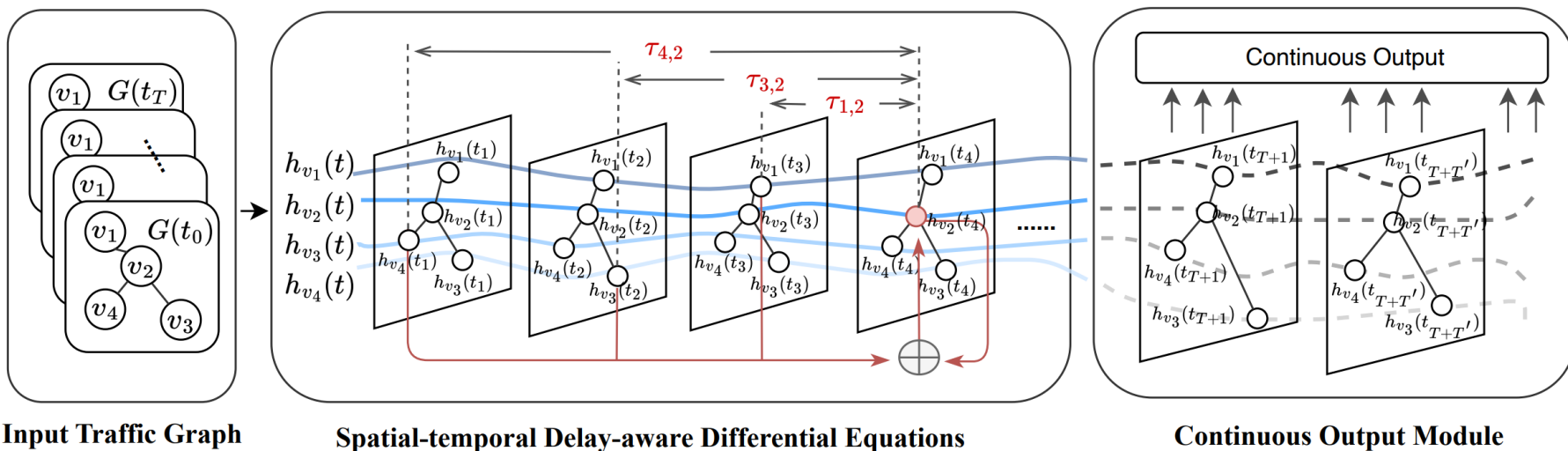
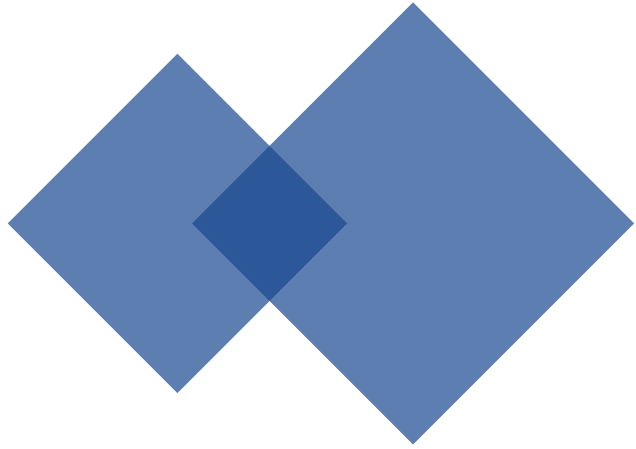


STDDE

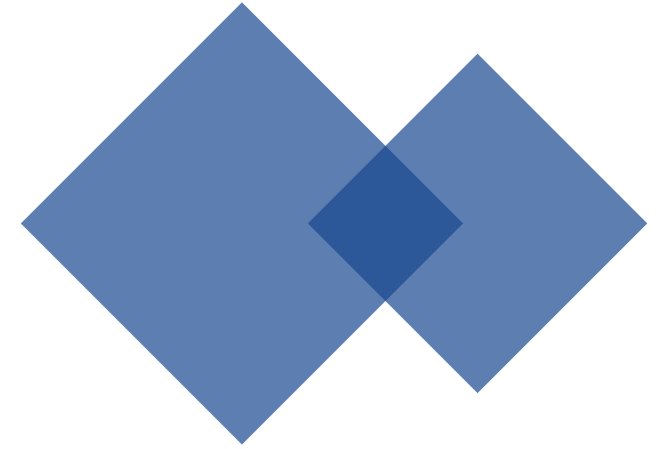
Unveiling Delay Effects in Traffic
Forecasting: A Perspective from
Spatial-**T**emporal **D**elay
Differential **E**quations





STDDE

Unveiling Delay Effects in Traffic
Forecasting: A Perspective from
Spatial-**T**emporal **D**elay
Differential **E**quations



