN to process each graph independently. The last (C3D) generated the prediction on entire temporal segments with C3D independently. Note that the second and the last methods can be viewed as context-level models as they utilized the entire video segment in a straightforward manner. Hence, for simplicity, the context model, which is further fused with GS-RNN for the complementary purpose, was selected from these two methods by referring to their performance in this study.

Fig. 5 (b) shows the ROC curves of these methods and Table III lists the AUC values of the ROC curves. The results indicate that simultaneously characterizing the structural and the sequential graph patterns enhanced the performance of FoG detection in terms of AUC. In addition, DeLong's test was applied to evaluate the statistical significance of the improvements [48], [49]. DeLong's test is a nonparametric approach by using the generalized U-statistics for the ROC curves. Table IV lists the results of the DeLong's test. The p-values of the bi-directional GS-GRU and the rest methods indicate that the improvements were statistically significant ($\alpha = 0.001$). Therefore, characterizing the structural and the sequential graph patterns of gaits was clearly helpful for FoG detection.

D. FoG Detection Performance of Fusion Strategies

To further improve the performance of FoG detection, the context prediction $\hat{y}_t^c$ was fused with the GS-RNN prediction $\hat{y}_t^g$. The context model was chosen as a general GRU network for sequential predictions, which utilized C3D context features as the input. Note that the GS-RNN model and the context model were trained independently avoiding an increase in model complexity. In terms of the fusion strategies, the linear fusion, the product fusion and the maximum pooling fusion were investigated. For the linear fusion, $\gamma = \frac{1}{2}$ and $\gamma = \frac{2}{3}$ were selected, which achieved the best image classification performance in [50].

Fig. 6 shows the ROC curves of different fusion methods and Table V lists the AUC values of these ROC curves. Fig. 6 (a) illustrates the curves of different fusion strategies. The linear fusion strategy with $\gamma = \frac{1}{2}$ achieved the best overall performance 0.900 in terms of AUC. In Fig. 6 (b), the curve of the best fusion strategy was compared with the cases that the graph sequence model and the context model are utilized independently. The fusion predictions improved the detection performance by taking advantage of the two independent methods. The improvement in terms of AUC to the best fusion method compared with $\hat{y}_t^c$ and $\hat{y}_t^g$ are 0.016 and 0.029, respectively.

In particular, the sensitivity, the specificity and the accuracy related to a threshold $\hat{\theta}$ were also adopted for the evaluation,

TABLE V
COMPARISON OF DIFFERENT FUSION STRATEGIES

	AUC	Youden's J	Sensitivity	Specificity	False positive rate	False negative rate	Likelihood ratio positive	Likelihood ratio negative	Accuracy
Linear fusion with $\gamma = \frac{2}{3}$	0.898	0.66	86.5%	79.6%	20.4%	13.5%	4.24	0.17	80.2%
Linear fusion with $\gamma = \frac{1}{2}$	**0.900**	**0.66**	**83.8%**	**82.3%**	**17.7%**	**16.2%**	**4.75**	**0.20**	**82.5%**
Maximum pooling fusion	0.898	0.66	83.7%	82.1%	17.9%	16.3%	4.67	0.20	82.2%
Product fusion	0.897	0.65	84.3%	80.4%	19.6%	15.7%	4.31	0.20	80.8%
Bi-directional GS-GRU	0.884	0.64	84.4%	79.2%	20.8%	15.6%	4.05	0.20	79.6%
Context model	0.878	0.61	77.4%	83.4%	16.6%	22.6%	4.65	0.27	82.8%

