SPATIO-TEMPORAL GRAPH COMPLEMENTARY SCATTERING NETWORKS

Zida Cheng, Siheng Chen✉, Ya Zhang✉

Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

Spatio-temporal graph signal analysis has a significant impact on a wide range of applications, including hand/body pose action recognition. To achieve effective analysis, spatio-temporal graph convolutional networks (ST-GCN) leverage the powerful learning ability to achieve great empirical successes; however, those methods need a huge amount of high-quality training data and lack theoretical interpretation. To address this issue, the spatio-temporal graph scattering transform (ST-GST) was proposed to put forth a theoretically interpretable framework; however, the empirical performance of this approach is constrained by the fully mathematical design. To benefit from both sides, this work proposes a novel complementary mechanism to organically combine the spatio-temporal graph scattering transform and neural networks, resulting in the proposed *spatio-temporal graph complementary scattering networks* (ST-GCSN). The essence is to leverage the mathematically designed graph wavelets with pruning techniques to cover major information and use trainable networks to capture complementary information. The empirical experiments on hand pose action recognition show that the proposed ST-GCSN outperforms both ST-GCN and ST-GST.

Index Terms— Spatio-temporal graph, graph scattering network, complementary, action recognition

1. INTRODUCTION

Spatio-temporal graph is an effective tool for modeling dynamic non-Euclidean data. Spatio-temporal graph signal analysis is widely useful in many applications, e.g., hand/body pose action recognition [1, 2, 3, 4, 5] and prediction [6, 7, 8], multi-agent trajectory prediction [9, 10] as well as traffic flow forecasting [11, 12]. Figure 1 shows a spatio-temporal graph of the hand pose.

To analyze spatio-temporal graph signals, spatio-temporal graph convolutional networks (ST-GCN) is emerging as a powerful learning-based method and has achieved great empirical success [1, 2]. However, the design of network architecture lacks theoretical interpretation. It is thus hard to explain the design rationale and further improve the architectures. Furthermore, ST-GCN also needs a large amount of high-quality labeled data, which might not be available
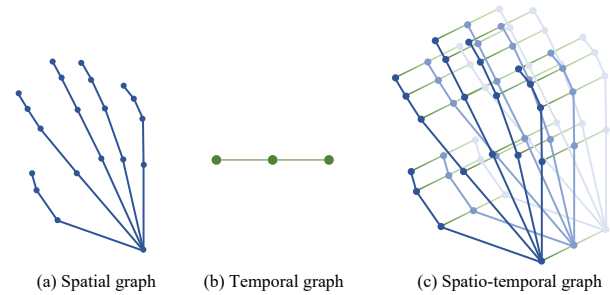

(a) Spatial graph   (b) Temporal graph   (c) Spatio-temporal graph

Fig. 1. A spatio-temporal graph representing hand pose. The vertices in temporal graph are sequentially connected.

in many practical scenarios. In contrast to ST-GCN, spatio-temporal graph scattering transform (ST-GST) was proposed to provide a mathematically interpretable framework [13]. It iteratively applies mathematically designed spatio-temporal graph wavelets and nonlinear activation functions on the graph signal. Since the filter banks do not need any training, ST-GST still works well with limited data. ST-GST also assures some nice theoretical properties, e.g., stability to perturbation of graph signals. However, the empirical performance of ST-GST is constrained by the nontrainable framework, especially when sufficient data are accessible.

In this work, we propose novel *spatio-temporal graph complementary scattering networks* (ST-GCSN), which organically combines the main architecture of ST-GST and the learning ability of ST-GCN. Our design rationale is to leverage the mathematically designed graph wavelets with pruning techniques to cover major information and use trainable graph convolutions to capture complementary information. The core of ST-GCSN is new *graph complementary scattering layer*. In each original graph scattering layer of ST-GST, we use the pruning techniques [14] to preserve major branches and prune non-informative ones; we next replace the pruned branches with trainable graph convolutions. Those trainable branches would adaptively capture complementary information missed by the preserved major branches. Therefore, the proposed graph complementary scattering layer has more learning ability than the original graph scattering layer; meanwhile, the proposed graph complementary scattering layer follows mathematically designed graph wavelets [15, 16] and has a more clear design rationale than purely trainable

graph convolution layer. We conduct experiments for hand pose action recognition task on FPHA [17] dataset and the results show that our method improves recognition accuracy significantly compared with ST-GCN and ST-GST.

