Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting

Zheng Fang et al.

KDD `21: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining

Psuedo Lab.

- Season 10 –

AI in Logistics & Transportation

Yungi Kwon (Learner)

2025.04.01

$$\mathcal{H}_{l+1} = \mathcal{H}_l \times$$

Specifically, th

distribution, let's

$$\mathcal{H}_{l+1} = \mathcal{H}_l \times_1 \hat{A} \times_2 U \times_3 W,$$

where the final output will be

$$\mathcal{H}_n = \mathcal{H}_0 \times_1 \hat{A}^n \times_2 U^n \times_3 W^n.$$

$$\begin{aligned}\hat{A}^n &= P\mathrm{diag}\left(\lambda_1^n, \lambda_2^n, \cdots, \lambda_m^n\right) P^T \\ &= \lambda_1^n P\mathrm{diag}\left(1, (\frac{\lambda_2}{\lambda_1})^n, \cdots, (\frac{\lambda_m}{\lambda_1})^n\right) P^T \\ &\longrightarrow \lambda_1^n P\mathrm{diag}\left(1, 0, \cdots, 0\right) P^T\end{aligned}$$

7.1 The calculation of the integration in Eq 19

PROOF. Suppose $\hat{A} - I, U - I, W - I$ have eigenvalue decompositions $P_1\Lambda_1P_1^{-1}, P_2\Lambda_2P_2^{-1}, P_3\Lambda_3P_3^{-1}$ respectively, then we have

$$\begin{aligned}&\int_0^t \mathcal{H}_0 \times_1 e^{(\hat{A}-I)(t-\tau)} \times_2 e^{(U-I)(t-\tau)} \times_3 e^{(W-I)(t-\tau)} \mathrm{d}\tau \\ =& \int_0^t \mathcal{H}_0 \times_1 P_1 e^{\Lambda_1(t-\tau)} P_1^{-1} \times_2 P_2 e^{\Lambda_2(t-\tau)} P_2^{-1} \times_3 P_3 e^{\Lambda_3(t-\tau)} P_3^{-1} \mathrm{d}\tau \\ =& \int_0^t \mathcal{H}_0 \times_1 P_1 \times_2 P_2 \times_3 P_3 \times_1 e^{\Lambda_1(t-\tau)} \times_2 e^{\Lambda_2(t-\tau)} \\ & \times_3 P_3 e^{\Lambda_3(t-\tau)} \times_1 P_1^{-1} \times_2 P_2^{-1} \times_3 P_3^{-1} \mathrm{d}\tau,\end{aligned}$$

denote $\tilde{\mathcal{H}}_0 = \mathcal{H}_0 \times_1 P_1 \times_2 P_2 \times_3 P_3$,

$$\begin{aligned}&\int_0^t \mathcal{H}_0 \times_1 e^{(\hat{A}-I)(t-\tau)} \times_2 e^{(U-I)(t-\tau)} \times_3 e^{(W-I)(t-\tau)} \mathrm{d}\tau \\ =& \int_0^t \tilde{\mathcal{H}}_0 \times_1 e^{\Lambda_1(t-\tau)} \times_2 e^{\Lambda_2(t-\tau)} \times_3 e^{\Lambda_3(t-\tau)} \mathrm{d}\tau \times_1 P_1^{-1} \times_2 P_2^{-1} \times_3 P_3^{-1},\end{aligned}$$

$$\begin{aligned}&\left(\int_0^t \tilde{\mathcal{H}}_0 \times_1 e^{\Lambda_1(t-\tau)} \times_2 e^{\Lambda_2(t-\tau)} \times_3 e^{\Lambda_3(t-\tau)} \mathrm{d}\tau\right)_{ijk} \\ =& \int_0^t \tilde{\mathcal{H}}_{0ijk} \times_1 e^{\Lambda_{1ii}(t-\tau)} \times_2 e^{\Lambda_{2jj}(t-\tau)} \times_3 e^{\Lambda_{3kk}(t-\tau)} \mathrm{d}\tau \\ =& -\frac{1}{\Lambda_{1ii} + \Lambda_{2jj} + \Lambda_{3kk}} \tilde{\mathcal{H}}_{0ijk} \times_1 e^{\Lambda_{1ii}(t-\tau)} \times_2 e^{\Lambda_{2jj}(t-\tau)} \times_3 e^{\Lambda_{3kk}(t-\tau)} \Big|_0^t \\ =& \frac{\tilde{\mathcal{H}}_{0ijk}}{\Lambda_{1ii} + \Lambda_{2jj} + \Lambda_{3kk}} \times_1 e^{\Lambda_{1ii}t} \times_2 e^{\Lambda_{2jj}t} \times_3 e^{\Lambda_{3kk}t} - \frac{\tilde{\mathcal{H}}_{0ijk}}{\Lambda_{1ii} + \Lambda_{2jj} + \Lambda_{3kk}}\end{aligned}$$

thus, the result of the integration is as the following,

$$\begin{aligned}&\int_0^t \mathcal{H}_0 \times_1 e^{(\hat{A}-I)(t-\tau)} \times_2 e^{(U-I)(t-\tau)} \times_3 e^{(W-I)(t-\tau)} \mathrm{d}\tau \\ =& \left(\frac{\tilde{\mathcal{H}}_{0ijk}}{\Lambda_{1ii} + \Lambda_{2jj} + \Lambda_{3kk}} \times_1 e^{\Lambda_{1ii}t} \times_2 e^{\Lambda_{2jj}t} \times_3 e^{\Lambda_{3kk}t} - \frac{\tilde{\mathcal{H}}_{0ijk}}{\Lambda_{1ii} + \Lambda_{2jj} + \Lambda_{3kk}}\right) \\ & \times_1 P_1^{-1} \times_2 P_2^{-1} \times_3 P_3^{-1}.\end{aligned}$$

□

we replace n with a continuous variable i, and view the expansion equation as a Riemann sum from 0 to n on i, which is,

$$\hat{A}^i \times_2 U^i \times_3 W^i\Big) \qquad (11)$$

$$\hat{A}^{(i-1)\times\Delta t} \times_2 U^{(i-1)\times\Delta t} \times_3 W^{(i-1)\times\Delta t} \Delta t\Big)$$

h $t = n$. When n goes to ∞, we can formulate

al:

$$\int_0^{t+1} \mathcal{H}_0 \times_1 \hat{A}^\tau \times_2 U^\tau \times_3 W^\tau \mathrm{d}\tau, \qquad (12)$$

GG...

Corollary 1 The discrete update in Eq 8 is a discretization of following ODE:

$$\frac{\mathrm{d}\mathcal{H}(t)}{\mathrm{d}t} = \mathcal{H}(t) \times_1 \ln \hat{A} + \mathcal{H}(t) \times_2 \ln U + \mathcal{H}(t) \times_3 \ln W + \mathcal{H}_0, \quad (14)$$

where $\mathcal{H}_0 = f(\mathcal{X})$ is the output of upstream networks.

PROOF. Starting from Eq 13, we consider the second-derivative of $\mathcal{H}(t)$ through derivative rules,

$$\frac{\mathrm{d}^2\mathcal{H}(t)}{\mathrm{d}t^2} = \frac{\mathrm{d}\mathcal{H}(t)}{\mathrm{d}t} \times_1 \ln \hat{A} + \frac{\mathrm{d}\mathcal{H}(t)}{\mathrm{d}t} \times_2 \ln U + \frac{\mathrm{d}\mathcal{H}(t)}{\mathrm{d}t} \times_3 \ln W \quad (15)$$

$$\begin{aligned}const =& \mathcal{H}_0 \times_1 \hat{A}^{t+1} \times_2 U^{t+1} \times_3 W^{t+1} \\ & - \left(\mathcal{H}(t) \times_1 \ln \hat{A} + \mathcal{H}(t) \times_2 \ln U + \mathcal{H}(t) \times_3 \ln W\right).\end{aligned}$$

Corollary 2. The analytic solution of the Eq 18 is given by

$$\begin{aligned}\mathcal{H}(t) =& \mathcal{H}_0 \times_1 e^{(\hat{A}-I)t} \times_2 e^{(U-I)t} \times_3 e^{(W-I)t} \\ & + \int_0^t \mathcal{H}_0 \times_1 e^{(\hat{A}-I)(t-s)} \times_2 e^{(U-I)(t-s)} \times_3 e^{(W-I)(t-s)} \mathrm{d}s\end{aligned} \qquad (19)$$

