Deep Learning for Human Mobility Analytics

-- L9-Advanced Topics in Human Mobility Analytics I

Yuxuan Liang (梁宇轩)

INTR & DSA Thrust

yuxuanliang@hkust-gz.edu.cn

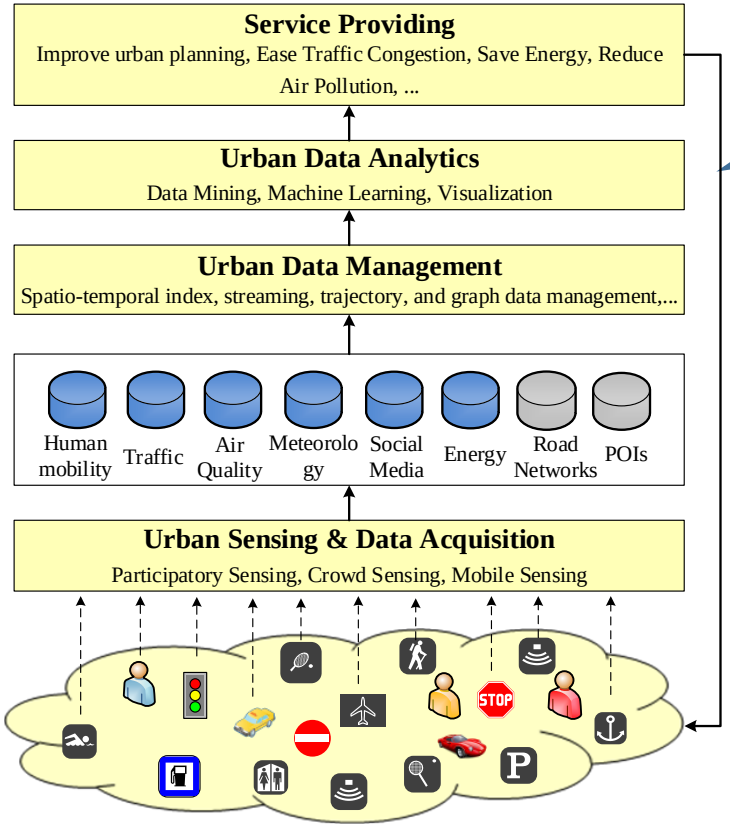
香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

Objectives of this Course

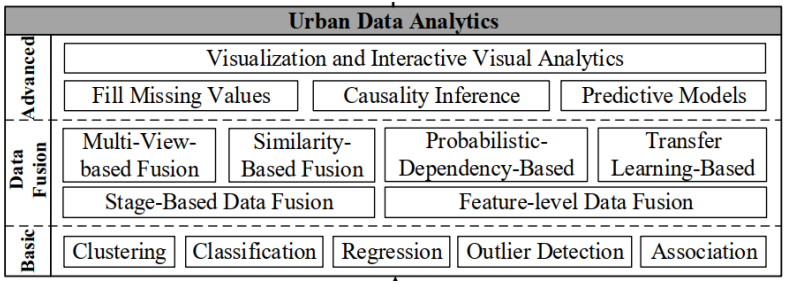
To introduce

- Transfer learning for ST data
- Self-supervised learning for ST data
  - Contrastive learning for ST data
  - Generative modeling for ST data
- Diffusion models for ST data

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

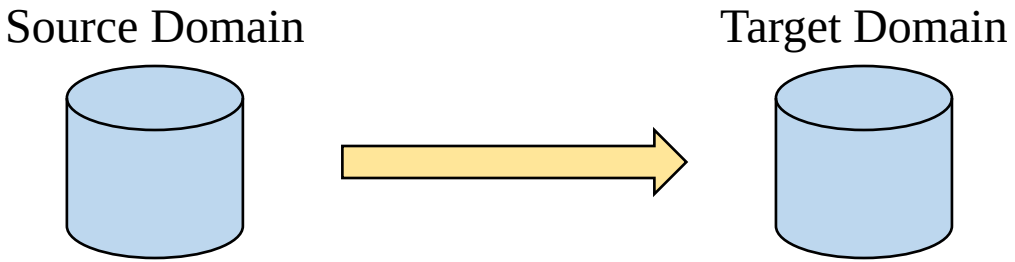


Outline

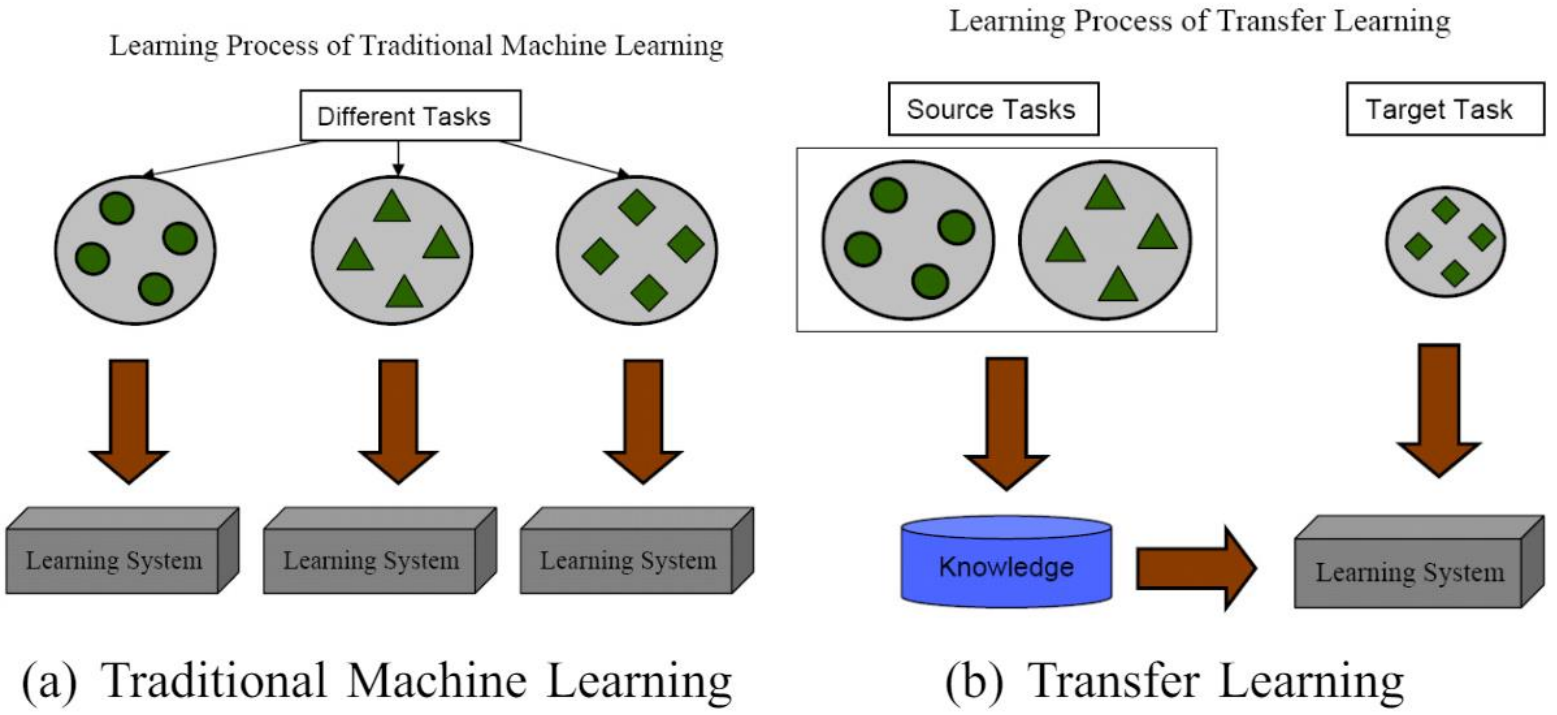
- Transfer learning for ST data
- Self-supervised learning for ST data
  - Contrastive learning for ST data
  - Generative modeling for ST data
- Diffusion models for ST data

What is Transfer Learning?

- Motivation for transfer learning
  - The goal is to transfer knowledge gathered from previous experience.
  - Also called Inductive Transfer or Learning to Learn.
- Once a predictive model is built, there are reasons to believe the model will cease to be valid at some point in time.

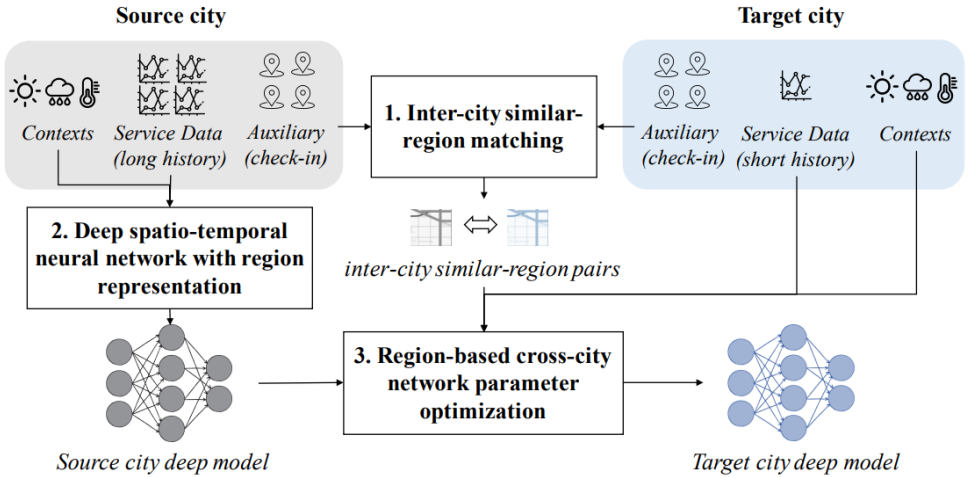


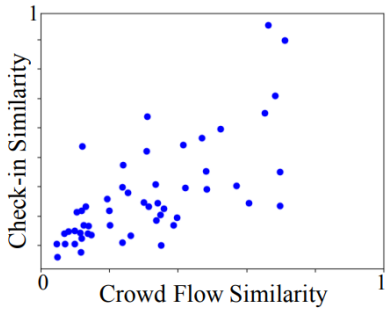
Traditional vs. Transfer Learning



Transfer Learning for Deep Spatio-Temporal Prediction

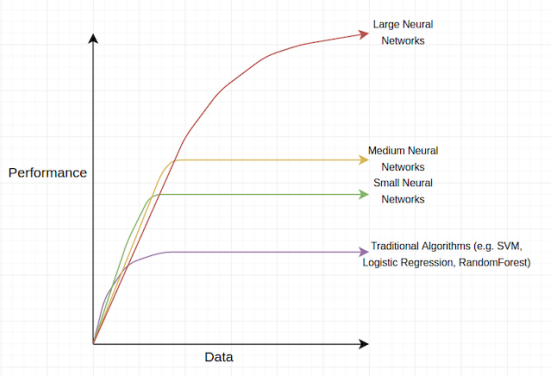
IJCAI 2019





Data Scarcity in Spatio-Temporal Domains

- Machine learning relies on big data
  - More data leads to better performance
- Data scarcity is common in spatio-temporal domains, e.g., in cities
  - Building new cities
  - Planning new urban services
- How to mitigate lack of data for spatio-temporal data mining?

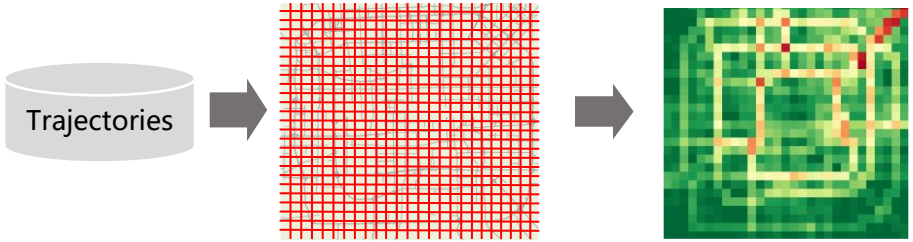


Solution: Transfer Learning

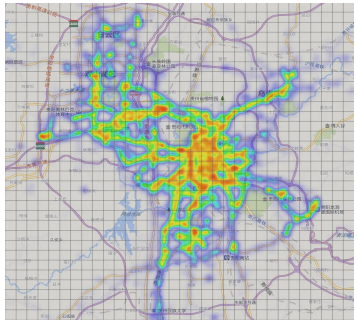
- Transfer learning:
  - Key idea: Borrow knowledge from different but related tasks
  - Effective in visual recognition, text mining, etc.
    - Pre-training & fine-tuning in computer vision
    - Cross-domain sentiment classification

Problem Statement

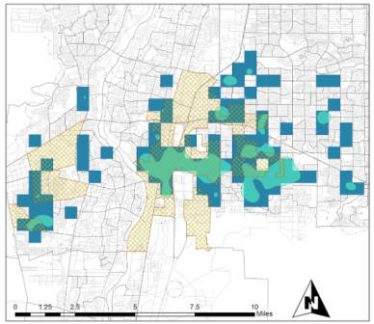
- We partition an area of interest (e.g., a metropolitan) evenly into grid cells, leading to an image-like data format called **ST grid**
  - A pixel → A region
  - RGB → Observations / Attributes



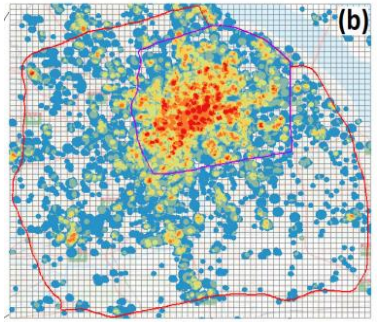
- Real-world examples



Taxi flows



Crime hotspots

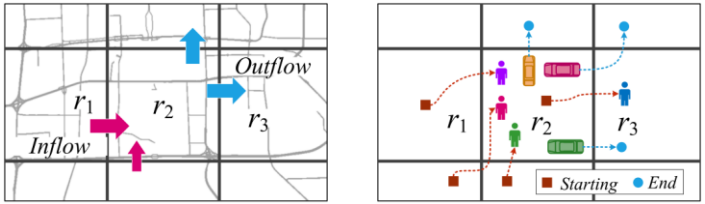


Bike-sharing demands

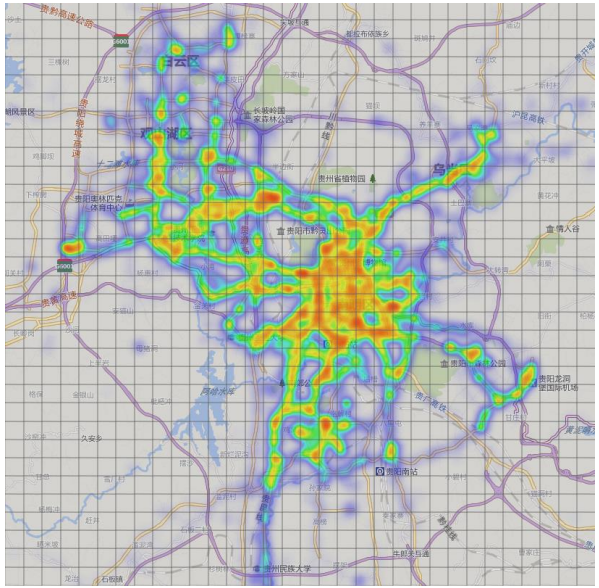
Application: Citywide Crowd Flow Forecasting

Grid-based citywide crowd flow prediction

- Predicting the inflow/outflow of every region in several hours



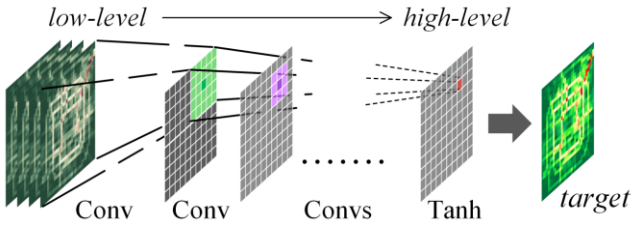
- It can provide insights to
  - Traffic control, risk assessment
- Challenges
  - Complex ST dependencies
  - External factor influence

