

Skeleton-Parted Graph Scattering Networks for 3D Human Motion Prediction

Maosen Li¹, Siheng Chen^{1,2} ✉, Zijing Zhang³, Lingxi Xie⁴,
Qi Tian⁴, and Ya Zhang^{1,2} ✉

¹ Cooperative Medianet Innovation Center, Shanghai Jiao Tong University

² Shanghai AI laboratory

³ Zhejiang University

⁴ Huawei Cloud & AI.

{maosen_li, sihengc, ya_zhang}@sjtu.edu.cn, zijing_zhang@163.com,
198808xc@gmail.com, tian.qi1@huawei.com

Abstract. Graph convolutional network based methods that model the body-joints' relations, have recently shown great promise in 3D skeleton-based human motion prediction. However, these methods have two critical issues: first, deep graph convolutions filter features within only limited graph spectrums, losing sufficient information in the full band; second, using a single graph to model the whole body underestimates the diverse patterns on various body-parts. To address the first issue, we propose adaptive graph scattering, which leverages multiple trainable band-pass graph filters to decompose pose features into richer graph spectrum bands. To address the second issue, body-parts are modeled separately to learn diverse dynamics, which enables finer feature extraction along the spatial dimensions. Integrating the above two designs, we propose a novel skeleton-parted graph scattering network (SPGSN). The cores of the model are cascaded multi-part graph scattering blocks (MPGSBs), building adaptive graph scattering on diverse body-parts, as well as fusing the decomposed features based on the inferred spectrum importance and body-part interactions. Extensive experiments have shown that SPGSN outperforms state-of-the-art methods by remarkable margins of 13.8%, 9.3% and 2.7% in terms of 3D mean per joint position error (MPJPE) on Human3.6M, CMU Mocap and 3DPW datasets, respectively ¹.

Keywords: Human motion prediction, adaptive graph scattering, spatial separation, bipartite cross-part fusion.

1 Introduction

3D skeleton-based human motion prediction has attracted increasing attention and shown broad applications, such as human-computer interaction [17] and autonomous driving [6]. Human motion prediction aims to generate the future human poses, in form of the 3D coordinates of a few key body joints, given

¹ The codes are available at <https://github.com/MediaBrain-SJTU/SPGSN>.

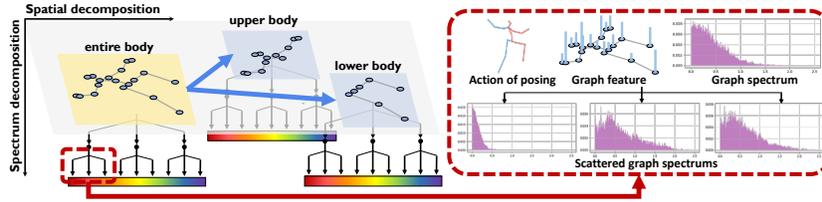


Fig. 1. Feature decomposition along the spatial and spectrum domains. For example, we separate the body into the upper and lower bodies, each of which uses a three-branch graph scattering tree with band-pass filtering to exploit rich graph spectrums.

the historical motions. Early attempts develop state models [31, 48, 55, 56] to capture the shallow dynamics. In the deep learning era, more implicit patterns are learned. For example, some recurrent-network-based methods [12, 16, 43, 58] aggregate the states and predict poses frame-by-frame; some feed-forward models [18, 32] directly output the predictions without state accumulation.

Recently, numerous graph-convolution-based models [7, 9, 35, 36, 38, 40, 41, 53] have achieved remarkable success in motion prediction by explicitly modeling the inherent body relations and extracting spatio-temporal features [27, 49, 59]. However, further development of graph-based methods encounters two critical issues. First, as long as the graph structure is given, standard graph convolution just filters the features within limited graph spectrum but cannot significantly preserve much richer bands (e.g., smoothness and difference on the graphs) at the same time. However, the pattern learning of human motions needs not only to capture the similarity or consistency of body-joints under the spatial constraints (low-frequency), but also to enhance the difference for diverse representation learning (high-frequency). For example, when the graph edge weights are purely positive or negative, deep graph convolution tends to average the distinct body-joints to be similar but ignores their specific characteristics. Second, existing methods usually use a single graph to model the whole body [40, 41], which underestimates diverse movement patterns in different body-parts [18]. For example, the upper and lower bodies have distinct motions, calling for using different graphs to represent them separately.

To address the first issue, we propose the adaptive graph scattering technique, which leverages multiple trainable band-pass graph filters arranged in a tree structure to decompose input features into various graph bands. With the mathematically designed band-specific filters, adaptive filter coefficients and feature transform layers, it preserves information from large graph spectrum. To address the second issue, we decompose a body into multiple body-parts, where comprehensive dynamics could be extracted. Therefore, our method achieves finer feature extraction along both graph spectrum and spatial dimensions. Fig. 1 sketches both the spatial and spectrum decomposition. As an example, we show three body-parts: the upper body, the lower body and the entire body. To understand the graph scattering, we show the output graph spectrums on different bands after the corresponding filtering processes.

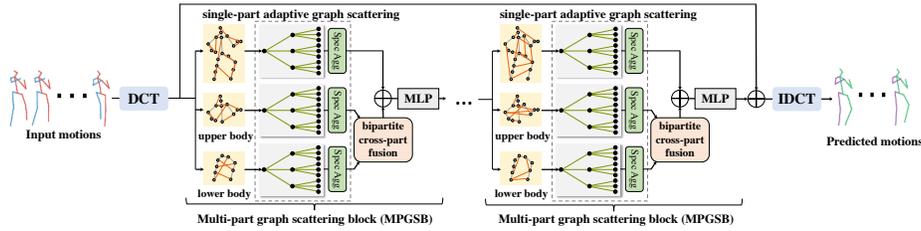


Fig. 2. Architecture of the SPGSN. The SPGSN first applies discrete cosine transform (DCT) to convert the body-joint positions along time to the frequency domain. Then, cascaded multi-part graph scattering blocks (MPGSBs) are built for deep feature extraction. Finally, we build a skip-connection between input and output features and use inverse DCT (IDCT) to recover the temporal information.

Integrating the above two designs, we propose a novel *skeleton-parted graph scattering network* (SPGSN). The core of SPGSN is the *multi-part graph scattering block* (MPGSB), consisting of two key modules: the single-part adaptive graph scattering, which uses multi-layer graph scatterings to extract spectrum features for each body-part, and bipartite cross-part fusion, which fuses body-part features based on part interactions. The SPGSN consists of multiple MPGSBs in a sequence; see Fig. 2. Taking the 3D motions as inputs, SPGSN first converts the feature along temporal dimension by discrete cosine transform (DCT) to obtain a more compact representation, which removes the complexity of temporal modeling [40,41]. Followed by the network pipeline, an inverse DCT recovers the responses to the temporal domain. A cross-model skip-connection is built to learn the residual DCT coefficients for stable prediction.

