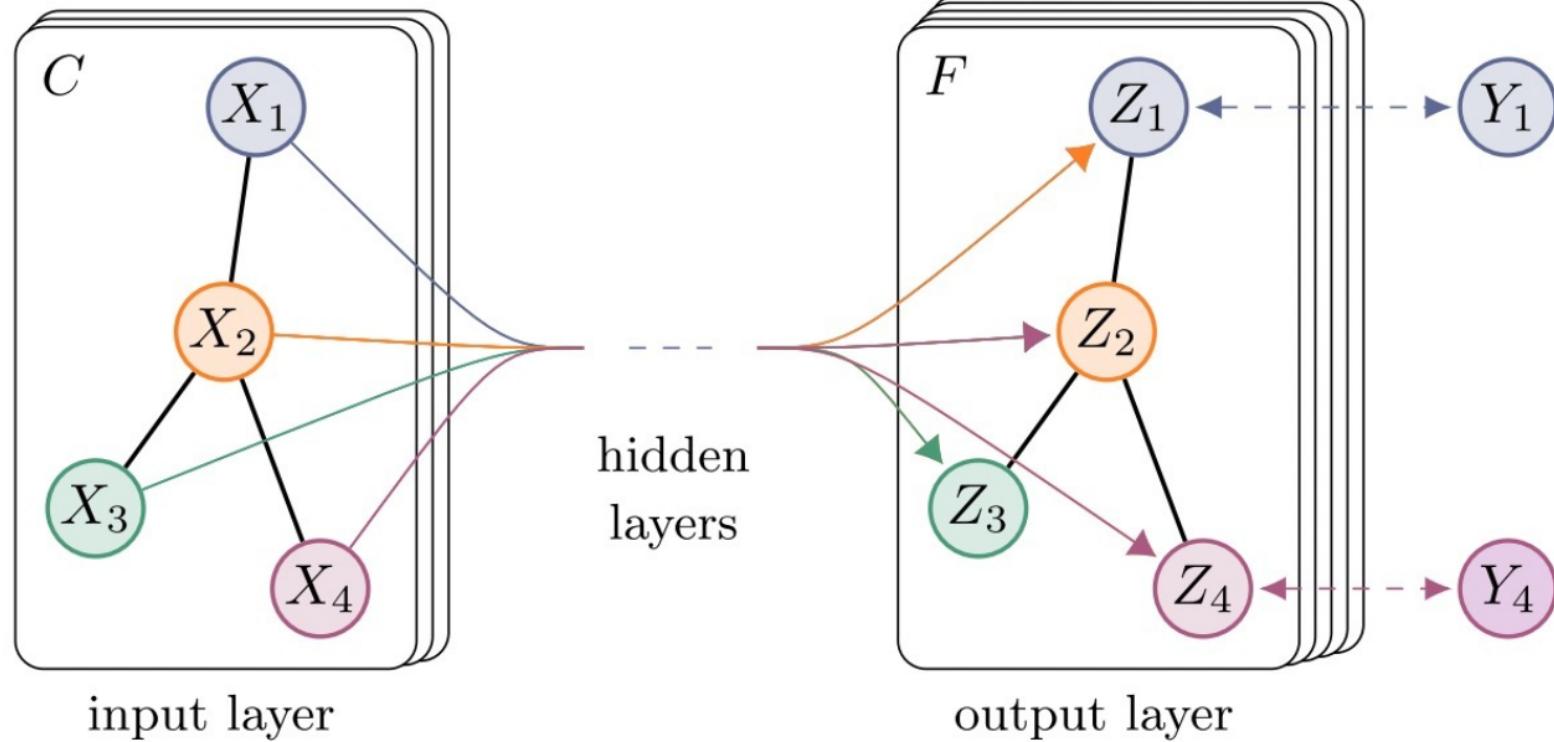


Learning on Graph



$$\begin{aligned}\mathbf{h}_u^{(k+1)} &= \text{UPDATE}^{(k)} \left(\mathbf{h}_u^{(k)}, \text{AGGREGATE}^{(k)}(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\}) \right) \\ &= \text{UPDATE}^{(k)} \left(\mathbf{h}_u^{(k)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)} \right),\end{aligned}$$

Extended Graph Neural Network

新增连边的表征，可以更好的利用边上的信息。

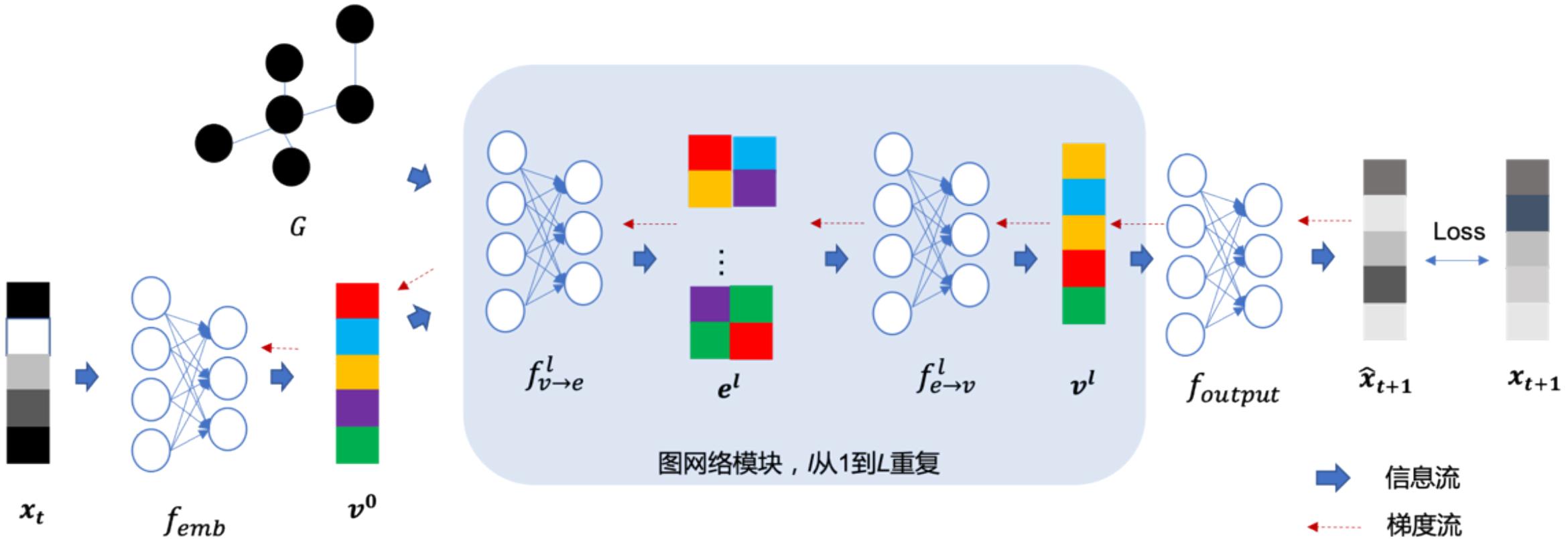
$$\mathbf{h}_{(u,v)}^{(k)} = \text{UPDATE}_{\text{edge}} \left(\mathbf{h}_{(u,v)}^{(k-1)}, \mathbf{h}_u^{(k-1)}, \mathbf{h}_v^{(k-1)}, \mathbf{h}_{\mathcal{G}}^{(k-1)} \right)$$

$$\mathbf{m}_{\mathcal{N}(u)} = \text{AGGREGATE}_{\text{node}} \left(\{\mathbf{h}_{(u,v)}^{(k)} \forall v \in \mathcal{N}(u)\} \right)$$

$$\mathbf{h}_u^{(k)} = \text{UPDATE}_{\text{node}} \left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}, \mathbf{h}_{\mathcal{G}}^{(k-1)} \right)$$

$$\mathbf{h}_{\mathcal{G}}^{(k)} = \text{UPDATE}_{\text{graph}} \left(\mathbf{h}_{\mathcal{G}}^{(k-1)}, \{\mathbf{h}_u^{(k)} \forall u \in \mathcal{V}\}, \{\mathbf{h}_{(u,v)}^{(k)} \forall (u,v) \in \mathcal{E}\} \right)$$

Dynamics Learning on Graph



PM2.5 Forecasting



王硕



李嫣然

Home > Conferences > GIS > Proceedings > SIGSPATIAL '20 > PM2.5-GNN: A Domain Knowledge Enhanced Graph Neural Network For PM2.5 Forecasting

POSTER

PM2.5-GNN: A Domain Knowledge Enhanced Graph Neural Network For PM2.5 Forecasting

Authors: Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, Fei Gao [Authors Info & Claims](#)

SIGSPATIAL '20: Proceedings of the 28th International Conference on Advances in Geographic Information Systems • November 2020 • Pages 163–166 • <https://doi.org/10.1145/3397536.3422208>

Published: 13 November 2020 [Publication History](#)

Check for updates

25 751

Get Access



SIGSPATIAL '20:
Proceedings of the...

PM2.5-GNN: A Domain
Knowledge Enhanced...

Pages 163–166

[← Previous](#) [Next →](#)

ABSTRACT

References

ABSTRACT

When predicting PM2.5 concentrations, it is necessary to consider complex information sources since the concentrations are influenced by various factors within a long period. In this paper, we identify a set of critical domain knowledge for PM2.5 forecasting and develop a novel graph based model, PM2.5-GNN, being capable of capturing long-term dependencies. On a real-world dataset, we validate the effectiveness of the proposed model and examine its abilities of capturing both fine-grained and long-term influences in PM2.5 process. The proposed PM2.5-GNN has also been

Tasks

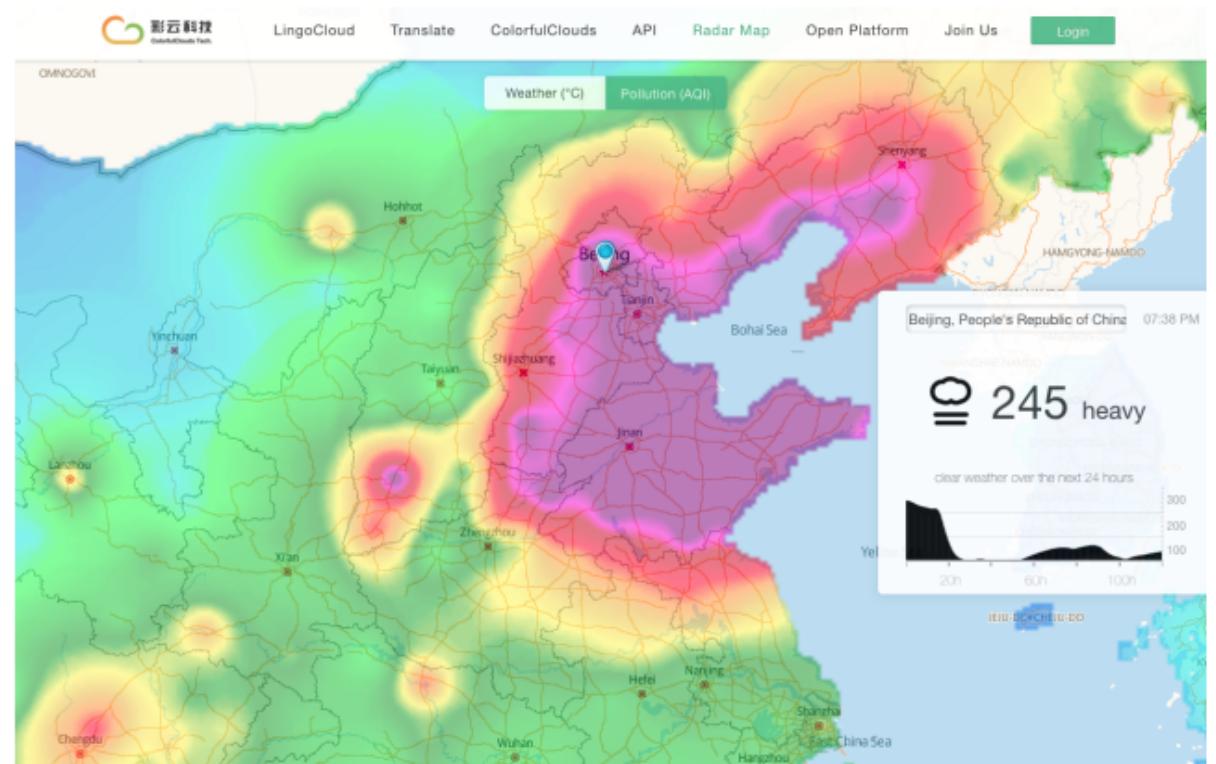
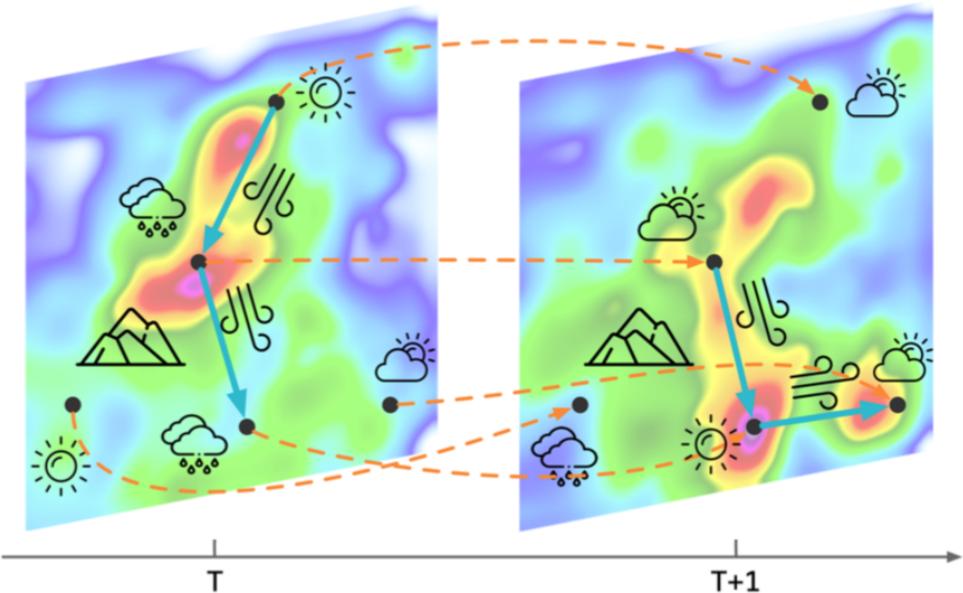