TABLE VI
COMPARISONS WITH EXISTING METHODS

Method	AUC	Sens.	Spec.	Acc.
C3D [27]	0.874	80.2	80.2	80.2
P3D [28]	0.819	77.1	74.1	74.5
Spatial + Dilated Temporal CNN [52]	0.844	80.0	78.9	79.0
2D CNN (ResNet-50) + LSTM [29]	0.863	84.9	74.8	75.7
Bilinear Attention Pooling [53]	0.848	85.1	75.0	75.9
Space-Time Region Graphs CNN [16]	0.846	78.3	77.5	77.6
C3DAN [54]	-	68.2	80.8	79.3
Global-local 2D CNN + LSTM [17]	0.869	84.5	76.5	77.2
Bi-directional GS-GRU	0.884	84.4	79.2	79.6
GS-GRU Fused with Context	**0.900**	83.8	82.3	82.5

which maximized the following Youden's J statistic [51]:

$$\hat{\theta} = \arg\min_{\theta} sensitivity + specificity - 1. \tag{32}$$

By maximizing this statistic, a threshold can be derived to treat the sensitivity and the specificity with equal importance. These evaluation metrics and associated J statistics are also listed in Table V. For the best fusion strategy, the sensitivity, specificity and accuracy values related to this threshold achieved 83.8%, 82.3% and 82.5%, respectively.

E. Comparison With State-of-the-Art Methods

We conducted comparisons with 6 existing video classification methods, including the 3D convolution methods [27], [28], the dilated temporal CNN method [52], the CNN-LSTM method [29], the bilinear pooling method [53] and the space-time region graphs CNN method [16], and our two recent studies, including C3DAN and global-local 2D CNN + LSTM are listed [?], [54]. As shown in Table VI, C3D and 2D CNN (ResNet-50) + LSTM outperformed the others, which suggests that integrating C3D and the ResNet-50 pre-trained neural networks to construct the vertex features and their relations is reasonable. Although dilated temporal convolutions have demonstrated their effectiveness for many temporal data related tasks, they were not able to capture the dynamic FoG patterns adequately. Space-time region graphs have been proposed to characterize the patch relations by utilizing GCNNs, however, the long-term temporal relations were not explored to characterize FoG patterns. Although bilinear methods have been successfully applied for fine-grained classification problems, the increased model complexity of bilinear models may compromise the performance for FoG detection.

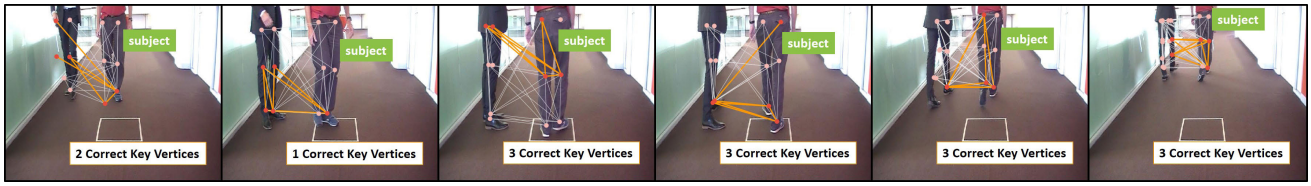
Our recent work C3DAN [54] aims to identify attended regions, but the movements of supporting staff may also be considered, which could compromise the performance of FoG detection. In [17], the structural patterns among joints were first explored for FoG detection with promising results. The improvement of GS-RNN over [17] demonstrates that temporal context is also important for characterizing the temporal dynamics of FoG events.

In summary, these comparisons clearly demonstrate the effectiveness of our proposed GS-RNN for FoG detection.