Extensive experiments are conducted for both short-term and long-term motion prediction on large-scale datasets, i.e., Human3.6M [24], CMU Mocap⁵ and 3DPW [42]. Our SPGSN significantly outperforms state-of-the-art methods in terms of mean per joint position error (MPJPE). The main contributions of our work are summarized here:

- We propose the skeleton-parted graph scattering networks (SPGSN) to promote finer feature extraction along both graph spectrum and spatial dimensions, resulting in more comprehensive feature extraction in large graph spectrum and spatially diverse dynamics learning to improve prediction.
- In SPGSN, we develop the multi-part graph scattering block (MPGSB), which contains single-part adaptive graph scattering and cross-part bipartite fusion to learn rich spectral representation and aggregate diverse part-based features for effective dynamics learning.
- We conduct experiments to verify that our SPGSN significantly outperforms existing works by 13.8%, 9.3% and 2.7% in terms of MPJPE for motion prediction on Human3.6M, CMU Mocap and 3DPW datasets, respectively.

⁵ <http://mocap.cs.cmu.edu/>

2 Related Works

2.1 Human Motion Prediction

For human motion prediction, early methods are developed based on state models [31, 48, 55, 56]. Recently, some recurrent-network-based models consider the sequential motion states. ERD [12] and Pose-VAE [58] build encoder-decoder in recurrent forms. Structural-RNN [25] transfers information between body-parts recurrently. Res-sup [43], AGED [16] and TPRNN [35] model the pose displacements in RNN models. Besides, some feed-forward networks use spatial convolutions to directly predict the whole sequences without state accumulation [18, 32]. Furthermore, considering an articulated pose, some methods exploit the correlations between body-components [4, 7, 40, 41, 53]. DMGNN [35] and MSR-GCN [9] build multiscale body graphs to capture local-global features. TrajCues [38] expands motion measurements to improve feature learning. Compared to previous models, our method leverages rich band-pass filters to preserve both smoothness and diversity of body joints and achieve more precise prediction.

2.2 Graph Representation Learning

Graphs explicitly depict the structural format [50, 62] of numerous data, such as social networks [34, 54], human poses and behaviors [5, 11, 20, 30, 39, 49, 51, 59, 60, 62–64], and dynamic systems [21, 22, 27, 29, 33, 61]. As effective methods of graph learning, some studies of the graph neural networks (GNNs) are developed to perform signal filtering based on the graph Laplacian eigen-decomposition [3, 10, 28, 65] or to aggregate vertex information [8, 19, 37, 46, 57]. Recently, graph scattering transform (GST) and related models are developed, promoting to capture rich graph spectrum with large bandwidth [15, 23, 44, 47]. GSTs generalize the image-based scattering transforms [1, 2, 14, 52], combining various graph signal filters with theoretically justified designs in terms of spectrum properties. [13, 66] develop diffusion wavelets. [44, 45] integrate designed scattering filters and parameterized feature learners. [47] expands GSTs on the spatio-temporal domain. In this work, we employ mathematical prior to initialize an adaptive graph scattering with trainable band-pass filters, filter coefficients and feature mapping.

3 Skeleton-Parted Graph Scattering Network

3.1 Problem Formulation

Skeleton-based motion prediction aims to generate the future poses given the historical ones. Mathematically, let $\mathbf{X}^{(t)} \in \mathbb{R}^{M \times 3}$ be a pose carrying the 3D coordinates of M body joints at time t , $\mathbb{X} = [\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(T)}] \in \mathbb{R}^{T \times M \times 3}$ be a three-mode tensor that concatenates moving poses within T timestamps. In motion prediction, let $\mathbb{X}^- = [\mathbf{X}^{(-T+1)}, \dots, \mathbf{X}^{(0)}] \in \mathbb{R}^{T \times M \times 3}$ represent T historical poses, $\mathbb{X}^+ = [\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(\Delta T)}] \in \mathbb{R}^{\Delta T \times M \times 3}$ represent ΔT future poses. We aim to propose a predictor $\mathcal{F}_{\text{pred}}(\cdot)$ to predict the future motions $\hat{\mathbb{X}}^+ = \mathcal{F}_{\text{pred}}(\mathbb{X}^-)$ to approximate the ground-truth \mathbb{X}^+ .

3.2 Model Architecture

Here we propose the model architecture and the operation pipeline of the *Skeleton-Parted Graph Scattering Network* (SPGSN), which is sketched in Fig. 2.

Taking the historical motion tensor $\mathbb{X}^- = [\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(T)}]$ as the input, we first apply the discrete cosine transform (DCT) along the time axis to convert the temporal dynamics of motions into the frequency domain, leading to a compact representation that eliminates the complexity of extra temporal embedding to promote easy learning [40,41]. Mathematically, we reshape \mathbb{X}^- into $\mathcal{X}^- \in \mathbb{R}^{T \times 3M}$ to consider all the joint coordinates at each timestamp independently as the basic units in the spatial domain; then we encode $\mathbf{X}^- = \text{DCT}(\mathcal{X}^-) \in \mathbb{R}^{M' \times C}$, where $M' = 3M$, and C denotes the number of DCT coefficients, also the feature dimension. In this way, although we triple the spatial scale, we compress the long sequence into a compact coefficient representation, resulting in a feature vector, and we do not need the additional sequential feature modeling. Compared to other frequency transformations, DCT fully preserves the temporal smoothness. The Fourier transform and wavelet transform usually introduce complex and multiscale responses, making the downstream modeling more complicated.

In the SPGSN, we develop a deep feed-forward architecture to learn the dynamics from the DCT-formed motion features \mathbf{X}^- . The network is constructed with cascaded *multi-part graph scattering blocks* (MPGSBs) as the core components. All MPGSBs do not share parameters, and the input of the following MPGSB is the output of the last one. In each MPGSB, the input motion is first decomposed into different body-parts. For example, Fig. 2 sketches the entire body, the upper and lower bodies, but different body separation strategies could be employed. There are trainable graphs on these body-parts. On each body-part, MPGSB takes a single-part adaptive graph scattering to preserve large-band spectrums of motion representation (see Sec. 4.1). On multiple body-parts, an bipartite cross-part fusion automatically performs body-part fusion based on the learned cross-part interaction for more coordinated motion estimation (see Sec. 4.2). Moreover, we build skip connections across all the MPGSBs, thus we force the SPGSN to capture the feature displacements for stable prediction. At the output end, we apply inverse DCT to recover the temporal information.

4 Multi-Part Graph Scattering Block

Here we present the *Multi-Part Graph Scattering Blocks* (MPGSBs). Each MPGSB contains two key modules, **1) single-part adaptive graph scattering** and **2) bipartite cross-part fusion**, to extract large graph spectrum from distinct body-parts and fuse body-parts into hybrid representation, respectively.

4.1 Single-Part Adaptive Graph Scattering

Given the determined graph topology, the information-aggregation-based graph filtering only captures features in limited spectrums, losing the rich frequency

bands of the inputs. To address this problem, the single-part adaptive graph scattering learns sufficient large-band information on each body-part. The single-part adaptive graph scattering contains two main operations: 1) *adaptive graph scattering decomposition* and 2) *adaptive graph spectrum aggregation*, which parses graph spectrum and aggregates the important bands, respectively.

Adaptive graph scattering decomposition. The adaptive graph scattering decomposition forms a tree-structure network with L layers and exponentially increasing tree nodes. These tree nodes perform graph filtering on the motion corresponding to various graph spectral bands.

