Symbiotic Graph Neural Networks for 3D Skeleton-Based Human Action Recognition and Motion Prediction

Maosen Li, *Student Member, IEEE*, Siheng Chen, *Member, IEEE*, Xu Chen, Ya Zhang, *Member, IEEE*, Yanfeng Wang, and Qi Tian, *Fellow, IEEE*

Abstract—3D skeleton-based action recognition and motion prediction are two essential problems of human activity understanding. In many previous works: 1) they studied two tasks separately, neglecting internal correlations; and 2) they did not capture sufficient relations inside the body. To address these issues, we propose a symbiotic model to handle two tasks jointly; and we propose two scales of graphs to explicitly capture relations among body-joints and body-parts. Together, we propose symbiotic graph neural networks, which contain a backbone, an action-recognition head, and a motion-prediction head. Two heads are trained jointly and enhance each other. For the backbone, we propose multi-branch multiscale graph convolution networks to extract spatial and temporal features. The multiscale graph convolution networks are based on joint-scale and part-scale graphs. The joint-scale graphs contain actional graphs, capturing action-based relations, and structural graphs, capturing physical constraints. The part-scale graphs integrate body-joints to form specific parts, representing high-level relations. Moreover, dual bone-based graphs and networks are proposed to learn complementary features. We conduct extensive experiments for skeleton-based action recognition and motion prediction with four datasets, NTU-RGB+D, Kinetics, Human3.6M, and CMU Mocap. Experiments show that our symbiotic graph neural networks achieve better performances on both tasks compared to the state-of-the-art methods.

Index Terms—3D skeleton-based action recognition, motion prediction, multiscale graph convolution networks, graph inference

1 Introduction

Human action recognition and motion prediction are crucial problems in computer vision, being widely applicable to surveillance [2], pedestrian tracking [3], and human-machine interaction [4]. Respectively, action recognition aims to classify the categories of query actions accurately [5]; and motion prediction forecasts the future movements based on observations [6].

The data of actions can be represented in various formats, including RGB videos [7] and 3D skeleton data [8]. Notably, skeleton data, locating 3D body-joints, is shown to be effective in action representation, efficient in computation, as well as robust against environmental noise [9]. In this work, we focus on action recognition and motion prediction based on the 3D skeleton data.

In most studies, 3D skeleton-based action recognition and motion prediction are treated separately due to the discriminative and generative properties of the two tasks. For action recognition, methods employed full action sequences for pattern learning [10], [11], [12]; however, with the long-term inputs, these methods failed in some real-time applications due to the hysteretic discrimination, while the model should respond as early as possible. As for motion prediction, previous works [13], [14], [15], [16] learned shallow dynamics, but often ignored semantics. Actually, there are mutual promotions between action recognition and motion prediction, which were rarely explored. For example, the classifier provides the action categories as the auxiliary information to guide prediction, as well as the predictor preserves more detailed information for accurate recognition via self-supervision. Considering the mutual promotions, we aim to develop a symbiotic method to enable action recognition and motion prediction simultaneously.

For both 3D skeleton-based action recognition and motion prediction, previous studies have been conducted. Concretely, some traditional attempts built hand-crafted models for feature learning [10], [13], [17], [18], [19]. Recently, some deep models based on either convolutional neural networks (CNN) or recurrent neural networks (RNN) learned high-level features from the vectorized poses [11], [14], [15], [16], [20], [21], [22], [23], [24], [25], [26]; however, these methods rarely investigated the joint relations, missing crucial dynamics. To capture richer features, several works exploited joint relations from various aspects. [12] proposed skeleton-graphs with nodes as joints and edges as bones. [10], [14], [27], [28], [29], [30] built the relations between coarser body-parts. These works essentially aggregated information based on body structures, while the

- Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, and Yanfeng Wang are with the Cooperative Medianet Innovation Center and the Shanghai Key Laboratory of Multimedia Processing and Transmissions, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {maosen_li, sihengc, xuchen2016, ya_zhang, wangyanfeng}@sjtu.edu.cn.
- Qi Tian is with Huawei Cloud & AI, Shenzhen 518129, China and also with the Department of Computer Science, University of Texas at San Antonio (UTSA), San Antonio, TX 78249 USA. E-mail: wywqtian@gmail.com.

Manuscript received 19 Aug. 2019; revised 19 Dec. 2020; accepted 7 Jan. 2021. Date of publication 22 Jan. 2021; date of current version 5 May 2022.
(Corresponding author: Maosen Li.)
Recommended for acceptance by Y. Liu.
Digital Object Identifier no. 10.1109/TPAMI.2021.3053765

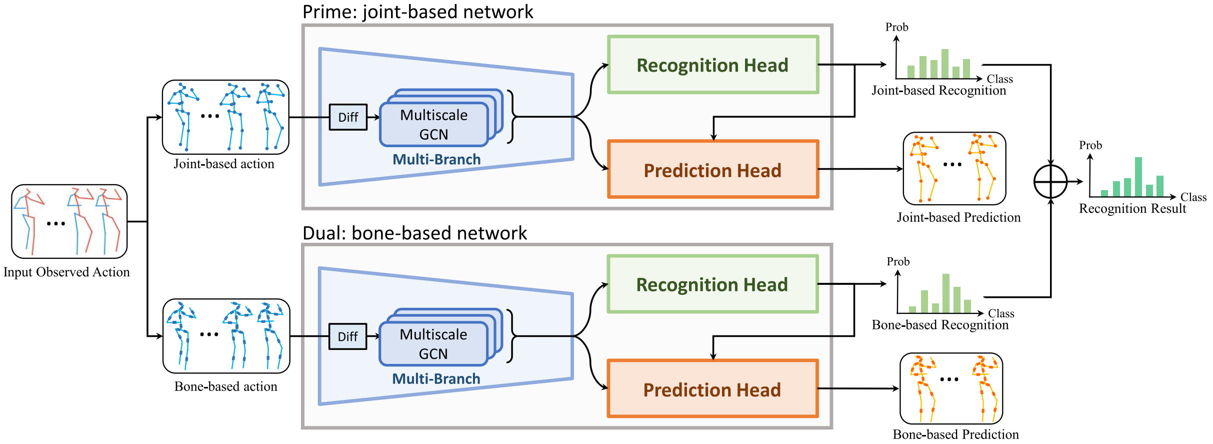


Fig. 1. Symbiotic graph neural networks (Sybio-GNN) contains a prime joint-based network and a dual bone-based network to learn both joint-based and bone-based action features. Each network has three main modules: a backbone, an action-recognition head, and a motion-prediction head. The backbone is multi-branch multiscale graph convolution networks. The action-recognition head and the motion-prediction head predict the action category and future poses. The predicted category further assists in motion prediction.

neglected some implicit relations over action-related joints, such as hands and feet moving collaboratively during walking. Moreover, some methods of motion prediction fed ground-truth action categories to enhance performance in both training and testing phases, but labels are hard to obtain in the real-world scenarios. To solve those issues, we construct graphs to model both local and long-range body relations and use graph convolutions to capture informative features.

In this paper, we propose a novel model called *symbiotic graph neural network* (Sybio-GNN), which handles 3D skeleton-based action recognition and motion prediction simultaneously and uses graph-based operations to extract features; see Fig. 1. Sybio-GNN consists of a joint-based network and a bone-based network, which focuses on learning features from body-joints and body-bones to obtain complementary patterns. Each network is constructed with a backbone, called *multi-branch multiscale graph convolutional network* (multi-branch multiscale GCN), an action-recognition head and a motion-prediction head, where two heads work on various tasks. Note that there are task promotions, i.e., the action-recognition head determines action categories, which is used to enhance prediction; the motion-prediction head predicts poses and improves recognition by promoting self-supervision and preserving detailed features.

As basic operators of the backbone, we propose the *joint-scale graph convolution* (JGC) and *part-scale graph convolution* (PGC) to extract multiscale spatial information. JGC is based on two types of graphs: actional graphs and structural graphs. The actional graphs are learned from data by an *actional graph inference module* (AGIM), capturing action-based relations; the structural graphs are built by extending the skeleton graphs, capturing physical constraints. PGC is based on a part-scale graph, whose nodes are integrated body-parts and edges are based on body-part connections. We also propose a difference operator to extract multiple orders of motion differences, reflecting positions, velocities, and accelerations of body-joints.

To validate the Sybio-GNN, we conduct extensive experiments on four large-scale datasets: NTU-RGB+D [31], Kinetics [12], Human3.6M [32], and CMU Mocap.[1] The results show that 1) Sybio-GNN outperforms the state-of-the-art methods in both action recognition and motion prediction; 2) Using the symbiotic model to train the two tasks simultaneously produces better performance than using individual models; and 3) the multiscale graphs model complicated relations between body-joints and body-parts, and the proposed JGC extract informative spatial features.

Overall, the main contributions in this paper are summarized as follows:

- *Multitasking framework*. We propose novel *symbiotic graph neural networks* (Sybio-GNN) to achieve 3D skeleton-based action recognition and motion prediction in a multitasking framework. Sybio-GNN contains a backbone, an action-recognition head, and a motion-prediction head. We exploit the mutual promotion between two heads, leading to improvements in both tasks; see Section 5;
- *Basic operators*. We propose novel operators to extract information from 3D skeleton data: 1) a *joint-scale graph convolution* operator extracts joint-level spatial features based on both actional and structral graphs; see Section 4.1.4; 2) a *part-scale graph convolution* operator extracts part-level spatial features based on part-scale graphs; see Section 4.2.1; 3) a pair of *bidirectional fusion* operators fuses information across two scales; see Section 5.1.2 and 4) a *difference* operator extracts temporal features; see Section 4.3;
- *Experimental findings*. We conduct extensive experiments for both tasks of 3D skeleton-based action recognition and motion prediction. The results show that Sybio-GNN outperforms the state-of-the-art methods in both tasks; see Section 6.

2 Related Works

3D Skeleton-Based Action Recognition. For 3D skeleton-based action recognition, conventionally, some models learned semantics based on hand-crafted features and physical intuitions [10], [17], [18], [33]. With the developed deep learning methods, some recurrent-neural-network-based (RNN-based) models captured temporal dependencies along

[1] http://mocap.cs.cmu.edu/.

consecutive frames [11], [34]. Moreover, convolutional neural networks (CNN) also achieve remarkable results [21], [22]. Recently, the graph-based approaches drew many attentions [1], [12], [27], [28], [30], [35], [36], [37], [38]. In this work, we adopt the graph-based approach. We construct multiscale graphs adaptively from data, capturing useful and comprehensive information about actions.

3D Skeleton-Based Motion Prediction. In earlier studies, state models were considered to predict future motions [13], [19], [39]. Recently, some RNN-based methods learned the dynamics from sequences [14], [15], [25], [40], [41], [42]. Moreover, adversarial mechanics and geodesic loss could further improve predictions [16]. As for our method, we use graph structures to explicitly model the relations between body-joints and body-parts, guiding the networks to learn local and non-local motion patterns.

Graph Deep Learning. Graphs, focused on by many recent studies, are effective to express data associated with non-grid structures [12], [27], [43], [44], [45], [46], [47], [48]. Given the fixed topologies, previous works explored to propagate node features based on the spectral domain [46], [47] or the vertex domain [48]. [1], [12], [28], [35], [36] leveraged graph convolution for 3D skeleton-based action recognition. [14] also considered the skeleton-based relations for motion prediction. In this paper, we propose multiscale graphs to represent multiple relations: joint-scale and part-scale relations. Then, we propose novel graph convolution operators to extract deep features for action recognition and motion prediction. Different from [1] obtaining multiple actional graphs with complicated inference processes, our method employs more efficient graph learning operations.

3 Problem Formulation

In this paper, we study 3D skeleton-based action recognition and motion prediction jointly. Let the action pose at timestamp t be $\mathbf{X}^{(t)} \in \mathbb{R}^{M \times D_{\mathbf{x}}}$, where $t > 0$ indicates the future frames, otherwise the observations; notably, $t = 0$ denotes the current frame. M is the number of joints and $D_{\mathbf{x}} = 3$ reflects the 3D joint positions. The action pose is essentially associated with a skeleton graph built with pairwise bone connectivity. Let the self-looped adjacency matrix of a skeleton graph be $\mathbf{A} \in \{0,1\}^{M \times M}$, where the (i,j)th elements $(\mathbf{A})_{ij}= 1$ when the ith and the jth body-joints are connected with bones, and $(\mathbf{A})_{ij}= 0$, otherwise.

For an action sequence belonging to one class, we have $\{\mathcal{X}_{\rm prev}, \mathcal{X}_{\rm pred}, \mathbf{y}\}$, where $\mathcal{X}_{\rm prev} = [\mathbf{X}^{(-T_{\rm prev})}, \ldots, \mathbf{X}^{(0)}] \in \mathbb{R}^{(T_{\rm prev}+1) \times M \times D_{\mathbf{x}}}$ denotes the previous motion tensor; $\mathcal{X}_{\rm pred} = [\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(T_{\rm pred})}] \in \mathbb{R}^{T_{\rm pred} \times M \times D_{\mathbf{x}}}$ denotes the future motion tensor; $T_{\rm prev}$ and $T_{\rm pred}$ are the frame numbers of previous and future motions, respectively; and one-hot vector $\mathbf{y} \in \{0,1\}^C$ denotes the class-label in C possible classes. Let $\mathcal{F}(\cdot)$ be the overall model. The discriminated class $\hat{\mathbf{y}}$ and the predicted motion $\hat{\mathcal{X}}_{\rm pred}$ are formulated as

$$\hat{\mathbf{y}}, \hat{\mathcal{X}}_{\rm pred} = \mathcal{F}(\mathcal{X}_{\rm prev}; \boldsymbol{\theta}_{\rm bk}, \boldsymbol{\theta}_{recg}, \boldsymbol{\theta}_{\rm pred}),$$

where $\boldsymbol{\theta}_{\rm bk}$, $\boldsymbol{\theta}_{recg}$ and $\boldsymbol{\theta}_{\rm pred}$ denote trainable parameters of the backbone, the action-recognition head and the motion-prediction head, respectively.

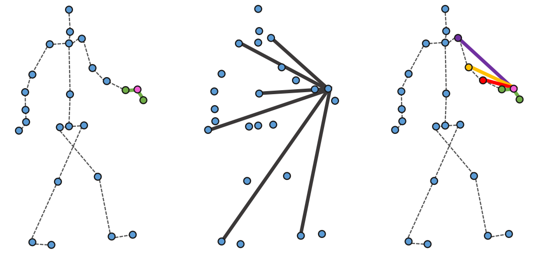


Fig. 2. Joint-scale graphs for walking. We consider an actional graph (b) and a structural graph (c), which is an extension of a skeleton graph (a). In each graph, the edges from ”Left Hand” to its neighbors are shown in solid lines and other links in the skeleton are shown in dashed lines.