PROOF. Suppose

$$\mathcal{H}^*(t) = \mathcal{H}(t) \times_1 e^{(\hat{A}-I)t} \times_2 e^{(U-I)t} \times_3 e^{(W-I)t}, \qquad (20)$$

then we have

$$\frac{\mathrm{d}\mathcal{H}^*(t)}{\mathrm{d}t} = \mathcal{H}_0 \times_1 e^{(\hat{A}-I)t} \times_2 e^{(U-I)t} \times_3 e^{(W-I)t}, \qquad (21)$$

and this is derived from Eq 18. Integrate Eq 21 on both sides, and lowing result,

$$\int_0^t \mathcal{H}_0 \times_1 e^{(\hat{A}-I)\tau} \times_2 e^{(U-I)\tau} \times_3 e^{(W-I)\tau} \mathrm{d}\tau, \qquad (22)$$

an be formulated as

$$\begin{aligned}\mathcal{H}(t) =& \mathcal{H}_0 \times_1 e^{(\hat{A}-I)t} \times_2 e^{(U-I)t} \times_3 e^{(W-I)t} \\ & + \int_0^t \mathcal{H}_0 \times_1 e^{(\hat{A}-I)(t-\tau)} \times_2 e^{(U-I)(t-\tau)} \times_3 e^{(W-I)(t-\tau)} \mathrm{d}\tau\end{aligned} \qquad (23)$$

□

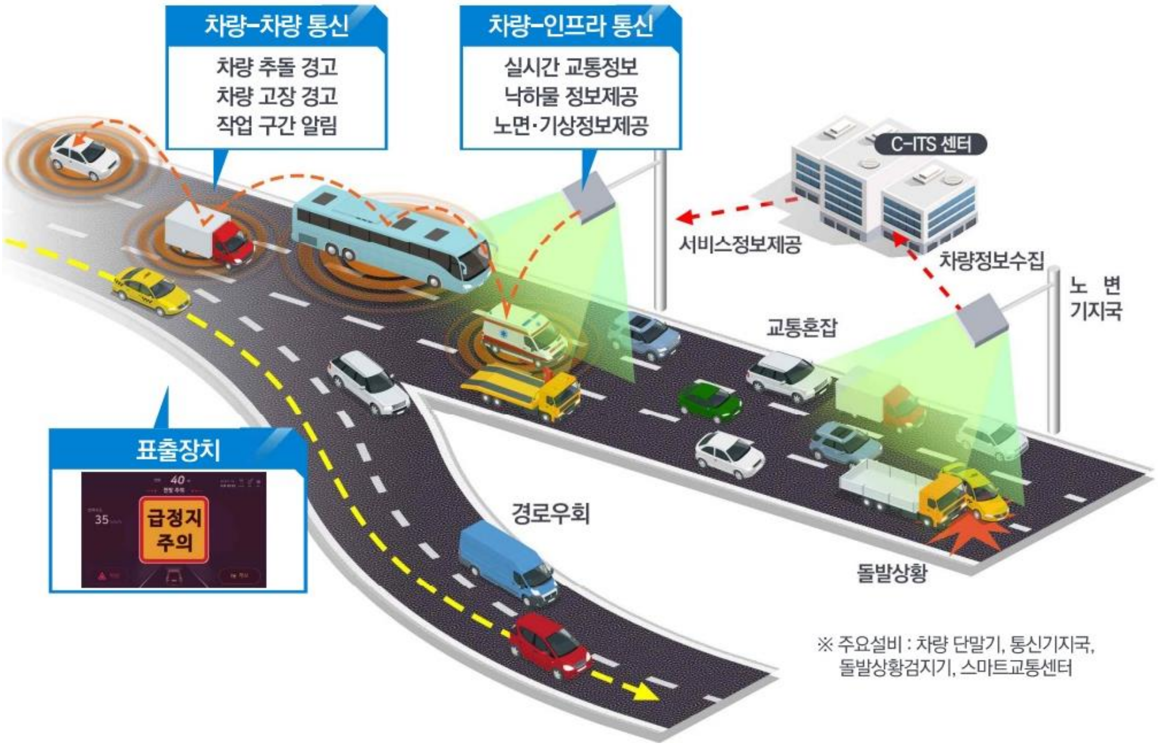
Introduction

Traffic Flow Forecasting

(교통 흐름 예측)

지능형 교통 체계 (ITS)

실시간 수집 + 통신 + **분석/예측** + 제어

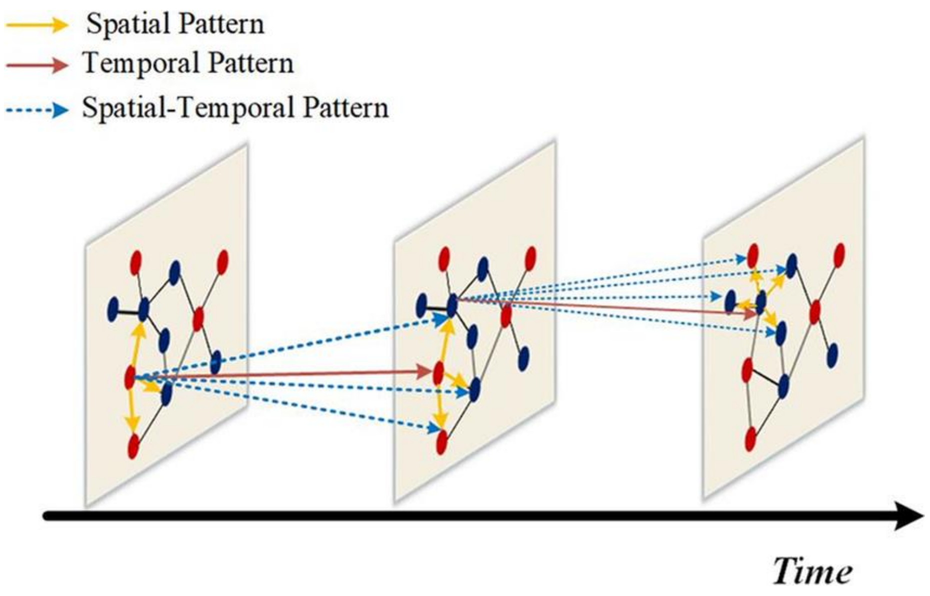


< Real-time Traffic Control >

"10분 뒤 고속도로 진입로에 정체 발생 예정"

☞ 신호등 제어, 경로 우회 (통행량 분산)

How'd we know?



Introduction

Based on
GNN + RNN Architectures

- DCRNN (2017)

Diffusion convolutional recurrent neural network: Data-driven traffic forecasting
Y Li, R Yu, C Shahabi, Y Liu
arXiv preprint arXiv:1707.01926, 2017 · arxiv.org

Spatiotemporal forecasting has various applications in neuroscience, climate and transportation domain. Traffic forecasting is one canonical example of such learning task. The task is challenging due to (1) complex spatial dependency on road networks, (2) non-linear temporal dynamics with changing road conditions and (3) inherent difficulty of long-term forecasting. To address these challenges, we propose to model the traffic flow as a diffusion process on a directed graph and introduce Diffusion Convolutional Recurrent

자세히 보기

☆ 저장 99 인용 4492회 인용 관련 학술자료 전체 10개의 버전

- STGCN (2017)

Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting
B Yu, H Yin, Z Zhu
arXiv preprint arXiv:1709.04875, 2017 · arxiv.org

Timely accurate traffic forecast is crucial for urban traffic control and guidance. Due to the high nonlinearity and complexity of traffic flow, traditional methods cannot satisfy the requirements of mid-and-long term prediction tasks and often neglect spatial and temporal dependencies. In this paper, we propose a novel deep learning framework, Spatio-Temporal Graph Convolutional Networks (STGCN), to tackle the time series prediction problem in traffic domain. Instead of applying regular convolutional and recurrent units, we

