MDG: A Multi-Task Dynamic Graph Generation Framework for Multivariate Time Series Forecasting (Supplementary Material)

Longhao Huang*† Shengbo Chen*† Jidong Yuan *†‡ Xu Li§

I Proofs of Theorems

In this section, we present detailed proofs of the theorems that were made in Section 4.4.

Theorem 1. For a random sample $\boldsymbol{x}$, the following inequality holds:

$$
\begin{aligned}
&|L_1(hx(\boldsymbol{x},y)) - L_1(hx_j(\boldsymbol{x}),y)| \leq \frac{nb_1\delta\tau B_{w_1}}{b_2 Z_1 c B_{w_2}^2} \\
&(\delta\tau B_{w_1} + \sqrt{(\delta\tau B_{w_1})^2 + \frac{4Z_1\delta_2 c B_{w_2}^2 B_{w_1} B_A \delta\varepsilon}{nb_1}}) \\
&+ 2B_{w_1} B_A \varepsilon.
\end{aligned} \tag{1.1}
$$

Proof: Suppose theoretically optimal $\frac{\nabla L(A)}{\nabla A} = 0$, $\frac{\nabla L_j(A_j)}{\nabla A_j} = 0$,

$$
\begin{aligned}
&|L_1(WA\boldsymbol{x},y) - L_1(WA_j\boldsymbol{x},y)| \\
&\leq \max_{(\boldsymbol{x},y)} L_1(WA\boldsymbol{x},y) - L_1(WA_j\boldsymbol{x},y)| \\
&\leq \max_{(\boldsymbol{x},y)} \delta||WA\boldsymbol{x} - WA_j\boldsymbol{x}||_2 \qquad (\textbf{Assumption 1}) \\
&= \max_{(\boldsymbol{x},y)} \delta||W(A-A_j)\boldsymbol{x}||_2 \\
&= \max_{\alpha} \delta||\sum_{i=1}^{n} d_i W(A-A_j) r_i + W(A-A_j)\eta|| \\
&\quad (\textbf{Assumption 3}) \\
&\leq \max_{\alpha} \delta(||\sum_{i=1}^{n} d_i W(A-A_j) r_i|| + ||W(A-A_j)\eta||) \\
&\overset{(9.1)}{\leq} \delta(\sqrt{\sum_{i=1}^{n} d_i^2}\sqrt{\sum_{i=1}^{n} ||W(A-A_j)r_i||^2} + 2B_w B_A\varepsilon) \\
&\leq \delta(\tau B_w \sqrt{\sum_{i=1}^{n} ||(A-A_j)r_i||^2} + 2B_w B_A\varepsilon),
\end{aligned} \tag{1.2}
$$

where inequality (9.1) comes from Cauchy-Schwarz inequality: $(\sum_{i=1}^{n} a_i b_i)^2 \leq (\sum_{i=1}^{n} a_i^2)(\sum_{i=1}^{n} b_i^2)$.

Bounding $\sqrt{\sum_{i=1}^{n} ||(A-A_j)r_i||^2}$

Define $B_L(f||g) = L(f) - L(g) - < f-g, \partial L(g) >$

$$
\begin{aligned}
L(A) &= b_1 l_1 + b_2 l_2 + b_3 l_3 \\
&= b_1 \frac{1}{Z_1} \sum_{i=1}^{Z_1} L_1(W_1 A X_{1i}, y_{1i}) \\
&\quad + b_2 \frac{1}{Z_2} \sum_{i=1}^{Z_2} L_2(W_2 A X_{2i}, y_{2i}) \\
&\quad + b_3 \frac{1}{Z_3} \sum_{i=1}^{Z_3} L_3(W_3 A X_{3i}, y_{3i}),
\end{aligned} \tag{1.3}
$$

$$
\begin{aligned}
L_j(A) &= b_1 \frac{1}{Z_1} \sum_{i=1, i\neq j}^{Z_1} [L_1(W_1 A X_{1i}, y_{1i}) \\
&\quad + L_1(W_1 A X_{1j}^{'}, y_{1j}^{'})] \\
&\quad + b_2 \frac{1}{Z_2} \sum_{i=1}^{Z_2} L_2(W_2 A X_{2i}, y_{2i}) \\
&\quad + b_3 \frac{1}{Z_3} \sum_{i=1}^{Z_3} L_3(W_3 A X_{3i}, y_{3i}), \\
N(A) &= \sum_{i=1}^{n} \frac{1}{Z_{k_i}} l(W_{k_i} A r_i, y_{r_i}).
\end{aligned} \tag{1.4}
$$

