Deep Learning for Human Mobility Analytics

-- L10-Advanced Topics in Human Mobility Analytics II

Yuxuan Liang (梁宇轩)

INTR & DSA Thrust

yuxuanliang@hkust-gz.edu.cn

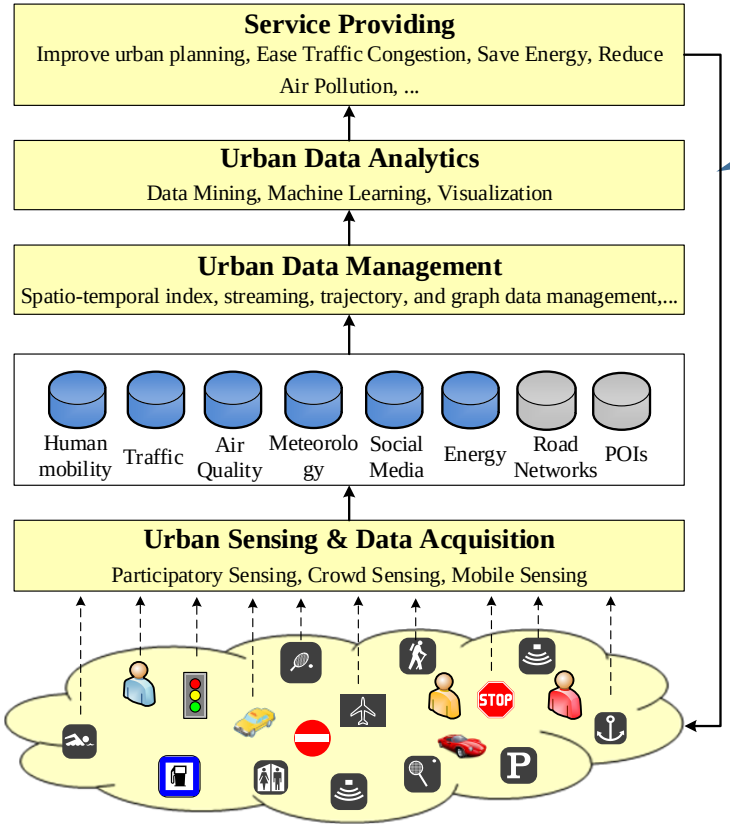
香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

Objectives of this Course

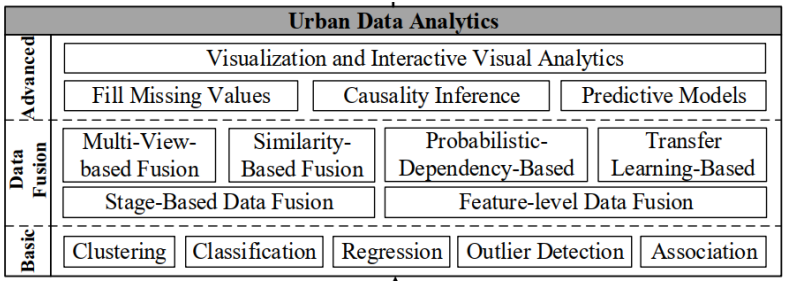
To introduce

- Generative models for Human Mobility Data
  - Adversarial training
  - Diffusion models
- Causal Learning for Human Mobility Data
- Continuous modeling for Human Mobility 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

- Generative models for Human Mobility Data
  - Adversarial training
  - Diffusion models
- Causal Learning for Human Mobility Data
- Continuous modeling for Human Mobility Data

GAN

- Generative
  - Learn a generative model
- Adversarial
  - Trained in an adversarial setting
- Network
  - Using deep neural networks

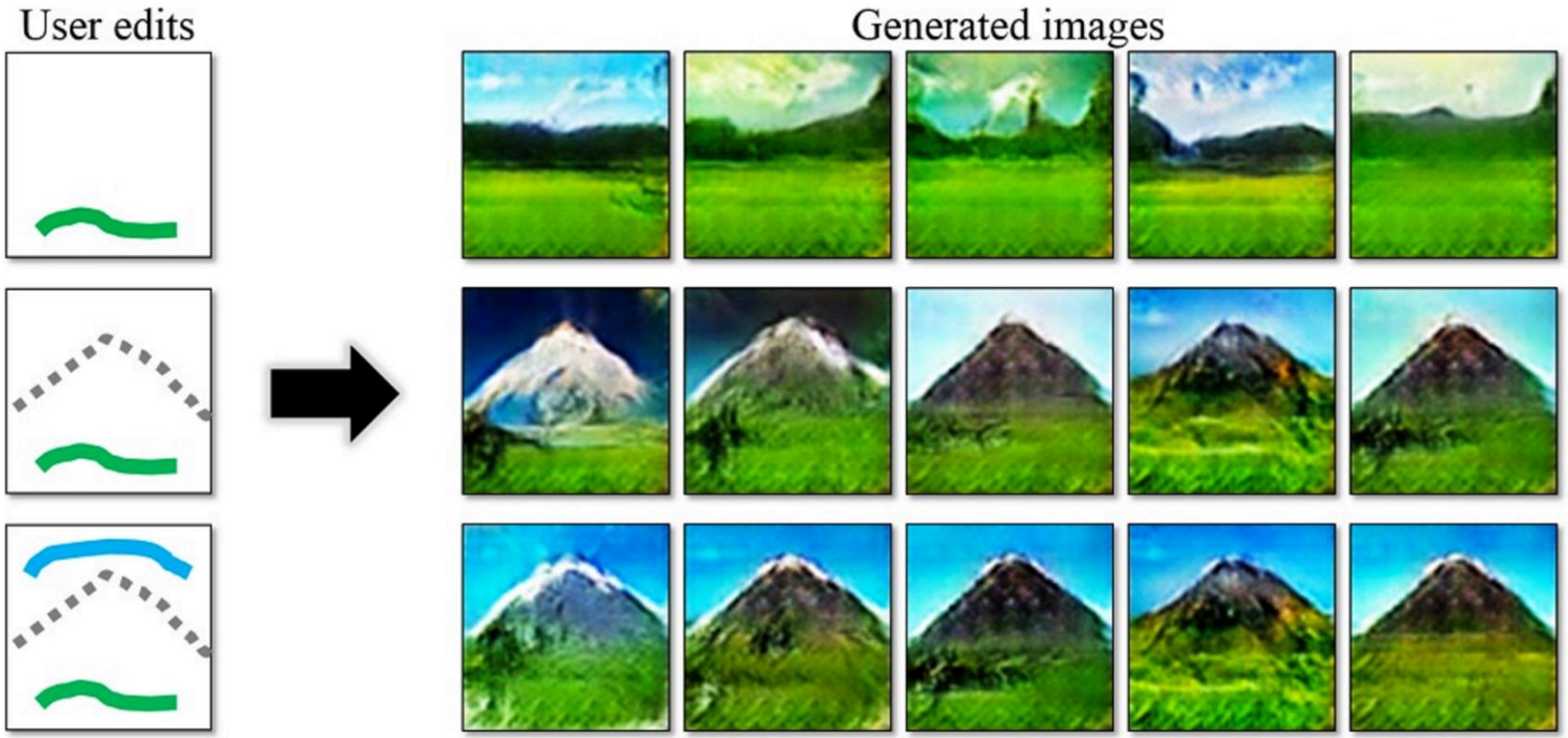
Why Generative Models?

- **We've only seen discriminative models so far**
  - Given an image **X**, predict a label **Y**
  - Estimates **P(Y|X)**

- **Discriminative models have several key limitations**
  - Can't model **P(X)**, i.e. the probability of seeing a certain image
  - Thus, can't sample from **P(X)**, i.e. **can't generate new images**

- **Generative models (in general) cope with all of above**
  - Can model **P(X)**
  - Can generate new images

Magic of GANs

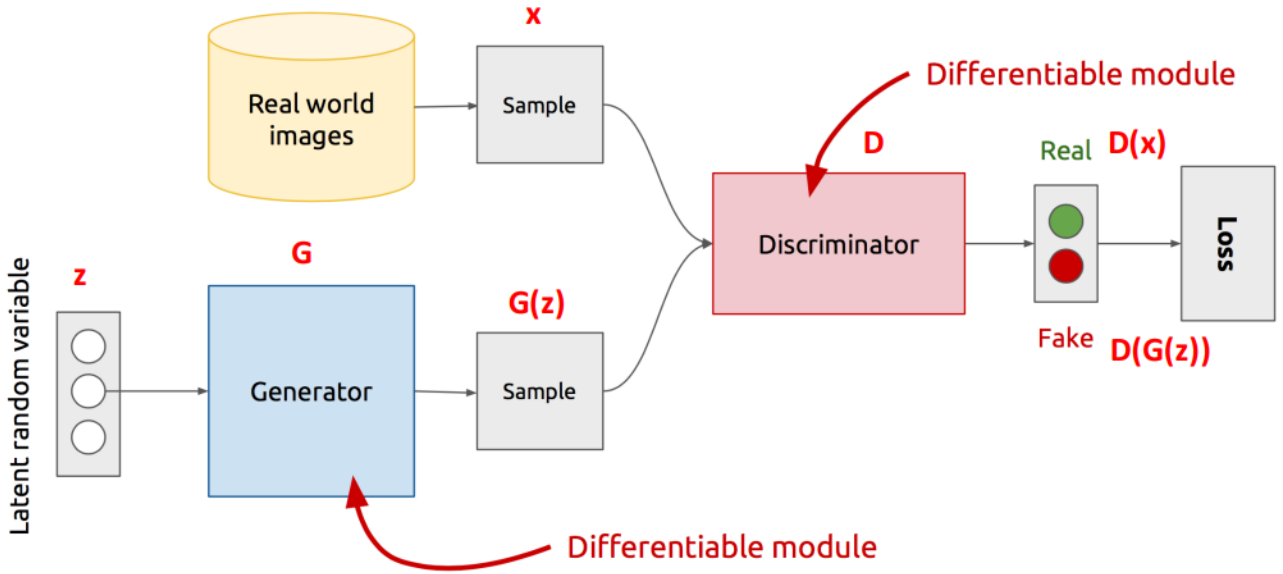


Adversarial Training

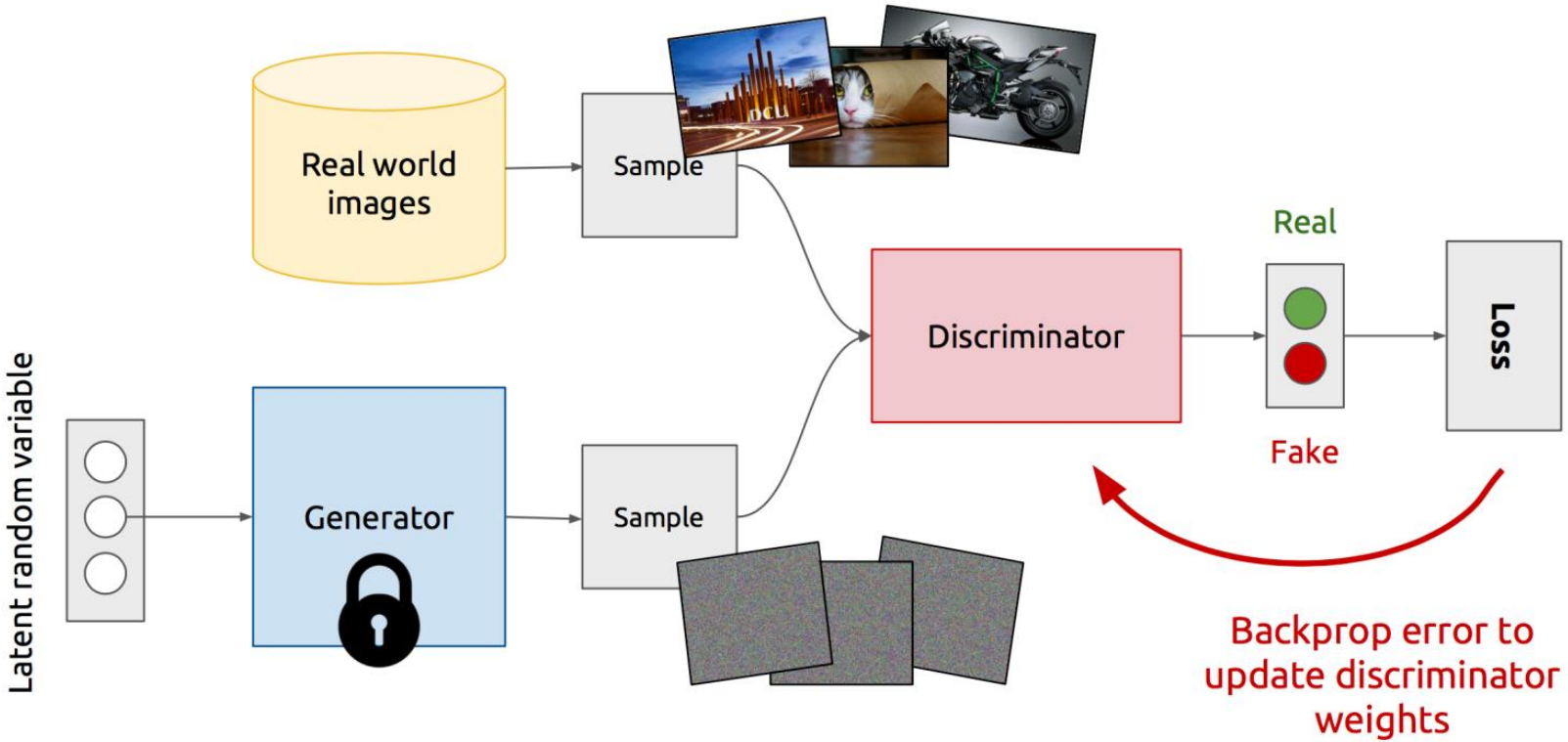
- Generator: generate fake samples, tries to fool the Discriminator
- Discriminator: tries to distinguish between real and fake samples
- Train them against each other
- Repeat this and we get better Generator and Discriminator

GAN's architecture

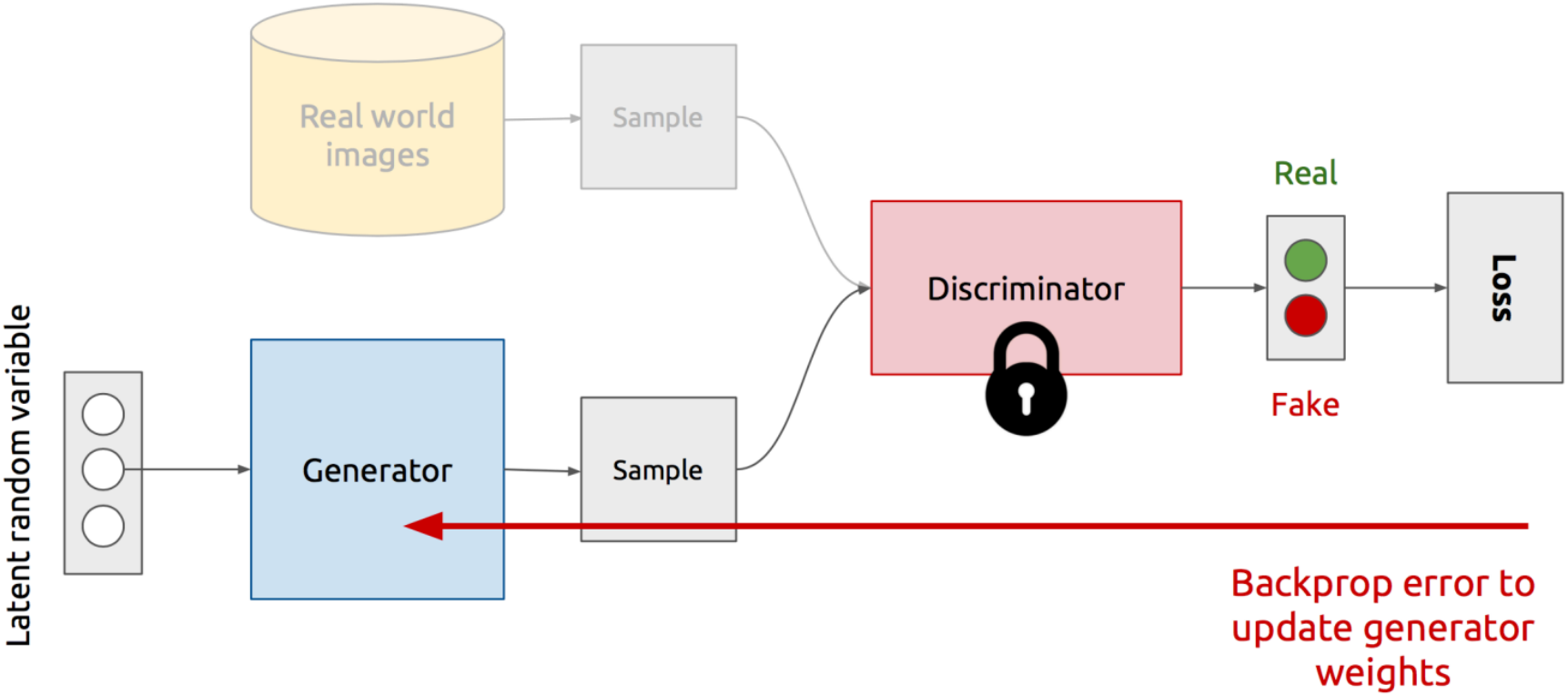
- **Z** is some random noise (Gaussian/Uniform)
- **Z** can be thought as the latent representation of the image



Training Discriminator



Training Generator



GAN's Formulation

$$\min_G \max_D V(D,G)$$

- It is formulated as a **minimax game**, where:
  - The Discriminator is trying to maximize its reward $\boldsymbol{V(D,G)}$
  - The Generator is trying to minimize Discriminator's reward (or maximize its loss)

$$V(D,G) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]$$

- The Nash equilibrium of this particular game is achieved at:
  - $P_{data}(x) = P_{gen}(x) \;\; \forall x$
  - $\mathrm{D}(x) = \frac{1}{2} \;\; \forall x$

Training Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

Discriminator updates

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

Generator updates

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

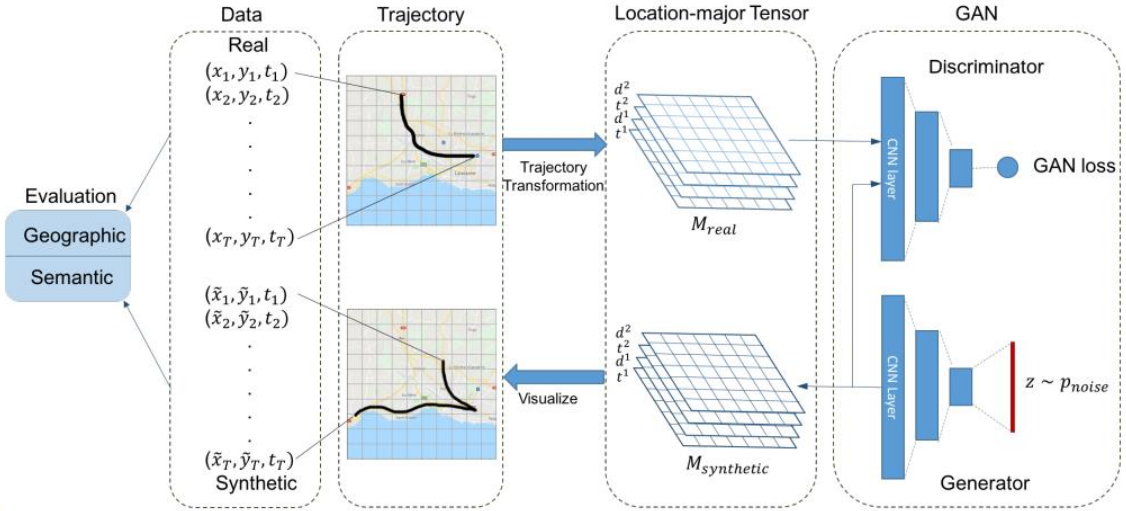
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