Figure 2: Online website (<http://caiyunapp.com/map/>) that provides 72-hour real-time PM_{2.5} concentration prediction using PM_{2.5}-GNN model proposed in this paper.

Input Attributes

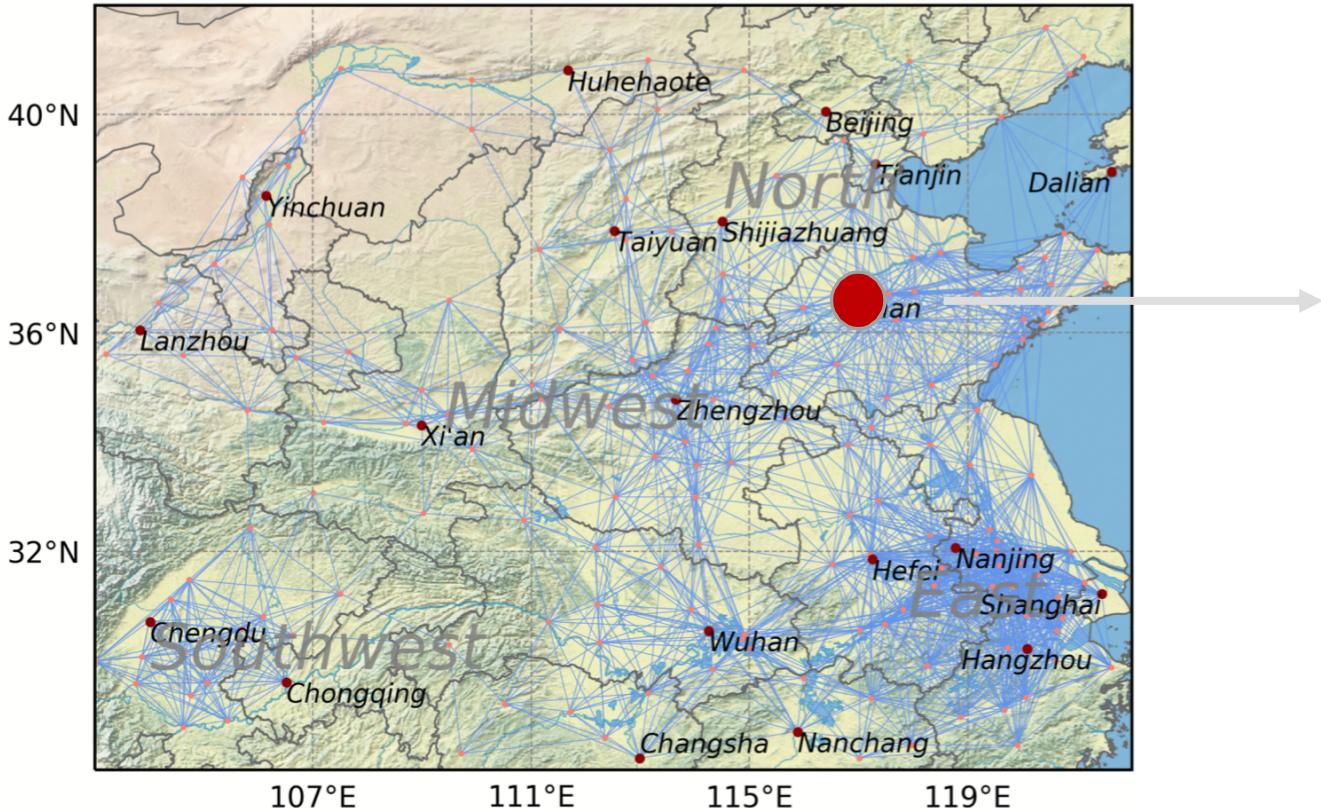
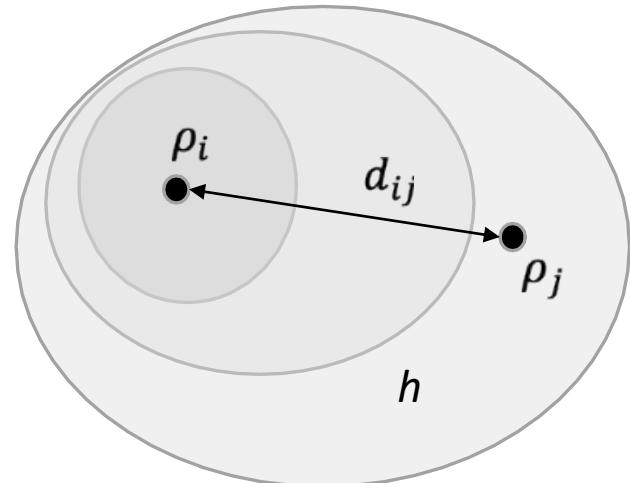


Table 1: Meteorological Attributes of Nodes (P)

Variable Name	Unit
Planetary Boundary Layer (PBL) height	m
K index	K
u-component of wind	m/s
v-component of wind	m/s
2m Temperature	K
Relative humidity	%
Total precipitation	m
Surface pressure	Pa

Network Constructing



$$A_{ij} = H(d_\theta - d_{ij}) \cdot H(m_\theta - m_{ij}), \quad \text{where } H(x) = \begin{cases} 1. & \text{if } x > 0 \\ 0. & \text{if } x \leq 0 \end{cases}$$

$$d_{ij} = \|\rho_i - \rho_j\|$$

$$m_{ij} = \sup_{\lambda \in (0,1)} \{h(\lambda\rho_i + (1-\lambda)\rho_j) - \max\{h(\rho_i), h(\rho_j)\}\} \quad (6)$$

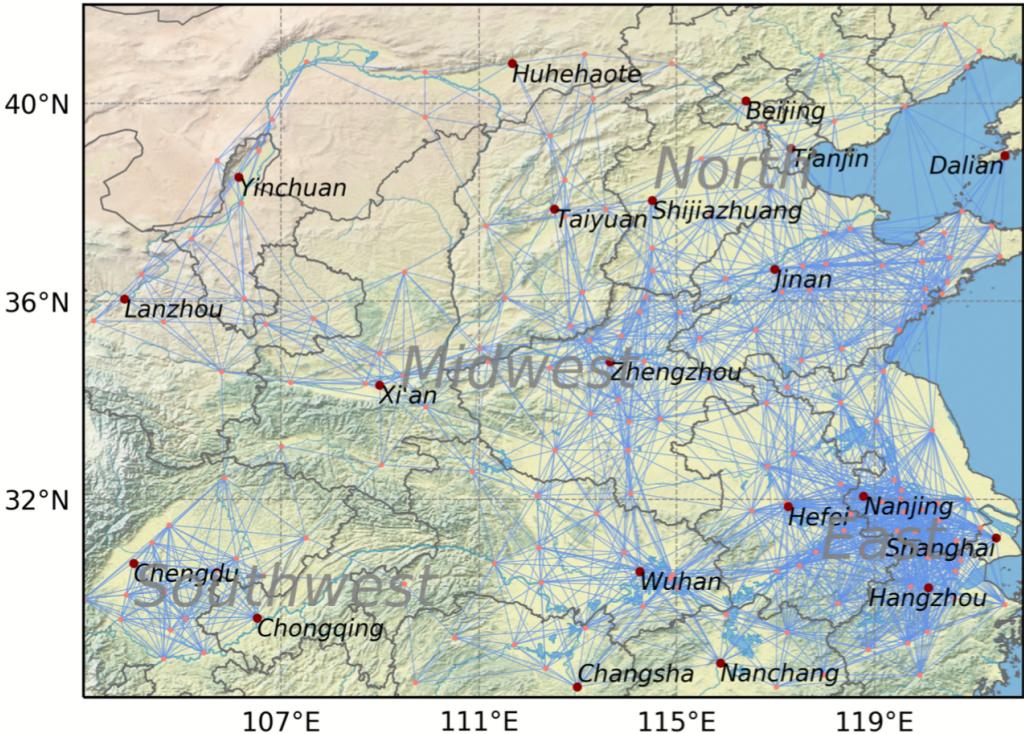


Table 2: Attributes of Edges (Q)

Variable Name	Unit
Wind speed of source node $ v $	km/h
Distance between source and sink d	km
Wind direction of source node β	(°)
Direction from source to sink γ	(°)
Advection coefficient S (Equation 5)	%

Model

$$\xi_i^t = [\hat{X}_i^{t-1}, P_i^t] \quad \forall i \in V$$

$$e_{j \rightarrow i}^t = \Psi([\xi_j^t, \xi_i^t, Q_{j \rightarrow i}^t]) \quad \forall \langle j, i \rangle \in E$$

$$\zeta_i^t = \Phi\left(\sum_{j \in N(i)} (e_{j \rightarrow i}^t - e_{i \rightarrow j}^t)\right) \quad \forall i \in V$$