4 Basic Components

In this paper, we propose a novel *Symbiotic Graph Neural Network* (Sybio-GNN), which simultaneously performs skeleton-based human action recognition and motion prediction. Before constructing the entire Sybio-GNN, We propose joint-scale graph operators and part-scale operators that are leveraged in the encoder of *multiscale GCNs* (see Fig. 1). The functionalities of these two operators are to learn body graphs at multiple scales and extract comprehensive motion features on each scale. Then, we propose a difference operator, which performs on the original human pose data and is located at the input of the entire model, providing richer motion priors for the subsequent learning modules.

4.1 Joint-Scale Graph Operators in Multiscale GCN

To model the joint relations, we build joint-scale graphs including actional graphs and structural graphs, which capture moving interactions and physical constraints between joints. Fig. 2 sketches some examples: (a) a skeleton graph with the local neighborhood; (b) an actional graph about action-based dependencies; (c) a structural graph, which allows ‘Left Hand’ to link with the entire arm. As follows, we propose the construction of joint-scale actional and structural graphs with the relevant operations.

4.1.1 Actional Graph Convolution

For different actions, some structurally distant joints may interact with various manners, leading to distinctive action-based relations. To represent these relations, we employ an actional graph: $\mathcal{G}_{\rm act}(V, \mathbf{A}_{\rm act})$, where $V = \{v_1, \ldots, v_M\}$ is the joint set and $\mathbf{A}_{\rm act} \in \mathbb{R}^{M \times M}$ is the adjacency matrix. To obtain the graph topology, we propose a data-adaptive module, called *actional graphs inference module* (AGIM), to learn $\mathbf{A}_{\rm act}$ purely from observed motions.

In AGIM, we let the vector representation of the ith joint positions across all observed frames be $\mathbf{x}_i = {\rm vec}(\mathcal{X}_{\rm prev}[:, i, :]) \in \mathbb{R}^{(D_{\mathbf{x}} T_{\rm prev})}$. To learn the relations, we propagate pose information between body-joints and possible edges. We first initialize $\mathbf{p}_i^{\langle 0 \rangle} = f_{\rm v}^{\langle 0 \rangle}(\mathbf{x}_i) \in \mathbb{R}^{D_{\rm v}}$, where $f_{\rm v}^{\langle 0 \rangle}(\cdot)$ is a multilayer perceptron (MLP) that maps the moving joints $\mathbf{x}_i$ to features $\mathbf{p}_i^{\langle 0 \rangle}$. In the kth updating iteration, the features are propagated as follows:

$$\mathbf{q}_{i,j}^{\langle k \rangle} = f_{\rm e}^{\langle k \rangle}\left(\left[\mathbf{p}_i^{\langle k-1 \rangle}, \mathbf{p}_j^{\langle k-1 \rangle}\right]\right) \in \mathbb{R}^{D_{\rm e}}, \quad (1a)$$

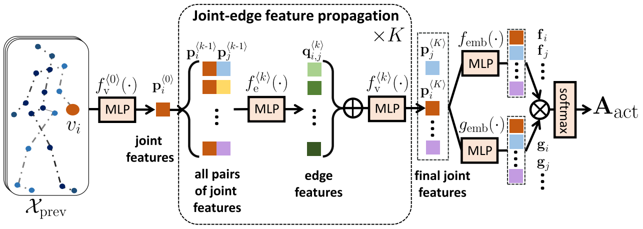

Joint-edge feature propagation
$\times K$
$\mathcal{X}_{\text{prev}}$
v_i
$f_{\text{v}}^{\langle 0\rangle}(\cdot)$
MLP
$\mathbf{p}_i^{\langle 0\rangle}$
joint features
$\mathbf{p}_i^{\langle k\text{-}1\rangle}$ $\mathbf{p}_j^{\langle k\text{-}1\rangle}$
all pairs of joint features
$f_{\text{e}}^{\langle k\rangle}(\cdot)$
MLP
$\mathbf{q}_{i,j}^{\langle k\rangle}$
edge features
$f_{\text{v}}^{\langle k\rangle}(\cdot)$
MLP
$\mathbf{p}_j^{\langle K\rangle}$
$\mathbf{p}_i^{\langle K\rangle}$
final joint features
$f_{\text{emb}}(\cdot)$
MLP
$g_{\text{emb}}(\cdot)$
MLP
$\mathbf{f}_i$
$\mathbf{f}_j$
$\mathbf{g}_i$
$\mathbf{g}_j$
softmax
$\mathbf{A}_{\text{act}}$

Fig. 3. Actional graphs inference module (AGIM) propagates features between joints and edges for K iterations and uses correlations between joint features to obtain actional graphs. In the box 'Joint-edge feature propagation $\times K$', any two joint features $\mathbf{p}_i^{\langle k-1\rangle}$ and $\mathbf{p}_j^{\langle k-1\rangle}$ are concatenated and fed in to an MLP, $f_{\text{e}}^{\langle k\rangle}(\cdot)$, corresponding to equation (1a). The edge features associated with the ith joint, $\mathbf{q}_{i,j}^{\langle k\rangle}$ are summed and fed into an MLP, $f_{\text{v}}^{\langle k\rangle}(\cdot)$, corresponding to Eq. (1b). The aggregated features mapped by two MLPs, $f_{\text{emb}}(\cdot)$ and $g_{\text{emb}}(\cdot)$ are used to calculate the adjacency matrix of the actional graph, corresponding to equation (2).

$$\mathbf{p}_i^{\langle k\rangle} = f_{\text{v}}^{\langle k\rangle}\left(\frac{1}{M-1}\sum_{v_j\in\mathcal{V},j\neq i}\mathbf{q}_{i,j}^{\langle k\rangle}\right) \in \mathbb{R}^{D_{\text{v}}}, \tag{1b}$$

where $\mathbf{p}_i^{\langle k\rangle}$, $\mathbf{q}_{i,j}^{\langle k\rangle}$ are the features of the ith joint v_i and the edge connecting v_i and v_j at the kth iteration; $f_{\text{e}}^{\langle k\rangle}(\cdot)$ and $f_{\text{v}}^{\langle k\rangle}(\cdot)$ are MLPs; $[\cdot,\cdot]$ is the concatenation. Eq. (1a) maps a pair of joint features to the in-between edge features; Eq. (1b) aggregates all edge features associated with the same joint to update the joint. After K iterations, each joint feature $\mathbf{p}^{\langle K\rangle}$ has integrated information in a long-range.

With the aggregated feature $\mathbf{p}^{\langle K\rangle}$ of any joint, we compute the affinity of each pair of joints, leading to an action-based relation. We build two individual embedding networks, $f_{\text{emb}}(\cdot)$ and $g_{\text{emb}}(\cdot)$, to further learn the high-level representations of joints. The (i,j)th element of the adjacency matrix of actional graph is

$$(\mathbf{A}_{\text{act}})_{i,j} = \frac{\exp(\mathbf{f}_i^{\text{T}}\mathbf{g}_j)}{\sum_{k=1}^{M}\exp(\mathbf{f}_i^{\text{T}}\mathbf{g}_k)} \in [0,1], \tag{2}$$

where $\mathbf{f}_i = f_{\text{emb}}(\mathbf{p}_i^{\langle K\rangle})$ and $\mathbf{g}_i = g_{\text{emb}}(\mathbf{p}_i^{\langle K\rangle}) \in \mathbb{R}^{D_{\text{emb}}}$ are the two different embeddings of joint v_i. Notably, $(\mathbf{A}_{\text{act}})_{i,j} \neq (\mathbf{A}_{\text{act}})_{j,i}$, indicating incoming and outgoing relations between joints. Eq. (2) uses the softmax to normalize the edge weights and promote a few large ones. The structure of AGIM is illustrated in Fig. 3.

Based on the inferred $\mathbf{A}_{\text{act}}$, we design an actional graph convolution (AGC) to capture the actional features. Mathematically, let the input and output features at frame t be $\mathbf{X}^{(t)} \in \mathbb{R}^{M\times D_{\mathbf{x}}}$ and $\mathbf{Y}_{\text{AGC}}^{(t)} \in \mathbb{R}^{M\times D_{\mathbf{y}}}$, the AGC works as

$$\mathbf{Y}_{\text{AGC}}^{(t)} = \text{AGC}(\mathbf{X}^{(t)}) = \mathbf{A}_{\text{act}}\mathbf{X}^{(t)}\mathbf{W}_{\text{act}}^{\top}, \tag{3}$$

where $\mathbf{W}_{\text{act}} \in \mathbb{R}^{D_{\mathbf{y}}\times D_{\mathbf{x}}}$ is a trainable weight. Therefore, the model aggregates the action-based information from collaboratively moving joints even in the distance.

4.1.2 Structural Graph Convolution

Intuitively, the joint dynamics is limited due to physical constraints, namely bone connections. To capture these relations, we develop a structural graph, $\mathcal{G}_{\text{str}}(V, \mathbf{A}_{\text{str}})$. According to the skeleton structure, we first normalize the skeleton graph adjacency matrix, $\mathbf{A}$, by

$$\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A},$$

where $\mathbf{D} \in \mathbb{N}^{M\times M}$ is a diagonal degree matrix of $\mathbf{A}$. $\tilde{\mathbf{A}}$ provides nice initialization to learn the edge weights and avoids multiplication explosion [49], [50].

Notably, $\tilde{\mathbf{A}}$ only describes the 1-hop neighborhood based on bone-connections. To represent long-range relations, we use the high-order polynomials of $\tilde{\mathbf{A}}$. Let the γ-order polynomial of $\tilde{\mathbf{A}}$ be $\tilde{\mathbf{A}}^{\gamma}$, which indicates the relations between each joint and its γ-hop neighbors on skeleton. Then, we introduce several individual edge-weight matrices $\mathbf{M}^{(\gamma)} \in \mathbb{R}^{M\times M}$ corresponding to $\tilde{\mathbf{A}}^{\gamma}$, whose elements are trainable to reflect the relation strength. We finally obtain the γ-order structural graph adjacency matrix

$$\mathbf{A}_{\text{str}}^{(\gamma)} = \tilde{\mathbf{A}}^{\gamma} \odot \mathbf{M}^{(\gamma)} \in \mathbb{R}^{M\times M},$$

where $\odot$ denotes the element-wise product. In this way, we model the structure-based relations between joints in relatively longer range. We consider $\gamma = 0, 1, \ldots, \Gamma$, thus we have multiple structural graphs for one body. See plot (c) in Fig. 2, the hand is correlated with the entire arm. Note that, when we set $\Gamma = 1$, the structural graph achieves the 'distance partitioning' proposed by [12]; as for $\Gamma > 1$, some related works like [51] also consider the high-order structural graphs for feature extraction.

Given the computed $\mathbf{A}_{\text{str}}^{(\gamma)}$, we propose the structural graph convolution (SGC) operator. Let the input and output features at frame t be $\mathbf{X}^{(t)} \in \mathbb{R}^{M\times D_{\mathbf{x}}}$ and $\mathbf{Y}_{\text{SGC}}^{(t)} \in \mathbb{R}^{M\times D_{\mathbf{y}}}$, the SGC operator is formulated as

$$\mathbf{Y}_{\text{SGC}}^{(t)} = \text{SGC}(\mathbf{X}^{(t)}) = \sum_{\gamma=1}^{\Gamma}\mathbf{A}_{str}^{(\gamma)}\mathbf{X}^{(t)}\mathbf{W}_{str}^{(\gamma)\top}, \tag{4}$$

where $\mathbf{W}_{\text{str}}^{(\gamma)} \in \mathbb{R}^{D_{\mathbf{y}}\times D_{\mathbf{x}}}$ is the trainable weights. Notably, the multiple structural graphs have different corresponding weights, which help to extract richer features.

4.1.3 Joint-Scale Graph Convolution

Based on AGC (Eq. (3)) and SGC (Eq. (4)), we present the joint-scale graph convolution (JGC) to capture comprehensive joint-scale spatial features. Mathematically, let the input joint features at frame t be $\mathbf{X}^{(t)} \in \mathbb{R}^{M\times D_{\mathbf{x}}}$, the output features be $\mathbf{Y}^{(t)} \in \mathbb{R}^{M\times D_{\mathbf{y}}}$, the JGC is formulated as

$$\mathbf{Y}^{(t)} = \rho(\text{JGC}(\mathbf{X}^{(t)})) = \rho(\lambda_{\text{act}}\text{AGC}(\mathbf{X}^{(t)}) + \text{SGC}(\mathbf{X}^{(t)})), \tag{5}$$

where λ_{act} is a hyper-parameter to trade off the contribution between actional and structural features; $\rho(\cdot)$ denotes the nonlinear ReLU function. In this way, the joint features are effectively aggregated to update each center joint according to the joint-scale graphs.

We further show the stability of the proposed activated joint-scale graph convolution layer; that is, when input 3D skeleton data is disturbed, the distortion of the output features is upper bounded.

Theorem 1 (Stability). *Let two joint-scale feature matrices be* $\mathbf{X}$ *and* $\mathbf{X}^{*} \in \mathbb{R}^{M\times D_{\mathbf{x}}}$ *associated with a skeleton graph* $\mathbf{A} \in \{0,1\}^{M\times M}$, *where* $D_{\mathbf{x}} = 3$ *and* $\|\mathbf{X}^{*} - \mathbf{X}\|_F \leq \epsilon$ $(\epsilon \geq 0)$.

Let $\mathbf{Y} = \rho(\mathrm{JGC}(\mathbf{X}))$ *and* $\mathbf{Y}^* = \rho(\mathrm{JGC}(\mathbf{X}^*)) \in \mathbb{R}^{M\times D_{\mathbf{y}}}$. *Let* $\mathbf{A}^*_{\mathrm{act}}$ *and* $\mathbf{A}_{\mathrm{act}} \in [0,1]^{M\times M}$ *be the joint-scale actional graph inferred from* $\mathbf{X}^*$ *and* $\mathbf{X}$, *respectively, where*

$$\left\|\mathbf{A}^*_{\mathrm{act}}\mathbf{X}^* - \mathbf{A}_{\mathrm{act}}\mathbf{X}\right\|_F \leq C\epsilon^q,$$

with q *the amplify factor and* C *some constant. Let* $\mu_{\mathrm{act}} = \|\mathbf{W}_{\mathrm{act}}\|_{\max}$, $\eta^{(\gamma)} = \|\mathbf{M}^{(\gamma)}\|_{\max}$ *and* $\mu^{(\gamma)}_{\mathrm{str}} = \|\mathbf{W}^{(\gamma)}_{\mathrm{str}}\|_{\max}$, *where* $\mathbf{W}_{\mathrm{act}}, \mathbf{W}^{(\gamma)}_{\mathrm{str}} \in \mathbb{R}^{D_{\mathbf{y}}\times D_{\mathbf{x}}}$ *and* $\mathbf{M}^{(\gamma)} \in \mathbb{R}^{M\times M}$. *Then*

$$\|\mathbf{Y}^* - \mathbf{Y}\|_F \leq \sqrt{3D_{\mathbf{y}}}\left(\epsilon^q\lambda_{\mathrm{act}}\mu_{\mathrm{act}}C + \epsilon\sum_{\gamma=1}^{\Gamma}\sqrt{\|\mathbf{A}^{\gamma}\|_0}\eta^{(\gamma)}\mu^{(\gamma)}_{\mathrm{str}}\right)$$

$$= O(\max(\epsilon^q, \epsilon)).$$

Note that $\|\cdot\|_F$ *denotes Frobenius norm and* $\|\cdot\|_0$ *is zero norm.* $\rho(\cdot)$ *denotes ReLU-activation function.* $O(\cdot)$ *denotes the effects that rely on* '$\max(\epsilon^q, \epsilon)$'.