We consider the first layer for example, and we could expand the design to any layers. Let the DCT-formed pose feature be $\mathbf{X} \in \mathbb{R}^{M' \times C}$, and the adaptive pose graph have adjacency matrix $\mathbf{A} \in \mathbb{R}^{M' \times M'}$. We utilize a series of band-pass graph filters: $\{h_{(k)}(\tilde{\mathbf{A}}) | k = 0, 1, \dots, K\}$, which are derived based the graph structure. Note that we have the normalized $\tilde{\mathbf{A}} = 1/2(\mathbf{I} + \mathbf{A}/\|\mathbf{A}\|_F^2)$ to handles the amplitudes of the trainable elements. Given the filter bank, $\{h_{(k)}(\tilde{\mathbf{A}})\}_{k=0}^K$, we obtain features $\{\mathbf{H}_{(k)} \in \mathbb{R}^{M' \times C'}\}_{k=0}^K$ through

$$\mathbf{H}_{(k)} = \sigma(h_{(k)}(\tilde{\mathbf{A}})\mathbf{X}\mathbf{W}_{(k)}), \quad (1)$$

where $\mathbf{W}_{(k)}$ is the trainable weights corresponding to the k th filter, and the nonlinear $\sigma(\cdot)$ (e.g. Tanh) disperses the graph frequency representation [23]. Note that $\tilde{\mathbf{A}}$ is also a parameterized matrix automatically tuned during training to adapt to the implicit interaction in motion data.

To ensure that various filters work on specific graph spectrums, we initialize $\{h_{(k)}(\tilde{\mathbf{A}})\}_{k=0}^K$ by leveraging the mathematical priors to constrain their filtering bands. Furthermore, we apply trainable coefficients in $\{h_{(k)}(\tilde{\mathbf{A}})\}_{k=0}^K$ to adaptively tune spectrum responses based on the predefined guidance; that is,

$$\begin{aligned} h_{(k)}(\tilde{\mathbf{A}}) &= \alpha_{(0,0)}\tilde{\mathbf{A}}, & k &= 0; \\ h_{(k)}(\tilde{\mathbf{A}}) &= \alpha_{(1,0)}\mathbf{I} + \alpha_{(1,1)}\tilde{\mathbf{A}}, & k &= 1; \\ h_{(k)}(\tilde{\mathbf{A}}) &= \sum_{j=1}^k \alpha_{(k,j)}\tilde{\mathbf{A}}^{2^j-1}, & k &= 2, \dots, K, \end{aligned} \quad (2)$$

where $\alpha_{(k,j)}$ is the trainable coefficient. For $k = 0$, we initialize $\alpha_{(0,0)} = 1$; for $k > 0$, we set $\alpha_{(k,k-1)} = 1$, $\alpha_{(k,k)} = -1$ and any other $\alpha_{(k,j)} = 0$. Notably, for $k > 0$, we could approximately obtain a series of graph wavelets to emphasize different frequencies. For example, we initialize $h_3(\tilde{\mathbf{A}}) = 0\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^2 - \tilde{\mathbf{A}}^4$ to depict the joint difference under the 2-order relations at the beginning. The power 2^j-1 are utilized based on diffusion wavelets, theoretically promoting optimal localization [13]. Besides, the intuition of this design is sketched in Fig. 3. Suppose all the edge weights are positive, $h_0(\tilde{\mathbf{A}})$ only preserves the low-frequency to enhance smoothness; the other filters obtain the band-pass features and promote joints' varieties. The real feature responses of the graph scattering during model inference is visualized in Appendix to analyze this module.

Plugging Eq. (2) to Eq. (1), we output the spectrum channels $\{\mathbf{H}_{(k)}\}$. At the next layer of the graph scattering, we repeat Eq. (1) on each $\mathbf{H}_{(k)}$. Thus, each

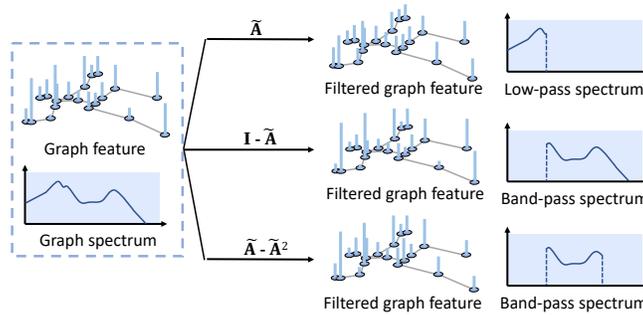


Fig. 3. Sketch of the graph filtering with graph filters. Note that we just show the initialized filters, and we apply trainable coefficients to achieve more flexible filtering.

non-leaf feature has $K + 1$ new branches; eventually, the output has $(K + 1)^L$ channels corresponding to different spectrums.

Adaptive graph spectrum aggregation. To abstract the key information from the $(K + 1)^L$ graph scattering responses and provide information to the downstream MPGSB, we propose the adaptive graph spectrum aggregation to fuse the spectral channels based on the inferred spectrum scores, which measure the importance of each channel over the whole spectrum. Given the output channels $\{\mathbf{H}_{(k)}\}$, the spectrum aggregation is formulated as

$$\mathbf{H} = \sum_{k=0}^{(K+1)^L} \omega_k \mathbf{H}_{(k)} \in \mathbb{R}^{M' \times C'}, \quad (3)$$

where ω_k is the inferred spectrum importance score of the k th feature $\mathbf{H}_{(k)}$, which is computed through

$$\omega_k = \frac{\exp(f_2(\tanh(f_1([\mathbf{H}_{\text{sp}}, \mathbf{H}_{(k)}])))})}{\sum_{j=0}^{(K+1)^L} \exp(f_2(\tanh(f_1([\mathbf{H}_{\text{sp}}, \mathbf{H}_{(j)}])))})}, \quad (4)$$

where $f_1(\cdot)$ and $f_2(\cdot)$ are MLPs, and $[\cdot, \cdot]$ is concatenation along feature dimensions. $\mathbf{H}_{\text{sp}} \in \mathbb{R}^{M' \times C'}$ carries the whole graph spectrum, which is

$$\mathbf{H}_{\text{sp}} = \text{ReLU} \left(\frac{1}{(K + 1)^L} \sum_{k=0}^{(K+1)^L} \mathbf{H}_{(k)} \mathbf{W}_{\text{sp}} \right), \quad (5)$$

where \mathbf{W}_{sp} denotes trainable weights and $\text{ReLU}(\cdot)$ is the ReLU activation. \mathbf{H}_{sp} employs the embedded spectrum to benefit the representation understanding.

For clearer understanding, Fig. 4 sketches an exemplar architecture of the single-part adaptive graph scattering ($L = 2$ and $K = 2$), where we briefly note various filtering with different subscripts. For multiple body-parts, we leverage shared parameters but different pose graphs on them to reduce the model complexity, since different body-parts carry the same feature modality but different graph views. In this way, we can capture the diverse features that reflects rich structural information.

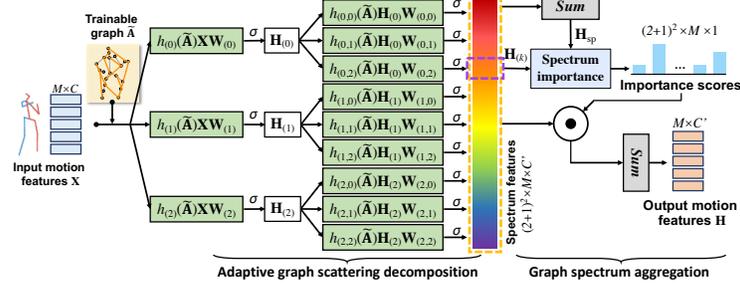


Fig. 4. A sketch of single-part adaptive graph scattering.