자세히 보기

☆ 저장 99 인용 4904회 인용 관련 학술자료 전체 11개의 버전

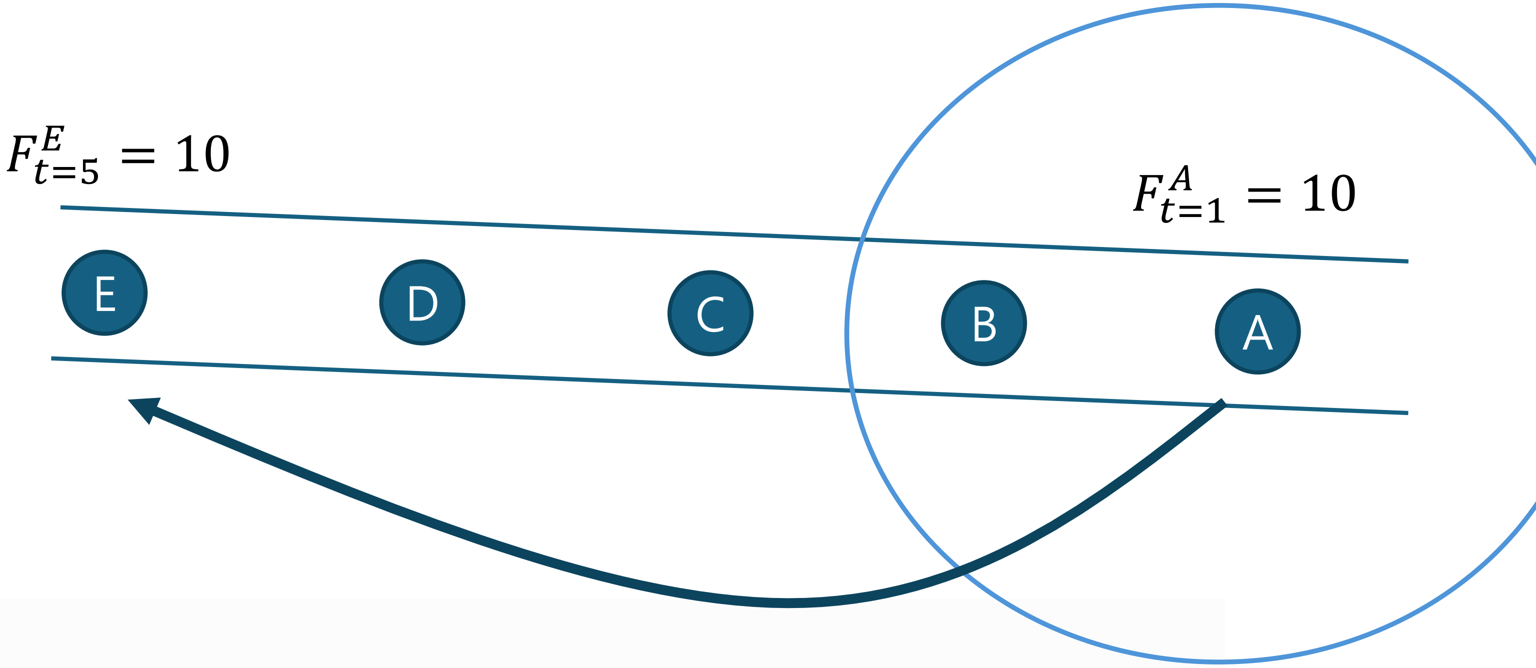
- STGAT (2019)

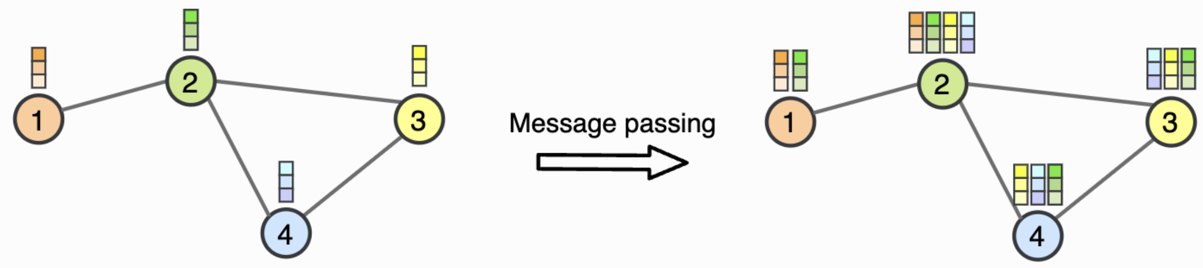
Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting
C Zhang, JQ James, Y Liu
Ieee Access, 2019 · ieeexplore.ieee.org

Traffic speed prediction, as one of the most important topics in Intelligent Transport Systems (ITS), has been investigated thoroughly in the literature. Nonetheless, traditional methods show their limitation in coping with complexity and high nonlinearity of traffic data as well as learning spatial-temporal dependencies. Particularly, they often neglect the dynamics happening to traffic network. Attention-based models witnessed extensive developments in recent years and have shown its efficacy in a host of fields, which

자세히 보기

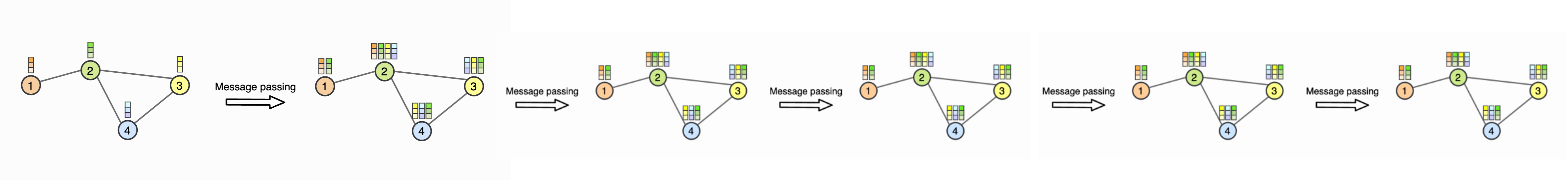
☆ 저장 99 인용 194회 인용 관련 학술자료 전체 4개의 버전



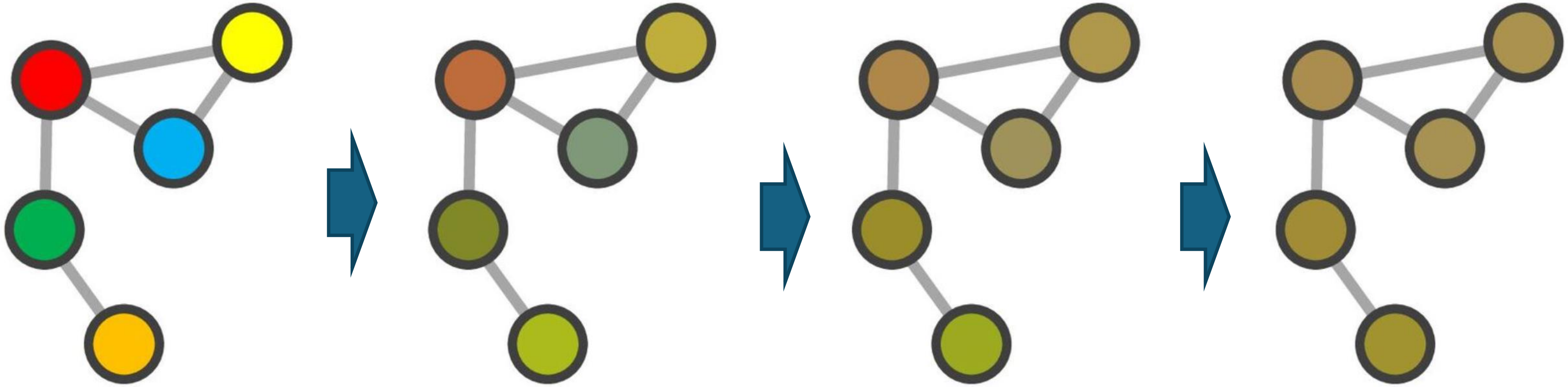


노드의 표현 구성이 지리적으로 가까운 노드들의 정보를 참조하는 수준에 그쳤음 (Geographical-based spatial correlations)

Introduction



However, too many convolutions causes over smoothing —
all node embeddings converge to the same value

