1. The Effect of Regularization Factor λ and Activation function

This section shows the effect of the regularization factor of the proposed Covariance Loss on the Metric. In addition, we present the effect of activation function on basis for the overall performance. As an example, we employ GWNET and Metr-LA dataset and compare performance of GWNET and GWNET-Cov by varying the λ of Covariance Loss. Figures 1 $\sim$ 3 show the effect of the factor, λ of Covariance Loss on the overall performance. As shown in the figures, in case of RMSE for middle- and long- term predictions, GWNET-Cov with higher λ reports more improved performances, while in the other metrics, GWNET-Cov shows similar or worse performances than GWNET. Figure 4 shows the effect of activation functions on basis. As shows in the figure, activation functions have limited contributions on the overall performances but *relu* shows slightly better performances than the others.

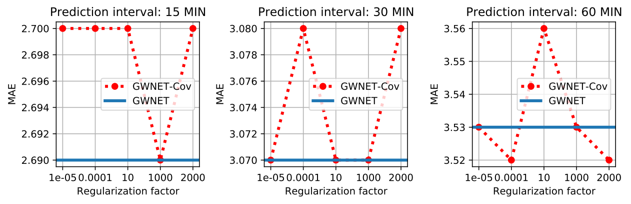

Figure 1. The effect of λ on MAE

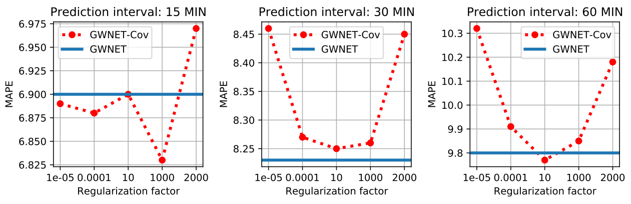

Figure 2. The effect of λ on MAPE

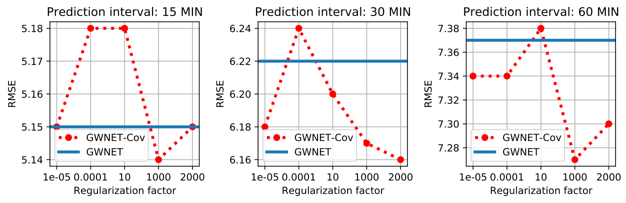

Figure 3. The effect of λ on RMSE

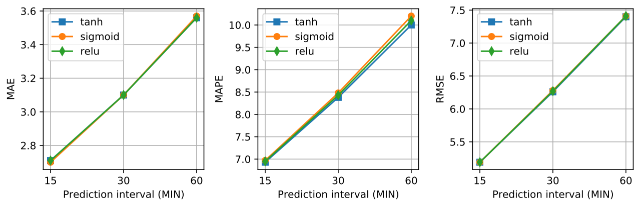

Figure 4. The effect of Activation

In Table 1$\sim$3, we present performance evaluations of GWNET on METR-LA dataset by varying the length of basis, λ, and activation functions. We observe that for the dataset, Covariance Loss improves middle- and long- term predictions and relu is the best activation for Covariance Loss.