$$x_i^t = [\xi_i^t, \zeta_i^t]$$

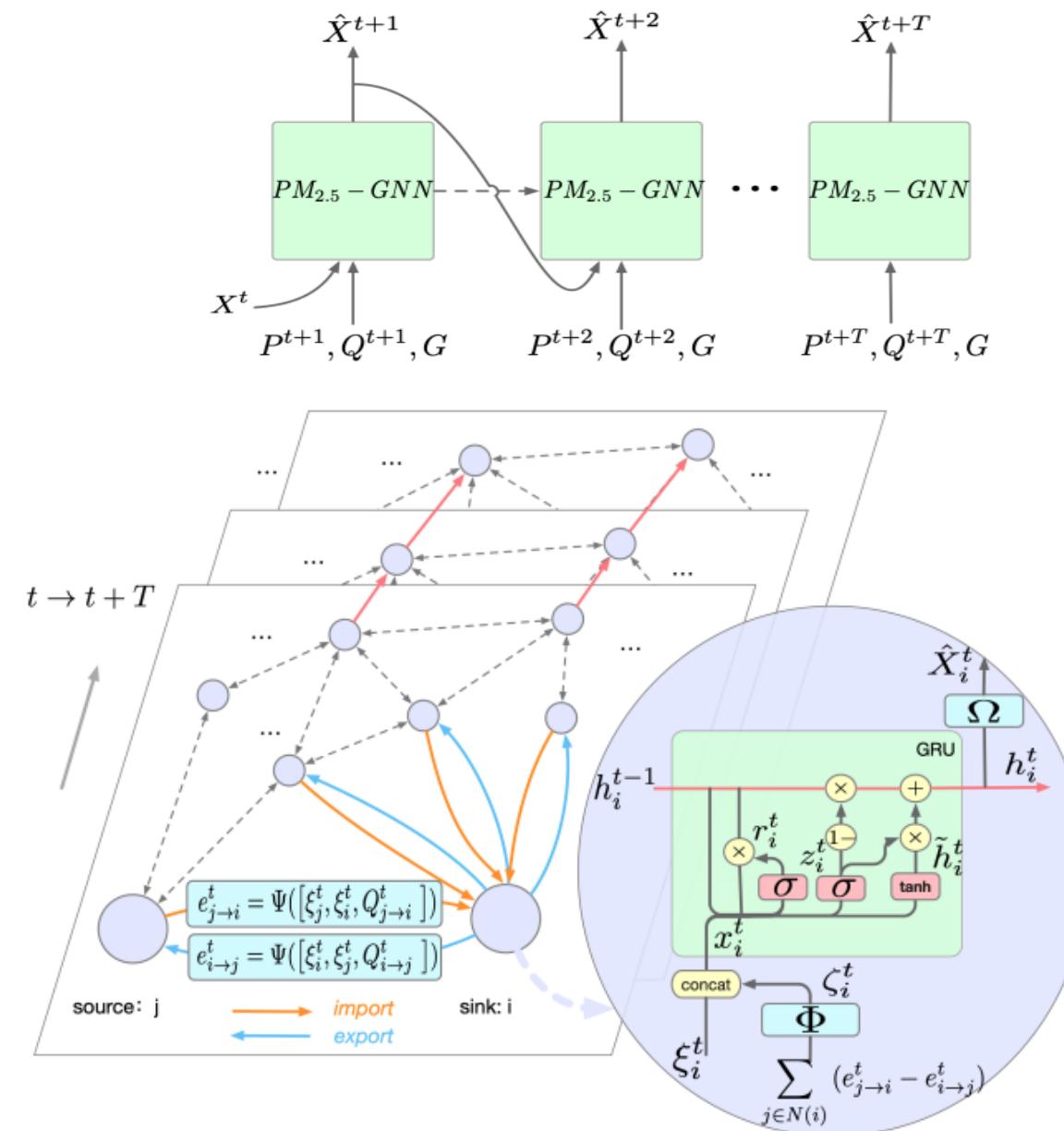
$$z_i^t = \sigma(W_z \cdot [h_i^{t-1}, x_i^t])$$

$$r_i^t = \sigma(W_r \cdot [h_i^{t-1}, x_i^t])$$

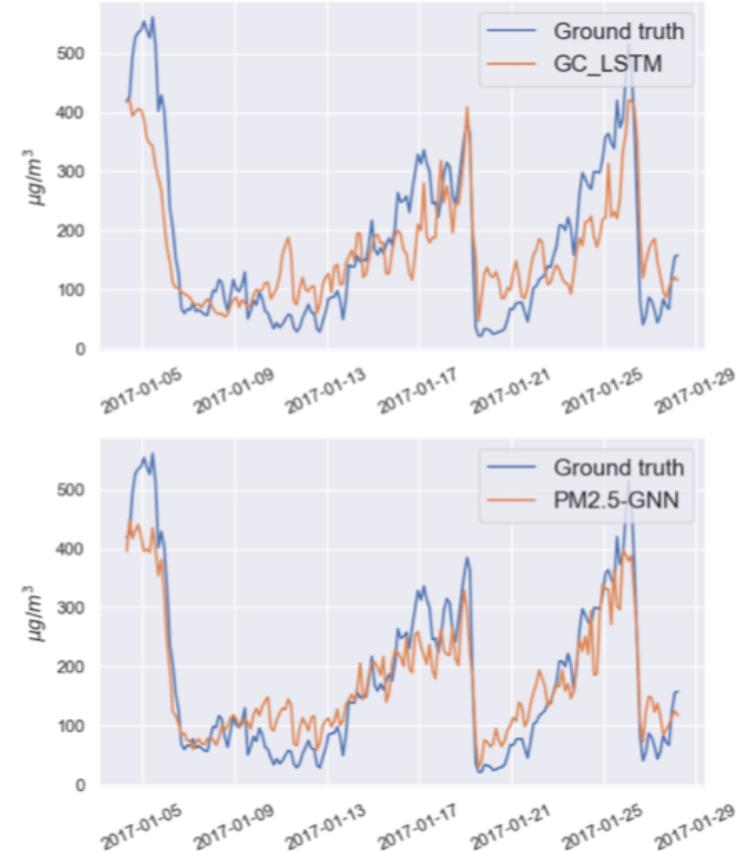
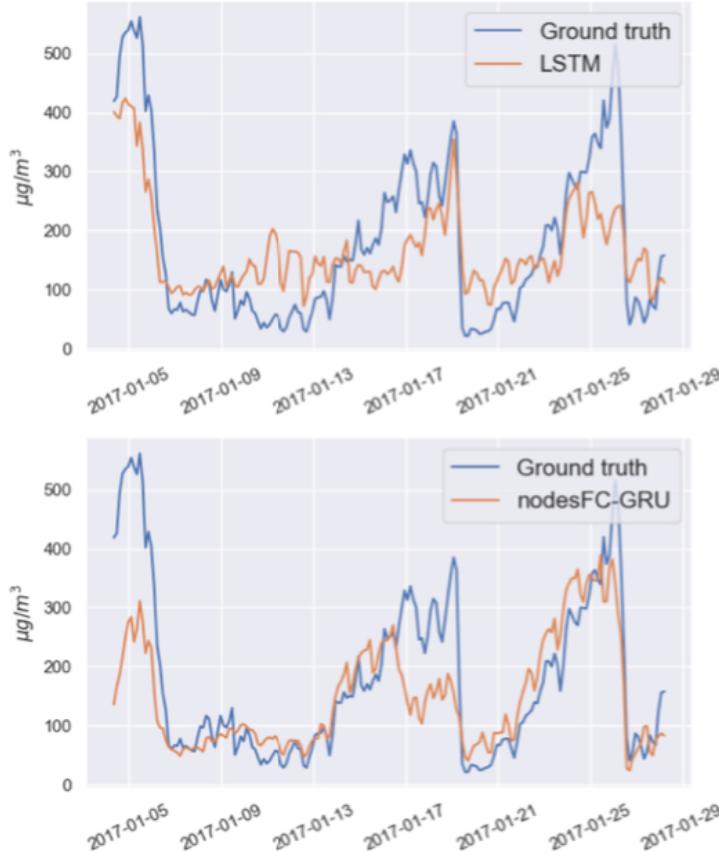
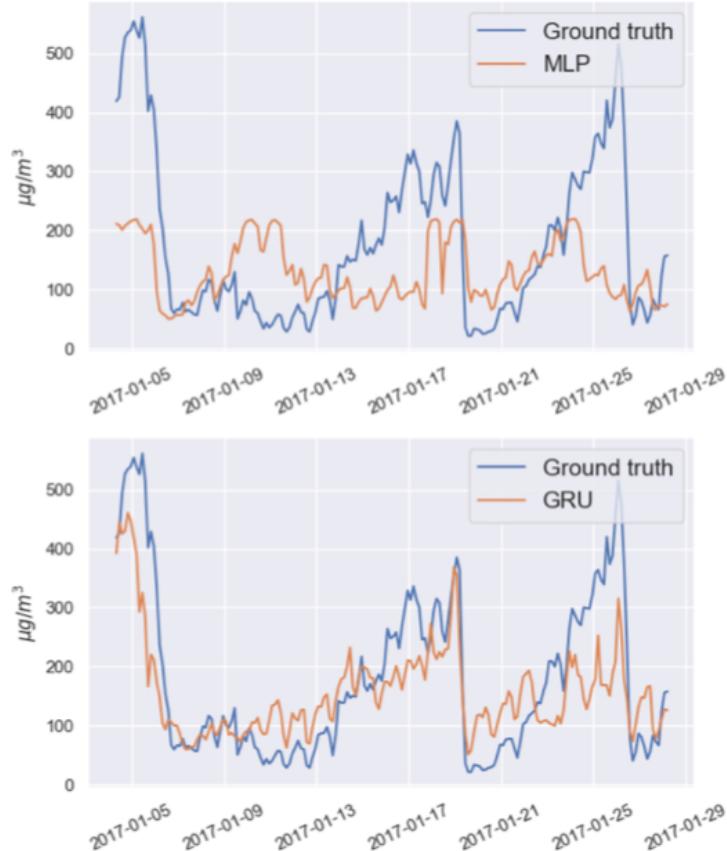
$$\tilde{h}_i^t = \tanh(W \cdot [r_i^t * h_i^{t-1}, x_i^t])$$

$$h_i^t = (1 - z_i^t) * h_i^{t-1} + z_i^t * \tilde{h}_i^t$$

$$\hat{X}_i^t = \Omega(h_i^t) \quad \forall i \in V$$



Results – Time Series



Results

Table 3: Dataset is spilt into 3 sub-datasets.

Dataset	Train	Validate	Test
1	2015/1/1 - 2016/12/31	2017/1/1 - 2017/12/31	2018/1/1 - 2018/12/31
2	2015/11/1 - 2016/2/28	2016/11/1 - 2017/2/28	2017/11/1 - 2018/2/28
3	2016/9/1 - 2016/11/30	2016/12/1 - 2016/12/31	2017/1/1 - 2017/1/31

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}$$

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n}$$

$$CSI = \frac{\text{hits}}{\text{hits} + \text{misses} + \text{falsealarms}}$$

$$FAR = \frac{\text{falsealarms}}{\text{hits} + \text{falsealarms}}$$

$$POD = \frac{\text{hits}}{\text{hits} + \text{misses}}$$

Table 4: Overall performance on all three sub-datasets. Best scores are in bold.

Dataset	Metric	MLP	LSTM	GRU	GC-LSTM	nodesFC-GRU	PM _{2.5} -GNN
	Train_Loss	0.5984 ± 0.0981	0.4424 ± 0.0399	0.4635 ± 0.0383	0.4157 ± 0.0484	0.3263 ± 0.0624	0.4692 ± 0.2853
	Validate_Loss	0.5279 ± 0.0495	0.4514 ± 0.0136	0.4545 ± 0.0129	0.4455 ± 0.0125	0.4138 ± 0.0097	0.4158 ± 0.0177
	Test_Loss	0.5647 ± 0.0454	0.4805 ± 0.0137	0.4830 ± 0.0124	0.4732 ± 0.0132	0.4395 ± 0.0088	0.4359 ± 0.0187
1	RMSE	22.98 ± 0.98	21.07 ± 0.38	21.13 ± 0.37	20.90 ± 0.40	20.28 ± 0.29	20.16 ± 0.48
	MAE	18.37 ± 0.94	16.68 ± 0.39	16.77 ± 0.37	16.53 ± 0.41	15.98 ± 0.30	15.91 ± 0.49
	CSI	40.77 ± 2.69%	44.87 ± 1.09%	44.71 ± 0.99%	45.64 ± 1.10%	47.61 ± 0.92%	47.91 ± 1.65%
	POD	51.43 ± 5.68%	56.43 ± 2.43%	56.17 ± 2.45%	57.98 ± 2.51%	59.79 ± 2.11%	60.33 ± 3.42%
	FAR	32.80 ± 4.29%	31.21 ± 1.68%	31.16 ± 1.80%	31.65 ± 1.73%	29.87 ± 1.43%	29.83 ± 2.36%
2	Train_Loss	0.6094 ± 0.1291	0.4638 ± 0.0861	0.4808 ± 0.0806	0.4417 ± 0.0951	0.3368 ± 0.1227	0.4437 ± 0.0988
	Validate_Loss	0.6356 ± 0.1113	0.5529 ± 0.0396	0.5465 ± 0.0359	0.5473 ± 0.0379	0.5575 ± 0.0339	0.5173 ± 0.0469
	Test_Loss	0.6570 ± 0.0963	0.5825 ± 0.0350	0.5700 ± 0.0353	0.5750 ± 0.0413	0.5747 ± 0.0337	0.5375 ± 0.0494
	RMSE	35.55 ± 2.76	33.53 ± 1.04	33.09 ± 1.00	33.20 ± 1.23	33.07 ± 1.03	32.11 ± 1.47
	MAE	28.67 ± 2.52	26.90 ± 1.04	26.54 ± 0.97	26.57 ± 1.22	26.40 ± 0.97	25.68 ± 1.42
3	CSI	45.52 ± 5.49%	49.75 ± 2.09%	49.83 ± 1.79%	50.13 ± 2.50%	48.79 ± 1.38%	51.35 ± 2.53%
	POD	60.85 ± 9.17%	64.94 ± 3.30%	64.58 ± 3.03%	64.54 ± 3.49%	61.29 ± 2.07%	66.24 ± 4.56%
	FAR	34.56 ± 6.21%	31.88 ± 2.28%	31.31 ± 2.44%	30.73 ± 2.80%	29.37 ± 2.60%	30.11 ± 3.67%
	Train_Loss	0.6443 ± 0.1234	0.4627 ± 0.0907	0.4631 ± 0.0877	0.4481 ± 0.1007	0.3134 ± 0.1226	0.4358 ± 0.0980
	Validate_Loss	0.7517 ± 0.1300	0.5864 ± 0.0587	0.5868 ± 0.0600	0.5798 ± 0.0638	0.6156 ± 0.0483	0.5390 ± 0.0708
4	Test_Loss	0.6389 ± 0.1119	0.5280 ± 0.0476	0.5224 ± 0.0464	0.5167 ± 0.0581	0.5757 ± 0.0340	0.4933 ± 0.0584
	RMSE	50.70 ± 4.57	46.19 ± 2.04	46.06 ± 2.03	45.71 ± 2.38	47.97 ± 1.67	44.36 ± 2.85
	MAE	41.89 ± 4.22	37.97 ± 1.94	37.94 ± 1.92	37.46 ± 2.29	39.03 ± 1.65	36.32 ± 2.81
	CSI	52.44 ± 3.81%	58.85 ± 2.36%	59.16 ± 1.87%	58.98 ± 2.47%	58.84 ± 1.60%	60.57 ± 2.78%
	POD	74.16 ± 7.25%	81.03 ± 3.14%	83.32 ± 1.95%	81.92 ± 2.91%	79.40 ± 1.71%	83.94 ± 3.34%
5	FAR	35.25 ± 5.32%	31.71 ± 2.38%	32.86 ± 2.37%	32.18 ± 2.36%	30.51 ± 2.28%	31.37 ± 3.63%