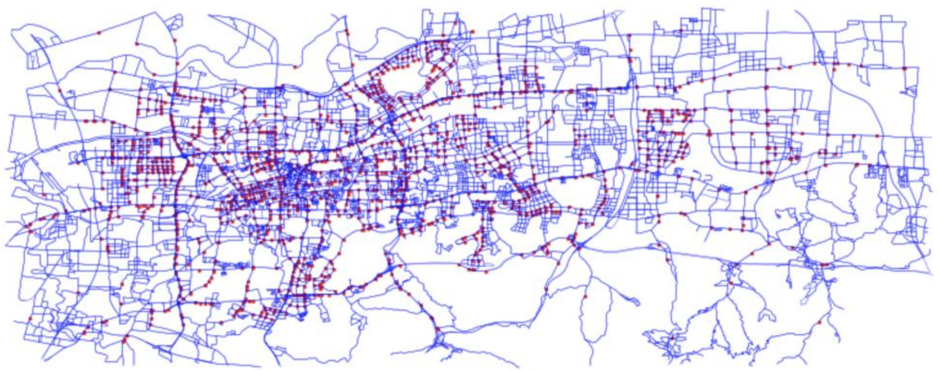


Preliminaries: Definitions

Traffic Network $\mathcal{G} = (V, E, A)$

$A \in \mathbb{R}^{N \times N}$ → Spatial Adj. Matrix A^{sp}

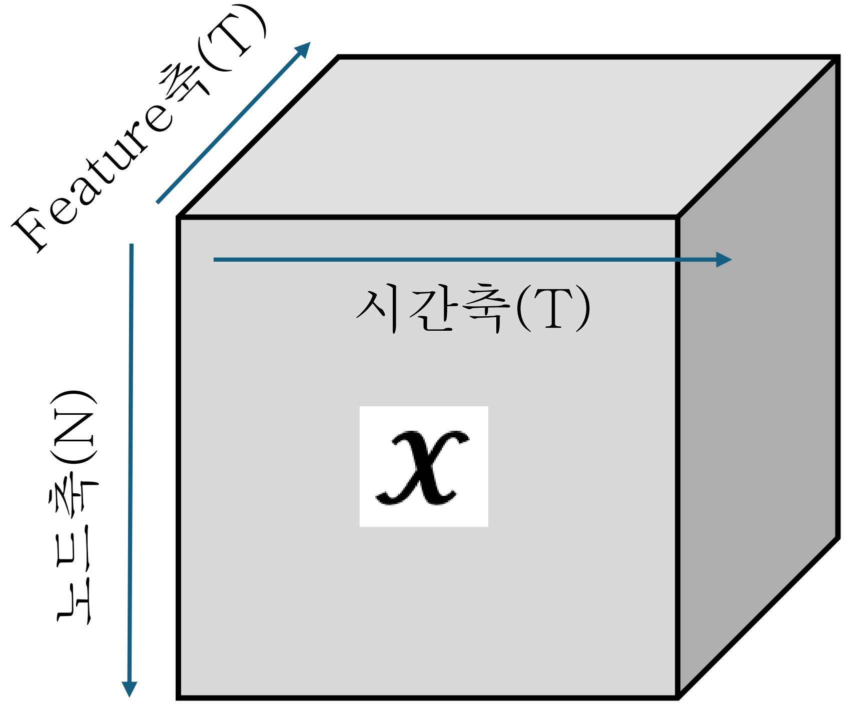
$A \in \mathbb{R}^{N \times N}$ → Semantic Adj. Matrix A^{se}



Road network of Jinan, China (Red nodes: road intersections having camera)

Graph Signal Tensor $\mathcal{X}$

Definition 2. (Graph signal tensor $\mathcal{X}$) We use $\mathbf{x}_t^i \in \mathbb{R}^F$ to denote the observation of node i at time t, and F is the length of an observation vector. $X_t = \left(\mathbf{x}_t^1, \mathbf{x}_t^2, \cdots, \mathbf{x}_t^N\right) \in \mathbb{R}^{N \times F}$ denotes the observations of all nodes at time t. $\mathcal{X} = (X_1, X_2, \cdots, X_T) \in \mathbb{R}^{T \times N \times F}$ denotes the observations of all nodes at all time.

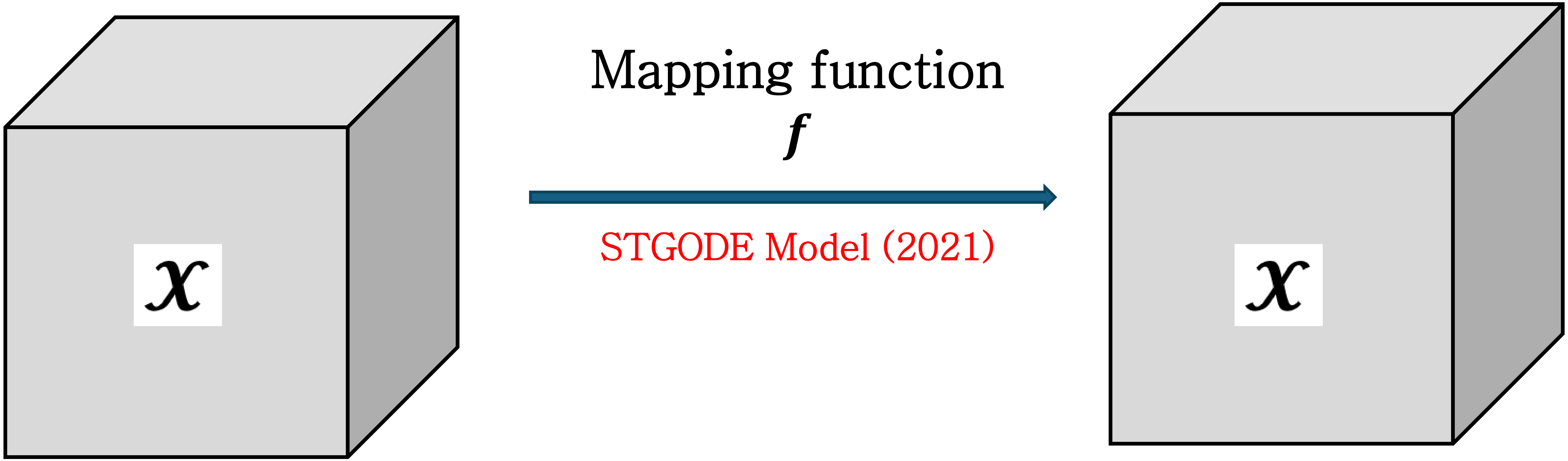


Problem Formulation

$$[X_{t-T+1}, X_{t-T+2}, \cdots, X_t; \mathcal{G}] \xrightarrow{f} [X_{t+1}, X_{t+2}, \cdots, X_{t+T'}] \, .$$

과거부터 현재까지의
T개 교통흐름을 바탕으로

미래 시점의
T’개 교통흐름을 예측



Regularized Adjacency Matrix

Kipf & Welling (2017): Regularized Adj. Matrix

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, \quad \text{where } \tilde{A} = A + I$$

Normalized itself with neighbors

➢ Eigenvalue ∈ [0, 1]

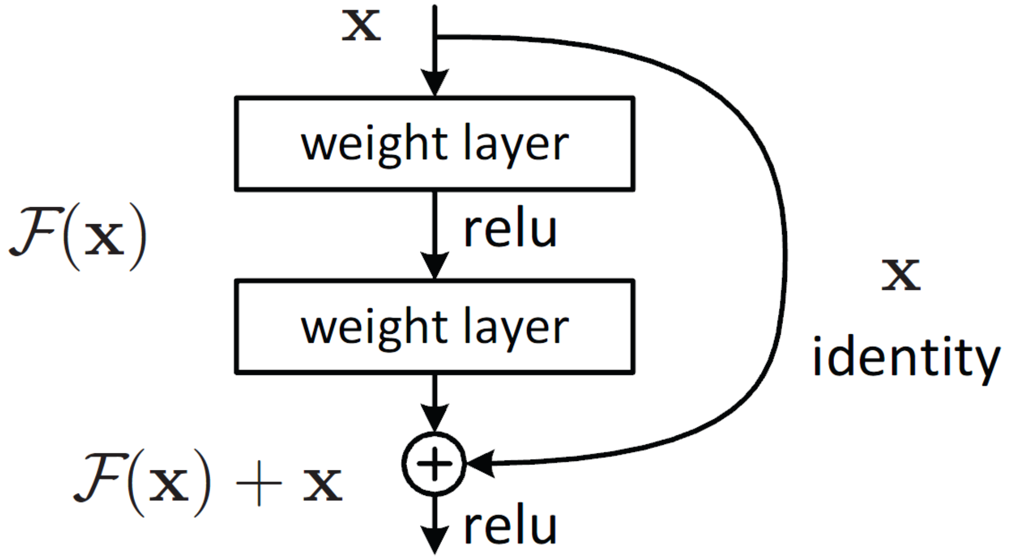
But, on this paper (STGODE),

$$\hat{A} = \frac{\alpha}{2}\left(I + \underline{D^{-\frac{1}{2}}AD^{-\frac{1}{2}}}\right),$$

Normalized neighbor info.

Sustaining its initial info.

➢ Eigenvalue ∈ [0, α]



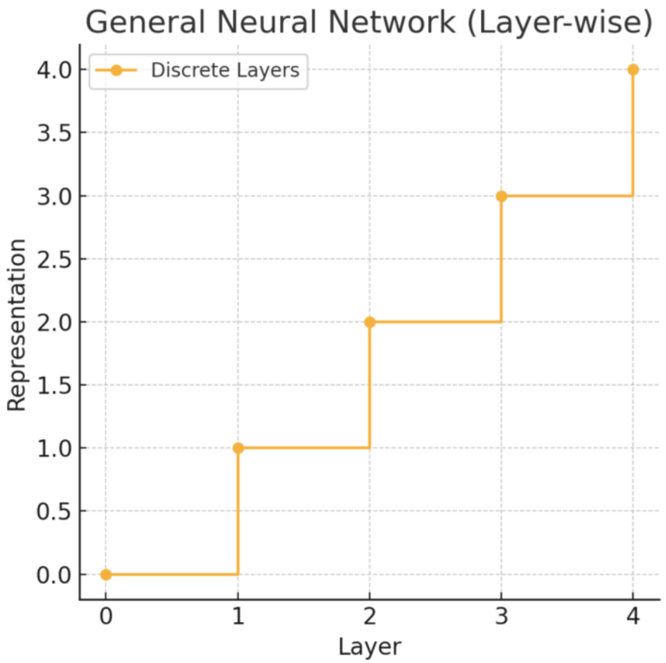
Compatible with
Neural ODE or CGNN

preserving the initial state
while allowing for continuous transformation

Neural ODE (1)