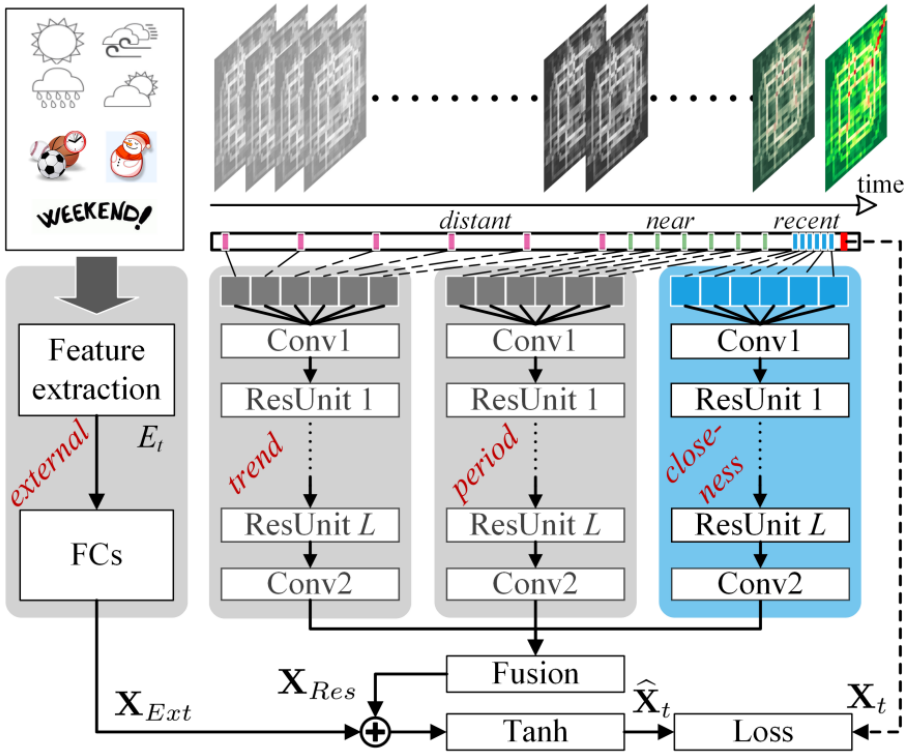


Y. Liang, R. Zimmermann et al., Fine-grained Urban Flow Prediction. WWW 2021.

ST-ResNet

- Temporal dependencies
  - Distant
  - Near
  - Recent
- Spatial dependencies





J. Zhang et al., Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. AAAI 2017.

Problem Statement

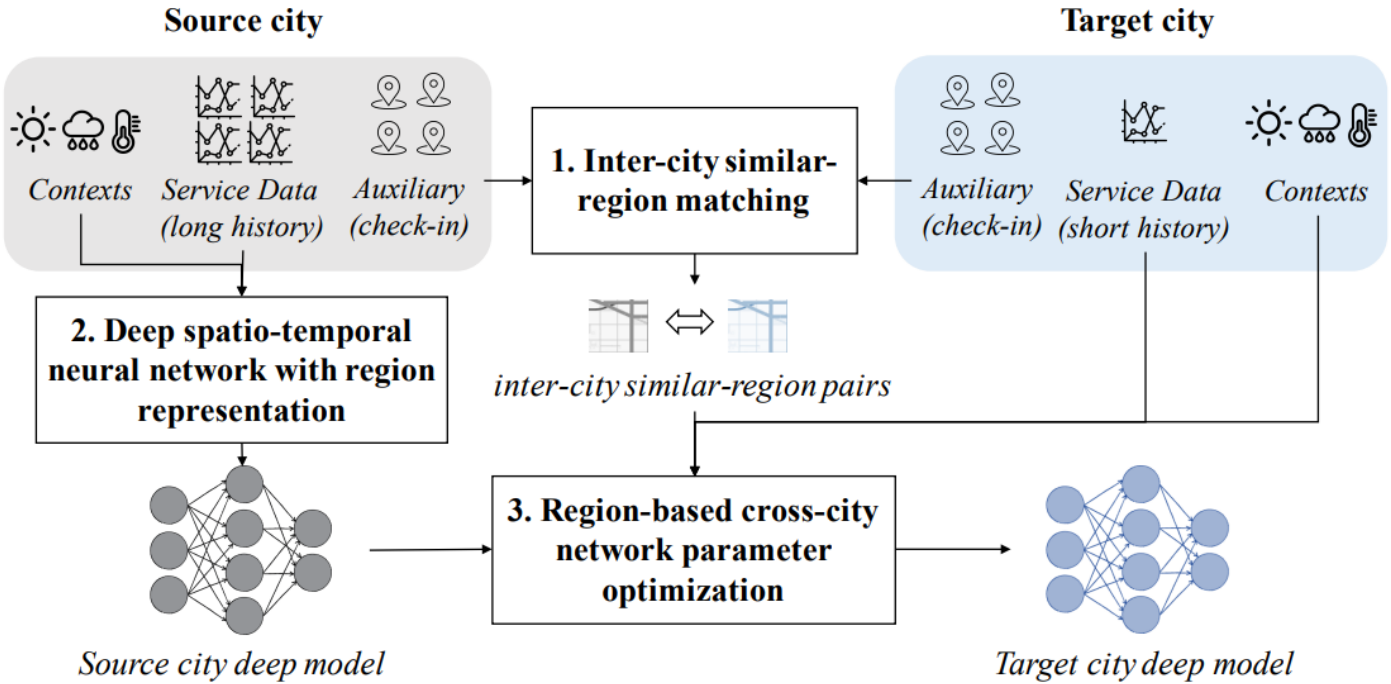
Given the little service data in target city $\mathcal{D}$ and rich service data in source city $\mathcal{D}'$, we aim to learn a function f to predict the citywide service data in the target city $\mathcal{D}$ at the next time-stamp $t_c + 1$:

$$\min_{f} \quad error(\widetilde{\mathcal{S}}_{t_c+1,\mathcal{D}}, \mathcal{S}_{t_c+1,\mathcal{D}}) \tag{5}$$

$$\text{where} \quad \widetilde{\mathcal{S}}_{t_c+1,\mathcal{D}} = f(\mathbb{S}_{\mathcal{D}}, \mathbb{S}_{\mathcal{D}'}), \quad |\mathbb{T}_{\mathcal{D}}| \ll |\mathbb{T}_{\mathcal{D}'}| \tag{6}$$

error metric may be mean absolute error, root mean squared error, etc., according to the real application requirement.

Framework


Source city
Target city
Contexts
Service Data (long history)
Auxiliary (check-in)
1. Inter-city similar-region matching
Auxiliary (check-in)
Service Data (short history)
Contexts
2. Deep spatio-temporal neural network with region representation
inter-city similar-region pairs
3. Region-based cross-city network parameter optimization
Source city deep model
Target city deep model

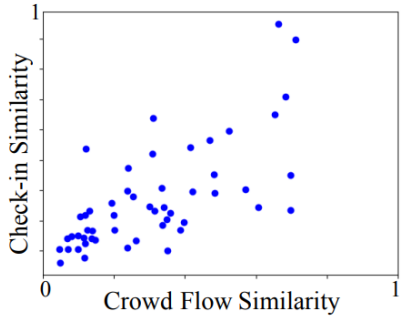
Inter-City Similar Region Matching

- Matching with a Short Period
  - For each target region, they choose the source region with the largest correlation value

$$
\begin{aligned}
&\mathcal{M}(r) = r^*, && r \in \mathbb{C}_{\mathcal{D}}, r^* \in \mathbb{C}_{\mathcal{D}'} \\
&\rho_{r,r^*} \geq \rho_{r,r'}, && \forall r' \in \mathbb{C}_{\mathcal{D}'} \\
&\rho_{r,r^*} = corr(\{s_{r,t}\}, \{s_{r^*,t}\}), && r \in \mathbb{C}_{\mathcal{D}}, r^* \in \mathbb{C}_{\mathcal{D}'}, t \in \mathbb{T}_{\mathcal{D}}
\end{aligned}
$$

Inter-City Similar Region Matching

- Matching with a Long Period of Auxiliary Data
  - Due to data scarsity in the target city, the above correlation similarity between a source region and a target region may not be very reliable
  - In reality, sometimes we can find another openly-accessible auxiliary data that correlates with the service data, which may help calculate the inter-city region similarity more robustly, such as check-in data

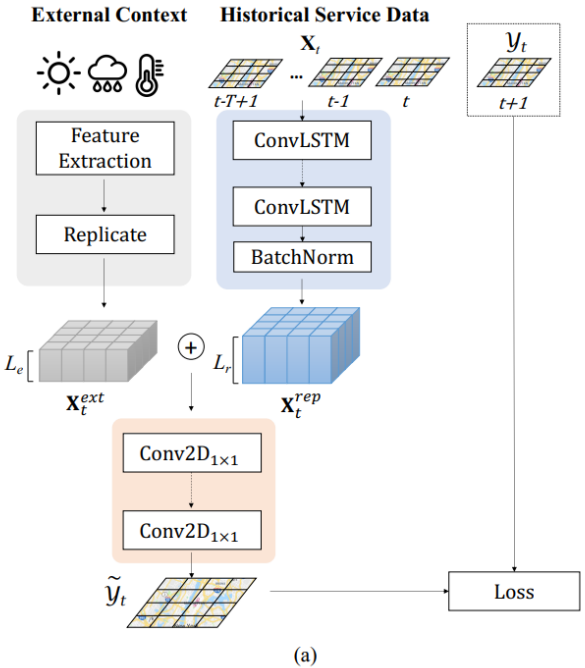


$$\rho_{r,r^*} = corr(\{a_{r,t}\}, \{a_{r^*,t}\}), \quad r \in \mathbb{C}_{\mathcal{D}}, r^* \in \mathbb{C}_{\mathcal{D}'}, t \in \mathbb{T}_{\mathcal{A}}$$

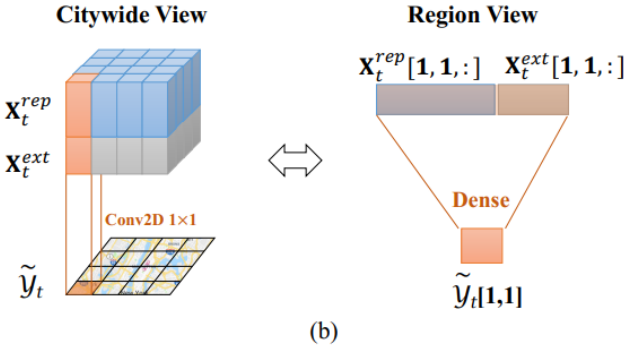
Figure 2: Check-in/crowd flow similarities.

ST Neural Networks with Region Representations

- Existing deep spatio-temporal models often take the whole city data for end-to-end prediction, e.g., ST-ResNet, which cannot be used for region-level transfer



(a)



(b)

$$\textit{ConvLSTM: } f_{\theta_1} : \mathbb{R}^{k\times W\times H} \rightarrow \mathbb{R}^{W\times H\times L_r} \quad (8)$$

$$\textit{Region representation: } \mathbf{X}_t^{rep} = f_{\theta_1}(\mathbf{X}_t) \quad (9)$$

$$\textit{Merge: } f_m : (\mathbb{R}^{W\times H\times L_r}, \mathbb{R}^{W\times H\times L_e}) \rightarrow \mathbb{R}^{W\times H\times (L_r+L_e)} \quad (10)$$

$$\textit{Conv2D}_{1\times1}\textit{: } f_{\theta_2} : \mathbb{R}^{W\times H\times (L_r+L_e)} \rightarrow \mathbb{R}^{W\times H} \quad (11)$$

$$\textit{Prediction : } \widetilde{\mathcal{Y}}_t = f_{\theta_2}(f_m(\mathbf{X}_t^{rep}, \mathbf{X}_t^{ext})) \quad (12)$$

$$= f_{\theta_2}(f_m(f_{\theta_1}(\mathbf{X}_t), \mathbf{X}_t^{ext})) \quad (13)$$

Region-based Cross-city Network Parameter Optimization

- Target 1: more accurate prediction in the target city
- Target 2: we try to minimize the squared error between the network hidden representations of the target region and its matched source region

$$\min_{\theta_{\mathcal{D}}} \quad (1-w) \sum_{t \in \mathbb{T}_{\mathcal{D}}} ||\widetilde{\mathcal{Y}}_t - \mathcal{Y}_t||_F^2 + w \sum_{r \in \mathbb{C}_{\mathcal{D}}} \sum_{t \in \mathbb{T}_{\mathcal{D}}} \rho_{r,r^*} \cdot ||\mathbf{x}_{r,t}^{rep} - \mathbf{x}_{r^*,t}^{rep}||^2$$

Evaluation

	D.C.→Chicago		Chicago→D.C.		D.C.→NYC		NYC→D.C.	
	1-day	3-day	1-day	3-day	1-day	3-day	1-day	3-day
Target Only								
ARIMA	0.740	0.694	0.707	0.661	0.360	0.341	0.707	0.661
DeepST	0.771	0.711	1.075	0.767	0.350	0.359	1.075	0.767
ST-ResNet	0.914	0.703	0.869	0.738	0.376	0.349	0.869	0.738
Source & Target								
DeepST (FT)	0.652	0.611	0.672	0.619	0.363	0.369	0.713	0.711
ST-ResNet (FT)	0.667	0.615	0.695	0.623	0.385	0.349	0.696	0.691
RegionTrans (S-Match)	0.605	0.594	0.631	0.602	**0.328**	**0.305**	**0.665**	**0.593**
RegionTrans (A-Match)	**0.587**	**0.576**	**0.600**	**0.581**	/	/	/	/

Table 1: Evaluation results. The target city holds 1 or 3-day crowd flow historical data. *RegionTrans (A-Match)* is available for D.C. $\rightleftharpoons$ Chicago as we have collected check-in data for Chicago and D.C.

Outline

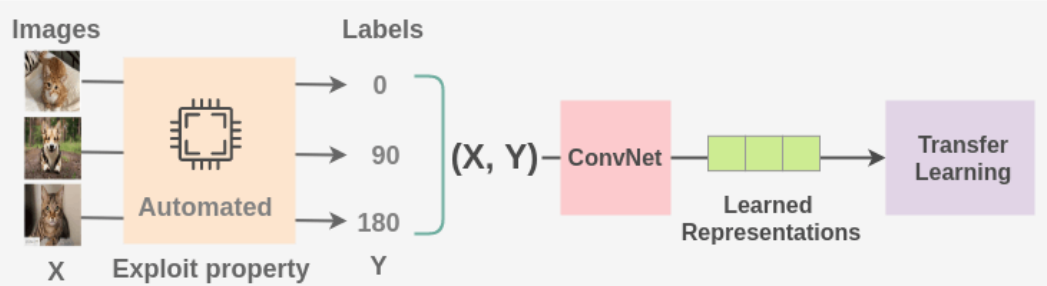
- Transfer learning for ST data
- Self-supervised learning for ST data
  - Contrastive learning for ST data
  - Generative modeling for ST data
- Diffusion models for ST data

Machine Learning Schemes

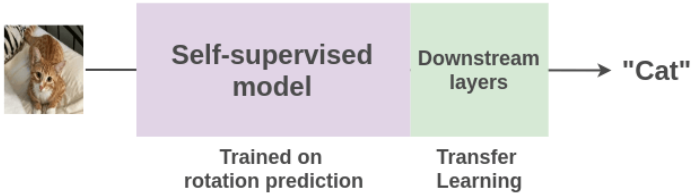
- ***Supervised learning*** – learning with labeled data
  - Approach: collect a large dataset, manually label the data, train a model, deploy
  - It is the dominant form of ML at present
  - Learned feature representations on large datasets are often transferred via pre-trained models to smaller domain-specific datasets
- ***Unsupervised learning*** – learning with unlabeled data
  - Approach: discover patterns in data either via clustering similar instances, or density estimation, or dimensionality reduction …
- ***Self-supervised learning*** – representation learning with unlabeled data
  - Learn useful feature representations from unlabeled data through pretext tasks
  - The term “self-supervised” refers to creating its own supervision (i.e., without supervision, without labels)
  - Self-supervised learning is one category of unsupervised learning

Self-Supervised Learning (SSL)

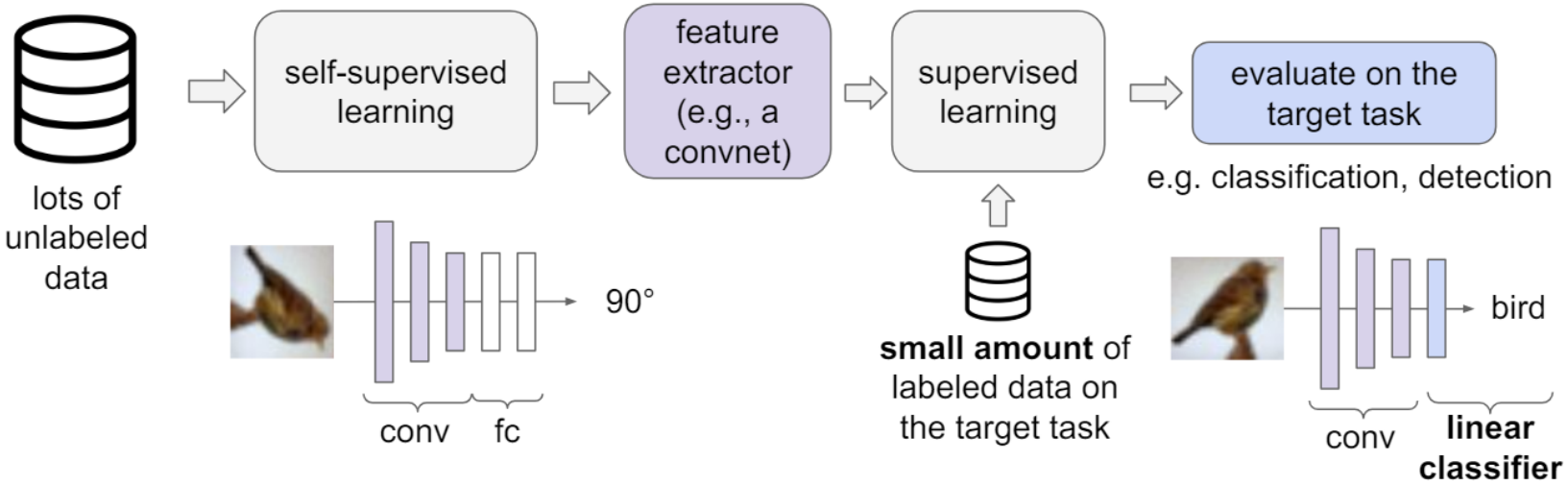
- Self-supervised learning example
  - ***Pretext task***: train a model to predict the rotation degree of rotated images with cats and dogs (we can collect million of images from internet, labeling is not required)



  - ***Downstream task***: use transfer learning and fine-tune the learned model from the pretext task for classification of cats vs dogs with very few labeled examples



SSL Pipeline



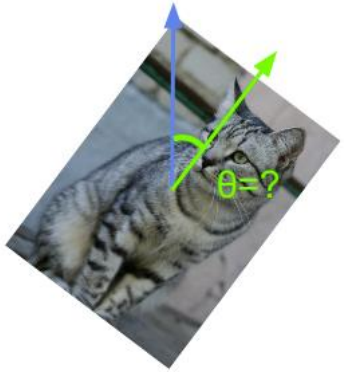
1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

2. Attach a shallow network on the feature extractor; train the shallow network on the target task with small amount of labeled data

SSL Pretext Tasks in Computer Vision

Example: learn to predict image transformations / complete corrupted images

image completion



rotation prediction

“jigsaw puzzle”