24.4.23

Presented by Yyyq

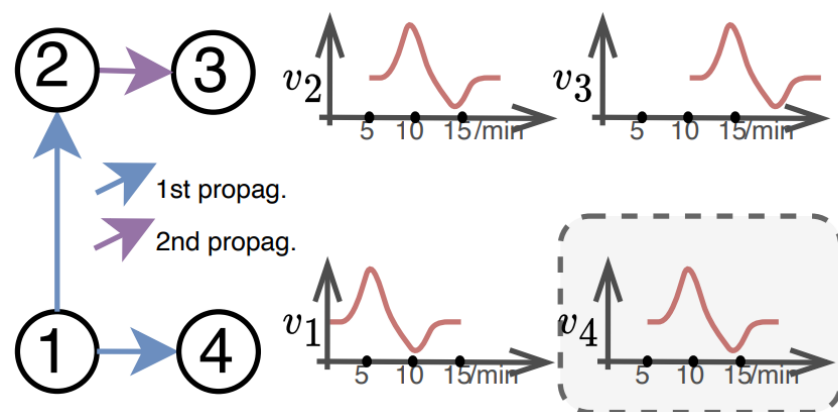
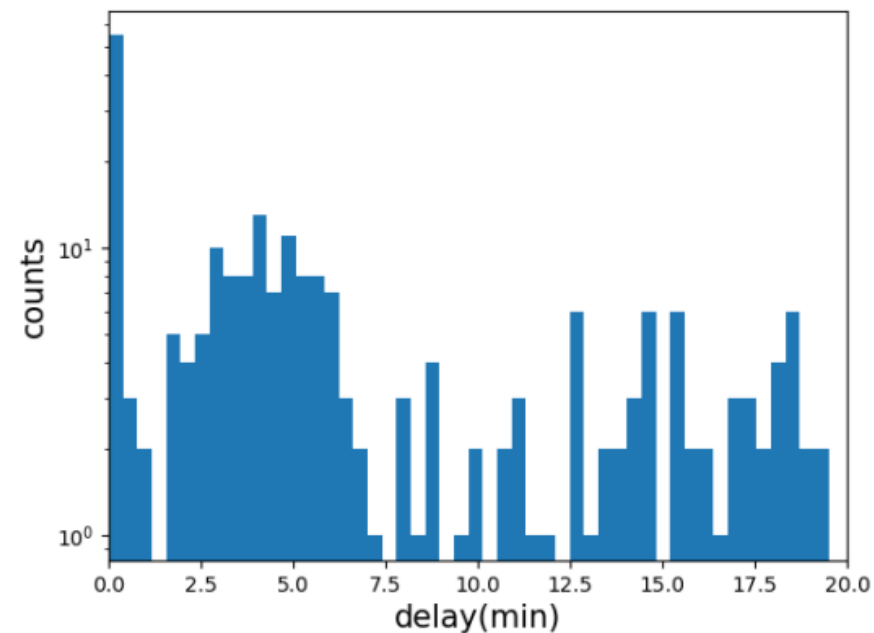
01

问题描述

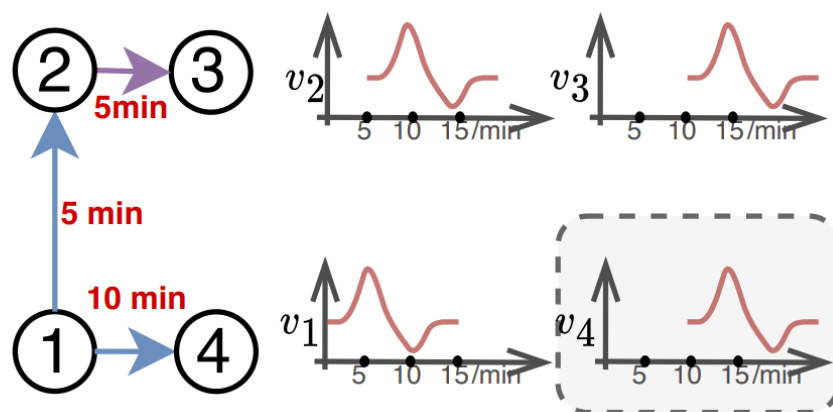
➤ 图信号传播过程中的延迟

➤ 交通系统固有的连续性

(传感器采样10分钟间隔, 希望更细粒度的预测精度)



(a) General GNN propagation process.



(b) Realistic propagation process.



➤ **时空延迟微分方程模型 STDDE**

将延迟效应和连续性纳入统一的框架

➤ **可学习的交通图时延估计器**

显式捕获并利用相邻节点之间延迟的空间交互

➤ **连续输出模块**

准确预测不同频率的交通流量



神经普通微分方程 (NODE) → 神经控制微分方程 (NCDE) → 神经延迟微分方程 (NDDE)

假设我们有一个动态系统，其演化由一个连续时间的微分方程描述。这个系统的输入是一个离散的时间序列数据，我们希望使用神经网络来对这个系统进行建模。

➤ 神经普通微分方程 (NODE)

- ResNet的连续版本，描述神经网络的动态演变过程
- 它的解完全由初始条件确定，后续到达的数据不会对方程产生影响
- 通常需要与RNN结合使用，以处理动态输入数据

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



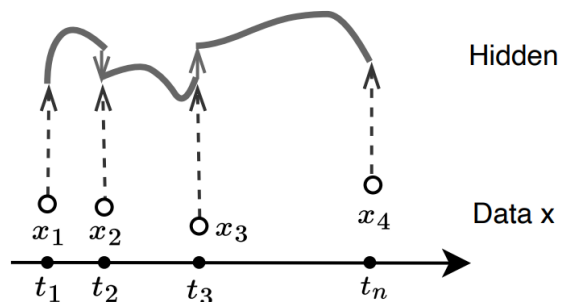
神经普通微分方程 (NODE) → 神经控制微分方程 (NCDE) → 神经延迟微分方程 (NDDE)