A Non-Parametric Generative Model for Human Trajectories

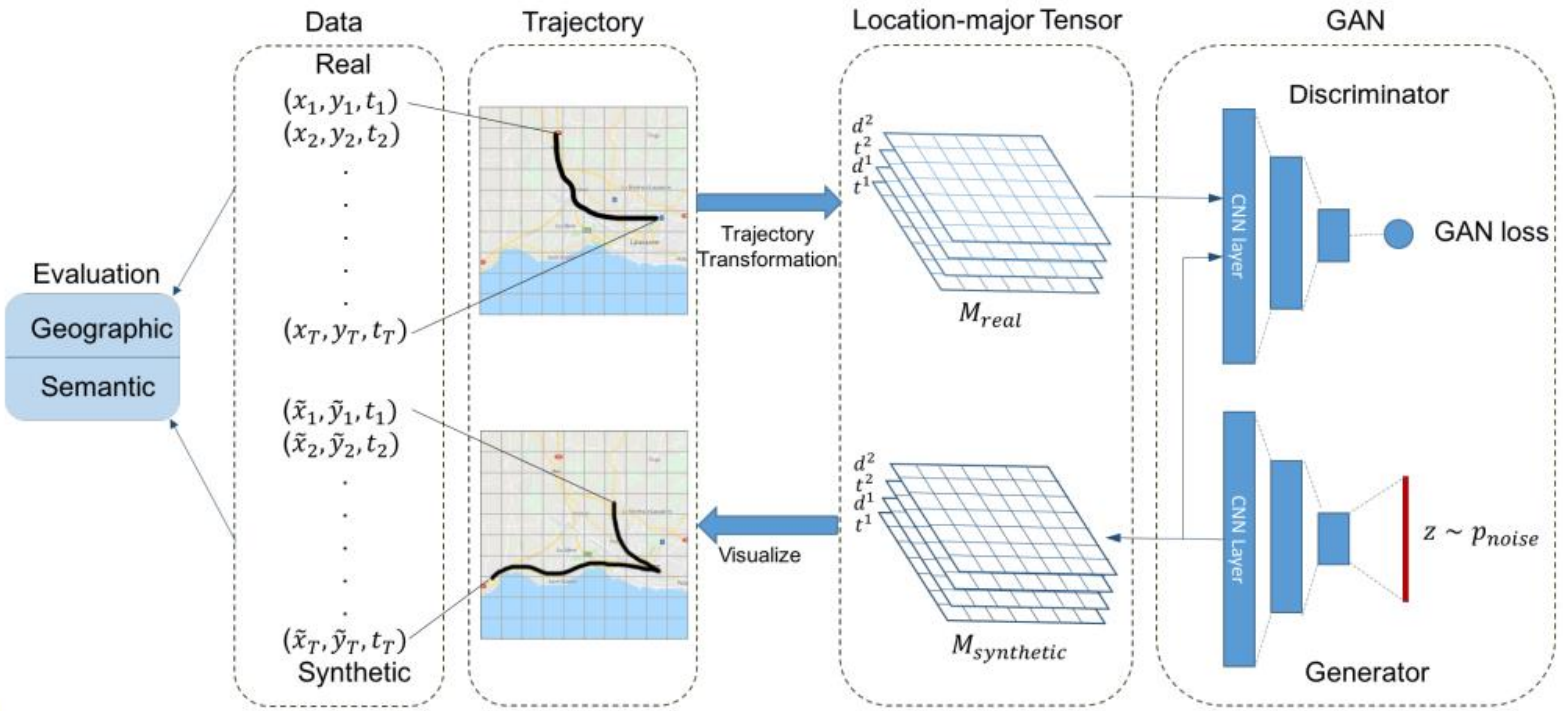
IJCAI 2018



Motivation

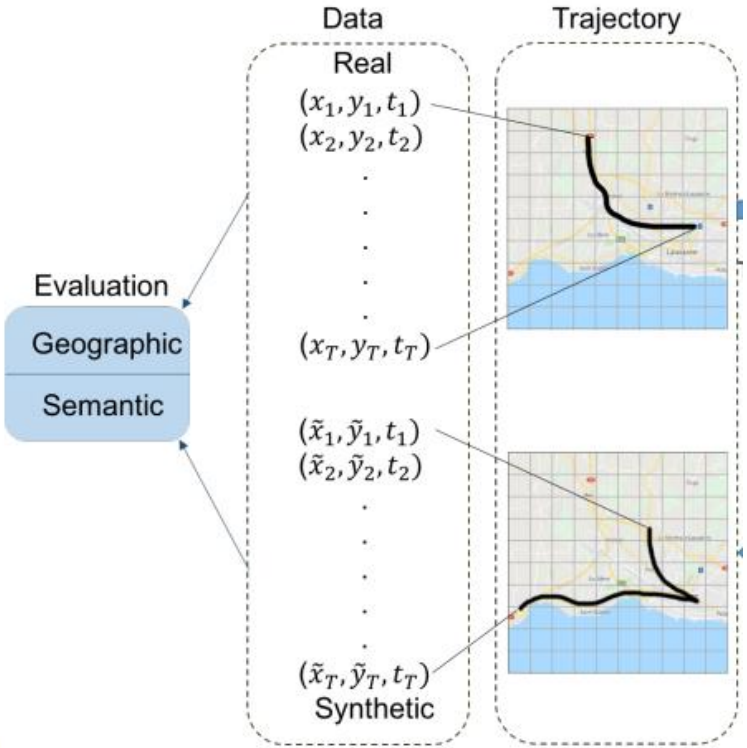
- Modeling human mobility and generating synthetic yet realistic location trajectories play a fundamental role in many privacy-aware analysis and design processes that operate on location data
- Modeling human mobility is a challenging task due to many
  - Human trajectories are extremely high dimensional
  - The trajectory of each individual is very unique in the geographical space
  - human mobilities are similar in an underlying semantic space, which gives meaning to a trajectory

Trajectory GAN



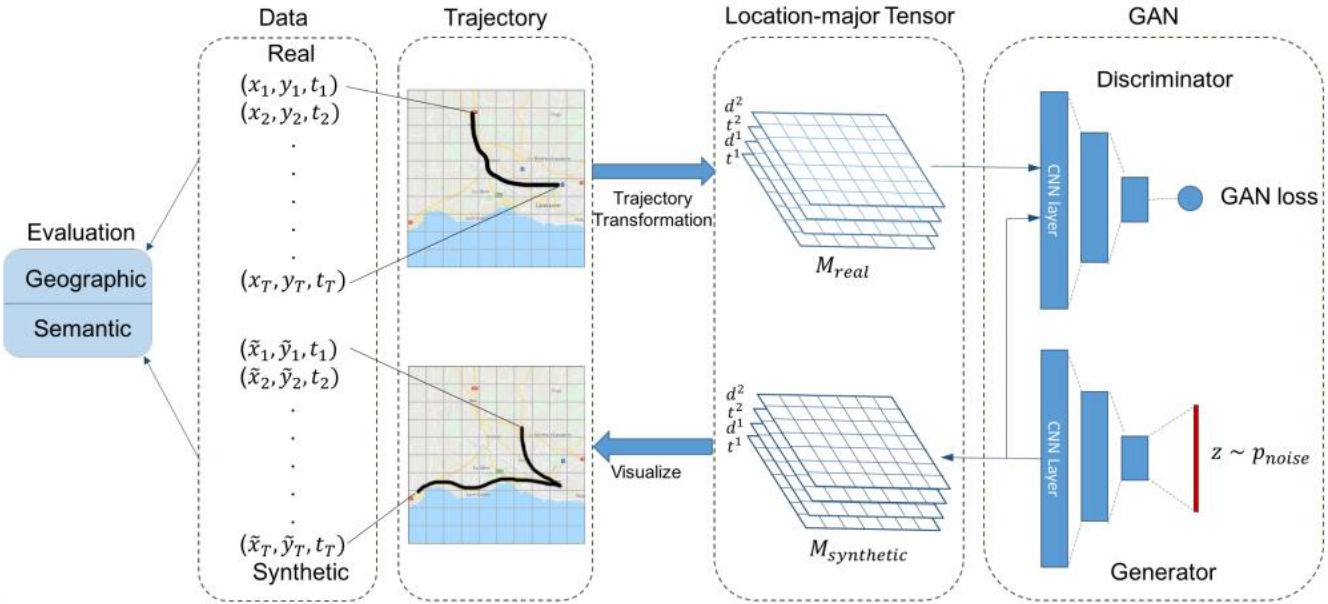
Representations

- It discretizes the region into a $N_1 \times N_2$ grid where each visit of a trajectory falls into one of these cells
- Note that
  - The human trajectory usually includes stationary activities, e.g., having lunch
  - Users might have multiple stays at the same location but at different times
- Solution: a stay-based representation



Generative Model

$$L = \mathbb{E}_{z \sim P_{\mathbf{z}}}[\mathcal{D}(\mathcal{G}(z)] - \mathbb{E}_{x \sim P_{real}}[\mathcal{D}(\mathbf{x})] + \lambda \mathbb{E}_{\tilde{x} \sim p_{\tilde{x}}}[(\|\nabla_{\tilde{\mathbf{x}}} \mathcal{D}(\tilde{\mathbf{x}})\| - 1)^2]$$



Results

	$JSD(p_{real}, p_{synthetic})$			
	$p(r)$	$p(r,t)$	$p(r,d)$	$p(r,d_{total})$
First-order MC	0.226045 (2.455)	0.377977 (1.898)	0.617615 (2.158)	0.282112 (2.235)
Time-dep MC	0.244647 (4.208)	0.383154 (2.756)	0.617357 (0.216)	0.299368 (3.789)
HMM	0.274112 (5.654)	0.438551 (3.497)	0.619812 (0.624)	0.420264 (0.819)
LSTM-MLE	0.386385 (14.140)	0.555498 (8.778)	0.642768 (6.427)	0.603640 (0.981)
Our method	**0.066109(1.818)**	**0.355129(1.779)**	**0.213122(7.308)**	**0.245102 (2.188)**

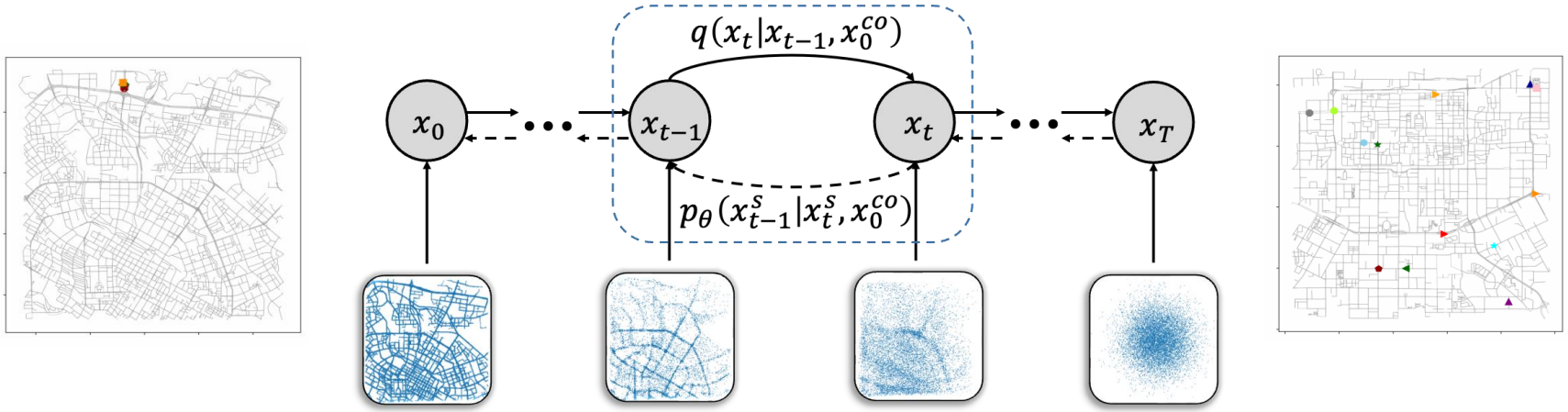
- $p(r)$ measures the visiting probability for a location r, reflecting the popularity of locations.
- $p(r,t)$ measures the visiting probability for a location r at any time t, reflecting the temporal popularity of locations (e.g. bars, homes).
- $p(r,d)$ measures the probability of visiting a location r for a duration d, reflecting the staying patterns in different places.
- $p(r,d_{total})$ measures the probability of visiting a location r for (multiple times) during a day and staying for a total duration of d_{total}, reflecting the overall importance of a location.

Outline

- Generative models for Human Mobility Data
  - Adversarial training
  - Diffusion models
- Causal Learning for Human Mobility Data
- Continuous modeling for Human Mobility Data

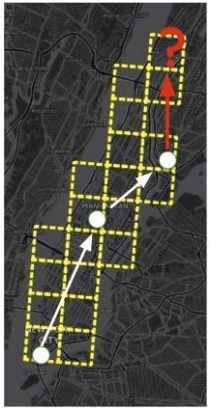
Generating GPS Trajectory with Diffusion Probabilistic Model

NeurIPS 2023



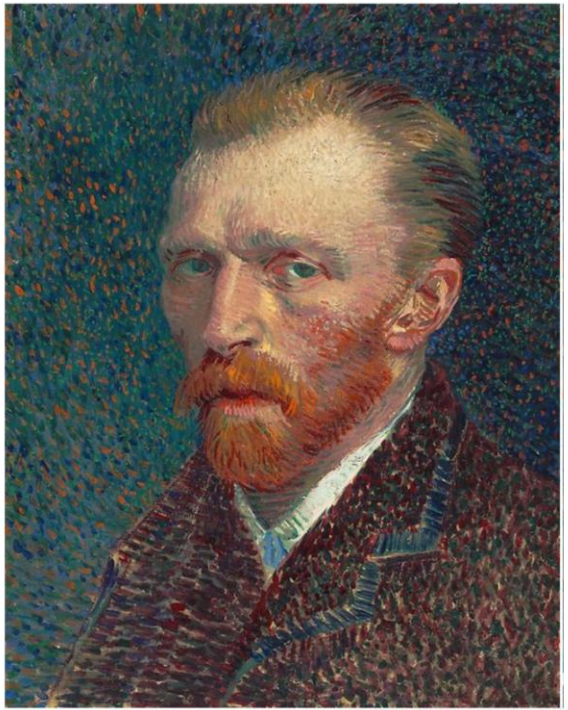
Introduction

- What is trajectory?
  - A sequence of captured spatial, temporal or spatio-temporal data.
  - Used to depict individual or group movement from one place to other place
  - Collected by different methods

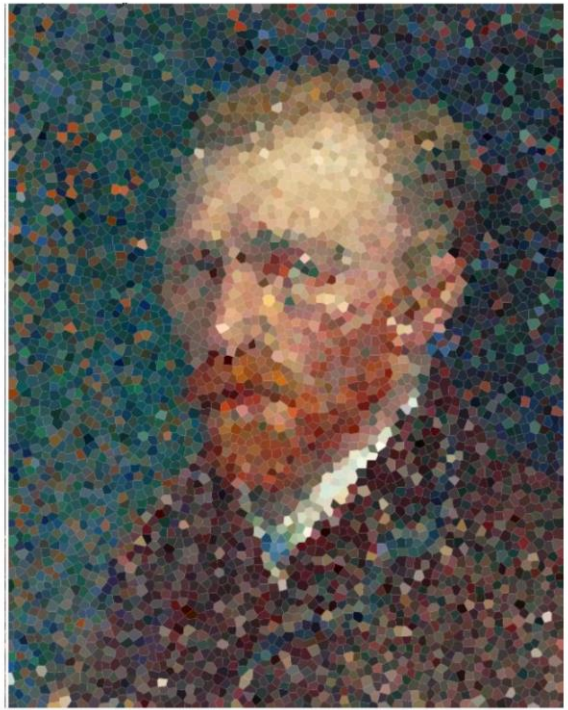


- What it can do?
  - Transportation planning and management
  - Human mobility analysis
  - Public health and epidemic prevention

Introduction



GPS Trajectory



Grids-based Trajectory

Introduction

- The current issues about GPS trajectories?
  - Personal geo-location is sensitive
  - Time-consuming and labor-intensive for collection
  - Difficulty in obtaining due to property rights restrictions
- A promise solution to address these issues
  - Generating synthetic trajectories by learning from real trajectory distributions and replacing privacy-sensitive with these synthetic counterparts

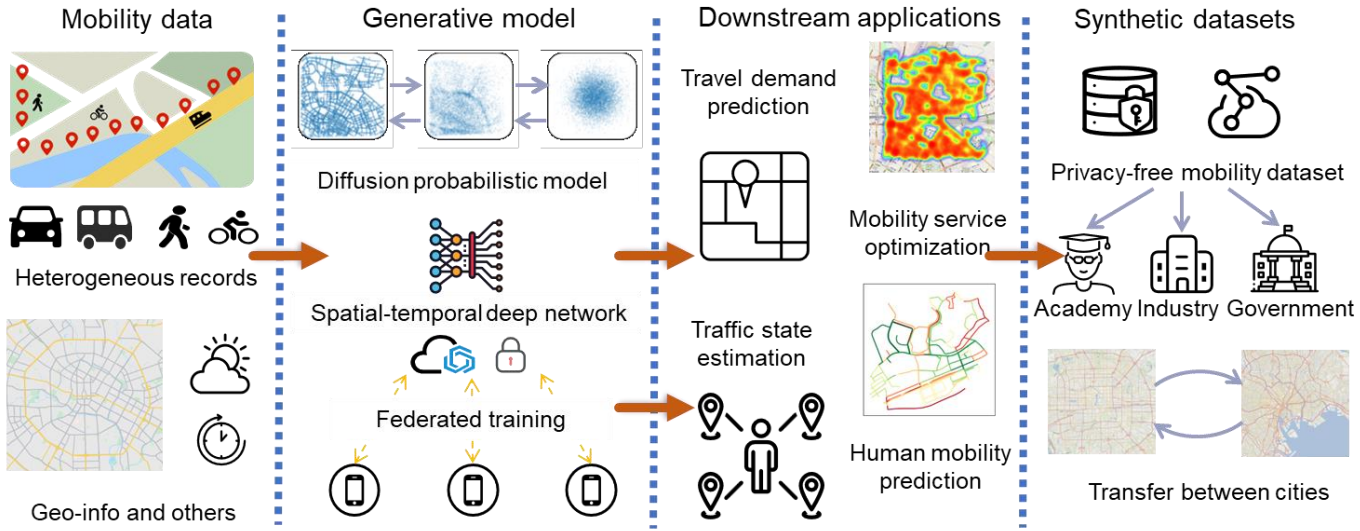
Generated by DALL.E

Introduction

- The challenge for accomplishing this task
  - Non-independent and identically distribution
  - Human activities are stochastic
  - Extra factors influent the trajectory moving

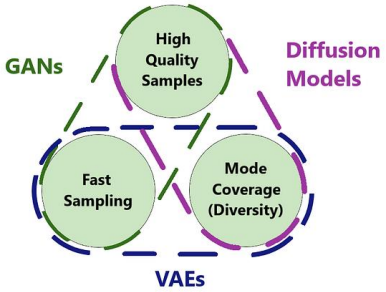
Objective for GPS Trajectories Generation

- Similarity: The generated trajectories can preserve the spatial-temporal characteristics and distribution of the real counterparts.
- Utility: The generated trajectories can maintain utility for downstream applications and analysis.
- Privacy: The generated trajectories do not reveal sensitive information associated with the individuals.

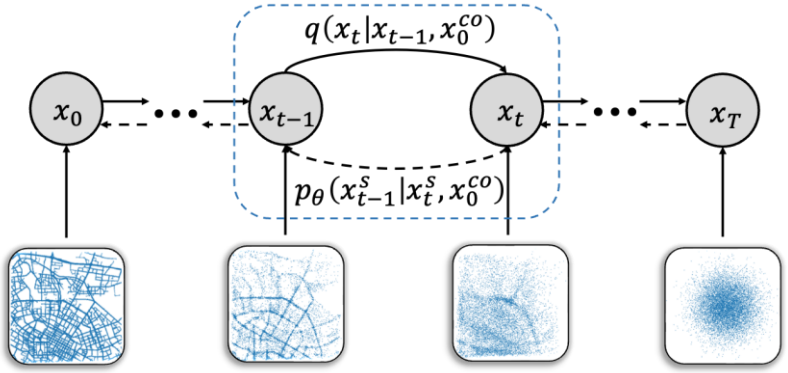


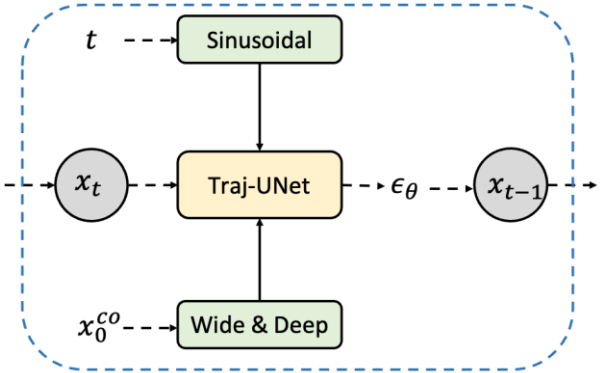
Preliminary

- GPS Trajectory
  - A sequence of GPS location points $x = \{p_1, p_2, \dots, p_n\}$, each point is a tuple $p_i = [\text{lat}_i, \text{lng}_i]$
- Trajectory Generation
  - Given a set of real-world GPS trajectories $X = \{x^1, x^2, \dots, x^m\}$.
  - Objective is to learn a generative model G, that can generate trajectories $Y = \{y^1, y^2, \dots, y^k\}$.
- Why diffusion model
  - Superior performance -- more reliable and robust method of generative tasks
  - High quality sample -- trajectory distribution and human activity is complex
  - Privacy free – generating trajectories from random noise



Diffusion Model-Based GPS Trajectory Generator





- Forward process:
  - Add Gaussian noise to trajectories

$$q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$$

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}\right)$$

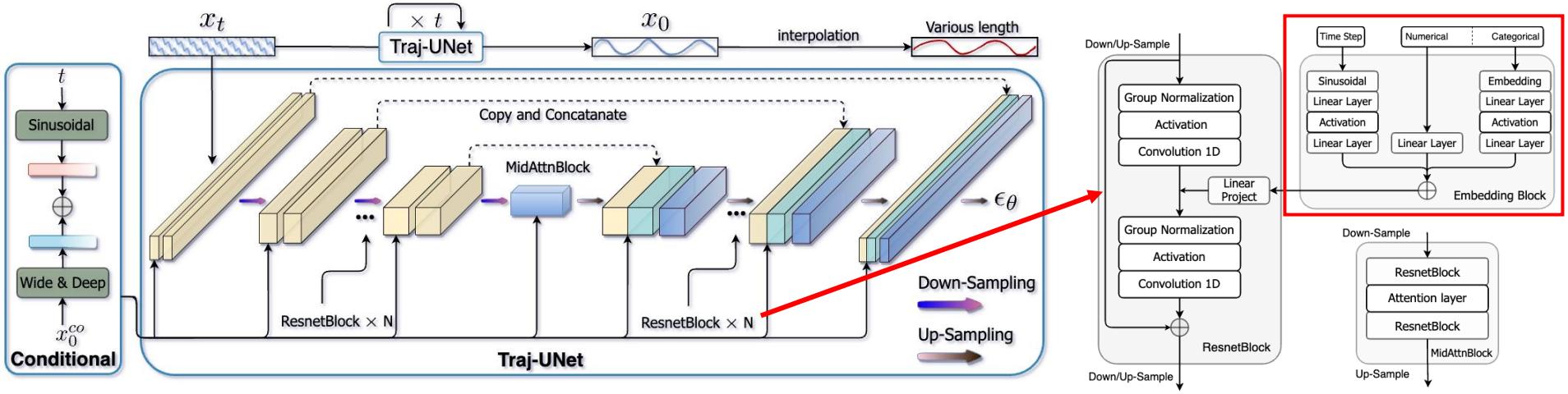
- Reverse process:
  - Recover the trajectories from the noise

$$p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

$$p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\sigma}_\theta(\boldsymbol{x}_t, t)^2 \mathbf{I}\right)$$

$$p_\theta\left(\boldsymbol{x}_{t-1}^{\mathrm{s}} \mid \boldsymbol{x}_t^{\mathrm{s}}, \boldsymbol{x}_0^{\mathrm{co}}\right) := \mathcal{N}\left(\boldsymbol{x}_{t-1}^{\mathrm{s}};\ \boldsymbol{\mu}_\theta\left(\boldsymbol{x}_t^{\mathrm{s}}, t \mid \boldsymbol{x}_0^{\mathrm{co}}\right),\ \boldsymbol{\sigma}_\theta\left(\boldsymbol{x}_t^{\mathrm{s}}, t \mid \boldsymbol{x}_0^{\mathrm{co}}\right)^2 \mathbf{I}\right)$$

Traj-UNet Architecture



Traj-UNet: Capturing local and global contextual in GPS trajectory by multi-level and enable multi-scale feature fusion.

