Deep Learning for Human Mobility Analytics

-- L8: Learning Spatio-Temporal Graph Data

Yuxuan Liang (梁宇轩)

INTR & DSA Thrust

yuxuanliang@hkust-gz.edu.cn

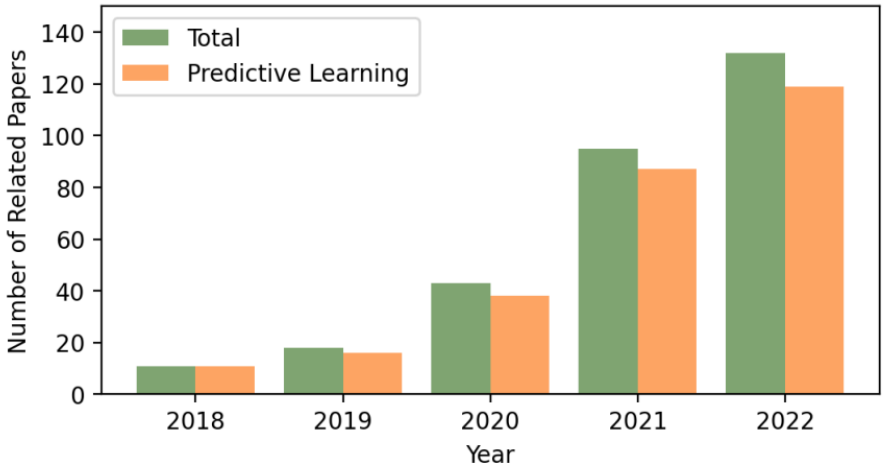
香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

Objectives of this Course

To introduce

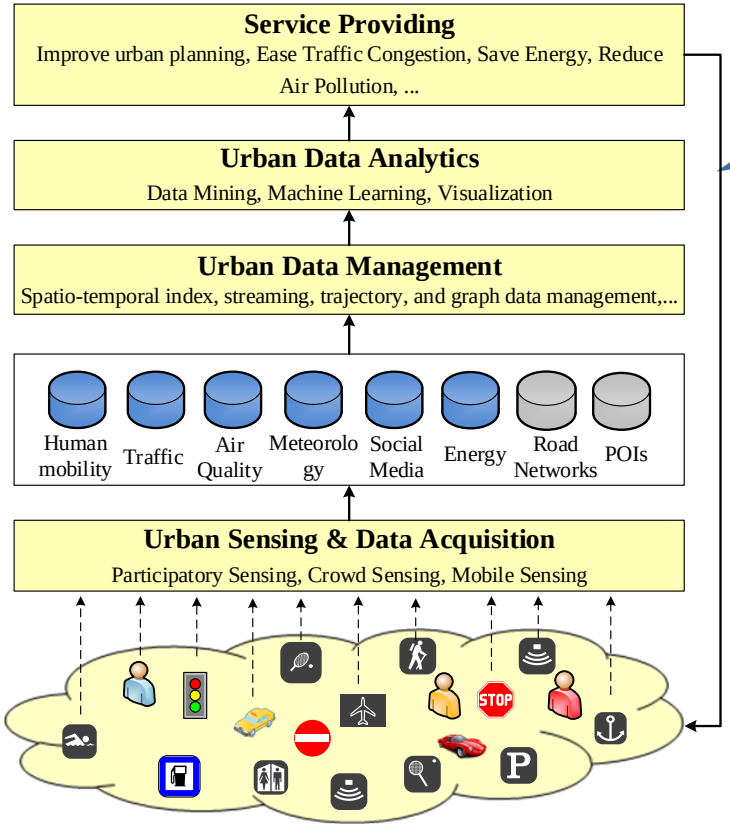
- Spatio-temporal graph construction
- Application domains
- Basic neural architecture
- STGNN variants
- Advanced learning framework

A hot topic!

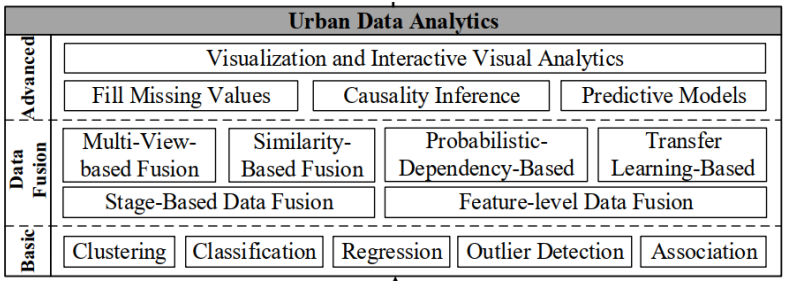


G. Jin, Y. Liang et al., Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey. TKDE 2023.

3rd Stage: Urban Data Analytics



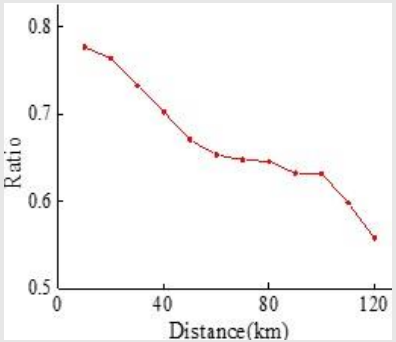
- Texts and images → spatio-temporal data
- A single data source → cross-domain data sources
- Separate data mining algorithms → ML + data management
- Visual and interactive data analytics



Spatio-Temporal Data is Unique

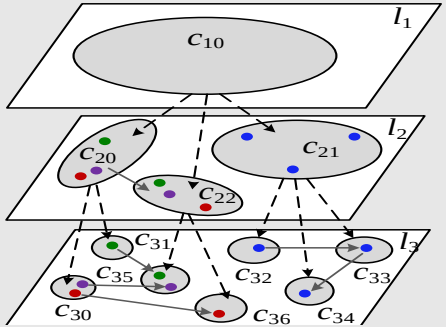
- Spatial property

Spatial closeness



Describing correlations

Spatial hierarchy



Structural constraints between different spatial granularity

Spatial distance

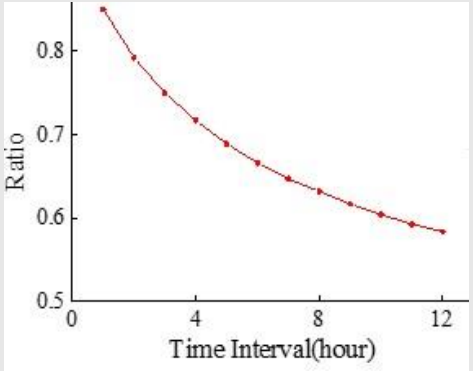
Triangle inequality:

$|d_1 - d_2| \leq d_3 \leq |d_1 + d_2|$

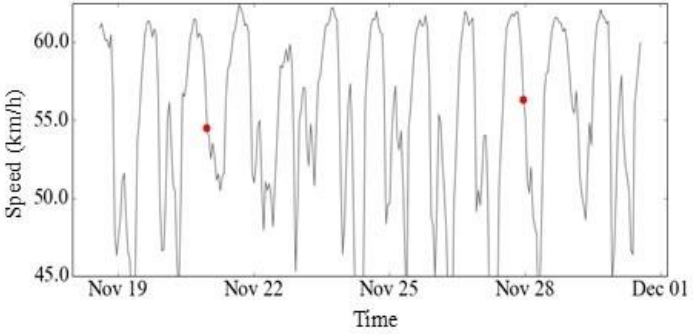
Spatio-Temporal Data is Unique

- Temporal property

Closeness

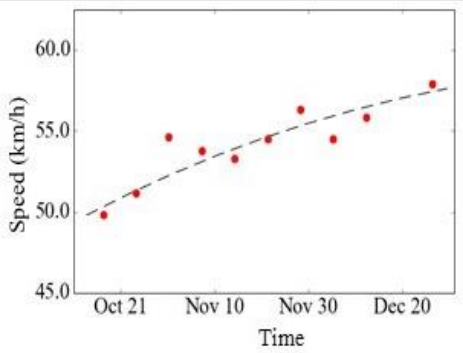


Periodicity



A) Hourly traffic speed on consecutive days

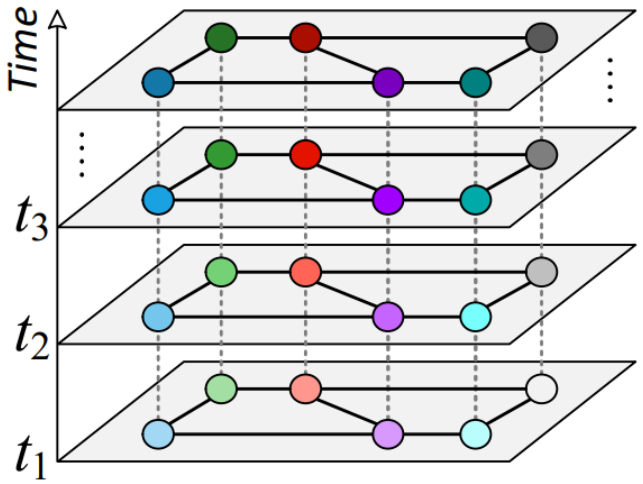
Trend



B) Traffic speed at 9-10am on consecutive weekends

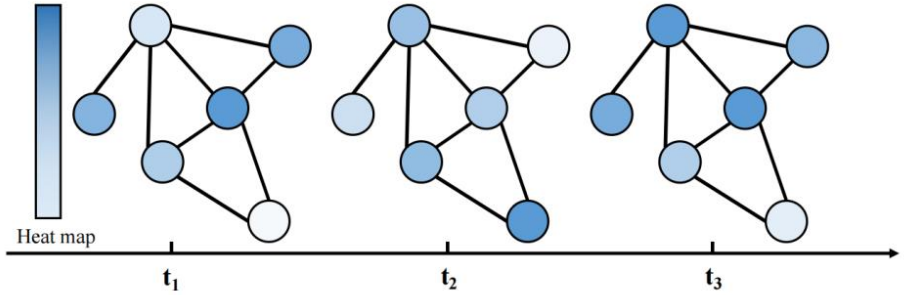
Definition of Spatio-Temporal Graphs (STG)

- There are numerous sensors deployed in the physical world
- Properties
  - Each sensor has a unique geospatial location
  - Constantly reporting time series readings
  - With structural correlation between readings
    - Usually represented as graphs
- **Examples in the human mobility domain**
  - Traffic speed/flow over road networks
  - Crowd flow in irregular urban regions

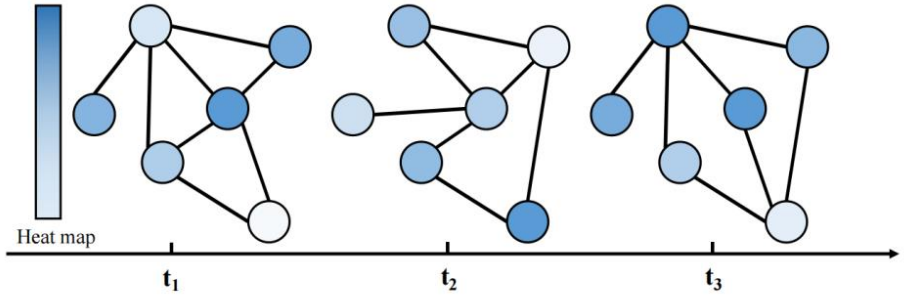


Spatio-Temporal Graph Construction

- Static or Dynamic?



(a) Static spatio-temporal graph



(b) Dynamic spatio-temporal graph

Construction Methods

- Topology-based graph

$$a_{ij}^{t} = \begin{cases} 1, \text{ if } v_i \text{ connects to } v_j \\ 0, \text{ otherwise} \end{cases},$$

- Distance-based graph

$$a_{ij}^{t} = \begin{cases} \dfrac{\exp(-\left\| d_{ij}^{t} \right\|_2)}{\sigma}, \text{if } d_{ij}^{t} < \epsilon, \\ 0, \quad \text{otherwise} \end{cases}$$

Construction Methods

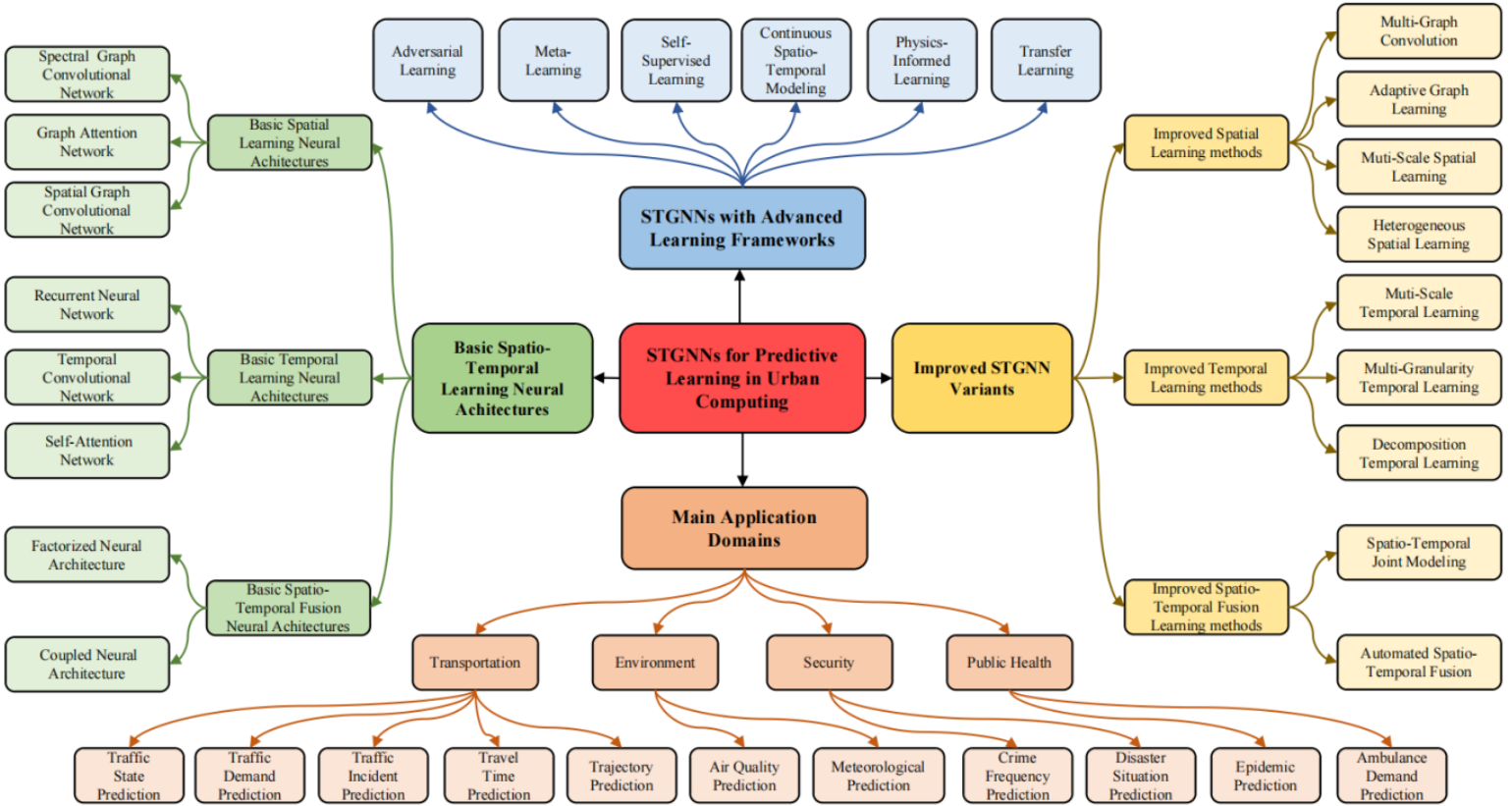
- Similarity-based graph

$$a_{ij}^t = \begin{cases} \dfrac{\sum_{i=1}^{n} \left(x_i^{0:t} - x_i^{\bar{0}:t}\right)\left(x_j^{0:t} - x_j^{\bar{0}:t}\right)}{\sqrt{\sum_{i=1}^{n} \left(x_i^{0:t} - x_i^{\bar{0}:t}\right)^2}\sqrt{\sum_{i=1}^{n} \left(x_j^{0:t} - x_j^{\bar{0}:t}\right)^2}}, \\ 0, \quad \text{otherwise} \end{cases}$$

- Interaction-based graph

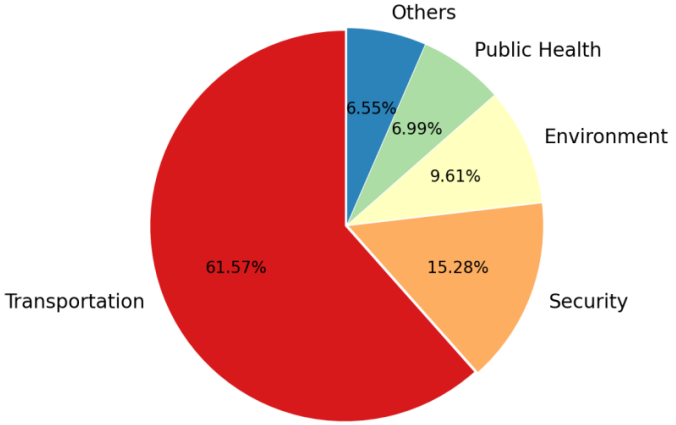
$$a_{ij}^t = \begin{cases} \dfrac{F_{ij}^t}{\sum_{m \in N(i)} F_{im}^t}, \ if \ F_{ij}^t > 0 \\ 0, \quad \text{otherwise} \end{cases},$$

Taxonomy


STGNNs for Predictive Learning in Urban Computing
STGNNs with Advanced Learning Frameworks
Adversarial Learning
Meta-Learning
Self-Supervised Learning
Continuous Spatio-Temporal Modeling
Physics-Informed Learning
Transfer Learning
Basic Spatio-Temporal Learning Neural Achitectures
Basic Spatial Learning Neural Achitectures
Spectral Graph Convolutional Network
Graph Attention Network
Spatial Graph Convolutional Network
Basic Temporal Learning Neural Achitectures
Recurrent Neural Network
Temporal Convolutional Network
Self-Attention Network
Basic Spatio-Temporal Fusion Neural Achitectures
Factorized Neural Architecture
Coupled Neural Architecture
Improved STGNN Variants
Improved Spatial Learning methods
Multi-Graph Convolution
Adaptive Graph Learning
Muti-Scale Spatial Learning
Heterogeneous Spatial Learning
Improved Temporal Learning methods
Muti-Scale Temporal Learning
Multi-Granularity Temporal Learning
Decomposition Temporal Learning
Improved Spatio-Temporal Fusion Learning methods
Spatio-Temporal Joint Modeling
Automated Spatio-Temporal Fusion
Main Application Domains
Transportation
Environment
Security
Public Health
Traffic State Prediction
Traffic Demand Prediction
Traffic Incident Prediction
Travel Time Prediction
Trajectory Prediction
Air Quality Prediction
Meteorological Prediction
Crime Frequency Prediction
Disaster Situation Prediction
Epidemic Prediction
Ambulance Demand Prediction

Application Domains

- Transportation
  - Traffic forecasting, demand anlaysis
- Environment
  - Air quality forecasting, weather forecasting
- Public safety
  - Crime prediction, disaster prediction
- Public health
  - Epidemic analysis, ambulance demand prediction

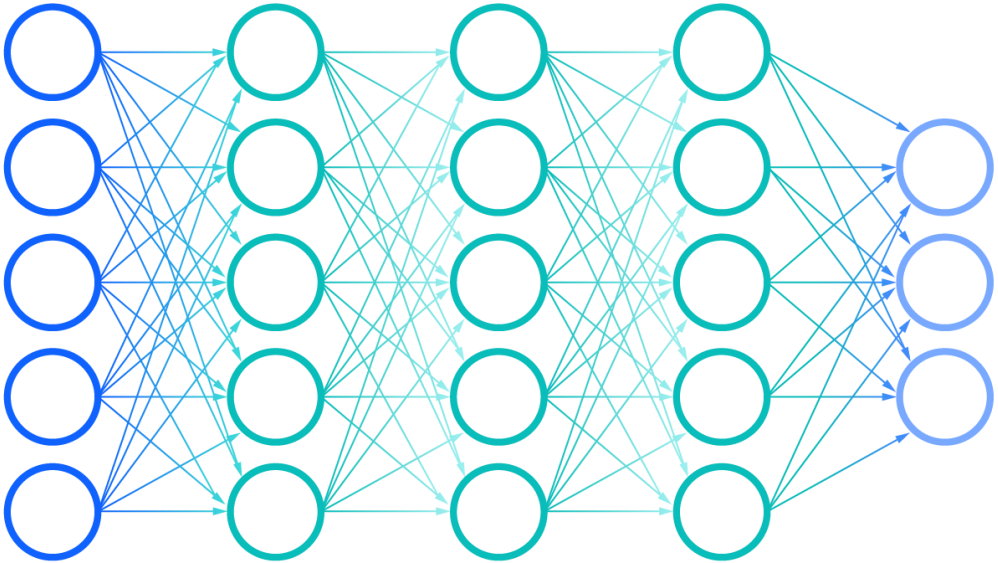


Preliminary

- Multi-Layer Perceptron (MLP)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Graph Neural Network (GNN)
- Self-attention & Transformer