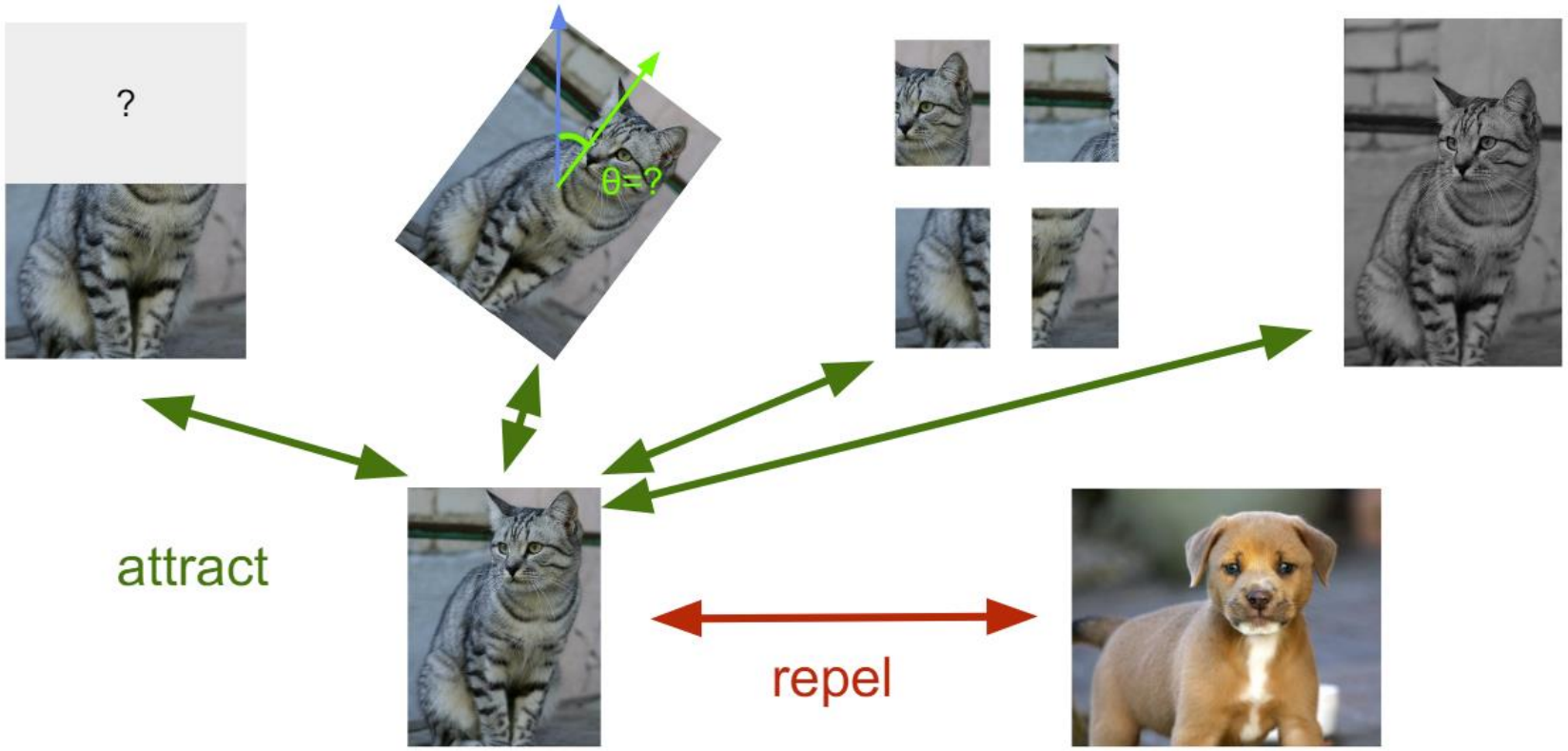
colorization

1. Solving the pretext tasks allow the model to learn good features.
2. We can automatically generate labels for the pretext tasks.

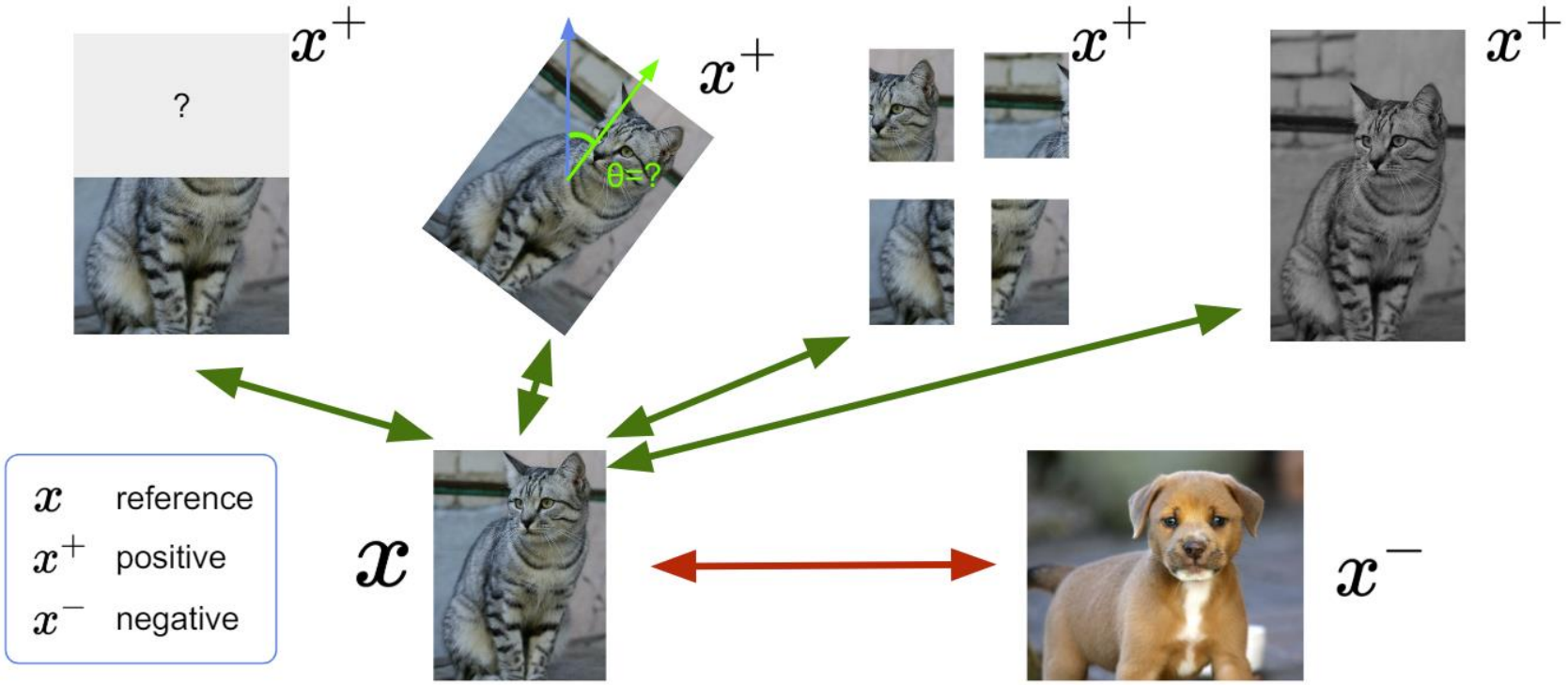
SSL Pretext Tasks in General Domains

- Contrastive
- Generative

Contrastive Learning



Contrastive Learning



Formulation of Contrastive Learning

What we want:

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

x: reference sample; x^+ positive sample; x^- negative sample

Given a chosen score function, we aim to learn an **encoder function** *f* that yields high score for positive pairs (x, x^+) and low scores for negative pairs (x, x^-).

Formulation of Contrastive Learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

x

x^+

x

x_1^-

x_2^-

x_3^-

...

Formulation of Contrastive Learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

score for the positive pair

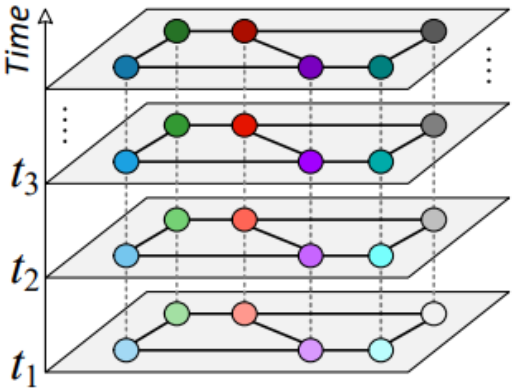
score for the N-1 negative pairs

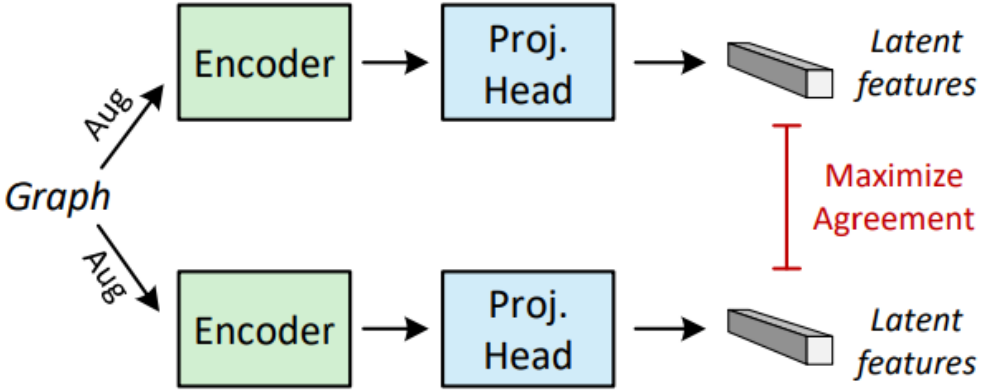
This seems familiar …
Cross entropy loss for a N-way softmax classifier!
I.e., learn to find the positive sample from the N samples

Spatio-Temporal Graph Contrastive Learning

SIGSPATIAL 2022

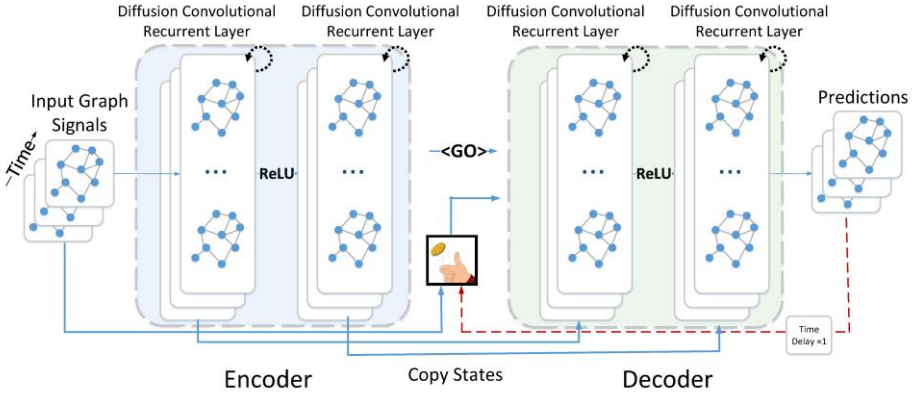
X. Liu et al. When Do Contrastive Learning Signals Help Spatio-Temporal Graph Forecasting?, SIGSPATIAL 2022



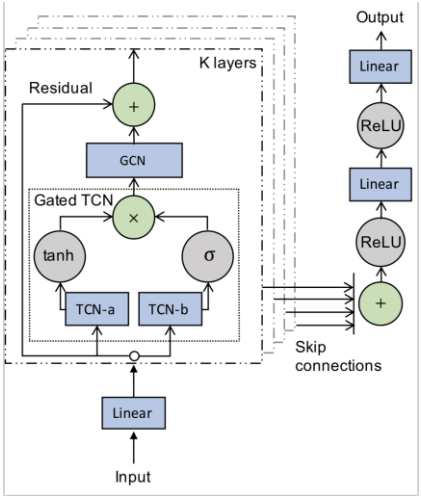


Spatio-Temporal Graph Neural Networks

- 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

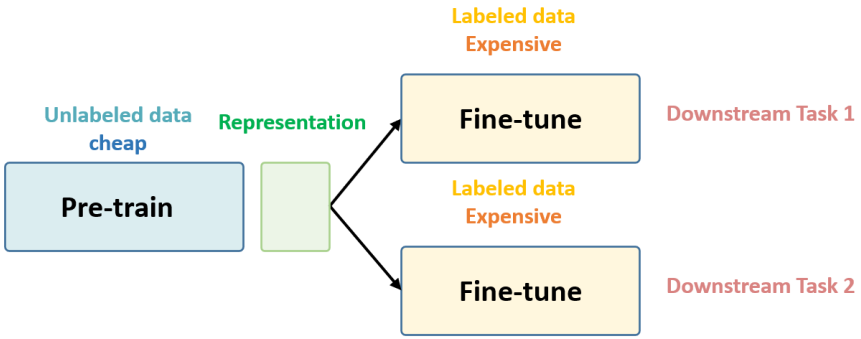
Challenges

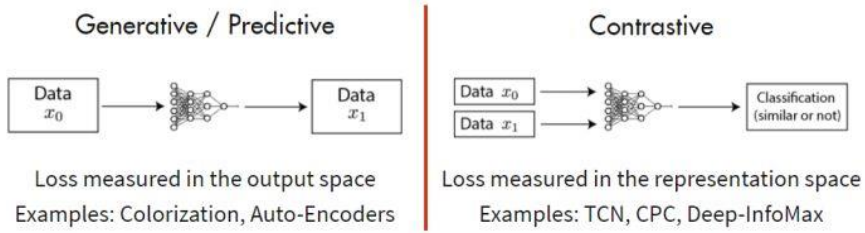
- Tremendous efforts have been made to design sophisticated architectures to capture complex spatio-temporal dependencies
- However, data scarcity is a crucial issue that may hinder the recent improvements on STG forecasting

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).

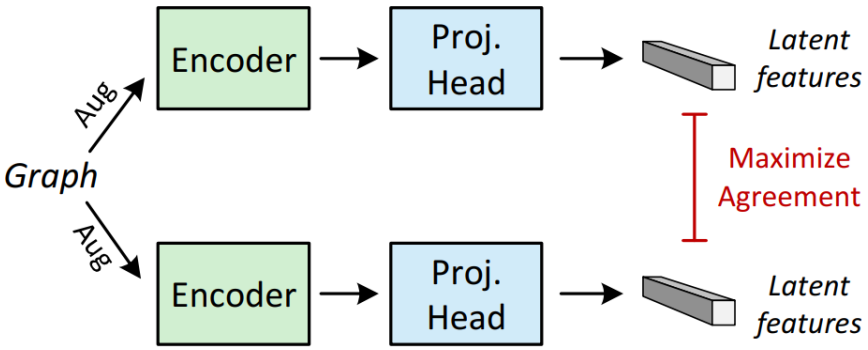


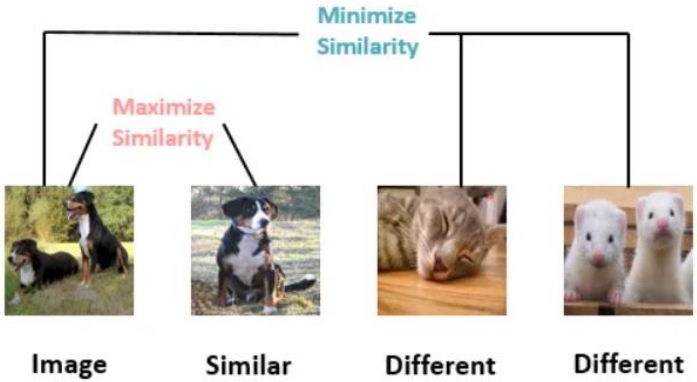


Contrastive Learning

- 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

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

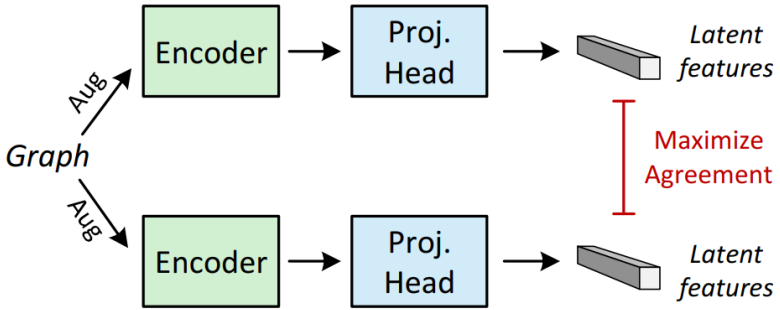




Our Contributions

- 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?



