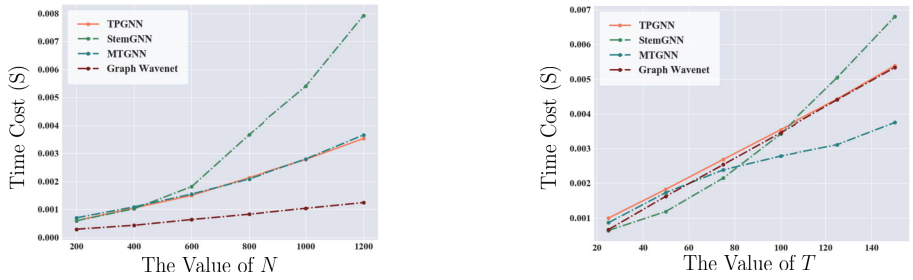
A Appendix

A.1 Limitation

Potential negative societal impacts. Accurate MTS forecasting benefits the application of intelligent systems. Since such systems contain many distributed sensors and produce a large amount of time-series data, predicting the systematic trend is crucial for controlling the system. However, suppose the algorithm is used for financial crimes or other illegal activities. In that case, it will have a bad influence on society. Fortunately, we can address such problems with privacy-preserving methods to improve data safety. Then, we can restrict the negative impacts since the data access is limited. Besides, the privacy-preserving approach also lowers the risk of personal data leakage.

A.2 Time Complexity Analysis



(a) The time complexity to varaible number.

(b) The time complexity to sequence length.

Figure 5: The efficiency of different methods on the synthetic datasets. The results demonstrate that TPGNN is efficient in processing MTS data.

Suppose that the MTS data has N variables, the input sequence length is T, the feature dimension is D_e, and the polynomial order is K. The TPGNN is composed of self-attention and TPG module, and it is known that self-attention has a time complexity as follows:

$$\mathcal{O}(NT^2D_e + NTD_e^2) \tag{11}$$

As the for TPG module, the time complexity for constructing the initial adjacency matrix is $\mathcal{O}(N^2)$, and the time complexity of Equation (7) is $\mathcal{O}(KT(ND_e^2 + N^2D_e))$. As a result, The overall time complexity of TPGNN is $\mathcal{O}(KTN^2D_e + KTND_e^2 + NT^2D_e)$. Since the K, D_E is small in practice, the efficiency of TPGNN is mainly decided by N and T.

In Figure 5, we investigate the efficiency of several SOTA methods on the synthetic datasets. The results show that TPGNN has competitive efficiency with MTGNN/Graph Wavenet, though these methods do not contain self-attention modules.

A.3 Data

We summarize the primary information of benchmark datasets in Table 1. All the six datasets come from the real-world application, we list the details as follows. All the data follow the MIT license.

A.3.1 Single-step forecasting

- Traffic: the traffic dataset from the California Department of Transportation contains road occupancy rates measured by 862 sensors in San Francisco Bay area freeways during 2015 and 2016.
- Solar-Energy: the solar-energy dataset from the National Renewable Energy Laboratory contains the solar power output collected from 137 PV plants in Alabama State in 2007.
- Electricity: the electricity dataset from the UCI Machine Learning Repository contains electricity consumption for 321 clients from 2012 to 2014.