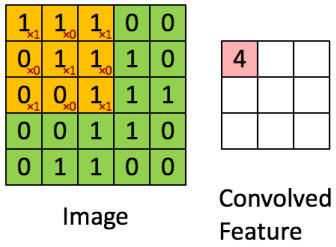
Multi-Layer Perceptron (MLP)

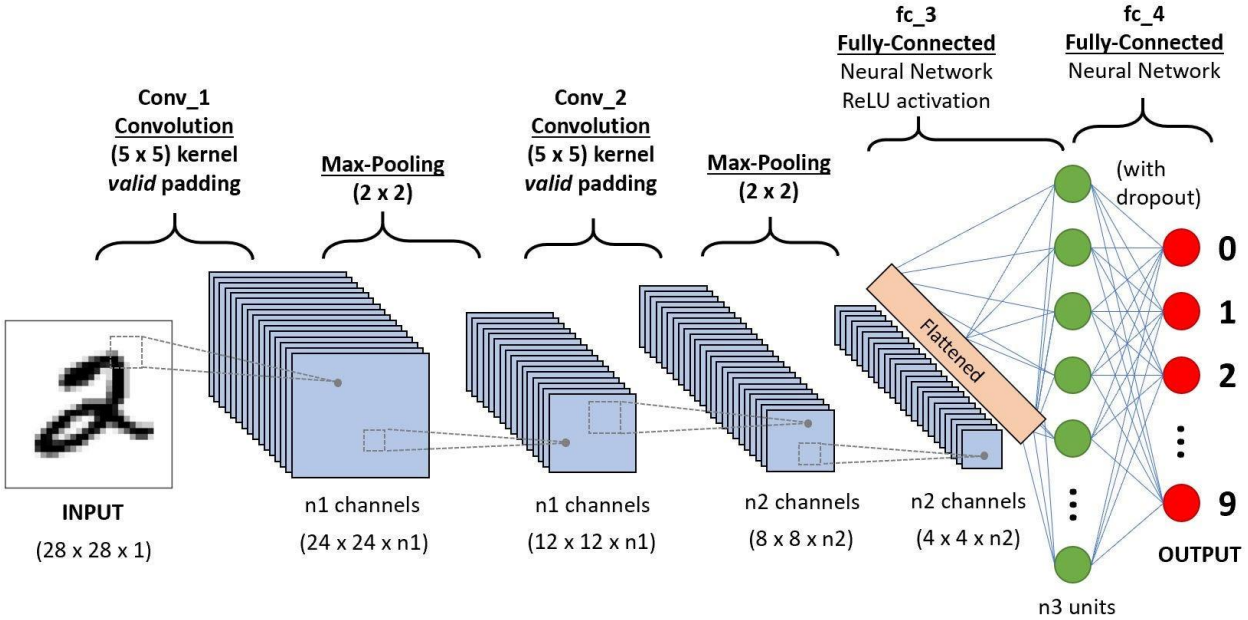
- Input layers
- Hidden layers
- Output layers



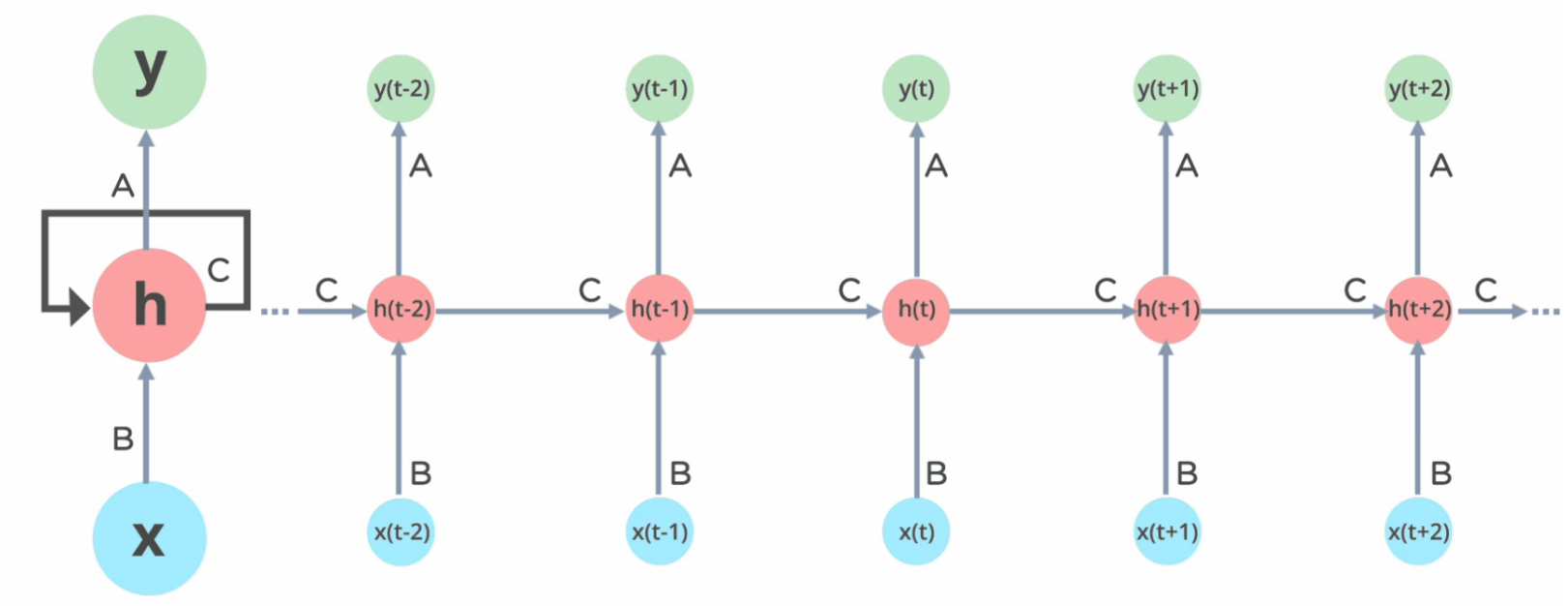
Convolutional Neural Networks (CNN)

- Convolution
- Max-pooling
- FC/MLP



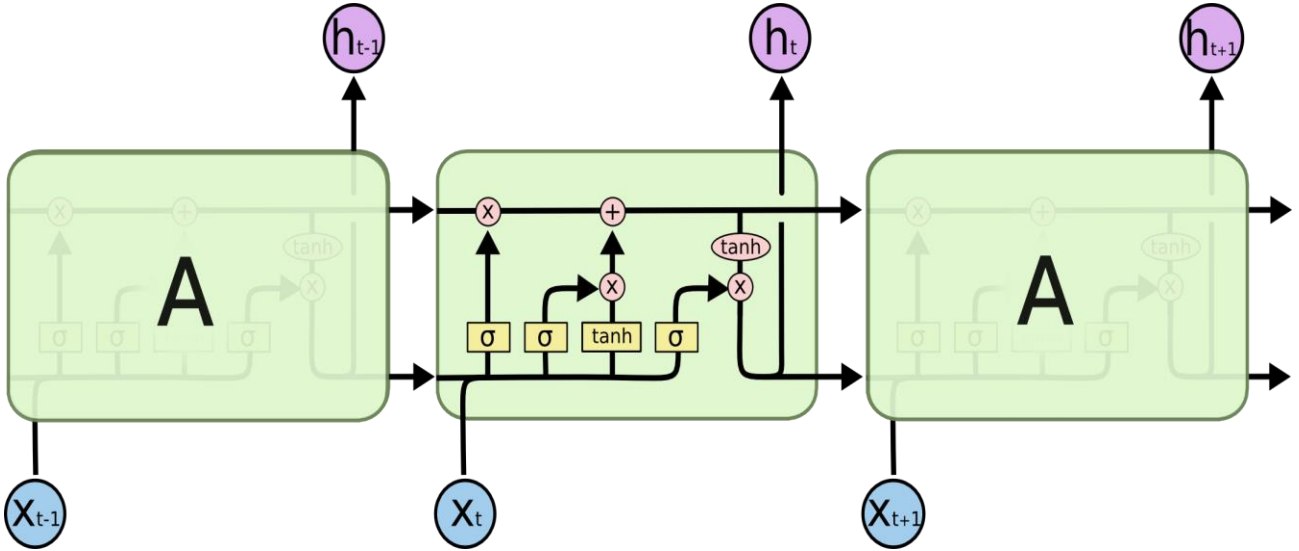


Recurrent Neural Networks (RNN)



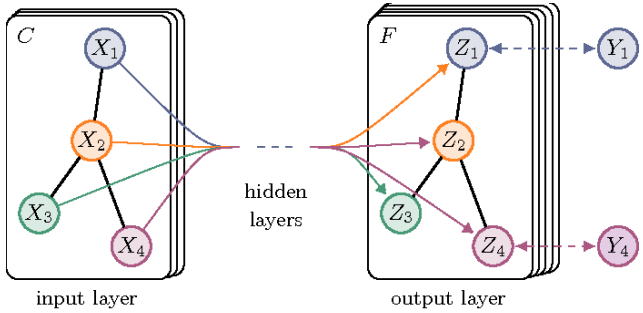
Long Short-Term Memory (LSTM)

- Avoiding gradient explosion and vanishing

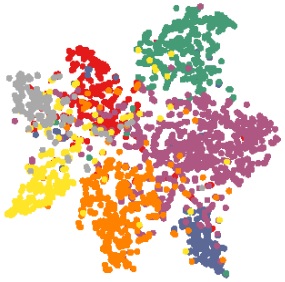


Graph Neural Networks (GNN)

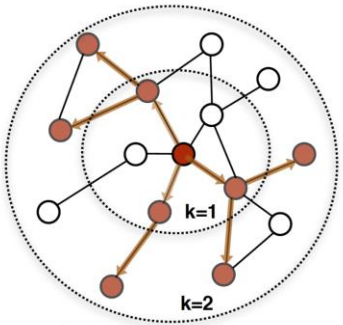
- Spectral-based GNNs
- Spatial-based GNNs



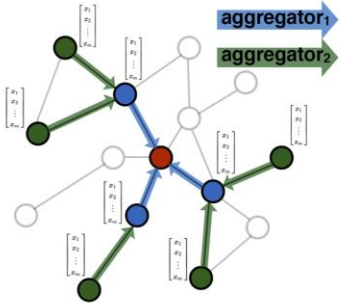
(a) Graph Convolutional Network



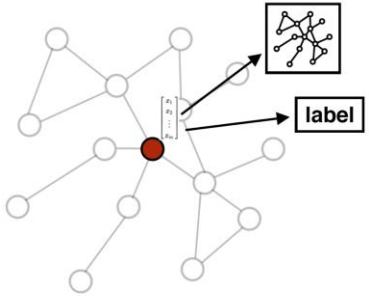
(b) Hidden layer activations



1. Sample neighborhood



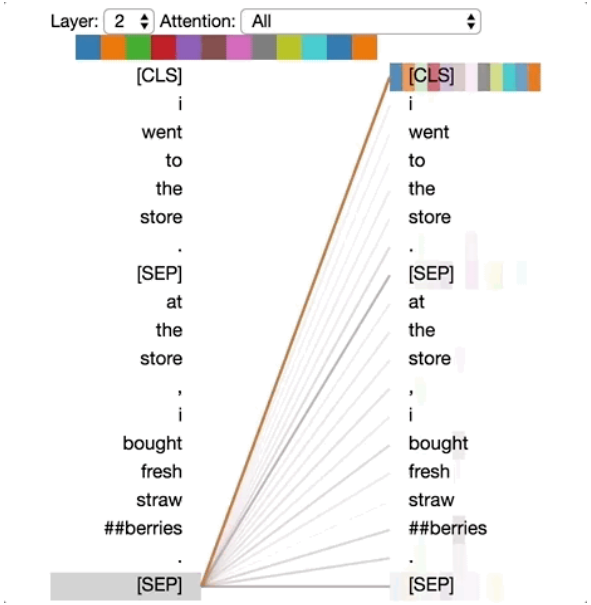
2. Aggregate feature information from neighbors

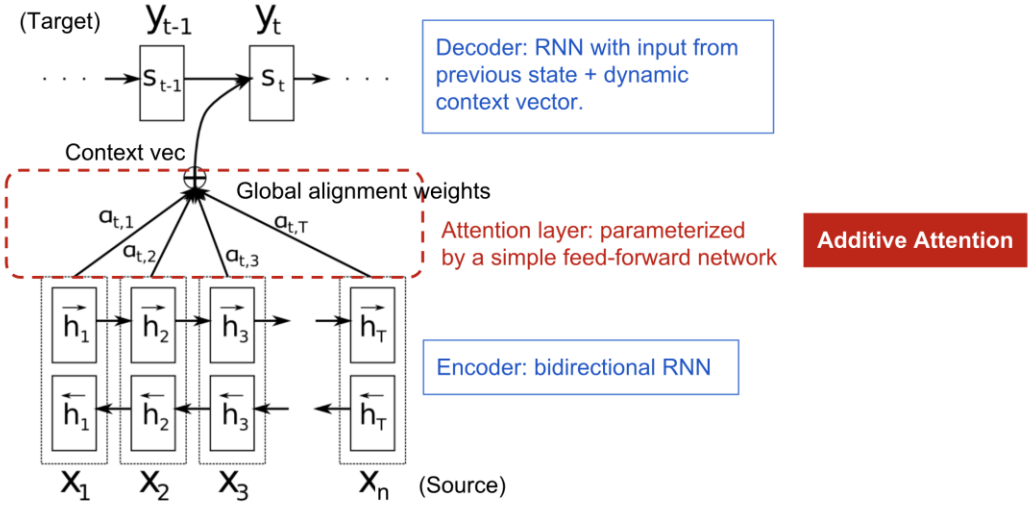


3. Predict graph context and label using aggregated information

Attention Mechanism

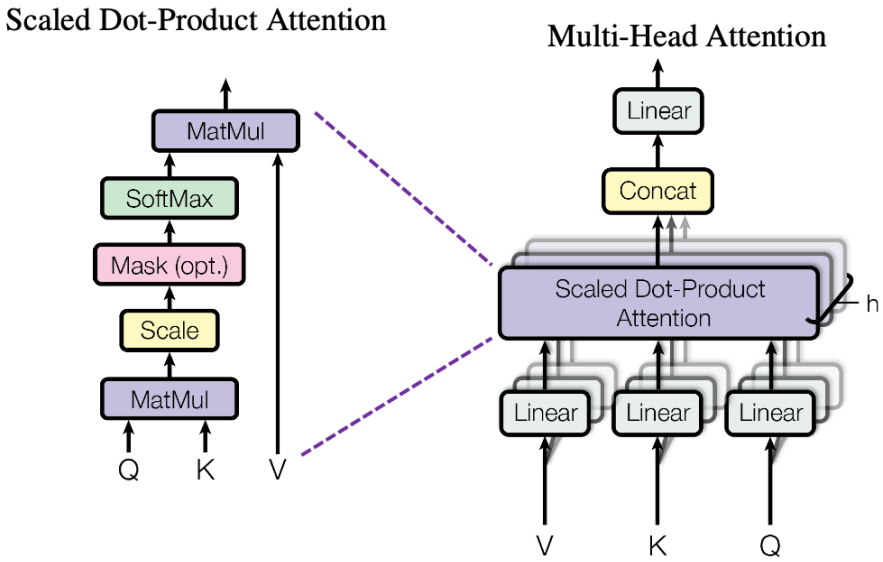
- Born for machine translation [Bahdanau et al. 2013]

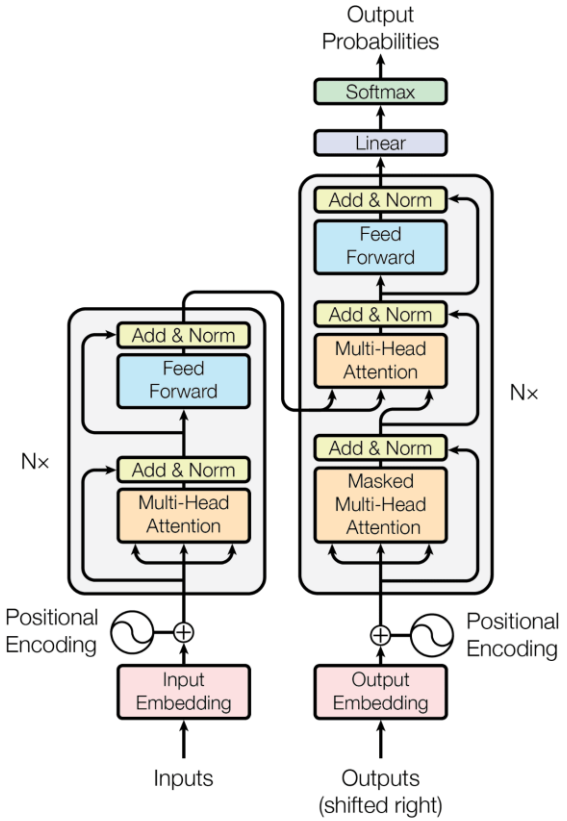




Transformers (Self-Attention)

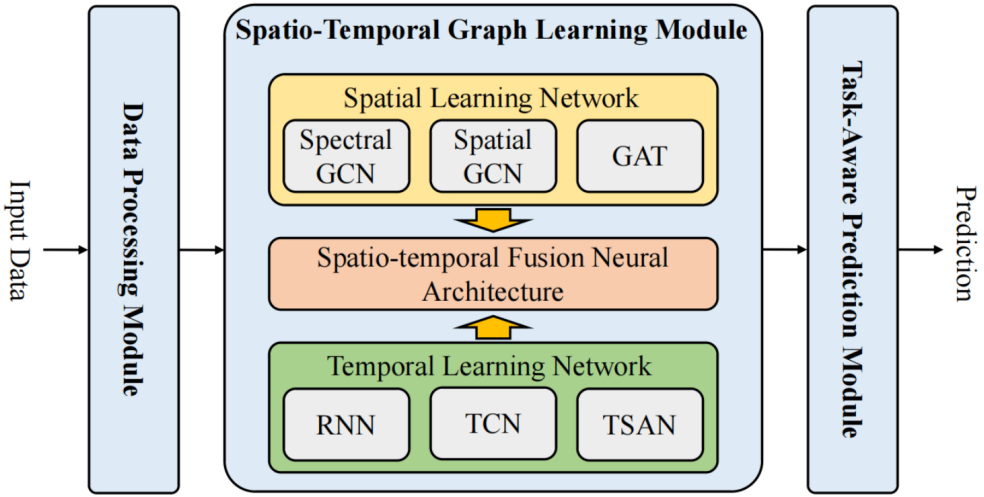
- Attention is all you need



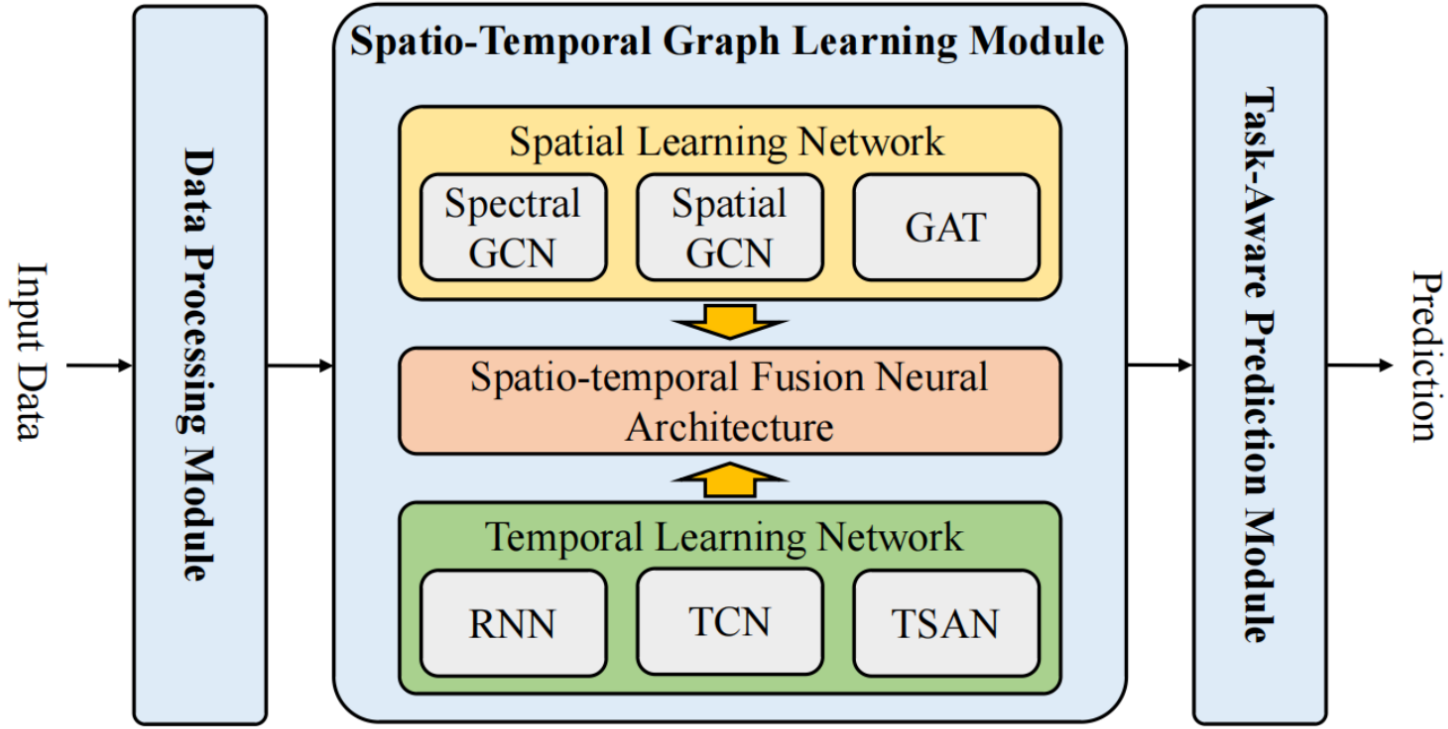


Insights of STG Modeling

- To predict the future of STGs, we should consider
  - The temporal relations across different time steps
  - The spatial relations between a node and its neighborhood