Table 1. The effect of λ with gwnet-Cov with 64 basis on METR-LA dataset

Activation	λ	MAE (15/30/60 min)	MAPE (15/30/60 min)	RMSE (15/30/60 min)	λ	MAE (15/30/60 min)	MAPE (15/30/60 min)	RMSE (15/30/60 min)
ReLU	1000.0	3.45/4.10/5.20	9.80/12.02/16.33	6.41/7.85/9.67	100.0	2.96/3.43/4.21	7.93/9.64/12.40	5.61/6.77/8.37
	10.0	2.72/3.11/3.57	7.00/8.50/10.20	5.18/6.23/**7.32**	1.0	2.72/3.08/**3.53**	7.01/8.43/9.99	5.22/**6.20/7.27**
	0.1	**2.69**/3.08/3.55	**6.90**/8.36/10.03	5.16/**6.22/7.34**	0.001	2.71/3.08/**3.53**	**6.90**/8.38/10.05	5.17/**6.17/7.33**
	0.0001	**2.69**/3.09/3.59	**6.90**/8.25/9.89	5.18/6.23/7.43	0.0002	2.71/3.09/**3.52**	7.10/8.50/10.10	5.17/**6.22/7.28**
	0.0003	2.70/3.09/**3.50**	**6.90**/8.40/10.10	5.19/6.24/**7.27**	0.0004	**2.69/3.07/3.53**	**6.90**/8.30/9.90	5.17/**6.21/7.36**
	0.0005	2.70/3.09/3.56	**6.90/8.19/9.80**	**5.15/6.18/7.36**	0.0006	**2.69/3.07**/3.54	7.00/8.34/9.90	5.16/**6.19/7.30**
	0.0007	2.70/3.10/3.55	**6.90**/8.32/9.99	5.16/**6.19/7.30**	0.0008	**2.69/3.07**/3.54	**6.86**/8.32/10.11	5.16/**6.22**/7.39
	0.0009	**2.69/3.07/3.52**	6.98/8.42/10.12	5.17/**6.20/7.31**	0.00001	2.70/3.08/**3.52**	**6.89/8.22**/9.83	**5.12/6.18/7.28**
Tanh	1000.0	2.70/3.09/3.55	6.92/8.41/10.05	5.15/6.19/7.38	10.0	2.71/3.11/3.57	6.95/8.41/10.19	5.20/6.27/7.43
	0.0001	2.70/3.07/3.53	6.86/8.31/10.01	5.16/6.18/7.32	0.00001	2.72/3.08/**3.53**	7.01/8.43/9.99	5.22/**6.20/7.27**
Sigmoid	1000.0	3.45/4.10/5.20	9.80/12.02/16.33	6.41/7.85/9.67	10.0	2.96/3.43/4.21	7.93/9.64/12.40	5.61/6.77/8.37
	0.0001	2.72/3.11/3.57	7.00/8.50/10.20	5.18/6.23/**7.32**	0.00001	2.72/3.08/**3.53**	7.01/8.43/9.99	5.22/**6.20/7.27**
gwnet (base line)		**2.69/3.07**/3.53	6.90/8.23/**9.80**	5.15/6.22/7.37	-	-	-	-

Table 2. The effect of λ with GWNET-Cov with 128 basis on METR-LA dataset

Activation	λ	MAE (15/30/60 min)	MAPE (15/30/60 min)	RMSE (15/30/60 min)	λ	MAE (15/30/60 min)	MAPE (15/30/60 min)	RMSE (15/30/60 min)
ReLU	1000.0	2.70/3.08/**3.53**	6.99/8.46/10.11	5.18/6.23/**7.32**	10.0	2.70/3.10/3.57	7.01/8.50/10.18	5.17/6.25/7.41
	0.0001	-	-	-	0.00001	2.71/3.11/3.54	7.04/8.58/10.23	5.20/6.32/7.38
Tanh	1000.0	2.71/3.10/3.57	7.01/8.32/9.83	5.16/**6.21**/7.37	10.0	2.70/3.09/3.54	6.90/8.45/10.09	**5.15/6.21**/7.33
	0.0001	2.72/3.10/3.56	6.92/8.34/9.89	5.26/6.30/7.43	0.00001	2.72/3.09/**3.53**	**6.86**/8.30/9.90	5.20/**6.21/7.32**
Sigmoid	1000.0	2.70/3.08/**3.53**	6.99/8.46/10.11	5.18/6.23/**7.32**	10.0	2.70/3.10/3.57	7.01/8.50/10.18	5.17/6.25/7.41
	0.0001	-	-	-	0.00001	2.71/3.11/3.54	7.04/8.58/10.23	5.20/6.32/7.38
GWNET (base line)		**2.69/3.07/3.53**	6.90/**8.23/9.80**	**5.15**/6.22/7.37	-	-	-	-