Wide & Deep: Employed to effectively embeding conditional information, such as departure time, trip distance, trip time.

Experiment Setups

- Dataset
- Metrics
  - Density error：measures the geo-distribution between the entire generated trajectory and the real one
  - Trip error：measures the distributed differences between trip start and destination areas
  - Length error：focuses on the differences in real and synthetic trajectory lengths
  - Pattern score：measures the pattern similarity of the generated trajectories

(a) Chengdu Trajectory (b) Chengdu Heatmap (c) Xi'an Trajectory (d) Xi'an Heatmap

Quantitatively Results

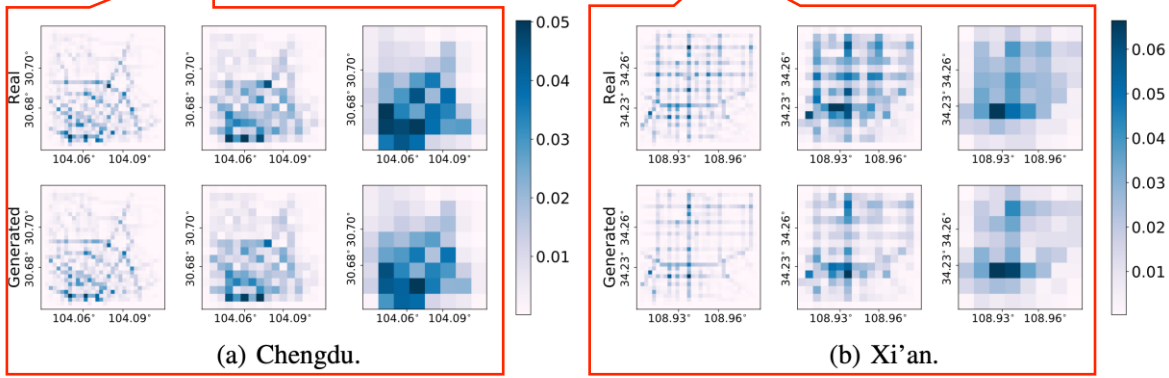
1. Generative models, VAE and TrajGAN, show better performance than RP and GP but are still inferior to DiffTraj (or DiffTraj-wo/Con).

2. Diff-LSTM achieves good results in some metrics compared to the model without UNet, but falls short of DiffTraj due to the differences in the backbone network.

3. The model without the Traj-UNet structure performs unfavorably.

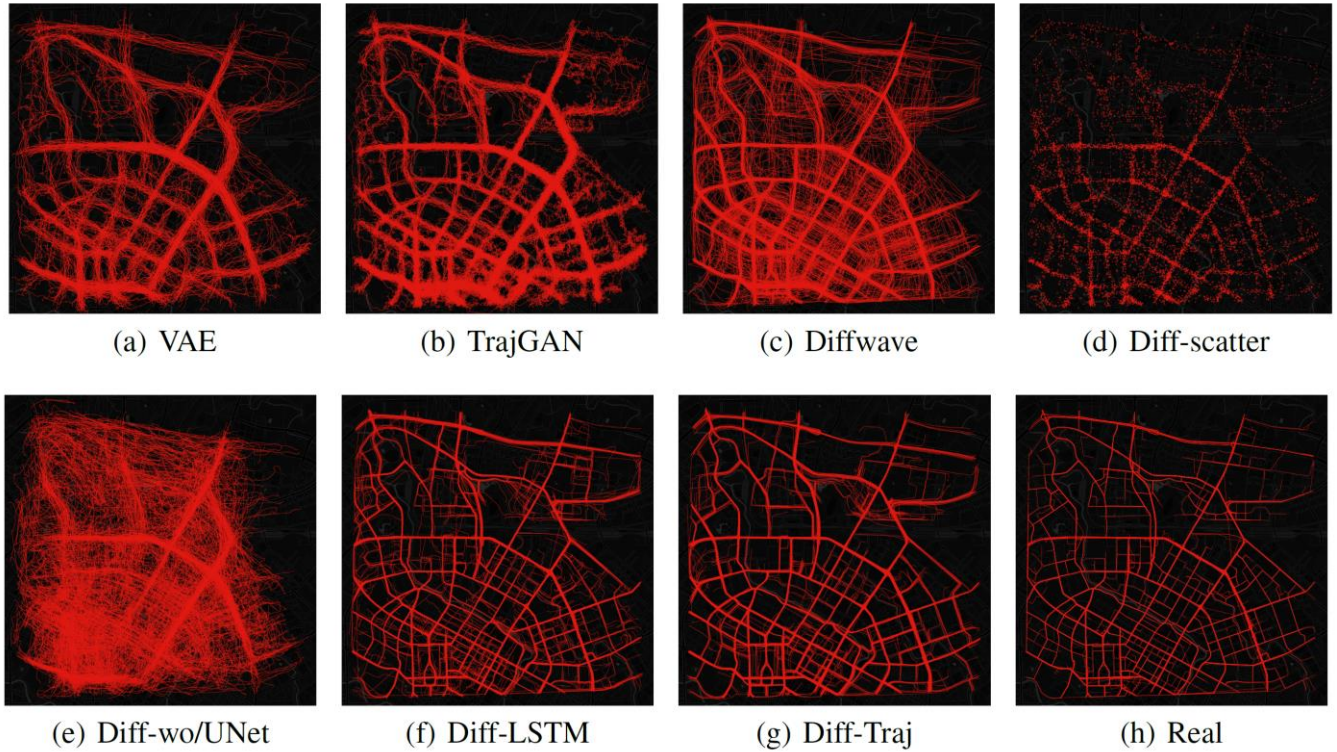
4. DiffTraj can generate high-quality trajectories and retain the original distribution

Methods	Chengdu				Xi'an			
	Density (↓)	Trip (↓)	Length (↓)	Pattern (↑)	Density (↓)	Trip (↓)	Length (↓)	Pattern (↑)
RP	0.0698	0.0835	0.2337	0.493	0.0543	0.0744	0.2067	0.381
GP	0.1365	0.1590	0.1423	0.233	0.0928	0.1013	0.2164	0.233
VAE	0.0148	0.0452	0.0383	0.356	0.0237	0.0608	0.0497	0.531
TrajGAN	0.0125	0.0497	0.0388	0.502	0.0220	0.0512	0.0386	0.565
DP-TrajGAN	0.0117	0.0443	0.0221	0.706	0.0207	0.0498	0.0436	0.664
Diffwave	0.0145	0.0253	0.0315	0.741	0.0213	0.0343	0.0321	0.574
Diff-scatter	0.0209	0.0685	–	–	0.0693	0.0762	–	–
Diff-wo/UNet	0.0356	0.0868	0.0378	0.422	0.0364	0.0832	0.0396	0.367
DiffTraj-wo/Con	0.0072	0.0239	0.0376	0.643	0.0138	0.0209	0.0357	0.692
Diff-LSTM	0.0068	0.0199	0.0217	0.737	0.0142	0.0195	0.0259	0.706
DiffTraj	**0.0055**	**0.0154**	**0.0169**	**0.823**	**0.0126**	**0.0165**	**0.0203**	**0.764**



(a) Chengdu. (b) Xi'an.

Visualization Results

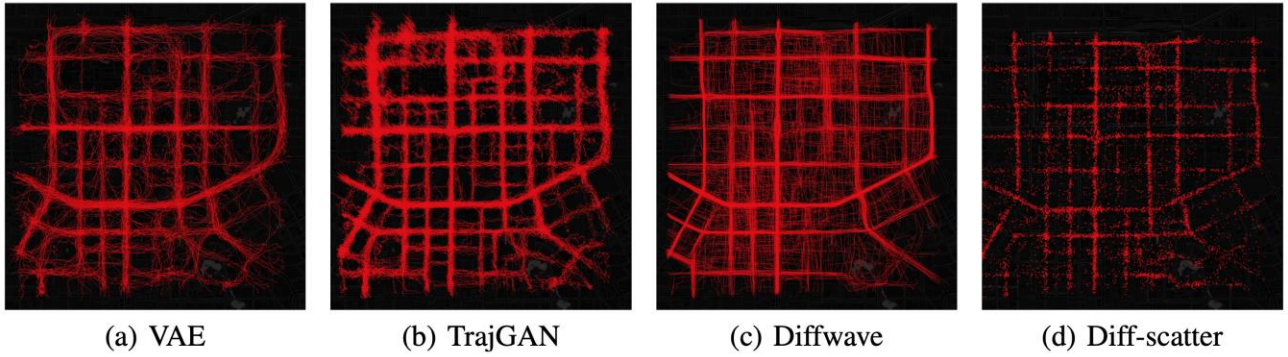


(a) VAE (b) TrajGAN (c) Diffwave (d) Diff-scatter

(e) Diff-wo/UNet (f) Diff-LSTM (g) Diff-Traj (h) Real

The generated trajectories are able to portray city profiles, and also accurately match the real roads.

Visualization Results

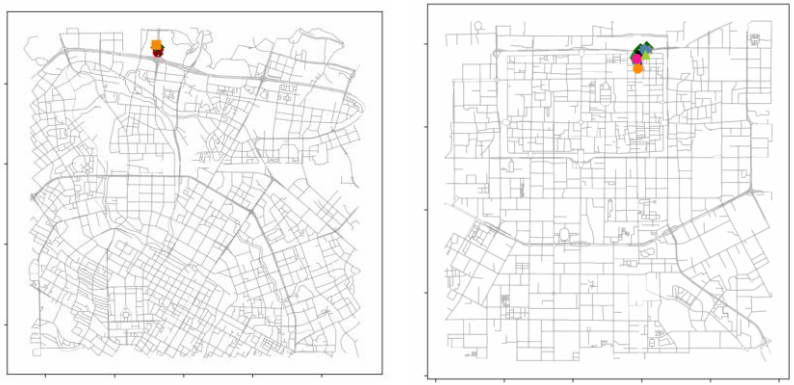


(a) VAE (b) TrajGAN (c) Diffwave (d) Diff-scatter

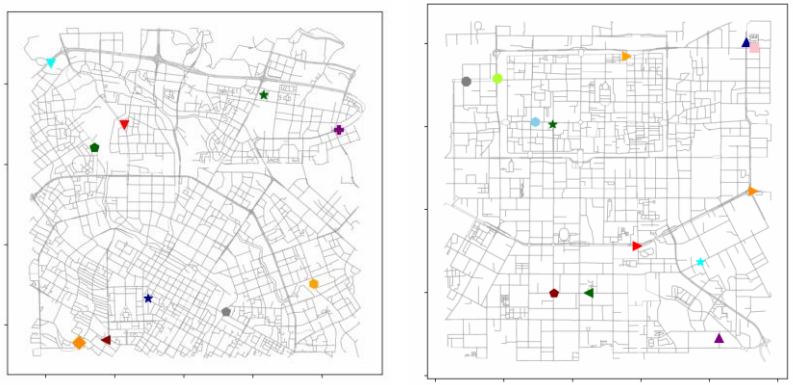
(e) Diff-wo/UNet (f) Diff-LSTM (g) Diff-Traj (h) Real

The generated trajectories are able to portray city profiles, and also accurately match the real roads.

Visualization Results



Same start and destination areas



Different start and destination areas

Utility of Generated Data

- Travel demand prediction
  - **Problem**: predict the vehicle inflow or outflow x_d^t for a given area d at time t (time-series prediction)

$$[x^{t-H+1}, \dots, x^t] \rightarrow [x^{t+1}, \dots, x^{t+h}]$$

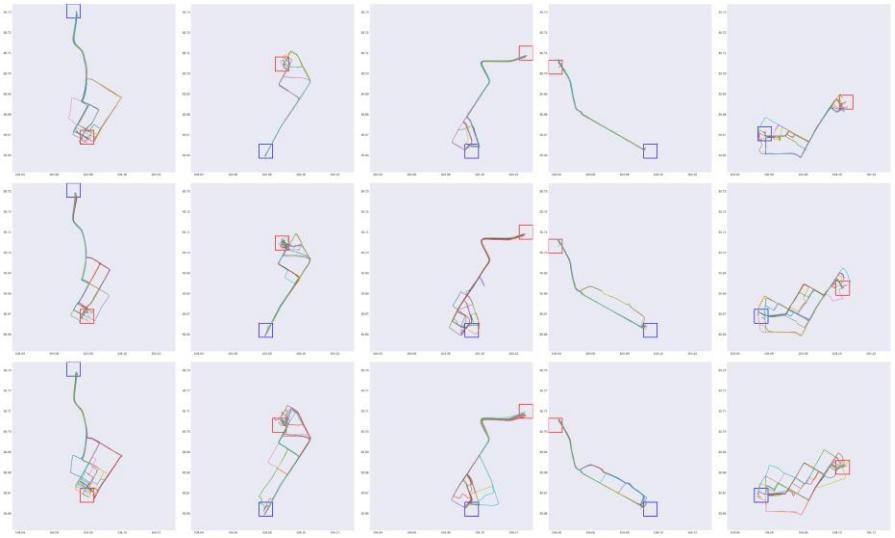
  - **Methods**: representative ST prediction model

Table 2: Data utility comparison by inflow/outflow prediction.

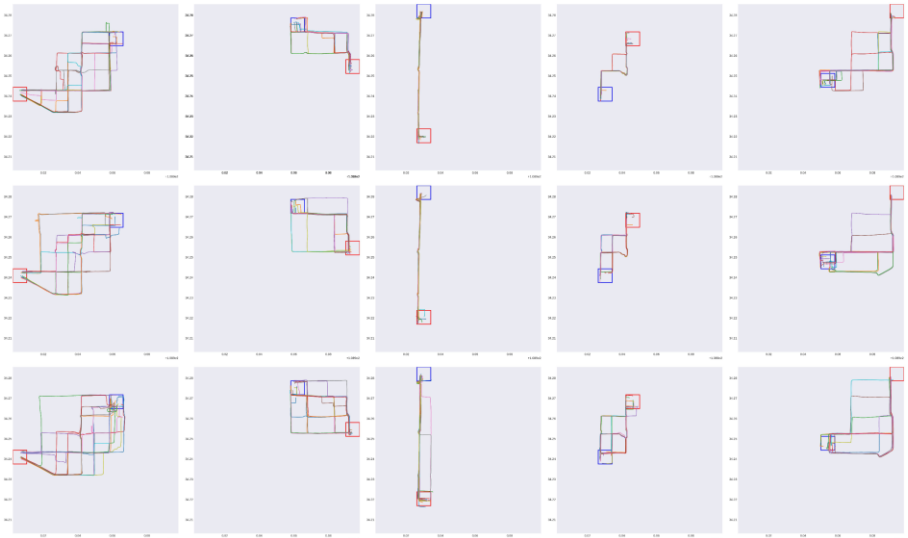
Task	Inflow (origin/generated)				Outflow (origin/generated)			
Methods	AGCRN	GWNet	DCRNN	MTGNN	AGCRN	GWNet	DCRNN	MTGNN
MSE	4.33/4.55	4.42/4.50	4.45/4.62	4.28/4.56	4.45/4.70	4.64/4.72	4.72/5.01	4.39/4.77
RMSE	2.08/2.13	2.10/2.12	2.11/2.15	2.07/2.13	2.11/2.16	2.15/2.17	2.17/2.24	2.10/2.18
MAE	1.49/1.52	1.50/1.50	1.51/1.53	1.48/1.51	1.50/1.54	1.53/1.54	1.53/1.57	1.49/1.53

Conditional Generation & Diversity Control

We can control the starting (destination) area and the diversity of generated trajectories.



$\omega = 0.1$



$\omega = 1$

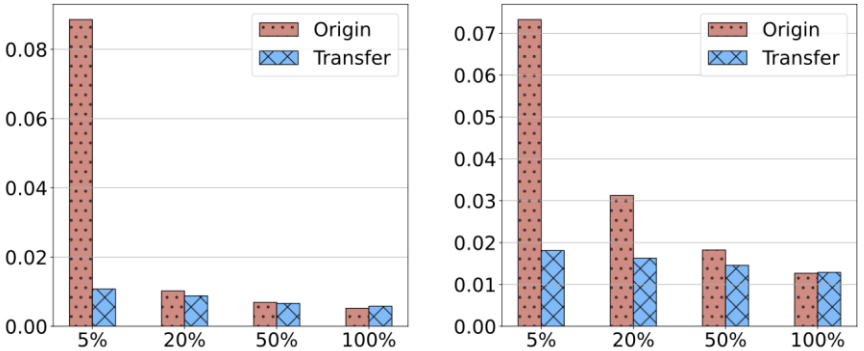
$\omega = 10$

$$\boldsymbol{\epsilon}_\theta = (1+\omega)\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t^{\mathrm{s}}, t \mid \boldsymbol{x}_0^{\mathrm{co}}\right) - \omega\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t^{\mathrm{s}}, t \mid \varnothing\right)$$

red square: start area

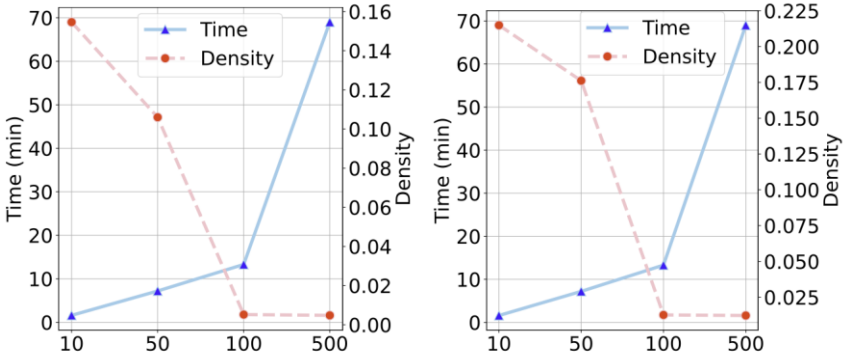
blue square: destination area

Transfer Learning & Speed up Sampling



(a) Chengdu (b) Xi'an

Transfer learning



(c) Chengdu (d) Xi'an

Speed up sampling

Only using 5% of the data, the transfer learning model achieves a significantly lower error compared to the original one.

The DiffTraj model matches the outcomes of the no skipped steps method at T = 100, saving 81% of the time cost.

Main Process Visualization

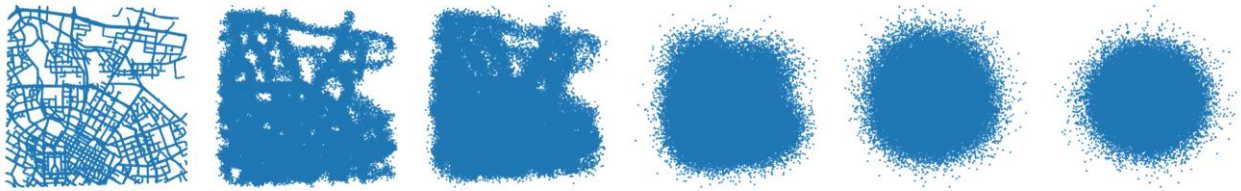


Figure 12: Forward trajectory noising process (Chengdu).