General Neural Networks

$$h^{(l+1)} = \sigma(W^{(l)}h^{(l)} + b^{(l)})$$

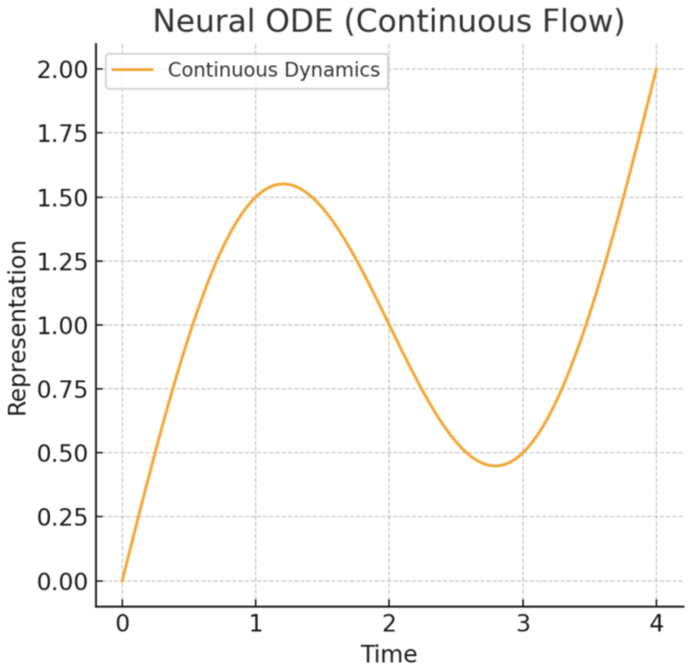


Forwarding

Neural ODE

$$\frac{dh(t)}{dt} = f(h(t), t)$$

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau}\mathrm{d}\tau = \mathbf{x}(0) + \int_0^t f(\mathbf{x}(\tau), \tau)\mathrm{d}\tau,$$



Neural ODE (2)

일반 신경망
(Graph Conv.도 마찬가지)

- Layer를 하나씩 쌓는 방식
- 레이어를 지날 때마다 $x_1, x_2, x_3, \ldots$ 값이 변화

Neural ODE

$$x(t) = x(0) + \int_0^t f(x(\tau), \tau) d\tau$$

- 출력값의 변화를 연속적인 변화로 모델링
(시간에 따라 점점 변하는 상태로 봄)

ODE Solver가 알아서 해줌
(Euler Method, Runge-Kutta 등등 있음)

PyTorch Implementation of Differentiable ODE Solvers

This library provides ordinary differential equation (ODE) solvers implemented in PyTorch. Backpropagation through ODE solutions is supported using the adjoint method for constant memory cost. For usage of ODE solvers in deep learning applications, see reference [1].

As the solvers are implemented in PyTorch, algorithms in this repository are fully supported to run on the GPU.

Installation

To install latest stable version:

```
pip install torchdiffeq
```

$$\frac{df(\tau)}{d\tau} = f(x(\tau), \tau)$$

이건 1차 Taylor 근사한 간소화버전

$$\frac{d\mathcal{H}(t)}{dt} = \mathcal{H}(t) \times_1 (\hat{A} - I) + \mathcal{H}(t) \times_2 (U - I) + \mathcal{H}(t) \times_3 (W - I) + \mathcal{H}_0,$$

- STGODE 모델의 Novelity -

공간적 전파($\times_1 \hat{A}$) + 시간적 전파($\times_2 U$)
+ Feature 전파($\times_3 W$) + Restart($\mathcal{H}_0$) 구조
로 변화율을 풀어낸 것

Two types of adjacency matrix

Spatial (Weighted) Adjacency Matrix A^{sp}

$$A_{ij}^{sp} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) & \text{, if } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \epsilon \\ 0 & \text{, otherwise} \end{cases},$$

$\boldsymbol{d_{ij}}$: 노드 i와 j 사이의 실제 물리적 거리

$\boldsymbol{\sigma^2}$: 거리 스케일을 조절하는 정규화 상수

→ d_{ij}(거리차이)에 따른 두 노드 사이 영향력이 얼마나 느리게 감쇠하는지

$\boldsymbol{\epsilon}$: 연결할지 말지를 결정하는 임계값(threshold)

Bing Yu, Haoteng Yin, Zhanxing Zhu: STGCN (2017)

Semantic Adjacency Matrix A^{SE}

Dynamic Time Warping(DTW) algorithm applied for calculating the similarity of two time-series.

$$A_{ij}^{SE} = \begin{cases} 1, & DTW(X^i, X^j) < \epsilon \\ 0, & \text{otherwise} \end{cases}$$

Two types of adjacency matrix

Semantic Adjacency Matrix A^{SE}

Example)

Node X (교통량)	Node Y (교통량)
$x_1 = 10$	$y_1 = 12$
$x_2 = 20$	$y_2 = 18$
$x_3 = 30$	$y_3 = 28$
$x_4 = 25$	$y_4 = 26$
$x_5 = 15$	$y_5 = 14$

$dist(X, Y)$ matrix

☞ Absolute distance or other distance metric can be used

	$y_1 = 12$	$y_2 = 18$	$y_3 = 28$	$y_4 = 26$	$y_5 = 14$
$x_1 = 10$	2	8	18	16	4
$x_2 = 20$	8	2	8	6	6
$x_3 = 30$	18	12	2	4	16
$x_4 = 25$	13	7	3	1	11
$x_5 = 15$	3	3	13	11	1

Two types of adjacency matrix

$dist(X, Y)$ matrix

	y_1	y_2	y_3	y_4	y_5
x_1	2	8	18	16	4
x_2	8	2	8	6	6
x_3	18	12	2	4	16
x_4	13	7	3	1	11
x_5	3	3	13	11	1

$$D(i, j) = dist(x_i, y_j) + \min\left(\overset{\rightarrow}{D(i-1, j)}, \overset{\downarrow}{D(i, j-1)}, \overset{\searrow}{D(i-1, j-1)}\right),$$

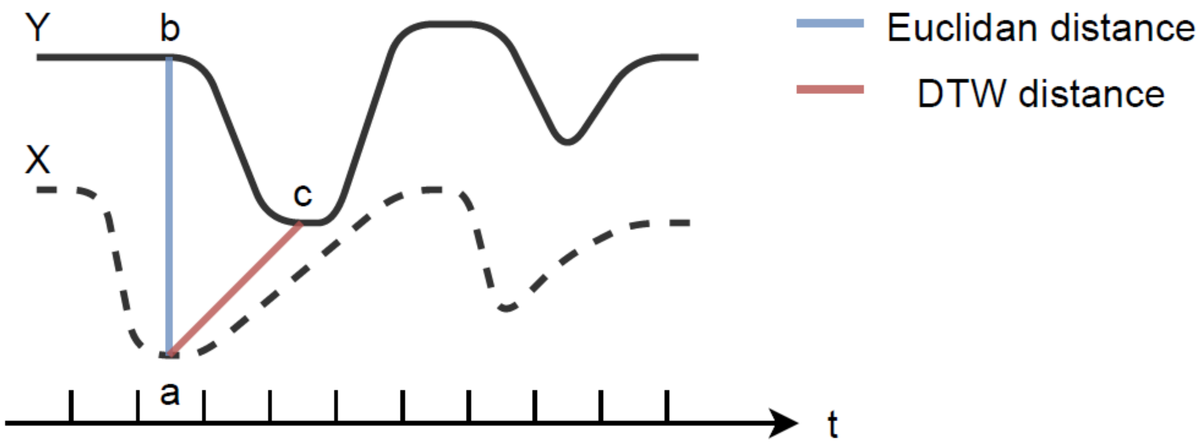
$D(i,j)$	1	2	3	4	5
1	**2 (Start)**	10	28	44	48
2	10	4	12	18	24
3	28	16	6	10	26
4	41	23	9	7	18
5	44	26	22	18	**8 (End)**

DTW(X, Y) = D(m, n) = D(5, 5) = 8

Two types of adjacency matrix

Semantic Adjacency Matrix A^{SE}

$$A_{ij}^{SE} = \begin{cases} 1, DTW(X^i, X^j) < \epsilon \\ 0, \text{otherwise} \end{cases}$$



DTW를 통해 전체적인 교통 흐름 패턴이 비슷한 노드, 멀지만 의미론적으로 유사한 노드를 식별할 수 있고, 이들의 연결 관계를 이용해 GNN 표현력을 향상시킨다.