Basic Neural Architecture



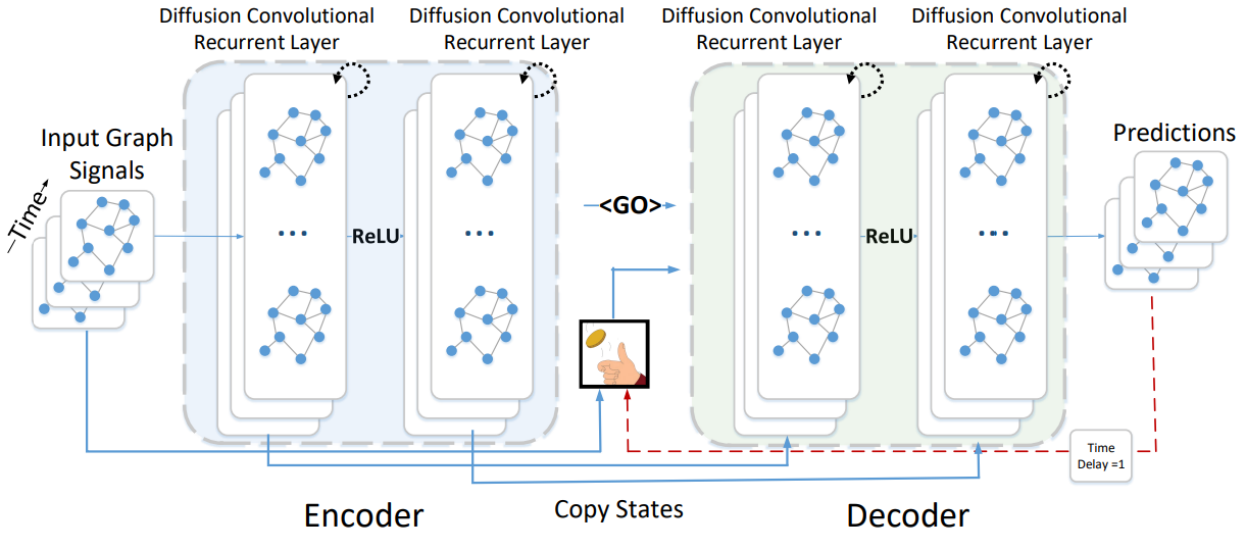
G. Jin, Y. Liang et al., Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey. arXiv 2023.

Representative Solutions

- DCRNN [ICLR'18]
- STGCN [IJCAI'18]
- T-GCN [TITS'19]
- Graph WaveNet [IJCAI'19]
- MTGNN [KDD'20]

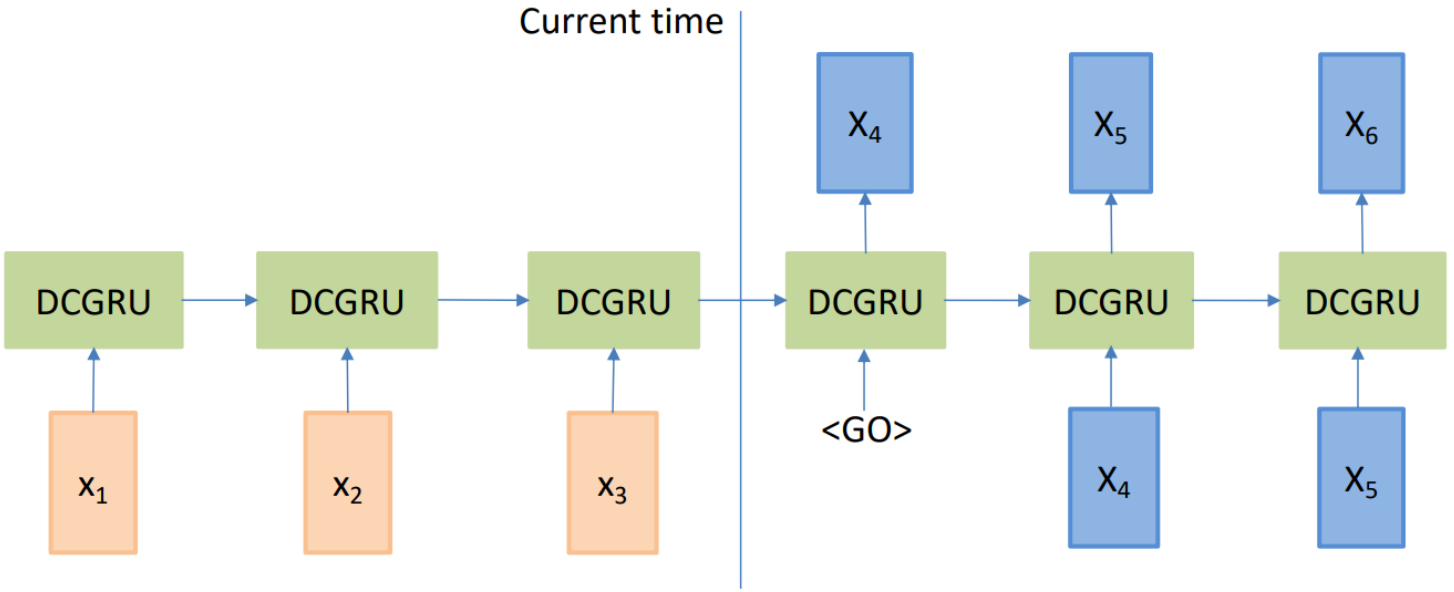
Diffusion Convolutional Recurrent Neural Network (DCRNN)

- Capturing temporal dependencies using RNNs (i.e., GRUs)
- Capturing spatial dependencies using Diffusion Convolution



Y. Li et al., Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. ICLR 2018.

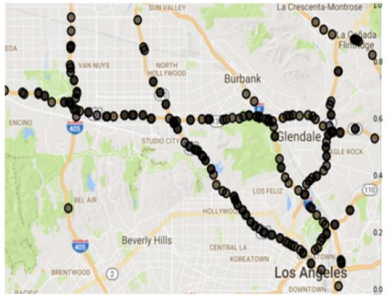
Formula



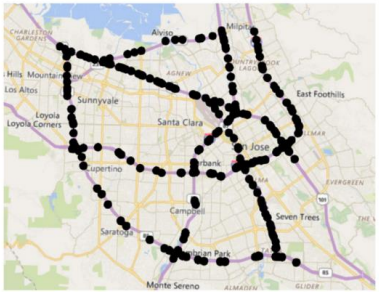
$$r^{(t)} = \sigma(\Theta_r \star_{\mathcal{G}} [X^{(t)}, H^{(t-1)}] + b_r) \qquad u^{(t)} = \sigma(\Theta_u \star_{\mathcal{G}} [X^{(t)}, H^{(t-1)}] + b_u)$$

$$C^{(t)} = \tanh(\Theta_C \star_{\mathcal{G}} [X^{(t)}, (r^{(t)} \odot H^{(t-1)})] + b_c) \qquad H^{(t)} = u^{(t)} \odot H^{(t-1)} + (1 - u^{(t)}) \odot C^{(t)}$$

Evaluation



(a) METR-LA



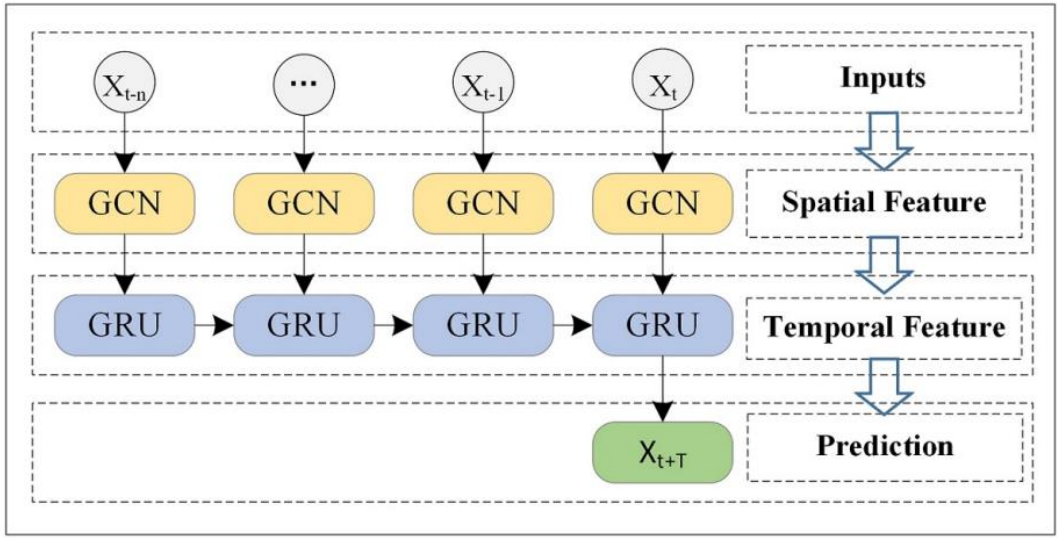
(b) PEMS-BAY

Table 1: Performance comparison of different approaches for traffic speed forecasting. DCRNN achieves the best performance with all three metrics for all forecasting horizons, and the advantage becomes more evident with the increase of the forecasting horizon.

	T	Metric	HA	ARIMA$_{Kal}$	VAR	SVR	FNN	FC-LSTM	*DCRNN*
METR-LA	15 min	MAE	4.16	3.99	4.42	3.99	3.99	3.44	**2.77**
		RMSE	7.80	8.21	7.89	8.45	7.94	6.30	**5.38**
		MAPE	13.0%	9.6%	10.2%	9.3%	9.9%	9.6%	**7.3%**
	30 min	MAE	4.16	5.15	5.41	5.05	4.23	3.77	**3.15**
		RMSE	7.80	10.45	9.13	10.87	8.17	7.23	**6.45**
		MAPE	13.0%	12.7%	12.7%	12.1%	12.9%	10.9%	**8.8%**
	1 hour	MAE	4.16	6.90	6.52	6.72	4.49	4.37	**3.60**
		RMSE	7.80	13.23	10.11	13.76	8.69	8.69	**7.59**
		MAPE	13.0%	17.4%	15.8%	16.7%	14.0%	13.2%	**10.5%**
PEMS-BAY	15 min	MAE	2.88	1.62	1.74	1.85	2.20	2.05	**1.38**
		RMSE	5.59	3.30	3.16	3.59	4.42	4.19	**2.95**
		MAPE	6.8%	3.5%	3.6%	3.8%	5.19%	4.8%	**2.9%**
	30 min	MAE	2.88	2.33	2.32	2.48	2.30	2.20	**1.74**
		RMSE	5.59	4.76	4.25	5.18	4.63	4.55	**3.97**
		MAPE	6.8%	5.4%	5.0%	5.5%	5.43%	5.2%	**3.9%**
	1 hour	MAE	2.88	3.38	2.93	3.28	2.46	2.37	**2.07**
		RMSE	5.59	6.50	5.44	7.08	4.98	4.96	**4.74**
		MAPE	6.8%	8.3%	6.5%	8.0%	5.89%	5.7%	**4.9%**

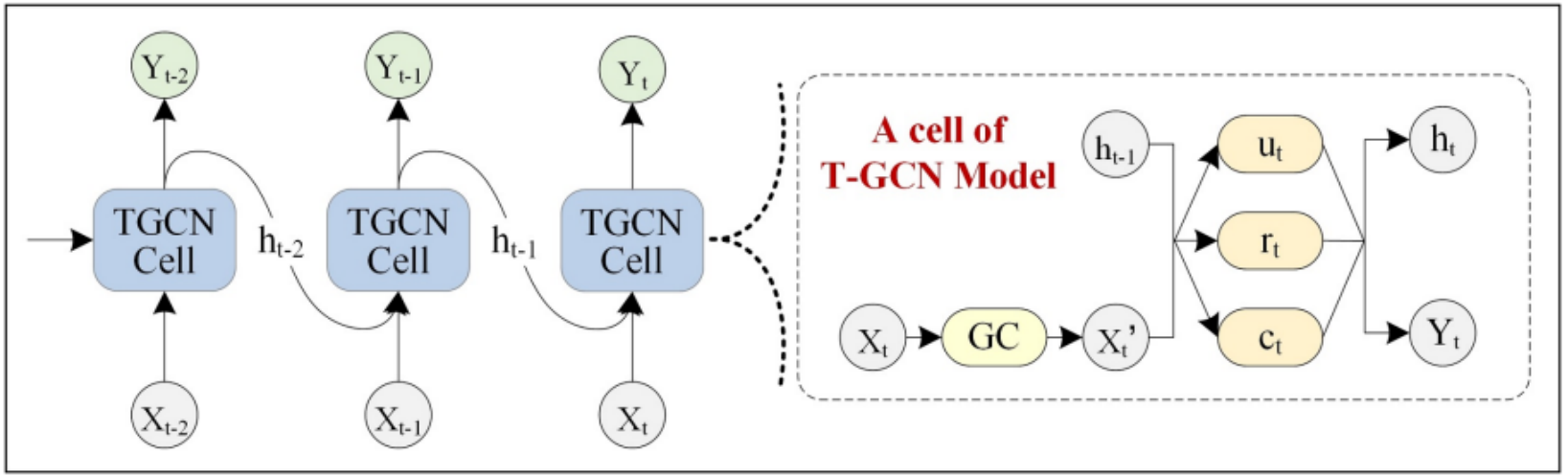
Temporal Graph Convolutional Network (T-GCN)

- Capturing temporal dependencies using RNNs (i.e., GRUs)
- Capturing spatial dependencies using Graph Convolutions



L. Zhao et al., T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. TITS 2019.

T-GCN Cell



$$u_t = \sigma(W_u\,[f(A, X_t), h_{t-1}] + b_u)$$

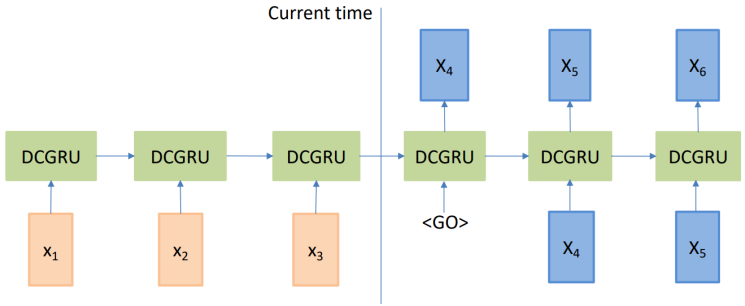
$$r_t = \sigma(W_r\,[f(A, X_t), h_{t-1}] + b_r)$$

$$c_t = tanh(W_c\,[f(A, X_t), (r_t * h_{t-1})] + b_c)$$

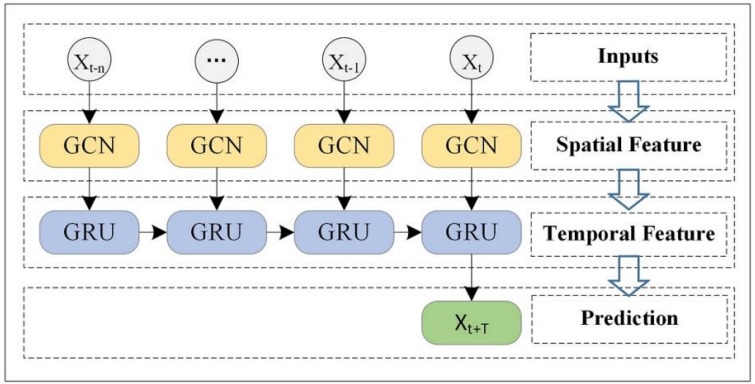
$$h_t = u_t * h_{t-1} + (1 - u_t) * c_t$$

Difference between DCRNN and T-GCN

- DCRNN: performing GNN inside each temporal cell
- T-GCN: performing GNN before each temporal cell



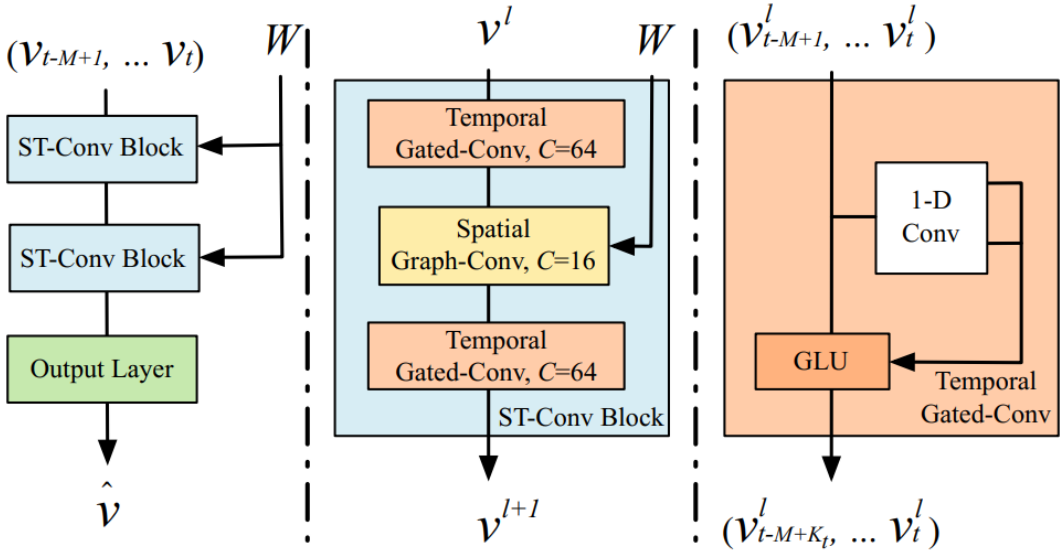
DCRNN



T-GCN

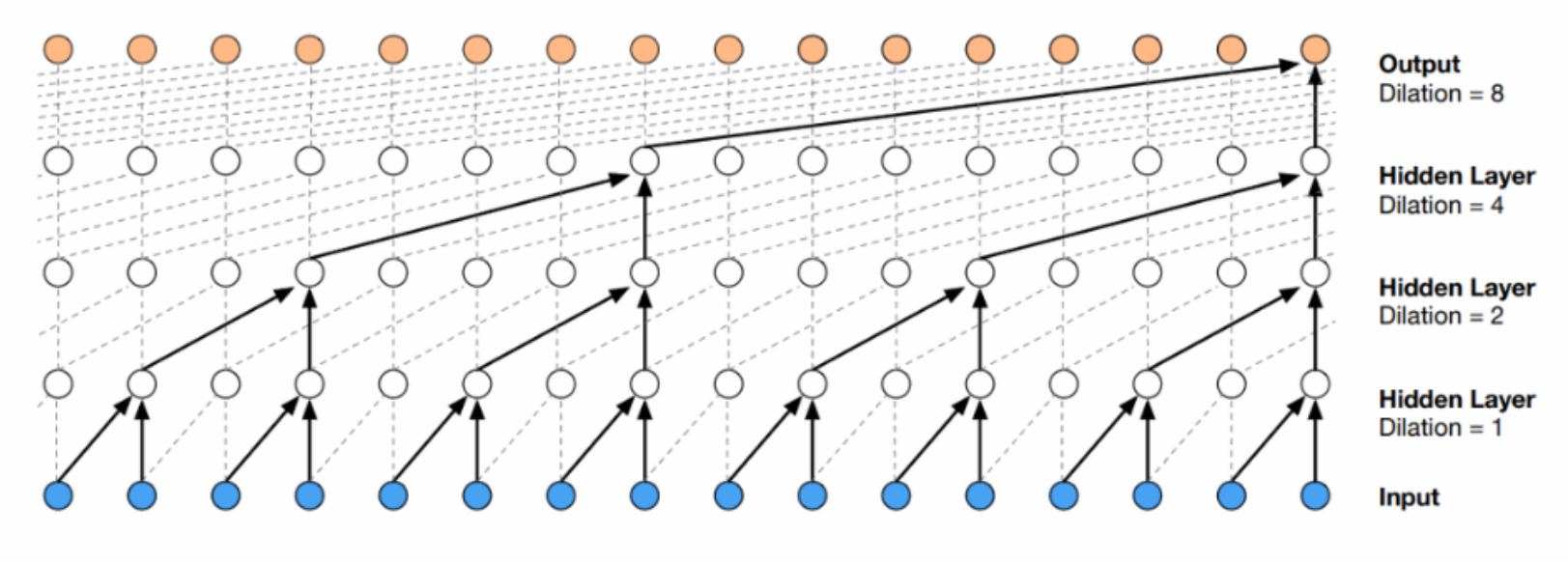
ST Graph Convolutional Network (STGCN)

- Capturing temporal dependencies using Temporal Convolutions
- Capturing spatial dependencies with GCNs



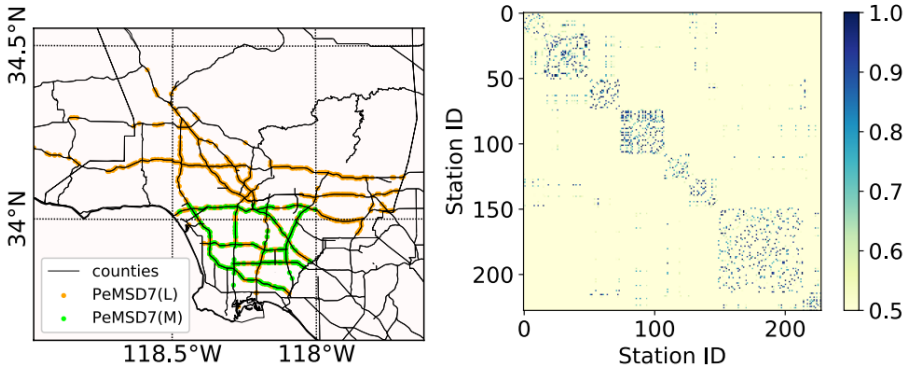
B. Yu et al., Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. IJCAI 2018.

