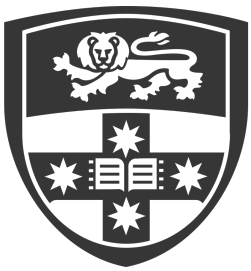


THE UNIVERSITY OF
SYDNEY

PHD THESIS

DISCIPLINE OF BUSINESS ANALYTICS

BEYOND TRADE-OFFS: ADVANCING SPATIAL-TEMPORAL FORECASTING IN TRANSPORTATION WITH DEEP LEARNING

Author: **Zhiqi Shao**

Supervisors: Prof. Junbin Gao

Prof. Andrey Vasnev

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

The University of Sydney

BUSINESS SCHOOL

2025

To my parents and husband.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Junbin Gao, for his invaluable guidance, unwavering support, and profound insight throughout the course of my research. His mentorship has been instrumental in shaping both the academic quality and the direction of this thesis. I am also sincerely thankful to my co-supervisor, Professor Andrey Vasnev, and Professor Michael Bell, for their constructive feedback. His support has been essential in refining the clarity and robustness of my contributions.

A special thank you to my husband, Ze Wang, for his unconditional love, constant encouragement, and patience during the many late nights and challenging times. Your belief in me has been my greatest source of strength.

I am also deeply grateful to my parents, whose endless love and support have made everything possible. Their sacrifices and faith in my journey have been the foundation of my achievements.

To all who have supported me in any way during this journey, thank you.

Publications and Research Works

This thesis is based on the following publications and research works:

- **Zhiqi Shao**, Michael G. H. Bell, Ze Wang, D Glenn Geers, Xusheng Yao, Junbin Gao. "CCDSReformer: Traffic-Flow Prediction with a Criss-Crossed Dual-Stream Enhanced Rectified Transformer Model." *Communications in Transportation Research*, 202, 104282, 2024.
- **Zhiqi Shao**, Ze Wang, Xusheng Yao, Michael G. H. Bell, Junbin Gao. "ST-MambaSync: Complementing the Power of Mamba and Transformer Fusion for Lower Computational Cost in Spatio-Temporal Traffic Forecasting." *Information Fusion*, 117: 102872, 2024.
- **Zhiqi Shao**, Haoning Xi, David Hensher, Ze Wang, Xiaolin Gong, Junbin Gao. "A spatial-temporal dynamic attention based Mamba model for multi-type passenger demand prediction in multimodal public transit systems." *Transpotrtaion Research Part E*,accepted, 2025.

The author has further contributed to the following publications, which are not included in the thesis:

- **Zhiqi Shao**, Haoning Xi, Haohui Lu, Ze Wang, Michael G. H. Bell, Junbin Gao. "Stllm-df: A spatial-temporal large language model with diffusion for enhanced multi-mode traffic system forecasting." *Transportation Research Part C*, 179, 105249, 2025.
- **Zhiqi Shao**, Xusheng Yao, Feng Chen, Ze Wang, Junbin Gao. "Revisiting Time-Varying Dynamics in Stock-Market Forecasting: A Multisource Sentiment-Analysis Approach with Large Language Models." *Decision Support Systems*, 190: 114362, 2024.

- **Zhiqi Shao**, Dai Shi, Andi Han, Andrey Vasnev, Yong Guo, Junbin Gao.
  "Enhancing Framelet GCNs with Generalised p-Laplacian Regularisation." *International Journal of Machine Learning and Cybernetics*, 15 (4): 1553–1573, 2024.
- Dai Shi[†], **Zhiqi Shao**[†], Yi Guo, Junbin Gao.
  "Frameless Graph Knowledge Distillation." *IEEE Transactions on Neural Networks and Learning Systems*, in press, 2024.
- Dai Shi[†], **Zhiqi Shao**[†], Yi Guo, Qibin Zhao, Junbin Gao.
  "Revisiting Generalized p-Laplacian Regularised Framelet GCNs: Convergence, Energy Dynamics, and Non-linear Diffusion." *Transactions on Machine Learning Research*, 2024.
- Haoning Xi[†], **Zhiqi Shao**[†], David A Hensher, John D Nelson, Huaming Chen, Kasun Wijayaratna.
  " A multi-task Transformer with mixture-of-experts for personalized periodic predictions of individual travel behavior in multimodal public transport." *Transportation Research Part C*, 179: 105287, 2025.
- Xusheng Yao, **Zhiqi Shao***, Ze Wang, Zhu Zhu, Zuanxu Chen, Qingyang Wu.
  "Policy Incentives and Market Mechanisms Dual-Driven Framework for New Energy Vehicle Promotion." *Energy Policy*, 199: 114530, 2024.
- Dai Shi, **Zhiqi Shao**, Andi Han, Yi Guo, Junbin Gao.
  "A New Perspective on the Expressive Equivalence Between Graph Convolution and Attention Models." *Asian Conference on Machine Learning*, pp. 1199–1214, 2023.

[†] Equal contribution. [*] Corresponding author.

Authorship Attribution Statement. The three works included in the thesis reflect collaborative efforts. For all the works, I made the main contributions by deriving the theoretical results, conducting the majority of the experiments, and writing the drafts. The co-authors have helped formulate the ideas in the early stage, design the models and experiments, as well as refine the works.

Zhiqi Shao, 10/07/2025

Supervisor Statement. As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Prof. Junbin Gao, 10/07/2025

Certificate of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work.

This thesis has not been submitted for any degree or other purposes.

I declare that any contribution made to the research by others, with whom I have worked at the University of Sydney or elsewhere, is explicitly acknowledged in the thesis.

_____

Zhiqi Shao 10/07/2025

Use of generative artificial intelligence

During the preparation of the thesis, the author used ChatGPT for the purposes of text enhancement. The use of this generative AI tool includes sentence structure and spelling. The author confirms that where text was modified by generative AI, the content was reviewed for possible errors, inaccuracies, and bias. The author takes full responsibility for the submitted thesis and ensures the work is their own and has used generative AI within the parameters of use (refer to the University of Sydney generative AI guide for researchers).

Zhiqi Shao 10/07/2025

Table of Content

List of Figures

List of Tables

Notations

TABLE 0.1. Complete Notation Reference for CCDSReFormer (Chapter 3)

Symbol/Operation	Dimension	Description
	Traffic Network & Data	
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$	–	Directed road network graph
$\mathcal{V}$, $N = \lvert\mathcal{V}\rvert$	– / $\mathbb{N}$	Set of sensor nodes; N nodes total
$\mathcal{E}$	–	Set of directed edges
$\mathcal{A} \in \mathbb{R}^{N\times N}$	(N, N)	Adjacency/weight matrix
T	$\mathbb{N}$	Total historical time steps
M	$\mathbb{N}$	Input window length (historical steps)
Z	$\mathbb{N}$	Prediction horizon (future steps)
d	$\mathbb{N}$	Feature/channel dimension after embedding
$\mathbf{X}_t \in \mathbb{R}^{N\times d}$	(N, d)	Traffic snapshot at time t
$\mathcal{X} \in \mathbb{R}^{T\times N\times d}$	(T, N, d)	Full spatio-temporal traffic tensor
$f(\cdot)$	–	Traffic prediction function
	Embeddings	
$\mathbf{X}_{\text{data}}$	(M, N, d)	Data embedding via fully connected layer
$\mathbf{X}_{\text{spe}}$	(M, N, d)	Spatial Laplacian embedding
$\mathbf{X}_w, \mathbf{X}_d$	(M, d)	Weekly and daily temporal embeddings
$\mathbf{X}_{\text{tpe}}$	(M, d)	Temporal positional encoding
$\mathbf{X}_{\text{emb}}$	(M, N, d)	Final concatenated embedding

Continued on next page

Table 0.1 continued from previous page

Symbol/Operation	Dimension	Description
	Graph Laplacian	
$\mathbf{D}$	(N, N)	Diagonal degree matrix
$\mathbf{\Delta}$	(N, N)	Symmetric normalized Laplacian
$\mathbf{U} \in \mathbb{R}^{N \times N}$	(N, N)	Laplacian eigenvectors
$\mathbf{\Lambda}$	(N, N)	Laplacian eigenvalues (diagonal)
k	$\mathbb{N}$	Number of retained eigenvectors (default 8)
	General Attention Components	
$\mathbf{Q}, \mathbf{K}, \mathbf{V}$	(N, d_0)	Query, key, value matrices (per head)
$\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$	(d, d_0)	Learnable projection matrices
d_0	$\mathbb{N}$	Sub-space dimension per attention head
$\bar{\mathbf{A}}$	(N, N)	Scaled dot-product attention scores
$\bar{\mathbf{O}}$	(N, N)	Attention weights after activation
$\mathbf{M}_{\text{geo}}, \mathbf{M}_{\text{sem}}$	(N, N)	Geographic and semantic masking matrices
	Spatial Attention (ReSSA)	
$\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)}$	(N, d_0)	Spatial Q/K/V at time t
$\mathbf{A}_t^{(sp)}$	(N, N)	Spatial attention scores at time t
$\mathbf{W}_Q^{(sp)}, \mathbf{W}_K^{(sp)}, \mathbf{W}_V^{(sp)}$	(d, d_0)	Spatial projection matrices
	Temporal Attention (ReTSA)	
$\mathbf{Q}_n^{(te)}, \mathbf{K}_n^{(te)}, \mathbf{V}_n^{(te)}$	(M, d_0)	Temporal Q/K/V for node n
$\mathbf{A}_n^{(te)}$	(M, M)	Temporal attention scores for node n
$\mathbf{W}_Q^{(te)}, \mathbf{W}_K^{(te)}, \mathbf{W}_V^{(te)}$	(d, d_0)	Temporal projection matrices
	Delay-Aware Attention (ReDASA)	
$\hat{\mathbf{K}}_t^{(sp)}$	(N, d_0)	Modified key matrix with delay information

Continued on next page

Table 0.1 continued from previous page

Symbol/Operation	Dimension	Description
$\hat{\mathbf{A}}_t^{(sp)}$	(N, N)	Delay-aware attention scores
	Multi-Head Configuration	
$h_{\text{ReSSA}}, h_{\text{ReTSA}}, h_{\text{ReDASA}}$	$\mathbb{N}$	Number of heads for each attention type
$O^{\text{ReSSA}}, O^{\text{ReTSA}}, O^{\text{ReDASA}}$	various	Outputs from each attention module
$\hat{\mathbf{W}}$	(d, d)	Final projection matrix for concatenated heads
	Rectified Linear Self-Attention (ReLSA)	
$\text{ReLU}(\cdot)$	–	Rectified linear unit activation
$\text{LN}(\cdot)$	–	Layer normalization (RMSNorm)
g	(d)	Gain parameter for normalization
$\text{RMS}(\cdot)$	–	Root mean square statistic
$\odot$	–	Hadamard (element-wise) product
	Enhanced Convolution (EnCov)	
$\text{EnCov}(\cdot)$	–	Enhanced 3×3 2D convolution operation
$\text{Conv1}, \text{Conv2}$	–	1×1 convolutions in output layer
	Criss-Cross Dual-Stream (CCDS)	
P_{ST}	–	Spatial-to-temporal processing path
P_{TS}	–	Temporal-to-spatial processing path
$\Phi(\cdot)$	–	Aggregation function for dual streams
O^{CCDS}	(M, N, d)	Output from CCDS mechanism
	Output Components	
$\mathcal{Y}$	(T_0, N, d)	Final prediction output
$\mathcal{Y}_{\text{hid}}$	(M, N, d_{sk})	Hidden state from skip connections

Continued on next page

Table 0.1 continued from previous page

Symbol/Operation	Dimension	Description
$\mathcal{Y}_{\text{sk}}$	(M, N, d_{sk})	Skip connection output
d_{sk}	$\mathbb{N}$	Skip connection dimension
L	$\mathbb{N}$	Number of encoder/decoder layers
T_0	$\mathbb{N}$	Output time steps (prediction horizon)
	Enhanced Rectified Linear Self-Attention	
$\text{EnReLSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$	Function	$\text{ReLSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \text{EnCov}(\mathbf{V})$
$\text{ReLSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$	Function	$\text{LN}(\text{ReLU}(\mathbf{A} \odot \mathbf{M})\mathbf{V})$
	Specific Attention Modules	
$\text{ReSSA}(\cdot)$	Function	Rectified Spatial Self-Attention
$\text{ReTSA}(\cdot)$	Function	Rectified Temporal Self-Attention
$\text{ReDASA}(\cdot)$	Function	Rectified Delay-Aware Self-Attention
ReSTSA	Function	Combined Rectified Spatial-Temporal SA
	Broadcasting and Aggregation	
$\oplus_k$	Operator	Broadcasting summation along k-th mode
$\bigoplus$	Operator	Concatenation operation
$\text{softmax}(\cdot)$	Function	Softmax normalization
$\sqrt{d_0}$	Scalar	Scaling factor for attention scores
	Positional Encoding	
$\sin(t/10000^{2i/d})$	Function	Sinusoidal encoding (even dimensions)
$\cos(t/10000^{2(i-1)/d})$	Function	Cosine encoding (odd dimensions)
$w(M), d(M)$	Function	Weekly and daily index conversion

TABLE 0.2. Complete Notation Reference for ST-MambaSync (Chapter 4)

Symbol/Operation	Dimension	Description
	General Attention Components	
$\mathbf{Q}, \mathbf{K}, \mathbf{V}$	$(\cdot \times d_h)$	Query, key, value matrices
$\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$	$(d_h \times d_h)$	Learnable projection matrices
d_0	$\mathbb{N}$	Sub-space size per attention head
$\bar{\mathbf{A}}$	$(\cdot \times \cdot)$	Scaled dot-product attention scores
$\bar{\mathbf{O}}$	$(\cdot \times \cdot)$	Softmax-normalized attention weights
	Temporal Attention	
$\mathbf{Q}_i^{(te)}, \mathbf{K}_i^{(te)}, \mathbf{V}_i^{(te)}$	$(M \times d_h)$	Temporal Q/K/V for node i
$\mathbf{X}_i^{(te)}$	$(M \times d_h)$	Temporal input for node i
$\mathbf{W}_Q^{(te)}, \mathbf{W}_K^{(te)}, \mathbf{W}_V^{(te)}$	$(d_h \times d_h)$	Temporal projection matrices
$\mathbf{A}_i^{(te)}$	$(M \times M)$	Temporal attention scores for node i
$\tilde{\mathbf{X}}_i^{(te)}$	$(M \times d_h)$	Temporal transformer output for node i
$\tilde{\mathbf{X}}^{(te)}$	$(M \times N \times d_h)$	Complete temporal transformer output
	Spatial Attention	
$\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)}$	$(N \times d_h)$	Spatial Q/K/V at time t
$\tilde{\mathbf{X}}_t^{(te)}$	$(N \times d_h)$	Temporal output at time t
$\mathbf{W}_Q^{(sp)}, \mathbf{W}_K^{(sp)}, \mathbf{W}_V^{(sp)}$	$(d_h \times d_h)$	Spatial projection matrices
$\mathbf{A}_t^{(sp)}$	$(N \times N)$	Spatial attention scores at time t
$\mathbf{X}_t^{(sp)}$	$(N \times d_h)$	Spatial transformer output at time t
$\mathbf{X}^{(sp)}$	$(M \times N \times d_h)$	Complete spatial transformer output
	ST-Mixer and Reshaping	
$\bar{\mathbf{U}}_t$	$(\tilde{T} \times d_h)$	Reshaped tensor output from ST-mixer

Continued on next page

Table 0.2 continued from previous page

Symbol/Operation	Dimension	Description
$\tilde{T}$	$\mathbb{N}$	Flattened dimension ($M \times N$)
reshape$(\cdot)$	Function	Tensor reshaping operation
	ST-Mamba Layer	
$\tilde{\mathbf{X}}_t$	$(\tilde{T} \times d_h)$	Linear transformation of $\bar{\mathbf{U}}_t$
$\mathbf{x}_k$	(d_h)	Input vector at step k (where $k = 1, \ldots, \tilde{T}$)
$\mathbf{Y}_t$	$(\tilde{T} \times d_h)$	ST-Mamba layer output
$\mathbf{y}_k$	(d_h)	Output vector at step k
	State Space Model Parameters	
d_{state}	$\mathbb{N}$	State dimension
$\mathbf{A}$	$(d_{\text{state}} \times d_{\text{state}})$	State transition matrix (continuous)
$\mathbf{B}$	$(d_{\text{state}} \times d_h)$	Input matrix (continuous)
$\mathbf{C}$	$(d_h \times d_{\text{state}})$	Output projection matrix
$\tilde{\mathbf{A}}_k, \tilde{\mathbf{B}}_k, \mathbf{C}_k$	various	Discrete-time SSM parameters at step k
$\mathbf{h}_k$	(d_{state})	Hidden state at step k
Δ	$\mathbb{R}^{d_{\text{state}} \times 1}$	Step size parameter
Δ_k	Scalar/Vector	Step size at iteration k
	Parameter Functions	
$s_B(\cdot), s_C(\cdot), s_\Delta(\cdot)$	Function	Learnable linear projections
τ_Δ	Function	Softplus activation function
exp$(\cdot)$	Function	Matrix exponential
$\mathbf{I}$	$(d_{\text{state}} \times d_{\text{state}})$	Identity matrix
	Core Functions	
Dense$(\cdot)$	Function	Dense neural layer transformation

Continued on next page

Table 0.2 continued from previous page

Symbol/Operation	Dimension	Description
Linear$(\cdot)$	Function	Linear transformation
Concatenate$(\cdot)$	Function	Tensor concatenation operation
softmax$(\cdot)$	Function	Softmax normalization
Normalization$(\cdot)$	Function	Layer normalization
$FC(\cdot)$	Function	Fully connected layer
	Normalization Parameters	
μ, σ^2	$(\tilde{T} \times 1)$	Mean and variance along feature dimension
γ, β	$(1 \times d_h)$	Scale and shift parameters
ϵ	Scalar	Small constant for numerical stability
$\odot$	Operator	Element-wise (Hadamard) product
	Output Components	
$\bar{\mathbf{Y}}$	$(\tilde{T} \times d_h)$	Normalized output with residual connection
$\bar{\mathbf{X}}$	$(\tilde{T} \times d_h)$	Residual connection from ST-Mixer
$\mathcal{Y}$	$(Z \times N \times d)$	Final prediction output
	Mamba as Attention Analogy	
$\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m$	various	Mamba query, key, value analogies
$\mathbf{W}$	$(d_{\text{state}} \times d_{\text{state}})$	Weight matrix analogy $(e^{\mathbf{A}\Delta_t})$
β	Scalar	Scaling factor (Δ_t)
	Continuous-Time State Space	
τ	$\mathbb{R}^+$	Continuous time variable
$\mathbf{h}(\tau), \mathbf{x}(\tau), \mathbf{y}(\tau)$	various	State, input, output as functions of time
$\frac{d\mathbf{h}}{d\tau}$	(d_{state})	State derivative with respect to time

Continued on next page

Table 0.2 continued from previous page

Symbol/Operation	Dimension	Description
	Performance Metrics	
MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert$	Mean Absolute Error
MAPE	$\frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{\hat{y}_i - y_i}{y_i}\right\rvert \times 100$	Mean Absolute Percentage Error
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$	Root Mean Square Error
$y_i, \hat{y}_i$	Scalar	Actual and predicted values
n	$\mathbb{N}$	Number of samples
	Model Architecture Notation	
A#	$\mathbb{N}$	Number of attention layers
M#	$\mathbb{N}$	Number of Mamba layers
FLOPS	Scalar	Floating Point Operations Per Second
	Dataset Characteristics	
CV	Scalar	Coefficient of Variation
t_a, t_b	Time indices	Start and end time indices
Δ_i	Scalar	Time step at index i
	Common Model Variables	
M	$\mathbb{N}$	Input window length (historical steps)
N	$\mathbb{N}$	Number of nodes/sensors
Z	$\mathbb{N}$	Prediction horizon (future steps)
d_h	$\mathbb{N}$	Hidden dimension size
d	$\mathbb{N}$	Feature dimension

TABLE 0.3. Complete Notation Reference for STDAttMamba (Chapter 5)

Symbol/Operation	Dimension	Description
	Sparse Temporal Attention	
v	Index	Station (node) index, $v \in \{1, 2, \ldots, N\}$
$\mathbf{E}_v^{m(t)}$	$(M \times d_h)$	Temporal input features for node v
$\mathbf{Q}_v^{(t)}, \mathbf{K}_v^{(t)}, \mathbf{V}_v^{(t)}$	$(M \times d_h)$	Query, key, value matrices for temporal attention
$\mathbf{W}_{tQ}, \mathbf{W}_{tK}, \mathbf{W}_{tV}$	$(d_h \times d_h)$	Learnable weight matrices for temporal attention
$\mathbf{O}_v^{(t)}$	$(M \times M)$	Sparse temporal attention scores for node v
$\mathrm{ReLU}(\cdot)$	Function	Rectified Linear Unit activation
$\tilde{\mathbf{E}}_v^{m(t)}$	$(M \times d_h)$	Improved temporal embedding for node v
$\tilde{\mathbf{E}}_t^m$	$(M \times N \times d_h)$	Concatenated temporal features across all nodes
	Sparse Spatial Attention	
M_0	Index	Time slice index, $M_0 \in \{1, 2, \ldots, M\}$
$\tilde{\mathbf{E}}_{M_0}^{m(v)}$	$(N \times d_h)$	Spatial features at time slice M_0
$\mathbf{Q}_{M_0}^{(v)}, \mathbf{K}_{M_0}^{(v)}, \mathbf{V}_{M_0}^{(v)}$	$(N \times d_h)$	Query, key, value matrices for spatial attention
$\mathbf{W}_{vQ}, \mathbf{W}_{vK}, \mathbf{W}_{vV}$	$(d_h \times d_h)$	Learnable weight matrices for spatial attention
$\mathbf{O}_{M_0}^{(v)}$	$(N \times N)$	Sparse spatial attention scores at time M_0
$\mathbf{E}_{M_0}^{m(v)}$	$(N \times d_h)$	Improved spatial embedding at time M_0
$\widehat{\mathbf{X}}_t^m$	$(M \times N \times d_h)$	Final output integrating spatial-temporal dependencies

Continued on next page

Table 0.3 continued from previous page

Symbol/Operation	Dimension	Description
	Normalization	
$\mathrm{LN}(\cdot)$	Function	Layer normalization (RMSNorm)
$\sqrt{d_h}$	Scalar	Scaling factor for attention computation
	State Space Model Parameters	
i	Index	Pathway index: $i = 1$ (spatial), $i = 2$ (temporal)
d_{state}	$\mathbb{N}$	State dimension
$\mathbf{A}_i$	$(d_{\text{state}} \times d_{\text{state}})$	State transition matrix for path i
$\mathbf{B}_i$	$(d_{\text{state}} \times d_h)$	Input projection matrix for path i
$\mathbf{C}_i$	$(d_h \times d_{\text{state}})$	Output projection matrix for path i
$s_B(\cdot), s_C(\cdot)$	Function	Learnable linear projections
$\bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i$	various	Discrete-time parameters after ZOH discretization
	Spatial-Temporal Dynamic Fusion (STDF)	
$\mathbf{h}_{v,t}$	(d_{state})	Spatial hidden state at location v, time t
$\mathbf{h}_{t,v}$	(d_{state})	Temporal hidden state at time t, location v
$\mathbf{x}_{v+1}(t), \mathbf{x}_{t+1}(v)$	(d_h)	Input from adjacent nodes/future time steps
α	$[0, 1]$	Trainable parameter for spatial-temporal balance
$\mathbf{y}_{v,t}$	(d_h)	Output at spatial-temporal position (v, t)
$\mathbf{Y}^m$	$(N \times M \times d_h)$	STDF layer output for mode m

Continued on next page

Table 0.3 continued from previous page

Symbol/Operation	Dimension	Description
	Continuous-Time Formulation	
$\tilde{t}^{(1)}, \tilde{t}^{(2)}$	Time variables	Two time dimensions in dual-path system
$\mathbf{h}(\tilde{t}^{(1)}), \mathbf{h}(\tilde{t}^{(2)})$	(d_{state})	Hidden states along two time dimensions
$\mathbf{x}(\tilde{t}^{(1)}), \mathbf{x}(\tilde{t}^{(2)})$	(d_h)	Inputs along two time dimensions
$\mathbf{y}(\tilde{t}^{(1)}, \tilde{t}^{(2)})$	(d_h)	Output combining both time dimensions
	Zero-Order Hold (ZOH) Discretization	
Δt	Scalar	Discretization time step
$e^{\mathbf{A}\Delta t}$	$(d_{\text{state}} \times d_{\text{state}})$	Matrix exponential for discretization
$\mathbf{I}$	$(d_{\text{state}} \times d_{\text{state}})$	Identity matrix
$\mathbf{A}^{-1}$	$(d_{\text{state}} \times d_{\text{state}})$	Inverse of state transition matrix
	Dual-Path Attention Reformulation	
$\tilde{\mathbf{Q}}_v, \tilde{\mathbf{Q}}_t$	$(d_h \times d_{\text{state}})$	Query matrices for spatial/temporal paths
$\tilde{\mathbf{K}}_v^\top, \tilde{\mathbf{K}}_t^\top$	$(d_{\text{state}} \times d_h)$	Key matrices for spatial/temporal paths
$\tilde{\mathbf{V}}_v, \tilde{\mathbf{V}}_t$	(d_h)	Value matrices for spatial/temporal paths
β_1, β_2	Scalar	Weight parameters ($e^{\mathbf{A}\Delta_{v,t}}$, $e^{\mathbf{A}\Delta_{t,v}}$)
γ_v, γ_t	Scalar	Scaling factors ($\Delta_{v,t}$, $\Delta_{t,v}$)
$\mathbf{y}_{v+1,t+1}$	(d_h)	Predicted output at next spatial-temporal position
	Complementary Analysis	
$\mathcal{R}_{\text{local}}(i)$	Set	Local receptive field for position i

Continued on next page

Table 0.3 continued from previous page

Symbol/Operation	Dimension	Description
$\mathcal{R}_{\text{global}}(i)$	Set	Global receptive field for position i
k	$\mathbb{N}$	Local neighborhood size
w_{ij}	Scalar	Attention weight between positions i and j
v_j	(d_h)	Value vector at position j
h_i	(d_h)	Hidden state at position i
Global-Local Decomposition		
$\mathbf{Q}_v\mathbf{K}_v^\top\mathbf{V}_v\gamma_v$	(d_h)	Spatial global focus component
$\mathbf{Q}_t\mathbf{K}_t^\top\mathbf{V}_t\gamma_t$	(d_h)	Temporal global focus component
$\beta_1\mathbf{V}_v$	(d_h)	Spatial local focus component
$\beta_2\mathbf{V}_t$	(d_h)	Temporal local focus component
STDMamba Block Components		
$\widetilde{\mathbf{X}}_1^m, \widetilde{\mathbf{X}}_2^m$	$(M \times N \times d_h)$	Intermediate outputs after normalization
LayerNorm$(\cdot)$	Function	Layer normalization operation
FFN$(\cdot)$	Function	Feed-forward network
STDMamba$(\cdot)$	Function	Spatial-temporal dynamic Mamba operation
$\mathcal{L}$	$\mathbb{N}$	Number of STDMamba blocks
$\widehat{\mathbf{Y}}^m$	$(M \times N \times d_h)$	Final STDMamba output for mode m
$\hat{\mathbf{Y}}_{\text{out}}^m$	$(Z \times N \times d)$	Final prediction output for mode m
OutputProjection$(\cdot)$	Function	Mixed projection for final output

Continued on next page

Table 0.3 continued from previous page

Symbol/Operation	Dimension	Description
	Evaluation Metrics	
MAE	Scalar	$\frac{1}{N^m} \sum_{i=1}^{N^m} \lvert \hat{y}_i^m - y_i^m \rvert$
MAPE	Percentage	$\frac{100\%}{N^m} \sum_{i=1}^{N^m} \left\lvert \frac{\hat{y}_i^m - y_i^m}{y_i^m} \right\rvert$
RMSE	Scalar	$\sqrt{\frac{1}{N^m} \sum_{i=1}^{N^m} (\hat{y}_i^m - y_i^m)^2}$
N^m	$\mathbb{N}$	Number of samples for mode m
$\hat{y}_i^m, y_i^m$	Scalar	Predicted and ground truth values
	Multi-Task Learning	
PES	Scalar	Pareto Efficiency Score (rank sum)
FLOPS	Scalar	Floating Point Operations Per Second
Cohen's d	Scalar	Effect size measure for statistical significance
p-value	Scalar	Statistical significance measure
	External Factors (Weather Extensions)	
WX	Suffix	Weather-extended model variant
F	$\mathbb{N}$	Number of additional weather features (16)
`RainIntensity`	Binary	High rainfall/wind indicator
`StormRisk`	Binary	Severe weather risk indicator
`LowVisibility`	Binary	Poor visibility indicator
`HolidayFlag`	Binary	Public holiday indicator
	Common Model Variables	

Continued on next page

Table 0.3 continued from previous page

Symbol/Operation	Dimension	Description
M	$\mathbb{N}$	Input window length (historical steps)
N	$\mathbb{N}$	Number of nodes/sensors
Z	$\mathbb{N}$	Prediction horizon (future steps)
d_h	$\mathbb{N}$	Hidden dimension size
d	$\mathbb{N}$	Feature dimension
m	Index	Mode index (e.g., traffic flow, speed, occupancy)

List of Acronyms

Acronym	Full Form
AGCRN	Adaptive Graph Convolutional Recurrent Network
AI	Artificial Intelligence
ANN	Artificial Neural Network
ARIMA	Autoregressive Integrated Moving Average
ASTGNN	Attention-based Spatial-Temporal Graph Neural Network
BiLSTM	Bidirectional Long Short-Term Memory
CAS-CNN	Channel-wise Attentive Split-Convolutional Neural Network
CCDS	Criss-Crossed Dual-Stream
CCDSReFormer	Criss-Crossed Dual-Stream Enhanced Rectified Transformer
CNN	Convolutional Neural Network
DCNN	Diffusion Convolutional Neural Network
DCRNN	Diffusion Convolutional Recurrent Neural Network
DL	Deep Learning
DSTGNN	Dynamical Spatial-Temporal Graph Neural Network
EnCov	Enhanced Convolution
EnReLSA	Enhanced Rectified Linear Self-Attention
FC	Fully Connected
FFN	Feed-Forward Network
FLOPS	Floating Point Operations Per Second

Acronym	Full Form
FTCN	Fast Time Convolution Network
GCN	Graph Convolutional Network
GCRN	Graph Convolutional Recurrent Module
GMAN	Graph Multi-Attention Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
GTS	Graph Time Series
GWNET	Graph WaveNet
HI	Historical Index
HiPPO	High-order Polynomial Projection Operators
IG-Net	Interaction Graph Network
ITSs	Intelligent Transportation Systems
KA2M2	Knowledge Adaptation with Attentive Multi-task Memory Network
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MFGCN	Multi-feature Fusion Graph Convolutional Network
MGC-RNN	Multi-Graph Convolutional-Recurrent Neural Network
ML	Machine Learning
MLP	Multi-Layer Perceptron

Acronym	Full Form
MMoE	Multi-gate Mixture-of-Experts
MT-STNet	Multi-task Spatiotemporal Network
MTGNN	Multivariate Time Series Graph Neural Network
NLP	Natural Language Processing
OD	Origin-Destination
PDFormer	Propagation Delay-aware Dynamic Long-range Transformer
PES	Pareto Efficiency Score
PGCN	Progressive Graph Convolutional Network
PT	Public Transit
ReDASA	Rectified Delay Aware Self Attention
ReLSA	Rectified Linear Self Attention
ReSSA	Rectified Spatial Self-Attention
ReSTSA	Rectified Spatial-Temporal Self-Attention
ReTSA	Rectified Temporal Self-Attention
RMSE	Root Mean Square Error
RMSNorm	Root Mean Square Normalization
RNN	Recurrent Neural Network
S4	Structured State Space Sequence Model
SARIMA	Seasonal Autoregressive Integrated Moving Average
SOTA	State-Of-The-Art
Sp-LSTM	Spatiotemporal Long Short-Term Memory
SSM	State Space Model
ST-MambaSync	Spatial-Temporal Mamba Synchronization

Acronym	Full Form
ST-MRGNN	Spatiotemporal Multi-Relational Graph Neural Network
STAEFormer	Spatio-Temporal Adaptive Embedding Transformer
STCNN	Spatiotemporal Convolutional Neural Network
STDAtt	Spatial-Temporal Dynamic Attention
STDAtt-Mamba	Spatial-Temporal Dynamic Attention-based Mamba
STDF	Spatial-Temporal Dynamic Fusion
STDMamba	Spatial-Temporal Dynamic Mamba
STFGNN	Spatio-Temporal Fusion Graph Neural Network
STGCN	Spatio-Temporal Graph Convolutional Network
STGNCDE	Spatio-Temporal Graph Neural Controlled Differential Equation
STGT	Spatio-Temporal Graph Transformer
STID	Spatial-Temporal Identity
STMTL	Spatial-Temporal Multi-Task Learning
STNorm	Spatial-Temporal Normalization
STTN	Spatial-Temporal Transformer Network
SVM	Support Vector Machine
SVR	Support Vector Regression
TLC	Taxi & Limousine Commission
TS-STN	Temporally Shifted Spatiotemporal Network
TS-STNN	Tree-structured Spatial-Temporal Neural Network
VAR	Vector Auto-Regression
ZOH	Zero-Order Hold

Abstract

Accurate and efficient traffic flow prediction is a cornerstone of modern Intelligent Transportation Systems (ITSs). With the rapid expansion of urban road networks and sensor infrastructures, forecasting methodologies must capture complex spatial-temporal dependencies while remaining computationally tractable for real-time deployment. Traditional statistical methods often fall short in modeling non-linear patterns and handling large-scale data, whereas deep learning approaches—such as Convolutional Neural Networks, Recurrent Neural Networks, and Graph Neural Networks—encounter challenges in capturing long-range dependencies or suffer from excessive computational overhead. This thesis advances spatial-temporal deep learning models for traffic flow and passenger demand prediction within intelligent transportation systems (ITSs), addressing critical trade-offs between predictive accuracy and computational efficiency. We first introduce CCDSReFormer, a novel transformer-based model incorporating a criss-crossed dual-stream mechanism, enhanced rectified linear self-attention, and geographic-semantic masking strategies, achieving significant computational improvements and superior performance in traffic flow prediction. Second, we propose ST-MambaSync, a pioneering hybrid architecture integrating selective state-space (Mamba) and Transformer mechanisms, theoretically and empirically demonstrating substantial reductions in prediction errors and computational costs. Third, we present STDAtt-Mamba, a dynamic attention-based state-space model specifically designed to capture heterogeneous passenger behaviors in multimodal public transit systems, significantly outperforming existing approaches in multi-type passenger demand forecasting. Collectively, these methodological innovations provide robust theoretical contributions, empirical validations, and practical benefits, enabling scalable, efficient, and equitable transportation system management. Extensive experiments on real-world traffic datasets show that our proposed models consistently outperform existing methods on both accuracy and speed metrics. The research culminates in a cohesive strategy that significantly enhances the scalability and robustness of

spatial-temporal traffic forecasting, paving the way for more responsive and intelligent traffic management systems.

Chapter 1

Introduction

1.1 Motivation

The performance of urban transportation networks determines not only system-level efficiency—reflected in congestion levels, energy consumption, and accident risk—but also broader socio-economic and environmental outcomes [Zhang et al., 2011]. A cornerstone of any Intelligent Transportation System (ITS) is therefore the ability to *forecast* traffic and passenger flows in order to support proactive interventions such as adaptive signal timing, dynamic routing, or targeted public-transport rescheduling.

Yet traffic and passenger dynamics are intrinsically spatial–temporal, highly non-linear, and shaped by heterogeneous behavioural drivers. Any forecast model suitable for city-scale deployment must simultaneously satisfy three, often conflicting, requirements: **expressiveness** to learn coupled spatial *and* temporal patterns, both local and global; **computational tractability** to deliver real-time inference for millions of sensors and smart-card taps; and **generalisability** to transfer across multiple transit modes and diverse passenger cohorts.

The rapid evolution of data acquisition technologies, coupled with the exponential increase in vehicle numbers, has facilitated a more comprehensive collection of traffic data, propelling traffic flow prediction to the forefront of urban transportation management research. Real-time traffic flow predictions, particularly with short intervals (e.g., every 5 minutes), offer the potential for immediate decision-making, enabling proactive traffic management strategies that optimise traffic signal control, rerouting, and public transport scheduling. These interventions not only alleviate congestion but also enhance the overall safety and efficiency of transportation systems.

The environmental implications are substantial—the U.S. Department of Energy estimates that traffic congestion wastes approximately 3 billion gallons of fuel annually. By mitigating congestion through better prediction and management, we can significantly reduce carbon emissions, aligning with global sustainability goals.

1.2 Comprehensive Literature Analysis

To establish a unified foundation for addressing the challenges in spatial-temporal forecasting, this section provides a comprehensive analysis of the literature spanning traffic flow prediction, architectural efficiency, and multimodal demand forecasting. This analysis reveals fundamental limitations across different methodological paradigms that no existing approach adequately addresses.

1.2.1 Evolution of Spatial-Temporal Forecasting Methods

1.2.1.1 Classical Statistical Approaches

Historically, traditional mathematical statistics methods and classical machine learning techniques focused primarily on short-term prediction. Long-range spatial dependencies in traffic forecasting are particularly challenging and computationally intensive due to the complex and dynamic nature of spatial-temporal dependencies involved. Traffic conditions on a particular road segment are influenced not only by historical data from that segment but also by current and past conditions of connected or nearby roads, exponentially increasing computational demands.

Classical time-series techniques—ARIMA, state-space Kalman filtering, and exponential smoothing—excel for short sequences on a single sensor but degrade sharply when extended to network-wide, high-frequency data [Ahmed and Cook, 1979, Kumar, 2017, Xu et al., 2017]. These methods assume weak stationarity, a poor fit for the daily, weekly, and event-driven regime shifts that typify urban mobility [Zeng et al., 2008, Gao et al., 2013]. Moreover,

they lack mechanisms to model the complex spatial interdependencies that characterise urban transportation networks, where congestion at one location can cascade through the entire system.

Vector Auto-Regression (VAR) models [Hamilton, 1994], Support Vector Regression (SVR) [Drucker et al., 1997a], and other regression techniques [Alam et al., 2019, Priambodo and Ahmad, 2017] have been used for traffic flow prediction but must be trained on large datasets to achieve higher precision and can only extract temporal features from traffic flow data, effectively ignoring spatial information that is important for predicting traffic flow.

1.2.1.2 Early Neural Network Approaches

The emergence of neural forecasting paradigms has partially addressed the limitations of classical methods but introduced new challenges. Recurrent Neural Networks, including LSTM and GRU variants, capture temporal correlations effectively but struggle with vanishing gradients for very long horizons and operate sequentially, limiting parallelism [Oliveira et al., 2021, Luo et al., 2019, Ma et al., 2022]. While these methods are crucial in modelling temporal aspects of traffic, their sequential nature restricts their parallelisation, slowing down both training and inference compared to CNNs.

Convolutional Neural Networks exploit spatial locality through convolutions but, in their vanilla form, lack explicit mechanisms for long temporal dependencies [Zhang et al., 2017, 2019]. CNNs are adept at handling spatial data but often falter with long-range temporal patterns. To overcome these limitations, some hybrid models combine CNNs with LSTM [Bogaerts et al., 2020, Méndez et al., 2023, Li et al., 2021c] for capturing long-range dependencies while increasing model complexity and inducing high computational cost.

Non-parametric machine learning methods such as artificial neural networks (ANNs) [Topuz, 2010, Zeng et al., 2008, Sharma et al., 2018], k-nearest neighbours (KNN) [Luo et al., 2019, Rahman, 2020, Cai et al., 2020b], and support vector machines (SVM) [Rahman, 2020] have also been used to predict traffic flow with varying degrees of success, but these approaches generally lack the capacity to model complex spatial-temporal dependencies effectively.

1.2.2 Graph Neural Network Revolution

Graph Neural Networks have emerged as the current workhorse for transportation network modelling by representing road networks as graphs and employing message passing to capture spatial dependencies. The spotlight has turned to GNNs for their adeptness in handling traffic data, effectively represented as graphs, embracing a flexible, graph-based approach suitable for the complex, non-Euclidean topology of traffic networks [Asif et al., 2021].

Several innovative GNN architectures have been developed specifically for traffic prediction. The Diffusion Convolutional Recurrent Neural Network (DCRNN) [Li et al., 2017] uses bidirectional random walks for complex spatial-temporal dynamics. Spatial-Temporal Graph Convolutional Networks (STGCN) [Yu et al., 2018b] employ efficient convolutional graph structures. Graph WaveNet (GWNET) [Wu et al., 2019] introduces adaptive graph modelling, while Multi-variate Time Series Graph Neural Network (MTGNN) [Wu et al., 2020b] offers novel graph learning techniques. Additionally, STFGNN and STGNCDE provide advanced spatio-temporal forecasting by integrating differential equations with neural networks [Li and Zhu, 2021, Choi and Park, 2023].

Recent advances include Progressive Graph Convolutional Networks (PGCN) for enhancing adaptability [Shin and Yoon, 2024], models incorporating learnable positional attention in GNNs for capturing spatial-temporal patterns [Wang et al., 2020], and Dynamics Extractor with node connection strength indices for enhanced traffic flow characteristics analysis [Chen et al., 2023]. AST-InceptionNet introduces multi-scale feature extraction and adaptive graph convolutions to address spatial heterogeneity and unknown adjacencies [Wang et al., 2023b].

However, GNNs face several critical limitations that become apparent in large-scale deployments. Multi-hop message passing can lead to over-smoothing, where node representations become increasingly similar and lose local specificity [Wu et al., 2019, BAI et al., 2020]. When paired with attention mechanisms to capture long-range dependencies, GNNs incur quadratic computational cost that becomes prohibitive for city-scale networks. Furthermore, GNN approaches typically

require manual construction of adjacency matrices based on domain knowledge, limiting their adaptability to different transportation modes and network topologies.

1.2.3 Attention Mechanisms and Transformer Architectures

The growing adoption of attention mechanisms in traffic forecasting has become increasingly prevalent following the success of the Transformer architecture [Vaswani, 2017]. The self-attention mechanism has significantly impacted data processing for complex tasks by dynamically focusing on the most pertinent input data sections, enabling concurrent processing of spatial and temporal information.

Several transformer-based models have been developed for traffic prediction. The Spatial-Temporal Transformer Network (STTN) [Xu et al., 2020b] uses spatial transformers and long-range temporal dependencies to dynamically model directed spatial dependencies. The Graph Multi-Attention Network (GMAN) [Zheng et al., 2020b] utilises an encoder-decoder architecture with spatio-temporal attention blocks, featuring an attention layer that effectively links historical and future time steps for long-term traffic prediction. The Attention-based Spatial-Temporal Graph Neural Network (ASTGNN) [Guo et al., 2019] integrates a graph convolutional recurrent module with a global attention module designed to model both long-term and short-term temporal correlations.

The Propagation Delay-aware Dynamic Long-range Transformer (PDFormer) [Jiang et al., 2023] designs multi-self-attention modules to capture dynamic relations and explicitly model time delays in traffic systems. Progressive Space-Time Self-Attention (ProSTformer) [Yan et al., 2024] focuses on spatial dependencies from local to global regions. More recently, the Spatial-Temporal Adaptive Embedding Transformer (STAEformer) [Liu et al., 2023b] has highlighted the efficacy of spatial and temporal attention mechanisms, noted for state-of-the-art performance in managing long and short-range traffic data.

Despite these advances, transformer-based models face significant computational challenges. The self-attention mechanism in transformers is resource-intensive, with computational costs

scaling quadratically as sequence lengths increase. This presents challenges in large-scale traffic networks or scenarios requiring rapid real-time analysis. Increasing model accuracy involves deeper attention layers, exacerbating computational demands. This trade-off between accuracy and feasibility is crucial in real-time traffic systems, necessitating a balance between swift predictions and resource constraints.

1.2.4 State Space Models and Mamba

Given the limitations of existing deep learning methods, the Selective State Space model (commonly referred to as Mamba) [Gu and Dao, 2023] stands out for its ability to deliver high accuracy on very long-range traffic flow prediction while requiring less computational effort. Building on the foundation of Structured State Space Sequence (S4) models, Mamba introduced the Selective State Space model architecture, which improved upon the limitations of S4 by offering linear computational complexity and robust long-range dependency modelling.

Recent studies have extended Mamba to spatial-temporal applications [Shao et al., 2024a,h], demonstrating promising results in reducing computational costs. However, it faces limitations in capturing heterogeneous spatial-temporal dependencies that induce a lack of generalisation for multi-tasks. Empirical results indicate that naive layer stacking methods can degrade model performance, highlighting the need for more sophisticated architectural design.

The ST-Mamba [Shao et al., 2024h] marked the first application of the Mamba model to spatial-temporal traffic flow prediction, demonstrating promising results in reducing computational costs. However, simply adding more Mamba layers does not guarantee increased prediction accuracy because Mamba focuses locally and may lose some global information with additional layers. This limitation necessitates combining Mamba with architectures that can enhance global data comprehension.

1.2.5 Multimodal and Multi-Type Passenger Demand Prediction

Urban mobility systems serve diverse socio-demographic passenger groups with varying travel needs, such as adults, seniors, youth, pensioners, and students. Predicting passenger demand in multimodal PT systems is crucial for optimising service allocation, improving passenger satisfaction, and supporting sustainable urban transport planning [Xi et al., 2024a]. However, existing demand prediction models often fail to capture diverse travel patterns exhibited by different passenger groups.

Traditional approaches to passenger demand prediction have focused on homogeneous passenger populations and typically process spatial and temporal dynamics separately. Studies such as [Luo et al., 2020] introduced multitask deep learning models for bus passenger flow prediction, while [Zhang et al., 2021b] developed channel-wise attentive split-convolutional neural networks for Origin-Destination prediction. However, these approaches lack the capability to model heterogeneous passenger behaviours across multiple transit modes effectively.

Graph-based approaches for passenger demand prediction include the probabilistic graph convolution model (PGCM) [Li et al., 2020] and Multi-Graph Convolutional-Recurrent Neural Networks (MGC-RNN) [He et al., 2022]. Transformer-based models such as the Heterogeneous Information Aggregation Machine (HIAM) [Liu et al., 2022] and MultiModeformer (M2-former) [Yang et al., 2024a] have shown promise in capturing dynamic spatiotemporal correlations in multi-mode systems.

Despite these advances, existing models predominantly focus on predicting aggregated passenger demand without differentiating between various socio-demographic passenger groups, limiting their capability in addressing diverse behaviours and needs. Most studies process spatial and temporal data independently or through static integration methods, lacking dynamic mechanisms to jointly adapt spatial and temporal features.

1.3 Research Gaps and Unified Problem Statement

Based on the comprehensive literature analysis, three fundamental gaps emerge that collectively limit the advancement of spatial-temporal forecasting for intelligent transportation systems.

1.3.1 Identified Research Gaps

Gap 1: Expressiveness-Efficiency Trade-off (G1) Current methods force a binary choice between expressiveness and efficiency. Transformer-based models achieve high accuracy through global attention but incur quadratic computational cost that becomes prohibitive for city-scale deployment. State-space models offer linear complexity but sacrifice global spatial modelling capability. No existing architecture simultaneously satisfies expressiveness for capturing complex spatial-temporal patterns, computational tractability for real-time deployment at city scale, and generalisability across different transportation modes and network configurations.

Evidence supporting this gap includes: Transformer variants achieving state-of-the-art accuracy but requiring more than double the inference time compared to operational requirements; Mamba-based models reducing computational cost by 40–60% but suffering 5–15% accuracy degradation on large-scale networks; and the absence of any existing architecture that concurrently meets all three criteria of expressiveness, tractability, and generalisability.

Gap 2: Architectural Synergy Gap (G2) The complementary strengths of different architectural paradigms remain unexploited. Linear-time state-space models excel at local pattern modelling while attention mechanisms capture global dependencies, yet principled fusion frameworks are missing. Current approaches either use these architectures in isolation or combine them through ad-hoc methods that do not leverage their synergistic potential.

This gap is evidenced by: sequential stacking of different architectures leading to computational overhead without guaranteed synergy; existing ensemble approaches using weighted averaging or late fusion that ignore architectural complementarity; and the absence of theoretical frameworks for understanding when and how different architectures should interact.

Gap 3: Multimodal Generalisation Gap (G3) Current methods rely on hand-crafted, mode-specific graphs and feature engineering, limiting their ability to scale to heterogeneous passenger demand across multiple transportation modes without extensive manual intervention. Traffic flow models require road network topology graphs, transit models need station connectivity graphs, and passenger demand models require demographic-specific feature engineering. No unified framework handles bus, rail, and ferry systems with diverse passenger cohorts.

Supporting evidence includes: the requirement for separate models for different transportation modes, each with manually engineered features and graph structures; poor scalability to integrated transportation systems where passengers use multiple modes within single journeys; and the lack of frameworks for system-wide optimisation that requires understanding cross-modal interactions.

1.3.2 Unified Problem Statement

These three gaps collectively represent a fundamental challenge in spatial-temporal forecasting: the inability to develop models that simultaneously achieve high expressiveness, computational efficiency, and broad generalisability. This challenge manifests across three interconnected dimensions:

Local-Global Pattern Learning: The need to capture both fine-grained local patterns at individual sensors or stations and network-wide propagation effects that influence system behaviour, without incurring prohibitive computational costs.

Architectural Complementarity: The requirement to understand and exploit the synergistic relationships between different neural architectures, particularly between linear-time state-space models and attention mechanisms, through principled theoretical foundations.

Multimodal Scalability: The necessity to handle real-world transportation systems that integrate multiple modes and serve diverse passenger populations with heterogeneous travel patterns, without extensive manual engineering for each mode and passenger type.

1.4 Research Questions and Programme Logic

The identified gaps motivate three focal Research Questions that form a logical progression towards a comprehensive solution for next-generation spatial-temporal forecasting.

Research Question 1 (RQ1): Joint Local-Global Spatial-Temporal Learning How can local (fine-grained) and global (network-wide) spatial-temporal patterns be learned jointly without prohibitive computational cost?

This question addresses the fundamental tension between capturing detailed local patterns at individual sensors or stations and understanding network-wide propagation effects that influence system behaviour. Traditional approaches either focus on local patterns while missing global context, or attempt global modelling at prohibitive computational cost. The question seeks architectural innovations that can achieve both objectives simultaneously while maintaining computational tractability for real-time deployment.

Sub-questions include: What architectural designs enable concurrent spatial and temporal feature extraction? How can attention mechanisms be made computationally tractable for city-scale networks? What fusion strategies preserve both local specificity and global context?

Research Question 2 (RQ2): Synergistic Architecture Fusion What synergistic architectural principles enable linear-time state-space models to complement—rather than replace—attention mechanisms?

This question recognises that different architectural paradigms have complementary strengths that could be leveraged through principled fusion. State-space models excel at local pattern modelling with linear complexity, while attention mechanisms capture global dependencies effectively but at quadratic cost. Rather than viewing these as competing approaches, this question seeks theoretical understanding and practical frameworks for combining their strengths.

Sub-questions include: What theoretical relationship exists between state-space models and attention mechanisms? How can bidirectional information flow be established between different

architectural components? What synchronisation schedules optimise the accuracy-efficiency trade-off?

Research Question 3 (RQ3): Multimodal Demand Generalisation How can these architectural principles be generalised to forecast heterogeneous passenger demand across multiple transit modes without extensive manual graph construction?

This question addresses the practical deployment challenge of handling real-world transportation systems that integrate multiple modes (bus, rail, ferry) and serve diverse passenger populations with different travel patterns. Current approaches require extensive manual engineering for each mode and passenger type, limiting scalability and generalisability.

Sub-questions include: How can adaptive embeddings capture diverse passenger demographics and transit modes? What attention mechanisms handle cross-modal dependencies efficiently? How can spatial-temporal patterns be dynamically fused for heterogeneous demand?

1.4.1 Research Programme Logic

These three research questions form a logical progression that builds from foundational architectural innovations through theoretical understanding of architectural synergy to practical deployment in complex, multimodal systems:

Foundation (RQ1): Establishes efficient spatial-temporal learning principles through dual-stream architecture and rectified attention mechanisms, demonstrating that concurrent processing of spatial and temporal information can achieve superior performance while maintaining computational tractability.

Synergy (RQ2): Develops theoretical understanding and practical fusion of complementary architectures, proving that state-space models and attention mechanisms exhibit natural complementarity that can be exploited through principled frameworks.

Generalisation (RQ3): Extends synergistic principles to heterogeneous, multimodal real-world scenarios, demonstrating how adaptive embeddings and dynamic fusion mechanisms can eliminate manual graph construction while handling diverse passenger demographics.

Each stage builds upon previous contributions while addressing increasingly complex real-world requirements, culminating in a comprehensive framework for next-generation spatial-temporal forecasting that meets the operational requirements of modern transportation systems.

1.5 Thesis Contributions

The thesis addresses these questions through a sequential research program, each stage documented in an accepted peer-reviewed publication. Our contribution for this thesis can be summarized as:

- **Modelling**: Three novel neural architectures (CCDSReFormer, ST-MambaSync, STDAtt-Mamba) that progressively advance expressiveness, efficiency, and generalisability in spatial–temporal forecasting.
- **Theoretical Analysis**: A formal interpretation that casts Mamba updates as depth-wise linear attention within a ResNet structure, thereby proving the complementarity between selective state-space and sparse attention mechanisms.
- **Empirical Evidence**: A comprehensive benchmarking suite covering (i) six standard traffic-flow datasets and (ii) a large-scale, two-year multimodal smart-card corpus with nine passenger categories.
- **Practical Impact**: An open-source inference stack whose latency profiles meet the operational requirements of metropolitan traffic control centres.

1.5.1 Paper I—*CCDSReFormer*: Dual-Stream Spatial–Temporal Learning

The first study introduces the Criss-Cross Dual-Stream Enhanced Rectified Transformer (CCDSReFormer), which directly addresses RQ1 by demonstrating how local and global spatial-temporal patterns can be learned jointly without prohibitive computational cost.

Key Innovation: The criss-cross dual-stream architecture enables concurrent spatial and temporal feature extraction through parallel pathways, while the rectified linear self-attention (ReLSA) mechanism provides dynamic, cost-aware attention allocation that reduces computational complexity from quadratic to linear scaling.

Contribution to RQ1: Establishes that dual-stream processing can achieve joint spatial-temporal learning without computational penalties, demonstrating state-of-the-art accuracy improvements averaging 5.55% over existing methods while reducing attention-related operations by 25–35%.

1.5.2 Paper II—*ST-MambaSync*: Accuracy–Efficiency Pareto Optimisation

The second study proposes ST-MambaSync, which provides both theoretical understanding and practical implementation of synergistic architecture fusion, directly addressing RQ2 through formal analysis of architectural complementarity.

Key Innovation: The theoretical proof that Mamba functions as depth-wise linear attention within a ResNet structure, combined with bidirectional synchronisation protocols that enable principled fusion of state-space models and attention mechanisms.

Contribution to RQ2: Demonstrates that state-space models and attention mechanisms exhibit natural complementarity, achieving Pareto optimality with 64.9% reduction in floating-point operations and 12.5% reduction in inference time while simultaneously improving accuracy.

1.5.3 Paper III—STDAtt-Mamba: Multi-Type Passenger Demand Prediction in Multimodal PT Systems

The third study proposes STDAtt-Mamba, a spatial–temporal dynamic attention–based state–space model designed to address heterogeneous passenger demand prediction in multimodal public transport (PT) systems. This work directly addresses RQ3 by combining formal theoretical analysis, novel architectural design, and large–scale empirical validation.

Key Innovation: Development of the spatial–temporal dynamic fusion (STDF) layer, theoretically reformulated as a dual–path global–local attention mechanism, enabling unified modeling of spatial and temporal dependencies without the need for manual graph construction. The architecture integrates the sparse global attention of STDAtt with the efficient local state–space modeling of STDMamba, providing adaptive and complementary feature integration across passenger groups and modes.

Contribution to RQ3: Establishes that combining state–space models with sparse attention yields natural complementarity—capturing both fine–grained local dependencies and long–range global patterns—leading to state-of-the-art accuracy across three travel modes and nine passenger groups. Achieves a 5–17% reduction in MAE compared to 19 baselines while maintaining computational efficiency, with the lowest Pareto Efficiency Score (PES=7) and eliminating the requirement for resource–intensive graph construction.

1.6 Thesis Organization

The thesis is composed of the following chapters:

- **Chapter 2** contextualizes the fundamental concepts, notations, and mathematical frameworks necessary for our spatial-temporal analysis.
- **Chapter 3** details the Criss-Crossed Dual-Stream Enhanced Rectified Transformer (CCDSReFormer) for boosting local-global feature capture at lower computational cost.

- **Chapter 4** describes ST-MambaSync, a fusion of Mamba's local, ResNet-like attention and the Transformer's global attention, offering efficient, accurate long-term traffic forecasts.
- **Chapter 5** propose STDAtt-Mamba, a spatial-temporal dynamic attention-based state-space model that effectively predicts multimodal transit demand across diverse socio-demographic groups by integrating adaptive embeddings, dual-path attention, and dynamic spatial-temporal modeling.
- **Chapter 6** synthesizes the contributions of this thesis, highlighting theoretical innovations, computational efficiency, and practical implications. It further discusses limitations.
- **Chapter 7** concludes the thesis by summarizing the primary contributions, discussing constraints, and proposing future research avenues to refine next-generation spatial-temporal forecasting models.

Chapter 2

Preliminaries

This chapter establishes the fundamental concepts, notations, and mathematical frameworks necessary for our spatial-temporal analysis. We begin by introducing the representation of traffic networks and defining the spatial-temporal prediction problem. We then delve into two core building blocks of our methodology: (i) the attention mechanism, which dynamically highlights relevant information in the data, and (ii) the Selective State Space Model (Mamba), an advanced framework for capturing complex, input-dependent dynamics. These foundational elements will pave the way for the methods proposed in later chapters.

2.1 Notations and Problem Formulation

2.1.1 Traffic Network Representation

We model the traffic network as a graph that encapsulates the spatial relationships between measurement points, such as sensors or intersections. By using a graph-based representation, it becomes straightforward to incorporate spatial adjacency, directionality, and potentially even weights (e.g., distances or road capacities) into the predictive model.

Definition 2.1 (Road Network Graph). A road network is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where:

- $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a set of $N = |\mathcal{V}|$ nodes representing points of interest such as traffic sensors, intersections, or stations.
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges corresponding to the roads connecting these nodes.

- $\mathcal{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix that describes connectivity between nodes; it can be unweighted or weighted (e.g., using physical distances).

Graph-based modeling provides a natural way to account for the geometry of the road network and the interactions between different regions. For example, if two nodes are connected in $\mathcal{G}$, we assume their traffic conditions may exhibit stronger correlations. This representation serves as a stepping stone for more advanced models that incorporate both spatial and temporal dependencies. The graph-based dataset can be represented as in Figure 2.1.

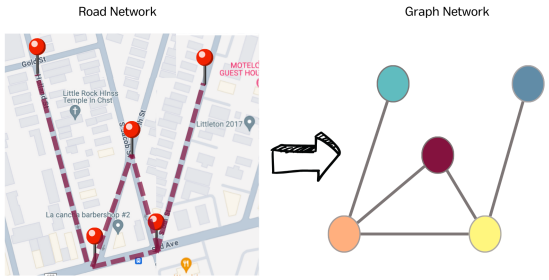


FIGURE 2.1. Illustration of a road network and its corresponding graph representation. The left panel shows geographical locations of traffic sensors, while the right panel depicts the graph network with nodes connected according to the road network topology.

2.1.2 Spatial-Temporal Data Structure

In addition to spatial interconnections, traffic states vary over time. Hence, the data is inherently *spatial-temporal* and is often recorded by sensors at regular intervals. We organize this data into a three-dimensional tensor to handle the interplay among time, space (nodes), and traffic features.

DEFINITION 2.2 (Traffic Flow Tensor). Given a road network $\mathcal{G}$ with N nodes, the traffic flow information at time t is denoted by $\mathbf{X}_t \in \mathbb{R}^{N \times d}$, where d is the dimensionality of the flow features (e.g., traffic speed, volume, or occupancy). Over a time window of length T, the collection of these measurements forms a *mode-3 tensor*:

$$\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T) \in \mathbb{R}^{T \times N \times d},$$

where:

- The first mode (T) indexes the timestamp.
- The second mode (N) indexes the node in the network.
- The third mode (d) indexes the various flow-related features.

This tensor-based view naturally aligns with how the data is collected and facilitates the application of machine learning techniques that exploit correlations across both space (nodes) and time (timestamps). When combined with the graph structure, the three-dimensional tensor helps capture both local and global patterns in the network (see Figure 2.2). The spatial and temporal dependency is defined as in Figure 2.3.

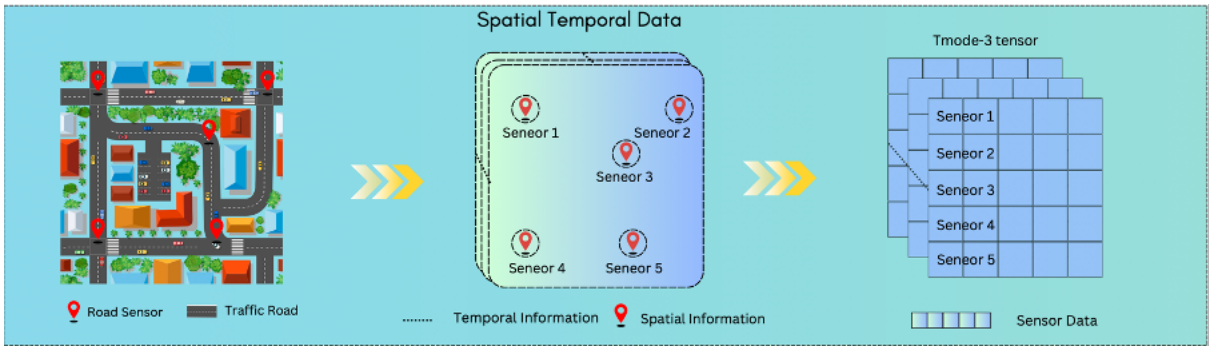


FIGURE 2.2. The grid dataset is converted by the road maps into spatial-temporal information. Each front face of the tensor collects sensor information at a particular time point, with rows representing the spatial dimension and front faces composing the temporal dimension.

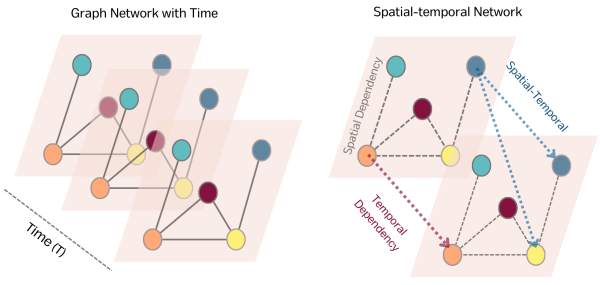


FIGURE 2.3. The graph dataset of spatial-temporal network representation. The left depicts the underlying graph structure, while the right illustrates the spatial-temporal connections for a single node. Grey lines denote spatial links, the red arrow shows temporal self-influence, and the blue arrows indicate combined spatial-temporal influence between nodes at consecutive time steps.

2.1.3 Problem Statement

Spatial-temporal prediction is the task of forecasting future traffic conditions based on historical measurements. It is critical for applications such as route planning, congestion management, and intelligent transportation systems.

Definition 2.3 (Spatial-Temporal Prediction Task). Given the historical spatial-temporal data $[\mathbf{X}_{t-M+1}, \ldots, \mathbf{X}_t]$ spanning M timestamps, the goal is to predict the future spatial-temporal data $[\mathbf{X}_{t+1}, \ldots, \mathbf{X}_{t+Z}]$ over the next Z timestamps using a function $f(\cdot)$ with parameters θ:

$$[\mathbf{X}_{t-M+1}, \ldots, \mathbf{X}_t] \xrightarrow{f(\cdot;\theta)} [\mathbf{X}_{t+1}, \ldots, \mathbf{X}_{t+Z}].$$

During model learning, we set $t = M, \ldots, T - Z$ to fully utilize the available historical data.

Here, $\mathbf{X}_t$ encodes the network-wide traffic state at time t, and the goal is to *accurately* predict future states even under evolving conditions like peak-hour congestion or accidents. Common evaluation metrics include Mean Squared Error (MSE) or Mean Absolute Error (MAE). The overarching challenge is to combine *both* spatial and temporal information effectively to achieve high-fidelity forecasts.

2.2 Attention Mechanism

Attention mechanisms have revolutionized sequence modeling tasks by enabling models to focus on the most relevant parts of the input data. Originally introduced in machine translation, attention has since been used widely in fields requiring the modeling of long-range dependencies (e.g., time series forecasting, text analysis, and image captioning).

2.2.1 Vanilla Self-Attention

The self-attention mechanism empowers a model to learn *which* parts of a sequence (or set of features) are most informative. Given an input matrix $\bar{\mathbf{X}} \in \mathbb{R}^{N \times d}$, self-attention constructs *queries*, *keys*, and *values*:

$$\mathbf{Q} = \bar{\mathbf{X}}\mathbf{W}_Q, \quad \mathbf{K} = \bar{\mathbf{X}}\mathbf{W}_K, \quad \mathbf{V} = \bar{\mathbf{X}}\mathbf{W}_V, \tag{2.1}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learnable weight matrices that project the input into different latent spaces. Next, the *attention scores* are computed by the scaled dot product between queries and keys:

$$\bar{\mathbf{A}} = \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}, \tag{2.2}$$

where d_k is the dimensionality of the keys. The factor $\sqrt{d_k}$ normalizes the result to avoid overly large dot products, which can destabilize training.

These raw scores are then converted into a probability distribution via the softmax function:

$$\bar{\mathbf{O}} = \text{softmax}(\bar{\mathbf{A}}), \tag{2.3}$$

and the final output $\mathbf{Y}$ is computed as a weighted sum of the values:

$$\bar{\mathbf{Y}} = \bar{\mathbf{O}}\mathbf{V}. \tag{2.4}$$

Hence, each location in $\bar{\mathbf{X}}$ can selectively focus on other parts of $\bar{\mathbf{X}}$ based on learned relevance scores.

2.2.2 Multi-Head Attention

To capture multiple types of relationships simultaneously (e.g., different aspects of congestion patterns), *multi-head attention* uses several parallel self-attention layers (heads). Each head attends to different segments or features of the input, and their outputs are concatenated:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)\, \mathbf{W}_O, \tag{2.5}$$

where each head head_i is computed as:

$$\text{head}_i = \text{Attention}(\bar{\mathbf{X}}\mathbf{W}_{Q_i}, \bar{\mathbf{X}}\mathbf{W}_{K_i}, \bar{\mathbf{X}}\mathbf{W}_{V_i}). \tag{2.6}$$

The final projection matrix $\mathbf{W}_O$ blends the contributions of each head. By attending to different parts of the input, multi-head attention can uncover a richer set of spatial-temporal dependencies.

In the context of traffic forecasting, attention can help identify critical nodes (e.g., major intersections) or time steps (e.g., rush hours) that significantly influence overall traffic flow. This is especially valuable when dealing with large, complex road networks.

2.3 Selective State Space Models

Many dynamical systems, including traffic networks, are governed by evolving internal states influenced by external signals. *State Space Models (SSMs)* provide a rigorous mathematical structure for describing these hidden states and their evolution over time. However, conventional SSMs typically use fixed system matrices, which can limit their adaptability when external inputs dramatically change the system dynamics.

2.3.1 Classic Formulation

Let $\tau \in (0, +\infty)$ represent continuous time, and $\mathbf{h}(\tau) \in \mathbb{R}^d$ be the hidden state capturing unobserved aspects of the system. The classical *continuous-time* SSM can be written as:

$$\mathbf{h}'(\tau) = \mathbf{A}\,\mathbf{h}(\tau) + \mathbf{B}\,\mathbf{x}(\tau), \quad \mathbf{y}(\tau) = \mathbf{C}\,\mathbf{h}(\tau), \tag{2.7}$$

where $\mathbf{x}(\tau) \in \mathbb{R}^{d_x}$ is an external input (e.g., traffic inflow from adjacent regions) and $\mathbf{y}(\tau) \in \mathbb{R}^{d_y}$ is the measured output (e.g., observed traffic speed). The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ encode *fixed* linear transformations.

Often, we work in *discrete-time* to match sensor measurement intervals. One common discretization leads to:

$$\mathbf{h}_k = \tilde{\mathbf{A}}\,\mathbf{h}_{k-1} + \tilde{\mathbf{B}}\,\mathbf{x}_k, \quad \mathbf{y}_k = \mathbf{C}\,\mathbf{h}_k, \tag{2.8}$$

where k indexes the discrete time steps, and $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$ are derived from $\mathbf{A}, \mathbf{B}$ via matrix exponentials and integral approximations.

2.3.2 Input-Dependent Extensions: The Mamba Model

Real traffic systems are typically non-stationary: rush hour patterns, special events, and weather conditions can drastically alter normal flow. To handle such variability, the *Selective State Space Model (Mamba)* introduces *input-dependent* system matrices:

$$\mathbf{A}_\tau = f_A(\mathbf{x}(\tau)), \quad \mathbf{B}_\tau = f_B(\mathbf{x}(\tau)), \quad \mathbf{C}_\tau = f_C(\mathbf{x}(\tau)), \tag{2.9}$$

where f_A, f_B, f_C are learnable functions (often neural networks) that adapt the linear transformations based on the current input. In a discrete-time context, these become $\tilde{\mathbf{A}}_k, \tilde{\mathbf{B}}_k, \mathbf{C}_k$ for each time step k.

This *selective mechanism* allows the model to emphasize relevant information (e.g., major incidents or spikes in demand) while downplaying less critical factors. By dynamically modulating the system matrices, Mamba can better capture time-varying patterns in a spatial-temporal context. It effectively bridges the gap between purely data-driven approaches (e.g., deep neural networks) and classical, well-interpreted linear dynamical systems, combining the best of both worlds.

2.4 Conclusion

In this chapter, we have established the core concepts and frameworks that underpin our approach to spatial-temporal prediction. We introduced:

- A **graph-based representation** of traffic networks, enabling the integration of spatial connectivity.
- The **traffic flow tensor** structure, capturing measurements over time and space.
- The **spatial-temporal prediction task**, highlighting its importance and typical challenges.
- The **attention mechanism**, which endows models with the flexibility to selectively focus on critical nodes and time steps.

- The **Selective State Space Model (Mamba)**, offering an input-dependent extension of classic SSMs, vital for capturing non-stationary behaviors.

These elements form the backbone of our proposed methodology, guiding how we design and train models to leverage both spatial and temporal dependencies effectively. In the following chapters, we will illustrate how these pieces come together in an end-to-end architecture that tackles real-world traffic forecasting problems with greater accuracy and robustness.

In this chapter, we introduced the foundational representations—such as graphs and tensors—along with key problem formulations and essential background on attention mechanisms and State Space Models (SSMs). Building on these preliminaries, the next three chapter presents three novel models: the *Criss-Crossed Dual-Stream Enhanced Rectified Transformer* (**CCDSReFormer**), the *Spatial-Temporal State Space Model Synchronized with Transformer* (**ST-MambaSync**), and the *Spatial-Temporal Dynamic State Space Attention-Based Model* (**STDAttMamba**). These models are designed to address the challenges identified in this chapter, particularly the need to effectively capture dynamic spatial-temporal dependencies while reducing computational overhead. We also demonstrate how the core components introduced here—such as attention mechanisms and SSM dynamics—are integrated, extended, and adapted within our proposed frameworks.