See the proof in Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3053765. Theorem 1 shows the joint-scale graph convolution at the first layer, but the bound can be extended to the subsequent layers. Theorem 1 shows that 1) the outputs of JGC followed by the activation function can be upper bounded, reflecting its robustness against input perturbation; and 2) given a fixed model, the bound is mainly related to the amplify factor q. The JGC's robustness ensures the stability of Sybio-GNN given small noises.

4.1.4 Joint-Scale Graph and Temporal Convolution Block

While the JGC leverages the joint spatial relations and extracts rich features, we should consider modeling the temporal dependencies among consecutive frames. We develop the temporal convolution operator (TC); that is, a convolution along time to learn the movements. Stacking JGC and TC, we build the *joint-scale graph and temporal convolution block* (J-GTC block), which learn the spatial and temporal features in tandem. Mathematically, let $\mathcal{X}_{\mathrm{in}} \in \mathbb{R}^{T\times M\times D_{\mathbf{x}}}$ be an input tensor, each J-GTC block works as

$$(\mathcal{X}')_{[t,:,:]} = \rho\Big(\mathrm{JGC}((\mathcal{X}_{\mathrm{in}})_{[t,:,:]})\Big) \in \mathbb{R}^{M\times D_{\mathbf{x}'}}, \tag{6a}$$

$$\mathcal{X}_{\mathrm{out}} = \rho(\mathrm{TC}(\mathcal{X}'; \tau, s)) \in \mathbb{R}^{T/s\times M\times (sD_{\mathbf{x}'})}. \tag{6b}$$

$\mathrm{T}C(\cdot)$ is a standard 1D convolution along the time axis, whose temporal kernel size is τ; s is the convolution stride along time to shrink the temporal dimension; t slice the pose at the corresponding timestamp. In each J-GTC block, Eq. (6a) extracts spatial features of joints; and Eq. (6b) extracts temporal features by aggregating the consecutive frames. Our J-GTC block also includes batch normalization and dropout operations. Moreover, there is a residual connection preserving the input features. The architecture of one J-GTC block is illustrated in Fig. 4. By stacking several J-GTC blocks in a hierarchy, we gradually convert the motion dynamics from the sample space to the feature space.

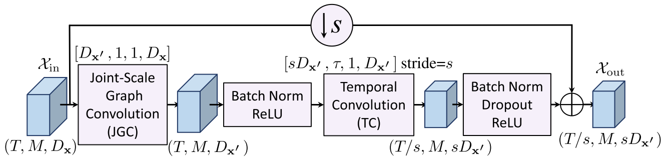

$\downarrow s$
$\mathcal{X}_{\mathrm{in}}$
$[D_{\mathbf{x}'}, 1, 1, D_{\mathbf{x}}]$
Joint-Scale Graph Convolution (JGC)
Batch Norm ReLU
$[sD_{\mathbf{x}'}, \tau, 1, D_{\mathbf{x}'}]$ stride=s
Temporal Convolution (TC)
Batch Norm Dropout ReLU
$\mathcal{X}_{\mathrm{out}}$
$(T, M, D_{\mathbf{x}})$
$(T, M, D_{\mathbf{x}'})$
$(T/s, M, sD_{\mathbf{x}'})$
$(T/s, M, sD_{\mathbf{x}'})$

Fig. 4. J-GTC block consists of JGC [equation (5)] and temporal convolution (TC). The triples below the blocks denote the tensor shapes. The quaternions are the shapes of parameters in JGC and TC operators. The first two pink boxes represent the joint-scale graph convolution, which is corresponding to equation (6a). The second two pink boxes denotes the temporal convolution, which is corresponding to equation (6b).

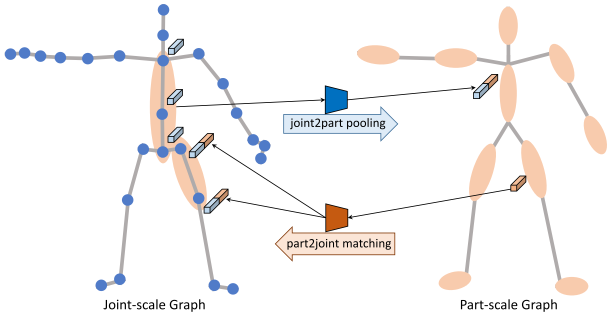


Fig. 5. A joint-scale graph consists of body-joints represented as blue nodes and a part-scale graph consists of body-parts represented as orange nodes. The bidirectional fusion converts features across two scales through the operations of joint2part pooling and part2joint matching. We only plot the one-hop structural graph for the joint-scale graph.

4.2 Part-Scale Graph Operators in Multiscale GCN

The joint-scale graphs treat body-joints as nodes and model their relations, but some action patterns depend on more abstract movements of body-parts. For example, 'hand waving' shows a rising arm, but the finger and wrist are less important. To model the part dynamics, we propose the part-scale graph and temporal convolution (P-GTC) block to extract part-scale features.

4.2.1 Part-Scale Graph Convolution

For a part-scale graph, we define $M_{\mathrm{p}} = 10$ body-parts as graph nodes: 'head', 'torso', pairs of 'upper arms', 'forearms', 'thighs' and 'crura', which integrates the covered joints on joint-scale body. And we build the edges according to body nature; see the right plot of Fig. 5. The self-looped adjacency matrix of the part-scale graph is defined as $\mathbf{A}_{\mathrm{p}} \in \{0,1\}^{M_{\mathrm{p}}\times M_{\mathrm{p}}}$, and we preprocess $\mathbf{A}_{\mathrm{p}}$ by

$$\mathbf{A}_{\mathrm{part}} = (\mathbf{D}_{\mathrm{p}}^{-1}\mathbf{A}_{\mathrm{p}}) \odot \mathbf{M}_{\mathrm{p}} \in \mathbb{R}^{M_{\mathrm{p}}\times M_{\mathrm{p}}},$$

where $\mathbf{D}_{\mathrm{p}} \in \mathbb{N}^{M_{\mathrm{p}}\times M_{\mathrm{p}}}$ is the diagonal degree matrix of $\mathbf{A}_{\mathrm{p}}$; $\mathbf{M}_{\mathrm{p}} \in \mathbb{R}^{M_{\mathrm{p}}\times M_{\mathrm{p}}}$ is a trainable weight matrix and $\odot$ is the element-wise product.

Similarly to the JGC operator (Eq. (5)), we propose the part-scale graph convolution (PGC) for spatial feature learning. Let the part features at time t be $\mathbf{X}^{(t)}_{\mathrm{p}} \in \mathbb{R}^{M_{\mathrm{p}}\times D_{\mathbf{x}}}$, the output features be $\mathbf{Y}^{(t)}_{\mathrm{PGC}} \in \mathbb{R}^{M\times D_{\mathbf{y}}}$, the PGC works as

$$\mathbf{Y}^{(t)}_{\mathrm{p}} = \mathrm{PGC}(\mathbf{X}^{(t)}_{\mathrm{p}}) = \mathbf{A}_{\mathrm{part}}\mathbf{X}^{(t)}_{\mathrm{p}}\mathbf{W}^{\top}_{\mathrm{part}}, \tag{7}$$

where $\mathbf{W}^{\top}_{\mathrm{part}}$ is the trainable parameters. With Eq. (7), we propagate information between body-parts, leading to abstract spatial patterns. Notably, We do not need a part-scale

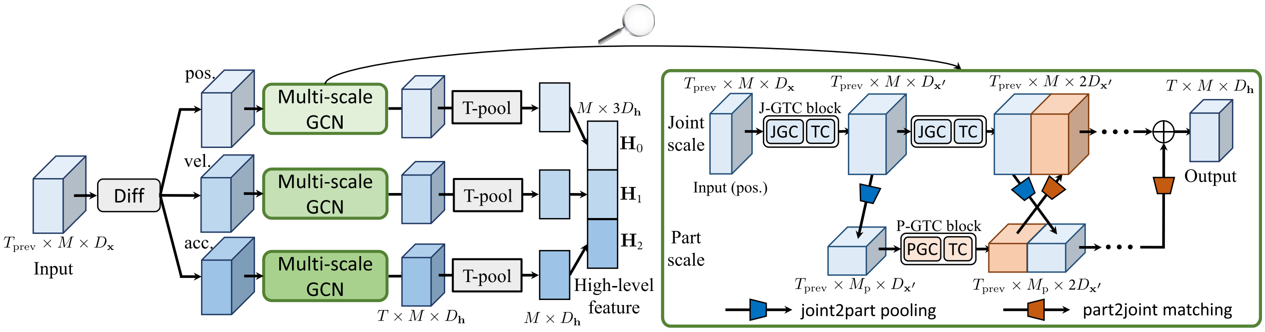

pos.
vel.
acc.
Diff
$T_{\mathrm{prev}} \times M \times D_{\mathbf{x}}$
Input
Multi-scale GCN
Multi-scale GCN
Multi-scale GCN
T-pool
T-pool
T-pool
$T \times M \times D_{\mathbf{h}}$
$M \times D_{\mathbf{h}}$
$M \times 3D_{\mathbf{h}}$
$\mathbf{H}_0$
$\mathbf{H}_1$
$\mathbf{H}_2$
High-level feature
$T_{\mathrm{prev}} \times M \times D_{\mathbf{x}}$
$T_{\mathrm{prev}} \times M \times D_{\mathbf{x}'}$
$T_{\mathrm{prev}} \times M \times 2D_{\mathbf{x}'}$
$T \times M \times D_{\mathbf{h}}$
Joint scale
J-GTC block
JGC
TC
JGC
TC
Input (pos.)
Output
Part scale
P-GTC block
PGC
TC
$T_{\mathrm{prev}} \times M_{\mathrm{p}} \times D_{\mathbf{x}'}$
$T_{\mathrm{prev}} \times M_{\mathrm{p}} \times 2D_{\mathbf{x}'}$
joint2part pooling
part2joint matching

Fig. 6. Backbone is essentially multi-branch multiscale graph convolution networks. It uses three individual multiscale GCNs to extract spatial and temporal features. A difference operator ('Diff') calculate three orders of differences, which represent joint positions ('pos.'), velocities ('vel.') and accelerations ('acc.'). Each multiscale GCN takes one order as input and uses multiple J-GTC, P-GTC blocks and bidirectional fusion to learn spatial and temporal features from two scales.

actional graph, because the part-scale graph includes some integrated relations internally, as well as it has a shorter distance to build long-range links.

4.2.2 *Part-Scale Graph and Temporal Convolution Block*

Considering the temporal evolution, we use the same temporal convolution as in J-GTC block to form the *part-scale graph and temporal convolution* (P-GTC) block . Let the input part feature tensor be $\mathcal{X}_{\mathrm{p,in}} \in \mathbb{R}^{T \times M_{\mathrm{p}} \times D_{\mathbf{x}}}$, we have

$$(\mathcal{X}'_{\mathrm{p}})_{[t,:,:]} = \rho\Big(\mathrm{PGC}((\mathcal{X}_{\mathrm{p,in}})_{[t,:,:]})\Big) \in \mathbb{R}^{M_{\mathrm{p}} \times D_{\mathbf{x}'}}, \tag{8a}$$

$$\mathcal{X}_{\mathrm{p,out}} = \rho\Big(\mathrm{TC}(\mathcal{X}'_{\mathrm{p}}; \tau, s)\Big) \in \mathbb{R}^{T/s \times M_{\mathrm{p}} \times (sD_{\mathbf{x}'})}, \tag{8b}$$

where t slices the pose at the timestamp and s is the temporal convolution stride. Comparing to the J-GTC block, the P-GTC block extracts the spatial and temporal features of actions in a higher scale. Given the part-scale features as the additional information, the model extracts more comprehensive features for pattern capturing. The effectiveness of leveraging the part-scale feature also has been verified by [10], [11], [27], [28], [29], [30], [31], [33], [52], [53], [54].

4.3 Difference Operator on Input Motions

Intuitively, the states of motion, such as velocity and acceleration, carry important dynamics information and make it easier to extract spatial-temporal features. To achieve this, we employ a difference operator to preprocess the input sequences. The idea is to compute high-order differences of the pose sequences. The zero-order difference is $\Delta^0 \mathbf{X}^{(t)} = \mathbf{X}^{(t)} \in \mathbb{R}^{M \times D_{\mathbf{x}}}$, where $\mathbf{X}^{(t)}$ is the pose at the time t, and the β-order difference ($\beta > 0$) of the pose is

$$\Delta^{\beta+1} \mathbf{X}^{(t)} = \Delta^{\beta} \mathbf{X}^{(t)} - \Delta^{\beta} \mathbf{X}^{(t-1)} \in \mathbb{R}^{M \times D_{\mathbf{x}}}, \tag{9}$$

where Δ^{β} denotes the βth-order difference operator. We use zero paddings after diference computation to handle boundary conditions. We take the first three orders ($\beta = 0, 1, 2$) to our model, reflecting positions, velocities, and accelerations. In the model, the three differences can be efficiently computed in parallel. Compared to previous works which only consider the position and velocity information [27], [28], [55], [56], [57], [58], we naturally introduce more detailed accelerations to depict the dynamics of motions.

5 Symbiotic Graph Neural Networks

In this section, we present the architecture of our Symbiotic Graph Neural Networks (Sybio-GNN), which is constructed with a deep backbone and two task-specific modules. We present the deep backbone network, which employs multiscale graphs to learn the high-level features of motions (see Sections 4.1 and 4.2) to provide the downstream tasks with informative representations; We then present the action-recognition head and the motion-prediction head with an effective multitasking scheme. Finally, we present a dual bone-based network to learn complementary features from body bones for downstream tasks.

5.1 Backbone: Multi-Branch Multiscale Graph Convolution Networks

To learn the high-level action pattern, Sybio-GNN consists of a deep backbone called *multi-branch multiscale graph convolution network* (multi-branch multiscale GCN). It employs parallel multiscale GCN branches to treat high-order action differences and also considers multiscale graphs for spatial feature extraction. Fig. 6 shows the backbone, where the left plot is the backbone framework including three branches of multiscale GCNs; the right plot is the structure of each branch. As follows, we propose the backbone in detail.