Inequality formula:

$$
\begin{aligned}
&min(b_2,b_3)[B_N(A_j||A) + B_N(A||A_j)] \leq \\
&B_L(A_j||A) + B_{L_j}(A||A_j),
\end{aligned} \tag{1.5}
$$

establishes because

$$
min(b_2,b_3) B_N(A_j||A) \leq B_L(A_j||A), \tag{1.6}
$$

$$
min(b_2,b_3) B_N(A||A_j) \leq B_L(A_j||A) + B_{L_j}(A||A_j).
$$

*School of Computer and Information Technology, Beijing Jiaotong University. {20120360, 21120338, yuanjd}@bjtu.edu.cn
†Beijing Key Laboratory of Traffic Data Analysis and Mining.
‡Corresponding author.
§School of Civil Engineering, Beijing Jiaotong University. xuli@bjtu.edu.cn

Then,

(1.7)

$$
\begin{aligned}
&B_L(A_j||A) + B_{L_j}(A||A_j) \\
&\geq min(b_2,b_3)(N(A_j) - N(A) - < A_j - A, \nabla N(A) > \\
&\quad + N(A) - N(A_j) - < A - A_j, \nabla N(A_j) >) \\
&= min(b_2,b_3) < A - A_j, \nabla N(A) - \nabla N(A_j) > \\
&= min(b_2,b_3) < A - A_j, \nabla \sum_{i=1}^{n} \frac{1}{Z_{k_i}} l(W_{k_i} A r_i, y_{r_i}) \\
&\quad - \nabla \sum_{i=1}^{n} \frac{1}{Z_{k_i}} l(W_{k_i} A_j r_i, y_{r_i}) > \\
&= min(b_2,b_3) < A - A_j, \sum_{i=1}^{n} \frac{1}{Z_{k_i}} \nabla W_{k_i}^T l(W_{k_i} A r_i, y_{r_i}) r_i^T \\
&\quad - \sum_{i=1}^{n} \frac{1}{Z_{k_i}} \nabla W_{k_i}^T l(W_{k_i} A_j r_i, y_{r_i}) r_i^T > \\
&= min(b_2,b_3) \sum_{i=1}^{n} \frac{1}{Z_{k_i}} < W_{k_i}(A - A_j) r_i, \\
&\quad \nabla l(W_{k_i} A r_i, y_{r_i}) - \nabla l(W_{k_i} A_j r_i, y_{r_i}) > \\
&\geq min(b_2,b_3) \sum_{i=1}^{n} \frac{c}{Z_{k_i}} || < min(W_2, W_3)(A - A_j) r_i > ||_2^2 \\
&\qquad (\textbf{Assumption 2}) \\
&\geq \frac{min(b_2,b_3)c}{max(Z_2,Z_3)} B^2_{min(W_2,W_3)} \sum_{i=1}^{n} ||(A - A_j) r_i||_2^2,
\end{aligned}
$$

(1.8)