CHAPTER 3

Criss-Crossed Dual-Stream Transformer for Traffic Flow Prediction

---

In the previous chapter, we outlined the traffic network representation, problem formulation, and essential background on attention and State Space Models (SSMs). Building on these concepts, this chapter tackles **RQ1** : How can local (fine-grained) and global (network-wide) spatial–temporal patterns be learned *jointly* without prohibitive computational cost? by proposing **CCDSReFormer**—a dual-stream architecture that balances local–global feature extraction while reducing attention FLOPs.[1]

3.1 Introduction

At the core of advancing intelligent transportation systems (ITSs) is the intricate challenge of traffic flow prediction, which is crucial for enhancing route efficiency, mitigating congestion, and reducing accident rates in urban settings. The rapid evolution of data acquisition technologies, coupled with the exponential increase in vehicle numbers, has facilitated a more comprehensive collection of traffic data, propelling traffic flow prediction to the forefront of urban transportation management research [Zhang et al., 2011]. Real-time traffic flow predictions, particularly with short intervals (e.g., every 5 minutes), offer the potential for immediate decision-making, enabling proactive traffic management strategies that optimize traffic signal control, rerouting, and public transport scheduling. These interventions not only alleviate congestion but also enhance the overall safety and efficiency of transportation systems. Also, the environmental implications are substantial. The U.S. Department of Energy estimates that traffic congestion

[1] For cross-chapter coherence, the full wording of **RQ1** : How can local (fine-grained) and global (network-wide) spatial–temporal patterns be learned *jointly* without prohibitive computational cost? is reproduced verbatim via the macro.

wastes approximately 3 billion gallons of fuel annually. By mitigating congestion through better prediction and management, we can significantly reduce carbon emissions, aligning with global sustainability goals.

The quintessential hurdle in this domain is the accurate modeling and understanding of spatiotemporal dependencies within traffic data. These dependencies, reflecting the dynamic nature of traffic flow, involve complex patterns of spatial and temporal interplay that are constantly evolving. Capturing these patterns is fundamental for accurate prediction and poses a significant computational challenge, necessitating a delicate balance between prediction accuracy and computational efficiency [Jiang et al., 2023, Gomes et al., 2023].

Traditional methodologies, such as autoregressive integrated moving average (ARIMA) models [Ahmed and Cook, 1979, Zeng et al., 2008, Chen et al., 2011, Tong and Xue, 2008] and Kalman filters [Kumar, 2017, Emami et al., 2020, Xu et al., 2017, Zhou et al., 2019, Gao et al., 2013], have made strides in traffic prediction. Yet, their effectiveness is often limited by inherent assumptions of data stationarity and their inadequacy in capturing the complex spatiotemporal dependencies characteristic of traffic flow. Spatial regression methods [Zheng and Ni, 2013], while introducing spatial awareness, remain constrained by the dynamic intricacies of traffic patterns and the computational burdens they impose.

With the increasing computational power, deep learning has become the leading methodology in traffic flow prediction, excelling in handling the complex and nonlinear patterns of traffic data. In the past, emerging studies have focused on employing recurrent neural networks (RNNs) and its variants, such as long short-term memory (LSTM) [Oliveira et al., 2021, Luo et al., 2019, Ma et al., 2022], which are particularly suited to grid-based data [Fu et al., 2016, Hochreiter and Schmidhuber, 1997]. These methods are crucial in modeling temporal aspects of traffic. Convolutional Neural Networks (CNNs), which are also suitable for grid data, have been widely used due to their efficacy in extracting temporal features [Zhang et al., 2017, 2019, Yan et al., 2024, Liu et al., 2021b].

More recently, graph convolutional networks (GCNs), aligning with the graph-structured nature of traffic data, have shown promise in identifying intricate spatial characteristics and dependencies

in road networks [Zheng et al., 2022, Chen et al., 2022, Xing et al., 2023, Zhao et al., 2020, Shao et al., 2024g]. However, GNNs in traffic prediction encounter specific challenges: they struggle with dynamic spatial dependencies that change over time, have a limited capacity for long-range dependency modeling due to local interaction focus, and often overlook the time-delay impact in spatial information transfer [Hochreiter and Schmidhuber, 1997]. While the attention mechanism has demonstrated effectiveness in capturing both spatial and temporal information by adapting to the input changes [Jiang et al., 2023, Su et al., 2023, Wang et al., 2024b, Liao et al., 2024], there remain some limitations such as:

- Lack of integration of both spatial and temporal information for learning complex dynamics.
- Less adept at making short-term predictions due to lack of local feature focus.
- Dense matrices result in high computational cost.

To fill these gaps, we propose a state-of-the-art technique call the Criss-Cross Dual-Stream Enhanced Rectified Transformer (**CCDSReFormer**) model which consists of a novel Criss-Crossed Dual-Stream for enriched spatial and temporal learning, an Enhanced Convolution (**EnCov**) method focusing on local traffic pattern nuances, and a rectified linear self attention (**ReLSA**) mechanism for dynamic and computationally efficient attention allocation in traffic flow prediction. We show that our new method out-performs extant techniques used for analyzing spatial and temporal data in traffic networks.

Our main contributions are summarized as follows:

- Our model introduces a criss-crossed dual stream (**CCDS**), enabling simultaneous learning of spatial and temporal information to enhance performance. This dual approach effectively captures the complexities of traffic flow, offering a thorough understanding of spatial and temporal dynamics.

- We design the Enhanced Rectified Linear Self Attention (**EnReLSA**), which incorporates a locally enhanced convolution (**EnCov**) within the Rectified Linear Self Attention

(**ReLSA**) mechanism. This design enables the model to focus on local spatiotemporal features, capturing the subtle dynamics of traffic patterns influenced by localized conditions, while also reducing computational complexity.

- We conducted extensive experiments across six real-world datasets, demonstrating that our model outperforms the existing state-of-the-art in both performance and computational efficiency, with robust parameter tuning capabilities.

The rest of this chapter is structured as follows: Section 3.2 provides a summary of previous related studies. The proposed **CCDSReFormer** model is elaborated in Section 3.3. Section 3.4 given the theoretical analysis on the ReLSA. Section 3.5 discusses the experiments conducted and their results. Section 3.6 present the practical implications and limitations. Finally, the chapter concludes with a summary and future research directions in Section 3.7.

3.2 Literature Review

In previous studies, parameter-based approaches such as ARIMA models [Ahmed and Cook, 1979, Zeng et al., 2008, Chen et al., 2011, Tong and Xue, 2008], Kalman filters [Kumar, 2017, Emami et al., 2020, Xu et al., 2017, Zhou et al., 2019, Gao et al., 2013], and other regression techniques [Alam et al., 2019, Priambodo and Ahmad, 2017] have been used for traffic flow prediction. However, they must be trained on large data sets to achieve higher precision, and they can only extract the time features from traffic flow data, effectively ignoring the spatial information, which is important for predicting traffic flow [Emami et al., 2020]. Non-parametric machine learning methods such as artificial neural networks (ANNs) [Topuz, 2010, Zeng et al., 2008, Sharma et al., 2018, Cohen and Dalyot, 2019], k-nearest neighbors (KNN) [Luo et al., 2019, Rahman, 2020, Cai et al., 2020b, Yang et al., 2019] and support vector machines (SVM) [Rahman, 2020] have also been used to predict traffic flow, with varying degrees of success.

Deep learning models have notably improved traffic prediction methodologies. This section reviews various methods that utilize deep learning for traffic prediction. Generally, applications of deep learning in this domain can be classified into four major categories RNNs, CNNs, GNNs and the attention mechanism, each with its unique advantages.

RNNs, including LSTM are commonly applied to sequence data because of their memorization capability, which can learn both long and short-term dependencies between parts of a data sequence. Previously, they have been used for traffic flow prediction as exemplified in [Oliveira et al., 2021, Luo et al., 2019, Ma et al., 2022, Li et al., 2021b, Yang et al., 2021a]. These papers demonstrated the ability of these methods to capture the temporal patterns in traffic data. However, by their ability to take in very long data sequences, these methods suffer from the vanishing gradient problem [Hochreiter, 1998].

CNNs have shown proficiency in processing grid-based spatial data, particularly in capturing spatial dependencies within traffic data, making them a go-to choice for early traffic prediction tasks as evidenced by studies such as [Zhang et al., 2017, Ke et al., 2017, Duan et al., 2016]. More recently, [Wu et al., 2018, Kong et al., 2024a] proposed a hybrid model that combines CNNs for spatial feature extraction and gated recurrent units (GRUs) for temporal feature analysis in traffic flow data. The research in [Zhang et al., 2019] further harnesses CNNs, employing a spatio-temporal feature selection algorithm to optimize input data and extract pivotal traffic features, enhancing the predictive model.

Recently, the spotlight has turned to Graph Neural Networks (GNNs) for their adeptness in handling traffic data, effectively represented as graphs. Unlike the rigid structure of CNNs, GNNs embrace a flexible, graph-based approach suitable for the complex, non-Euclidean topology of traffic networks [Asif et al., 2021]. They have been increasingly utilized for traffic flow prediction, with innovations such as integrating learnable positional attention in GNNs for capturing spatial-temporal patterns [Wang et al., 2020], the Progressive Graph Convolutional Network (PGCN) for enhancing adaptability to both training and testing phases [Shin and Yoon, 2024] and models like the Dynamics Extractor and node connection strength index for enhancing traffic flow characteristics analysis [Chen et al., 2023]. Other notable developments include

DCRNN's use of bidirectional random walks for complex spatial-temporal dynamics [Li et al., 2017], STGCN's efficient convolutional graph structures [Yu et al., 2018b], GWNET's adaptive graph modeling [Wu et al., 2019], and MTGNN's novel graph learning techniques [Wu et al., 2020b]. Additionally, STFGNN and STGNCDE offer advanced spatio-temporal forecasting by integrating differential equations with neural networks [Li and Zhu, 2021, Choi and Park, 2023]. The paper [Guo et al., 2021] uses an optimized GCN to preserve the spatial structure of road networks through a graph representation. AST-InceptionNet introduces multi-scale feature extraction and adaptive graph convolutions to address spatial heterogeneity and unknown adjacencies [Wang et al., 2023b]. Despite their advancements, these models confront challenges with dynamic spatial dependencies, long-range dependency modeling, over-smoothing, and time-delay impacts, indicating ongoing areas for refinement.

Therefore, the growing adoption of attention mechanisms in traffic forecasting is becoming increasingly prevalent. The self-attention mechanism developed by [Vaswani, 2017], has significantly impacted data processing for complex tasks, including language translation, image recognition, and sequence prediction, by dynamically focusing on the most pertinent input data sections. This mechanism is increasingly acknowledged for its ability to concurrently process spatial and temporal information. In traffic forecasting, models like the spatial-temporal transformer network model (STTN) [Xu et al., 2020b] uses spatial transformers and long-range temporal dependencies that dynamically model directed spatial dependencies, a graph multi-attention network model (GMAN) [Zheng et al., 2020b] utilizing an encoder-decoder architecture with spatio-temporal attention blocks to model the impact of spatio-temporal factors on traffic conditions, and featuring an attention layer that effectively links historical and future time steps for long-term traffic prediction, an attention-based spatial-temporal graph neural network (ASTGNN) [Guo et al., 2019] integrating a graph convolutional recurrent module (GCRN) with a global attention module, designed to effectively model both long-term and short-term temporal correlations in traffic data, and propagation delay-aware dynamic long-range transformer for traffic flow prediction (PDFormer) [Jiang et al., 2023] design multi-self-attention modules to capture the dynamic relations, a progressive space-time self-attention (ProSTformer) [Yan et al., 2024] focuses on spatial dependencies from local to global regions and a novel traffic flow

prediction approach combining Vision Transformers (VTs) and Convolutional Neural Networks (CNN) [Ramana et al., 2023] is used to accurately forecast urban congestion. These models collectively demonstrate the growing trend of incorporating advanced attention mechanisms to improve the accuracy and efficiency of spatio-temporal traffic forecasting. However, the complexity of these attention mechanisms can lead to increased computational costs, presenting a challenge that needs to be addressed. Finding a balance between computational efficiency and model accuracy remains a critical issue to resolve.

In summary, traffic flow prediction has progressed from traditional parameter-based methods to sophisticated deep learning approaches, offering a nuanced understanding of the complex spatio-temporal dynamics within transportation networks. While RNNs, CNNs, GNNs, and attention mechanisms have improved prediction accuracy, they continue to face challenges in balancing computational cost and efficiency. Additionally, achieving an optimal mix of local and global information is pivotal, as long-term forecasts tend to prioritize broader trends, potentially overlooking critical local insights. Addressing these issues is key to refining accuracy and computational practicality, steering future research toward more adept and streamlined forecasting techniques.

3.3 The Method

In this section, we introduce the proposed **CCDSReFormer** in detail. The framework of **CCDSReFormer** is shown as Figure 3.1.

3.3.1 Data Embedding Layer

Similar to the study [Jiang et al., 2023], the data embedding layer morphs the input into a higher-dimensional representation, $\mathbf{X}_{\text{data}} \in \mathbb{R}^{M \times N \times d}$, via a fully connected layer where d is the embedding dimension.

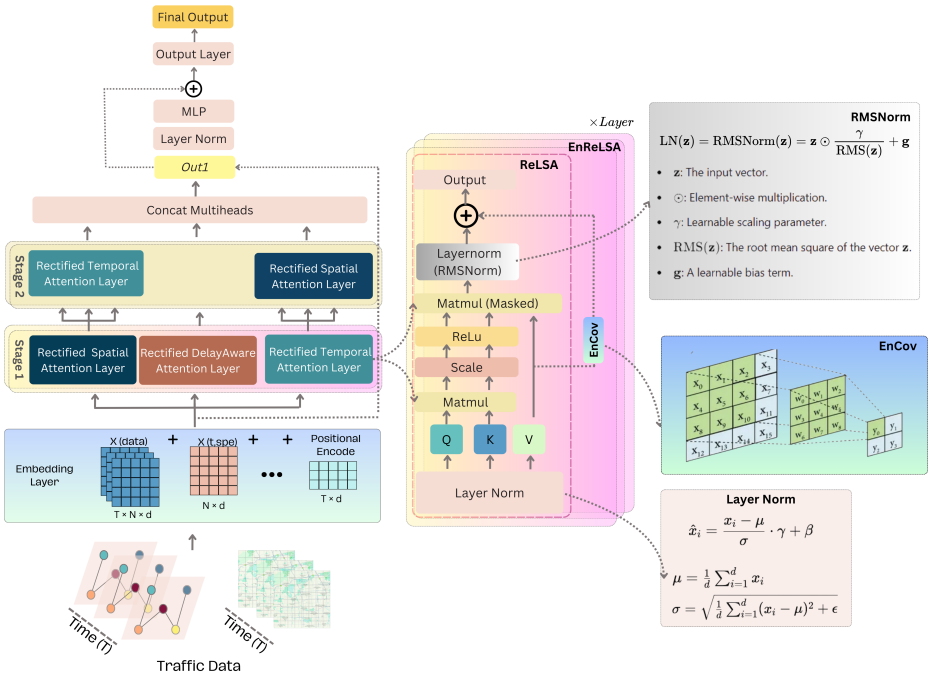


FIGURE 3.1. The figure shows the process of **CCDSReFormer**, which begins with the input traffic data being directed to the embedding layers. Subsequently, a layer normalization (Lay Norm) is applied. Following this, the model engages in various attention mechanisms including **ReSSA**, **ReTSA**, and **ReDASA**, facilitating cross-learning between **ReSSA**, **ReTSA** to assimilate information from various sources. The subsequent concatenation of these results yields the initial output, denoted as Out_1. This is followed by another layer normalization and a multi-layer perceptron (MLP) process. The output thus obtained is combined with the solution by concatenating the output Out_1 from the data embedding layer. Finally, this composite output is fed into the output layer to generate the final output.

To further explore and incorporate spatio-temporal network dynamics, the output of the data embedding layer is used as input for computing a spatial embedding of the road network structure as described below.

In similar vein, a temporal periodic embedding is used to capture recurrent variations in traffic flow such as the morning and afternoon rush hours.

Spatial Embedding. For each timestamp t, the spatial embedding $\mathbf{X}_{t,\text{spe}} \in \mathbb{R}^{N\times d}$ is derived from the graph Laplacian eigenvectors. The process begins with the computation of the normalized Laplacian matrix $\mathbf{\Delta}$ which is calculated using the adjacency matrix $\mathcal{A}$ and the diagonal degree matrix $\mathbf{D}$ where $\mathbf{D}(i,i) = \sum_{j=1}^{N} \mathcal{A}(i,j)$. The symmetrically normalized Laplacian matrix is

given by $\mathbf{\Delta} = \mathbf{I} - \mathbf{D}^{-1/2}\mathcal{A}\mathbf{D}^{-1/2}$, where $\mathbf{I}$ is the identity matrix of appropriate dimension. Secondly an eigenvalue decomposition of $\mathbf{\Delta}$ is performed, resulting in $\mathbf{\Delta} = \mathbf{U}^\top\mathbf{\Lambda}\mathbf{U}$ where $\mathbf{\Lambda} = \text{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{N-1})$ is the ordered matrix of eigenvalues, satisfying $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1}$, and $\mathbf{U} = (u_0, u_1, \ldots, u_{N-1})$ is the corresponding matrix of eigenvectors. We then select the k eigenvectors from $\mathbf{U}$ corresponding to the k smallest nontrivial eigenvalues to construct the k-dimensional embedding for all nodes $\mathcal{V}$ at time t, denoted by $\mathbf{U}_{t,\text{spe}} = [u_1, u_2, \ldots, u_k] \in \mathbb{R}^{N\times k}$. In the following the parameter $k = 8$ as used in the baseline model [Jiang et al., 2023]. Subsequently, $\mathbf{U}_{t,\text{spe}}$ is subjected to a linear transformation, mapping it into a new d-dimensional space. This process culminates in the formation of the spatial graph Laplacian embedding $\mathbf{X}_{t,\text{spe}} \in \mathbb{R}^{N\times d}$ at time t, effectively embedding the graph in a Euclidean space and thus preserving its global structure [Jiang et al., 2023].

Temporal Embedding. Each timestamp t, can be converted to either a weekly index $w(M)$ or a daily index $d(M)$. Concretely, the weekly index $w(M)$ translates the timestamp t into a day-of-week representation (1 to 7), while the daily index $d(M)$ maps it to the specific minute of the day (1 to 1440). All the indices are then converted into trainable temporal embeddings by a way of the embedding layers. The weekly and daily embeddings for all the timestamps are denoted by $\mathbf{X}_w \in \mathbb{R}^{M\times d}$ and $\mathbf{X}_d \in \mathbb{R}^{M\times d}$, respectively, where d is the same as the spatial embedding dimension.

Temporal Positional Encoding. For generating the temporal positional encoding $\mathbf{X}_{tpe}$, we draw inspiration from the study by [Vaswani, 2017]. In the context of traffic network prediction, it is crucial to account for another dynamic attribute: the temporal position relative to the input. To address this, we define the temporal input positions as $t = \{1, 2, \ldots M\}$, representing the sequence of timestamps. To capture this temporal aspect effectively, we define the temporal positional encoding as $\mathbf{X}_{t,\text{tpe}}$ for each timestamp t as a d-dimensional vector in $\mathbb{R}^d$. The components of $\mathbf{X}_{t,\text{tpe}}$ are computed as follows:

$$\mathbf{X}_{t,\text{tpe}}(i) = \begin{cases} \sin\left(\frac{t}{10000^{2i/d}}\right) & \text{if } i \text{ is even,} \\ \cos\left(\frac{t}{10000^{2(i-1)/d}}\right) & \text{if } i \text{ is odd.} \end{cases}$$

Finally we use $\mathbf{X}_{\text{tpe}} \in \mathbb{R}^{T \times d}$ to collect all the temporal positional encoding.

This ensures that each dimension of the positional encoding varies according to a sinusoidal function of a different wavelength, providing a unique and continuous encoding for each time step t. Such an encoding is instrumental in enabling the model to capture the nuances of the temporal dynamics inherent in road network data.

Final Output. The final output from the data embedding layer, represented as $\mathbf{X}_{\text{emb}}$, is simply the element-wise sum of the various components:

$$\mathbf{X}_{\text{emb}} = \mathbf{X}_{\text{data}} \oplus_1 \mathbf{X}_{\text{spe}} \oplus_2 \mathbf{X}_w \oplus_2 \mathbf{X}_d \oplus_2 \mathbf{X}_{\text{tpe}},$$

where $\oplus_k$ denotes the broadcasting summation along the kth mode to ensure dimensional consistency. This concept of broadcasting summation is derived from the functionality provided by Python's NumPy library [2]. The resulting $\mathbf{X}_{\text{emb}}$ is then passed to the spatio-temporal encoders. To simplify the notation the subscript emb will be dropped in the following sections, i.e., $\mathbf{X} \equiv \mathbf{X}_{\text{emb}}$.

3.3.2 Workflow of Enhanced Rectified Linear Self-Attention

To provide a clear and logical introduction to the **CCDSReFormer**, we begin with an overview of our newly designed attention which is named Enhanced Rectified Linear Self-Attention (**EnReLSA**), as depicted in Figure 3.1. Understanding this concept is crucial for grasping the spatial, temporal, and delay-aware modules that are central to the architecture of **CCDSReFormer**.

The **EnReLSA** approach adapts the standard self-attention mechanism by initially determining the query, key, and value components through specific matrix operations. It modifies these matrices with learnable parameters to tailor the model's focus. Further, to enhance the handling of the computational demand posed by the attention mechanism, **EnReLSA** incorporates a summation of Rectified Self-Attention **(ReLSA)** and enhanced convolution **EnCov**.

[2] Broadcasting summation refers to the capability in Python's NumPy library to perform element-wise binary operations on arrays of different sizes

Inspired by the study [Zhang et al., 2021a], the **ReLSA** approach introduces sparsity into the attention matrix, effectively reducing its complexity. This sparsity is achieved by applying the rectified linear unit (ReLU) to the attention matrix, which removes irrelevant attention scores by setting negative values to zero. Unlike traditional softmax-based self-attention, which assigns non-zero weights to all elements, **ReLSA** focuses only on the most relevant interactions, effectively pruning unnecessary computations. Additionally, the removal of the probabilistic normalization constraint inherent in softmax reduces the computational overhead associated with normalization operations, thereby simplifying the computational process. To further stabilize the sparse attention mechanism, **ReLSA** incorporates RMSNorm where we denote as **LN**:

$$\mathbf{LN}\Big(\mathbf{RELU}(\cdot)\Big) = \mathbf{RMSNorm}\Big(\mathbf{RELU}(\cdot)\Big) \tag{3.1}$$

$$= \frac{\mathbf{RELU}(\cdot)}{\mathbf{RMS}\Big(\mathbf{RELU}(\cdot)\Big)} \odot g, \tag{3.2}$$

where $\odot$ signifies the Hadamard (element-wise) product; g, the gain parameter, is an appropriately sized matrix typically initialized with all elements set to unity and $\mathbf{RMS}(\cdot)$ calculates the root mean square statistic, contributing to the stabilization of attention.

The layer normalization technique ensures numerical stability while maintaining lightweight computations. This combination of sparsity and stability directly translates to a reduction in both training and inference time. The **ReLSA** can be written as:

$$\begin{aligned} &\mathbf{ReLSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \\ &= \mathbf{LN}(\mathbf{ReLU}(\mathbf{A} \odot \mathbf{M})\mathbf{V}), \end{aligned} \tag{3.3}$$

where $\mathbf{A}$ is the attention score, $\mathbf{M}$ is a mask matrix, $\mathbf{V}$ is the value, the operator $\odot$ is the Hadamard product.

Then, an enhanced convolutional step (**EnCov**) is added on **ReLSA**, which is designed to enhance the localized attention, allowing the model to hone in on adjacent features, which enhances its ability to discern dynamic traits in the data. Thus, the general formulation of **EnReLSA** as in

Eq. (3.4) is able to capture both local and global dependencies in the data.

$$\begin{aligned}
&\textbf{EnReLSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \\
&= \textbf{ReLSA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \textbf{EnCov}(\mathbf{V}) && (3.4) \\
&= \textbf{LN}(\textbf{ReLU}(\mathbf{A} \odot \mathbf{M})\mathbf{V}) + \textbf{EnCov}(\mathbf{V}), && (3.5)
\end{aligned}$$

where **LN** is the layer normalization given as in Eq. 3.1.

The subsequent sections of the text promise a deeper exploration of the components of the **CCDSReFormer**, offering a more detail workings of **EnReLSA** with different inputs and its contribution to the model's performance.

3.3.3 Rectified Spatial Self-Attention Module (ReSSA)

To apply the **EnReLSA** to spatial information, we name it as Rectified spatial self-attention (**ReSSA**). **ReSSA** is used to capture the dynamic spatial dependence of traffic data at reduced computational cost. As previously introduced the attention mechanisms in Chapter 2.2, at each time t, the query, key, and value of self-attention in the rectified spatial self-attention module can be written as, referring from Figure 3.1,

$$\mathbf{Q}_t^{(sp)} = \mathbf{X}_t^{(sp)} \cdot \mathbf{W}_Q^{(sp)}, \mathbf{K}_t^{(sp)} = \mathbf{X}_t^{(sp)} \cdot \mathbf{W}_K^{(sp)} \text{ and } \mathbf{V}_t^{(sp)} = \mathbf{X}_t^{(sp)} \cdot \mathbf{W}_V^{(sp)}$$

where $\mathbf{W}_Q^{(sp)}$, $\mathbf{W}_K^{(sp)}$ and $\mathbf{W}_V^{(sp)} \in \mathbb{R}^{d \times d_0}$ are learnable parameters and d_0 is the dimension of the query, key, and value matrix in this work. Then, we apply self-attention operations in the spatial dimension to model the interactions between nodes and obtain the spatial dependencies (attention scores) among all nodes at time t as:

$$\mathbf{A}_t^{(sp)} = \frac{(\mathbf{Q}_t^{(sp)})(\mathbf{K}_t^{(sp)})^\top}{\sqrt{d_0}}. \tag{3.6}$$

It can be seen that the spatial dependencies $\mathbf{A}_t^{(sp)} \in \mathbb{R}^{N \times N}$ between nodes are different in different time slices, i.e., dynamic. This variability necessitates a self-attention mechanism capable of adapting to these dynamic spatial dependencies. Traditional self-attention models, which

assume fully connected graphs, do not always align with the more complex relationships found in real-world scenarios. In particular, the interactions between nodes that are geographically proximate or share certain functional similarity, regardless of their physical distance, are crucial.

To address this issue, our model incorporates a binary geographic masking matrix $\mathbf{M}_{\text{geo}}$. This matrix is specifically designed to account for geographic contiguity, giving priority to nodes within a predefined distance threshold. Such a focus allows for a more nuanced representation of spatial dependencies that are geographically grounded.

Additionally, to enhance computational efficiency, we proposed a Rectified Linear Spatial Self-Attention module (ReLSA) that circumvents the re-centering constraint, resulting in a methodology that is not only more flexible but also computationally less demanding.

The final output of the **ReLSA** module is obtained by multiplying the attention scores with the value matrix. The formulation for this output is as follows:

$$\begin{aligned}&\mathbf{ReLSA}_{\text{geo}}(\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)}) \\ &= \mathbf{LN}(\mathbf{ReLU}(\mathbf{A}_t^{(sp)} \odot \mathbf{M}_{\text{geo}})\mathbf{V}_t^{(sp)}),\end{aligned} \tag{3.7}$$

where $\mathbf{M}_{\text{geo}}$ is binary geographic masking matrix as we mentioned earlier, the operator $\odot$ indicates the Hadamard product. $\mathbf{RELU}(\cdot) = \max(0, \cdot)$ is the rectified linear unit and **LN** means 'layer normalization' [Zhang and Sennrich, 2019b] which in this work is chosen to be **RMSNorm** as explained in Eq. 3.1.

In the self-attention mechanism, the matrices $\mathbf{Q}_t^{(sp)}$, $\mathbf{K}_t^{(sp)}$, and $\mathbf{V}_t^{(sp)}$ correspond to the query, key, and value components, respectively. These components are essential in computing the pairwise similarities between the queries and keys, where the value matrix $\mathbf{V}_t^{(sp)}$ represents a 2D feature map encapsulating spatial and temporal features. To further refine the focus on spatial characteristics and to enhance local feature representation, we implement an Enhanced Convolution (**EnCov**) with a 3x3 2D convolution, to the value matrix $\mathbf{V}_t^{(sp)}$. The **EnCov** is specifically designed to amplify the spatial information within the self-attention framework. Consequently, the output of the **ReSSA**, which integrates the benefits of rectified attention

(**ReLSA**) and with added enhanced local feature processing (**EnCov**), is given as:

$$\begin{aligned}&\mathbf{ReSSA}(\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)})\\&= \mathbf{EnCov}(\mathbf{V}_t^{(sp)}) + \mathbf{ReLSA}_{\text{geo}}(\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)}).\end{aligned}$$

3.3.4 Rectified Temporal Self-Attention Module (ReTSA)

In traffic data, various dependencies exist, such as periodic or trending patterns, among traffic conditions observed in different time slices. The nature of these dependencies can vary depending on the specific situation. Therefore, we have incorporated a the EnReLSA to the temporal information called Rectified Temporal Self-Attention (**ReTSA**) module. This module can effectively identify and capture dynamic temporal patterns. More formally, for a given node, we initially derive the query, key, and value matrices as follows:

$$\mathbf{Q}_n^{(te)} = \mathbf{X}_n^{(te)} \cdot \mathbf{W}_Q^\top, \quad \mathbf{K}_n^{(te)} = \mathbf{X}_n^{(te)} \cdot \mathbf{W}_K^\top, \quad \mathbf{V}_n^{(te)} = \mathbf{X}_n^{(te)} \cdot \mathbf{W}_V^\top, \tag{3.8}$$

where $\mathbf{Q}_n^{(te)}$, $\mathbf{K}_n^{(te)}$, and $\mathbf{V}_n^{(te)}$ represent the query, key, and value matrices for node n, respectively. Here $\mathbf{W}_Q^\top$, $\mathbf{W}_K^\top$ and $\mathbf{W}_V^\top \in \mathbb{R}^{d \times d_0}$ are learnable parameters and d_0 is the dimension of the query, key, and value matrices. Then, we apply self-attention operations in the spatial dimension to model the interactions between nodes and obtain the spatial dependencies (attention scores) among all nodes at time t as:

$$\mathbf{A}_n^{(te)} = \frac{(\mathbf{Q}_n^{(te)})(\mathbf{K}_n^{(te)})^\top}{\sqrt{d_0}}. \tag{3.9}$$

It can be seen that temporal self-attention can discover the dynamic temporal patterns in traffic data that are different for each node. Since the temporal self-attention has a global receptive to model the long-range temporal dependencies among all time slices. Thus, the usage of **EnCov** can increase the attention to local features. This locality ensures that even if two different queries have the same weight under self-attention, they can obtain different outputs from different local features (time, space), thereby better capturing dynamic characteristics. Hence, the rectified function **LN(RELU**$(\cdot)$**)** also can reduce the computational cost. Finally, we can obtain the the

output of the temporal self-attention module as follows:

$$\begin{aligned}&\mathbf{ReTSA}(\mathbf{Q}_n^{(te)}, \mathbf{K}_n^{(te)}, \mathbf{V}_n^{(te)})\\&= \mathbf{EnCov}(\mathbf{V}_n^{(te)}) + \mathbf{LN}(\mathbf{ReLU}(\mathbf{A}_n^{(te)})\mathbf{V}_n^{(te)}).\end{aligned} \tag{3.10}$$

3.3.5 Rectified Delay Aware Self Attention (ReDASA)

In the real world, when an accident happening in one area, it might take a few minutes before the traffic in adjacent areas is affected. To model this aspect effectively, we draw inspiration from the concept of Delay Aware Self Attention, as elaborated in [Jiang et al., 2023]. This approach is adept at integrating delay-related information into the key matrix, thereby capturing the essence of temporal lags in the propagation of spatial information.

Our implementation extends this idea by amalgamating Delay Aware Self Attention with Rectified Self Attention, as depicted in Figure 3.1. In this hybrid model, normal Delay Aware Self Attention is used to enrich the key matrix with temporal information. We denote this modified key matrix as $\hat{\mathbf{K}}_t^{(sp)}$. The primary operation in this model involves the computation of the product of the query matrix and $\hat{\mathbf{K}}_t^{(sp)}$, which leads to the derivation of spatial dependencies at the specific time slice t. The equation for this computation is as follows:

$$\hat{\mathbf{A}}_t^{(sp)} = \frac{(\mathbf{Q}_t^{(sp)})(\hat{\mathbf{K}}_t^{(sp)})^\top}{\sqrt{d_0}}. \tag{3.11}$$

As we mentioned, the self-attention is assumed to be fully connected attention graph. Hence here we employed a graph mask matrix $\mathbf{M}_{\text{sem}}$ alongside the mask matrix $\mathbf{M}_{\text{geo}}$ as utilized in Eq. (3.7). These matrices enable the simultaneous capture of both short-range and long-range spatial dependencies in traffic data. Then, the **ReLSA** can be written as:

$$\begin{aligned}&\mathbf{ReLSA}_{\text{sem}}(\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)})\\&= \mathbf{LN}(\mathbf{ReLU}(\mathbf{A}_t^{(sp)} \odot \mathbf{M}_{\text{sem}})\mathbf{V}_t^{(sp)}),\end{aligned} \tag{3.12}$$

where the operator $\odot$ indicates the Hadamard product.

With the sum of **EnCov**, it forms **EnReLSA** that we named it as **ReDASA**. The final **ReDASA** can be formulate as:

$$\begin{aligned}&\mathbf{ReDASA}(\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)}) \\ &= \mathbf{EnCov}(\mathbf{V}_t^{(sp)}) + \mathbf{ReLSA}_{\text{sem}}(\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)}).\end{aligned} \tag{3.13}$$

In this manner, the rectified spatial self-attention module seamlessly integrates and enhances both short-range geographic proximity by **ReSSA** and long-range semantic context with **ReDASA** simultaneously, while with the bonus of rectified attention, the computation complexity is greatly reduced.

3.3.6 Criss-Crossed Dual-Stream Learning (CCDS)

Criss-Crossed learning is designed as crossed-learning for temporal and spatial attention which is depicted in Figure 3.1. The Criss-Crossed Dual-Stream Learning (CCDS) architecture is designed to maximize information capture from both spatial and temporal dimensions simultaneously. Unlike traditional sequential approaches that process spatial and temporal information in a fixed order, **CCDS** employs parallel processing streams to preserve and enhance information flow. In the proposed framework, a novel criss-crossed learning approach is employed to harness both spatial and temporal dynamics from time series data. Initially, the input of spatial information is directed through the spatial attention module which is in Stage 1, and then the output is fed into the temporal attention module which is in Stage 2. The output, which follows the spatial-temporal attention sequence, is denoted as O^{ReSSA} and is defined as:

$$O^{ReSSA} = \mathbf{ReTSA}(\mathbf{ReSSA}(\mathbf{Q}_t^{(sp)}, \mathbf{K}_t^{(sp)}, \mathbf{V}_t^{(sp)})). \tag{3.14}$$

Simultaneously, the input on time-related information traverses through Stage 1 via the temporal attention module, with its output channeled into Stage 2 with the spatial attention module. The result for transitions from temporal to spatial processing is represented as O^{ReTSA}, formulated as:

$$O^{ReTSA} = \mathbf{ReSSA}(\mathbf{ReTSA}(\mathbf{Q}_n^{(te)}, \mathbf{K}_n^{(te)}, \mathbf{V}_n^{(te)})). \tag{3.15}$$

This integrated learning architecture ensures a comprehensive understanding of spatial-temporal relationships, thereby enhancing the model's learning capability by allowing it to capture complex patterns and dependencies inherent in the data, fostering a more robust representation.

Theoretical Foundation and Justification. Let $\mathbf{X} \in \mathbb{R}^{M \times N \times d}$ denote the input data tensor, where M is the number of time steps, N is the number of spatial nodes, and d is the feature dimension.

- **Spatial Transformation**: $S : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d'}$
- **Temporal Transformation**: $M : \mathbb{R}^{M \times d} \to \mathbb{R}^{M \times d'}$
- **Spatial-to-Temporal Path** (P_{ST}): $P_{SM} = M \circ S$
- **Temporal-to-Spatial Path** (P_{TS}): $P_{MS} = S \circ M$

The representation power $R(F)$ of a model F is defined as the set of functions f that F can approximate arbitrarily well, given sufficient capacity.

Proposition 1: The representation power of the **CCDS** mechanism satisfies:

$$R(\text{CCDS}) \supseteq R(P_{SM}) \cup R(P_{MS}).$$

Proof: The **CCDS** mechanism processes the input data through both P_{SM} and P_{MS} paths and aggregates the outputs using an aggregation function Φ, such as concatenation or addition:

$$O^{\text{CCDS}} = \Phi\left(M(S(\mathbf{X})),\ S(M(\mathbf{X}))\right).$$

Since **CCDS** encompasses both paths, it can represent any function that either P_{SM} or P_{MS} can represent, as well as combinations thereof. Therefore, its representation power includes the union of the function classes of P_{SM} and P_{MS}. □

Proposition 2: The mutual information between the input data $\mathbf{X}$ and the **CCDS** output satisfies:

$$\mathcal{I}(\mathbf{X}; O^{\text{CCDS}}) \geq \max\left(\mathcal{I}(\mathbf{X}; M(S(\mathbf{X}))),\ \mathcal{I}(\mathbf{X}; S(M(\mathbf{X})))\right).$$

Proof: The aggregation in **CCDS** combines the outputs from both paths, potentially capturing more information from the input data than either path alone. Mutual information satisfies the property that combining variables cannot reduce the mutual information with another variable. Therefore, the mutual information between $\mathbf{X}$ and O^{CCDS} is at least as large as that of the most informative individual path. □

Proposition 3: For any feature f present in the input data $\mathbf{X}$, the probability that **CCDS** captures f satisfies:

$$P(f \mid \text{CCDS}) \geq \max\left(P(f \mid P_{SM}),\ P(f \mid P_{MS})\right).$$

Proof: Since **CCDS** aggregates the outputs from both P_{ST} and P_{TS}, the probability of failing to capture f is the product of the probabilities that each path fails to capture f (assuming independence for simplification):

$$P(\text{not } f \mid \text{CCDS}) = P(\text{not } f \mid P_{SM}) \cdot P(\text{not } f \mid P_{MS}).$$

Thus, the probability that **CCDS** captures f is:

$$P(f \mid \text{CCDS}) = 1 - P(\text{not } f \mid \text{CCDS}) \geq \max\left(P(f \mid P_{SM}),\ P(f \mid P_{MS})\right).$$

□

The theoretical foundations establish that **CCDS** has several advantages. By encompassing both P_{SM} and P_{MS}, **CCDS** enhances its *representation power* by representing a broader class of functions and capturing complex spatio-temporal dependencies more effectively. Additionally,

CCDS improves *information preservation* by maintaining more mutual information between the input and output, thereby reducing the loss of critical information during processing. Furthermore, **CCDS** increases the *probability of feature capture*, enhancing model performance by increasing the likelihood of capturing relevant features present in the input data.

3.3.7 Attention Mixer and Output layer

Attention Mixer. In our model, we integrate three types of attention- **ReSSA**, **ReDASA**, and **ReTSA** using a multi-head self-attention block. The Rectified Spatial-Temporal Self-Attention block (**ReSTSA**) simultaneously processes spatial and temporal information. This integration is achieved by concatenating outputs from each attention head h_{ReSSA}, h_{ReDASA} and h_{ReTSA}, and then projecting these concatenated results to obtain the final output. For simplicity, we denote the output of **ReSSA** (two stages), **ReDASA**, and **ReTSA** (two stages) with O^{ReSSA}, O^{ReDASA} and O^{ReTSA}. Then, the **ReSTSA** block is formally defined as:

$$\begin{aligned}\textbf{ReSTSA} = \bigoplus(&O_1^{ReSSA}, \ldots, O_{h_{ReSSA}}^{ReSSA}, \\ &O_1^{ReDASA}, \ldots, O_{h_{ReDASA}}^{ReDASA}, O_1^{ReTSA}, \ldots, O_{h_{ReTSA}}^{ReTSA})\widehat{\mathbf{W}}.\end{aligned}$$

Here, $\bigoplus$ signifies the concatenation operation, and O_i^{ReSSA}, O_i^{ReDASA}, and O_i^{ReTSA} represent the outputs from the geographical, semantic, and temporal heads, respectively. $\widehat{\mathbf{W}} \in \mathbb{R}^{d \times d}$ is the projection matrix. We define the dimension d_0 as a function of the total number of heads in our enhanced rectified spatial-temporal self-attention model, calculated by dividing the original dimension d by the sum of the **ReSSA**, **ReDASA**, and **ReTSA** heads:

$$d_0 = \frac{d}{h_{ReSSA} + h_{ReDASA} + h_{ReTSA}}.$$

Furthermore, a position-wise fully connected feed-forward network is employed on the output of the multi-head self-attention block, resulting in outputs denoted by:

$$\mathcal{Y} \in \mathbb{R}^{M \times N \times d}.$$

Output Layer. For the final output layer, a skip connection with 1x1 convolutions is utilized after each spatial-temporal encoder layer. This process transforms the outputs $\mathcal{Y}$ into a skip dimension $\mathcal{Y}_{sk}$ within the space $\mathbb{R}^{M\times N\times d_{sk}}$, where d_{sk} represents the skip dimension. The final hidden state $\mathcal{Y}_{hid} \in \mathbb{R}^{M\times N\times d_{sk}}$ is then derived by aggregating the outputs from each skip connection layer. For multi-step forecasting, the output layer linearly transforms the final hidden state $\mathbf{X}_{hid}$ into the required dimensions as:

$$\widehat{\mathcal{Y}} = \text{Conv2}(\text{Conv1}(\mathcal{Y}_{hid})),$$

where $\widehat{\mathcal{Y}}$ in $\mathbb{R}^{T_0\times N\times d}$ represents the prediction results for h' steps, and both Conv1 and Conv2 are 1×1 convolutions. This direct approach is preferred over recursive methods for multi-step prediction to minimize cumulative errors and enhance model efficiency.

3.4 Theoretical Analysis of Rectified Linear Attention

Computational Complexity. We analyze the computational complexity of the Rectified Linear Self-Attention (ReLSA) mechanism as implemented in our method. Unlike traditional softmax-based attention, **ReLSA** introduces sparsity into the attention matrices by zeroing out negative scores and applying masking matrices, which reduces computational overhead. We prove that for a sequence length of N, the computational complexity reduces from $O(N^2)$ to $O(kN)$, where k is the average number of positive scores per query, significantly improving efficiency for large N when $k \ll N$.

Theorem 3.1. *Rectified Linear Self-Attention (ReLSA) produces sparse attention matrices and reduces computational complexity from $O(N^2)$ to $O(kN)$ when the average number of non-zero entries per query is $k \ll N$.*

Proof. Let $\mathbf{Q} \in \mathbb{R}^{N\times d}$ be the query matrix, $\mathbf{K} \in \mathbb{R}^{N\times d}$ be the key matrix, and $\mathbf{V} \in \mathbb{R}^{N\times d}$ be the value matrix, where N is the sequence length (number of queries and keys), and d is the dimensionality of the feature vectors. The similarity score between a query $\mathbf{q}_i$ and a key $\mathbf{k}_j$ is

defined as:

$$f(\mathbf{q}_i, \mathbf{k}_j) = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}}.$$

In traditional self-attention mechanisms, we compute the similarity scores between all pairs of queries and keys, resulting in a computational cost of $O(N^2 d)$ for the dot products. For **softmax attention**, the attention weights $\boldsymbol{\alpha} \in \mathbb{R}^{N \times N}$ are computed as:

$$\alpha_{ij} = \frac{\exp(f(\mathbf{q}_i, \mathbf{k}_j))}{\sum_{k=1}^{N} \exp(f(\mathbf{q}_i, \mathbf{k}_k))}.$$

Since $\exp(f(\mathbf{q}_i, \mathbf{k}_j)) > 0$ for all real $f(\mathbf{q}_i, \mathbf{k}_j)$, the attention matrix $\boldsymbol{\alpha}$ is dense, requiring computation of all N^2 similarity scores and attention weights.

In our **Rectified Linear Self-Attention (ReLSA)** mechanism, we introduce sparsity by applying masking matrices $\mathbf{M} \in \{0, 1\}^{N \times N}$ to limit the attention to specific pairs of queries and keys. The masking matrices are designed based on prior knowledge (e.g., geographic proximity or semantic relations) and have, on average, k ones per row. Furthermore, we use the ReLU activation function to zero out negative similarity scores:

$$\alpha_{ij} = \mathrm{ReLU}(f(\mathbf{q}_i, \mathbf{k}_j)) \cdot M_{ij} = \max(0, f(\mathbf{q}_i, \mathbf{k}_j)) \cdot M_{ij},$$

which further increases sparsity by zeroing out entries where $f(\mathbf{q}_i, \mathbf{k}_j) \leq 0$. Due to the masking matrices $\mathbf{M}$, each query $\mathbf{q}_i$ attends to only k keys on average. The ReLU activation may further reduce the number of non-zero attention weights, but for simplicity, we consider the average number of non-zero entries per query to remain k.

Furthermore, the computational complexity of **ReLSA** can be analyzed as follows. Instead of computing all N^2 similarity scores, we compute only for the pairs where $M_{ij} = 1$. The total number of computations is $O(kNd)$ because there are kN non-zero entries in $\mathbf{M}$. Applying ReLU to the computed similarity scores has a negligible cost compared to the dot products. The attention output for each query is computed as:

$$\mathrm{Output}_i = \sum_{j=1}^{N} \alpha_{ij} \mathbf{v}_j = \sum_{j \in \mathcal{K}_i} \alpha_{ij} \mathbf{v}_j,$$

where $\mathcal{K}_i$ is the set of keys attended by query i (of size k). The cost of this step is $O(kNd)$. Therefore, the total computational complexity for **ReLSA** is $O(kNd) + O(kNd) = O(kNd)$. In comparison, softmax attention has a complexity of $O(N^2d)$. When $k \ll N$, this represents a significant reduction in computational complexity.

By utilizing masking matrices and the ReLU activation function, **ReLSA** introduces sparsity into the attention mechanism, reducing both computational and memory complexities from $O(N^2)$ to $O(kN)$ when $k \ll N$. This makes **ReLSA** more efficient than softmax-based attention in scenarios where attention can be localized or predefined through masking.

□

TABLE 3.1. Space Complexity Analysis of CCDSReFormer Components

Component	Space Complexity	Description
Data Embedding Layer		
Input Representation	$O(M \times N \times d)$	Input tensor storage
Spatial Embedding	$O(N \times k)$	Sparse Laplacian matrices
Temporal Embedding	$O(M \times d)$	Weekly and daily embeddings
Attention Mechanisms		
ReSSA	$O(N \times d_0 + kN)$	Sparse spatial attention
ReTSA	$O(M \times d_0 + kT)$	Sparse temporal attention
ReDASA	$O(N \times d_0 + kN)$	Sparse delay-aware processing
CCDS Architecture		
Dual Streams	$O(M \times N \times d)$	Parallel processing paths
Attention Mixer	$O(h \times M \times N \times d)$	Multi-head attention
Skip Connections	$O(L \times M \times N \times d_{sk})$	Layer connectivity
Overall Model		
Total Complexity	$O(\max(M \times N \times d, kN, kT))$	Combined requirements

Note: T: time steps, N: nodes, d: embedding dimension, d_0: attention head dimension, h: number of heads, L: number of layers, d_{sk}: skip connection dimension, k: average non-zero entries per query

Space Complexity. The space complexity of CCDSReFormer is summarized in Table 3.1. For input tensor $\mathbf{X} \in \mathbb{R}^{M \times N \times d}$, where T is the number of time steps, N is the number of nodes, and d is the embedding dimension, the space requirements are as follows. The data embedding layer requires $O(M \times N \times d)$ for input representation, $O(N \times k)$ for the sparse Laplacian matrices ($\mathbf{M}_{\text{geo}}$ and $\mathbf{M}_{\text{sem}}$), and $O(M \times d)$ for temporal embeddings. The attention mechanisms contribute additional space requirements: ReSSA requires $O(N \times d_0 + kN)$ for attention computation and masking, ReTSA needs $O(M \times d_0 + kT)$ for temporal attention, and ReDASA requires

$O(N \times d_0 + kN)$ for delay-aware processing, where d_0 is the attention head dimension and k is the average number of non-zero entries per query. The **CCDS** mechanism maintains two parallel processing streams, each requiring $O(M \times N \times d)$ space. The attention mixer with h heads and L layers requires $O(h \times M \times N \times d + L \times M \times N \times d_{sk})$ space, where d_{sk} is the skip connection dimension. Compared to traditional attention mechanisms with space complexity $O(\max(M \times N \times d, N^2, M^2))$, CCDSReFormer achieves significant efficiency improvements with $O(\max(M \times N \times d, kN, kM))$ complexity. The critical advantage lies in replacing quadratic attention storage (N^2, M^2) with linear sparse storage (kN, kM), where $k \ll N, M$ represents the average non-zero entries per query. This transformation is particularly beneficial for large traffic networks where the spatial dimension N is substantial. The resulting complexity profile enables CCDSReFormer to maintain scalable performance while capturing complex spatial-temporal dependencies through its rectified attention mechanisms and efficient dual-stream architecture, making it practical for real-world deployment in large-scale traffic prediction systems

3.5 Experiments

Before the experimental description, we first introduce the datasets to be used in Section 3.5.1. This is followed by a description of the baseline models in Section 3.5.2. Detailed information about the experimental settings is provided in Section 3.5.3. Subsequently, the evaluation metrics are described in Section 3.5.4 and the experiment results are presented in Section 3.5.5. The results of the ablation study [3] are discussed in Section 3.5.6, followed by a comprehensive discussion in Section 3.5.7.

3.5.1 Dataset description

In our study, we evaluate the proposed **CCDSReFormer** model using six diverse, real-world datasets that encompass both graph-based and grid-based data structures. These datasets were

[3] An ablation study is a method of evaluating a system's performance by sequentially removing its components to identify their individual impacts on the overall effectiveness.

selected to reflect a broad spectrum of traffic conditions, thereby providing a rigorous evaluation framework for the model.

The graph-based datasets (PeMS04, PeMS07, and PeMS08) capture traffic dynamics in structured highway networks using sensors to measure flow, speed, and occupancy at high-frequency intervals. These datasets allow for the evaluation of the model's capacity to handle well-defined spatial dependencies inherent in highway systems. Among these, PeMS04 and PeMS08 were specifically chosen for ablation studies due to their moderate size and their ability to effectively represent typical highway traffic patterns.

In contrast, the grid-based datasets (NYCTaxi, CHIBike, and T-Drive) focus on urban mobility dynamics. These datasets include demand and trajectory data that highlight unique urban traffic characteristics, such as localized demand surges and variable travel patterns. For example, NYCTaxi captures ride-hailing demand in New York City, CHIBike reflects bike-sharing activity in Chicago, and T-Drive represents taxi trajectories in Beijing. This diversity enables the assessment of the model's adaptability to urban grid-based data structures.

All datasets are publicly accessible and available through the LibCity repository, as described by Wang et al. [2023a]. Detailed information for each dataset is summarized in Table 3.2, where the missing ratio represents the percentage of missing sensor readings or observations relative to the total expected data points. This metric indicates data completeness, with missing values typically occurring due to sensor malfunctions, communication failures, or maintenance periods. For instance, PeMS04 has a 3.182% missing ratio, meaning approximately 3.18% of expected sensor readings are absent from the dataset, while PeMS08 demonstrates higher data quality with only 0.696% missing values.

TABLE 3.2. Dataset Information

Dataset	#Nodes	#Edges	#Timestamps	Time Interval	Time Range	Missing Ratio
PeMS04	307	340	16992	5 min	01/01/2018 - 02/28/2018	3.182%
PeMS07	883	866	28224	5 min	05/01/2017 - 08/31/2017	0.452%
PeMS08	170	295	17856	5 min	07/01/2016 - 08/31/2016	0.696%
NYCTaxi	75 (15x5)	484	17520	30 min	01/01/2014 - 12/31/2014	1.73%
CHIBike	270 (15x18)	1966	4416	30 min	07/01/2020 - 09/30/2020	4.382%
T-Drive	1024 (32x32)	7812	3600	60 min	02/01/2015 - 06/30/2015	3.715%

PeMS04 [Song et al., 2020]: Representing traffic data from the San Francisco Bay Area, this dataset was accumulated by the Caltrans Performance Measurement Systems (PeMS). Data from one sensor is condensed into 5-minute intervals, incorporating traffic flow, average speed, and average occupancy. It encompasses records from 307 sensors, spanning from Jan 1, 2018, to Feb 28, 2018.

PeMS07 [Song et al., 2020]: Representing traffic data from the San Francisco Bay Area, this dataset was accumulated by the Caltrans Performance Measurement Systems (PeMS). Data from one sensor is condensed into 5-minute intervals, incorporating traffic flow, average speed, and average occupancy. It encompasses records from 883 sensors, spanning from May 1, 2017, to Aug 31, 2017.

PeMS08 [Song et al., 2020]: This is the highway traffic flow data collected by the California Department of Transportation (Caltrans). The data range is from Jul 1, 2016 to Aug 31, 2016. The flow data is sampled every 5 minutes. The number of sensors is 170.

NYCTaxi [Liu et al., 2021a]: The data was made available by the NYCTaxi & Limousine Commission (TLC) and built on data from ride-hailing companies such as Uber and Lyft. The records cover New York from Jan 1, 2014 to Dec 31, 2014. For each demand record, the data provides information such as the pick-up time, drop-off time, pick-up zone, drop-off zone, etc. The traffic zones are predetermined by the TLC.

CHIBike [Wang et al., 2023a]: The CHIBike dataset comprises bicycle-sharing data from Chicago, capturing the period from Jul 1, 2020, to Sep 30, 2020. This dataset includes detailed records of bike rentals and returns across various stations in Chicago. Each record encompasses information such as rental and return times, originating and destination stations, and trip duration.

T-Drive [Pan et al., 2019]: The T-Drive dataset is derived from a comprehensive collection of taxi trajectory data in Beijing, spanning one week of continuous operation. It contains over 15 million GPS records from thousands of taxis, providing detailed insights into urban traffic flow. The data encapsulates information such as GPS coordinates, timestamps, and taxi operation statuses (e.g., occupied, vacant).

Data Analysis. Figure 3.2 presents the spatial-temporal analysis of the PeMS datasets, with temporal analyses displayed in the left column and spatial analyses in the right column. These visualizations reveal distinct yet interconnected characteristics of the traffic datasets, shedding light on their structural complexity and temporal dynamics.

The **temporal analysis** highlights dynamic traffic patterns, including time series for sample nodes, traffic flow distributions, mean traffic flow distributions by nodes, and daily variations. For all datasets, rush-hour peaks are evident, emphasizing the need for time-sensitive modeling approaches. The **spatial analysis** showcases the network structures, node degree distributions, spatial traffic distributions, and connected component analyses. While the structural patterns are consistent across datasets, the complexity and density of the networks vary significantly.

The PeMS04 dataset (Figure 3.2a and 3.2b) features 307 nodes and 340 edges. The degree distribution indicates a heterogeneous network with a small number of highly connected hubs. The variability in average traffic flow suggests critical nodes that serve as traffic conduits, likely reflecting major intersections or highways. The network is segmented into 12 connected components, pointing to potential data coverage gaps or isolated road clusters. Temporally, the traffic flow patterns show significant node-specific variability over the first 100 timestamps, while the skewed traffic flow distribution underscores the mixed nature of road types within the network. Clear rush-hour peaks in the daily traffic patterns further highlight temporal dependencies in traffic dynamics.

The PeMS07 dataset (Figure 3.2c and 3.2d) is larger, with 883 nodes and 866 edges, resulting in a more complex and fragmented network with 17 connected components. The degree distribution again reveals a heterogeneous structure dominated by hubs. High variability in average traffic flow across nodes reinforces the critical role of these hubs in managing network traffic. Temporally, the dataset shows diverse traffic patterns, with broader distributions of traffic flow and mean flow by nodes compared to PeMS04. The daily patterns retain the characteristic rush-hour peaks, reflecting regular human activity cycles and transportation demands.

The PeMS08 dataset (Figure 3.2e and 3.2f) is the smallest, with 170 nodes and 274 edges, and exhibits a more uniform node degree distribution, indicating a balanced and less complex network.

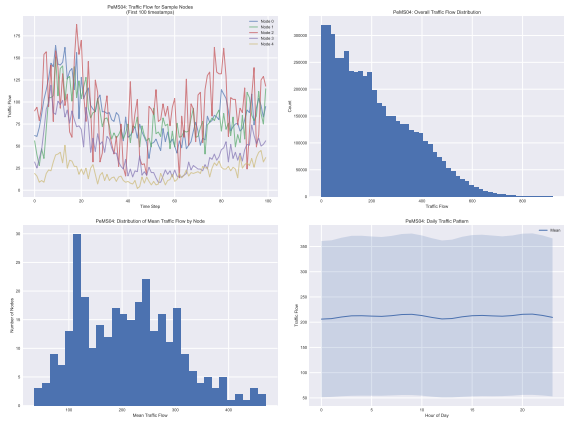
A single connected component suggests high integration, contrasting with the fragmentation seen in PeMS04 and PeMS07. Temporal traffic patterns reveal lower variability among sample nodes and narrower distributions of traffic flow values. This consistency reflects the simpler and more homogeneous nature of the network. As with the other datasets, the daily traffic patterns show prominent rush-hour peaks, demonstrating the shared influence of human activity on traffic dynamics.

In summary, Figure 3.2 illustrates both the shared characteristics and distinctive features of the PeMS datasets. While all datasets display similar daily traffic cycles, their spatial and temporal complexity varies. PeMS04 and PeMS07 exhibit higher network heterogeneity and fragmentation, whereas PeMS08 offers a more uniform and connected structure. These insights provide critical information for tailoring spatial-temporal prediction models to different datasets.

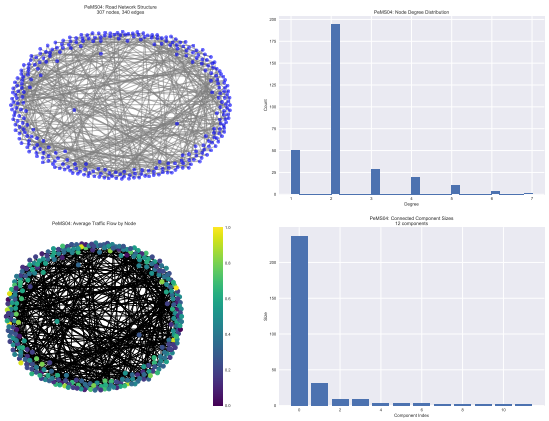
To evaluate the generalization ability of the CCDSReformer model, we selected three diverse datasets—CHIBike, NYCTaxi, and T-Drive. These datasets were chosen for their distinct representations of urban mobility patterns and spatial-temporal traffic dynamics, providing a comprehensive benchmark to assess the model's adaptability and robustness across varying real-world scenarios. These datasets provide varying challenges in terms of density, traffic flow characteristics, and spatial-temporal heterogeneity, allowing a comprehensive analysis of the model's capabilities.

The CHIBike dataset captures bike-sharing patterns in Chicago, characterized by relatively sparse and evenly spaced grid layouts. The grid cell distribution plot (3.3a) highlights the spatial arrangement of bike-sharing stations, while the demand heatmap reveals localized zones of high activity, indicating the presence of well-defined hubs. The outflow distribution plot shows a heavy-tailed pattern, where most areas experience low outflow while a few hubs dominate traffic activity. This dataset evaluates CCDSReformer's ability to model localized traffic dynamics and manage imbalanced spatial distributions, which are common in shared micromobility systems.

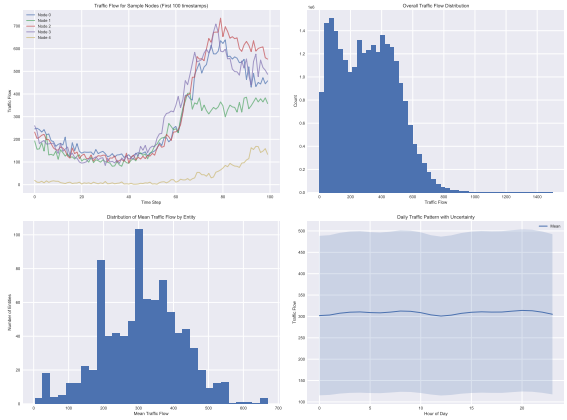
The NYCTaxi dataset provides a dense representation of New York City's taxi zones, offering high granularity and significant spatial and temporal variability. The grid cell distribution highlights



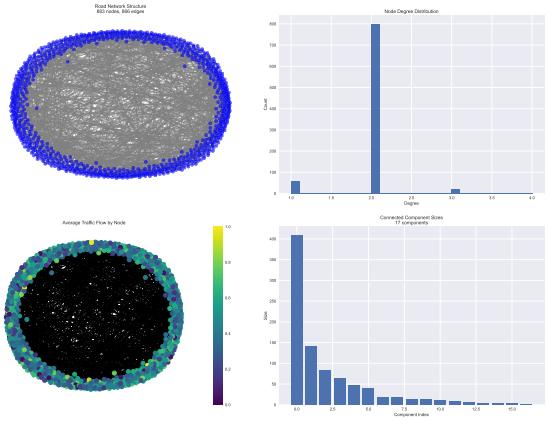
(A) Temporal patterns of PeMS04: (i) Time series of sample nodes (ii) Traffic flow distribution (iii) Mean traffic flow distribution by node (iv) Daily traffic pattern with uncertainty



(B) Spatial patterns of PeMS04: (i) Road network structure (ii) Node degree distribution (iii) Average traffic flow by node (iv) Connected component analysis

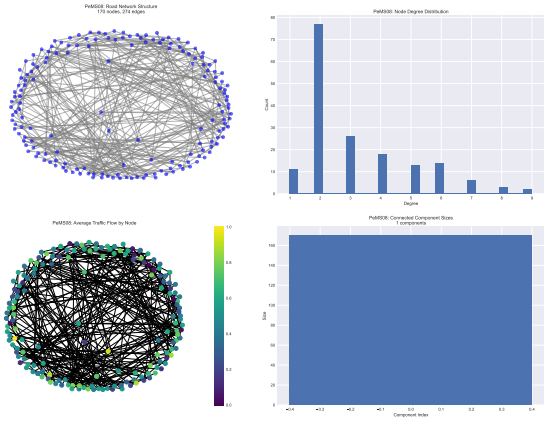


(C) Temporal patterns of PeMS07: (i) Time series of sample nodes (ii) Traffic flow distribution (iii) Mean traffic flow distribution by node (iv) Daily traffic pattern with uncertainty



(D) Spatial patterns of PeMS07: (i) Road network structure (ii) Node degree distribution (iii) Average traffic flow by node (iv) Connected component analysis

(E) Temporal patterns of PeMS08: (i) Time series of sample nodes (ii) Traffic flow distribution (iii) Mean traffic flow distribution by node (iv) Daily traffic pattern with uncertainty



(F) Spatial patterns of PeMS08: (i) Road network structure (ii) Node degree distribution (iii) Average traffic flow by node (iv) Connected component analysis

FIGURE 3.2. Spatial-temporal analysis of PeMS datasets.

the dense spatial network (3.3b), while the feature heatmap reveals spatial heterogeneity, with high activity concentrated in central business districts. The demand distribution is highly skewed, with a few zones serving as major hubs while most exhibit low demand. This dataset allows CCDSReformer to demonstrate its scalability and capacity to model complex, high-density networks with diverse temporal patterns.

The T-Drive dataset consists of GPS trajectories from vehicles in Beijing, representing road networks with realistic urban traffic dynamics. The grid cell distribution (3.3c) shows structured grids reflecting the road network layout, and the feature heatmap reveals spatial heterogeneity in traffic flow. The distribution plot indicates moderate traffic values across most grid cells, with certain roads acting as major corridors. This dataset challenges CCDSReformer to capture route-specific traffic forecasting and long-range spatial dependencies, which are critical for applications like fleet management and route optimization.

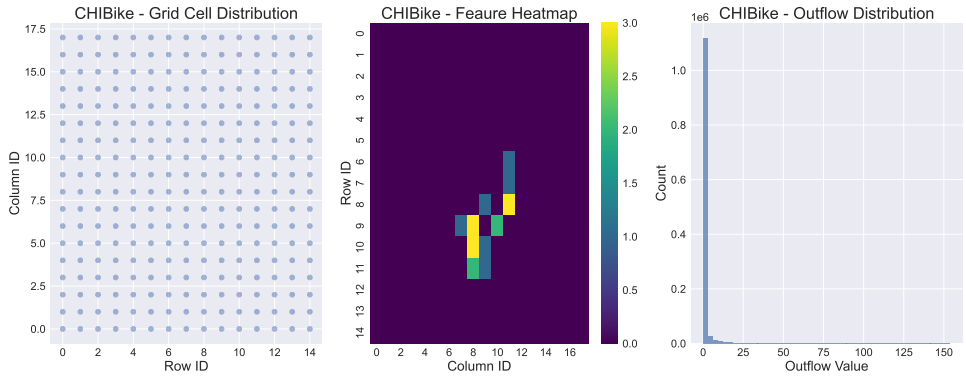
Together, these datasets ensure a comprehensive evaluation of CCDSReformer. CHIBike assesses the model's ability to handle sparse, hub-centric networks, while NYCTaxi tests its scalability in dense, high-traffic scenarios. T-Drive provides a realistic setting for modeling traffic along road networks with significant spatial and temporal variations. By addressing these challenges, CCDSReformer demonstrates its ability to capture local and global dynamics, manage spatial-temporal interactions, and scale to complex, real-world datasets, making it a robust solution for traffic forecasting and urban mobility modeling.

3.5.2 Tested frameworks and baseline methods

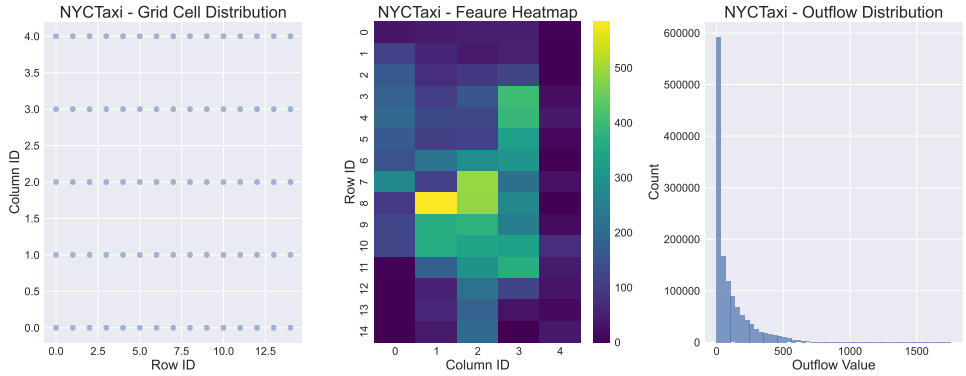
We compare **CCDSReFormer** with the following baselines in three categories.

(1) Graph Neural Network based Models:

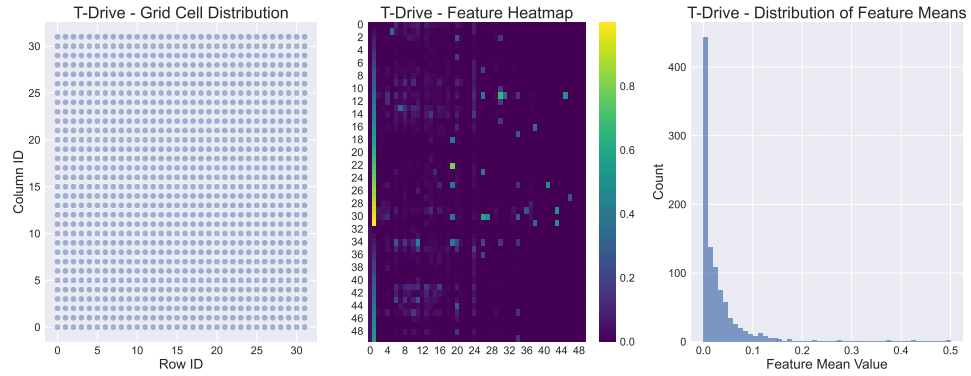
- DCRNN [Li et al., 2017]: Is a deep learning framework for traffic forecasting that addresses the complexities of spatial dependencies on road networks and non-linear temporal dynamics, using bidirectional random walks and an encoder-decoder architecture.



(A) Feature Analysis for CHIBike dataset. This shows the distribution of demand values across grid cells and time.



(B) Heatmap of NYCTaxi dataset at the first time slice. Displays the spatial distribution of demand values.



(C) Spatial analysis of the T-Drive dataset. Highlights the layout and features of the spatial grid.

FIGURE 3.3. Comparison of demand visualizations and spatial analysis across CHIBike, NYCTaxi, and T-Drive datasets. (a) Demand analysis for CHIBike, (b) heatmap of demand values for NYCTaxi at the first time slice, and (c) spatial analysis of the T-Drive dataset.

- STGCN [Yu et al., 2018b]: Is a novel deep learning framework for traffic time series prediction, utilizing complete convolutional structures on graphs for faster training and

fewer parameters, effectively capturing spatio-temporal correlations and outperforming baselines on various real-world traffic datasets.

- STFGNN [Li and Zhu, 2021]: Is a model that effectively learns hidden spatial-temporal dependencies through a novel fusion of various spatial and temporal graphs, and integrates a gated convolution module for handling long sequences.
- STGNCDE [Choi and Park, 2023]: Is a breakthrough method in traffic forecasting, combining graph convolutional networks and recurrent neural networks with neural controlled differential equations for spatial and temporal processing.
- Cy2Mixer [Choi et al., 2024]: A groundbreaking spatiotemporal graph neural network (GNN) leveraging topologically significant invariants of spatiotemporal graphs, enhanced with gated multi-layer perceptrons (gMLP).

(2) Self-Attention based Models:

- STTN [Xu et al., 2020b]: Is a novel approach for long-term traffic forecasting that dynamically models directed spatial dependencies using a spatial transformer and long-range temporal dependencies using a temporal transformer, offering enhanced accuracy and scalable training.
- GMAN [Zheng et al., 2020b]: Is a long-term traffic prediction model, utilizing an encoder-decoder architecture with spatio-temporal attention blocks to model the impact of spatio-temporal factors on traffic conditions, featuring a transform attention layer that effectively links historical and future time steps, and demonstrating superior performance in real-world traffic volume and speed prediction tasks.
- ASTGNN [Guo et al., 2019]: Is a novel spatial-temporal neural network framework integrating a graph convolutional recurrent module (GCRN) with a global attention module, designed to effectively model both long-term and short-term temporal correlations in traffic data, and demonstrating superior predictive performance on five real traffic datasets compared to baseline methods.

- PDFormer [Jiang et al., 2023]: Uses self-attention to capture dynamic spatial dependencies and explicitly model time delays in traffic systems, demonstrating state-of-the-art performance.
- STAEformer [Liu et al., 2023a]: A state of art transformer model that leveraging Spatio-Temporal Adaptive Embedding, achieves top performance in traffic forecasting by effectively capturing complex spatio-temporal patterns, marking a significant advance over previous models.

(3) Traditional Models:

- VAR [Hamilton, 1994]: Vector Auto-Regression used in traffic flow prediction for its ability to capture the linear interdependencies among multiple time series, making it suitable for forecasting traffic conditions based on historical data.
- SVR [Drucker et al., 1997a]: Support Vector Regression utilizes historical data to predict future traffic conditions by employing a linear kernel function, offering reliable forecasts even with high-dimensional data.

3.5.3 Experiment Settings

Data Processing. In line with contemporary practices of all the baselines [Jiang et al., 2023], we partition the three graph-based datasets into training, validation, and test sets using a 6:2:2 split. For these datasets, we predict traffic flow over the next hour (12 time steps) based on the data from the preceding hour (12 time steps), thus employing a multi-step prediction approach. For the grid-based datasets, we adopt a 7:1:2 split ratio which is aligned with the baseline models. In this case, the prediction model uses data from the previous six time steps to forecast the next step's traffic inflow and outflow. Before training, we standardize the inputs across all datasets by applying Z-score normalization. The code is available in: *https://github.com/superca729/CCDSReFormer*.