假设我们有一个动态系统，其演化由一个连续时间的微分方程描述。这个系统的输入是一个离散的时间序列数据，我们希望使用神经网络来对这个系统进行建模。

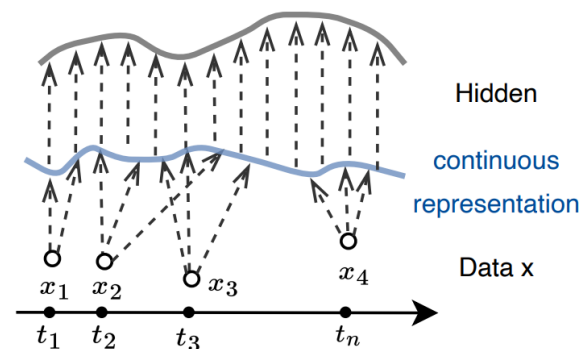
➤ 神经控制微分方程 (NCDE)

- 根据控制信号调整模型的演化轨迹
- 灵活地适应输入数据的变化

$$\frac{dy(t)}{dt} = f(y(t), u(t), t; \theta)$$



(a) ODE-RNN



(b) CDE



神经普通微分方程 (NODE) → 神经控制微分方程 (NCDE) → 神经延迟微分方程 (NDDE)

假设我们有一个动态系统，其演化由一个连续时间的微分方程描述。这个系统的输入是一个离散的时间序列数据，我们希望使用神经网络来对这个系统进行建模。

➤ **神经延迟微分方程 (NDDE)**

- 引入延迟项来处理具有时间延迟效应的系统

$$\frac{d\mathbf{y}(t)}{dt} = f(\mathbf{y}(t), \mathbf{y}(t - \tau), t; \theta)$$

➤ **本文将NDDE扩展到多变量条件下进行时空建模，并配合NCDE构建连续的交通信号演化**

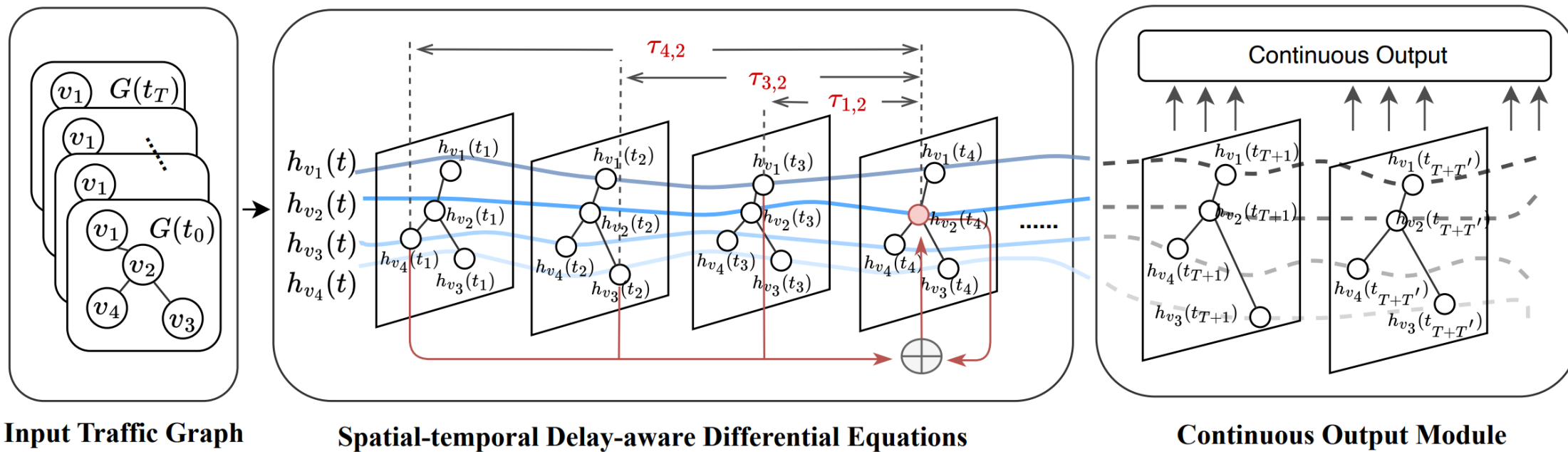


➤ 预测问题

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

➤ 生成不同粒度的预测 (t+任意正数)

$$\left[X_{t-T+1,:}, X_{t-T+2,:}, \cdots, X_{t,:}; \mathcal{G} \right] \xrightarrow[infer]{f} \left[X_{t+dt_1,:}, X_{t+dt_2,:}, \cdots, X_{t+dt_n,:} \right].$$





➤ **延迟感知神经微分方程框架：**系统的整体演化过程

- $\phi(t)$ 表示历史函数
- $t < 0$ 表示过去的时间点， τ 表示延迟值

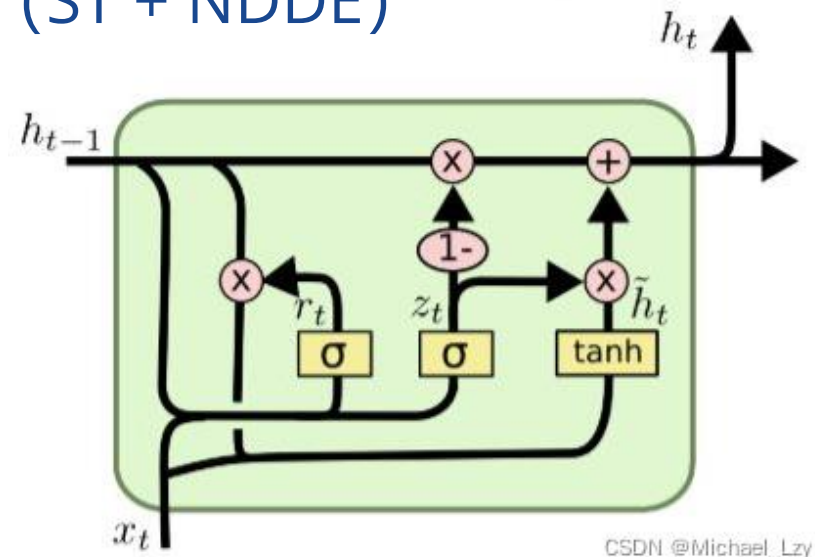
$$h(t) = \begin{cases} \phi(t), & t \leq 0, \\ h(0) + \int_0^T f(h(t), h(t - \tau), t, \theta) dt, & t > 0. \end{cases}$$