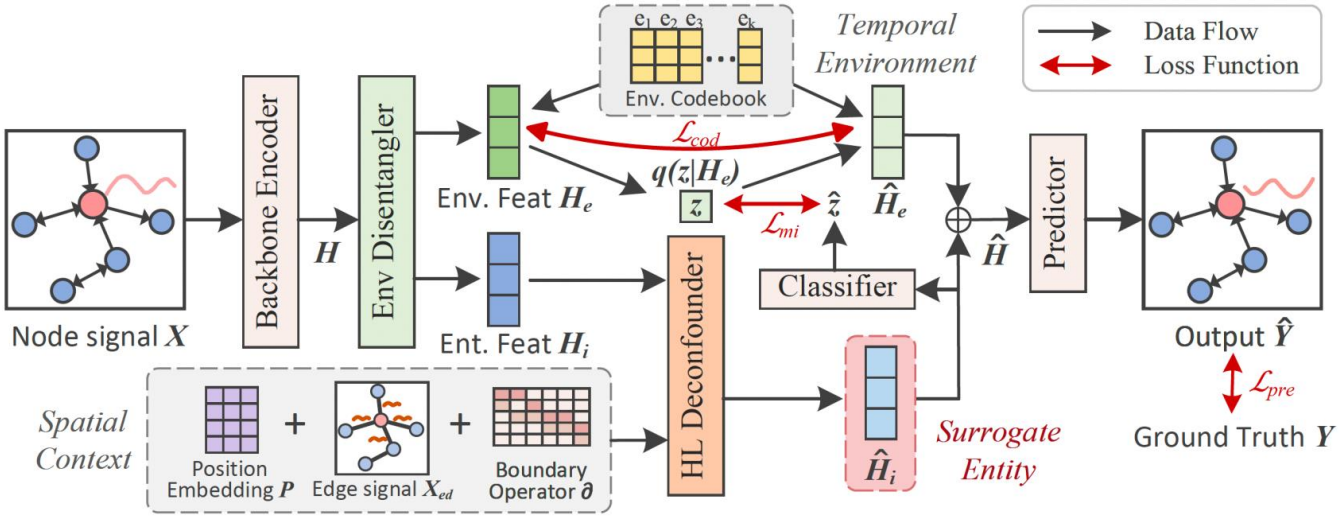
Figure 13: Reverse trajectory denoising process (Chengdu).

Outline

- Generative models for Human Mobility Data
  - Adversarial training
  - Diffusion models
- Causal Learning for Human Mobility Data
- Continuous modeling for Human Mobility Data

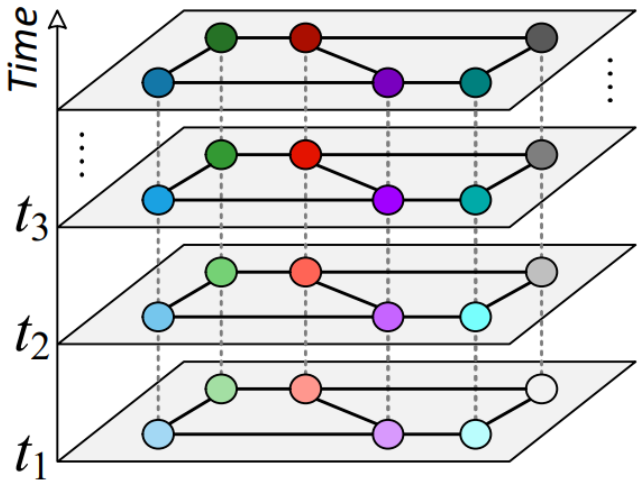
Deciphering Spatio-Temporal Graph Forecasting: A Causal Lens and Treatment

NeurIPS 2023



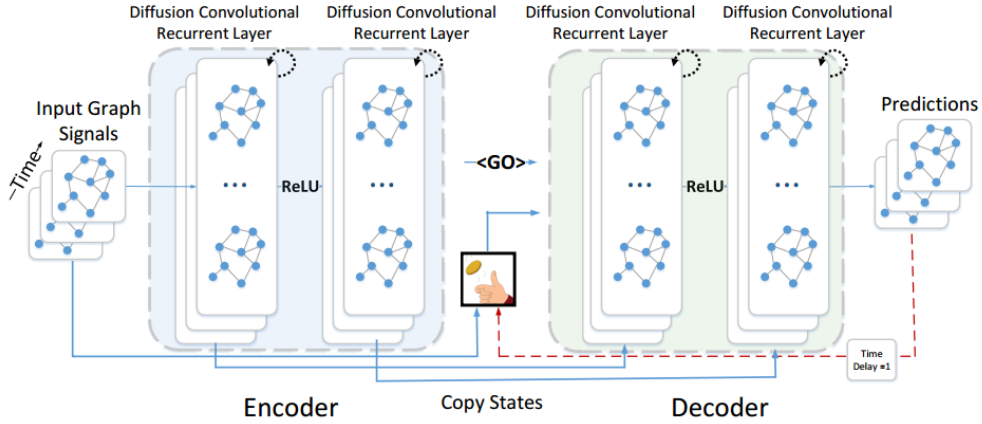
Definition of ST Graphs

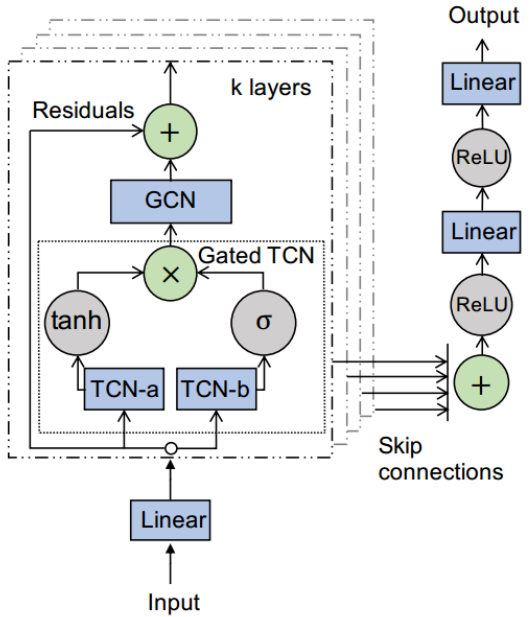
- There are numerous sensors deployed in the physical world
- Properties
  - Each sensor has a unique geospatial location
  - Constantly reporting time series readings
  - With structural correlation between readings
    - Usually represented as graphs
- Examples
  - Traffic speed/flow over road networks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



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





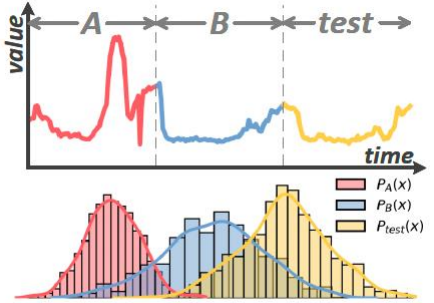
G. Jin, **Y. Liang*** et al., Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey. TKDE 2023.

Challenges

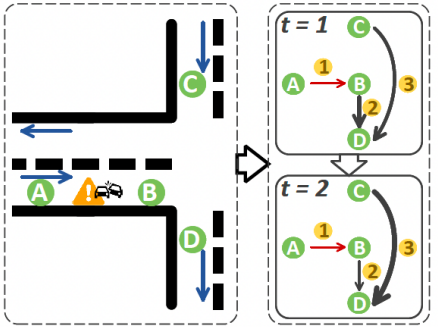
- Temporal Distribution Shift

$$P_A(x) \neq P_B(x) \neq P_{test}(x)$$

- Dynamic Spatial Causation
  - Existing work:
    - Distance-based adjacency matrices
    - Attention mechanism
  - However, the ripple effect of causations



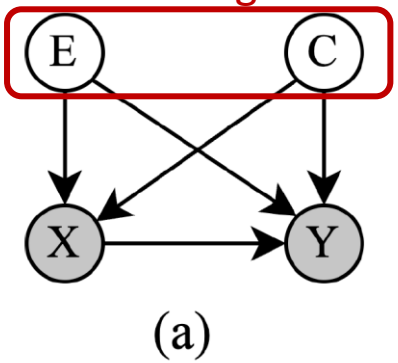
(a) Temporal Distribution Shift



(b) Ripple Effect of Causations

A Causal Lens

Confounding factors



(a)

E: temporal environment
C: spatial context
X: historical signal
Y: future signal

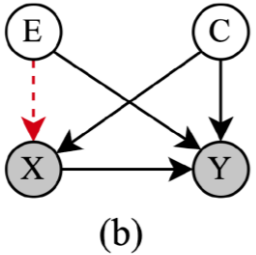
Backdoor paths

- **X ← E → Y** The temporal OoD can arise due to changes in external variables over time. (e.g., weather can affect traffic flow observations)
- **X ← C → Y** X and Y are intrinsically affected by the surrounding spatial context, comprising both spurious and genuine causal components.
- **X → Y** Our primary goal.

Causal Treatments

- Back-door adjustment for E

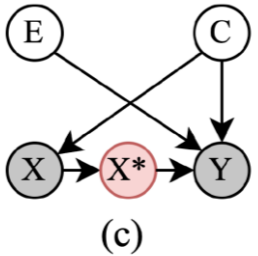
$$
\begin{aligned}
P(Y|do(X)) &= \sum\nolimits_e P(Y|do(X), E=e)P(E=e|do(X)) \\
&= \sum\nolimits_e P(Y|do(X), E=e)P(E=e) \\
&= \sum\nolimits_e P(Y|X, E=e)P(E=e)
\end{aligned}
$$



(b)

- Front-door adjustment for C

$$
\begin{aligned}
P(Y|do(X)) &= \sum\nolimits_{x^*} P(Y|do(X^*=x^*))P(X^*=x^*|do(X)) \\
&= \sum\nolimits_{x^*} \sum\nolimits_{x'} P(Y|X^*=x^*, X=x')P(X=x')P(X^*=x^*|do(X)) \\
&= \sum\nolimits_{x^*} \sum\nolimits_{x'} P(X^*=x^*|X)P(Y|X^*=x^*, X=x')P(X=x')
\end{aligned}
$$

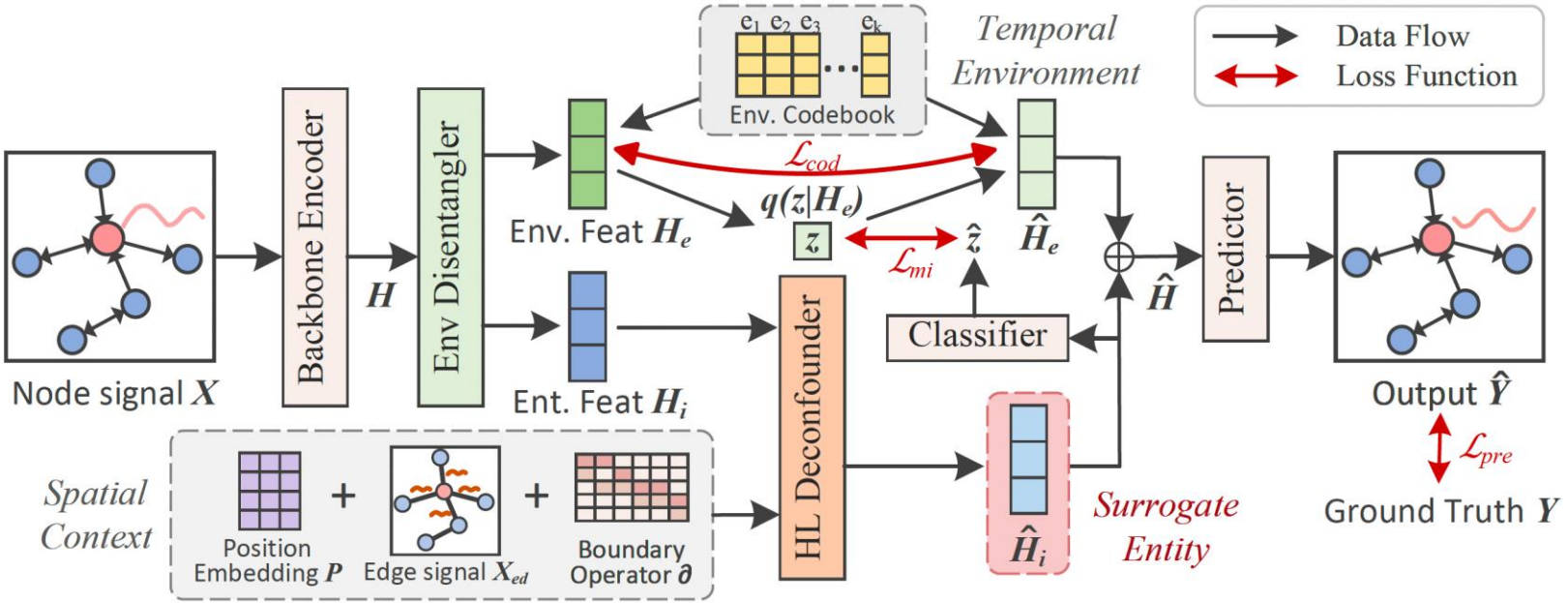


(c)

Y. Xia, **Y. Liang*** et al., Deciphering Spatio-Temporal Graph Forecasting: A Causal Lens and Treatment. NeurIPS 2023.

Model Instantiations

- Causal Spatio-Temporal neural network (CaST)



Y. Xia, **Y. Liang*** et al., Deciphering Spatio-Temporal Graph Forecasting: A Causal Lens and Treatment. NeurIPS 2023.

Model Instantiations

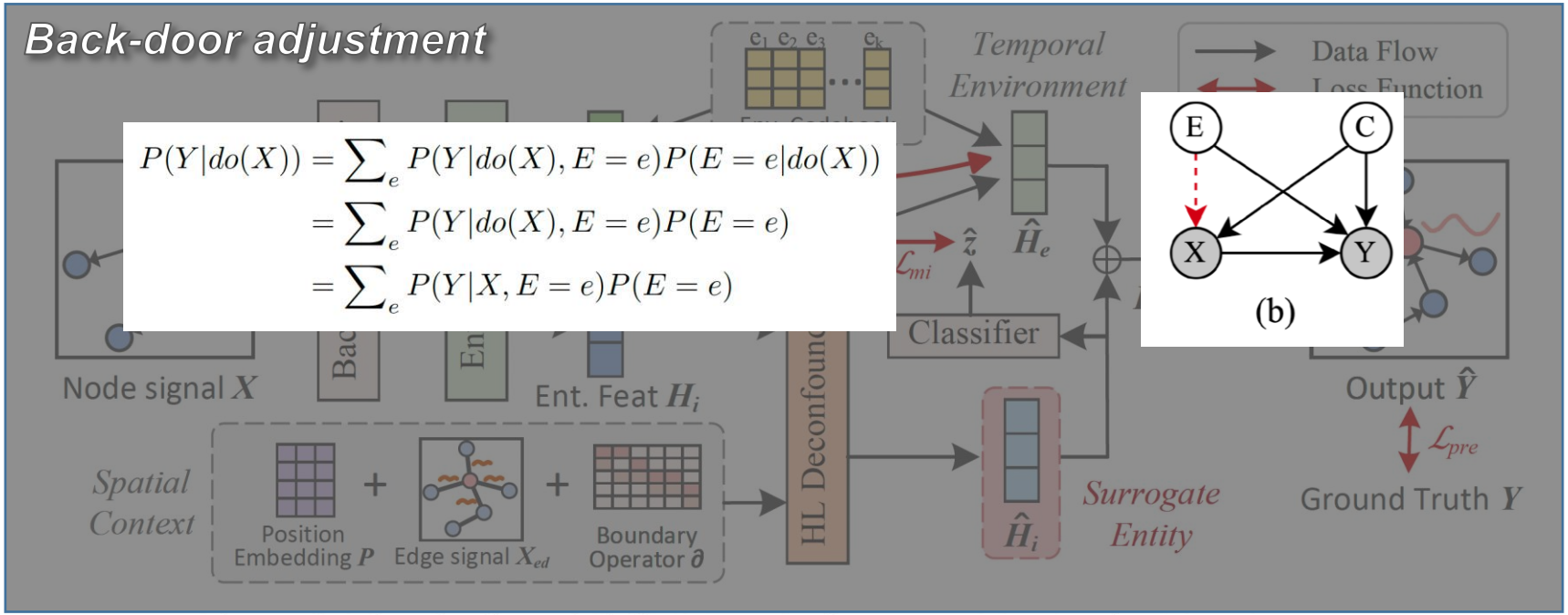


Figure 3: The pipeline of CaST. Env: Environment. Ent: Entity. Feat: Feature.

Model Instantiations

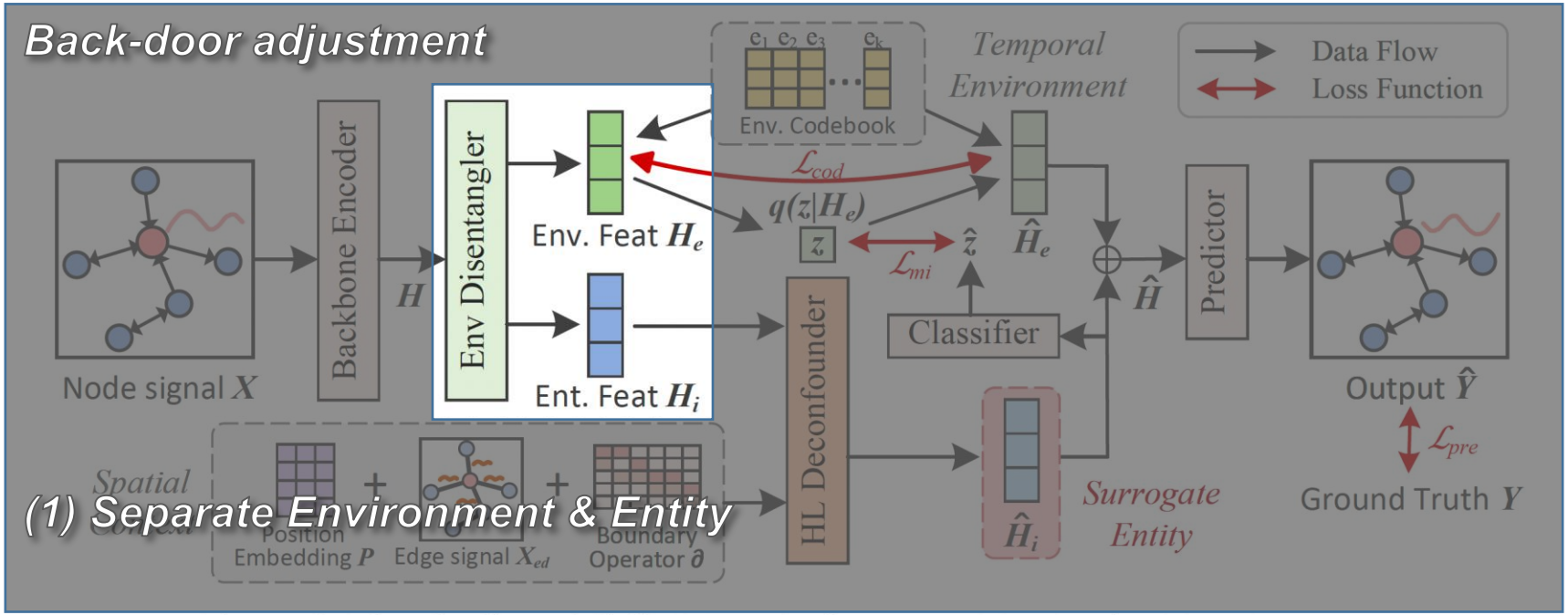


Figure 3: The pipeline of CaST. Env: Environment. Ent: Entity. Feat: Feature.

Model Instantiations

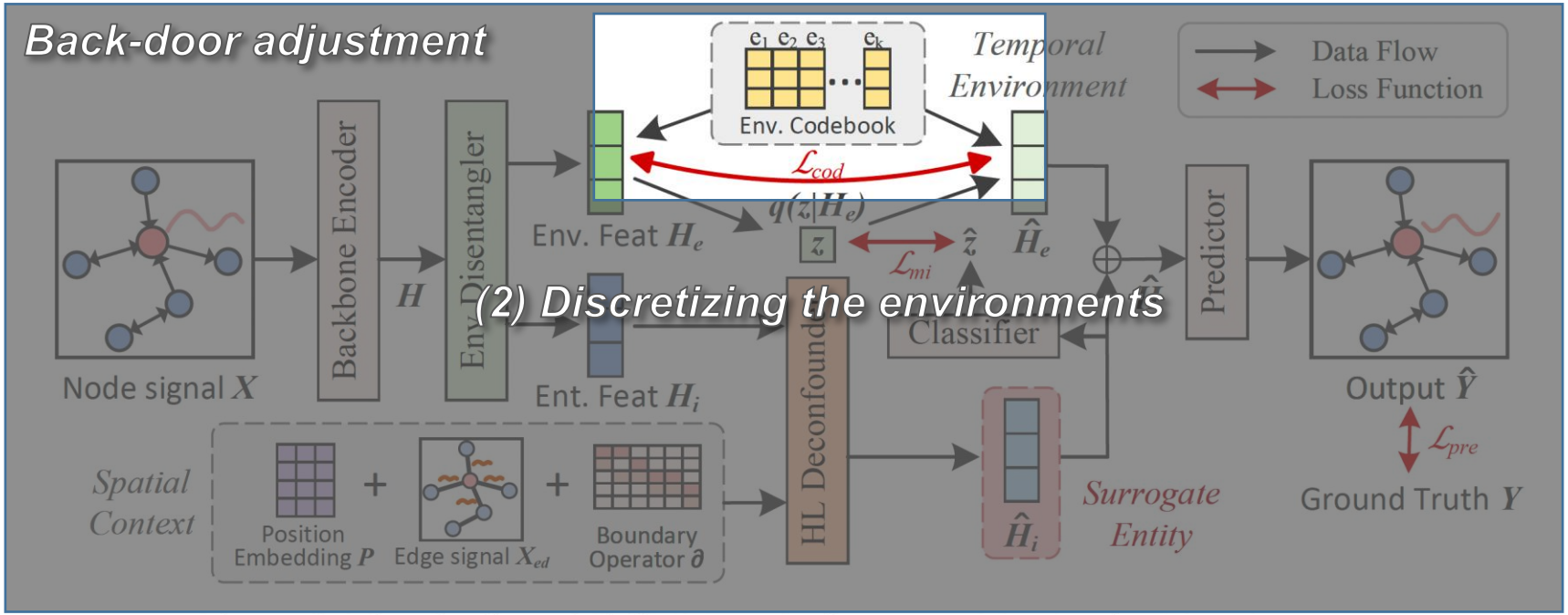


Figure 3: The pipeline of CaST. Env: Environment. Ent: Entity. Feat: Feature.