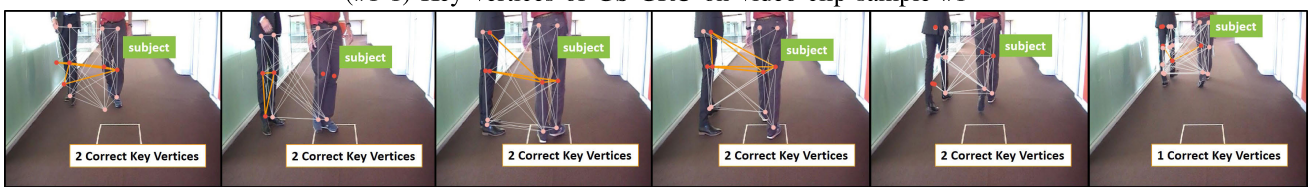
F. Key Vertex Localization

In order to further understand how the sequential graph representations contribute to FoG detection, Fig. 7 visualizes the vertex-level responses of the bi-directional GS-GRU and GCNN methods. Note that the bi-directional GS-GRU utilizes sequential patterns while GCNN treats input graphs individually. Two positive 6-second FoG video clips are visualized and one frame is selected per second for the illustration purpose. The vertices which achieve the top-5 FoG scores in a graph are highlighted in red and defined as key vertices.

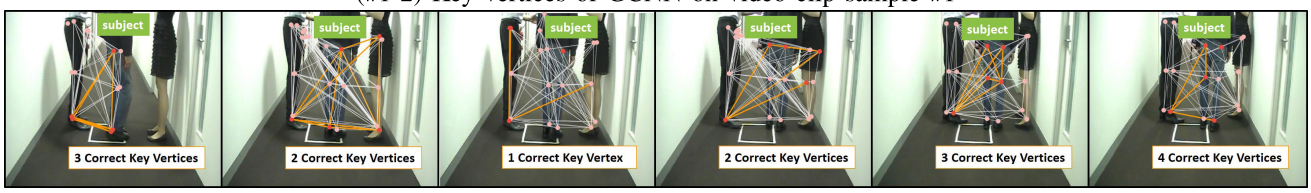
In general, multiple persons can appear in a video and the anatomic joint proposals (vertices) are produced collectively. For positive FoG video clips, the key vertices should be correctly located on the patient subjects because FoG events should not be associated with supporting staffs. Otherwise, the algorithm would produce incorrect predictions. Therefore, the key vertices correctly located on patient subjects are counted per second. For the bi-directional GS-GRU, most of the key vertices are located on the patient subjects while some top vertices occasionally appear on the supporting staffs, which demonstrates that GS-GRU takes the correct joint proposals to recognize FoG events. However, for GWN, which does not introduce graph sequential patterns, the count of correctly located key vertices decreases. It indicates that sequential graph representations play an important role in



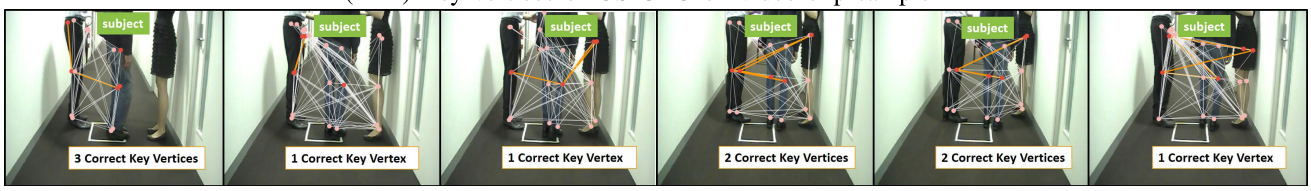
(#1-1) Key vertices of GS-GRU on video clip sample #1



(#1-2) Key vertices of GCNN on video clip sample #1



(#2-1) Key vertices of GS-GRU on video clip sample #2



(#2-2) Key vertices of GCNN on video clip sample #2

Fig. 7. Illustration of the key vertex localization task in two FoG video clip samples by utilizing the bi-directional GS-GRU and GCNN methods. Each clip is of 6-second length, of which one frame is selected per second for the figure, and the top-5 scored vertices (key vertices) are highlighted in red. The count of the key vertices correctly locating on the subjects are noted on each frame, and it can be observed that GS-GRU is more likely to focus on the subjects, which is benefited from the graph temporal relations.

improving the performance of FoG detection and can provide additional insights associated with FoG events.