➤ 用GRU来推导微分方程：系统的瞬时变化率

$$z_t = \sigma(W_z h_{t-1} + U_z g_t + b_z),$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t,$$

$\begin{cases} h_t: \text{隐藏状态} \\ z_t: \text{更新门} \\ g_t: \text{更新向量 (当前时间步的输入)} \end{cases}$



CSDN @Michael_Lzy

1. 更新门 z_t 的计算:

$$z_t = \sigma(W_z h_{t-1} + U_z g_t + b_z)$$

2. 重置门 r_t 的计算:

~~$$r_t = \sigma(W_r h_{t-1} + U_r g_t + b_r)$$~~

3. 候选更新 \tilde{h}_t 的计算:

~~$$\tilde{h}_t = \tanh(W_h (r_t \odot h_{t-1}) + U_h g_t + b_h)$$~~

4. 更新隐藏状态 h_t :

~~$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$~~
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t$$

- 用GRU来推导微分方程：系统的瞬时变化率

$$z_t = \sigma(W_z h_{t-1} + U_z g_t + b_z),$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t - h_{t-1}$$

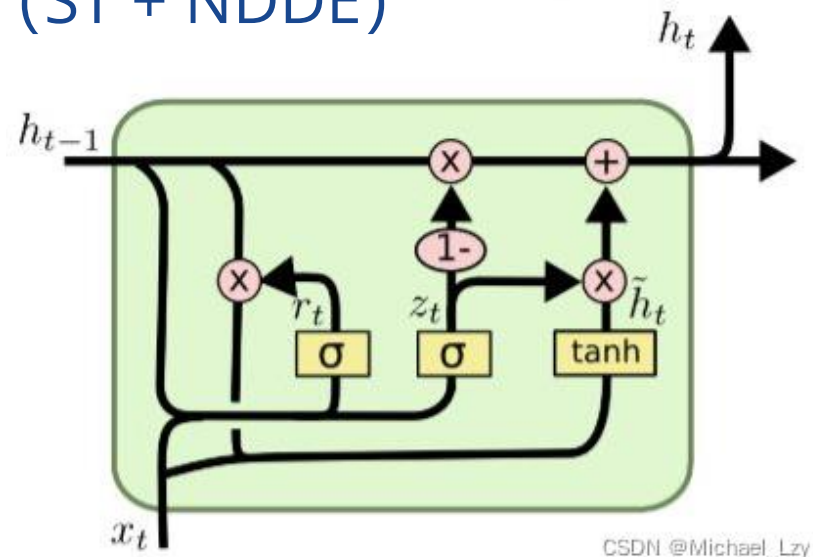
$$\begin{cases} h_t: \text{隐藏状态} \\ z_t: \text{更新门} \\ g_t: \text{更新向量 (当前时间步的输入)} \end{cases}$$



$$\Delta h = h_t - h_{t-1} = (1 - z_t) \odot (g_t - h_{t-1}).$$

- 用微分来近似隐藏状态的变化率：用增量 Δh 近似切线斜率

$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t)).$$





$$h(t) = \begin{cases} \phi(t), & t \leq 0, \\ h(0) + \int_0^T f(h(t), h(t - \tau), t, \theta) dt, & t > 0. \end{cases}$$

$$\begin{aligned} z_t &= \sigma(W_z h_{t-1} + U_z g_t + b_z), \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot g_t, \\ \Delta h &= h_t - h_{t-1} = (1 - z_t) \odot (g_t - h_{t-1}). \end{aligned}$$

***注：**

- 假设隐藏状态 $h(t)$ 在相邻时间点 Δt 和 $t + \Delta t$ 之间的变化是连续的
- 这个微分方程并没有直接体现延迟值。在GRU模型中，通过模型中的历史隐藏状态来间接地考虑延迟值。
- 微分方程描述了系统的瞬时变化率，而积分式子描述了系统的整体演化过程。



$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t)).$$



$$h(t) = \begin{cases} \phi(t), & t \leq 0, \\ h(0) + \int_0^T f(h(t), h(t-\tau), t, \theta) dt, & t > 0. \end{cases}$$

➤ 设置历史函数为时间常数函数：MLP

$$\phi(t) = \text{constant} = \text{MLP}(x), \quad t \leq 0$$

$$\begin{aligned} z_t &= \sigma(W_z h_{t-1} + U_z g_t + b_z), \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot g_t, \\ \Delta h &= h_t - h_{t-1} = (1 - z_t) \odot (g_t - h_{t-1}). \end{aligned}$$



$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t)).$$



➤ 将时空延迟相关性纳入NDDE

- 利用图神经网络提取空间特征并将其视为更新向量

节点*i*和*j*之间边的权重

$$\underline{g_i(t)} = c \sum_{\underline{j \in \mathcal{N}(i)}} \boxed{\alpha_{ij}} \underline{f(h_j(t - \tau_{ij}))},$$