4.2 Bipartite Cross-Part Fusion

To combine the diverse and hybrid features learned from different body-parts, we propose a bipartite cross-part fusion module based on body-part interactions, which allow the isolated single-part features to adapt to each other, leading to more coordinated and reasonable patterns on the whole body.

In this paper, we mainly consider to separate the body into two parts besides the entire body, because according to the experiences, the separation of the two parts (e.g. separating as upper and lower bodies or left and right bodies) can distinguish the different global movement patterns and explore their interactions to reflect the movement coordination. Meanwhile, the two-part separation needs only one interaction-based fusion, which reduces the model complexity. As an example, we consider the upper-lower-separation. We first model their cross-part influence, to reflect the implicit upper-lower interactions. Two directed bipartite graphs are adaptively constructed to propagate influence from the upper to the lower bodies and vice versa. Here we present the ‘upper-to-lower’ graph as an example. Let the joint features on the upper body and lower body be $\mathbf{H}_\uparrow \in \mathbb{R}^{M_\uparrow \times C'}$ and $\mathbf{H}_\downarrow \in \mathbb{R}^{M_\downarrow \times C'}$, respectively, where M_\uparrow and M_\downarrow are the numbers of nodes in these two parts; we calculate the upper-to-lower affinity matrix through

$$\mathbf{A}_{\uparrow 2\downarrow} = \text{softmax}(f_\uparrow(\mathbf{H}_\uparrow)f_\downarrow(\mathbf{H}_\downarrow)^\top) \in [0, 1]^{M_\uparrow \times M_\downarrow}, \quad (6)$$

where $\text{softmax}(\cdot)$ is the softmax function across rows to normalize the affinity effects and enhance the strong correlations; $f_\uparrow(\cdot)$ and $f_\downarrow(\cdot)$ are two embedding networks. Each column of $\mathbf{A}_{\uparrow 2\downarrow}$ reflects the influence levels of all the upper-body-joints to the corresponding lower-body-joint.

Given the $\mathbf{A}_{\uparrow 2\downarrow}$, we update the lower body via

$$\mathbf{H}'_\downarrow = \mathbf{H}_\downarrow + \mathbf{A}_{\uparrow 2\downarrow}^\top \mathbf{H}_\uparrow, \quad (7)$$

where the new lower body aggregates the information from the upper body by emphasizing the influence across body-parts. In the similar manner, we also update the upper body based on the bipartite lower-to-upper graph.

Finally, given the updated upper and lower body, \mathbf{H}'_{\uparrow} and \mathbf{H}'_{\downarrow} , we fuse them on the whole body and obtain the hybrid feature $\mathbf{H}' \in \mathbb{R}^{M' \times C'}$ by

$$\mathbf{H}' = \text{MLP}(\mathbf{H} + (\mathbf{H}'_{\uparrow} \oplus \mathbf{H}'_{\downarrow})), \quad (8)$$

where $\oplus : \mathbb{R}^{M_{\uparrow} \times C'} \times \mathbb{R}^{M_{\downarrow} \times C'} \rightarrow \mathbb{R}^{M' \times C'}$ places joints from different body-parts to align with the original body. $\text{MLP}(\cdot)$ further embeds the fused body. In this way, the output features carry the comprehensive graph spectrum and multi-part representation to promote motion prediction.

4.3 Loss Function

To train the proposed SPGSN, we define the loss function. Suppose that we take N samples in a mini-batch as inputs, and let the n th ground-truth and predicted motion sample be \mathbb{X}_n^+ and $\widehat{\mathbb{X}}_n^+$. The loss function \mathcal{L} is defined as the average ℓ_2 distance between the targets and predictions:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N \|\mathbb{X}_n^+ - \widehat{\mathbb{X}}_n^+\|^2. \quad (9)$$

All the trainable parameters in our SPGSN are tuned end-to-end, including the body graph structures, adaptive filter coefficients and the network weights.

5 Experiments

5.1 Datasets

Dataset 1: Human 3.6M (H3.6M). There are 7 subjects performing 15 classes of actions in H3.6M [24], and each subject has 22 body joints. All sequences are downsampled by two along time. Following previous paradigms [9], the models are trained on the segmented clips in the 6 subjects and tested on the clips in the 5th subject.

Dataset 2: CMU Mocap. CMU Mocap consists of 5 general classes of actions. On each pose, we use 25 joints in the 3D space. Following [9,41], we use 8 actions: ‘basketball’, ‘basketball signal’, ‘directing traffic’, ‘jumping’, ‘running’, ‘soccer’, ‘walking’ and ‘washing window’.

Dataset 3: 3D Pose in the Wild (3DPW). 3DPW [42] contains more than 51k frames with 3D poses for indoor and outdoor activities. We adopt the training, test and validation separation suggested by the official setting. Each subject has 23 joints. The frame rate of the motions is 30Hz.

5.2 Model and Experimental Settings

Implementation details. We implement SPGSN with PyTorch 1.4 on one NVIDIA Tesla V100 GPU. We set 10 MPGSBs to form the entire model. In each MPGSB, the single-part adaptive graph scattering has $L = 2$ layers of graph scattering decomposition, and the filter order $K = 2$. The hidden dimension in a MPGSB is 256. We use Adam optimizer [26] to train the SPGSN with batch

Table 1. Prediction MPJPEs of various models for short-term motion prediction on 5 representative actions in H3.6M. We also introduce an SPGSN variant called SPGSN (1body), which only considers the entire non-separated bodies. Since the original STS-GCN [53] uses a different protocol from all the other methods, we update its code for a fair comparison; see results in STSGCN*.

Motion millisecond	Walking				Eating				Smoking				Discussion				Directions			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup [43]	29.36	50.82	76.03	81.52	16.84	30.60	56.92	68.65	22.96	42.64	70.24	83.68	32.94	61.18	90.92	96.19	35.36	57.27	76.30	87.67
CSM [32]	21.70	43.56	66.29	75.48	14.50	26.13	47.47	55.63	19.42	37.70	62.49	68.55	26.35	53.41	79.12	83.01	27.07	44.72	63.94	75.37
SkelNet [18]	20.49	34.36	59.64	68.76	11.80	22.38	39.88	48.11	11.33	23.71	45.30	52.85	21.79	40.24	65.93	77.91	16.06	27.12	62.97	72.75
DMGNN [35]	17.32	30.67	54.56	65.20	10.96	21.39	36.18	43.88	8.97	17.62	32.05	40.30	17.33	34.78	61.03	69.80	13.14	24.62	64.68	81.86
Traj-GCN [41]	12.29	23.03	39.77	46.12	8.36	16.90	33.19	40.70	7.94	16.24	31.90	38.90	12.50	27.40	58.51	71.68	8.97	19.87	43.35	53.74
HisRep [40]	10.53	19.96	34.88	42.05	7.39	15.53	31.26	38.58	7.17	14.54	28.83	35.67	10.89	25.19	56.15	69.30	7.77	18.23	41.34	51.61
MSR-GCN [9]	12.16	22.65	38.64	45.24	8.39	17.05	33.03	40.43	8.02	16.27	31.32	38.15	11.98	26.76	57.08	69.74	8.61	19.65	43.28	53.82
STSGCN* [53]	16.26	24.63	40.06	45.94	14.32	22.14	37.91	45.03	13.10	20.20	37.71	44.65	14.33	24.28	52.62	68.53	14.24	24.27	44.24	53.21
SPGSN (1body)	10.13	19.51	35.52	44.67	7.13	15.02	31.87	41.18	6.83	13.94	28.77	36.78	10.42	23.90	54.13	69.99	7.38	17.48	40.54	53.09
SPGSN	10.14	19.39	34.80	41.47	7.07	14.85	30.48	37.91	6.72	13.79	27.97	34.61	10.37	23.79	53.61	67.12	7.35	17.15	39.80	50.25

Table 2. MPJPEs for short-term motion prediction on other 9 actions in H3.6M.