Our main contributions are as follows:

• We propose a novel spatio-temporal graph complementary scattering network. It organically combines mathematically designed ST-GST and trainable ST-GCN, achieving both theoretical interpretability and learning ability.

• We propose a novel graph complementary scattering layer (GCSL) as the basic block, which leverages mathematically designed graph wavelets with pruning techniques to cover major information and uses trainable graph convolutional layer to capture complementary information.

• Extensive experiments are conducted on FPHA dataset for hand pose action recognition task. The proposed method improve classification accuracy by 2.43% / 1.56% compared with ST-GCN/ST-GST. Ablation studies show GCSLs function as complementary component as expected.

2. PRELIMINARY

2.1. Spatio-temporal graph scattering transform

ST-GST [13] iteratively applys spatio-temporal graph wavelets and nonlinear activation, acquiring a tree-like structure. Each node in the tree is a spatio-temporal graph signal.

With the hand pose in Figure 1 as an example, the hand has N joints and the action sequence has T time steps. We see it as a spatio-temporal graph with N spatial vertices and T temporal vertices. The input graph signal is $\mathbf{X} \in \mathbb{R}^{N \times T}$, a general spatio-temporal graph filter is defined as

$$\mathbf{H}\left(\mathbf{P}_s\right) \mathbf{X} \mathbf{G}^{\top}\left(\mathbf{P}_t\right) = \left(\sum_{p=0}^{P-1} h_p \mathbf{P}_s^p\right) \mathbf{X} \left(\sum_{q=0}^{Q-1} g_q \mathbf{P}_t^q\right)^{\top},$$

where $\mathbf{P}_s$ and $\mathbf{P}_t$ are the spatial and temporal graph shift matrices, respectively. The spatial and temporal graph filters, $\mathbf{H}\left(\mathbf{P}_s\right)$ and $\mathbf{G}^{\top}\left(\mathbf{P}_t\right)$ are polynomial of graph shift matrices. Then we can design spatial and temporal graph wavelets. Here we use the lazy random walk matrix as the graph shift: $\mathbf{P}_s = \frac{1}{2}(\mathbf{I} + \mathbf{D}_s^{-1}\mathbf{A}_s)$, where $\mathbf{A}_s$ is the adjacent matrix in spatial domain and $\mathbf{D}_s$ is the diagonal degree matrix; a similar rule is used for $\mathbf{P}_t$ as the time-domain shift. The $\mathbf{P}$ matrices are Markov matrices and row sums are all one, which can represent a signal diffusion process over a graph. The commonly used graph wavelets in ST-GST are defined as

$$\{\mathbf{H}_{j_1}\left(\mathbf{P}_s\right) = \mathbf{P}_s^{2^{j_1-1}} - \mathbf{P}_s^{2^{j_1}}\}_{j_1=1}^{J_s},$$
$$\{\mathbf{G}_{j_2}\left(\mathbf{P}_t\right) = \mathbf{P}_t^{2^{j_2-1}} - \mathbf{P}_t^{2^{j_2}}\}_{j_2=1}^{J_t},$$

where j_1 / j_2 are the spatial/temporal scale. J_s and J_t are space/time scale numbers.

Let $\mathbf{Z} \in \mathbb{R}^{N \times T}$ be a spatio-temporal graph signal input. Each spatio-temporal filter is composed of a spatial filter chosen from $\{\mathbf{H}_{j_1}\}_{j_1=1}^{J_s}$ and a temporal filter from $\{\mathbf{G}_{j_2}\}_{j_2=1}^{J_t}$. Thus we can define $J = J_s \times J_t$ filters and get J output signals by convolving them with $\mathbf{Z}$. Finally, each output signal is $\mathbf{Z}_{(j_1,j_2)} = \sigma\left(\mathbf{H}_{j_1}\left(\mathbf{P}_s\right)\mathbf{Z}\mathbf{G}_{j_2}^{\top}\left(\mathbf{P}_t\right)\right)$. σ is the absolute value function for nonlinear activation. With spatio-temporal graph signals seen as tree nodes, a parent tree node $\mathbf{Z}$ generates J children tree nodes in one scattering layer.