Ablation – Domain Knowledge

Table 5: Experimental results of PM_{2.5}-GNN’s different configurations. Lack of PBL feature or subtraction component worsens PM_{2.5}-GNN’s performance.

Dataset	Metric	PM _{2.5} -GNN	no PBL height	no export
1	RMSE	20.16 ± 0.48	20.46 ± 0.43	20.98 ± 0.33
	MAE	15.91 ± 0.49	16.12 ± 0.44	16.67 ± 0.35
	CSI	47.91 ± 1.65%	46.70 ± 1.48%	45.41 ± 1.17%
2	RMSE	32.11 ± 1.47	33.25 ± 1.65	32.70 ± 1.31
	MAE	25.68 ± 1.42	26.67 ± 1.59	26.16 ± 1.27
	CSI	51.35 ± 2.53%	49.42 ± 2.90%	50.41 ± 2.43%
3	RMSE	44.36 ± 2.85	46.12 ± 3.38	44.80 ± 2.59
	MAE	36.32 ± 2.81	38.04 ± 3.38	36.78 ± 2.53
	CSI	60.57 ± 2.78%	58.72 ± 3.15%	60.12 ± 2.38%

Large Models

Large Models for Time Series and Spatio-Temporal Data: A Survey and Outlook

Ming Jin, Qingsong Wen[†], Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang,

James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, Shirui Pan[†],

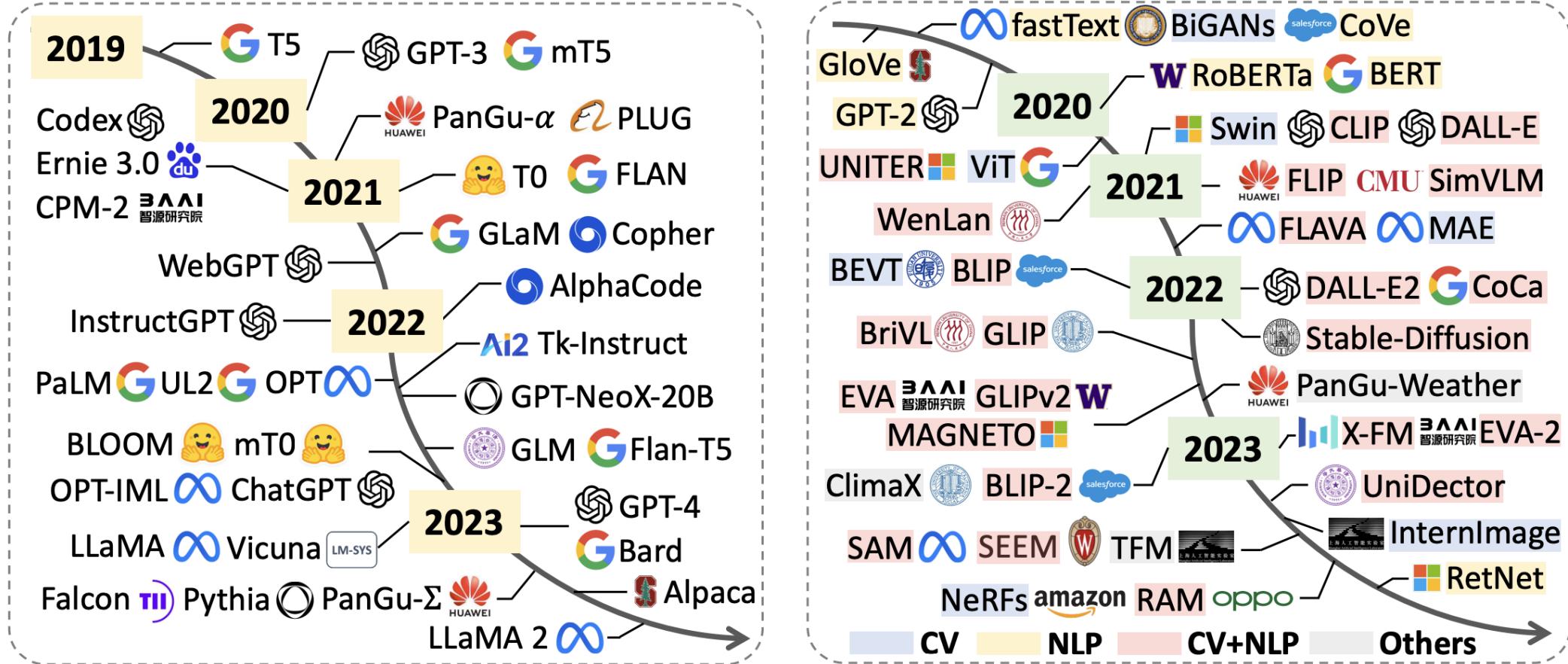
Vincent S. Tseng, *Fellow, IEEE*, Yu Zheng, *Fellow, IEEE*, Lei Chen, *Fellow, IEEE*, Hui Xiong, *Fellow, IEEE*

Abstract—Temporal data, notably time series and spatio-temporal data, are prevalent in real-world applications. They capture dynamic system measurements and are produced in vast quantities by both physical and virtual sensors. Analyzing these data types is vital to harnessing the rich information they encompass and thus benefits a wide range of downstream tasks. Recent advances in large language and other foundational models have spurred increased use of these models in time series and spatio-temporal data mining. Such methodologies not only enable enhanced pattern recognition and reasoning across diverse domains but also lay the groundwork for artificial general intelligence capable of comprehending and processing common temporal data. In this survey, we offer a comprehensive and up-to-date review of large models tailored (or adapted) for time series and spatio-temporal data, spanning four key facets: data types, model categories, model scopes, and application areas/tasks. Our objective is to equip practitioners with the knowledge to develop applications and further research in this underexplored domain. We primarily categorize the existing literature into two major clusters: large models for time series analysis (LM4TS) and spatio-temporal data mining (LM4STD). On this basis, we further classify research based on model scopes (i.e., general vs. domain-specific) and application areas/tasks. We also provide a comprehensive collection of pertinent resources, including datasets, model assets, and useful tools, categorized by mainstream applications. This survey coalesces the latest strides in large model-centric research on time series and spatio-temporal data, underscoring the solid foundations, current advances, practical applications, abundant resources, and future research opportunities.

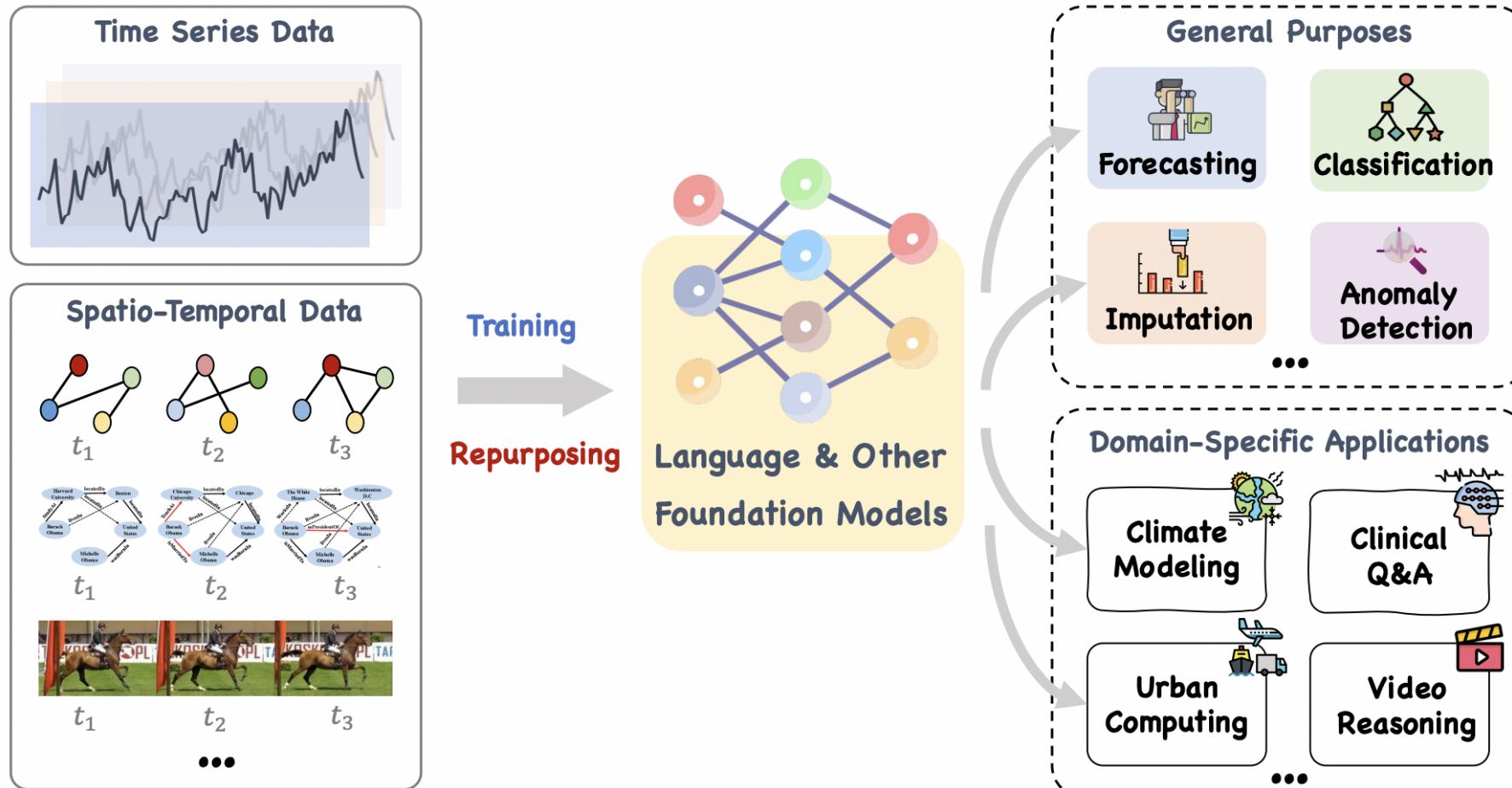
Index Terms—Large language models, pre-trained foundation models, large models, time series, spatio-temporal data, temporal data.