5.1.1 *Multiple Branches*

The backbone has three branches of multiscale GCN. Each branch uses a distinct order of action differences as input, treating the motion states ('positions', 'velocities' and 'accelerations') for dynamics learning (see Fig. 6). The three branches have identical network architectures. We obtain semantics of high-order differences and concatenate them together for action recognition and motion prediction.

5.1.2 *Multiscale GCN*

To learn the multiscale action features comprehensively, each branch of the backbone is a multiscale GCN based on the *joint-scale* and *part-scale* graphs. Concretely, for the joint-scale graphs, both the actional and structural graphs are used to capture body-joint correlations. We employ a cascade of J-GTC blocks (see Section 4.1.4) based on the joint-scale graphs for feature capturing. With the part-scale graphs whose nodes represent high-level body instances, we stack P-GTC blocks (see Section 4.2.1) for feature capturing. To be aware of the multiscale immediate representations and learn rich and consistent patterns, we introduce a fusion mechanism between the hidden layers of two scales; called *bidirectional fusion*.

Bidirectional Fusion. The bidirectional fusion exchanges features from both the joint scale and the part scale; see illustrations in Figs. 5 and 6. It contains two operations:

- *Joint2part pooling.* For the joint scale, we use pooling to average the joint features on the same part to represent a super node. Then, we concatenate the pooling result to the corresponding part feature in the part scale. As shown in Fig. 5, we average torso joints to obtain a node in the part-scale graph and concatenate it to the original part-scale features.
- *Part2joint matching.* For the part scale, the part features are copied for several times to match the number of corresponding joints, as well as we concatenate the copied parts to the joints. As shown in Fig. 5, we copy the thigh twice and concatenate them to the hip and knee in the joint scale.

Fig. 6 (right plot) shows the operations and one bidirectional fusion in multiscale GCN. Given the joint-scale input, we first use a J-GTC block to extract the initial joint-scale features, and a joint2part pooling is applied on the joint-scale features to compute the initial part-scale features. We next feed them into two parallel J-GTC and P-GTC blocks. Then we concatenate the responses mapped by joint2part pooling and part2joint matching to the features in opposite scales. Compared to previous methods without the bidirectional fusion at immediate layers [10], [11], [27], [28], [29], [30], [31], [33], [52], [53], [54], the proposed Sybio-GNN model captures rich multiscale features at each network layer to enhance the information flow and comprehensive feature extraction. Therefore, both scales have good adaptability to multiscale information. After multiple interactive J-GTC and P-GTC blocks, we fuse the outputs of two scales through summation, followed by the average pooling to remove the temporal dimension, and obtain the high-level features.

Finally, we concatenate the outputs from three branches together and use them as the comprehensive semantics for action recognition and motion prediction.

5.2 Multitasking I: Action Recognition

For action recognition, Sybio-GNN can be represented as $\hat{\mathbf{y}} = \mathcal{F}_{\text{recg}}(\mathcal{X}_{\text{prev}}; \boldsymbol{\theta}_{\text{bk}}, \boldsymbol{\theta}_{\text{recg}})$, where $\mathcal{F}_{\text{recg}}(\cdot)$ is the recognition sub-model in Sybio-GNN, which combines the backbone network and a recognition head. Given the high-level features extracted by three branches of backbone, $\mathbf{H}_0$, $\mathbf{H}_1$ and $\mathbf{H}_2 \in \mathbb{R}^{M \times D_{\mathbf{h}}}$, we concatenate them and employ an MLP to produce the fused feature

$$\mathbf{H}_{\text{recg}} = \text{MLP}_{\text{recg}}([\mathbf{H}_0, \mathbf{H}_1, \mathbf{H}_2]) \in \mathbb{R}^{M \times D_{\mathbf{h}}},$$

where $\text{MLP}_{\text{recg}}(\cdot)$ denotes the fusing network of recognition task and $[\cdot, \cdot, \cdot]$ is the concatenation of three matrices along feature dimension. To integrate the joint dynamics, we apply the global averaging pooling on the M joints of $\mathbf{H}_{\text{recg}}$ and obtain a feature vector $\mathbf{h}_{\text{recg}} \in \mathbb{R}^{D_{\mathbf{h}}}$ that represents the whole body. We finally feed the vector into a 1-layer softmax classifier, obtaining $\hat{\mathbf{y}} \in [0, 1]^C$.

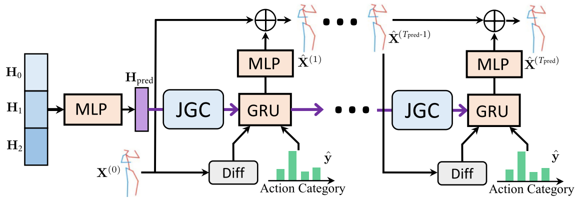


Fig. 7. The motion-prediction head of Sybio-GNN uses JGC [equation (5)], the difference operator [equation (9)] and GRU to predict the future poses sequentially. The box named 'JGC' performing on the hidden states corresponds to equation (10a). The box named 'GRU' performing on the input dynamics and action category corresponds to equation (10b). The output MLP and the residual connection summing up the input poses and outputs correspond to equation (10c).

5.3 Multitasking II: Motion Prediction

For motion preidction, Sybio-GNN works as $\hat{\mathbf{X}}_{\text{pred}} = \mathcal{F}_{\text{pred}}(\mathcal{X}_{\text{prev}}; \boldsymbol{\theta}_{\text{bk}}, \boldsymbol{\theta}_{\text{pred}})$, where $\mathcal{F}_{\text{pred}}(\cdot)$ is the prediction submodel that uses the backbone and a prediction head. Fig. 7 shows the structure of the prediction head, whose functionality is to sequentially predict the future poses. We adopt the self-regressive mechanics and identical connection in the prediction head, and we use gated recurrent unit (GRU) to model the temporal evolution.

Concretely, an MLP is first employed to embed the features of three action differences

$$\mathbf{H}_{\text{pred}} = \text{MLP}_{\text{pred}}([\mathbf{H}_0, \mathbf{H}_1, \mathbf{H}_2]) \in \mathbb{R}^{M \times D_{\mathbf{h}}},$$

where $\text{MLP}_{\text{pred}}(\cdot)$ denotes the fusing network of prediction task. Let $\mathbf{H}_{\text{pred}}^{(0)} = \mathbf{H}_{\text{pred}}$ be the initial states of the GRU-based predictor and $\hat{\mathbf{X}}^{(0)} = \mathbf{X}^{(0)}$ be the pose in the current timestamp. To produce the $(t+1)$th pose $(t \geq 0)$, the motion-prediction head works as

$$\widetilde{\mathbf{H}}_{\text{pred}}^{(t)} = \text{JGC}(\mathbf{H}_{\text{pred}}^{(t)}), \tag{10a}$$

$$\mathbf{H}_{\text{pred}}^{(t+1)} = \text{GRU}([\hat{\mathbf{X}}^{(t)}, \Delta^1\hat{\mathbf{X}}^{(t)}, \Delta^2\hat{\mathbf{X}}^{(t)}, \hat{\mathbf{y}}], \widetilde{\mathbf{H}}_{\text{pred}}^{(t)}), \tag{10b}$$

$$\hat{\mathbf{X}}^{(t+1)} = \hat{\mathbf{X}}^{(t)} + f_{\text{pred}}(\mathbf{H}_{\text{pred}}^{(t+1)}), \tag{10c}$$

where $\text{JGC}(\cdot)$, $\text{GRU}(\cdot)$ and $f_{\text{pred}}(\cdot)$ represent JGC operator, GRU cell and output MLP, respectively. The following $\hat{\mathbf{X}}^{(t)}$s are the predictions obtained sequentially and used recylingly. In Step (10a), we apply the JGC to update the hidden states; in Step (10b), we feed the updated hidden states, current pose and classified labels into the GRU cell to produce the features that reflect future displacement; In Step (10c), we add the predicted displacement to the previous pose to predict the next frame.

The motion-prediction head has three advantages: (i) we use JGC to update hidden features, capturing more complicated motion patterns; (ii) we input multiple orders of poses differences and classified labels to the GRU, providing explicit motion priors; and (iii) Connected by the residual, the GRU and MLP predict the displacement for each frame; this makes predictions precise and robust.

5.4 Multi-Objective Optimization

To train action recognition and motion prediction simultaneously, we consider a multi-objective scheme.

To recognize actions, we minimize the cross entropy between the ground-truth categorical labels and the inferred ones. Let the true label of the nth sample be $(\mathbf{y})_n \in \{0,1\}^C$ and the corresponding classification results be $(\hat{\mathbf{y}})_n \in \{0,1\}^C$. For N training samples in one mini-batch, the action recognition loss is formulated as

$$\mathcal{L}_{\text{recg}} = -\frac{1}{N}\sum_{n=1}^{N}(\mathbf{y})_n^\top \log{(\hat{\mathbf{y}})_n}, \tag{11}$$

where $\top$ denotes the transpose operation.

For motion prediction, we minimize the ℓ_1 distance between the target motions and the predicted clips. Let the nth target and predictions be $(\mathcal{X}_{\text{pred}})_n$ and $(\hat{\mathcal{X}}_{\text{pred}})_n$, for N samples in one mini-batch, the prediction loss is

$$\mathcal{L}_{\text{pred}} = \frac{1}{N}\sum_{n=1}^{N}\|(\mathcal{X}_{\text{pred}})_n - (\hat{\mathcal{X}}_{\text{pred}})_n\|_1, \tag{12}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm. According to our experiments, the ℓ_1 norm leads to more precise predictions compared to the common ℓ_2 norm.

To integrate two losses for training, we propose a convex combination that weighted sums Eqs. (11) and (12)

$$\mathcal{L} = \alpha\mathcal{L}_{\text{recg}} + (1-\alpha)\mathcal{L}_{\text{pred}},$$

where α trade-offs the importances of two tasks. In our training scheme, all the model parameters are trained end-to-end with the stochastic gradient descent algorithm [59].

5.5 Bone-Based Dual Graph Neural Networks

While the joints contain some information of action representation from the joint aspect, the attributes of bones, such as lengths and orientations, are crucial to provide some complementary information [35], [36], [60]. In this section, we construct a bone-based dual graph against the original joint-scale graph, whose vertices are bones and edges link bones.

To represent the feature of each bone, we compute the subtraction of two endpoint joint coordinates, which includes information of bone lengths and orientations. The subtraction order is from the centrifugal joint v_j to the centripetal v_i. Let the joint locations along time be $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^{(D_\mathbf{x}T_{\text{prev}})}$, the bone attribute is $\mathbf{b}_{i,j} = \mathbf{x}_j - \mathbf{x}_i \in \mathbb{R}^{(D_\mathbf{x}T_{\text{prev}})}$. Then, we construct the bone-based dual actional and structural graphs to model the bone relations; and we also build the part-scale dual graph. The dual actional graph is learned from bone features by AGIM (see Section 4.1.1); for dual structural graph, the 1-hop edges are linked when two bones with articulated joints and the high-hop edges are extended from the 1-hop edges; the part-scale attributes are obtained by integrating bone attributes and the part-scale graph is built according to body nature; The bone-based graphs are dual of joint-scale graphs, which are employed to extract complementary bone features.

Given the bone-based graphs, we train a *bone-based graph neural network* besides the original joint-based network, which takes body-bones as inputs and learns underlying action patterns from the bone-based perspective. We note that, for action recognition, we leverage both the joint-based and bone-based networks in training and test phases to make action recognition more effective. We stress that for action recognition, the output space of action recognition is the space of action labels. This naturally allows action recognition to extract and fuse motion information from various perspectives to improve the quality of features. Besides the joint-based network, the bone-based network could provide crucial and complementary motion features, providing more comprehensive information. However, as for motion prediction, we leverage both the joint-based network and the bone-based network in the training phase, but only use the joint-based network in the test phase. The reasons for this design mainly include: 1) In the training phases, both the joint-based and the bone-based motion-prediction heads work in a self-supervision manner and capture detailed motion information from two related, yet different perspectives, leading to the mutual enhancement of the model training. 2) In the test phases, the joint-based motion-prediction head can directly produce the future positions of the body-joints. On the other hand, we need to convert the bone-based prediction output to the joint-based representation, which might affect the overall prediction performance. We thus do not need the bone-based prediction output.

5.6 Discussion About the Actional Graph

Here we compare the learned actional graphs in our Sybio-GNN to the previous models which model long-range spatial dependencies of human actions.

To learn the actional graph from data, taking the joint-based network as an example, we estimate the affinities of arbitrary two joints given the joint embeddings by the proposed 'Actional Graph Inference Module' (AGIM). Specifically, the feature of any joint is learned by aggregating the features of any other joints through the estimated possible links for several times of information propagation, where the link features are learned based on the end-point joints. We finally construct the actional graph based on the affinity weights of any two joints by computing the normalized embedded Gaussian.

The intuition of our AGIM includes that we explicitly model the links and build the information propagation process between any two joints in distance, obtaining reliable information in a long-range for graph construction. Here are the comparisons between the proposed method and the existing methods [1], [35], [61], [62]:

- Compared to 2s-AGCN [35], both Sybio-GNN and 2s-AGCN calculate the joint affinities based on joint embeddings; however, when learning the joint embeddings, 2s-AGCN uses ego information and Sybio-GNN uses neighboring information.
- Compared to sLnL [61], both Sybio-GNN and sLnL consider local and non-local dependencies; however, sLnL captures only the joint-based dependencies in multiple ranges while Sybio-GNN learns dependencies of comprehensive body-joints and body-parts.
- Compared to GR-GCN [62], both Sybio-GNN and GR-GCN build relations between joints; however, GR-GCN designs a separate objective function to adjust the graph, which is independent from the final

task, while the proposed Sybio-GNN learns data-adaptive graphs in an end-to-end manner.

- Compared to our previous work AS-GCN [1], both Sybio-GNN and AS-GCN employ joint-edge feature propagation before actional graph inference, while Sybio-GNN computes joint affinities in a more efficient manner and AS-GCN pretrains an encoder-decoder to capture the latent joint relations with a complex data sampling.

In summary, the novelty of our learnable actional graph mainly focuses on a rich joint-edge propagation process for information aggregation; meanwhile, based on the affinity calculation, we obtain the stable and reasonable graph structures through end-to-end training.

6 Experiments and Analysis

In this section, we evaluate the proposed Sybio-GNN. First, we introduce the datasets and model settings; then, the performance comparisons between Sybio-GNN and other state-of-the-art methods are presented; and we finally show the ablation studies of our model.

6.1 Datasets and Model Setting

6.1.1 Dataset

We conduct extensive experiments on four large-scale datasets: NTU-RGB+D [31], Kinetics [12], Human3.6M [32] and CMU Mocap. The details is shown as follow.