Model Instantiations

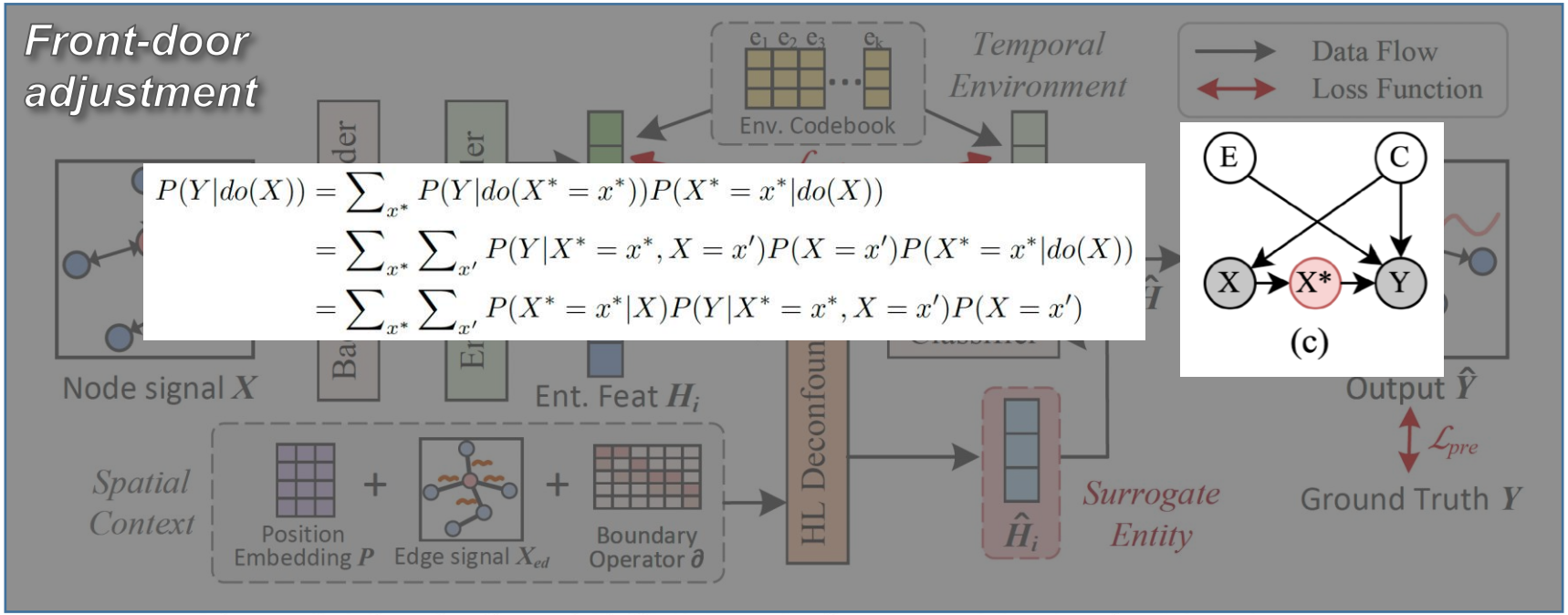


Figure 3: The pipeline of CaST. Env: Environment. Ent: Entity. Feat: Feature.

Model Instantiations

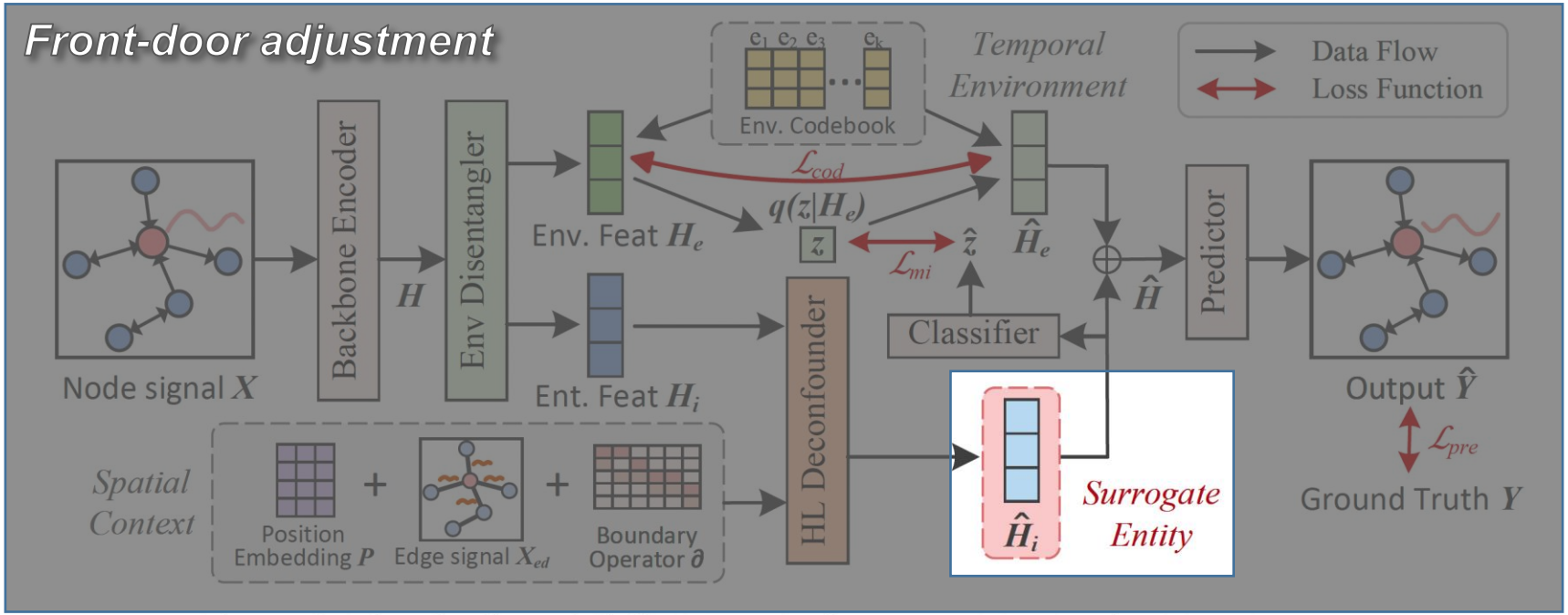


Figure 3: The pipeline of CaST. Env: Environment. Ent: Entity. Feat: Feature.

Model Instantiations

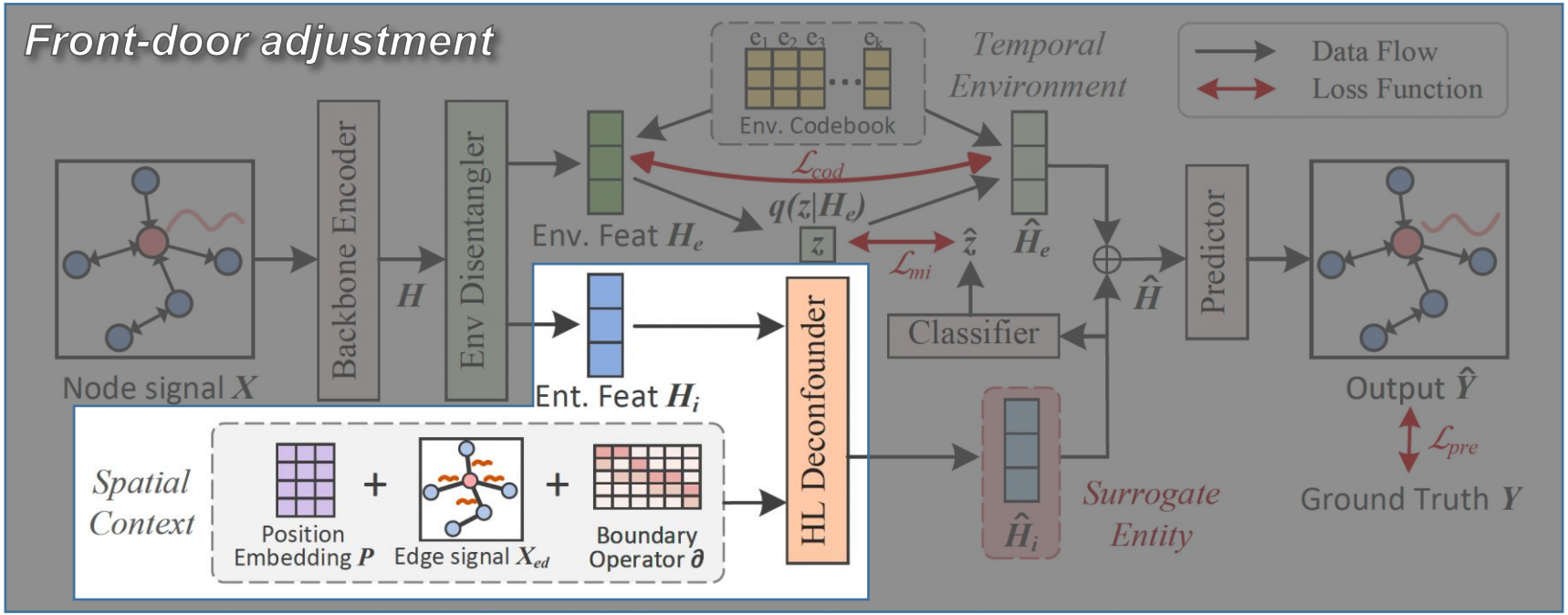


Figure 3: The pipeline of CaST. Env: Environment. Ent: Entity. Feat: Feature.

Model Instantiations

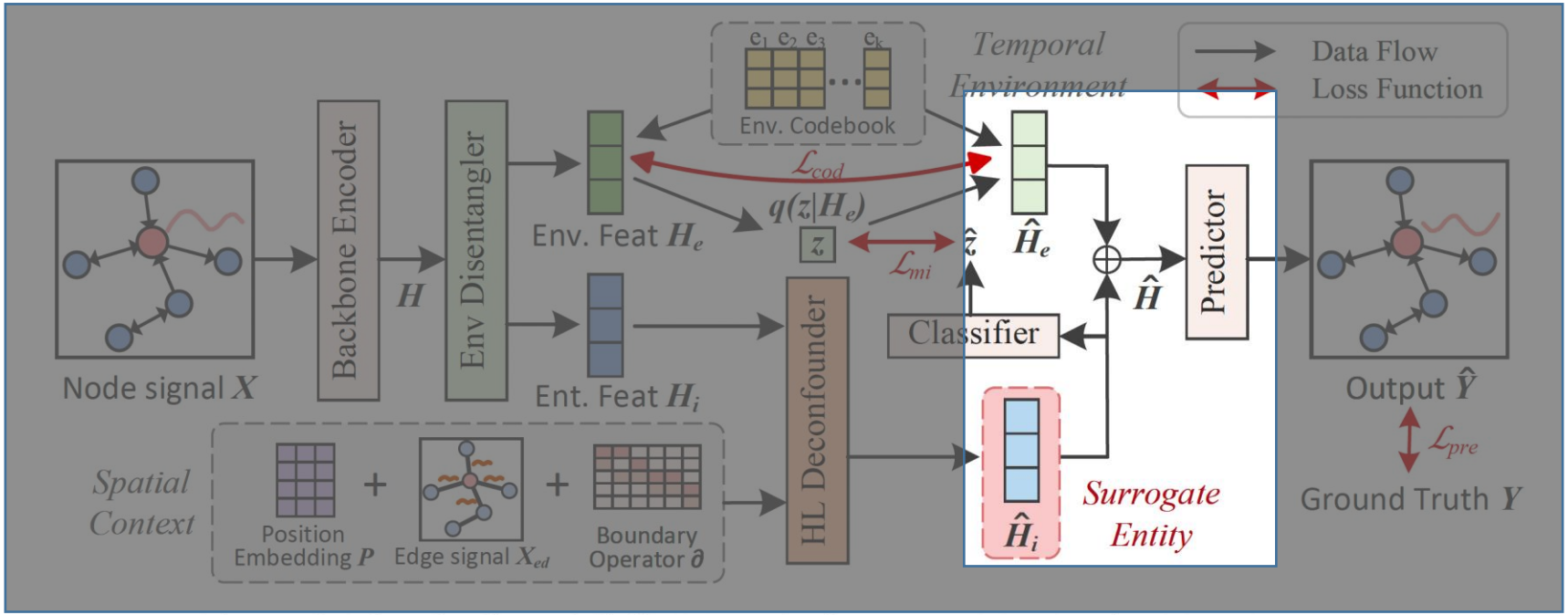


Figure 3: The pipeline of CaST. Env: Environment. Ent: Entity. Feat: Feature.

Model Instantiations

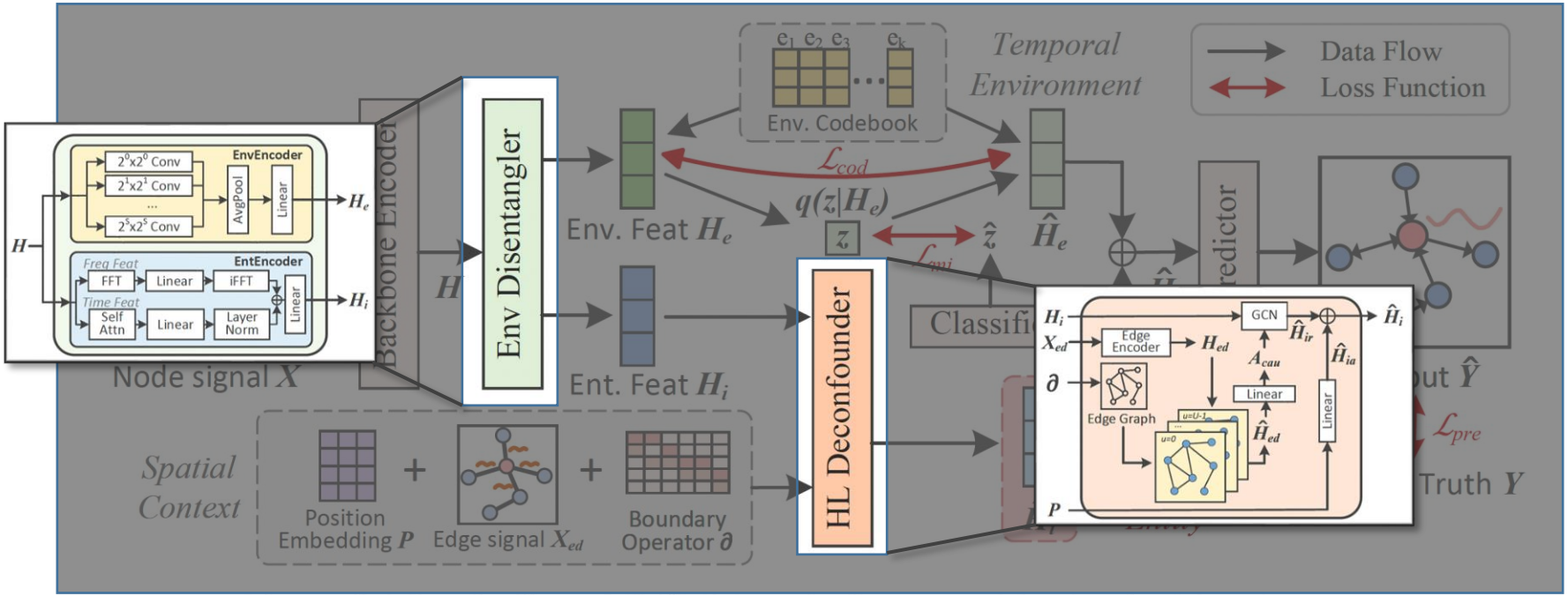


Figure 3: The pipeline of CaST. Env: Environment. Ent: Entity. Feat: Feature.

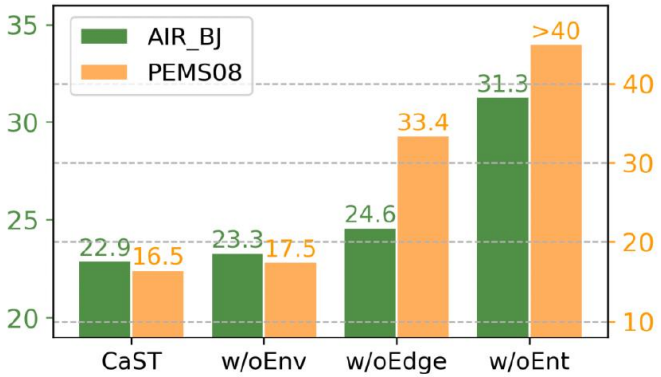
Experiments

- Datasets: PEMS08, AIR-BJ, AIR-GZ
- Experiment settings: predict over the next 24 steps given the past 24 steps
- Evaluation metrics: MAE, RMSE

Model	PEMS08 (24→24)		AIR-BJ (24→24)		AIR-GZ (24→24)	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA(2017)	58.83	81.96	32.12	43.95	19.56	25.77
VAR(1991)	37.04	53.08	29.79	42.04	14.97	20.61
DCRNN(2017)	22.10 ± 0.45	33.96 ± 0.59	23.72 ± 0.36	35.84 ± 0.56	12.99 ± 0.26	18.27 ± 0.41
STGCN(2018)	18.60 ± 0.08	28.44 ± 0.15	23.71 ± 0.21	36.30 ± 0.58	12.69 ± 0.04	17.66 ± 0.09
ASTGCN(2019)	20.36 ± 0.48	30.87 ± 0.55	23.78 ± 0.22	35.91 ± 0.11	12.91 ± 0.15	18.02 ± 0.27
MTGNN(2020)	18.13 ± 0.10	28.85 ± 0.12	24.35 ± 0.74	38.97 ± 1.81	12.43 ± 0.11	17.99 ± 0.18
AGCRN(2020)	17.06 ± 0.14	26.80 ± 0.15	23.43 ± 0.29	35.66 ± 0.57	12.74 ± 0.01	17.49 ± 0.01
GMSDR(2022)	18.34 ± 0.68	28.36 ± 1.01	25.92 ± 0.52	39.60 ± 0.44	13.47 ± 0.31	19.04 ± 0.46
STGNCDE(2022)	17.55 ± 0.30	27.28 ± 0.36	24.35 ± 0.31	35.91 ± 0.48	13.70 ± 0.10	19.15 ± 0.07
CaST (ours)	**16.44** ± 0.10	**26.61** ± 0.15	**22.90** ± 0.09	**34.84** ± 0.11	**12.36** ± 0.01	**17.25** ± 0.05

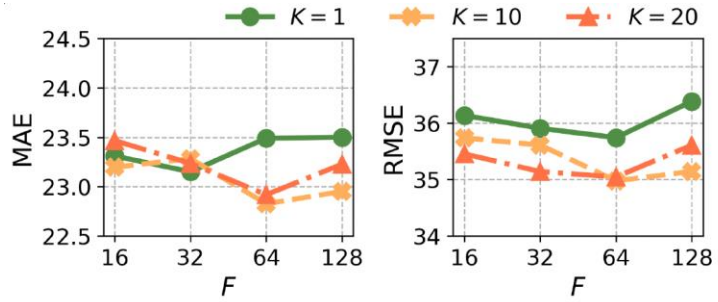
Ablation Study & Interpretation Analysis

- Effects of Core Components
  - w/o Env: excludes environment features for prediction.
  - w/o Ent: omits entity features for prediction.
  - w/o Edge: not utilize the causal score to guide the spatial message passing

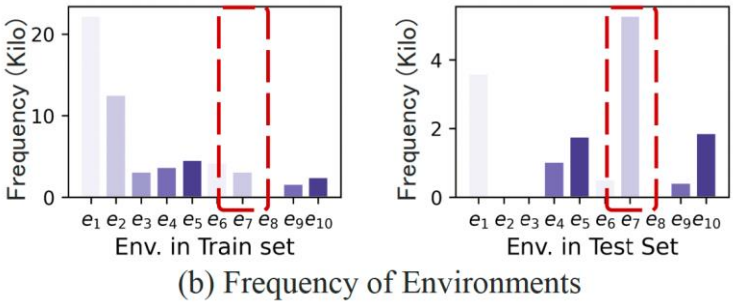


Ablation Study & Interpretation Analysis

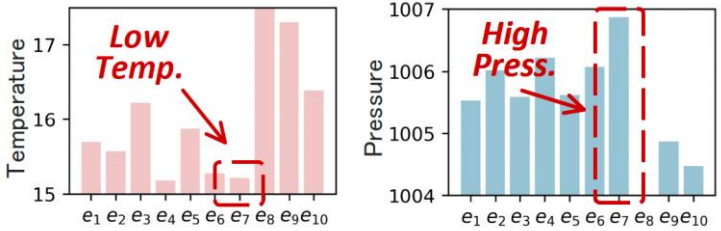
- Analysis on Environmental Codebook



- Interpretation of Temporal Environments



(b) Frequency of Environments



(c) External Factors of Environments

Conclusion

- Took a causal look at the STG forecasting problem
  - Utilizing back-door and front-door adjustments for resolving challenges
  - Introducing a novel Causal Spatio-Temporal neural network (CaST)
  - Verifying effectiveness, generalizability, and interpretability through extensive experiments on three datasets

Outline

- Generative models for Human Mobility Data
  - Adversarial training
  - Diffusion models
- Causal Learning for Human Mobility Data
- Continuous modeling for Human Mobility Data

Preliminary - Neural ODE

- Models such as ResNet and RNN build complicated transformations by

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$

- In this paper, they parameterize the continuous dynamics of hidden units using an ordinary differential equations (ODE) specified by a neural network

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$$

Integrating Neural Networks - ResNet analogy

- Euler method is the simplest one:

$$\mathbf{h}_N = \mathbf{h}_{N-1} + \Delta t g\left((N-1)\,\Delta t, \mathbf{h}_{N-1}\right)$$

- Some people find this equation similar to ResNet skip connection

$$\mathbf{h}_{l+1} = \mathbf{h}_l + \mathrm{NNetwork}(\mathbf{h}_l)$$

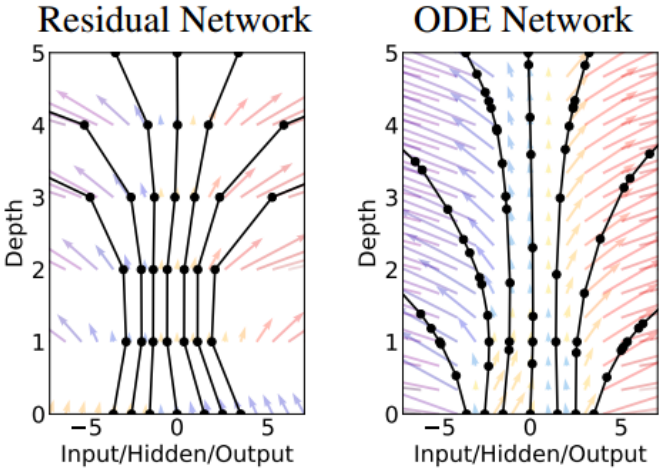


Figure 1: *Left:* A Residual network defines a discrete sequence of finite transformations. *Right:* A ODE network defines a vector field, which continuously transforms the state. *Both:* Circles represent evaluation locations.