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

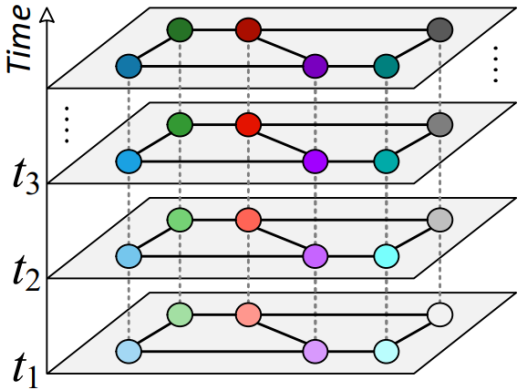
Our Contributions

- 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

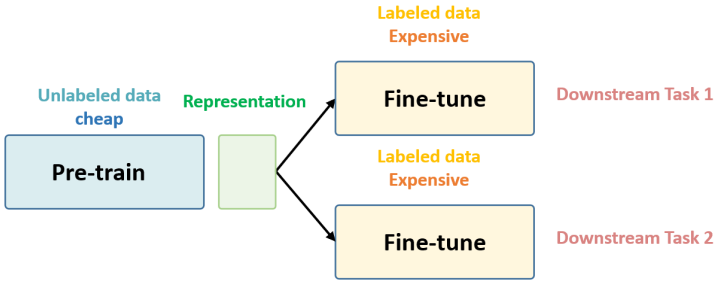
- A sensor network is denoted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$
- The observation at a time step t is $\mathbf{X}^t \in \mathbb{R}^{N \times F}$
- STG forecasting problem

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

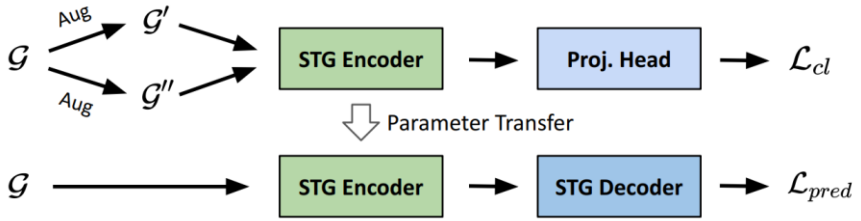


Training Schemes (Q1)

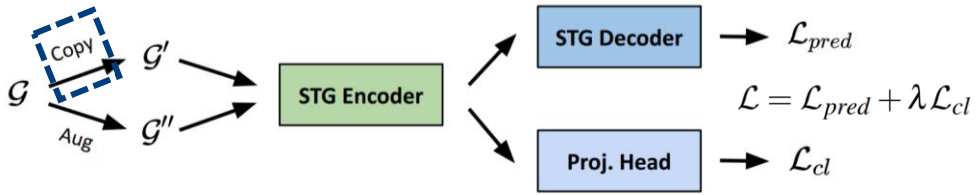
- **Q1**: *What is the appropriate training scheme when integrating contrastive learning with STG forecasting?*
- We identify two candidate schemes to incorporate contrastive learning
  - Pretraining & Fine-tuning
  - Joint learning



Standard CL for CV, Graphs



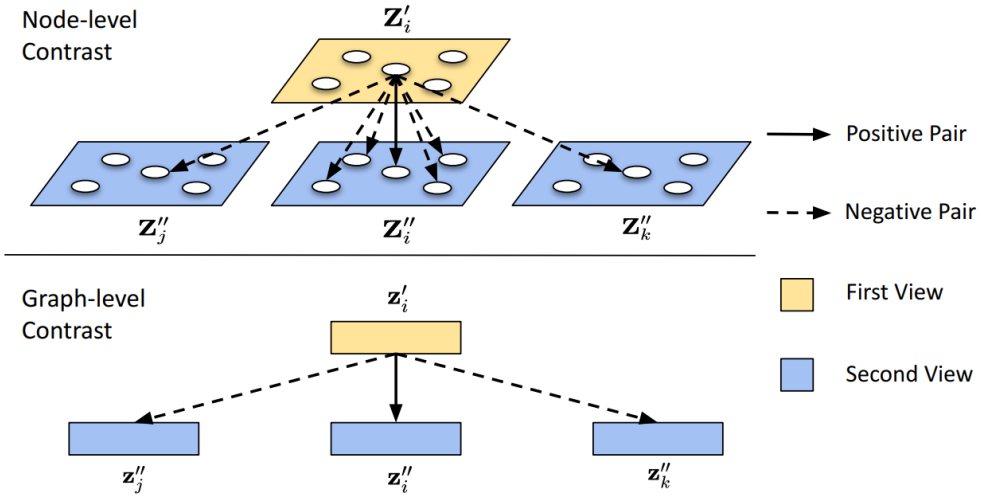
(a) Pretraining & Fine-tuning



(b) Joint Learning

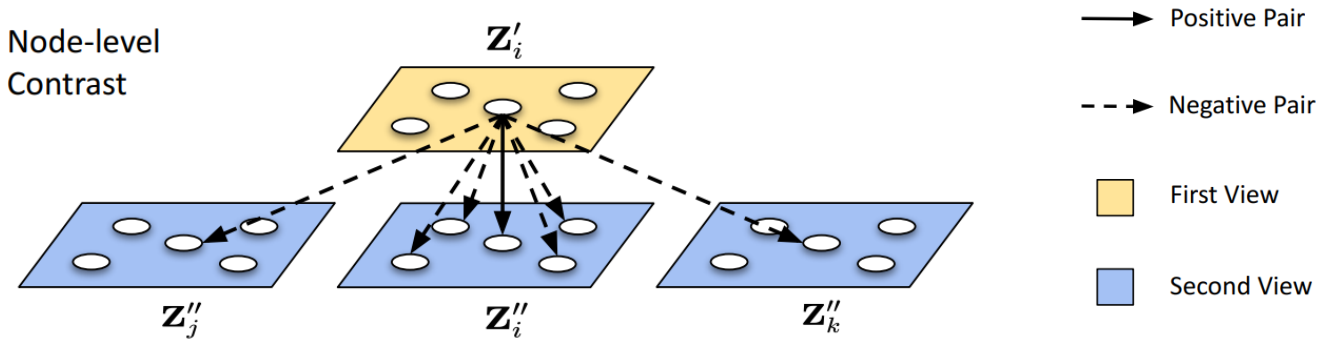
What & How to Contrast (Q2)

- **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



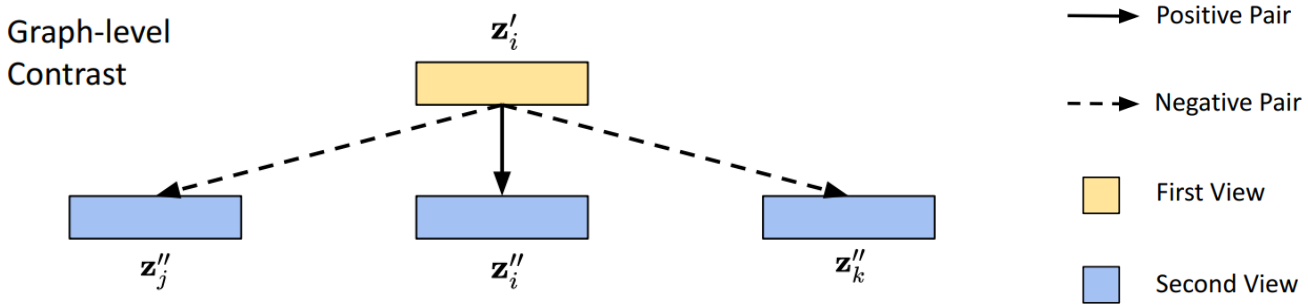
What & How to Contrast (Q2)

- 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)

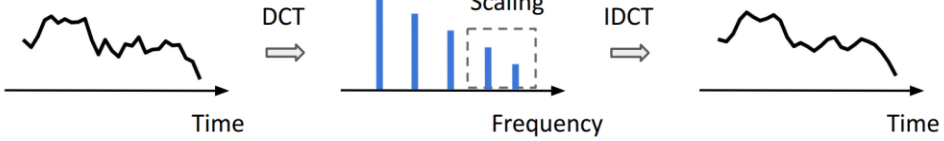
- **Q3**: *How should we perform data augmentation to generate a positive pair?*
- We propose four methods perturb data in different aspects: graph structure, time domain, and frequency domain

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

Edge Masking

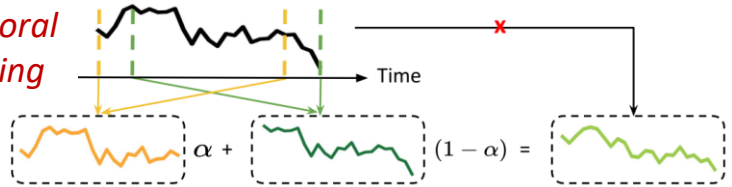
$$\mathbf{P}_{ij}^{(t-S):t} = \begin{cases} \mathbf{X}_{ij}^{(t-S):t}, & \text{if } \mathbf{M}_{ij} \geqslant r_{im} \\ -1, & \text{otherwise} \end{cases}$$

Input Masking



Input Smoothing

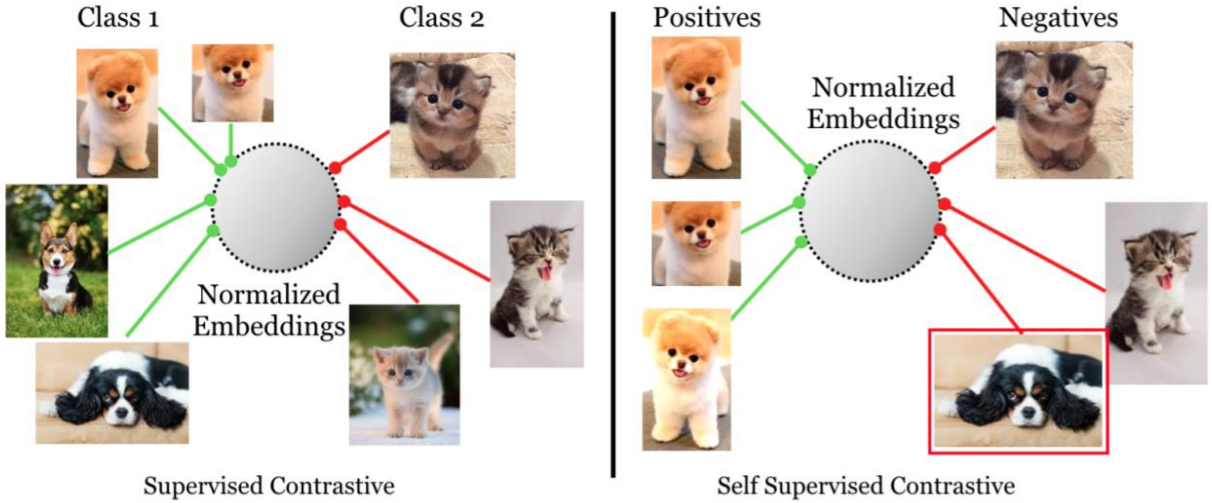
Temporal Shifting



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

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

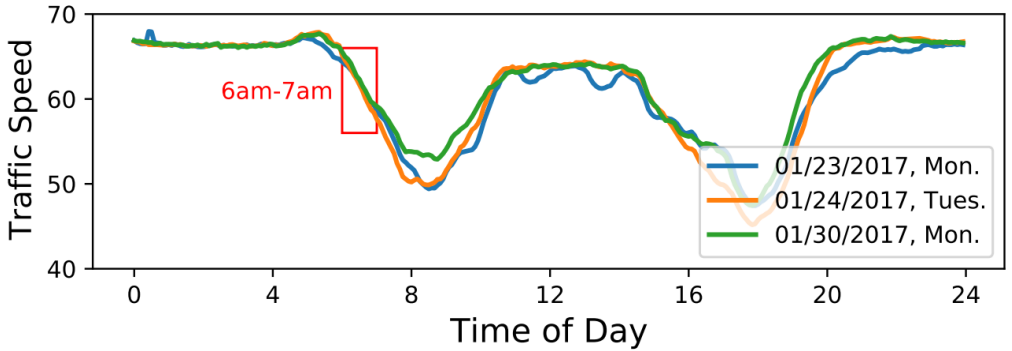
$$\mathcal{L}_{cl} = \frac{1}{M}\sum_{i=1}^{M} -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_i'')/\tau)}{\sum_{j=1, j\neq i}^{M} \exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_j'')/\tau)} \quad \Rightarrow \quad \frac{1}{M}\sum_{i=1}^{M} -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i', \mathbf{z}_i'')/\tau)}{\sum_{j\in\chi_i}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

A set of acceptable negatives for the i-th sample

How to filter out useless negatives?

Negative Filtering (Q4)

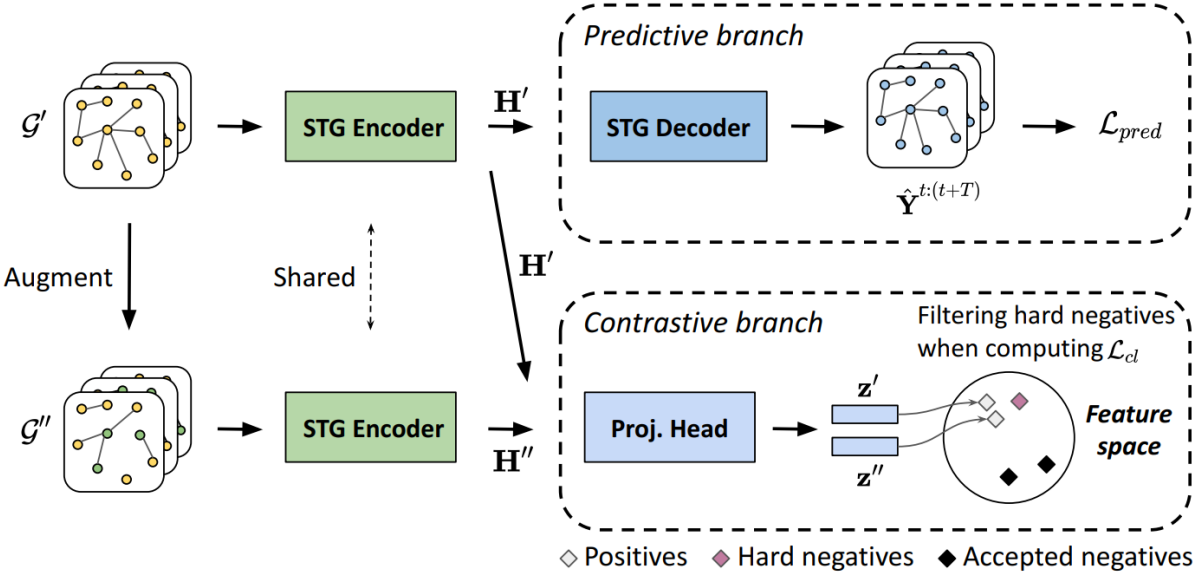
- **Temporal negative filtering**: exclude unsuitable negatives by utilizing ubiquitous temporal properties of STG – closeness and periodicity



- **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**

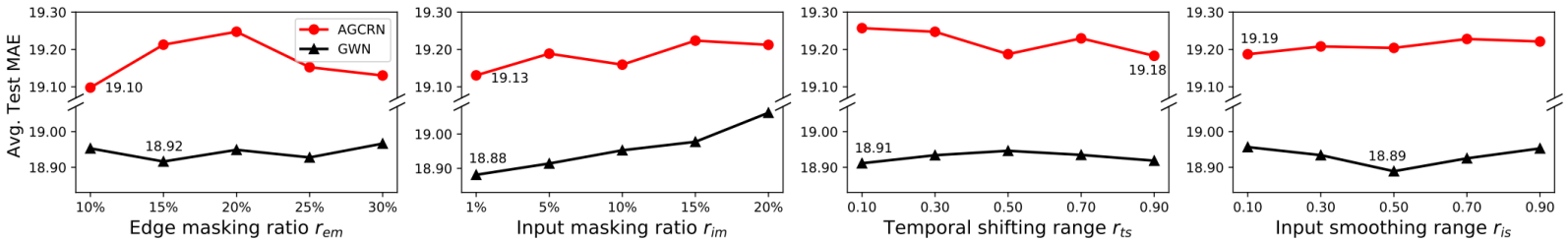
Empirical Results (Q2)

- 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

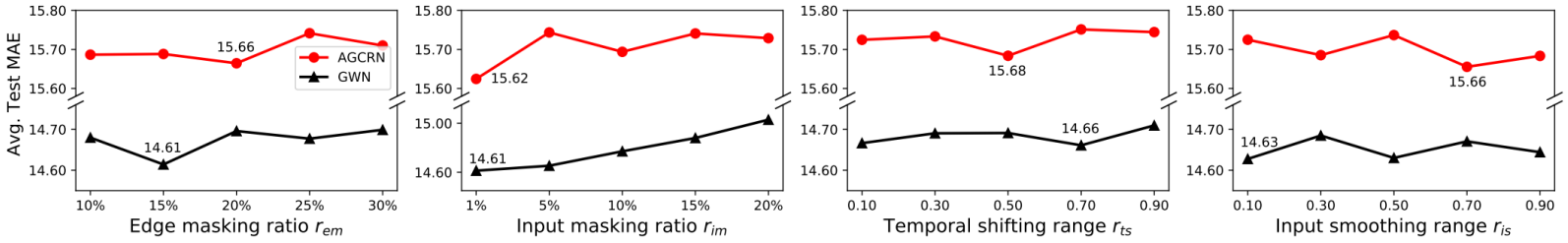
Methods	PEMS-04				PEMS-08			
	15 min	30 min	60 min	Δ	15 min	30 min	60 min	Δ
GWN	18.20±0.09	19.32±0.13	21.10±0.18	–	13.80±0.05	14.75±0.04	16.39±0.09	–
w/ JL-node	17.97±0.05	18.90±0.07	**20.38±0.07**	-1.37	13.67±0.06	14.63±0.08	16.14±0.13	-0.50
w/ JL-graph	**17.93±0.04**	**18.87±0.04**	20.40±0.09	**-1.42**	**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



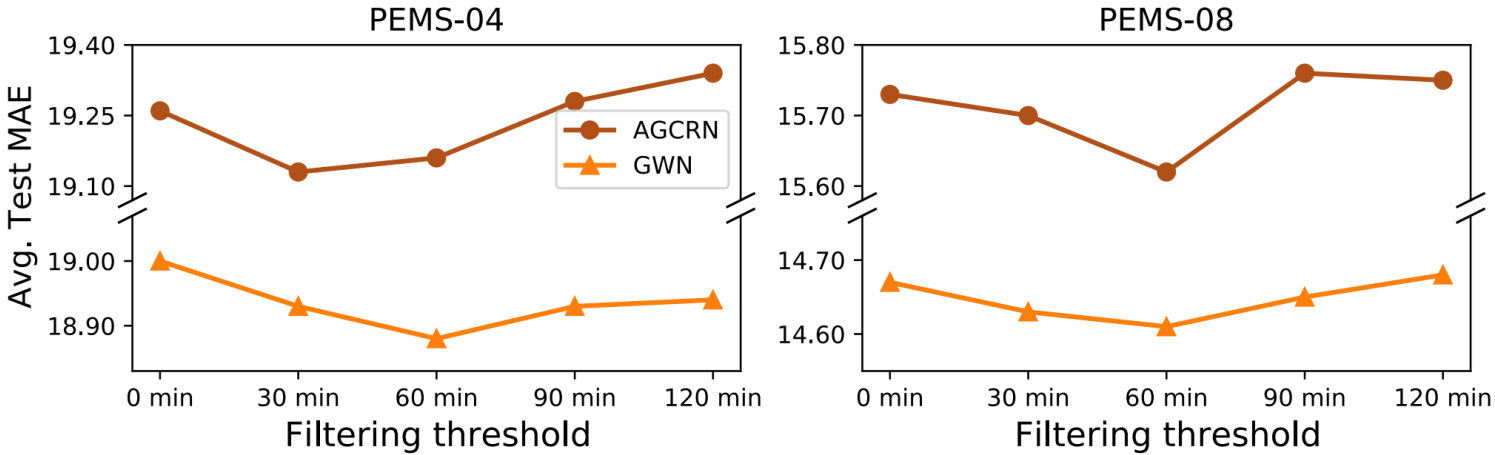
(a) PEMS-04



(b) PEMS-08

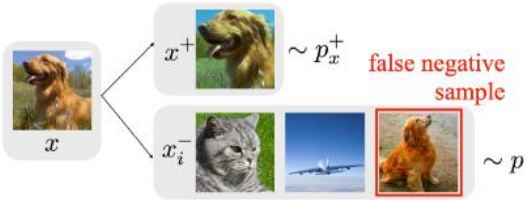
Empirical Results (Q4)