The full ST-GST framework is shown in Figure 2(a). $\mathbf{Z}_0 = \mathbf{X}$ is the root node. We iteratively apply scattering layers and the tree grow exponentially with base of J. Each tree node can be indexed by the path from root to it. For example, $p^{(\ell)} = \left(j_1^{(1)}, j_2^{(1)}\right), \ldots, \left(j_1^{(\ell)}, j_2^{(\ell)}\right)$ is the path from root to a node in the ℓ-th layer, denoted as $\mathbf{Z}_{(p^{(\ell)})}$. Finally, the graph signals across all the tree nodes are concatenated to form the final scattering feature, which is used for downstream tasks. ST-GST has nice theoretical properties, such as the stability to the graph signal and structural perturbations.

2.2. Pruned graph scattering transform

The complexity of ST-GST grows exponentially with layer number. To reduce complexity, [14] prunes the scattering tree according to the ratio of a child node energy to the parent node energy. We can easily extend it to the spatio-temporal graph domain. For a parent node $\mathbf{Z}_p$ and a child node $\mathbf{Z}_c$, $\mathbf{Z}_c$ (and its children) are pruned if $\frac{||\mathbf{Z}_c||}{||\mathbf{Z}_p||} < \tau$, where $||\cdot||$ means Frobenius norm and τ is a user-specific threshold. Figure 2(a) shows an example of pruned scattering tree. [14] shows that this pruning design can still preserve the scattering's stability property.

3. SPATIO-TEMPORAL GRAPH COMPLEMENTARY SCATTERING NETWORK

We first introduce a trainable graph complementary scattering layer as the basic block of ST-GCSN. We then present agent-parameter training, which enables the training of ST-GCSN.

3.1. Graph complementary scattering layer

Scattering layer with pruning. The idea is to use a pruned spatio-temporal graph scattering layer to be the initial structure of a graph complementary scattering layer. For an input spatio-temporal graph signal $\mathbf{Z}_{(p^{(\ell)})}$, an ordinary spatio-temporal graph scattering layer outputs:

$$\mathbf{Z}_{\left(p^{(\ell)},(j_1,j_2)\right)} = \sigma\left(\mathbf{H}_{j_1}\left(\mathbf{P}_s\right)\mathbf{Z}_{(p^{(\ell)})}\mathbf{G}_{j_2}^{\top}\left(\mathbf{P}_t\right)\right),$$
$$j_1 = 1, ..., J_s \quad j_2 = 1, ...J_t,$$

That means each input parent tree node generates $J = J_s \times J_t$ children tree nodes. We then perform the pruning technique

(a) Pruned ST-GST

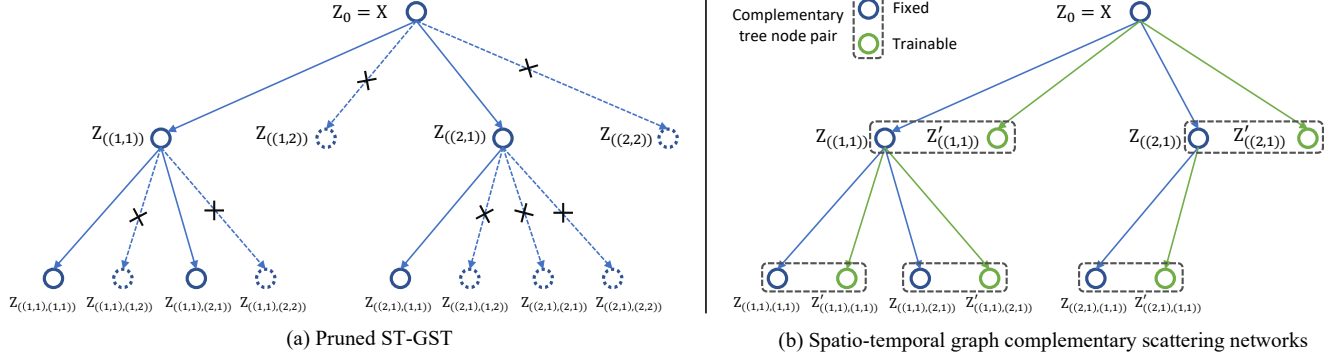(b) Spatio-temporal graph complementary scattering networks