Tensor-based Spatial-Temporal Graph ODE

기본 GCN 수식

$$H_{l+1} = \sigma(\hat{A} H_l W)$$

Layer가 깊어질수록 Over-smoothing

STGODE Block 구조 (Eq. 8)

$$H_{l+1} = H_l \times_1 \hat{A} \times_2 U \times_3 W + H_0$$

3모드 연산을 통해 공간/시간/피처를 각각 처리

H0는 초기값(restart-term) – OverSmoothing 방지

연속적인 변화로 확장 (Eq. 12)

$$H(t) = \int_0^{t+1} H_0 \times_1 \hat{A}^\tau \times_2 U^\tau \times_3 W^\tau d\tau$$

Eq. 8 Discrete Summation 형태를 연속시간 변수 형태로 확장

이 때 테일러 근사하면
슬라이드 11에서 나온 미방

여긴 Exact Solution을 얻기 위한 여정

$$\frac{d\mathcal{H}(t)}{dt} = \mathcal{H}(t) \times_1 \ln \hat{A} + \mathcal{H}(t) \times_2 \ln U + \mathcal{H}(t) \times_3 \ln W + \mathcal{H}_0$$

(Eq. 14)

$$\mathcal{H}(t) = \mathcal{H}_0 \times_1 e^{(\hat{A}-I)t} \times_2 e^{(U-I)t} \times_3 e^{(W-I)t} + \int_0^t \mathcal{H}_0 \times_1 e^{(\hat{A}-I)(t-\tau)} \times_2 e^{(U-I)(t-\tau)} \times_3 e^{(W-I)(t-\tau)} d\tau \quad (23)$$

Temporal Convolutional (Network) Blocks

4.3 Temporal Convolutional Blocks

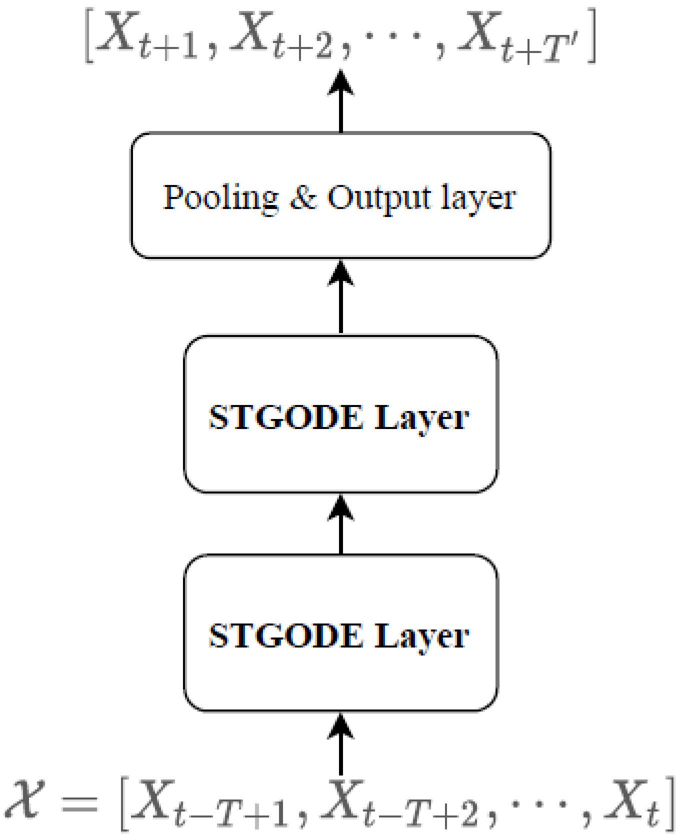
Besides spatial correlations among different nodes, the long-term temporal correlations of the nodes themselves also matter. Although RNN-based models, like LSTM and GRU, are widely applied in time-series analysis, recurrent networks still suffer from some intrinsic drawbacks like time-consuming iterations, unstable gradients, and delayed responses to dynamic changes.

To enhance the performance of extracting long term temporal dependencies, a 1-D dilated temporal convolutional network along the time axis is adopted here.

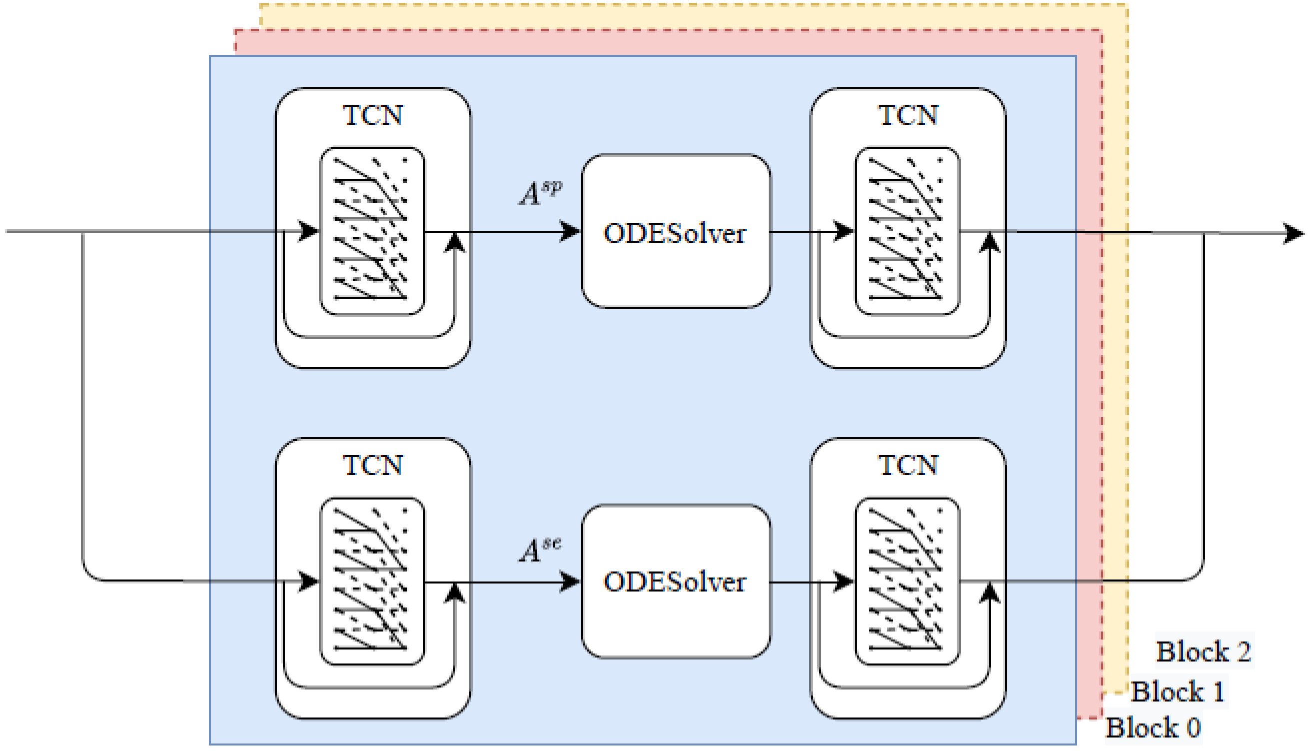
$$H_{tcn}^{l} = \begin{cases} X & , l = 0 \\ \sigma(W^{l} *_{d^{l}} H_{tcn}^{l-1}) & , l = 1, 2, \cdots, L \end{cases} \tag{25}$$

where $X \in \mathbb{R}^{N \times T \times F}$ is the input of TCN, $H_{tcn}^{l} \in \mathbb{R}^{N \times T \times F}$ is the output of the l-th layer of TCN, and W^{l} denotes the l-th convolution kernel. To expand the receptive field , an exponential dilation rate $d^{l} = 2^{l-1}$ is adopted in temporal convolution. In the process, zero-padding strategy is utilized to keep time sequence length unchanged. What's more, a residual structure [12] is added to strengthen convolution performance as shown in Fig 5(b).

STGODE Layer & Others

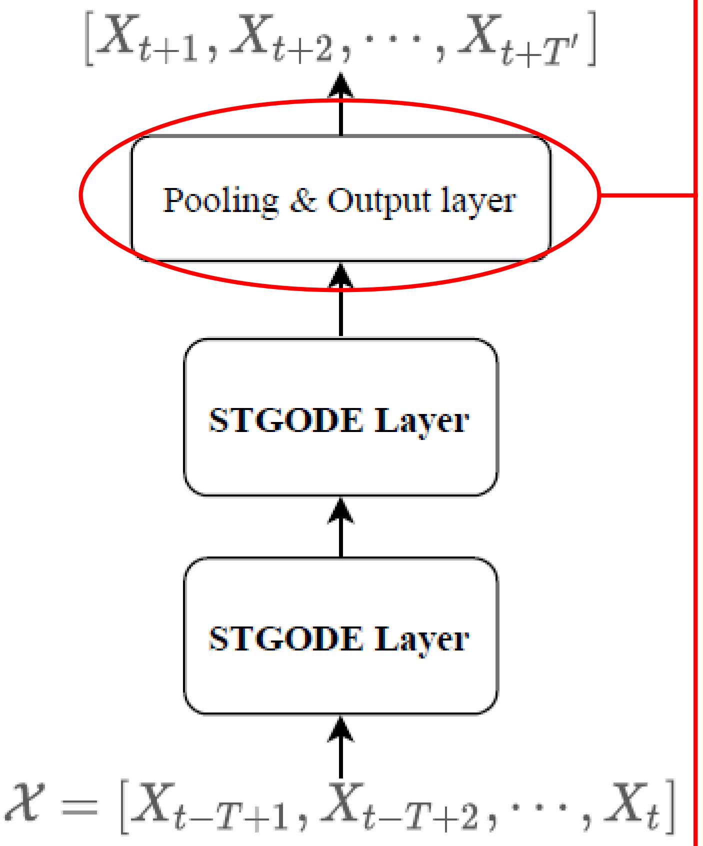


(a) Framework



(b) STGODE Layer

STGODE Layer & Others



(a) Framework

4.5 Others

After the STGODE layers, a max-pooling operation is carried out to aggregate information from different blocks selectively. Finally, a two-layer MLP is designed as the output layer to transform the output of the max-pooling layer to the final prediction.