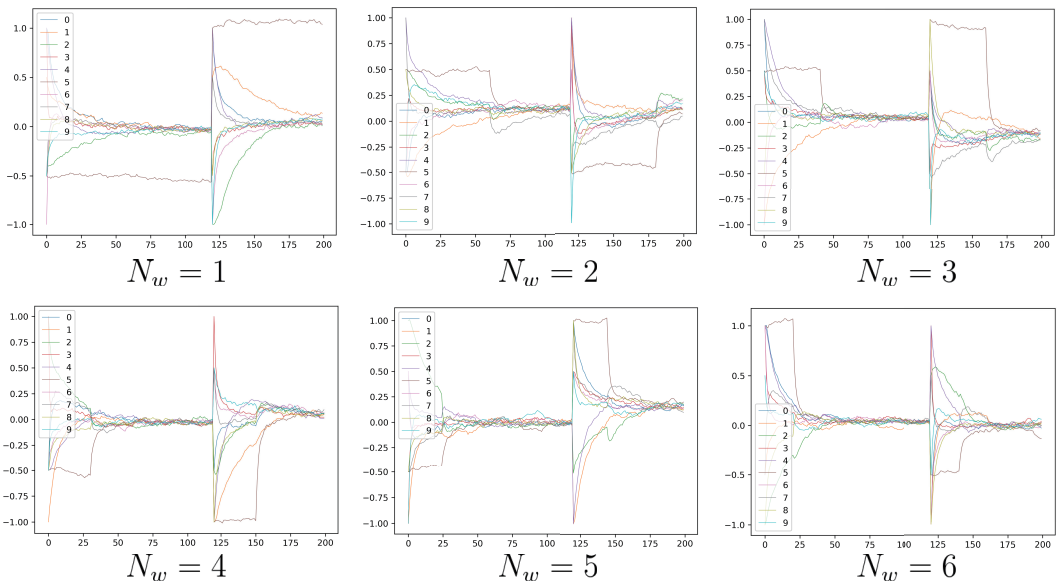


Figure 6: The first ten time-series data of the synthetic results, we show the intial 200 time steps. We observe that the increases of the N_w introduce complex behaviors in the MTS data.

- Exchange-Rate: the exchange-rate dataset contains the daily exchange rates of eight foreign countries, including Australia, British, Canada, Switzerland, China, Japan, New Zealand, and Singapore, ranging from 1990 to 2016.

The input sequence length is 168 and the output sequence length is 1. Models are trained to predict the target future step (horizon) 3, 6, 12, and 24.

A.3.2 Multi-step forecasting

- PEMS-D7: 44-day traffic data collected by 228 sensors in the California state highway system during the weekdays from May through June in 2012.
- PEMS-BAY: 6-month traffic data collected by 325 sensors in the Bay Area of California from January 1st through May 31th in 2017.

For the two traffic datasets, we forecast the subsequent 12 steps by observing a sequence of length 12 and evaluate the model performance at 3, 6, 12 steps.

A.3.3 Synthetic data with the NPR model

For clarification, we illustrate the synthetic algorithm for generating the MTS data in this section, the pseudo-code of our algorithm is showed in Algorithm 1. We firstly generate a ranomd orthogonal matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$ by decomposing a random matrix with SVD decomposition. Then, we define N_w constant matrices with random positive eigenvalues. Moreover, we sparsify the results to the given sparisty threshold to remove unnecessary connections. We then generate the required dynamic dependence MTS data with an NRW model, illustrated in lines 12 to 20 of the pseudo-code.

We further show some examples of the synthetic results in Figure 6, it illustrates that the dependence complexity increases with the N_w. In the evaluation, we set $\sigma = 0.01$, $T = 2400$, $T_p = 120$, and $\delta = 0.05$.

A.4 Baselines

There are several state-of-the-art baseline methods, we summarize them as follows.

Algorithm 1 Generating MTS data with the NPR model

Require: Total length T of the MTS data, number of variables N, number of constant matrices N_w.
Require: Length of the cycle T_p, stand deviation of the random walk σ, matrix sparsity δ.
Ensure: Synthetic MTS data $\mathbf{X} \in \mathbb{R}^{T \times N}$

Generate a random orthogonal matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$;
for $i = 1 \rightarrow N_w$ **do**
    $\mathbf{\Sigma}_i = diag(|\mathcal{N}(0,1)|, \ldots, |\mathcal{N}(0,1)|)$
    $\mathbf{G}_i = \mathbf{P}^T \mathbf{\Sigma}_i \mathbf{P}$, $\alpha = 0$
    $\mathbf{G}_i[\mathbf{G}_i < \alpha] = 0$
    while sparsity of $\mathbf{G}_i > \delta$ **do**
        $\alpha = \alpha + 0.02$
        $\mathbf{G}_i[\mathbf{G}_i < \alpha] = 0$
    end while
    $\mathbf{G}_i =$ Symmetric normalized Laplacian of $\mathbf{G}_i$
end for
$\mathbf{X} = \mathbf{0} \in \mathbb{R}^{T \times N}$
$T_l = T/T_p$
for $t = 1 \rightarrow T$ **do**
    if t-1% T_p=0 **then**
        initialize $\mathbf{x} \in \mathbb{R}^N$ randomly from $\{-1, -0.5, 0.5, 1\}$
    else
        $\mathbf{x} = \mathcal{N}(\mathbf{G}_{(t-1\%T_p)/T_l} \mathbf{X}[t-1], \sigma)$
    end if
    $\mathbf{X}[t] = \mathbf{x}$
end for

A.4.1 Single-step forecasting

- VAR-MLP: A model utilizes both of the multilayer perception (MLP) and auto-regressive model (VAR) [47].
- GP: Modeling time-series data with non-parametric method and Gaussian processes [10, 31].
- RNN-GRU: A model combines the recurrent unit and fully connected GRU.
- LSTNet: A deep neural network, which combines convolutional neural networks and recurrent neural networks [20].
- TPA-LSTM: A model utilizes convolution layers to extract temporal patterns and captures time series correlation with attention mechanism [34].
- MTGNN: MTGNN learns a static graph to represent the inter-series correlation and proposes a novel mix-hop propagation layer to capture the inter-series relation [41].