Table 3. The effect of λ with gwnet-Cov with 512 basis on METR-LA dataset

Activation	λ	MAE (15/30/60 min)	MAPE (15/30/60 min)	RMSE (15/30/60 min)	λ	MAE (15/30/60 min)	MAPE (15/30/60 min)	RMSE (15/30/60 min)
ReLU	2000.0	2.7/3.08/**3.52**	6.97/8.45/10.18	**5.15/6.16/7.30**	1000.0	**2.69/3.07/3.53**	**6.83**/8.26/9.85	**5.14/6.17/7.27**
	10	2.7/**3.07**/3.56	**6.9**/8.25/**9.77**	5.18/**6.20**/7.38	0.0001	2.7/3.08/**3.52**	6.88/8.27/9.91	5.18/6.24/**7.34**
	0.00001	2.7/**3.07**/3.53	**6.89**/8.46/10.32	**5.15/6.18/7.34**	-	-	-	-
Tanh	1000.0	2.70/3.08/3.55	6.92/8.44/10.04	5.17/6.23/**7.35**	10.0	2.71/3.10/3.58	7.01/8.38/10.08	5.19/6.28/7.45
	0.0001	-	-	-	0.00001	2.71/3.11/3.57	6.98/8.62/10.36	5.20/6.32/7.44
Sigmoid	1000.0	-	-	-	10.0	2.71/3.10/3.57	**6.84**/8.29/10.02	5.19/6.26/7.39
	0.0001	2.70/3.10/3.57	6.97/8.41/9.93	5.18/6.27/7.44	0.00001	2.71/3.09/3.54	6.99/8.43/10.14	5.21/6.25/**7.37**
GWNET (base line)		**2.69/3.07**/3.53	6.90/**8.23**/9.80	5.15/6.22/7.37	-	-	-	-

2. Case Study for Noise Robustness

Figure 5 depicts RMSE degeneration of predictions on node 7, 78, 114 and 156 as a function of the number of noisy node. When the number of noisy nodes is small, the degeneration of STGCN-Cov is negligible due to explicit learning of spatial dependecies, but the number of noisy node becomes large, the shape of the kernel distribution changes and predicted value also changes more sensitively (Node 7, 78). And there are nodes that are affected by the spatial change of the kernel more slower (Node 114) or faster (Node 156).

2.1. Comparisons with L1 and L2 Regularization

Since the proposed loss function has a form of regularization, we evaluate the performance by comparing it against L1 and L2 regularization methods. For this set of experiment, we employ STGCN with 3rd order of chebyshev expansion as a baseline model and apply L1, L2 and the proposed loss function varying their importance factor, λ. We run each method 10 times and measure RMSE. Figure 6 show RMSE of STGCN (baseline), STGCN-Cov (cov-λ), STGCN-L1 (L1-λ), STGCN-L2 (L2-λ) on PeMSD7(M) dataset. As shown in the figures, the proposed loss outperforms all the others regularization methods.

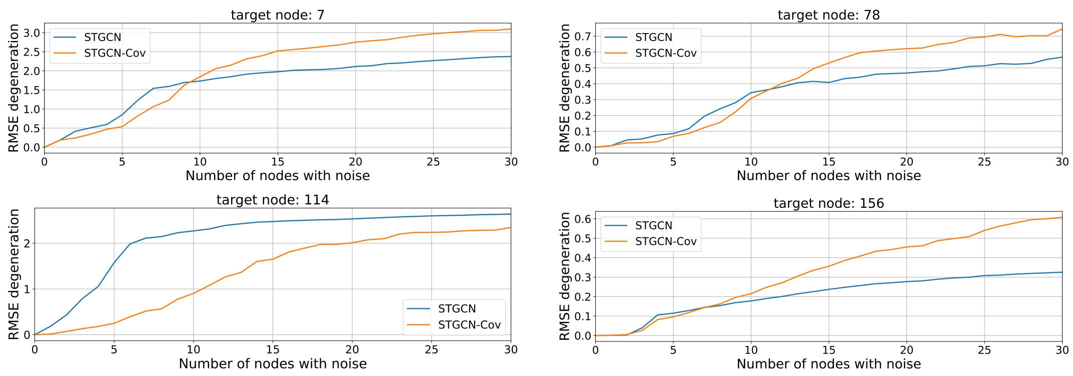

Figure 5. Noise Robustness as a function of the number of noisy nodes on PeMSD7(M) dataset