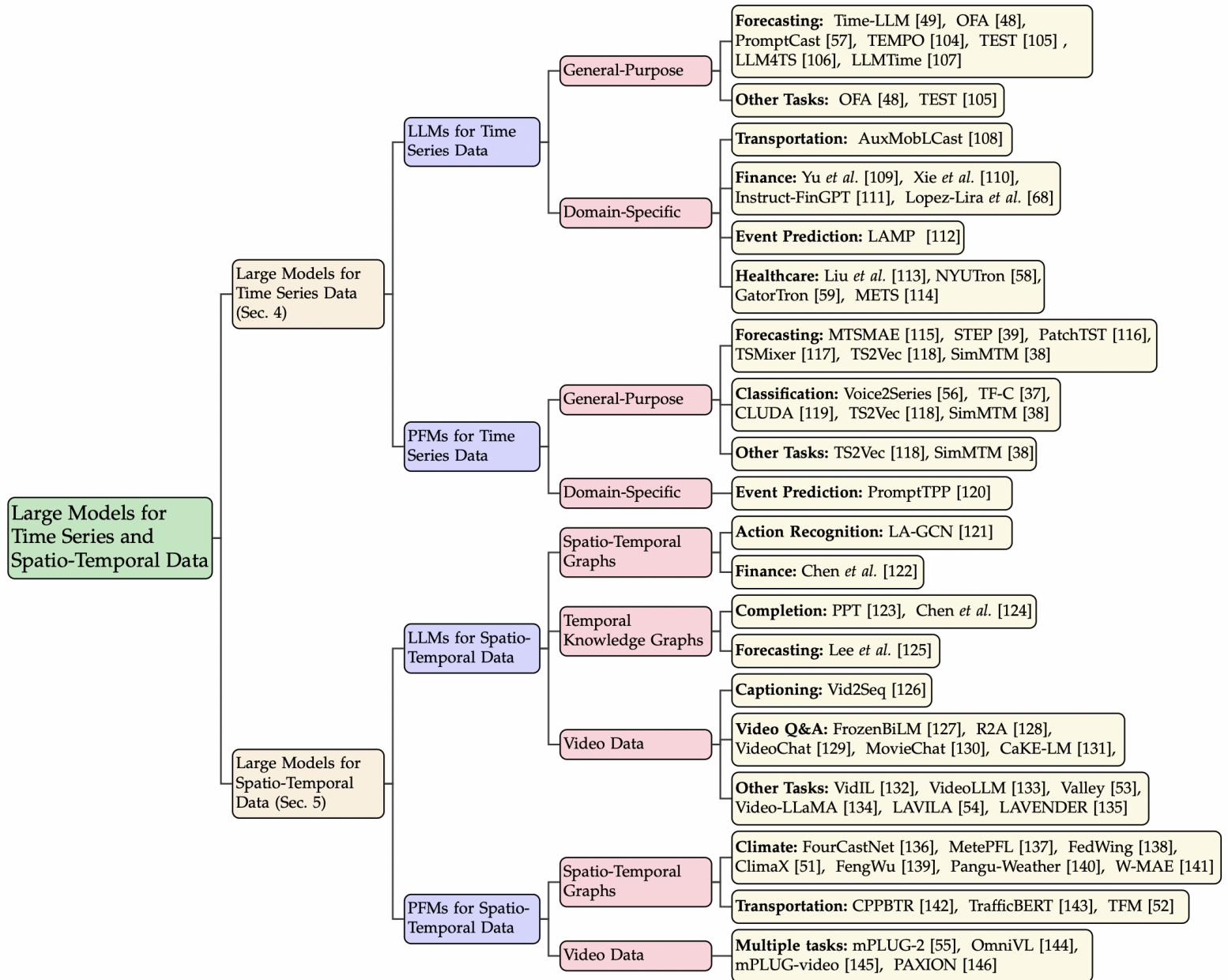
Large Models



Large Models



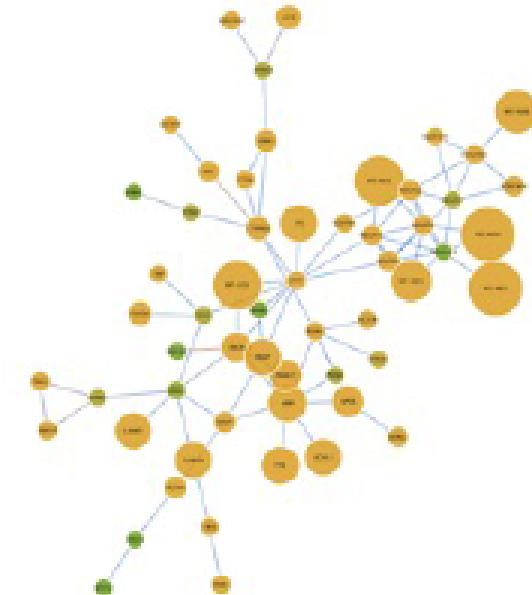
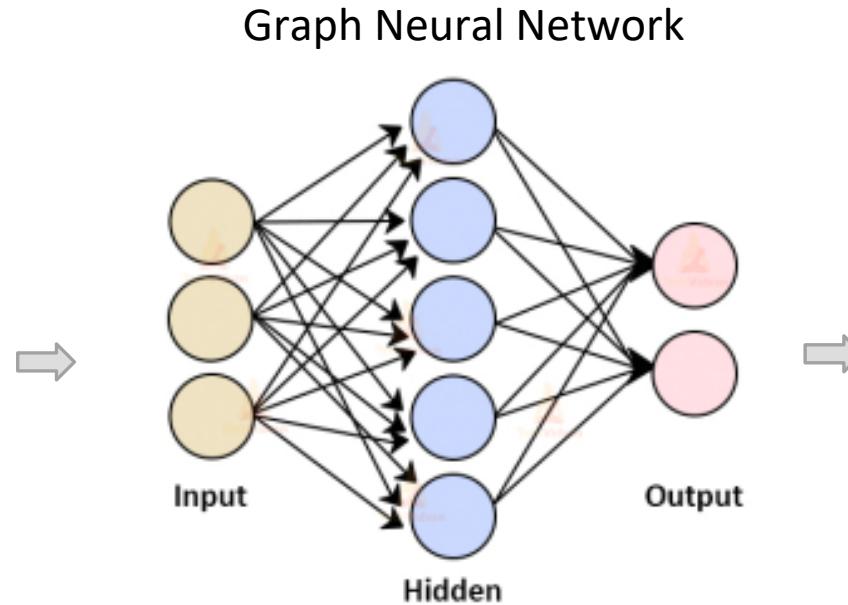
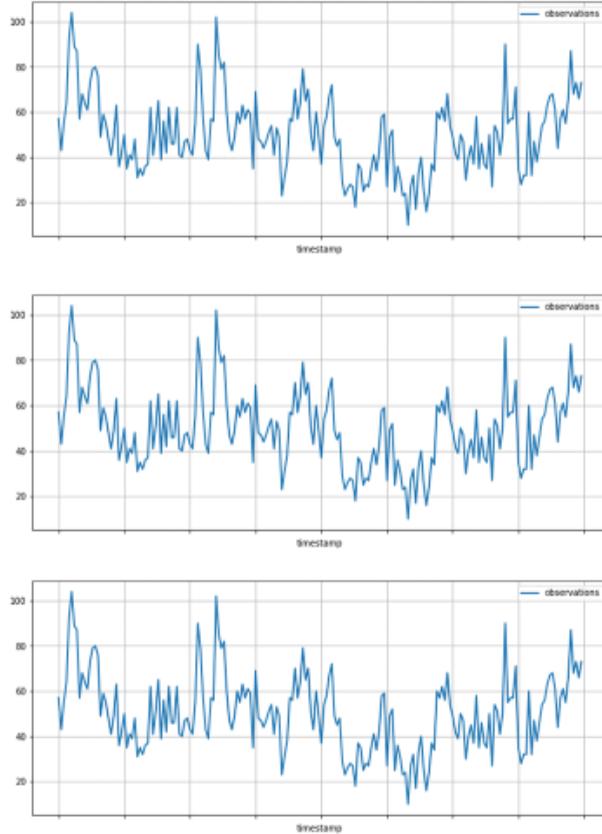
Large Models



Outline

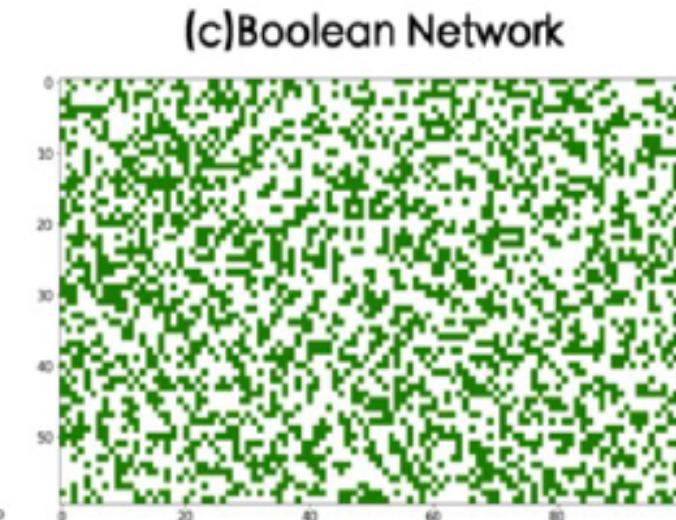
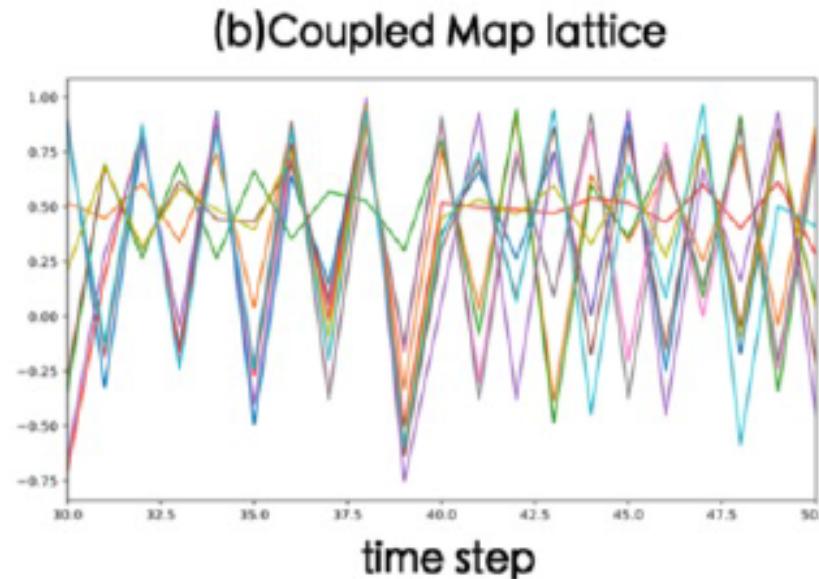
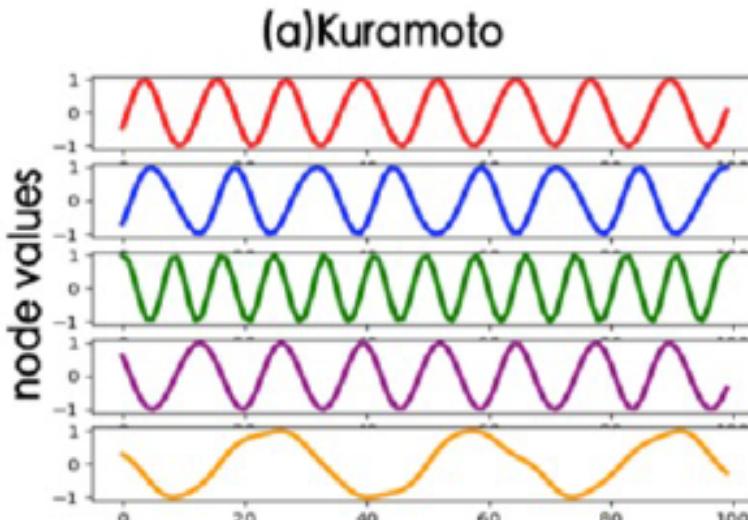
- AI for complex systems
 - Complex Systems and Modelling Methods
 - Representation & Generation
 - Dynamics Learning
 - Network Reconstruction
 - Multi-scale Modelling
 - Simulation, optimization, and control
- Complexity science for AI