A.4.2 Multi-step forecasting

- ARIMA: A representative univariate time-series forecasting method based on the Kalman filter [23].
- FC-LSTM: Long Short-Term Memory Network, which is a recurrent neural network with fully connected LSTM hidden units [36].
- LSVR: A linear support vector regression (LSVR) model for travel-time predictions [39].
- STGCN: A spatial-temporal graph convolutional network, which incorporates graph convolutions with 1×1-kernel convolution layers [44].
- DCRNN: A diffusion convolutional recurrent neural network, which utilizes the diffusion graph convolution layer and reccurent unit [22].

Table 5: Results for selecting layers of encoder and decoder L and state embedding dimension D_e. We find that $L = 1, D_e = 64$ is an optimal configuration.

L	1	2	4	1	2	4	1	2	4
D_e	MAE			MAPE(%)			RMSE		
16	2.15/2.76/3.29	2.14/2.73/3.23	2.15/2.75/3.25	5.06/6.86/8.52	5.08/6.88/8.48	5.08/6.90/8.49	4.10/5.55/6.75	4.09/5.51/6.61	4.12/5.57/6.972
32	**2.11**/2.72/3.26	2.12/**2.70**/3.22	2.14/2.73/3.24	4.99/6.79/8.49	5.00/6.73/8.35	5.04/6.80/8.38	**4.07**/5.52/6.68	4.07/5.48/6.67	4.08/5.49/6.63
64	2.12/2.73/**3.22**	2.13/2.73/3.36	2.12/2.71/3.29	5.00/6.74/**8.25**	5.02/6.76/8.40	**4.97**/**6.73**/8.34	4.08/**5.45**/**6.56**	4.08/5.49/6.66	4.09/5.52/6.79

- StemGNN: A graph neural network, which combines Graph Fourier Transform (GFT) and Discrete Fourier Transform (DFT) together to capture inter-series correlations and temporal dependencies jointly in the spectral domain [4].
- Graph WaveNet: A spatial-temporal graph convolutional network, which learns a static graph to capture the spatial correlations [42].
- MTGNN: The same model to single-step forecasting, which is trained to predict multiple steps.

Parameter scale. We further list the model size of SOTA methods to illustrate our method is lightweight in the model scale. TPGNN: 0.31M, MTGNN: 0.44M, Graph WaveNet: 0.25M, STGCN:0.33M, StemGNN:1.22M, TPA-LSTM:0.12M, DCRNN:0.37M.

A.5 Metrics

Let $\tilde{\mathbf{Y}}^{(t)}$ and $\mathbf{Y}^{(t)}$ be the predicted and ground truth signal matrix at step t respectively, N is variable number. The evaluation metrics we use in the experiments are computed by:

$$
\begin{aligned}
MAE &= \frac{1}{N}\Sigma_{i=1}^{N}|\tilde{\mathbf{Y}}^{(t)}[i] - \mathbf{Y}^{(t)}[i]| \\
MAPE &= \frac{1}{N}\Sigma_{i=1}^{N}|\frac{\tilde{\mathbf{Y}}^{(t)}[i] - \mathbf{Y}^{(t)}[i]}{\mathbf{Y}^{(t)}[i]}| \times 100\% \\
RMSE &= \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(\tilde{\mathbf{Y}}^{(t)}[i] - \mathbf{Y}^{(t)}[i])^2} \\
RRSE &= \frac{\sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(\tilde{\mathbf{Y}}^{(t)}[i] - \mathbf{Y}^{(t)}[i])^2}}{\mathrm{std}(\mathbf{Y}^{(t)}) * \sqrt{N/(N-1)}} \\
CORR &= \frac{\Sigma_{i=1}^{N}(\tilde{\mathbf{Y}}^{(t)}[i] - \mathrm{mean}(\tilde{\mathbf{Y}}^{(t)}))(\mathbf{Y}^{(t)}[i] - \mathrm{mean}(\mathbf{Y}^{(t)}))}{N\,\mathrm{std}(\tilde{\mathbf{Y}}^{(t)})\,\mathrm{std}(\mathbf{Y}^{(t)})}
\end{aligned}
\tag{12}
$$