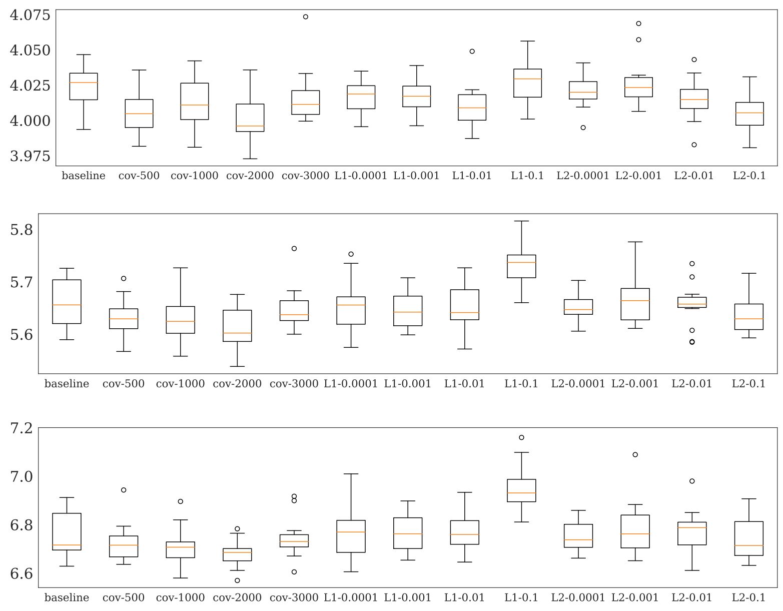

Figure 6. RMSE comparison of STGCN with L1, L2 and Covariance loss for prediction step 3, 6, and 9 (in order from top to bottom).

3. Case Study for Contributions of Cross Terms

With the proposed loss function, we impose an addition constraints on the cross terms to have zero effect for the predictions. In Figure 7, we plot the overall contributions of cross terms for predicting velocities of node 11, 78, 156 and 177 of PeMSD7(M) dataset. As we expected, overall contributions of cross-terms consistently become near to zero except for the cases where velocities change rapidly and significantly. Figure 8 also shows the effect of the importance factor (λ) of covariance loss on the distribution of basis values (left) and RMSE degeneration as a function of the number of noisy nodes for varying λ (right). We find that with the bigger λ, basis values approach to zero which further indicates that contribution of basis cross-terms becomes smaller. We also find that the bigger λ reduces the effect of noisy nodes. However, even though the bigger λ may bring noisy robustness, the overall performance is worse than that of the baseline model ($\lambda = 0$) in this case.

4. The Basis functions for Independent Predictions

In Figure 9, we plot predictions results and corresponding basis functions for node 1, 10, and 100 of METR-LA dataset. As shown in the figures, the basis functions follow the shape of predictions but they are not as clear as the former experiments that follow only predictions for the next 5 minutes in STGCN-Cov. This is because all of the predictions share the basis functions. The basis functions consider predictions of the next $5 \sim 60$ minutes, the shape of the basis functions are distracted.