Backpropagation: Main Technical Difficulty

- Differentiating through the operations of the forward pass is straightforward, but incurs
  - high memory costs
  - additional numerical error.

Adjoint method!

Algorithm 1 Reverse-mode derivative of an ODE initial value problem

Input: dynamics parameters θ, start time t_0, stop time t_1, final state $\mathbf{z}(t_1)$, loss gradient $\partial L / \partial \mathbf{z}(t_1)$

$s_0 = [\mathbf{z}(t_1), \frac{\partial L}{\partial \mathbf{z}(t_1)}, \mathbf{0}_{|\theta|}]$ ▷ Define initial augmented state

def aug_dynamics($[\mathbf{z}(t), \mathbf{a}(t), \cdot], t, \theta$): ▷ Define dynamics on augmented state

  return $[f(\mathbf{z}(t), t, \theta), -\mathbf{a}(t)^\mathsf{T} \frac{\partial f}{\partial \mathbf{z}}, -\mathbf{a}(t)^\mathsf{T} \frac{\partial f}{\partial \theta}]$ ▷ Compute vector-Jacobian products

$[\mathbf{z}(t_0), \frac{\partial L}{\partial \mathbf{z}(t_0)}, \frac{\partial L}{\partial \theta}] = \text{ODESolve}(s_0, \text{aug\_dynamics}, t_1, t_0, \theta)$ ▷ Solve reverse-time ODE

return $\frac{\partial L}{\partial \mathbf{z}(t_0)}, \frac{\partial L}{\partial \theta}$ ▷ Return gradients

Advantages of Neural ODE

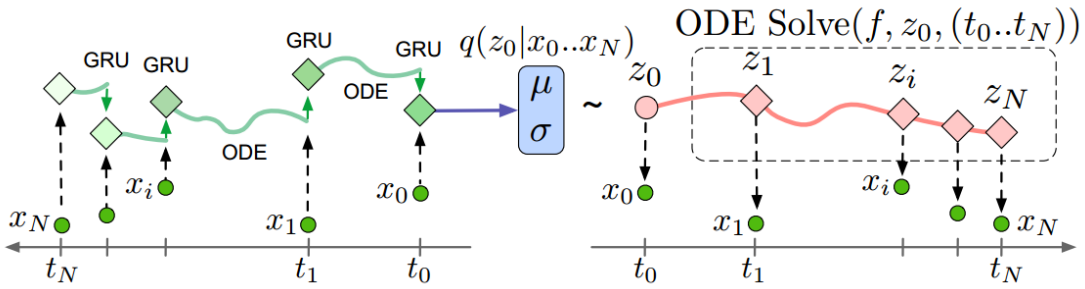
- Memory efficiency
- Adaptive computation
- Parameter efficiency
- Scalable and invertible normalizing flows
- Continuous time-series models

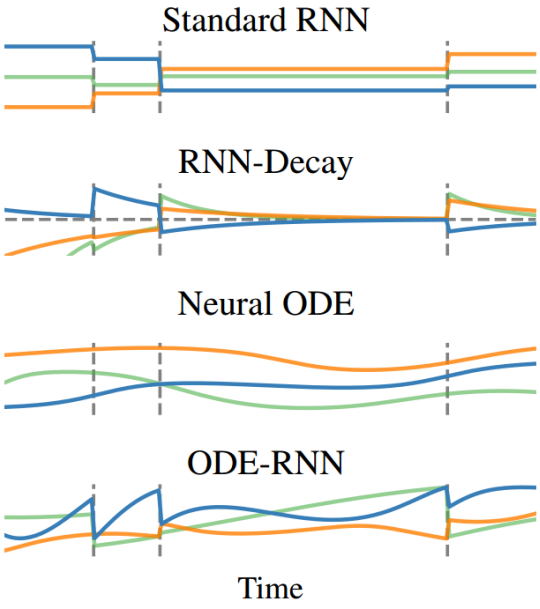
Time Series Models with Neural ODEs

- Latent ODEs for irregularly-sampled time series (NeurIPS 2019)

Algorithm 1 The ODE-RNN. The only difference, highlighted in blue, from standard RNNs is that the pre-activations h' evolve according to an ODE between observations, instead of being fixed.

Input: Data points and their timestamps $\{(x_i, t_i)\}_{i=1..N}$
$h_0 = \mathbf{0}$
for i in 1, 2, . . . , N **do**
  $h'_i = \text{ODESolve}(f_\theta, h_{i-1}, (t_{i-1}, t_i))$ ▷ Solve ODE to get state at t_i
  $h_i = \text{RNNCell}(h'_i, x_i)$ ▷ Update hidden state given current observation x_i
end for
$o_i = \text{OutputNN}(h_i)$ for all $i = 1..N$
Return: $\{o_i\}_{i=1..N}; h_N$





Neural CDE

- Nerual Controlled Differential Equations (NCDEs) are considered as a continuous analogue to RNNs
- NCDEs show the state-of-the-art accuracy in many time-series tasks

Let $\tau, T \in \mathbb{R}$ with $\tau < T$, and let $v, w \in \mathbb{N}$. Let $X \colon [\tau, T] \to \mathbb{R}^v$ be a continuous function of bounded variation; for example this is implied by X being Lipschitz. Let $\zeta \in \mathbb{R}^w$. Let $f \colon \mathbb{R}^w \to \mathbb{R}^{w \times v}$ be continuous.

Then we may define a continuous path $z \colon [\tau, T] \to \mathbb{R}^w$ by $z_\tau = \zeta$ and

$$z_t = z_\tau + \int_\tau^t f(z_s)\mathrm{d}X_s \quad \text{for } t \in (\tau, T], \tag{2}$$

where the integral is a Riemann–Stieltjes integral. As $f(z_s) \in \mathbb{R}^{w \times v}$ and $X_s \in \mathbb{R}^v$, the notation "$f(z_s)\mathrm{d}X_s$" refers to matrix-vector multiplication. The subscript notation refers to function evaluation, for example as is common in stochastic calculus.

Evaluating NCDE $$z_t = z_{t_0} + \int_{t_0}^t f_\theta(z_s)\mathrm{d}X_s = z_{t_0} + \int_{t_0}^t f_\theta(z_s)\frac{\mathrm{d}X}{\mathrm{d}s}(s)\mathrm{d}s = z_{t_0} + \int_{t_0}^t g_{\theta,X}(z_s, s)\mathrm{d}s.$$

Neural CDE

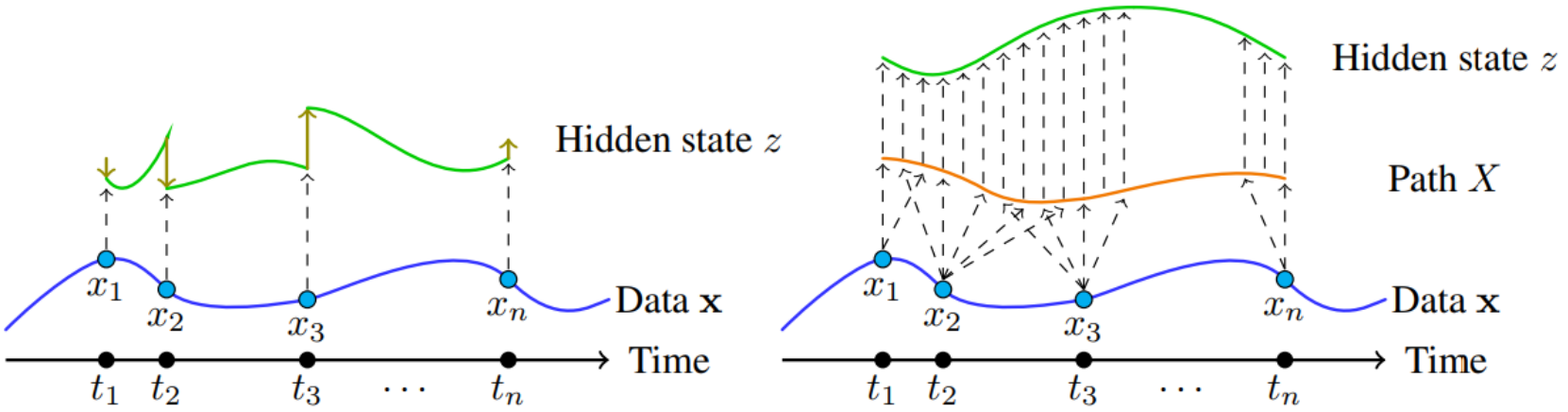
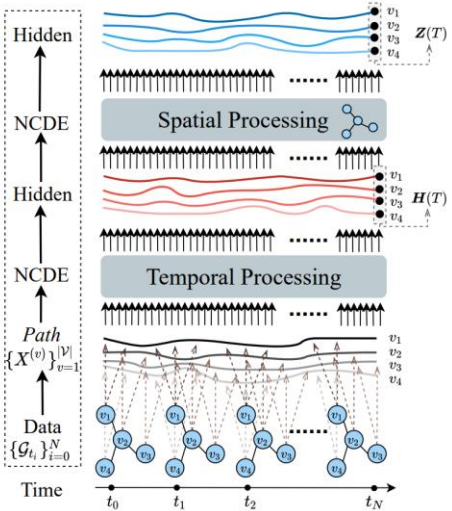


Figure 1: Some data process is observed at times $t_1, \ldots, t_n$ to give observations $x_1, \ldots, x_n$. It is otherwise unobserved. **Left:** Previous work has typically modified hidden state at each observation, and perhaps continuously evolved the hidden state between observations. **Right:** In contrast, the hidden state of the Neural CDE model has continuous dependence on the observed data.

Graph Neural Controlled Differential Equations for Traffic Forecasting

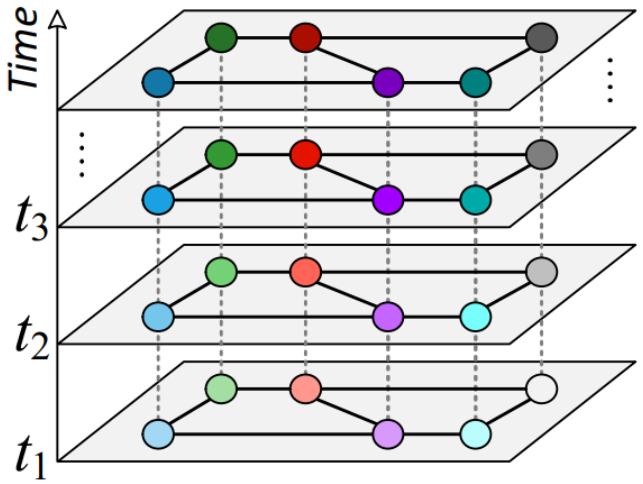
AAAI 2022



Model	PeMSD3			PeMSD4			PeMSD7			PeMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	31.58	52.39	33.78%	38.03	59.24	27.88%	45.12	65.64	24.51%	34.86	59.24	27.88%
ARIMA	35.41	47.59	33.78%	33.73	48.80	24.18%	38.17	59.27	19.46%	31.09	44.32	22.73%
VAR	23.65	38.26	24.51%	24.54	38.61	17.24%	50.22	75.63	32.22%	19.19	29.81	13.10%
FC-LSTM	21.33	35.11	23.33%	26.77	40.65	18.23%	29.98	45.94	13.20%	23.09	35.17	14.99%
TCN	19.32	33.55	19.93%	23.22	37.26	15.59%	32.72	42.23	14.26%	22.72	35.79	14.03%
TCN(w/o causal)	18.87	32.24	18.63%	22.81	36.87	14.31%	30.53	41.02	13.88%	21.42	34.03	13.09%
GRU-ED	19.12	32.85	19.31%	23.68	39.27	16.44%	27.66	43.49	12.20%	22.00	36.22	13.33%
DSANet	21.29	34.55	23.21%	22.79	35.77	16.03%	31.36	49.11	14.43%	17.14	26.96	11.32%
STGCN	17.55	30.42	17.34%	21.16	34.89	13.83%	25.33	39.34	11.21%	17.50	27.09	11.29%
DCRNN	17.99	30.31	18.34%	21.22	33.44	14.17%	25.22	38.61	11.82%	16.82	26.36	10.92%
GraphWaveNet	19.12	32.77	18.89%	24.89	39.66	17.29%	26.39	41.50	11.97%	18.28	30.05	12.15%
ASTGCN(r)	17.34	29.56	17.21%	22.93	35.22	16.56%	24.01	37.87	10.73%	18.25	28.06	11.64%
MSTGCN	19.54	31.93	23.86%	23.96	37.21	14.33%	29.00	43.73	14.30%	19.00	29.15	12.38%
STG2Seq	19.03	29.83	21.55%	25.20	38.48	18.77%	32.77	47.16	20.16%	20.17	30.71	17.32%
LSGCN	17.94	29.85	16.98%	21.53	33.86	13.18%	27.31	41.46	11.98%	17.73	26.76	11.20%
STSGCN	17.48	29.21	16.78%	21.19	33.65	13.90%	24.26	39.03	10.21%	17.13	26.80	10.96%
AGCRN	15.98	28.25	15.23%	19.83	32.26	12.97%	22.37	36.55	9.12%	15.95	25.22	10.09%
STFGNN	16.77	28.34	16.30%	20.48	32.51	16.77%	23.46	36.60	9.21%	16.94	26.25	10.60%
STGODE	16.50	27.84	16.69%	20.84	32.82	13.77%	22.59	37.54	10.14%	16.81	25.97	10.62%
Z-GCNETs	16.64	28.15	16.39%	19.50	31.61	12.78%	21.77	35.17	9.25%	15.76	25.11	10.01%
STG-NCDE	**15.57**	**27.09**	**15.06%**	**19.21**	**31.09**	**12.76%**	**20.53**	**33.84**	**8.80%**	**15.45**	**24.81**	**9.92%**
Only temporal	20.44	32.82	20.03%	26.31	40.97	17.95%	28.77	44.39	12.60%	20.83	32.55	13.01%
Only spatial	15.92	27.17	15.14%	19.86	31.92	13.35%	21.72	34.73	9.24%	17.58	27,76	11.27%

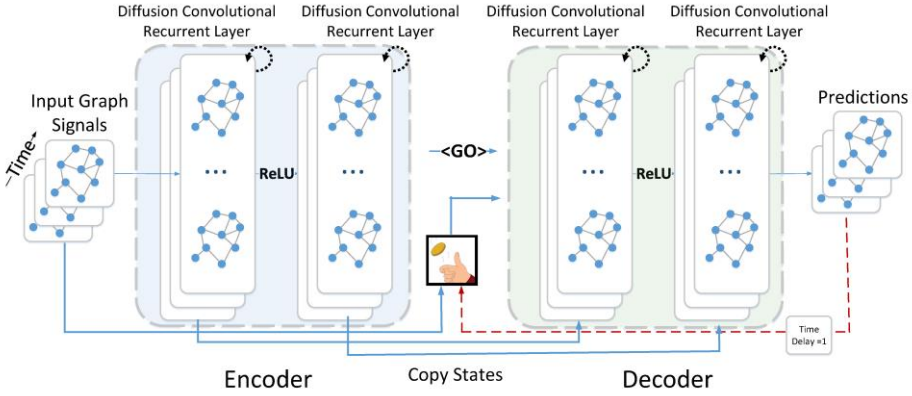
Definition of Spatio-Temporal Graphs (STG)

- 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
  - Traffic speed/flow over road networks
  - Crowd flow in irregular urban regions

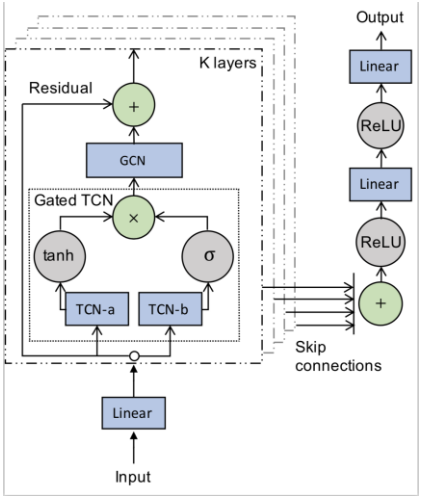


Spatio-Temporal Graph Neural Networks

- 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

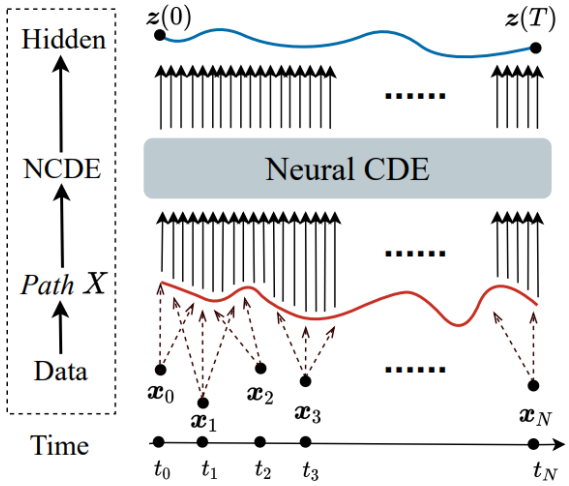


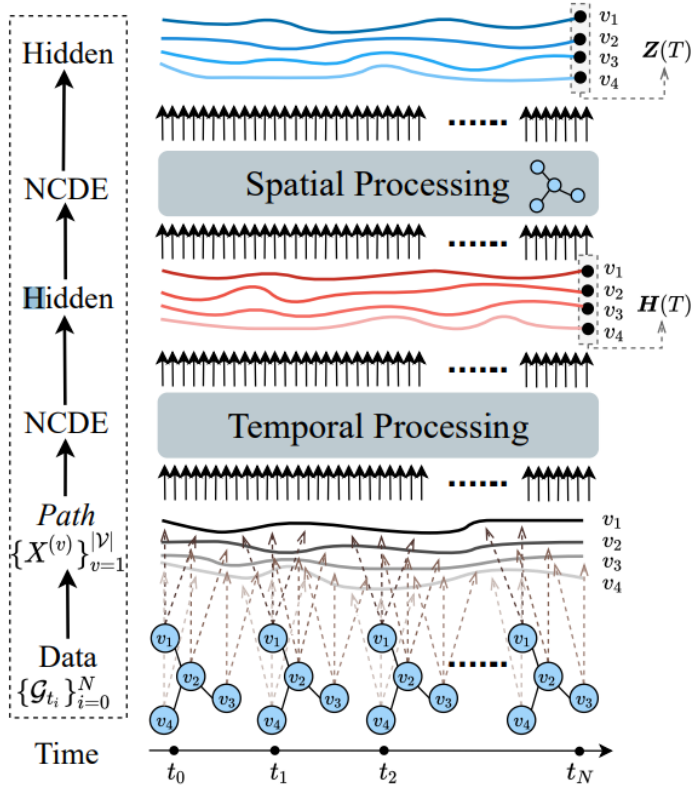
DCRNN



Graph WaveNet

Graph NCDE for Traffic Forecasting





Comparison to Existing Apporaches

Model	Temporal Dependency	Spatial Dependency
STGCN [11]	GLU	GCN
DCRNN [8]	LSTM	GCN (Pre-defined Graph)
GraphWaveNet [10]	TCN	GCN (Adaptive Graph)
ASTGCN [5]	1D CNN+Attention	Attention+GCN(Pre-defined Graph)
STSGCN [9]	GCN (Sliding Window)	GCN (Multiple Graph)
AGCRN [1]	GRU	GCN (Adaptive Graph)
STFGNN [7]	1D CNN	GCN (Multiple Graph)
STGODE [4]	TCN	ODE+GCN (Dynamic Graph)
STG-NDCE	NCDE	NCDE + GCN (Adaptive Graph)