NTU-RGB+D. NTU-RGB+D, containing 56,880 skeleton action sequences completed by one or two performers and categorized into 60 classes, is one of the largest datasets for 3D skeleton-based action recognition. It provides the 3D spatial coordinates of 25 joints for each subject in an action. For method evaluation, two protocols are recommended: 'Cross-Subject' (CS) and 'Cross-View' (CV). In CS, 40,320 samples performed by 20 subjects are separated into the training set, and the rest belong to the test set. CV assigns data according to camera views, where training and test set have 37,920 and 18,960 samples, respectively.

Kinetics. Kinetics is a large dataset for human action analysis, containing over 240,000 video clips. There are 400 classes of actions. Due to only RGB videos, we obtain skeleton data by estimating joint locations on pixels with OpenPose toolbox [63]. The toolbox generates 2D pixel coordinates (x, y) and confidence score c for totally 18 joints. We represent each joint as a three-element feature vector: $[x, y, c]^\top$. For the multiple-person cases, we select the body with the highest average joint confidence in each sequence. Therefore, one clip with T frames is transformed into a skeleton sequence with the dimension of $18 \times 3 \times T$.

Human3.6M. Human3.6M (H3.6M) is a large motion capture dataset. Seven subjects are performing 15 classes of actions, where each subject has 32 joints. We downsample all sequences by two. The models are trained on six subjects and tested on the specific clips of the 5th subject. Notably, the dataset provides the joint locations in angle space, and we transform them into exponential maps and only use the joints with non-zero values.

CMU Mocap. CMU Mocap includes five major action categories, and each subject in CMU Mocap has 38 joints, which are presented by angle positions. We use the same strategy presented in [26] to select the actions. Thus we choose eight actions: 'Basketball', 'Basketball Signal', 'Directing Traffic', 'Jumping', 'Running', 'Soccer', 'Walking' and 'Washing Window'. We preprocess the data and compute the corresponding exponential maps with the same approach as we do for Human3.6M dataset.

6.1.2 Model Setting and Implementation Details

The models are implemented with PyTorch 0.4.1. Since different datasets have distinctive patterns and complexities, we employ specific configurations of Sybio-GNN networks on corresponding datasets.

For NTU-RGB+D and Kinetics, the backbone network of Sybio-GNN contains 9 J-GTC blocks and 8 P-GTC blocks. In each three J-GTC and P-GTC blocks, the feature dimensions are respectively 64, 128 and 256. The kernel size of TC is 9 and it shrinks the temporal dimension with stride 2 after the 3rd and 6th blocks, where we use bidirectional fusion mechanisms. $\lambda_{\text{act}} = 0.5$. The action-recognition head is a 2-layer MLP, whose hidden dimension is 256. For the motion-prediction head, the hidden dimensions of GRU and output MLP are 256. For the actional graph inference module (AGIM), we use 2-layer 128-D MLPs with ReLU, batch normalization and dropout in each iteration. For the loss function, we the coefficient α in a range of [0.8,0.95]. We use SGD algorithm to train Sybio-GNN, where the learning rate is initially 0.1 and decays by 10 every 30 epochs. The model is trained with batch size 64 for 100 epochs on 8 GTX-1080Ti GPUs. For both NTU-RGB+D and Kinetics, the last 10 frames are used for motion prediction and other previous frames are fed into Sybio-GNN for action recognition.

As for Human3.6M and CMU Mocap, due to the simpler dynamics and fewer categories, we propose a light version of Sybio-GNN, which extracts meaningful features with more shallow networks, improving efficiency for motion prediction. In the backbone, we use 4 J-GTC blocks and 3 P-GTC blocks, whose feature dimensions are 32, 64, 128 and 256; the temporal convolution strides in 4 blocks are: 1, 2, 2, 2, respectively. We apply bidirectional fusions at the last 3 layers. $\lambda_{\text{act}} = 1.0$. The recognition and motion-prediction heads, as well as AGIM, leverage the same architecture as we set for NTU-RGB+D. We train the model using Adam optimizer with the learning rate 1×10^{-4} and batch size 64 for 10^5 iterations on one GTX-1080Ti GPU. All the hyper-parameters are selected using a validation set.

6.2 Comparison With State-of-the-Arts

On the three large-scale skeleton-formed datasets, we compare the proposed Sybio-GNN with state-of-the-art methods for human action recognition and motion prediction.

6.2.1 3D Skeleton-Based Action Recognition

For action recognition, we first show the classification accuracies and model complexities of Sybio-GNN and baselines on two recommended benchmarks of NTU-RGB+D, i.e., Cross-Subject and Cross-View [31]. The state-of-the-art models are based on manifold analysis [10], recurrent neural networks [11], [27], [31], [34], convolution networks [9], [21], [22], and graph networks [1], [12], [27], [28], [30], [35], [36], [51], [62], [64]. Moreover, to investigate different

TABLE 1
Comparison of Action Recognition on NTU-RGB+D

Methods	CS	CV	Param (M)	Test time (s)
Lie Group [10]	50.1%	52.8%	-	-
H-RNN [11]	59.1%	64.0%	-	-
Deep LSTM [31]	60.7%	67.3%	-	-
PA-LSTM [31]	62.9%	70.3%	-	-
ST-LSTM+TS [34]	69.2%	77.7%	-	-
Temporal Conv [21]	74.3%	83.1%	-	-
Visualize CNN [22]	76.0%	82.6%	-	-
ST-GCN [12]	81.5%	88.3%	3.10	2.28×10^{-2}
DPRL [64]	83.5%	89.8%	-	-
SR-TSL [27]	84.8%	92.4%	-	-
HCN [9]	86.5%	91.1%	2.06	2.42×10^{-2}
PB-GCN [30]	87.5%	93.2%	3.55	3.37×10^{-2}
AS-GCN [1]	86.8%	94.2%	6.99	2.64×10^{-1}
2s-AGCN [35]	88.5%	95.1%	6.94	4.05×10^{-2}
AGC-LSTM [28]	89.2%	95.0%	-	-
DGNN [36]	89.9%	96.1%	8.18	4.17×10^{-2}
sLnL [61]	89.1%	94.1%	-	-
GR-GCN [62]	87.5%	94.3%	-	-
RA-GCN [51]	87.3%	93.6%	6.21	4.12×10^{-2}
Sybio-GNN (Only J-S)	88.3%	94.5%	7.83	2.37×10^{-2}
Sybio-GNN (Only J-A)	85.7%	93.7%	5.12	2.64×10^{-2}
Sybio-GNN (Only P)	86.5%	87.3%	5.14	2.29×10^{-2}
Sybio-GNN (No bone)	87.1%	93.8%	7.43	5.97×10^{-2}
Sybio-GNN (No pred)	89.0%	94.7%	9.14	4.23×10^{-2}
Sybio-GNN	**90.1%**	**95.4%**	14.85	6.03×10^{-2}

The accuracies on both Cross-Subject (CS) and Cross-View (CV) benchmarks.

components of Sybio-GNN, such as multiple graphs and multitasking, we test several model variants, including Sybio-GNN using only joint-scale structural graphs (Only J-S), only joint-scale actional graphs (Only J-A), only-part scale graph (Only P), no bone-based networks (No bone), no prediction for multitasking (No pred) and the complete model. For these methods, Table 1 presents recognition accuracies. We also count the numbers of model parameters and the average inference time to run each test sample of several methods with their open-source codes. We see that the complete Sybio-GNN outperforms the baselines on both benchmarks. The results reveal that richer joint relations promote to capture more useful patterns, and additional motion prediction and complementary bone-based features improve the discrimination. In addition, to compare the model complexities, many recent models, such as AS-GCN, DGBNN and Sybio-GNN have a similar magnitude of parameter numbers, but Sybio-GNN effectively improves the recognition performances. For the test time, Sybio-GNN costs time that is similar to the previous works, and it outperforms the previous AS-GCN with a large margin.

Then, we evaluate the recognition performance and complexity of Sybio-GNN on Kinetics, and we it with eights baselines, including a hand-crafted based method, Feature Encoding [20], two deep models, Deep LTSM [31] and Temporal ConvNet [21], and five graph-based methods [1], [12], [35], [36], [37]. Table 2 shows the top-1 and top-5 classification results, parameter numbers and average test times for each sample, where Sybio-GNN (No pred) denotes the Sybio-GNN variant without motion-prediction head for multitasking. We see that Sybio-GNN outperforms other methods on top-1 recognition accuracy and achieves competitive results on top-5 accuracy, as well as Sybio-GNN has acceptable model weights and running speed.

TABLE 2
Comparison of Action Recognition on Kinetics

Methods	Top-1	Top-5	Param (M)	Test time (s)
Feature Encoding [20]	14.9%	25.8%	-	-
Deep LSTM [31]	16.4%	35.3%	-	-
Temporal Conv [21]	20.3%	40.0%	-	-
ST-GCN [12]	30.7%	52.8%	3.10	2.17×10^{-2}
STGR-GCN [37]	33.6%	56.1%	-	-
AS-GCN [1]	34.8%	56.3%	6.99	2.80×10^{-1}
2s-AGCN [35]	36.1%	58.7%	6.94	3.94×10^{-2}
DGNN [36]	36.9%	56.9%	8.18	4.09×10^{-2}
sLnL [61]	36.6%	**59.1**%	-	-
Sybio-GNN (No pred)	36.4%	57.4%	9.14	4.30×10^{-2}
Sybio-GNN	**37.2%**	58.1%	14.85	5.97×10^{-2}

The top-1 and top-5 classification accuracies are listed.

TABLE 3
Comparison of Action Recognition on Human 3.6M
and CMU Mocap Dataset

	Human3.6M		CMU Mocap	
Methods	Top-1	Top-5	Top-1	Top-5
ST-GCN [12]	40.2%	78.4%	87.5%	96.9%
HCN [9]	47.6%	88.8%	95.4%	99.2%
2s-AGCN [35]	55.4%	94.1%	97.1%	99.8%
Sybio-GNN (Only J)	55.6%	93.9%	96.5%	99.4%
Sybio-GNN (Only P)	54.3%	93.1%	94.9%	98.0%
Sybio-GNN (No bone)	53.5%	93.2%	93.5%	95.8%
Sybio-GNN (No pred)	55.2%	94.1%	96.6%	99.4%
Sybio-GNN	**56.5%**	**95.3%**	**98.8%**	**100%**

The top-1 and top-5 classification accuracies are listed.

Additionally, we evaluate our model for action recognition on Human3.6M and CMU Mocap. Table 3 presents the top-1 and top-5 classification accuracies for both two datasets. Here we compare Sybio-GNN with a few recently proposed methods: ST-GCN [12], HCN [9], and 2s-AGCN [35]. We also show the effectiveness of our model.

Notably, for Human3.6M, there is a relatively large gap between the top-1 and top-5 accuracies, because the input motions are some fragmentary clips of long sequences with incomplete semantics and activities have subtle differences (e.g., 'Eating' and 'Smoking' are similar). In other words, Sybio-GNN learns the common features and provides reasonable discrimination, resulting in high top-5 accuracy; but it confuses in non-semantic variances, causing not high top-1 accuracy. However, CMU Mocap has more distinctive actions, where we obtain high classification accuracies.

6.2.2 3D Skeleton-Based Motion Prediction

To validate the model of motion prediction, we train the Sybio-GNN on NTU-RGB+D, Human3.6M, and CMU Mocap. There are two specific tasks: short-term and long-term motion prediction. Concretely, the target of short-term prediction is commonly to predict poses within 400 milliseconds, while the long-term prediction aims to predict poses in 1,000 ms or longer. To reveal the effectiveness of Sybio-GNN, we introduce many state-of-the-art methods, which

TABLE 4
Comparisons of MAEs Between Sybio-GNN and State-of-the-Art Methods for Short-Term Motion Prediction on the Four Representative Actions of H3.6M

Motion	Walking				Eating				Smoking				Discussion			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
ZeroV [15]	0.39	0.68	0.99	1.15	0.27	0.48	0.73	0.86	0.26	0.48	0.97	0.95	0.31	0.67	0.94	1.04
ERD [42]	0.93	1.18	1.59	1.78	1.27	1.45	1.66	1.80	1.66	1.95	2.35	2.42	2.27	2.47	2.68	2.76
Lstm3LR [42]	0.77	1.00	1.29	1.47	0.89	1.09	1.35	1.46	1.34	1.65	2.04	2.16	1.88	2.12	2.25	2.23
SRNN [14]	0.81	0.94	1.16	1.30	0.97	1.14	1.35	1.46	1.45	1.68	1.94	2.08	1.22	1.49	1.83	1.93
DropAE [65]	1.00	1.11	1.39	/	1.31	1.49	1.86	/	0.92	1.03	1.15	/	1.11	1.20	1.38	/
Samp-loss [15]	0.92	0.98	1.02	1.20	0.98	0.99	1.18	1.31	1.38	1.39	1.56	1.65	1.78	1.80	1.83	1.90
Res-sup [15]	0.27	0.46	0.67	0.75	0.23	0.37	0.59	0.73	0.32	0.59	1.01	1.10	0.30	0.67	0.98	1.06
CSM [26]	0.33	0.54	0.68	0.73	0.22	0.36	0.58	0.71	0.26	0.49	0.96	0.92	0.32	0.67	0.94	1.01
TP-RNN [66]	0.25	0.41	0.58	0.65	0.20	0.33	0.53	0.67	0.26	0.47	0.88	0.90	0.30	0.66	0.96	1.04
QuaterNet [23]	0.21	0.34	0.56	0.62	0.20	0.35	0.58	0.70	0.25	0.47	0.93	0.90	0.26	0.60	0.85	0.93
AGED [16]	0.21	0.35	0.55	0.64	0.18	**0.28**	0.50	0.63	0.27	0.43	0.81	0.83	0.26	0.56	**0.77**	**0.84**
BiHMP-GAN [25]	0.33	0.52	0.63	0.67	0.20	0.33	0.54	0.70	0.26	0.50	0.91	0.86	0.33	0.65	0.91	0.95
Skel-TNet [67]	0.31	0.50	0.69	0.76	0.20	0.31	0.53	0.69	0.25	0.50	0.93	0.89	0.30	0.64	0.89	0.98
VGRU-r1 [68]	0.34	0.47	0.64	0.72	0.27	0.40	0.64	0.79	0.36	0.61	0.85	0.92	0.46	0.82	0.95	1.21
Sybio-GNN (Only J-A)	0.19	0.35	0.54	0.63	0.18	0.34	0.54	0.66	0.23	0.43	0.84	0.82	0.26	0.62	0.81	0.87
Sybio-GNN (Only J-S)	0.19	0.33	0.54	0.69	0.17	0.32	0.52	0.66	0.21	0.41	0.83	0.82	0.24	0.64	0.93	1.01
Sybio-GNN (No recg)	0.18	**0.31**	**0.50**	**0.59**	**0.16**	0.29	0.49	0.61	**0.21**	**0.40**	0.80	**0.80**	0.22	0.57	0.85	0.93
Sybio-GNN	**0.17**	**0.31**	**0.50**	0.60	**0.16**	0.29	**0.48**	**0.60**	**0.21**	**0.40**	**0.76**	**0.80**	**0.21**	**0.55**	**0.77**	0.85

Sybio-GNN (J-A) and Sybio-GNN (J-S) are Sybio-GNN with joint-scale actional graphs only and with joint-scale structural graph only, respectively. Sybio-GNN (No recg) represents the model trained without action classification.