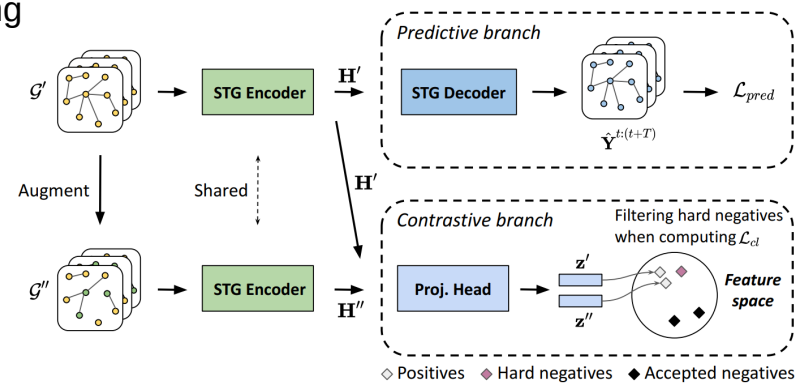
- Effects of temporal negative filtering



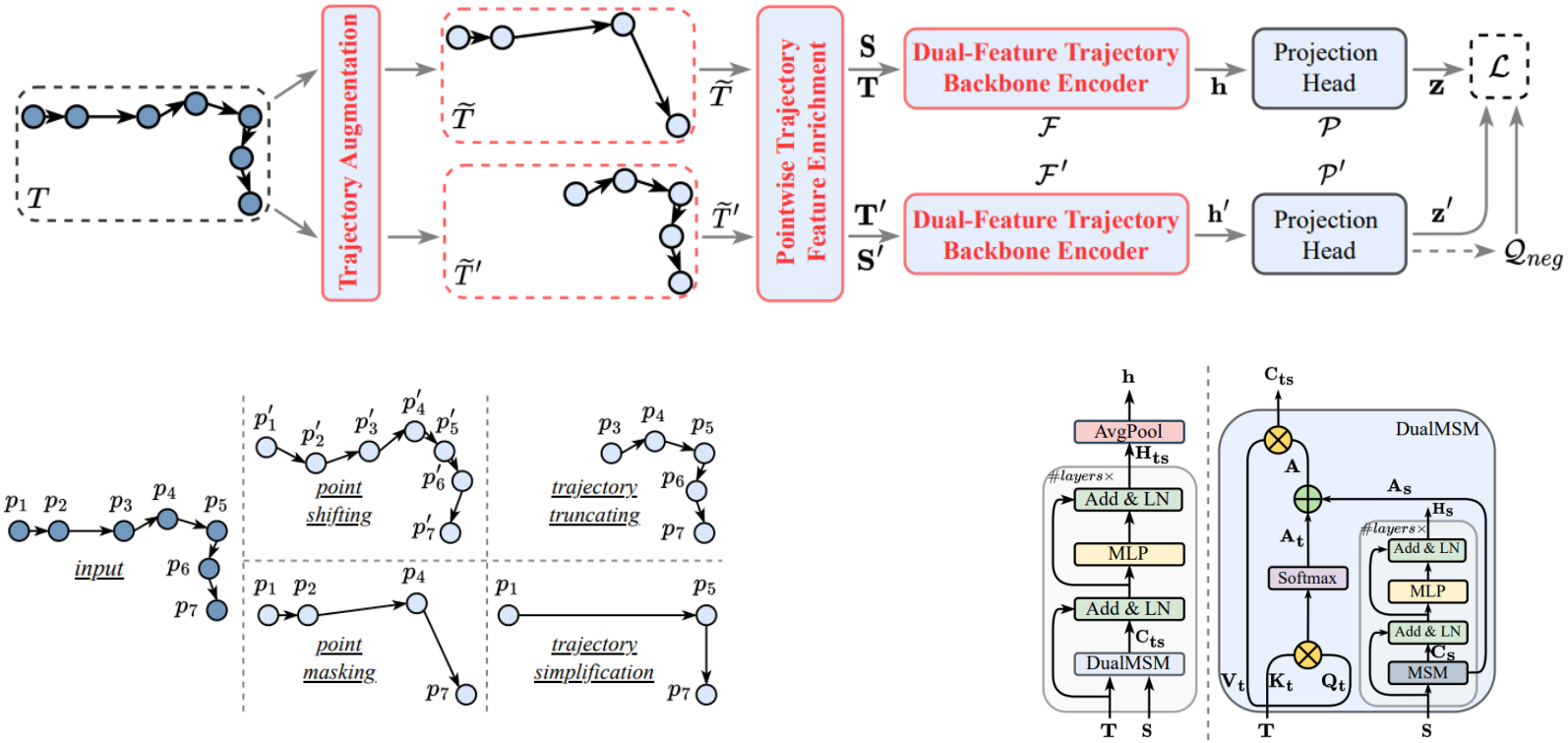
Further Discussion

- 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





Contrastive Learning for ST Trajectories



Y. Chang et al., Contrastive Trajectory Similarity Learning with Dual-Feature Attention. ICDE 2023.

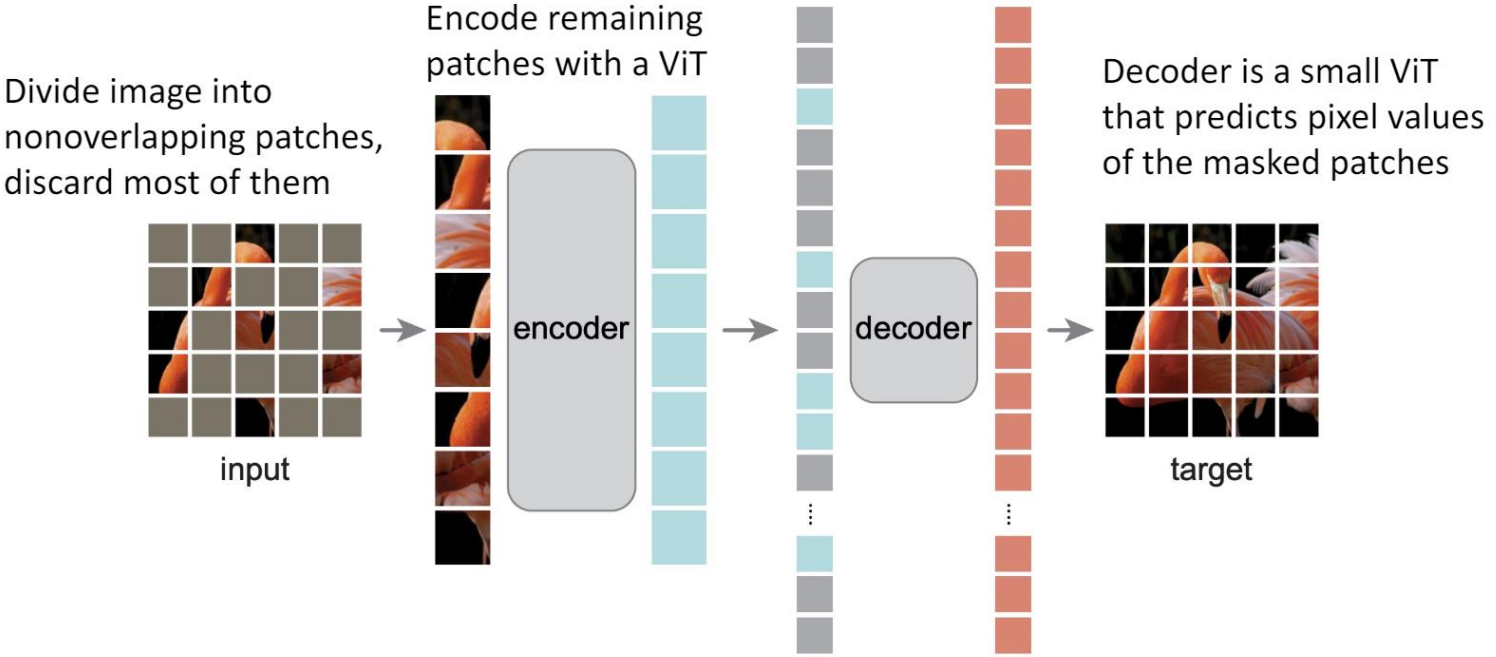
SSL Pretext Tasks in General Domains

- Contrastive
- Generative

(Masked) Generative Learning

- Masked AutoEncoder (MAE) [He et al. 2021]

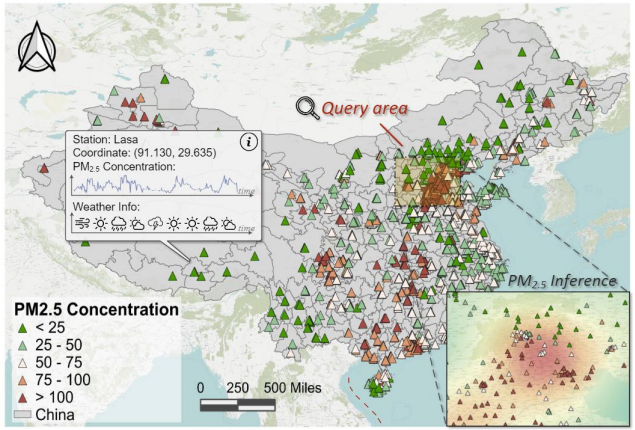
A ~~new~~ old method dethrones contrastive learning? Denoising Autoencoder with Vision Transformer

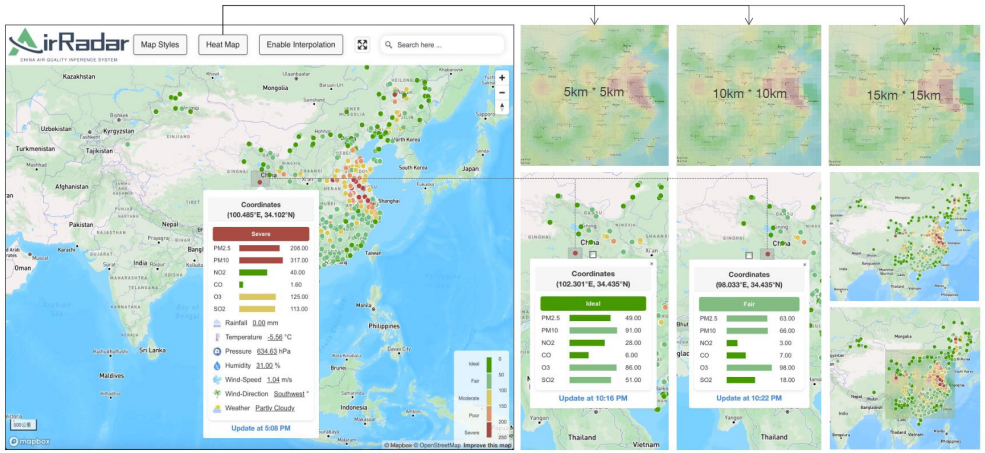


AirRadar: Inferring Nationwide Air Quality in China with Deep Neural Networks

Under Review

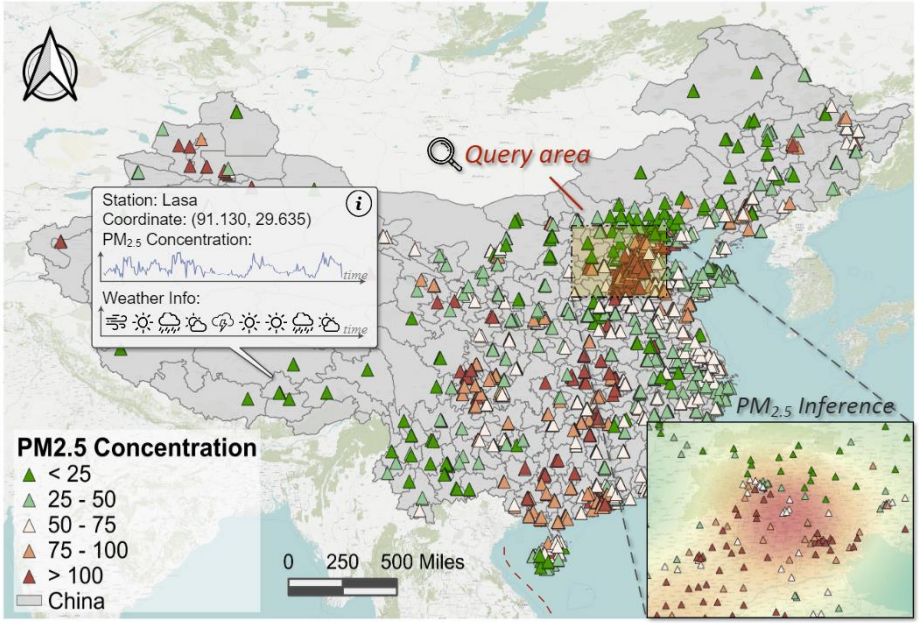
Wang et al. AirRadar: Inferring Nationwide Air Quality in China with Deep Neural Networks, Under Review





Background

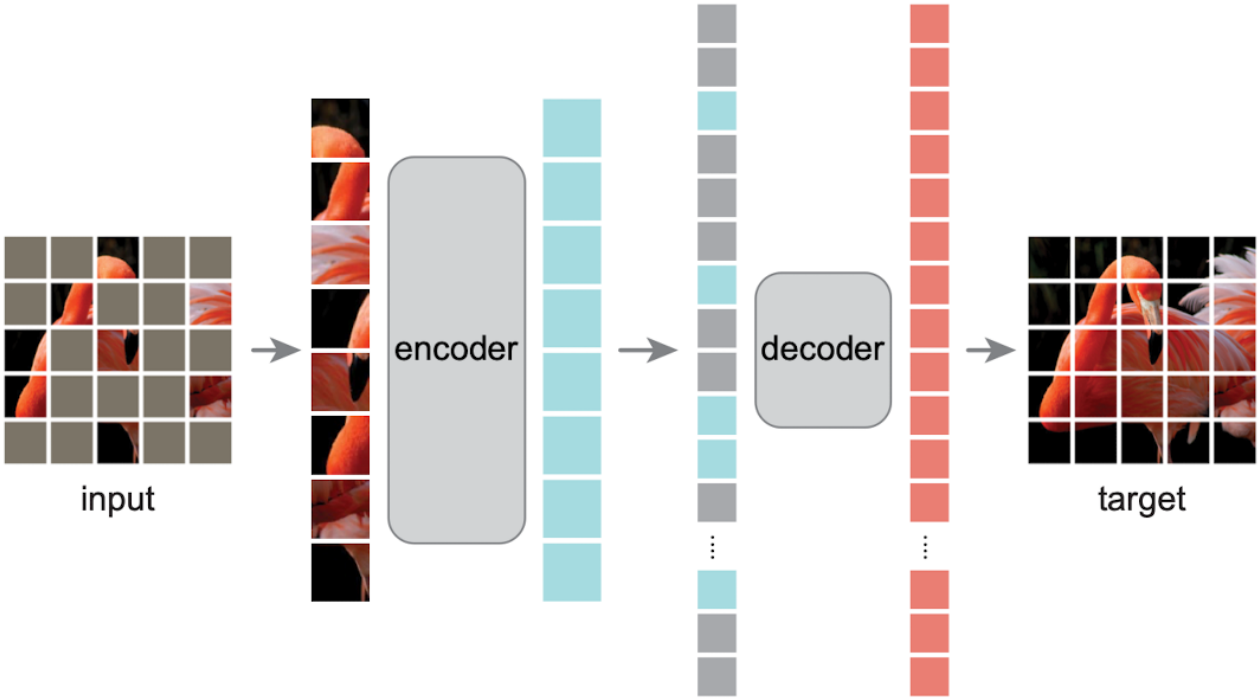
- Air pollution poses a significant threat to global health
  - Real-time monitoring is required
- Establishing monitoring stations incurs huge costs
  - Unable to cover most areas
- Goal: Inferring air quality spanning nationwide



Wang et al. AirRadar: Inferring Nationwide Air Quality in China with Deep Neural Networks, Under Review

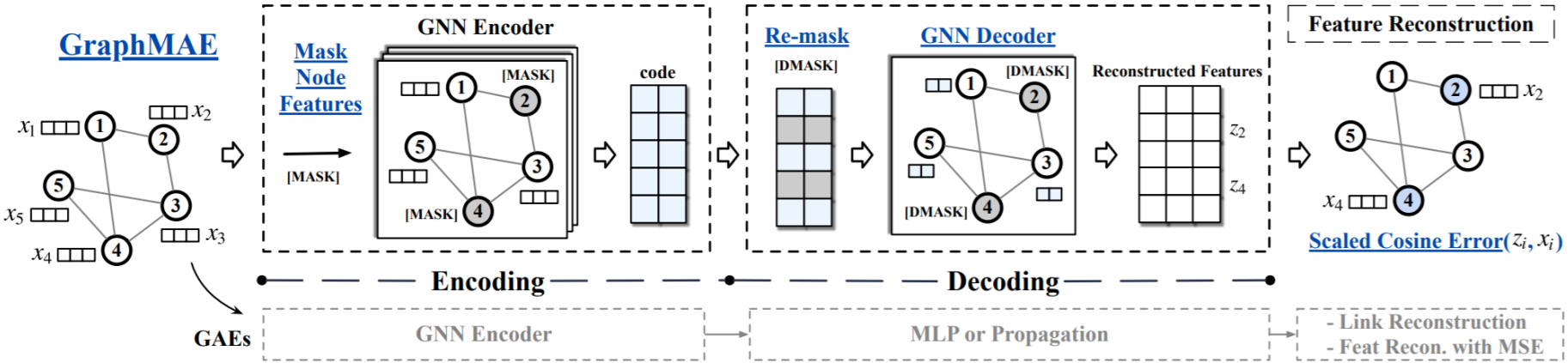
Related Work

- Masked AutoEncoder (MAE) [He et al. 2021]



Related Work

- GraphMAE [He et al. 2021]



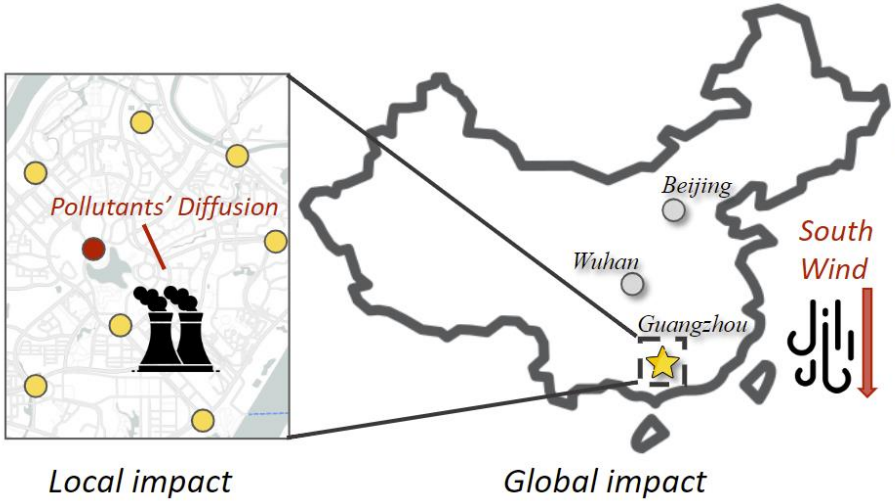
Results of GraphMAE

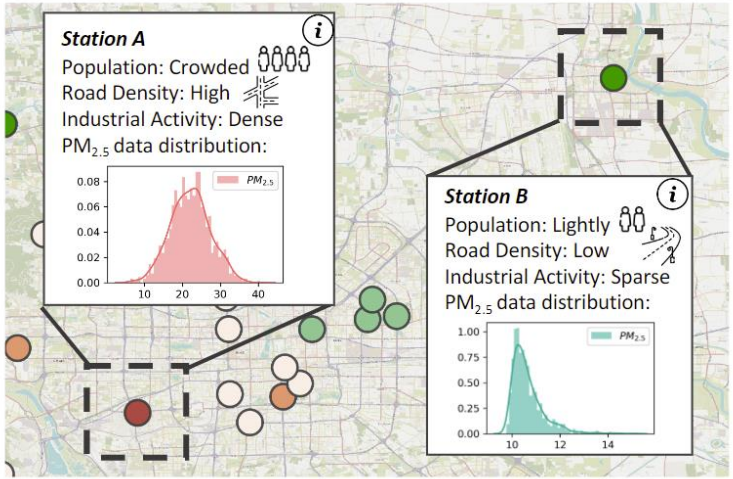
Table III. Model comparison on the nationwide dataset. The parameter count, denoted as #Param, is in the order of million (M). The symbol Δ represents the reduction in MAE compared to GraphMAE. The mask ratio represents the proportion of unobserved nodes to all nodes.