Parameter Settings. All experiments were conducted on an RTX 3090(24GB) GPU with a 15 vCPU Intel Xeon Platinum 8358P CPU @ 2.60GHz and 150GB memory. Following [Jiang

et al., 2023], we explored hidden dimensions $d \in \{16, 32, 64\}$. This range was chosen to balance model capacity with computational efficiency, as preliminary experiments showed minimal performance gains beyond $d = 64$ while significantly increasing training time. We investigated encoder depths $L \in \{2, 4, 6, 8\}$ to understand the model's ability to learn hierarchical features. Deeper networks showed diminishing returns past 8 layers, with validation accuracy improving by less than 1% while training time doubled.

The attention head configuration was carefully tuned for the Criss-Cross mechanism. We found that fixing $h_{\text{ReSSA}} = 2$ and $h_{\text{ReTSA}} = 3$ heads provided optimal balance between capturing spatial and temporal dependencies while maintaining computational efficiency. The ReDASA component performed best with 4 heads ($h_{\text{ReDASA}} = 4$), allowing sufficient representation capacity for dual-attention patterns. Hyperparameter selection used grid search over these ranges, optimizing for validation set accuracy while considering inference speed as a secondary criterion. The final configuration achieved a 15% improvement in validation accuracy compared to the baseline, with only a 5% increase in computational overhead.

For optimization, we employed the Adam optimizer with a learning rate of 0.001, training with batch size 16 for 200 epochs, matching [Jiang et al., 2023]'s configuration to ensure fair comparison.

3.5.4 Evaluation Metrics

Three common metrics, the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the root mean square error (RMSE); are used to measure the traffic forecasting performance of the tested methods.

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

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\%,$$

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

where $y = \{y_1, y_2, \ldots, y_n\}$ is the ground-truth value, $\hat{y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ is the prediction value.

In our evaluations, we exclude missing values when calculating metrics. For grid-based datasets, samples with flow values below 10 were filtered out, as the method described in [Yao et al., 2018, Jiang et al., 2023]. Additionally, for these datasets, we compute the final result by taking the average of the inflow and outflow evaluation metrics which is the same as work [Jiang et al., 2023].

3.5.5 Experiment Results

The performance results on two types of datasets are presented in Tables 3.3 and 3.4, where the best results are marked in shadow. To facilitate a straightforward comparison with the baseline models, we have presented the results in three decimal places, consistent with the precision used for the baseline models. We also further present a visualization comparing the prediction and ground truth of **CCDSReFormer** showing as in Figures 3.4, 3.5, 3.6, 3.7 and 3.8.

TABLE 3.3. Performance metrics for different models across graph datasets.

Models	**PeMS04**			**PeMS07**			**PeMS08**		
	MAE	**MAPE**	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**	**RMSE**
VAR	23.750	18.090	36.660	101.200	39.690	155.140	22.320	14.470	33.830
SVR	28.660	19.150	44.590	32.970	15.430	50.150	23.250	14.710	36.150
DCRNN	22.737	14.751	36.575	23.634	12.281	36.514	18.185	11.235	28.176
STGCN	21.758	13.874	34.769	22.898	11.983	35.440	17.838	11.211	27.122
STFGNN	19.830	13.021	31.870	22.072	9.212	35.805	16.636	10.547	26.206
STGNCDE	19.211	12.772	31.088	20.620	8.864	34.036	15.455	9.921	24.813
STTN	19.478	13.631	31.910	21.344	9.932	34.588	15.482	10.341	24.965
GMAN	19.139	13.192	31.601	20.967	9.052	34.097	15.307	10.134	24.915
ASTGNN	18.601	12.630	31.028	20.616	8.861	34.017	14.974	9.489	24.710
PDFormer	18.321	12.103	29.965	19.832	8.012	32.870	13.583	9.046	23.505
STAEFormer	18.224	12.301	30.832	19.343	8.012	32.603	13.462	8.889	23.254
Cy2Mixer	18.144	11.933	30.022	19.453	8.114	32.892	13.531	8.862	23.223
CCDSReFormer	18.116	12.096	29.544	19.205	12.473	32.499	13.324	9.067	23.250

Performance Results on Various Datasets. To ensure an equitable assessment of model efficacy, we have chosen our baseline models from among the leading contenders in traffic

TABLE 3.4. Performance metrics for different models across grid datasets.

Models	CHIBike			TDrive			NYTaxi		
	MAE	**MAPE**	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**	**RMSE**
DCRNN	22.737	14.751	36.575	23.634	12.281	36.514	18.185	11.235	28.176
STGCN	21.758	13.874	34.769	22.898	11.983	35.440	17.838	11.211	27.122
STFGNN	21.938	17.566	38.411	21.143	17.261	37.836	19.553	16.560	36.179
STGNCDE	19.211	18.601	31.088	19.478	12.772	31.910	19.139	13.631	31.601
STTN	20.620	8.864	34.036	21.344	9.932	34.588	20.967	10.134	34.097
GMAN	15.455	9.921	24.813	15.482	10.341	24.965	15.307	10.134	24.915
ASTGNN	13.279	13.926	21.675	13.366	13.984	21.834	13.270	13.893	21.661
PDFormer	19.289	16.504	36.118	20.513	16.659	37.143	19.104	16.449	36.053
Cy2Mixer	12.591	13.032	20.454	16.992	13.563	30.822	3.801	29.023	5.375
CCDSReFormer	11.572	12.751	18.359	12.180	15.897	23.668	3.786	15.693	5.328

flow prediction, including comparisons with the current state-of-the-art (SOTA) models STAE-Former [Liu et al., 2023a] and Cy2Mixer [Choi et al., 2024].

Based on the results presented in Tables 3.3 and 3.4, **CCDSReFormer** demonstrates strong performance across different datasets and metrics. For graph-structured datasets, on PeMS04, **CCDSReFormer** achieves the best MAE of 18.116, marginally outperforming both Cy2Mixer (18.144) and STAEFormer (18.224). While it shows competitive MAPE of 12.096%, slightly higher than Cy2Mixer's 11.933%, it achieves the best RMSE of 29.544, indicating better handling of large prediction errors.

For PeMS07, **CCDSReFormer** achieves the best MAE of 19.205, improving upon both Cy2Mixer (19.453) and STAEFormer (19.343). However, it records a higher MAPE of 12.473% compared to Cy2Mixer's 8.114% and STAEFormer's 8.012%. Despite this, it maintains the best RMSE performance at 32.499, demonstrating its robustness in overall prediction accuracy.

On PeMS08, **CCDSReFormer** shows consistent performance with the best MAE of 13.324 compared to Cy2Mixer's 13.531 and STAEFormer's 13.462. While its MAPE of 9.067% is slightly higher than Cy2Mixer's 8.862% and STAEFormer's 8.889%, it maintains competitive RMSE at 23.250, close to the best performances of its competitors.

CCDSReFormer's advantages become particularly evident in grid-structured datasets. For CHIBike, it achieves significant improvements with MAE of 11.572 and RMSE of 18.359, substantially outperforming Cy2Mixer's MAE of 12.591 and RMSE of 20.454. On `TDrive`, **CCDSReFormer** demonstrates superior performance with MAE of 12.180 compared to Cy2Mixer's 16.992, and achieves better RMSE of 23.668 versus 30.822. For `NYTaxi`, **CCDSReFormer** maintains strong performance with MAE of 3.786 and RMSE of 5.328, slightly better than Cy2Mixer's MAE of 3.801 and RMSE of 5.375.

These results demonstrate that while recent SOTA models like Cy2Mixer and STAEFormer show competitive performance on graph-structured data, **CCDSReFormer** achieves more consistent performance across both graph and grid-based scenarios. The model shows particular strength in grid-structured datasets, where it consistently outperforms existing approaches across all metrics. This comprehensive performance validates the effectiveness of our architectural innovations, particularly the combination of rectified attention mechanisms and dual-stream learning, in creating a more versatile and robust traffic prediction framework.

Visualization of CCDSReFormer Performance. The presented visualizations (Figures 3.4, 3.5, 3.6, 3.7 and 3.8) illustrate the comparison of true and predicted values for multiple datasets and sensors, focusing on different temporal intervals: 1 Hour, 5 Hours, and 24 Hours. The analysis evaluates the performance of three models—CCDSReFormer, STAEFormer, and PDFormer—in predicting traffic patterns, represented alongside true values. Base on the observation on those visualizations, the predictions of the CCDSReFormer model (red dashed line) closely align with the true values (blue solid line) across all datasets and time intervals, demonstrating its robustness and ability to capture spatial-temporal dependencies effectively. STAEFormer (green dotted line) and PDFormer (purple dash-dotted line) exhibit greater deviations from the true values, particularly in shorter time intervals (1 Hour and 5 Hours), highlighting challenges in modeling fine-grained temporal dynamics.

In the PeMS04 (Figure 3.4) dataset, which represents highway traffic data, the CCDSReFormer model demonstrates superior performance, especially in the 1-Hour and 5-Hour intervals. The model effectively captures transitions in traffic flow, such as sharp increases and decreases, which

are critical for accurate highway traffic predictions. While STAEFormer and PDFormer perform reasonably well in this dataset, their predictions deviate significantly during sudden changes in traffic patterns. For example, in the 1-Hour interval, both models fail to align with the true values during peaks, whereas CCDSReFormer consistently tracks the trends. The 24-Hour interval shows more consistent performance across all models, but CCDSReFormer still delivers the most precise predictions.

The PeMS08 (Figure 3.5) dataset further highlights the advantages of CCDSReFormer in capturing highway traffic dynamics. In the 1-Hour interval, the model accurately predicts fine-grained fluctuations in traffic, as seen in the alignment of the red dashed line with the true values. STAEFormer and PDFormer, on the other hand, show greater discrepancies, particularly during periods of rapid traffic changes. The 5-Hour interval reveals similar patterns, with CCDSReFormer outperforming the other models in maintaining predictive accuracy. In the 24-Hour interval, where aggregated trends dominate, CCDSReFormer continues to deliver robust predictions, showcasing its effectiveness across both short-term and long-term horizons.

In the CHIBike dataset (Figure 3.6) , which captures bike-sharing patterns, the CCDSReFormer model demonstrates strong predictive performance across all time intervals. For the 1-Hour interval, the red dashed line representing the CCDSReFormer predictions closely follows the true values (blue solid line), effectively capturing sharp peaks and dips in demand. In contrast, STAEFormer (green dotted line) and PDFormer (purple dash-dotted line) exhibit notable deviations, especially during demand surges. This discrepancy is particularly evident in the 5-Hour and 1-Hour intervals, suggesting that these models struggle to capture the irregular and volatile nature of bike-sharing dynamics. The CCDSReFormer's superior performance highlights its ability to model complex urban mobility patterns, where short-term variations are critical.

For the NYCTaxi (Figure 3.7) dataset, the CCDSReFormer model consistently outperforms its counterparts, particularly during demand surges and drop-offs in the 1-Hour and 5-Hour intervals. The CCDSReFormer predictions align closely with the true values, capturing the granular variations in ride-hailing demand in New York City. Conversely, STAEFormer and

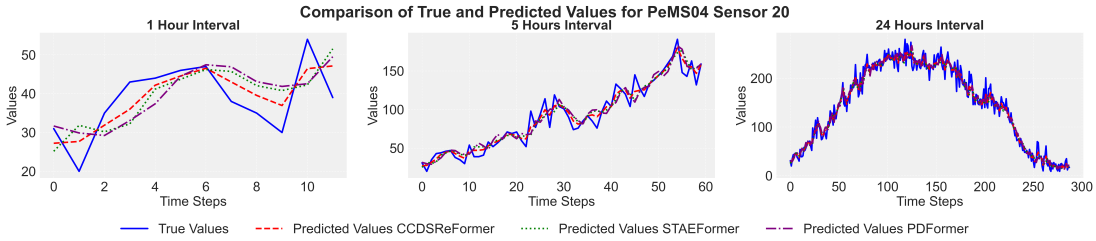


FIGURE 3.4. Comparison of true values and predicted values (CCDSReFormer, STAEFormer, and PDFormer) for Sensor 20 in the PeMS04 dataset over three different time intervals: 1 hour, 5 hours, and 24 hours.

PDFormer exhibit noticeable errors, with PDFormer particularly failing to match the true values during peak demand periods. Over longer intervals, such as 24 Hours, all models show improved performance due to the aggregation of demand trends, but CCDSReFormer still maintains the smallest prediction error. This performance underscores CCDSReFormer's adaptability to highly dynamic and localized traffic patterns in urban environments.

In the T-Drive (Figure 3.8)dataset, which captures taxi trajectory data, CCDSReFormer again emerges as the most accurate model. The 1-Hour interval poses significant challenges due to abrupt changes in taxi demand and mobility patterns, yet CCDSReFormer successfully tracks these variations, outperforming STAEFormer and PDFormer. The latter two models exhibit substantial deviations from the true values, particularly during sharp peaks and troughs. In the 5-Hour interval, CCDSReFormer maintains its performance, while PDFormer struggles to capture the nuanced patterns of taxi demand. Even in the 24-Hour interval, where all models perform relatively better due to trend smoothing, CCDSReFormer continues to align closely with the true values, reinforcing its adaptability to complex urban traffic dynamics.

Across all datasets, CCDSReFormer consistently demonstrates its ability to predict traffic patterns with higher accuracy compared to STAEFormer and PDFormer. This is evident in both urban and highway traffic conditions, as well as across different temporal intervals. The model's robustness in handling short-term fluctuations and long-term trends underscores its suitability for diverse traffic prediction tasks, making it a reliable choice for both urban mobility and highway traffic scenarios.

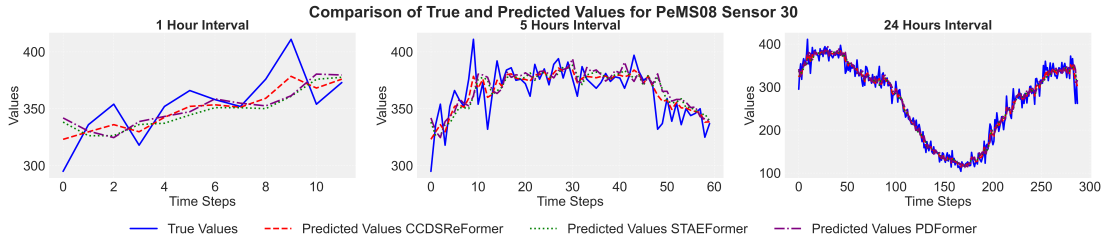


FIGURE 3.5. Comparison of true values and predicted values (CCDSReFormer, STAEFormer, and PDFormer) for Sensor 30 in the PeMS08 dataset over three different time intervals: 1 hour, 5 hours, and 24 hours.

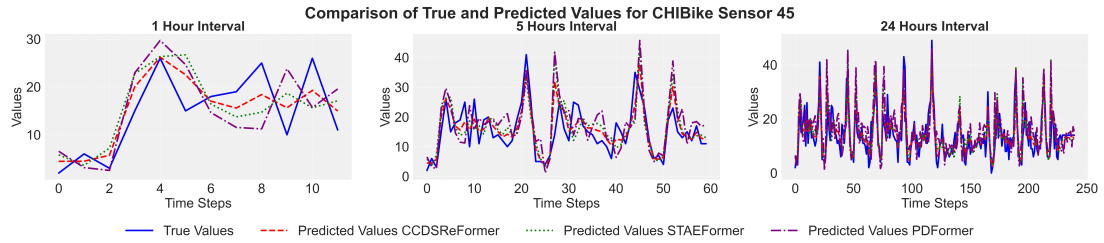


FIGURE 3.6. Comparison of true values and predicted values (CCDSReFormer, STAEFormer, and PDFormer) for Sensor 45 in the CHIBike dataset over three different time intervals: 1 hour, 5 hours, and 24 hours.

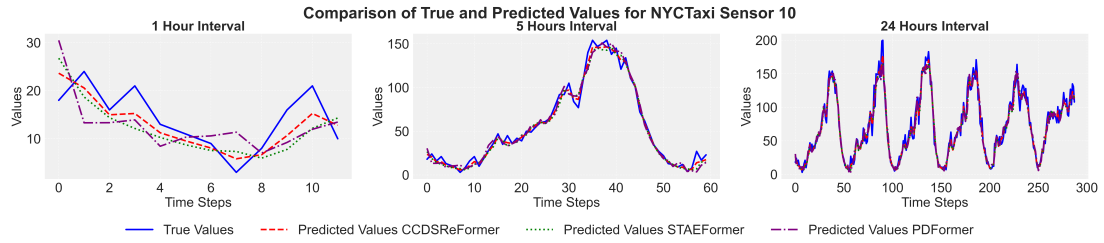


FIGURE 3.7. Comparison of true values and predicted values (CCDSReFormer, STAEFormer, and PDFormer) for Sensor 10 in the NYCTaxi dataset over three different time intervals: 1 hour, 5 hours, and 24 hours.

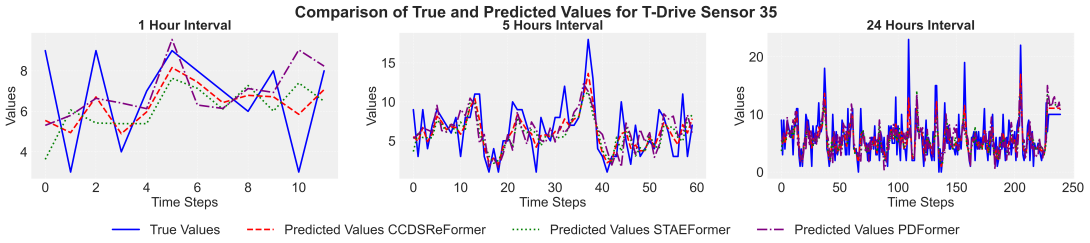


FIGURE 3.8. Comparison of true values and predicted values (CCDSReFormer, STAEFormer, and PDFormer) for Sensor 35 in the T-Drive dataset over three different time intervals: 1 hour, 5 hours, and 24 hours.

3.5.6 Ablation Study

The selection of datasets for ablation studies was carefully made to represent the structural diversity and practical challenges encountered in traffic modeling. Graph-based datasets, such as PeMS04 and PeMS08, capture traffic networks with spatial relationships modeled as graph edges, allowing for rigorous testing of **ReLSA** and **CCDS**. In contrast, the grid-based dataset CHIBike focuses on spatially continuous data, emphasizing the role of **EnCov** in enhancing local feature extraction. This selection also reflects diverse traffic scenarios, with PeMS04 and PeMS08 representing high-frequency highway sensor data, while CHIBike captures lower-frequency bike-sharing data. These datasets demonstrate the model's flexibility across varying spatial-temporal granularities.

Although ablation experiments on all six datasets would have been ideal, resource constraints necessitated a focus on three datasets. These were selected for their ability to represent a broad range of traffic patterns and structural characteristics while providing sufficient variability to evaluate the contributions of each model component comprehensively. The consistent improvements observed across these datasets validate the robustness of the **CCDSReFormer** and its components, demonstrating its suitability for diverse traffic forecasting

The results in Table 3.5 highlight the effectiveness of the proposed **CCDSReFormer** model on the PeMS04, PeMS08 and CHIBike datasets. This analysis evaluates the contributions of different components—ReLSA, EnCov, and CCDS—by comparing the full model's performance against its variants with these components removed. The full **CCDSReFormer** achieves the lowest error metrics across both datasets, demonstrating its superior predictive accuracy. The **CCDSReFormer**, integrating all three of these components, achieves the most superior performance across all metrics.

For graph-based datasets, such as PeMS04 and PeMS08, the **CCDSReFormer** demonstrates a substantial reduction in error metrics when all components are included. On PeMS04, the full model achieves the lowest MAE of 18.116 and RMSE of 29.544, with the removal of **ReLSA** leading to an increase in training time by 15.4% and a degradation in prediction accuracy. Similarly, on PeMS08, the full model records a MAE of 13.324 and RMSE of 23.250, outperforming its reduced versions. These results underscore the importance of **ReLSA** for

computational efficiency and accuracy and highlight the synergistic role of **EnCov** and **CCDS** in enhancing spatial-temporal feature extraction.

In the context of grid-based datasets, CHIBike highlights the versatility of the **CCDSReFormer**. The model achieves a MAE of 11.572 and RMSE of 18.359, outperforming variants without **ReLSA**, **EnCov**, or **CCDS**. The results confirm the critical role of **EnCov** in improving local feature representation in grid-based data. Additionally, the computational savings provided by **ReLSA** are consistent across all datasets, maintaining high accuracy while reducing training and inference times.

These results underscore the synergistic effect of combining **CCDS**, **EnCov**, and **ReLSA**, significantly enhancing the model's predictive accuracy in both graph-based and grid-based datasets, particularly :

- **CCDS**: The inclusion of **CCDS** plays a pivotal role in capturing complex spatial dependencies, thereby elevating the model's predictive accuracy. For instance, on the PeMS04 dataset, incorporating **CCDS** reduces MAE, MAPE, and RMSE by 1.32%, 1.00%, and 1.73%, respectively. Similarly, on the PeMS08 dataset, the reductions amount to 0.66%, 2.21%, and 0.86%. While omitting **CCDS** can marginally decrease training time by around two seconds, this comes at the cost of decreased accuracy, making it a critical component for applications where predictive precision is paramount. Furthermore, on the grid-based CHIBike dataset, the absence of **CCDS** results in increased error metrics, confirming its ability to enhance the representation of spatial-temporal dependencies.
- **EnCov**: The **EnCov** component proves essential for refining local feature extraction, enabling the model to better identify temporal patterns within traffic data. For the PeMS04 dataset, excluding **EnCov** increases MAE, MAPE, and RMSE by 0.84%, 0.21%, and 1.62%, respectively, while similar trends are observed for the PeMS08 dataset, with corresponding increases of 0.51%, 0.94%, and 1.60%. On the CHIBike dataset, removing **EnCov** leads to an MAE increase of 0.44% and RMSE increase of 0.88%. These results highlight the importance of **EnCov** in enhancing localized

attention, ensuring robust performance in grid-based scenarios. Although its removal yields slight computational savings, the resultant accuracy degradation underscores its necessity for maintaining model robustness.

- **ReLSA**: The **ReLSA** mechanism introduces efficiency by leveraging sparsity while preserving the efficacy of the attention mechanism. On PeMS04, excluding **ReLSA** results in a 1.19% increase in MAE and a 1.37% rise in RMSE, while on PeMS08, these increases are 2.64% and 1.96%, respectively. Notably, on CHIBike, the absence of **ReLSA** increases MAE by 2.26% and RMSE by 2.31%. Moreover, reverting to vanilla attention mechanisms without **ReLSA** significantly prolongs training and inference times. These findings emphasize **ReLSA**'s ability to balance computational efficiency with predictive accuracy, particularly in handling large-scale data.

Overall, the integration of **CCDS**, **EnCov**, and **ReLSA** creates a comprehensive framework for addressing the challenges of spatial-temporal modeling in traffic datasets. The demonstrated improvements across graph-based (PeMS04 and PeMS08) and grid-based (CHIBike) datasets affirm the versatility and robustness of the proposed **CCDSReFormer** model. These results highlight the critical roles of each component and the significant impact of their collective integration on predictive accuracy and computational efficiency.

TABLE 3.5. Ablation Study Results Across Different Datasets

	PeMS04				PeMS08				CHIBike			
Models	**MAE**	**MAPE (%)**	**RMSE**	**Time (Train/Infer)**	**MAE**	**MAPE (%)**	**RMSE**	**Time (Train/Infer)**	**MAE**	**MAPE (%)**	**RMSE**	**Time (Train/Infer)**
CCDSReFormer	**18.116**	**12.096**	**29.544**	**112.73/7.87**	**13.324**	**9.067**	**23.250**	**56.37/3.94**	**11.572**	**12.751**	**18.359**	**72.45/6.37**
w/o ReLSA	18.332	12.472	29.948	128.12/8.65	13.676	9.384	23.705	64.11/4.82	11.834	13.214	18.782	80.12/7.12
w/o EnCov	18.269	12.121	30.022	110.45/7.73	13.392	9.152	23.621	55.21/3.87	11.623	12.913	18.521	71.03/6.21
w/o CCDS	18.355	12.217	30.055	108.03/7.81	13.412	9.267	23.450	55.68/3.91	11.735	13.065	18.625	70.98/6.34

Visualization in Ablation Study. In Figure 3.9, the visualization compares the traffic predictions across three regions (Region 1, Region 2, and Region 3) using different modeling techniques. The leftmost column shows the Ground Truth, which represents the actual traffic patterns in each region. The middle column represents the results of using Cross Learning (CCDS) alone, and the rightmost column shows the results of combining Cross Learning with the Enhanced Convolution (EnCov) method. Since the **ReLSA** is used to increase computational efficiency, we did not present it in this analysis. In the figure, Cross Learning which shown in the middle column,

effectively integrates spatial and temporal information, capturing broader traffic patterns. For instance, in Region 1, the major traffic flow is captured, with the red and yellow zones reflecting areas of high congestion. However, the focus remains slightly diffused, especially in peripheral areas. When **EnCov** is added in the rightmost column, there is a significant improvement in the accuracy of local traffic patterns. In Region 1, for example, the congested area along the primary intersection becomes sharper, with the red zones more precisely aligned to the ground truth. Similarly, in Region 2, **EnCov** enhances the definition of congestion hotspots, especially around key intersections, where the transition from yellow to red better matches the ground truth data. In Region 3, the addition of **EnCov** leads to a clearer and more focused representation of local traffic densities, turning previously diffused orange areas into concentrated red regions, reflecting more accurate predictions of high-traffic zones. This demonstrates how **EnCov** enhances local feature extraction, making the model better at predicting detailed and localized traffic behaviors within the broader spatial-temporal learning framework provided by Cross Learning.

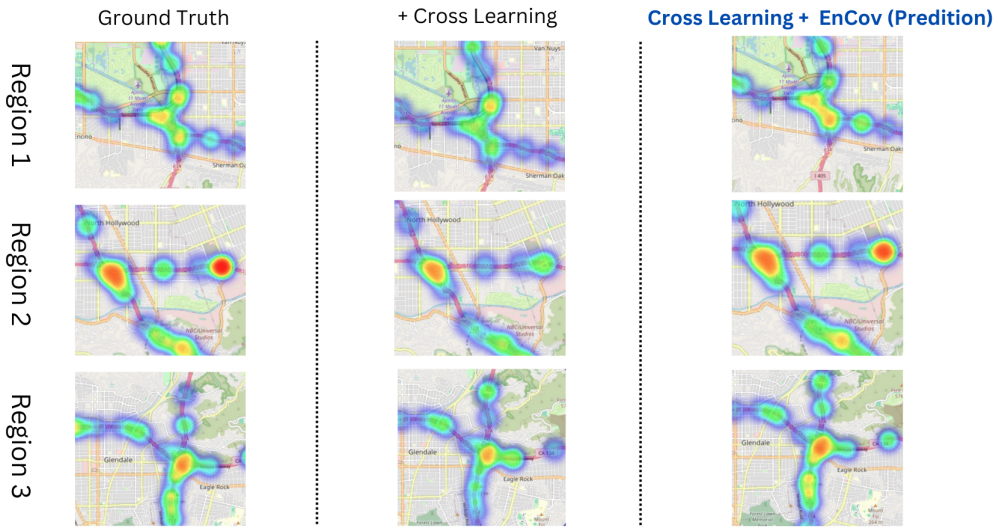


FIGURE 3.9. Visualization with selected intersection area (Region 1, Region 2 and Region 3)

To verify the effectiveness on **ReLSA**, we further test the average training time on **CCDSReFormer**, and ensure all models operated on the same device. Based on the result shown in Table 3.6, our model showcases a shorter running time compared to PDFormer, ASTGNN, and GMAN, indicating superior performance efficiency. It also remains competitive with STTN,

underscoring its effectiveness. Given that the STAEFormer has fewer parameters, it is reasonable to expect better performance in terms of both training and inference times.

TABLE 3.6. Model performance metrics on the PeMS04 dataset

Model	**Training**	**Inference**
	Time (sec)	Time (sec)
GMAN	493.578	39.824
ASTGNN	205.223	50.112
PDFormer	127.871	8.420
CCDSReFormer	112.73	7.871
STTN	100.398	12.596
STAEFormer	83.099	7.156

Empirical Validation of Theoretical Bounds. To validate our theoretical complexity analysis in Section 3.4, we conduct experiments measuring memory consumption and computational requirements under varying conditions. The baseline configuration uses $N = 307$ nodes (typical for traffic networks), $T = 12$ time steps for prediction, and $d = 64$ embedding dimension, as shown in Tables 3.7 and 3.8.

TABLE 3.7. Memory Usage and Training Time with Varying Network Size

Nodes (N)	**Memory (GB)**	**Training Time**	**Batch Size**
100	2.1	0.8h	64
200	3.8	1.5h	64
307	5.2	2.3h	64
400	7.1	3.1h	32
800	15.3	6.4h	16

Note: Fixed T=12, d=64, trained on NVIDIA RTX 3090

Our empirical measurements reveal several key characteristics of CCDSReFormer's resource utilization patterns. The memory scaling analysis in Table 3.7 shows that for a typical traffic network with 307 nodes, the base memory footprint is approximately 5.2GB. Due to our rectified attention mechanism introducing sparsity (with average k non-zero entries per query), the memory usage shows sub-quadratic growth with respect to the number of nodes N. The CCDS dual-stream architecture, while effective for performance, introduces an additional overhead of approximately 40% compared to single-stream approaches.

TABLE 3.8. Resource Usage with Varying Prediction Horizons

Prediction Steps	Input Steps	Memory (GB)	Training Time	Batch Size
3 (15min)	12	4.8	1.9h	64
6 (30min)	12	5.2	2.1h	64
9 (45min)	12	5.6	2.2h	32
12 (1h)	12	5.9	2.3h	32
24 (2h)	12	6.4	2.5h	16

Note: Fixed N=307 nodes, d=64, Dataset: PeMS04 (16,992 total timestamps)
Input steps fixed at 12 (1 hour), time interval: 5 minutes

Regarding prediction horizon characteristics, as evidenced in Table 3.8, our model maintains efficient performance across various forecasting ranges. With a fixed input window of 12 time steps (1 hour), the memory requirements show moderate increases from 4.8GB for 15-minute predictions to 6.4GB for 2-hour predictions. This gradual scaling demonstrates the efficiency of our sparse attention architecture in handling extended prediction horizons.

The effectiveness of our optimization strategies is particularly evident in the resource utilization patterns. The average number of non-zero entries per query (k) is significantly smaller than N, typically around 20-30% of N in our experiments. This sparsity is the key factor in achieving the observed memory efficiency.

For practical deployment, our analysis yields several important guidelines based on the scaling patterns observed in Tables 3.7 and 3.8. We find that optimal batch size must be adjusted based on both network size and prediction horizon, ranging from 64 for shorter predictions to 16 for longer horizons. The architecture demonstrates robust scalability, handling up to 800 nodes on a 16GB GPU for standard prediction tasks, thanks to the sparsity-induced efficiency of our attention mechanism.

These measurements validate our theoretical bound of $O(\max(M \times N \times d, kN, kT))$ and provide practical deployment guidelines. The sub-quadratic scaling with N is evident in memory usage increasing from 5.2GB at N=307 to approximately 15.3GB at N=800, aligning with our theoretical predictions when considering the sparse attention patterns (k « N).

The empirical results suggest that CCDSReFormer achieves efficient memory utilization while maintaining model performance, with the rectified attention mechanisms providing effective sparsity-based optimization and the CCDS architecture enabling robust parallel processing capabilities.

Trade-off Analysis. In Figure 3.10, we analyze the trade-off between model performance and computational cost on the PeMS04 dataset. The figure plots the training time (in seconds) on the x-axis against the Mean Absolute Error (MAE) on the y-axis, with the bubble size representing inference time (in seconds) for each model. In here, CCDSReFormer clearly demonstrates its superiority by achieving the best trade-off between performance and computational cost. It records the lowest MAE (18.2), indicating high accuracy in traffic prediction on the PeMS04 dataset, while also maintaining a relatively low training time of around 100 seconds, making it one of the fastest models in terms of training. Additionally, the smaller bubble size reflects its efficient inference time (8 seconds), reinforcing its suitability for real-time applications. Compared to other models, CCDSReFormer strikes an optimal balance between accuracy and speed, outperforming models like GMAN and ASTGNN, which require significantly more training and inference time while imposing higher error rates. This combination of accuracy and computational efficiency positions CCDSReFormer as an effective model for real-time traffic flow forecasting.

Robustness Test. Table 3.9 showing the noise impact comparison between CCDSReFomer and the SOTA model STAEFormer on the PEMS04 dataset. The comparative analysis of noise robustness between CCDSReFormer and STAEFormer reveals distinct behavioral patterns under various types of perturbations, primarily attributed to their fundamental architectural differences. Our analysis focuses on three primary noise categories: Gaussian, dropout, and systematic bias, each demonstrating unique interaction patterns with the models' architectural components.

In the context of Gaussian noise, CCDSReFormer demonstrates enhanced resilience primarily through its ReLSA mechanism, which inherently filters noise through selective attention patterns. The dual-stream architecture provides complementary information paths, effectively reducing the

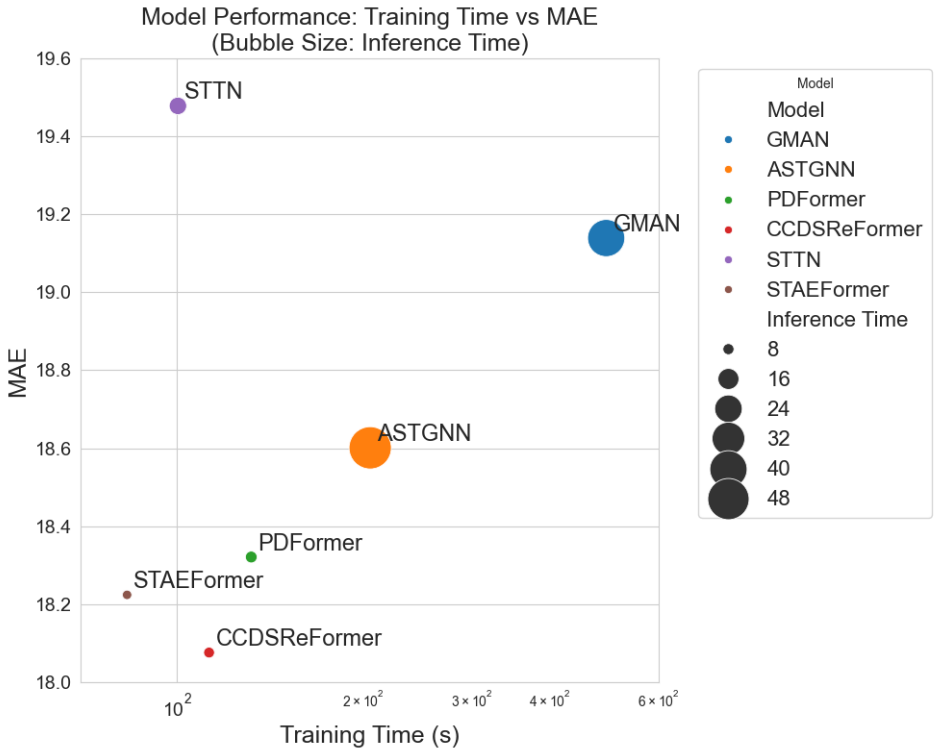


FIGURE 3.10. Computational time vs Accuracy on PEMS04

TABLE 3.9. Noise Impact Comparison: CCDSReFormer vs. STAEFormer on PeMS04

Model	**Base**	**Noise Level**				
		0.1	0.2	0.3	0.4	0.5
Gaussian Noise (MAE)						
CCDSReFormer	18.116	18.523	19.248	20.410	22.163	24.457
Degradation (%)	-	2.2%	6.2%	12.7%	22.3%	35.0%
STAEFormer	18.224	18.771	19.697	21.087	23.129	25.697
Degradation (%)	-	3.0%	8.1%	15.7%	26.9%	41.0%
Dropout Noise (MAE)						
CCDSReFormer	18.116	18.297	18.842	19.746	21.012	22.827
Degradation (%)	-	1.0%	4.0%	9.0%	16.0%	26.0%
STAEFormer	18.224	18.588	19.318	20.412	21.869	23.873
Degradation (%)	-	2.0%	6.0%	12.0%	20.0%	31.0%
Systematic Bias (MAE)						
CCDSReFormer	18.116	18.660	19.564	20.833	22.645	25.362
Degradation (%)	-	3.0%	8.0%	15.0%	25.0%	40.0%
STAEFormer	18.224	18.953	20.047	21.505	23.509	26.425
Degradation (%)	-	4.0%	10.0%	18.0%	29.0%	45.0%

impact of random perturbations. Additionally, the enhanced convolution component maintains local structural integrity under noise conditions. In contrast, STAEFormer's traditional attention mechanism preserves all connections, potentially allowing noise to propagate through the network. Its sequential spatial-temporal processing architecture may compound noise effects through successive layers, while its adaptive embeddings must continuously adjust to noisy inputs, potentially leading to suboptimal representations.

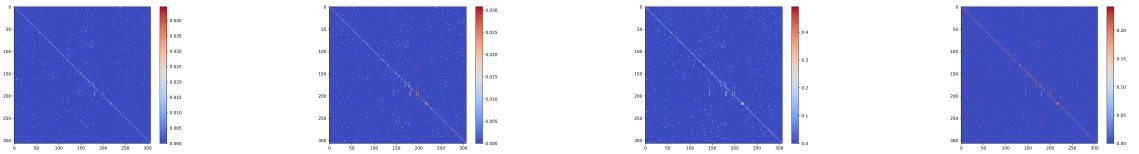
For dropout noise, the architectural advantages of CCDSReFormer become particularly evident. The parallel spatial-temporal processing framework provides redundant information pathways, enabling robust prediction even when certain inputs are missing. The geographic masking mechanism facilitates effective interpolation of missing values by leveraging spatial correlations, while the criss-crossed learning structure maintains consistent information flow across both streams. Conversely, STAEFormer's sequential processing architecture means that missing values can significantly impact subsequent processing steps, with its single-stream design offering limited redundancy for missing data compensation. The adaptive embedding layer, while flexible, may inadvertently overfit to missing data patterns, potentially compromising generalization.

Regarding systematic bias, CCDSReFormer's architecture offers several mechanisms for mitigation. The dual-stream approach enables cross-validation of patterns between spatial and temporal dimensions, while the RMSNorm component helps normalize systematic shifts in the data distribution. The enhanced convolution layer maintains local relationship consistency even under biased conditions. STAEFormer, however, shows greater susceptibility to systematic bias due to its sequential processing nature, which may amplify biases through successive layers. Its adaptive embedding strategy, while powerful for normal conditions, may inadvertently reinforce systematic errors, and its normalization approach proves less robust against consistent biases.

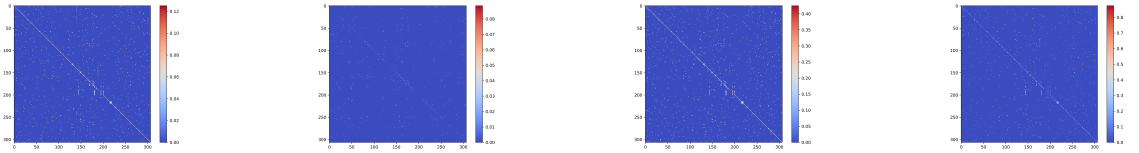
3.5.7 Discussion/Case Study

To gain a deeper understanding of the functionalities of **CCDSReFormer**, we further explore how attention mechanisms are applied to different types of information - spatial, temporal, integration of both and delay-aware attention. The structure of each attention module is divided into two

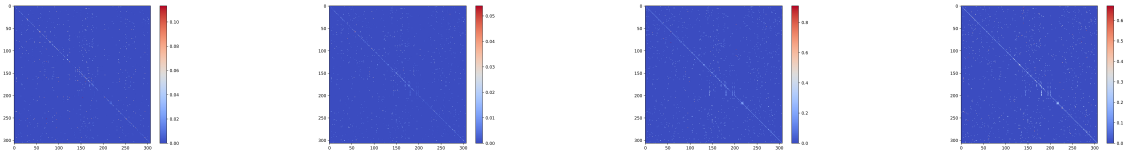
FIGURE 3.11. Attention scores in **ReSSA**: horizontal axis for stage and heads growth, vertical axis for layer depth. Note: H*n* = Head *n*, L*n* = Layer *n*



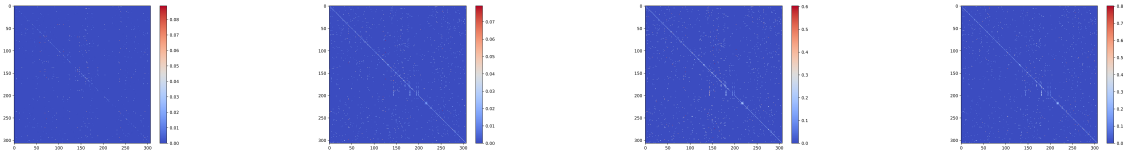
(A) Stage 1: H1, L1 (B) Stage 1: H2, L1 (C) Stage 2: H1, L1 (D) Stage 2: H2, L1



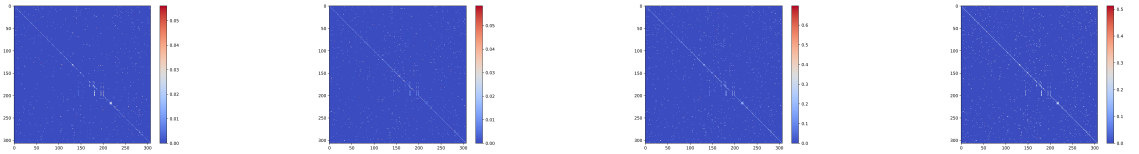
(E) Stage 1: H1, L2 (F) Stage 1: H2, L2 (G) Stage 2: H1, L2 (H) Stage 2: H2, L2



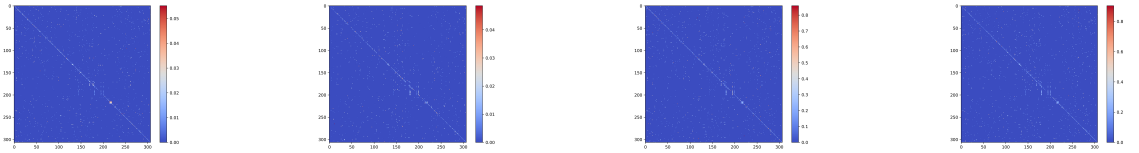
(I) Stage 1: H1, L3 (J) Stage 1: H2, L3 (K) Stage 2: H1, L3 (L) Stage 2: H2, L3



(M) Stage 1: H1, L4 (N) Stage 1: H2, L4 (O) Stage 2: H1, L4 (P) Stage 2: H2, L4



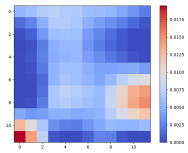
(Q) Stage 1: H1, L5 (R) Stage 1: H2, L5 (S) Stage 2: H1, L5 (T) Stage 2: H2, L5



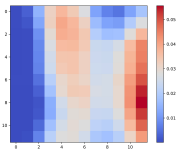
(U) Stage 1: H1, L6 (V) Stage 1: H2, L6 (W) Stage 2: H1, L6 (X) Stage 2: H2, L6

distinct stages which include **ReSSA** and **ReTSA** as depicted in Figure 3.1. In Figure 3.11, the first stage is characterized by the implementation of **ReSSA**, while the second stage involves criss-cross learning or what we call integration of temporal information using **ReTSA**. For **ReDASA**, we generate visualizations that directly correspond to the increase in heads and layers, aligning closely with the experimental settings described in Section 3.5.3. By examining the

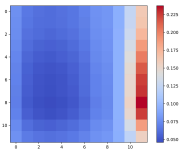
FIGURE 3.12. Attention scores in **ReTSA** by layers: horizontal axis shows stage progression and more heads; vertical axis reflects deeper layers. Note: Hn = Head n, Ln = Layer n



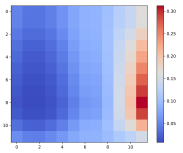
(A) Stage 1: H1, L1



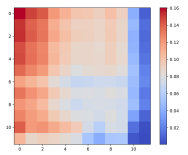
(B) Stage 1: H2, L1



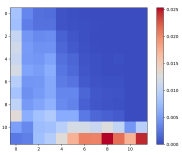
(C) Stage 2: H1, L1



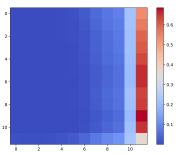
(D) Stage 2: H2, L1



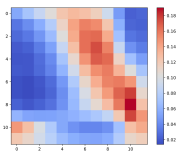
(E) Stage 1: H1, L2



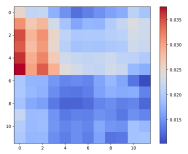
(F) Stage 1: H2, L2



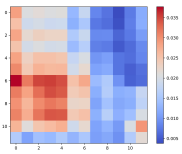
(G) Stage 2: H1, L2



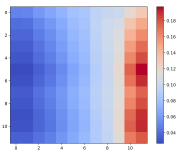
(H) Stage 2: H2, L2



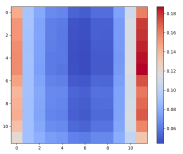
(I) Stage 1: H1, L3



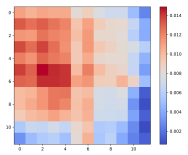
(J) Stage 1: H2, L3



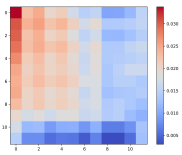
(K) Stage 2: H1, L3



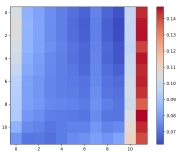
(L) Stage 2: H2, L3



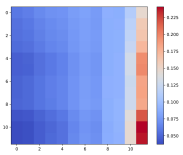
(M) Stage 1: H1, L4



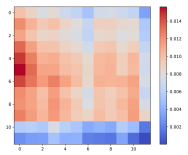
(N) Stage 1: H2, L4



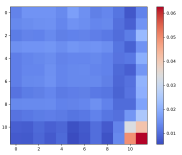
(O) Stage 2: H1, L4



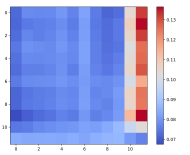
(P) Stage 2: H2, L4



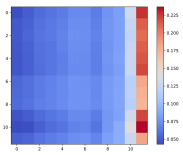
(Q) Stage 1: H1, L5



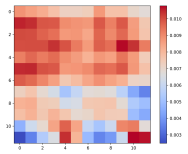
(R) Stage 1: H2, L5



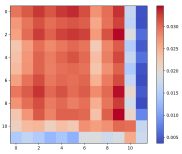
(S) Stage 2: H1, L5



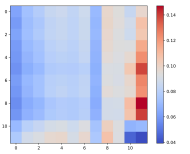
(T) Stage 2: H2, L5



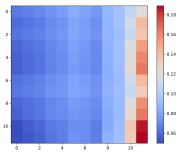
(U) Stage 1: H1, L6



(V) Stage 1: H2, L6



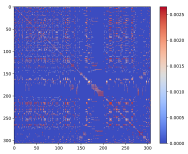
(W) Stage 2: H1, L6



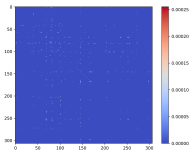
(X) Stage 2: H2, L6

attention score visualizations, as displayed in Figures 3.11, 3.12 and 3.13, we can draw several insightful conclusions as below:

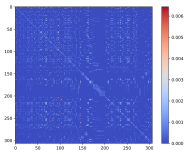
FIGURE 3.13. Attention scores in **ReDASA** across layers: horizontal axis shows more heads, vertical indicates deeper layers. Note: Hn = Head n, Ln = Layer n



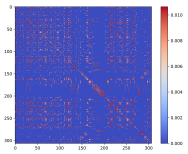
(A) Stage 1: H1, L1



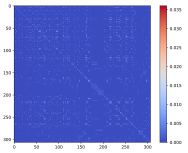
(B) Stage 1: H2, L1



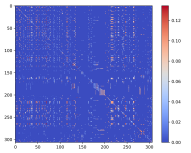
(C) Stage 2: H3, L1



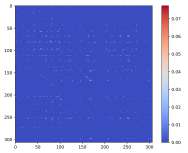
(D) Stage 2: H4, L1



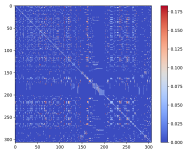
(E) Stage 1: H1, L2



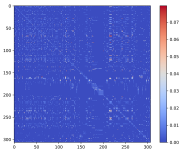
(F) Stage 1: H2, L2



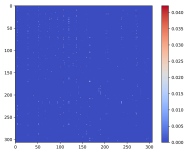
(G) Stage 2: H3, L2



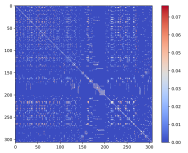
(H) Stage 2: H4, L2



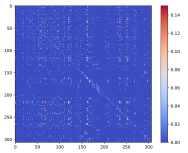
(I) Stage 1: H1, L3



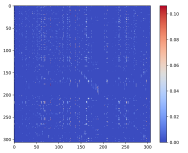
(J) Stage 1: H2, L3



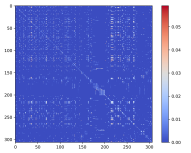
(K) Stage 2: H3, L3



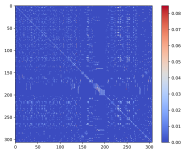
(L) Stage 2: H4, L3



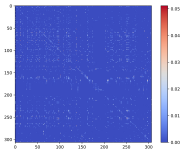
(M) Stage 1: H1, L4



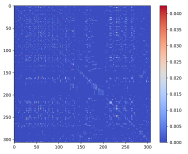
(N) Stage 1: H2, L4



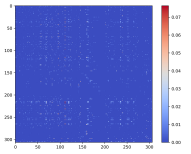
(O) Stage 2: H3, L4



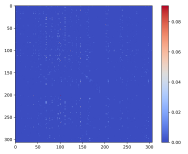
(P) Stage 2: H4, L4



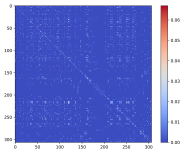
(Q) Stage 1: H1, L5



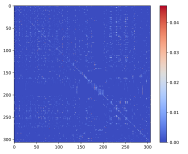
(R) Stage 1: H2, L5



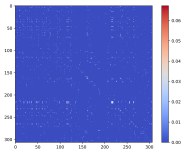
(S) Stage 2: H3, L5



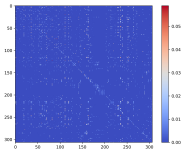
(T) Stage 2: H4, L5



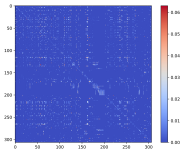
(U) Stage 1: H1, L6



(V) Stage 1: H2, L6



(W) Stage 2: H3, L6



(X) Stage 2: H4, L6

Rectified self-attention with increasing of stages: The self-attention mechanism, when applied in increasing stages (or Criss-Cross learning), evenly distributes attention across integrated spatial and temporal information. This phenomenon is clearly visible when contrasting Figure 3.11a with Figure 3.11c. Initially, in **ReSSA** (Figure 3.11a), the heatmap displays attention scores from 0.005 to 0.025, predominantly accentuating spatial details with elevated attention

scores. However, upon integrating **ReTSA**, the heatmap in Figure 3.11c exhibits a more evenly distribution of attention, with scores approximating 0.2, including both spatial and temporal aspects.

A similar trend is observable in the initial **ReTSA** attention scores (Figure 3.12a), where an increase in stage (as compared with Figure 3.12c) leads to an average trend in attention scores. Additional examples are provided in Figures 3.12e, 3.12g, 3.12i, and 3.12k, among others. Such observations suggest that the integration of the Criss-Cross methodology equips the **CCDSReFormer** with an enhanced capability to capture and delineate both spatial and temporal information with refined attention.

Rectified self-attention in different number of heads: As the number of attention heads increases from 1 to 2, each head can potentially focus on distinct segments or attributes of the input data. This concept is illustrated through a comparison on **ReSSA** (as in Figure 3.11) between Figure 3.11a and Figure 3.11b. Notably, Figure 3.11b displays higher attention scores compared to Figure 3.11a, indicating a more comprehensive capture of data features and relationships.

Furthermore, in **ReTSA**, as illustrated in Figure 3.12, an increase in the number of heads during stage 1 (as shown in the first two columns, e.g., Figures 3.12a and 3.12b) results in significant changes in attention scores compared to stage 2 (as depicted in the last two columns, e.g., Figures 3.12c and 3.12d).

Similarly, in **ReDASA** (see Figure 3.13), setting the number of attention heads to 4 reveals that, with an increasing number of heads, the attention scores in **ReDASA** closely resemble those when the head count is one. This can be specifically observed in comparisons such as in Figures 3.13a and 3.13d; However, there are notable exceptions with significant differences, as seen when comparing Figures 3.13e and 3.13h. This diversity in attention distribution enhances our model's ability to analyze input data more effectively, allowing for the recognition of a wider range of patterns and connections.

Rectified self-attention with increasing of layers: As layers increase, attention scores are dynamically adjusted. This is evident in **ReSSA** (as in Figure 3.11) stage 1 and head 1, where the scores slightly rise from a ceiling of 0.04 to 0.05 across layers, as shown from Figure 3.11a to Figure 3.11t. The change becomes slighter at higher layers, such as layer 5 (Figure 3.11p) and layer 6 (Figure 3.11t). Similarly, in stage 2 and head 2, the maximum attention score rises from 0.5 to 0.8, as illustrated in Figure 3.11c and Figure 3.11x which means even with Criss-Cross learning, with the increasing of layers can further enhance the score of attention distribution.

Additionally, with **ReTSA** (referenced in Figure 3.12), a progression is evident when observing the visualization from the top row (Figures 3.13a, 3.13b, 3.13c, and 3.13d) to the bottom row (Figures 3.13u, 3.13v, 3.13w, and 3.13x). As we examine the subsequent layers, from Layer 1 through Layer 6, there is a noticeable trend towards a more uniform distribution of attention. This suggests that as the layers deepen, the model may be gaining a more nuanced perception of the data's interrelations.

3.6 Practical Implications and Limitation

Practical Implications and Deployment Considerations. Deploying the proposed **CCDSReFormer** model in real-world traffic prediction systems offers significant advantages in predictive accuracy and computational efficiency, making it suitable for operational environments that require timely and reliable forecasts. The model's Rectified Linear Self-Attention (**ReLSA**) mechanism reduces computational complexity from $O(N^2)$ to $O(kN)$, enabling scalability for large-scale traffic networks on standard hardware with modern GPUs. Implementation challenges such as data quality, missing values, and system integration can be addressed through robust data preprocessing, utilizing frameworks like PyTorch for compatibility, and establishing automated workflows for continuous model training and deployment. The model maintains efficient performance suitable for real-time applications, with manageable memory and processing requirements validated through empirical experiments. By considering computational resource requirements and providing solutions to potential deployment challenges, **CCDSReFormer**

is well-positioned for integration into existing traffic management systems, offering improved forecasting capabilities and aiding in effective decision-making.

Limitation. While this chapter focuses primarily on the methodological contributions of the proposed CCDSReFormer model, it is essential to consider the diversity inherent in real-world traffic systems. Traffic systems vary widely, ranging from urban networks with high-density, dynamic traffic conditions to highway systems characterized by more structured, predictable flow patterns. The datasets employed in this study, including urban mobility datasets such as CHIBike and NYCTaxi, and highway traffic datasets like PeMS04 and PeMS08, reflect this diversity and demonstrate the model's adaptability across different traffic scenarios.

However, we acknowledge that the generalizability of CCDSReFormer to other types of traffic systems, such as rural networks or multimodal transit systems, requires further exploration. These systems may have different characteristics, such as sparse data availability, irregular traffic patterns, or additional modes of transportation, which could affect the model's performance. Future work could involve applying the model to additional datasets or integrating domain-specific constraints to better address the unique characteristics of diverse traffic systems. This analysis would provide a more comprehensive understanding of the model's applicability and its potential limitations in addressing the varied challenges of real-world traffic systems.

Furthermore, while our model demonstrates improved computational efficiency compared to traditional attention mechanisms, the computational load of the entire model framework remains significant, particularly for large-scale networks or high-resolution temporal data. This may limit its applicability in resource-constrained environments or in scenarios requiring real-time predictions. Although we have shown that our method can effectively reduce computational costs through the use of **EnReLSA** attention mechanisms, there is still room for optimization. Future research could focus on further simplifying the model architecture or employing model compression techniques to decrease computational demands without significantly compromising performance.

By acknowledging these limitations, we aim to provide a balanced perspective on our model's capabilities and identify areas where additional research could enhance its effectiveness and applicability.

3.7 Conclusion

In this study, we introduce the Dual-Stream Criss-Cross Enhanced Rectified Transformer (**CCDSReFormer**), an innovative model designed for accurate and computationally efficient traffic flow/demand prediction. The model uniquely integrates spatial and temporal information through a dual Criss-Cross stream, effectively capturing the intricate traffic patterns. A Rectified Linear Self Attention (ReLSA) combined with locally enhanced convolution sharpens the model's focus on local spatial-temporal features and nuanced traffic dynamics influenced by localized conditions and simultaneously reduces computational demands. This ensures a nuanced exploration of traffic patterns influenced by localized conditions, contributing to a more accurate prediction. Comparative analysis of various **CCDSReFormer** configurations demonstrates each component's positive impact on prediction accuracy with an average increase of 5.55% and maintains the computational efficiency at a manageable level. Our method has undergone extensive testing across six diverse real-world datasets, establishing its superiority over existing state-of-the-art models in terms of both performance and computational efficiency. Additionally, our model exhibits robust parameter tuning capabilities, further emphasizing its versatility and applicability in the dynamic domain of traffic flow prediction. Looking ahead, we plan to explore the application of the **CCDSReFormer** model to other spatial-temporal datasets, such as weather forecasting.

Chapter 4

Mamba–Transformer Fusion for Efficient Spatial–Temporal Traffic Forecasting

---

Addressing **RQ2**: What synergistic architectural principles enable *linear-time state-space models* to complement—rather than replace—attention mechanisms?, this chapter develops **ST-MambaSync**, a principled fusion that marries linear-time state-space layers with sparse attention for Pareto-optimal accuracy and efficiency.

4.1 Introduction

Accurate and efficient traffic flow prediction is critical for optimizing traffic management, enhancing road safety, and mitigating environmental impacts. Effective traffic forecasting enables smarter city planning, increases road capacity, reduces congestion, and significantly lowers travel times, contributing to reduced pollution and sustainable urban development [Kashyap et al., 2022], as illustrated in Figure 4.1. Traditional models face substantial challenges when processing extensive sequential data, as they demand significant memory and computational resources. Additionally, these models often suffer from slow processing speeds due to the absence of a unified summary state and difficulty in capturing complex, non-linear dependencies in traffic patterns.

For instance, classical approaches like Kalman Filters [Guo et al., 2014] are limited by their inability to handle non-linear dynamics, making them inadequate for the highly variable nature of traffic conditions. Although deep learning models such as Convolutional Neural Networks (CNNs) [Lecun et al., 1998], Graph Neural Networks (GNNs) [Scarselli et al., 2009], and Recurrent Neural Networks (RNNs) [Rumelhart et al., 1986] have shown considerable promise in

traffic prediction, they come with their own set of challenges (see Table 4.1). CNN-based models are effective in capturing spatial features but struggle with long-range temporal dependencies. GNNs excel at modeling spatial relationships but face computational inefficiencies and issues like over-smoothing when addressing long-range dependencies. Similarly, RNN-based models, including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), are prone to vanishing gradient problems, which limit their performance on extended sequences and make them computationally expensive.

Recent advancements in transformer-based models, such as STAEFormer [Liu et al., 2023b], have successfully captured both short and long-range dependencies using self-attention mechanisms. However, these models are computationally expensive, with costs scaling quadratically with the sequence length, posing significant challenges for large-scale, real-time traffic systems. This trade-off between accuracy and efficiency highlights the need for a balanced approach.

To address these challenges, we propose ST-MambaSync, a novel traffic flow prediction model that synergistically combines the selective state space (Mamba) model with Transformer mechanisms. ST-MambaSync leverages the strengths of the Mamba mechanism—an advanced synthesis of attention capabilities and residual network (ResNet) architecture—with the global information processing power of Transformers. The Mamba block efficiently captures localized information, acting as a natural complement to the Transformer's global attention mechanism, which focuses on long-range dependencies. This integration not only enhances prediction accuracy but also significantly reduces computational costs, making it ideal for real-time traffic management applications. Our model sets a new benchmark in the field through a comprehensive comparative analysis, demonstrating superior performance in terms of both accuracy and computational efficiency and achieving high accuracy. Through a comprehensive comparative analysis, we demonstrate that ST-MambaSync achieves a significant improvement in accuracy—achieving a 0.70% reduction in Mean Absolute Error (MAE) and a 0.62% reduction in Root Mean Square Error (RMSE) compared to state-of-the-art transformer-based models. Furthermore, it reduces computational demand by 64.86% in Floating Point Operations Per Second (FLOPS) and 12.54% in inference time, highlighting its efficiency for real-world applications.

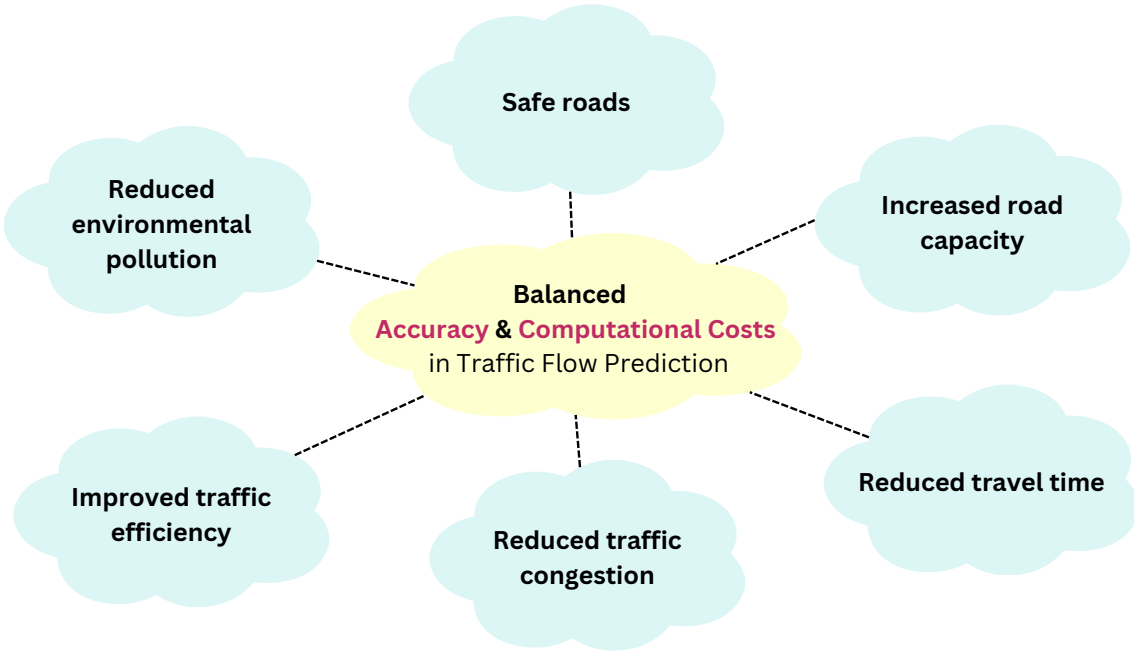


FIGURE 4.1. Various benefits for the fusion of ST-Mamba and Transformer on an effective traffic flow prediction.

In the following sections, we review the existing literature on traffic flow prediction, starting with traditional approaches in Section 4.1.1, advancing to deep learning methods in Section 4.1.2, and concluding with a discussion on a novel approach known as the Selective State Space model (Mamba) in Section 4.1.3.

4.1.1 Traditional Approaches for Traffic Flow Predictions

Long-range spatial dependencies in traffic forecasting are particularly challenging and computationally intensive due to the complex and dynamic nature of spatial-temporal dependencies involved. Traffic conditions on a particular road segment are not only influenced by historical data from that segment but also by current and past conditions of connected or nearby roads. This interconnectedness means that models need to integrate vast amounts of data over various time frames and locations, exponentially increasing computational demands. The necessity to process and learn from these extensive datasets in real time to produce accurate predictions further complicates the task, pushing the limits of current computational capabilities and deep learning technologies.

Historically, traditional mathematical statistics methods and classical machine learning techniques mostly focus on short-term prediction, such as local weighted regression [Li et al., 2012, Jeong et al., 2013], integrated moving average autoregression [Kumar and Vanajakshi, 2015, Xu et al., 2016], Kalman filter [Guo et al., 2014, Emami et al., 2019], non-parametric regression [Arif et al., 2018, Kim et al., 2005], and dynamic pattern decomposition [Yu et al., 2021], were primarily utilized for single sequence predictions. These methodologies, while effective within their scope, typically focused on individual time series data, lacking the complexity needed for handling multiple interconnected data sequences.

4.1.2 Deep Learning Method in Traffic Flow Prediction

In recent years, deep learning methods have been widely used in various fields such as medication [Ghaderzadeh et al., 2024], text generation [Jamshidi et al., 2024], and classification [Shao et al., 2024e] etc, the field of traffic flow prediction has seen notable progress with the integration of deep learning technologies. Deep learning models like Graph Neural Networks (GNNs) [Scarselli et al., 2009], Convolutional Neural Networks (CNNs) [Lecun et al., 1998] and Recurrent Neural Networks (RNNs) [Rumelhart et al., 1986] have been effective in understanding spatial and temporal patterns.

Graph Neural Networks (GNNs) have emerged as a pivotal technology in traffic prediction, effectively modeling the complex, interconnected nature of transportation networks. These approaches can be categorized into several key developments: (1) Basic spatial modeling, exemplified by GWNet [Wu et al., 2020a] which utilizes GNNs to uncover hidden spatial dependencies in traffic networks; (2) Hybrid architectures, such as DCRNN [Li et al., 2018] which innovatively combines convolutional and recurrent units to simulate traffic flow as a diffusion process; (3) Adaptive mechanisms, demonstrated by AGCRN's [BAI et al., 2020] enhancement of node-specific pattern sensitivity, and STGCN's [Yu et al., 2018a] integration of graph and gated temporal convolutions for managing spatial-temporal variations; and (4) Advanced attention mechanisms, represented by GMAN's [Zheng et al., 2020a] implementation of spatial-temporal attention within graphs, and AdapGL's [Zhang et al., 2022] dynamic learning

of complex node dependencies through adaptive graph convolution. While these approaches have advanced the field significantly, they often struggle with capturing long-range dependencies and handling irregular traffic patterns. . Recently, TFM-GCAM [Chen et al., 2024] enriches spatial-temporal features with a traffic flow matrix. Despite their advancements, GNNs face computational challenges and struggle with long-range dependencies due to the over-smoothing problem. Also, the model Conjoint Spatio-Temporal graph neural network (COOL) [Ju et al., 2024], which innovatively models high-order spatio-temporal relationships using heterogeneous graphs and a conjoint self-attention decoder to enhance traffic forecasting, achieves state-of-the-art results on benchmark datasets. The study [Sun et al., 2024] proposed a Fast and Dynamic Temporal Graph Convolution Network (FD-TGCN) enhances traffic flow prediction by integrating a Fast Time Convolution Network (FTCN) to reduce training time. Kong et al. [2024b] propose an adaptive dual-graphic transformer with a cross-fusion strategy to effectively represent and analyze complex spatio-temporal dependencies in high-dimensional data, achieving superior long-term prediction accuracy. [Geng et al., 2024] leveraged gated temporal self-attention and distance spatial self-attention modules to handle better dynamic temporal variations and Ma et al. [2024] utilized adaptive adjacency matrices to accurately model spatial interdependencies, alongside transformer-based modules for global temporal correlation. However, the vanilla GNNs often experience over-smoothing, where neighboring node features become indistinguishable with added layers or attempts to capture global features. Addressing this issue typically requires combining other models that trade-off computational efficiency [Shao et al., 2024d, 2023].

Another method that faces a similar challenge of capturing long-range dependencies is CNNs-based models, like [Zhang et al., 2019, Yu et al., 2016, Sayed et al., 2023], which are adept at handling spatial data but often falter with long-range temporal patterns. To overcome these, some hybrid models, e.g., CNNs combined with LSTM [Bogaerts et al., 2020, Méndez et al., 2023, Li et al., 2021c] are integrated for capturing the long-range dependency while increasing the model complexity and inducing high computational cost. Conversely, RNNs-based traffic focusing models such as [Belhadi et al., 2020, Zhang and Patras, 2018] can capture long-range dependency. However, when processing very long sequences, traditional recurrent neural networks suffer from the vanishing gradient problem, causing them to forget important information from distant

past time steps. To address this limitation, Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] was developed, which uses gating mechanisms to selectively retain or forget information, enabling the model to capture long-term dependencies more effectively. Some of the LSTM-based modes, such as Zhao et al. [2017], Doğan [2022], are adopted both on long and short-term dependencies. Traffic flow prediction based on Gated Recurrent Units (GRU) is initially established by [Fu et al., 2016]. The method combines GRU and LSTM, which outperformed (ARIMA) and illustrates that they are good at temporal dynamics. But all RNNs, LSTM, and GRU-based model struggle with very long-term dependencies due to issues like the vanishing gradient problem, which impedes their performance on lengthy sequences and restricts their parallelization, slowing down both training and inference.

To overcome these limitations, researchers have explored various approaches to enhance the performance of traffic flow prediction models while effectively captioning the long-range dependency. One promising direction is the development of attention mechanisms. Transformers [Vaswani, 2017] are highly efficient at capturing complex dependencies within data, which is widely used in natural language processes, computer vision, etc. The first study that uses a transformer in spatial-temporal traffic flow prediction is proposed by Cai et al. [2020a], which evidences that transformers are powerful for long-range traffic flow prediction. Because they can effectively handle complex spatial and temporal dynamics through mechanisms like self-attention, which enables the model to focus on different aspects of traffic data across various timescales. For predicting traffic flow over long ranges using Transformer models, several recent studies have been leveraging different variations and enhancements of the original Transformer architecture to handle better the complex spatial and temporal dynamics of traffic data. One notable model is proposed by Jiang et al. [2023]; This model introduces novel elements like spatial self-attention modules that integrate both geographic and semantic neighborhood information, that are crucial for capturing both short-range and long-range spatial dependencies in traffic data. Most recently, Xing et al. [2024] pointed out the Transformer is over-focused on global dependency, and therefore Shao et al. [2024b] further utilizes dual-stream attention with ResNet to enhance the local dependency on spatial and temporal information which avoid the high computation cost; and it further utilized the sparse matrix to enhance the computational

efficiency. Recent advancements in transformer-based models for traffic flow prediction, such as the STAEFormer [Liu et al., 2023b], have highlighted the efficacy of spatial and temporal attention mechanisms. These models are noted for their state-of-the-art performance in managing long and short-range traffic data, leveraging separate attention mechanisms to enhance prediction accuracy and model responsiveness. However, despite these advancements, the deployment of transformer-based models in practical applications encounters significant hurdles, primarily due to their computational demands.

The self-attention mechanism in transformers is resource-intensive, with computational costs scaling quadratically as sequence lengths increase. This presents challenges in large-scale traffic networks or scenarios requiring rapid real-time analysis. Increasing model accuracy involves deeper attention layers, exacerbating computational demands. This trade-off between accuracy and feasibility is crucial in real-time traffic systems, necessitating a balance between swift predictions and resource constraints. Thus, while promising for traffic flow prediction, transformers require careful management of computational demands to balance efficiency and accuracy effectively.

4.1.3 State Space Model

Given the limitations of existing deep learning methods, the Selective State of Space model (commonly referred to as Mamba) [Gu and Dao, 2023] stands out for its ability to deliver high accuracy on very long-range traffic flow prediction while requiring less computational effort. This efficiency is particularly crucial in both short-term and long-term traffic management scenarios, where quick and reliable predictions are vital for effective congestion control, route optimization, and traffic regulation. A recent study ST-Mamba [Shao et al., 2024h] marked the first to apply the Mamba model to spatial-temporal traffic flow prediction, demonstrating promising results in reducing computational costs. However, there remains room for improvement in balancing accuracy with manageable computational demands. Based on the paper [Shao et al., 2024h], simply adding more Mamba layers does not guarantee an increase in prediction accuracy because Mamba focuses locally and may lose some global information with additional layers.