Motion millisecond	Greeting				Phoning				Posing				Purchases				Sitting			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup [43]	34.46	63.36	124.60	142.50	37.96	69.32	115.00	126.73	36.10	69.12	130.46	157.08	36.33	60.30	86.53	95.92	42.55	81.40	134.70	151.78
DMGNN [35]	23.30	50.32	107.30	132.10	12.47	25.77	48.08	58.29	15.27	29.27	71.54	96.65	21.35	38.71	75.67	92.74	11.92	25.11	44.59	50.20
Traj-GCN [41]	18.65	38.68	77.74	93.39	10.24	21.02	42.54	52.30	13.66	29.89	66.62	84.05	15.60	32.78	65.72	79.25	10.62	21.90	46.33	57.91
MSR-GCN [9]	16.48	36.95	77.32	93.38	10.10	20.74	41.51	51.26	12.79	29.38	66.95	85.01	14.75	32.39	66.13	79.64	10.53	21.99	46.26	57.80
STSGCN* [53]	15.02	30.70	67.11	87.63	14.88	21.40	46.55	52.03	15.01	25.69	58.38	73.08	15.26	26.26	63.45	74.25	15.19	22.95	46.82	58.34
SPGSN (1body)	15.16	33.61	71.89	88.74	8.78	18.50	39.85	51.53	10.92	25.46	61.38	78.87	12.78	28.86	62.59	77.01	9.25	19.58	43.47	56.32
SPGSN	14.64	32.59	70.64	86.44	8.67	18.32	38.73	48.46	10.73	25.31	59.91	76.46	12.75	28.58	61.01	74.38	9.28	19.40	42.25	53.56
Motion millisecond	Sitting Down				Taking Photo				Waiting				Walking Together				Average			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup [43]	47.28	85.95	145.75	168.86	26.10	47.61	81.40	94.73	30.62	57.82	106.22	121.45	26.79	50.07	80.16	92.23	34.66	61.97	101.08	115.49
DMGNN [35]	14.95	32.88	77.06	93.00	13.61	28.95	45.99	58.76	12.20	24.17	59.62	77.54	14.34	26.67	50.08	63.22	16.95	33.62	65.90	79.65
Traj-GCN [41]	16.14	31.12	61.47	75.46	9.88	20.89	44.95	56.58	11.43	23.99	50.06	61.48	10.47	21.04	38.47	45.19	12.68	26.06	52.27	63.51
MSR-GCN [9]	16.10	31.63	62.45	76.84	9.89	21.01	44.56	56.30	10.68	23.06	48.25	59.23	10.56	20.92	37.40	43.85	12.11	25.56	51.64	62.93
STSGCN* [53]	16.70	28.05	56.15	72.03	16.61	24.84	45.98	61.79	16.30	24.33	48.12	59.79	11.38	22.39	39.90	47.48	15.34	25.52	50.64	60.61
SPGSN (1body)	14.34	28.10	58.23	74.44	8.72	18.95	42.62	55.22	9.24	20.02	43.80	56.80	8.91	18.46	34.88	42.98	10.55	22.63	48.21	60.96
SPGSN	14.18	27.72	56.75	70.74	8.79	18.90	41.49	52.66	9.21	19.79	43.10	54.14	8.94	18.19	33.84	40.88	10.44	22.33	47.07	58.26

size 32. The learning rate is 0.001 with a 0.96 decay for every two epochs. To obtain more generalized evaluation with lower test bias, we utilize all the clips in the 5th subject of H3.6M and the test folder of CMU Mocap, instead of testing on a few samples picked from the test sequences like in [16, 32, 35, 41, 43].

Baselines. We compare our model to many state-of-the-art methods, including the RNN-based Res-sup [43], feed-forward-based CSM [32], SkelNet [18], and graph-based Traj-GCN [41], DMGNN [35], HisRep [40], STSGCN [53] and MSR-GCN [9]. We test these methods under the same protocol.

Evaluation Metrics. We use the Mean Per Joint Position Error (MPJPE), where we record the average ℓ_2 distance between predicted joints and target ones in 3D Euclidean space at each prediction timestamp. Compared to previous mean angle error (MAE) [35, 43], the MPJPE reflects larger degrees of freedom of human poses and covers larger ranges of errors for clearer comparison.

5.3 Comparison to State-of-the-Art Methods

To validate SPGSN, we show the quantitative results for both short-term and long-term motion prediction on H3.6M, CMU Mocap and 3DPW. We also illustrate the predicted samples for qualitative evaluation.

Table 3. Prediction MPJPEs of methods for long-term prediction on 8 actions in H3.6M and the average MPJPEs across all the actions.

Motion millisecond	Walking		Eating		Smoking		Directions		Phoning		Sitting		TakingPhoto		Waiting		Average	
	560	1k	560	1k	560	1k	560	1k	560	1k	560	1k	560	1k	560	1k	560	1k
Res-sup. [43]	81.73	100.68	79.87	100.20	94.83	137.44	110.05	152.48	143.92	186.79	166.20	185.16	107.03	162.38	126.70	153.14	129.19	164.96
Traj-GCN [41]	54.05	59.75	53.39	77.75	50.74	72.62	71.01	101.79	69.55	104.19	77.63	118.36	78.73	120.06	79.08	107.32	81.07	113.01
DMGNN [35]	71.36	95.82	58.11	86.66	50.85	72.15	102.06	135.75	71.33	108.37	75.51	115.44	78.38	123.65	85.54	113.68	93.57	127.62
MSR-GCN [9]	52.72	63.05	52.54	77.11	49.45	71.64	71.18	100.59	68.28	104.36	78.19	120.02	77.94	121.87	76.33	106.25	81.13	114.18
STSGCN* [53]	50.64	64.74	56.46	75.08	55.55	74.13	75.61	109.89	79.19	109.88	82.32	119.83	87.70	119.79	78.41	108.04	80.66	113.33
SPGSN	46.89	53.59	49.76	73.39	46.68	68.62	70.05	100.52	66.70	102.52	75.00	116.24	75.58	118.22	73.50	103.62	77.40	109.64

Table 4. Prediction MPJPEs of methods on CMU Mocap for both short-term and long-term prediction, as well as the average prediction results across all the actions.