Fig. 2. (a) Pruned scattering tree of ST-GST with 2 layer, $J_s = J_t = 2$. Dotted tree nodes are pruned and the children tree nodes of the pruned ones are omitted. (b) Architecture of the proposed spatio-temporal graph complementary scattering network. Each fixed tree node (in blue) in the pruned scattering tree is complemented by a trainable tree node (in green).

in Section 2.2 with threshold τ, the preserved tree node set is:

$$\mathcal{Z}_f = \{\mathbf{Z}_{(p^{(\ell)},(j_1,j_2))} \mid \frac{||\mathbf{Z}_{(p^{(\ell)},(j_1,j_2))}||}{||\mathbf{Z}_{(p^{(\ell)})}||} > \tau,$$

$$j_1 = 1, ..., J_s, \, j_2 = 1, ...J_t\}.$$

This forms the fixed component in a graph complementary scattering layer. The process is shown in Figure 2(a).

Complementary tree nodes. The idea is to construct trainable tree nodes to replace the pruned tree nodes, adaptively complementing to the preserved tree node set in each graph complementary scattering layer; see Figure 2(b). For each preserved tree node $\mathbf{Z}_{(p^{(\ell)},(j_1,j_2))} \in \mathcal{Z}_f$, we construct its complementary tree node $\mathbf{Z}'_{(p^{(\ell)},(j_1,j_2))}$. These two tree nodes form a complementary pair:

$$\mathbf{Z}_{(p^{(\ell)},(j_1,j_2))} = \sigma \left(\mathbf{H}_{j_1} \left(\mathbf{P}_s \right) \mathbf{Z}_{(p^{(\ell)})} \mathbf{G}_{j_2}^{\top} \left(\mathbf{P}_t \right) \right),$$

$$\mathbf{Z}'_{(p^{(\ell)},(j_1,j_2))} = \sigma \left((\mathbf{I} - \mathbf{H}_{j_1} \left(\mathbf{P}'_s \right)) \mathbf{Z}_{(p^{(\ell)})} \left(\mathbf{I} - \mathbf{G}_{j_2}^{\top} \left(\mathbf{P}'_t \right) \right) \right),$$

$$(1)$$

where $\mathbf{P}'_s$ and $\mathbf{P}'_t$ are trainable through backpropagation and initialized as $\mathbf{P}_s$ and $\mathbf{P}_t$, respectively. We thus obtain the trainable tree node set, whose cardinality is the same with the fixed tree node set. This design involves two key-points before training: (1) the filters of fixed and trainable tree nodes are explicitly complementary. The filter $\mathbf{H}$ in a fixed tree node corresponds to the complementary filter $\mathbf{I} - \mathbf{H}$ in the associated trainable tree node; and (2) all the trainable tree nodes with the same parent share the same $\mathbf{P}'_s$ and $\mathbf{P}'_t$, following designing rule of graph wavelets. That gives a more clear design rationale than purely trainable graph convolutional layer.

Overall layer. Finally, a graph complementary scattering layer is formed by both fixed tree nodes and trainable tree nodes. By iteratively applying graph complementary scattering, we obtain a spatio-temporal graph complementary scattering network, which organically combines the advantages of

both ST-GST and ST-GCN; see Figure 2(b). For downstream tasks, all tree nodes are concatenated as the final feature. The trainable parameters are optimized by stochastic gradient descent.

3.2. Agent-parameter training

To ensure that the trainable tree nodes in graph complementary scattering layers still retain nice theoretical properties as the fixed nodes, we propose the agent-parameter training mechanism. Recall that we make $\mathbf{P}'_s$ and $\mathbf{P}'_t$ trainable, which are Markov matrices with rows normalized. To preserve this property, for a Markov matrix $\mathbf{P}'$, we do not train it directly but use an agent parameter matrix $\mathbf{M}$ and let: $\mathbf{P}' = \mathcal{E}(\mathbf{M})$, where $\mathcal{E}(\cdot)$ means row-wise softmax normalization. We then train $\mathbf{M}$ as the network parameter. At the start of training, we initialize $\mathbf{M}$ so that the resultant random walk matrix is equal to the corresponding matrix in the fixed tree nodes: $\mathbf{P}'_{init} = \mathcal{E}(\mathbf{M}_{init}) = \mathbf{P}$.