Temporal Convolutional Networks (TCN)



$$\Gamma *_{\mathcal{T}} Y = P \odot \sigma(Q) \in \mathbb{R}^{(M-K_t+1)\times C_o}$$

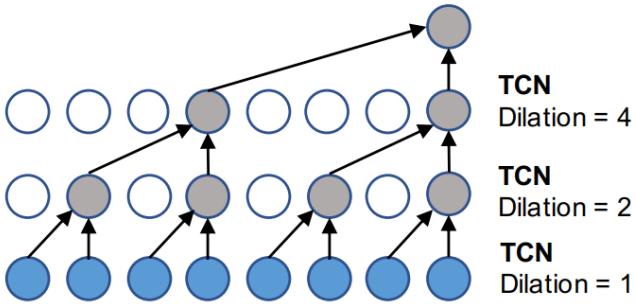
Evaluation

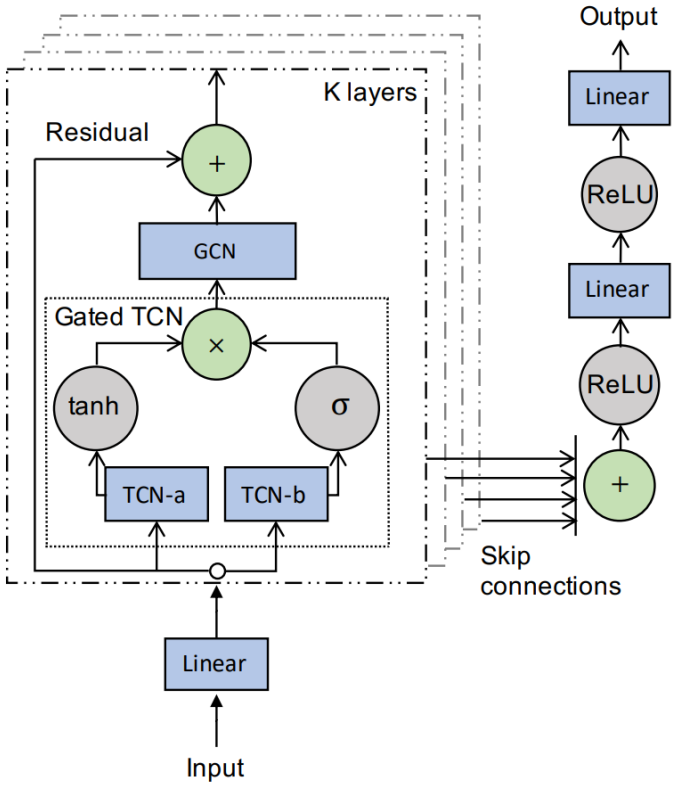


Model	PeMSD7(M) (15/ 30/ 45 min)			PeMSD7(L) (15/ 30/ 45 min)		
	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE
HA	4.01	10.61	7.20	4.60	12.50	8.05
LSVR	2.50/ 3.63/ 4.54	5.81/ 8.88/ 11.50	4.55/ 6.67/ 8.28	2.69/ 3.85/ 4.79	6.27/ 9.48/ 12.42	4.88/ 7.10/ 8.72
ARIMA	5.55/ 5.86/ 6.27	12.92/ 13.94/ 15.20	9.00/ 9.13/ 9.38	5.50/ 5.87/ 6.30	12.30/ 13.54/ 14.85	8.63/ 8.96/ 9.39
FNN	2.74/ 4.02/ 5.04	6.38/ 9.72/ 12.38	4.75/ 6.98/ 8.58	2.74/ 3.92/ 4.78	7.11/ 10.89/ 13.56	4.87/ 7.02/ 8.46
FC-LSTM	3.57/ 3.94/ 4.16	8.60/ 9.55/ 10.10	6.20/ 7.03/ 7.51	4.38/ 4.51/ 4.66	11.10/ 11.41/ 11.69	7.68/ 7.94/ 8.20
GCGRU	2.37/ 3.31/ 4.01	5.54/ 8.06/ 9.99	4.21/ 5.96/ 7.13	2.48/ 3.43/ 4.12 *	5.76/ 8.45/ 10.51 *	4.40/ 6.25/ 7.49 *
STGCN(Cheb)	**2.25/ 3.03/ 3.57**	5.26/ **7.33/ 8.69**	**4.04/ 5.70/ 6.77**	**2.37/ 3.27/ 3.97**	**5.56/ 7.98/ 9.73**	**4.32/ 6.21/ 7.45**
STGCN(1^{st})	2.26/ 3.09/ 3.79	**5.24**/ 7.39/ 9.12	4.07/ 5.77/ 7.03	2.40/ 3.31/ 4.01	5.63/ 8.21/ 10.12	4.38/ 6.43/ 7.81

Graph WaveNet

- Using GCN to model spatial relations
- Using dilated TCNs to capture temporal relations





Z. Wu et al., Graph WaveNet for Deep Spatial-Temporal Graph Modeling. IJCAI 2019.

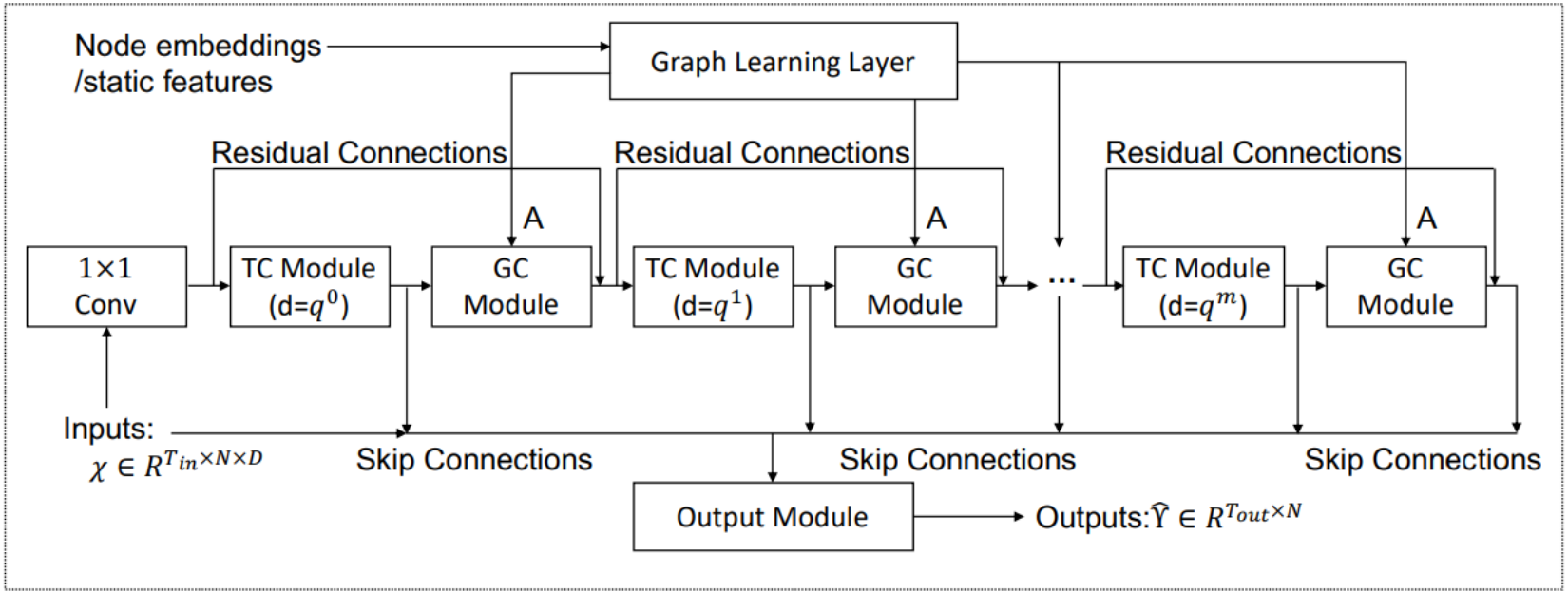
Evaluation

Data	Models	15 min			30 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	ARIMA [Li *et al.*, 2018b]	3.99	8.21	9.60%	5.15	10.45	12.70%	6.90	13.23	17.40%
	FC-LSTM [Li *et al.*, 2018b]	3.44	6.30	9.60%	3.77	7.23	10.90%	4.37	8.69	13.20%
	WaveNet [Oord *et al.*, 2016]	2.99	5.89	8.04%	3.59	7.28	10.25%	4.45	8.93	13.62%
	DCRNN [Li *et al.*, 2018b]	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.60	10.50%
	GGRU [Zhang *et al.*, 2018]	2.71	5.24	6.99%	3.12	6.36	8.56%	3.64	7.65	10.62%
	STGCN [Yu *et al.*, 2018]	2.88	5.74	7.62%	3.47	7.24	9.57%	4.59	9.40	12.70%
	Graph WaveNet	**2.69**	**5.15**	**6.90%**	**3.07**	**6.22**	**8.37%**	**3.53**	**7.37**	**10.01%**
PEMS-BAY	ARIMA [Li *et al.*, 2018b]	1.62	3.30	3.50%	2.33	4.76	5.40%	3.38	6.50	8.30%
	FC-LSTM [Li *et al.*, 2018b]	2.05	4.19	4.80%	2.20	4.55	5.20%	2.37	4.96	5.70%
	WaveNet [Oord *et al.*, 2016]	1.39	3.01	2.91%	1.83	4.21	4.16%	2.35	5.43	5.87%
	DCRNN [Li *et al.*, 2018b]	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	GGRU [Zhang *et al.*, 2018]	-	-	-	-	-	-	-	-	-
	STGCN [Yu *et al.*, 2018]	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	Graph WaveNet	**1.30**	**2.74**	**2.73%**	**1.63**	**3.70**	**3.67%**	**1.95**	**4.52**	**4.63%**

Evaluation

Dataset	Model Name	Adjacency Matrix Configuration	Mean MAE	Mean RMSE	Mean MAPE
METR-LR	Identity	$[\mathbf{I}]$	3.58	7.18	10.21%
	Forward-only	$[\mathbf{P}]$	3.13	6.26	8.65%
	Adaptive-only	$[\tilde{\mathbf{A}}_{adp}]$	3.10	6.21	8.68%
	Forward-backward	$[\mathbf{P}_f, \mathbf{P}_b]$	3.08	6.13	8.25%
	Forward-backward-adaptive	$[\mathbf{P}_f, \mathbf{P}_b, \tilde{\mathbf{A}}_{adp}]$	**3.04**	**6.09**	**8.23%**
PEMS-BAY	Identity	$[\mathbf{I}]$	1.80	4.05	4.18%
	Forward-only	$[\mathbf{P}_f]$	1.62	3.61	3.72%
	Adaptive-only	$[\tilde{\mathbf{A}}_{adp}]$	1.61	3.63	3.59%
	Forward-backward	$[\mathbf{P}_f, \mathbf{P}_b]$	1.59	3.55	3.57%
	Forward-backward-adaptive	$[\mathbf{P}_f, \mathbf{P}_b, \tilde{\mathbf{A}}_{adp}]$	**1.58**	**3.52**	**3.55%**

MTGNN

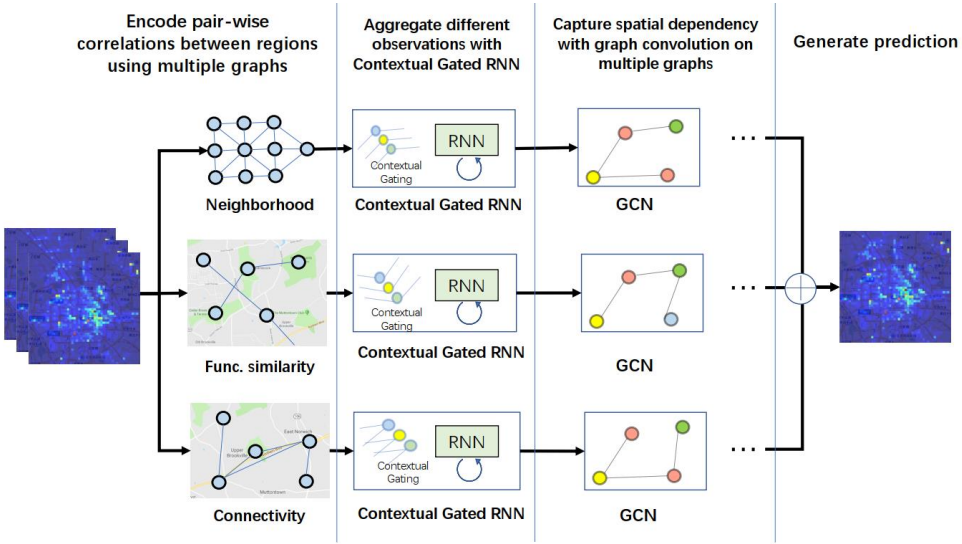

Node embeddings
/static features
Graph Learning Layer
Residual Connections
Residual Connections
Residual Connections
A
A
A
1×1
Conv
TC Module
(d=q^0)
GC
Module
TC Module
(d=q^1)
GC
Module
...
TC Module
(d=q^m)
GC
Module
Inputs:
$\chi \in R^{T_{in} \times N \times D}$
Skip Connections
Skip Connections
Skip Connections
Output Module
Outputs: $\hat{Y} \in R^{T_{out} \times N}$

Variants of STGNN

- Variants of Spatial Learning
  - Spatio-Temporal Multi-Graph Convolution Networks
  - Adaptive Graph Learning
  - Multi-Scale Spatial Learning
- Variants of Temporal Learning
  - Multi-Granularity Temporal Learning
  - Decomposed Temporal Learning

ST Multi-Graph Convolution Network (ST-MGCN)

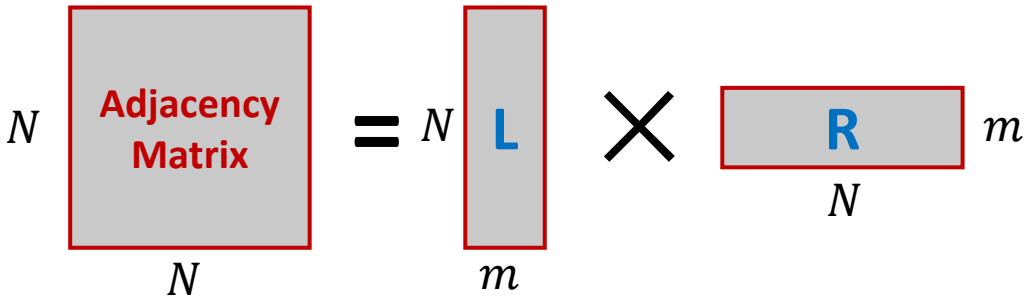
Figure 1: An example of different correlations among regions. To predict the demand in region **1**, spatially adjacent region **2**, functionality similar region **3** and transportation connected region **4** are considered more important, while distant and irrelevant regions **5** are less relevant.

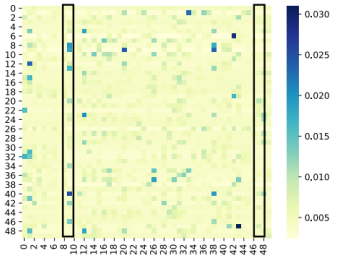


X. Geng et al., Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. AAAI 2019.

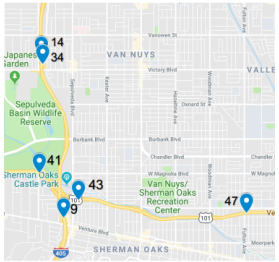
Adaptive Graph Learning

- **Adaptive adjacency matrix** (inspired by RecSys)
  - Adaptively generated from two low-rank matrices
  - L and R can be viewed as location embeddings
  - The embeddings can reflect the similarity between different nodes in the latent space





(a) The heatmap of the learned self-adaptive adjacency matrix for the first 50 nodes.



(b) The geographical location of a part of nodes marked on Google Maps.

Z. Wu et al., Graph WaveNet for Deep Spatial-Temporal Graph Modeling. IJCAI 2019.

Adaptive Graph Learning

- Random initialization-based methods

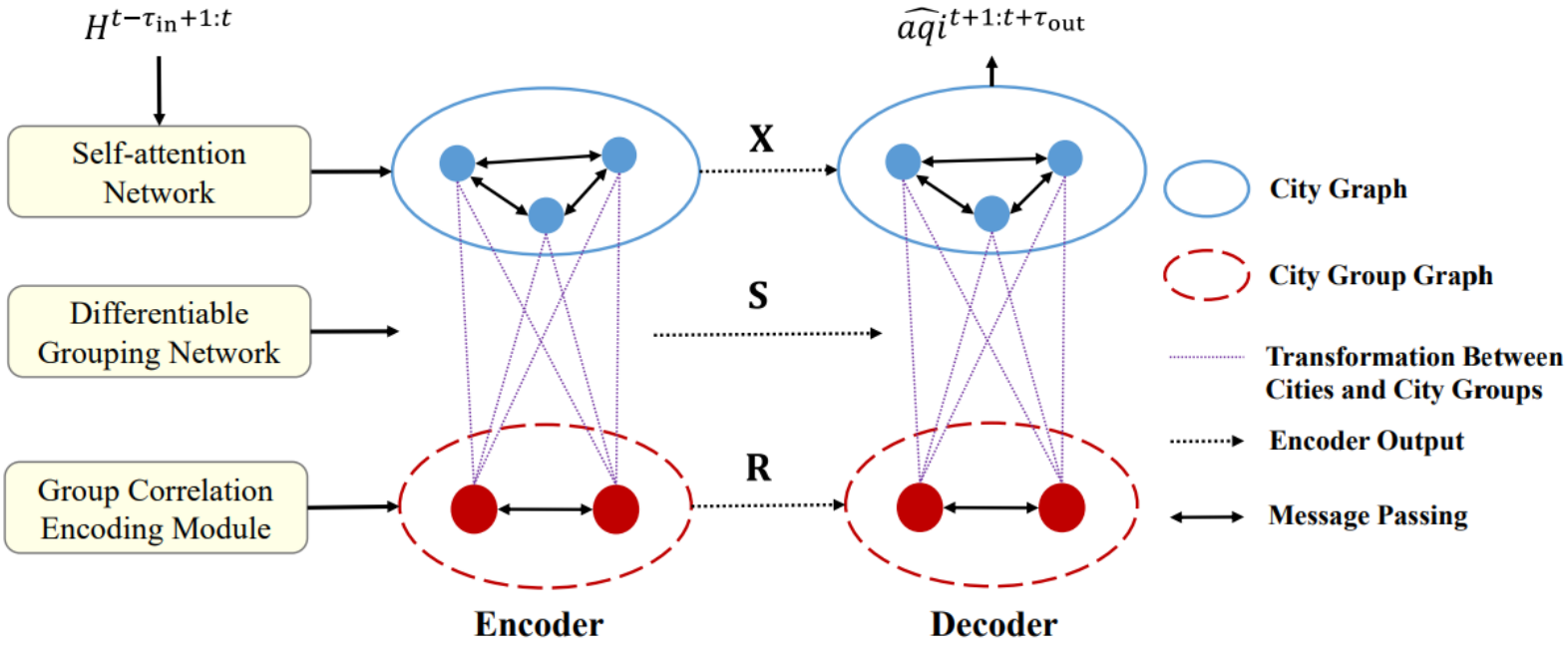
$$\tilde{\boldsymbol{A}}_{adp} = \mathrm{SoftMax}\left(\mathrm{ReLU}\left(\boldsymbol{E}_1 \boldsymbol{E}_2^T\right)\right),$$

$$\begin{aligned}
\boldsymbol{M}_1 &= \tanh\left(\alpha \boldsymbol{E}_1 \boldsymbol{\Theta}_1\right), \\
\boldsymbol{M}_2 &= \tanh\left(\alpha \boldsymbol{E}_2 \boldsymbol{\Theta}_2\right), \\
\tilde{\boldsymbol{A}}_{adp} &= \mathrm{ReLU}\left(\tanh\left(\alpha\left(\boldsymbol{M}_1 \boldsymbol{M}_2^T - \boldsymbol{M}_2 \boldsymbol{M}_1^T\right)\right)\right),
\end{aligned}$$