Table: Classification of spatial-temporal modeling methods

Temporal Modeling

- For a node v, its temporal processing can be written as follows:

$$\boldsymbol{h}^{(v)}(T) = \boldsymbol{h}^{(v)}(0) + \int_0^T f(\boldsymbol{h}^{(v)}(t); \boldsymbol{\theta}_f) \frac{dX^{(v)}(t)}{dt} dt,$$

Matrix form

$$\boldsymbol{H}(T) = \boldsymbol{H}(0) + \int_0^T f(\boldsymbol{H}(t); \boldsymbol{\theta}_f) \frac{d\boldsymbol{X}(t)}{dt} dt,$$

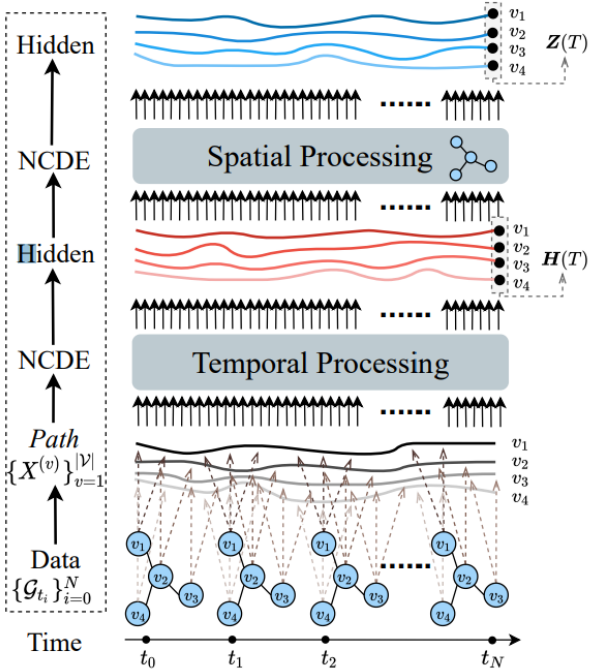
CDE functions

$$f(\boldsymbol{H}(t); \boldsymbol{\theta}_f) = \psi(\mathrm{FC}_{|\mathcal{V}|\times \dim(\boldsymbol{h}^{(v)}) \to |\mathcal{V}|\times \dim(\boldsymbol{h}^{(v)})}(\boldsymbol{A}_K)),$$

$$\vdots$$

$$\boldsymbol{A}_1 = \sigma(\mathrm{FC}_{|\mathcal{V}|\times \dim(\boldsymbol{h}^{(v)}) \to |\mathcal{V}|\times \dim(\boldsymbol{h}^{(v)})}(\boldsymbol{A}_0)),$$

$$\boldsymbol{A}_0 = \sigma(\mathrm{FC}_{|\mathcal{V}|\times \dim(\boldsymbol{h}^{(v)}) \to |\mathcal{V}|\times \dim(\boldsymbol{h}^{(v)})}(\boldsymbol{H}(t))),$$



Spatial Processing

- Similarly, the second NCDE starts for its spatial processing as follows:

Matrix form

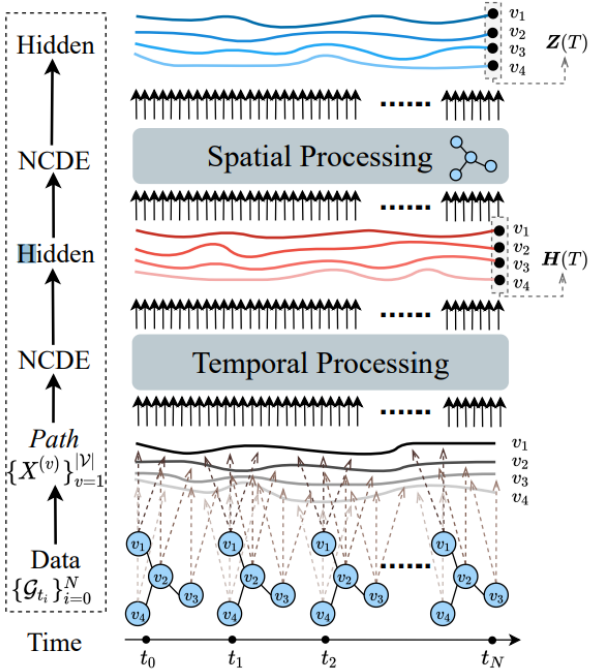
$$\boldsymbol{Z}(T) = \boldsymbol{Z}(0) + \int_0^T g(\boldsymbol{Z}(t); \boldsymbol{\theta}_g) \frac{d\boldsymbol{H}(t)}{dt} dt,$$

CDE functions

$$g(\boldsymbol{Z}(t); \boldsymbol{\theta}_g) = \psi(\mathrm{FC}_{|\mathcal{V}| \times \dim(\boldsymbol{z}^{(v)}) \to |\mathcal{V}| \times \dim(\boldsymbol{z}^{(v)})}(\boldsymbol{B}_1)), \tag{9}$$

$$\boldsymbol{B}_1 = (\boldsymbol{I} + \phi(\sigma(\boldsymbol{E} \cdot \boldsymbol{E}^\intercal)))\boldsymbol{B}_0 \boldsymbol{W}_{spatial}, \tag{10}$$

$$\boldsymbol{B}_0 = \sigma(\mathrm{FC}_{|\mathcal{V}| \times \dim(\boldsymbol{z}^{(v)}) \to |\mathcal{V}| \times \dim(\boldsymbol{z}^{(v)})}(\boldsymbol{Z}(t))), \tag{11}$$



Combination of Spatial and Temporal Processing

$$\boldsymbol{H}(T) = \boldsymbol{H}(0) + \int_0^T f(\boldsymbol{H}(t); \boldsymbol{\theta}_f) \frac{d\boldsymbol{X}(t)}{dt} dt, \qquad \boldsymbol{Z}(T) = \boldsymbol{Z}(0) + \int_0^T g(\boldsymbol{Z}(t); \boldsymbol{\theta}_g) \frac{d\boldsymbol{H}(t)}{dt} dt,$$

$$\boldsymbol{Z}(T) = \boldsymbol{Z}(0) + \int_0^T g(\boldsymbol{Z}(t); \boldsymbol{\theta}_g) f(\boldsymbol{H}(t); \boldsymbol{\theta}_f) \frac{d\boldsymbol{X}(t)}{dt} dt$$

Experimental Results

Model	PeMSD3			PeMSD4			PeMSD7			PeMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	31.58	52.39	33.78%	38.03	59.24	27.88%	45.12	65.64	24.51%	34.86	59.24	27.88%
ARIMA	35.41	47.59	33.78%	33.73	48.80	24.18%	38.17	59.27	19.46%	31.09	44.32	22.73%
VAR	23.65	38.26	24.51%	24.54	38.61	17.24%	50.22	75.63	32.22%	19.19	29.81	13.10%
FC-LSTM	21.33	35.11	23.33%	26.77	40.65	18.23%	29.98	45.94	13.20%	23.09	35.17	14.99%
TCN	19.32	33.55	19.93%	23.22	37.26	15.59%	32.72	42.23	14.26%	22.72	35.79	14.03%
TCN(w/o causal)	18.87	32.24	18.63%	22.81	36.87	14.31%	30.53	41.02	13.88%	21.42	34.03	13.09%
GRU-ED	19.12	32.85	19.31%	23.68	39.27	16.44%	27.66	43.49	12.20%	22.00	36.22	13.33%
DSANet	21.29	34.55	23.21%	22.79	35.77	16.03%	31.36	49.11	14.43%	17.14	26.96	11.32%
STGCN	17.55	30.42	17.34%	21.16	34.89	13.83%	25.33	39.34	11.21%	17.50	27.09	11.29%
DCRNN	17.99	30.31	18.34%	21.22	33.44	14.17%	25.22	38.61	11.82%	16.82	26.36	10.92%
GraphWaveNet	19.12	32.77	18.89%	24.89	39.66	17.29%	26.39	41.50	11.97%	18.28	30.05	12.15%
ASTGCN(r)	17.34	29.56	17.21%	22.93	35.22	16.56%	24.01	37.87	10.73%	18.25	28.06	11.64%
MSTGCN	19.54	31.93	23.86%	23.96	37.21	14.33%	29.00	43.73	14.30%	19.00	29.15	12.38%
STG2Seq	19.03	29.83	21.55%	25.20	38.48	18.77%	32.77	47.16	20.16%	20.17	30.71	17.32%
LSGCN	17.94	29.85	16.98%	21.53	33.86	13.18%	27.31	41.46	11.98%	17.73	26.76	11.20%
STSGCN	17.48	29.21	16.78%	21.19	33.65	13.90%	24.26	39.03	10.21%	17.13	26.80	10.96%
AGCRN	15.98	28.25	15.23%	19.83	32.26	12.97%	22.37	36.55	9.12%	15.95	25.22	10.09%
STFGNN	16.77	28.34	16.30%	20.48	32.51	16.77%	23.46	36.60	9.21%	16.94	26.25	10.60%
STGODE	16.50	27.84	16.69%	20.84	32.82	13.77%	22.59	37.54	10.14%	16.81	25.97	10.62%
Z-GCNETs	16.64	28.15	16.39%	19.50	31.61	12.78%	21.77	35.17	9.25%	15.76	25.11	10.01%
STG-NCDE	**15.57**	**27.09**	**15.06%**	**19.21**	**31.09**	**12.76%**	**20.53**	**33.84**	**8.80%**	**15.45**	**24.81**	**9.92%**
Only temporal	20.44	32.82	20.03%	26.31	40.97	17.95%	28.77	44.39	12.60%	20.83	32.55	13.01%
Only spatial	15.92	27.17	15.14%	19.86	31.92	13.35%	21.72	34.73	9.24%	17.58	27,76	11.27%

Other Advanced Topics

- Physics-Informed Neural Networks (PINN)
- Reinforcement Learning (RL)

1	Dynamic Bike Reposition: A Spatio-Temporal Reinforcement Learning Approach	.com/pdf/KDD2018dyna	KDD	2018	HKUST	Yixuan Wang
2	STDEN: Towards Physics-Guided Neural Networks for Traffic Flow Prediction	s://arxiv.org/pdf/2209.00	AAAI	2022	State Key Laboratory of Softw	Jingtao HE
3	Trace and Pace: Controllable Pedestrian Animation via Guided Trajectory Diffusion	Trace and Pace: Contro	CVPR	2023	NVIDIA	Ruiguo Zhong

Note on Project Presentation

- Each group has 20 mins for presentation (**15-min Talk** + **5-min QA**)
- How to determine the order? Random draw!

11	25 Nov	L11: Large Language Models for Human Mobility Analytics	✓	
12	2 Dec	L12: Summary and Future Trends	✓	* Project presentation * Submit Systems Report & Codes (by **9 Dec @ 23:59**)

A Slide Template for Project Presentation

- Abstract
- **Introduction**
  - Background (e.g., target, motivation)
  - Challenges & Research Gaps
  - Your contributions
- Related works & Problem statement
- **Methodology**
  - Framework
  - Model details
- Experiments
  - Datasets & settings
  - Results & discussion
  - Case studies & Visualization
- Conclusion
- Future works

NUS
National University
of Singapore

AirFormer: Predicting Nationwide Air Quality in China with Transformers

Yuxuan Liang, Yutong Xia, Songyu Ke, Yiwei Wang, Qingsong Wen, Junbo Zhang, Yu Zheng, Roger Zimmermann

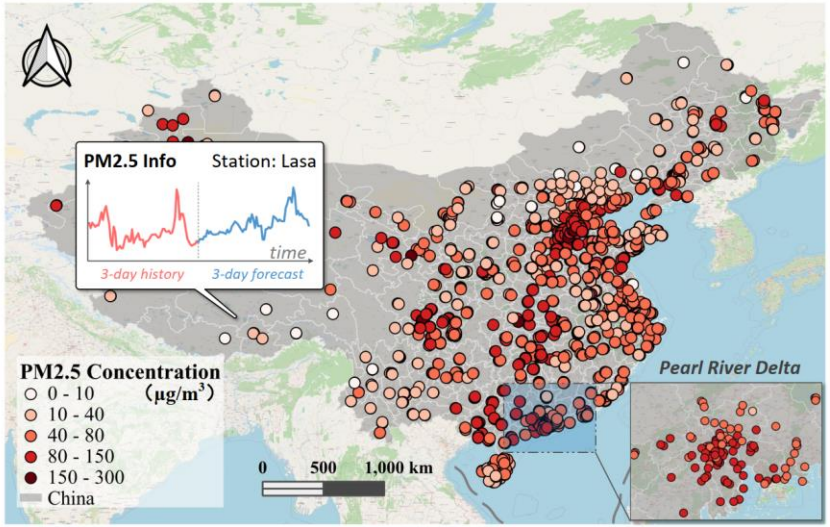
Background

- According to World Health Organization (WHO), air pollution is one of the leading causes of death in the world today, accounting for 7 million fatalities per year
- Over 90% of people breathe air that contains more contaminants than the WHO's recommended levels, with those in emerging countries suffering the most, especially in China with 1.4 billion people

Nationwide Air Quality Prediction in China

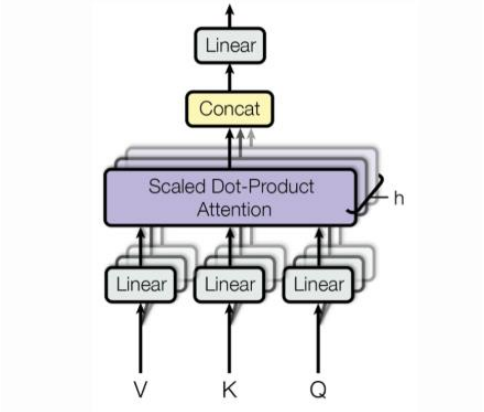
- We present the first attempt to *collectively* predict air quality in the Chinese mainland with an unprecedented fine spatial granularity using **transformers**, covering thousands of stations.

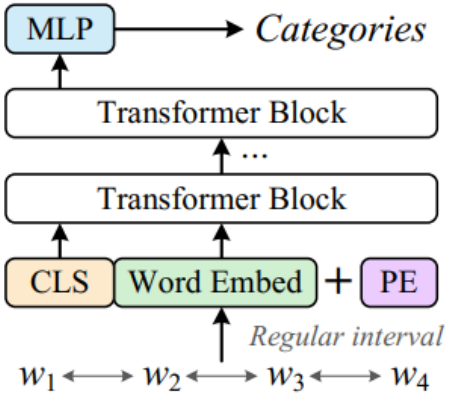
- Station distribution
- Advantages
  - Providing more useful information to the public with high social impacts
  - Containing more data samples that benefit model training



Transformer Architecture

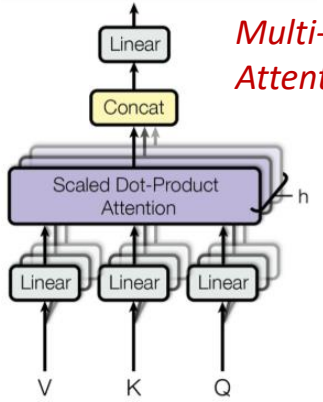
- Recently, Transformers has emerged as the dominant model in some research fields, especially in NLP
  - Key operation: Multi-head Self-attention mechanism (MSA)
  - **Parallel training and inference**
  - Achieving new state-of-the-art results across various tasks





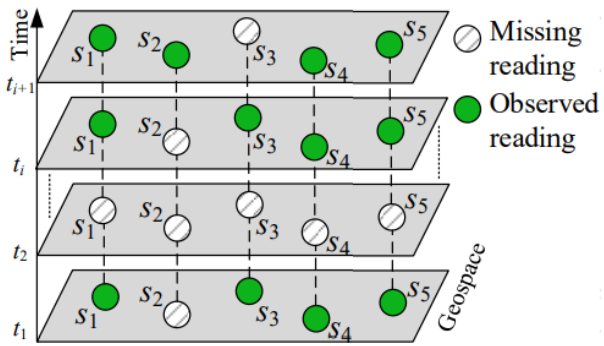
Challenges

- Inefficiency: Transformers suffer notoriously high computation costs when scaling up to predictions over thousands of stations
- Uncertainty: the air quality readings are intrinsically uncertain due to
  - Inaccurate or missing observations
  - Unpredictable factors, e.g., vehicle exhaust, policy, and industrial emission



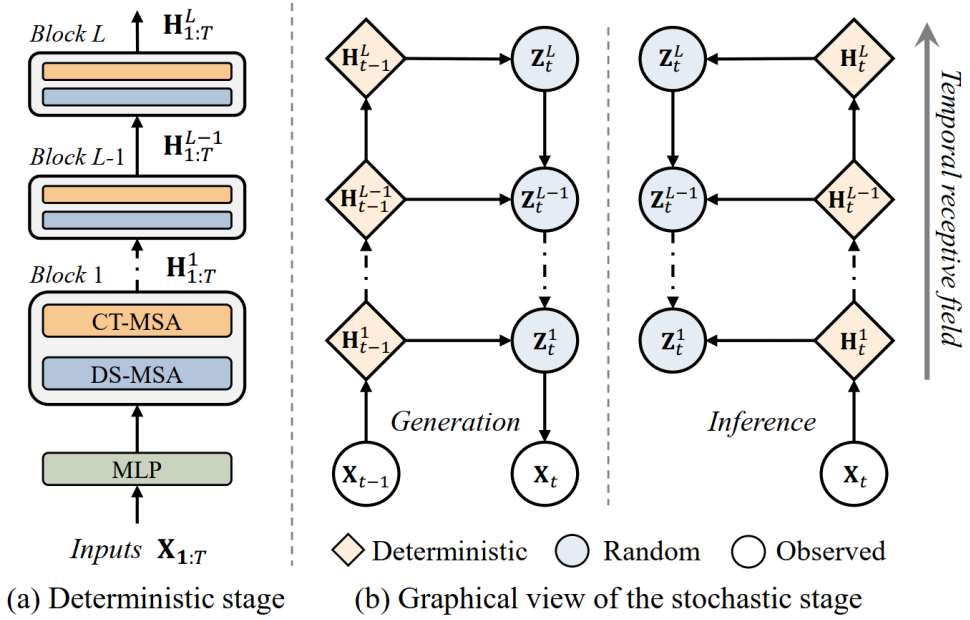
Multi-head Self-Attention (MSA)

Quadratic costs w.r.t # tokens



Framework

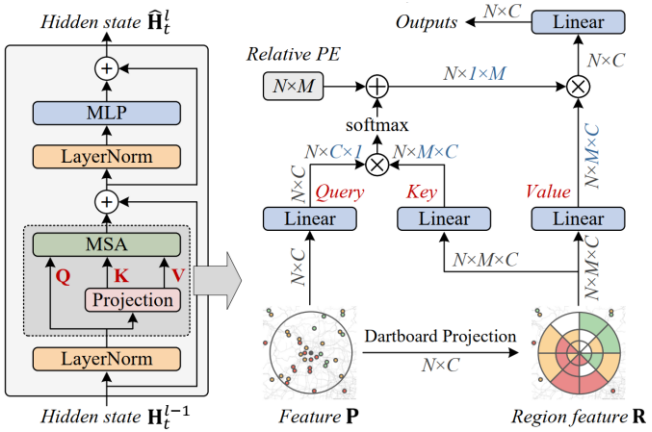
- To tackle these issues, we present **AirFormer**, a novel transformer architecture for nationwide air quality prediction in China
- Two-stage framework
  - Bottom-up deterministic stage
  - Top-down stochastic stage



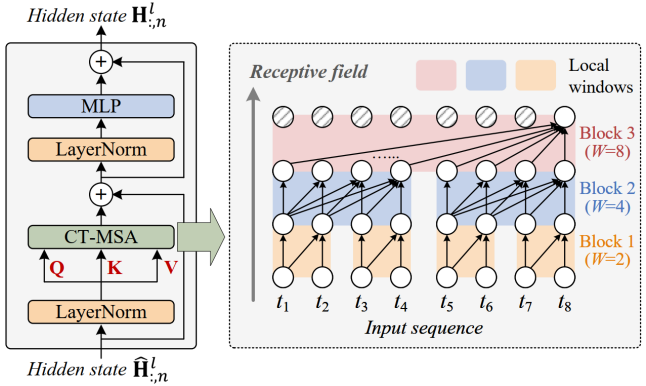
(a) Deterministic stage (b) Graphical view of the stochastic stage

Bottom-up Deterministic Stage

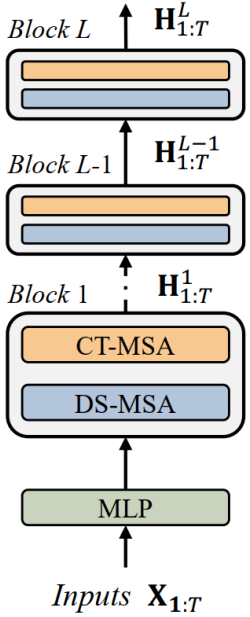
- In the first stage, we propose two new types of MSA to efficiently capture the spatial and temporal dependencies, respectively
  - Dartboard Spatial-MSA (DS-MSA)
  - Causal Temporal-MSA (CT-MSA)



DS-MSA

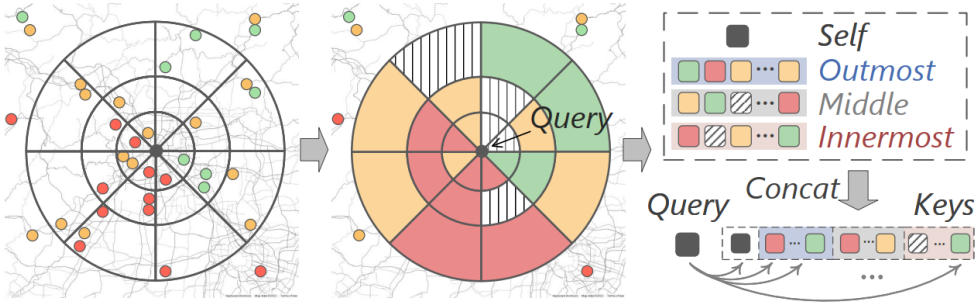


CT-MSA



Dartboard Spatial-MSA

- Considering the spatial correlations among nearby locations are often stronger than those far away, we devise DS-MSA to efficiently capture spatial relations.
  - Insight: as its name suggests, each location attends to its close surroundings at a fine granularity and faraway stations at a coarse granularity
  - Pipeline: see the following figure
  - Result: DS-MSA only takes **linear complexity** w.r.t the number of stations.