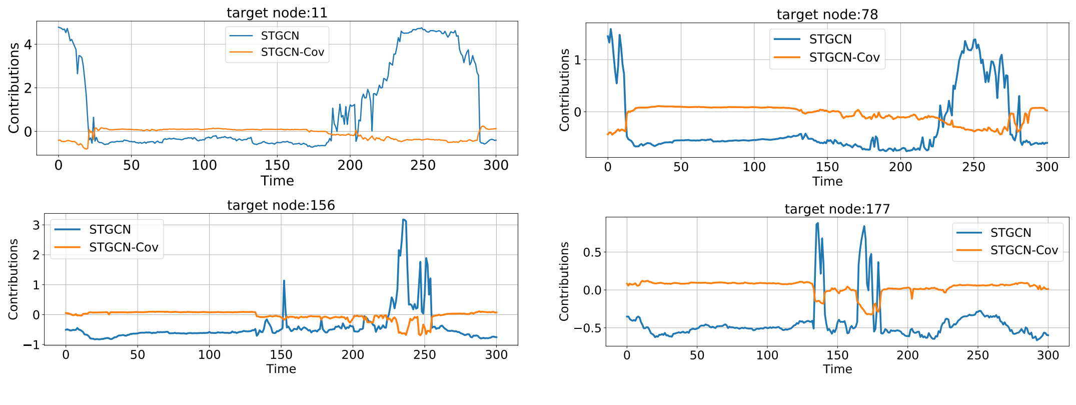

Figure 7. Predictions on velocities over the next $5 \sim 60$ minutes.

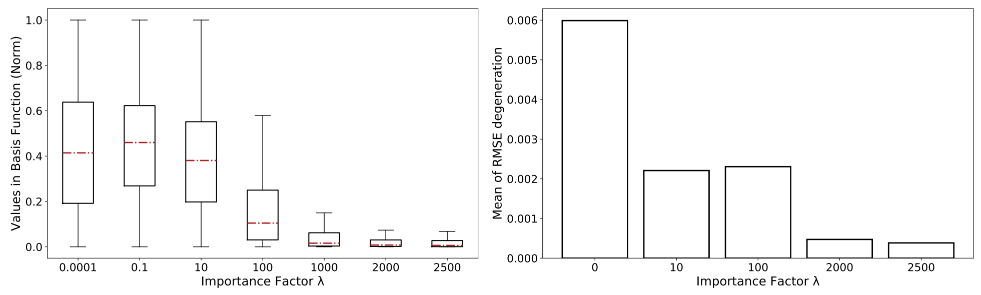

Figure 8. The effect of λ on cross-term contribution (left) and noisy robustness (right).

5. Spatio-Temporal Graph Convolutional Network (STGCN)

STGCN is a variant of Graph Convolutional Network (GCN). that generalizes traditional convolution operation onto graph structured data by defining a graph convolution operator. The traditional convolution operator considers spatial locality specified by filters to extract meaningful features on grid-shaped data structures. The graph convolution operates similarly and spatial locality is specified by a *graph laplacian* matrix. The graph laplacian matrix specifies neighbor relations between nodes in a graph and how strong the relations are. The graph laplacian matrix, (L) is defined as follows.

$$(Lf)(i) = \sum_{j \in N_i} W_{i,j}(f(i) - f(j)), \tag{1}$$

where, N_i and $W_{i,j}$ indicate neighbors of node i and weights between node i and node j. As shown in the Equation 1, with graph laplacian matrix, it is able to infer node i's signal, $f(i)$ by aggregating the neighbor nodes' signals, $f(j)$, which is analogous to the traditional convolution operation that consider spatial locality to extract meaningful features.