$$
\begin{aligned}
&min(b_2,b_3)[B_N(A_j||A) + B_N(A||A_j)] \\
&\leq L(A_j) - L(A) - < A_j - A, \frac{\nabla L(A)}{\nabla A} > \\
&\quad L_j(A) - L_j(A_j) - < A - A_j, \frac{\nabla L_j(A_j)}{\nabla A_j} > \\
&= L(A_j) - L(A) + L_j(A) - L_j(A_j) \\
&= b_1 \frac{1}{Z_1} \sum_{i=1}^{Z_1} L_1(W_1 A_j X_{1i}, y_{1i}) \\
&\quad + b_2 \frac{1}{Z_2} \sum_{i=1}^{Z_2} L_2(W_2 A_j X_{2i}, y_{2i}) \\
&\quad + b_3 \frac{1}{Z_3} \sum_{i=1}^{Z_3} L_3(W_3 A_j X_{3i}, y_{3i}) \\
&\quad - b_1 \frac{1}{Z_1} \sum_{i=1}^{Z_1} L_1(W_1 A X_{1i}, y_{1i}) \\
&\quad - b_2 \frac{1}{Z_2} \sum_{i=1}^{Z_2} L_2(W_2 A X_{2i}, y_{2i}) \\
&\quad - b_3 \frac{1}{Z_3} \sum_{i=1}^{Z_3} L_3(W_3 A X_{3i}, y_{3i}) \\
&\quad + \frac{b_1}{Z_1} \sum_{i=1, i\neq j}^{Z_1} [L_1(W_1 A X_{1i}, y_{1i}) + L_1(W_1 A X_{1j}, y_{1j})] \\
&\quad + \frac{b_2}{Z_2} \sum_{i=1}^{Z_2} L_2(W_2 A X_{2i}, y_{2i}) + \frac{b_3}{Z_3} \sum_{i=1}^{Z_3} L_3(W_3 A X_{3i}, y_{3i}) \\
&\quad - \frac{b_1}{Z_1} \sum_{i=1, i\neq j}^{Z_1} [L_1(W_1 A_j X_{1i}, y_{1i}) + L_1(W_1 A_j X_{1j}, y_{1j})] \\
&\quad - \frac{b_2}{Z_2} \sum_{i=1}^{Z_2} L_2(W_2 A_j X_{2i}, y_{2i}) - \frac{b_3}{Z_3} \sum_{i=1}^{Z_3} L_3(W_3 A_j X_{3i}, y_{3i}) \\
&= \frac{b_1}{Z_1}[L_1(W_1 A_j X_{1j}, y_{1j}) - L_1(W_1 A_j X'_{1j}, y'_{1j})] \\
&\quad + \frac{\delta_1}{Z_1}[-L_1(W_1 A X_{1j}, y_{1j}) + L_1(W_1 A X'_{1j}, y'_{1j})] \\
&\leq \frac{b_1}{Z_1}(\delta ||W_1(A_j - A) X_{1j}|| + \delta ||W_1(A - A_j) X'_{1j}||) \\
&\quad (\textbf{Assumption 1}) \\
&\leq \frac{2\delta b_1}{Z_1}(max||W_1(A_j - A) X_{1j}||) \\
&\leq \frac{2\delta b_1}{Z_1}(\tau B_{W_1} \sqrt{\sum_{i=1}^{n} ||(A - A_j) r_i||^2} + 2 B_{W_1} B_A \varepsilon). \\
&\quad (\text{According to Eq. 1.2})
\end{aligned}
$$

Let $\sqrt{\sum_{i=1}^{n} ||(A - A_j) r_i||} = \kappa$. Putting Eq. 1.7 and Eq. 1.8 together, we can obtain:

(1.9)

$$
\frac{min(b_2,b_3)c}{max(Z_2,Z_3)} B^2_{W_2} \kappa^2 \leq \frac{2\delta b_1 \tau}{Z_1} B_{W_1} \kappa + \frac{4\delta b_1 B_{W_1} B_A \varepsilon}{Z_1}.
$$

According to the properties of the real root of one-place 2-th order equation, the solution of Eq. 1.9 is:

(1.10)

$$
\begin{aligned}
\kappa \leq \frac{max(Z_2,Z_3) b_1}{min(b_2,b_3) Z_1 c B_{W_2}} &(\delta \tau B_{W_1} \\
&+ \sqrt{(\delta \tau B_{W_1})^2 + \frac{4 Z_1 min(b_2,b_3) c B^2_{W_2} B_{W_1} B_A \delta \varepsilon}{max(Z_2,Z_3) b_1}}).
\end{aligned}
$$

Putting Eq. 1.2 and Eq. 1.10 together, we can prove Theorem 1.