节点*i*在时间*t*的
更新向量

$\underline{j \in \mathcal{N}(i)}$
节点*i*的邻居集合

节点*j*在时间*t* - τ_{ij} 的隐藏状态，
考虑延迟值 τ_{ij}



➤ **1、最大化交叉相关 (MCC)：** 静态，预处理步骤

- 寻找两个时间序列经过偏移后交叉相关的峰值来确定延迟
- k表示k步偏移，相关函数使用person相关系数

$$\tau_{ij} = \arg \max_k \text{corr}(\tilde{x}_i^{\rightarrow k}, \tilde{x}_j),$$



➤ 2、将时间延迟建模为可学习的模式：连续建模， τ 可学习

- 为每条边分配两个可学习的延迟参数：一个用于高峰时段，另一个用于非高峰时段。

$$g_i(t) = c \sum_{j \in \mathcal{N}(i)} \alpha_{ij} f(h_j(t - \tau_{ij})),$$

注：

- 通常情况下， τ 被认为是不可学习的
- 但是由于我们采用了连续建模的方式，我们得到了 h 关于 t 的导数
- 因此我们可以通过定义 h 关于 τ 的反向计算，来将 τ 的梯度纳入神经网络中



➤ **NCDE**：考虑外部的新输入数据

- 从原始输入数据中通过自然三次样条插值方法生成连续表示 $X(t)$

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

$$X(t) = \Phi \left(\{x^0, t_0\}, \{x^1, t_1\}, \dots, \{x^n, t_n\} \right) \quad \left\{ \begin{array}{l} \varphi \text{ 是自然三次样条函数} \\ x_i \text{ 是时刻 } t_i \text{ 的输入数据} \end{array} \right.$$

- 将这个连续表示应用于动态演化方程

$$\frac{dh_i(t)}{dt} = (1 - z_i(t)) \odot (g_i(t) - h_i(t)) \tilde{f} \left(\frac{dX_i(t)}{dt} \right)$$

表示交通流量的趋势和波动，
不断地影响动态演化的方向



➤ 使用另一个STDDE来生成最终输出

- 将输入过程中隐藏流的最后阶段视为输出过程的历史函数
- 隐藏状态在流形空间内保持连续，确保了输入和输出过程之间的一致性
- 预测不同频率的交通流量

$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t))$$

$$h_i(t') = h_i(T) + \int_T^{t'} \frac{dh_i(t)}{dt} dt,$$

$$y_i(t') = f(h_i(t')),$$

$$Y_i = [y_i(t_{T+1}), y_i(t_{T+2}), \dots, y_i(t_{T+T'})]$$



➤ **STDDE的泛化能力**

- GRU 可以被看作是 STDDE 的离散形式
- 所有的时延都设为零时，STDDE 会退化为一般的图神经网络（GNNs）

➤ **DDE的稳定性**

- 当平衡常数（ $c \leq 1/K$ ）的时候，DDE是渐进稳定的，上下界渐进稳定（ k 应该是GNN中的延迟值参数）

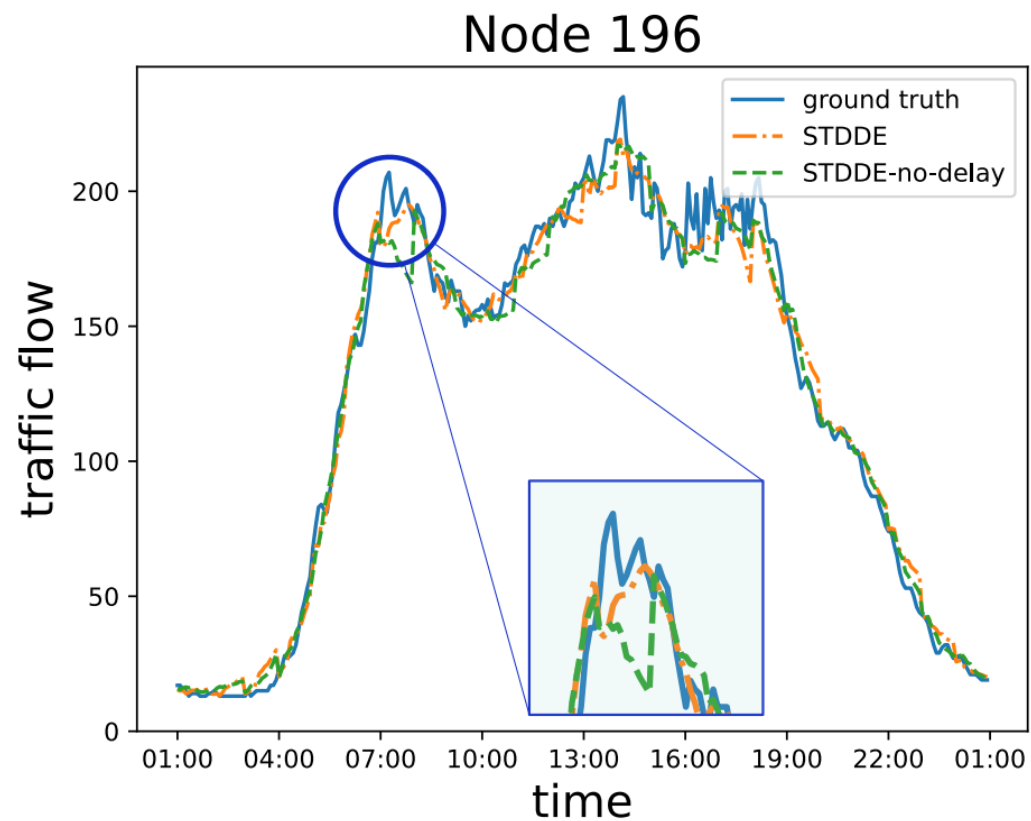


Datasets	#Sensors	#Edges	Time Steps
PeMSD7 (M)	228	1132	12672
PeMSD7 (L)	1026	10150	12672
PeMS03	358	547	26208
PeMS04	307	340	16992
PeMS07	883	866	28224
PeMS08	170	295	17856