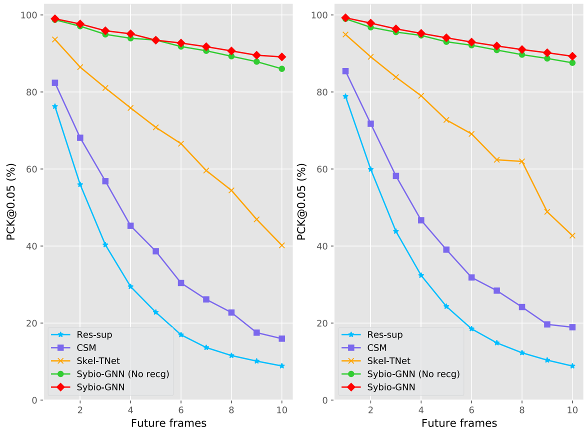
TABLE 5
Comparisons of MAEs Between Sybio-GNN and Previous Methods for Short-Term Motion Prediction on Other 11 Actions of H3.6M Dataset

Motion	Directions				Greeting				Phoning				Posing				Purchases				Sitting			
millisecond	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
ZeroV [15]	0.39	0.59	0.79	0.89	0.54	0.89	1.30	1.49	0.64	1.21	1.65	1.83	0.28	0.57	1.13	1.37	0.62	0.88	1.19	1.27	0.40	1.63	1.02	1.18
Res-sup [15]	0.41	0.64	0.80	0.92	0.57	0.83	1.45	1.60	0.59	1.06	1.45	1.60	0.45	0.85	1.34	1.56	0.58	0.79	1.08	1.15	0.41	0.68	1.12	1.33
CSM [26]	0.39	0.60	0.80	0.91	0.51	0.82	1.21	1.38	0.59	1.13	1.51	1.65	0.29	0.60	1.12	1.37	0.63	0.91	1.19	1.29	0.39	0.61	1.02	1.18
TP-RNN [66]	0.38	0.59	0.75	0.83	0.51	0.86	1.27	1.44	0.57	1.08	1.44	1.59	0.42	0.76	1.29	1.54	0.59	0.82	1.12	1.18	0.41	0.66	1.07	1.22
Skel-TNet [67]	0.36	0.58	0.77	0.86	0.50	0.84	1.28	1.45	0.58	1.12	1.52	1.64	0.29	0.62	1.19	1.44	0.58	0.84	1.17	1.24	0.40	0.61	1.01	1.15
Sybio-GNN (No recg)	0.24	0.45	0.61	0.67	0.36	0.61	0.98	1.17	0.50	0.86	1.29	1.43	**0.18**	**0.44**	0.99	1.22	**0.40**	0.62	1.00	1.08	**0.23**	**0.41**	0.80	0.97
Sybio-GNN	**0.23**	0.42	**0.57**	**0.65**	**0.35**	**0.60**	**0.95**	**1.15**	**0.48**	**0.80**	1.28	**1.41**	**0.18**	0.45	**0.97**	**1.20**	**0.40**	**0.60**	**0.97**	**1.04**	0.24	**0.41**	**0.77**	**0.95**
Motion	Sitting Down				Taking Photo				Waiting				Walking Dog				Walking Together				Average			
millisecond	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
ZeroV [15]	0.39	0.74	1.07	1.19	0.25	0.51	0.79	0.92	0.34	0.67	1.22	1.47	0.60	0.98	1.36	1.50	0.33	0.66	0.94	0.99	0.39	0.77	1.05	1.21
Res-sup. [15]	0.47	0.88	1.37	1.54	0.28	0.57	0.90	1.02	0.32	0.63	1.07	1.26	0.52	0.89	1.25	1.40	0.27	0.53	0.74	0.79	0.40	0.69	1.04	1.18
CSM [26]	0.41	0.78	1.16	1.31	0.23	0.49	0.88	1.06	0.30	0.62	1.09	1.30	0.59	1.00	1.32	1.44	0.27	0.52	0.71	0.74	0.38	0.68	1.01	1.13
TP-RNN [66]	0.41	0.79	1.13	1.27	0.26	0.51	0.80	0.95	0.30	0.60	1.09	1.28	0.53	0.93	1.24	1.38	0.23	0.47	0.67	0.71	0.37	0.66	0.99	1.11
Skel-TNet [67]	0.37	0.72	1.05	1.17	0.24	0.47	0.78	0.93	0.30	0.63	1.17	1.40	0.54	0.88	1.20	1.35	0.27	0.53	0.68	0.74	0.36	0.64	0.99	1.02
Sybio-GNN (No recg)	0.30	0.62	0.91	1.03	0.16	0.34	0.55	0.66	**0.22**	0.49	0.89	1.09	**0.42**	0.74	1.09	1.25	0.17	0.34	0.52	0.57	**0.26**	0.50	0.82	0.94
Sybio-GNN	**0.28**	**0.60**	**0.89**	**0.99**	**0.14**	**0.32**	**0.53**	**0.64**	**0.22**	**0.48**	**0.87**	**1.06**	**0.42**	**0.73**	**1.08**	**1.22**	**0.16**	**0.33**	**0.50**	**0.56**	**0.26**	**0.49**	**0.79**	**0.92**

learned dynamics from pose vectors [15], [16], [25], [42], [66] or separate body-parts [14], [26], [67]. We also introduce a naive baseline, named ZeroV [15], which sets all predictions to be the last observed frame.

Short-Term Motion Prediction. We validate Sybio-GNN on two datasets: Human3.6M and NTU-RGB+D. We first compare Sybio-GNN to baselines for short-term prediction on Human3.6M, where the models generate poses up to the future 400 ms. We analyze several variants of Sybio-GNN with different components, including using only joint-scale actional graphs (Only J-A) or joint-scale structural graphs (Only J-S), and no recognition head (No recg). For the metric, the mean angle errors (MAE) between the predictions and the ground truths are computed, representing the prediction errors in angle space. We first test 4 representative actions: 'Walking', 'Eating', 'Smoking' and 'Discussion'. Table 4 shows MAEs of different methods. As we see, when Sybio-GNN simultaneously employs multiple graphs and multitasking, our method outperforms all the baselines and its own ablations.

We also test Sybio-GNN on the remaining 11 actions in Human3.6M, where the MAEs of some recent methods are shown in Table 5. Sybio-GNN also achieves the best performance on most actions and the lowest average MAE on 15 motions. Although the mentioned top-1 classification accuracy on this dataset is not very high (see Table 3), we note



(a) Cross-Subject (b) Cross-View

Fig. 8. Comparison of PCK@0.05 (%) between Sybio-GNN and state-of-the-art methods for short-term motion prediction on NTU-RGB+D. The variant of Sybio-GNN (No recg) denotes our model without using the recognition task to enhance motion prediction.

TABLE 7
Comparisons of MAEs Between Our Model and Other Methods for Long-Term Motion Prediction on Four Actions of H3.6M

Motion	Walking		Eating		Smoking		Discussion	
milliseconds	560	1k	560	1k	560	1k	560	1k
ZeroV [15]	1.35	1.32	1.04	1.38	1.02	1.69	1.41	1.96
ERD [42]	2.00	2.38	2.36	2.41	3.68	3.82	3.47	2.92
Lstm3LR [42]	1.81	2.20	2.49	2.82	3.24	3.42	2.48	2.93
SRNN [14]	1.90	2.13	2.28	2.58	3.21	3.23	2.39	2.43
DropAE [65]	1.55	1.39	1.76	2.01	1.38	1.77	1.53	1.73
Res-sup. [15]	0.93	1.03	0.95	1.08	1.25	1.50	1.43	1.69
CSM [26]	0.86	0.92	0.89	1.24	0.97	1.62	1.44	1.86
TP-RNN [66]	**0.74**	**0.77**	0.84	1.14	0.98	1.66	1.39	1.74
AGED [16]	0.78	0.91	0.86	0.93	1.06	1.21	1.25	1.30
BiHMP-GAN [25]	/	0.85	/	1.20	/	**1.11**	/	1.77
Skel-TNet [67]	0.79	0.83	0.84	1.06	0.98	1.21	1.19	1.75
Sybio-GNN	0.75	0.78	**0.77**	**0.88**	**0.92**	1.18	**1.17**	**1.28**

that the estimated soft labels cover the common motion factors (reflected by top-5 accuracy), such as the walking factor in 'Walking', 'Walking Dog' and 'Walking Together', and we need the walking factors instead of the specific labels for motion generation. Given the soft labels, the model tends to obtain precise predictions.

For NTU-RGB+D, Sybio-GNN aims to forecast future 10 frames. We compare our Sybio-GNN with several previous methods. We also construct a Sybio-GNN variant that abandons action-recognition head (No recg). Specifically, let the input pose vector have dimension of $3M$ to represent M 3D joint positions, and the input dimension of GRU is also set to be $3M$ to match the pose dimension. As for the metric, we use the percentage of correct keypoints within a normalized region 0.05 (PCK@0.05), where a joint is correctly predicted if the normalized distance between the predicted position and ground-truth is less than 0.05. The PCK@0.05 of different models is illustrated in Fig. 8. We see: 1) our model outperforms the baselines with a large margin especially at the last several frames; 2) Using action recognition and motion prediction together obtains the highest PCK@0.05 along time, demonstrating the enhancements from recognition task for dynamics learning.

Long-Term Motion Prediction. For long-term prediction, Sybio-GNN is tested on Human3.6M and CMU Mocap. We predict the future poses with high variation up to 1,000 millisecond. Table 6 presents the MAEs of various models for predicting the 4 motions in Human3.6M at the future 560 and 1,000 ms. We see that Sybio-GNN outperforms the competitors on 'Eating', 'Smoking' and 'Discussion', and obtain competitive results on 'Walking'.

To further evaluate Sybio-GNN, we conduct long-term prediction on CMU Mocap. We present the MAEs of Sybio-GNN with or without using the action-recognition head. Table 7 shows the predicting MAEs ranging from future 80 to 1,000 ms. We see that Sybio-GNN significantly

TABLE 6
Comparisons of MAEs Between Our Model and the State-of-the-Art Methods on the Eight Actions of CMU Mocap Dataset

Motion	Basketball					Basketball Signal					Directing Traffic					Jumping				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
Res-sup [15]	0.49	0.77	1.26	1.45	1.77	0.42	0.76	1.33	1.54	2.17	0.31	0.58	0.94	1.10	2.06	0.57	0.86	1.76	2.03	2.42
Res-uns [15]	0.53	0.82	1.30	1.47	1.81	0.44	0.80	1.35	1.55	2.17	0.35	0.62	0.95	1.14	2.08	0.59	0.90	1.82	2.05	2.46
CSM [26]	0.37	0.62	1.07	1.18	1.95	0.32	0.59	1.04	1.24	1.96	0.25	0.56	0.89	1.00	2.04	0.39	0.60	1.36	1.56	2.01
BiHMP-GAN [25]	0.37	0.62	1.02	1.11	1.83	0.32	0.56	1.01	1.18	1.89	0.25	0.51	0.85	0.96	1.95	0.39	0.57	1.32	1.51	1.94
Skel-TNet [67]	0.35	0.63	1.04	1.14	1.78	0.24	0.40	0.69	0.80	1.07	0.22	0.44	0.78	0.90	1.88	0.35	**0.53**	**1.28**	**1.49**	1.85
Sybio-GNN (No recg)	0.33	**0.48**	0.95	1.09	**1.47**	0.15	0.26	0.47	0.56	1.04	**0.20**	**0.41**	0.77	0.89	1.95	**0.32**	0.55	1.40	1.60	1.87
Sybio-GNN	**0.32**	**0.48**	**0.91**	**1.06**	**1.47**	**0.12**	**0.21**	**0.38**	**0.49**	**0.94**	**0.20**	**0.41**	**0.75**	**0.87**	**1.84**	**0.32**	0.55	1.40	1.60	**1.82**

Motion	Running					Soccer					Walking					Washing Window				
milliseconds	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
Res-sup [15]	0.32	0.48	0.65	0.74	1.00	0.29	0.50	0.87	0.98	1.73	0.35	0.45	0.59	0.64	0.88	0.32	0.47	0.74	0.93	1.37
Res-uns [15]	0.35	0.50	0.69	0.76	1.04	0.31	0.51	0.90	1.00	1.77	0.36	0.47	0.62	0.65	0.93	0.33	0.47	0.75	0.95	1.40
CSM [26]	0.28	0.41	**0.52**	0.57	0.67	0.26	0.44	0.75	0.87	1.56	0.35	0.44	0.45	0.50	0.78	0.30	0.47	0.80	1.01	1.39
BiHMP-GAN [25]	0.28	0.40	0.50	0.53	0.62	0.26	0.44	0.72	0.82	1.51	0.35	0.45	0.44	0.46	0.72	0.31	0.46	0.77	0.92	1.31
Skel-TNet [67]	0.38	0.48	0.57	0.62	0.71	0.24	0.41	0.69	0.79	1.44	0.33	0.41	0.45	0.48	0.73	0.31	0.46	0.79	0.96	1.37
Sybio-GNN (No recg)	**0.21**	**0.33**	0.53	**0.56**	0.66	0.22	0.38	0.72	0.83	1.38	**0.26**	**0.32**	0.38	0.41	0.54	**0.22**	**0.33**	0.62	0.83	1.07
Sybio-GNN	**0.21**	**0.33**	0.53	**0.56**	**0.65**	**0.19**	**0.32**	**0.66**	**0.78**	**1.32**	**0.26**	**0.32**	**0.35**	**0.39**	**0.52**	**0.22**	**0.33**	**0.55**	**0.73**	**1.05**

We evaluate the model for long-term prediction and present the MAEs at both short and long-term prediction timestamps.

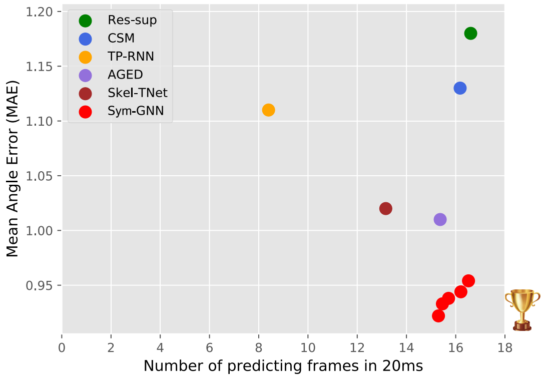


Fig. 9. Sybio-GNN is both faster and more precise compared to others. Various red circles denote different iteration numbers K in AGIM, where $K = 0, 1, 2, 3, 4$. The bottom right corner (highlighted by a trophy cup) indicates higher speed and lower error, showing an ideal target.