A.6 Experimental Settings

For the datasets with a prior distance matrix D, we follow the pre-processing method of STGCN [44]. It constructs the adjacency matrix W as follows, and they select the hyperparameters as $\sigma^2 = 10, \epsilon = 0.5$:

$$
W_{ij} = \begin{cases} \exp(\frac{-D_{ij}^2}{\sigma^2}), & i \neq j \text{ and } \exp(\frac{-D_{ij}^2}{\sigma^2}) \geq \epsilon; \\ 0, & \text{otherwise} \end{cases}
\tag{13}
$$

We conduct all experiments under the environment of one Intel(R) Xeon(R) Gold 6254 CPU @ 3.10GHz and NVIDIA GeForce RTX 2080Ti GPU. We repeat the experiment 10 times and report the average of evaluation metrics. In the training stage, we adopt L1 loss and Adam optimizer. The initial learning rate is 10^-3 and it decays by the ratio of 0.3 for every 300 epochs. The layer number L is 1, the embedding dimension D_e is 64, the polynomial order K is 2, and the dimension of variable embedding is 10. The training epoch is 1000, 1500 for single-step and multi-step forecasting, respectively. The batch size is 50. We use dropout to improve the model robustness to the noisy data. The rate for dropout is 0.2.

Table 6: Results for selecting the polynomial order of the TPG module K. We find that $K = 2$ is an optimal configuration.

K	1	2	3	4
MAE	2.13/2.75/3.37	**2.12/2.72/3.22**	2.13/2.84/3.29	2.16/2.88/3.44
MAPE(%)	5.10/6.78/8.53	**5.00/6.73/8.22**	5.04/6.77/8.26	5.12/6.98/8.57
RMSE	4.11/5.53/6.79	**4.05/5.45/6.56**	4.06/5.48/6.59	4.19/5.84/6.98

Table 7: Results for selecting the dimension of node (variable) embedding C. We find that $C = 10$ is an optimal configuration.

C	5	10	20
MAE	2.12/2.72/3.27	**2.12/2.72/3.22**	2.14/2.79/3.26
MAPE(%)	5.00/6.75/8.39	**5.00/6.73/8.22**	5.04/6.81/8.35
RMSE	4.07/5.49/6.71	**4.05/5.45/6.56**	4.06/5.47/6.63

A.7 Parameter Study

This section investigates the model configuration on the PEMSD7(M) dataset. The first two hyperparameters are the layer number of encoder and decoder L and the embedding dimension of the matrix signal D_e, we conclude the results in Table 5, which indicates that the configuration $L = 1, D_e = 64$ is a good setting since it has good performance and efficiency.

Another two crucial hyperparameters are the K and C. The first one decides the polynomial order of the TPGNN, the second one controls the embedding dimension of the variable embedding. We show the results in Table 6 and Table 7, which implies the $K = 2, C = 10$ is the best parameter selection.

A.8 More results for ablation study

This section shows more ablation results on the PEMS-D7 of the different horizons to illustrate that our designs are indispensable. In Table 8 and Table 9, we find TPGNN outperforms other variants consistently, which demonstrate the effectiveness of our components. Although w/o normalize achieves better performance on MAPE, it performs badly in the long-term forecasting.

Table 8: Prediction results for ablation study on the PEMS-D7 dataset over the horizon of 3.

Metrics	TPGNN	w/o TPG	w/o dynamic	w/o overview	w/o normalize	w/o K-matrices
MAE	**2.124±0.012**	2.168±0.028	2.169±0.035	2.138±0.029	2.132±0.039	2.143±0.033
MAPE(%)	5.003±0.032	5.017±0.077	5.051±0.026	5.057±0.089	**5.000±0.067**	5.017±0.112
RMSE	**4.049±0.027**	4.127±0.043	4.106±0.078	4.072±0.101	4.074±0.046	4.077±0.056

Table 9: Prediction results for ablation study on the PEMS-D7 dataset over the horizon of 6.