Network Reconstruction



Inferred Network and
Dynamics

Various Dynamics



$$\frac{d\phi_i}{dt} = \omega_i + k \sum_{j \neq i} A_{ij} \sin(\phi_i - \phi_j)$$

$$x_i(t+1) = (1-s)h(x_i(t)) + \frac{s}{D} \sum_{j \neq i} h(x_j(t))$$

$$h(x) = \lambda x(1-x)$$

x_1, x_2, x_3, x_4	\longrightarrow	x_1, x_2, x_3, x_4
0 0 0 0	\longrightarrow	0 0 0 0
0 0 0 1	\longrightarrow	1 0 0 1
0 0 1 0	\longrightarrow	1 0 0 0
0 0 1 1	\longrightarrow	1 0 0 0
0 1 0 0	\longrightarrow	0 1 0 0

Gumbel Graph Network

RESEARCH

A General Deep Learning Framework for Network Reconstruction and Dynamics Learning

Zhang Zhang¹, Yi Zhao³, Jing Liu¹, Shuo Wang², Ruyi Tao⁴, Ruyue Xin¹ and Jiang Zhang^{1*}

*Correspondence:
zhangjiang@bnu.edu.cn

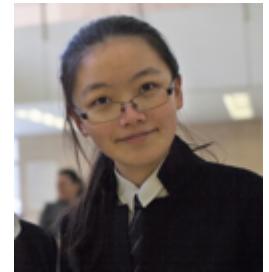
¹School of Systems Science,
Beijing Normal University, 100875
Beijing, People's Republic of China
Full list of author information is
available at the end of the article

Abstract

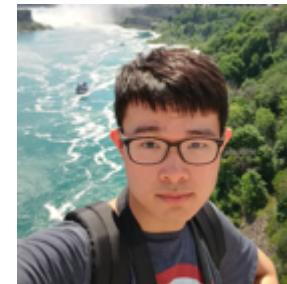
Many complex processes can be viewed as dynamical systems on networks. However, in real cases, only the performances of the system are known, the network structure and the dynamical rules are not observed. Therefore, recovering latent network structure and dynamics from observed time series data are important tasks because it may help us to open the black box, and even to build up the model of a complex system automatically. Although this problem hosts a wealth of potential applications in biology, earth science, and epidemics etc., conventional methods have limitations. In this work, we introduce a new framework, Gumbel Graph Network (GGN), which is a model-free, data-driven deep learning framework to accomplish the reconstruction of both network connections and the dynamics on it. Our model consists of two jointly trained parts: a network generator that generating a discrete network with the Gumbel Softmax technique; and a dynamics learner that utilizing the generated network and one-step trajectory value to predict the states in future steps. We exhibit the universality of our framework on different kinds of time-series data: with the same structure, our model can be trained to accurately recover the network structure and predict future states on continuous, discrete, and binary dynamics, and outperforms competing network reconstruction methods.



张章



赵旨



刘晶



王硕



陶如意



张江



张妍

PHYSICAL REVIEW E

covering statistical, nonlinear, biological, and soft matter physics[Highlights](#) [Recent](#) [Accepted](#) [Collections](#) [Authors](#) [Referees](#) [Search](#) [Press](#) [About](#) [Editorial Team](#)

Universal framework for reconstructing complex networks and node dynamics from discrete or continuous dynamics data

Yan Zhang, Yu Guo, Zhang Zhang, Mengyuan Chen, Shuo Wang, and Jiang Zhang
Phys. Rev. E **106**, 034315 – Published 16 September 2022[Article](#)[References](#)[No Citing Articles](#)[Supplemental Material](#)[PDF](#)[HTML](#)[Export Citation](#)

ABSTRACT

Many dynamical processes of complex systems can be understood as the dynamics of a group of nodes interacting on a given network structure. However, finding such interaction structure and node dynamics from time series of node behaviors is tough. Conventional methods focus on either network structure inference task or dynamics reconstruction problem, very few of them can work well on both. This paper proposes a universal framework for reconstructing network structure and node dynamics at the same time from observed time-series data of nodes. We use a differentiable Bernoulli sampling process to generate a candidate network structure, and we use neural networks to simulate the node

Issue

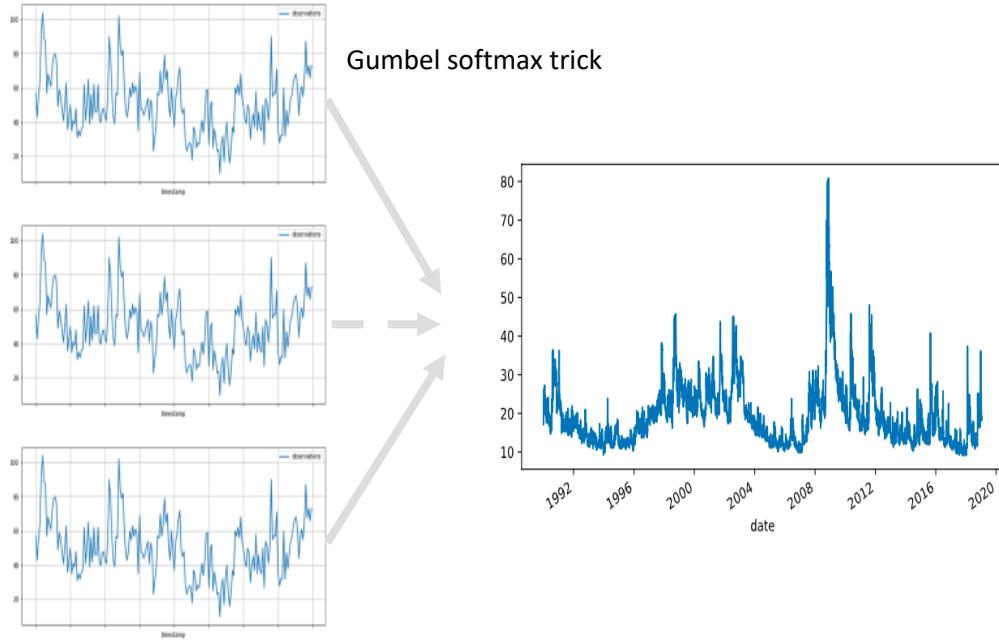
Vol. 106, Iss. 3 — September 2022



Check for updates

<https://journals.aps.org/pre/abstract/10.1103/PhysRevE.106.034315>

Key Idea



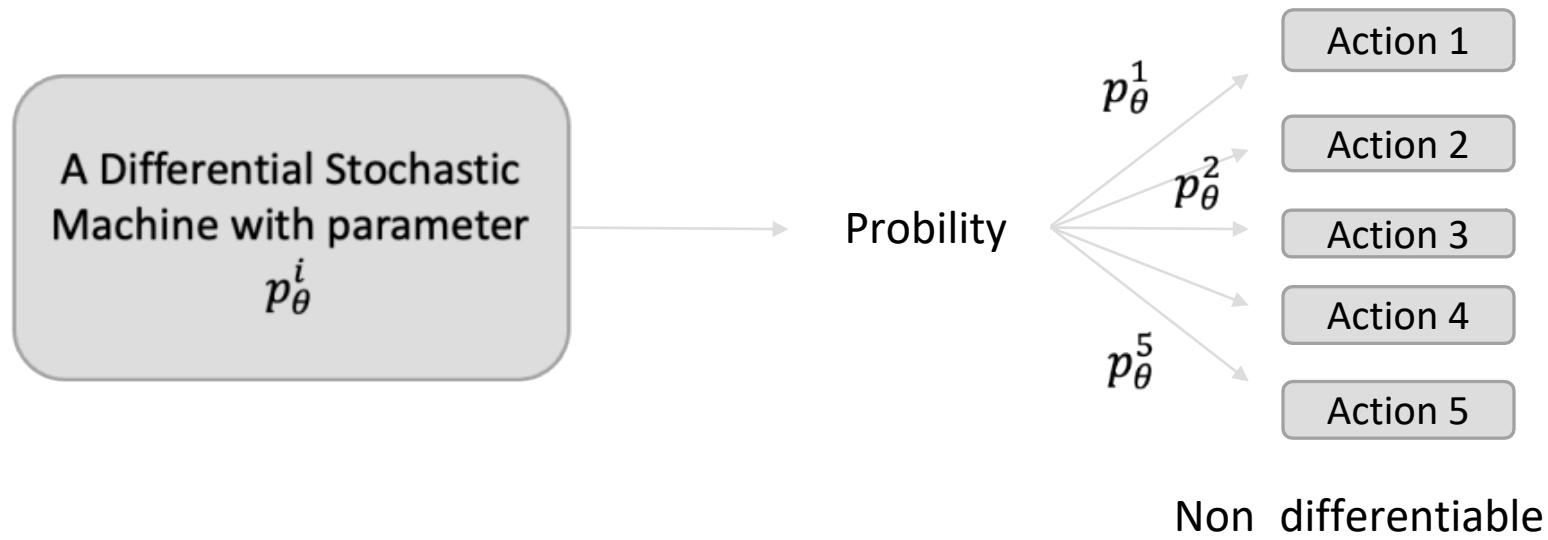
Idea from Granger Causality

$$\begin{aligned} L(\Theta, \Phi) &= \mathbb{E}_{\hat{A} \sim B(\Theta)} \left(- \sum_{t=1}^T \log P(\mathbf{x}^{t+1} | \mathbf{x}^t, \hat{A}, \Phi) \right) + \lambda \sum_{ij} |\theta_{ij}| \\ &\approx -\frac{1}{K} \sum_{k=1}^K \sum_{t=1}^T \sum_{i=1}^N L_i(\mathbf{x}_i^{t+1} | \mathbf{x}^t, \hat{A}_{\cdot,i}^k) + \lambda \sum_{ij} |\theta_{ij}|, \end{aligned}$$

where,

$$L_i(\mathbf{x}_i^{t+1} | \mathbf{x}^t, \hat{A}_{\cdot,i}^k) = \log \hat{f}_i(\mathbf{x}_i^{t+1} | \mathbf{x}^t \odot \hat{A}_{\cdot,i}^k; \phi_i)$$

Problem



Gumbel softmax

$$f(z) = f(y)$$

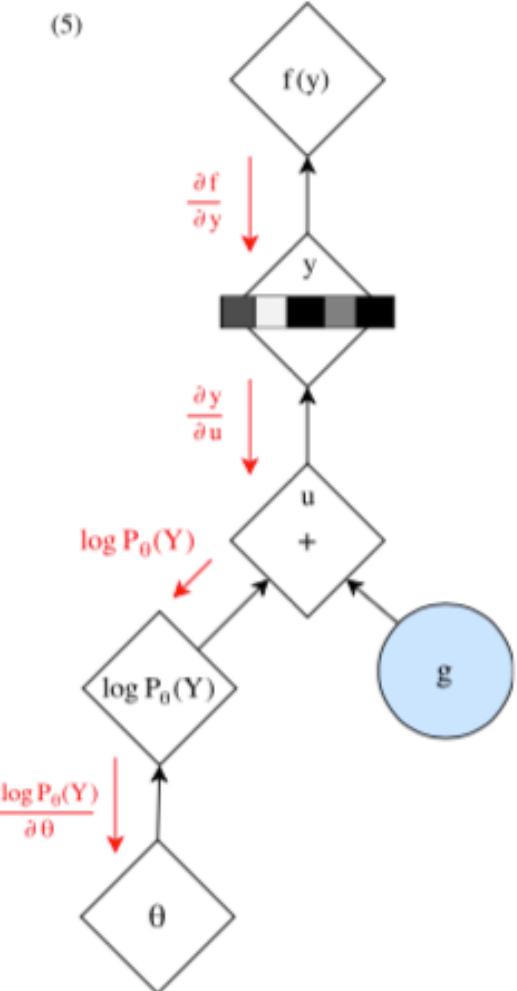
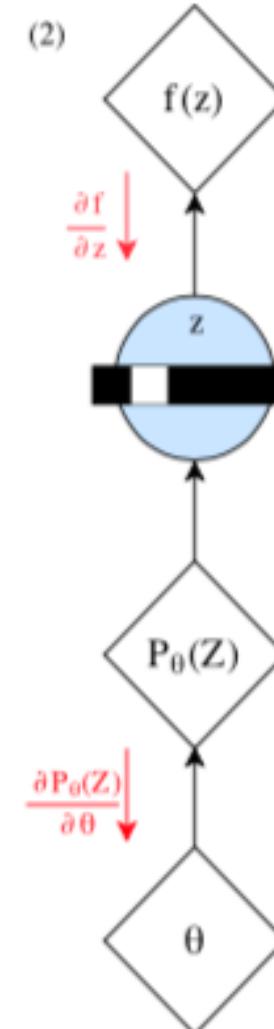
$$y = \text{softmax}(\log P_\theta(Y) + g) = \frac{\exp(\frac{\log p_{\theta^i} + g^i}{\tau})}{\sum_i \exp(\frac{\log p_{\theta^i} + g^i}{\tau})}$$