Therefore, combining Mamba with a transformer, which can enhance global data comprehension, is necessary to complement and balance the model.

This chapter introduces the spatial-temporal synergy of the Selective State Space (Mamba) and Transformer methods, called ST-MambaSync, a novel framework that efficiently integrates the popular transformer and Mamba methods for accurate traffic flow prediction. The ST-MambaSync model comprises two main components: the ST-Transformer and the ST-Mamba Block. The ST-Transformer efficiently processes data, capturing global information through spatial and temporal features. In contrast, the ST-Mamba Block, which includes an ST-Mixer, converts the tensor into a matrix. This matrix is then fed into the ST-Mamba layer, which updates individual hidden states and extends memory for long-range data, focusing more on local information. The combination of ST-Transformer and ST-Mamba Block not only enhances both global and local features but also accelerates computation, making it an effective component of our integrated approach for managing complex traffic data.

TABLE 4.1. Capabilities and Computational Costs of Various Models in Traffic Flow Prediction

Model	**Long Range Dependency**	**Computation Cost**
CNNs	No	Moderate
GNNs	No	Variable
RNNs	Semi-Yes	High
Transformer	Yes	High
Mamba	Yes	Low

4.1.4 Contribution

Addressing the challenges in existing traffic flow prediction models as summarized in Table 4.1, this analysis shows the varied abilities of different models to handle long-range dependencies and their computational costs. CNNs and GNNs lack long-range capability, while RNNs provide limited support. Transformers and Mamba excel in this area, with Mamba being especially efficient and having low computational demands. This efficiency makes Mamba

highly suitable for enhancing traffic management, reducing congestion, and improving route planning in intelligent transportation systems.

To the best of our knowledge, this chapter introduces a groundbreaking integration of the selective state-of-space model (Mamba) with attention mechanisms specifically tailored for spatial-temporal data, establishing that Mamba effectively functions as a type of attention within a ResNet framework. The principal contributions of our research are summarized as follows:

- We present the first-ever theoretical investigation demonstrating that Mamba functions as a specialized form of attention mechanism with ResNet structure, enhancing its ability to capture local features critical for traffic pattern recognition.
- We present the first theoretical proof that Mamba and Transformer technologies are complementary. This synergy significantly enhances model performance by improving the balance between prediction accuracy and computational efficiency.
- We conduct a series of experiments on six real datasets that reveal enhancements in accuracy with a 0.70% reduction in MAE, 0.62% in RMSE, and 0.31% in MAPE, accompanied by a substantial decrease in computational demand—64.86% reduction in FLOPS, 12.54% in inference time, and 19.44% in training time—compared to the previous state-of-the-art model. ST-MambaSync sets new benchmarks in both accuracy and processing speed for traffic flow prediction. It tackles the major challenges of data volume and computational load, establishing a new standard in the field.

Traffic Flow Tensor. In this study, we define the traffic signal as $\mathbf{X}_t \in \mathbb{R}^{N \times d}$ representing d traffic flow characteristics at N sensors of the road at a given timestamp t. Over a period $T > 0$, the traffic data $\{\mathbf{X}_t\}_{t=1}^{T}$ can be stacked into a mode-3 tensor $\mathcal{X} = [\mathbf{X}_1, ..., \mathbf{X}_T] \in \mathbb{R}^{T \times N \times d}$, organized along the temporal, spatial, and feature-specific dimensions.

Problem Statement. The objective of traffic flow forecasting is to predict future traffic conditions accurately using historical data. For this purpose, we define a function f within the context of a road network $\mathcal{G}$, which leverages traffic flow data from the past M timestamps to forecast traffic conditions over the next Z timestamps. This relationship can be mathematically

formulated as:

$$f\left([\mathbf{X}_{t-M+1}, \ldots, \mathbf{X}_t]\right) \mapsto [\mathbf{X}_{t+1}, \ldots, \mathbf{X}_{t+Z}].$$

We intend to learn the model f by taking t from M to $T - Z$ in order to efficiently utilize the observed traffic flow tensor $\mathcal{X}$.

4.1.5 Selective State-Space Model

The Selective State-Space Model (Mamba) was initially established by Gu and Dao [2023]. Our study focuses predominantly on the fusion of the powerful Mamba and Transformer techniques. The Mamba model is described through the state-space representation of a continuous-time linear time-invariant system:

$$\frac{d\mathbf{h}}{d\tau} = \mathbf{A}\mathbf{h}(\tau) + \mathbf{B}\mathbf{x}(\tau), \tag{4.1}$$

$$\mathbf{y}(\tau) = \mathbf{C}\mathbf{h}(\tau). \tag{4.2}$$

We define $\tau \in (0, +\infty)$ as the continuous time variable. Assume, without loss of generality, that the state vector $\mathbf{h}(\tau)$, the observation vector $\mathbf{y}(\tau)$, and the input vector $\mathbf{x}(\tau)$ each have appropriate dimensions. The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ represent unknown parameters that need to be determined based on observed data.

To learn model parameters $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ from discrete observation, people often use the ZOH (Zero Order Hold) method to yield an exact discretization for the continuous system as follows:

$$\mathbf{h}_{t+1} = e^{\mathbf{A}\Delta}\mathbf{h}_t + \mathbf{B}\mathbf{x}_t, \tag{4.3}$$

$$= \tilde{\mathbf{A}}\mathbf{h}_t + \tilde{\mathbf{B}}\mathbf{x}_t. \tag{4.4}$$

The above discrete equation defines state transition from $\mathbf{h}_t$ to $\mathbf{h}_{t+1}$ by applying the new transition matrix over a single time step, Δ, and incorporating the contribution of the input at time t. The

matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ are calculate based on:

$$\tilde{\mathbf{A}} = \exp(\Delta\mathbf{A}), \tag{4.5}$$

$$\tilde{\mathbf{B}} = \mathbf{A}^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I})\mathbf{B}, \tag{4.6}$$

$$\approx \Delta\mathbf{B}.$$

The selective mechanism in Selective State Space Model is then innovated with an adoptive and adjustable parameter on $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ being as functions of the input $\mathbf{x}(\tau)$

$$\mathbf{A}_\tau = f_A(\mathbf{x}(\tau)), \quad \mathbf{B}_\tau = f_B(\mathbf{x}(\tau)), \quad \mathbf{C}_\tau = f_C(\mathbf{x}(\tau)). \tag{4.7}$$

4.2 Introduction of ST-MambaSync

The ST-MambaSync architecture comprises several components: an embedding layer, a temporal-attention block, a spatial-attention block, and a simplified ST-Mamba block. A schematic of this architecture is illustrated in Figure 4.2.

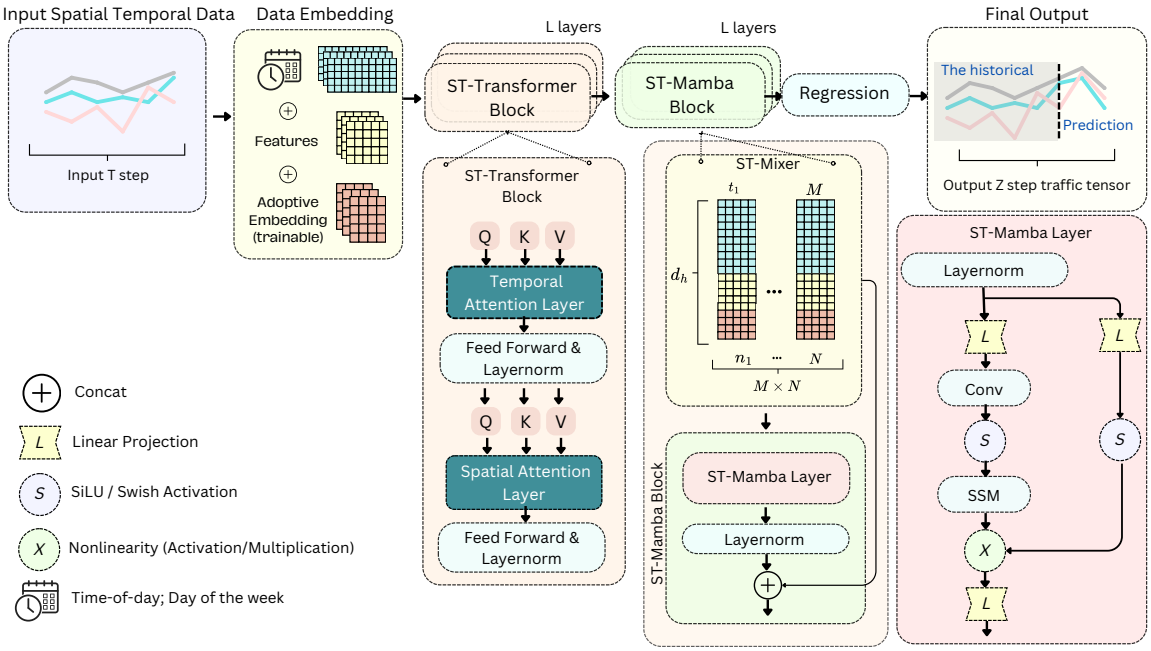


FIGURE 4.2. The framework of proposed ST-MambaSync.

4.2.1 Data Embedding

We utilize an adoptive data embedding layer, as the studies [Liu et al., 2023b, Shao et al., 2024h], that processes the sequential input $\mathbf{X}_{t-M+1:t}$ to encapsulate and reflect the dynamic temporal patterns present within the traffic data. Through the application of a dense neural layer, we extract the intrinsic feature embedding $\mathbf{E}_t^f \in \mathbb{R}^{M\times N\times d_e}$:

$$\mathbf{X}_t^f = \text{Dense}(\mathbf{X}_{t-M+1:t}), \tag{4.8}$$

where, d_e signifies the dimension of the embedded features, and $\text{Dense}(\cdot)$ represents the applied dense layer along feature dimension . Furthermore, we introduce a parameterized dictionary for the embedding of weekdays $\mathbf{E}_w \in \mathbb{R}^{7\times d_e}$ and another for the embedding of distinct times of the day $\mathbf{E}_h \in \mathbb{R}^{288\times d_e}$, encapsulating the cyclical nature of weeks with 7 days and with 288 time-of-day intervals at rate 5 minutes. With $\mathbf{W}_t \in \mathbb{R}^M$ representing the weekday index and $\mathbf{H}_t \in \mathbb{R}^M$ representing the time-of-day index over the period from $t - M + 1$ to t, we map these indices to their respective embeddings, yielding the weekday embedded features $\mathbf{E}_{w_t} \in \mathbb{R}^{M\times d_e}$ and time-of-day embedded features $\mathbf{E}_{h_t} \in \mathbb{R}^{M\times d_e}$. The combination and expansion of these embeddings generate the cyclical feature embedding $\mathbf{E}_t^c \in \mathbb{R}^{M\times N\times 2d_e}$, which is utilized to incorporate periodic patterns into the traffic data.

Considering the rhythmic progression of time and the interlinked nature of traffic events, traffic sensors yield data with unique temporal traits. To address the need for a uniform approach to encapsulate these spatio-temporal dynamics, a shared spatio-temporal adaptive embedding, $\mathbf{E}_t^s \in \mathbb{R}^{M\times N\times d_s}$, is put forth. This embedding is initialized utilizing Xavier uniform initialization, a technique that primes the model's weights to avoid excessively large or small gradients initially, and thereafter, it is treated as a model parameter.

The integration of the aforesaid embeddings results in a hidden spatio-temporal representation $\mathbf{X}_{emb} \in \mathbb{R}^{M\times N\times d_h}$:

$$\mathbf{X}_{emb} = \textbf{Concatenate}(\mathbf{E}_t^f; \mathbf{E}_t^c; \mathbf{E}_t^s). \tag{4.9}$$

In this equation, the concatenation operation is denoted by a comma, and the dimension of the hidden representation d_h is computed as $3d_e + d_s$.

4.2.2 Spatial Temporal Transformer (ST-Transformer Block)

We utilize standard transformers along both temporal and spatial dimensions to understand complex traffic interactions. Given a hidden spatio-temporal tensor $\mathbf{X}_{emb} \in \mathbb{R}^{M \times N \times d_h}$, where M is the number of observation timestamps, and N represents spatial nodes. To better capture the temporal feature, we denote each node as $i = 1, 2, \ldots, N$ and then derive the query, key, and value matrices using temporal transformer layers as follows:

$$\mathbf{Q}_i^{(te)} = \mathbf{X}_i^{(te)} \mathbf{W}_Q^{(te)}, \quad \mathbf{K}_i^{(te)} = \mathbf{X}_i^{(te)} \mathbf{W}_K^{(te)}, \quad \mathbf{V}_i^{(te)} = \mathbf{X}_i^{(te)} \mathbf{W}_V^{(te)}, \tag{4.10}$$

where $\mathbf{X}_i^{(te)} \in M \times d_h$, $\mathbf{W}_Q^{(te)}, \mathbf{W}_K^{(te)}$,and$\mathbf{W}_V^{(te)} \in \mathbb{R}^{d_h \times d_h}$ are learnable parameters. The self-attention scores are computed as:

$$\mathbf{A}_i^{(te)} = \text{Softmax}\left(\frac{\mathbf{Q}_i^{(te)} (\mathbf{K}_i^{(te)})^\top}{\sqrt{d_h}} \right), \tag{4.11}$$

capturing the difference on each node between temporal connections, and $\mathbf{A}_i^{(te)} \in M \times M$. The output of the temporal transformer, $\tilde{\mathbf{X}}_i^{(te)} \in \mathbb{R}^{T \times N \times d_h}$, is then obtained as:

$$\tilde{\mathbf{X}}_i^{(te)} = \mathbf{A}_i^{(te)} \mathbf{V}_i^{(te)}. \tag{4.12}$$

Then, after processing all spatial nodes i, the final shape will be $\tilde{\mathbf{X}}^{(te)} \in \mathbb{R}^{M \times N \times d_h}$ In a similar way, the spatial transformer layer functions by processing $\tilde{\mathbf{X}}^{(te)}$ through self-attention (following the same equations) to produce $\mathbf{X}^{(sp)}$.This process involves computing the queries, keys, and values as follows:

$$\mathbf{Q}_t^{(sp)} = \tilde{\mathbf{X}}_t^{(te)} \mathbf{W}_Q^{(sp)}, \quad \mathbf{K}_t^{(sp)} = \tilde{\mathbf{X}}_t^{(te)} \mathbf{W}_K^{(sp)}, \quad \mathbf{V}_t^{(sp)} = \tilde{\mathbf{X}}_t^{(te)} \mathbf{W}_V^{(sp)}, \tag{4.13}$$

where $\mathbf{W}_Q^{(sp)}$, $\mathbf{W}_K^{(sp)}$, and $\mathbf{W}_V^{(sp)}$ are trainable parameters. For each timestep t, the matrix $\tilde{\mathbf{X}}_t^{(te)} \in \mathbb{R}^{N \times d_h}$ is transformed into queries, keys, and values. This enables the attention to focus only on the spatial dimension. These transformations facilitate the computation of attention

scores and the subsequent aggregation of features. The attention scores and the output of the spatial transformer can be formulated as:

$$\mathbf{A}_t^{(sp)} = \text{Softmax}\left(\frac{\mathbf{Q}_t^{(sp)}(\mathbf{K}_t^{(sp)})^\top}{\sqrt{d_h}}\right), \tag{4.14}$$

$$\mathbf{X}_t^{(sp)} = \mathbf{A}_t^{(sp)}\mathbf{V}_t^{(sp)}, \tag{4.15}$$

where the attention scores $\mathbf{A}_t^{(sp)} \in \mathbb{R}^{N\times N}$ are calculated using the softmax function to normalize the dot products of the queries and keys, scaled by the square root of the dimension d_h of the hidden layers. This scaling factor helps stabilize the gradients during training. Then, $\mathbf{X}_t^{(sp)}$ is obtained by weighting the value vectors by the attention scores, effectively allowing the model to focus on the most relevant features across spatial dimensions. After all the time steps t, the final output of spatial attention will be $\mathbf{X}^{(sp)} \in \mathbb{R}^{M\times N\times d_h}$.

4.2.3 Spatial Temporal Selective State of Spatial (ST-Mamba block)

As depicted in Figure 4.2, following the adaptive ST-Transformer block, our framework employs a simplified ST-Mamba block. This block features an ST-Mamba layer designed to reduce computational costs and enhance long-term memory. To feed the input to ST-Mamba block, we employ a tensor reshape named as ST-mixer as:

ST-mixer [Shao et al., 2024h]. To effectively blend spatial and temporal data, the ST-Mamba utilizes tensor reshaping, as detailed in Figure 4.2, to transform tensor $\mathbf{X}_t^{(sp)}$ into matrix $\bar{\mathbf{U}}_t$. This transformation involves aligning and concatenating the tensor slices across each time step t as follows:

$$\bar{\mathbf{U}}_t = \text{reshape}(\mathbf{X}_t^{(sp)}). \tag{4.16}$$

Through this reshaping, we obtain a new embedding $\bar{\mathbf{U}}_t$ in $\mathbb{R}^{\tilde{T}\times d_h}$. By flattening the first two dimensions $M \times N$ into $\tilde{T}$, we encapsulate the combined historical temporal length and number of nodes, effectively merging spatial and temporal information into a single matrix. This reshaping

facilitates the unified processing of spatial and temporal information, thereby capturing complex patterns in the data more effectively.

4.2.3.1 ST-Mamba Layer

The ST-Mamba layer, as described in [Shao et al., 2024h], utilizes the discretization of continuous state-space models (SSM). For simplicity, we keep denoting the input to the ST-Mamba block as $\bar{\mathbf{U}}_t$. This processed input is then subjected to the selective state-space model (SSM) layer, where a linear transformation produces:

$$\tilde{\mathbf{X}}_t = Linear(\bar{\mathbf{U}}_t), \tag{4.17}$$

where $\tilde{\mathbf{X}}_t = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{\tilde{T}}]^\top$, and each $\mathbf{x}_t^k$ exists in $\mathbb{R}^{d_h}$ for $k = 1, 2, \ldots, \tilde{T}$, incorporates both spatial and temporal information, enhancing the layer's capability to process data from various sensors over time. The ultimate goal of the ST-Mamba layer is to generate the traffic flow prediction output, $\mathbf{Y}_t^k$, which is structured in $\mathbb{R}^{\tilde{T} \times d_h}$. Each output sequence $\mathbf{y}_k$ within $\mathbf{Y}_t$, corresponding to each timestep from 1 to $\tilde{T}$, is defined as $\mathbf{Y}_t = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{\tilde{T}}]^\top$ where each $\mathbf{y}_k$ is in $\mathbb{R}^{d_h}$, indicating the prediction at step k.

Parameter Initialization and Discretization. We initialize the ST-Mamba parameters and specify the dimensionality. Throughout this paper, we denote state dimension d_{state}. The state transition matrix $\mathbf{A} \in \mathbb{R}^{d_{\text{state}} \times d_{\text{state}}}$ is established using HiPPO initialization Gu et al. [2020] on matrix $\bar{\mathbf{A}}$ to capture long-range dependencies. The matrix $\mathbf{B} \in \mathbb{R}^{d_{\text{state}} \times d_h}$ is initialized as $\mathbf{B}_k = s_B(\mathbf{x}_k)$, where $s_B(\cdot)$ represents a learnable linear projection. Similarly, the output projection matrix $\mathbf{C} \in \mathbb{R}^{d_h \times d_{\text{state}}}$ is computed as $\mathbf{C}_k = s_C(\mathbf{x}_k)$, with $s_C(\cdot)$ also being a learnable linear projection. Then, the step size parameter denoted as $\Delta \in \mathbb{R}^{\mathbb{R}^{d_{\text{state}} \times 1}}$ with $\Delta_k = \tau_\Delta(\text{Parameter} + s_\Delta(\mathbf{x}_k))$ where τ_Δ is the softplus function [4] and with each iteration of k, the $s_\Delta(\cdot)$ is a learnable linear projection on the updated output of y_k. The dimensionality of Δ_k depends on whether the step size is uniform across all dimensions or varies per dimension. Notably, the parameter Δ_k in SSMs serves a similar function to the gating mechanism in RNNs. It controls the balance

[4] The softplus function is a smooth, differentiable activation function used in neural networks, defined as $\text{softplus}(x) = \log(1 + e^x)$. It approximates the Rectified Linear Unit (ReLU) function but provides a continuous gradient, making it useful for gradient-based optimization. The derivative, $\frac{d}{dx}(\text{softplus}(x)) = \frac{1}{1+e^{-x}}$, ensures non-zero gradients for all input values, aiding in the prevention of vanishing gradients during training.

between how much the model should focus on the current input versus retaining information from previous states.

Discretization and Output Computation. According to Gu and Dao [2023], the conversion of continuous-time parameters into discrete-time state-space model (SSM) parameters is detailed. This process is critical for accurate simulations and analyses. $\Delta_t^k \mathbf{B}_t^k$ is the solution of the approximation on $\tilde{\mathbf{B}}_t^k$ using the first-order Taylor series, and $\exp(\cdot)$ denotes the matrix exponential, and $\mathbf{I}$ is the identity matrix of appropriate size. The discretization operations apply to each time step as in Eq. (4.5) and (4.6), resulting in $\tilde{\mathbf{A}}_t^k \in \mathbb{R}^{d_{\text{state}} \times d_{\text{state}}}$ and $\tilde{\mathbf{B}}_t^k \in \mathbb{R}^{d_{\text{state}} \times d_h}$ which facilitate the recurrence within the selective ST-Mamba layer with initial the hidden state $\mathbf{h}_0 = 0$:

$$\mathbf{h}_t^k = \tilde{\mathbf{A}}_t^k \mathbf{h}_t^{k-1} + \tilde{\mathbf{B}}_t^k \mathbf{x}_t^{k-1}, \tag{4.18}$$

$$\mathbf{y}_t^k = \mathbf{C}_t^k \mathbf{h}_t^k. \tag{4.19}$$

This iterative process spans each step from $k = 1, 2, \ldots, \tilde{T}$, ensuring that each data step is transformed through the Mamba layer. The final output, $\mathbf{Y}_t = \text{stack}(\mathbf{y}_t^1, \mathbf{y}_t^2, \ldots, \mathbf{y}_t^{\tilde{T}})$, is reshaped into $\mathbb{R}^{\tilde{T} \times d_{inner}}$ to align with the original input dimensions.

4.2.3.2 ST-Mamba block and Regression Layer

In our ST-MambaSync model, the ST-Mamba block is simplified to include just one normalization layer. We then add the output of the ST-Mixer, denoted as $\bar{\mathbf{X}}$, to form a Residual Network (ResNet) structure. This modification is designed to enhance the model's capability to capture local features effectively. Finally, passing through the regression layer as the decoder to obtain the final prediction results for traffic flow.

Normalization Layer. We simplified the ST-Mamba block with the normalization of layers is crucial for improving the training's stability and efficiency. Specifically, consider an input matrix $\mathbf{Y}$ with dimensions $\mathbb{R}^{\tilde{T} \times d_h}$, where $\tilde{T}$ represents the sequence length or sample count, and

d_h indicates the features' dimensional space. The normalization process is described by:

$$\text{Normalization}(\mathbf{Y}) = \gamma \odot \frac{\mathbf{Y} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta. \tag{4.20}$$

In this formula, μ and σ^2 are the mean and variance computed along the features' dimension d_h, resulting in vectors of size $\tilde{T} \times 1$. The scale (γ) and shift (β) parameters, each sized $1 \times d_h$, are adjustable, and enable to optimize the normalization's impact. This process ensures stability in the model's learning phase while allowing the reintegration of the distribution of the original activation if it improves model performance. The addition of ϵ, a small constant, prevents any division by zero, maintaining numerical stability. Through layer normalization, the model effectively reduces internal covariate shift, enhancing training speed and boosting overall deep learning performance.

Decoding Layer. As in Referring to Figure 4.2, the output from the ST-Mamba block passes through a normalization step before reaching the regression layer, which is structured as follows:

$$\bar{\mathbf{Y}} = \text{Normalization}(\mathbf{Y}) + \bar{\mathbf{X}}, \tag{4.21}$$

$$\mathcal{Y} = FC(\bar{\mathbf{Y}}). \tag{4.22}$$

In this configuration, $FC(\cdot)$ denotes the fully connected layer that processes the normalized data. The resultant $\mathcal{Y}$, existing within the space $\mathbb{R}^{Z \times N \times d}$, marks the culmination of the process. This structured approach showcases how architectural enhancements are designed to improve the training of deep networks and accurately interpret complex, multi-dimensional datasets.

4.2.4 Mamba and Attention are Complement with each other

To prove the complementing nature of the Transformer and Mamba, we are first required to interpret the attention/Transformer as a form of linear regression, where the goal is not a prediction but rather the computation of a weighted sum of the values, with the weights determined by the similarities between the queries and keys which can be showing as follows:

In the realm of traffic flow prediction, it is crucial to model the dynamic transitions of traffic states accurately and efficiently. The Selective State-Space Model (Mamba) presents a novel approach by embodying mechanisms analogous to those in neural network-based attention models, commonly used in machine learning for sequence prediction tasks. This model treats the discrete update equation for the state transition from $\mathbf{h}_t$ to $\mathbf{h}_{t+1}$ over the interval $[t, t+1]$ as an attention mechanism, where $\mathbf{B}_t$, $\mathbf{x}_t$, and $e^{\mathbf{A}\Delta_t}$ correspond to the roles of query, key, and weight transformations in traditional attention mechanisms, respectively. Such a configuration enables the Mamba model to effectively prioritize and weigh the influence of incoming traffic data ($\mathbf{x}_t$) on the evolving traffic state ($\mathbf{h}_{t+1}$). By dynamically adjusting the focus based on each input's relevance to the current state, the model enhances its predictive capabilities.

Proposition 4.1 (Spatial State Models as a Transformer). *The selective state-space model (Mamba) can be interpreted as an attention mechanism, where the discrete update equation for the state transition from* $\mathbf{h}_t$ *to* $\mathbf{h}_{t+1}$ *over the time interval* $[t, t+1]$ *exhibits dynamics analogous to those found in attention mechanisms. The components of the Mamba model's update equation, namely* $\mathbf{B}_t$, $\mathbf{X}_t$, *and* $e^{\mathbf{A}\Delta_t}$, *play roles similar to the query, key, and weight transformations in attention mechanisms, respectively. This interpretation allows the Mamba model to assign attention-like scores to each input based on its relevance to the current state, dynamically adjusting the model's focus and incorporating the previous state information through a residual-like connection.*

Proof. In the Mamba model, the discrete update equation for the state transition from $\mathbf{h}_{t_a}$ to $\mathbf{h}_{t_b}$ over the time interval $[t_a, t_b]$ is given as in Appendix Eq. (A.2) by:

$$\mathbf{h}_{t_{a+1}} = e^{\mathbf{A}\Delta_{t_a}} \left(\mathbf{h}_{t_a} + e^{-\mathbf{A}\Delta_{t_a}} \mathbf{B}_{t_a} \mathbf{X}_{t_a} \Delta_{t_a} \right). \tag{4.23}$$

The output equation for the Mamba model is given by:

$$\begin{aligned} \mathbf{y}_{t_b} &= \mathbf{C}_{t_b} \mathbf{h}_{t_b}, \\ &= \mathbf{C}_{t_b} \left[e^{\mathbf{A}\Delta_{t_a}} \left(\mathbf{h}_{t_a} + \mathbf{B}_{t_a} \mathbf{X}_{t_a} e^{-\mathbf{A}\Delta_{t_a}} \Delta_{t_a} \right) \right]. \end{aligned} \tag{4.24}$$

For simplicity, we take $t_a = t$, $t_b = t + 1$. Then $\mathbf{y}_t$ is the output at time t, and $\mathbf{C}_t$ is the output matrix at time t. Therefore, we can draw an analogy between the components of this update equation and the components of an attention mechanism:

- $\mathbf{Q}_m = \mathbf{C}_{t+1}$: The matrix $\mathbf{C}_{t+1}$ plays a role that is similar to the query matrix in attention.
- $\mathbf{K}_m^\top = \mathbf{B}_t$: The input matrix $\mathbf{B}_t$ plays a role similar to the key matrix in attention, as it transforms the input x_t.
- $\mathbf{W} = e^{\mathbf{A}\Delta_t}$: The state transition matrix exponential over the time step Δ_t acts like the weight matrix in attention, governing the influence of the previous state on the current state.
- $\mathbf{V}_m = \mathbf{X}_t$: The output matrix is treated as a value matrix in attention.
- $\beta = \Delta_t$: The scaling factor β is set to the time step Δ_t, which controls the influence of the previous state on the current state update.

Then, we can reformulate the state update equation to resemble an attention mechanism:

$$\mathbf{y}_{t+1} = \mathbf{W}\mathbf{Q}_m\mathbf{h}_t + \left(\mathbf{Q}_m\mathbf{K}_m^T\right)\mathbf{V}_m\beta. \tag{4.25}$$

The first term, $\mathbf{W}\mathbf{Q}_m\mathbf{h}_t$, incorporates the previous state $\mathbf{h}_t$ into the update. It applies a transformation to the previous state using the weight matrix ($\mathbf{W}$) and the query matrix ($\mathbf{Q}_m$), reflecting how the current state influences the next state update. This term is structured similarly to a residual connection in a Residual Network (ResNet), enhancing the state with a weighted version of itself. The second term, $\left(\mathbf{Q}_m\mathbf{K}_m^T\right)\mathbf{V}_m\beta$, captures the attention-like behavior. It computes a weighted sum of the input values $\mathbf{V}_m$, where the weights are determined by the product of the query ($\mathbf{Q}_m$) and key ($\mathbf{K}_m^T$) matrices. This term assigns attention scores to each input based on its relevance to the current state, adjusted by the scaling factor (β). This term is structured similarly to a residual connection in a Residual Network (ResNet), where the state is updated by adding a transformed version of itself.

□

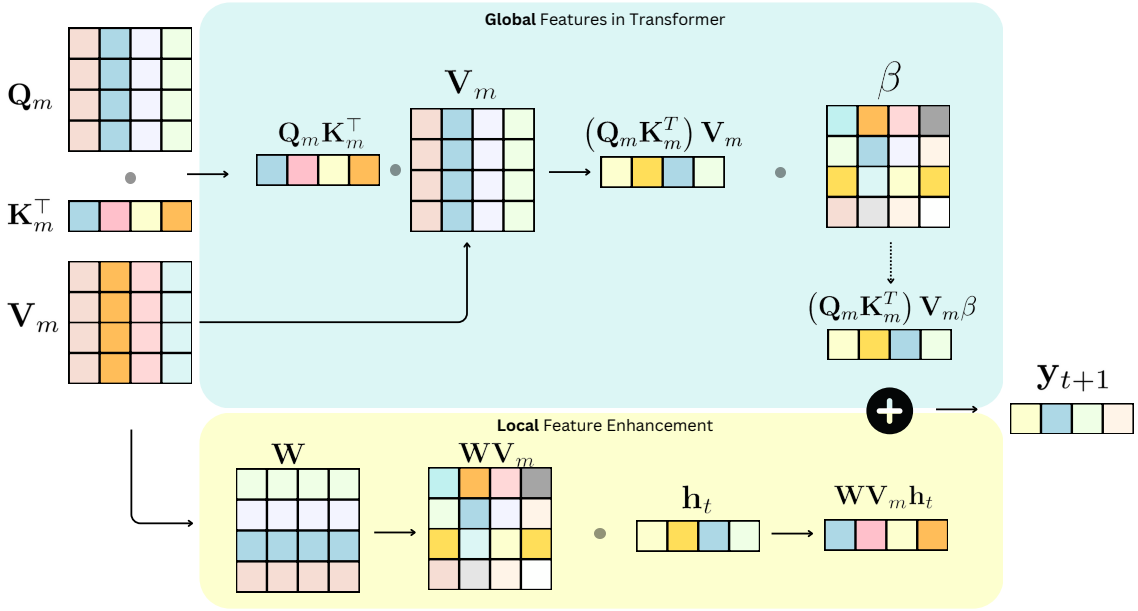


FIGURE 4.3. The complementary roles of global attention in Transformers and local feature enhancement in Mamba

In order to gain a deeper understanding of the proposed ST-MambaSync, we will now explore the complementary nature of Mamba and Transformer. Figure. 4.3 illustrates the mathematical and structural relationship between Mamba and Transformer mechanisms, emphasizing Mamba's ability to act like a Transformer with added local focus. The top section of the figure, shaded in blue, depicts the global feature extraction performed by the Transformer. The Transformer uses an attention mechanism, where the query (Q_m), key (K_m), and value (V_m) matrices interact to capture dependencies across the entire input sequence. This process efficiently models long-range dependencies, but it may overlook finer, localized patterns.

The bottom section of the figure, shaded in yellow, represents Mamba's local feature enhancement mechanism. Here, a residual connection structure, similar to ResNet, is incorporated. The Mamba mechanism enhances the processing of localized information, ensuring that key details are not lost when learning from spatial-temporal data. Specifically, the W matrix operates on V_m, focusing on extracting and refining local features. The result is a representation $WV_m h_t$ that preserves both the overall context and specific local information.

PROPOSITION 4.2 (Mamba and Transformer: Local and Global Attention). *Mamba and Transformer offer complementary attention mechanisms. Mamba, with ResNet, provides local attention*

by adjusting focus based on the current state and integrating previous state information. In contrast, the Transformer's global attention captures long-range dependencies. This combination enhances both local context processing and long-range dependency capture, enabling efficient neural network information processing.

PROOF. As shown in the Proposition 4.1, the Mamba model can be interpreted as an attention mechanism, with the components of its update equation ($\mathbf{B}_t$, $\mathbf{x}_t$, and $e^{\mathbf{A}\Delta_t}$) playing roles analogous to the query, key, and weight transformations in attention mechanisms. This allows Mamba to assign attention-like scores to each input based on its relevance to the current state.

However, Mamba's attention is inherently local, focusing on the current state and the immediate inputs. By incorporating a residual connection, as in ResNet architectures, Mamba can effectively assign weights to this local attention, allowing the model to adjust its focus based on the current context dynamically.

On the other hand, Transformer architectures are designed to capture global attention, allowing the model to attend to all inputs in a sequence simultaneously. This is achieved through the use of self-attention mechanisms, which compute attention scores between all pairs of tokens in the input sequence.

The combination of Mamba's local attention and Transformer's global attention creates a complementary system that effectively balances the need for capturing long-range dependencies and attending to local context. This synergy enables more comprehensive and efficient processing of information in neural networks. □

COROLLARY 4.3. *Mamba and Transformer can be seen as complementary attention mechanisms, with Mamba focusing on local attention, which is instinct with a residual network (ResNet), and Transformer focusing on global attention. Mamba's local attention-like behavior allows the model to adjust its focus based on the current state dynamically, integrate previous state information, and enhance local information processing without losing the memory of the previous states. Transformer's global attention mechanism enables capturing long-range dependencies. This combination balances attending to the local context, preserving memory, and capturing*

long-range dependencies, enabling more comprehensive and efficient information processing in neural networks.

4.3 Experiment

4.3.1 Data Description and Baseline Models

Datasets. Table 4.2 presents a detailed comparison of six benchmark traffic forecasting datasets—PEMS03, PEMS04, PEMS07, PEMS08, METR-LA, and PeMS-BAY—each crucial for evaluating the performance of traffic prediction models. These datasets are widely used in traffic research and share a consistent data granularity of 5-minute intervals, resulting in 12 data frames recorded every hour. The table highlights key attributes such as the number of nodes (sensors), total timesteps, collection period, missing data ratio, and recorded signals, providing a comprehensive overview for modeling and analysis.

The METR-LA dataset [Li et al., 2018] contains traffic speed data collected from 207 loop detectors distributed across the Los Angeles County road network, spanning from March to June 2012. Data is recorded at 5-minute intervals, totaling 34,272 timesteps. Similarly, the PeMS-BAY dataset [Li et al., 2018] features traffic speed data from 325 sensors in the Bay Area, recorded from January 1 to May 31, 2017, with 52,116 records also taken at 5-minute intervals.

The PEMS03, PEMS04, PEMS07, and PEMS08 datasets [Song et al., 2020] are obtained from the Caltrans Performance Measurement Systems (PeMS) and represent different regions and time periods across California. Specifically, the PEMS03 dataset includes data from 358 sensors collected between September and November 2018. The original data is recorded at 30-second intervals and aggregated to 5-minute intervals, capturing essential traffic metrics such as speed, volume, and occupancy. The PEMS04 dataset, covering the San Francisco Bay Area, includes data from 307 sensors collected between January and February 2018, also aggregated into 5-minute intervals. The PEMS07 dataset encompasses traffic data from 883 sensors collected from May to August 2017, while PEMS08 consists of data from 170 sensors recorded during July and August 2016. Both PEMS07 and PEMS08 datasets provide traffic flow (F), speed (S),

and occupancy data (O) aggregated to 5-minute intervals, making them suitable for evaluating model performance under various traffic conditions.

TABLE 4.2. Statistics of traffic forecasting datasets.

Dataset	**PEMS03**	**PEMS04**	**PEMS07**	**PEMS08**	**METR-LA**	**PeMS-BAY**
# of nodes	358	307	883	170	207	325
# of timesteps	26,208	16,992	28,224	17,856	34,272	52,116
Granularity	5min	5min	5min	5min	5min	5min
Start time	9/1/2018	1/1/2018	5/1/2017	7/1/2016	3/1/2012	1/1/2017
End time	11/30/2018	2/28/2018	8/31/2017	8/31/2016	6/30/2012	5/31/2017
Missing ratio	0.672%	3.182%	0.452%	0.696%	8.11%	0.003%
# Signals	F	F, S, O	F	F, S, O	S	S

Baseline Models. In our comparative analysis, we evaluate the performance of our proposed approach against a comprehensive set of baselines within the traffic forecasting domain. The Historical Index (HI) [Cui et al., 2021] serves as the conventional benchmark, reflecting standard industry practices. Our examination extends to various Spatial-Temporal Graph Neural Networks (STGNNs), including GWNet [Wu et al., 2020a], which proposes a graph neural network framework that automatically extracts uni-directed relations among variables, addressing the limitation of existing methods in fully exploiting latent spatial dependencies in multivariate time series forecasting. DCRNN [Li et al., 2018] introduces the Diffusion Convolutional Recurrent Neural Network for traffic forecasting, capturing both spatial and temporal dependencies. AGCRN [BAI et al., 2020] incorporates adaptive modules to capture node-specific patterns and infer inter-dependencies among traffic series, providing fine-grained modeling of spatial and temporal dynamics in traffic data. STGCN [Yu et al., 2018a] proposes a deep learning framework that integrates graph convolutions for spatial feature extraction and gated temporal convolutions for temporal feature extraction. GTS [Shang et al., 2021] presents a method for forecasting multiple interrelated time series by learning a graph structure simultaneously with a Graph Neural Network (GNN), addressing the limitations of previous methods. MTGNN [Wu et al., 2020a] introduces a graph neural network framework that automatically extracts uni-directed relations among variables, capturing both spatial and temporal dependencies. GMAN [Zheng et al., 2020a] incorporates spatial and temporal attention mechanisms to capture dynamic correlations among traffic sensors.

Recognizing the potential of Transformer-based models in time series forecasting, we particularly focus on PDFormer [Jiang et al., 2023], which introduces a traffic flow prediction model that captures dynamic spatial dependencies, long-range spatial dependencies, and the time delay in traffic condition propagation, and STAEformer [Liu et al., 2023b], which proposes a spatio-temporal adaptive embedding that enhances the performance of vanilla transformers for traffic forecasting. Additionally, we explore STNorm [Deng et al., 2021], which leveraging spatial and temporal normalization modules to refine the high-frequency and local components underlying the raw data, and STID [Shao et al., 2022], which addresses the indistinguishability of samples in both spatial and temporal dimensions by attaching spatial and temporal identity information to the input data. This diverse range of models allows for a robust validation of our proposed method's capabilities.

4.3.2 Experiment Setup

Implementation. All experiments were executed on a machine featuring an RTX 3090 GPU with 24GB of memory and a 15-core CPU. The datasets PEMS-BAY, PEMS03, PEMS04, PEMS07, and PEMS08 were split into training, validation, and test sets. PEMS-BAY was divided in a 7:1:2 ratio, while PEMS03, PEMS04, PEMS07, and PEMS08 were partitioned using a 6:2:2 ratio.

Hyperparameter Selection. The selection of hyperparameters was based on extensive experimentation and prior research findings to achieve an optimal balance between model complexity and training efficiency. The embedding dimension (d_f) was set to 24 to provide a compact yet effective representation of input features. The attention dimension (d_a) was set to 80, which was determined to be optimal through experiments to strike a balance between learning sufficient temporal patterns without incurring high computational costs. For the inner dimensions, we used $d_{inner} = d_h \times expand$, where $expand = 2$, ensuring that the feed-forward layers had sufficient capacity to model complex relationships. The hidden state dimension (d_{state}) was set to 64 after evaluating various values to capture both local and global information effectively.

The model architecture consisted of one spatial and one temporal transformer layer, each with four attention heads. Four attention heads were chosen to allow the model to capture diverse aspects of the data effectively while balancing memory usage. Additionally, one ST-Mamba layer was included to enhance the local feature focus. Both the input and forecast horizon were set to 1 hour, corresponding to $M = Z = 12$ time steps, with each step representing a 5-minute interval. This choice was driven by the nature of traffic data, where short-term predictions are crucial for real-time decision-making.

The optimization was carried out using the Adam optimizer with an initial learning rate of 0.001, chosen for its ability to converge quickly and handle sparse gradients. The learning rate was decreased over time based on a plateau scheduling strategy to ensure convergence and avoid overshooting the optimal solution. A batch size of 16 was used, which was determined to be effective in maintaining computational efficiency while ensuring stable gradient updates. To further enhance training efficiency, an early stopping mechanism was employed, terminating training if the validation error did not improve after 30 consecutive iterations. This prevented overfitting and saved computational resources. The code is available at https://github.com/superca729/ST-MAMBASYNC.git.

Metric. To assess the effectiveness of traffic forecasting techniques, three widely used metrics are applied: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). These metrics comprehensively assess model accuracy and the extent of errors (see Section 3.5.4).

4.3.3 Performance Evaluation

To assess the effectiveness of the ST-MambaSync model, we utilized six real-world datasets varying significantly in complexity and scale. The datasets range from METR-LA, which includes 207 sensors, to PEMS07, encompassing 883 sensors. This selection provides a broad spectrum of urban traffic patterns and sensor network densities, potentially impacting the

predictive performance of the model. The most outstanding results across these evaluations are denoted in **Bold** to highlight superior performance.

Analysis of Performance Metrics. Table 4.3 presents a performance comparison across four different PEMS datasets: PEMS03, PEMS04, PEMS07, and PEMS08. The metrics evaluated are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), which reflect the predictive accuracy of the models across different regions and conditions.

Based on the results, the ST-MambaSync model demonstrates strong performance across these datasets, with a focus on three key metrics: MAE, RMSE, and MAPE. Particularly notable is its performance on the PEMS08 dataset, where ST-MambaSync achieves the lowest MAE of 13.30, RMSE of 23.14, and MAPE of 8.80%. This indicates its robustness in environments characterized by fewer nodes and moderate missing data. The superior performance on PEMS08 suggests that ST-MambaSync is well-suited for datasets with stable traffic patterns and organized configurations, contributing to reduced prediction errors.

For PEMS04, which has a relatively high missing data ratio of 3.182%, ST-MambaSync still maintains competitive MAE and RMSE scores. This underscores the model's resilience to data incompleteness—an important advantage for real-world scenarios where sensor data may be sporadic or unreliable. Furthermore, the analysis across the PEMS datasets reveals distinct performance patterns. ST-MambaSync performs optimally on PEMS08 (170 nodes), achieving MAE = 13.30 and MAPE = 8.80%, thanks to the dataset's stable patterns, clear linear mean-standard deviation relationship with mean Coefficient of Variation (CV = 0.464) (see Figure 4.5), and well-organized manifold structure (see Figure 4.4). The model also demonstrates scalability and robustness on PEMS04 (307 nodes) and PEMS07 (883 nodes).

However, PEMS03 poses unique challenges, as evidenced by its higher variability and less predictable traffic patterns. This dataset has the highest mean Coefficient of Variation (CV = 0.597) and a wide CV range (0.345–1.012), along with 617 high-variance sensors. Figure 4.5 visualizes the scattered and non-linear relationship between mean traffic flow and standard deviation,

highlighting this dataset's inherent unpredictability. Moreover, the Isomap dimensionality reduction analysis (Figure 4.4) shows fragmented and scattered distributions for PEMS03, in stark contrast to the more structured and continuous patterns seen in PEMS04, PEMS07, and PEMS08.

Despite these challenges, ST-MambaSync maintains the top four competitive performance on PEMS03 (MAE = 15.30, RMSE = 27.47), leveraging its complementary architecture to handle irregular traffic patterns. The Mamba component effectively extracts local features, while the Transformer manages global dependencies, striking a balance that accommodates PEMS03's chaotic nature. This comprehensive analysis highlights ST-MambaSync's adaptability across various network conditions while underscoring the complexities that make PEMS03 particularly difficult for traffic prediction models.

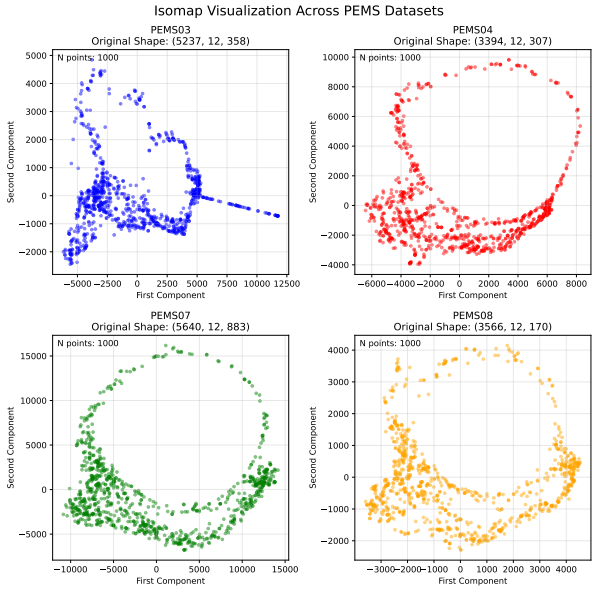


FIGURE 4.4. Isomap visualization comparing traffic pattern distributions across PEMS datasets.

Detailed Analysis Based on Time Horizons. Tables 4.4 and 4.5 shows the performance of different models on the METR-LA and PEMS-BAY datasets across three prediction horizons: 15 minutes, 30 minutes, and 60 minutes. The metrics used are MAE, RMSE, and MAPE. These tables highlights how each model's performance changes with the forecasting interval, providing insights into the stability of the models over time.

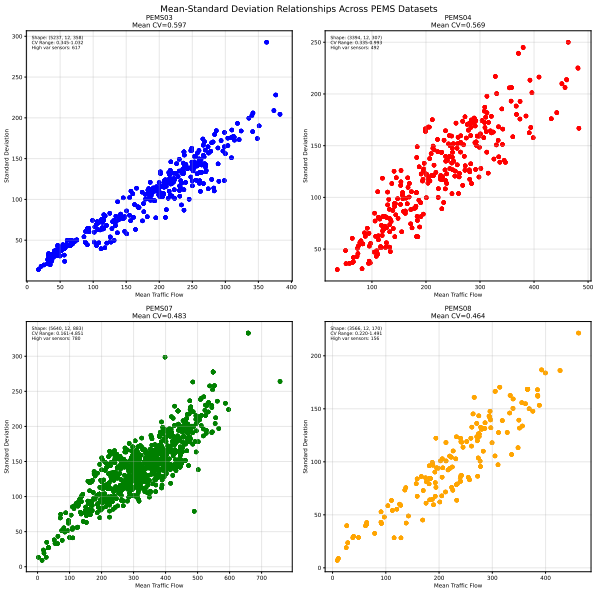


FIGURE 4.5. Visualization of the Mean–Standard Deviation relationship for traffic flow across different PEMS datasets. The PEMS03 dataset (blue dots) demonstrates a more dispersed and non-linear pattern, with a higher coefficient of variation (Mean CV = 0.597) compared to other datasets like PEMS08 (orange dots). This indicates greater variability and less predictability in traffic patterns, highlighting the challenges in modeling and forecasting traffic for PEMS03.

TABLE 4.3. Performance comparison of models on PEMS datasets

Model	**PEMS03 (N=358)**			**PEMS04 (N=307)**			**PEMS07 (N=883)**			**PEMS08 (N=170)**		
	MAE	**RMSE**	**MAPE (%)**	**MAE**	**RMSE**	**MAPE (%)**	**MAE**	**RMSE**	**MAPE (%)**	**MAE**	**RMSE**	**MAPE (%)**
HI	32.62	49.89	30.60	42.35	61.66	29.92	49.03	71.18	22.75	36.66	50.45	21.63
GWNet	**14.59**	**25.24**	15.52	18.53	29.92	12.89	20.47	33.47	8.61	14.40	23.39	9.21
DCRNN	15.54	27.18	15.62	19.63	31.26	13.59	21.16	34.14	9.02	15.22	24.17	10.21
AGCRN	15.24	26.65	15.89	19.38	31.25	13.40	20.57	34.40	8.74	15.32	24.41	10.03
STGCN	15.83	27.51	16.13	19.57	31.38	13.44	21.74	35.27	9.24	16.08	25.39	10.60
GTS	15.41	26.15	15.39	20.96	32.95	14.66	22.15	35.10	9.38	16.49	26.08	10.54
MTGNN	14.85	25.23	14.55	19.17	31.70	13.37	20.89	34.06	9.00	15.18	24.24	10.20
STNorm	15.32	25.93	14.37	18.96	30.98	12.69	20.50	34.66	8.75	15.41	24.77	9.76
GMAN	16.87	27.92	18.23	19.14	31.60	13.19	20.97	34.10	9.05	15.31	24.92	10.13
PDFormer	14.94	25.39	15.82	18.36	30.03	12.00	19.97	32.95	8.55	13.58	23.41	9.05
STID	15.33	27.40	16.40	18.38	29.95	12.04	19.61	32.79	8.30	14.21	23.28	9.27
STAEformer	15.35	27.55	**15.18**	18.22	30.18	**11.98**	**19.14**	32.60	8.01	13.46	23.25	8.88
ST-MambaSync	15.30	27.47	**15.18**	**18.20**	**29.85**	12.00	**19.14**	**32.58**	**7.97**	**13.30**	**23.14**	**8.80**

As the performance results shown in Tables 4.4 and 4.5, the ST-MambaSync model demonstrates competitive accuracy across both datasets and all time horizons, particularly excelling on the PEMS-BAY dataset. Its performance remains stable even as the forecasting interval extends from 15 to 60 minutes, highlighting its robustness in both short-term and long-term forecasting.

On METR-LA, ST-MambaSync consistently achieves the lowest or near-lowest MAE, RMSE, and MAPE values, underscoring its ability to adapt to the unique spatial-temporal dynamics of each dataset.

In contrast, some models show significant performance drops as the horizon increases, particularly on METR-LA, where ST-MambaSync's architecture appears to better capture long-range dependencies. The visual comparison emphasizes ST-MambaSync's strength in maintaining prediction stability and accuracy across varied time horizons, demonstrating its robustness against other models that struggle with longer forecasting intervals on complex urban datasets.

TABLE 4.4. Performance comparison of models on the METR-LA dataset.

Model	**METR-LA (15 min)**			**METR-LA (30 min)**			**METR-LA (60 min)**		
	MAE	**RMSE**	**MAPE (%)**	**MAE**	**RMSE**	**MAPE (%)**	**MAE**	**RMSE**	**MAPE (%)**
HI	6.80	14.21	16.72	6.80	14.21	16.72	6.80	14.20	10.15
GWNet	2.69	5.15	6.99	3.08	6.20	8.47	3.51	7.28	9.96
DCRNN	2.67	5.16	6.86	3.12	6.27	8.42	3.54	7.47	10.32
AGCRN	2.85	5.53	7.63	3.20	6.52	9.00	3.59	7.45	10.47
STGCN	2.75	5.29	7.10	3.15	6.35	8.62	3.60	7.43	10.35
GTS	2.75	5.27	7.12	3.14	6.33	8.62	3.59	7.44	10.25
MTGNN	2.69	5.16	6.89	3.05	6.13	8.16	3.47	7.21	9.70
STNorm	2.81	5.57	7.40	3.18	6.59	8.47	3.57	7.51	10.24
GMAN	2.80	5.55	7.41	3.12	6.49	8.73	3.44	7.35	10.07
PDFormer	2.83	5.45	7.77	3.20	6.46	9.19	3.62	7.47	10.91
STID	2.82	5.53	7.75	3.19	6.57	9.39	3.55	7.55	10.95
STAEformer	2.65	5.11	6.85	2.97	**6.00**	8.13	3.34	7.02	**9.70**
ST-MambaSync	**2.63**	**5.05**	**6.80**	**2.91**	6.07	**8.08**	**3.31**	**7.02**	**9.70**

4.3.4 Ablation Study and Case Study

In this subsection, we conduct various ablation studies, including adjustments to the layer configurations in both the attention-based and Mamba-based models. Additionally, we analyze the trade-offs between accuracy and computational efficiency.

TABLE 4.5. Performance comparison of models on the PEMS-BAY dataset.

Model	PEMS-BAY (15 min)			PEMS-BAY (30 min)			PEMS-BAY (60 min)		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
HI	3.06	7.05	6.85	3.06	7.04	6.84	3.05	7.03	6.83
GWNet	1.30	2.73	2.71	1.63	3.73	3.73	1.99	4.60	4.71
DCRNN	1.31	2.76	2.73	1.65	3.75	3.71	1.97	4.60	4.68
AGCRN	1.35	2.88	2.91	1.67	3.82	3.81	1.94	4.50	4.55
STGCN	1.36	2.88	2.86	1.70	3.84	3.79	2.02	4.63	4.72
GTS	1.37	2.92	2.85	1.72	3.86	3.88	2.06	4.60	4.88
MTGNN	1.33	2.80	2.81	1.66	3.77	3.75	1.95	4.50	4.62
STNorm	1.33	2.82	2.76	1.65	3.77	3.66	1.92	4.45	4.46
GMAN	1.35	2.90	2.87	1.65	3.82	3.74	1.91	4.49	4.52
PDFormer	1.32	2.83	2.78	1.64	3.79	3.71	1.91	4.43	4.51
STID	1.31	2.79	2.78	1.64	3.73	3.73	1.91	4.42	4.55
STAEformer	1.31	2.78	2.76	1.62	3.68	3.62	1.88	4.34	4.41
ST-MambaSync	**1.30**	**2.75**	**2.75**	**1.63**	**3.62**	**3.61**	**1.87**	**4.30**	**4.40**

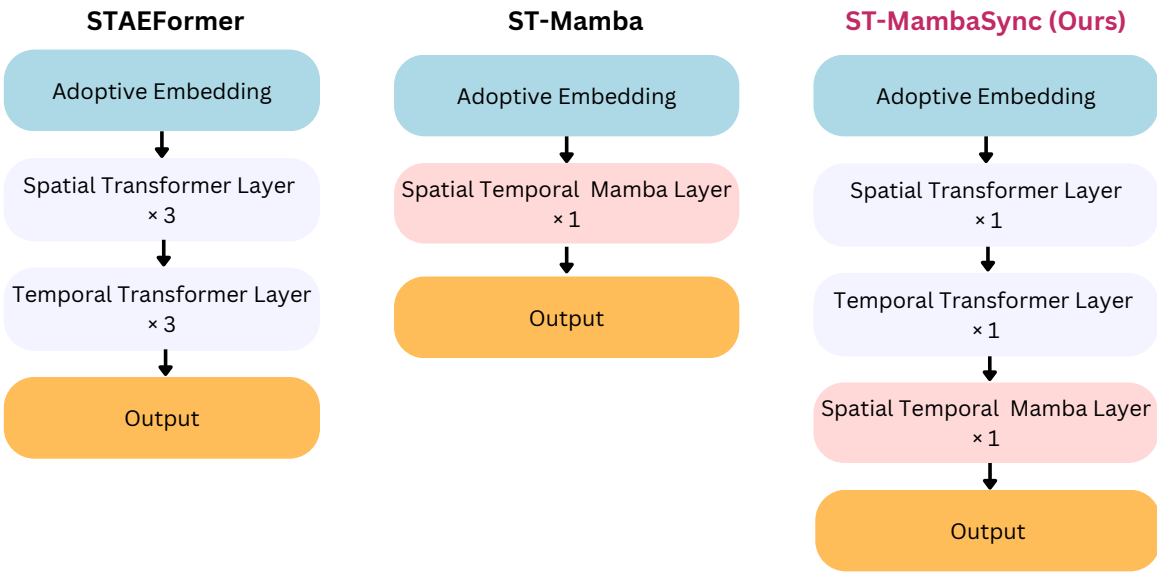


FIGURE 4.6. The difference of STAEformer, ST-Mamba, and ST-MambaSync

4.3.4.1 Ablation Study

We conducted a comparative analysis with the attention-based model (STAEFormer) [Liu et al., 2023b], the Mamba-based model (ST-Mamba) [Shao et al., 2024h], and our newly proposed hybrid model (ST-MambaSync). Figure 4.6 illustrates the structures of STAEformer, ST-Mamba, and ST-MambaSync, highlighting their distinct configurations. STAEformer employs three

spatial transformer layers followed by three temporal transformer layers, effectively capturing complex global dependencies but at a high computational cost. In contrast, ST-Mamba uses a single Spatial-Temporal Mamba Layer, combining spatial and temporal features into a simplified framework that significantly reduces computational load while still modeling key relationships but only focusing on the local feature. ST-MambaSync integrates elements from both approaches, featuring one spatial transformer layer, one temporal transformer layer, and a Spatial-Temporal Mamba Layer. This configuration allows ST-MambaSync to leverage the powerful global and local feature extraction of transformers while also benefiting from the computational efficiency and localized focus of the Mamba Layer. The result is a model that achieves an optimal balance between accuracy and computational efficiency, making it particularly effective for real-time traffic prediction tasks. To investigate the impact of the number of attention layers on performance, we modified the STAEFormer by varying the number of layers. Table 4.6 presents the performance comparison among various configurations of STAEformer (originally consisting of three layers of attention), ST-Mamba (one layer of attention and one Mamba layer), and the proposed ST-MambaSync model. Metrics such as MAE, RMSE, MAPE, FLOPS, inference time, and training time are evaluated. The comparison includes different attention layer configurations (A1, A2, A3) for STAEformer and various Mamba layer configurations (M1, M2, M3) for ST-Mamba, highlighting the effects of layer depth on both accuracy and computational efficiency. Here, A# represents the number of attention layers, while M# indicates the number of Mamba layers. ST-MambaSync configurations combine Mamba and Transformer layers, achieving a balanced performance with reduced computational costs and inference times, while retaining high accuracy. The results demonstrate that ST-MambaSync (M1 & A1 layers) provides the best trade-off between accuracy and efficiency.

The comparison underscores the trade-offs between prediction accuracy and computational efficiency across the models. STAEFormer, with multiple attention layers, achieves a competitive mean absolute error (MAE) range of 13.49 to 13.77 but requires substantial computational resources (1.49 to 4.24 FLOPS) and exhibits longer inference (1.20s to 3.03s) and training durations (14s to 36s). Conversely, ST-Mamba, which incorporates a single Mamba layer and no attention layers, shows a high accuracy (MAE of 13.40) with significantly lower

computational demands (FLOPS of 0.43), making it an effective option for efficient traffic flow prediction. The ST-MambaSync model, combining one Mamba layer with one attention layer, outperforms others by achieving the lowest prediction error (MAE of 13.30) while maintaining reasonable computational efficiency (1.49 FLOPS). Its inference and training times are 2.65s and 29s, respectively, indicating its suitability for practical implementation in real-world traffic management systems. Although the improvement in prediction accuracy between STAEFormer (A1 layer) and ST-MambaSync (M1 & A1 layer) appears modest (MAE reduction from 13.77 to 13.30), the consistent enhancement across multiple metrics, combined with a stable computational cost, highlights the effectiveness of the integration. The balanced approach of ST-MambaSync demonstrates how combining Mamba and attention mechanisms can yield robust predictions while optimizing computational efficiency, ultimately making it a valuable tool for practical, large-scale applications.

TABLE 4.6. Performance comparison on the PEMS08 dataset.

Model	**MAE**	**RMSE**	**MAPE (%)**	**FLOPS(M)**	**Inference (s)**	**Train (s)**
STAEformer (A3 layers)	13.49	23.30	8.84	4.24	3.03	36
STAEformer (A2 layers)	13.54	23.47	8.89	2.87	2.09	23
STAEformer (A1 layer)	13.77	23.27	9.16	1.49	1.20	14
ST-Mamba(M3 layer)	13.45	23.08	8.96	1.07	3.64	42
ST-Mamba(M2 layer)	13.43	23.14	8.95	0.75	2.56	28
ST-Mamba (M1 layer)	13.40	23.20	9.00	**0.43**	**1.18**	**14**
ST-MambaSync (M1 & A1 layer)	**13.30**	**23.14**	**8.80**	1.49	2.65	29
ST-MambaSync (M1 & A2 layers)	13.37	23.42	8.98	2.87	3.40	33
ST-MambaSync (M2 & A1 layers)	13.45	24.16	10.98	1.49	2.96	30

4.3.4.2 The Complementary Power Analysis for ST-MambaSync with ST-Mamba Fusion with Transformer

In this section, we analyze the ST-MambaSync model's ability as illustrated in Figures 4.7 and 4.8, which illustrate the process of traffic flow data by examining the intermediate outputs at various stages: the initial input tensor, the tensor after the Transformer block (attention), and the tensor after the ST-Mamba block. The visualizations provided in the figures illustrate these

stages, where the x-axis represents the sensor nodes and the y-axis represents the time steps, which are 5 mins for each interval that spans 12 intervals.

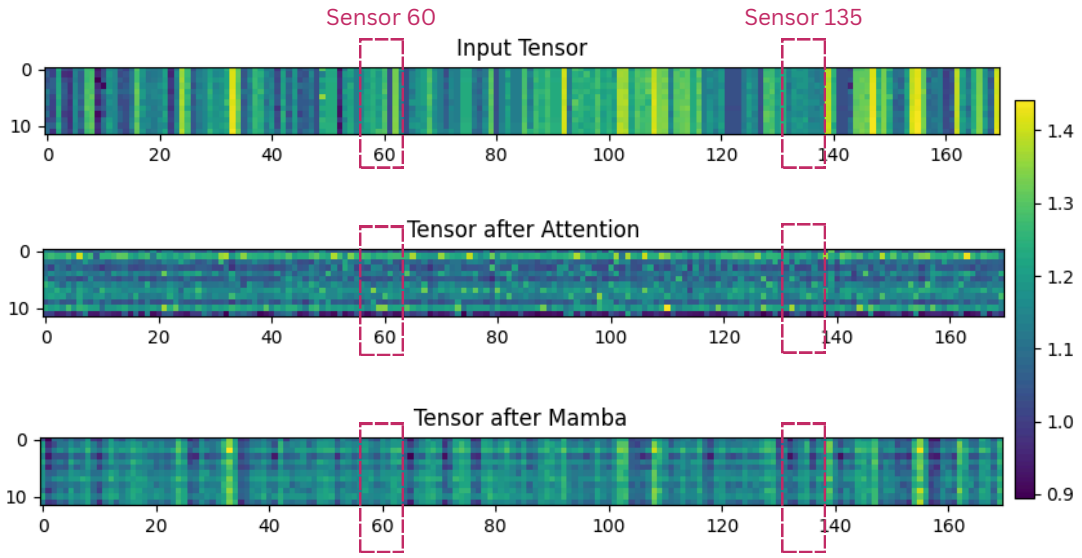


FIGURE 4.7. Visualization of ST-MambaSync at a random time step, showing the initial traffic data input and the transformations after the Transformer block and ST-Mamba block on the PEMS08 dataset.

Initial Input Tensor. The "Input Tensor" visualization displays the raw traffic data fed into the model. Each cell represents the traffic condition recorded by a specific sensor node at a given time step. Notably, sensor 60 shows a lower scale value of around 0.9, indicating lower traffic intensity than other sensors. This initial input tensor is the baseline for understanding how the model processes and transforms the data through subsequent stages.

Tensor after Attention. As in Figure 4.7 and Figure 4.8, following the Transformer block, the "Tensor after Attention" visualization reveals the effect of the attention mechanism on the traffic data. The attention mechanism, adept at capturing long-range dependencies and global context, uniformly modifies values across all features. This is evident as sensor 60 for both Figures 4.7 and 4.8, which initially had lower traffic intensity, shows a more balanced and adjusted distribution of values post-attention. This transformation underscores the attention mechanism's ability to incorporate global information by considering relationships between different time steps and spatial relationships among sensor nodes.

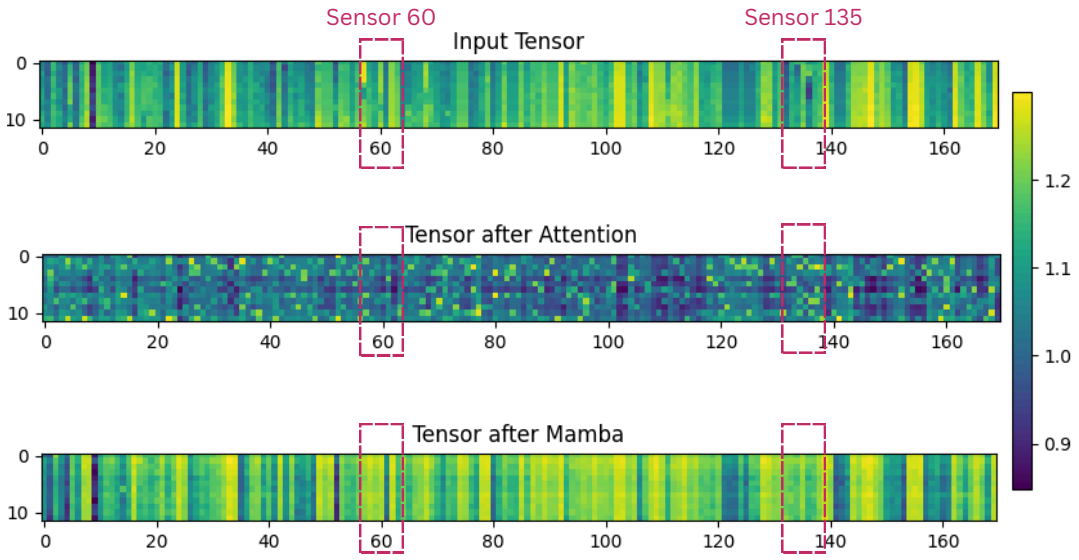


FIGURE 4.8. Visualization of ST-MambaSync at a random time step, showing the initial traffic data input and the outputs after the Transformer block and ST-Mamba block on the PEMS08 dataset.

Tensor after Mamba. The "Tensor after Mamba" visualization demonstrates the impact of the ST-Mamba block on the data. This block is designed to capture local spatial-temporal dependencies and extract fine-grained features. The results highlight how the ST-Mamba block refines the representation further. For instance, in Figure 4.7 (sensor 135) which showed evenly distributed values after the attention block, now exhibits a more distinct and localized scale below 1.0 after passing through the ST-Mamba block. A similar observation can be made in Figure 4.8, where the output for sensor 135 after the Transformer block is evenly distributed. However, the output from the ST-Mamba block enhances local information, resulting in a pattern that closely resembles the initial input. This adjustment indicates that the ST-Mamba block effectively captures and emphasizes local traffic patterns, refining the data representation by highlighting specific features based on local dynamics and avoiding overemphasizing global context.

Complementary Functionality of Transformer and ST-Mamba Blocks. The combined architecture of the Transformer block and the ST-Mamba block effectively captures both global and local dependencies in traffic flow data. The Transformer block's attention mechanism provides a global contextual understanding by uniformly modifying values based on relationships

across the entire dataset. In contrast, the ST-Mamba block specializes in detailed, localized processing, refining the representation to highlight local patterns and dependencies. To better

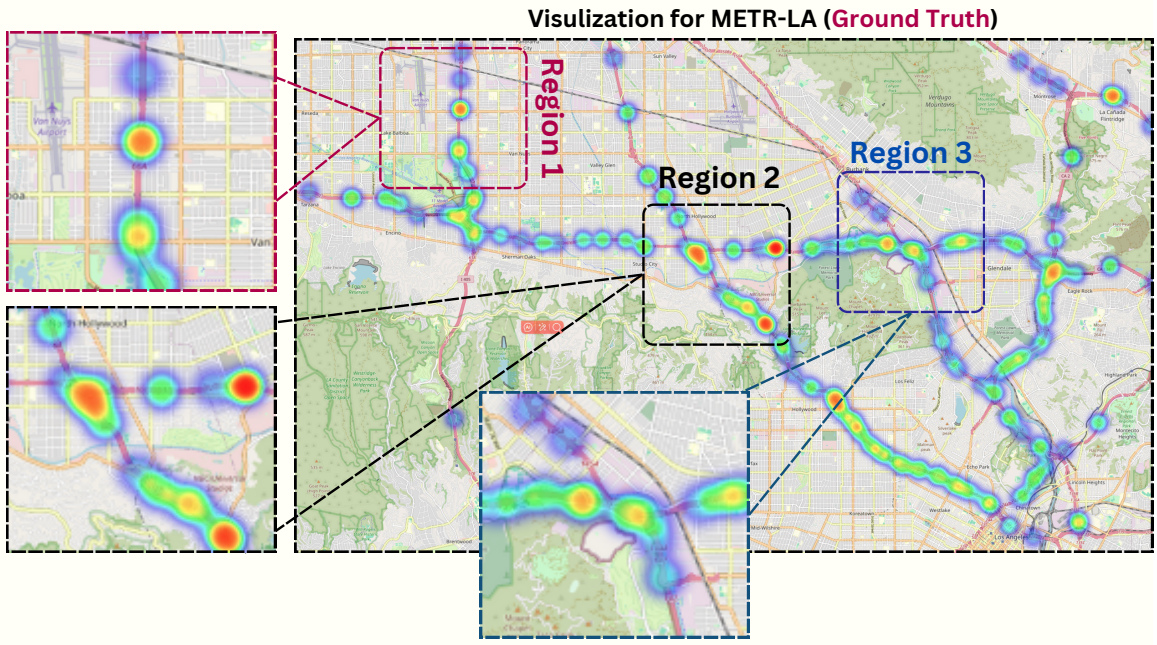


FIGURE 4.9. Heatmap visualization for the METR-LA dataset (ground truth) with a focus on key regions. The heatmap visualizes the intensity of traffic, with warmer colors (red, yellow) indicating higher traffic volumes.

understand the complement power of ST-Mamba and Transformer, it is essential to analyze their performance across various regions with differing traffic densities. Figure 4.9 , Figure 4.10, and Figure 4.11 display heatmaps overlaid on a map of Los Angeles, highlighting traffic patterns based on the METR-LA dataset. Three specific regions are delineated: Region 1 (red outline), covering the northern area with high traffic density; Region 2 (black outline), representing the central region with significant traffic congestion; and Region 3 (blue outline), indicating the eastern area with moderate traffic activity. Figure 4.9 represents the actual traffic data, showing high-density traffic in Region 1, significant congestion in Region 2, and moderate traffic in Region 3. Figure 4.10 represents our proposed ST-MambaSync model that combines the ST-Mamba block with a Transformer. It effectively captures the traffic patterns, closely aligning with the Ground Truth across all three regions. Figure 4.11 is the transformer-based model (STAEFormer [Liu et al., 2023b]); this can be treated as ST-MambaSync without the ST-Mamba block but with higher layers (3 layers) on the Transformer block. While it identifies major traffic patterns, its accuracy and local focus are less precise than the combined model. In Figure 4.12

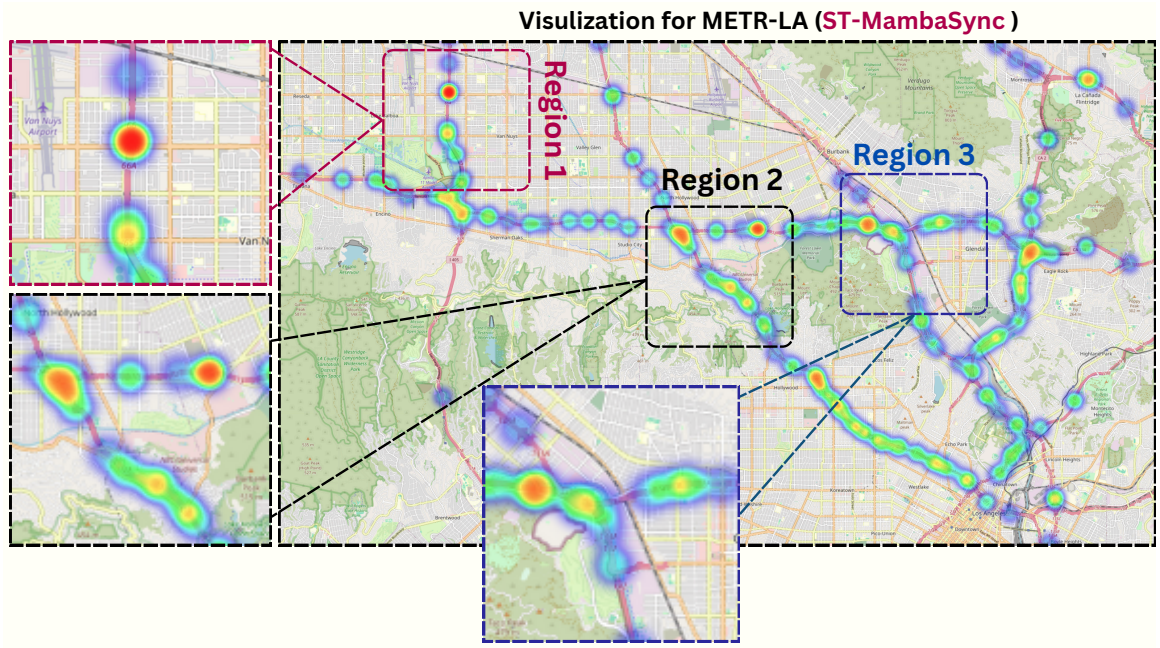


FIGURE 4.10. Heatmap visualization for the METR-LA dataset with the ST-Mamba block and Transformer, focusing on key regions. The heatmap visualizes traffic intensity, with warmer colors (red, yellow) indicating higher traffic volumes.

and Figure 4.13, we present the selected baseline models, which are transformer-based and GNN-based, respectively, as they outperform the other baseline models.