Huber loss is selected as the loss function since it is less sensitive to outliers than the squared error loss [13],

$$L(Y, \hat{Y}) = \begin{cases} \frac{1}{2}(Y - \hat{Y})^2 & , |Y - \hat{Y}| \leq \delta \\ \delta|Y - \hat{Y}| - \frac{1}{2}\delta^2 & , \text{otherwise} \end{cases} \tag{26}$$

where δ is a hyperparameter which controls the sensitivity to outliers.

Experiments: Experimental Settings for STGODE

Training set (60%)	Valid (20%)	Test (20%)

Purpose: One hour of historical data is inputted, and we predict traffic cond. in the next 60 minutes.

Regularized adjacent matrix hyperparameter $\boldsymbol{\alpha} = 0.8$

$$\hat{A} = \frac{\alpha}{2}\left(I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right),$$

Spatial weighted adjacency matrix $\boldsymbol{\sigma = 10,\ \epsilon = 0.5}$

$$A_{ij}^{sp} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) & \text{, if } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \epsilon \\ 0 & \text{, otherwise} \end{cases},$$

Semantic adjacency matrix $\boldsymbol{\epsilon = 0.6}$

$$A_{ij}^{SE} = \begin{cases} 1, DTW(X^i, X^j) < \epsilon \\ 0, \text{otherwise} \end{cases}$$

And **Adam Optimizer & learning rate 0.01**

, We use three kinds of evaluation metrics

RMS, MAE, MAPE

Experiments: Baselines

Baseline Models

1) ARIMA: (1970)

Distribution of residual autocorrelations in autoregressive-integrated moving average time series models
GEP Box, DA Pierce - Journal of the American statistical …, 1970 - Taylor & Francis
Many statistical models, and in particular autoregressive—moving average time series models, can be regarded as means of transforming the data to white noise, that is, to an uncorrected sequence of errors. If the parameters are known exactly, this random sequence can be computed directly from the observations; when this calculation is made with estimates substituted for the true parameter values, the resulting sequence is referred to as the "residuals," which can be regarded as estimates of the errors. If the appropriate model …
☆ 저장 99 인용 4796회 인용 관련 학술자료 전체 6개의 버전

2) STGCN: (2017)

Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting
B Yu, H Yin, Z Zhu
arXiv preprint arXiv:1709.04875, 2017 · arxiv.org
Timely accurate traffic forecast is crucial for urban traffic control and guidance. Due to the high nonlinearity and complexity of traffic flow, traditional methods cannot satisfy the requirements of mid-and-long term prediction tasks and often neglect spatial and temporal dependencies. In this paper, we propose a novel deep learning framework, Spatio-Temporal Graph Convolutional Networks (STGCN), to tackle the time series prediction problem in traffic domain. Instead of applying regular convolutional and recurrent units, we
자세히 보기 ⌄
☆ 저장 99 인용 4904회 인용 관련 학술자료 전체 11개의 버전 »

3) DCRNN: (2017)

Diffusion convolutional recurrent neural network: Data-driven traffic forecasting
Y Li, R Yu, C Shahabi, Y Liu
arXiv preprint arXiv:1707.01926, 2017 · arxiv.org
Spatiotemporal forecasting has various applications in neuroscience, climate and transportation domain. Traffic forecasting is one canonical example of such learning task. The task is challenging due to (1) complex spatial dependency on road networks, (2) non-linear temporal dynamics with changing road conditions and (3) inherent difficulty of long-term forecasting. To address these challenges, we propose to model the traffic flow as a diffusion process on a directed graph and introduce Diffusion Convolutional Recurrent
자세히 보기 ⌄
☆ 저장 99 인용 4492회 인용 관련 학술자료 전체 10개의 버전 »

4) GraphWaveNet: (2019)

Graph wavenet for deep spatial-temporal graph modeling
Z Wu, S Pan, G Long, J Jiang, C Zhang - arXiv preprint arXiv:1906.00121, 2019 - arxiv.org
Spatial-temporal graph modeling is an important task to analyze the spatial relations and temporal trends of components in a system. Existing approaches mostly capture the spatial dependency on a fixed graph structure, assuming that the underlying relation between entities is pre-determined. However, the explicit graph structure (relation) does not necessarily reflect the true dependency and genuine relation may be missing due to the incomplete connections in the data. Furthermore, existing methods are ineffective to capture …
☆ 저장 99 인용 2627회 인용 관련 학술자료 전체 11개의 버전 »

5) ASTGCN(r): (2019)

Attention based spatial-temporal graph convolutional networks for traffic flow forecasting
S Guo, Y Lin, N Feng, C Song, H Wan - Proceedings of the AAAI …, 2019 - ojs.aaai.org
Forecasting the traffic flows is a critical issue for researchers and practitioners in the field of transportation. However, it is very challenging since the traffic flows usually show high nonlinearities and complex patterns. Most existing traffic flow prediction methods, lacking abilities of modeling the dynamic spatial-temporal correlations of traffic data, thus cannot yield satisfactory prediction results. In this paper, we propose a novel attention based spatial-temporal graph convolutional network (ASTGCN) model to solve traffic flow forecasting …
☆ 저장 99 인용 2821회 인용 관련 학술자료 전체 10개의 버전 »

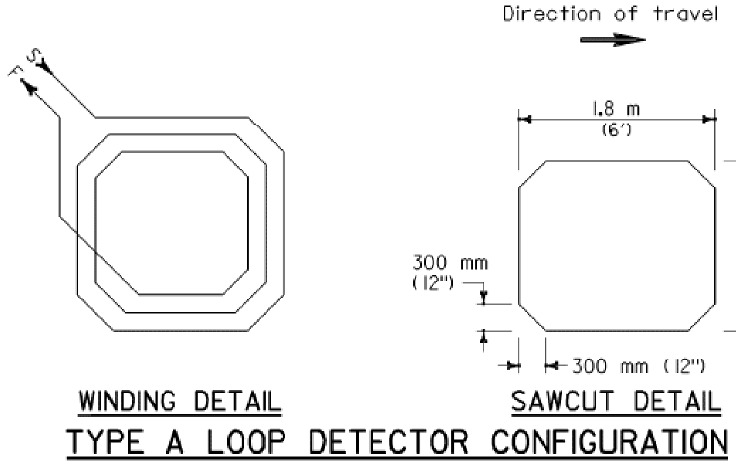
6) STSGCN: (2020)

Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting
C Song, Y Lin, S Guo, H Wan - Proceedings of the AAAI conference on …, 2020 - ojs.aaai.org
Spatial-temporal network data forecasting is of great importance in a huge amount of applications for traffic management and urban planning. However, the underlying complex spatial-temporal correlations and heterogeneities make this problem challenging. Existing methods usually use separate components to capture spatial and temporal correlations and ignore the heterogeneities in spatial-temporal data. In this paper, we propose a novel model, named Spatial-Temporal Synchronous Graph Convolutional Networks (STSGCN), for spatial …
☆ 저장 99 인용 1395회 인용 관련 학술자료 전체 6개의 버전 »

Experiments: Datasets

PeMS Datasets

California Dept. of Transportation (Caltrans)



Datasets	#Sensors	#Edges	Time Steps
PeMSD7(M)	228	1132	12672
PeMSD7(L)	1026	10150	12672
PeMS03	358	547	26208
PeMS04	307	340	16992
PeMS07	883	866	28224
PeMS08	170	295	17856

Table 1: Datasets description

Traffic Flow(교통량; 차량대수) **&**
Avg. Speed & Avg. Occupancy(평균점유시간)