TABLE 8
Action Recognition Accuracies With Noisy Motion Prediction Targets in Varying Degrees on NTU-RGB+D Dataset

Noise ratio	CS	CV
0%	**90.1%**	**96.4%**
10%	89.8%	96.1%
20%	89.5%	96.1%
50%	89.1%	95.5%
70%	88.5%	94.9%
100%	87.7%	93.9%
No pred	89.0%	95.7%

'No pred' denotes model without motion prediction task.

outperforms the state-of-the-art methods on actions 'Basketball', 'Basketball Signal' and 'Washing Window', and obtains competitive performance on 'Jumping' and 'Running'.

Effectiveness-Efficiency Tradeoff. We compare the prediction errors and efficiency of various models on motion prediction. The high response speed and precise generation are both essential in real-time scenarios. In Symbio-GNN, the AGIM propagates the features between joints and edges iteratively. The iteration times K trades off between effectiveness and speed; i.e., a larger K leads to a lower MAE but slower speed. To represent the running speed, we use the generated frame numbers in each 20 ms (frame period) when we predict up to 400 ms. Tuning K from 0 to 4, we compare Sybio-GNN to other methods on Human3.6M and show the running speeds and MAEs in Fig. 9, where different red circles denote different numbers of iterations K in AGIM, i.e., from the rightmost circle to the leftmost one, $K = 0, 1, 2, 3, 4$. We see that the Sybio-GNN is both faster and more precise compared to its competitors.

6.3 Ablation Studies

6.3.1 Symbiosis of Recognition and Prediction

To analyze the mutual effects of action recognition and motion prediction, we conduct several experiments.

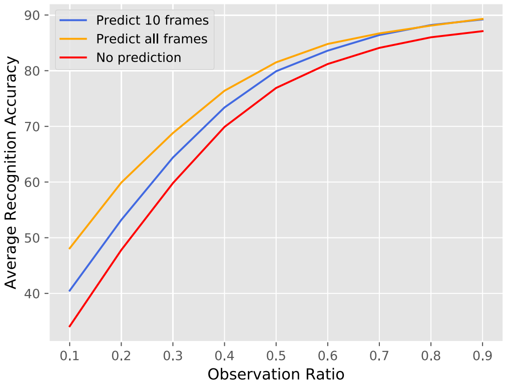


Fig. 10. Given the same input, predicting more future poses leads to a better performance of action recognition. We see that across all the observation ratios, predicting all future poses is better than predicting 10 future poses; and both are better than no prediction.

We first study the effects on action recognition from motion prediction. We use accurate class labels but noisy future poses to train the multitasking Sybio-GNN for action recognition. To represent noisy supervisions, we randomly shuffle a percentage of targets motions among training data. Table 8 presents the recognition accuracies with various ratios of noisy prediction targets on two benchmarks of NTU-RGB+D. We also show the recognition results of the model without motion-prediction head. We see that 1) the predicted head benefits the action-recognition head. Introducing a motion-prediction head is beneficial even when the noise ratio is around 50 percent; 2) when the noise ratio exceeds 50 percent, the recognition performance is just slightly worse than that of the model without the motion-prediction head, reflecting that the action recognition is robust against the deflected motion prediction. Consequently, we show that motion prediction strengthens action recognition. Similarly, we also test how the confused recognition results affect motion prediction by using noisy action categories, which are presented in Appendix, available in the online supplemental material.

We finally test the promotion on recognition when the observed data is limited, where we intercept the early motions by a ratio (e.g., 10 percent) for action recognition. There are three models with various prediction strategies: 1) predicting the future 10 frames ('Pred 10 frames'); 2) predicting all future frames ('Pred all frames'); 3) no prediction ('No pred'). Fig. 10 illustrates the recognition accuracies of three models on different observation ratios. As we see, when the observation ratio is low, 'Pred all frames' can be aware of the entire action sequences and capture richer dynamics, showing the best performance; when the observation ratio is high, predicting 10 or all frames are similar because the inputs carry sufficient patterns, but they outperform 'No pred' as they preserve information. By introducing the motion-prediction head, our Sybio-GNN has the potential for action classification in the early period.

6.3.2 Effects of Graphs

In this section, we study the abilities of various graphs, namely, only joint-scale structural graphs (Only J-S), only joint-scale actional graphs (Only J-A), only part-scale graphs (Only P), and combining them (full).

For action recognition, on Cross-Subject of NTU-RGB+D, we investigate different graph configurations. While

TABLE 9
Recognition Accuracies on NTU-RGB+D, CS With Various Graphs: Only Joint-Scale Structural Graphs (Only J-S), Only Joint-Scale Actional Graph (Only J-A), Only Part-Scale Graph (Only P) and all Graphs (Full)

JS-Hop (Γ)	Only J-S	Only J-A	Only P	full
1	85.9%	85.7%	87.3%	86.1%
2	86.2%			86.9%
3	87.5%			88.3%
4	88.3%			**90.1%**

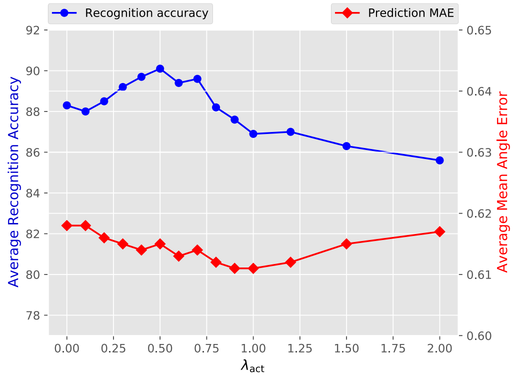


Fig. 11. Average action recognition accuracies and motion prediction MAEs of models with different λ_{act}.

involving a joint-scale structural graph, we respectively set the number of hop in the joint-scale structural graphs (JS-Hop) to be $\Gamma = 1, 2, 3, 4$. Note that when we use only joint-scale structural graphs with $\Gamma = 1$, the corresponding graph is exactly the skeleton itself. Table 9 presents the results of Sybio-GNN with different graph components for action recognition. We see that 1) representing long-range structural relations, higher Γ leads to more effective action recognition; 2) combining the multiple graphs introduced from different perspectives improves the action recognition performance significantly. We also test various graph components for motion prediction in Appendix, available in the online supplemental material.

6.3.3 *Balance Joint-Scale Actional and Structural Graphs*

In our model, we present that the power of actional and structural graphs in JGC operator is traded off by a hyper-parameter λ_{act} (see Eq. (5)). Here we analyze how λ_{act} affects the model performances.

For action recognition, we test our model on NTU-RGB+D, Cross-Subject, and present the classification accuracies with different λ_{act}; for motion prediction, we show the average MAEs for short-term prediction on Human3.6M. Fig. 11 illustrates the model performances for both tasks. We see: 1) when $\lambda_{act} = 0.5$, we obtain the highest recognition accuracies, showing large improvements than other λ_{act}; 2) for motion prediction, the performance is robust against different λ_{act}, where the MAEs fluctuate around 0.615, but $\lambda_{\text{act}} = 0.9$ and 1.0 lead to the lowest errors.

TABLE 10
The Recognition Accuracies of Model With Different Parallel Networks on NTU-RGB+D

Parallel Network	CS	CV
Only Joint	87.1%	93.8%
Only Bone	87.4%	93.5%
Joint & Bone	**90.1%**	**95.4%**

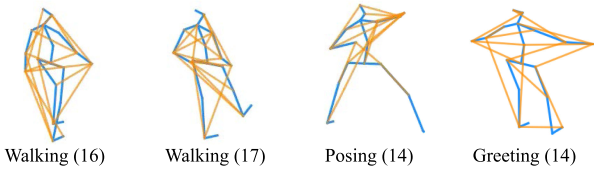


Fig. 12. Joint-scale actional graphs on different motions in H3.6M. Orange lines indicate the connections whose weights are greater than 0.5 in actional graphs, and bule lines indicate the original skeletons. The numbers with brackets presented below plots denotes the numbers of actional relations that are drawn here.

6.3.4 *Bone-Based Dual Graph Neural Networks*

We validate the effectiveness of using dual networks which take joint and bone features as inputs for action recognition, respectively. Table 10 presents the recognition accuracies for different combinations of joint-based and bone-based dual networks on two benchmarks of NTU-RGB+D dataset. We see that only using joint features or bone features for action recognition cannot obtain the most accurate recognition, but combining joint and bone features could improve the classification performances with a large margin, indicating the complementary information carried by the two networks.

6.4 Visualization

In this section, we visualize some representations of Sybio-GNN, including the learned joint-scale actional graphs and their low dimensional manifolds. Moreover, we show some predicted motions to evaluate the model qualitatively.

6.4.1 *Joint-Scale Actional Graphs*

We first show the learned joint-scale actional graphs on four motions in Human3.6M. Fig. 12 highlights a few edges whose weights are greater than 0.5. We see: 1) The joint-scale actional graphs capture some action-based long-range relations beyond direct bone-connections; 2) Some reasonable relations associated with the motions themselves are captured, e.g., for 'Greeting', the stretched arms are correlated to other joints; 3) For motions with the same category, we tend to obtain the similar graphs; see two plots of 'Walking', while different classes of motions have distinct actional graphs; see 'Walking' and the other motions, where our model learns the discriminative patterns from data.

6.4.2 *Manifolds of Joint-Scale Actional Graphs*

To verify how discriminative the patterns embedded in the joint-scale actional graphs, we visualize the low-dimension manifolds of different joint-scale actional graphs. We select

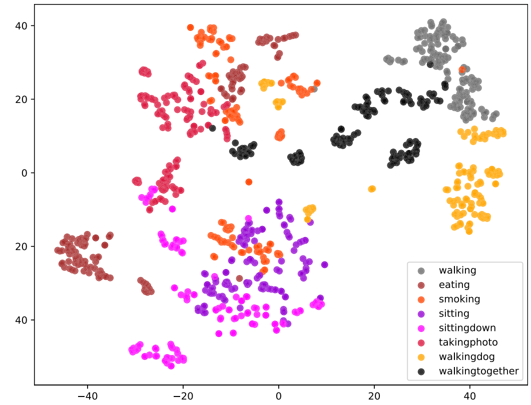


Fig. 13. 2D T-SNE map of learned actional graphs corresponding to 8 activities in H3.6M. Walking-related graphs are separated from sitting-related graphs with a large margin.

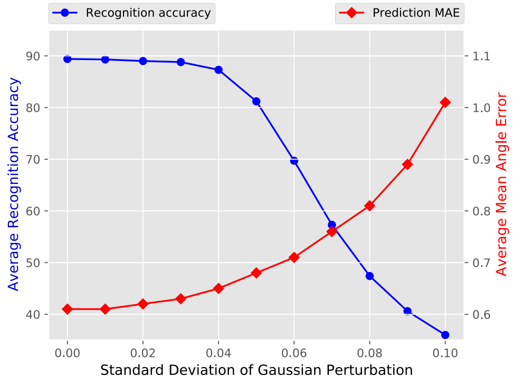


Fig. 15. The recognition accuracy and prediction MAE perturbed Gaussian noises with different standard deviations.

8 representative classes of actions in Human3.6M and sample more clips from long test motion sequences. Here we treat all the joint-scale actional graphs as vectors and obtain their 2D T-SNE map; see Fig. 13. We see that 'Walking', 'Walking Dog' and 'Walking Together', which have the common walking dynamics, are distributed closely, as well as 'Sitting' and 'Sitting Down' are clustered; however, walking-related actions and sitting-related actions are separated with a large margin; as for 'Eating', 'Smoking' and 'Taking Photo', they have similar movements on arms, showing a new cluster.

6.4.3 Predicted Sequences

Finally, to qualitatively show the prediction performances, we compare the generated samples of Sybio-GNN to those of Res-sup [15] and CSM [26] on Human3.6M. We also show the prediction errors by plotting a line segment that connects the predicted and ground-truth positions of each corresponding joint. A longer line segment indicates a larger prediction error on one joint. Fig. 14 illustrates the predictions of 'Walking' over 1,000 ms with the frame interval of 40 ms. Compared to baselines, Sybio-GNN has significantly better performances. The poses generated by Res-sup has large early discontinuity, where the error map shows several long line segments on knees and feet. CSM overcomes the early discontinuity to some extent, while the errors become large after the 600th millisecond. Sybio-GNN completes the action accurately and reasonably.

6.5 Robustness Against Input Perturbation

According to Theorem 1, we present that Sybio-GNN is robust against perturbation on inputs, where we calculate an upper bound of output deviation. To verify the stability, we add Gaussian noises sampled from $\mathcal{N}(0, \sigma^2)$ on input actions. We show the recognition accuracies on NTU-RGB+D (Cross-Subject) and short-term prediction MAEs on Human3.6M with standard deviation σ varied from 0.01 to 0.1. The recognition/prediction performances with different σ are illustrated in Fig. 15. We see: 1) for action recognition, Sybio-GNN stays a high accuracy when the noise has $\sigma \leq 0.04$, but it tends to deteriorate due to severe perturbation when $\sigma > 0.04$; 2) for motion prediction, Sybio-GNN produces precise poses when the noise has $\sigma < 0.03$, but the

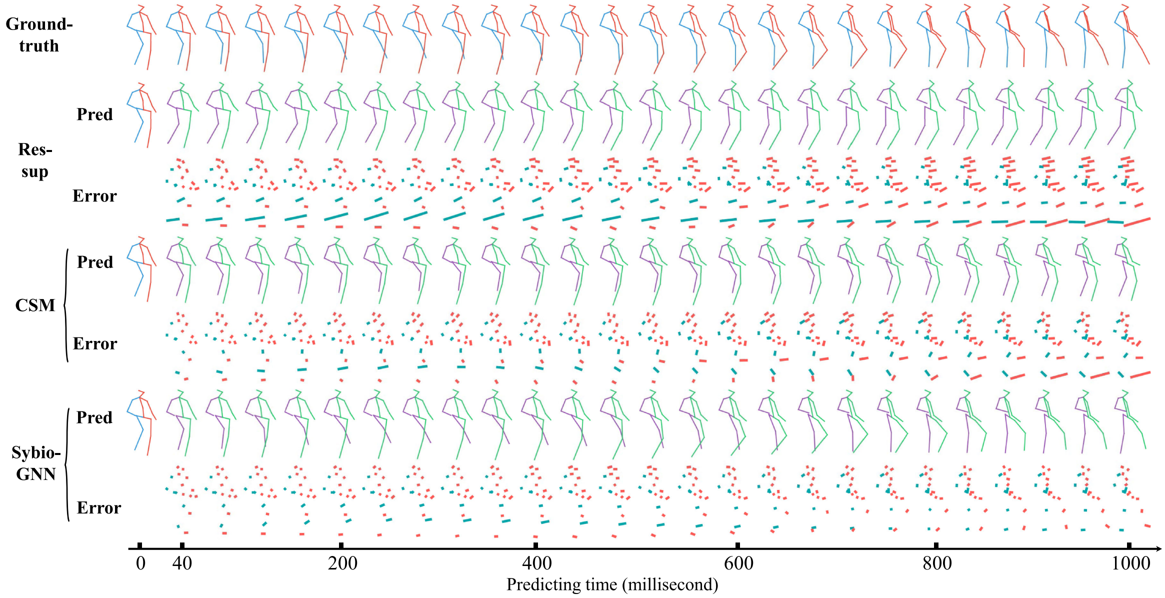


Fig. 14. Visualization of motion prediction on Human3.6M over the future 100 millisecond. We shows the predictions of 'Waling' in Human3.6M. We compare the predictions of Sybio-GNN, Res-sup, and CSM to the ground truth (GT).

prediction performance is degraded for larger σ. In all, Sybio-GNN is robust against small perturbation. More experimental analysis of model robustness and verification of Theorem 1 are presented in Appendix, available in the online supplemental material.