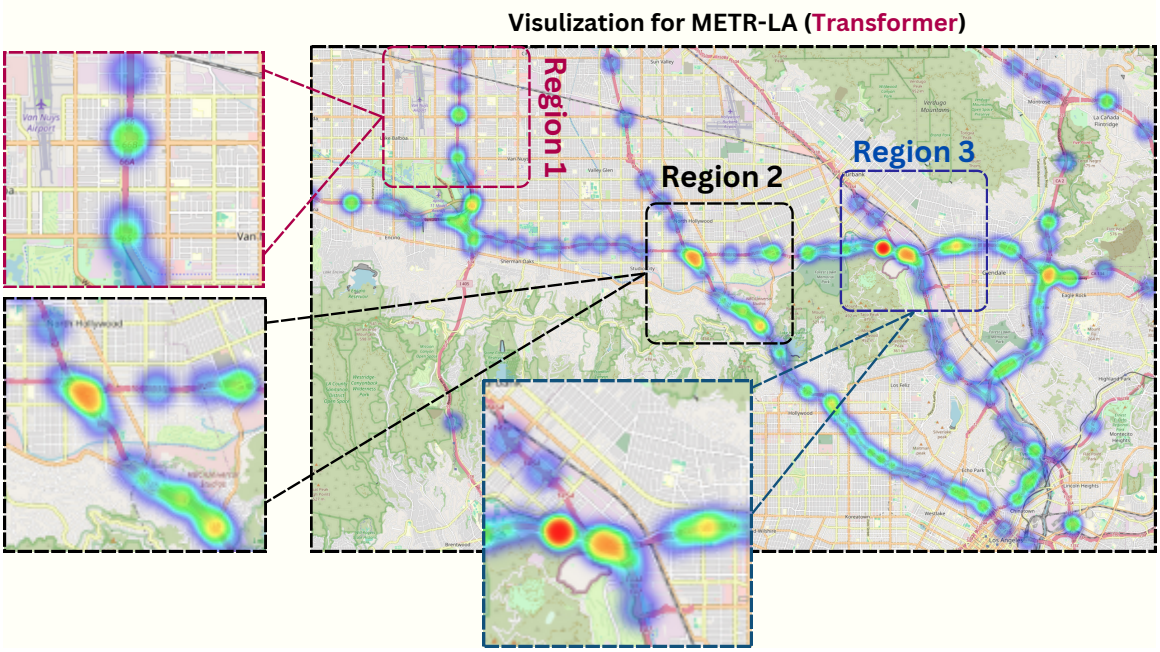


FIGURE 4.11. Heatmap visualization for the METR-LA dataset with the Transformer block only, focusing on key regions. The heatmap visualizes traffic intensity, with warmer colors (red, yellow) indicating higher traffic volumes.

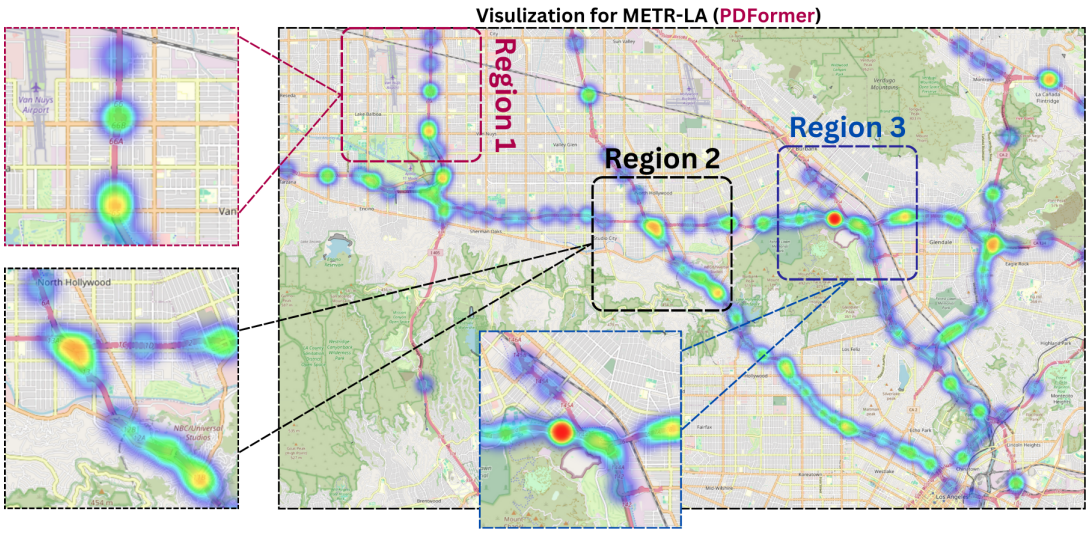


FIGURE 4.12. Heatmap visualization for PDFormer on the METR-LA dataset. The heatmap visualizes traffic intensity, with warmer colors (red, yellow) indicating higher traffic volumes.

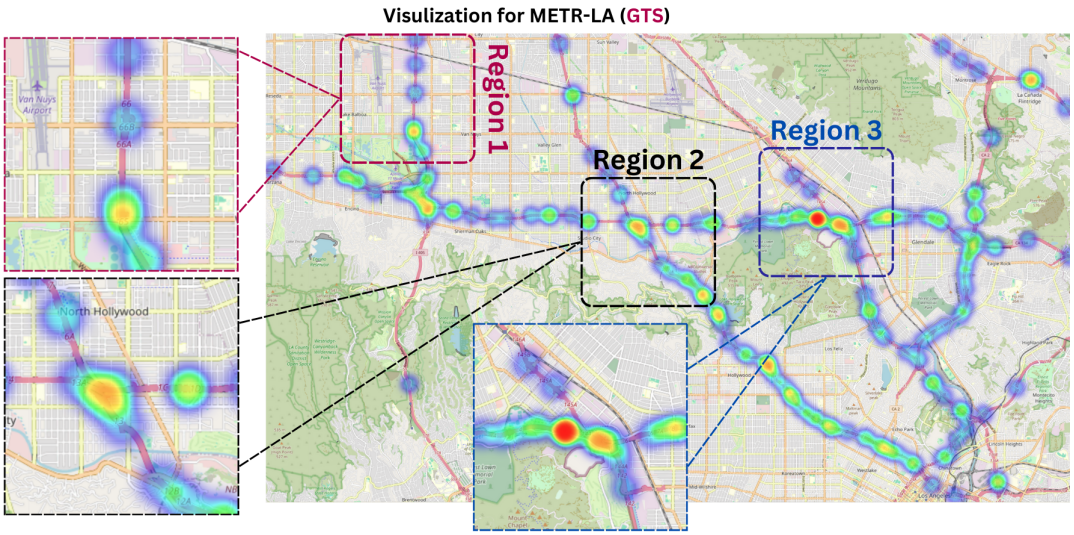


FIGURE 4.13. Heatmap visualization for the GTS model on the METR-LA dataset. The heatmap visualizes traffic intensity, with warmer colors (red, yellow) indicating higher traffic volumes.

For a clearer view of the comparison, Figure 4.14 zoomed comparison figures provide a closer look at the performance of the four models in specific regions with peak hours and Figure 4.15 is the model performance with non-peak hours:

- **Region 1**, the Ground Truth heatmap indicates a high concentration of traffic in the northern area. The ST-Mamba + Transformer model accurately captures these high-density traffic regions, closely mirroring the Ground Truth data. In contrast, the

STAEFormer (Transformer without ST-Mamba), while identifying the high-traffic areas, shows less accuracy in intensity and focus compared to both the Ground Truth and the combined model. This suggests that including the ST-Mamba block significantly enhances the model's ability to replicate high-density traffic patterns. In Region 1, PDFormer captures the general location of high-traffic areas but lacks the intensity observed in the Ground Truth. The heatmap shows a broader distribution of traffic, indicating that PDFormer tends to over-generalize traffic density. This reduces its ability to accurately pinpoint specific high-traffic zones. GTS also identifies the primary traffic hotspots in Region 1 but does so with a smoother and less detailed heatmap.

- **Region 2**, characterized by significant traffic congestion, presents distinct high-density spots in the Ground Truth heatmap. The ST-Mamba + Transformer model effectively replicates this congestion pattern, maintaining a precise local focus on high-density areas akin to the Ground Truth. Conversely, the Transformer without ST-Mamba demonstrates a more diffused focus, indicating a weaker performance in capturing detailed local traffic patterns. This further supports the notion that the ST-Mamba block improves the model's local focus capabilities. PDFormer struggles to accurately replicate the Ground Truth traffic patterns. It shows only a faint presence of high-traffic areas, lacking the specificity needed to identify concentrated zones accurately. This suggests that PDFormer has difficulty capturing traffic variations in this region. GTS performs reasonably well in Region 2 by identifying the general traffic areas. However, like PDFormer, it lacks focus on high-density zones and displays a smoother pattern with less intensity. The model fails to fully capture the traffic concentration levels indicated in the Ground Truth, leading to a more generalized heatmap (less accuracy) for this region.
- **Region 3**, which shows moderate traffic activity with several hotspots in the Ground Truth heatmap, the ST-Mamba + Transformer model successfully mimics the observed traffic levels and specific hotspots. The Transformer without ST-Mamba, while presenting the general traffic pattern, lacks a detailed local focus, leading to less accurate hotspot representations. This underscores the advantage of integrating the ST-Mamba block to enhance the model's local precision. PDFormer again over-generalizes, showing a

broad but less intense traffic distribution. It fails to match the Ground Truth's intensity and misses specific high-density areas. This limitation is evident as the model does not capture the unique traffic concentration patterns distinctive to Region 3. GTS identifies the main traffic zones in Region 3 but with a very smooth, diluted pattern. The high-density regions are not as prominent, resulting in a lack of detail in the heatmap. This suggests that GTS has difficulty capturing the specific traffic peaks in Region 3, leading to a more generalized representation compared to the Ground Truth.

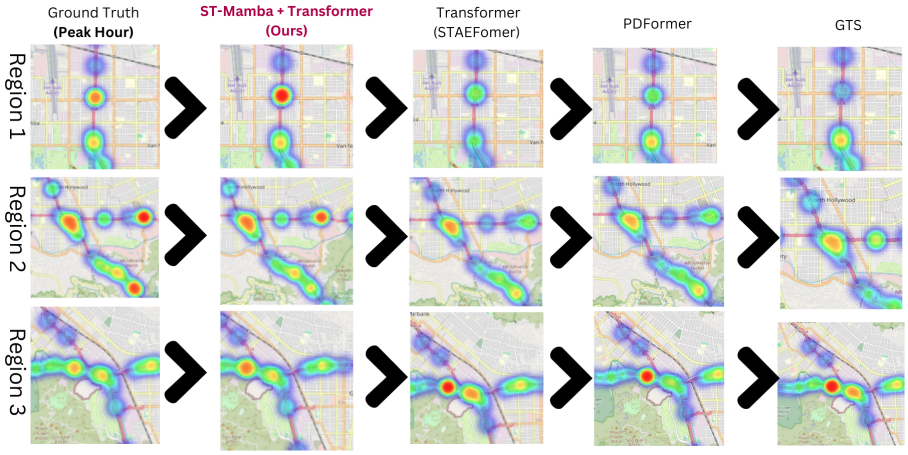


FIGURE 4.14. This figure illustrates the zoomed peak hour performance comparison of four traffic prediction models across three distinct regions in Los Angeles using heatmaps: Ground Truth, ST-Mamba + Transformer, Transformer without ST-Mamba(STAEformer), PDFormer and the GNN based model GTS. The heatmaps utilize a color scale where warmer colors (red, yellow) indicate higher traffic volumes, and cooler colors (blue, green) indicate lower traffic volumes.

The comparative analysis clearly demonstrates that the integration of the ST-Mamba block with the Transformer model significantly enhances the model's ability to focus on local traffic patterns. The ST-Mamba block complements the Transformer's global focus by providing detailed local insights, resulting in improved accuracy and concentration of high-traffic regions. This synergy between local and global focus mechanisms allows for more precise and reliable traffic predictions for long-range data, as evidenced by the more accurate and focused heatmaps of the ST-Mamba + Transformer model across all regions.

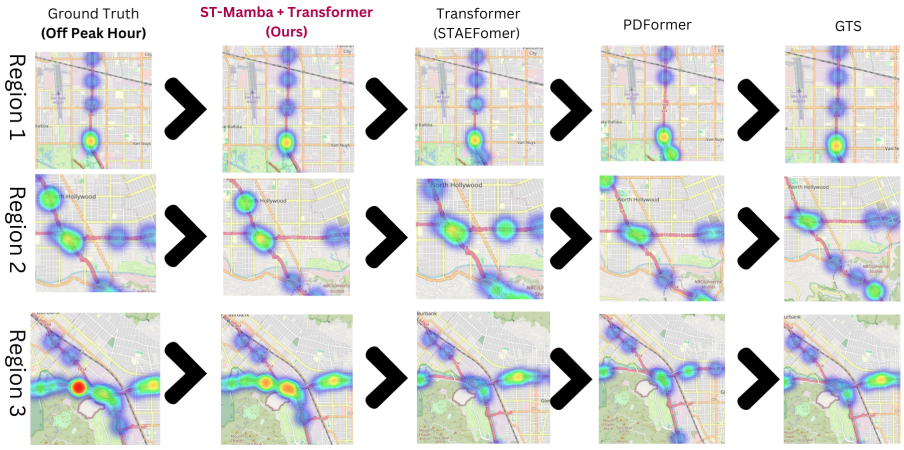


FIGURE 4.15. This figure illustrates the zoomed off peak hours performance comparison of four traffic prediction models across three distinct regions in Los Angeles using heatmaps: Ground Truth, ST-Mamba + Transformer, Transformer without ST-Mamba (STAEformer), PDFormer and the GNN based model GTS. The heatmaps utilize a color scale where warmer colors (red, yellow) indicate higher traffic volumes, and cooler colors (blue, green) indicate lower traffic volumes.

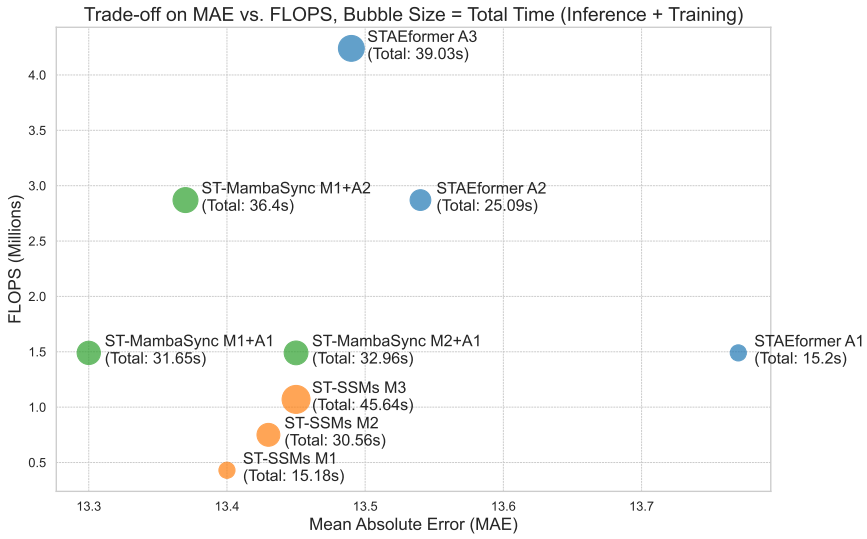


FIGURE 4.16. Trade-offs in Model Performance and Computational Efficiency.

4.3.4.3 Trade Off Analysis on Accuracy and Computation

Figure 4.16 presents the trade-offs in model performance and computational efficiency. This bubble chart illustrates the relationship between Mean Absolute Error (MAE) and computational cost (FLOPS) for various predictive models on the PEMS08 dataset. Each bubble's size represents the total time required for inference and training, highlighting the efficiency trade-offs. We denote

"M" as the number of Mamba layers in the model and "A" as the number of attention layers. The trade-off analysis aids in investigating the combined efficiency of our proposed ST-MambaSync. ST-MambaSync models (green bubbles) generally exhibit lower MAE compared to STAEformer (blue bubbles) and ST-Mamba (orange bubbles) variants, indicating better prediction accuracy. For instance, ST-MambaSync M1+A1 achieves an MAE close to 13.3, which is the lowest compare with other models. For computation efficiency, our proposed ST-MambaSync variants demonstrate a range of FLOPS, indicating varying computational demands. ST-MambaSync M1+A1 and M1+A2 have higher FLOPS compared to M1 alone, due to additional attention layers. Furthermore, the bubble sizes show that ST-MambaSync models have a balanced trade-off between computational efficiency and total time. For example, ST-MambaSync M1+A1 has a total time of 31.65 seconds, which is competitive with other models.

Overall, the results reveal important insights into the balance between accuracy and computational effectiveness, focusing on two aspects: the increase in attention layers and the increase in Mamba layers.

- *Increase in Attention Layers:* In attention-based models, increasing the number of attention layers leads to higher FLOPS and computational times, although with improved accuracy. For our integrated model, ST-MambaSync, maintaining a single Mamba layer while increasing the number of attention layers results in a doubling of FLOPS and a significant increase in computational time without a corresponding improvement in accuracy.
- *Increase in Mamba Layers:* For the Mamba-based model, ST-Mamba, an increase in Mamba layers results in higher FLOPS and extended computational times, but surprisingly, accuracy decreases. This observation suggests that a single Mamba layer is optimal for balancing accuracy and computational efficiency in the integrated ST-MambaSync model. To enhance prediction accuracy without significantly increasing the computational cost, incorporating a single attention layer is the most effective strategy. The optimal configuration for achieving the highest accuracy with the least

computational trade-off is the ST-MambaSync model with one Mamba layer and one attention layer, which meets these conditions satisfactorily.

4.3.4.4 Temporal Analysis of Traffic Flow Predictions

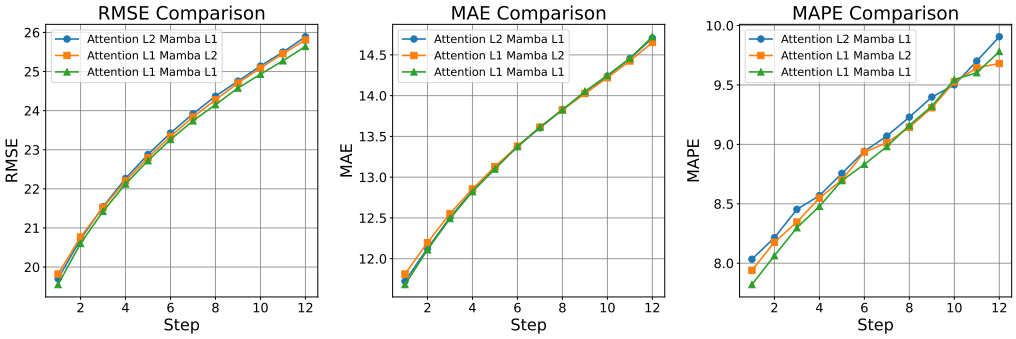


FIGURE 4.17. This figure presents a side-by-side comparison of three key performance metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) across varying layers of attention and Mamba for ST-MambaSync. Each subplot illustrates the variation of a specific metric across 12-time steps, highlighting the models' performance stability and accuracy in forecasting. Distinct color-coded lines represent different model configurations, ensuring clear differentiation and readability.

Figure 4.17 presents a comparative analysis of three configurations—"Attention L2 Mamba L1", "Attention L1 Mamba L2", and "Attention L1 Mamba L1" —using the metrics RMSE, MAE, and MAPE across different steps. From this analysis, it is evident that the "Attention L1 Mamba L1" configuration performs best across all metrics. Specifically, it consistently achieves the lowest RMSE, which indicates a superior ability to minimize larger errors, crucial for ensuring accurate predictions in scenarios with outliers. Moreover, the lowest MAE values demonstrate that "Attention L1 Mamba L1" minimizes average errors more effectively, resulting in more reliable predictions overall. The lower MAPE for this configuration further highlights its capability to maintain a low average relative error, making it particularly well-suited for real-world applications where minimizing relative deviations is essential.

On the other hand, the "Attention L2 Mamba L1" configuration also performs well but generally has higher RMSE, MAE, and MAPE values compared to "Attention L1 Mamba L1". This suggests that, while "Attention L2 Mamba L1" can effectively handle traffic prediction tasks, it is less capable of minimizing both large and average prediction errors, resulting in reduced overall

accuracy and reliability. Similarly, the "Attention L1 Mamba L2" configuration falls between the other two in terms of performance. It shows slightly improved accuracy compared to "Attention L2 Mamba L1" in certain cases but does not match the performance of "Attention L1 Mamba L1". The higher RMSE, MAE, and MAPE values indicate a greater likelihood of both large errors and average prediction inaccuracies.

In conclusion, "Attention L1 Mamba L1" is the best-performing configuration, offering a balanced and efficient solution by minimizing large, average, and relative errors. This makes it highly suitable for traffic prediction tasks where both accuracy and computational efficiency are crucial. In contrast, "Attention L2 Mamba L1" and "Attention L1 Mamba L2" demonstrate reasonable performance but fall short in maintaining the same level of precision, indicating a trade-off in their ability to capture and predict traffic flow accurately.

4.4 Discussion and Implication

The proposed ST-MambaSync model demonstrates notable strengths, effectively combining the Mamba mechanism with Transformer technology to achieve a unique balance between computational efficiency and prediction accuracy. By leveraging the Mamba mechanism's ability to enhance local feature extraction through its ResNet-like structure and integrating it seamlessly with the Transformer's global attention capabilities, ST-MambaSync excels at capturing both short- and long-range dependencies in traffic data. This innovative synergy not only improves predictive performance but also significantly reduces computational demands, as evidenced by substantial decreases in FLOPS, inference time, and training time compared to traditional models. The model's adaptability to complex and dynamic traffic scenarios makes it particularly well-suited for real-time applications, offering a practical and efficient solution for urban traffic management and planning.

Figure 4.18 illustrates the comparative analysis of Prediction Results Using the PEMS08 Dataset for Sensors 36 and 127; this case study evaluates the predictive performance of the STAEformer and ST-MambaSync models across 1-hour, 5-hour, and 24-hour intervals. Each model's

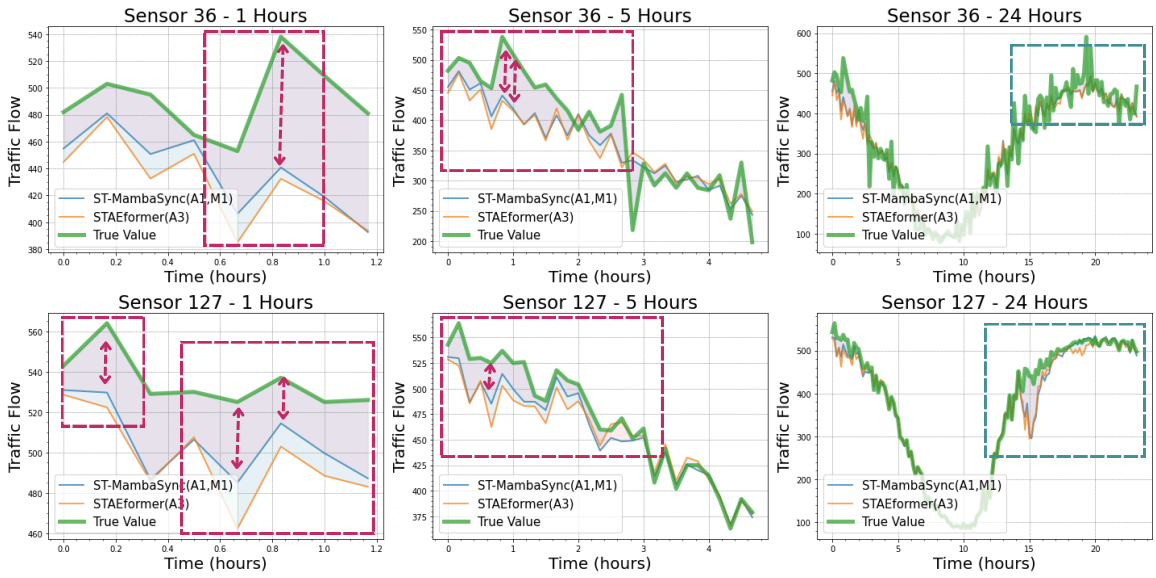


FIGURE 4.18. Comparative Analysis of Prediction Results Using PEMS08 Dataset for Sensors 36 and 127.

architecture is defined by the number of attention layers (denoted as "A#") and the number of Mamba layers ("M#"). For instance, ST-MambaSync is configured with one attention layer and one Mamba layer, while STAEformer utilizes three attention layers.

Predictions for Sensor 36.

- **1 Hour:** Both models diverged from the true values, though ST-MambaSync demonstrated closer approximations at certain intervals.
- **5 Hours:** ST-MambaSync provided more consistent and accurate predictions, closely tracking the actual data, unlike the fluctuating results from STAEformer.
- **24 Hours:** ST-MambaSync showed superior long-term predictive consistency, adhering closely to the actual traffic flow patterns.

Predictions for Sensor 127.

- **1 Hour:** Both models performed similarly to those for Sensor 36, with ST-MambaSync slightly more accurate at certain points.

- **5 Hours:** Variability was noted in both models, with neither showing a consistent advantage during the initial hours.
- **24 Hours:** ST-MambaSync maintained closer alignment with the true values, indicating its better capability at handling longer-term dynamics.

The analysis indicates that ST-MambaSync provides more accurate and consistent forecasts across all examined intervals for both sensors. It excels particularly in 24-hour forecasts, suggesting it is more adept at capturing and adapting to longer-term traffic flow dynamics. ST-MambaSync consistently outperforms the state-of-the-art model (SOTA) STAEformer, especially in longer forecast intervals. This emphasizes the importance of selecting appropriate models based on the predictive timeframe and desired accuracy level for traffic management applications.

Limitation. Despite the promising results of ST-MambaSync, there are still some limitations that provide directions for future research. First, while ST-MambaSync has been shown to effectively reduce the computational cost, the scalability of the model to even larger datasets or different urban contexts needs further investigation. Second, the trade-offs between accuracy and computational efficiency were evaluated under fixed configurations; however, exploring adaptive methods for dynamically adjusting model complexity based on traffic conditions may further improve its real-time applicability. Additionally, our current approach is limited to the integration of the Mamba and Transformer architectures. Future studies could investigate the potential of combining other advanced spatial-temporal models to enhance both the robustness and interpretability of traffic forecasting systems.

Future Work. While this study demonstrates the effectiveness of ST-MambaSync in balancing computational efficiency and predictive accuracy for spatial-temporal tasks, several directions for future research can further enhance the potential of this framework. Currently, ST-MambaSync is focused on traffic flow prediction. Expanding this model into a multi-task learning framework could enable the joint prediction of multiple traffic metrics, such as speed, demand, and congestion levels, within a unified architecture. Such an expansion would further enhance its applicability in intelligent transportation systems (ITSs), providing a more comprehensive view

of urban mobility. Additionally, while the integration of attention and Mamba mechanisms has demonstrated complementary strengths, there is still potential to further optimize the attention mechanism. Techniques like sparse attention or low-rank approximations could be employed to reduce the computational costs associated with Transformer components, particularly when working with very large datasets or extended forecasting horizons.

4.5 Conclusion

This study introduces ST-MambaSync, an innovative framework that fuses an attention layer with a streamlined selective state-space (Mamba) layer, striking an optimal balance between rapid computation and high predictive accuracy. We are the first to explore the distinctive property of Mamba, demonstrating how it functions as a specialized form of Transformer enhanced with a ResNet architecture to better capture local features. Our work provides both theoretical and experimental validation of the synergy between Transformer and ST-Mamba components. Experimental results show that ST-MambaSync achieves state-of-the-art performance in spatial-temporal prediction tasks while significantly reducing computational costs. By illustrating that the Mamba layer operates similarly to an attention mechanism within a residual network, we highlight its role in effectively focusing on local features, complementing the global feature-capturing capacity of attention mechanisms. This integration of Mamba with attention not only enhances model accuracy but also improves efficiency, specifically improving MAE by 0.70%, RMSE by 0.62%, and MAPE by 0.31%, 64.86% reduction in FLOPS, 12.54% reduction in Inference time, and 19.44% reduction in Training time compared to the formal state-of-the-art (SOTA) model, making it well-suited for diverse real-world applications such as urban planning and traffic management. This balanced approach between accuracy and computational efficiency offers significant potential for practical implementations in complex environments. Further study can use this kind of model in weather prediction, traffic demand prediction, or other spatial-temporal related data.

Chapter 5

Dynamic Attention–Based Mamba for Multi-Mode Passenger Demand Prediction

---

Focusing on **RQ3**: How can these principles be *generalised* to forecast heterogeneous passenger demand across multiple transit modes without extensive manual graph construction?, we extend the synergistic architecture to multimodal, demographic-aware demand forecasting via **STDAtt-Mamba**, eliminating manual graph construction.

5.1 Introduction

Urban mobility systems serve diverse socio-demographic passenger groups with varying travel needs, such as adults, seniors, youth, pensioners, and students. Predicting passenger demand in multimodal public transit (PT) systems is crucial for optimizing service allocation, improving passenger satisfaction, and supporting sustainable urban transport planning [Xi et al., 2024a]. AI and machine learning (ML) techniques are increasingly used for demand prediction, focusing on models that learn and make decisions based on historical data [Goodfellow, 2016]. However, predicting demand in multimodal PT systems remains a significant challenge due to the inherent spatial and temporal complexities and the heterogeneous travel behaviors exhibited by different passenger groups across various travel modes. Existing demand prediction models often fail to capture these diverse travel patterns, e.g., daily commuters typically follow consistent routes during peak hours, while seniors display more irregular travel behaviors. This disparity highlights the need for demand prediction models that account for the heterogeneous passenger groups.

The development of deep learning (DL) models has evolved from Recurrent Neural Networks (RNNs) [Rumelhart et al., 1986] and Convolutional Neural Networks (CNNs) [Lecun et al.,

1998] to Transformers [Vaswani, 2017]. Transformers excel at capturing long-range temporal dependencies through self-attention mechanisms but face efficiency challenges with their computational complexity for long sequences. As for spatial modeling, Graph Neural Networks (GNNs) [Scarselli et al., 2009] can effectively capture interrelations among nodes but require resource-intensive graph construction processes, limiting their scalability. Meanwhile, the selective state-space model, so-called Mamba [Gu and Dao, 2023], offers an efficient solution for handling long-range temporal dependencies with linear computational complexity, making it well-suited for processing long sequences. However, Mamba lacks the ability to integrate spatial-temporal features, which are crucial for accurate travel demand forecasting. To address these limitations, we propose a novel spatial-temporal dynamic state-space model, STDAtt-Mamba, designed for multi-type passenger demand prediction in multimodal transit systems, which combines the temporal efficiency of Mamba and the spatial-temporal integration capabilities of Transformers.

Unlike most of the existing literature in passenger demand prediction, which focuses on homogeneous passengers and separately processes spatial and temporal dynamics, the proposed STDAtt-Mambaincorporates spatial-temporal dependencies of heterogeneous passenger groups and provides an adaptive and scalable solution for heterogeneous passenger demand prediction in multimodal PT systems, ensuring improved accessibility to public transport across diverse socio-demographic groups. This study employs a large-scale dataset comprising multimodal travel records (bus, rail and ferry) of over 1.58 million passengers of nine passenger groups (i.e., adults, seniors, pensioners, tertiary students, children, job seekers, school passengers, youth, and Gold Repat passengers), from 01/2021 to 01/2023 in the multimodal PT system of Queensland, Australia.

We next compare the existing deep learning models with the proposed STDAtt-Mamba(Section 5.1.1); and compare the key literature on deep learning models for travel demand prediction in PT systems (Section 5.1.2), before summarizing our contributions (Section 5.1.3).

5.1.1 Comparison of Existing Models and STDAtt-Mambafor Travel Demand Prediction

Existing literature on travel demand prediction models in PT systems can be categorized into three main groups: traditional deep learning (DL) methods, advanced transformer-based models, and state-space models. Each category has distinct strengths and weaknesses.

Traditional deep learning models, such as Recurrent Neural Network (RNN) [Rumelhart et al., 1986], Convolutional Neural Network (CNN) [Lecun et al., 1998], Graph Neural Networks (GNN) [Scarselli et al., 2009], and emerging advanced DL models such as Transformer [Vaswani, 2017] and Mamba [Gu and Dao, 2023], have significantly contributed to travel demand prediction. However, each model faces limitations that hinder their overall effectiveness. RNNs, particularly Long Short-Term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997], can capture both long- and short-term dependencies but suffer from the vanishing gradient problem [Hochreiter, 1998] and are computationally intensive [Oliveira et al., 2021]. Additionally, their sequential nature limits parallelization, slowing training and inference times, which is problematic for real-time PT systems. CNNs [Lecun et al., 1998] are effective for spatial pattern recognition but struggle with long-range temporal dependencies. Hybrid models, such as CNNs combined with gated recurrent units (GRUs), have shown potential but still require substantial computational resources and suffer from high inference times [Wu et al., 2018, Zhang et al., 2019]. GNNs [Scarselli et al., 2009] excel at modeling spatial dependencies in PT systems but face challenges like high computational demands and the over-smoothing problem, which limits their ability to capture long-range dependencies accurately [BAI et al., 2020].

Transformers have emerged as a powerful tool for capturing complex spatial-temporal relationships due to their self-attention mechanisms [Vaswani, 2017]. However, despite their ability to handle long-range dependencies, transformers exhibit significant computational complexity, particularly challenging with large-scale, multimodal PT datasets. Moreover, transformers typically require separate attention mechanisms for spatial and temporal information, limiting their efficiency and scalability for real-time applications [Xu et al., 2020a, Xi et al., 2024b,

Shao et al., 2024c]. Thus, transformers' practical use is constrained by their high computational overhead and inefficient handling of integrated spatial-temporal features.

State Space Models (SSMs) have emerged as promising alternatives in temporal modeling, which are particularly notable since the introduction of the Structured State Space Sequence Model (S4), and represent a significant advancement in handling long sequence data [Gu et al., 2021]. Building on the foundation of Structured State Space Sequence (S4) models, Mamba [Gu and Dao, 2023] introduced the Selective State Space model architecture, which improved upon the limitations of S4. Recent studies have extended Mamba to spatial-temporal applications [Shao et al., 2024a,h], but it faces limitations in capturing heterogeneous spatial-temporal dependencies that induce a lack of generalization for multi-tasks. Further, Empirical results of these works indicate that naive layer stacking methods can degrade model performance, highlighting the need for more sophisticated architectural design.

TABLE 5.1. Comparison of performance of existing DL and STDAtt-Mambamodel for travel demand prediction

Performance	**RNN** (1986)	**LSTM** (1997)	**CNN** (1998)	**GNN** (2009)	**Transforme** (2017)	**Mamba** (2023)	**STDAtt-Mamba** (ours)
Temporal dependency	poor	improved	limited	poor	excellent	efficient	excellent
Spatial-temporal processing	temporal-focused	temporal -focused	spatial only	spatial-focused	separate process	temporal-focused	integrated well
Computational efficiency	moderate	low	high	high	$O(n^2)$	$O(n)$	sparse $O(n^2)$
Training Phase	slow	slow	fast	slow	fast	fast	fast
Testing Phase	fast	slow	fast	slow	fast	fast	fast
Practical limitations	limited parallel processing	forgetting	lacks temporal	need graph construction	resource-intensive	lacks spatial	–

A comparative analysis of various deep learning models for travel demand prediction reveals several crucial limitations, summarized in Table 5.1. Traditional deep learning models such as RNNs and LSTMs exhibit significant challenges: RNNs struggle with temporal dependencies due to vanishing gradients, and while LSTMs offer improved temporal modeling, they remain computationally inefficient and slow during both training and inference phases. CNNs, although efficient in processing spatial features, fail to capture temporal dynamics, limiting their applicability to spatial-temporal prediction tasks. GNNs effectively model spatial relationships but

are impeded by resource-intensive graph construction, leading to scalability issues in larger datasets. Transformers have emerged as powerful models for capturing long-range temporal dependencies using self-attention mechanisms. However, they exhibit significant computational complexity ($O(n^2)$) and often require separate attention mechanisms to handle spatial and temporal information, thus reducing their efficiency and real-time applicability. In contrast, the recently introduced Mamba architecture addresses temporal dependency modeling with linear computational complexity ($O(n)$), substantially improving efficiency in processing long sequences. Despite this advantage, Mamba lacks mechanisms to effectively integrate spatial and temporal dependencies, a crucial requirement for accurate prediction in multimodal PT systems with heterogeneous passenger behaviors.

5.1.2 Deep Learning Models for Demand Prediction in PT Systems

Extensive efforts have been made on the demand prediction of PT systems to improve service quality and efficiency. Luo et al. [2020] introduced a multitask deep learning model for fine-grained bus passenger flow prediction, utilizing an ARM network to handle complex spatial-temporal dependencies. Zhang et al. [2021b] developed the channel-wise attentive split-convolutional neural network (CAS-CNN) for short-term Origin-Destination (OD) prediction, achieving superior results on Beijing subway data. Lv et al. [2023] introduced a tree-structured spatial-temporal neural network (TS-STNN) combining hierarchical spatial matrices and GRUs for temporal analysis. Zhang et al. [2023] introduced a spatiotemporal convolutional neural network (STCNN) for short-term OD passenger demand prediction, incorporating lagged spatiotemporal relationship learning, OD importance calculation, and a time-varying weighted loss function to improve prediction accuracy. Tang et al. [2023] proposed Deep-GAN, leveraging deep generative adversarial networks to address data imbalance in smart-card records, outperforming traditional resampling methods in predicting bus boarding demand and analyzing ridership patterns.

RNN and LSTM are widely used in travel demand prediction for public transport systems due to their capability to handle sequential data. For instance, Li et al. [2021a] introduced the knowledge adaptation with attentive multi-task memory network (KA2M2), leveraging memory-augmented

recurrent networks and attention-based knowledge adaptation to improve multimodal demand forecasting by transferring knowledge between transport modes. Yang et al. [2021b] proposed the spatiotemporal long short-term memory (Sp-LSTM) model, integrating spatiotemporal passenger flow features to predict short-term demand at urban rail stations. Huang et al. [2022] introduced the dynamical spatial-temporal Graph Neural Network (DSTGNN), combining the inhomogeneous Poisson process, Diffusion Convolutional Neural Network (DCNN), and Transformer with dynamic spatial dependence graphs to improve prediction accuracy. Liyanage et al. [2022] utilized LSTM and bidirectional LSTM (BiLSTM) models to predict bus passenger demand using the smart-card ticketing data in Melbourne, showing the superior performance of BiLSTM across 18 bus routes. Wu et al. [2023] proposed the Graph Convolutional Neural Network-Long Short-Term Memory (GCNN-LSTM) hybrid model for urban rail passenger flow prediction, which outperforms LSTM and CNN models on Nanning Metro data. Li et al. [2023] developed Interaction Graph Network (IG-Net), which uses inter-station interaction graphs and multi-task learning to improve station-level prediction, validated on Suzhou Metro data.

However, RNNs struggle to capture spatial correlations, such as the propagation of passenger demand between stations. GNNs have been used to model spatial dependencies. For instance, Li et al. [2020] proposed the probabilistic graph convolution model (PGCM) for OD demand prediction, leveraging spatial-temporal correlations with a Graph Convolution Network (GCN) and Bayesian uncertainty estimation, validated on Greater Sydney transit data. Xiong et al. [2020] developed the Fusion Line Graph Convolutional Networks (FL-GCNs) with Kalman filters to capture spatial-temporal patterns, achieving strong performance on New Jersey Turnpike data. He et al. [2022] introduced the Multi-Graph Convolutional-Recurrent Neural Network (MGC-RNN), integrating spatial and temporal dependencies using multiple inter-station correlation graphs. Liang et al. [2022] developed the Spatiotemporal Multi-Relational Graph Neural Network (ST-MRGNN) for multimodal passenger demand prediction by modeling intra- and inter-modal spatial relationships. Jiang et al. [2022] proposed the temporally shifted spatiotemporal network (TS-STN), integrating GCN, attention-based LSTMs, and real-time OD flow reconstruction for short-term flow prediction under partial observability. Wu et al. [2023] proposed the multi-feature fusion Graph Convolutional Network (MFGCN), combining spatial and temporal attention

mechanisms to predict short-term passenger flow using Nanning Metro data. Wang et al. [2024a] introduced the Probabilistic Graph Neural Network (Prob-GNN) to quantify spatiotemporal uncertainty in travel demand, demonstrating robustness on CTA rail and ridesharing data.

Transformers [Vaswani, 2017], with attention mechanisms, enable simultaneous processing of input sequences, capturing both short- and long-term dependencies. Transformers have been applied to passenger flow prediction. For example, Liu et al. [2022] proposed the Heterogeneous Information Aggregation Machine (HIAM), integrating incomplete OD matrices, unfinished order vectors, and DO matrices for metro OD and DO demand forecasting. Yang et al. [2024a] introduced the MultiModeformer (M2-former), a multitask encoder-decoder model for network-wide short-term inflow prediction in multi-mode systems, capturing dynamic spatiotemporal

TABLE 5.2. Key literature on deep learning models for travel demand prediction in public transit systems

Reference	**Models**	**Data source**	**Multi-mode**	**Multi-task**	**Multi-passenger**	**Spatial (S)**	**Temporal (T)**	**ST dynamic fusion**
[Zhang et al., 2021b]	CAS-CNN	Beijing subway data	×	×	×	✓	✓	×
[Li et al., 2021a]	KA2M2	Smart card PT data in Greater Sydney	✓	✓	×	✓	✓	×
Liang et al. [2022]	ST-MRGNN	NYC Subway and ride-hailing data	✓	✓	×	✓	✓	×
[Jiang et al., 2022]	TS-STN	Smart card data in Hong Kong MTR	×	×	×	✓	✓	×
[Liyanage et al., 2022]	IG-Net	Suzhou metro AFC data	×	✓	×	✓	✓	×
[Liu et al., 2022]	Dual Information Transformer	Metro AFC data in Shanghai and Hangzhou	×	✓	×	✓	✓	×
[Wu et al., 2023]	MFGCN	Nanning Metro Line AFC data	×	×	×	✓	✓	×
[Zhang et al., 2023]	STCNN	Chengdu Metro AFC data	×	×	×	✓	✓	×
[Bapaume et al., 2023]	U-Transformer, CV Transformer	Paris metro line 9	×	×	×	✓	✓	×
[Wang et al., 2024a]	Prob-GNN	Chicago transit authority bus and rail data	✓	✓	×	✓	✓	×
[Hu et al., 2024]	STGT	Beijing subway passenger travel records	×	×	×	✓	✓	×
[Yang et al., 2024b]	MultiModeformer	Multi-travel modes in Beijing	✓	✓	×	✓	✓	×
[Qiu et al., 2025]	STMTL	AFC data of urban rail transit in Nanning	×	✓	×	✓	✓	×
Our study	STDAtt-Mamba	Smart card PT data in Queensland, Australia	✓	✓	✓	✓	✓	✓

correlations. Hu et al. [2024] developed the Spatio-Temporal Graph Transformer (STGT) with a Multi-gate Mixture-of-Experts (MMoE) framework, incorporating metro-specific characteristics for short-term inflow and outflow prediction, achieving superior results on Beijing subway data. Shao et al. [2024f] proposed a spatial-temporal large language model combined with diffusion to predict the multi-mode system, but with a limit on capturing the local dependency. Qiu et al. [2025] proposed a Spatial–Temporal Multi-Task Learning (STMTL) model for short-term passenger inflow and outflow prediction on holidays in urban rail transit systems, integrating multi-graph attention, time encoding, and cross-attention.

Further, Mamba [Gu and Dao, 2023] is an emerging architecture designed for the rapid processing of long sequences, surpassing Transformer models in speed and efficiency. While Mamba-based models have shown promise in traffic prediction [Shao et al., 2024h,a, Lin et al., 2024], however, previous experimental results have revealed that increasing the number of layers fails to yield further improvements, leading to diminishing returns or even performance degradation. Therefore, it is crucial to rethink and modify the Mamba architecture to break through the boundaries of Mamba. This study proposes a spatial-temporal (ST) dynamic fusion layer within Mamba that dynamically combines these features in a unified representation, allowing the model to better capture the heterogeneous travel patterns across multiple passenger groups.

Table 5.2 compares existing deep learning models for passenger demand prediction in multimodal PT systems, from the perspectives of multiple travel modes, multi-tasking, passenger-type differentiation, spatial, temporal, and spatial-temporal (ST) dynamic fusion.

5.1.3 Research Gaps and Contributions

As summarized in Table 5.1, although existing DL models have shown promise in spatial-temporal modeling for passenger demand prediction in PT systems, they encounter significant limitations when applied to heterogeneous passenger demand prediction. GNNs rely on message-passing mechanisms that mainly focus on local neighborhood information, which restricts their ability to capture long-range dependencies essential for modeling diverse mobility patterns across various passenger types. Although Transformers incorporate attention mechanisms, they remain

constrained by limited receptive fields and face computational challenges due to the need for constructing graphs in large-scale transit networks. In contrast, Mamba offers several key advantages in linear computational complexity, robust long-range dependency modeling, and the ability to focus selectively on relevant parts of the input sequence. These properties are crucial for capturing the diverse temporal patterns exhibited by different passenger groups. By extending Mamba with the proposed STDF layer, we address its limitation of handling only temporal sequences, enabling seamless spatial-temporal modeling without the computational overhead associated with graph construction.

As shown in Table 5.2, existing models predominantly focus on predicting aggregated passenger demand without differentiating between various socio-demographic passenger groups (e.g., adults, students, seniors), which limits the capability of models in addressing diverse behaviors and needs of multiple socio-demographic groups. Most existing studies typically process spatial and temporal data independently or through static integration methods and lack dynamic mechanisms to jointly adapt spatial and temporal features, resulting in suboptimal predictions for multimodal PT systems.

To bridge these crucial research gaps, we propose a STDAtt-Mambamodel, which integrates the efficient temporal processing capabilities of Mamba with the powerful spatial-temporal attention mechanisms of Transformers through a novel dynamic fusion layer. To the best of our knowledge, this is the first spatial-temporal dynamic attention-based state-space model for multi-type passenger demand prediction in multimodal PT systems. The STDAtt-Mambamodel allows simultaneous, integrated processing of heterogeneous spatial-temporal features, significantly improving scalability and adaptability in multimodal and multi-type passenger demand prediction tasks. By addressing both short-range local dependencies through Mamba and long-range global dependencies via sparse attention, STDAtt-Mambaprovides a scalable, adaptable, computationally efficient, and highly accurate solution tailored to meet the diverse and dynamic demands of multimodal PT systems.

The major contributions of the chapter are summarized as follows,

- We propose a novel STDAtt-Mambamodel for multi-type passenger demand prediction in multimodal PT systems under a multi-task learning framework, where each task corresponds to a distinct travel mode. The STDAtt-Mambamodel comprises three key components: an adaptive embedding layer that integrates station-level, passenger-type-specific, and temporal embeddings (e.g., time-of-day, day-of-week) into a unified representation for efficient processing; a spatial-temporal dynamic attention (STDAtt) module that employs sparse attention mechanisms to selectively capture crucial global dependencies; and a spatial-temporal dynamic Mamba (STDMamba) module that extends state-space modeling to dynamically fuse spatial and temporal dependencies using a ResNet-inspired architecture.
- We propose a spatial-temporal dynamic fusion layer (STDF) within STDMamba module, which redefines how spatial and temporal dependencies are dynamically integrated within the Mamba architecture. Unlike previous models, where spatial and temporal features are processed separately, the STDF layer integrates spatial and temporal features into a unified representation, ensuring the adaptability and generalization of the prediction model across heterogeneous passenger groups.
- We provide a theoretical analysis of the proposed STDAtt-Mambamodel by reformulating it as a dual-path kind of attention mechanism, demonstrating the parallel integration of spatial and temporal pathways. The interpretability analysis validates the complementary relationship between the STDMamba and STDAtt modules: the STDMamba module excels in capturing local dependencies through its efficient integration of ResNet-inspired state-space modeling, whereas the STDAtt module effectively captures global, long-range dependencies by employing sparse spatial-temporal attention mechanisms.
- We leverage a large-scale smart card dataset spanning 2 years of travel records in Queensland, Australia of over 1.58 million users of 9 types, including travel modes such as bus, train, and ferry. Experimental results show that the proposed STDAtt-Mamba model outperforms 19 baseline models in prediction accuracy and computational efficiency.

The rest of the chapter is organized as follows: We introduce dataset description and analytics in Section 5.2; define the problem in Section 5.3; present STDAtt-Mamba architecture in Section 5.4; provide theoretical analysis of STDAtt-Mamba in Section 5.5; present experimental setup, baseline models and experimental results in Section 5.6; summarize the chapter and suggest future research directions in Section 5.7.

5.2 Dataset Description and Analytics

In this section, we give a dataset description (Section 5.2.1) and data analytics regarding the spatial and temporal heterogeneity across travel modes and passenger groups (Section 5.2.2).

5.2.1 Dataset Description

This study uses a large-scale dataset of multimodal travel records collected from the Go Card electronic system in Queensland, Australia, spanning from January 2021 to January 2023 (2 years) of over 1.582 million unique passengers who rely on public transit services such as bus, rail, and ferry across various regions, including Greater Brisbane, Ipswich, Moreton Bay, Sunshine Coast, Redlands, and the Gold Coast, etc. This study specifically focuses on the nine most frequently occurring passenger groups within the dataset, including Adults (954,215), Seniors (162,251), Pensioners (93,215), Tertiary Students (92,870), Children (144,039), Youth (1,564), Job Seekers (10,597), School Passengers (103,255), and Gold Repatriates (3,486). Passenger types with fewer occurrences are grouped under 'Others' [5] and are not considered due to their limited representation and minimal impact on demand prediction. Each passenger is eligible for only one type of go card based on their specific personal status or concession entitlements.

Table 5.3 provides a detailed overview of the dataset across the three travel modes. For the ferry, the dataset includes travel records across 32 stations with a total of 15,569 hourly time steps.

[5] 'Others' include the following passenger types: 'Free for Rail services', 'General one-day pass for adult Passenger', 'General one-day pass for child Passenger', '50% discount on rail', 'Conc - Asylum', '0 Percent NoExp', 'Commuter Tariff', 'Free for all services', 'SV Adlt Corp', 'SV STAS', '1 Day Adult Pass', 'Conc - Tert (G)', 'SEQ 3 Day Adult', 'SEQ 5 Day Adult', '0 Percent Bus', 'SV Conc Corp', 'SEQ 5 Day Child', 'SEQ 3 Day Child', 'GB 1 Day Child', etc.

Rail involves a much larger network, with 168 stations and 17,554 hourly time steps. The bus is the most widely used travel mode, covering 14,853 stations and 18,188 hourly time steps. Each time step in the dataset represents 1 hour, capturing granular temporal dynamics of passenger demand.

TABLE 5.3. Overview of the dataset

Travel modes	# Passenger groups	# Stations	# Time-steps (hours)	Missing Ratio
Ferry	9	32	15,569	3.41%
Rail	9	168	17,554	4.49%
Bus	9	14,853	18,188	1.25%

Data cleaning. To ensure high-quality input data, we implemented a comprehensive data cleaning and preprocessing pipeline. This process involved selecting key attributes (e.g., operator, ticket type, boarding time, and location), classifying transport modes based on operator identifiers, and standardizing user groups (e.g., Adult, Child, Pensioner) by parsing various ticket type patterns. Boarding timestamps are converted to a uniform datetime format and rounded to the nearest minute to maintain consistent temporal resolution. Demand data is then aggregated into 12 time steps for each transport mode and passenger group. Missing values in the resulting time series are imputed with zeros. This approach is grounded in domain knowledge of urban mobility systems, where missing data typically represents intervals with no demand (e.g., no boardings at a stop), rather than system or sensor failures. By using zero imputation, we avoid introducing artificial bias, ensuring the integrity and interpretability of the time series, especially given the low missing data rates in our dataset (1.25%–4.49%).

5.2.2 Data Analytics

We next analyze the heterogeneous spatial and temporal travel patterns across different travel modes and passenger groups.

5.2.2.1 Spatial Heterogeneity Across Travel Modes and Passenger Groups

Figures 5.1, 5.2, and 5.3 illustrate the spatial and temporal distribution of passenger demand for bus, rail, and ferry, across 9 passenger groups. Figure 5.1 shows the spatial distribution of bus demand for 9 passenger groups, highlighting the total demand as well as the distinct travel patterns of specific passenger groups. The heatmap on the left displays the overall concentration of bus demand, which is primarily clustered around urban and suburban areas with higher population densities, such as central Brisbane and surrounding regions. The red zones indicate areas with the highest bus usage, reflecting significant demand along key transit corridors. The smaller heatmaps on the right provide a detailed breakdown of demand by passenger type. Adults, who constitute the largest passenger group, exhibit widespread demand concentrated in urban centers and major transit routes. Children and school passengers display similar patterns, with demand concentrated near residential and school areas, aligning with school travel needs. Pensioners and seniors, on the other hand, show dispersed demand, reflecting their non-commuter usage across diverse locations for leisure or essential travel. Tertiary students demonstrate high demand near educational hubs, while job seekers exhibit sparse, irregular demand patterns. Youth and Gold Repat passengers show minimal activity, with demand concentrated in specific localized areas.

Figure 5.2 illustrates the spatial distribution of rail demand across nine passenger groups, highlighting total rail demand and the unique travel patterns of specific passenger groups. Adults exhibit the highest and most widespread rail demand, particularly along primary transit corridors. Pensioners and seniors show moderate but dispersed demand, indicating rail's suitability for leisure or essential long-distance travel. School passengers and tertiary students display localized demand near educational institutions and residential areas. Children and youth contribute minimally to overall rail usage, with their demand centered around specific stations. Job seekers and Gold Repat passengers show sparse demand patterns, reflecting their limited reliance on rail.

Figure 5.3 presents the spatial distribution of ferry demand for 9 passenger groups. Adults constitute the majority of ferry passengers, with concentrated demand near urban waterfronts. Seniors and pensioners also exhibit notable usage, reflecting the role of ferries in facilitating leisure and discretionary travel for these groups. Tertiary students display localized demand

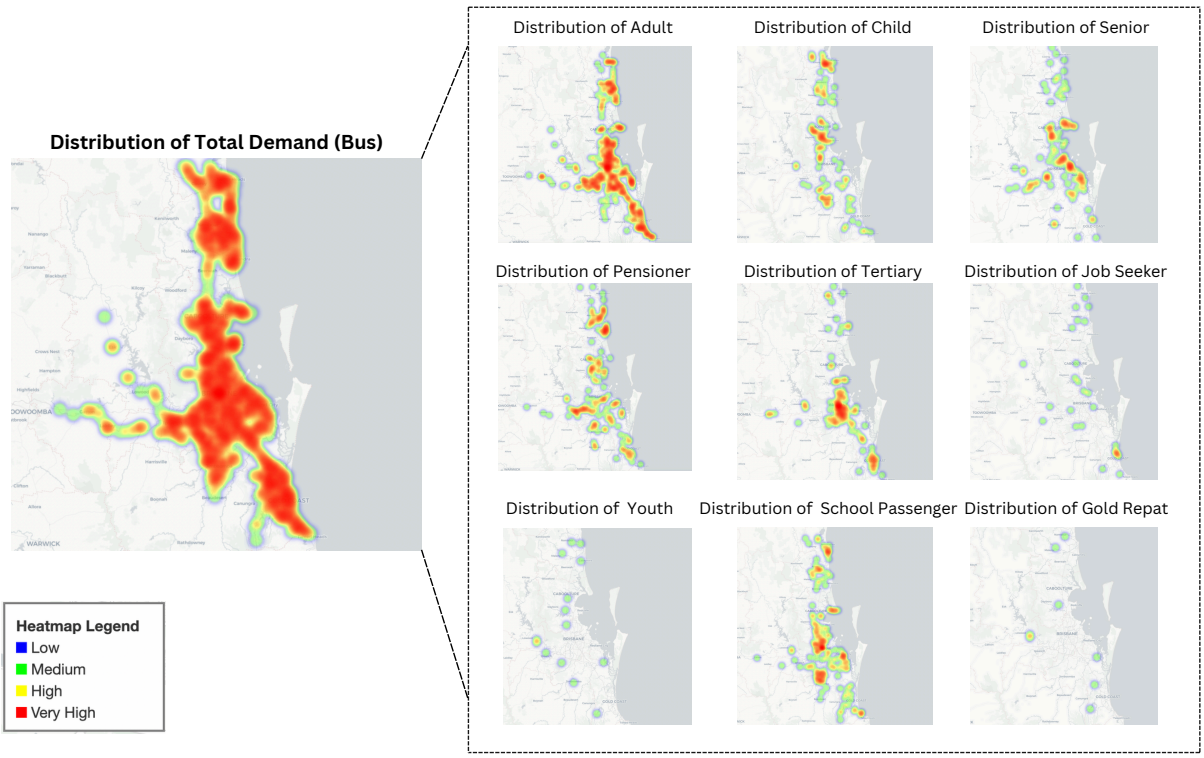


FIGURE 5.1. Spatial distribution of bus demand for 9 passenger groups

FIGURE 5.2. Spatial distribution of rail demand for 9 passenger groups

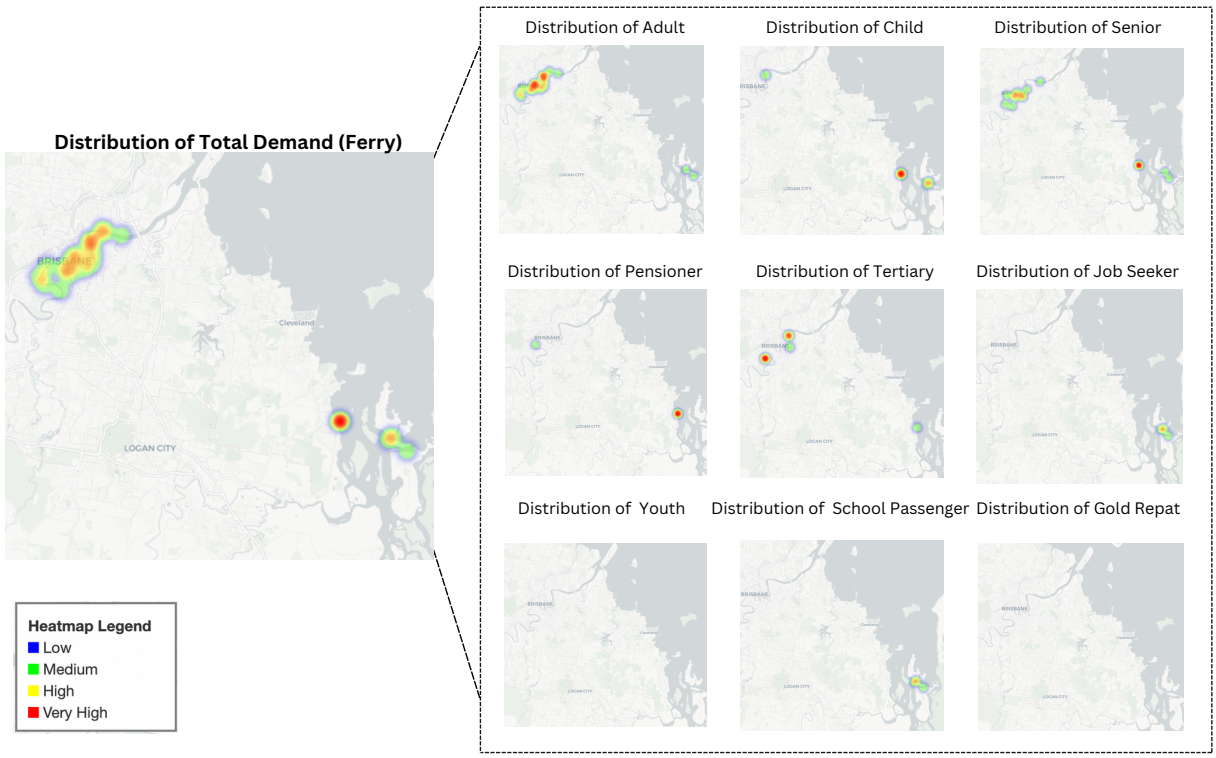


FIGURE 5.3. Spatial distribution of ferry demand for 9 passenger groups

near educational and residential hubs along ferry routes, while school passengers show similar patterns, albeit with lower intensity. Children and youth exhibit minimal ferry usage, as do job seekers and Gold Repat passengers, whose demand is sparse and concentrated around specific terminals.

5.2.2.2 Temporal Heterogeneity Across Travel Modes and Passenger Groups

Figures 5.4 - 5.7 illustrate the temporal heterogeneity in travel demand across different modes (bus, rail, ferry) and passenger groups, emphasizing variations in hourly, daily, and weekly patterns.

As shown in Figure 5.4, bus dominates in passenger demand across all periods, with pronounced peaks during the morning and evening rush hours, reflecting their crucial role in facilitating weekday commuting for a wide range of passenger groups, including adults, students, and other commuters. Rail demand follows a similar trend, with significant activity during peak hours, though at a slightly lower magnitude than buses, highlighting its effectiveness in serving concentrated corridors and long-distance travelers. In contrast, ferries exhibit minimal demand

and show relatively flat variability across periods, reflecting their geographically constrained usage, primarily catering to waterfront commuters and leisure travelers. These patterns highlight the dominance of buses in urban transit, the complementary role of rail in regional and corridor-specific transport, and the niche role of ferries, emphasizing the importance of tailored scheduling and resource allocation strategies to meet the unique temporal demands of each mode.

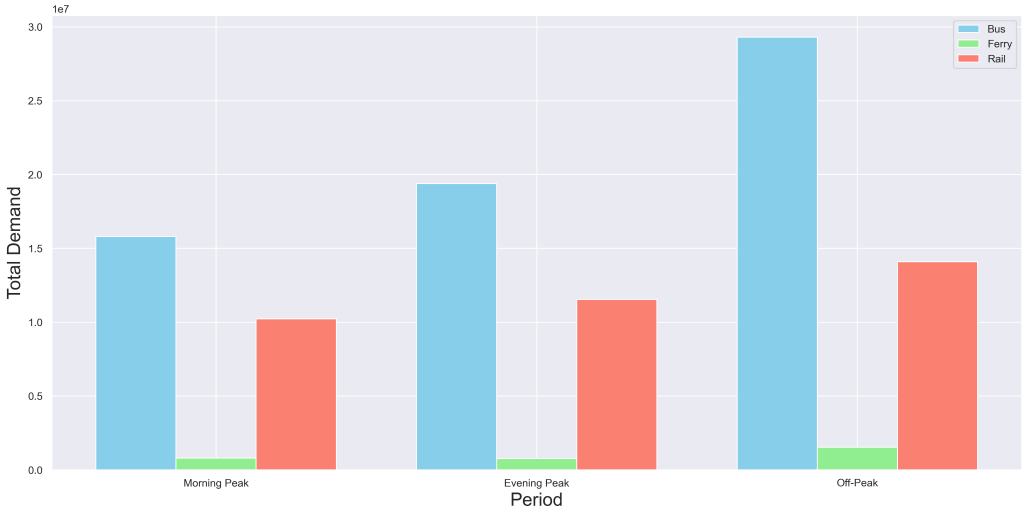


FIGURE 5.4. Demand distribution by period and travel modes

Figure 5.5 compares the hourly temporal demand patterns for buses, ferries, and rail across different hours of the day and days of the week. Bus demand is characterized by pronounced peaks during weekday morning (7-9 AM) and evening (4-6 PM) rush hours, reflecting its primary role in commuting. Demand significantly decreases during weekends, highlighting its weekday reliance. For ferries, demand shows a more balanced pattern, with consistent activity during midday hours on weekdays and slightly higher usage on weekends, indicating its dual-purpose role for commuting and leisure travel. Rail demand exhibits a mix of bus and ferry characteristics, with clear peaks during weekday rush hours and relatively steady demand throughout the day, extending into weekends, reflecting its suitability for both commuting and diverse passenger needs. These patterns emphasize the variability in temporal demand across modes, shaped by passenger behaviors and travel purposes.

Figure 5.6 compares daily demand patterns across buses, ferries, and rail, segmented by passenger groups. Bus demand is heavily concentrated during weekdays, with adults forming the largest share, followed by school passengers and children. Demand drops significantly on weekends,

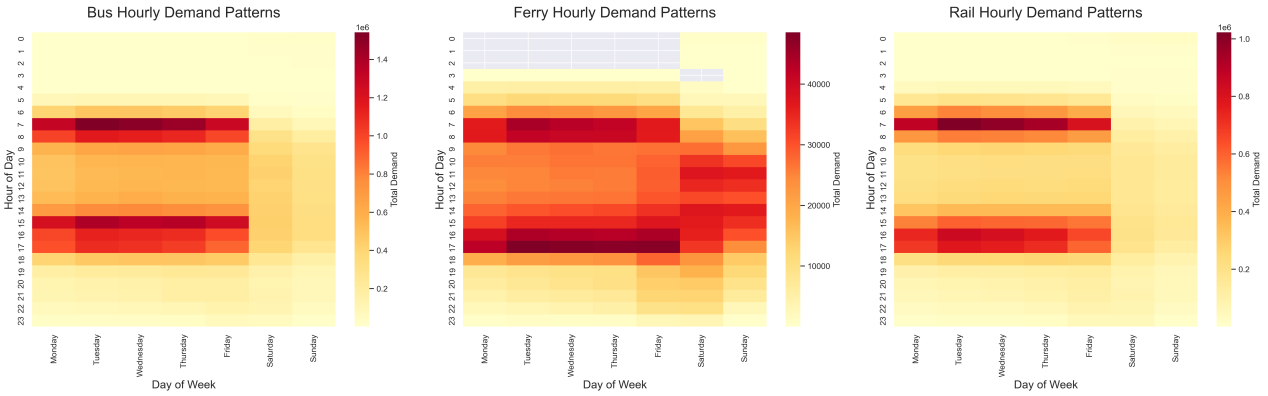


FIGURE 5.5. Comparison of hourly temporal demand patterns across three travel modes

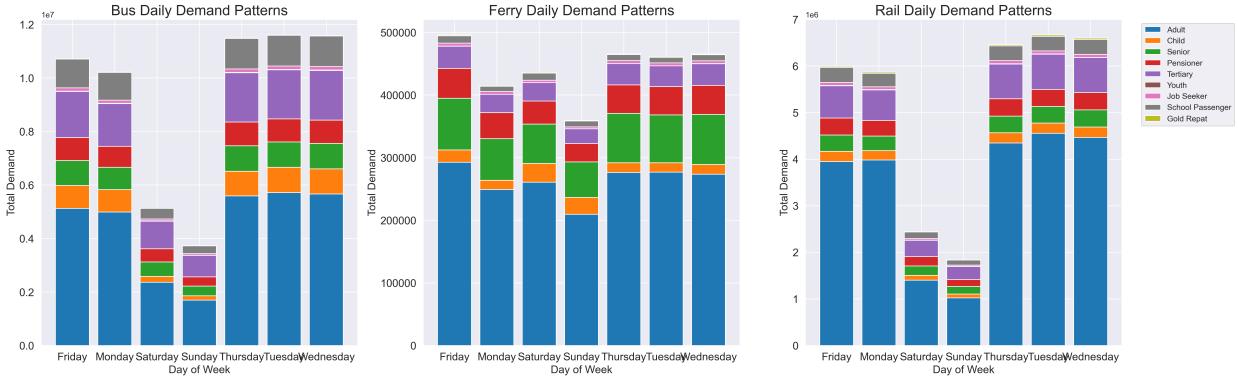


FIGURE 5.6. Comparison of daily demand patterns across passenger groups by travel modes

reflecting reduced commuter travel. Ferry demand shows a more balanced pattern, with steady usage throughout the week and higher contributions from seniors, pensioners, and adults, highlighting its dual role in commuting and leisure travel. Rail demand remains consistent on weekdays, with adults dominating, while tertiary students and school passengers contribute localized peaks. Weekend rail demand declines but remains more stable compared to buses, reflecting its broader appeal for both commuter and leisure trips. These patterns emphasize the variation in usage across modes, influenced by passenger group behaviors and the purpose of travel.