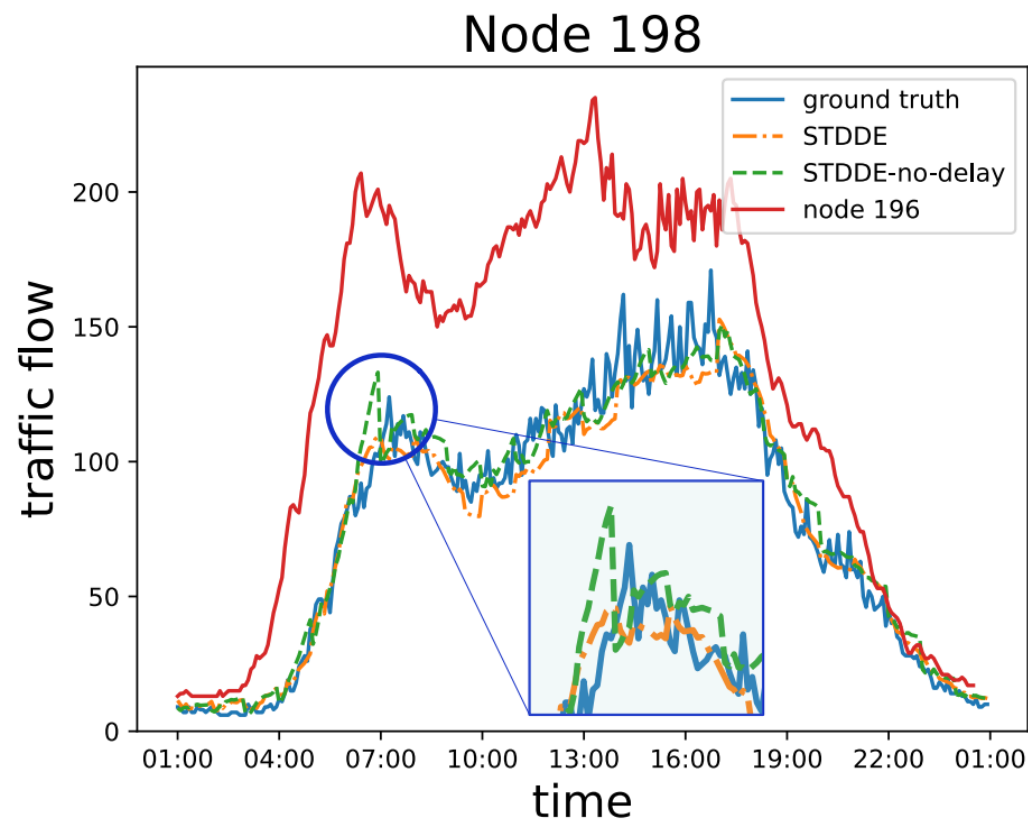


Data	Metric	STDDE-no-delay	STDDE	Gain
First 15%	RMSE	16.97	16.86	0.65%
	MAE	11.54	11.47	0.61%
	MAPE	19.71	18.37	6.80%
Last 15%	RMSE	37.72	34.59	8.30%
	MAE	25.06	23.24	7.26%
	MAPE	14.63	13.96	4.65%

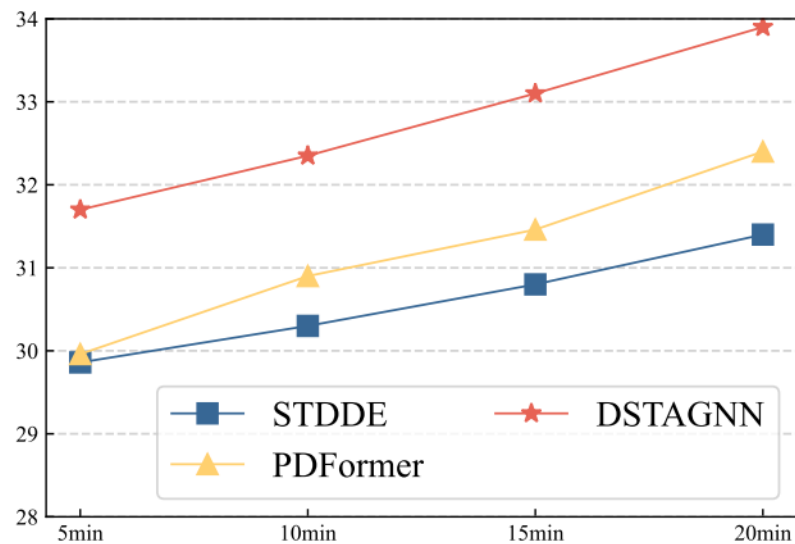
Table 3: Performances facing delays of different extent.