- Feature initialization-based approaches
  - E.g., DGCRN, DSTAGNN, BSTGCN

Multi-Scale Spatial Learning

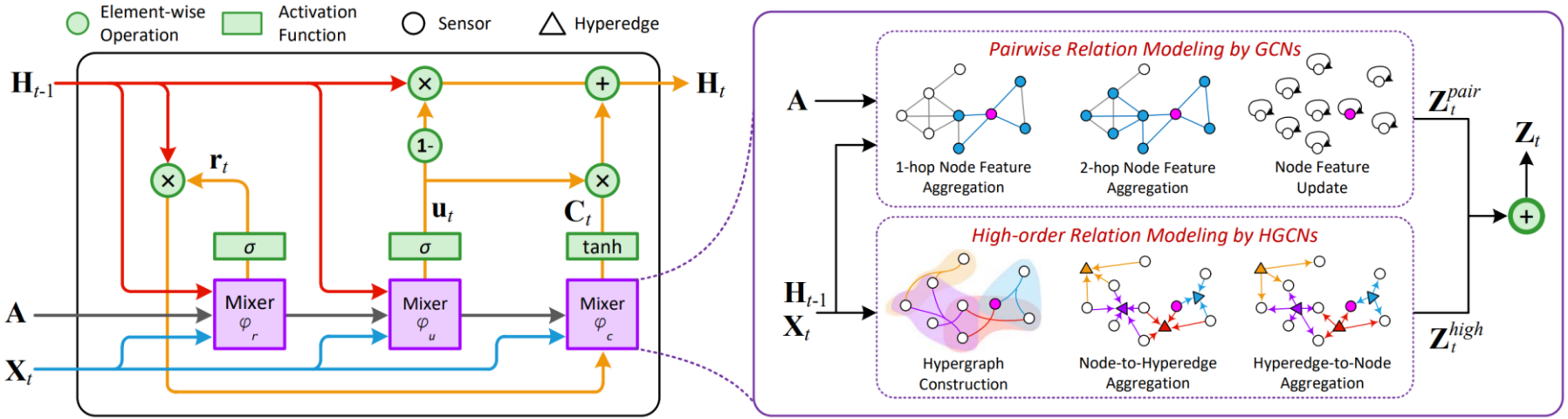
- Modeling spatial dependences at different scales



L. Chen et al., Group-Aware Graph Neural Network for Nationwide City Air Quality Forecasting. arXiv 2021.

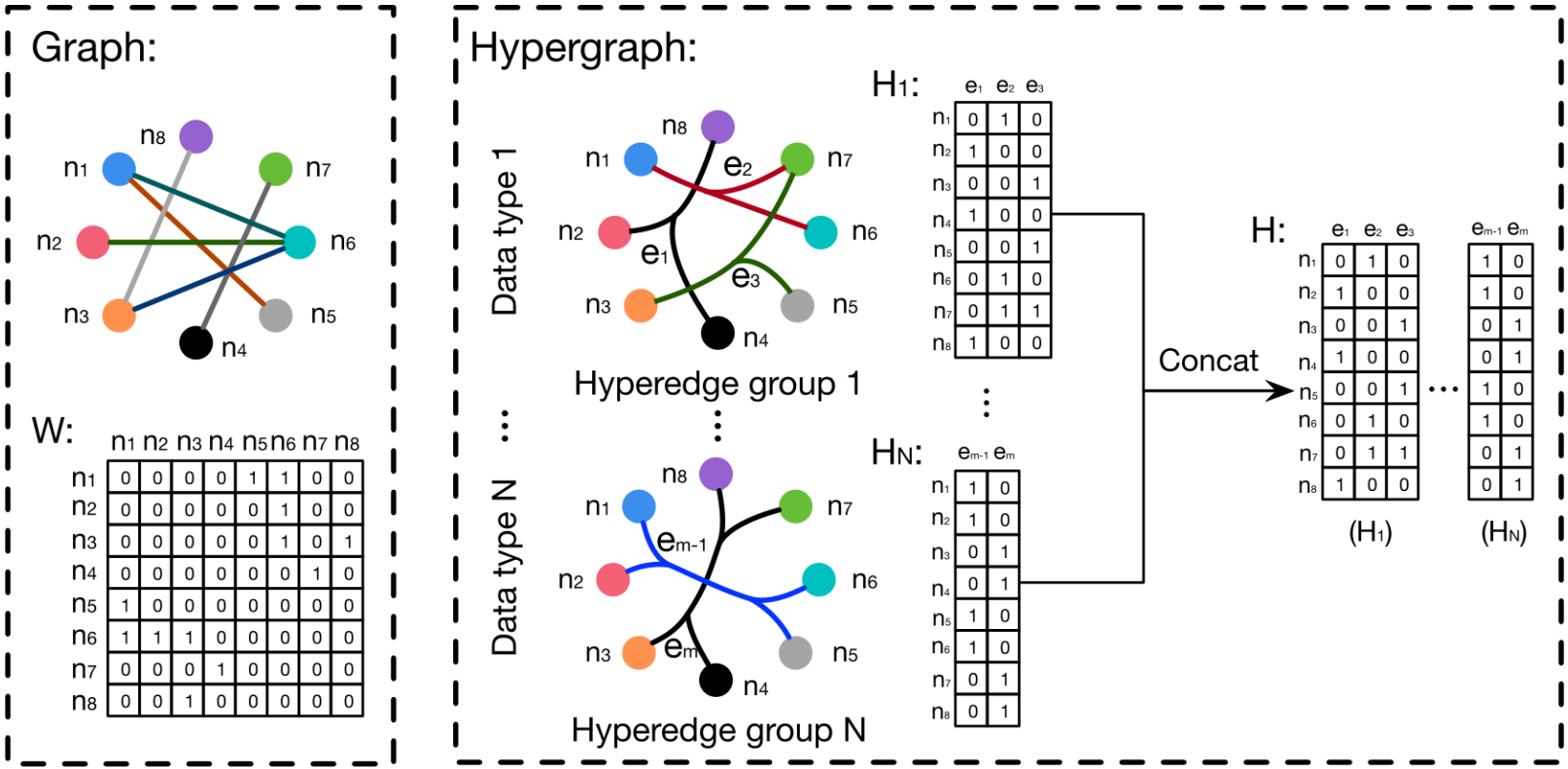
Multi-Scale Spatial Learning

- Can also be linked to hypergraphs

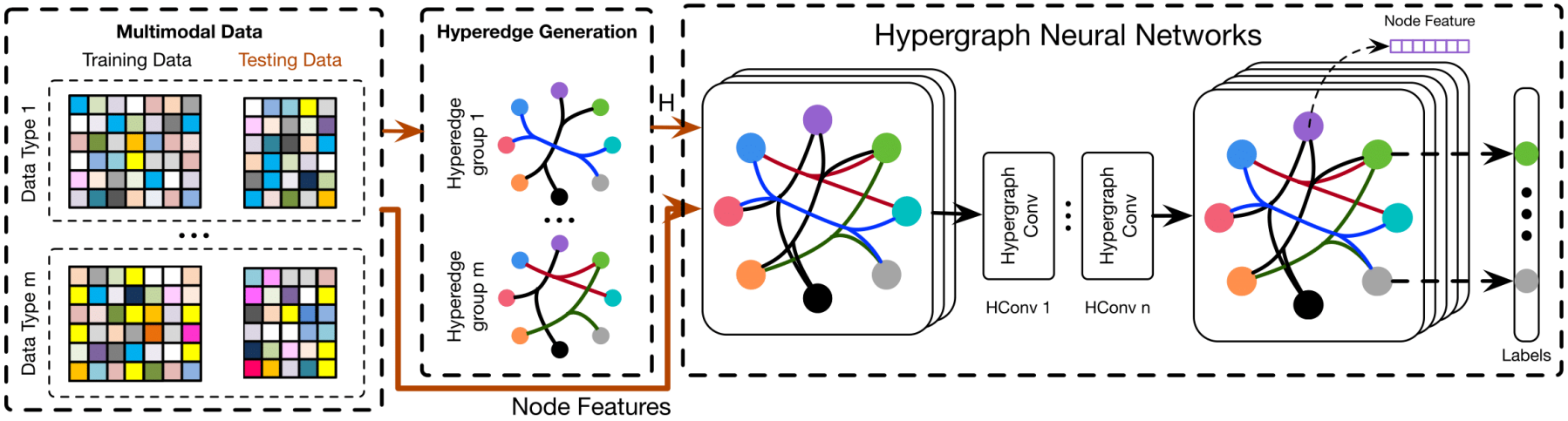


Y. Liang et al., Relation-Aware Recurrent Neural Networks for Spatio-Temporal Forecasting. TKDE 2022.

Hypergraph


Graph:
W:
Hypergraph:
Data type 1
Data type N
Hyperedge group 1
Hyperedge group N
H1:
HN:
H:
Concat
(H1)
(HN)

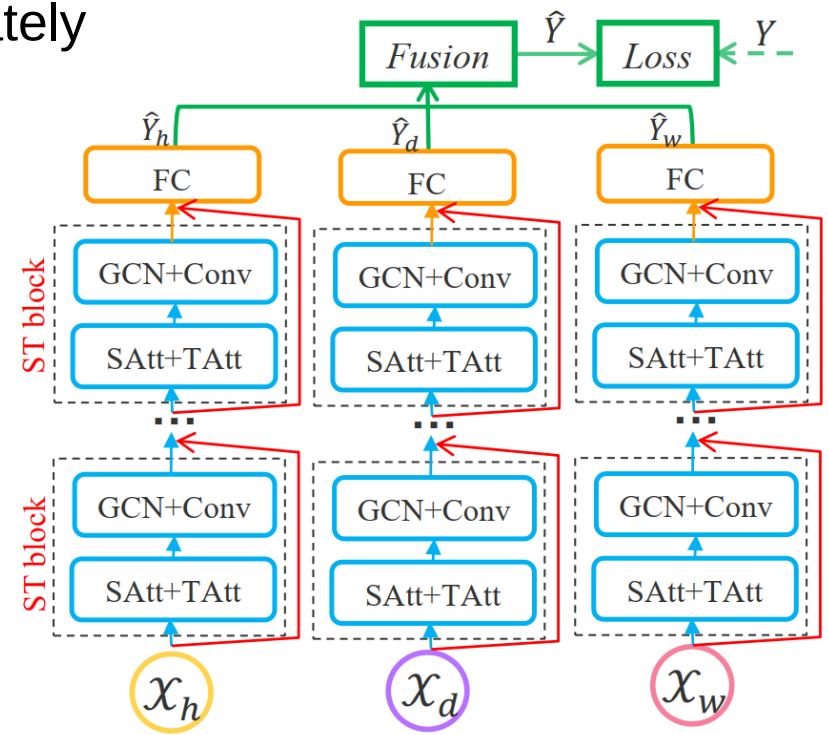
Hypergraph Convolution



$$\mathbf{X}^{(l+1)} = \sigma(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^{\top}\mathbf{D}_v^{-1/2}\mathbf{X}^{(l)}\mathbf{\Theta}^{(l)}),$$

Multi-Granularity Temporal Learning

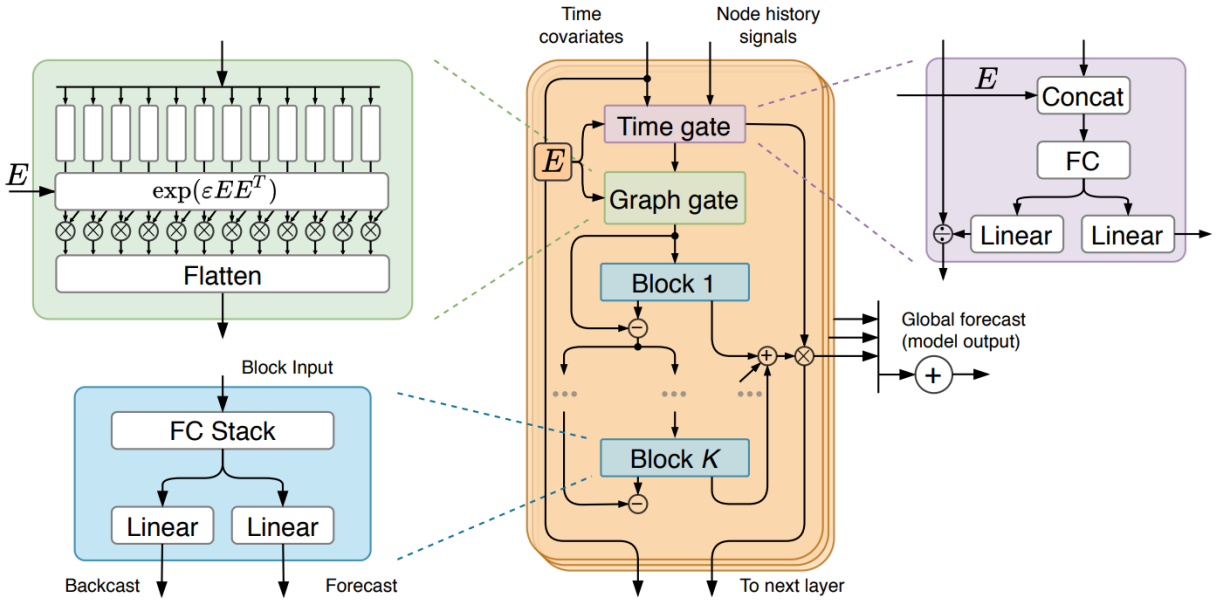
- Learning temporal properties separately
  - Closeness
  - Periodicity
  - Trend



S. Guo et al., Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. AAAI 2019.

Decomposed Temporal Learning

- This kind of variant aims to automatically decompose and integrate different temporal components (e.g., periodicity) through special neural designs



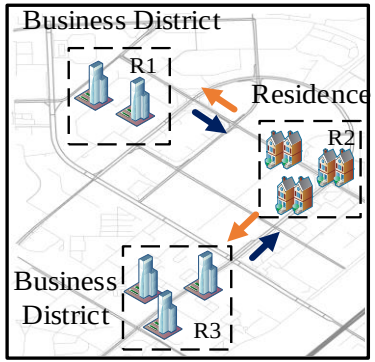
B. Oreshkin et al., FC-GAGA: Fully Connected Gated Graph Architecture for Spatio-Temporal Traffic Forecasting. AAAI 2021.

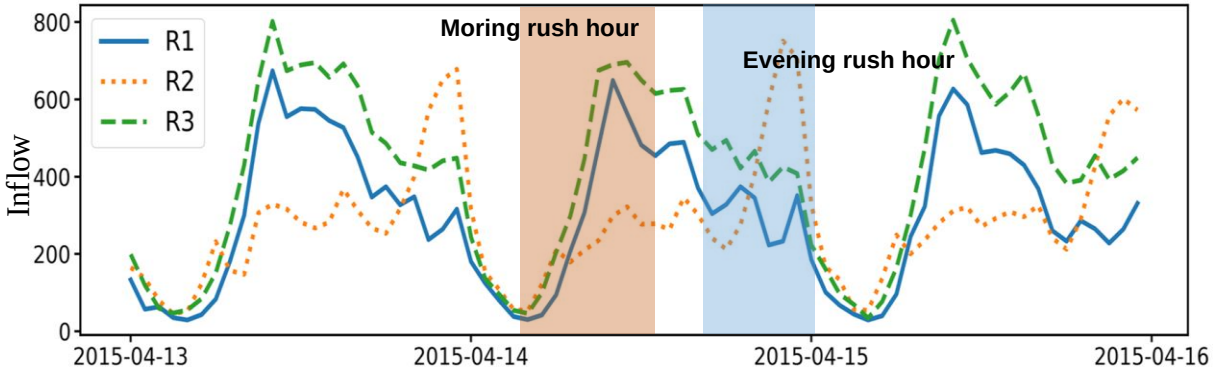
Advanced Learning Frameworks

- Adversarial Learning
- Meta Learning
- Self-Supervised Learning
- Continuous Spatio-Temporal modeling
- Physics-Informed Learning
- Transfer Learning

Challenge – Diverse ST Dependencies

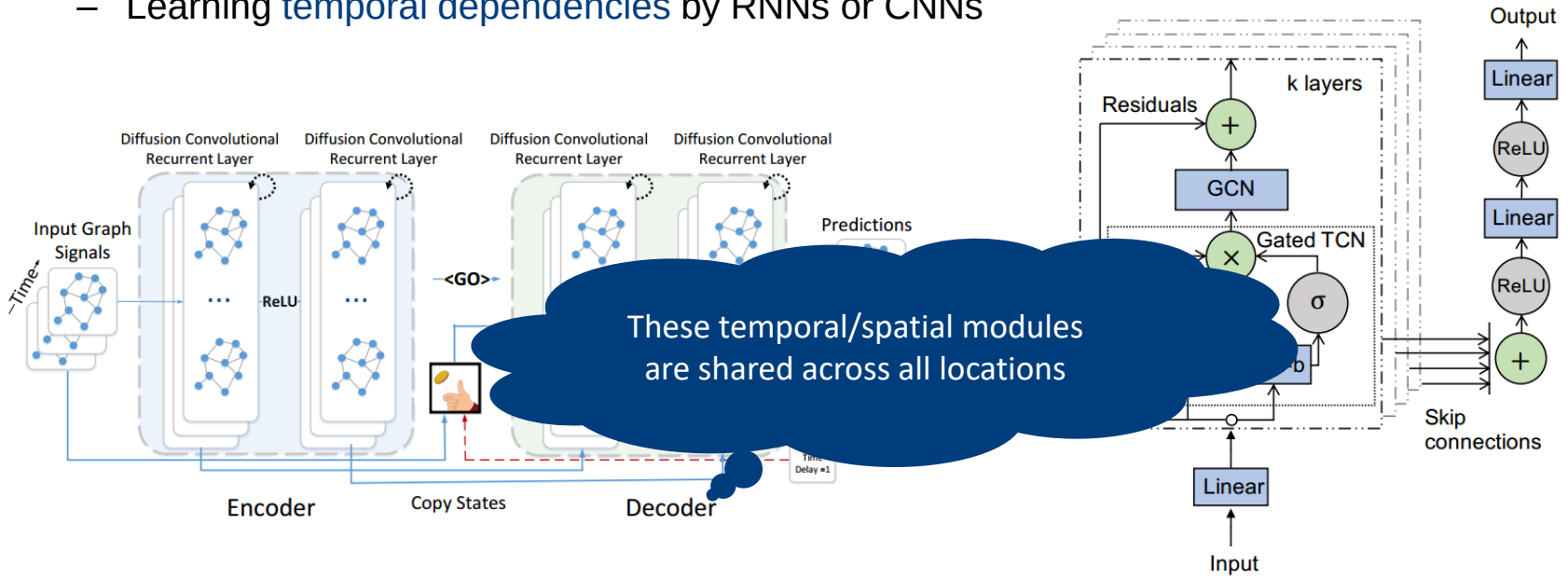
- However, spatial location characteristics and spatial interrelationships are diverse and often depend on **spatial attributes**
- Spatial locations with similar geo-attributes tend to have
  - Similar temporal properties
  - Similar spatial correlations





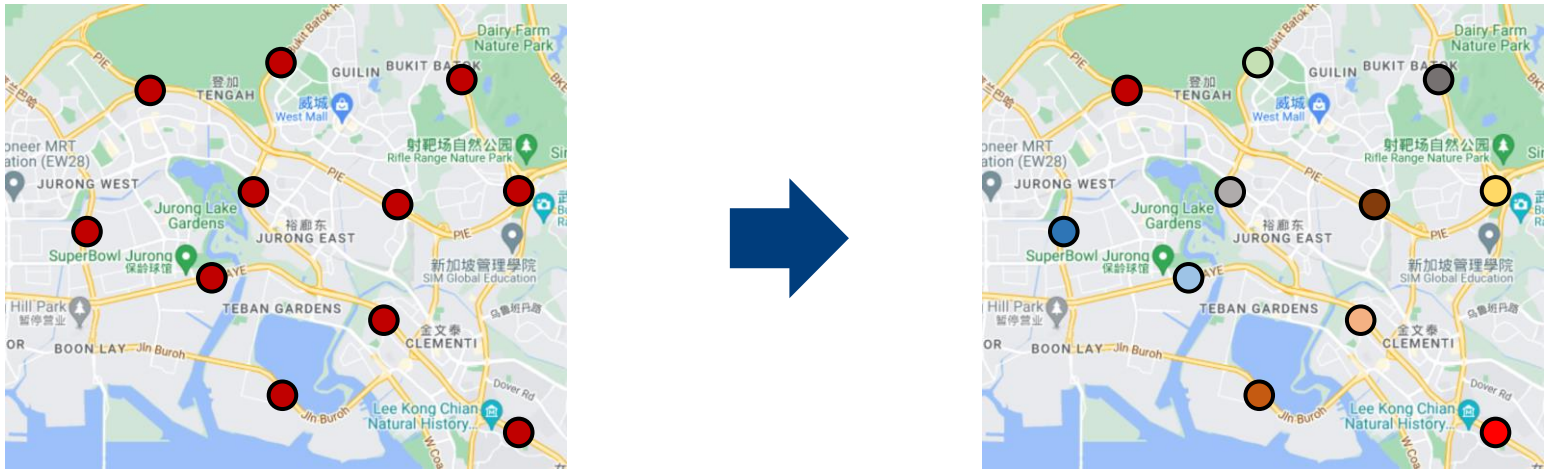
Existing Approaches

- Capturing ST dependencies is of great importance to traffic forecasting
  - Capturing spatial correlations by Graph Neural Networks (GNNs)
  - Learning temporal dependencies by RNNs or CNNs