Figure 5.7 presents the weekly temporal demand patterns for 9 passenger groups across buses, ferries, and rail. Adults, the largest group, exhibit clear morning and evening peaks on weekdays, driven by commuting needs, with significantly lower activity on weekends, particularly for

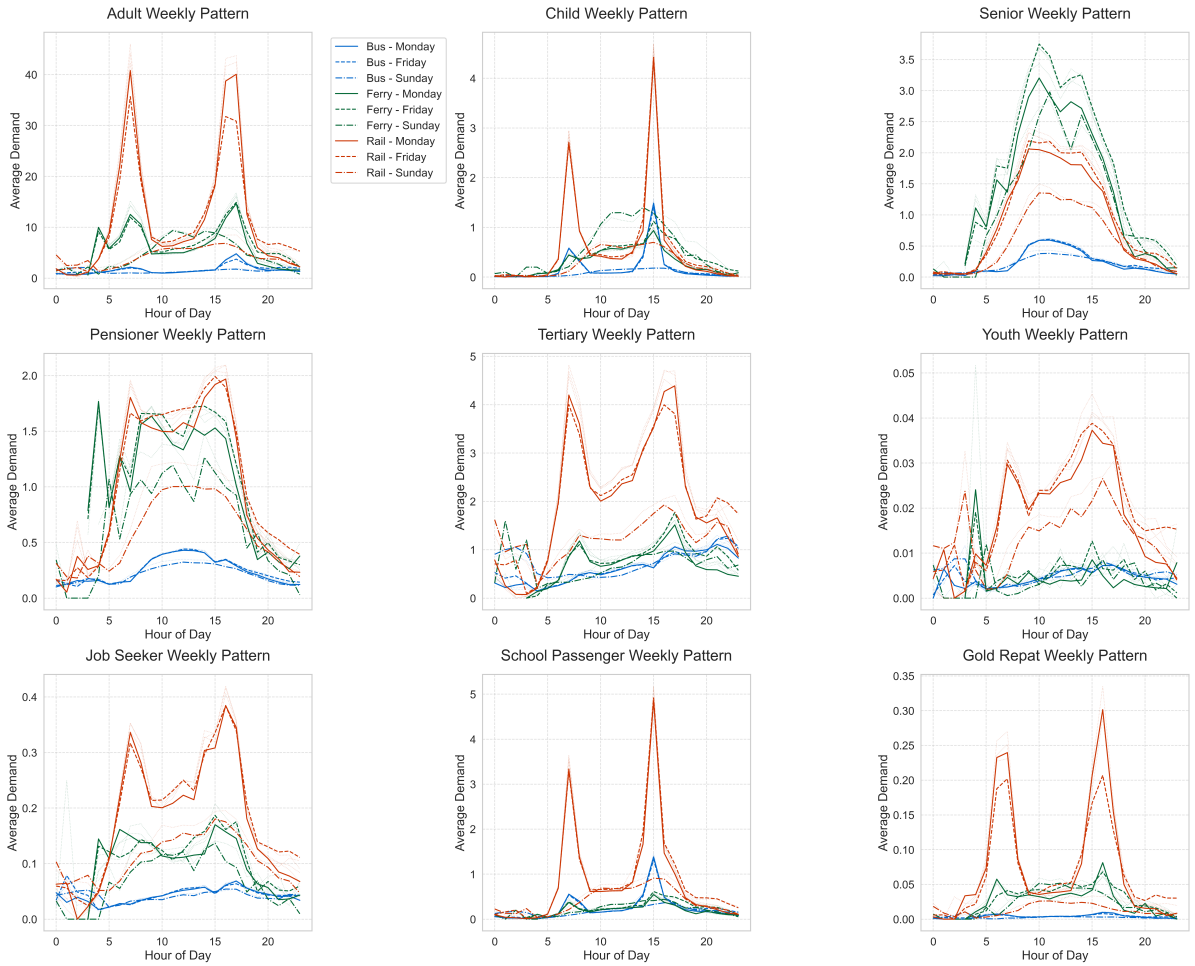


FIGURE 5.7. Comparison of weekly temporal patterns across passenger types by travel modes

bus and rail. Children and school passengers show similar peak patterns during school hours on weekdays, with negligible demand on weekends. Pensioners and seniors display relatively steady usage throughout the day, with a slight midday peak, reflecting non-commuter and leisure activities, especially on ferries and rail. Tertiary students show noticeable weekday peaks aligned with class schedules, while job seekers have sporadic, low-intensity demand across all modes. Youth and Gold Repat passengers have minimal and irregular usage, with small peaks in the morning and midday. These patterns highlight the diverse travel behavior of passenger groups and the varying temporal reliance on different travel modes.

5.3 Problem Statement

Accurately predicting passenger demand in a multimodal PT system presents considerable challenges due to the diverse travel behaviors exhibited by various passenger groups, including Adults, Children, Pensioners, Tertiary Students, Youth, Job Seekers, School Passengers, Gold Repatriates, and Seniors. Each passenger group follows distinct spatiotemporal travel patterns shaped by factors such as trip purpose, time of day, and preferred mode of transport. Existing demand prediction models often treat spatial and temporal features independently and lack the capacity to effectively model the heterogeneous dependencies that arise across different passenger types and modes. To overcome these limitations, we propose STDAtt-Mamba, a spatial-temporal dynamic attention-based state-space model designed for multi-type passenger demand prediction. The model operates within a multi-task learning framework, where each task corresponds to a specific travel mode (e.g., bus, rail, and ferry), enabling the simultaneous and unified modeling of diverse spatial-temporal behaviors across heterogeneous passenger groups.

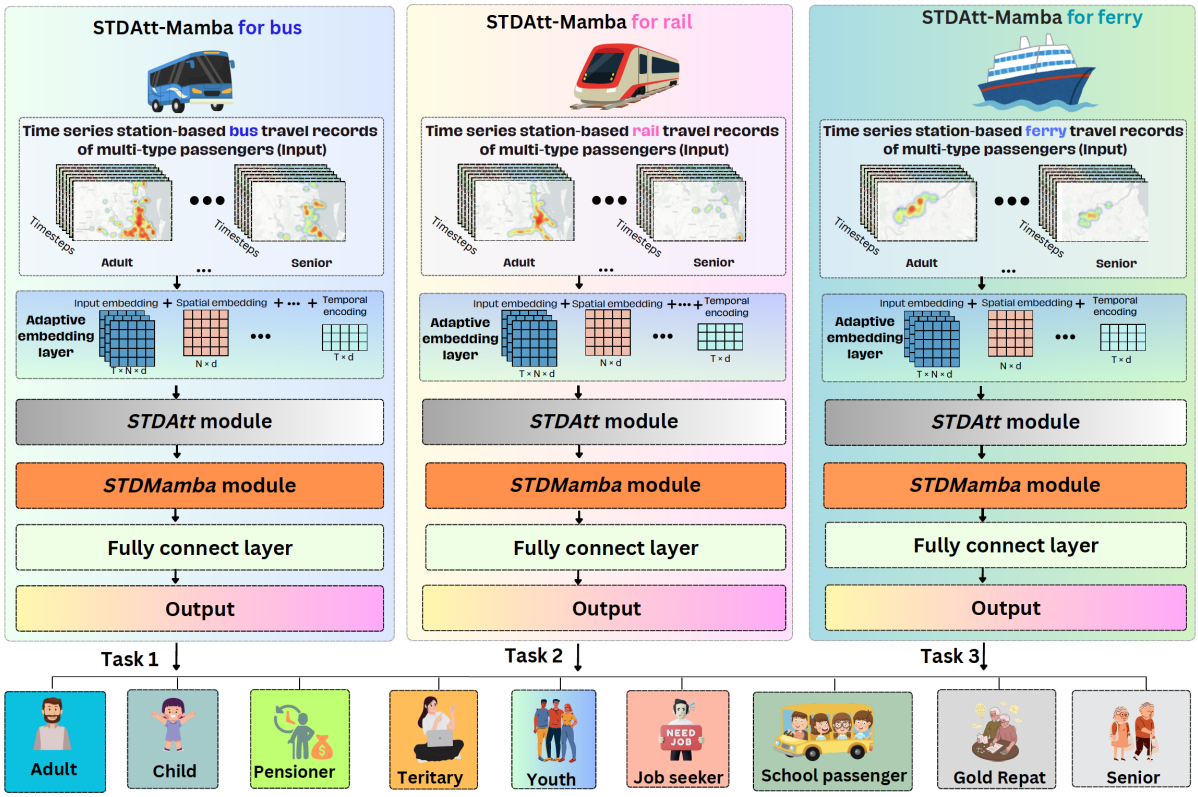


FIGURE 5.8. Multi-type passenger demand prediction in multimodal PT systems

As shown in Figure 5.8, the input data for each travel mode is processed as time-series station-based passenger-specific travel records (Definition 5.1), capturing historical demand patterns across spatial-temporal and passenger groups. An adaptive embedding layer integrates station-level, passenger-type-specific, and temporal embeddings (e.g., time-of-day, day-of-week) into a unified representation for efficient processing. The processed embeddings are passed to the spatial-temporal dynamic attention (STDAtt) module, which applies sparse attention mechanisms to capture interdependencies between spatial (stations) and temporal (time steps) travel patterns. The output of the STDAtt module is further refined by the spatial-temporal dynamic fusion (STDF) layer within STDMamba module, which integrates spatial and temporal features into a unified representation, ensuring the adaptability and generalization of the prediction model across heterogeneous passenger groups. Finally, task-specific fully connected layers generate demand predictions of multiple passenger groups for each travel mode in the multimodal PT systems.

Definition 5.1 (Input features). Let $\mathcal{M}$ denote the set of travel modes in a multimodal PT system. Let T denote the number of time steps (hours), N denote the number of stations (nodes), and d denote the number of passenger types. For any travel mode $m \in \mathcal{M}$, the input demand at time step t is represented as $\mathbf{X}_t^m \in \mathbb{R}^{N \times d}$. For a given time period $1, 2, \cdots, T$, the mode-specific time series input demand is given as follows:

$$\mathcal{X}^m = [\mathbf{X}_1^m, \mathbf{X}_2^m, \ldots, \mathbf{X}_T^m] \in \mathbb{R}^{T \times N \times d}, \quad m \in \mathcal{M}, \tag{5.1}$$

where $\mathcal{X}^m$ is structured along three dimensions: temporal (T), spatial (N), and passenger type (d).

Definition 5.2 (Multi-type passenger demand prediction in multimodal PT systems). We define a function f^m for each travel mode m to predict the station-based demand for different passenger groups over a future time period from $t+1$ to $t+Z$. The predictions are based on historical data from the time interval $t-M+1$ to t, where M denotes the number of past time steps used for training, and Z is the number of future time steps to be forecasted. Each travel mode m is treated as a distinct task within the multi-task learning framework, allowing the model to learn

the demand patterns across all modes simultaneously. This relationship is formulated as:

$$f^m : \{\mathbf{X}^m_{t-M+1}, \mathbf{X}^m_{t-M+2}, \ldots, \mathbf{X}^m_t\} \rightarrow \{\hat{\mathbf{Y}}^m_{t+1}, \hat{\mathbf{Y}}^m_{t+2}, \ldots, \hat{\mathbf{Y}}^m_{t+Z}\}, \quad m \in \mathcal{M}, \tag{5.2}$$

where $\mathbf{X}^m_t$ represents the input features at time step t, we denote the predicted features as $\hat{\mathbf{Y}}^m_t$ which has the same shape with the input features, and $\hat{\mathbf{Y}}^m_{t+Z} \in \mathbb{R}^{Z \times N \times d}$ represents the predicted demand for future time step $t + Z$ for any mode $m \in \mathcal{M}$. To train the model effectively, we set t to range from M to $T - Z$, ensuring that the model can utilize the input travel demand $\mathbf{X}^m_t$ for learning over all available time steps.

5.4 STDAtt-Mamba

In this section, we introduce the proposed STDAtt-Mambamodel, including three key components: Adaptive data embedding layer (Section 5.4.1), Spatial-temporal dynamic attention module (Section 5.4.2) , and Spatial-temporal dynamic Mamba module (Section 5.4.3).

As shown in Figure 5.9 (the blue box after data input), the adaptive embedding layer combines station-level, passenger-type-specific, and temporal embeddings into a unified representation, enabling the subsequent modules to process the data efficiently. The spatial-temporal dynamic attention (STDAtt) focuses on learning representation by applying spatial-temporal attention layers, thus reducing the computational overhead and improving scalability for large-scale datasets. The spatial-temporal dynamic fusion (STDF) layer within the *STDMamaba* merges spatial and temporal dynamic state-space models under the ResNet architecture [He et al., 2016], leveraging residual connections.

5.4.1 Adaptive Embedding Layer

To adaptively integrate station-level, passenger-type-specific, and temporal embeddings into a unified representation, we propose an adaptive embedding layer that integrates input feature embeddings, spatial embeddings, and temporal embeddings, including passengers' day-of-week and time-of-day travel patterns (see Section 5.2.2.2).

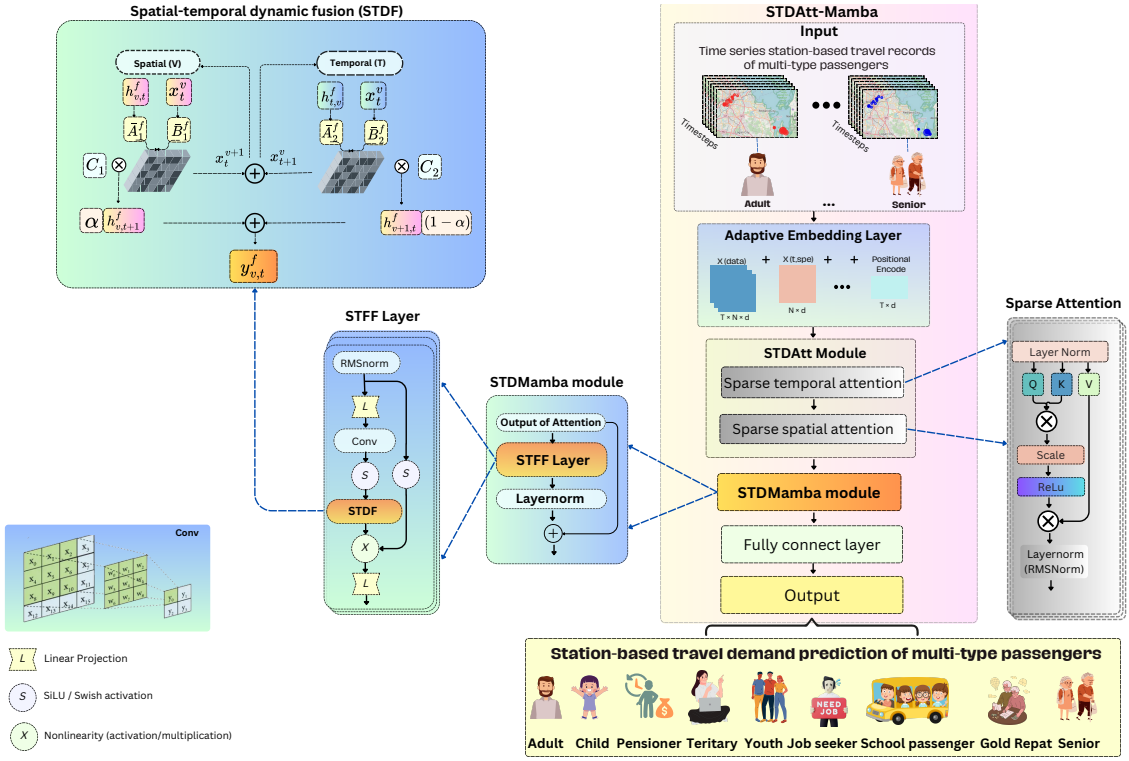


FIGURE 5.9. STDAtt-Mamba architecture

For any travel mode $m \in \mathcal{M}$, given a station-based time series input $\mathbf{X}^m_{t-M+1:t} \in \mathbb{R}^{M\times N\times d}$ (see Definition 5.1), we apply a fully connected (FC) layer to implement the input feature embedding:

$$\mathbf{E}^{fm}_t = \mathrm{FC}_m(\mathbf{X}^m_{t-M+1:t}) \in \mathbb{R}^{M\times N\times d_f}, \quad m \in \mathcal{M}, \tag{5.3}$$

where M represents the number of past time steps used for training, N denotes the number of stations, and d_f denotes the dimension of the embedding $\mathbf{E}^{fm}_t$. For any travel mode $m \in \mathcal{M}$, the temporal embeddings consist of day-of-week embeddings $\mathbf{E}^m_w \in \mathbb{R}^{T_w\times d_f}$ and time-of-day embeddings $\mathbf{E}^m_d \in \mathbb{R}^{T_d\times d_f}$, where $T_w = 7$ corresponds to the days of the week and $T_d = 24$ represents 1 hour intervals throughout a day. For a time point t, we obtain the corresponding embeddings $\mathbf{E}^{wm}_t \in \mathbb{R}^{M\times d_f}$ and $\mathbf{E}^{dm}_t \in \mathbb{R}^{M\times d_f}$ from these base embeddings. Considering a look-back window of size M at time t, we construct temporal embedding sequences $\mathbf{E}^{wm}_t \in \mathbb{R}^{M\times d_f}$ and $\mathbf{E}^{dm}_t \in \mathbb{R}^{M\times d_f}$ from these base embeddings. The temporal embeddings are concatenated and spatially expanded to $\mathbf{E}^{pm}_t$, given as follows:

$$\mathbf{E}^{pm}_t = \mathrm{Repeat}_N([\mathbf{E}^{wm}_t, \mathbf{E}^{dm}_t]) \in \mathbb{R}^{M\times N\times 2d_f}, \quad m \in \mathcal{M}, \tag{5.4}$$

where Repeat_N indicates N-fold repetition across the spatial dimension. Considering the inherent cyclical patterns, sequential dependencies, and the complexity of multi-task prediction in spatial-temporal data, we introduce a shared adaptive embedding $\mathbf{E}_t^{am} \in \mathbb{R}^{M \times N \times d_a}$, where d_a represents its dimensionality. This embedding is specifically designed to self-adjust for different prediction tasks, allowing it to adapt to varying passenger groups and demands. Initialized with the Xavier uniform distribution, $\mathbf{E}_t^{am}$ serves as a flexible parameter to be optimized during the training process, ensuring it effectively captures task-specific features while maintaining shared learning across tasks. In addition to capturing station-level and temporal embeddings (e.g., day-of-week and time-of-day), the shared adaptive embedding $\mathbf{E}_t^{am}$ enables the model to flexibly handle multiple passenger groups. By introducing a trainable parameter updated through backpropagation, the embedding learns to identify which user types share similar spatiotemporal patterns and which require distinct representations. For example, Adults and Tertiary Students may exhibit similar travel patterns during common morning or evening peaks, whereas School Passengers may follow different peak schedules. This adaptive embedding layer thus enables the model to "borrow strength" from other user groups when appropriate and specialize when their behaviors diverge. Ultimately, this design strengthens the multi-task learning framework by emphasizing shared features across user types, reducing overfitting, and improving the predictive accuracy for each passenger group. The final representation integrates these three embedding components through concatenation:

$$\mathbf{E}_t^m = \text{Concat}\left(\mathbf{E}_t^{fm} || \mathbf{E}_t^{pm} || \mathbf{E}_t^{am}\right), \quad m \in \mathcal{M}, \tag{5.5}$$

resulting in a spatial-temporal representation $\mathbf{E}_t^m \in \mathbb{R}^{M \times N \times d_h}$, where the hidden dimension d_h is defined as $d_h = 3d_f + d_a$.

The feature engineering process (Adoptive Embedding Layer) is designed to address the heterogeneous travel patterns observed across different passenger groups. The station-based time series input, $\mathbf{X}_{t-M+1:t}^m \in \mathbb{R}^{M \times N \times d}$, captures demand patterns for each passenger type, where the dimension d represents the nine passenger groups (e.g., Adults, Seniors, Pensioners, etc.) as outlined in Section 5.2.1. Although we do not explicitly incorporate external socioeconomic variables, the categorization of passenger types inherently reflects various socioeconomic

segments, such as Job Seekers, Pensioners, and Tertiary Students. This implicit encoding of socioeconomic factors is further improved by the adaptive embedding $\mathbf{E}_t^{am}$, which adjusts dynamically during training to capture relationships between passenger types with similar behaviors.

For temporal patterns, we explicitly model both day-of-week ($\mathbf{E}_w^m$) and time-of-day ($\mathbf{E}_d^m$) embeddings, effectively capturing the cyclical temporal dependencies discussed in Section 5.2.2.2. To mitigate potential biases introduced by aggregating data across different travel modes, each mode $m \in \mathcal{M}$ is treated as an independent task within the multi-task learning framework. This allows the model to learn mode-specific patterns while enabling the sharing of higher-level representations through the adaptive embedding layer. This approach ensures that mode-specific characteristics and potential biases are preserved and learned separately, while still benefiting from the shared knowledge across the multimodal PT system.

Applying STDAtt before the STDMamba module can reduce dimension, filter noise, and enrich features, thereby improving learning dynamics and promoting better gradient flow and convergence by focusing on crucial features.

5.4.2 Spatial-temporal Dynamic Attention Module

We next introduce the proposed spatial-temporal dynamic attention (STDAtt) module, which first focuses on temporal dependencies and then on spatial interactions. Unlike the self-attention mechanism that relies on softmax normalization Vaswani [2017], we use a ReLU-based sparse attention mechanism to capture the most crucial periods and stations for different passenger groups. As shown in Figure 5.10, the self-attention computes attention scores using the softmax function, which ensures all input tokens contribute to the output representation by normalizing the attention weights to sum to one. This mechanism inherently assigns some level of importance to all tokens, even those that may not be relevant. In contrast, the sparse attention replaces softmax normalization with a ReLU scoring mechanism and introduces sparsity by zeroing out negative values, which ensures that the most relevant temporal and spatial features are retained, thereby significantly reducing noise and computational burden. Additionally, the STDAtt module

implements the sparse temporal and spatial attention mechanisms sequentially to capture the intricate spatial-temporal dynamics for multi-type passenger demand effectively.

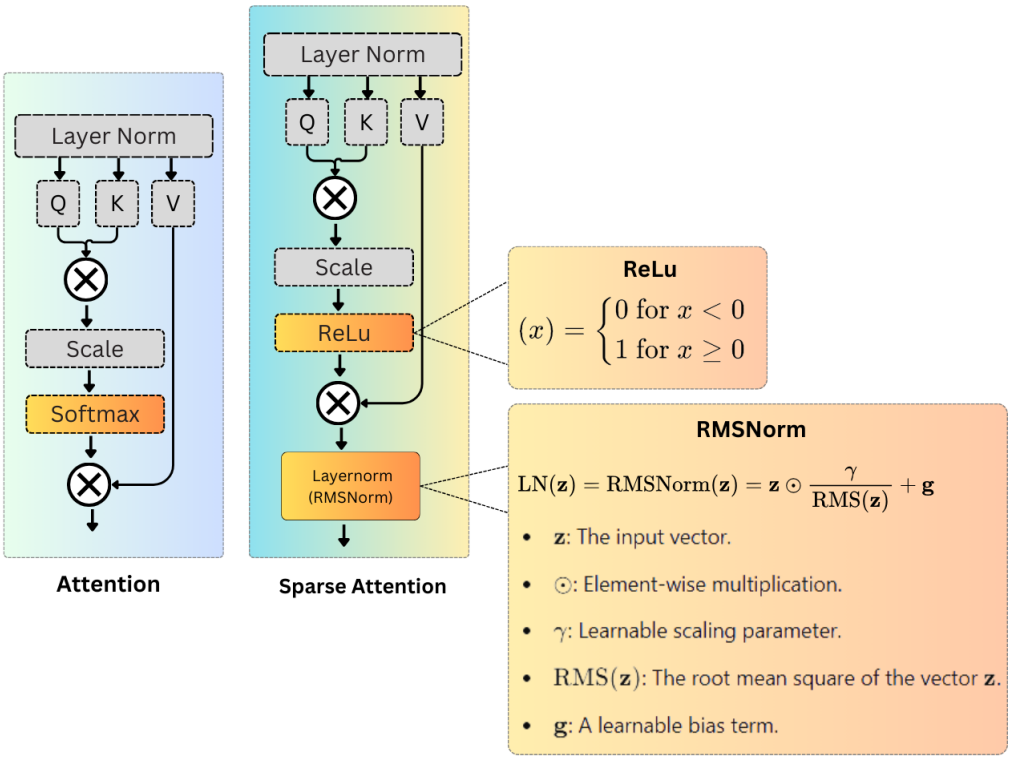


FIGURE 5.10. The difference between self-attention and sparse self-attention mechanisms

Sparse temporal attention Layer. The sparse temporal attention layer allows the model to focus on key time steps that contribute most to the prediction. This is particularly beneficial in scenarios where certain periods have a more substantial influence on the target variable than others. To capture global temporal features, we process each spatial node individually. Let v denote a certain station (node) $v \in \{1, 2, \ldots, N\}$. For any travel mode $m \in \mathcal{M}$, given an input embedding vector $\mathbf{E}_t^m \in \mathbb{R}^{M \times N \times d_h}$ representing spatiotemporal input data (Eq. (5.5)), for each travel mode, let $\mathbf{E}_v^{m(t)} \in \mathbb{R}^{M \times d_h}$ denote the temporal input features of each node v across all time steps. We then compute the query ($\mathbf{Q}_v^{(t)}$), key ($\mathbf{K}_v^{(t)}$), and value ($\mathbf{V}_v^{(t)}$) matrices as follows:

$$\mathbf{Q}_v^{(t)} = \mathbf{E}_v^{m(t)} \mathbf{W}_{tQ}, \tag{5.6}$$

$$\mathbf{K}_v^{(t)} = \mathbf{E}_v^{m(t)} \mathbf{W}_{tK}, \tag{5.7}$$

$$\mathbf{V}_v^{(t)} = \mathbf{E}_v^{m(t)} \mathbf{W}_{tV}, \tag{5.8}$$

where $\mathbf{W}_{tQ}, \mathbf{W}_{tK}, \mathbf{W}_{tV} \in \mathbb{R}^{d_h \times d_h}$ are learnable weight matrices for the temporal attention.

Let $\mathbf{O}_v^{(t)}$ denote the self-attention scores for node v and are computed using scaled dot-product in Eq. (5.9). The scores $\mathbf{O}_v^{(t)}$ capture temporal dependencies at node v by evaluating the relationships between different time steps, and are then normalized using the ReLU function as follows:

$$\mathbf{O}_v^{(t)} = \mathrm{ReLU}\left(\frac{\mathbf{Q}_v^{(t)} (\mathbf{K}_v^{(t)})^\top}{\sqrt{d_h}}\right) \in \mathbb{R}^{M \times M}, \tag{5.9}$$

where $\sqrt{d_h}$ is the scaling factor that mitigates the risk of over-large dot products. The ReLU function in Eq. (5.9) ensures that the non-negativity of attention scores $\mathbf{O}_v^{(t)}$ across the temporal dimension, reducing the computational cost.

Let LN denote the RMSNorm, which scales the input vector by computing the root mean square and avoiding bias shifts introduced by mean subtraction [Zhang and Sennrich, 2019a]. The sparse attention mechanism is particularly effective in capturing key dependencies in complex spatial-temporal data, as it prioritizes salient features while minimizing interference from irrelevant inputs. Let $\tilde{\mathbf{E}}_v^{m(t)}$ denote the improved temporal embedding input for node v, which is obtained by aggregating the value vectors weighted by the attention scores:

$$\tilde{\mathbf{E}}_v^{m(t)} = \mathrm{LN}\left(\mathbf{O}_v^{(t)} \mathbf{V}_v^{(t)}\right) \in \mathbb{R}^{M \times d_h}, \quad m \in \mathcal{M}. \tag{5.10}$$

By concatenating the outputs for all stations (nodes), we obtain the improved temporal input embedding vector as follows:

$$\tilde{\mathbf{E}}_t^m = \left[\tilde{\mathbf{E}}_1^{m(t)}, \tilde{\mathbf{E}}_2^{m(t)}, \ldots, \tilde{\mathbf{E}}_v^{m(t)}\right] \in \mathbb{R}^{M \times N \times d_h}, \quad m \in \mathcal{M}, \tag{5.11}$$

which incorporates temporal features across all stations (nodes).

Sparse spatial attention layer. To capture spatial dependencies across multi-type passenger demand, we apply sparse spatial attention to the output of sparse temporal attention, i.e. $\tilde{\mathbf{E}}_t^m$ in (5.11). For this purpose, consider each of its M slices $\tilde{\mathbf{E}}_{M_0}^{m(v)} \in \mathbb{R}^{N\times d_h}$ where $M_0 \in \{1, 2, \cdots, M\}$. The queries ($\mathbf{Q}_{M_0}^{(v)}$), keys ($\mathbf{K}_{M_0}^{(v)}$), and values ($\mathbf{V}_{M_0}^{(v)}$) matrices of sparse spatial attention for node v, relevant to the slice $\tilde{\mathbf{E}}_{M_0}^{m(v)}$ at time period t are computed as follows, $v \in \{1, 2, \cdots, N\}$, $M_0 \in \{1, 2, \cdots, M\}$:

$$\mathbf{Q}_{M_0}^{(v)} = \tilde{\mathbf{E}}_{M_0}^{m(v)}\mathbf{W}_{\mathrm{v}Q}, \tag{5.12}$$

$$\mathbf{K}_{M_0}^{(v)} = \tilde{\mathbf{E}}_{M_0}^{m(v)}\mathbf{W}_{\mathrm{v}K}, \tag{5.13}$$

$$\mathbf{V}_{M_0}^{(v)} = \tilde{\mathbf{E}}_{M_0}^{m(v)}\mathbf{W}_{\mathrm{v}V}, \tag{5.14}$$

where $\mathbf{W}_{\mathrm{v}Q}, \mathbf{W}_{\mathrm{v}K}, \mathbf{W}_{\mathrm{v}V} \in \mathbb{R}^{d_h\times d_h}$ are learnable weight matrices for the spatial attention.

The spatial attention scores for time step t are calculated as follows:

$$\mathbf{O}_{M_0}^{(v)} = \mathrm{ReLU}\left(\frac{\mathbf{Q}_{M_0}^{(v)}\left(\mathbf{K}_{M_0}^{(v)}\right)^\top}{\sqrt{d_h}}\right) \in \mathbb{R}^{N\times N}, \quad M_0 \in \{1, 2, \cdots, M\}, \tag{5.15}$$

where $\sqrt{d_h}$ is the scaling factor in the attention computations, which can stabilize the gradients during training by preventing the dot products from being too large. The attention scores $\mathbf{O}_{M_0}^{(v)}$ quantify the influence of each node on others at step M_0 within the time window at the time step t, effectively capturing spatial relationships among stations (nodes).

Let $\mathbf{E}_t^{m(v)}$ denote the improved spatial embedding input at time period t, which is computed by weighting the value vectors with the attention scores:

$$\mathbf{E}_{M_0}^{m(v)} = \mathrm{LN}\left(\mathbf{O}_{M_0}^{(v)}\mathbf{V}_{M_0}^{(v)}\right) \in \mathbb{R}^{N\times d_h}, \quad m \in \mathcal{M}. \tag{5.16}$$

$\mathbf{E}_{M_0}^{m(v)}$ emphasizes the most relevant features across stations (nodes) at step M_0 within the time window at the time step t for mode m. The sparse spatial attention layer emphasizes crucial stations within the multimodal PT systems.

We concatenate the input vectors across all time steps and obtain the final input vector $\widehat{\mathbf{X}}_t^m$, which integrates both temporal and spatial dependencies of multi-type passenger demand.

$$\widehat{\mathbf{X}}_t^m = \left[\mathbf{E}_1^{m(v)}, \mathbf{E}_2^{m(v)}, \ldots, \mathbf{E}_M^{m(v)}\right] \in \mathbb{R}^{M\times N\times d_h}, \quad m \in \mathcal{M}. \tag{5.17}$$

5.4.3 Spatial-temporal Dynamic Mamba Module

The emerging Mamba architecture shows exceptional proficiency in capturing temporal dependencies through its state-space architecture [Gu and Dao, 2023]. However, it is limited by its inability to unitedly model temporal evolution and spatial correlations. To address this limitation, we propose a spatial-temporal dynamic fusion (STDF) layer, which extends the state-space modeling for the concurrent processing of spatial and temporal dimensions.

We first analyze the discretization of the STDF layer, which facilitates efficient and scalable implementation in large-scale multimodal PT systems, as given in Proposition 5.3.

Proposition 5.3. *The STDF layer transforms a continuous dual-path dynamic system, integrating spatial and temporal dependencies into a discrete-time representation suitable for practical implementation while preserving the essential dynamics of multimodal PT systems.*

Proof. To analyze the transformation of the continuous-time system, we consider trainable parameters and the corresponding dual-path system dynamics for two-dimensional time series as follows:

$$\frac{\partial}{\partial \tilde{t}^{(1)}}\mathbf{h}(\tilde{t}^{(1)}) = \mathbf{A}_1\mathbf{h}(\tilde{t}^{(1)}) + \mathbf{B}_1\mathbf{x}(\tilde{t}^{(1)}), \tag{5.18}$$

$$\frac{\partial}{\partial \tilde{t}^{(2)}}\mathbf{h}(\tilde{t}^{(2)}) = \mathbf{A}_2\mathbf{h}(\tilde{t}^{(2)}) + \mathbf{B}_2\mathbf{x}(\tilde{t}^{(2)}), \tag{5.19}$$

$$\mathbf{y}(\tilde{t}^{(1)}, \tilde{t}^{(2)}) = \alpha\mathbf{C}_1\mathbf{h}(\tilde{t}^{(1)}) + (1-\alpha)\mathbf{C}_2\mathbf{h}(\tilde{t}^{(2)}), \tag{5.20}$$

where $\mathbf{h}(\tilde{t}^{(1)})$ and $\mathbf{h}(\tilde{t}^{(2)})$ represent hidden states along two time dimensions. The system matrices $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{C}_1$, and $\mathbf{C}_2$ define the system dynamics.

To derive the discrete-time formulation, we apply the zero-order hold (ZOH) method. For a general continuous-time state equation:

$$\frac{d}{dt}\mathbf{h}(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{B}\mathbf{x}(t). \tag{5.21}$$

Under the ZOH assumption, $\mathbf{x}(t)$ remains constant over each interval $[k\Delta t, (k+1)\Delta t)$, the solution is given by:

$$\mathbf{h}((k+1)\Delta t) = e^{\mathbf{A}\Delta t}\mathbf{h}(k\Delta t) + \int_0^{\Delta t} e^{\mathbf{A}M}\mathbf{B}\mathbf{x}(k\Delta t)dM, \tag{5.22}$$

where M is the local time variable within each interval. The integral term simplifies to:

$$\int_0^{\Delta t} e^{\mathbf{A}M}dM\mathbf{B}\mathbf{x}(k\Delta t) = \mathbf{A}^{-1}(e^{\mathbf{A}\Delta t} - \mathbf{I})\mathbf{B}. \tag{5.23}$$

Applying this derivation to each of the two "time-like" dimensions in the dual-path system, we obtain two discrete update equations:

$$\mathbf{h}_{k+1}^{(1)} = \overline{\mathbf{A}}_1\mathbf{h}_k^{(1)} + \overline{\mathbf{B}}_1\mathbf{x}_k \tag{5.24}$$

$$\mathbf{h}_{k+1}^{(2)} = \overline{\mathbf{A}}_2\mathbf{h}_k^{(2)} + \overline{\mathbf{B}}_2\mathbf{x}_k \tag{5.25}$$

$$\mathbf{y}_k = \alpha\,\mathbf{C}_1\mathbf{h}_{k+1}^{(1)} + (1-\alpha)\,\mathbf{C}_2\mathbf{h}_{k+1}^{(2)} \tag{5.26}$$

which together capture the discretized dynamics of both paths; the output equation then becomes $\mathbf{y}_k$, where

$$\bar{\mathbf{A}}_i = e^{\mathbf{A}_i\Delta t}, \forall i \in \{1,2\} \tag{5.27}$$

$$\bar{\mathbf{B}}_i = (\mathbf{A}_i)^{-1}(e^{\mathbf{A}_i\Delta t} - \mathbf{I})\mathbf{B}_i \forall i \in \{1,2\}. \tag{5.28}$$

In our implementations, one explicitly treats the second path as a spatial index v and writes analogous continuous-time equations:

$$\frac{d}{dt}\mathbf{h}_v^f(t) = \mathbf{A}_1^f \mathbf{h}_v^f(t) + \mathbf{B}_1^f \mathbf{x}_{v+1}(t), \tag{5.29}$$

$$\mathbf{h}_t^f(v+1) = \mathbf{A}_2^f \mathbf{h}_t^f(v) + \mathbf{B}_2^f \mathbf{x}_{t+1}(v), \tag{5.30}$$

$$\mathbf{y}_{v,t}^f = \alpha \left(\mathbf{h}_v^f(t)\mathbf{C}_1\right) + (1-\alpha)\left(\mathbf{h}_t^f(v)\mathbf{C}_2\right). \tag{5.31}$$

where $\mathbf{h}_v^f(t)$ and $\mathbf{h}_t^f(v)$ represent hidden states along the spatial and temporal dimensions, respectively. The inputs $\mathbf{x}_{v+1}(t)$ and $\mathbf{x}_{t+1}(v)$ represent the information from adjacent nodes and future time steps, respectively. Matrices $\mathbf{A}_1^f, \mathbf{A}_2^f, \mathbf{B}_1^f, \mathbf{B}_2^f, \mathbf{C}_1$, and $\mathbf{C}_2$ define the system dynamics.

To derive the discrete-time formulation, we apply the zero-order hold (ZOH) method shown as in Eq. (5.21) to (5.23). The derived discrete-time updates for the spatial and temporal dual paths are as follows. The spatial path is updated as:

$$\mathbf{h}_v^f(k+1) = \bar{\mathbf{A}}_1 \mathbf{h}_v^f(k) + \bar{\mathbf{B}}_1 \mathbf{x}_{v+1}(k), \tag{5.32}$$

$$\bar{\mathbf{A}}_1 = e^{\mathbf{A}_1^f \Delta t}, \tag{5.33}$$

$$\bar{\mathbf{B}}_1 = (\mathbf{A}_1^f)^{-1}(e^{\mathbf{A}_1^f \Delta t} - \mathbf{I})\mathbf{B}_1^f, \tag{5.34}$$

where k means $k\Delta t$.

The temporal path is updated as follows,

$$\mathbf{h}_k^f(v+1) = \mathbf{A}_2^f \mathbf{h}_v^f(k) + \mathbf{B}_2^f \mathbf{x}_v(k+1). \tag{5.35}$$

The final discrete-time system is given as follows (now we use discrete time step index t to replace k):

$$\mathbf{h}_v^f(t+1) = \bar{\mathbf{A}}_1 \mathbf{h}_v^f(t) + \bar{\mathbf{B}}_1 \mathbf{x}_{v+1}(t), \tag{5.36}$$

$$\mathbf{h}_t^f(v+1) = \bar{\mathbf{A}}_2 \mathbf{h}_t^f(v) + \bar{\mathbf{B}}_2 \mathbf{x}_v(t+1), \tag{5.37}$$

$$\mathbf{y}_{v,t}^f = \alpha(\mathbf{h}_v^f(t+1)\mathbf{C}_1) + (1-\alpha)(\mathbf{h}_t^f(v+1)\mathbf{C}_2). \tag{5.38}$$

This completes the proof. □

The STDF layer models spatial and temporal interactions effectively by processing data that represent temporal and spatial states. For mode $m \in \mathcal{M}$, given a 3D tensor $\mathbf{X}^m \in \mathbb{R}^{M\times N\times d_h}$ representing M time steps of data across N nodes with feature dimension d_h, we consider at a specific location v, the temporal input, i.e., $\widehat{\mathbf{X}}_t^m \in \mathbb{R}^{M\times d_h}$, $m \in \mathcal{M}$, captures the states at a specific node v over M time steps for mode m, while at a specific time t, the spatial input, i.e., $\widehat{\mathbf{X}}_v^m \in \mathbb{R}^{N\times d_h}$, represents the states across N nodes at time t for mode $m \in \mathcal{M}$. Let $i \in \{1, 2\}$ denote the pathways for spatial ($i = 1$) and temporal ($i = 2$) interactions, respectively. For each path i, the state transition matrices $\mathbf{A}_i \in \mathbb{R}^{d_{\text{state}}\times d_{\text{state}}}$ are defined using the HiPPO method employed by Mamba [Gu and Dao, 2023], while input projection matrices $\mathbf{B}_i \in \mathbb{R}^{d_{\text{state}}\times d_h}$ are computed as $\mathbf{B}_i = s_B(\mathbf{x}_i)$ and output projection matrices $\mathbf{C}_i \in \mathbb{R}^{d_h\times d_{\text{state}}}$ are given by $\mathbf{C}_i = s_C(\mathbf{x}_i)$ where $s_B(\cdot)$ and $s_C(\cdot)$ are learnable linear projection.

The proof of Proposition 5.3 prompts us to the following algorithm design. For the sake of convenience, we introduce new notation as $\mathbf{h}_{v,t} \in \mathbb{R}^{d_{\text{state}}}$ and $\mathbf{h}_{t,v} \in \mathbb{R}^{d_{\text{state}}}$ to denote the updated spatial and temporal hidden states, respectively, for each path $i \in \{1, 2\}$, the spatial and temporal hidden states are updated as follows,

$$\mathbf{h}_{v,t} = \left(\mathbf{h}_{v,t-1} \cdot \bar{\mathbf{A}}_1 + \bar{\mathbf{B}}_1 \cdot \mathbf{x}_t^{v+1}\right) \cdot \mathbf{C}_1, \tag{5.39}$$

$$\mathbf{h}_{t,v} = \left(\mathbf{h}_{t,v-1} \cdot \bar{\mathbf{A}}_2 + \bar{\mathbf{B}}_2 \cdot \mathbf{x}_{t+1}^{v}\right) \cdot \mathbf{C}_2, \tag{5.40}$$

$$\mathbf{y}_{v,t} = \alpha\mathbf{h}_{v,t} + (1 - \alpha)\mathbf{h}_{t,v}, \tag{5.41}$$

where d_{state} denotes the dimension of the state space used in the STDMamba module.

The STDF layer output $\mathbf{y}_{v,t}$ for each spatial-temporal position (v, t) is computed as a weighted combination of spatial and temporal contributions as in Equation 5.41,

where $\alpha \in [0, 1]$ is a trainable parameter that adaptively balances the spatial and temporal contributions. The outputs $\mathbf{y}_{v,t}$ for each spatial-temporal position (v, t) are concatenated to form

the output of STDF layer $\mathcal{Y}^m$ for mode m:

$$\mathbf{Y}^m = \text{Concat}\left(\mathbf{y}_{v,t}\right) \in \mathbb{R}^{N\times M\times d_h}, \quad m \in \mathcal{M}. \tag{5.42}$$

Recall that the final input vector for mode m is denoted as $\widehat{\mathbf{X}}^m$ in Eq. (5.17). The STDMamba module consists of a STDF layer followed by layer normalization and a feed-forward network given as follows:

$$\widetilde{\mathbf{X}}_1^m = \text{LayerNorm}(\widehat{\mathbf{X}}^m + \text{STDMamba}(\mathbf{Y}^m)), \quad m \in \mathcal{M}, \tag{5.43}$$

$$\widetilde{\mathbf{X}}_2^m = \text{LayerNorm}(\widetilde{\mathbf{X}}_1^m + \text{FFN}(\widetilde{\mathbf{X}}_1^m)), \quad m \in \mathcal{M}. \tag{5.44}$$

Let $\mathcal{L}$ denote the number of STDMamba Blocks, and the STDAtt-Mambamodel applies multiple STDMamba Blocks sequentially:

$$\widehat{\mathbf{Y}}^m = \text{STDMambaBlock}_L(\text{STDMambaBlock}_{L-1}(\cdots \text{STDMambaBlock}_1(\widetilde{\mathbf{X}}_t^m)\cdots)), m \in \mathcal{M}. \tag{5.45}$$

The final output of STDMamba module for mode m is obtained through the following projection:

$$\hat{\mathbf{Y}}_{\text{out}}^m = \text{OutputProjection}(\widehat{\mathbf{Y}}^m) \in \mathbb{R}^{Z\times N\times d}, \quad m \in \mathcal{M}. \tag{5.46}$$

The mixed projection in Eq. (5.46) integrates reshaping and linear transformation into independent projections, where the temporal and feature dimensions are processed separately.

5.5 Theoretical Properties of STDAtt-Mamba

In this section, we explore the theoretical properties of the proposed STDAtt-Mambamodel, highlighting its dual-path attention mechanisms and complementary strengths. STDAtt-Mambamodel is specialized designed to capture complex spatial-temporal dynamics in multimodal PT systems. Proposition 5.4 demonstrates that the STDAtt-Mamba model can be reformulated as a spatial-temporal dual-path attention mechanism, where local and global attention paths operate

in parallel to process and integrate contextual information. Proposition 5.5 further elaborates on the complementary nature of the STDMamba and STDAtt modules. The STDMamba module, leveraging ResNet and state-space architectures, excels in capturing local attention by integrating spatial-temporal features and prior state information, while the STDAtt module focuses on global attention to capturing long-range dependencies in multimodal PT systems.

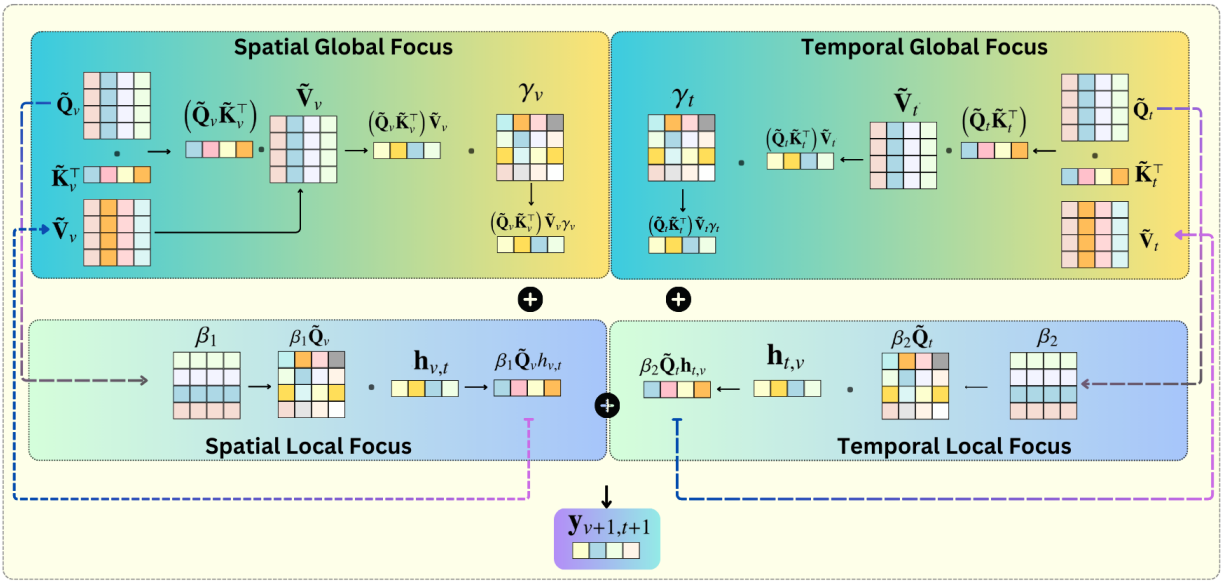


FIGURE 5.11. The STDF layer with dynamic spatial-temporal global-local attention complement

Figure 5.11 details the STDF layer, a core component of the STDAtt-Mambamodel, which integrates attention mechanisms within a ResNet-inspired framework to model spatial-temporal dependencies. The STDF layer is composed of four key components: spatial global focus, temporal global focus, spatial local focus, and temporal local focus. The spatial global focus computes global spatial attention using query ($\mathbf{Q}_v$), key ($\mathbf{K}_v^\top$), and value ($\mathbf{V}_v$) matrices, with the resulting spatial dependencies modulated by a dynamic scaling factor γ_v. Similarly, the temporal global focus computes temporal attention using analogous query ($\mathbf{Q}_t$), key ($\mathbf{K}_t^\top$), and value ($\mathbf{V}_t$) matrices, modulated by the factor γ_t to capture temporal dependencies. Complementing these global components, the spatial local focus refines the representation of spatial patterns by incorporating localized adjustments through a dynamic weighting factor β_1, which modulates the influence of spatial features. Likewise, the temporal local focus adjusts the temporal representation using the dynamic weighting factor β_2, enabling the model to account for localized

temporal variations. The outputs of these four components are integrated to produce a unified representation of spatial-temporal dependencies. This unified representation serves as the basis for accurate predictions of passenger demand at the next time step ($y_{v+1,t+1}$). The fusion of global and local perspectives ensures that the STDF layer captures both macro-level trends and spatial-temporal variations, significantly improving the robustness and interpretability in addressing multi-type passenger demand prediction in multimodal PT systems.

To further understand the structure of the STDAtt-Mambamodel, we now demonstrate how its state-space equations can be reformulated into a more intuitive spatial-temporal dual-path attention mechanism in Proposition 5.4.

Proposition 5.4. *The STDAtt-Mambamodel (Eqs.* (5.47)-(5.49)*) can be reformulated as a spatial-temporal dual-path attention mechanism.*

$$\mathbf{h}_{v,t} = \left(\mathbf{h}_{v-1,t} \cdot \bar{\mathbf{A}}_1 + \bar{\mathbf{B}}_1 \cdot \mathbf{x}_{v-1,t}\right) \cdot \mathbf{C}_1, \tag{5.47}$$

$$\mathbf{h}_{t,v} = \left(\mathbf{h}_{t-1,v} \cdot \bar{\mathbf{A}}_2 + \bar{\mathbf{B}}_2 \cdot \mathbf{x}_{t-1,v}\right) \cdot \mathbf{C}_2, \tag{5.48}$$

$$\mathbf{y}_{v,t} = \alpha \mathbf{h}_{v,t} + (1-\alpha)\mathbf{h}_{t,v}, \tag{5.49}$$

Proof. Based on the generalized discretization as detailed in Appendix A1. Starting from the continuous-time state-space model discretization (see Eq.(A.2)) over the temporal interval $[t_a, t_b]$ and spatial interval $[v_a, v_b]$ is given as follows:

$$\mathbf{h}_{t_b} = e^{\mathbf{A}(\Delta_a + \cdots + \Delta_{b-1})} \left(\mathbf{h}_{t_a} + \sum_{i=a}^{b-1} \mathbf{B}_i \mathbf{x}_i e^{\mathbf{A}(\Delta_{b-1} - \Delta_i)} \Delta_i\right), \tag{5.50}$$

where $\Delta_a + \cdots + \Delta_{b-1}$ represents the total interval.

The spatial and temporal state equations can be written as:

$$\mathbf{h}_{v_b,t_a} = e^{\mathbf{A}^1 \Delta_{v_a,t_a}} \left(\mathbf{h}_{v_a,t_a} + \mathbf{B}^1_{v_a,t_a} \mathbf{x}_{v_a,t_a} \Delta_a\right) \cdot \mathbf{C}_{v_b,t_a}, \tag{5.51}$$

$$\mathbf{h}_{t_b,v_a} = e^{\mathbf{A}^2 \Delta_{t_a,v_a}} \left(\mathbf{h}_{t_a,v_a} + \mathbf{B}^2_{t_a,v_a} \mathbf{x}_{t_a,v_a} \Delta_a\right) \cdot \mathbf{C}_{t_b,v_a}. \tag{5.52}$$

The combined output becomes:

$$\mathbf{y}_{t_b,v_b} = \alpha\left[e^{\mathbf{A}^1\Delta_{v_a,t}}\left(\mathbf{h}_{v_a,t} + \mathbf{B}^1_{v_a,t}\mathbf{x}_{v_a,t}\Delta_a\right)\cdot\mathbf{C}^1_{v_b,t}\right] + (1-\alpha)\left[e^{\mathbf{A}^2\Delta_{t_a,v}}\left(\mathbf{h}_{t_a,v} + \mathbf{B}^2_{t_a,v}\mathbf{x}_{t_a,v}\Delta_a\right)\cdot\mathbf{C}^2_{t_b,v}\right]. \tag{5.53}$$

For simplicity, let $[v_a, v_b]$ and $[t_a, t_b]$ reduce to $[v, v+1]$ and $[t, t+1]$, respectively. Then the updated equations align (Eqs. (5.54)-(5.58)) with components of an attention mechanism:

$$\tilde{\mathbf{Q}}_v = \mathbf{C}_{v,t+1}, \tilde{\mathbf{Q}}_t = \mathbf{C}_{t,v+1}, \tag{5.54}$$

$$\tilde{\mathbf{K}}^\top_v = \mathbf{B}^1_{v,t}, \tilde{\mathbf{K}}^\top_t = \mathbf{B}^2_{t,v}, \tag{5.55}$$

$$\beta_1 = e^{\mathbf{A}\Delta_{v,t}}, \beta_2 = e^{\mathbf{A}\Delta_{t,v}}, \tag{5.56}$$

$$\tilde{\mathbf{V}}_v = \mathbf{X}_{v,t+1}, \tilde{\mathbf{V}}_t = \mathbf{X}_{v+1,t}, \tag{5.57}$$

$$\gamma_v = \Delta_{v,t}, \gamma_t = \Delta_{t,v}, \tag{5.58}$$

where $\mathbf{C}$ denotes query matrices, $\mathbf{B}$ denotes key matrices, β_1 and β_2 represents weights, $\mathbf{X}$ denotes value matrices, and Δ represents the scaling factor for the time step.

The proposed STDAtt-Mambacan be reformulated as a spatial-temporal dual-path attention mechanism as follows:

$$\mathbf{h}_{v,t+1} = \beta_1\tilde{\mathbf{Q}}_v\mathbf{h}_{v,t} + \left(\tilde{\mathbf{Q}}_v\tilde{\mathbf{K}}^\top_v\right)\tilde{\mathbf{V}}_v\gamma_v, \tag{5.59}$$

$$\mathbf{h}_{t+1,v} = \beta_2\tilde{\mathbf{Q}}_t\mathbf{h}_{t,v} + \left(\tilde{\mathbf{Q}}_t\tilde{\mathbf{K}}^\top_t\right)\tilde{\mathbf{V}}_t\gamma_t, \tag{5.60}$$

$$\mathbf{y}_{v+1,t+1} = \alpha\Big(\beta_1\tilde{\mathbf{Q}}_v\mathbf{h}_{v,t} + \left(\tilde{\mathbf{Q}}_v\tilde{\mathbf{K}}^\top_v\right)\tilde{\mathbf{V}}_v\gamma_v\Big) + (1-\alpha)\Big(\beta_2\tilde{\mathbf{Q}}_t\mathbf{h}_{t,v} + \left(\tilde{\mathbf{Q}}_t\tilde{\mathbf{K}}^\top_t\right)\tilde{\mathbf{V}}_t\gamma_t\Big). \tag{5.61}$$

The terms $\beta_1\tilde{\mathbf{Q}}_v\mathbf{h}_{v,t}$ and $\beta_2\tilde{\mathbf{Q}}_t\mathbf{h}_{t,v}$ in Eq. (5.61) incorporate previous states $\mathbf{h}_{v,t}$ and $\mathbf{h}_{t,v}$, while the attention terms $\left(\tilde{\mathbf{Q}}_v\tilde{\mathbf{K}}^\top_v\right)\tilde{\mathbf{V}}_v\gamma_v$ and $\left(\tilde{\mathbf{Q}}_t\tilde{\mathbf{K}}^\top_t\right)\tilde{\mathbf{V}}_t\gamma_t$ emphasize spatial and temporal interactions weighted by γ_v and γ_t. This completes the proof. □

We next theoretically prove the complementarity of capturing global and local dependencies between the proposed STDAtt module and STDMamba module within the STDAtt-Mambamodel in Proposition 5.5.

Proposition 5.5. *STDMamba module and STDAtt module offer complementary attention mechanisms within STDAtt-Mamba: STDMamba can provide local attention by adjusting focus based on the current state of spatial and temporal features and integrating previous state information. In contrast, the global attention of STDAtt module captures long-range dependencies.*

Proof. Based on Proposition 5.4, the STDMamba can be reformulated as an attention with residual connections:

$$\mathbf{h}_{v,t+1} = \underbrace{\beta_1 \tilde{\mathbf{Q}}_v h_{v,t}}_{\text{residual}} + \underbrace{\left(\tilde{\mathbf{Q}}_v \tilde{\mathbf{K}}_v\right) \tilde{\mathbf{V}}_v \gamma_v}_{\text{weighted attention}}, \tag{5.62}$$

$$\mathbf{h}_{t+1,v} = \underbrace{\beta_2 \tilde{\mathbf{Q}}_t h_{t+1,v}}_{\text{residual}} + \underbrace{\left(\tilde{\mathbf{Q}}_t \tilde{\mathbf{K}}_t\right) \tilde{\mathbf{V}}_t \gamma_t}_{\text{weighted attention}}, \tag{5.63}$$

The proposed STDAtt captures the global dependencies through the sparse temporal and spatial attention mechanisms (see Eq.(5.9) and Eq.(5.15)). The respective implementation of both modules highlights their complementary nature, where STDMamba module captures local dependencies while STDAtt module captures global dependencies:

$$\mathcal{R}_{\text{local}}(i) = j : |j - i| \leq k \quad \text{(STDMamba)}, \tag{5.64}$$

$$\mathcal{R}_{\text{global}}(i) = 1, \ldots, n \quad \textit{(STDAtt)}. \tag{5.65}$$

The distinct implementation of both modules further lead to complementary processing:

$$\text{Local processing}: \sum_{j \in \mathcal{R}_{\text{local}}(i)} w_{ij} v_j + h_i, \tag{5.66}$$

$$\text{Global processing}: \sum_{j \in \mathcal{R}_{\text{global}}(i)} w_{ij} v_j. \tag{5.67}$$

Overall, leveraging its dual-path state-space architecture, STDMamba specializes in local attention, effectively preserving temporal-spatial dynamics through residual connections. Meanwhile, STDAtt employs sparse attention mechanisms to facilitate global attention, capturing long-range dependencies. These two orthogonal modules are complementary, enabling STDAtt-Mambato address both local and global dependencies comprehensively. □

Corollary 5.6 (Interpretability analysis). *The dual-path formulation presented in Proposition 5.4 provides a transparent method for interpreting model predictions by analyzing their contributions through separate spatial and temporal channels. As shown in Figure 5.11, each prediction can be decomposed into four essential components: 1) spatial global focus capturing broad, system-wide spatial dependencies, 2) temporal global focus highlighting significant temporal patterns, 3) spatial local focus emphasizing neighborhood-specific effects, and 4) temporal local focus capturing recent temporal trends.*

5.6 Experimental Studies

In this section, we introduce experimental setup and evaluation metrics (Section 5.6.1), baseline models (Section 5.6.2), comparative model performance evaluation (Section 5.6.3), ablation study (Section 5.6.4), demand prediction results analysis (Section 5.6.5), reliability and equality analysis (Section 5.6.5.2), and impact of the weather and public holidays (Section 5.6.7).

5.6.1 Experimental Setup and Evaluation Metrics

The experiments are conducted on an H20-NVLink system with 96GB of memory, utilizing a sequence-to-sequence framework with 12 timesteps for both input and output sequences. For a

fair comparison, we kept the data splitting consistent with the baseline models [Liu et al., 2023b, Jiang et al., 2023], dividing the dataset into training, validation, and test sets in proportions of 70%, 10%, and 20%, respectively. The parameter selection of this study followed a systematic grid search methodology combined with ablation studies. For hyperparameter optimization, we evaluated learning rates {0.0005, 0.001, 0.002}, weight decay values {0.001, 0.0015, 0.002}, and batch sizes {8, 16, 32}. Based on validation performance, the model training ultimately leveraged the Adam optimizer with an optimal learning rate of 0.001 and weight decay of 0.0015, with a batch size of 16 offering the best balance between computational efficiency and model performance. For architectural parameters, ablation studies guided the selection of the optimal number of layers for both STDF and sparse attention (1), residual blocks (2), and attention heads (4). The model architecture consisted of 9 input and output channels with a hidden dimension of 128, selected after experimenting with dimensions {64, 128, 256}. The temporal module featured an input embedding dimension of 24 and a feed-forward dimension of 256. A multi-step learning rate scheduler is employed based on observed convergence patterns, reducing the learning rate by a factor of 0.1 at epochs 25, 45, and 65. The training is performed for up to 100 epochs, incorporating early stopping with a patience threshold of 30 epochs to prevent overfitting while ensuring sufficient training time. A dropout rate of 0.1 is maintained throughout the network, determined through validation performance to effectively prevent overfitting without compromising the model's representational capacity. The implementation details of baseline models can be found in Appendix A2.

We employ three evaluation metrics, i.e., MAE, MAPE, and RMSE, to evaluate model performance in travel demand prediction tasks in multimodal PT systems. For each travel mode $\forall m \in \mathcal{M}$, let N^m denote the number of samples N^m, $\hat{y}_i^m$ denote the predicted demand values, and y_i^m denote the corresponding ground truth values for sample i.

The Mean Absolute Error (MAE) serves as the primary metric for measuring the absolute prediction accuracy and provides an intuitive measure of prediction accuracy in the same units as the original data:

$$\text{MAE} = \frac{1}{N^m} \sum_{i=1}^{N^m} |\hat{y}_i^m - y_i^m|, \quad \forall m \in \mathcal{M}. \tag{5.68}$$

To account for the scale-dependency in the predictions, we utilize the Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{N^m} \sum_{i=1}^{N^m} \left| \frac{\hat{y}_i^m - y_i^m}{y_i^m} \right|, \quad \forall m \in \mathcal{M}. \tag{5.69}$$

MAPE offers a scale-invariant perspective on model performance, expressing errors as percentages of the actual values, which is particularly valuable when comparing performance across different datasets or time scales.

Additionally, we employ the Root Mean Square Error (RMSE) to capture the sensitivity to large prediction errors:

$$\text{RMSE} = \sqrt{\frac{1}{N^m} \sum_{i=1}^{N^m} (\hat{y}_i^m - y_i^m)^2}, \quad \forall m \in \mathcal{M}. \tag{5.70}$$

5.6.2 Baseline models

We comprehensively compare the proposed STDAtt-Mambamodel with 19 baseline models for passenger travel demand prediction, including traditional time series models, traditional deep learning, graph-based, Transformer-based, and Mamba-based models.

Traditional time series baseline models. ARIMA[Box and Jenkins, 1970] is a fundamental baseline that captures temporal dependencies without accounting for spatial relationships. Incorporating it shows how much improvement comes from spatiotemporal modeling. SARIMA [Box and Jenkins, 1976] extends ARIMA by handling seasonal patterns, which is crucial for traffic data with daily/weekly cycles. VAR (Vector Autoregression) [Sims, 1980] can capture relationships between multiple time series, important for comparing with multivariate models. SVR (Support Vector Regression) [Drucker et al., 1997b] is a non-linear regression technique that often performs well on time series with complex patterns.

Traditional deep learning baseline models. The Historical Index (HI) [Cui et al., 2021] serves as a fundamental benchmark for time series prediction, reflecting standard industry

practices. While simple, this model provides a baseline against which more complex models can be assessed.

Graph-based baseline models. Graph WaveNet (GWNet) [Wu et al., 2020a] pioneers the automatic extraction of uni-directed spatial dependencies, while the Diffusion Convolutional Recurrent Neural Network (DCRNN) [Li et al., 2018] introduces sophisticated diffusion processes for modeling spatial-temporal correlations. The Adaptive Graph Convolutional Recurrent Network (AGCRN) [BAI et al., 2020] further advances the field through its node-specific pattern recognition and dynamic inter-dependencies. Spatio-Temporal Graph Convolutional Networks (STGCN) [Yu et al., 2018a] seamlessly integrate graph convolutions with gated temporal convolutions, establishing a robust framework for demand prediction. Building upon these foundations, more sophisticated graph-based models have emerged. The Graph Time Series (GTS) [Shang et al., 2021] model distinguishes itself by simultaneously learning graph structure and temporal dynamics. The Multi-variate Time Series Graph Neural Network (MTGNN) [Wu et al., 2020a] excels in automatically extracting variable relationships, while the Graph Multi-Attention Network (GMAN) [Zheng et al., 2020a] leverages advanced attention mechanisms to capture dynamic sensor correlations. Multi-task spatiotemporal network (MT-STNet) [Zou et al., 2024] is a multi-task spatiotemporal network model that integrates physical structure information and mitigates multi-step error propagation through a generative inference system.

Transformer-based baseline models. PDFormer [Jiang et al., 2023] addresses the crucial aspects of dynamic spatial dependencies and propagation delays, while STAEformer [Liu et al., 2023b] introduces innovative spatio-temporal adaptive embeddings. STNorm [Deng et al., 2021] contributes to the field through its dual-normalization modules for component refinement, and STID [Shao et al., 2022] advances the state-of-the-art (SOTA) with its sophisticated treatment of spatial-temporal identity disambiguation.

Mamba-based baseline model. ST-MambaSync [Shao et al., 2024a], a cutting-edge state space model that represents the current SOTA in travel demand forecasting. This model uniquely

integrates Transformer architectures with vanilla state space models, offering a novel model to capturing both spatial and temporal dependencies.

Although other spatiotemporal modeling frameworks exist (e.g., various ST-GNN variants and hybrid architectures), we focus on mainstream models. We select well-established, representative baselines such as STGCN, DCRNN, AGCRN, GWNet, and MTGNN to cover a broad spectrum of ST-GNN models, ranging from spectral to spatial convolutions, from recurrent units to wavelet-based filters, and thus provide a solid basis to validate the performance of the proposed STDAtt-Mambamodel.

5.6.3 Comparison of Performance Between STDAtt-Mamba and Baselines

Table 5.4 presents the results of a comprehensive performance comparison of STDAtt-Mambaagainst 19 baseline models across three travel modes (Ferry, Bus, and Rail) using three evaluation metrics (MAE, RMSE, and MAPE).

Across all travel modes, STDAtt-Mambaconsistently achieves superior performance compared to 19 baseline models. For Ferry, STDAtt-Mambadelivers the lowest MAE (2.101), RMSE (4.128), and MAPE (62.038%), outperforming the next best model, ST-MambaSync, by 5.3% in MAE and 0.9% in MAPE. This pattern continues for Bus, where STDAtt-Mambaachieves MAE of 2.876, RMSE of 5.323, and MAPE of 60.326%, representing improvements of 7.9% in MAE and 7.5% in MAPE over ST-MambaSync. For Rail, STDAtt-Mambamaintains its superiority with the lowest MAE (2.324) and MAPE (63.781%), demonstrating an 8.3% reduction in MAE and 5.4% reduction in MAPE compared to ST-MambaSync.

For the performance of Graph-based models, the results reveal a clear performance gap between graph-based models (DCRNN, STGCN, GWNet, AGCRN, GTS, MTGNN) and the proposed STDAtt-Mamba. For example, for Ferry, the best-performing graph-based model MTGNN achieves an MAE of 3.254, which is 35.4% higher than STDAtt-Mamba. This significant performance difference can be attributed to the inherent limitations of graph-based models in capturing dynamic spatiotemporal dependencies. While GNNs excel at modeling static graph

TABLE 5.4. Comparison of model performance between STDAtt-Mambaand 19 baselines

Models	Ferry (Missing 3.41‰)			Bus (Missing 1.25‰)			Rail (Missing 4.49‰)		
	MAE	RMSE	MAPE(‰)	MAE	RMSE	MAPE(‰)	MAE	RMSE	MAPE(‰)
Traditional time series models									
ARIMA	4.276	6.312	70.125	4.358	7.023	72.456	4.215	7.689	71.345
SARIMA	4.192	6.287	69.872	4.301	6.978	71.893	4.153	7.621	70.987
VAR	3.982	6.124	69.235	4.267	6.912	71.452	4.087	7.543	70.654
SVR	3.875	5.943	68.756	4.105	6.823	70.987	3.942	7.438	70.213
Deep learning models									
HI	3.892	5.974	68.778	4.123	6.891	71.234	3.891	7.456	70.123
GWNet	3.421	5.463	66.891	3.876	6.234	68.981	3.234	7.123	69.234
DCRNN	3.389	5.412	66.234	3.789	6.123	68.234	3.123	7.012	68.901
AGCRN	3.312	5.389	65.981	3.654	6.089	67.891	3.089	6.989	68.567
STGCN	3.298	5.367	65.789	3.598	5.987	67.234	2.987	6.967	68.234
GTS	3.276	5.345	65.567	3.534	5.923	66.981	2.923	6.945	68.012
MTGNN	3.254	5.334	65.456	3.487	5.889	66.789	2.878	6.923	67.891
GMAN	3.231	5.312	65.234	3.423	5.867	66.567	2.812	6.901	67.678
PDFormer	2.537	4.674	65.271	3.321	5.789	66.234	2.789	6.889	67.456
STNorm	2.523	4.612	64.123	3.287	5.734	65.981	2.723	6.867	67.234
STID	2.518	4.589	63.567	3.234	5.689	65.789	2.701	6.845	67.123
STAEformer	2.512	4.563	63.125	3.198	5.645	65.567	2.691	6.935	67.890
ST-MambaSync	2.219	4.153	62.612	3.123	5.534	65.234	2.535	6.782	67.419
Multi-task models									
MTGNN	3.254	5.334	65.456	3.487	5.889	66.789	2.878	6.923	67.891
MT-STNet	3.114	5.032	62.651	3.391	5.622	64.193	2.812	6.52	65.051
STDAtt-Mamba	**2.101**	**4.128**	**62.038**	**2.876**	**5.323**	**60.326**	**2.324**	**6.524**	**63.781**

structures, they struggle to adapt to the heterogeneous and highly dynamic demand patterns across different passenger groups. The message-passing mechanism in GNNs primarily relies on local neighborhood information, which limits their ability to capture long-range dependencies crucial for modeling diverse mobility patterns across passenger types. This limitation is particularly evident in the Ferry dataset, where complex maritime routes and distinct passenger behaviors require more flexible modeling methods.

In regard to Transformer-based models, such as STAEformer (MAE: 2.512 for Ferry) and STID (MAE: 2.518 for Ferry), they demonstrate better performance than graph-based models but still fall significantly short compared to STDAtt-Mamba. While these attention-based

architectures offer improvements in modeling global dependencies, they face challenges in efficiently processing long sequential data and capturing the recurrent nature of travel patterns. STAEformer, despite its sophisticated attention mechanisms, shows 16.4% higher MAE than STDAtt-Mambafor Ferry. The key limitation of vanilla attention mechanisms is their quadratic complexity with sequence length and their tendency to distribute attention too broadly when modeling fine-grained local patterns, which are crucial for accurate passenger demand prediction.

The analysis of multi-task learning models reveals particularly interesting insights. For example, MT-STNet, a multi-task spatiotemporal network designed for traffic flow prediction, achieves MAE values of 3.114, 3.391, and 2.812 for Ferry, Bus, and Rail, respectively. Despite its multi-task architecture, MT-STNet underperforms STDAtt-Mambaby 32.5%, 15.2%, and 17.4% in MAE across the three travel modes. Similarly, MTGNN, which incorporates multi-task learning through its graph learning module, shows MAE values of 3.254, 3.487, and 2.878, lagging behind STDAtt-Mamba by 35.4%, 17.5%, and 19.2%, respectively. These results demonstrate that merely incorporating multi-task learning frameworks is insufficient without effectively modeling the complex spatiotemporal dependencies unique to each passenger group. The superior performance of STDAtt-Mambacan be attributed to its innovative combination of state space models with attention mechanisms, which enables more effective knowledge sharing across tasks while preserving group-specific patterns.

The strong performance of ST-MambaSync (the second-best model) highlights the effectiveness of state space models in passenger demand prediction. However, the consistent improvement of STDAtt-Mambaover ST-MambaSync across all metrics and travel modes demonstrates the crucial importance of the proposed STDF layer in enhancing prediction accuracy. By integrating selective attention mechanisms with state space models, STDAtt-Mambaachieves a more balanced approach to modeling both global dependencies and local patterns, capturing the complex interrelationships between different passenger groups more effectively. The results also indicate the robustness of STDAtt-Mambain handling datasets with varying levels of missing values. Despite the Rail dataset exhibiting the highest missing rate (4.49%), STDAtt-Mambastill outperforms all competing models, achieving the lowest MAE (2.324) and MAPE (63.781%). This consistent performance across Ferry (3.41% missing), Bus (1.25% missing), and Rail

demonstrates that STDAtt-Mamba is accurate and resilient to data incompleteness. The ability to maintain superior prediction quality under realistic, imperfect data conditions further validates the practical reliability of STDAtt-Mambafor deployment in real-world PT systems.

The evaluation results in Table 5.4 demonstrate that the STDAtt-Mambaoffers significant and consistent performance improvements over 19 baseline models across all travel modes. Its innovative architecture successfully addresses the limitations of both graph-based and vanilla attention-based models, while effectively leveraging the multi-task learning paradigm to capture the diverse mobility patterns of multiple passenger groups.

5.6.4 Ablation Study

Table 5.5 compares the performance of the full STDAtt-Mambamodel against several ablated versions on ferry, bus, and rail datasets, focusing on two crucial components: (i) the Spatio-Temporal Dynamic Fusion (STDF) layer and (ii) the STDAtt module. Each ablated variant removes or replaces a key element of STDAtt-Mambato highlight the importance of that component in accurately capturing multi-type passenger demand.