Metrics	TPGNN	w/o TPG	w/o dynamic	w/o overview	w/o normalize	w/o K-matrices
MAE	**2.716±0.021**	2.845±0.023	2.823±0.065	2.748±0.057	2.727±0.044	2.748±0.029
MAPE(%)	6.728±0.052	6.998±0.087	6.842±0.034	6.825±0.082	**6.714±0.055**	6.751±0.119
RMSE	**5.452±0.048**	5.662±0.067	5.594±0.056	5.492±0.071	5.471±0.046	5.511±0.062

A.9 Theoretical Analysis

Lemma 1 *Let $F = \{\mathbf{A}_1, \ldots, \mathbf{A}_k\}, \mathbf{A}_i \in \mathbb{R}^{N\times N}$, be a set of diagbolizable matrices. Then F is a commuting($\mathbf{A}_i\mathbf{A}_j = \mathbf{A}_j\mathbf{A}_i, \forall 1 \leq i, j \leq k$), if and only if there exists and ivertible $\mathbf{S} \in \mathbb{R}^{N\times N}$ such that $\mathbf{S}^{-1}\mathbf{A}_i\mathbf{S}, \mathbf{A}_i \in F$ is diagonal.*

Proof. This is a known result, which indicates that the commutation and simultaneously diagonalization are equivalent properties. We briefly conclude the proving process. If F can be diagonalized by a $\mathbf{S}$, it is trivial to prove that F is commuting due to the diagonal matrices are commutative. We

inductively prove that F can be diagonalized by a $\mathbf{S}$. If $N = 1$, then it's a trivial result. Suppose the theorem holds for matrices of size $N \leq k-1$, we prove that the theorem is true for $N = k$. Let $N = k$, $\mathbf{A} \in F$, and $\mathbf{A}$ is diagnolizable with eigenvalues $\{\lambda_1, \ldots, \lambda_r\}$ where $r \geq 2$ and $\mathbf{AB} = \mathbf{BA}$ for each $\mathbf{B} \in F$. $\mathbf{A}$ is similar to a diagonal matrix, so without loss of generality assume $\mathbf{A}$ is diagonal. Since each $\mathbf{B}$ commutes with the diagonal matrix $\mathbf{A}$, each $\mathbf{B} \in F$ is a block diagnoal matrix. Since $\mathbf{A}$ has at least two distinct entries, each block of each $\mathbf{B}$ has size $k-1$ or less. By the block-wise commutation property, together with the inductive hypothesis, all the blocks in all $\mathbf{B} \in F$ are simultaneously diagonalizable. Thus there exist fixed matrices $T_1, T_2, \ldots, T_r$ such that conjugating with each $\mathbf{B} \in F$ with $T = diag(T_1, T_2, \ldots, T_r)$ gives a block matrix in which the blocks are diagonal, which is a diagonal matrix.

Lemma 2 *Let $\mathbf{G}, \mathbf{A} \in \mathbb{R}^{N \times N}$ are symmetric matrices. Since they are real symmetric matrices, $\mathbf{G}, \mathbf{A}$ can be diagonalized by orthogonal matrices. Let $\mathbf{P}(\mathbf{A}), \mathbf{P}(\mathbf{G})$ are the corresponding orthogonal matrices, we have $\mathbf{P}(\mathbf{A})^T\mathbf{A}\mathbf{P}(\mathbf{A}) = diag(\lambda_1, \ldots, \lambda_N), \mathbf{P}(\mathbf{G})^T\mathbf{G}\mathbf{P}(\mathbf{G}) = diag(\sigma_1, \ldots, \sigma_N)$. Let $\mathbf{G_A} = \mathbf{P}(\mathbf{A})^T\mathbf{G}\mathbf{P}(\mathbf{A})$, then we have the following results:*

$$||\mathbf{G} - \Sigma_{k=0}^{K} a_k \mathbf{A}^k||_F^2 = tr((\mathbf{G_A} - diag(\mathbf{G_A}))^2) + tr((diag(\mathbf{G_A}) - p_K(\mathbf{D}))^2) \tag{14}$$

where $tr(\cdot)$ is the trace of matrix, $(a_0, \ldots, a_K)$ is the polynomial coefficients, $p_K(\mathbf{D}) = \Sigma_{k=0}^{K} a_k \operatorname{diag}(\lambda_1, \ldots, \lambda_N)^k$ is the matrix polynomial of $\mathbf{A}$'s eigenvalues.