7 Conclusion

In this paper, we propose a novel symbiotic graph neural network (Sybio-GNN), which handles action recognition and motion prediction jointly and use graph-based operations to capture action patterns. Our model consists of a backbone, an action-recognition head, and a motion-prediction head, where the two heads enhance each other. As building components in the backbone and the motion-prediction head, graph convolution operators based on learnable joint-scale and part-scale graphs are used to extract spatial information. We conduct extensive experiments for action recognition and motion prediction with four datasets, NTU-RGB+D, Kinetics, Human3.6M, and CMU Mocap. Experiments show that our model achieves consistent improvements compared to the previous methods.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2019YFB1804304), SHEITC (No. 2018-RGZN-02046), 111 plan (No. BP0719010), and STCSM (No. 18DZ2270700), and State Key Laboratory of UHD Video and Audio Production and Presentation. Parts of this paper appear in [1]. This work was done when S. Chen was working at Mitsubishi Electric Research Laboratories (MERL).

References

[1] 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.*, 2019, pp. 3590–3598.

[2] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, “A string of feature graphs model for recognition of complex activities in natural videos,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2595–2602.

[3] D. Huang and K. Kitani, “Action-reaction: Forecasting the dynamics of human interaction,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 489–504.

[4] L.-Y. Gui, K. Zhang, Y.-X. Wang, X. Liang, J. M. F. Moura, and M. Veloso, “Teaching robots to predict human motion,” in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2018, pp. 562–567.

[5] S. Mathe and C. Sminchisescu, “Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1408–1424, Jul. 2015.

[6] Y. Shi, B. Fernando, and R. Hartley, “Action anticipation with RBF kernelized feature mapping RNN,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 301–317.

[7] D. Wang, W. Ouyang, W. Li, and D. Xu, “Dividing and aggregating network for multi-view action recognition,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 451–476.

[8] X. Liang, K. Gong, X. Shen, and L. Lin, “Look into person: Joint body parsing pose estimation network and a new benchmark,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 871–885, Apr. 2019.

[9] C. Li, Q. Zhong, D. Xie, and S. Pu, “Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 786–792.

[10] R. Vemulapalli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3D skeletons as points in a lie group,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 588–595.

[11] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1110–1118.

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

[13] A. M. Lehrmann, P. V. Gehler, and S. Nowozin, “Efficient nonlinear Markov models for human motion,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1314–1321.

[14] 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.*, 2016, pp. 5308–5317.

[15] J. Martinez, M. J. Black, and J. Romero, “On human motion prediction using recurrent neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4674–4683.

[16] L. Gui, Y. Wang, X. Liang, and J. Moura, “Adversarial geometry-aware human motion prediction,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 786–803.

[17] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1290–1297.

[18] M. Hussein, M. Torki, M. Gowayyed, and M. El-Saban, “Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2466–2472.

[19] J. Wang, A. Hertzmann, and D. Fleet, “Gaussian process dynamical models,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 1441–1448.

[20] B. Fernando, E. Gavves, M. J. Oramas, A. Ghodrati, and T. Tuytelaars, “Modeling video evolution for action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5378–5387.

[21] T. S. Kim and A. Reiter, “Interpretable 3D human action analysis with temporal convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1623–1631.

[22] M. Liu, H. Liu, and C. Chen, “Enhanced skeleton visualization for view invariant human action recognition,” *Pattern Recognit.*, vol. 68, pp. 346–362, Aug. 2017.

[23] D. Pavllo, D. Grangier, and M. Auli, “QuaterNet: A quaternion-based recurrent model for human motion,” in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 1–14.

[24] L. Gui, Y. Wang, D. Ramanan, and J. Moura, “Few-shot human motion prediction via meta-learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 432–450.

[25] J. Kundu, M. Gor, and R. Babu, “Bihmp-GAN: Bidirectional 3D human motion prediction GAN,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8553–8560.

[26] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, “Convolutional sequence to sequence model for human dynamics,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5226–5234.

[27] C. Si, Y. Jing, W. Wang, L. Wang, and T. Tan, “Skeleton-based action recognition with spatial reasoning and temporal stack learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 106–121.

[28] 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.*, 2019, pp. 1227–1236.

[29] Z. Huang, C. Wan, T. Probst, and L. Van Gool, “Deep learning on lie groups for skeleton-based action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1243–1252.

[30] K. C. Thakkar and P. J. Narayanan, “Part-based graph convolutional network for action recognition,” *CoRR*, vol. abs/1809.04983, 2018.

[31] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, “Ntu RGB+D: A large scale dataset for 3D human activity analysis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1010–1019.

[32] C. Ionescu, P. Papaca, V. Olaru, and C. Sminchisescu, “Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, 2014.

[33] Y. Yacoob and M. J. Black, “Parameterized modeling and recognition of activities,” *Comput. Vis. Image Understanding*, vol. 73, no. 2, pp. 232–247, 1999.

[34] J. Liu, A. Shahroudy, D. Xu, and G. Wang, “Spatio-temporal LSTM with trust gates for 3D human action recognition,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 816–833.

[35] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12018–12027.

[36] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7904–7913.

[37] B. Li, X. Li, Z. Zhang, and F. Wu, "Spatio-temporal graph routing for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8561–8568.

[38] Y. Wen, L. Gao, H. Fu, F. Zhang, and S. Xia, "Graph CNNs with motif and variable temporal block for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8989–8996.

[39] V. Pavlovic, J. Rehg, and J. MacCormick, "Learning switching linear models of human motion," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 942–948.

[40] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: Video forecasting by generating pose futures," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3352–3361.

[41] T. Xue, J. Wu, K. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 91–99.

[42] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4346–4354.

[43] N. Verma, E. Boyer, and J. Verbeek, "FeaStNet: Feature-steered graph convolutions for 3D shape analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2598–2606.

[44] D. Valsesia, G. Fracastoro, and E. Magli, "Learning localized generative models for 3D point clouds via graph convolution," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–15.

[45] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–20.

[46] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[47] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.

[48] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[49] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proc. Int. World-Wide Web Conf.*, 1998, pp. 107–117.

[50] S. Chen, D. Tian, C. Feng, A. Vetro, and J. Kovačević, "Fast resampling of three-dimensional point clouds via graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 666–681, Feb. 2018.

[51] Y.-F. Song, Z. Zhang, and L. Wang, "Richly activated graph convolutional network for action recognition with incomplete skeletons," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1–5.

[52] E. Ohn-Bar and M. M. Trivedi, "Joint angles similarities and HOG2 for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2013, pp. 465–470.

[53] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," *J. Vis. Commun. Image Representation*, vol. 25, no. 1, pp. 24–38, 2014.

[54] R. Vemulapalli and R. Chellapa, "Rolling rotations for recognizing human actions from 3D skeletal data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4471–4479.

[55] K. Cho and X. Chen, "Classifying and visualizing motion capture sequences using deep neural networks," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2014, pp. 122–130.

[56] J. Weng, C. Weng, and J. Yuan, "Spatio-temporal Naive-Bayes nearest-neighbor (ST-NBNN) for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 445–454.

[57] C. Li, Q. Zhong, D. Xie, and S. Pu, "Skeleton-based action recognition with convolutional neural networks," in *Proc. IEEE Int. Conf. Multimedia Expo Workshop*, 2017, pp. 597–600.

[58] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1012–1020.

[59] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. Int. Conf. Comput. Statist.*, 2010, pp. 177–187.

[60] X. Zhang, C. Xu, X. Tian, and D. Tao, "Graph edge convolutional neural networks for skeleton-based action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3047–3060, Aug. 2020.

[61] G. Hu, B. Cui, and S. Yu, "Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2019, pp. 1216–1221.

[62] X. Gao, W. Hu, J. Tang, J. Liu, and Z. Guo, "Optimized skeleton-based action recognition via sparsified graph regression," in *Proc. ACM Int. Conf. Multimedia*, 2019, pp. 601–610.

[63] 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.*, 2017, pp. 1302–1310.

[64] Y. Tang, Y. Tian, J. Lu, P. Li, and J. Zhou, "Deep progressive reinforcement learning for skeleton-based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5323–5332.

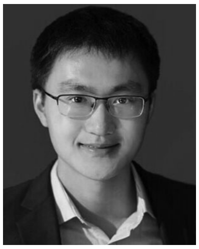
[65] P. Ghosh, J. Song, E. Aksan, and O. Hilliges, "Learning human motion models for long-term predictions," *CoRR*, vol. abs/1704.02827, 2017.

[66] H. Chiu, E. Adeli, B. Wang, D. Huang, and J. Niebles, "Action-agnostic human pose forecasting," *CoRR*, vol. abs/1810.09676, 2018.

[67] X. Guo and J. Choi, "Human motion prediction via learning local structure representations and temporal dependencies," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 2580–2587.

[68] A. Gopalakrishnan, A. Mali, D. Kifer, L. Giles, and A. G. Ororbia, "A neural temporal model for human motion prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12108–12117.

Maosen Li (Student Member, IEEE) recieved the BE degree in optical engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017. He is currently working toward the PhD degree from the Cooperative Meidianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China, since 2017. His research interests include computer vision, machine learning, graph representation learning, and video analysis. He is the reviewer of some prestigious international journals and conferences, including the *International Journal of Computer Vision*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Multimedia*, and *Pattern Recognition*, and AAAI.



Siheng Chen (Member, IEEE) received the bachelor's degree in electronics engineering from the Beijing Institute of Technology, Beijing, China, in 2011, the master's degrees in electrical and computer engineering and machine learning from Carnegie Mellon University, Pittsburgh, Pennsylvania, and the doctorate degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, Pennsylvania, in 2016. He is currently an associate professor at Shanghai Jiao Tong University. Before that, he was a research scientist with Mitsubishi Electric Research Laboratories (MERL). Before joining MERL, he was an autonomy engineer with Uber Advanced Technologies Group, working on the perception and prediction systems of self-driving cars. Before joining Uber, he was a postdoctoral research associate with Carnegie Mellon University. He was the recipient of the 2018 IEEE Signal Processing Society Young Author Best Paper Award. His coauthored paper received the Best Student Paper Award at IEEE GlobalSIP 2018. He organized the special session "Bridging graph signal processing and graph neural networks" at ICASSP 2020. His research interests include graph signal processing, graph neural networks, and autonomous driving.

Xu Chen received the BE degree in electronics engineering from Xidian University, Xi'an, China, in 2016. He is currently working toward the dual PhD degrees from Cooperative Meidianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China, and the University of Technology Sydney, Ultimo, Australia. His research interests include machine learning, graph representation learning, recommendation systems, and computer vision.

Ya Zhang (Member, IEEE) received the bachelor's degree from Tsinghua University, Beijing, China, and the PhD degree in information sciences and technology from Pennsylvania State University, State College, Pennsylvania. She is currently a professor at the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University. Her research interest include mainly in machine learning with applications to multimedia and healthcare. Before joining Shanghai Jiao Tong University, she was a research manager with Yahoo! Labs, where she led an R&D team of researchers with strong backgrounds in data mining and machine learning to improve the web search quality of Yahoo international markets. Prior to joining Yahoo, she was an assistant professor with the University of Kansas with a research focus on machine learning applications in bioinformatics and information retrieval. She has published more than 70 refereed papers in prestigious international conferences and journals, including the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Neural Networks and Learning Systems*, ICDM, CVPR, ICCV, ECCV, and ECML. She currently holds five US patents and four Chinese patents and has nine pending patents in the areas of multimedia analysis. She was appointed the chief expert for the 'Research of Key Technologies and Demonstration for Digital Media Self-organizing' project under the 863 program by the Ministry of Science and Technology of China.

Yanfeng Wang received the BE degree in information engineering from the University of PLA, Beijing, China, and the MS and PhD degrees in business management from the Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, China. He is currently the vice director of Cooperative Medianet Innovation Center and also the vice dean of the School of Electrical and Information Engineering, Shanghai Jiao Tong University. His research interest mainly include media Big Data and emerging commercial applications of information technology.

Qi Tian (Fellow, IEEE) received the BE degree in electronic engineering from Tsinghua University, Beijing, China, in 1992, the MS degree in ECE from Drexel University, Philadelphia, Pennsylvania, in 1996, and the PhD degree in ECE from the University of Illinois at Urbana-Champaign (UIUC), Champaign, Illinois, in 2002. He is currently the chief scientist of computer vision at Huawei Cloud and AI, and a full professor at the Department of Computer Science, University of Texas at San Antonio (UTSA). He was a tenured associate professor during 2008-2012 and a tenure-track assistant professor during 2002-2008. From 2008 to 2009, he took faculty leave for one year with Microsoft Research Asia (MSRA) as lead researcher with the Media Computing Group. His research interests include multimedia information retrieval, computer vision, pattern recognition. He has published more than 700 refereed journal and conference papers. He was the coauthor of a Best Paper at ACM ICMR 2015, a Best Paper at PCM 2013, a Best Paper at MMM 2013, a Best Paper at ACM ICIMCS 2012, a Top 10 percent Paper at MMSP 2011, a Best Student Paper at ICASSP 2006, a Best Student Paper Candidate at ICME 2015, and a Best Paper Candidate at PCM 2007. He received the 2017 UTSA President's Distinguished Award for Research Achievement, the 2016 UTSA Innovation Award, the 2014 Research Achievement Awards from the College of Science, UTSA, the 2010 Google Faculty Award, and the 2010 ACM Service Award. He is an associate editor of many journals and on the editorial board of the *Journal of Multimedia* (JMM) and the *Journal of Machine Vision and Applications* (MVA).