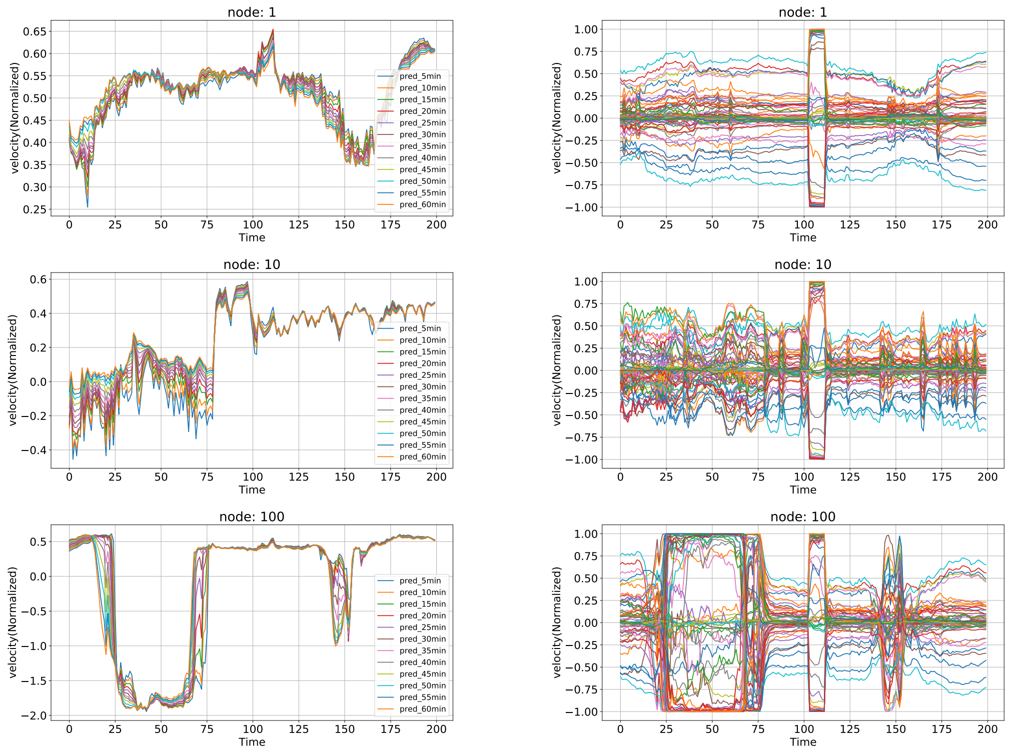

Figure 9. Velocity predictions of METR-LA dataset over next 60 minutes (Left) and its basis functions (Right) on node 1, 10, and 100 (in order from top to bottom).

Another interesting nature of graph laplacian matrix is that the matrix is eigen-decomposable so that we have $L = U\Lambda U^T$ if the matrix is symmetric, i.e. the graph is undirected. Thus, a feature extraction, $h = f_\theta(L)x$, becomes filtering at spectrum domain (a whole set of eigen-vectors), $h = Uf_\theta(\Lambda)U^T x$. Notice that aforementioned graph convolutions are not fully utilizing locality (only 1-hop neighbor nodes are included in the estimation) or computationally expensive (additional matrix multiplications for U and U^T are required). One well-known solution for the limitations is to employ polynomial approximations such as Chebyshev polynomials as follows,

$$f_\theta(L)\mathbf{x} \approx \sum_{k=0}^{K} \theta_k T^k(\bar{L})\mathbf{x}, \tag{2}$$

where, $\bar{L}$ indicates $2L/\lambda_{max} - I_N$. With k-order Chebyshev expansion, we can consider signals of k-hop neighbors for feature extractions.

GCN introduces a first order approximation on chebyshev expansion (Equation 2). Assuming $K = 1$ and $\lambda_{max} = 2$, Eq.2 is simplified as

$$f_\theta(L)\mathbf{x} = \theta_0\mathbf{x} + \theta_1 D^{-1/2}AD^{-1/2}\mathbf{x}, \tag{3}$$

where A is an adjacent matrix and D is a diagonal matrix of node degree. With the definition on normalized graph laplacian matrix, $L = I_N - D^{-1/2}AD^{-1/2}$ and further assumption that $\theta_0 \approx \theta_1$, the authors of GCN introduces the well-known form of graph convolution operation,

$$H^{l+1} = f(\bar{A}H^l W^l), \tag{4}$$

where $\bar{A} = I_N - D^{-1/2}AD^{-1/2}$, H^l and W^l are input and weight of l^{th} layer while $H^0 = \mathbf{x}$ and f denotes activation function. With the mathematically well established graph convolution operation, GCN has achieved successful performance, while being computationally efficient.

STGCN extends GCN by stacking 1-dimensional convolution layers and graph convolution layers to extract features considering temporal and spatial locality. STGCN consists of 2 ST-conv blocks in which a 1-dimensional convolution layer with gated linear unit and two graph convolution layers are stacked. In the 1-dimensional convolutional layer, filters extract features from each time series variable. After extracting temporal features, STGCN extracts spatial features with graph convolution operation by considering signals of neighboring nodes specified by graph laplacian matrix. The STGCN repeats the spatial and temporal feature extractions and thus, the output of last hidden layer represents condensed spatio-temporal information of given dataset.