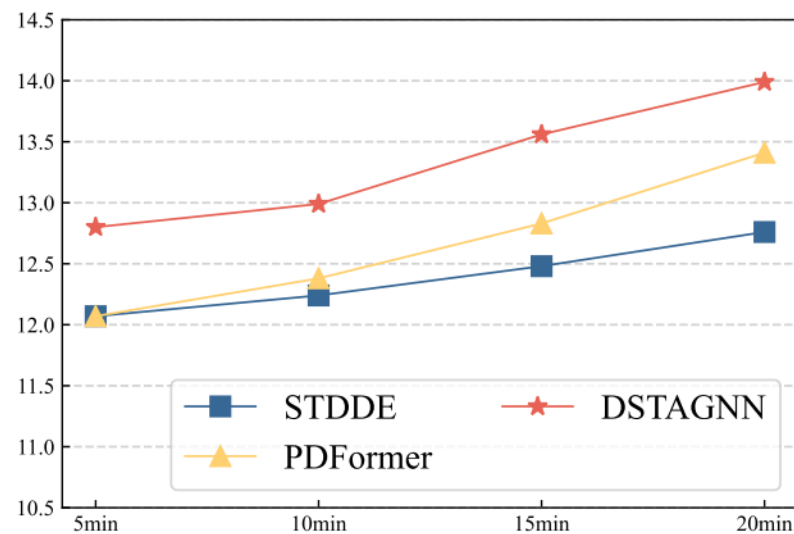
(a)



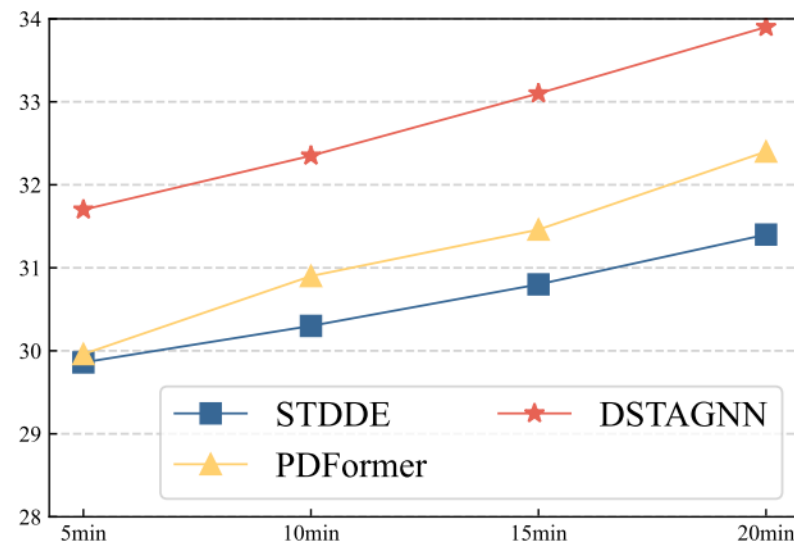
(b)



(a) RMSE



(b) MAE



(c) MAPE

Figure 5: Performance comparison with input time intervals greater than inference intervals.

在训练过程中，将时间间隔设置为10/15/20分钟，即输入步长为6/4/3；
在推理过程中，我们将时间间隔更改为5分钟



Dataset	Metric	STGCN	DCRNN	ASTGCN(r)	STSGCN	STGODE	AGCRN	STG-NCDE	DSTAGNN	FOGS	PDFormer	STDDE
PeMSD7(M)	RMSE	7.55	7.18	6.87	5.93	5.66	5.54	5.39	5.54	5.54	5.60	5.19
	MAE	4.01	3.83	3.61	3.01	2.97	2.79	2.68	2.78	2.76	2.81	2.56
	MAPE	9.67	9.81	8.84	7.55	7.36	7.02	6.76	6.93	6.83	7.06	6.61
PeMSD7(L)	RMSE	8.28	8.33	7.64	6.88	5.98	5.92	5.76	5.98	6.04	5.90	5.63
	MAE	4.84	4.33	4.09	3.61	3.22	2.99	2.87	2.98	2.96	2.92	2.77
	MAPE	11.76	11.41	10.25	9.13	7.94	7.59	7.31	7.50	7.48	7.54	7.26
PeMS03	RMSE	30.42	30.31	29.56	29.21	27.84	28.25	27.09	27.39	24.85	25.96	24.52
	MAE	17.55	17.99	17.34	17.48	16.50	15.98	15.57	15.62	15.06	14.95	15.03
	MAPE	17.43	18.34	17.21	16.78	16.69	15.23	15.06	14.74	15.03	15.58	14.69
PeMS04	RMSE	36.01	37.65	35.22	33.65	32.82	32.26	31.09	31.71	31.29	29.96	29.86
	MAE	22.66	24.63	22.94	21.19	20.84	19.83	19.21	19.38	19.44	18.31	18.11
	MAPE	14.34	17.01	16.43	13.90	13.77	12.97	12.76	12.77	12.81	12.07	12.07
PeMS07	RMSE	39.34	38.61	37.87	39.03	37.54	36.55	33.84	34.88	34.09	32.80	32.59
	MAE	25.33	25.22	24.01	24.26	22.99	22.37	20.53	21.62	20.79	19.78	19.47
	MAPE	11.21	11.82	10.73	10.21	10.14	9.12	8.80	9.24	8.75	8.54	8.49
PeMS08	RMSE	27.88	27.83	26.22	26.80	25.97	25.22	24.81	25.08	25.36	24.61	24.31
	MAE	18.11	17.46	16.64	17.13	16.81	15.95	15.45	15.85	16.10	15.66	15.12
	MAPE	11.34	11.39	10.6	10.96	10.62	10.09	9.92	9.93	9.85	9.61	9.74

Table 2: Performance comparison of baselines and proposed STDDE on six popular used real-world traffic datasets.