To summarize, we build ST-GCSN by graph complementary scattering layers, which are composed of fixed tree nodes and trainable tree nodes. We use spatio-temporal graph scattering layer and pruning technique to get fixed nodes. Then we construct trainable tree nodes to capture complementary information that fixed tree nodes missed. The adaptive nodes are designed in an explicitly complementary manner and also follow the graph wavelets design rule.

4. EXPERIMENTAL RESULTS

4.1. Experimental setup

Dataset. First-Person Hand Action Recognition (FPHA) [17] contains 1175 videos from 45 different action categories. Each video contains a right hand manipulating an object. The dataset provide 3D coordinate annotation of 21 hand joints

Method	Accuracy (%)	Training time
Transition Forest [18]	80.69	——
Huang et al. [19]	84.35	——
ST-GCN [2]	86.32	10 h
ST-GST [13]	87.19	1.3 h
ST-GCSN	**88.75**	2.5 h

Table 1. Hand action recognition accuracy on FPHA. Our method outperms both ST-GST and ST-GCN.

for every frame. We clip and pad the skeleton coordinate sequences so that each sequence contains 200 frames. We then uniformly sample 67 frames from each sequence. We use 600 sequences for training and 575 for testing. The original data consist of 3 coordinate dimensions (x, y, z), we use them as three independent channels in the root graph signal node.

Implementation details. We use 2 scattering layers with $J_s = 20, J_t = 5$, resulting in 10101 nodes before pruning. We set pruning threshold $\tau = 0.002$, preserving 2693 fixed tree nodes after pruning. We pool the final feature by temporal average to reduce cardinality. The classifier is implemented by a multi-layer perceptron with one hidden layer.

4.2. Results

Primary results Table 1 reports the action recognition accuracy on FPHA. The proposed ST-GCSN outperforms ST-GST, ST-GCN and other strong baselines. Our method shows superiority over purely trained graph convolutions and fully designed methods. We also conduct experiments for human body action recognition on a subset of NTU-RGBD[20]. Our method achieves 50.05% accuracy, outperforming ST-GST (48.24%) and ST-GCN (45.36%).

Effect of complementary mechanism The complementary mechanism is the core of ST-GCSN. It involves two aspects: (1) the fixed tree nodes and trainable tree nodes capture complementary information; (2) the filters in fixed tree nodes and trainable tree nodes are in the complementary form $(\mathbf{H}, \mathbf{I} - \mathbf{H})$, see Eq.1. In Table 2, We conduct ablation experiments for both aspects: (1) We construct two networks using only fixed/trainable tree nodes. Results show that best accuracy is achieved only when fixed and trainable nodes are both utilized. That validates that the fixed and trainable component of ST-GCSN work in a complementary manner. (2) We replace the $(\mathbf{H}, \mathbf{I} - \mathbf{H})$ filter design with $(\mathbf{H}, \mathbf{H})$, that is, $\mathbf{Z}'_{p^{(\ell)},(j_1,j_2)} = \sigma\left((\mathbf{H}_{j_1}(\mathbf{P}'_s)) \mathbf{Z}_{(p^{(\ell)})} \mathbf{G}^\top_{j_2}(\mathbf{P}'_t)\right)$; called *w/o complementary mechanism*. Accuracy drops significantly without the explicit complementary filters, validating the necessity of design in Eq. 1.

Effect of number of tree nodes. Compared with ST-GST, the proposed ST-GCSN have both fixed and trainable tree nodes, leading to twice of the tree node cardinality. To analyze the effect of the tree node number, we vary it by applying different pruning thresholds on ST-GST. Figure 3

Method	Acc. (%)
Fixed tree nodes only	87.19
Trainable tree nodes only	87.82
w/o complementary mechanism	86.26
complementary mechanism	**88.75**

Table 2. Effect of complementary mechanism. Both fixed and trainable tree nodes with the complementary design significantly improves the performance.
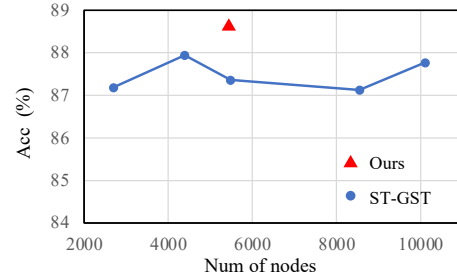

Fig. 3. Effect of tree node number. The proposed ST-GCSN outperforms ST-GST with varying number of tree nodes.