Proof. Let $\mathbf{E} = \mathbf{G} - \Sigma_{k=0}^{K} a_k \mathbf{A}^k$, we have the following equations:

$$\begin{aligned}
\mathbf{E} &= \mathbf{P}(\mathbf{A})\mathbf{P}(\mathbf{A})^\mathbf{T}\mathbf{G}\mathbf{P}(\mathbf{A})\mathbf{P}(\mathbf{A})^\mathbf{T} - \mathbf{P}(\mathbf{A})(\Sigma_{k=0}^{K} a_k \operatorname{diag}(\lambda_1, \ldots, \lambda_N)^k)\mathbf{P}(\mathbf{A})^\mathbf{T} \\
&= \mathbf{P}(\mathbf{A})(\mathbf{G_A} - \Sigma_{k=0}^{K} a_k \operatorname{diag}(\lambda_1, \ldots, \lambda_N)^k))\mathbf{P}(\mathbf{A})^\mathbf{T} \\
&= \mathbf{P}(\mathbf{A})(\mathbf{G_A} - p_K(\mathbf{D}))\mathbf{P}(\mathbf{A})^\mathbf{T}
\end{aligned} \tag{15}$$

Moreover, $||\mathbf{G} - \Sigma_{k=0}^{K} a_k \mathbf{A}^k||_F^2 = tr(\mathbf{E}^2)$ ($\mathbf{E}$ is symmetric). Therefore, we have:

$$\begin{aligned}
||\mathbf{G} - \Sigma_{k=0}^{K} a_k \mathbf{A}^k||_F^2 &= tr((\mathbf{G_A} - \operatorname{diag}(\mathbf{G_A}) + \operatorname{diag}(\mathbf{G_A}) - p_K(\mathbf{D}))^2) \\
&= tr((\mathbf{G_A} - \operatorname{diag}(\mathbf{G_A}))^2) + 2tr((\mathbf{G_A} - \operatorname{diag}(\mathbf{G_A}))(\operatorname{diag}(\mathbf{G_A}) - p_K(\mathbf{D}))) \\
&\quad + tr((\operatorname{diag}(\mathbf{G_A}) - p_K(\mathbf{D}))^2) \\
&= tr((\mathbf{G_A} - diag(\mathbf{G_A}))^2) + tr((diag(\mathbf{G_A}) - p_K(\mathbf{D}))^2)
\end{aligned} \tag{16}$$

Review. Lemma 2 clearly illustrates the approximation ability of a matrix polynomial. The error is controlled by two terms. The value of the first term depends on whether $\mathbf{P}(\mathbf{A})$ is able to diagonalize $\mathbf{G}$. As for the second term, it corresponds to a least square regression problem.

Theorem 2 *Given the setting same as Lemma 2. Let $\mathbf{Q} = \mathbf{P}(\mathbf{A})^T\mathbf{G}$, we define a discrepancy matrix $\mathbf{D}_{AG}, (\mathbf{D}_{AG})_{ij} = \mathbf{Q}_{ij}^2$. Let $\mathbf{d}_k$ is the k-th row vector of $\mathbf{D}_{AG}$, and $\lambda_{min}, \lambda_{max}$ be the minimum and maximum eigenvalues of matrix $\Sigma_{k=1}^{N} \mathbf{d}_k \mathbf{d}_k^T$. We then have the following estimation of the frist term of Equation 16:*

$$(1 - \lambda_{max})tr(\mathbf{G}^2) \leq tr((\mathbf{G_A} - diag(\mathbf{G_A}))^2) \leq (1 - \lambda_{min})tr(\mathbf{G}^2) \tag{17}$$

Proof. We first expand the first term of Equation 16.

$$\begin{aligned}
tr((\mathbf{G_A} - diag(\mathbf{G_A}))^2) &= tr(\mathbf{G}_A^2 + diag(\mathbf{G_A})^2 - 2diag(\mathbf{G_A})\mathbf{G_A}) \\
&= tr(\mathbf{G}^2) + tr(diag(\mathbf{G_A})^2 - 2diag(\mathbf{G_A})\mathbf{G_A})
\end{aligned} \tag{18}$$

Besides, we observe that:

$$(diag(\mathbf{G_A})\mathbf{G_A})_{ij} = (\mathbf{G_A})_{ii}(\mathbf{G_A})_{ij} \tag{19}$$

Therefore,

$$\begin{aligned}
tr(diag(\mathbf{G_A})\mathbf{G_A}) &= \Sigma_i (diag(\mathbf{G_A})\mathbf{G_A})_{ii} \\
&= \Sigma_i (\mathbf{G_A})_{ii}^2 \\
&= tr(diag(\mathbf{G_A})^2)
\end{aligned} \tag{20}$$

Combining the result with Equation 18, we derive the following result:

$$\begin{aligned} tr((\mathbf{G_A} - diag(\mathbf{G_A}))^2) &= tr(\mathbf{G}^2) + tr(diag(\mathbf{G_A})^2 - 2diag(\mathbf{G_A})\mathbf{G_A}) \\ &= tr(\mathbf{G}^2) + tr(diag(\mathbf{G_A})^2) - 2tr(diag(\mathbf{G_A})^2) \\ &= tr(\mathbf{G}^2) - tr(diag(\mathbf{G_A})^2) \end{aligned} \tag{21}$$