V. Conclusion

In this study, a novel deep neural network architecture GS-RNN is presented to process the spatial temporal data represented as dynamic graph sequences. Graph RNN cells and vertex-wise RNN cells are devised as the building blocks of GS-RNNs, which model the structural and the temporal graph patterns simultaneously. To this end, GS-RNNs can be used to formulate vision-based FoG detection as a fine-grained sequential modelling task. Comprehensive experimental results on an in-house dataset, which has the largest number of subjects in the literature of video-based PD gait analysis, demonstrate the superior performance of the proposed GS-RNN architectures. In addition, the graph representation of anatomic joints provides an intuitive interpretation of the detection results by localizing the key vertices of an FoG video, which is helpful for clinical assessments in practice. In our future work, we will focus on simplifying GS-RNN cells and architectures to reduce the computational cost for training and prediction phases without compromising the model's learning capacity.

Acknowledgment

The authors thank their patients who participated into the data collection and they all provided written informed consent. They would also like to thank Moran Gilat, Julie Hall, Alana Muller, Jennifer Szeto and Courtney Walton for conducting and scoring the freezing of gait assessments. They would also like to thank ForeFront, a large collaborative research group dedicated to the study of neurodegenerative diseases.

References

[1] E. R. Dorsey *et al.*, "Global, regional, and national burden of Parkinson's disease, 1990–2016: A systematic analysis for the global burden of disease study 2016," *Lancet Neurol.*, vol. 17, no. 11, pp. 939–953, Nov. 2018.

[2] J. Jankovic, "ParkinsonâĽ™s disease: Clinical features and diagnosis," *J. Neurol., Neurosurgery Psychiatry*, vol. 79, no. 4, pp. 368–376, 2008.

[3] M. A. Hely, W. G. J. Reid, M. A. Adena, G. M. Halliday, and J. G. L. Morris, "The Sydney multicenter study of Parkinson's disease: The inevitability of dementia at 20 years," *Movement Disorders*, vol. 23, no. 6, pp. 837–844, Apr. 2008.

[4] M. Macht *et al.*, "Predictors of freezing in Parkinson's disease: A survey of 6,620 patients," *Movement Disorders*, vol. 22, no. 7, pp. 953–956, May 2007.

[5] B. R. Bloem, J. M. Hausdorff, J. E. Visser, and N. Giladi, "Falls and freezing of gait in Parkinson's disease: A review of two interconnected, episodic phenomena," *Movement Disorders*, vol. 19, no. 8, pp. 871–884, Aug. 2004.

[6] S. J. G. Lewis and R. A. Barker, "A pathophysiological model of freezing of gait in Parkinson's disease," *Parkinsonism Related Disorders*, vol. 15, no. 5, pp. 333–338, Jun. 2009.

[7] J. D. Schaafsma, Y. Balash, T. Gurevich, A. L. Bartels, J. M. Hausdorff, and N. Giladi, "Characterization of freezing of gait subtypes and the response of each to levodopa in Parkinson's disease," *Eur. J. Neurol.*, vol. 10, no. 4, pp. 391–398, Jul. 2003.

[8] S. Donovan *et al.*, "Laserlight cues for gait freezing in ParkinsonâĽ™s disease: An open-label study," *Parkinsonism Related Disorders*, vol. 17, no. 4, pp. 240–245, May 2011.

[9] T. R. Morris *et al.*, "Clinical assessment of freezing of gait in Parkinson's disease from computer-generated animation," *Gait Posture*, vol. 38, no. 2, pp. 326–329, Jun. 2013.

[10] T. Khan, J. Westin, and M. Dougherty, "Motion cue analysis for Parkinsonian gait recognition," *Open Biomed. Eng. J.*, vol. 7, pp. 1–8, Jan. 2013.

[11] M. Nieto-Hidalgo, F. J. Ferrández-Pastor, R. J. Valdivieso-Sarabia, J. Mora-Pascual, and J. M. García-Chamizo, "A vision based proposal for classification of normal and abnormal gait using RGB camera," *J. Biomed. Inform.*, vol. 63, pp. 82–89, Oct. 2016.