Finally, STGCN applies a fully convolutional operation on the basis functions. Here, the the number of features of the basis function is the same with the width of filters. This indicates that the last layer of STGCN considers all the features at the same time, which is equivalent to traditional linear regression models. With the characteristics of STGCN, our Spatio-Temporal Covariance loss function promotes STGCN to learn temporally and spatially condensed information of given data and thus we can fully utilize temporal and spatial covariance for the learning of STGCN.

6. Graph-WaveNet (GWNET)

To capture true spatial dependencies and genuine underlying relationships among relevant nodes which might not be explicitly reflected on a fixed (pre-determined) graph structure, a novel graph neural network architecture is proposed, Graph-WaveNet (GWNET), which designs a self-adaptive matrix to take the variations of the influence between each node and its neighbors into account, which is learned in a data-driven manner. This self-adaptive adjacency matrix does not require any prior knowledge and is learned end-to-end through stochastic gradient descent, which makes the model discover hidden spatial dependencies by itself. The self-adaptive adjacency matrix is represented as

$$\tilde{\mathbf{A}}_{apt} = SoftMax(ReLU(\mathbf{E}_1\mathbf{E}_2)) \tag{5}$$

where $\mathbf{E}_1, \mathbf{E}_2 \in \mathbf{R}^{N\times c}$ are two node embedding dictionaries, randomly initialized learnable parameters.

To capture the spatial-temporal dependencies, the GWNET framework consists of K spatial-temporal layers, each of which contain two building blocks, the graph convolution layer (GCN) and the temporal convolution layer (TCN).

In graph convolution layer (GCN), the graph convolution is an essential operation, which is equivalent to the equation 4. Further, the diffusion convolution layer is utilized for the diffusion process of graph signals with K finite steps (hops). As a result, it is generalized into the form of equation 6,

$$\mathbf{H}^{l+1} = f(\sum_{k=0}^{K} \tilde{\mathbf{A}}_{apt}^{k} \mathbf{H}^{l} \mathbf{W}_{k}^{l}) \tag{6}$$

where the graph structure is not given, and only the self-adaptive adjacency matrix is utilized alone to capture hidden spatial dependencies.

In temporal convolution layer, dilated casual convolution is an essential operation to model the temporal correlations to increase the receptive field exponentially. Given a 1D sequence input $\mathbf{x} \in \mathbf{R}^T$ and filter $\mathbf{f} \in \mathbf{R}^m$, the dilated casual convolution operation of $\mathbf{x}$ with $\mathbf{f}$ at time step t is represented as,

$$\mathbf{x} \star \mathbf{f}(t) = \sum_{s=0}^{m-1} \mathbf{f}(s)\mathbf{x}(t - d \times s) \tag{7}$$

where d is the dilation factor which controls the skipping distance, and $\star$ is the dilated casual convolution operator. The receptive field of a model grows exponentially as the number of stacked dilated causal convolution layers increases, which enables dilated causal convolution networks to capture longer sequences.

With this, a simple Gated TCN is easily fitted into the framework. Given the input X, it takes the form

$$\mathbf{h} = g(\Theta_1 \star X + b) \odot \sigma(\Theta_2 \star X + c) \tag{8}$$

where Θ_1, Θ_2, b, c are model parameters, $g(\cdot)$ is an activation function of the output, $\sigma(\cdot)$ is the sigmoid function and $\odot$ is the element-wise product.

In short, not only the GWNET takes advantage of a self-adaptive graph matrix to capture the hidden spatial dependency in the data, but also utilizes the network combined the graph convolution with the 1D dilated casual convolution to capture the spatial-temporal dependency even from very long sequences of the input, which are integrated seamlessly in a unified framework.