Model	Year	#Param(M)	Mask Ratio = 25%				Mask Ratio = 50%				Mask Ratio = 75%			
			MAE	Δ	RMSE	MAPE	MAE	Δ	RMSE	MAPE	MAE	Δ	RMSE	MAPE
KNN	1967	-	30.50	+146.0%	65.40	1.36	30.25	+145.5%	72.23	0.71	34.07	+194.0%	74.55	0.64
RF	2001	-	29.22	+135.6%	68.95	0.76	29.71	+141.2%	71.61	0.75	29.82	+157.3%	70.99	0.74
MCAM	2021	0.408	23.94	+93.1%	36.25	0.95	25.01	+103.0%	37.94	0.92	25.19	+117.3%	37.82	1.04
SGNP	2019	0.114	23.60	+90.3%	37.58	0.83	24.06	+95.3%	37.08	0.93	21.68	+87.1%	33.68	0.84
STGNP	2022	0.108	23.21	+87.2%	38.13	0.62	21.95	+78.2%	37.13	0.67	19.58	+68.9%	31.95	0.69
VAE	2013	0.011	28.49	+129.8%	67.11	0.94	28.92	+134.7%	69.67	0.94	29.00	+150.2%	69.11	0.93
GAE	2016	0.073	12.63	+1.9%	23.80	0.46	12.78	+3.7%	24.11	0.46	12.57	+8.5%	23.73	0.46
GraphMAE	2022	0.073	12.40	-	23.20	0.46	12.32	-	23.11	0.46	11.59	-	21.51	0.43

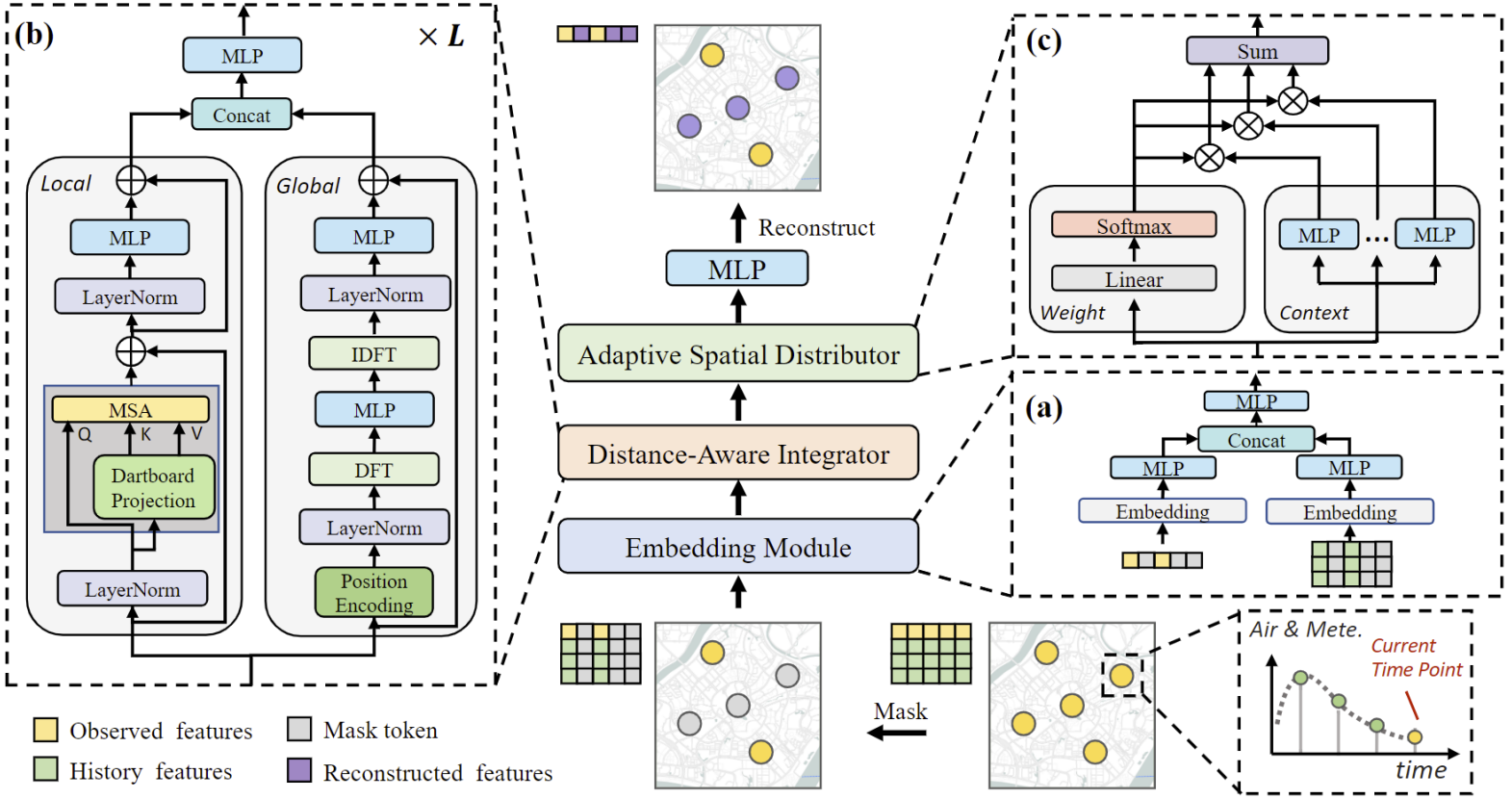
Challenges

- Local and global impact
- Spatial distribution shift



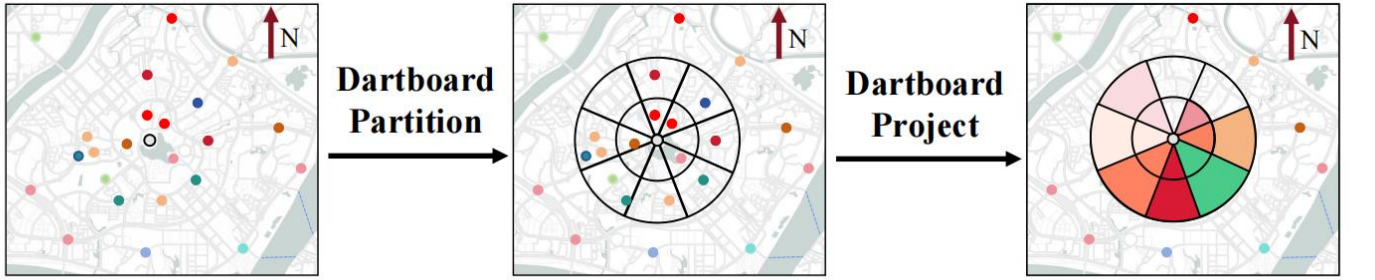


Framework


(b)
× L
MLP
Concat
Local
MLP
LayerNorm
MSA
Q K V
Dartboard Projection
LayerNorm
Global
MLP
LayerNorm
IDFT
MLP
DFT
LayerNorm
Position Encoding
Reconstruct
MLP
Adaptive Spatial Distributor
Distance-Aware Integrator
Embedding Module
Mask
(c)
Sum
Softmax
Linear
Weight
MLP
MLP
Context
(a)
MLP
Concat
MLP
MLP
Embedding
Embedding
Air & Mete.
Current Time Point
time
Observed features
Mask token
History features
Reconstructed features

Local spatial correlation

- Consider the impact from neighbor
- Divide the local scope into regions



(a) Initial feature (b) Partition matrix (c) Projected feature

Global spatial correlation

- Consider the impact of other cities
- Using the FNO to capture the global spatial relationship

$$K(X)(i) = F^{-1}(F(\kappa) \cdot F(X))(i) \quad i \in D$$

- Reduce computational complexity
  - $O(NE^2/K + NE\log N) < O(N^2E + 3NE^2)$

Adaptive Spatial Distributor

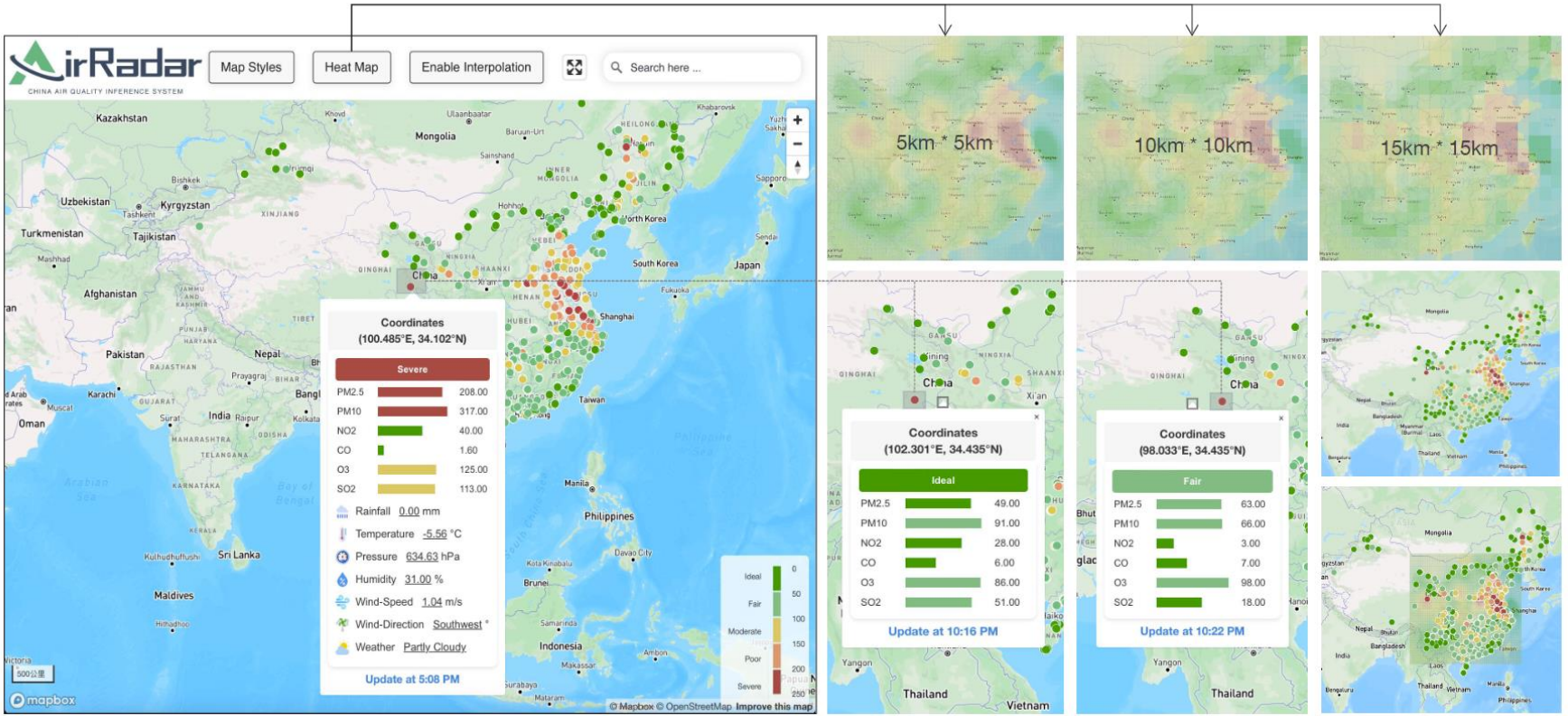
- Address spatial distribution shift
- Introduce dynamic weight allocation

$$\mathbf{Y} = \sum_{i=1}^{|C|} J(\mathbf{Z}^l)_i \mathbf{H} c_i$$

Results

Model	Year	#Param(M)	Mask Ratio = 25%				Mask Ratio = 50%				Mask Ratio = 75%			
			MAE	Δ	RMSE	MAPE	MAE	Δ	RMSE	MAPE	MAE	Δ	RMSE	MAPE
KNN	1967	-	30.50	+146.0%	65.40	1.36	30.25	+145.5%	72.23	0.71	34.07	+194.0%	74.55	0.64
RF	2001	-	29.22	+135.6%	68.95	0.76	29.71	+141.2%	71.61	0.75	29.82	+157.3%	70.99	0.74
MCAM	2021	0.408	23.94	+93.1%	36.25	0.95	25.01	+103.0%	37.94	0.92	25.19	+117.3%	37.82	1.04
SGNP	2019	0.114	23.60	+90.3%	37.58	0.83	24.06	+95.3%	37.08	0.93	21.68	+87.1%	33.68	0.84
STGNP	2022	0.108	23.21	+87.2%	38.13	0.62	21.95	+78.2%	37.13	0.67	19.58	+68.9%	31.95	0.69
VAE	2013	0.011	28.49	+129.8%	67.11	0.94	28.92	+134.7%	69.67	0.94	29.00	+150.2%	69.11	0.93
GAE	2016	0.073	12.63	+1.9%	23.80	0.46	12.78	+3.7%	24.11	0.46	12.57	+8.5%	23.73	0.46
GraphMAE	2022	0.073	12.40	-	23.20	0.46	12.32	-	23.11	0.46	11.59	-	21.51	0.43
AirRadar	-	0.343	**6.41**	**-48.3%**	**12.60**	**0.24**	**6.79**	**-44.9%**	**12.90**	**0.26**	**8.11**	**-30.0%**	**14.84**	**0.29**

Demo System

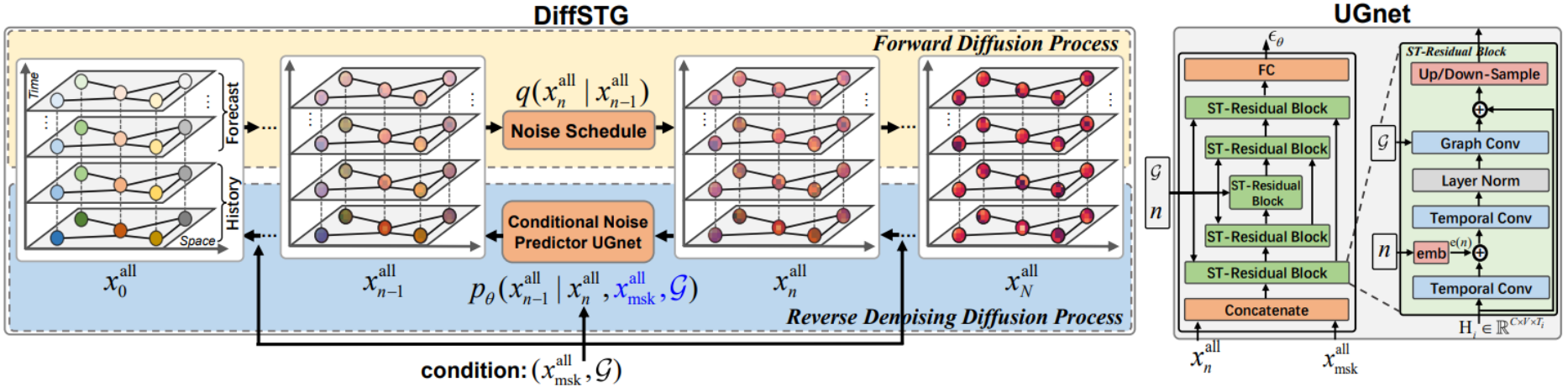

AirRadar
CHINA AIR QUALITY INFERENCE SYSTEM
Map Styles
Heat Map
Enable Interpolation
Search here ...
Coordinates
(100.485°E, 34.102°N)
Severe
PM2.5 208.00
PM10 317.00
NO2 40.00
CO 1.60
O3 125.00
SO2 113.00
Rainfall 0.00 mm
Temperature -5.56 °C
Pressure 634.63 hPa
Humidity 31.00 %
Wind-Speed 1.04 m/s
Wind-Direction Southwest °
Weather Partly Cloudy
Update at 5:08 PM
Ideal
Fair
Moderate
Poor
Severe
5km * 5km
10km * 10km
15km * 15km
Coordinates
(102.301°E, 34.435°N)
Ideal
PM2.5 49.00
PM10 91.00
NO2 28.00
CO 6.00
O3 86.00
SO2 51.00
Update at 10:16 PM
Coordinates
(98.033°E, 34.435°N)
Fair
PM2.5 63.00
PM10 66.00
NO2 3.00
CO 7.00
O3 98.00
SO2 18.00
Update at 10:22 PM
mapbox
© Mapbox © OpenStreetMap Improve this map