[12] M. Nieto-Hidalgo, F. J. Ferrández-Pastor, R. J. Valdivieso-Sarabia, J. Mora-Pascual, and J. M. García-Chamizo, "Vision based gait analysis for frontal view gait sequences using RGB camera," in *Proc. Int. Conf. Ubiquitous Comput. Ambient Intell.*. Cham, Switzerland: Springer, 2016, pp. 26–37.

[13] S. Sedai *et al.*, "3D human pose tracking using Gaussian process regression and particle filter applied to gait analysis of Parkinson's disease patients," in *Proc. IEEE 8th Conf. Ind. Electron. Appl. (ICIEA)*, Jun. 2013, pp. 1636–1642.

[14] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 318–335.

[15] G. Chéron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN features for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3218–3226.

[16] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proc. Eur. Conf. Comput. Vis.*, pp. 399–417, 2018.

[17] K. Hu, Z. Wang, K. E. Martens, and S. Lewis, "Vision-based freezing of gait detection with anatomic patch based representation," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 564–576.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," Jun. 2014, *arXiv:1406.1078*. [Online]. Available: https://arxiv.org/abs/1406.1078

[20] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5308–5317.

[21] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Apr. 2018, pp. 7444–7452.

[22] Y. Du, Y. Fu, and L. Wang, "Representation learning of temporal dynamics for skeleton-based action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3010–3022, Jul. 2016.

[23] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, Apr. 2018.

[24] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, Dec. 2018.

[25] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[26] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.

[28] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5534–5542.

[29] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2625–2634, 2015.

[30] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[31] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1933–1941.

[32] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[33] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," Dec. 2013, *arXiv:1312.6203*. [Online]. Available: https://arxiv.org/abs/1312.6203

[34] D. K. Duvenaud *et al.*, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[35] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," Jun. 2015, *arXiv:1506.05163*. [Online]. Available: https://arxiv.org/abs/1506.05163

[36] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," Sep. 2016, *arXiv:1609.02907*. [Online]. Available: https://arxiv.org/abs/1609.02907

[37] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, "Semantic graph convolutional networks for 3D human pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3425–3435.

[38] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3595–3603.

[39] X. Liu *et al.*, "Social relation recognition from videos via multi-scale spatial-temporal reasoning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3566–3574.

[40] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," Dec. 2014, *arXiv:1412.3555*. [Online]. Available: https://arxiv.org/abs/1412.3555

[41] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," Nov. 2015, *arXiv:1511.05493*. [Online]. Available: https://arxiv.org/abs/1511.05493

[42] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 362–373.

[43] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 1227–1236.

[44] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7291–7299.

[45] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4724–4732.

[46] A. Shumway-Cook, S. Brauer, and M. Woollacott, "Predicting the probability for falls in community-dwelling older adults using the timed Up & Go test," *Phys. Therapy*, vol. 80, no. 9, pp. 896–903, Sep. 2000.

[47] M. M. Hoehn and M. D. Yahr, "Parkinsonism: Onset, progression, and mortality," *Neurology*, vol. 17, no. 5, pp. 427–442, May 1967.

[48] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach," *Biometrics*, vol. 44, no. 3, pp. 837–845, Sep. 1988.

[49] X. Robin *et al.*, "pROC: An open-source package for R and S+ to analyze and compare ROC curves," *BMC Bioinformatics*, vol. 12, no. 1, p. 77, Mar. 2011.

[50] Y. Peng, X. He, and JŻhao, "Object-part attention model for fine-grained image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1487–1500, Mar. 2018.

[51] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.

[52] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 156–165, Jul. 2017.

[53] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 33–44.

[54] R. Sun, Z. Wang, K. E. Martens, and S. Lewis, "Convolutional 3D attention network for video based freezing of gait recognition," in *Proc. IEEE Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Dec. 2018, pp. 1–7.