Method	Acc. (%)	Params.
Purely trainable	88.99	13.27M
ST-GCSN	**88.75**	0.23M

Table 3. Effect of trainability. Proposed ST-GCSN achieves similar performance with 50 times less trainable parameters.

shows that simply increasing the node number brings little gain. That indicates the improvement of our method is mainly credited to the proposed complementary framework, instead of preserving more tree nodes in ST-GST.

Effect of trainability The proposed graph complementary scattering layer follows the design of graph wavelets, thus acquire more clear design rationale than purely trainable graph convolution. To analyze the effect of trainability, we replace the trainable tree nodes with purely trainable graph convolutional filters. Table 3 shows that the purely trainable network reaches negligible gain by sacrificing 50 times more trainable parameters.

5. CONCLUSION

We propose the spatio-temporal graph complementary scattering networks (ST-GCSN) for spatio-temporal graph data analysis. As the basic block of ST-GCSN, each graph complementary scattering layer consists of both mathematically designed tree nodes and trainable tree nodes. With the proposed complementary mechanism, graph complementary scattering layers also have both theoretical interpretability and flexible learning ability. Experimental results validate that the proposed method outperforms both ST-GCN and ST-GST on the task of hand pose action recognition.

6. REFERENCES

[1] Sijie Yan, Yuanjun Xiong, and Dahua Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, pp. 7444–7452.

[2] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Negar Heidari and Alexandras Iosifidis, "Progressive spatio-temporal graph convolutional network for skeleton-based human action recognition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3220–3224.

[5] Pratyusha Das, Jiun-Yu Kao, Antonio Ortega, Tomoya Sawada, Hassan Mansour, Anthony Vetro, and Akira Minezawa, "Hand graph representations for unsupervised segmentation of complex activities," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 4075–4079.

[6] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian, "Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[7] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian, "Multiscale spatio-temporal graph neural networks for 3d skeleton-based motion prediction," *IEEE Transactions on Image Processing*, vol. 30, pp. 7760–7775, 2021.

[8] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li, "Learning trajectory dependencies for human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[9] Yue Hu, Siheng Chen, Ya Zhang, and Xiao Gu, "Collaborative motion prediction via neural motion message passing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, Eds., Cham, 2020, pp. 507–523, Springer International Publishing.

[11] Xiyue Zhang, Chao Huang, Yong Xu, Lianghao Xia, Peng Dai, Liefeng Bo, Junbo Zhang, and Yu Zheng, "Traffic flow forecasting with spatial-temporal graph diffusion network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, pp. 15008–15015, May 2021.

[12] Yuyol Shin and Yoonjin Yoon, "Incorporating dynamicity of transportation network with multi-weight traffic graph convolutional network for traffic forecasting," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[13] Chao Pan, Siheng Chen, and Antonio Ortega, "Spatio-temporal graph scattering transform," in *International Conference on Learning Representations*, 2021.

[14] Vassilis N. Ioannidis, Siheng Chen, and Georgios B. Giannakis, "Pruned graph scattering transforms," in *International Conference on Learning Representations*, 2020.

[15] Feng Gao, Guy Wolf, and Matthew Hirn, "Geometric scattering for graph data analysis," in *Proceedings of the 36th International Conference on Machine Learning*, 09–15 Jun 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 2122–2131.

[16] Fernando Gama, Alejandro Ribeiro, and Joan Bruna, "Diffusion scattering transforms on graphs," in *International Conference on Learning Representations*, 2019.

[17] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] Guillermo Garcia-Hernando and Tae-Kyun Kim, "Transition forests: Learning discriminative temporal transitions for action recognition and detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[19] Zhiwu Huang and Luc Van Gool, "A riemannian network for spd matrix learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[20] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.