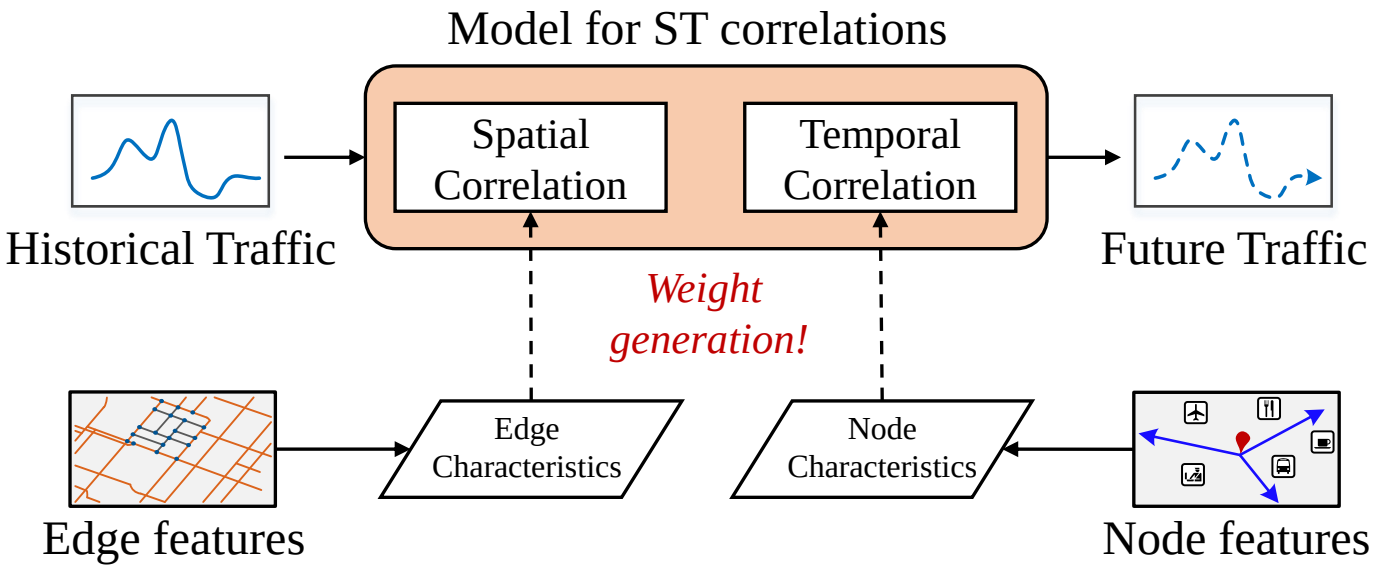
An Intuitive Idea

- Using non-shared modules to customize the model for each location
  - **Parameter explosion**: the parameter size will increase by #node times
  - Prone to overfitting, verified by empirical studies

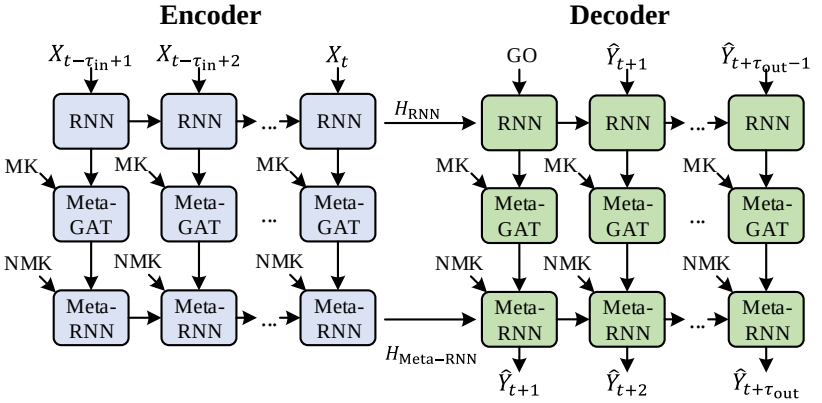


Our Intuition

- **Geospatial attributes** can reflect the characteristics of nodes and edges, and affect different kinds of spatio-temporal characteristics



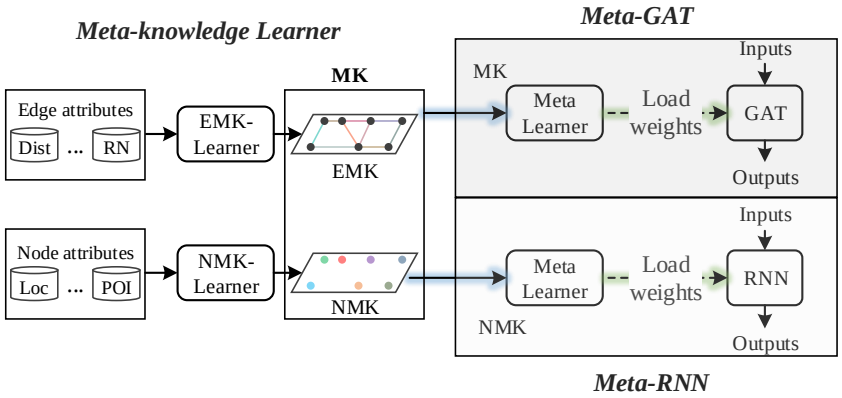
ST-MetaNet: Spatio-Temporal Meta Network



Recurrent Neural Network (RNN)
- Embedding the sequence of urban traffic.

Meta Graph Attention Network (Meta-GAT)
- Modeling diverse spatial correlations.

Meta Recurrent Neural Network (Meta-RNN)
- Modeling diverse temporal correlations.

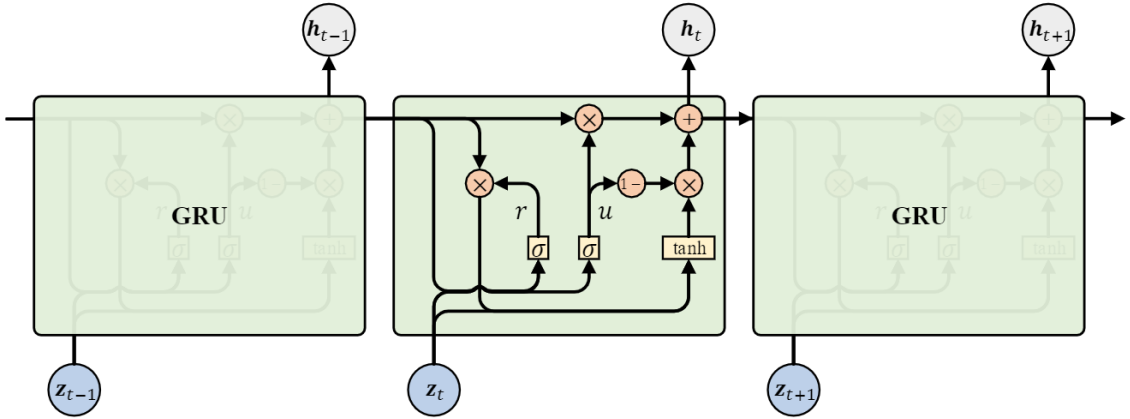


Meta-knowledge Learner
- Learning node & edge characteristics from geo-attributes.

Meta Learner
- Generating parameter weights in GAT and RNN.

Z. Pan, Y. Liang et al., Urban Traffic Prediction from Spatio-Temporal Data Using Deep Meta Learning. KDD 2019.

GRUs (a variant of RNNs)



$$u = \text{sigmoid}\left(\mathbf{W}_{u,i}\boldsymbol{z}_{t,i} + \mathbf{U}_{u,i}\boldsymbol{h}_{t-1,i} + b_{u,i}\right),$$

$$r = \text{sigmoid}\left(\mathbf{W}_{r,i}\boldsymbol{z}_{t,i} + \mathbf{U}_{r,i}\boldsymbol{h}_{t-1,i} + b_{r,i}\right),$$

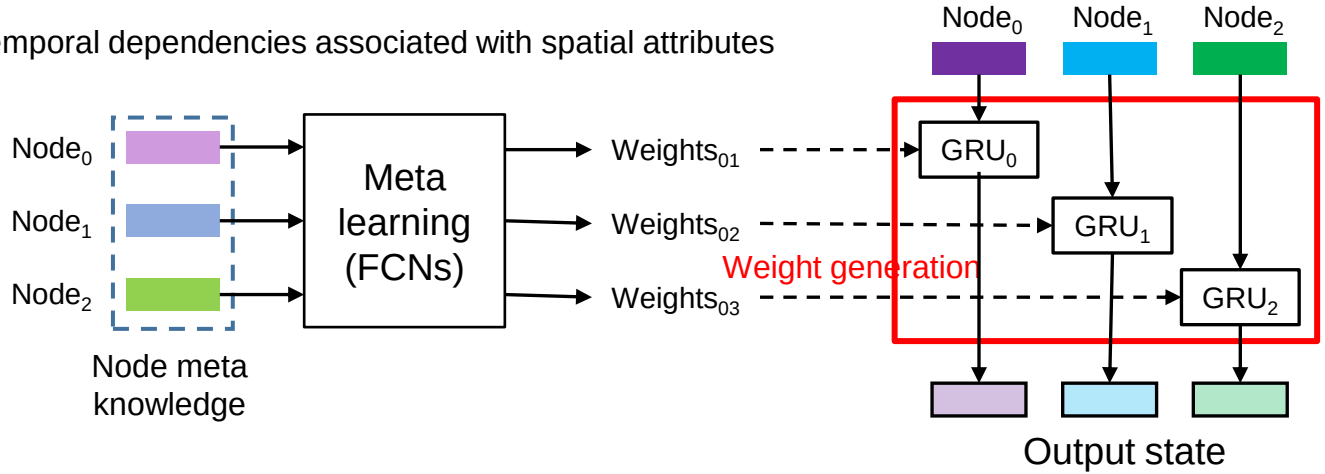
$$\boldsymbol{h}_{t,i} = u \circ \boldsymbol{h}_{t-1,i} + (1-u) \circ \tanh\left(\mathbf{W}_{h,i}\boldsymbol{z}_{t,i} + \mathbf{U}_{h,i}\left(r \circ \boldsymbol{h}_{t-1,i}\right) + b_{h,i}\right).$$

The parameters shared by all nodes for feature embedding

Meta-GRU

Meta-GRUs

- Modeling temporal dependencies associated with spatial attributes



$$u = \text{sigmoid}\left(\mathbf{W}_{u,i}\boldsymbol{z}_{t,i} + \mathbf{U}_{u,i}\boldsymbol{h}_{t-1,i} + b_{u,i}\right),$$

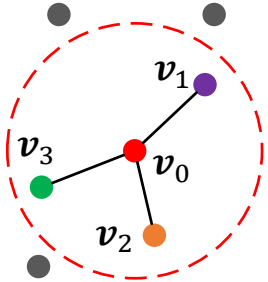
$$r = \text{sigmoid}\left(\mathbf{W}_{r,i}\boldsymbol{z}_{t,i} + \mathbf{U}_{r,i}\boldsymbol{h}_{t-1,i} + b_{r,i}\right)$$

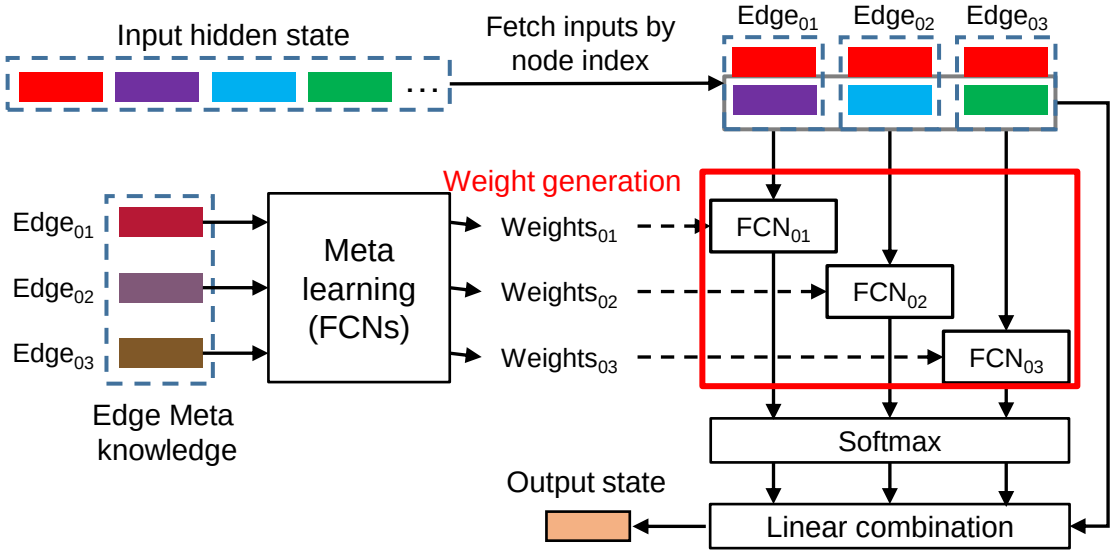
$$\boldsymbol{h}_{t,i} = u \circ \boldsymbol{h}_{t-1,i} + (1-u) \circ \tanh\left(\mathbf{W}_{h,i}\boldsymbol{z}_{t,i} + \mathbf{U}_{h,i}\left(r \circ \boldsymbol{h}_{t-1,i}\right) + b_{h,i}\right).$$

Meta Graph Attention Network (Meta-GAT)

Meta-GAT

- Modeling spatial correlation associated with spatial attributes





Computing attention weights

$$\boldsymbol{w}_{ij} = \text{LeakyReLU}(\mathbf{W}_{ij}[\boldsymbol{h}_i \parallel \boldsymbol{h}_j] + b_{ij})$$

Extracting edge-related meta knowledge

$$\text{MK}_{ij} = \text{NMK}(\boldsymbol{v}_i) \parallel \text{NMK}(\boldsymbol{v}_j) \parallel \text{EMK}(\boldsymbol{e}_{ij})$$

Network weight generation

$$\mathbf{W}_{ij} = \text{G}_{\mathbf{W}}(\text{MK}_{ij}), \quad b_{ij} = \text{G}_b(\text{MK}_{ij})$$

Normalization by softmax

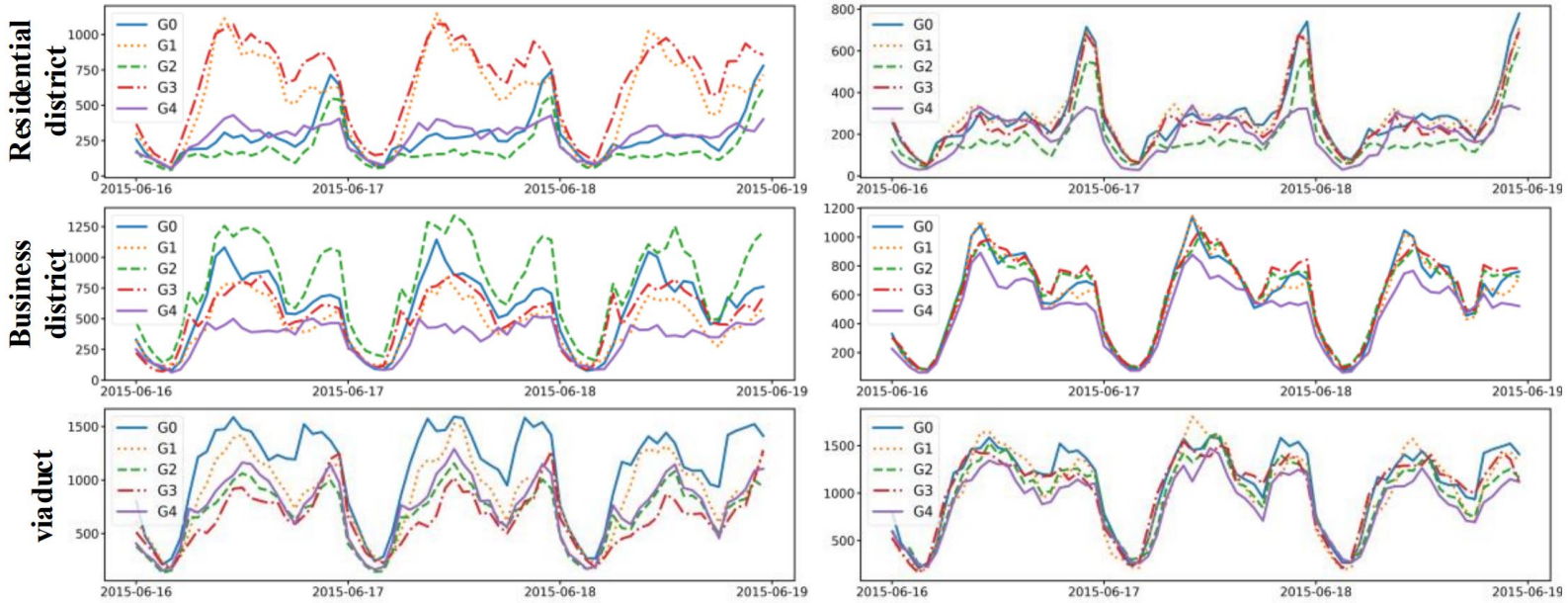
$$(1-\lambda_i)\boldsymbol{h}_i + \lambda_i \text{ReLU}(\sum_j \frac{\exp(\boldsymbol{w}_{ij})}{\sum_k \exp(\boldsymbol{w}_{ik})} \boldsymbol{h}_j)$$

Model Comparison

- We evaluate our model on two different tasks, predicting taxi flow in Beijing and forecasting traffic speed on highways of Los Angeles.
- Compared to SOTA, our method achieves lower error while using fewer parameters

Models			HA	ARIMA	GBRT	Seq2Seq	GAT-SeqSeq	SOTA	**ST-MetaNet**
Taxi flow	overall	MAE	26.2	40.0	28.8	21.3 ± 0.06	18.3 ± 0.13	18.7 ± 0.53	**16.9** ± 0.13
		RMSE	56.5	86.8	60.9	42.6 ± 0.14	35.6 ± 0.23	36.1 ± 0.59	**34.0** ± 0.25
	1h	MAE	26.2	27.1	22.3	17.8 ± 0.05	16.3 ± 0.12	16.8 ± 0.50	**15.0** ± 0.14
		RMSE	56.5	58.3	47.7	35.1 ± 0.07	31.9 ± 0.21	31.9 ± 0.69	**29.9** ± 0.08
	2h	MAE	26.2	41.2	29.8	22.0 ± 0.06	18.7 ± 0.12	18.9 ± 0.57	**17.3** ± 0.14
		RMSE	56.5	77.0	62.6	43.6 ± 0.16	36.3 ± 0.20	36.4 ± 0.71	**34.7** ± 0.25
	3h	MAE	26.2	51.8	34.2	24.2 ± 0.09	19.9 ± 0.14	20.3 ± 0.52	**18.4** ± 0.10
		RMSE	56.5	108.0	70.3	48.1 ± 0.20	38.4 ± 0.30	39.5 ± 0.46	**37.1** ± 0.41
	# params		-	-	-	333k	407k	445k	268k
Traffic Speed	overall	MAE	4.79	4.03	3.85	3.55 ± 0.01	3.28 ± 0.00	3.10 ± 0.01	**3.05** ± 0.02
		RMSE	8.72	7.94	7.48	7.27 ± 0.01	6.66 ± 0.01	6.31 ± 0.03	**6.25** ± 0.02
	15min	MAE	4.79	3.27	3.16	2.98 ± 0.01	2.83 ± 0.01	2.75 ± 0.01	**2.68** ± 0.02
		RMSE	8.72	6.14	6.05	5.88 ± 0.01	5.47 ± 0.01	5.33 ± 0.02	**5.15** ± 0.02
	30min	MAE	4.79	3.99	3.85	3.57 ± 0.01	3.31 ± 0.00	3.14 ± 0.01	**3.09** ± 0.03
		RMSE	8.72	7.78	7.50	7.26 ± 0.01	6.68 ± 0.00	6.45 ± 0.04	**6.28** ± 0.02
	60min	MAE	4.79	5.18	4.85	4.38 ± 0.01	3.93 ± 0.01	**3.60** ± 0.02	**3.60** ± 0.04
		RMSE	8.72	10.10	9.08	8.88 ± 0.02	8.03 ± 0.02	7.65 ± 0.06	**7.52** ± 0.01
	# params		-	-	-	81k	113k	373k	85k