Z is a categorical variable (1,2,...,N) with probability

$$P_\theta(Y) = (p_\theta^1, p_\theta^2, \dots, p_\theta^n)$$

$g = (g^1, g^2, \dots, g^n)$ are iid for given distribution

gumbel distribution (0,1)



Network Generator

$$\begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1N} \\ \vdots & \alpha_{ij} & \vdots \\ \alpha_{N1} & \cdots & \alpha_{NN} \end{pmatrix} \rightarrow \begin{pmatrix} \exp\left(\frac{\log \alpha_{ij} + g_{ij}}{\tau}\right) \\ \exp\left(\frac{\log \alpha_{ij} + g_{ij}}{\tau}\right) + \exp\left(\frac{\log(1 - \alpha_{ij}) + g_{ij}}{\tau}\right) \end{pmatrix}_{N \times N}$$

↓

$$\begin{pmatrix} 0 & \cdots & 1 \\ \vdots & a_{ij} & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$

Network Generator

Legend

• Vector and Matrix Product

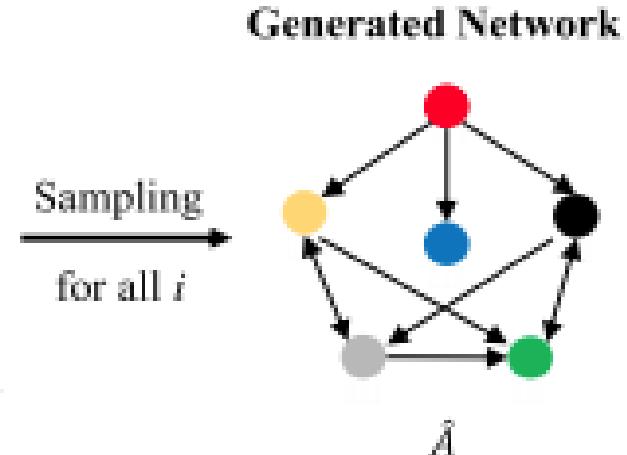
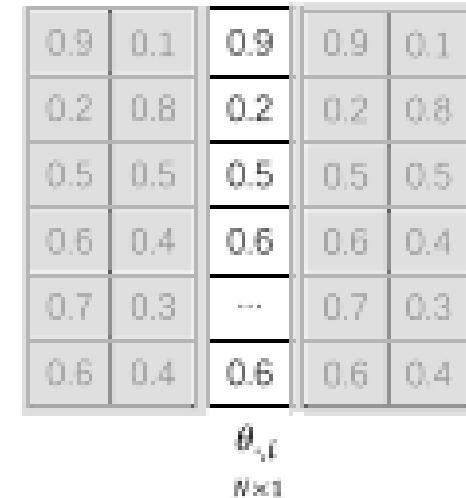
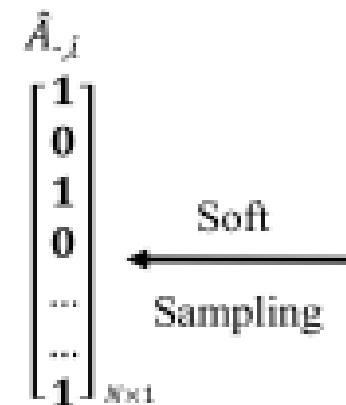
◎ Vectors Concatenation

○ Mapping Vectors to Log(Probability)

Input: x_i^t

$2.1, 0.3, \dots, 0.5$

$1 \times D$



Input: x^t

$0.1, 0.2, \dots, 0.5$

$2.1, 0.3, \dots, 0.5$

$3.3, 1.1, \dots, 0.5$

\dots

$1.1, 0.3, \dots, 1.1$

$N \times D$

$1 \times D$

$NN^{(1)}$

First step

$$\begin{bmatrix} 2.1, \dots, 3.1 \\ 4.3, \dots, 2.4 \\ 3.1, \dots, 5.5 \\ \vdots \\ -4, \dots, 1.8 \end{bmatrix} \xrightarrow{\circlearrowleft} \begin{bmatrix} 1.1 \\ 0.3 \\ 0.1 \\ \vdots \\ 0.4 \end{bmatrix}^T$$

$1 \times F$

$$\begin{bmatrix} 1.2 \\ -3 \\ 3.5 \\ \vdots \\ 0.8 \end{bmatrix}^T \xrightarrow{\circlearrowleft} \begin{bmatrix} 1.1 \\ 2.3 \\ 3.1 \\ \vdots \\ 0.1 \end{bmatrix}^T$$

$$h_i^{(2)} \quad 1 \times F$$

$NN^{(2)}$

Second step

$$\begin{bmatrix} 0.1 \\ 2.7 \\ 3.3 \\ \vdots \\ 0.9 \end{bmatrix}^T \xrightarrow{\circlearrowleft} \begin{bmatrix} 1.8 \end{bmatrix}^T$$

$$h_i^{(3)} \quad 1 \times D$$

$NN^{(3)}$

Third step

$$L_i \quad 1 \times 1$$

$$2.4, 0.9, \dots, 0.7$$

Data: x_i^{t+1}

Dynamics Learner

ARNI(Algorithm for Revealing Network Interactions)



ARTICLE

DOI: 10.1038/s41467-017-02288-4

OPEN

Model-free inference of direct network interactions from nonlinear collective dynamics

Jose Casadiego^{1,2}, Mor Nitzan^{3,4,5}, Sarah Hallerberg^{2,6} & Marc Timme^{1,2,7,8}

The topology of interactions in network dynamical systems fundamentally underlies their function. Accelerating technological progress creates massively available data about collective nonlinear dynamics in physical, biological, and technological systems. Detecting direct interaction patterns from those dynamics still constitutes a major open problem. In particular, current nonlinear dynamics approaches mostly require to know a priori a model of the (often high dimensional) system dynamics. Here we develop a model-independent framework for inferring direct interactions solely from recording the nonlinear collective dynamics generated. Introducing an explicit dependency matrix in combination with a block-orthogonal regression algorithm, the approach works reliably across many dynamical regimes, including transient dynamics toward steady states, periodic and non-periodic dynamics, and chaos. Together with its capabilities to reveal network (two point) as well as hypernetwork (e.g., three point) interactions, this framework may thus open up nonlinear dynamics options of inferring direct interaction patterns across systems where no model is known.

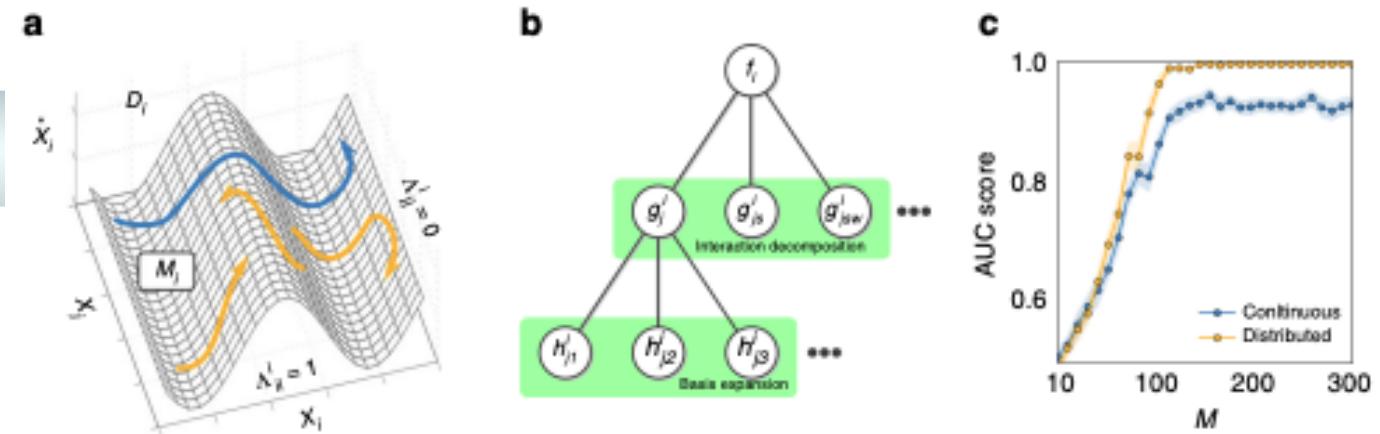
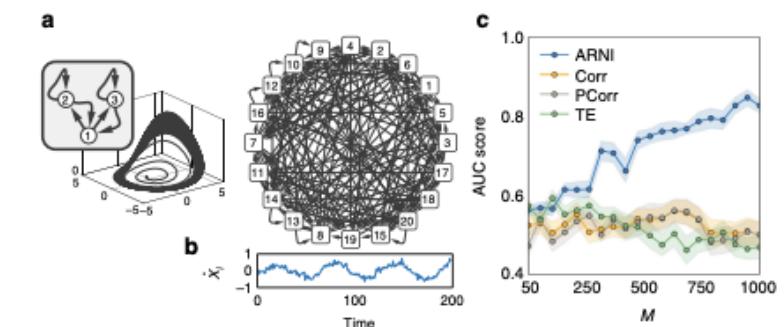


Table 1 Interactions may be represented in different basis functions

Index	Basis function
a	$h_{j,p}^i(x_i, x_j) = (x_j - x_i)^p$
b	$h_{j,p_1 p_2}^i(x_i, x_j) = x_i^{p_1} x_j^{p_2}$
c	$h_{j,p}^i(x_i, x_j) = \sin(p(x_j - x_i))$ and $h_{j,p}^i(x_i, x_j) = \cos(p(x_j - x_i))$
d	$h_{j,p}^i(x_i, x_j) = 1 + \ x_j - x_{i,p}\ ^2$, $x_{i,j} = (x_i, x_j)$ and $x_{i,p} = (x_{i,p}, x_{j,p})$
e	$h_{j,p}^i(x_i) = x_i^p$
f	$h_{j,p}^i(x_i) = \sin(px_i)$ and $h_{j,p}^i(x_i) = \cos(px_i)$

Different basis functions can be employed to represent a specific interaction. Basis functions of correct order are more likely to reveal true interactions than functions of incorrect order (see also Supplementary Note 6). Fig. 5, Basis function d belongs to a class of radial basis functions, so $x_{i,p}$ represent the p -th center of the expansion⁵⁶.



<https://www.nature.com/articles/s41467-017-02288-4>



NRI(Neural Relational Inference)

Neural Relational Inference for Interacting Systems

Thomas Kipf^{*1} Ethan Fetaya^{*23} Kuan-Chieh Wang²³ Max Welling¹⁴ Richard Zemel²³⁴

Abstract

Interacting systems are prevalent in nature, from dynamical systems in physics to complex societal dynamics. The interplay of components can give rise to complex behavior, which can often be explained using a simple model of the system's constituent parts. In this work, we introduce the neural relational inference (NRI) model: an unsupervised model that learns to infer interactions while simultaneously learning the dynamics purely from observational data. Our model takes the form of a variational auto-encoder, in which the latent code represents the underlying interaction graph and the reconstruction is based on graph neural networks. In experiments on simulated physical systems, we show that our NRI model can accurately recover ground-truth interactions in an unsupervised manner. We further demonstrate that we can find an interpretable structure and predict complex dynamics in real motion capture and sports tracking data.