Theorem 2. Known that L_1 is bounded by $\mathcal{M} > 0$. The algorithm for learning $\boldsymbol{F}$ is uniformly stable with respect to the domain of the prediction task and any $\lambda > 0$, with probability at least $1 - \lambda$, the generalization

bound is [1]:

(1.11)

$$E_{(x,y)\sim\mu}L_1(\hat{y},y) - \frac{1}{Z}\sum_{i=1}^{Z}L_1(\hat{y}_i,y_i) \leq \frac{2max(Z_2,Z_3)b_1\delta^2\tau^2B_{w_1}^2}{min(b_2,b_3)Z_1cB_{w_2}^2} + (\frac{4max(Z_2,Z_3)b_1\delta^2\tau^2B_{w_1}^2}{min(b_2,b_3)cB_{w_2}^2} + \mathcal{M})\sqrt{\frac{log(1/\lambda)}{2Z_1}}.$$

Proof: According to Theorem 1, we have proven that MDG for learning $\boldsymbol{F}$ is uniformly stable. For simplicity, we let $\varepsilon = 0$. The following inequality holds:

$$\beta \leq \frac{2max(Z_2,Z_3)b_1\delta^2\tau^2B_{W_1}^2}{min(b_2,b_3)Z_1cB_{W_2}^2}. \tag{1.12}$$

The generalization bound derived using uniform stability, we can obtain:

(1.13)

$$\begin{aligned} &E_{(x,y)\sim\mu}L_1(\hat{y},y) - \frac{1}{Z}\sum_{i=1}^{Z}L_1(\hat{y}_i,y_i) \\ &\leq \beta + (2Z_1\beta + \mathcal{M})\sqrt{\frac{log(1/\lambda)}{2Z_1}} \\ &= \frac{2max(Z_2,Z_3)b_1\delta^2\tau^2B_{w_1}^2}{min(b_2,b_3)Z_1cB_{w_2}^2} + (\frac{4max(Z_2,Z_3)b_1\delta^2\tau^2B_{w_1}^2}{min(b_2,b_3)cB_{w_2}^2} \\ &+ \mathcal{M})\sqrt{\frac{log(1/\lambda)}{2Z_1}}. \end{aligned}$$

II Symbol Table

Table 1 summarizes the major notations in this paper.

Table 1: Symbol Table

Symbol	Explanation
X	multivariate time series
$\hat{X}$	reconstructed multivariate time series
X^{train}	the training set of X
$\mathcal{X}$	the series prediction model input
T	the total time stpes of series
N	the total dimensions of series
$\boldsymbol{x}_t$	the t-th value of series X
G	the graph of series sensors
V	the set of nodes in G
E	the set of edges in G
A	the adjacency matrix of G
A^{global}	the generated global matrix
$A^{dynamic}$	the generated dynamic matrix
f	a mapping function
H	the historical time steps of series
$H^{'}$	the future time steps of series
L	the Laplace matrix of A
D	the degree matrix of A
U	the eigenvector matrix of A
Λ	the eigenvector value of A
l	the layer of model
H^l	the output of l-layer model
W^l	the parametric matrix of l-layer model
$\mathbf{h}$	the output of model encoder
g_{θ_1}	the encoder of model
g_{θ_2}	the decoder of model
d	the hidden dimension of embedding
θ_{ij}	the success probability of matrix A_{ij}
g_{ij}	the output of Gumbel distribution
s	the annealing factor
seq	the sliding window size
α	the coefficient of graph confusion block
b	the weight of loss functions
Z	the sequence length of tasks
$\mathcal{T}$	the epoch size
$hx(\cdot)$	the prediction function
$\boldsymbol{R}$	the subset of X
n	rotation speed
Tr	torque
W	power
p	penetration rate
F	total thrust
v_T	advance speed
d	distance

III Model Setting

For the global graph structure generation block, the encoder part uses a 2-layer 1D-convolution network whose output dimension is {8, 16} and the window size is set to 9 with full padding. Batch normalization and the ReLU activation function are followed by two convolutional layers. The input channel of the fully connected layer is dependent on the length of the dataset, and the output channel is set to 100. The decoder part is symmetric. The input and output channels of the fully connected layer are opposite relative to the encoder. The decoder's 2-layer 1D-convolution network has an output dimension of {8, 1}, and the window size is also set to 9 with full padding. Batch normalization and the ReLU activation function are followed by a fully connected layer and the first convolution layer.