7. Constraint Analysis

In linear regression problems with mean square error (MSE), the optimization goal is to learn the best mapping of $\mathbf{x}$, $\phi(\mathbf{x})$ and $\mathbf{w}$ such that,

$$\mathbf{y} = \Sigma_{i=1}^{D} w_i \phi_i(\mathbf{x}), \tag{9}$$

where D is a dimension of the mapping. Meanwhile, the proposed loss function includes minimizing MSE between covariance matrices of empirical target variables and corresponding basis functions which have a form of inner product such that,

$$\mathbf{y} \cdot \mathbf{y}' = \Sigma_{i=1}^{D} w_i^2 \phi_i(\mathbf{x}) \phi_i(\mathbf{x}'). \tag{10}$$

Combining Equation 9 and Equation 10 reveals the constraint more explicitly.

$$\begin{aligned} \mathbf{y} \cdot \mathbf{y}' &= \Sigma_{i=1}^{D} w_i \phi_i(\mathbf{x}) \cdot \Sigma_{i=1}^{D} w_i \phi_i(\mathbf{x}') \\ &= \Sigma_{i=1}^{D} w_i^2 \phi_i(\mathbf{x}) \phi_i(\mathbf{x}'). \end{aligned} \tag{11}$$

Thus, the summation of all of the cross terms that consist of $\phi_i(x)\phi_j(x')$, where $i \neq j$, should converge to zero as the optimization proceeds. To achieve further insight on the constraint, we assume $D = 3$ and take a break-down analysis.

$$\begin{aligned} & w_1\phi_1(\mathbf{x})(w_2\phi_2(\mathbf{x}') + w_3\phi_3(\mathbf{x}')) \\ +& w_2\phi_2(\mathbf{x})(w_1\phi_1(\mathbf{x}') + w_3\phi_3(\mathbf{x}')) \\ +& w_3\phi_3(\mathbf{x})(w_1\phi_1(\mathbf{x}') + w_2\phi_2(\mathbf{x}')) = 0 \end{aligned} \tag{12}$$

Then, we take a matrix form such that,

$$\begin{bmatrix} w_1\phi_1(\mathbf{x}) & w_2\phi_2(\mathbf{x}) & w_3\phi_3(\mathbf{x}) \end{bmatrix} \begin{bmatrix} 0 & w_2 & w_3 \\ w_1 & 0 & w_3 \\ w_1 & w_2 & 0 \end{bmatrix} \begin{bmatrix} \phi_1(\mathbf{x}') \\ \phi_2(\mathbf{x}') \\ \phi_3(\mathbf{x}') \end{bmatrix} = 0. \tag{13}$$

The matrix multiplication of the first two matrix gives

$$\begin{bmatrix} w_1w_2\phi_2(\mathbf{x}) + w_1w_3\phi_3(\mathbf{x}) \\ w_1w_2\phi_1(\mathbf{x}) + w_2w_3\phi_3(\mathbf{x}) \\ w_1w_3\phi_1(\mathbf{x}) + w_2w_3\phi_2(\mathbf{x}) \end{bmatrix}^T \begin{bmatrix} \phi_1(\mathbf{x}') \\ \phi_2(\mathbf{x}') \\ \phi_3(\mathbf{x}') \end{bmatrix} = 0. \tag{14}$$

Now, we can decompose the transposed matrix such as,

$$\left(\begin{bmatrix} 0 & w_1w_2 & w_1w_3 \\ w_1w_2 & 0 & w_2w_3 \\ w_1w_3 & w_2w_3 & 0 \end{bmatrix} \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \phi_3(\mathbf{x}) \end{bmatrix} \right)^T \begin{bmatrix} \phi_1(\mathbf{x}') \\ \phi_2(\mathbf{x}') \\ \phi_3(\mathbf{x}') \end{bmatrix} = 0. \tag{15}$$

Releasing the transpose operator gives the final form of the constraint on ϕ and **w** imposed by the proposed Spatio-Temporal Covariance loss function such as,

$$\begin{bmatrix} \phi_1(\mathbf{x}) & \phi_2(\mathbf{x}) & \phi_3(\mathbf{x}) \end{bmatrix} \begin{bmatrix} 0 & w_1w_2 & w_1w_3 \\ w_1w_2 & 0 & w_2w_3 \\ w_1w_3 & w_2w_3 & 0 \end{bmatrix} \begin{bmatrix} \phi_1(\mathbf{x}') \\ \phi_2(\mathbf{x}') \\ \phi_3(\mathbf{x}') \end{bmatrix} = 0. \tag{16}$$