Since $\mathbf{G_A} = \mathbf{P}(\mathbf{A})^T\mathbf{G}\mathbf{P}(\mathbf{A})$. Let $\mathbf{u}_1, \ldots, \mathbf{u}_N$ are the column vectors of $\mathbf{P}(\mathbf{A})$, we have $(\mathbf{G_A})_{ij} = \mathbf{u}_i^T\mathbf{G}\mathbf{u}_j$. Therefore, we derive the following equation:

$$\begin{aligned} tr(diag(\mathbf{G_A})^2) &= \Sigma_{k=1}^N(\mathbf{G_A})_{kk}^2 \\ &= \Sigma_{k=1}^N(\mathbf{u}_k^T\mathbf{G}\mathbf{u}_k)^2 \end{aligned} \tag{22}$$

Let $\mathbf{Q} = \mathbf{P}(\mathbf{A})^T\mathbf{P}(\mathbf{G})$, $\mathbf{v}_1, \ldots, \mathbf{v}_N$ are the column vectors of $\mathbf{P}(\mathbf{G})$. We then have $\mathbf{P}(\mathbf{A}) = \mathbf{P}(\mathbf{G})\mathbf{Q}^T$, and $\mathbf{u}_k = \Sigma_{s=1}^m Q_{sk}\mathbf{v}_s$. Therefore, we derive the following result:

$$\begin{aligned} tr(diag(\mathbf{G_A})^2) &= \Sigma_{k=1}^m(\mathbf{u}_k^T\mathbf{G}\mathbf{u}_k)^2 \\ &= \Sigma_{k=1}^m((\Sigma_{s=1}^m Q_{sk}\mathbf{v}_s^T)\mathbf{G}(\Sigma_{s=1}^m Q_{sk}\mathbf{v}_s))^2 \end{aligned} \tag{23}$$

Because $\mathbf{v}_s^T\mathbf{G}\mathbf{v}_s = \sigma_s, \mathbf{v}_s^T\mathbf{G}\mathbf{v}_t = 0, if\ s \neq t$, we have $tr(diag(\mathbf{G_A})^2) = \Sigma_{k=1}^m(\Sigma_{s=1}^m\sigma_s Q_{sk}^2)^2$. We then define a discrepancy $\mathbf{D}_{AG}$, $(\mathbf{D}_{AG})_{ij} = (Q)_{ij}^2$. Let $\mathbf{d}_k$ be the k-th column vector of $\mathbf{D}_{AG}$, $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N)^T$ is a vector composed of $\mathbf{G}$'s eigenvalues. Therefore, we have the following equation:

$$\begin{aligned} tr(diag(\mathbf{G_A})^2) &= \Sigma_{k=1}^m(\sigma^T\mathbf{d}_k)^2 \\ &= \Sigma_{k=1}^m\sigma^T\mathbf{d}_k\mathbf{d}_k^T\sigma \\ &= \sigma^T(\Sigma_{k=1}^N\mathbf{d}_k\mathbf{d}_k^T)\sigma \end{aligned} \tag{24}$$

Due to $\Sigma_{k=1}^N\mathbf{d}_k\mathbf{d}_k^T$ is a symmetric matrix, according the property of Rayleigh quotient, we know:

$$\lambda_{min} \leq \frac{\sigma^T(\Sigma_{k=1}^m\mathbf{d}_k\mathbf{d}_k^T)\sigma}{\sigma^T\sigma} \leq \lambda_{max} \tag{25}$$

Where $\lambda_{min}, \lambda_{max}$ are the minimum and maximum eigenvalues of $\Sigma_{k=1}^N\mathbf{d}_k\mathbf{d}_k^T$. Besides, according to the property of matrix trace, $tr(\mathbf{G}^2) = tr(\mathbf{P}(\mathbf{G})diag(\sigma_1, \ldots, \sigma_N)^2\mathbf{P}(\mathbf{G})^T) = \sigma^T\sigma$. Therefore, we have the following result:

$$\lambda_{min}tr(\mathbf{G}^2) \leq tr(diag(\mathbf{G_A})^2) \leq \lambda_{max}tr(\mathbf{G}^2) \tag{26}$$

According to Equation 21 and Equation 26, we finish the proof of Theorem 2.