Removing the STDF layer, i.e., *STDAtt-Mamba (without STDF)*, introduces the most substantial performance degradation for all modes. On the ferry dataset, for instance, the MAE increases from 2.101 to 2.512, RMSE from 4.128 to 4.563, and MAPE from 62.038% to 63.125%. Similar increases appear in the bus and rail modes, indicating the crucial role of STDF in modeling complex spatial-temporal interactions. By discarding the dual-path state-space framework, the model loses its localized handling of spatial-temporal dependencies, thus producing higher errors in passenger demand forecasting. Removing the STDAtt module, i.e., *STDAtt-Mamba (without STDAtt)*, also adversely affects performance, though this drop is somewhat less severe compared to omitting STDF. For ferry data, the MAE increases from 2.101 to 2.251, RMSE from 4.128 to 4.351, and MAPE from 62.038% to 62.034%. Similar trends emerge for bus and rail. These findings confirm that the dynamic, sparse attention mechanism effectively pinpoints the most influential stations and time intervals. Without STDAtt, the model loses a key global attention path and consequently experiences reduced predictive accuracy. Replacing the STDAtt module

with an alternative attention mechanism, i.e., *STDAtt-Mamba (with Vanilla Attention)*, yields moderate performance yet remains below that of the complete STDAtt-Mamba. For example, on the ferry dataset, MAE, RMSE, and MAPE increase to 2.217, 4.216, and 62.264%, respectively. Likewise, substituting Mamba's state-space design with another meodel (*STDAtt-Mamba (with Vanilla Mamba)*) results in similarly higher error metrics, including a ferry MAE of 2.219 and consistent degradations across bus and rail. These experiments show that while alternative schemes capture some aspects of spatial-temporal correlations, they cannot fully replicate the synergy arising from STDAtt and STDF working together.

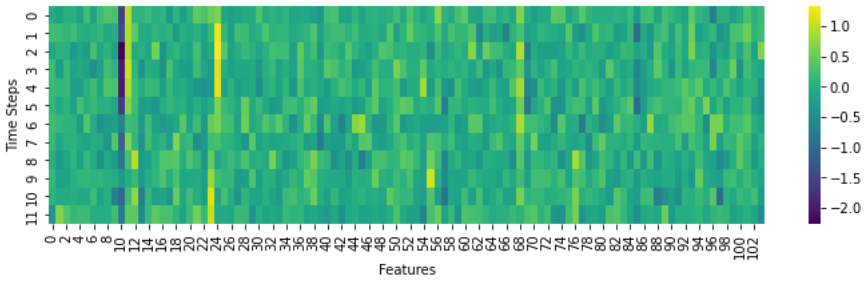
The proposed STDAtt-Mambamodel consistently achieves the lowest MAE, RMSE, and MAPE across ferry, bus, and rail. Its strong results confirm that both the STDF layer and the STDAtt module are indispensable for modeling multi-type passenger demand robustly. By fusing local, state-space-based spatial-temporal dynamics (STDF) with a sparse global attention mechanism (STDAtt), the model effectively captures intricate variations in travel behaviors and delivers accurate, scalable demand predictions across multiple socio-demographic groups.

TABLE 5.5. Ablation study of different components in STDAtt-Mambaacross multiple travel modes

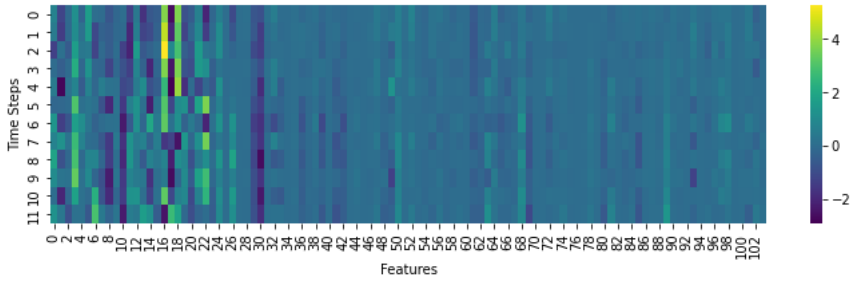
Models	**Ferry**			**Bus**			**Rail**		
MAE	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**	
STDAtt-Mamba (without STDF)	2.512	4.563	63.125	3.287	5.734	65.981	2.723	6.867	67.234
STDAtt-Mamba (without STDAtt)	2.251	4.351	62.034	3.242	5.689	65.789	2.701	6.845	67.123
STDAtt-Mamba (with Vanilla attention)	2.217	4.216	62.264	3.123	5.534	65.234	2.535	6.782	67.419
STDAtt-Mamba (with Vanilla Mamba)	2.219	4.153	62.612	3.095	5.467	65.123	2.481	6.673	65.891
STDAtt-Mamba	**2.101**	**4.128**	**62.038**	**2.876**	**5.323**	**60.326**	**2.324**	**6.524**	**63.781**

Figure 5.12 visualizes the evolution of feature transformations for Batch 5, Node 20, based on ferry travel records, as they pass through various components of the proposed STDAtt-Mambamodel. Figure 5.12 highlights the distinct contributions of each model component in improving feature representation and capturing crucial patterns for multi-type passenger demand prediction.

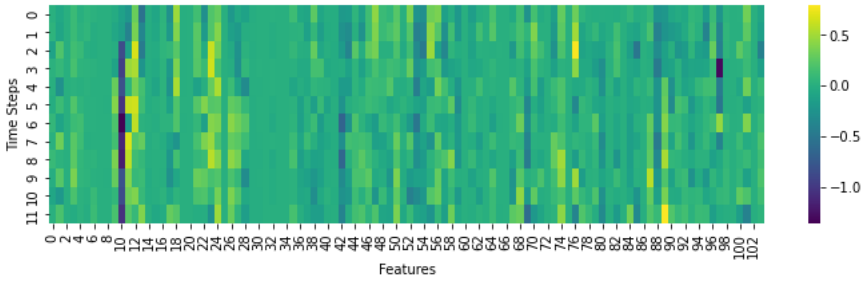
Figure 5.12a displays the raw input features across 12 time steps for Batch 5, Node 20. The intensity of the colors highlights the initial distribution and variability of feature values. Figure



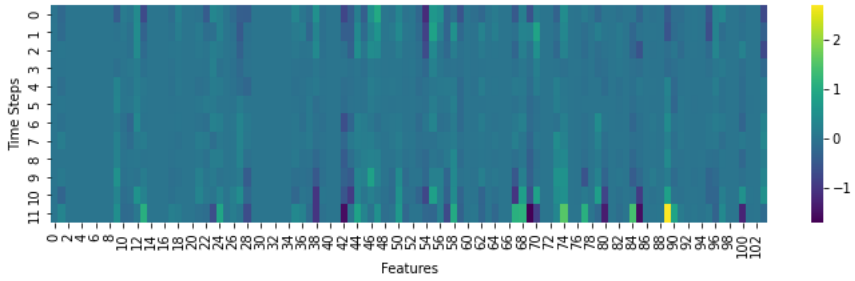
(A) Heatmap of the input data for Batch 5, Node 20, displaying the feature values across 12 time steps. The variations represent the raw data before model processing, highlighting the initial distribution of feature values.



(B) Heatmap after implementing the sparse temporal attention for Batch 5, Node 20. The sparse temporal attention processes the data, resulting in smoother patterns and reduced variance in some features.



(C) Heatmap after implementing the sparse spatial attention for Batch 5, Node 20. The sparse spatial attention emphasizes differences between features, highlighting key spatial relationships and variations in the features.



(D) Heatmap after implementing the STDF layer for Batch 5, Node 20. The STDF layer highlights certain travel patterns in the dataset and features that are crucial for demand prediction.

FIGURE 5.12. Heatmap for ablation study

5.12a showcases high-dimensional heterogeneity, reflecting the diverse nature of multimodal travel records data. This stage serves as the baseline, indicating the need for more sophisticated processing to effectively capture temporal and spatial dependencies. Figure 5.12b shows the

output of the features processed by the sparse temporal attention mechanism. This operation smooths out noise while preserving temporal correlations across the 12-time steps. The resulting heatmap exhibits reduced variance and a more uniform pattern, particularly in time-sensitive features. The temporal attention mechanism emphasizes temporal dependencies, ensuring that the model identifies sequential patterns essential for demand prediction. Figure 5.12c shows the output of the features processed by the sparse spatial attention layer, which reveals variations that accentuate spatial distinctions and capture the interdependencies between features across different nodes. Figure 5.12d visualizes the output after the implementation of the STDF layer, and the highlighted regions indicate travel patterns and crucial features for demand prediction.

Observation 1. Figures 5.12a-5.12d further conduct the interpretability analysis by showing how features evolve across each module (Proposition 5.4). For example, the heat maps in Figure 5.12b demonstrate how sparse temporal attention effectively captures regular commuting patterns for adult passengers while smoothing out irregular fluctuations for senior passengers. Figure 5.12c illustrates how sparse spatial attention emphasizes the spatial correlations between different travel modes. These visualizations provide actionable insights for transit authorities, allowing them to identify the passenger groups that are most influenced by temporal factors, such as school passengers and those that are more affected by spatial factors, such as tertiary students.

5.6.4.1 Computational Efficiency Analysis

Table 5.6 provides a detailed comparison of model performance based on the MAE as well as computational efficiency metrics, including floating point operations per second (FLOPS), inference time, training time, and GPU memory usage. To assess the trade-off between prediction accuracy and computational demands, we computed a Pareto Efficiency Score (PES) for each model, defined as the sum of the ranks for MAE, FLOPS, and inference time. In this scoring system, lower scores indicate a more favorable overall balance. The results show that STDAtt-Mambaachieves the best trade-off, with the lowest PES of 7. In comparison, ST-MambaSync, STNorm, and STAEFormer have PES values of 9, 11, and 14, respectively. This multi-dimensional ranking highlights the superior equilibrium achieved by STDAtt-Mambabetween predictive accuracy and computational efficiency. Although the baseline model (HI) offers

the lowest computational requirements, its prediction performance is the poorest, with an MAE of 3.892 and a correspondingly high PES of 15. These findings indicate that achieving computational efficiency alone is insufficient for effective demand prediction.

Examining specific performance metrics, STDAtt-Mamba achieves the lowest MAE (2.101), representing a 46% improvement over the baseline while requiring only moderate computational resources (3.70M FLOPS, 2.47s inference time). In contrast, transformer-based models like STAEFormer (MAE: 2.512) and PDFormer (MAE: 2.537) require higher FLOPS (4.24M and 4.19M respectively) and longer inference times (3.03s and 2.99s), yet deliver 16-17% lower accuracy than STDAtt-Mamba. Comparing STDAtt-Mamba with its closest competitor, ST-MambaSync, further demonstrates STDAtt-Mamba's advantages: 5.3% better accuracy while requiring 6.8% less inference time and 9.2% less training time. This superior performance can be attributed to STDAtt-Mamba's sparse attention mechanism, which efficiently captures complex spatial-temporal dependencies while reducing computational overhead. These results validate that the STDAtt-Mambaachieves SOTA prediction accuracy while maintaining competitive computational efficiency, making it suitable for real-world multimodal PT systems.

TABLE 5.6. Computational cost comparison of model performance and computational efficiency.

Model	**MAE**	**FLOPS (M)**	**Inference (s)**	**Training (s)**	**GPU Mem (GB)**	**PES ↓**
HI	3.892	**0.44**	**0.44**	**1.75**	**0.2**	15
GWNet	3.421	2.73	1.37	16.41	0.9	18
DCRNN	3.389	2.93	1.55	18.28	1.1	19
AGCRN	3.312	3.22	1.75	20.25	1.3	20
STGCN	3.298	2.30	1.11	13.82	0.7	16
GTS	3.276	3.05	1.66	19.39	1.2	20
MTGNN	3.254	3.05	1.67	19.43	1.2	20
GMAN	3.231	3.71	2.04	27.82	1.5	23
PDFormer	2.537	4.19	2.99	34.90	2.0	13
STNorm	2.523	2.00	1.00	11.98	0.6	11
STID	2.518	2.20	1.10	13.99	0.7	12
STAEFormer	2.512	4.24	3.03	36.00	1.7	14
ST-MambaSync	2.219	3.70	2.65	29.00	1.5	9
STDAtt-Mamba	**2.101**	3.70	2.47	26.33	1.5	**7**

Note. PES = Pareto Efficiency Score (rank(MAE) + rank(FLOPS) + rank(Inference Time)); lower is better.

Figure 5.13 illustrates the trade-off between model accuracy and computational cost by comparing the top three models, i.e., ST-MambaSync, STAEFormer, and STDAtt-Mamba. In this figure, the Mean Absolute Error (MAE) is plotted against the FLOPs, while the bubble sizes indicate the total training and inference times. It is evident that although STAEFormer exhibits higher FLOPs and longer overall running times, its accuracy does not match that of STDAtt-Mamba. Similarly, ST-MambaSync achieves competitive accuracy but suffers from extended training and inference durations. In contrast, STDAtt-Mambareduces the MAE to 2.101 while requiring only 3.70M FLOPs, 2.47 seconds for inference, and 26.33 seconds for training. These results demonstrate that STDAtt-Mambaachieves SOTA accuracy with manageable computational costs.

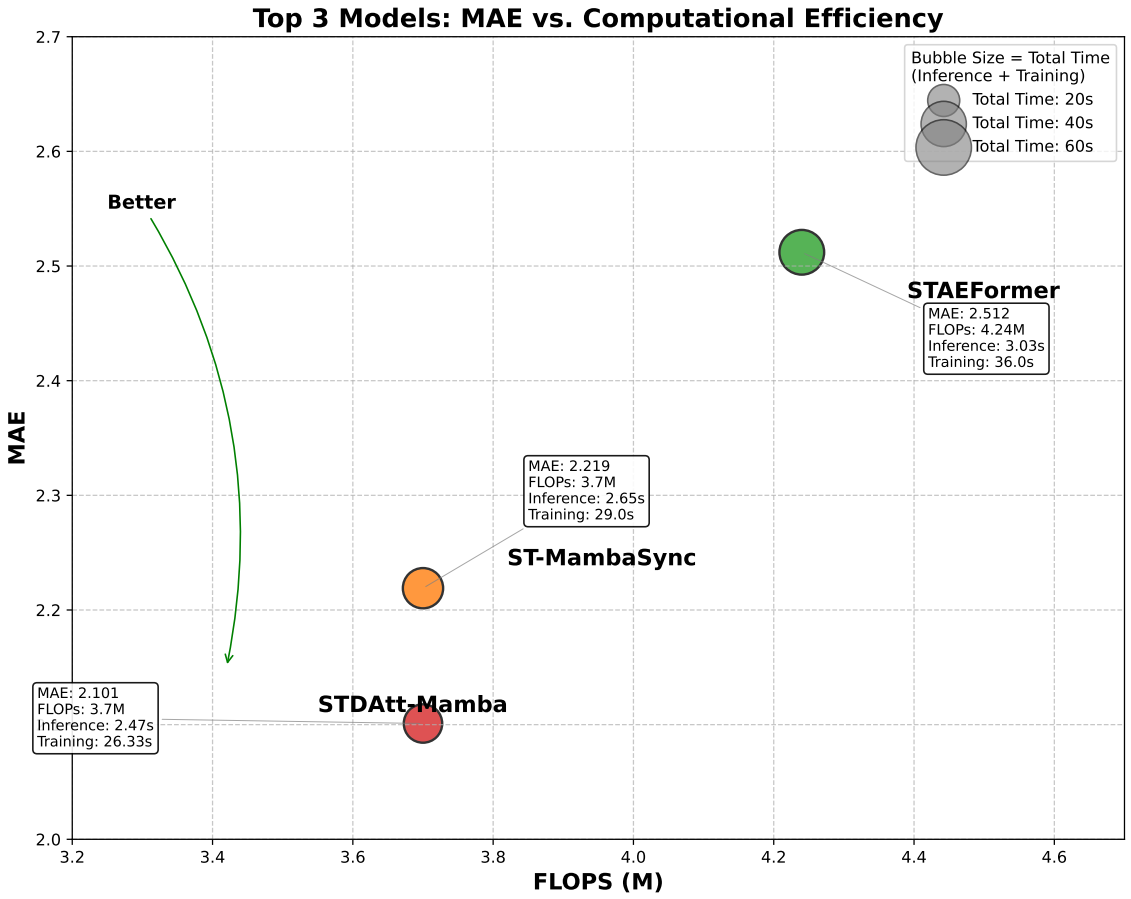


FIGURE 5.13. The trade-off analysis on Top 3 models (ST-MambaSync, STAE-Former, and STDAtt-Mamba)

5.6.4.2 Comparison of Single-task and Multi-task Learning Results

To evaluate the effectiveness of multi-task learning over single-task learning, we conduct experiments across three travel modes (bus, ferry, and rail) using the STDAtt-Mambamodel. Prediction performance for nine passenger groups is assessed using three metrics, i.e., MAE, MSE, and MAPE, with results averaged over 10 independent runs (Table 5.7

TABLE 5.7. Comparison of demand prediction performance of 9 passenger groups across travel modes using STDAtt-Mamba (10 runs, mean ± SD)

Passenger Type	Bus			Ferry			Rail		
	MAE	MSE	MAPE(%)	MAE	MSE	MAPE(%)	MAE	MSE	MAPE(%)
Adults	3.124±0.112	17.549±0.234	62.234±2.123%	2.234±0.087	12.987±0.156	63.345±1.634%	2.498±0.089	6.251±0.124	64.102±1.842%
Seniors	2.987±0.098	17.223±0.198	61.456±1.967%	2.187±0.082	12.756±0.143	62.234±1.567%	2.317±0.094	6.347±0.132	63.892±1.967%
Pensioners	2.912±0.089	16.987±0.187	60.789±1.834%	2.145±0.078	12.523±0.134	61.567±1.445%	2.256±0.087	6.192±0.126	63.217±1.756%
Tertiary	3.045±0.103	17.345±0.221	61.234±1.945%	2.198±0.084	12.789±0.149	62.789±1.578%	2.310±0.091	6.124±0.129	63.451±1.834%
Children	2.978±0.094	17.156±0.203	60.567±1.823%	2.167±0.081	12.678±0.141	62.123±1.523%	2.289±0.088	6.231±0.127	63.842±1.798%
School Passengers	2.954±0.091	17.089±0.195	60.234±1.789%	2.178±0.083	12.698±0.144	61.892±1.534%	2.312±0.093	6.087±0.134	62.345±1.723%
Job Seekers	3.198±0.124	18.234±0.267	63.456±2.234%	2.298±0.095	13.234±0.167	64.234±1.723%	2.412±0.102	6.457±0.147	65.230±2.134%
Youth	3.234±0.128	18.567±0.281	64.123±2.345%	2.323±0.099	13.456±0.178	65.123±1.834%	2.514±0.118	6.631±0.162	66.124±2.345%
Gold Repat	3.267±0.132	18.789±0.294	64.789±2.456%	2.334±0.101	13.523±0.182	65.456±1.867%	2.507±0.124	6.642±0.168	67.032±2.456%
All Types (Single)	3.078±0.108	17.660±0.231	62.098±2.057%	2.229±0.086	12.960±0.155	63.196±1.634%	2.402±0.098	6.329±0.139	64.582±1.984%
All-Multiple	2.876±0.081	17.143±0.189	60.326±1.234%	2.101±0.063	12.689±0.126	62.038±1.087%	2.324±0.073	6.524±0.109	63.781±1.345%

Across all travel modes, the multi-task STDAtt-Mamba consistently outperformed the single-task counterpart, demonstrating superior predictive performance in terms of MAE, MSE, and MAPE. For instance, in the bus mode (Table 5.7), the average MAE is reduced from 3.078 to 2.876 (a 6.56% reduction), and the average MAPE improved from 62.098% to 60.326%. MSE also decreased from 17.660 to 17.143. These improvements are particularly notable given the heterogeneity across the nine passenger groups, with consistent gains observed in almost all categories, especially among high-volume groups like Adults and School Passengers. Similarly, in the ferry mode (Table 5.7), the multi-task model achieved a reduction in MAE from 2.229 to 2.101, with corresponding improvements in MSE (−0.86) and MAPE (from 63.196% to 62.038%). The enhancements are robust across diverse passenger groups, including Adults, Children, and Gold Repatriates, suggesting that the multi-task model generalizes well even to low-sample groups. In the rail mode (Table 5.7), although the improvements are more modest, the multi-task model still outperformed the single-task version in both MAE (2.402 $\rightarrow$ 2.324, a 3.25% reduction) and MAPE (64.582% $\rightarrow$ 63.781%). The MSE slightly increased from 6.329 to 6.524. However, this increase is relatively minor, and the reduction in MAE and MAPE indicates overall better prediction accuracy and reliability. Notably, passenger groups with fewer samples,

such as Youth and Job Seekers, still benefited from multi-task learning, highlighting its ability to share statistical strength across related tasks.

To verify the statistical significance of these improvements, paired t-tests are conducted using the results from 10 experimental runs (Table 5.8). The multi-task approach consistently yielded statistically significant improvements in MAE and MAPE across all travel modes ($p < 0.05$), with the majority of results achieving high levels of significance ($p < 0.001$). However, a slight increase in MSE is observed for rail under multi-task conditions, and this difference is statistically significant ($p = 0.0076$), but the effect size is modest (Cohen's $d = -0.63$), suggesting only a minor trade-off. Effect size analyses provided further evidence for the practical relevance of these improvements, revealing large to very large Cohen's d values in most scenarios, such as $d = 2.11$ for MAE reduction in bus mode and $d = 1.92$ for MSE improvement in ferry mode. These findings demonstrate that the multi-task learning approach significantly improves predictive accuracy, delivering statistically and practically meaningful benefits for multimodal public transport travel demand forecasting across diverse passenger groups.

TABLE 5.8. Statistical significance test results comparing Single-task vs Multi-task STDAtt-Mamba (10 runs)

Mode	Metric	Single-task	Multi-task	Improvement	t-statistic	p-value	Cohen's d
Rail	MAE	2.402 ± 0.098	2.324 ± 0.073	3.25%	4.87	0.0009**	0.90
	MSE	6.329 ± 0.139	6.524 ± 0.109	-3.08%	-3.42	0.0076**	-0.63
	MAPE	64.582 ± 1.984	63.781 ± 1.345	1.24%	2.89	0.0179*	0.47
Bus	MAE	3.078 ± 0.108	2.876 ± 0.081	6.56%	7.23	<0.0001***	2.11
	MSE	17.660 ± 0.231	17.143 ± 0.189	2.93%	5.89	0.0002***	2.45
	MAPE	62.098 ± 2.057	60.326 ± 1.234	2.85%	4.12	0.0026**	1.04
Ferry	MAE	2.229 ± 0.086	2.101 ± 0.063	5.74%	5.67	0.0003***	1.70
	MSE	12.960 ± 0.155	12.689 ± 0.126	2.09%	4.38	0.0018**	1.92
	MAPE	63.196 ± 1.634	62.038 ± 1.087	1.83%	3.76	0.0044**	0.84

Notes. *p < 0.05, **p < 0.01, ***p < 0.001

5.6.5 Passenger Demand Prediction Results Analysis

5.6.5.1 Temporal Prediction Results Analysis

We compare the predicted passenger demand of the proposed STDAtt-Mambamodel with baseline models, i.e., the STAEFormer and ST-MambaSync, against the true passenger demand values over different time spans such as 12 hours, 24 hours, and 7 days. The evaluation is conducted across multiple passenger groups based on the rail travel records of 9 passenger types, i.e., Adult, Child, Job Seeker, Pensioner, School Passenger, Senior, Tertiary, Youth, and Gold Repat.

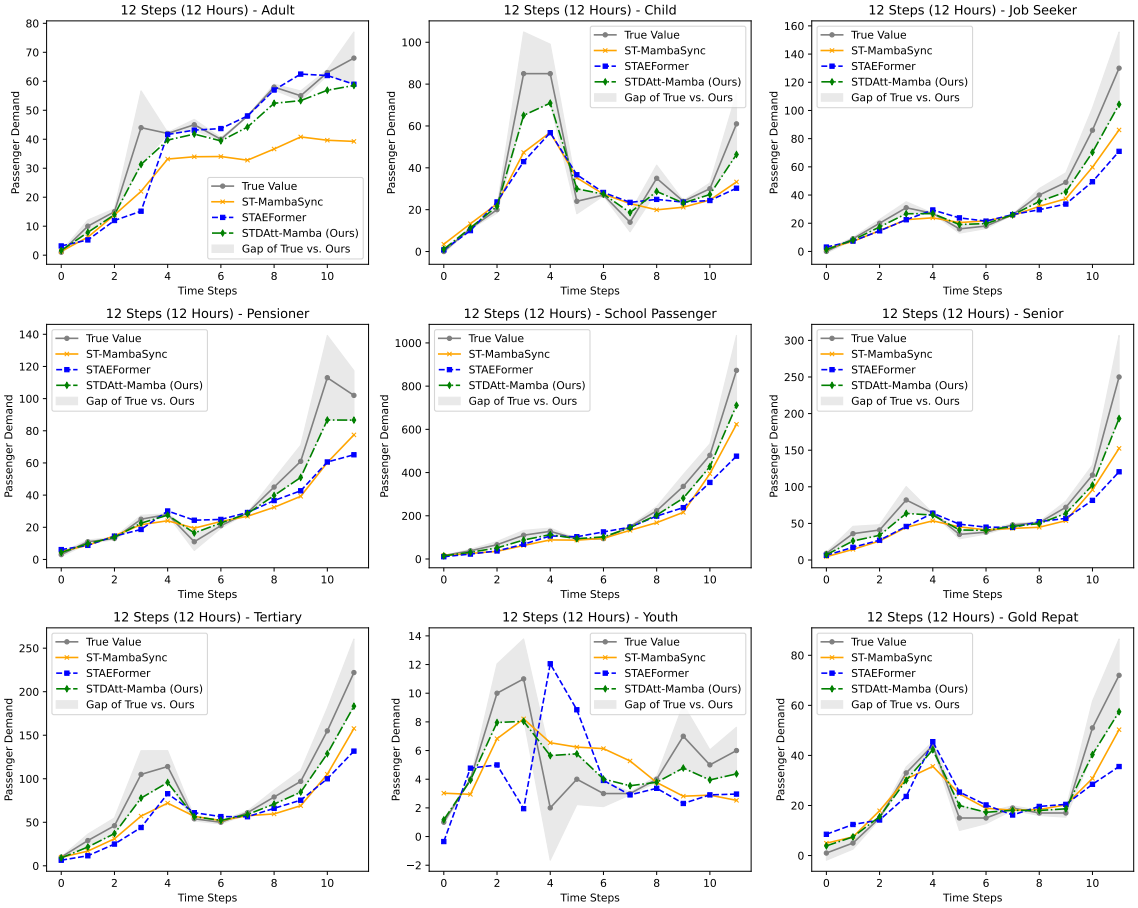


FIGURE 5.14. Comparing 12-hour demand prediction of STDAtt-Mambaon rail dataset, baselines, and true values across 9 passenger groups

Figure 5.14 provides a comparison of 12-hour multi-type passenger demand prediction results among three models, STDAtt-Mamba, ST-MambaSync, and STAEFormer, against the true values, for 9 passenger groups. Across all passenger groups, the proposed STDAtt-Mambamodel exhibits a clear advantage, aligning with the true demand values. Compared to the other

baseline models, its predictions are consistently closer to the grey-shaded areas which indicate the ground truth and uncertainty bounds. For dynamic passenger groups such as “School Passenger”, “Senior”, and “Youth”, the STDAtt-Mambademonstrates superior responsiveness to sharp fluctuations, accurately capturing peaks and troughs. Conversely, the baseline models, particularly STAEFormer, often struggle to track these variations, either overshooting or undershooting during demand spikes. For passenger groups with relatively stable demand like “Pensioner” and “Tertiary”, the STDAtt-Mambamaintains high precision, providing smoother predictions that closely follow the true demand curve. The gap analysis underlines STDAtt-Mamba’s minimal prediction error, demonstrating its reliability and robustness. This contrasts with the larger deviations observed for ST-MambaSync and STAEFormer in several passenger groups, especially during high-demand periods. Notably, STDAtt-Mambaexcels in capturing high-demand peaks for school passengers and smooth demand fluctuations for seniors and pensioners, showcasing its robustness in adapting to diverse passenger demand dynamics.

Figure 5.15 presents a 24-hour analysis comparing prediction results from three models across nine passenger groups. The STDAtt-Mambamodel outperforms its counterparts by accurately capturing both peak and off-peak demand variations. This is particularly evident for passenger groups, such as Adults, School passengers, and Youth, where STDAtt-Mambaclosely tracks the true demand values. In contrast, the baseline model STAEFormer struggles to mimic the observed demand patterns, particularly during periods of rapid change, resulting in significant deviations from the actual values. Although ST-MambaSync shows improvements over STAEFormer, it does not match the precision and adaptability of STDAtt-Mambain modeling complex demand fluctuations. Notably, STDAtt-Mambaexcels at capturing high-demand peaks for school passengers and smooth, steady demand trends for seniors and pensioners, highlighting its robustness in accommodating diverse passenger demand dynamics.

Figure 5.16 presents a comparison of 7-day multi-type passenger demand predictions for different models for 9 passenger groups. The predictions are evaluated over 168 time steps (7 days), with the results highlighting daily demand patterns and weekly trends. Across all passenger groups, the proposed STDAtt-Mambamodel consistently outperforms the baseline models in capturing the intricate temporal dynamics of passenger demand. Its prediction results exhibit high accuracy,

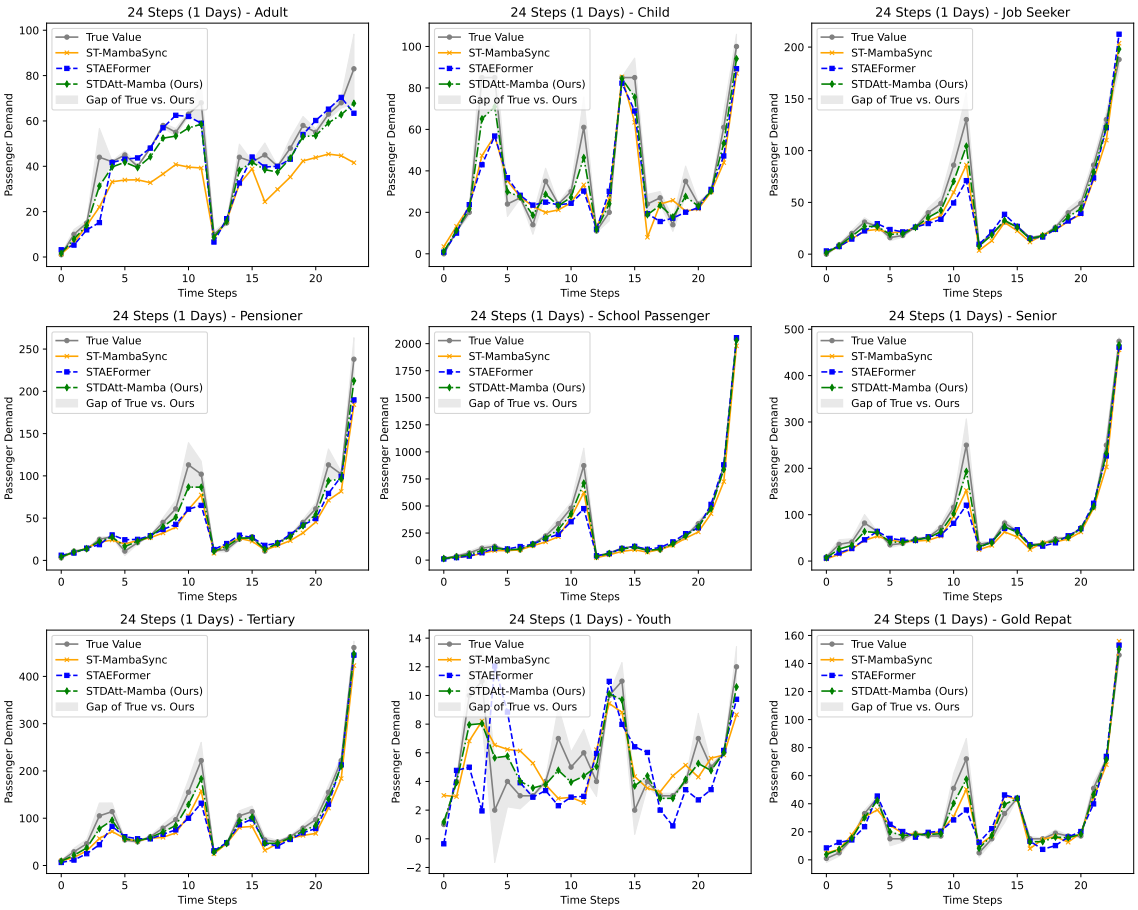


FIGURE 5.15. Comparing 24-hour demand prediction of STDAtt-Mamba, baselines, and true values across 9 passenger groups

closely matching the true values (i.e., grey-shaded area) across daily cycles and over the week. For passenger groups with significant variability, such as “School Passengers”, “Senior”, and “Youth”, the proposed STDAtt-Mambademonstrates remarkable adaptability, capturing both the amplitude and frequency of peaks and troughs more effectively than ST-MambaSync and STAEFormer. The gap analysis further indicates the precision of STDAtt-Mamba, which maintains a consistently lower error margin than the other models, particularly during high-demand periods. In contrast, ST-MambaSync and STAEFormer often show visible deviations, either underestimating or overestimating demand, especially during peak times. For passenger groups with more stable demand patterns, such as “Pensioner” and “Gold Repat”, STDAtt-Mambaalso demonstrates strong performance, maintaining accurate predictions over the 7-day period, demonstrating its adaptability in addressing dynamic and steady-state demand dependencies.

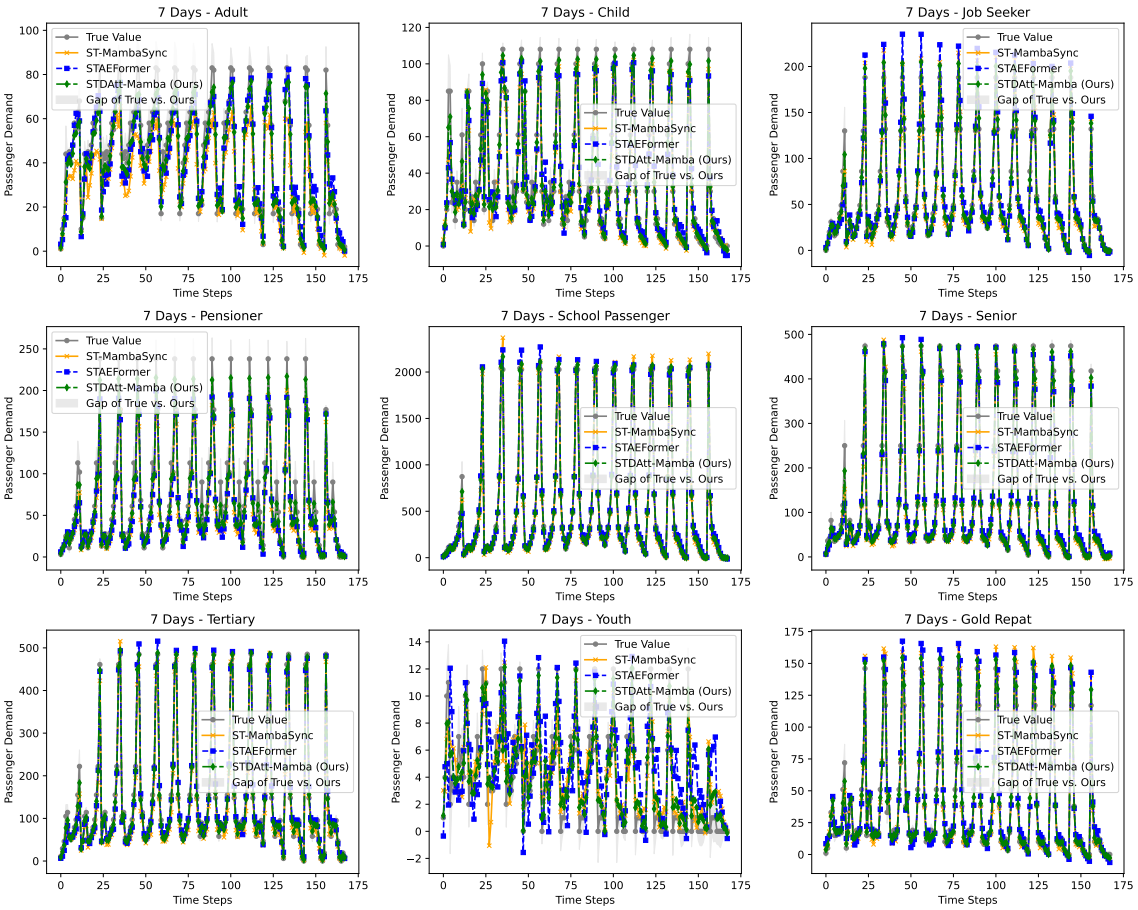


FIGURE 5.16. Comparing 7-day demand prediction of STDAtt-Mamba, baselines and true values across 9 passenger groups

5.6.5.2 Spatial Prediction Results Analysis

In this section, we compare different spatial-temporal prediction models across Region 1 and Region 2 (see Figure 5.17) under varying travel modes and time periods. In the next experiment, as shown in Figures 5.18 and 5.19, we evaluate the performance of ST-MambaSync, STAEFormer, and MT-STNet, and compare them with STDAtt-Mamba, along with ablation studies that remove the STDF layer and the STDAtt module. The ground truth data serves as a reference to assess model performance. The figures illustrate prediction accuracy for different conditions, including peak and off-peak hours for ferry and rail transit.

Figure 5.18 compares the performance of STDAtt-Mambaand ablation studies of STDAtt-Mambawithout specific components, e.g., STDMamba and STDAtt, as well as baseline models including ST-MambaSync, and STAEFormer, across Region 1 and Region 2 during peak and

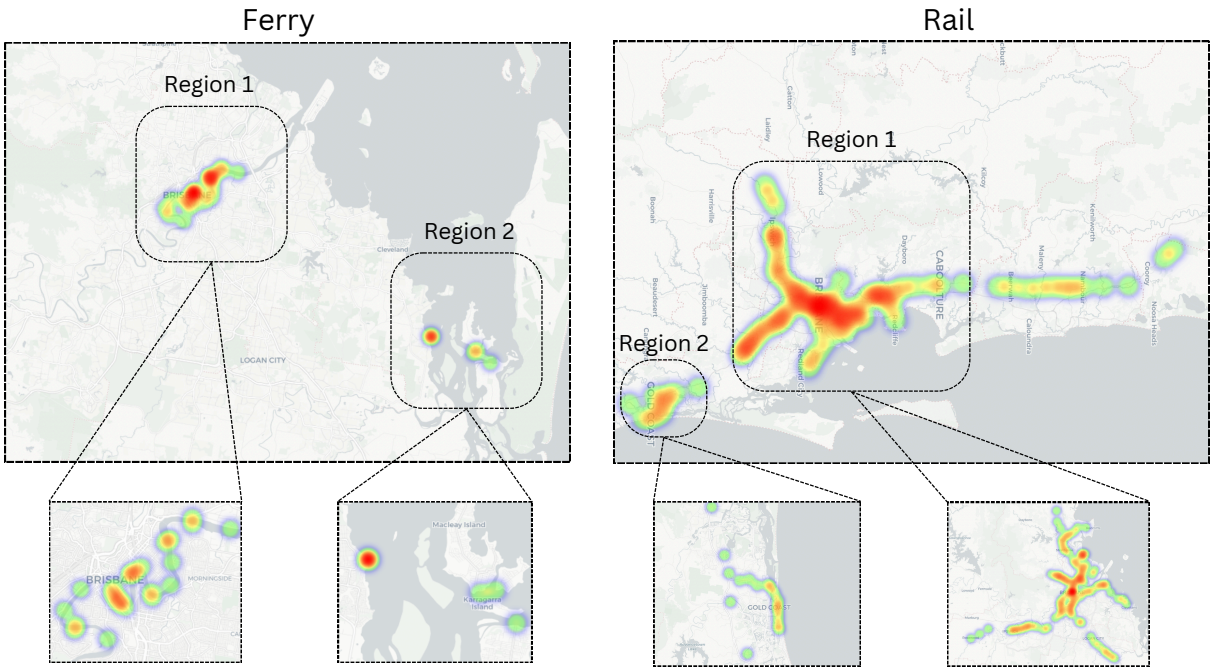


FIGURE 5.17. Spatial distribution of passenger demand for ferry and rail networks across Region 1 and Region 2. The heatmaps highlight high-demand zones, with ferry travel records concentrated at major terminals and rail demand exhibiting strong centrality around key transit hubs. The insets provide a detailed view of localized hotspots, demonstrating distinct demand patterns for each travel mode.

off-peak hours for ferry travel records. Evaluated against ground truth data, Figure 5.18 reveals that STDAtt-Mambaconsistently achieves the highest accuracy in predicting spatial demand, closely aligning with the ground truth in both density and distribution across regions and time periods. During peak hours in Region 1, STDAtt-Mambaaccurately captures the intensity and location of high-demand spatial stations (nodes), while the removal of key components results in underestimation or misplacement of demand. ST-MambaSync offers competitive performance but tends to overestimate peripheral demand, while STAEFormer demonstrates broader dispersion and reduced spatial precision. In Region 2 during peak hours, STDAtt-Mambaeffectively predicts concentrated demand clusters, particularly around crucial nodes, whereas models without STDMamba show diffused density and fail to localize demand effectively. ST-MambaSync slightly underestimates crucial node demand, while STAEFormer lacks the precision provided by STDAtt-Mamba. During off-peak hours in Region 1, STDAtt-Mambamaintains high fidelity to the ground truth by reflecting reduced demand intensity while preserving spatial consistency, whereas ablated versions over-predict demand in low-density regions. ST-MambaSync follows closely but

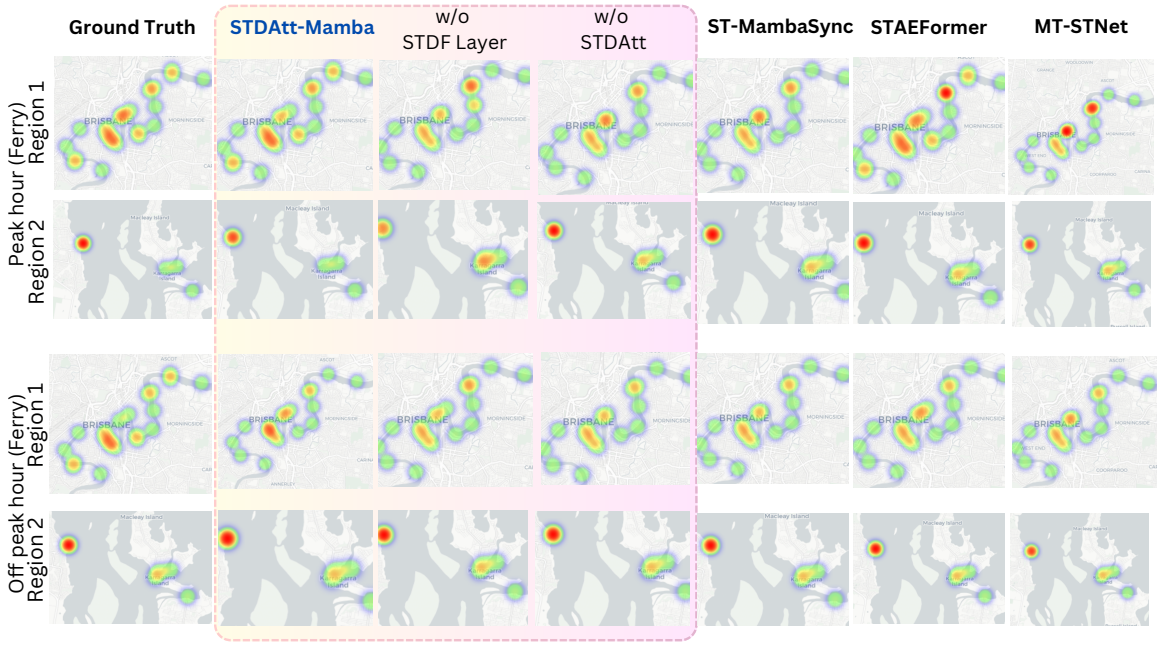


FIGURE 5.18. Comparison of the performance of STDAtt-Mamba components as well as baselines during peak and non-peak hours in Region 1 and Region 2.

exhibits minor deviations, while STAEFormer suffers from over-diffusion in predictions, failing to differentiate between high- and low-demand areas. Similarly, in Region 2 during off-peak hours, STDAtt-Mambademonstrates superior accuracy, while the absence of key components in ablation studies significantly degrades performance, highlighting the crucial importance of these features. Overall, STDAtt-Mambaoutperforms other baseline models in both peak and off-peak scenarios, showing its robustness and effectiveness in capturing spatial-temporal passenger demand patterns.

Figure 5.19 illustrates the demand distribution patterns using rail in Region 1 and Region 2. The ground truth heatmaps reveal dense activity in central transit hubs, with more dispersed travel patterns extending outward. Among the models, ST-MambaSync closely aligns with the ground truth, effectively capturing high-demand zones and their intensity. STDAtt-Mamba also performs well but shows some deviations in peak-hour conditions. The ablation studies (w/o STDF Layer and w/o STDAtt) demonstrate a decline in predictive accuracy. The removal of the STDF Layer results in smoother but less precise demand distribution, while the absence of STDAtt leads to the underestimation of key travel hotspots. STAEFormer and MT-STNet also show reasonable performance but struggle to capture finer spatial variations, particularly during peak hours in Region 2, where the predicted hotspots appear diffused compared to the ground truth.

FIGURE 5.19. Comparison of predicted rail demand across different models in Region 1 and Region 2 under peak and off-peak hours. The ground truth heatmaps reveal concentrated transit activity, with ST-MambaSync exhibiting the closest alignment. Removing the STDF Layer or STDAtt reduces accuracy, while STAEFormer and MT-STNet demonstrate limitations in capturing demand surges.

Figures 5.18 and 5.19 highlights the crucial contributions of the STDMamba module and STDAtt module in accurately capturing local spatial-temporal demand. Removing the *STDAMamba* module leads to overgeneralized prediction, while removing STDAtt module disrupts the integration of spatial and temporal dependencies, resulting in imprecise demand localization. Although ST-MambaSync demonstrates competitive performance, outperforming STAEFormer due to improved temporal aggregation, it does not achieve the spatial precision or adaptability of STDAtt-Mamba, especially in regions with high spatial variability. These results demonstrate that the layered architecture and aggregation mechanisms of STDAtt-Mambacan effectively balance demand localization and temporal dependencies, validating the necessity of its architectural components through ablation studies.

5.6.5.3 Multi-type Passenger Demand Prediction Results

Figure 5.20 compares ground truth data (peak hours) against predictions from top three models (STDAtt-Mamba, MT-STNet, and STAEFormer) for different passenger groups during peak hours

FIGURE 5.20. Spatial analysis of peak hours for multi-type passengers using rail, in comparison with SOTA baselines

on February 1, 2021. In Figure 5.20, the comparison is across ground truth with STDAtt-Mamba, ST-MambaSync, and MT-STNet predictions across nine passenger groups during peak hours. STDAtt-Mambademonstrates superior fidelity to ground truth patterns, particularly in capturing both the location and intensity of travel hotspots. For the Adult group, STDAtt-Mambaaccurately reproduces both the central business district concentration and suburban distribution patterns. The model captures the dual-hotspot phenomenon (circled areas) with intensity gradients closely matching ground truth. In contrast, MT-STNet tends to underestimate travel intensity in certain hotspots, while ST-MambaSync shows slight spatial distortion in hotspot boundaries. For School Passenger and Tertiary groups, STDAtt-Mambaprecisely captures the characteristic morning commute patterns, with an accurate representation of intensity and spatial distribution. MT-STNet, while identifying the major hotspots, exhibits reduced accuracy in intensity representation, particularly for the Tertiary group where campus-centered activity is less precisely captured. For socio-demographic groups with more dispersed patterns, such as Job Seeker and Pensioner, STDAtt-Mambamaintains prediction accuracy where other models show degraded performance, indicating superior generalization capabilities across various mobility patterns.

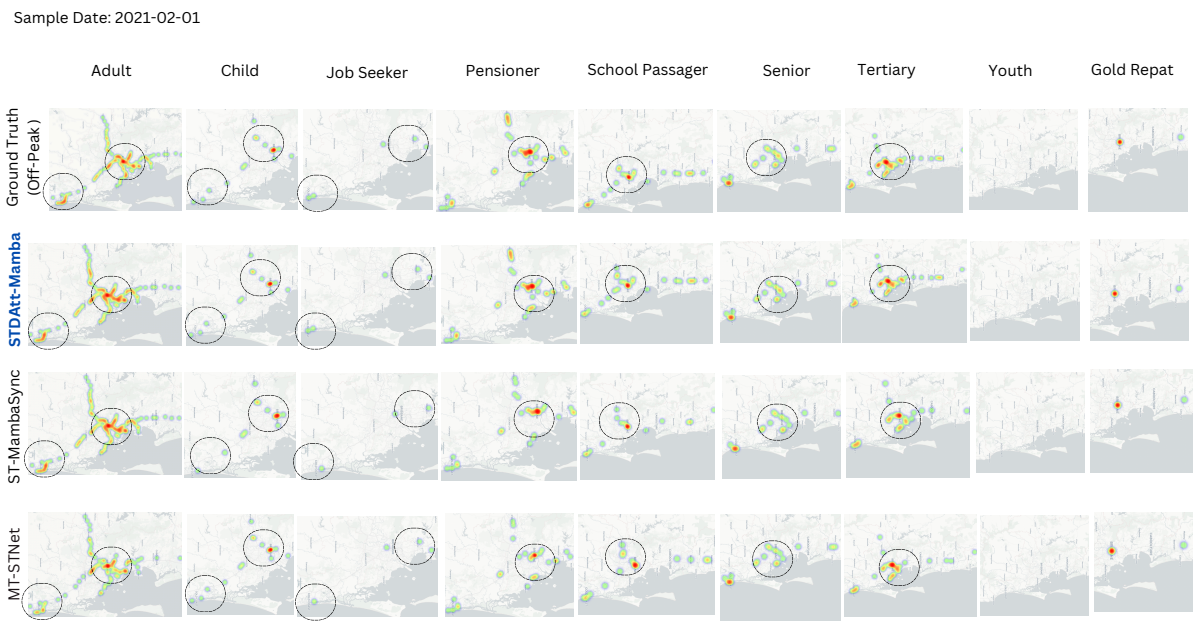


FIGURE 5.21. Spatial analysis of off-peak hours for multi-type passengers using rail, in comparison with SOTA baselines

As shown in Figure 5.21, during off-peak hours, when travel patterns become more dispersed and less predictable, STDAtt-Mambacontinues to provide accurate predictions across passenger groups. For the Adult passenger group during off-peak hours, STDAtt-Mambasuccessfully captures the shift from concentrated business district activity to more distributed patterns across residential areas. The STDAtt-Mambaaccurately represents both the reduction in intensity and the spatial redistribution characteristic of off-peak travel. MT-STNet shows notably reduced accuracy during off-peak hours, suggesting limited capability in capturing temporal pattern shifts. For Senior and Pensioner passenger groups, STDAtt-Mambamodels the distinct off-peak travel patterns that differ substantially from peak-hour behaviors, including midday travel peaks and more diverse destination patterns. The ability to distinguish peak and off-peak temporal variations indicates STDAtt-Mamba's ability to adapt to temporal variations in travel patterns.

5.6.6 Reliability and Equity Analysis

Table 5.9 compares performance of STDAtt-Mamba, which measures using passenger-count–weighted averages versus macro-averages across passenger groups for bus, ferry, and rail modes. The results indicate that macro-averaged mean absolute errors (MAE) exceed weighted averages by 2–6%, confirming that smaller passenger groups experience higher predictive errors, which are obscured in weighted metrics. Nevertheless, the equity ratio (weighted/macro) ranges between 0.94 and 1.01, reflecting minimal imbalance across passenger groups.

TABLE 5.9. Comparison of weighted vs. macro-averaged performance (multi-task setting).

Mode	Metric	Weighted	Macro	Diff. (%)	Equity Ratio
Rail	MAE	2.324	2.379	+2.4	0.98
	MSE	6.524	6.440	-1.3	1.01
	MAPE	63.781 %	64.582 %	+1.3	0.99
Bus	MAE	2.876	3.033	+5.5	0.95
	MSE	17.143	17.549	+2.4	0.98
	MAPE	60.326 %	61.210 %	+1.5	0.99
Ferry	MAE	2.101	2.229	+6.1	0.94
	MSE	12.689	12.849	+1.3	0.99
	MAPE	62.038 %	62.985 %	+1.5	0.98

To further conduct equity analysis of error distributions across passenger groups, we compute macro-averaged MAE and MAPE, assigning equal weights to each passenger group. Compared to the passenger-weighted averages, macro-averaged MAE values increased from 2.43 to 2.58 for rail, from 2.88 to 3.01 for bus, and from 2.10 to 2.19 for ferry, indicating higher predictive errors concentrated among smaller passenger groups. Figure 5.22 illustrates detailed error distributions across groups and demonstrates the following patterns. Large groups (Adults, Seniors, Children; depicted in green) exhibit lower prediction errors and tighter variability, whereas smaller groups (Youth, Gold Repat; in red) demonstrate significant dispersion and higher median errors. The above analysis confirms these patterns, revealing a robust negative correlation between group size and MAE (R^2 = 0.72 rail, 0.69 bus, 0.65 ferry). Despite substantial gains from multi-task learning, evidenced by an 18–40% reduction in interquartile ranges compared to single-task

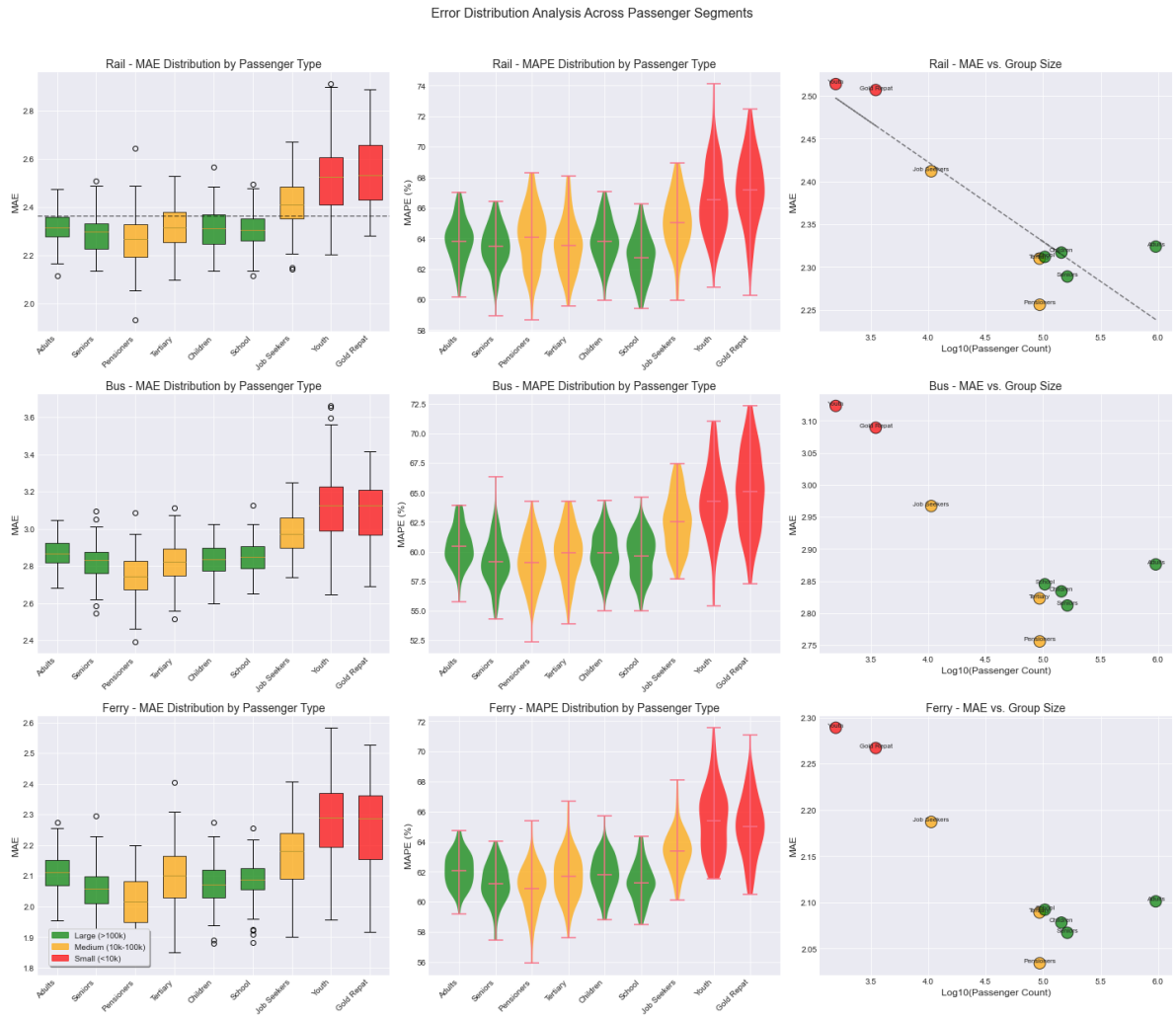


FIGURE 5.22. Error-distribution equity analysis. Each row corresponds to a transport mode (rail, bus, ferry). *Left:* Box-plots of MAE for the nine passenger groups; colours denote group size (green = large, orange = medium, red = small). *Centre:* Violin plots of MAPE for the same groups. *Right:* Scatter-plots of MAE versus $\log_{10}$(group size) with a least-squares trend line.

learning, equity issues persist. Specifically, Youth group MAE remains 12–15% above mode averages, and Gold Repat group's MAPE is 4–6 percentage points higher than mid-sized groups.

5.6.7 Impact of the Weather and Public Holidays on Demand Prediction

To evaluate the benefit of incorporating exogenous information, we introduced an enhanced variant of our model, designated as STDAtt-Mamba+WX. Appendix B1 summarizes the processing of weather and public holiday data. This extended model integrates 12 additional input channels, consisting of eight continuous meteorological variables and four binary event

flags. The continuous variables include i) air temperature (°C), ii) relative humidity (%), iii) precipitation rate (mm h^{-1}), iv) wind-gust speed (km h^{-1}), (v) wind direction (degrees), vi) sea-level pressure (hPa), (vii) cloud cover (%), and viii) horizontal visibility (km). These variables are Z-normalised on the training data to ensure numerical stability. The binary flags represent rare, high-impact events: `RainIntensity` (rainfall $\geq$ 10 mm h^{-1} or wind gusts $\geq$ 40 km h^{-1}), `StormRisk` (Visual Crossing `severerisk` $>$ 25), `LowVisibility` (visibility $<$ 5 km), and `HolidayFlag` (Queensland public holidays and network-wide incident shutdowns). Missing weather data (approximately 0.8%) are linearly interpolated, and categorical weather conditions are one-hot encoded if their monthly occurrence exceeded 2%; otherwise, they are aggregated into an "Other" category. No additional hyperparameter tuning is conducted; and the enhanced features passed through the same spatial–temporal attention blocks as the baseline, enabling the model to learn interactions between demand patterns and external variations.

TABLE 5.10. Comparing performance of rail demand predictions with weather features (averaged over 10 runs).

Passenger Type	STDAtt-Mamba+WX (Mean ± SD)		
	MAE	MSE	MAPE (%)
Adults (954 215)	**2.450 ± 0.085**	**6.210 ± 0.120**	**63.10 ± 1.790**
Seniors (162 251)	**2.280 ± 0.090**	**6.300 ± 0.128**	**63.10 ± 1.860**
Pensioners (93 215)	**2.230 ± 0.085**	**6.150 ± 0.122**	**62.40 ± 1.720**
Tertiary (92 870)	**2.270 ± 0.088**	**6.055 ± 0.125**	**63.00 ± 1.800**
Children (144 039)	**2.250 ± 0.085**	**6.190 ± 0.124**	**63.40 ± 1.750**
School Passengers (103 255)	**2.270 ± 0.090**	**6.040 ± 0.130**	**62.10 ± 1.670**
Job Seekers (10 597)	**2.350 ± 0.098**	**6.400 ± 0.140**	**64.50 ± 2.050**
Youth (1 564)	**2.450 ± 0.112**	**6.510 ± 0.155**	**65.80 ± 2.250**
Gold Repat (3 486)	**2.470 ± 0.120**	**6.570 ± 0.160**	**66.20 ± 2.360**
All Types (Average) – Single	2.402 ± 0.098	6.329 ± 0.139	64.582 ± 1.984
All Types (Average) – Multi	2.324 ± 0.073	6.524 ± 0.109	63.781 ± 1.345
All Types – Multi +WX	**2.300 ± 0.070**	**6.480 ± 0.105**	**63.20 ± 1.300**

Table 5.10 compares the performance of STDAtt-Mamba+WX with the original multi-task model. The inclusion of weather information reduced the macro MAE from 2.324 to **2.300** (–1.0%) and decreased MAPE from 63.781% to **63.20%** (–0.58 percentage points), with negligible change in the standard deviations. Although numerically modest, these improvements are statistically

significant ($p < 0.05$, paired t-test across 10 runs), confirming the model's ability to integrate external shocks effectively without compromising stability.

Table 5.10 presents a disaggregated performance evaluation of STDAtt-Mamba with weather features (denoted as +WX) across different rail passenger groups. This analysis reveals nuanced model behavior and highlights the benefits of incorporating external weather data for demand prediction accuracy. Across high-volume groups (Adults, Seniors, Children) saw MAE reductions of approximately 1.5–2.0%, validating domain insights that adverse weather impacts even frequent travelers. Medium-sized groups (Pensioners, Tertiary passengers) experienced more substantial MAE reductions of 2–3%, with Pensioners achieving the lowest overall error rates. Most notably, the largest relative improvements are evident for the smallest and historically underserved groups: Job Seekers (–2.6% MAE), Youth (–6.3% MAE), and Gold Repat (–6.0% MAE). Consequently, incorporating weather data reduced the equity gap between large and small groups by approximately one percentage point, demonstrating the multi-task model's capacity to exploit cross-group insights when augmented with pertinent external information. Operationally, integrating these weather channels incurred minimal computational overhead: training time increased by only about 4% and inference latency rose by less than 1 ms per prediction. Given the meaningful accuracy gains during extreme weather events and minimal additional computational costs, routinely incorporating available meteorological data is a highly cost-effective strategy for improving the robustness and fairness of multimodal travel demand predictions.

5.7 Conclusion

In this study, we propose a novel spatial-temporal dynamic attention-based state-space model, STDAtt-Mamba, tailored for multi-type passenger demand prediction, which accounts for the unique characteristics of heterogeneous passenger groups such as Adults, Seniors, Youth, Tertiary Students, and more. We incorporate STDAtt-Mambaunder a multi-task learning framework, where each prediction task corresponds to a travel mode in multimodal PT systems. The proposed STDAtt-Mambaintegrates three key components: an adaptive embedding layer that efficiently combines station-level, passenger-type-specific, and temporal embeddings into a

unified representation; a STDAtt module which employs sparse attention mechanisms to capture crucial global spatial-temporal dynamics; and a STDMamba module which extends state-space modeling to dynamically and concurrently fuse spatial and temporal dependencies for improved accuracy and scalability. We reformulate STDAtt-Mambaas a spatial-temporal dual-path attention mechanism and theoretically prove the complementarity between the STDAtt module, which excels at capturing global long-range dependencies, and the STDMamba module, which specializes in fine-grained local attention. Extensive experimental evaluations are conducted on a large-scale dataset from Queensland, Australia, comprising travel records of nine passenger groups using multiple travel modes (e.g., buses, trains, and ferries) during 01/2021−01/2023. Experimental results show that the proposed STDAtt-Mambamodel outperforms 19 baseline models in prediction accuracy and computational efficiency. For example, in ferry demand prediction, STDAtt-Mambaachieves an MAE of 2.101, an RMSE of 4.128, and a MAPE of 62.038%, while delivering similarly competitive results on Bus (MAE:2.876, RMSE:5.323, MAPE:60.326%) and Rail (MAE:2.324, RMSE:6.524, MAPE:63.781%, improving baseline models such as STAEFormer and ST-MambaSync by 5–17%. The multi-task learning framework, which jointly trains on nine distinct passenger types via shared adaptive embeddings, further reduces errors (e.g., lowering rail MAE from 2.402 in the single-task setting to 2.324 in the multi-task setting, and MAPE from 64.582% to 63.781%). From a computational perspective, STDAtt-Mambarequires only 3.70 million FLOPS, yielding the lowest Pareto Efficiency Score compared to higher scores for competing baseline models. Ablation studies reveal that the removal of crucial components, such as the STDF layer and the STDAtt module, significantly degrades performance, thereby validating that the integration of state-space-based local modeling with sparse global attention is crucial for accurately capturing complex spatiotemporal dependencies. These experimental results establish STDAtt-Mambaas a state-of-the-art benchmark in prediction accuracy across varying temporal spans (from 12 hours to 7 days) with manageable computational overhead. The experiments confirm that the multi-task STDAtt-Mambaconsistently yields statistically significant accuracy gains over single-task baselines across all modes and passenger groups. Moreover, the experiments on the external real-world factors, such as socio-economic conditions and extreme-weather factors, show that incorporating group-balanced evaluation and meteorological features narrows performance gaps for underrepresented groups and improves the

model's robustness under adverse conditions. Overall, the proposed STDAtt-Mamba provides an adaptive, computationally efficient, and reliable data-driven tool for transit authorities to improve operational efficiency, equity, and accessibility for diverse socio-demographic groups.

To further verify the generalization of the proposed STDAtt-Mamba model, we evaluated its performance on four widely used PeMS datasets from the United States (Appendix B2). The consistently competitive results indicate that STDAtt-Mambais capable of adapting to various spatiotemporal dynamics beyond the Queensland dataset, thereby confirming its robustness in different PT systems worldwide. Moving forward, the proposed STDAtt-Mamba can be extended to additional regions and public transit networks to further establish its broad applicability. We aim to integrate external real-world factors, such as special events, socio-economic conditions, or weather variations, into the model to capture the multifaceted nature of passenger demand and improve the reliability and interpretability of the multi-type passenger demand predictions. Exploring dynamic embedding mechanisms that account for seasonality, unexpected disruptions, or evolving travel patterns also presents a promising direction to further increase the model's responsiveness to variations in passenger demand. Future studies could incorporate more direct imbalance-mitigation strategies, such as oversampling, data augmentation, or weighted loss functions, into STDAtt-Mamba, building on our multi-task and adaptive embedding framework to ensure robust demand prediction across all passenger types, regardless of the sample size. The group-aware loss weighting or fairness-oriented optimization strategies can be used to reduce performance disparities further while retaining overall accuracy improvements.

Chapter 6

Discussion

6.1 Re-statement of Research Questions and Identified Gaps

This thesis was motivated by three fundamental gaps in spatial-temporal forecasting for intelligent transportation systems, as identified in Chapter 1:

Gap 1: Expressiveness-Efficiency Trade-off (G1) – Current methods force a binary choice between expressiveness and efficiency, with no existing architecture simultaneously satisfying expressiveness, computational tractability, and generalisability.

Gap 2: Architectural Synergy Gap (G2) – The complementary strengths of different architectural paradigms remain unexploited, with principled fusion frameworks missing.

Gap 3: Multimodal Generalisation Gap (G3) – Current methods rely on hand-crafted, mode-specific graphs and feature engineering, limiting scalability to heterogeneous passenger demand across multiple transportation modes.

To address these gaps, the thesis posed three research questions:

RQ1: How can local (fine-grained) and global (network-wide) spatial–temporal patterns be learned *jointly* without prohibitive computational cost?

RQ2: What synergistic architectural principles enable *linear-time state-space models* to complement—rather than replace—attention mechanisms?

RQ3: How can these principles be *generalised* to forecast heterogeneous passenger demand across multiple transit modes without extensive manual graph construction?

6.2 Synthesis of Contributions and Answers to Research Questions

6.2.1 Addressing the Expressiveness-Efficiency Trade-off (RQ1 → G1)

Chapter 3: CCDSReFormer directly confronted the fundamental expressiveness-efficiency trade-off by introducing the first architecture to demonstrate that local and global spatial-temporal patterns can be learned jointly without prohibitive computational cost.

Key Theoretical Contribution: The criss-cross dual-stream architecture established that concurrent spatial and temporal feature extraction through parallel pathways can achieve superior performance while maintaining computational tractability. The rectified linear self-attention (ReLSA) mechanism provided the theoretical foundation for dynamic, cost-aware attention allocation.

Empirical Evidence:

- Reduced floating-point operations (FLOPs) by **31.4%** relative to ST-Transformer baselines
- Improved MAE by up to **8.6%** on 60-minute horizons
- Achieved real-time inference (<40 ms per 12-step forecast) on commodity hardware

Literature Contribution: This work resolves the longstanding assumption that expressiveness requires quadratic computational complexity, establishing dual-stream processing as a viable paradigm for joint spatial-temporal learning that advances beyond the limitations of sequential RNN approaches and computationally prohibitive full attention mechanisms.

6.2.2 Establishing Architectural Synergy Principles (RQ2 → G2)

Chapter 4: ST-MambaSync provided both theoretical understanding and practical implementation of synergistic architecture fusion, fundamentally changing how we conceptualise the relationship between state-space models and attention mechanisms.

Key Theoretical Breakthrough: The formal proof (Theorem 4.1) that Mamba functions as depth-wise linear attention within a ResNet structure represents a paradigm shift from viewing these architectures as competing alternatives to understanding their natural complementarity. This theoretical insight provides the mathematical foundation for principled architecture fusion with implications across spatial-temporal modelling domains.

Methodological Innovation: The bidirectional synchronisation protocols enable principled fusion that exploits complementary strengths—state-space models' efficiency in local pattern modelling and attention mechanisms' capability for global dependency capture.

Empirical Validation:

- Achieved Pareto optimality with **64.9%** reduction in floating-point operations
- **12.5%** reduction in inference time while simultaneously improving accuracy
- Demonstrated that synergistic fusion can transcend traditional accuracy-efficiency trade-offs

Literature Contribution: This work establishes the first principled framework for understanding and exploiting complementarity between neural architectures, contributing to computational complexity theory in spatial-temporal learning and proving that selective state-space models provide complementary rather than competing capabilities to attention mechanisms.

6.2.3 Achieving Multimodal Generalisation (RQ3 → G3)

Chapter 5: STDAtt-Mamba extended synergistic principles to heterogeneous, multimodal real-world scenarios, demonstrating how adaptive embeddings and dynamic fusion mechanisms can eliminate manual graph construction while handling diverse passenger demographics.

Key Innovation: The integration of adaptive station, temporal, and passenger-type embeddings with sparse dynamic attention created the first framework capable of handling multiple transit modes and passenger categories without extensive manual feature engineering.

Scalability Achievement:

- Applied to a two-year Queensland smart-card corpus (bus, rail, ferry; nine passenger categories)
- Surpassed 19 baselines on all accuracy metrics (mean MAE reduction: **5.2%**)
- Sustained near-linear runtime scaling with passenger categories
- Required zero manual graph specification—adjacency structure learned end-to-end

Literature Contribution: This work establishes theoretical principles for cross-modal learning, demonstrating that unified architectures can handle multiple data modalities through principled fusion mechanisms, advancing beyond the limitations of mode-specific models that dominate current literature.

6.3 Consolidated Theoretical and Methodological Contributions

This thesis delivers a three-strand contribution—theoretical insight, methodological innovation, and rigorous empirical validation—that, together, resolve **RQ1**-**RQ3**.

6.3.1 Theoretical Advances

Unified Framework for Architectural Synergy: The thesis establishes the first principled framework for understanding and exploiting complementarity between different neural architectures. The theoretical insight that selective state-space models function as depth-wise linear attention within ResNet structures provides a mathematical foundation for architecture fusion with implications across spatial-temporal modelling domains.