(a) Spatial partition (b) Spatial projection (c) Spatial attention

$\mathcal{O}(N^2C)$ MSA

Reduced

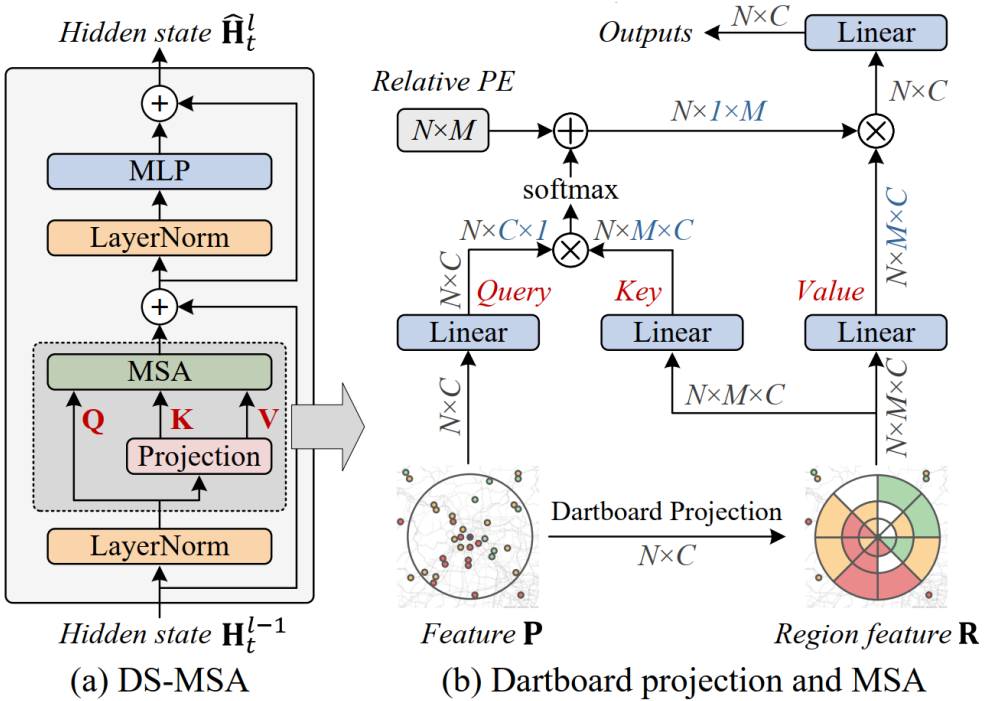
N: #stations

M: #regions

$\mathcal{O}(NMC)$ our DS-MSA

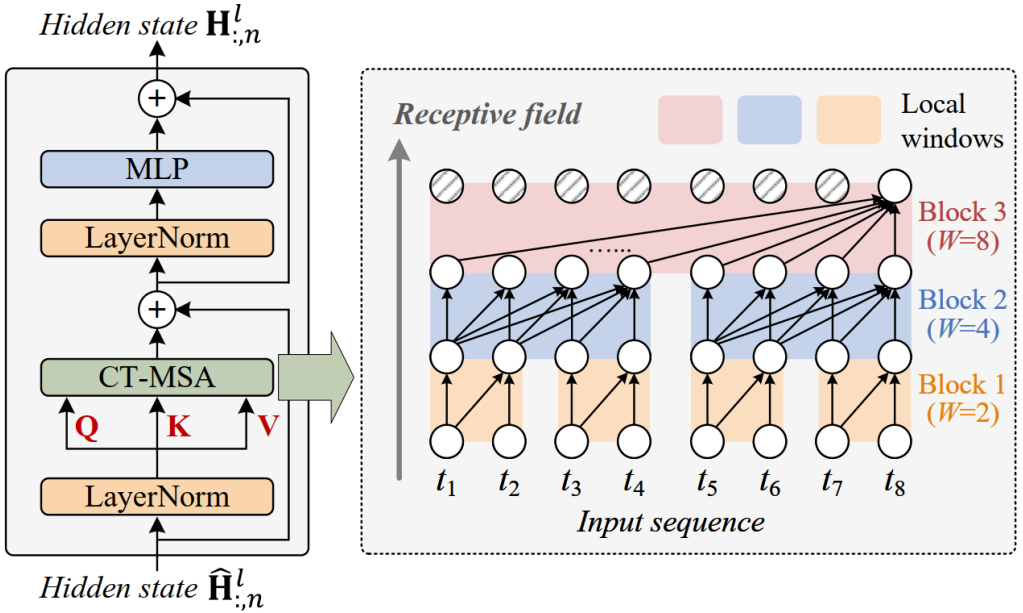
Dartboard Spatial-MSA

- Implementation



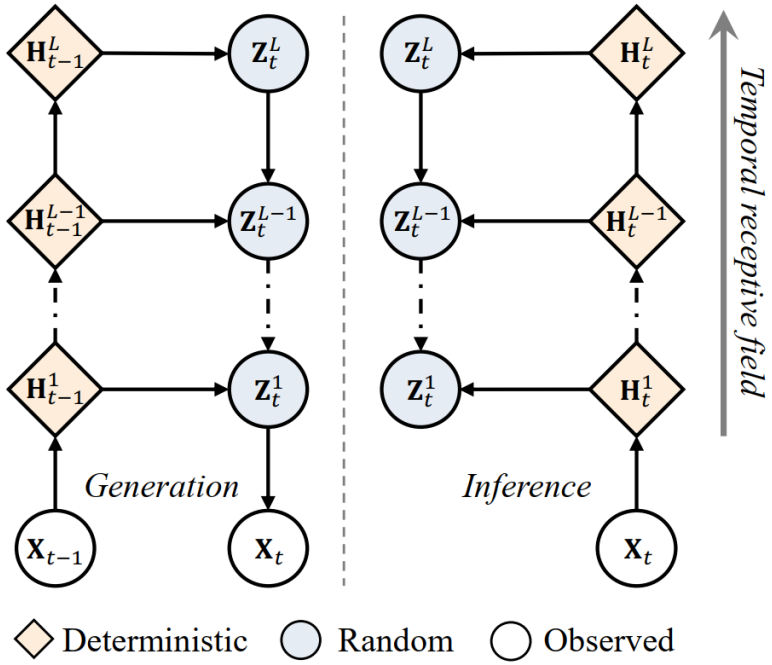
Causal Temporal-MSA

- In addition to spatial relations, the air quality of a location depends on its history
- Leveraging the domain knowledge of time series, CT-MSA has two major modifications beyond standard MSA
  - Local window
  - Temporal causality



Top-Down Stochastic Stage

- Once the deterministic representations are obtained, we produce latent variables $\boldsymbol{Z}$ at each level
- To maintain the parallelism of transformers
  - We do not establish explicit dependencies between various time steps.
  - We implicitly build temporal dependencies by conditioning a latent variable $\boldsymbol{Z}_t^{l-1}$ on its higher-level variable $\boldsymbol{Z}_t^l$



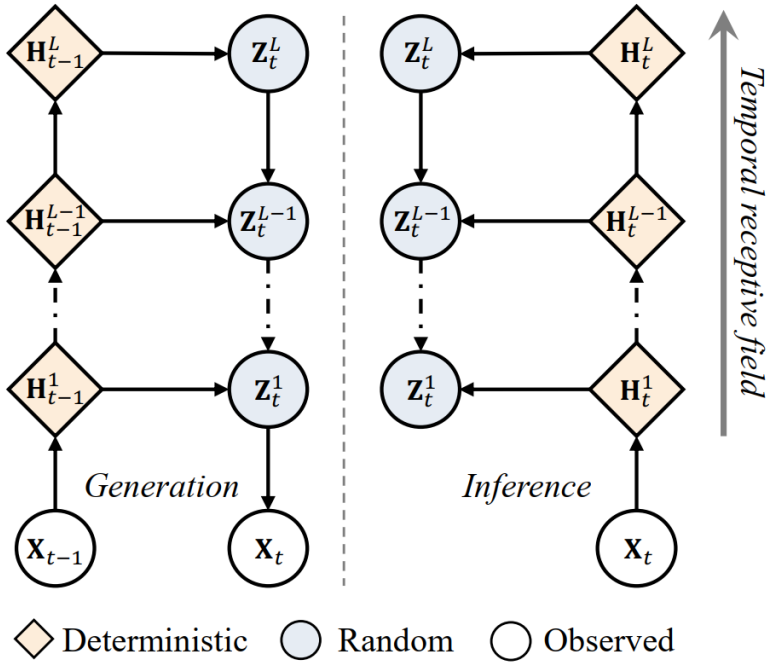
Top-Down Stochastic Stage

- Generation model

$$p_\theta\left(\mathcal{Z}_t|\mathbf{X}_{1:t-1}\right) = \prod_{n=1}^{N} p_\theta\left(\left\{\mathbf{z}_{t,n}^1, \ldots, \mathbf{z}_{t,n}^L\right\}|\mathbf{X}_{1:t-1}\right)$$
$$= \prod_{n=1}^{N} p_\theta\left(\mathbf{z}_{t,n}^L|\mathbf{h}_{t-1,n}^L\right) \prod_{l=1}^{L-1} p_\theta\left(\mathbf{z}_{t,n}^l|\mathbf{z}_{t,n}^{l+1}, \mathbf{h}_{t-1,n}^l\right),$$

- Inference model

$$q_\phi\left(\mathcal{Z}_t|\mathbf{X}_{1:t}\right) = \prod_{n=1}^{N} q_\phi\left(\mathbf{z}_{t,n}^L|\mathbf{h}_{t,n}^L\right) \prod_{l=1}^{L-1} q_\phi\left(\mathbf{z}_{t,n}^l|\mathbf{z}_{t,n}^{l+1}, \mathbf{h}_{t,n}^l\right)$$



Prediction & Optimization

- Making predictions

$$\hat{\mathbf{Y}}_{1:\tau} = \mathrm{MLP}([\mathbf{H}_T^1, \ldots, \mathbf{H}_T^L, \mathbf{Z}_T^1, \ldots, \mathbf{Z}_T^L])$$

- Loss function
  - Prediction loss
  - ELBO

$$\mathcal{L} = \mathcal{L}_{pred} + \mathcal{L}_{\mathrm{ELBO}}$$

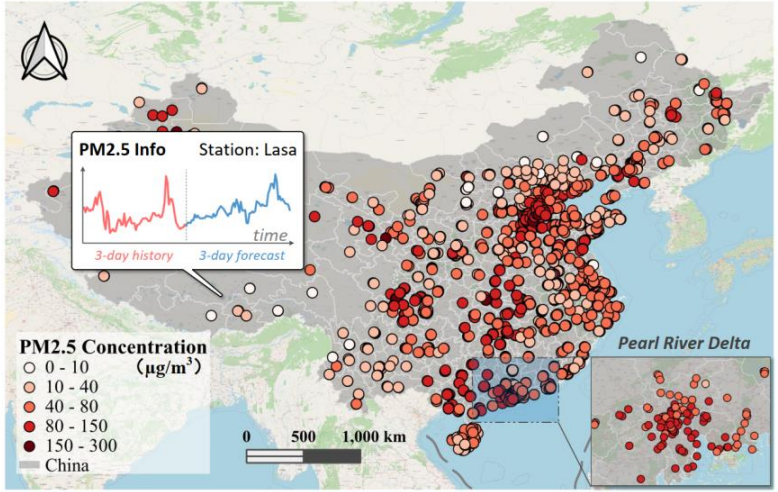
$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}}(\theta, \phi; t) &= \mathcal{L}_{rec}(\theta, \phi; t) + \mathcal{L}_{kl}(\theta, \phi; t) \\ &= -\mathbb{E}_{q_\phi(\mathcal{Z}_t|\mathbf{X}_t)}\left[\log p_\theta\left(\mathbf{X}_t|\mathcal{Z}_t\right)\right] \\ &\quad + KL\left(q_\phi\left(\mathcal{Z}_t|\mathbf{X}_{1:t}\right) \| p_\theta\left(\mathcal{Z}_t|\mathbf{X}_{1:t-1}\right)\right)\end{aligned}$$

$$\mathcal{L}_{kl} = \sum_{n=1}^{N} KL\left(q_\phi\left(\mathbf{z}_{t,n}^L|\mathbf{h}_{t,n}^L\right) \| p_\theta\left(\mathbf{z}_{t,n}^L|\mathbf{h}_{t-1,n}^L\right)\right) + \sum_{n=1}^{N}\sum_{l=1}^{L-1}$$
$$\mathbb{E}_{q_\phi}\left[KL\left(q_\phi\left(\mathbf{z}_{t,n}^l|\mathbf{z}_{t,n}^{l+1}, \mathbf{h}_{t,n}^l\right) \| p_\theta\left(\mathbf{z}_{t,n}^l|\mathbf{z}_{t,n}^{l+1}, \mathbf{h}_{t-1,n}^l\right)\right)\right]$$

Our Collected Dataset

- We evaluate our model on 4-year air quality data with 1,085 stations
- The dataset is partitioned in chronological order
  - With the first two years for training, i.e., 17,532 instances
  - The third year for validation, and the last year for test

Dataset	Venue	#Node	Range	Scale	Joint?
(Zheng et al. 2015)	KDD	35	48m	City	✗
(Yi et al. 2018)	KDD	35	48m	City	✗
(Lin et al. 2018)	GIS	35	14m	City	✓
(Wang et al. 2020)	GIS	184	48m	Nation	✓
(Xu et al. 2021)	preprint	56	12m	Province	✓
(Chen et al. 2021)	preprint	209	28m	Nation	✓
Our dataset	N/A	**1,085**	48m	Nation	✓



Model Comparison

- Following prior studies, we predict the concentration of PM2.5 over the next 24 steps given the past 24 steps, where each step stands for 3 hours
- Evaluation metrics: MAE, RMSE

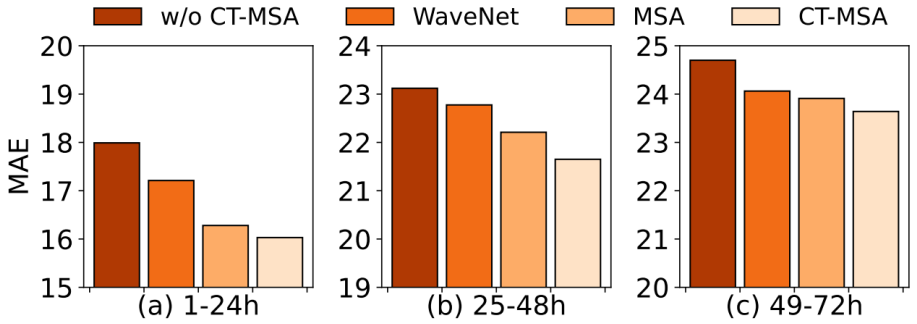
Table 2: 5-run results. The magnitude of #Param (the number of parameters) is Kilo. The bold/underlined font mean the best/the second best result. * denotes the improvement over the second best model is statistically significant at level 0.01 (Kim 2015).

Model	#Param (K)	1-24h		25-48h		49-72h		Sudden change	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA (Zhang, Zheng, and Qi 2017)	-	31.25	62.52	31.19	62.49	31.16	62.49	72.89	132.42
VAR (Toda 1991)	-	29.91	52.10	31.61	64.59	30.18	65.10	70.86	114.78
DCRNN (Li et al. 2017)	397	19.35	46.40	24.06	57.38	25.30	58.24	63.22	112.30
STGCN (Yu, Yin, and Zhu 2018)	453	19.06	42.69	24.09	56.50	25.10	58.96	61.35	111.89
GWNET (Wu et al. 2019)	825	17.79	39.49	23.32	53.17	25.00	57.01	59.33	105.75
MTGNN (Wu et al. 2020)	207	18.15	38.99	23.47	52.21	24.77	55.73	59.24	103.71
ASTGCN (Guo et al. 2019)	4,812	20.76	50.29	24.37	56.04	25.22	57.77	63.19	114.57
GMAN (Zheng et al. 2020)	269	19.60	45.70	23.79	54.25	24.89	56.33	61.83	109.57
STTN (Xu et al. 2020)	188	18.22	37.44	24.16	52.91	25.35	56.14	60.36	105.20
DeepAir (Yi et al. 2018)	183	17.47	39.12	23.40	53.48	24.95	56.92	60.26	109.95
$PM_{2.5}$-GNN (Wang et al. 2020)	101	20.20	48.96	25.04	59.56	26.31	60.46	63.64	119.00
GAGNN (Chen et al. 2021)	412	19.53	45.68	24.56	58.59	25.56	59.61	64.38	116.51
AirFormer (ours)	246	**16.03***	**32.36***	**21.65***	**44.67***	**23.64***	**50.22***	**54.92***	**90.15***

Ablation Study

- Effects of DS-MSA & CT-MSA

Variant	Time/epo	1-24h	25-48h	49-72h
w/o DS-MSA	1,284	17.24	23.55	24.96
MSA	2,818	18.19	23.24	24.23
Local MSA	2,157	18.63	23.79	24.98
DS-MSA (50-200)	1,708	**16.03**	21.65	23.64
DS-MSA (50)	1,547	16.43	22.87	24.32
DS-MSA (50-200-500)	2,285	16.09	**21.30**	**22.85**

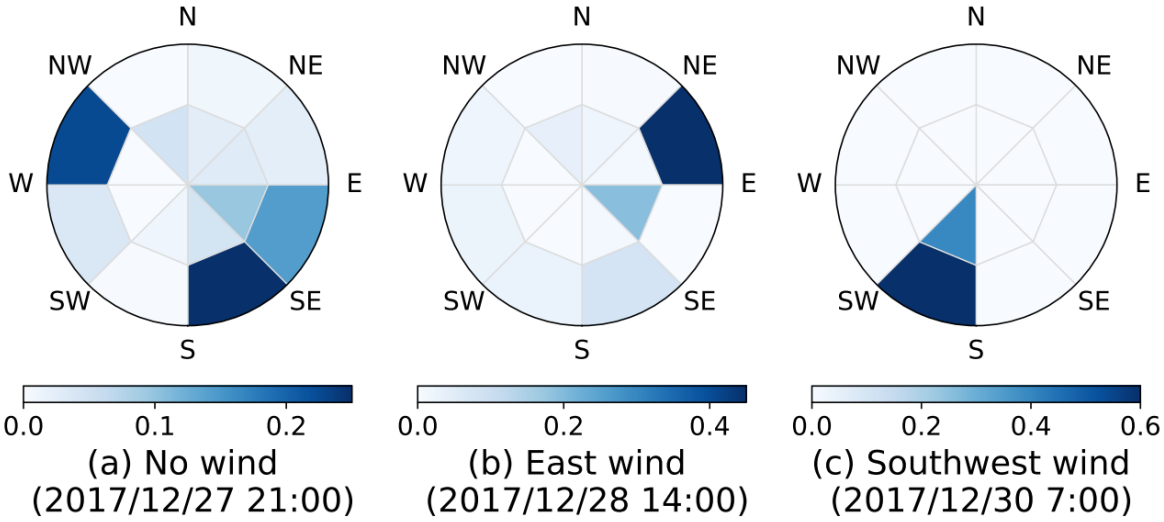


- Effects of the stochastic stage

Varaint	Time/epo	1-72h	Sudden change
w/o Stochastic	1,559	20.97	57.52
AirFormer	1,708	**20.44**	**54.92**

Visualization of DS-MSA

- We perform a case study on a 2-circle dartboard centered by Xizhimen (a hybrid district in Beijing) from Dec. 27-30, 2017



Conclusion

- We have devised AirFormer for nationwide air quality prediction in China. To the best of our knowledge, this is the first work for collectively forecasting air quality among thousands of locations.
- Our model elaborately combines the efficient spatio-temporal learning capabilities of transformers with the uncertainty measurement of stochastic latent spaces
- Compared to prior methods, AirFormer reduces the prediction errors by 4.6~8.2%
- In the future, we will explore
  - Online learning and deploy our model to support public use
  - Defeating the distribution shift within air quality data

Q&A

Thanks!

CityMind Lab

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