Case Study on Learned Location Embedding



(a) Seq2seq integrated with graph attention

(b) The proposed model

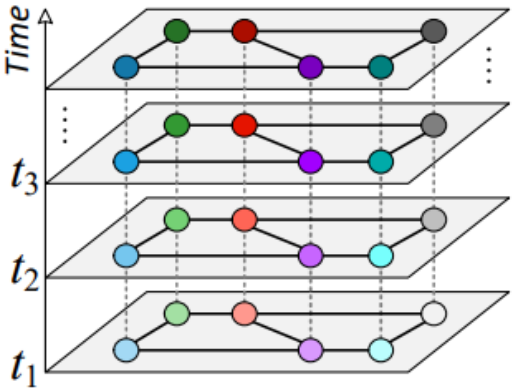
Advanced Learning Frameworks

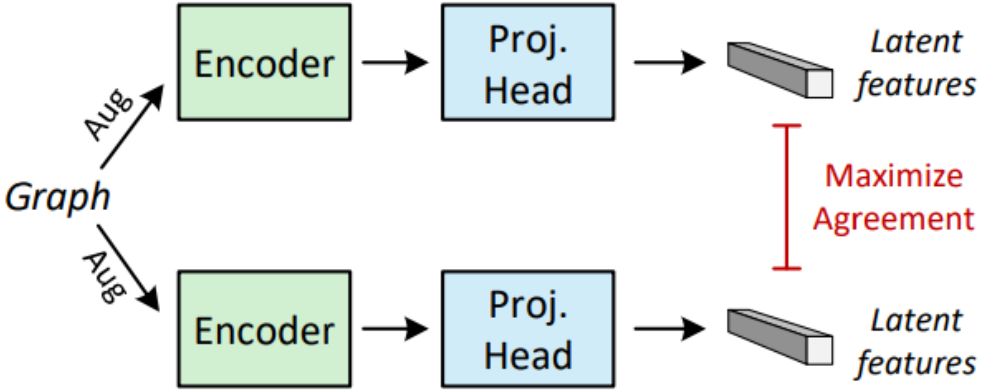
- Adversarial Learning
- Meta Learning
- Self-Supervised Learning
- Continuous Spatio-Temporal modeling
- Physics-Informed Learning
- Transfer Learning

Spatio-Temporal Graph Contrastive Learning

SIGSPATIAL 2022

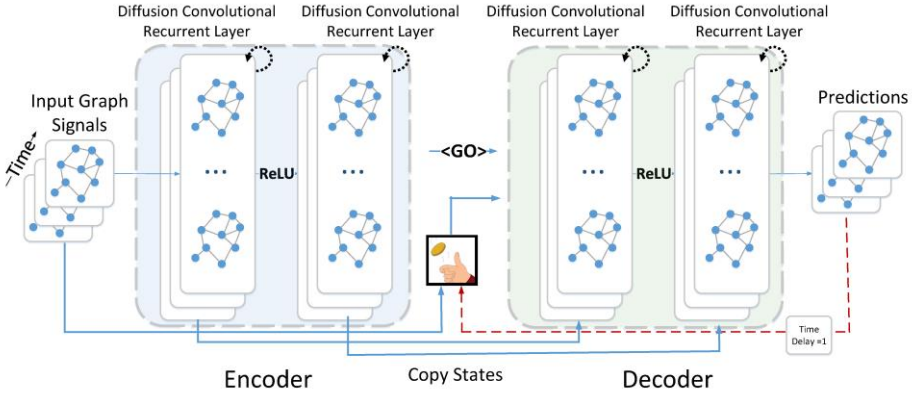
X. Liu et al. When Do Contrastive Learning Signals Help Spatio-Temporal Graph Forecasting?, SIGSPATIAL 2022



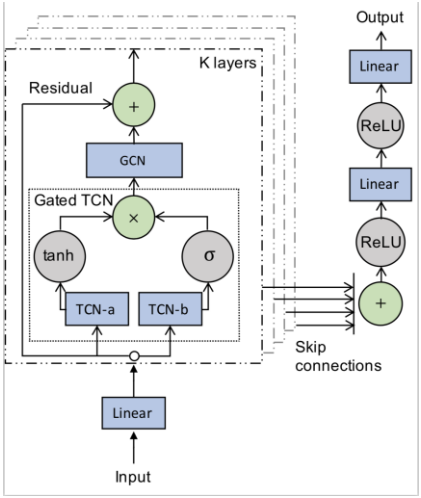


Spatio-Temporal Graph Neural Networks

- STGNNs are the modern tools for modeling STG, e.g., forecasting
  - Learning spatial relations via GNNs or Attention
  - Modeling temporal dependencies with RNNs, Attention, or TCNs



DCRNN



Graph WaveNet

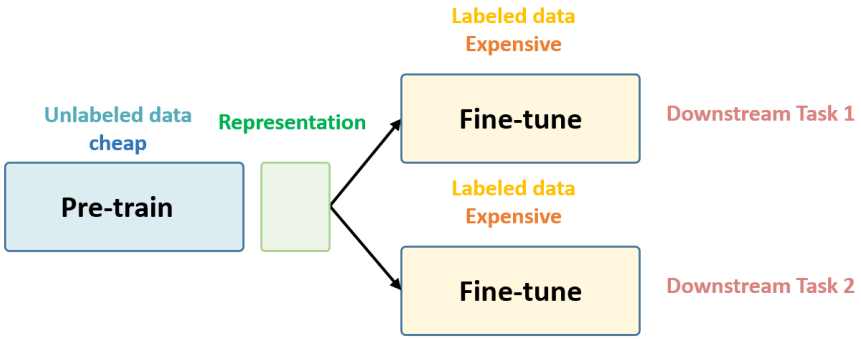
Challenges

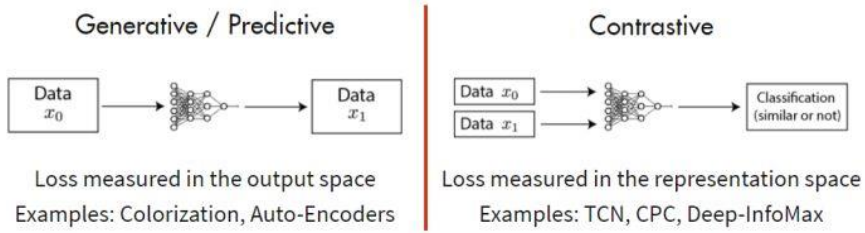
- Tremendous efforts have been made to design sophisticated architectures to capture complex spatio-temporal dependencies
- However, data scarcity is a crucial issue that may hinder the recent improvements on STG forecasting

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

Self-Supervised Learning

- Meanwhile, self-supervised learning have demonstrated great promise in a series of tasks on graphs.
  - It derives supervisory signals from the data itself, usually exploiting the underlying structure of the data.
  - Most of the self-supervised methods are based on Contrastive Learning (CL).

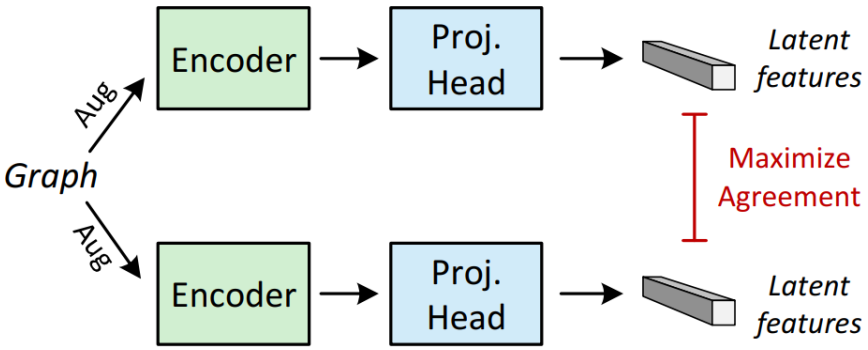


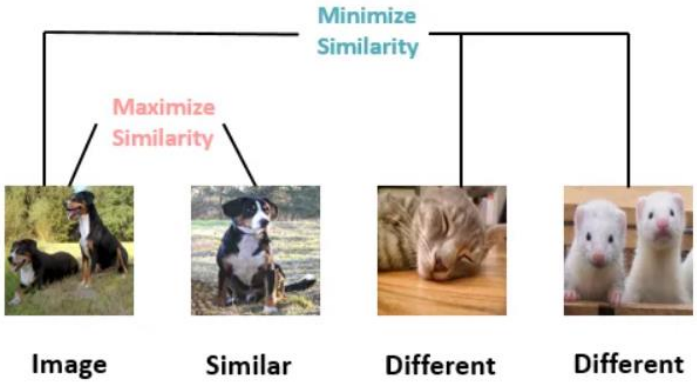


Contrastive Learning

- Contrastive learning is used to learn the general features of a dataset without labels by teaching the model which data points are similar or different
  - Examples: SimCLR, MoCo, GraphCL

$$\mathcal{L}_{cl} = \frac{1}{M}\sum_{i=1}^{M} -\log \frac{\exp(\text{sim}(\mathbf{z}_i', \mathbf{z}_i'')/\tau)}{\sum_{j=1, j\neq i}^{M} \exp(\text{sim}(\mathbf{z}_i', \mathbf{z}_j'')/\tau)}$$

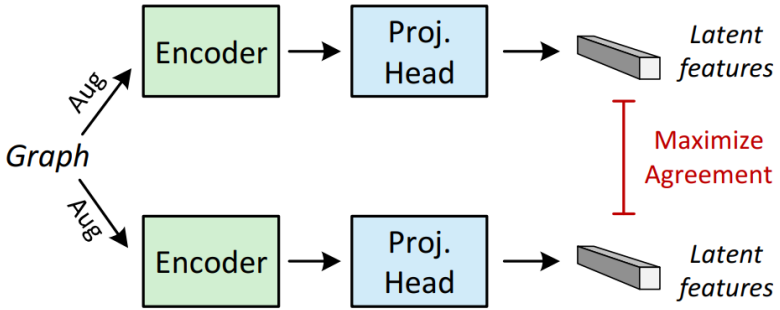




Our Contributions

- In light of the success of contrastive learning, we present the first systematic study to answer a key question

Can we leverage the additional self-supervised signals derived from CL to alleviate data scarcity, so as to benefit STG forecasting?



$$\mathcal{L}_{cl} = \frac{1}{M}\sum_{i=1}^{M} -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_i'')/\tau)}{\sum_{j=1, j\neq i}^{M} \exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_j'')/\tau)}$$

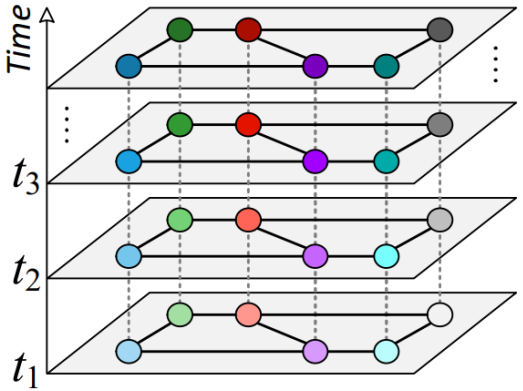
Our Contributions

- We give an affirmative answer by identifying and addressing four essential questions in a unified framework.
  - Training schemes (Q1)
  - What & how to contrast (Q2)
  - Data augmentation (Q3)
  - Negative filtering (Q4)
- We propose a model-agnostic framework called ***STGCL*** to incorporate contrastive learning into current STGNNs for STG modeling

Notations

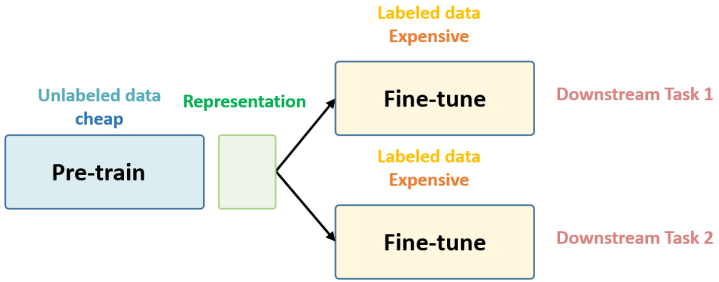
- A sensor network is denoted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$
- The observation at a time step t is $\mathbf{X}^t \in \mathbb{R}^{N \times F}$
- STG forecasting problem

$$\mathcal{G} : [\mathbf{X}^{(t-S):t}; G] \xrightarrow{f} \mathbf{Y}^{t:(t+T)}$$

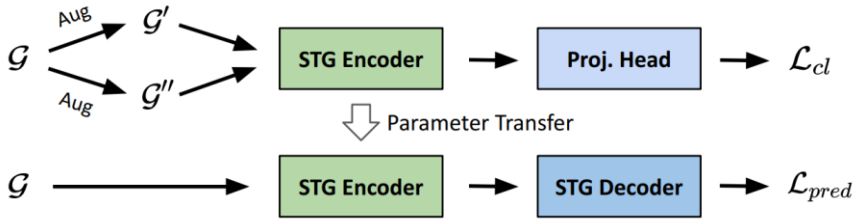


Training Schemes (Q1)

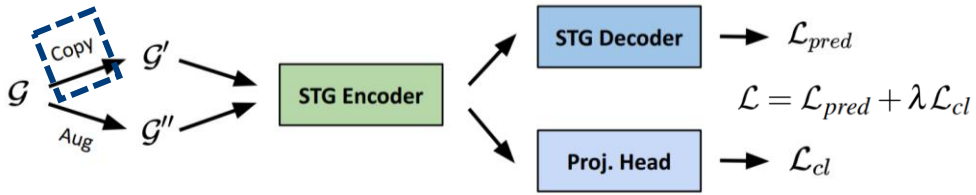
- **Q1**: *What is the appropriate training scheme when integrating contrastive learning with STG forecasting?*
- We identify two candidate schemes to incorporate contrastive learning
  - Pretraining & Fine-tuning
  - Joint learning



Standard CL for CV, Graphs



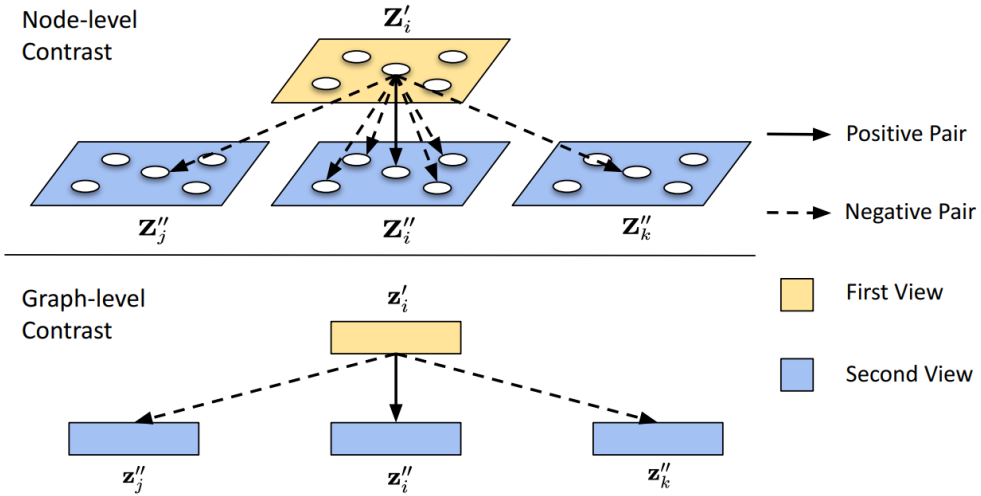
(a) Pretraining & Fine-tuning



(b) Joint Learning

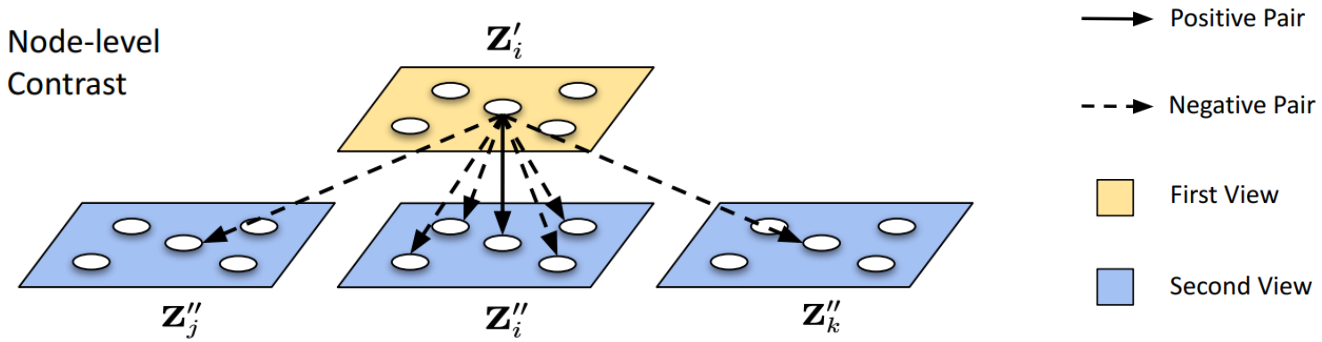
What & How to Contrast (Q2)

- **Q2**: *Which level should we select as the object of contrastive learning?*
- We propose two feasible designs with different rationales
  - Node-level: more fine-grained and matches to the level of the forecasting task
  - Graph-level: considers global knowledge of the whole graph



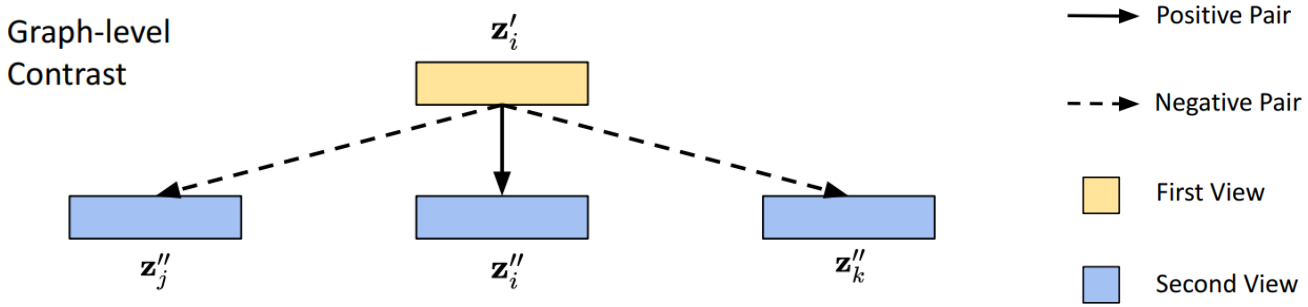
What & How to Contrast (Q2)

- Suppose we have a batch of M STG with N nodes
- Node-level contrast – node as the object
  - Full spatio-temporal contrast induces $\mathcal{O}(M^2N^2)$ complexity
  - We thus factorize the spatio-temporal contrast into spatial and temporal domains, leading to $\mathcal{O}(M+N)$ complexity



What & How to Contrast (Q2)

- Graph-level contrast – graph as the object
  - Encouraging model to distinguish the spatio-temporal patterns of different inputs
  - Summarizing the representation of STG using a **readout** function
  - Can be interpreted as a sample-level contrast



Data Augmentation (Q3)

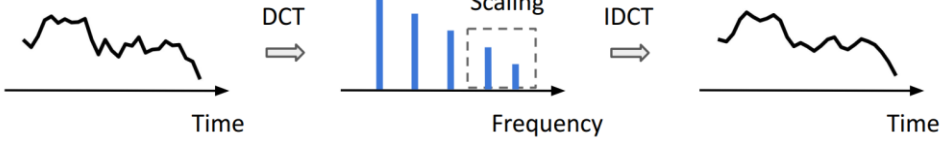
- **Q3**: *How should we perform data augmentation to generate a positive pair?*
- We propose four methods perturb data in different aspects: graph structure, time domain, and frequency domain

$$\mathbf{A}'_{ij} = \begin{cases} \mathbf{A}_{ij}, & \text{if } \mathbf{M}_{ij} \geqslant r_{em} \\ 0, & \text{otherwise} \end{cases}$$

Edge Masking

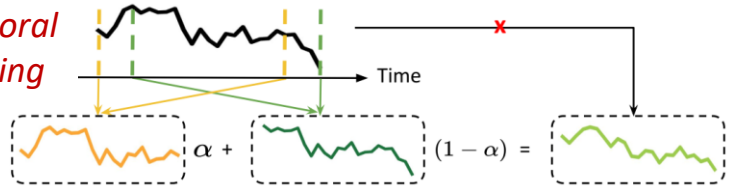
$$\mathbf{P}_{ij}^{(t-S):t} = \begin{cases} \mathbf{X}_{ij}^{(t-S):t}, & \text{if } \mathbf{M}_{ij} \geqslant r_{im} \\ -1, & \text{otherwise} \end{cases}$$