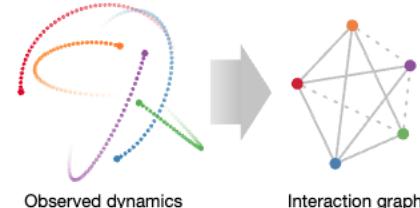
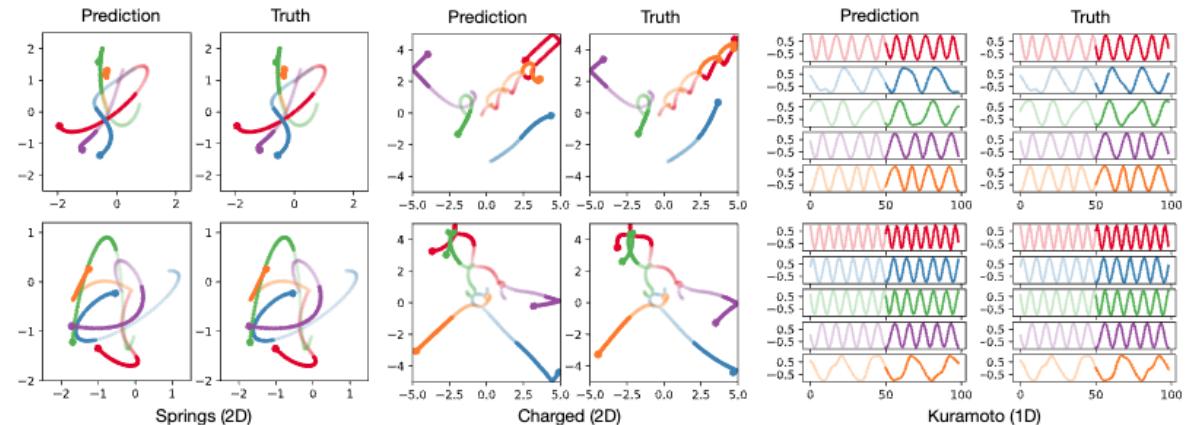
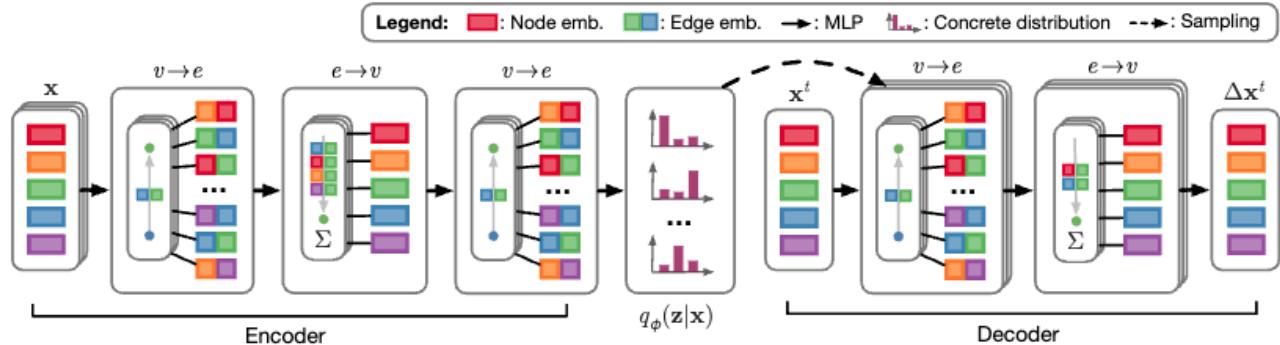


Figure 1. Physical simulation of 2D particles coupled by invisible springs (left) according to a latent interaction graph (right). In this example, solid lines between two particle nodes denote connections via springs whereas dashed lines denote the absence of a coupling. In general, multiple, directed edge types – each with a different associated relation – are possible.

able to reason about the different types of interactions that might arise, e.g. defending a player or setting a screen for a teammate. It might be feasible, though tedious, to manually annotate certain interactions given a task of interest. It is more promising to learn the underlying interactions, perhaps shared across many tasks, in an unsupervised fashion.



Comparisons

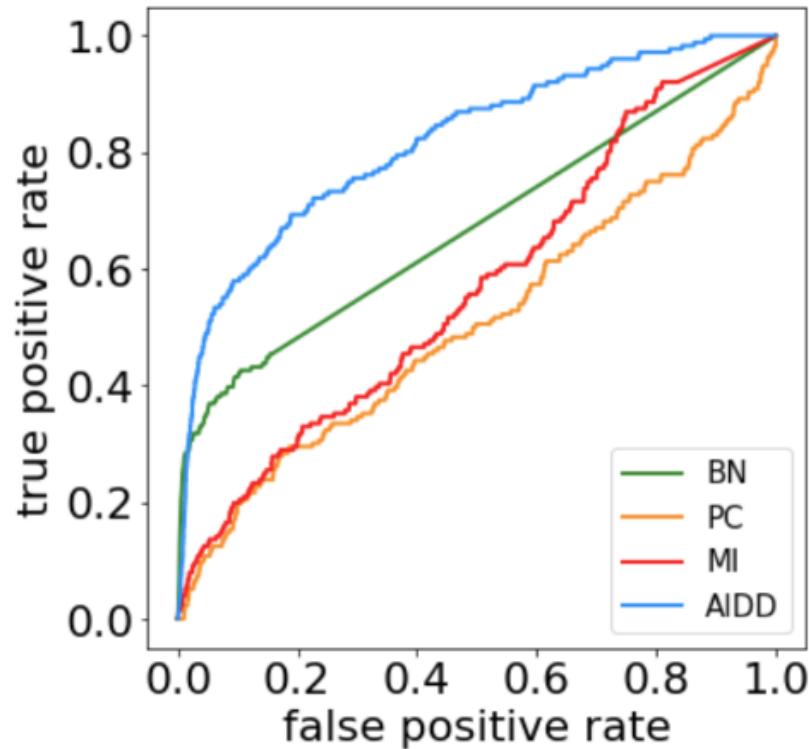
Type	Model	Network	ARNI		MI	PC	NRI		LSTM	OURS	
			AUC	MSE	AUC	AUC	MSE/ACC	AUC	MSE/ACC	MSE/ACC	AUC
Con.	Spring	ER-10	0.5853	1.33E-03	0.7500	0.8250	2.60E-08	0.9998*	2.98E-04	2.70E-04*	1.0
		WS-10	0.5125	1.58E-03	0.6875	0.7875	8.40E-08	0.9997*	3.35E-04	3.31E-04*	1.0
		BA-10	0.5169	1.10E-03	0.6422	0.6571	7.00E-10	0.9999*	2.14E-04	2.90E-05*	1.0
		ER-2000	-	-	0.4997	-	-	-	2.25E-03	1.18E-05	0.9886
		WS-2000	-	-	0.5002	-	-	-	5.89E-03	8.51E-06	0.9933
		BA-2000	-	-	0.5010	-	-	-	4.54E-03	2.09E-03	0.9523
	SIR	SIR-371(D)	0.5424*	8.25E-03	0.5027	0.5119	-	-	2.28E-03*	2.98E-05	0.9156
	Menten	Gene-100(D)	1.0	9.71E-03	0.5416	0.6574	-	-	2.29E-03*	4.37E-05	0.9960*
	Dis.	ER-10	1.0	2.33E-09	0.5745	0.7804	1.40E-05	0.8850	2.60E-04	5.60E-06*	1.0
		WS-10	1.0	2.35E-09	0.6875	0.8375	9.40E-06	0.9331	2.40E-04	2.80E-06*	1.0
		BA-10	1.0	2.40E-09	0.4390	0.7439	1.30E-05	0.6753	9.21E-05	6.90E-06*	1.0
		ER-200	0.8441*	4.17E-02	0.5774	0.7648	-	-	5.91E-05	2.04E-06	0.9987
		WS-200	1.0	2.36E-09	0.6969	0.7506	-	-	1.63E-04	1.95E-06	0.9987*
		BA-200	0.8840*	2.45E-02	0.5533	0.7493	-	-	1.46E-04	2.57E-06	0.9874
		WS-1000	-	-	0.5670	-	-	-	3.54E-05	2.92E-06	0.9795
		BA-1000	-	-	0.5290	-	-	-	3.46E-05	5.48E-05	0.9105
Bin.	Voter	ER-10	-	-	-	0.4552	0.8447*	0.5000	0.5242	0.9647	1.0
		WS-10	-	-	-	0.5250	0.9062*	0.5037	0.6007	0.9463	1.0
		BA-10	-	-	-	0.4607	0.9588*	0.4999	0.6917	0.9866	1.0
		WS-1000	-	-	0.5470*	-	-	-	0.5317	0.6650	0.9996
		BA-1000	-	-	0.5030	-	-	-	0.5208*	0.6758	0.9942
		EMAIL-1133	-	-	0.4999	-	-	-	0.5333*	0.7212	0.9576
		ROAD-1174	-	-	0.5004	-	-	-	0.5455*	0.8942	0.9996
		DORM-217(D)	-	-	0.5219	-	-	-	0.5735*	0.6951	0.9901
		BLOG-1224(D)	-	-	0.4995	-	-	-	0.5295*	0.6793	0.8603

<https://journals.aps.org/pre/abstract/10.1103/PhysRevE.106.034315>

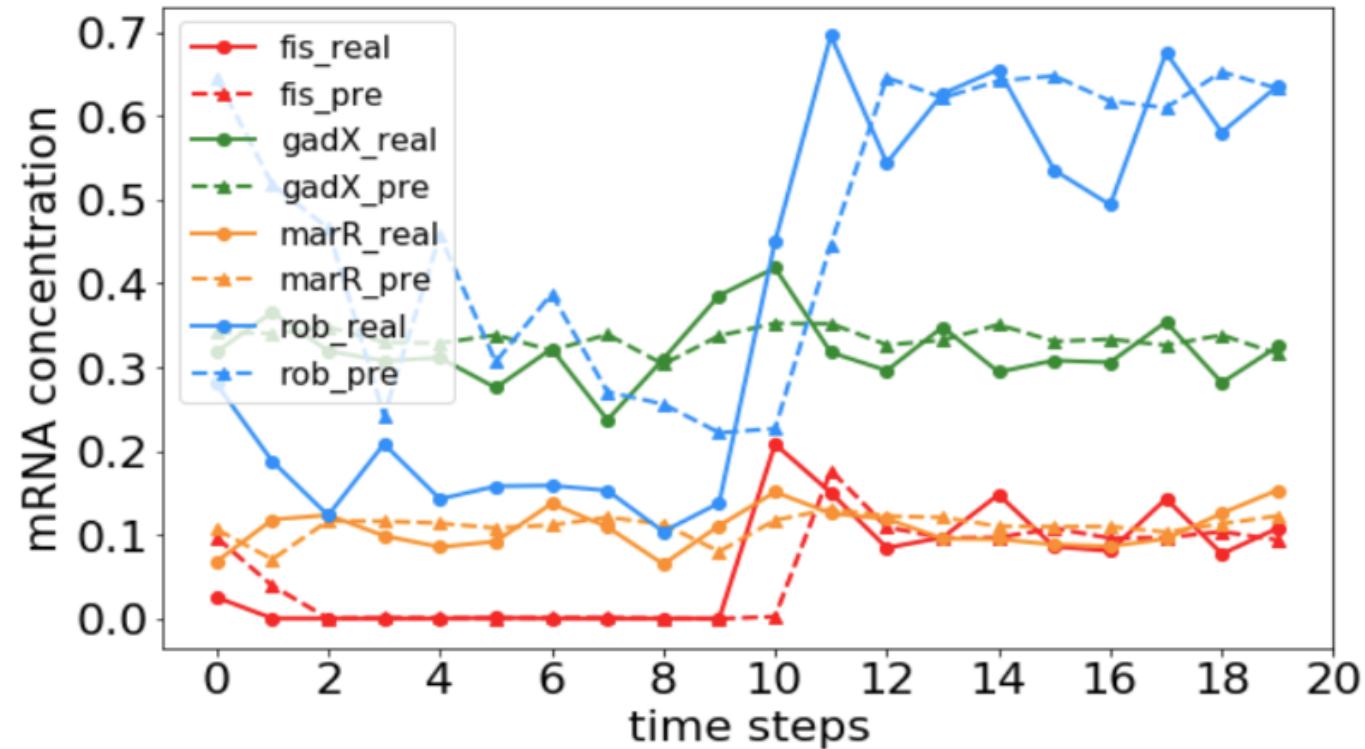


Gene Network Reconstruction

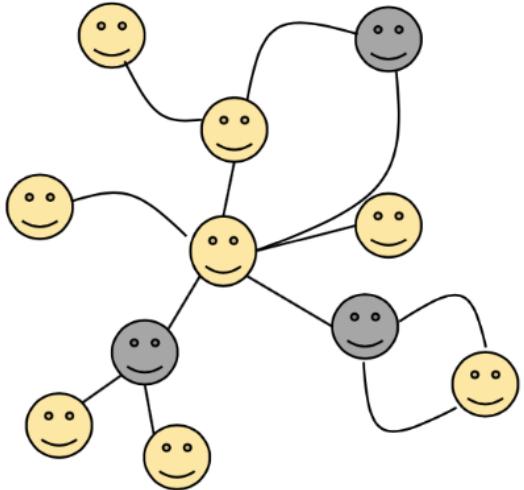
a



b



Missing Nodes



b