Motion millisecond	Basketball				Basketball Signal				Directing Traffic				Jumping							
	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
Res-sup. [43]	15.45	26.88	43.51	49.23	88.73	20.17	32.98	42.75	44.65	60.57	20.52	40.58	75.38	90.36	153.12	26.85	48.07	93.50	108.90	162.84
DMGNN [35]	15.57	28.72	59.01	73.05	138.62	5.03	9.28	20.21	26.23	52.04	10.21	20.90	41.55	52.28	111.23	31.97	54.32	96.66	119.92	224.63
Traj-GCN [41]	11.68	21.26	40.99	50.78	97.99	3.33	6.25	13.58	17.98	54.00	6.92	13.69	30.30	39.97	114.16	17.18	32.37	60.12	72.55	127.41
MST-GCN [9]	10.28	18.94	37.68	47.03	86.96	3.03	5.68	12.35	16.26	47.91	5.92	12.09	28.36	38.04	111.04	14.99	28.66	55.86	69.05	124.79
STSGCN [53]	12.56	23.04	41.92	50.33	94.17	4.72	6.69	14.53	17.88	49.52	6.41	12.38	29.05	38.86	109.42	17.52	31.48	58.74	72.06	127.40
SPGSN	10.24	18.54	38.22	48.68	89.58	2.91	5.25	11.31	15.01	47.31	5.52	11.16	25.48	37.06	108.14	14.93	28.16	56.72	71.16	125.20
Motion millisecond	Soccer				Walking				Washing Window				Average							
	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
Res-sup. [43]	17.75	31.30	52.55	61.40	107.37	44.35	76.66	126.83	151.43	194.33	22.84	44.71	86.78	104.68	202.73	24.21	43.75	76.19	88.93	139.00
DMGNN [35]	14.86	25.29	52.21	65.42	111.90	9.57	15.53	26.03	30.37	67.01	7.93	14.68	33.34	44.24	82.84	14.07	24.44	45.90	55.45	104.33
Traj-GCN [41]	13.33	24.00	43.77	53.20	108.26	6.62	10.74	17.40	20.35	34.41	5.96	11.62	24.77	31.63	66.85	9.94	18.02	33.55	40.95	81.85
MSR-GCN [9]	10.92	19.50	37.05	46.38	99.32	6.31	10.30	17.64	21.12	39.70	5.49	11.07	25.05	32.51	71.30	8.72	15.83	30.57	38.10	79.01
STSGCN [53]	13.49	25.24	39.87	51.58	109.63	7.18	10.99	17.84	22.61	44.12	6.79	12.10	24.92	36.66	69.48	10.80	18.19	31.18	41.05	81.76
SPGSN	10.86	18.99	35.05	45.16	99.51	6.32	10.21	16.34	20.19	34.83	4.86	9.44	21.50	28.37	65.08	8.30	14.80	28.64	36.96	77.82

Short-term prediction. Short-term motion prediction aims to predict the poses within 400 milliseconds. First, on H3.6M, Table 1 presents the MPJPEs of SPGSN and many previous methods on 5 representative actions at multiple prediction timestamps. We see that, SPGSN obtains superior performance at most timestamps; also, learning the diverse patterns from separated body-parts, SPGSN outperforms the variant SPGSN (1body) that only uses the entire human body. Besides, Table 2 presents the MPJPEs on other 9 actions in H3.6M and the average MPJPEs over the dataset. Compared to the baselines, SPGSN achieves much lower MPJPEs by 9.3% in average. Notably, the original STSGCN [53] uses a different protocol from all the other methods, we update its code for a fair comparison; see STSGCN* and more details are in Appendix.

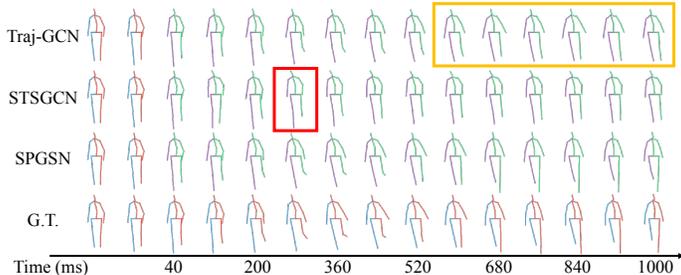
Long-term prediction. Long-term motion prediction aims to predict the poses over 400 milliseconds, which is challenging due to the pose variation and elusive human intention. Table 3 presents the prediction MPJPEs of various methods at the 560 ms and 1000 ms on 8 actions in H3.6M. We see that, SPGSN achieves more effective prediction on most actions and has lower MPJPEs by 3.6% in average. The results on the other actions are shown in Appendix.

We also test the SPGSN for both short-term and long-term prediction on CMU Mocap. Table 4 shows the MPJPEs on 7 actions within the future 1000 ms. We see that, SPGSN outperforms the baselines on most actions, and the average prediction MPJPE is much lower by 9.3% than previous methods.

Furthermore, we test the SPGSN on 3DPW dataset for both short-term and long-term motion prediction. We present the average MPJPEs across all the test samples at different prediction steps in Table 5. Compared to the state-of-the-art methods, SPGSN reduces the MPJPE by 2.7% in average.

Table 5. The average prediction MPJPEs across the test set of 3DPW at various prediction time steps.

millisecond	Average MAE							
	100	200	400	500	600	800	900	1000
Res-sup. [43]	102.28	113.24	173.94	185.35	191.47	201.39	205.12	210.58
CSM [32]	57.83	71.53	124.01	142.47	155.16	174.87	183.40	187.06
Traj-GCN [41]	16.28	35.62	67.46	80.19	90.36	106.79	113.93	117.84
DMGNN [35]	17.80	37.11	70.38	83.02	94.12	109.67	117.25	123.93
HisRep [40]	15.88	35.14	66.82	78.49	93.55	107.63	114.59	114.75
MSR-GCN [9]	15.70	33.48	65.02	77.59	93.81	108.15	114.88	116.31
STSGCN [53]	18.32	37.79	67.51	77.34	92.75	106.65	113.14	112.22
SPGSN	15.39	32.91	64.54	76.23	91.62	103.98	109.41	111.05

**Fig. 5.** Prediction samples of different methods on action ‘Walking’ of H3.6M for long-term prediction. The predictions of Traj-GCN collapse to static ‘mean poses’ after the 600th ms (orange box); STSGCN starts to suffer from large errors at the 280th ms (red box); SPGSN completes the action more accurately.

Prediction Visualization. To qualitatively evaluate the prediction, we compare the synthesized samples of SPGSN to those of Traj-GCN and STSGCN on H3.6M. Figure. 5 illustrates the future poses of ‘Walking’ in 1000 ms with the frame interval of 80 ms. Compared to the baselines, SPGSN completes the action more accurately. The predictions of Traj-GCN collapse to static ‘mean poses’ after the 600th ms (orange box). STSGCN starts to suffer from large errors at the 280th ms (red box). See more results in Appendix.

5.4 Model Analysis

Numbers of MPGSBs and graph scattering layers. We test the SPGSN frameworks with various numbers of MPGSBs (8-11) and layers of adaptive graph scattering (1-3) on H3.6M for both short-term and long-term prediction. Table 6 presents the average MPJPEs of different architectures. The SPGSN with 10 MPGSBs and 2 layers of adaptive graph scattering obtains the lowest MPJPEs. The models have stable prediction for 9-11 MSGCBs, and 2 or 3 graph scattering layers usually show better results than only using one layer.

Numbers of spectral channels. We investigate the numbers of graph scattering decomposition channels, K , in each MPGSB. We test the model that

Table 6. Performance analysis of SPGSNs with different numbers of graph scattering blocks and graph scattering layers.