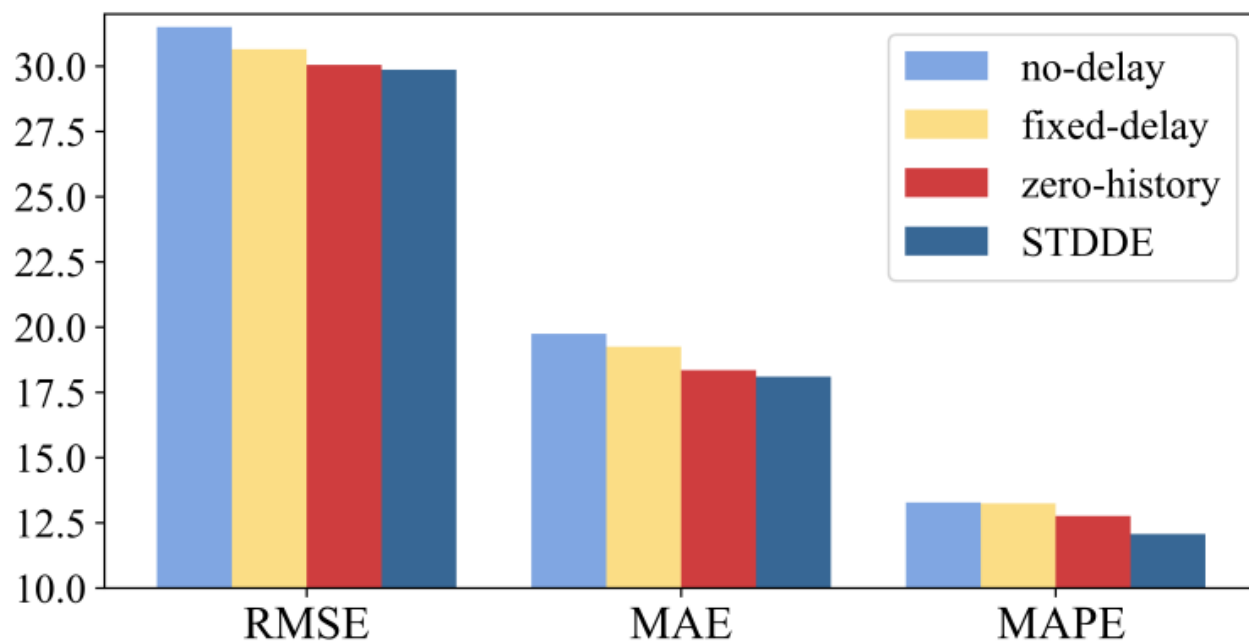


Figure 6: Ablation experiments of STDDE.



Model	# Parameters	PeMSD7 (M)		PeMSD7 (L)	
		Train	Infer	Train	Infer
STGODE	328,646	131	13	1107	146
FOGS	1,674,188	50	3	531	42
DSTAGNN	2,784,988	168	43	1222	209
PDFormer	531,165	120	11	1292	138
STDDE	175,830	82	9	734	84

Table 4: Comparison of # parameters and running time in one epoch. (Unit: seconds)

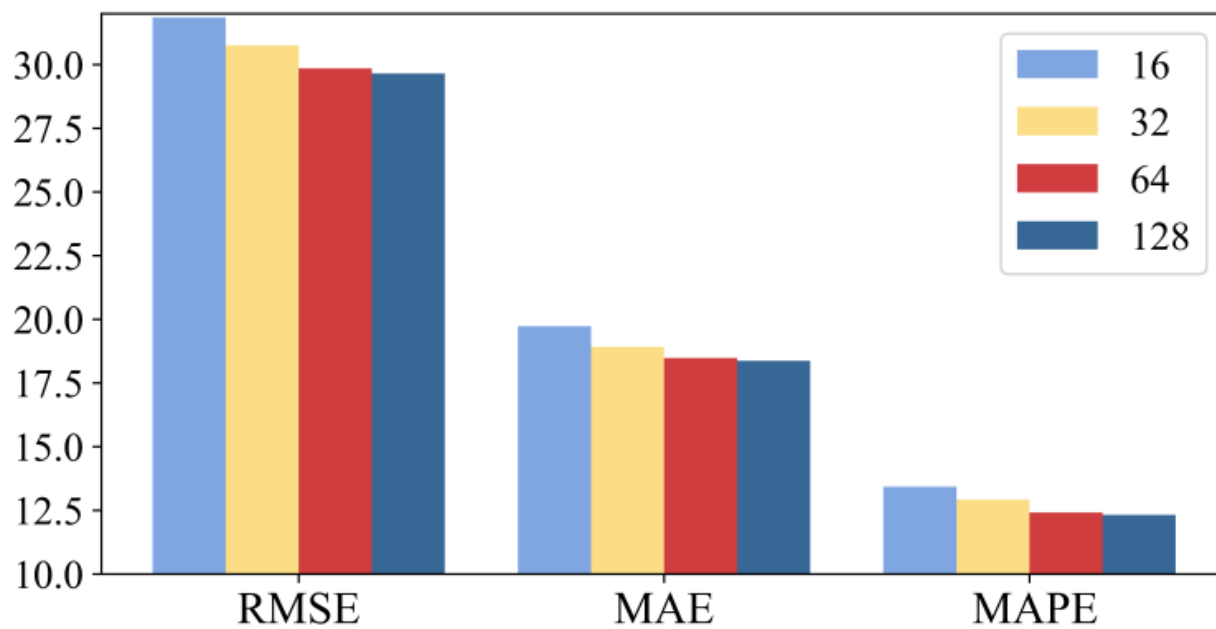


Figure 7: STDDE results with the change of hidden size.



谢谢观看

MANY THANKS !



24.4.23