Input Masking



Input Smoothing

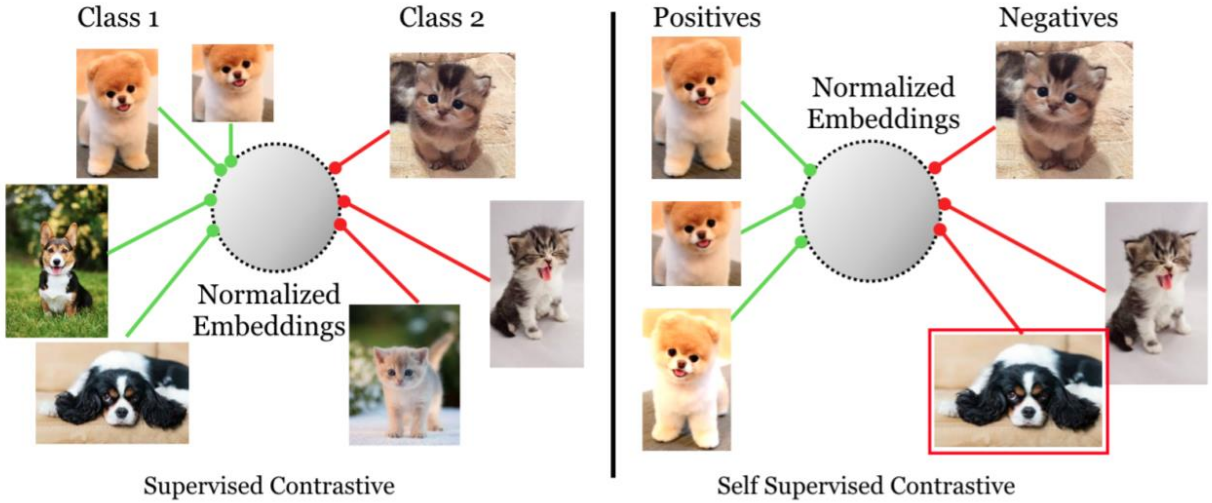
Temporal Shifting



$$\mathbf{P}^{(t-S):t} = \alpha \mathbf{X}^{(t-S):t} + (1-\alpha)\mathbf{X}^{(t-S+1):(t+1)}$$

Negative Filtering (Q4)

- **Q4**: *Given an anchor, should all other objects be considered as negatives? If not, how should we filter out unsuitable negatives?*
- Treating all other objects as negatives ignores instances' semantic similarity



Negative Filtering (Q4)

- **Q4**: *Given an anchor, should all other objects be considered as negatives? If not, how should we filter out unsuitable negatives?*
- Challenge: no available semantic labels in STG
- We propose to filter out unsuitable negatives based on the unique properties of STG data

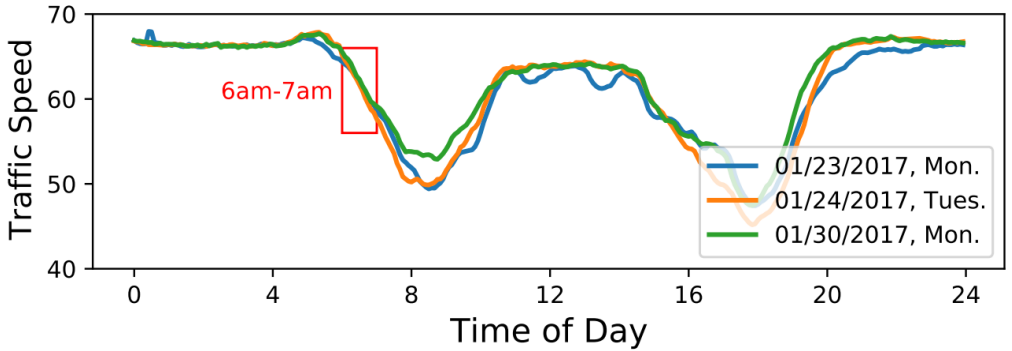
$$\mathcal{L}_{cl} = \frac{1}{M}\sum_{i=1}^{M} -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_i'')/\tau)}{\sum_{j=1, j\neq i}^{M} \exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_j'')/\tau)} \quad \Rightarrow \quad \frac{1}{M}\sum_{i=1}^{M} -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_i'')/\tau)}{\sum_{j\in\chi_i} \exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_j'')/\tau)}$$

A set of acceptable negatives for the i-th sample

How to filter out useless negatives?

Negative Filtering (Q4)

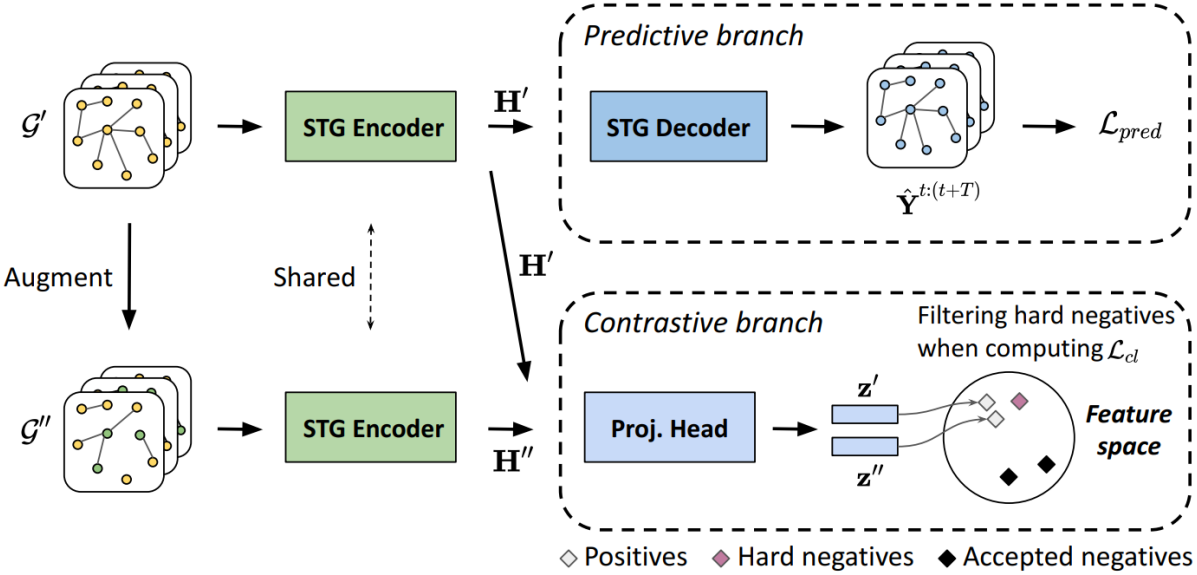
- **Temporal negative filtering**: exclude unsuitable negatives by utilizing ubiquitous temporal properties of STG – closeness and periodicity



- **Spatial negative filtering**
  - Utilizing the information from the predefined adjacency matrix
  - Specifically, the first-order neighbors of each node are excluded during contrastive loss
  - Note that this part is only applicable for node-level contrast

Sample Implementation

- We give a sample implementation to link all the introduced techniques and to facilitate understanding of our framework.
  - Joint learning + Graph-level contrast
  - Acceleration



Experiments

- Task setting: using 1-hour historical data to predict the next one hour
- Base models as ST encoder
  - CNN-based: Graph WaveNet (GWN), MTGNN
  - RNN-based methods: DCRNN, AGCRN
- Datasets

Datasets	#Nodes	#Edges	#Instances	Interval
PEMS-04	307	209	16,992	5 min
PEMS-08	170	137	17,856	5 min

Empirical Results (Q1)

- Training schemes evaluation.
  - P&F: pretrain and fine-tune. JL: Joint learning.
  - Observation: Joint learning is the preferable scheme

Methods	PEMS-04	PEMS-08
GWN	19.33±0.11	14.78±0.03
w/ P&F-node	20.22±0.22	15.37±0.08
w/ P&F-graph	20.67±0.13	15.86±0.15
w/ JL-node	18.89±0.05	14.63±0.07
w/ JL-graph	**18.88±0.04**	**14.61±0.03**
AGCRN	19.39±0.03	15.79±0.06
w/ P&F-node	19.70±0.07	17.21±0.14
w/ P&F-graph	20.39±0.10	17.92±0.10
w/ JL-node	19.32±0.06	15.78±0.09
w/ JL-graph	**19.13±0.05**	**15.62±0.07**

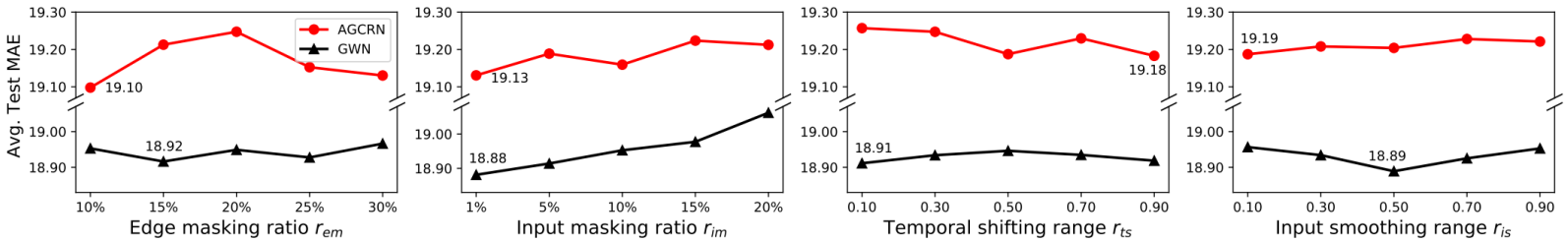
Empirical Results (Q2)

- The reasons why the graph-level method surpasses the node-level
  - Nodes are enforced to perform forecasting/contrastive tasks simultaneously, which is non-trivial
  - While for the graph level, it might be easier to distinguish patterns at the graph level and it is important to provide global information to each node

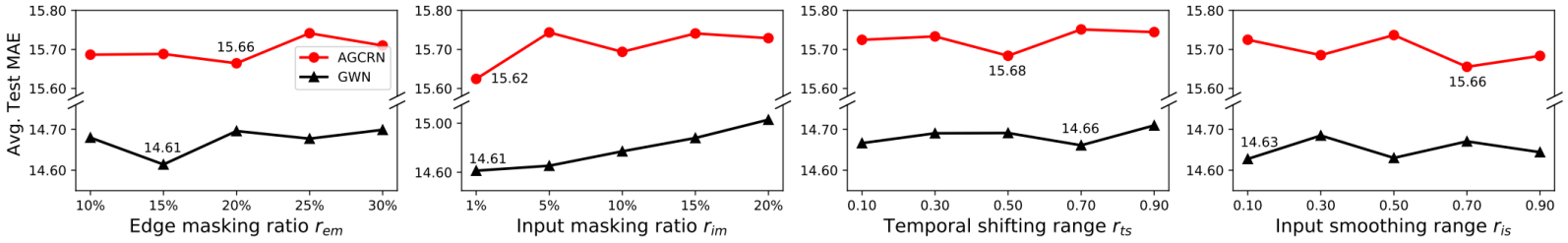
Methods	PEMS-04				PEMS-08			
	15 min	30 min	60 min	Δ	15 min	30 min	60 min	Δ
GWN	18.20±0.09	19.32±0.13	21.10±0.18	–	13.80±0.05	14.75±0.04	16.39±0.09	–
w/ JL-node	17.97±0.05	18.90±0.07	**20.38±0.07**	-1.37	13.67±0.06	14.63±0.08	16.14±0.13	-0.50
w/ JL-graph	**17.93±0.04**	**18.87±0.04**	20.40±0.09	**-1.42**	**13.67±0.04**	**14.61±0.03**	**16.09±0.05**	**-0.57**
MTGNN	18.32±0.05	19.10±0.05	20.39±0.09	–	14.36±0.06	15.34±0.10	16.91±0.16	–
w/ JL-node	18.03±0.02	18.79±0.06	19.94±0.03	-1.05	14.05±0.05	14.94±0.04	16.38±0.09	-1.24
w/ JL-graph	**17.99±0.03**	**18.72±0.05**	**19.88±0.07**	**-1.22**	**14.04±0.05**	**14.90±0.05**	**16.23±0.08**	**-1.44**
DCRNN	19.99±0.11	22.40±0.19	27.15±0.35	–	15.23±0.15	16.98±0.25	20.27±0.41	–
w/ JL-node	19.94±0.08	22.38±0.14	27.15±0.26	-0.07	**15.15±0.05**	**16.85±0.11**	**20.02±0.23**	**-0.46**
w/ JL-graph	**19.82±0.08**	**22.07±0.12**	**26.51±0.21**	**-1.14**	15.19±0.11	16.89±0.19	20.09±0.34	-0.31
AGCRN	18.53±0.03	19.43±0.06	20.72±0.03	–	14.58±0.07	15.71±0.07	17.82±0.11	–
w/ JL-node	18.46±0.04	19.37±0.07	20.69±0.11	-0.16	14.55±0.06	15.69±0.10	17.82±0.14	-0.05
w/ JL-graph	**18.31±0.04**	**19.17±0.06**	**20.39±0.03**	**-0.81**	**14.51±0.05**	**15.56±0.06**	**17.51±0.10**	**-0.53**

Empirical Results (Q3)

- Effects of different data augmentation methods
  - Observation: STGCL is not sensitive to augmentation methods



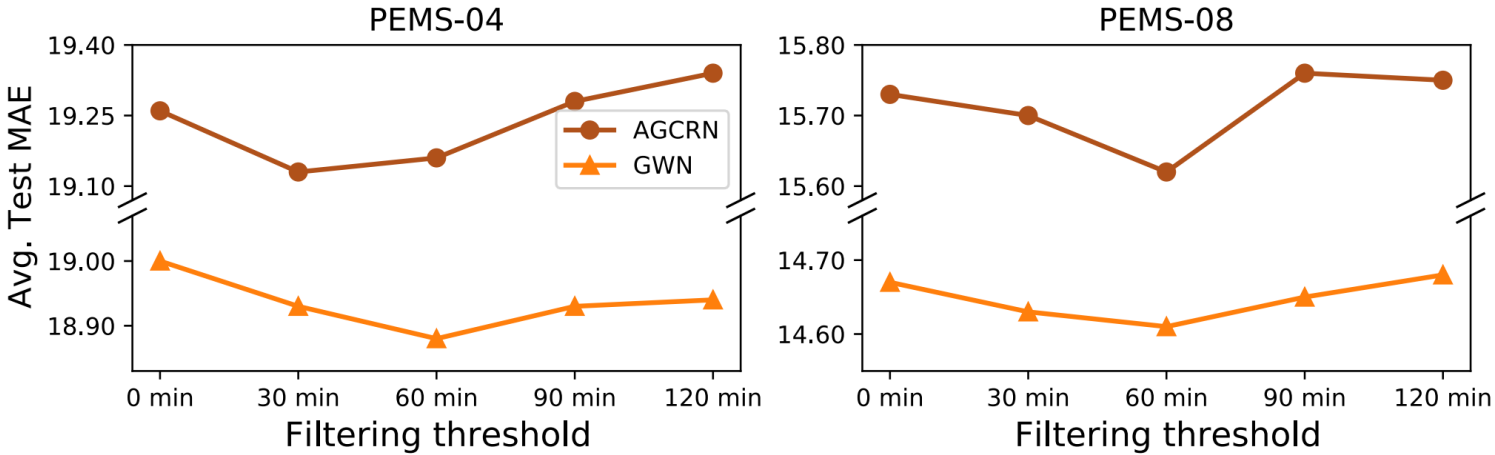
(a) PEMS-04



(b) PEMS-08

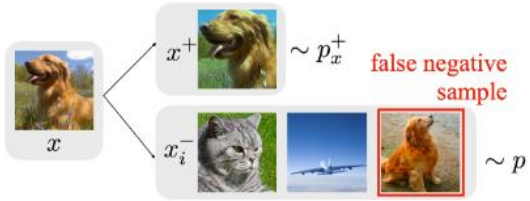
Empirical Results (Q4)

- Effects of temporal negative filtering



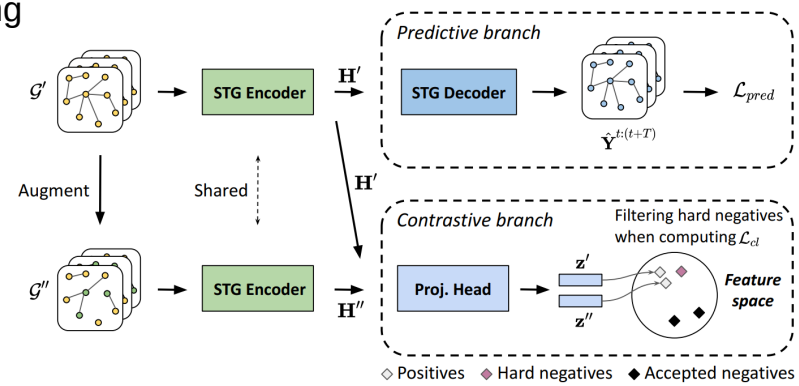
Further Thinking

- Key ingredients in CV-based contrastive learning
  - Heavy data augmentation
  - Large batch size
  - Hard negative mining, e.g., by labels



- In contrast, STG-based contrastive learning has different insights
  - Moderate data augmentation is preferable
  - As opposite to CV tasks, more negative samples may not help
  - Using spatio-temporal prior knowledge for negative filtering

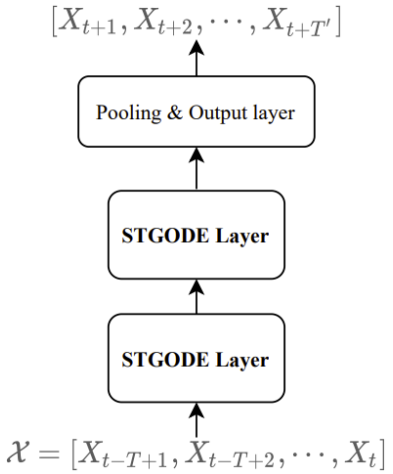
- More discussion
  - Prediction is indeed a pretext task
  - Representation learning for STG is challenging
    - ➢ No semantic labels to represent global information
    - ➢ Hard to reduce dimensions, e.g., pooling



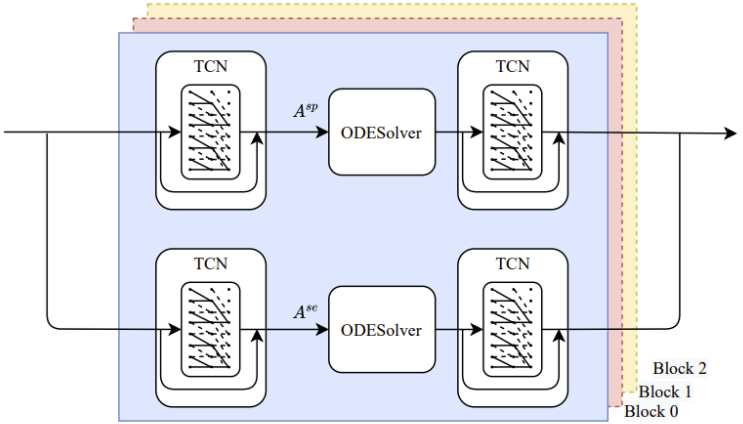
Advanced Learning Frameworks

- Adversarial Learning
- Meta Learning
- Self-Supervised Learning
- Continuous Spatio-Temporal modeling
- Physics-Informed Learning
- Transfer Learning

Continuous Spatio-Temporal modeling



(a) Framework



(b) STGODE Layer

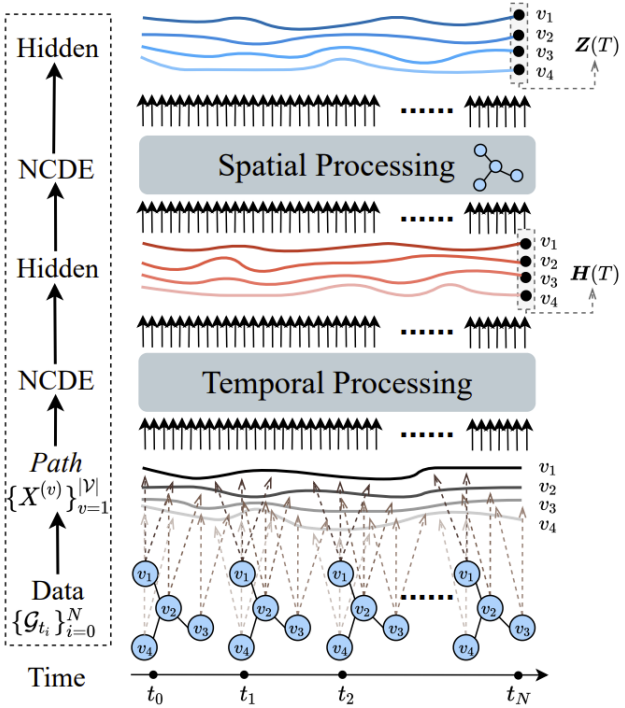
$$\mathcal{H}(t) = ODESolve\left(\frac{d\mathcal{H}(t)}{dt}, \mathcal{H}_0, t\right), \tag{24}$$

where

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

Fang et al., Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. KDD 2021.

Continuous Spatio-Temporal modeling



Choi et al., Neural Controlled Differential Equations for Traffic Forecasting. AAAI 2022.

Presented Papers

	UrbanFM: Inferring Fine-Grained Urban Flows		KDD	2019	NUS	Gangyong Zhu
	SPATIO-TEMPORAL FEW-SHOT LEARNING VIA DIFFUSIVE NEURAL NETWORK GENERATION	Link	ICLR	2024	Tsinghua University	Pei LIU
	Towards Unifying Diffusion Models for Probabilistic Spatio-Temporal Graph Learning	Link	SIGSPATIAL	2024	HKUST(GZ)	Yongzi Yu

Q&A

Thanks!

CityMind Lab

amazon Microsoft Tencent Alibaba.com 京东 JD.COM CAINIAO 菜鸟