Lemma 3 *Given the setting same as Lemma 2, we have the following estimation of the second term of Equation 16:*

$$0 \leq tr((diag(\mathbf{G_A}) - p_K(\mathbf{D}))^2) \leq tr(diag(\mathbf{G_A})^2) - \frac{1}{N}(tr(\mathbf{G_A}))^2 - \frac{tr(\mathbf{D}\mathbf{G_A}) - \frac{1}{N}tr(\mathbf{D})tr(\mathbf{G_A})}{tr(\mathbf{D}^2) - \frac{1}{N}tr(\mathbf{D})^2} \tag{27}$$

Specifically, if $\mathbf{D}$ has N different values and $K = N - 1$, then the upper bound is zero.

Proof. Since $tr((diag(\mathbf{G_A}) - p_K(\mathbf{D}))^2) = \Sigma_{i=1}^N(\mathbf{G_A}_{ii} - \Sigma_{k=0}^K a_k\lambda_i^k)^2$, it corresponds to a least square problem. Equation 3 is a known result of least square optimization under $K = 1$. Due to $K \geq 1$, Equation 3 holds obviously. Besides, if $\mathbf{D}$ has N different values, i.e., $\mathbf{A}$ has N different eigenvalues, and $K = N - 1$. Then the upper bound of $tr((diag(\mathbf{G_A}) - p_K(\mathbf{D}))^2)$ is zero with Lagrange interpolation.

Theorem 3 ***(Formal statement of Theorem 1).****Let $G = \{\mathbf{G}^{(1)}, \ldots, \mathbf{G}^{(T)}\}, \mathbf{G}^{(t)} \in \mathbb{R}^{N\times N}$ be the symmetric normalized Laplacian of the optimal structures for time step 1 to T,$\mathbf{A} \in \mathbb{R}^{N\times N}$ is the initial adjacency of TPGNN. Suppose G is commuting, $\mathbf{A}$ and elements of G are symmetric. Then, there is a $\mathbf{P}(\mathbf{G}) \in \mathbb{R}^{N\times N}$ such that $\mathbf{P}(\mathbf{G})^T\mathbf{G}^{(t)}\mathbf{P}(\mathbf{G})$ is diagonal matrix for each t. Let*

$\mathbf{P}(\mathbf{A})^T\mathbf{A}\mathbf{P}(\mathbf{A})$ *be a diagonal matrix, we define* $\mathbf{D}_{AG}$ *the same as Theorem 2, and* $\lambda_{min}, \lambda_{max}$ *are the minimum and maximum eigenvalues of* $\Sigma_{k=1}^{N}\mathbf{d}_k\mathbf{d}_k^T$. *If* $\mathbf{A}$ *has* N *different singular values, the polynomial's order* $K = N - 1$. *Then the approximation error* $e^{(1:T)} = \frac{1}{T}\Sigma_{t=1}^{T}||\mathbf{W}^{(t)} - \mathbf{G}^{(t)}||_F^2$ *satisfies the following estimation:*

$$(1 - \lambda_{max})\mathbb{E}_t||\mathbf{G}^{(t)}||_F^2 \leq \mathrm{e}^{(1:T)} \leq (1 - \lambda_{min})\mathbb{E}_t||\mathbf{G}^{(t)}||_F^2, \tag{28}$$

where $\mathbb{E}_t||\mathbf{G}^{(t)}||_F^2 = \frac{1}{T}\Sigma_{t=1}^{T}||\mathbf{G}^{(t)}||_F^2$ *is the average norm of the Laplacians.*

Proof. Firstly, G is diagonalizable since its elements are symmetric. According to Lemma 1, we can find a $\mathbf{S} \in \mathbb{R}^{N\times N}$ such that $\mathbf{S}^{-1}\mathbf{G}^{(t)}\mathbf{S}$ is diagonal for each t. Therefore, we can find an othogonal matrix $\mathbf{P}(\mathbf{G})$ such that $\mathbf{P}(\mathbf{G})^T\mathbf{G}^{(t)}\mathbf{P}(\mathbf{G})$ is diagonal using Gram-Schmidt orthogonalization. We now consider each $||\mathbf{W}^{(t)} - \mathbf{G}^{(t)}||_F^2$. According to Theorem 2 and Lemma 3, we have the following estimation:

$$(1 - \lambda_{max})||\mathbf{G}^{(t)}||_F^2 \leq ||\mathbf{W}^{(t)} - \mathbf{G}^{(t)}||_F^2 \leq (1 - \lambda_{min})||\mathbf{G}^{(t)}||_F^2 \tag{29}$$

We thus prove the Theorem 3 with the Equation 29.