MPGSB number	Scatter-layers			Average MPJPE					
	1	2	3	80	160	320	400	560	1000
8	✓			11.34	23.86	49.66	61.22	80.32	114.26
		✓		11.05	23.27	48.58	59.97	78.81	112.29
			✓	10.88	23.12	48.43	59.76	79.32	112.86
9	✓			10.91	23.22	48.39	59.63	79.31	113.09
		✓		10.58	22.56	47.49	58.73	78.49	112.06
			✓	10.40	22.33	47.29	58.58	77.79	110.55
10	✓			10.48	22.41	47.53	58.78	78.81	111.29
		✓		10.44	22.33	47.07	58.26	77.40	109.64
			✓	10.46	22.46	47.25	58.48	77.86	110.64
11	✓			10.51	22.39	47.24	58.42	77.91	110.35
		✓		10.57	22.53	47.15	58.49	79.06	112.05
			✓	10.66	22.69	47.43	58.66	79.18	112.83

Table 7. SPGSNs with different numbers of graph scattering channels on each non-leaf scattering feature.

channel numbers ($K+1$)	Average MAE					
	80	160	320	400	560	1000
1	12.03	25.87	51.40	61.98	80.52	113.15
2	10.50	22.39	47.40	58.44	77.53	110.35
3	10.44	22.33	47.07	58.26	77.40	109.64
4	10.47	22.41	47.23	58.32	77.37	109.89
5	10.57	22.46	47.38	58.46	77.66	110.52

applies 1 to 5 channels on H3.6M; see the average MPJPEs in Table 7. We see that 3 channels lead to the lowest prediction errors. The model with only 1 channel cannot capture the sufficiently large spectrums and rich information. Five channels cause much heavy model and over-fitting.

Bipartite cross-part interaction. To investigate the proposed body-part separation and cross-part interaction, we compare the SPGSN with two model variants. First, SPGSN that directly aligns the independent part features on the entire body; second, SPGSN separates the body into left and right parts. We test these variants on H3.6M; see Table 8. The SPGSN with upper-lower interaction and fusion consistently outperforms the baselines. For SPGSN (no CrossPart), due to the lack of mutual influence, it is hard to ensure higher coordination and rationality. As for SPGSN (left-right), given the interaction of body-parts, the prediction error is reduced, while the upper-lower separation promotes more special dynamics than the left-right separation due to the movement diversity.

To verify the inferred interactions between the upper and lower bodies, we visualize the learned bipartite graphs on different actions. We show the upper-to-lower and lower-to-upper graphs on ‘walking’ and ‘posing’, where we plot the edges whose weights are larger than 0.25; see Fig. 6. Different actions reflect different bipartite graphs: walking connects the contralateral hands and feet on both upper-to-lower and lower-to-upper graphs; posing connects the ipsilateral joints on the two body-parts, which delivers near-torso features to the body.

Table 8. Comparison between SPGSN and its variants, including the model without cross-part fusion, i.e., SPGSN (no CrossPart), and the model separating the body into left and right parts, i.e., SPGSN (left-right).

prediction time	80	160	320	400	560	1000
SPGSN (no CrossPart)	10.55	22.63	48.21	60.96	78.02	111.74
SPGSN (left-right)	10.47	22.51	47.48	58.85	77.79	110.16
SPGSN (ours, upper-lower)	10.44	22.33	47.07	58.26	77.40	109.64

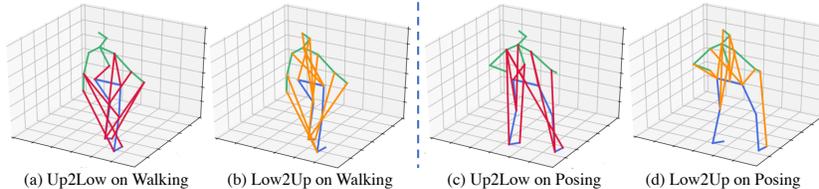


Fig. 6. Inferred directed bipartite cross-part graphs on posing and walking. We denote the upper-to-lower and the lower-to-upper graphs as ‘Up2Low’ and ‘Low2Up’.

Table 9. Comparison of running time, model sizes and MPJPE

	DMGNN [35]	Traj-GCN [41]	MSR-GCN [9]	STSGCN [53]	HisRep [40]	SPGSN (Ours)
RunTime (ms)	33.13	26.35	42.36	17.6	31.57	30.07
ParaSize (M)	4.82	2.56	6.30	0.04	3.18	5.66
FLOPs (M)	2.82	0.49	3.89	1.35	3.08	1.77
MPJPE	49.03	38.63	38.06	38.03	36.41	34.53

Efficiency Analysis. To verify the applicability of SPGSN, we compare SPGSN to existing methods in terms of the running times, parameter numbers, FLOPs and prediction results in short-term prediction on H3.6M (Table 9). SPGSN has the lowest MPJPE and efficient running based on the parallel computation. SPGSN also has the acceptable model size.

6 Conclusion

We propose a novel skeleton graph scattering network for human motion prediction, which contains cascaded multi-part graph scattering blocks (MPGSBs) to capture fine representation along both spatial and spectrum dimensions. Each MPGSB builds adaptive graph scattering on separated body-parts. In this way, the model carries large graph spectrum and considers the distinct part-based dynamics for precise motion prediction. Experiments reveal the effectiveness of our model for motion prediction on Human3.6M, CMU Mocap and 3DPW datasets.

Acknowledgements: This work is supported by the National Key Research and Development Program of China (2020YFB1406801), the National Natural Science Foundation of China under Grant (62171276), 111 plan (BP0719010), STCSM (18DZ2270700, 21511100900), State Key Laboratory of UHD Video and Audio Production and Presentation.

References

1. Andén, J., Mallat, S.: Deep scattering spectrum. *IEEE Transactions on Signal Processing* **62**(16), 4114–4128 (2014)
2. Bruna, J., Mallat, S.: Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8), 1872–1886 (2013)
3. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. In: *ICLR* (Apr 2014)
4. Cai, Y., Huang, L., Wang, Y., Cham, T.J., Cai, J., Yuan, J., Liu, J., Yang, X., Zhu, Y., Shen, X., Liu, D., Liu, J., Thalmann, N.M.: Learning progressive joint propagation for human motion prediction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) *ECCV*. pp. 226–242 (2020)
5. Chen, G., Song, X., Zeng, H., Jiang, S.: Scene recognition with prototype-agnostic scene layout. *IEEE Transactions of Image Processing* **29**(*IEEE Transactions on Image Processing*), 5877–5888 (2020)
6. Chen, S., Liu, B., Feng, C., Vallespi-Gonzalez, C., Wellington, C.: 3d point cloud processing and learning for autonomous driving. *IEEE Signal Processing Magazine* (2020)
7. Cui, Q., Sun, H., Yang, F.: Learning dynamic relationships for 3d human motion prediction. In: *CVPR* (June 2020)
8. Dai, H., Dai, B., Song, L.: Discriminative embeddings of latent variable models for structured data. In: *ICML* (June 2016)
9. Dang, L., Nie, Y., Long, C., Zhang, Q., Li, G.: Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 11467–11476 (October 2021)
10. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: *NeurIPS* (Dec 2016)
11. Fan, L., Wang, W., Huang, S., Tang, X., Zhu, S.C.: Understanding human gaze communication by spatio-temporal graph reasoning. In: *ICCV* (Oct 2019)
12. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: *ICCV*. pp. 4346–4354 (December 2015)
13. Gama, F., Ribeiro, A., Bruna, J.: Diffusion scattering transforms on graphs. In: *ICLR* (May 2019)
14. Gama, F., Ribeiro, A., Bruna, J.: Stability of graph scattering transforms. In: *NeurIPS*. vol. 32 (December 2019)
15. Gao, F., Wolf, G., Hirn, M.: Geometric scattering for graph data analysis. In: *ICML*. pp. 2122–2131 (June 2019)
16. Gui, L., Wang, Y., Liang, X., Moura, J.: Adversarial geometry-aware human motion prediction. In: *ECCV*. pp. 786–803 (Sept 2018)
17. Gui, L., Zhang, K., Wang, Y., Liang, X., Moura, J., Veloso, M.: Teaching robots to predict human motion. In: *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Oct 2018)
18. Guo, X., Choi, J.: Human motion prediction via learning local structure representations and temporal dependencies. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 33, pp. 2580–2587 (2019)
19. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: *NeurIPS* (Dec 2017)
20. Hu, G., Cui, B., Yu, S.: Skeleton-based action recognition with synchronous local and non-local spatio-temporal learning and frequency attention. In: *ICME* (July 2019)