Outline

- Transfer learning for ST data
- Self-supervised learning for ST data
  - Contrastive learning for ST data
  - Generative modeling for ST data
- Diffusion models for ST data

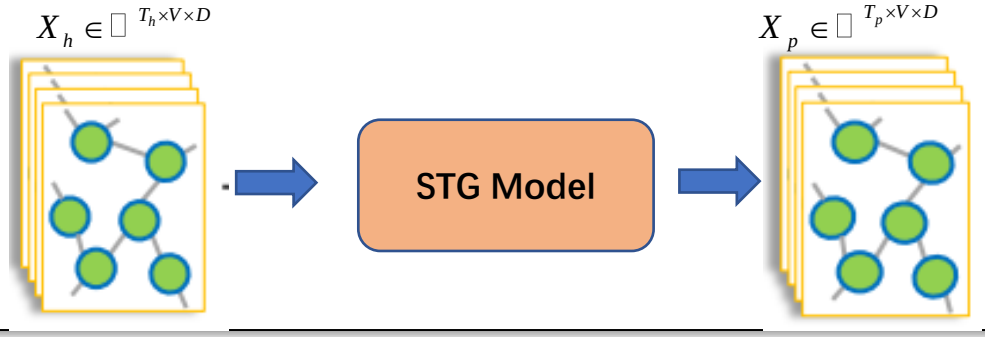
DiffSTG: Probabilistic Spatio-Temporal Graph Forecasting with Denoising Diffusion Models

SIGSPATIAL 2023

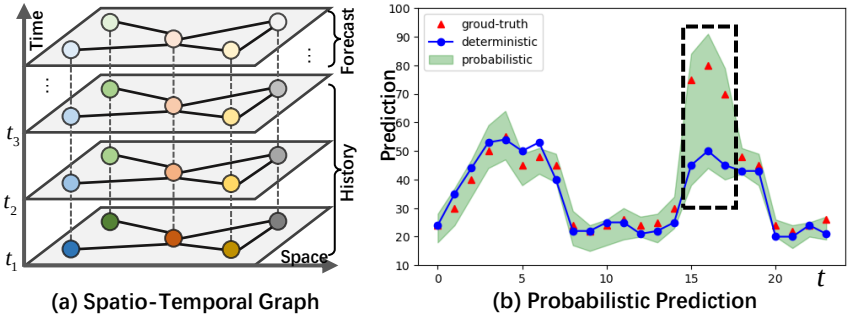


Problem Definition

Spatio-temporal Graph (STG) Forecasting



Stochastic STG Forecasting



(a) Spatio-Temporal Graph (b) Probabilistic Prediction

- Given the historical spatial-temporal graph (STG) to predict the future STG.
- Stochastic Prediction

Related Work: Spatial-Temporal Probabilistic Forecasting

Spatial-Temporal Stochastics Forecasting

[1] Liu Y, Qin H, Zhang Z, et al. Probabilistic spatiotemporal wind speed forecasting based on a variational Bayesian deep learning model[J]. Applied Energy, 2020, 260: 114259.

[2] Agoua X G, Girard R, Kariniotakis G. Probabilistic models for spatio-temporal photovoltaic power forecasting[J]. IEEE Transactions on Sustainable Energy, 2018, 10(2): 780-789.

[3] Sun M, Feng C, Zhang J. Conditional aggregated probabilistic wind power forecasting based on spatio-temporal correlation[J]. Applied Energy, 2019, 256: 113842.

[4] Furtlehner C, Lasgouttes J M, Attanasi A, et al. Spatio-temporal probabilistic short-term forecasting on urban networks[D]. Inria Saclay-Île de France; Inria de Paris; PTV-SISTeMA, 2019.

- most of them study the window power forecasting
- most works comes from the field of applied energy, not computer science

Still lack of works in spatial-temporal stochastic prediction

Related Work: Probabilistic Time Series Forecasting

Probabilistic Time Series Forecasting

[1] DeepAR: presented the first attempt to utilize a low-rank approximation (i.e., Gaussian distribution) to model the data distribution.

[2] Neural ODE: innovatively define a probabilistic generative process over time series from a latent initial state, which can be trained with variational inference.

Diffusion-based Methods

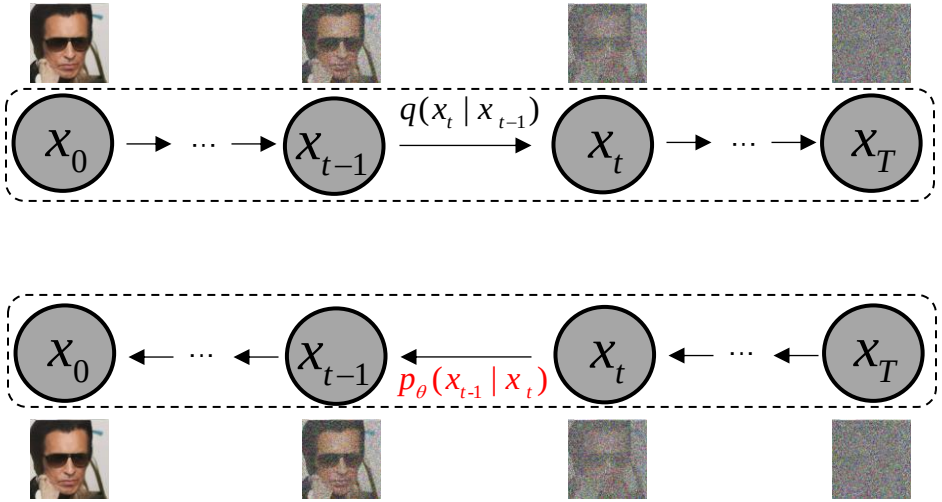
[3] TimeGrad: sets the LSTM-encoded representation of the current time series as condition, and makes the prediction in an auto-regressive way.

[4] CSDI: directly utilizes observed value as the condition and outputs the distribution in a non-autoregressive way.

only model the temporal dependencies without capturing the spatial correlations between different nodes.

Brief Introduction on Diffusion Model

Diffusion Model -- A powerful generative model



Forward Process
- Markov chain
- Fixed process
- add noise

Reverse Process
- Markow chain
- remove noise

$$p_\theta(\boldsymbol{x}_{0:N}) := p(\boldsymbol{x}_N) \prod_{n=N}^{1} p_\theta(\boldsymbol{x}_{n-1}|\boldsymbol{x}_n)$$

$$p_\theta(\boldsymbol{x}_{n-1}|\boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{x}_{n-1}; \mu_\theta(\boldsymbol{x}_n, n), \sigma_\theta(\boldsymbol{x}_n, n))$$

$$\mu_\theta(\boldsymbol{x}_n, n) = \frac{1}{\alpha_n}\left(\boldsymbol{x}_n - \frac{\beta_n}{\sqrt{1-\alpha_n}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_n, n)\right),$$

and

$$\sigma_\theta(\boldsymbol{x}_n, n) = \frac{1-\alpha_{n-1}}{1-\alpha_n}\beta_n,$$

Related Work: Probabilistic Time Series Forecasting

Probabilistic Time Series Forecasting

[1] DeepAR: presented the first attempt to utilize a low-rank approximation (i.e., Gaussian distribution) to model the data distribution.

[2] Neural ODE: innovatively define a probabilistic generative process over time series from a latent initial state, which can be trained with variational inference.

Diffusion-based Methods

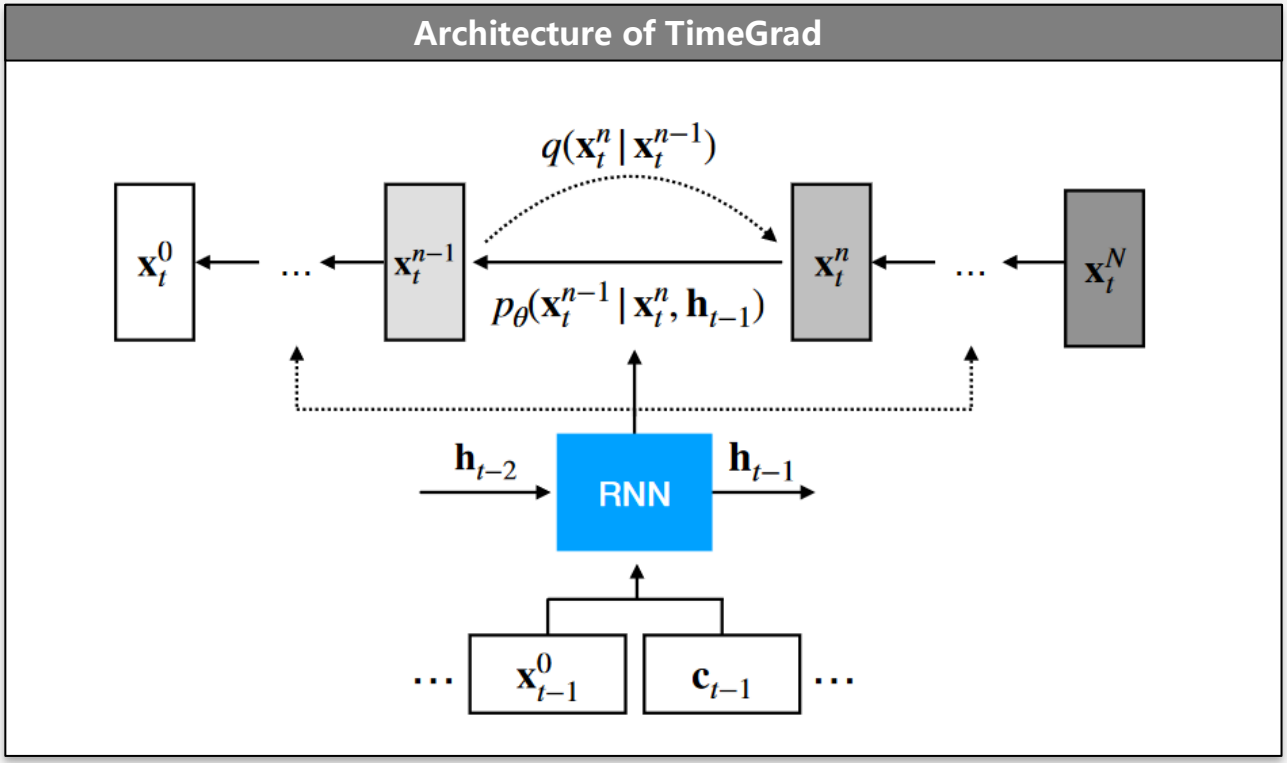
[3] TimeGrad: sets the LSTM-encoded representation of the current time series as condition, and makes the prediction in an auto-regressive way.

[4] CSDI: directly utilizes observed value as the condition and outputs the distribution in a non-autoregressive way.

only model the temporal dependencies without capturing the spatial correlations between different nodes.

> Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting, ICML 2021

Related Work: Diffusion-based Stochastics Time Series Forecasting



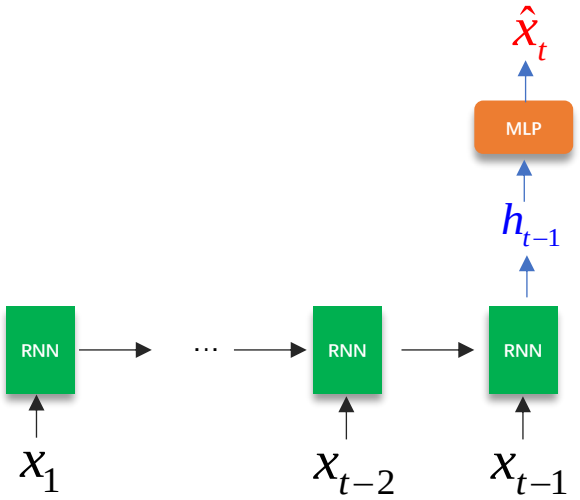
> Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting, ICML 2021

Diffusion-based Stochastics Time Series Forecasting

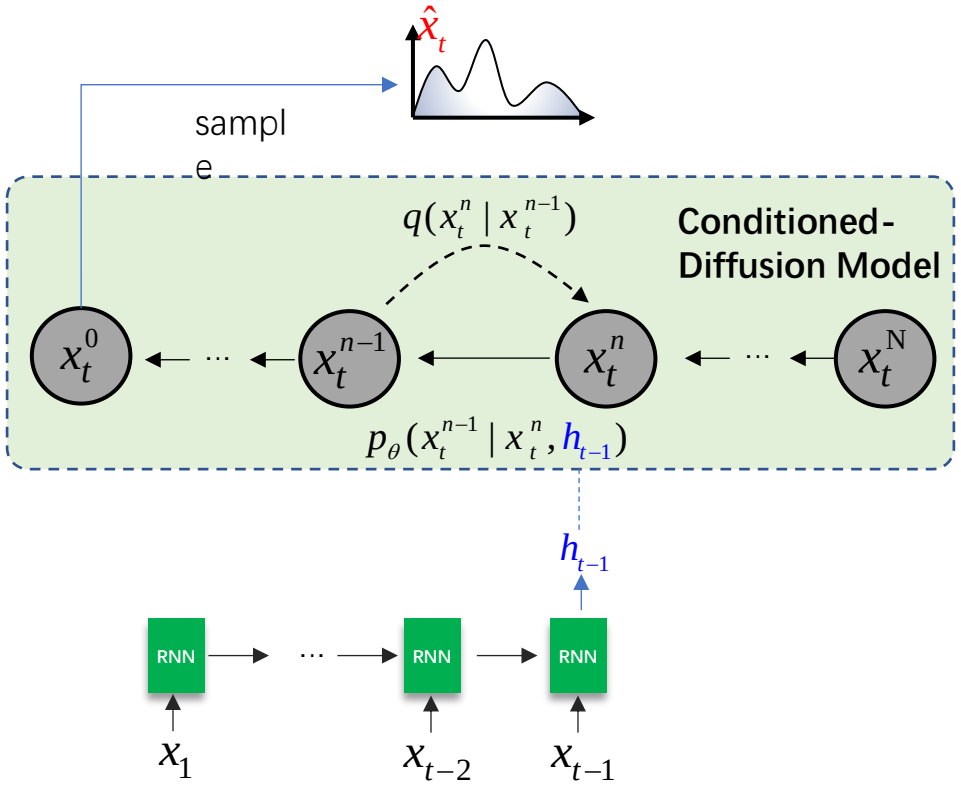
Drawbacks

1. Time-Consuming with two loops
2. No modeling of spatial relation

Traditional RNN-based Model



Diffusion-based RNN Method



Motivation

Limitations of current works

1. **Lack of effective stochastic prediction solutions** for the spatial-temporal graph.

2. Current diffusion-based time series prediction model is **ineffective for spatial correlation and inefficient**.

Our Goal

Effective Diffusion-based Probabilistic Model for Spatial-Temporal Graph Forecasting

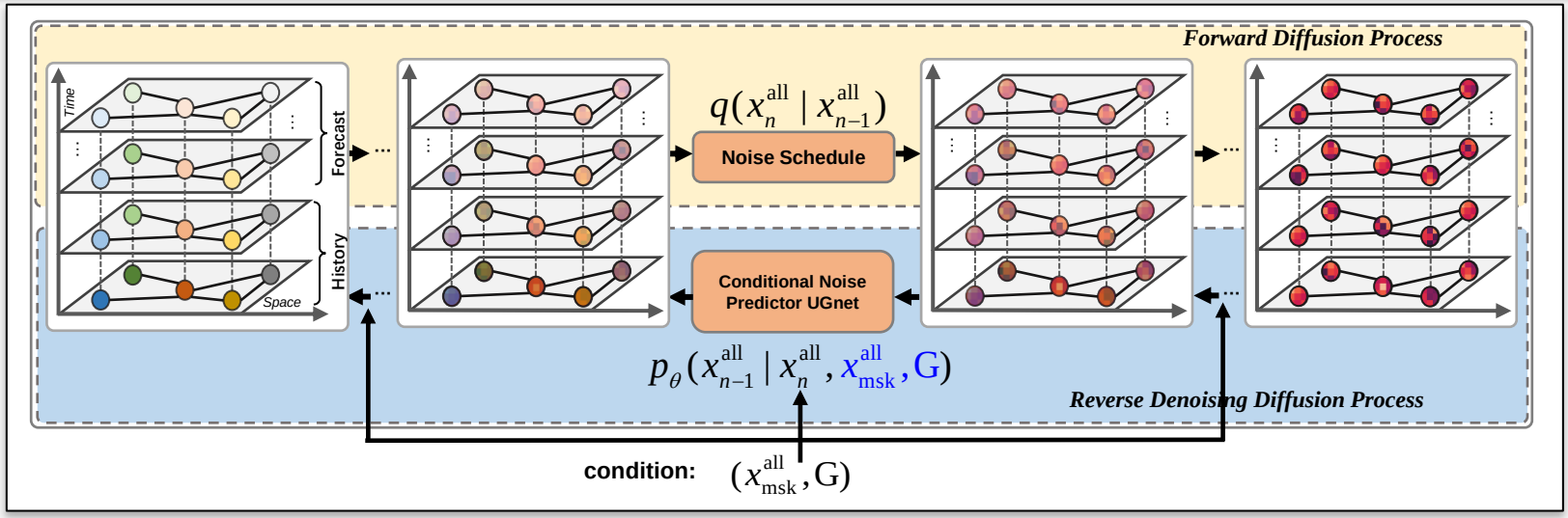
Solution: Key problems in the model

- How to generalize diffusion model to STG forecasting?
- How to capture the ST-correlation in p_θ?
- How to make it efficient in the reverse process?

Solution: Key problems in the model

- How to generalize diffusion model to STG forecasting?
- How to capture the ST-correlation in p_θ?
- How to make it efficient in the reverse process?