For the dynamic graph encoder, the output dimension of the 1D-convolution layer is {16, 1}, and the win-

Table 2: Diebold-Mariano Test comparing MAE and RMSE between the best and the second best model

Datasets	3				6				12			
	MAE		RMSE		MAE		RMSE		MAE		RMSE	
	p	dm	p	dm	p	dm	p	dm	p	dm	p	dm
ECG5000	0.135	1.658	0.117	1.735	0.124	1.684	0.128	1.765	0.077	1.848	0.081	1.821
METR-LA	0.102	1.724	0.109	1.895	0.053	2.166	**0.040**	**2.381**	0.086	1.863	**0.042**	**2.213**
PeMSD4	0.176	1.513	0.124	1.650	0.079	1.908	0.093	1.794	**0.043**	**2.169**	0.076	1.854
PeMSD8	0.114	1.765	0.105	1.701	**0.045**	**2.274**	0.085	1.834	0.056	2.036	0.083	1.879
TBM	0.189	1.542	0.164	1.489	0.095	1.845	**0.041**	**2.311**	0.092	1.873	**0.039**	**2.386**

dow size is set to 3 with zero padding. The decoder part uses a fully connected layer that transforms the num of nodes into series lengths which both depend on datasets. Deconvolution layers have output dimensions of $\{16, batch\_size\}$ using zero padding, and the window size is also set to 3.

IV Diebold-Mariano test

A summary of the Diebold-Mariano test [2] at the $p < 0.05$ level comparing their prediction results between the best and the second best model is shown in Table 2. If the p value is smaller than 0.05 and dm value is larger than 0, we can draw the conclusion that the best model performs better than the second. Combined with previous forecasting metrics, MDG reaches better effects than other baselines.

V Hyperparameter Setting and Model Convergence

Two key hyperparameters are the embedding dimensions of the global graph generation block and the convolution kernel size of both the encoder and decoder. Table 3 shows the effects of different dimensions of node embeddings and the size of the window on PeMSD8 dataset. The two parameters vary in a range of $D \in \{25, 50, 100, 200, 400\}$ and $w \in \{5, 7, 9, 11, 13\}$. In general, to achieve the best performance of the model, we fixed the model hyperparameters D at 100 and w at 9.

Depending on the optimal parameters, we verify the convergence of the model. Multi-task losses of the dataset are displayed in Figure 1. As shown in the illustration, three different train losses finally converge to a certain value as the training proceeds, and the test loss of the main prediction task reaches the optimum. The losses all converge rapidly at the beginning and then reach a steady state. Similar trends can be observed from the other datasets. The effectiveness of task-dependent loss is verified by analysis.

Table 3: Hyperparameter Analysis on PeMSD8 Dataset

Setting		RMSE	MAE
D	25	24.423	14.982
	50	23.754	14.671
	100	**23.714**	**14.657**
	200	23.730	14.669
	400	23.746	14.675
w	5	23.732	14.668
	7	23.726	14.603
	9	**23.714**	**14.657**
	11	23.757	14.670
	13	23.915	14.682

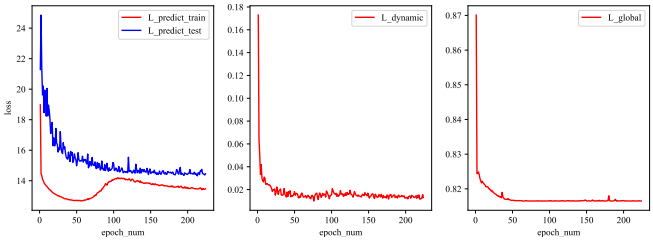

Figure 1: Convergences of the Learning Algorithm on PeMSD8 Dataset

References

[1] L. Le, A. Patterson, and M. White, *Supervised autoencoders: improving generalization performance with unsupervised regularizers*, NIPS, 2018.

[2] D. Harvey, S. Leybourne, and P. Newbold, *Testing the equality of prediction mean squared errors*, International Journal of forecasting, 1997.