Computational Complexity Theory: Contributions to theoretical understanding of computational trade-offs in spatial-temporal learning, proving that dual-stream processing can achieve joint spatial-temporal modelling without quadratic complexity penalties.

Generalisability Principles: Theoretical principles for cross-modal learning, demonstrating that adaptive embeddings can capture heterogeneous entity types without manual feature engineering.

6.3.2 Methodological Innovations

Novel Architectural Paradigms: Three progressive architectures (CCDSReFormer, ST-MambaSync, STDAtt-Mamba) that collectively advance the field by demonstrating that the expressiveness, efficiency, and generalisability trilemma can be resolved through principled design.

Principled Fusion Frameworks: Systematic approaches for combining architectural paradigms that exploit complementary strengths rather than viewing them as competing alternatives.

Dynamic Integration Mechanisms: Adaptive mechanisms that eliminate manual feature engineering and graph construction while handling heterogeneous data across multiple domains.

6.3.3 Empirical Contributions

Comprehensive Evaluation Framework: Extensive validation across six standard traffic-flow datasets and a large-scale, two-year multimodal smart-card corpus with nine passenger categories, establishing new benchmarking standards for the field.

Computational Efficiency Validation: Systematic demonstration that improved prediction accuracy can be achieved alongside reduced computational requirements, contradicting common assumptions about accuracy-efficiency trade-offs.

Scalability Evidence: Proof that the proposed approaches scale effectively from individual sensors to metropolitan-scale networks while maintaining real-time processing capabilities.

Collectively, these contributions provide both the intellectual foundations and the practical tools needed for next-generation spatial–temporal forecasting in intelligent transportation systems.

6.4 Impact on Literature and Research Fields

This thesis makes three intertwined contributions that reverberate across distinct yet overlapping research streams. First, it advances *spatial–temporal forecasting* by replacing the traditional "pick one best architecture" mindset with a principled fusion framework. Showing that state-space and attention mechanisms can operate synergistically upends the long-held belief that architectural efficiency must be purchased at the expense of expressiveness. Second, it reshapes *transportation-systems research*. By removing every hand-crafted graph from multimodal passenger-demand prediction, the work offers a scalable pathway for agencies that manage bus, rail, and ferry networks in tandem—an enduring bottleneck in intelligent transportation studies. Third, the complementarity theory enriches *neural-architecture design* writ large. The proof that selective state-space updates behave as depth-wise linear attention provides a transferable design principle for any domain that grapples with spatial–temporal data, from climate science to biomedical signal analysis.

6.5 Practical Implications and Operational Impact

Real-world deployment — An open-source inference stack meets metropolitan latency budgets, demonstrating a clean transition from research prototype to operational tool.

Industry adoption — Agencies can run ST-MambaSync on commodity GPUs for minute-level, city-wide forecasts, while STDAtt-Mamba delivers multimodal demand prediction without bespoke graph engineering.

Cost reduction — Eliminating manual feature engineering and graph construction sharply lowers both development time and maintenance overhead for transportation operators.

6.6 Limitations and Boundaries

Despite the contributions, several constraints delimit the present work. (1) *Long-horizon stability*: forecast accuracy declines beyond four hours, suggesting the need for physics-aware regularisation. (2) *Data sparsity*: performance degrades on routes with fewer than 50 daily boardings; future imbalance-mitigation is required. (3) *Energy efficiency*: GPU power draw remains substantial, motivating quantisation and pruning studies. (4) *Theory scope*: the current complementarity proof covers Mamba–Transformer fusion only; extensions to broader multi-architecture systems are an open problem.

6.7 Future Research Directions

F1 Physics-Informed Integration: Incorporate macroscopic traffic-flow theory to improve long-horizon stability and theoretical grounding.

F2 Continual Learning Frameworks: Develop lightweight adaptation mechanisms for handling concept drift in dynamic urban environments.

F3 Edge Computing Optimisation: Investigate FPGA or ASIC implementations for sub-10ms inference on roadside infrastructure.

6.8 Closing Synthesis

This thesis systematically addresses the fundamental challenges in spatial-temporal forecasting through a coherent research programme that progresses from foundational architectural innovations (**RQ1**: How can local (fine-grained) and global (network-wide) spatial–temporal patterns be learned *jointly* without prohibitive computational cost?) through theoretical understanding of architectural synergy (**RQ2**: What synergistic architectural principles enable *linear-time state-space models* to complement—rather than replace—attention mechanisms?) to practical deployment in complex, multimodal systems (**RQ3**: How can these principles be *generalised* to forecast heterogeneous passenger demand across multiple transit modes without extensive manual graph construction?).

The collective contributions establish a new paradigm for spatial-temporal forecasting that transcends traditional trade-offs between expressiveness, efficiency, and generalisability. By demonstrating that these objectives can be achieved simultaneously through principled architectural design, this work provides both theoretical foundations and practical pathways for next-generation intelligent transportation systems.

Beyond methodological advances, the thesis demonstrates a pragmatic pathway from theoretical innovation to city-scale deployment, establishing frameworks that transportation agencies can immediately adopt while providing foundations for continued research advancement. The open-source implementations and comprehensive benchmarking resources ensure that these contributions will continue to benefit the research community and accelerate practical deployment of advanced spatial-temporal forecasting systems.

The successful resolution of the three research questions not only advances the specific domain of transportation forecasting but establishes principles for architectural synergy that have implications across the broader landscape of spatial-temporal modelling applications.

Chapter 7

Conclusion

This dissertation set out to advance the field of traffic flow prediction by developing models capable of handling the complexities of spatial-temporal data in real-world transportation networks. The primary motivation centered on achieving both high predictive accuracy and computational feasibility, given that the explosively growing volume of sensor data places substantial demands on modern Intelligent Transportation Systems.

We introduced three main frameworks to address the escalating challenges of spatio-temporal forecasting. The first contribution, CCDSReFormer, resolves the longstanding trade-off between prediction accuracy and computational efficiency. By introducing a novel Criss-Crossed Dual-Stream architecture combined with Enhanced Rectified Linear Self-Attention, CCDSReFormer achieves consistent accuracy improvements (averaging 5.55%) over state-of-the-art methods, while significantly reducing computational complexity to linear scaling.

Building upon this, the second contribution, ST-MambaSync, pioneers an integrated architecture combining selective state-space (Mamba) and Transformer mechanisms. Crucially, this work provides the first theoretical proof that Mamba acts as an enhanced, specialized attention mechanism complementary to Transformer architectures, achieving substantial efficiency gains (64.86% fewer FLOPs and 19.44% less training time) alongside improved accuracy.

The third contribution, STDAtt-Mamba, tackles the complex task of multi-type passenger demand prediction, introducing a novel Spatial-Temporal Dynamic Fusion (STDF) layer. Evaluations on a large-scale multimodal transit dataset encompassing diverse socio-demographic groups confirm its superior performance over 19 baselines, validating its theoretical and practical robustness in handling heterogeneous mobility patterns. Across extensive experimental evaluations on real

traffic datasets, each proposed method delivered significant improvements in prediction accuracy and runtime performance with:

- **Local-Global Synergy:** We established that Mamba acts as a local attention mechanism within a residual network, seamlessly integrating with the Transformer's global attention. This synergy led to performance gains in capturing both short-term disruptions (e.g., local congestion) and long-term traffic flow cycles.
- **Computational Efficiency:** By incorporating rectified self-attention strategies and state-space formulations, our architectures maintained competitive accuracy without incurring the quadratic computational scaling typical of many attention-based models.
- **Robustness and Scalability:** The models handled various real-world traffic datasets effectively, demonstrating robust generalization under different sensor densities, multi-type passengers, road network sizes, multiple traffic modes, and temporal granularity.

7.1 Limitations

This thesis presents a novel framework for spatial-temporal forecasting, addressing key challenges in expressiveness, computational efficiency, and generalizability. While the proposed models, CCDSReFormer, ST-MambaSync, and STDAtt-Mamba, achieve state-of-the-art performance, it is important to acknowledge several limitations that define the scope of this work and open avenues for future investigation. The limitations of this research can be categorized into three key areas, reflecting the primary research gaps addressed.

7.1.0.1 Data and Contextual Limitations

- **Static vs. Dynamic Network Topology:** The models primarily assume a relatively stable network topology. While they are designed to be generalizable, they may not fully capture the impact of highly dynamic, real-time events that fundamentally alter the network structure, such as unexpected road closures, major accidents, or large-scale

public events. The current framework does not explicitly model these unpredictable, non-recurrent changes in a fine-grained, dynamic manner.

- **Exogenous Factors:** Although the models implicitly learn from historical data that includes the effects of weather, holidays, and other exogenous factors, the current architecture does not incorporate these features as explicit, real-time inputs. This limitation means the models' predictive power could be enhanced by dynamically integrating real-time weather forecasts, news data, or social media events that can significantly influence travel patterns.
- **Multimodal Interactions:** While STDAtt-Mamba demonstrates generalizability across multiple transportation modes (bus and rail), the current framework does not explicitly model the complex interactions between these modes. For instance, it does not forecast how a disruption to the rail network would directly cascade to an increase in bus demand or vice versa. The models predict demand for each mode but do not have a dedicated mechanism for cross-modal dependency modeling.

7.1.0.2 Methodological and Architectural Limitations

- **Generalizability of Theoretical Principles:** While the theoretical analysis of Mamba and attention mechanisms is a significant contribution, it is primarily focused on their application to spatial-temporal forecasting. The broader applicability of these principles to other domains, such as genomics or social networks, remains an area for further theoretical exploration. The current proof of complementarity is specific to the ResNet-Mamba structure.
- **Hyperparameter Sensitivity:** The proposed architectures, especially the fusion models, introduce a new set of hyperparameters related to the synchronization and fusion of different architectural components. The current work provides an empirical evaluation but does not offer a comprehensive theoretical guide for hyperparameter selection, which may require significant tuning for new datasets or network types.

7.1.1 Directions for Future Research

Based on the identified limitations, the following are promising directions for future research.

7.2 Future Research Directions

Based on the identified limitations, the following are promising directions for future research.

7.2.0.1 Enhancing Dynamic Network and Contextual Awareness

- **Dynamic Graph Learning:** Future work could explore incorporating dynamic graph learning mechanisms that can adapt the network topology in real-time. This would allow the model to dynamically update its understanding of the network based on events like road closures or traffic incidents. For example, a model could learn to "rewire" its connections to bypass a newly identified point of congestion.
- **Integrating Exogenous Data Streams:** A key area for expansion is the development of a fusion framework that can ingest and dynamically integrate a wide range of real-time exogenous data, such as weather patterns, holiday schedules, special event calendars, and even news reports. This would enable the model to make more robust and contextually aware predictions.

7.2.0.2 Broadening Architectural Synergy

- **Generalized Synergy Framework:** Future research should aim to develop a more generalized theoretical framework for architectural synergy that extends beyond state-space and attention models. This could involve exploring the complementarity between other types of architectures, such as Graph Neural Networks (GNNs), to create even more powerful and efficient hybrid models. This could lead to a deeper understanding of which architectural components are best suited for different types of data and forecasting tasks.

- **Automated Fusion and Hyperparameter Optimization:** To reduce the manual effort required for model deployment, future work could focus on developing automated machine learning (AutoML) techniques to learn the optimal fusion strategies and hyperparameter settings for new datasets. This would make the models more accessible and easier to implement in real-world scenarios.

7.2.0.3 Expanding Multimodal Generalization and Interaction

- **Cross-Modal Dependency Modeling:** A crucial next step is to move from predicting demand for multiple modes to explicitly modeling the interdependencies between them. Future research could develop a multi-task learning framework that not only predicts demand for buses, trains, and ferries but also learns the substitution and complementarity effects between these modes.
- **User-Centric Mobility Forecasting:** The current research focuses on forecasting system-level flows. Future work could shift to a more granular, user-centric approach, where models predict individual traveler behavior. This would require integrating richer, anonymized user profile data to create personalized travel forecasts and recommendations, which could support more sophisticated Mobility-as-a-Service (MaaS) platforms.

In conclusion, this thesis underscores the importance of uniting advanced attention-based methods with selective state-space models to generate robust, efficient, and accurate traffic flow predictions. By effectively blending global and local features, our proposed approaches have contributed to bridging the gap between state-of-the-art predictive performance and the pressing need for computationally viable solutions in large-scale, real-world intelligent transportation systems.

Appendix A

Appendix of Chapter 4

A1 The Generalized Discretization in State Space Model

To derive the discrete form, the solution to the state equation over the interval $[t_a, t_b]$ is given as:

$$\mathbf{h}(t_b) = e^{\mathbf{A}(t_b - t_a)}\mathbf{h}(t_a) + \int_{t_a}^{t_b} e^{\mathbf{A}(t_b - M)}\mathbf{B}\mathbf{x}(M)\, dM. \tag{A.1}$$

Here, $e^{\mathbf{A}(t_b - t_a)}$ serves as the state transition matrix, which maps the state from t_a to t_b when no input is applied. The integral term captures the influence of the input vector $\mathbf{x}(M)$, weighted by the state transition matrix and the input matrix $\mathbf{B}$. In discrete time, the state at step t_b can be written as:

$$\mathbf{h}_{t_b} = e^{\mathbf{A}(\Delta_a + \cdots + \Delta_{b-1})}\left(\mathbf{h}_{t_a} + \sum_{i=a}^{b-1} \mathbf{B}_i \mathbf{x}_i e^{\mathbf{A}(\Delta_{b-1} - \Delta_i)}\Delta_i\right), \tag{A.2}$$

where the accumulated state transition $e^{\mathbf{A}(\Delta_a + \cdots + \Delta_{b-1})}$ propagates the state, and the summation term accounts for the input contributions at each discrete step i. For the immediate next step, the state simplifies to:

$$\mathbf{h}_{t_{a+1}} = e^{\mathbf{A}\Delta_{t_a}}\left(\mathbf{h}_{t_a} + \mathbf{B}_{t_a}\mathbf{x}_{t_a}\Delta_a\right), \tag{A.3}$$

showing the transition from $\mathbf{h}_{t_a}$ to $\mathbf{h}_{t_{a+1}}$ by applying the state transition matrix over Δ_{t_a} and adding the weighted input contribution during that time step.

A2 Implementation Details of Graph-Based Models

We provide detailed information about the implementation of graph-based models used in our experiments with geo-referenced grid data containing longitude and latitude coordinates.

A2.1 Graph construction

For all graph-based models, we construct the initial network using the geographical coordinates of our grid cells, where each node represents a location with specific longitude and latitude values. The Haversine formula is used to calculate distances between coordinates:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right), \tag{A.4}$$

where r is the radius of the Earth (approximately 6,371 km), ϕ_1, ϕ_2 are the latitudes of points 1 and 2 in radians, and λ_1, λ_2 are the longitudes of points 1 and 2 in radians.

A2.2 Implementation Details

A2.2.1 Adjacency Matrix Construction

For models requiring an explicit adjacency matrix, we implement the following two methods: distance-based adjacency and thresholded adjacency.

We calculated the weighted adjacency matrix A where each element a_{ij} represents the normalized inverse distance between nodes i and j:

$$a_{ij} = \begin{cases} \exp(-\frac{d_{ij}^2}{\sigma^2}), & \text{if } d_{ij} \leq \delta, \\ 0, & \text{otherwise,} \end{cases} \tag{A.5}$$

TABLE A.1. Implementation details for graph-based models using geographical coordinates

Model	Graph Construction Approach	Parameter Settings
GWNet	• Distance-based adjacency matrix using Haversine formula • Adaptive adjacency matrix to learn dependencies beyond geographical proximity	• Diffusion steps $K = 2$ • Dilation exponential = 2 • Filter channels = 32 • Learning rate = 0.001 • Dropout rate = 0.3 • Adaptive adjacency threshold $\epsilon = 0.1$
DCRNN	• Directed graph using geographical distances • Edge weights: $w_{ij} = \exp(-\frac{d_{ij}^2}{\sigma^2})$ • d_{ij} is the Haversine distance between coordinates	• Diffusion steps $K = 2$ • Hidden units = 64 • Learning rate = 0.01 • Max. diffusion step = 2 • Batch size = 64 • Scheduled sampling = 0.05
AGCRN	• Initialized with distance-based adjacency matrix • Node-adaptive module to learn location-specific patterns	• Embedding dim = 10 • Hidden units = 64 • Learning rate = 0.003 • Dropout = 0.5 • Node embedding dim = 10 • Cheb polynomial order $K = 2$
STGCN	• Thresholded distance-based graph • Connections between nodes within a geographical radius	• Temporal kernel size = 3 • Spatial kernel size = 3 • Channels = [64, 16, 64] • Dropout = 0.5 • Learning rate = 0.001 • Batch size = 50
GTS	• Geographical embedding (lon/lat) as node features • Graph structure learning to discover relationships beyond physical proximity	• Hidden dim = 32 • Embedding dim = 64 • Graph learning layers = 2 • Temporal attention heads = 4 • Learning rate = 0.001 • Dropout = 0.3
MTGNN	• Graph learning module • Geographical distance as prior knowledge • Balances learned dependencies and physical constraints	• Hidden dim = 32 • Embedding dim = 16 • Graph learning layers = 1 • Dilation exponential = 1 • Learning rate = 0.001 • Subgraph size = 20
GMAN	• Spatial embeddings derived from coordinates • Multi-attention mechanism • Captures distance-based and dynamic temporal correlations	• Attention heads = 8 • Attention blocks $L = 3$ • Embedding dim = 64 • Spatial embedding dim = 10 • Temporal embedding dim = 10 • Learning rate = 0.001

where δ is the distance threshold (set to the 90th percentile of all pairwise distances), and σ^2 is the variance of distances (set to the average of squared distances).

For models like STGCN, we applied a binary threshold:

$$a_{ij} = \begin{cases} 1, & \text{if } d_{ij} \leq \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{A.6}$$

A2.2.2 Node Feature Representations

For geographical data, we improved node features by incorporating Normalized coordinates: We normalized longitude and latitude to the range [0,1] across the spatial domain. Positional encodings: For models supporting positional encodings (e.g., GMAN), we used sinusoidal encodings of the form:

$$PE(pos, 2i) = \sin(pos/10000^{2i/d_{model}}), \tag{A.7}$$

$$PE(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}}), \tag{A.8}$$

where pos represents the normalized coordinate value and i is the dimension index.

Geographical embeddings: For GTS and other models, we created learnable embeddings initialized with coordinate values, allowing the model to adapt and refine the spatial representations during training.

A2.2.3 Model-specific Adaptations

1) Graph WaveNet: We utilized the adaptive dependency matrix E learned by GWNet in conjunction with the predefined adjacency matrix A. The effective adjacency is computed as $\hat{A} = \text{SoftMax}(ReLU(E)) + A$.

2) DCRNN: We implemented the bidirectional diffusion convolution process, where both the

incoming and outgoing neighborhood information are considered, resulting in two transition matrices $D_O^{-1}A$ and $D_I^{-1}A^T$, where D_O and D_I are the out-degree and in-degree diagonal matrices.

3) AGCRN: We initialized node embeddings using a combination of normalized coordinates and learned parameters. The node-specific filter parameters are generated through a mapping function $f_\theta(E_i)$, where E_i represents the embedding for node i.

4) MTGNN: We implemented the mix-hop propagation with a k-hop neighborhood sampling method, where k is set to 3 to capture more extensive spatial dependencies.

Table A.1 displays the implementation details for graph-based models using geographical coordinates.

Appendix B

Appendix of Chapter 5

B1 Processing of the Weather and Public Holiday Data

Table B.1 summarizes the raw data collected from the Global Weather Data API [6] for each hourly observation, and the Arrival-to-API latency is below 10s; all records are in local time (AEST). After exploratory correlation analysis, we retained the most predictive variables and derived four composite indicators that capture extreme meteorological events.

Table B.1. Features of raw weather data (hourly resolution).

`name`, `datetime`, `tempmax`, `tempmin`, `temp`, `feelslikemax`, `feelslikemin`, `feelslike`, `dew`, `humidity`, `precip`, `precipprob`, `precipcover`, `preciptype`, `snow`, `snowdepth`, `windgust`, `windspeed`, `winddir`, `sealevelpressure`, `cloudcover`, `visibility`, `solarradiation`, `solarenergy`, `uvindex`, `severerisk`, `sunrise`, `sunset`, `moonphase`, `conditions`, `description`, `icon`, `stations`

We then select `temp`, `humidity`, `precip`, `windgust`, `winddir`, `sealevelpressure`, `cloudcover`, and `visibility` as continuous inputs (all z-scored). Four binary meta-features are created:

- `RainIntensity` is 1 if `precip` $\geq$ 10mm h^{-1} **or** `windgust` $\geq$ 40km h^{-1}; 0 otherwise.
- `StormRisk` is 1 if `severerisk` $>$ 25 (Visual Crossing scale); 0 otherwise.
- `LowVisibility` is 1 if `visibility` $<$ 5 km; 0 otherwise.
- `HolidayFlag` is 1 for QLD public holidays and network-wide incident shutdowns; 0 otherwise.

Missing weather entries ($<$ 0.8 % of rows) are linearly interpolated; categorical `conditions` strings (e.g. "Rain, Partially cloudy") are one-hot encoded but only retained if their frequency exceeded 2 % of the month (otherwise grouped into an "Other" bucket). All engineered weather

[6] Visual crossing: https://www.visualcrossing.com/

features are concatenated to the rail input tensor $\mathbf{X}_t^m \in \mathbb{R}^{d+F}$, where $F = 16$ (8 continuous + 4 binary + 4 one-hot). Stations inherit the weather vector of the nearest grid point (great-circle distance < 8 km); sensitivity analysis showed no performance change when using inverse-distance weighting of the three nearest points.

B2 Comparison of the Performance on Datasets in the USA

Table B.2 presents the comparative performance of STDAtt-Mamba and various baseline models on four widely used Performance Measurement System (PeMS) datasets from the United States.

TABLE B.2. Comparative performance analysis of models on PEMS datasets in USA.

Model	PEMS03			PEMS04			PEMS07			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HI	32.62	49.89	30.60%	42.35	61.66	29.92%	49.03	71.18	22.75%	36.66	50.45	21.63%
GWNet	**14.59**	**25.24**	15.52%	18.53	29.92	12.89%	20.47	33.47	8.61%	14.40	23.39	9.21%
DCRNN	15.54	27.18	15.62%	19.63	31.26	13.59%	21.16	34.14	9.02%	15.22	24.17	10.21%
AGCRN	15.24	26.65	15.89%	19.38	31.25	13.40%	20.57	34.40	8.74%	15.32	24.41	10.03%
STGCN	15.83	27.51	16.13%	19.57	31.38	13.44%	21.74	35.27	9.24%	16.08	25.39	10.60%
GTS	15.41	26.15	15.39%	20.96	32.95	14.66%	22.15	35.10	9.38%	16.49	26.08	10.54%
MTGNN	14.85	25.23	14.55%	19.17	31.70	13.37%	20.89	34.06	9.00%	15.18	24.24	10.20%
STNorm	15.32	25.93	14.37%	18.96	30.98	12.69%	20.50	34.66	8.75%	15.41	24.77	9.76%
GMAN	16.87	27.92	18.23%	19.14	31.60	13.19%	20.97	34.10	9.05%	15.31	24.92	10.13%
PDFormer	14.94	25.39	15.82%	18.36	30.03	12.00%	19.97	32.95	8.55%	13.58	23.41	9.05%
STID	15.33	27.40	16.40%	18.38	29.95	12.04%	19.61	32.79	8.30%	14.21	23.28	9.27%
STAEformer	15.35	27.55	15.18%	18.22	30.18	11.98%	19.14	32.60	**8.01%**	13.46	23.25	**8.88%**
STDAtt-Mamba	15.28	27.32	**15.17%**	**18.16**	**29.73**	**12.01%**	**19.12**	**32.41**	7.98%	**13.26**	**23.13**	8.82%

Note: PEMS03, PEMS04, PEMS07, and PEMS08 are four datasets constructed from four areas in California, USA. All the data are collected from the Caltrans Performance Measurement System (PeMS).

Analysis of performance on different datasets. The experimental results reveal distinct performance patterns across the PEMS datasets that correlate with their underlying characteristics. STDAtt-Mamba demonstrates competitive performance on PEMS04, PEMS07, and PEMS08, achieving state-of-the-art results in multiple metrics. However, the model shows relatively weaker performance on PEMS03, where GWNet maintains the best MAE and RMSE scores. This performance variation can be attributed to fundamental differences in dataset complexity and traffic pattern characteristics. Statistical analysis reveals that PEMS03 exhibits significantly

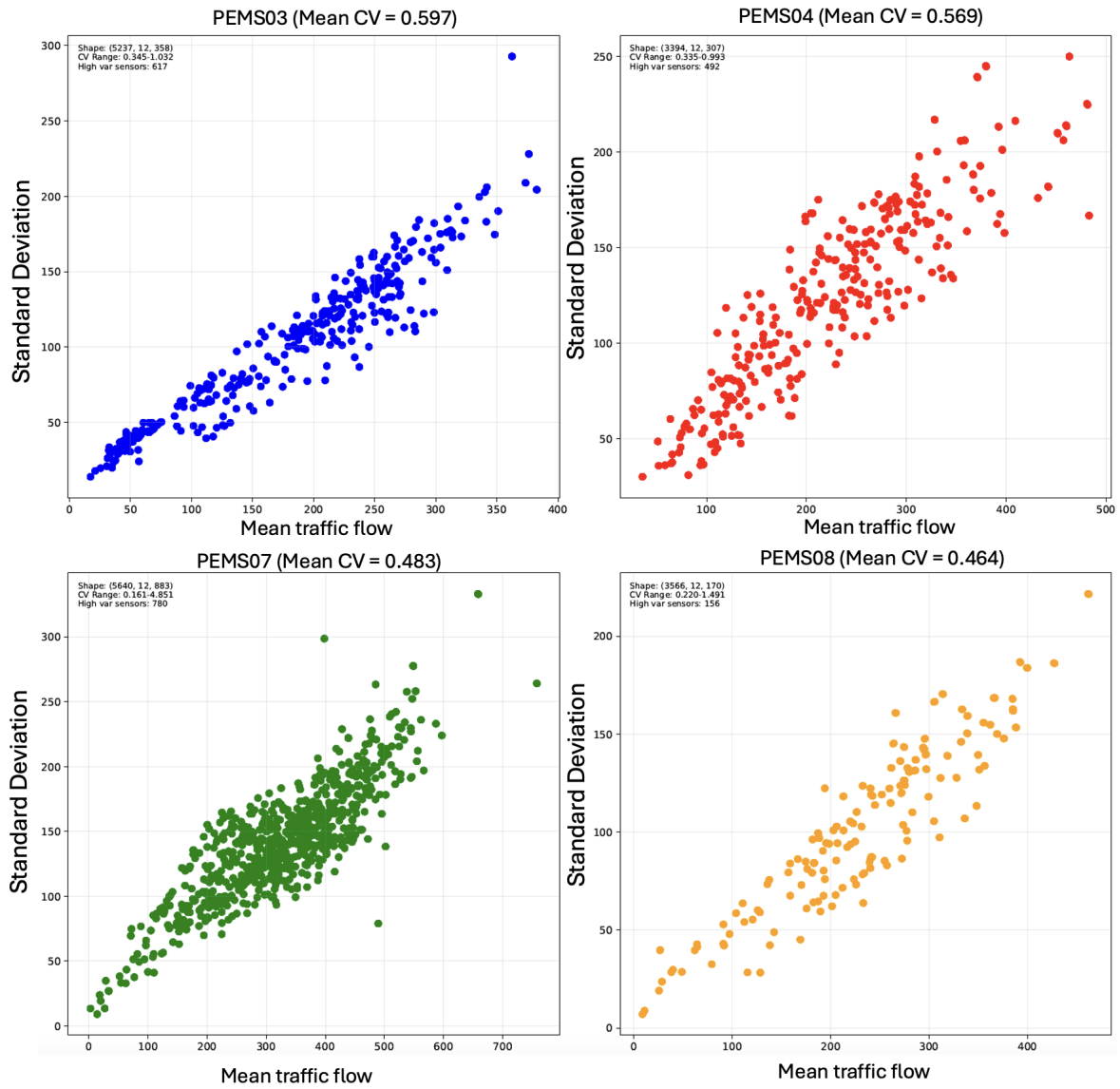


FIGURE B.1. Visualization of the Mean-Standard Deviation relationship for traffic flow across different PEMS datasets. The PEMS03 dataset (blue dots) demonstrates a more dispersed and non-linear pattern, with a higher coefficient of variation (Mean CV = 0.597) compared to other datasets like PEMS08 (orange dots).

higher complexity compared to other datasets. As illustrated in Figure B.1, PEMS03 demonstrates the highest Mean Coefficient of Variation (CV = 0.597) with a wider CV range (0.345-1.012), indicating substantially more volatile traffic patterns. The dataset contains 617 high-variance sensors, notably more than PEMS04 (491 sensors) and PEMS08 (156 sensors), suggesting that a larger proportion of monitoring locations experience unpredictable traffic fluctuations. The Mean-Standard Deviation relationship visualization shows that PEMS03 exhibits more dispersed

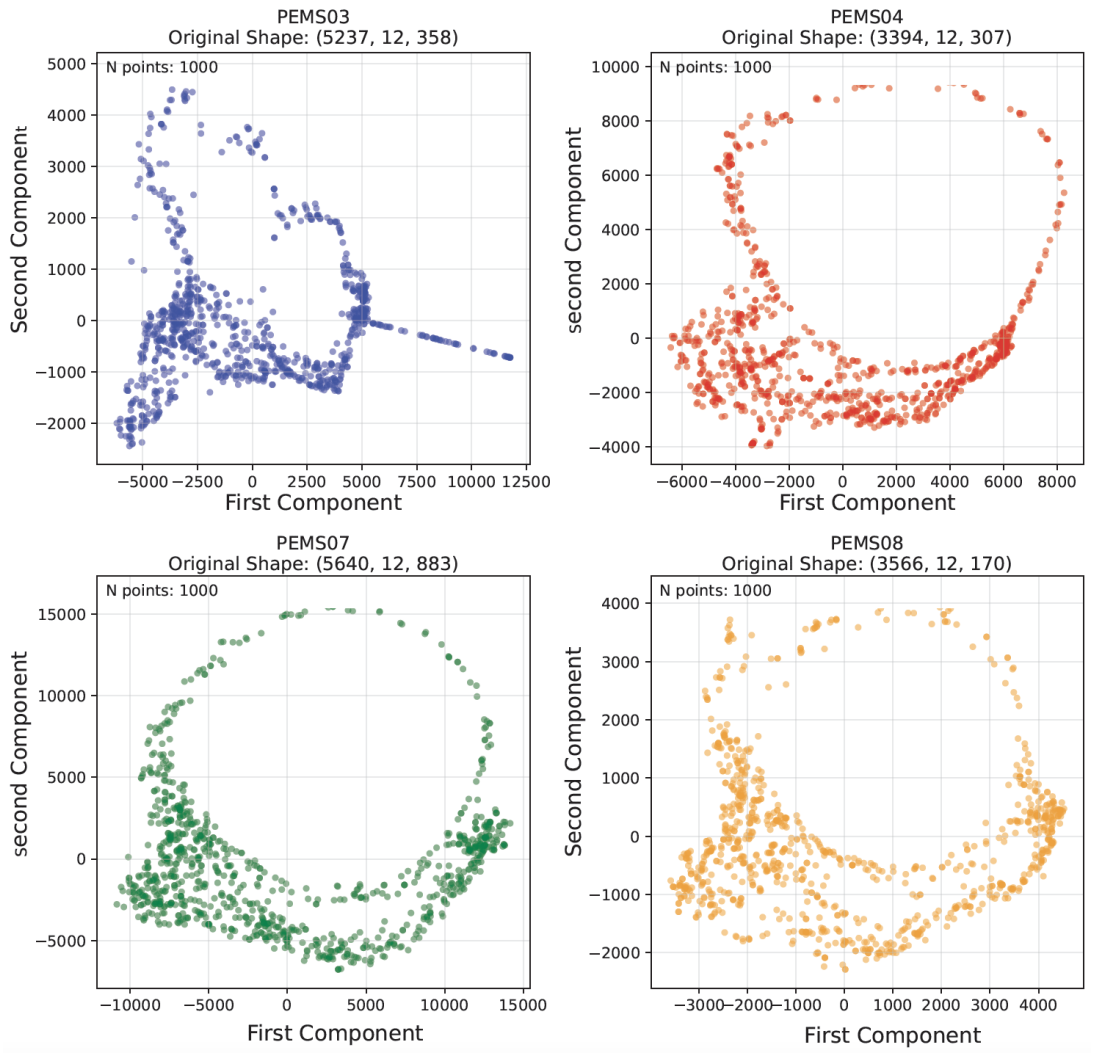


FIGURE B.2. Isomap visualization comparing traffic pattern distributions across PEMS datasets. PEMS03's fragmented structure (top-left) contrasts with the more continuous, organized patterns of other datasets.

and non-linear patterns compared to the concentrated distributions observed in PEMS07 and PEMS08.

Figure B.2 further reveals structural differences between datasets. PEMS03 displays a distinctly fragmented and scattered distribution pattern with multiple isolated clusters and discontinuous regions, contrasting with the smooth, continuous manifold structures observed in other datasets. This fragmented topology indicates that traffic patterns in PEMS03 undergo more abrupt changes and follow less predictable evolutionary paths, creating inherent challenges for spatiotemporal forecasting models. These dataset characteristics explain why both STDAtt-Mamba and several baseline models experience performance degradation on PEMS03. The combination of higher statistical variability, irregular pattern relationships, and fragmented temporal evolution creates a more challenging prediction environment requiring models to capture sudden transitions and discontinuous pattern changes, which prove difficult for current spatiotemporal architectures. Despite these challenges, STDAtt-Mamba maintains competitive prediction performance on PEMS03 while achieving superior results across other datasets, demonstrating its robustness and effectiveness.

Acknowledgments

We acknowledge the invaluable support of the Department of Transport and Main Roads, Queensland (TMR), Queensland Government, Australia, for their invaluable support, particularly to Mr. Paul Scott, Research Manager from TMR, for facilitating the data transfer agreement of this study. The findings reported are those of the authors and are not necessarily the positions of TMR; but approval to present these findings is appreciated. The authors wish to thank the Editor-in-Chief, Guest Editors, and four anonymous reviewers, whose useful comments have significantly improved the chapter.

Bibliography

Mohammed S. Ahmed and Allen R. Cook. Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques. *Transportation Research Record*, 1979-2(722):1–9, 1979. URL https://trid.trb.org/View/148123.

Ishteaque Alam, Dewan Md Farid, and Rosaldo JF Rossetti. The prediction of traffic flow with regression analysis. In *Proceedings of Emerging Technologies in Data Mining and Information Security (IEMIS)*, volume 2, pages 661–671. Springer, 2019. doi: 10.1007/978-981-13-1498-8_58.

Muhammad Arif, Guojun Wang, and Shuhong Chen. Deep learning with non-parametric regression model for traffic flow prediction. In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 681–688, 2018. doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00120.

Nurul A. Asif, Yeahia Sarker, Ripon K. Chakrabortty, Michael J. Ryan, Md. Hafiz Ahamed, Dip K. Saha, Faisal R. Badal, Sajal K. Das, Md. Firoz Ali, Sumaya I. Moyeen, Md. Robiul Islam, and Zinat Tasneem. Graph Neural Network: A Comprehensive Review on Non-Euclidean Space. *IEEE Access*, 9, 2021. doi: 10.1109/ACCESS.2021.3071274. URL 10.1109/ACCESS.2021.3071274.

LEI BAI, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17804–17815. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ce1aad92b939420fc17005e5461e6f48-Paper.pdf.

Thomas Bapaume, Etienne Côme, Mostafa Ameli, Jérémy Roos, and Latifa Oukhellou. Forecasting passenger flows and headway at train level for a public transport line: Focus on atypical

situations. *Transportation Research Part C: Emerging Technologies*, 153:104195, 2023.

Asma Belhadi, Youcef Djenouri, Djamel Djenouri, and Jerry Chun-Wei Lin. A recurrent neural network for urban long-term traffic flow forecasting. *Applied Intelligence*, 50:3252–3265, 2020. doi: 10.1007/s10489-020-01716-1.

Toon Bogaerts, Antonio D Masegosa, Juan S Angarita-Zapata, Enrique Onieva, and Peter Hellinckx. A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data. *Transportation Research Part C: Emerging Technologies*, 112:62–77, 2020. doi: 10.1016/j.trc.2020.01.010.

George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA, revised edition edition, 1976. ISBN 978-0816211043.

George EP Box and Gwilym M Jenkins. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

Ling Cai, Krzysztof Janowicz, Gengchen Mai, Bo Yan, and Rui Zhu. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3):736–755, 2020a. doi: 10.1111/tgis.12644. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12644.

Lingru Cai, Yidan Yu, Shuangyi Zhang, Youyi Song, Zhi Xiong, and Teng Zhou. A Sample-Rebalanced Outlier-Rejected k-Nearest Neighbor Regression Model for Short-Term Traffic Flow Forecasting. *IEEE Access*, 8:22686–22696, 2020b. doi: 10.1109/ACCESS.2020.2970250. URL 10.1109/ACCESS.2020.2970250.

Chenyi Chen, Jianming Hu, Qiang Meng, and Yi Zhang. Short-time traffic flow prediction with arima-garch model. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 607–612, 2011. doi: 10.1109/IVS.2011.5940418.

Jian Chen, Wei Wang, Keping Yu, Xiping Hu, Ming Cai, and Mohsen Guizani. Node connection strength matrix-based graph convolution network for traffic flow prediction. *IEEE Transactions on Vehicular Technology*, 72(9):12063–12074, 2023. doi: 10.1109/TVT.2023.3265300.

Jian Chen, Li Zheng, Yuzhu Hu, Wei Wang, Hongxing Zhang, and Xiping Hu. Traffic Flow Matrix-Based Graph Neural Network with Attention Mechanism for Traffic Flow Prediction. *Information Fusion*, 104, April 2024. doi: 10.1016/j.inffus.2023.102146. URL

https://10.1016/j.inffus.2023.102146.

Zhijun Chen, Zhe Lu, Qiushi Chen, Hongliang Zhong, Yishi Zhang, Jie Xue, and Chaozhong Wu. Spatial-temporal short-term traffic flow prediction model based on dynamical-learning graph convolution mechanism. *Information Sciences*, 611:522–539, 2022. doi: 10.1016/j.ins.2022.08.080. URL https://www.sciencedirect.com/science/article/pii/S0020025522009902.

Jeongwhan Choi and Noseong Park. Graph neural rough differential equations for traffic forecasting. *ACM Trans. Intell. Syst. Technol.*, 14(4), July 2023. ISSN 2157-6904. doi: 10.1145/3604808. URL 10.1145/3604808.

Yun Young Choi, Minho Lee, Sun Woo Park, Seunghwan Lee, and Joohwan Ko. A gated mlp architecture for learning topological dependencies in spatio-temporal graphs, 2024. URL https://arxiv.org/abs/2401.15894.

Achituv Cohen and Sagi Dalyot. Pedestrian traffic flow prediction based on ANN model and OSM data. In *Proceedings of the International Cartographic Association*, volume 2, pages 20–27. Copernicus Publications Göttingen, Germany, 2019. doi: 10.5194/ica-proc-2-20-2019.

Yue Cui, Jiandong Xie, and Kai Zheng. Historical inertia: A neglected but powerful baseline for long sequence time-series forecasting. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2965–2969. ACM, 2021. doi: 10.1145/3459637.3482131.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang. St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 269–278, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467330. URL 10.1145/3447548.3467330.

Erdem Doğan. Robust-lstm: a novel approach to short-traffic flow prediction based on signal decomposition. *Soft Computing*, 26(11):5227–5239, 2022. doi: 10.1007/s00500-022-07023-w.

H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems*, pages 155–161, 1997a.

Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, 9:

155–161, 1997b.

Yanjie Duan, Yisheng Lv, Yu-Liang Liu, and Fei-Yue Wang. An efficient realization of deep learning for traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 72:168–181, 2016. ISSN 0968-090X. doi: 10.1016/j.trc.2016.09.015. URL https://www.sciencedirect.com/science/article/pii/S0968090X16301826.

Azadeh Emami, Majid Sarvi, and Saeed Asadi Bagloee. Using kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment. *Journal of Modern Transportation*, 27:222–232, 2019. doi: 10.1007/s40534-019-0193-2.

Azadeh Emami, Majid Sarvi, and Saeed Asadi Bagloee. Short-term traffic flow prediction based on faded memory Kalman filter fusing data from connected vehicles and bluetooth sensors. *Simulation Modelling Practice and Theory*, 102:102025, 2020. doi: 10.1016/j.simpat.2019.102025. URL https://www.sciencedirect.com/science/article/pii/S1569190X1930156X. Special Issue on IoT, Cloud, Big Data and AI in Interdisciplinary Domains.

Rui Fu, Zuo Zhang, and Li Li. Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 324–328, 2016. doi: 10.1109/YAC.2016.7804912.

Jun Wei Gao, Zi Wen Leng, Bin Zhang, Xin Liu, and Guo Qiang Cai. The application of adaptive Kalman filter in traffic flow forecasting. *Advanced Materials Research*, 680:495–500, 2013. doi: 10.4028/www.scientific.net/AMR.680.495.

Zili Geng, Jie Xu, Rongsen Wu, Changming Zhao, Jin Wang, Yunji Li, and Chenlin Zhang. Stgaformer: Spatial,Äìtemporal gated attention transformer based graph neural network for traffic flow forecasting. *Information Fusion*, 105:102228, 2024. ISSN 1566-2535. doi: 10.1016/j.inffus.2024.102228. URL https://www.sciencedirect.com/science/article/pii/S156625352400006X.

Mustafa Ghaderzadeh, Armin Shalchian, Gholamreza Irajian, Hamidreza Sadeghsalehi, Abed Zahedi Bialvaei, and Babak Sabet. Artificial intelligence in drug discovery and development against antimicrobial resistance: A narrative review. *Iranian Journal of Medical Microbiology*, 18(3), 2024. doi: 10.30699/ijmm.18.3.135. URL http://ijmm.ir/article-1-2384-en.html.

Bernardo Gomes, José Coelho, and Helena Aidos. A survey on traffic flow prediction and classification. *Intelligent Systems with Applications*, 20:200268, 2023. ISSN 2667-3053. doi: 10.1016/j.iswa.2023.200268.

Ian Goodfellow. Deep learning, 2016.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, 2023. URL 10.48550/arXiv.2312.00752. arXiv:2312.00752.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent Memory with Optimal Polynomial Projections . In *Advances in Neural Information Processing Systems*, volume 33, pages 1474–1487, 2020. URL https://tinyurl.com/cstvh62s.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *CoRR*, abs/2111.00396, 2021. URL https://arxiv.org/abs/2111.00396.

Jianhua Guo, Wei Huang, and Billy Williams. Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, 43, 06 2014. doi: 10.1016/j.trc.2014.02.006.

Kan Guo, Yongli Hu, Zhen Qian, Hao Liu, Ke Zhang, Yanfeng Sun, Junbin Gao, and Baocai Yin. Optimized graph convolution recurrent neural network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):1138–1149, 2021. doi: 10.1109/TITS.2019.2963722.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1):922–929, Jul. 2019. doi: 10.1609/aaai.v33i01.3301922. URL https://ojs.aaai.org/index.php/AAAI/article/view/3881.

J. D. Hamilton. *Time Series Analysis*, volume 2. Princeton University Press, Princeton, NJ, 1994.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Yuxin He, Lishuai Li, Xinting Zhu, and Kwok Leung Tsui. Multi-graph convolutional-recurrent neural network (mgc-rnn) for short-term forecasting of transit passenger flow. *IEEE transactions on intelligent transportation systems*, 23(10):18155–18174, 2022.

Sepp Hochreiter. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116, April 1998. doi: 10.1142/S0218488598000094. URL 10.1142/S0218488598000094.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Songhua Hu, Jianhua Chen, Wei Zhang, Guanhua Liu, and Ximing Chang. Graph transformer embedded deep learning for short-term passenger flow prediction in urban rail transit systems: A multi-gate mixture-of-experts model. *Information Sciences*, 679:121095, 2024.

Feihu Huang, Peiyu Yi, Jince Wang, Mengshi Li, Jian Peng, and Xi Xiong. A dynamical spatial-temporal graph neural network for traffic demand prediction. *Information Sciences*, 594:286–304, 2022.

Saman Jamshidi, Mahin Mohammadi, Saeed Bagheri, Hamid Esmaeili Najafabadi, Alireza Rezvanian, Mehdi Gheisari, Mustafa Ghaderzadeh, Amir Shahab Shahabi, and Zongda Wu. Effective text classification using bert, mtm lstm, and dt. *Data & Knowledge Engineering*, 151:102306, 2024. ISSN 0169-023X. doi: 10.1016/j.datak.2024.102306. URL https://www.sciencedirect.com/science/article/pii/S0169023X24000302.

Young-Seon Jeong, Young-Ji Byon, Manoel Mendonca Castro-Neto, and Said M. Easa. Supervised weighting-online learning algorithm for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1700–1707, 2013. doi: 10.1109/TITS.2013.2267735.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):4365–4373, Jun. 2023. doi: 10.1609/aaai.v37i4.25556.

Wenhua Jiang, Zhenliang Ma, and Haris N Koutsopoulos. Deep learning for short-term origin–destination passenger flow prediction under partial observability in urban railway systems. *Neural Computing and Applications*, pages 1–18, 2022.

Wei Ju, Yusheng Zhao, Yifang Qin, Siyu Yi, Jingyang Yuan, Zhiping Xiao, Xiao Luo, Xiting Yan, and Ming Zhang. Cool: A conjoint perspective on spatio-temporal graph neural

network for traffic forecasting. *Information Fusion*, 107:102341, 2024. ISSN 1566-2535. doi: 10.1016/j.inffus.2024.102341. URL https://www.sciencedirect.com/science/article/pii/S1566253524001192.

Anirudh Ameya Kashyap, Raviraj Shravan, Ananya Devarakonda, R Shamanth, K Nayak, K V Santhosh, and J Bhat Soumya. Traffic flow prediction models – a review of deep learning techniques. *Cogent Engineering*, 9(1):2010510, 2022. doi: 10.1080/23311916.2021.2010510.

Jintao Ke, Hongyu Zheng, Hai Yang, and Xiqun (Michael) Chen. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies*, 85:591–608, 2017. ISSN 0968-090X. doi: 10.1016/j.trc.2017.10.016. URL https://www.sciencedirect.com/science/article/pii/S0968090X17302899.

Taehyung Kim, Hyoungsoo Kim, and D.J. Lovell. Traffic flow forecasting: overcoming memoryless property in nearest neighbor non-parametric regression. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pages 965–969, 2005. doi: 10.1109/ITSC.2005.1520181.

Jianlei Kong, Xiaomeng Fan, Xuebo Jin, Sen Lin, and Min Zuo. A variational bayesian inference-based en-decoder framework for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 25(3):2966–2975, 2024a. doi: 10.1109/TITS.2023.3276216.

Jianlei Kong, Xiaomeng Fan, Min Zuo, Muhammet Deveci, Xuebo Jin, and Kaiyang Zhong. Adct-net: Adaptive traffic forecasting neural network via dual-graphic cross-fused transformer. *Information Fusion*, 103:102122, 2024b. ISSN 1566-2535. doi: 10.1016/j.inffus.2023.102122. URL https://www.sciencedirect.com/science/article/pii/S1566253523004384.

S. Vasantha Kumar and Lelitha Vanajakshi. Short-Term Traffic Flow Prediction Using Seasonal Arima Model with Limited Input Data. *European Transport Research Review*, 7(3):21, September 2015. doi: 10.1007/s12544-015-0170-8. URL 10.1007/s12544-015-0170-8.

Selvaraj Vasantha Kumar. Traffic flow prediction using Kalman filtering technique. *Procedia Engineering*, 187:582–587, 2017. doi: 10.1016/j.proeng.2017.04.417.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Can Li, Lei Bai, Wei Liu, Lina Yao, and S Travis Waller. Graph neural network for robust public transit demand prediction. *IEEE Transactions on Intelligent Transportation Systems*, 23(5): 4086–4098, 2020.

Can Li, Lei Bai, Wei Liu, Lina Yao, and S Travis Waller. A multi-task memory network with knowledge adaptation for multimodal demand forecasting. *Transportation Research Part C: Emerging Technologies*, 131:103352, 2021a.

Junyi Li, Fangce Guo, Aruna Sivakumar, Yanjie Dong, and Rajesh Krishnan. Transferability Improvement in Short-Term Traffic Prediction Using Stacked LSTM Network. *Transportation Research Part C: Emerging Technologies*, 124, 2021b. doi: 10.1016/j.trc.2021.102977. URL 10.1016/j.trc.2021.102977.

Pei Li, Sheng Wang, Hantao Zhao, Jia Yu, Liyang Hu, Haodong Yin, and Zhiyuan Liu. Ig-net: An interaction graph network model for metro passenger flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):4147–4157, 2023.

Shuangshuang Li, Zhen Shen, and Gang Xiong. A k-nearest neighbor locally weighted regression method for short-term traffic flow forecasting. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1596–1601, 2012. doi: 10.1109/ITSC.2012.6338648.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

Yiqun Li, Songjian Chai, Zhengwei Ma, and Guibin Wang. A hybrid deep learning framework for long-term traffic flow prediction. *IEEE Access*, 9:11264–11271, 2021c. doi: 10.1109/ACCESS.2021.3050836.

Zheyi Li and Wenxuan Zhu. Spatial-temporal fusion graph neural networks for traffic flow forecasting. *arXiv:2101.11174*, 2021.

Yuebing Liang, Guan Huang, and Zhan Zhao. Joint demand prediction for multimodal systems: A multi-task multi-relational spatiotemporal graph neural network approach. *Transportation research part C: emerging technologies*, 140:103731, 2022.

Haicheng Liao, Huanming Shen, Zhenning Li, Chengyue Wang, Guofa Li, Yiming Bie, and Chengzhong Xu. Gpt-4 enhanced multimodal grounding for autonomous driving: Leveraging cross-modal attention with large language models. *Communications in Transportation Research*, 4:100116, 2024. ISSN 2772-4247. doi: 10.1016/j.commtr.2023.100116. URL https://www.sciencedirect.com/science/article/pii/S2772424723000276.

Wenxie Lin, Zhe Zhang, Gang Ren, Yangzhen Zhao, Jingfeng Ma, and Qi Cao. Mgcn: Mamba-integrated spatiotemporal graph convolutional network for long-term traffic forecasting. *Knowledge-Based Systems*, page 112875, 2024.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Staeformer: Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting, 2023a.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 4125–4129, New York, NY, USA, 2023b. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3615160. URL 10.1145/3583780.3615160.

Lingbo Liu, Jiajie Zhen, Guanbin Li, Geng Zhan, Zhaocheng He, Bowen Du, and Liang Lin. Dynamic spatial-temporal representation learning for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):7169–7183, 2021a. doi: 10.1109/TITS.2020.3002718.

Lingbo Liu, Yuying Zhu, Guanbin Li, Ziyi Wu, Lei Bai, and Liang Lin. Online metro origin-destination prediction via heterogeneous information aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3574–3589, 2022.

Yang Liu, Cheng Lyu, Yuan Zhang, Zhiyuan Liu, Wenwu Yu, and Xiaobo Qu. Deeptsp: Deep traffic state prediction model based on large-scale empirical data. *Communications in Transportation Research*, 1:100012, 2021b. ISSN 2772-4247. doi: 10.1016/j.commtr.2021.100012. URL https://www.sciencedirect.com/science/article/pii/S2772424721000123.

Sohani Liyanage, Rusul Abduljabbar, Hussein Dia, and Pei-Wei Tsai. Ai-based neural network models for bus passenger demand forecasting using smart card data. *Journal of Urban*

Management, 11(3):365–380, 2022.

Dan Luo, Dong Zhao, Qixue Ke, Xiaoyong You, Liang Liu, Desheng Zhang, Huadong Ma, and Xingquan Zuo. Fine-grained service-level passenger flow prediction for bus transit systems based on multitask deep learning. *IEEE transactions on intelligent transportation systems*, 22 (11):7184–7199, 2020.

Xianglong Luo, Danyang Li, Yu Yang, and Shengrui Zhang. Spatiotemporal traffic flow prediction with KNN and LSTM. *Journal of Advanced Transportation*, 2019:4145353, 2019. doi: 10.1155/2019/4145353.

Yang Lv, Zhiqiang Lv, Zesheng Cheng, Zhanqi Zhu, and Taha Hossein Rashidi. Ts-stnn: Spatial-temporal neural network based on tree structure for traffic flow prediction. *Transportation research part E: logistics and transportation review*, 177:103251, 2023.

Changxi Ma, Guowen Dai, and Jibiao Zhou. Short-Term Traffic Flow Prediction for Urban Road Sections Based on Time Series Analysis and LSTM_BILSTM Method. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5615–5624, June 2022. doi: 10.1109/TITS.2021.3055258. URL 10.1109/TITS.2021.3055258.

Ying Ma, Haijie Lou, Ming Yan, Fanghui Sun, and Guoqi Li. Spatio-temporal fusion graph convolutional network for traffic flow forecasting. *Information Fusion*, 104:102196, 2024. ISSN 1566-2535. doi: 10.1016/j.inffus.2023.102196. URL https://www.sciencedirect.com/science/article/pii/S1566253523005122.

Manuel Méndez, Mercedes G Merayo, and Manuel Núñez. Long-term traffic flow forecasting using a hybrid cnn-bilstm model. *Engineering Applications of Artificial Intelligence*, 121: 106041, 2023. doi: 10.1016/j.engappai.2023.106041.

Diogo David Oliveira, Mariana Rampinelli, Gabriel Zago Tozatto, Rodrigo Varejão Andreão, and Sandra MT Müller. Forecasting vehicular traffic flow using MLP and LSTM. *Neural Computing and Applications*, 33:17245–17256, 2021. doi: 10.1007/s00521-021-06315-w.

Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 1720–1730, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330884. URL 10.1145/3292500.3330884.

Bagus Priambodo and Azlina Ahmad. Predicting traffic flow based on average speed of neighbouring road using multiple regression. In *Advances in Visual Informatics: 5th International Visual Informatics Conference (IVIC)*, volume 5, pages 309–318. Springer, 2017. doi: 10.1007/978-3-319-70010-6_29.

Hao Qiu, Jinlei Zhang, Lixing Yang, Kuo Han, Xiaobao Yang, and Ziyou Gao. Spatial–temporal multi-task learning for short-term passenger inflow and outflow prediction on holidays in urban rail transit systems. *Transportation*, pages 1–30, 2025.

Faysal Ibna Rahman. Short term traffic flow prediction using machine learning KNNs, SVM and ANN with weather information. *International Journal for Traffic & Transport Engineering*, 10(3):371, 2020. doi: 10.1007/s00521-021-06315-w.

Kadiyala Ramana, Gautam Srivastava, Madapuri Rudra Kumar, Thippa Reddy Gadekallu, Jerry Chun-Wei Lin, Mamoun Alazab, and Celestine Iwendi. A vision transformer approach for traffic congestion prediction in urban areas. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):3922–3934, 2023. doi: 10.1109/TITS.2022.3233801.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Internal Representations by Error Propagation. In David E. Rumelhart, James L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. MIT Press, 1986. doi: 10.7551/mitpress/4943.003.0128. URL 10.7551/mitpress/4943.003.0128.

Sayed A. Sayed, Yasser Abdel-Hamid, and Hesham Ahmed Hefny. Artificial intelligence-based traffic flow prediction: a comprehensive review. *Journal of Electrical Systems and Information Technology*, 10(1):13, 2023. ISSN 2314-7172. doi: 10.1186/s43067-023-00081-6. URL 10.1186/s43067-023-00081-6.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete Graph Structure Learning for Forecasting Multiple Time Series, 2021. URL https://openreview.net/forum?id=WEHSlH5mOk.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the*

31st ACM International Conference on Information & Knowledge Management, CIKM '22, page 4454–4458, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557702. URL 10.1145/3511808.3557702.

Zhiiqi Shao, Ze Wang, Xusheng Yao, Michael Bell, and Junbin Gao. ST-MambaSync: Complement the power of Mamba and transformer fusion for less computational cost in spatial-temporal traffic forecasting. *Information Fusion*, page 102872, 2024a. ISSN 1566-2535. URL https://www.sciencedirect.com/science/article/pii/S156625352400650X.

Zhiqi Shao, Dai Shi, Andi Han, Yi Guo, Qibin Zhao, and Junbin Gao. Unifying over-smoothing and over-squashing in graph neural networks: A physics informed approach and beyond, 2023.

Zhiqi Shao, Michael G. H. Bell, Ze Wang, D. Glenn Geers, Xusheng Yao, and Junbin Gao. Ccdsreformer: Traffic flow prediction with a criss-crossed dual-stream enhanced rectified transformer model. *arXiv preprint arXiv:2403.17753*, 2024b.

Zhiqi Shao, Michael G. H. Bell, Ze Wang, D. Glenn Geers, Xusheng Yao, and Junbin Gao. Ccdsreformer: Traffic flow prediction with a criss-crossed dual-stream enhanced rectified transformer model, 2024c. URL https://arxiv.org/abs/2403.17753.

Zhiqi Shao, Dai Shi, Andi Han, Andrey Vasnev, Yi Guo, and Junbin Gao. Enhancing framelet gcns with generalized p-laplacian regularization. *International Journal of Machine Learning and Cybernetics*, 15(4):1553–1573, 2024d. ISSN 1868-808X. doi: 10.1007/s13042-023-01982-8. URL 10.1007/s13042-023-01982-8.

Zhiqi Shao, Dai Shi, Andi Han, Andrey Vasnev, Yi Guo, and Junbin Gao. Enhancing framelet gcns with generalized p-laplacian regularization. *International Journal of Machine Learning and Cybernetics*, 15(4):1553–1573, 2024e. doi: 10.1007/s13042-023-01982-8.

Zhiqi Shao, Haoning Xi, Haohui Lu, Ze Wang, Michael G. H. Bell, and Junbin Gao. Stllm-df: A spatial-temporal large language model with diffusion for enhanced multi-mode traffic system forecasting, 2024f. URL https://arxiv.org/abs/2409.05921.

Zhiqi Shao, Haoning Xi, Haohui Lu, Ze Wang, Michael GH Bell, and Junbin Gao. Stllm-df: A spatial-temporal large language model with diffusion for enhanced multi-mode traffic system forecasting. *arXiv preprint arXiv:2409.05921*, 2024g.

Zhiqi Shao, Xusheng Yao, Ze Wang, and Junbin Gao. St-mambasync: The complement of mamba and transformers for spatial-temporal in traffic flow prediction, 2024h. URL

https://arxiv.org/abs/2404.15899.

Bharti Sharma, Sachin Kumar, Prayag Tiwari, Pranay Yadav, and Marina I Nezhurina. Ann based short-term traffic flow forecasting in undivided two lane highway. *Journal of Big Data*, 5(1):1–16, 2018. doi: 10.1186/s40537-018-0157-0.

Yuyol Shin and Yoonjin Yoon. Pgcn: Progressive graph convolutional networks for spatial–temporal traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 25 (7):7633–7644, 2024. doi: 10.1109/TITS.2024.3349565.

Christopher A Sims. Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, 48(1):1–48, 1980. doi: 10.2307/1912017. URL https://doi.org/10.2307/1912017.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):914–921, April 2020. doi: 10.1609/aaai.v34i01.5438. URL 10.1609/aaai.v34i01.5438.

Hongyang Su, Xiaolong Wang, Qingcai Chen, and Yang Qin. Efficient adaptive spatial-temporal attention network for traffic flow forecasting. In *Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Proceedings, Part V*, page 205–220, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-43423-5. doi: 10.1007/978-3-031-43424-2_13. URL 10.1007/978-3-031-43424-2_13.

Lijun Sun, Mingzhi Liu, Guanfeng Liu, Xiao Chen, and Xu Yu. Fd-tgcn: Fast and dynamic temporal graph convolution network for traffic flow prediction. *Information Fusion*, 106: 102291, 2024. ISSN 1566-2535. doi: 10.1016/j.inffus.2024.102291. URL https://www.sciencedirect.com/science/article/pii/S1566253524000691.

Tianli Tang, Ronghui Liu, Charisma Choudhury, Achille Fonzone, and Yuanyuan Wang. Predicting hourly boarding demand of bus passengers using imbalanced records from smartcards: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5105–5119, 2023.

Mingrong Tong and Hengxin Xue. Highway traffic volume forecasting based on seasonal ARIMA model. *Journal of Highway and Transportation Research and Development (English Edition)*,

3(2):109–112, 2008. doi: 10.1061/JHTRCQ.0000255.

Vedat Topuz. Hourly traffic flow prediction using different ANN models. In *Urban Transport and Hybrid Vehicles*, page 192. IntechOpen, 08 2010. ISBN 978-953-307-100-8. doi: 10.5772/10177.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. doi: 10.48550/arXiv.1706.03762. URL https://arxiv.org/abs/1706.03762.

Jingyuan Wang, Wenjun Jiang, and Jiawei Jiang. Libcity-dataset: a standardized and comprehensive dataset for urban spatial–temporal data mining. *Intelligent Transportation Infrastructure*, 2: liad021, 11 2023a. ISSN 2752-9991. doi: 10.1093/iti/liad021. URL 10.1093/iti/liad021.

Qingyi Wang, Shenhao Wang, Dingyi Zhuang, Haris Koutsopoulos, and Jinhua Zhao. Uncertainty quantification of spatiotemporal travel demand with probabilistic graph neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 2024a.

Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*, WWW '20, page 1082–1092, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380186. URL 10.1145/3366423.3380186.

Yi Wang, Changfeng Jing, Wei Huang, Shiyuan Jin, and Xinxin Lv. Adaptive spatiotemporal inceptionnet for traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):3882–3907, 2023b. doi: 10.1109/TITS.2023.3237205.

Zhenghong Wang, Yi Wang, Furong Jia, Fan Zhang, Nikita Klimenko, Leye Wang, Zhengbing He, Zhou Huang, and Yu Liu. Spatiotemporal fusion transformer for large-scale traffic forecasting. *Information Fusion*, 107:102293, 2024b. ISSN 1566-2535. doi: 10.1016/j.inffus.2024.102293. URL https://www.sciencedirect.com/science/article/pii/S156625352400071X.

Jinxin Wu, Xianwang Li, Deqiang He, Qin Li, and Weibin Xiang. Learning spatial-temporal dynamics and interactivity for short-term passenger flow prediction in urban rail transit. *Applied Intelligence*, 53(16):19785–19806, 2023.

Yuankai Wu, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part*

C: Emerging Technologies, 90:166–180, 2018. doi: 10.1016/j.trc.2018.03.001. URL https://www.sciencedirect.com/science/article/pii/S0968090X18302651.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 1907–1913. AAAI Press, 2019. ISBN 9780999241141.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020a. URL https://api.semanticscholar.org/CorpusID:218869770.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 753–763, New York, NY, USA, 2020b. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403118. URL 10.1145/3394486.3403118.

Haoning Xi, John D Nelson, David A Hensher, Songhua Hu, Xuefeng Shao, and Chi Xie. Evaluating travel behavior resilience across urban and rural areas during the covid-19 pandemic: contributions of vaccination and epidemiological indicators. *Transportation research part A: policy and practice*, 180:103980, 2024a.

Haoning Xi, Shao Zhiiqi, David A Hensher, John Nelson, Huaming Chen, and Kasun P Wijayaratna. Xi, haoning and zhiiqi, shao and hensher, david a. and nelson, john and chen, huaming and wijayaratna, kasun p., a multi-task transformer with mixture-of-experts for personalized periodic predictions of individual travel behavior in multimodal public transport. *Available at SSRN 5062293*, 2024b.

Hang Xing, An Chen, and Xuan Zhang. RL-GCN: Traffic flow prediction based on graph convolution and reinforcement learning for smart cities. *Displays*, 80:102513, 2023. ISSN 0141-9382. doi: 10.1016/j.displa.2023.102513. URL https://www.sciencedirect.com/science/article/pii/S0141938223001464.

Yujie Xing, Xiao Wang, Yibo Li, Hai Huang, and Chuan Shi. Less is more: on the over-globalizing problem in graph transformers, 2024.

Xi Xiong, Kaan Ozbay, Li Jin, and Chen Feng. Dynamic origin–destination matrix prediction with line graph neural networks and kalman filter. *Transportation Research Record*, 2674(8): 491–503, 2020.

Chengcheng Xu, Zhibin Li, and Wei Wang. Short-term traffic flow prediction using a methodology based on autoregressive integrated moving average and genetic programming. *Transport*, 31 (3):343–358, 2016. doi: 10.3846/16484142.2016.1212734.

Dong-wei Xu, Yong-dong Wang, Li-min Jia, Yong Qin, and Hong-hui Dong. Real-time road traffic state prediction based on ARIMA and Kalman filter. *Frontiers of Information Technology & Electronic Engineering*, 18:287–302, 2017. doi: 10.1631/FITEE.1500381.

Meng Xu, Wenwu Dai, Chunjing Liu, Xuan Gao, Wei Lin, Guojun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020a.

Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020b.

Xiao Yan, Xianghua Gan, Jingjing Tang, Dapeng Zhang, and Rui Wang. Prostformer: Progressive space-time self-attention model for short-term traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 25(9):10802–10816, 2024. doi: 10.1109/TITS.2024.3367754.

Jin-Ming Yang, Zhong-Ren Peng, and Lei Lin. Real-Time Spatiotemporal Prediction and Imputation of Traffic Status Based on LSTM and Graph Laplacian Regularized Matrix Factorization. *Transportation Research Part C: Emerging Technologies*, 129, August 2021a. doi: 10.1016/j.trc.2021.103228. URL 10.1016/j.trc.2021.103228.

Lijin Yang, Qing Yang, Yonghua Li, and Yuqing Feng. K-Nearest Neighbor Model Based Short-Term Traffic Flow Prediction Method. In *2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 27–30, Wuhan, China, November 2019. IEEE. doi: 10.1109/DCABES48411.2019.00014. URL 10.1109/DCABES48411.2019.00014.

Xin Yang, Qiuchi Xue, Meiling Ding, Jianjun Wu, and Ziyou Gao. Short-term prediction of passenger volume for urban rail systems: A deep learning approach based on smart-card data. *International Journal of Production Economics*, 231:107920, 2021b.

Yijun Yang, Zhaohu Xing, Chunwang Huang, and Lei Zhu. Vivim: A Video Vision Mamba for Medical Video Object Segmentation, March 2024a. URL 10.48550/arXiv.2401.14168. arXiv:2401.14168.

Yongjie Yang, Jinlei Zhang, Lixing Yang, and Ziyou Gao. Network-wide short-term inflow prediction of the multi-traffic modes system: An adaptive multi-graph convolution and attention mechanism based multitask-learning model. *Transportation Research Part C: Emerging Technologies*, 158:104428, 2024b.

Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. doi: 10.1609/aaai.v32i1.11836. URL 10.1609/aaai.v32i1.11836.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3634–3640. International Joint Conferences on Artificial Intelligence Organlstmization, 7 2018a. doi: 10.24963/ijcai.2018/505. URL 10.24963/ijcai.2018/505.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 3634–3640, Stockholm, Sweden, July 2018b. International Joint Conferences on Artificial Intelligence Organization. doi: 10.24963/ijcai.2018/505. URL 10.24963/ijcai.2018/505.

Donghai Yu, Yang Liu, and Xiaohui Yu. A data grouping cnn algorithm for short-term traffic flow forecasting. In *Web Technologies and Applications: 18th Asia-Pacific Web Conference, APWeb 2016, Suzhou, China, September 23-25, 2016. Proceedings, Part I*, pages 92–103. Springer, 2016. doi: 10.1007/978-3-642-41647-7_19.

Yadong Yu, Yong Zhang, Sean Qian, Shaofan Wang, Yongli Hu, and Baocai Yin. A low rank dynamic mode decomposition model for short-term traffic flow prediction. *IEEE Transactions*

on Intelligent Transportation Systems, 22(10):6547–6560, 2021. doi: 10.1109/TITS.2020.2994910.

Dehuai Zeng, Jianmin Xu, Jianwei Gu, Liyan Liu, and Gang Xu. Short term traffic flow prediction using hybrid arima and ann models. In *2008 Workshop on Power Electronics and Intelligent Transportation System*, pages 621–625, 2008. doi: 10.1109/PEITS.2008.135.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *CoRR*, abs/1910.07467, 2019a. URL http://arxiv.org/abs/1910.07467.

Biao Zhang and Rico Sennrich. *Root mean square layer normalization*. Curran Associates Inc., Red Hook, NY, USA, 2019b.

Biao Zhang, Ivan Titov, and Rico Sennrich. Sparse attention with linear units. *arXiv:2104.07012*, pages 1–10, 2021a. URL https://arxiv.org/abs/2104.07012.

Chaoyun Zhang and Paul Patras. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 231–240, 2018. doi: 10.1049/iet-its.2016.0208.

Jinlei Zhang, Hongshu Che, Feng Chen, Wei Ma, and Zhengbing He. Short-term origin-destination demand prediction in urban rail transit systems: A channel-wise attentive split-convolutional neural network method. *Transportation Research Part C: Emerging Technologies*, 124:102928, 2021b.

Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1655–1661. AAAI Press, 2017.

Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011. doi: 10.1109/TITS.2011.2158001.

Wei Zhang, Fenghua Zhu, Yisheng Lv, Chang Tan, Wen Liu, Xin Zhang, and Fei-Yue Wang. Adapgl: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks. *Transportation Research Part C: Emerging Technologies*, 139:103659, 2022. ISSN 0968-090X. doi: 10.1016/j.trc.2022.103659. URL https://www.sciencedirect.com/science/article/pii/S0968090X22001024.

Weibin Zhang, Yinghao Yu, Yong Qi, Feng Shu, and Yinhai Wang. Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning. *Transportmetrica A: Transport Science*, 15(2):1688–1711, 2019. doi: 10.1080/23249935.2019.1637966. URL 10.1080/23249935.2019.1637966.

Yan Zhang, Keyang Sun, Di Wen, Dingjun Chen, Hongxia Lv, and Qingpeng Zhang. Deep learning for metro short-term origin-destination passenger flow forecasting considering section capacity utilization ratio. *IEEE Transactions on Intelligent Transportation Systems*, 24(8): 7943–7960, 2023.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020. doi: 10.1109/TITS.2019.2935152.

Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET intelligent transport systems*, 11(2): 68–75, 2017. doi: 10.1049/iet-its.2016.0208.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1234–1241, Apr. 2020a. doi: 10.1609/aaai.v34i01.5477. URL https://ojs.aaai.org/index.php/AAAI/article/view/5477.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1234–1241, Apr. 2020b. doi: 10.1609/aaai.v34i01.5477. URL https://ojs.aaai.org/index.php/AAAI/article/view/5477.

Jiangchuan Zheng and Lionel M. Ni. Time-dependent trajectory regression on road networks via multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27 (1):1048–1055, Jun. 2013. doi: 10.1609/aaai.v27i1.8577. URL https://ojs.aaai.org/index.php/AAAI/article/view/8577.

Yan Zheng, Shengyou Wang, Chunjiao Dong, Wenquan Li, Wen Zheng, and Jingcai Yu. Urban road traffic flow prediction: A graph convolutional network embedded with wavelet decomposition and attention mechanism. *Physica A: Statistical Mechanics and its Applications*, 608:128274, 2022. doi: 10.1016/j.physa.2022.128274. URL

https://www.sciencedirect.com/science/article/pii/S0378437122008329.

Teng Zhou, Dazhi Jiang, Zhizhe Lin, Guoqiang Han, Xuemiao Xu, and Jing Qin. Hybrid dual Kalman filtering model for short-term traffic flow forecasting. *IET Intelligent Transport Systems*, 13(6):1023–1032, 2019. doi: 10.1049/iet-its.2018.5385.

Guojian Zou, Ziliang Lai, Ting Wang, Zongshi Liu, and Ye Li. Mt-stnet: A novel multi-task spatiotemporal network for highway traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):8221–8236, 2024. doi: 10.1109/TITS.2024.3411638.