Solution: How to generalize Diffusion model to STG forecasting



Historical STG conditioned diffusion

$$p_\theta(\boldsymbol{x}_{0:N}^{\mathrm{p}}|\boldsymbol{x}^{\mathrm{h}}, G) := p(\boldsymbol{x}_N^{\mathrm{p}}) \prod^{1} p_\theta(\boldsymbol{x}_{n-1}^{\mathrm{p}}|\boldsymbol{x}_n^{\mathrm{p}}, \boldsymbol{x}^{\mathrm{h}}, G).$$

$$p_\theta(\boldsymbol{x}_{n-1}^{\mathrm{p}}|\boldsymbol{x}_n^{\mathrm{p}}, \boldsymbol{x}^{\mathrm{h}}, G) = \mathcal{N}(\boldsymbol{x}_{n-1}^{\mathrm{p}}; \mu_\theta(\boldsymbol{x}_n^{\mathrm{p}}, n|\boldsymbol{x}^{\mathrm{h}}, G), \sigma_\theta(\boldsymbol{x}_n^{\mathrm{p}}, n|\boldsymbol{x}^{\mathrm{h}}, G)).$$

$$\min_\theta \mathcal{L}(\theta) := \min_\theta \mathbb{E}_{\boldsymbol{x}_0^{\mathrm{p}}, \boldsymbol{\epsilon}} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\boldsymbol{x}_n^{\mathrm{p}}, n|\boldsymbol{x}^{\mathrm{h}}, G \right) \right\|_2^2,$$

Generalized STG conditioned diffusion

$$p_\theta(\boldsymbol{x}_{0:N}^{\mathrm{all}}|\boldsymbol{x}_{\mathrm{mask}}^{\mathrm{all}}, G) := p(\boldsymbol{x}_N^{\mathrm{all}}) \prod_{n=N}^{1} p_\theta(\boldsymbol{x}_{n-1}^{\mathrm{all}}|\boldsymbol{x}_n^{\mathrm{all}}, \boldsymbol{x}_{\mathrm{mask}}^{\mathrm{all}}, G)$$

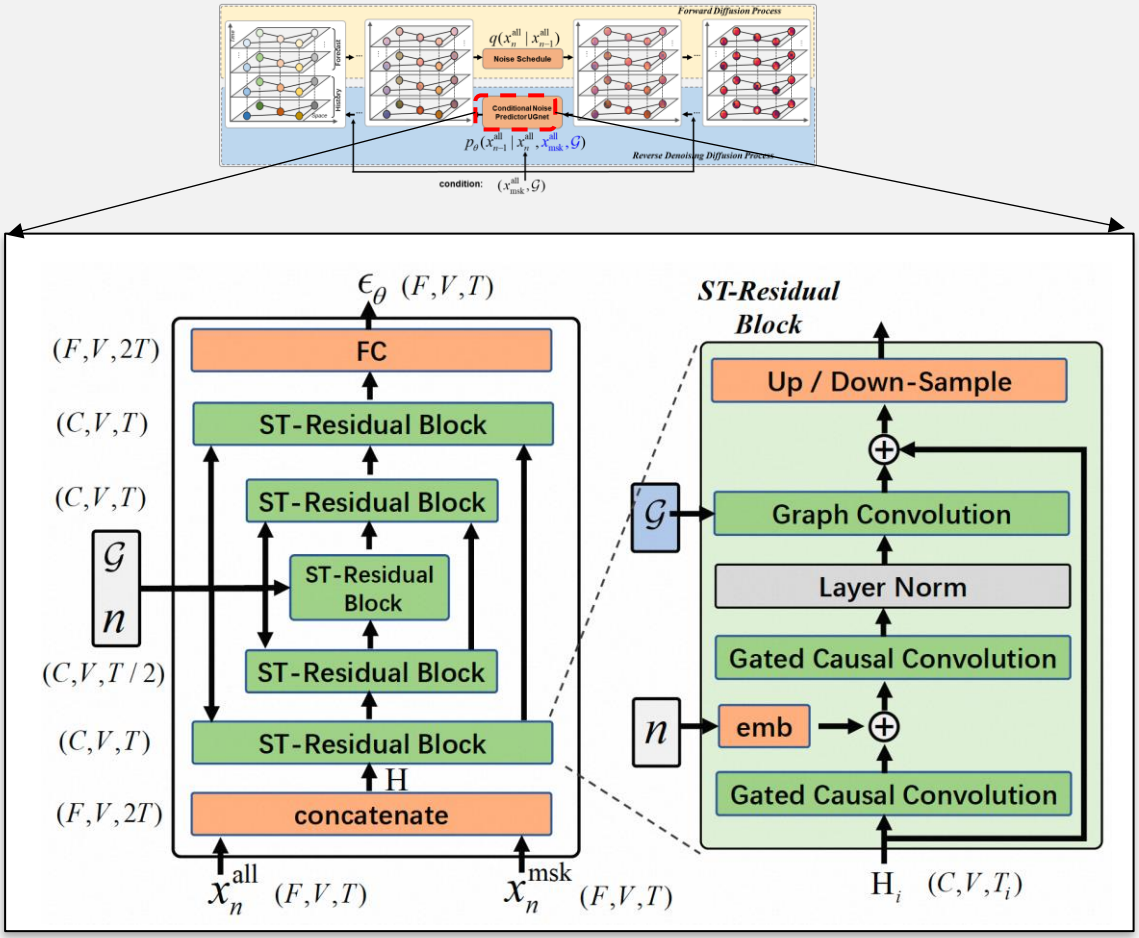
$$\min_\theta \mathcal{L}(\theta) := \min_\theta \mathbb{E}_{\boldsymbol{x}_0^{\mathrm{all}}, \boldsymbol{\epsilon}} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\boldsymbol{x}_n^{\mathrm{all}}, n|\boldsymbol{x}_{\mathrm{mask}}^{\mathrm{all}}, G \right) \right\|_2^2.$$

- Reconstruct the historical data
- Able to conduct more tasks

Solution: Key problems in the model

- How to generalize diffusion model to STG forecasting?
- How to capture the ST-correlation in p_θ?
- How to make it efficient in the reverse process?

Solution: How to capture the ST-correlation in p_θ?

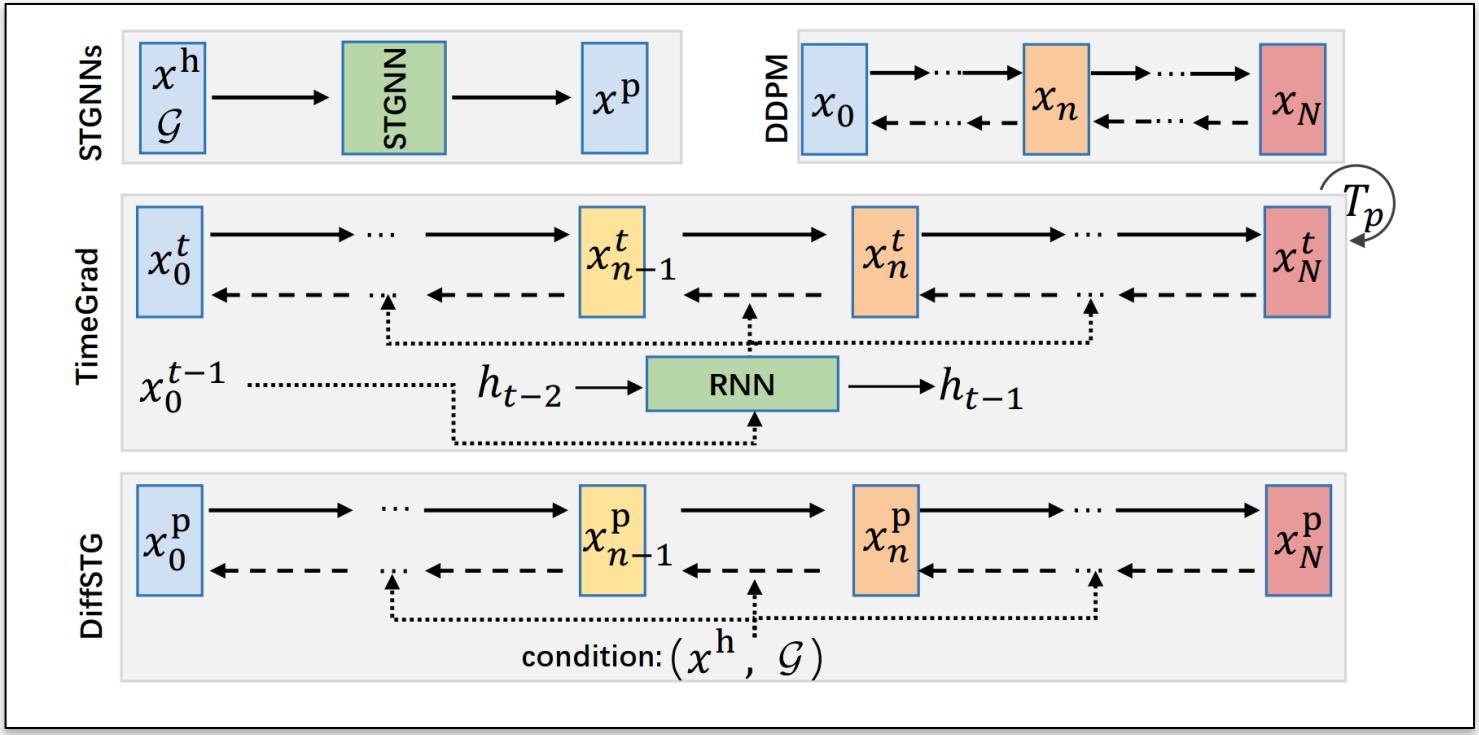


- Unet-like architecture
- Temporal conv: Down/up scale in the temporal dimension.
- Graph conv: capture the spatial correlation in the spatial dimension.

Solution: Key problems in the model

- How to generalize diffusion model to STG forecasting?
- How to capture the ST-correlation in p_θ?
- How to make it efficient in the reverse process?

Solution: How to make it efficient in the diffusion process?



- One diffusion loop, all predictions
- Skip diffusion steps: DDIM

Experiment Results

Datasets

Dataset	Nodes	F	Data Type	Time interval	#Samples
PEMS08	170	1	Traffic flow	5 minutes	17,856
AIR-BJ	34	1	$PM_{2.5}$	1 hour	8,760
AIR-GZ	41	1	$PM_{2.5}$	1 hour	8,760

Metrics

$$\mathrm{CRPS}(F, x) = \int_{\mathbb{R}} \left(F(z) - \mathbb{I}\{x \le z\}\right)^2 \mathrm{d}z,$$

$$\mathrm{MAE}(Y, \hat{Y}) = \frac{1}{|Y|} \sum_{i=1}^{|Y|} \left|Y_i - \hat{Y}_i\right|,$$

Results

Method	AIR-BJ			AIR-GZ			PEMS08		
	MAE	RMSE	CRPS	MAE	RMSE	CRPS	MAE	RMSE	CRPS
Latent ODE [30]	20.61	32.27	0.47	12.92	18.76	0.30	26.05	39.50	0.11
DeepAR [31]	20.15	32.09	0.37	11.77	17.45	<u>0.23</u>	21.56	33.37	<u>0.07</u>
CSDI [37]	26.52	40.33	0.50	13.75	19.40	0.28	32.11	47.40	0.11
TimeGrad [26]	<u>18.64</u>	<u>31.86</u>	<u>0.36</u>	12.36	18.15	0.25	24.46	38.06	0.09
MC Dropout [44]	20.80	40.54	0.45	<u>11.12</u>	<u>17.07</u>	0.25	<u>19.01</u>	<u>29.35</u>	0.07
DiffSTG (ours)	**17.88**	**29.60**	**0.34**	**10.95**	**16.66**	**0.22**	**18.60**	**28.20**	**0.06**
Error reduction	-4.1%	-7.1%	-5.6%	-1.5%	-2.4%	-4.3%	-2.2%	-3.9%	-14.3%

> DiffSTG outperforms all the probabilistic baselines

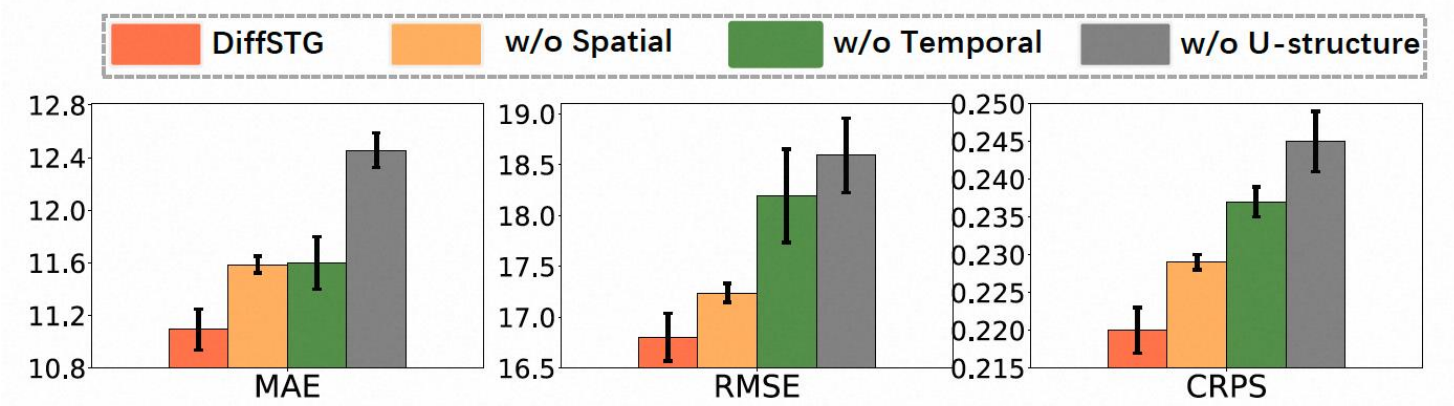
Experiment Results

Time cost of diffusion-based models

Method	$S = 8$	$S = 16$	$S = 32$
TimeGrad [26]	9.58	128.40	672.12
DiffSTG (M=100, k=1)	0.24	0.48	0.95
DiffSTG (M=40, k=1)	0.12	0.20	0.71
DiffSTG (M=40, k=2)	0.07	0.12	0.21
CSDI	0.51	0.88	1.82

DiffSTG achieves 40 times speed-up compared to TimeGrad

Ablation Study



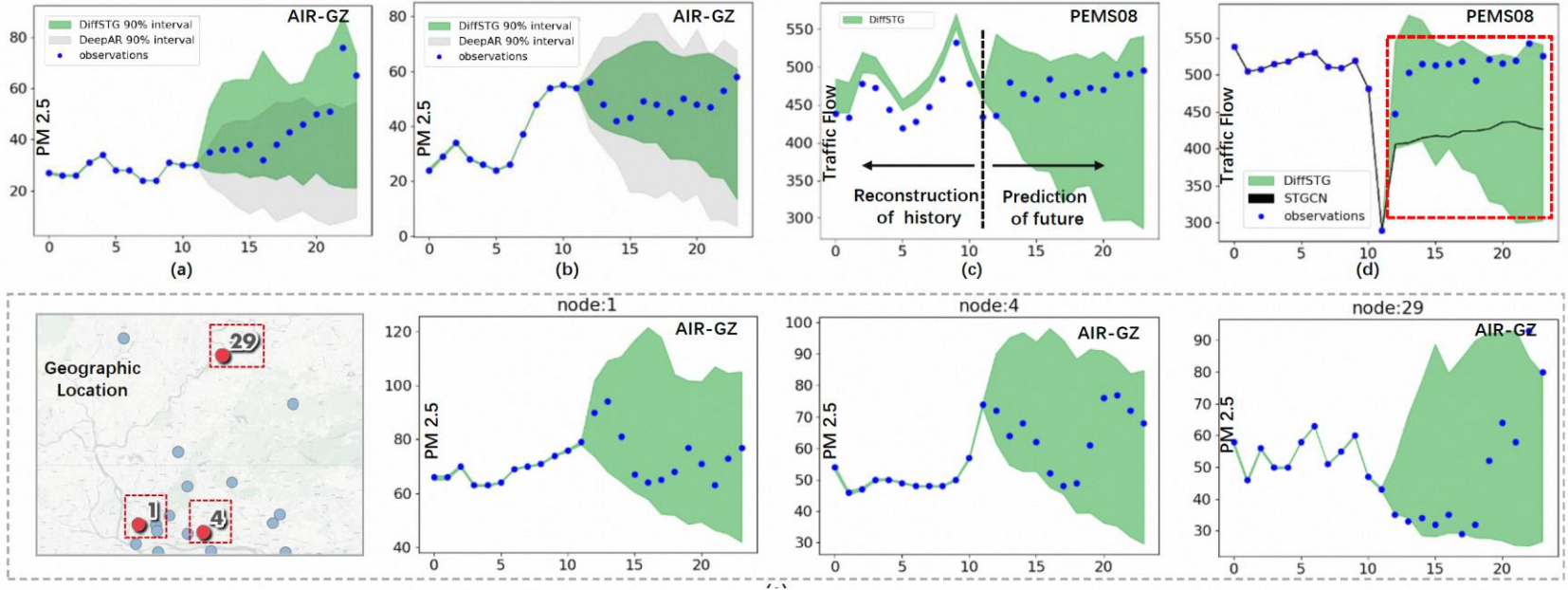
> The GNN, TCN, and the Unet-based structure are important for the performance improvement

Experiment: Case Study

DiffSTG can capture the data distribution more precisely

DiffSTG provides a more compact prediction interval

capability in history reconstruction



spatial dependency learning ability

Contribution

1.We hit the problem of **probabilistic STG forecasting** from a **score-based diffusion** perspective with the first shot. Our DiffSTG can effectively model the **complex ST dependencies and intrinsic uncertainties** within STG data.

2. We develop a novel denoising network called UGNet dedicated to STGs for the first time. It contributes as **a new and powerful member of denoising network family** for modeling ST-dependencies in STG data.

3. We empirically show that DiffSTG reduces the Continuous Ranked Probability Score (**CRPS**) by 4%-14%, and Root Mean Squared Error (**RMSE**) by 2%-7% over existing probabilistic methods on three real-world datasets.

Presented Papers

	Pre-training enhanced spatial-temporal graph neural network for multivariate time series fo	https://arxiv.org/abs/220	KDD	2022	ese Academy of Sciences, Uni	Weilin Ruan
2	Autost: Efficient neural architecture search for spatio-temporal prediction	net/pdf/AutoST_kdd20_c	SIGKDD	2020	JD	Yongkai GAO
3	What is the Human Mobility in a New City: Transfer Mobility Knowledge Across Cities	ar.nsf.gov/servlets/purl/1	WWW	2020	Harbin Institute of Technology	Tianyu Wei

Q&A

Thanks!

CityMind Lab

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