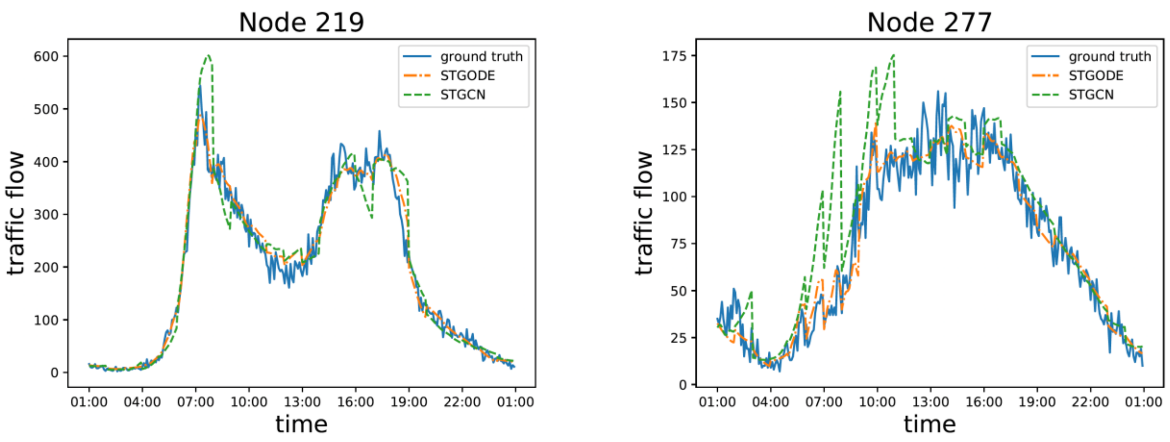
30 seconds resolutions
=> Aggregated into 5-mins intervals

\+ Z-Score Normalization

Experiments: Results and Analysis

Dataset	Metric	ARIMA	STGCN	DCRNN	ASTGCN(r)	GraphWaveNet	STSGCN	STODE
PeMSD7(M)	RMSE	13.20	7.55	7.18	6.87	6.24	5.93	**5.66**
	MAE	7.27	4.01	3.83	3.61	3.19	3.01	**2.97**
	MAPE	10.38	9.67	9.81	8.84	8.02	7.55	**7.36**
PeMSD7(L)	RMSE	12.39	8.28	8.33	7.64	7.09	6.88	**5.98**
	MAE	7.51	4.84	4.33	4.09	3.75	3.61	**3.22**
	MAPE	15.83	11.76	11.41	10.25	9.41	9.13	**7.94**
PeMS03	RMSE	47.59	30.42	30.31	29.56	32.77	29.21	**27.84**
	MAE	35.41	17.55	17.99	17.34	19.12	17.48	**16.50**
	MAPE	33.78	17.43	18.34	17.21	18.89	16.78	**16.69**
PeMS04	RMSE	48.80	36.01	37.65	35.22	39.66	33.65	**32.82**
	MAE	33.73	22.66	24.63	22.94	24.89	21.19	**20.84**
	MAPE	24.18	14.34	17.01	16.43	17.29	13.90	**13.77**
PeMS07	RMSE	59.27	39.34	38.61	37.87	41.50	39.03	**37.54**
	MAE	38.17	25.33	25.22	24.01	26.39	24.26	**22.99**
	MAPE	19.46	11.21	11.82	10.73	11.97	10.21	**10.14**
PeMS08	RMSE	44.32	27.88	27.83	26.22	30.04	26.80	**25.97**
	MAE	31.09	18.11	17.46	16.64	18.28	17.13	**16.81**
	MAPE	22.73	11.34	11.39	10.6	12.15	10.96	**10.62**

Table 2: Performance comparison of baseline models and STGODE on PeMS datasets.



↑ Case Study:
임의의 2개 노드 선정 후
실제 교통흐름 예측을 STGCN 모델과 비교

1) Our ODE framework is allowed to extract longer-range spatial-temporal dependencies.

2) Semantical neighbours allowed us to establish global & comprehensive spatial relationship.

3) 1D dilated conv. networks with residual help to capture long-term temporal dependencies.

Ablation Experiments and Parameter Analysis

- Ablation Experiments: 딥러닝 핵심 개념. 매우 중요하며 모델 성능 실험에서 거의 표준이라고 보면 됨.
(Top-tier 논문(NeurIPS, ICML, CVPR, KDD, ICLR 등)에선 거의 필수로 들어가는 파트)

>> 모델의 구성 요소 중 일부를 제거하거나 바꿔서, 그 요소가 전체 성능에 얼마나 기여하는지를 분석하는 실험

1) STGCN*: ODE Solver부분을 GCN Layer로 대체

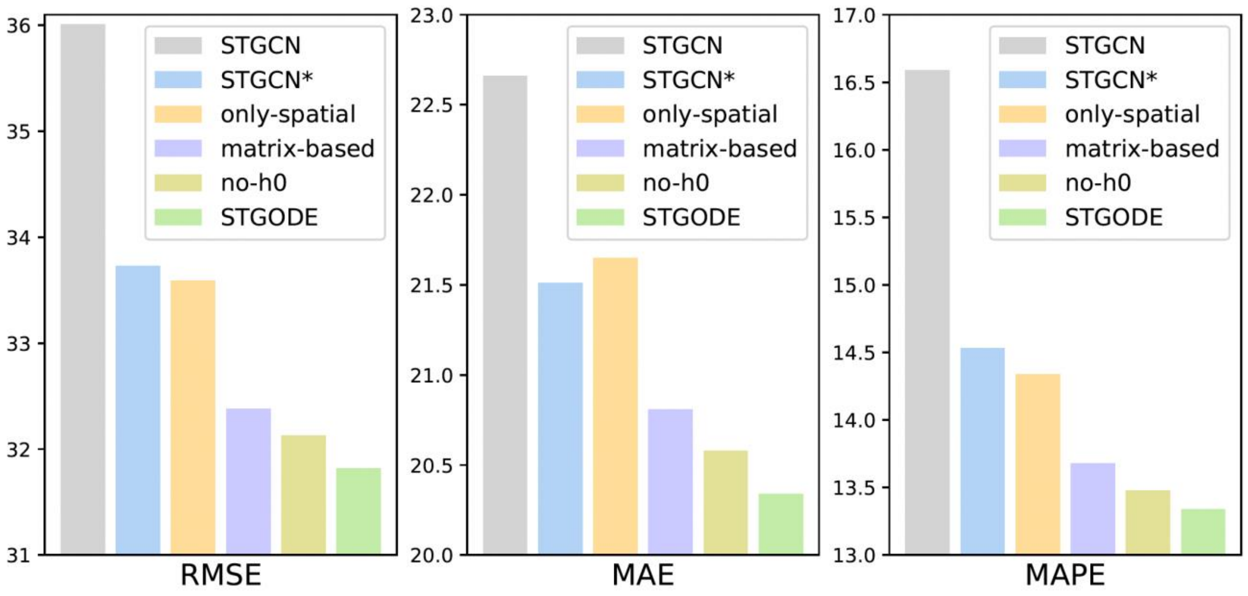
2) Only spatial: Spatial Adj. Matrix A^{SP}만 사용

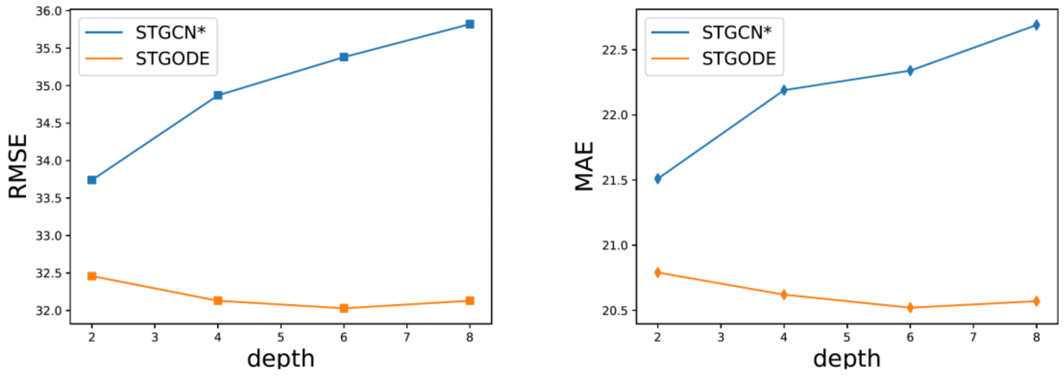
3) no-h0: 초기 상태값(H_0)을 ODE 연산에서 뺌

4) matrix-based:
mode간 곱을 무시하고 단순 행렬 연산 중심 구조로 바꿈.

$$\frac{dH(t)}{dt} = H(t) \times_1 (\hat{A} - I) + H(t) \times_2 (U - I) + H(t) \times_3 (W - I) + H_0$$

$$\Rightarrow \quad \frac{dH(t)}{dt} = \ln \hat{A} H(t) + H(t) \ln W + H_0$$





Over-smoothing 문제없는
우리의 STGODE Model !!

Take-Home Messages

1. **Spatial-Temporal Graph ODE (STGODE) model** to address limitations of existing spatial-temporal prediction models, which struggle to capture **long-range dependencies** b/w distant roads and suffer from **over-smoothing** when stacked deeply.

2. **Graph ODE** enables deep network architectures without suffering from the over-smoothing problem.

3. **Spatial and Semantic adjacency matrices** are combined to model both physical proximity and traffic pattern similarity among roads.

Many thanks for your attention