21. Hu, Y., Chen, S., Zhang, Y., Gu, X.: Collaborative motion prediction via neural motion message passing. In: CVPR (June 2020)
22. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In: ICCV. pp. 6272–6281 (2019)
23. Ioannidis, V.N., Chen, S., Giannakis, G.B.: Pruned graph scattering transforms. In: ICLR (Apr 2020)
24. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on Pattern Analysis and Machine Intelligence* **36**(7), 1325–1339 (2013)
25. Jain, A., Zamir, A., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: CVPR. pp. 5308–5317 (June 2016)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980 (2014)
27. Kipf, T., Fetaya, E., Wang, K.C., Welling, M., Zemel, R.: Neural relational inference for interacting systems. In: ICML. pp. 2688–2697 (2018)
28. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (Apr 2017)
29. Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofghi, S.H., Savarese, S.: Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. arXiv preprint arXiv:1907.03395 (2019)
30. Lee, S., Lim, J., Suh, I.H.: Progressive feature matching: Incremental graph construction and optimization. *IEEE Transactions of Image Processing* **29**, 6992–7005 (2020)
31. Lehrmann, A., Gehler, P., Nowozin, S.: Efficient nonlinear markov models for human motion. In: CVPR. pp. 1314–1321 (June 2014)
32. Li, C., Zhang, Z., Sun Lee, W., Hee Lee, G.: Convolutional sequence to sequence model for human dynamics. In: CVPR (June 2018)
33. Li, J., Yang, F., Tomizuka, M., Choi, C.: Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *NeurIPS* (2020)
34. Li, M., Chen, S., Zhang, Y., Tsang, I.: Graph cross networks with vertex infomax pooling. In: *NeurIPS*. vol. 33, pp. 14093–14105 (2020)
35. Li, M., Chen, S., Zhao, Y., Zhang, Y., Wang, Y., Tian, Q.: Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In: CVPR (June 2020)
36. Li, M., Chen, S., Zhao, Y., Zhang, Y., Wang, Y., Tian, Q.: Multiscale spatio-temporal graph neural networks for 3d skeleton-based motion prediction. *IEEE Transactions on Image Processing* **30**, 7760–7775 (2021)
37. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. In: ICLR (May 2016)
38. Liu, Z., Su, P., Wu, S., Shen, X., Chen, H., Hao, Y., Wang, M.: Motion prediction using trajectory cues. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 13299–13308 (October 2021)
39. Lu, X., Wang, W., Danelljan, M., Zhou, T., Shen, J., Van Gool, L.: Video object segmentation with episodic graph memory networks. In: *ECCV*. pp. 661–679 (2020)
40. Mao, W., Liu, M., Salzmann, M.: History repeats itself: Human motion prediction via motion attention. In: *ECCV* (Aug 2020)
41. Mao, W., Liu, M., Salzmann, M., Li, H.: Learning trajectory dependencies for human motion prediction. In: *ICCV* (Oct 2019)

42. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: ECCV (September 2018)
43. Martinez, J., Black, M., Romero, J.: On human motion prediction using recurrent neural networks. In: CVPR. pp. 4674–4683 (July 2017)
44. Min, Y., Wenkel, F., Wolf, G.: Scattering gcn: Overcoming oversmoothness in graph convolutional networks. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 14498–14508 (Dec 2020)
45. Min, Y., Wenkel, F., Wolf, G.: Geometric scattering attention networks. In: ICASSP. pp. 8518–8522 (2021)
46. Niepert, M., Ahmed, M., Kutzkovi, K.: Learning convolutional neural networks for graphs. In: ICML (June 2016)
47. Pan, C., Chen, S., Ortega, A.: Spatio-temporal graph scattering transform. In: ICLR (May 2021)
48. Pavlovic, V., Rehg, J.M., MacCormick, J.: Learning switching linear models of human motion. In: NeurIPS (2001)
49. Qi, S., Wang, W., Jia, B., Shen, J., Zhu, S.C.: Learning human-object interactions by graph parsing neural networks. In: ECCV. pp. 401–417 (2018)
50. Rizkallah, M., Su, X., Maugey, T., Guillemot, C.: Geometry-aware graph transforms for light field compact representation. *IEEE Transactions of Image Processing* **29**, 602–616 (2020)
51. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with directed graph neural networks. In: CVPR (June 2019)
52. Sifre, L., Mallat, S.: Rotation, scaling and deformation invariant scattering for texture discrimination. In: CVPR. pp. 1233–1240 (June 2013)
53. Sofianos, T., Sampieri, A., Franco, L., Galasso, F.: Space-time-separable graph convolutional network for pose forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11209–11218 (October 2021)
54. Tabassum, S., Pereira, F.S., Fernandes, S., Gama, J.: Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **8**(5) (2018)
55. Taylor, G., Hinton, G.: Factored conditional restricted Boltzmann machines for modeling motion style. In: ICML (June 2009)
56. Taylor, G., Hinton, G., Roweis, S.: Modeling human motion using binary latent variables. In: NeurIPS (December 2007)
57. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (Apr 2018)
58. Walker, J., Marino, K., Gupta, A., Hebert, M.: The pose knows: Video forecasting by generating pose futures. In: ICCV. pp. 3332–3341 (Oct 2017)
59. Wang, W., Zhu, H., Dai, J., Pang, Y., Shen, J., Shao, L.: Hierarchical human parsing with typed part-relation reasoning. In: CVPR (June 2020)
60. Xu, C., Chen, S., Li, M., Zhang, Y.: Invariant teacher and equivariant student for unsupervised 3d human pose estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 3013–3021 (2021)
61. Xu, C., Li, M., Ni, Z., Zhang, Y., Chen, S.: Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6498–6507 (2022)
62. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI (Feb 2018)

63. Zhang, J., Shen, F., Xu, X., Shen, H.T.: Temporal reasoning graph for activity recognition. *IEEE Transactions of Image Processing* **29**, 5491–5506 (2020)
64. Zhang, X., Xu, C., Tian, X., Tao, D.: Graph edge convolutional neural networks for skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems* **31**(8), 3047–3060 (2019)
65. Zheng, C., Pan, L., Wu, P.: Multimodal deep network embedding with integrated structure and attribute information. *IEEE Transactions on Neural Networks and Learning Systems* **31**(5), 1437–1449 (2020)
66. Zou, D., Lerman, G.: Graph convolutional neural networks via scattering. *Applied and Computational Harmonic Analysis* **49**(3), 1046–1074 (2020)