LEARNING WEEKLY URBAN EVOLUTION: DELAY-COUPLED NEURAL DYNAMICS FOR EXTREMELY LONG-HORIZON SPATIOTEMPORAL FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding how cities evolve over long horizons is essential for capacity allocation, operational scheduling, and emergency management. As horizons extend from hours to days, city-wide traffic states evolve continuously across temporal phases, giving rise to complex spatiotemporal dynamics. Our analysis of real-world traffic data identifies three interrelated challenges: delayed interactions among nodes and their temporal accumulation, changing dependencies across phases and days, and the joint representation of long-term periodicity and local peak and trough variations. To address these challenges, we propose Delay-Coupled Neural Dynamics (DeCoDy), a graph dynamics framework with time delays for modeling extremely long-horizon urban evolution. DeCoDy addresses them through Multi-Delay Interaction and Accumulation, Phase Evolution and Adaptive Coupling, and Global and Local Residual Representation, while generating continuous future trajectories with a constrained full-horizon response solver. We evaluate DeCoDy on four real-world urban traffic datasets with horizons from 8 hours to 7 days. DeCoDy achieves the lowest MAE in 14 of 16 dataset–horizon settings and the lowest RMSE on all four datasets for 7-day forecasting, demonstrating its effectiveness in modeling extremely long-horizon urban evolution.

1 INTRODUCTION

Spatiotemporal forecasting supports resource allocation, operational scheduling, and advance decision-making by anticipating how urban systems evolve over time. As the prediction horizon extendsZhang et al. (2024), the task increasingly requires modeling the continuous evolution of spatially distributed states rather than only mapping historical observations to future values. Existing approaches improve long-horizon forecasting through direct predictionShao et al. (2022a); Zeng et al. (2023), periodic modeling, attention-based temporal dependency learningZhou et al. (2021); Wu et al. (2021); Zhou et al. (2022); Liu et al. (2021); Nie et al. (2022); Zhang & Yan (2023); Lin et al. (2024), or graph-based spatial interaction modelingShao et al. (2022b); Liu et al. (2023a); Hu et al. (2025); Li et al. (2017); Song et al. (2020); Gao et al. (2023). However, from hours to daysZhang et al. (2024); He et al. (2025), city-wide traffic states evolve continuously across locations, phases, and day boundaries, while recurring patterns coexist with local peak and trough variations. Capturing such multiscale and interconnected evolution remains challenging for extremely long-horizon urban forecastingFang et al. (2025).

To better understand the challenges of modeling multi-day urban traffic evolution, we analyze real-world traffic data, as shown in Figure 1. First, Figure 1(a) shows that neighboring nodes exhibit different deviation onset times and durations within the same episode. Deviations at multiple nodes overlap in time, while the target node also changes throughout this ongoing evolution. These observations indicate delayed interactions among nodes together with accumulation over time, as deviations remain active and overlap in subsequent periods. Second, Figure 1(b) shows continuous traffic evolution across intraday periods and a day boundary: deviations persist across midnight, while local lagged correlation patterns between nodes change over time. This indicates that traffic states evolve continuously across phases and days, while inter-node dependencies vary with temporal conditions. Finally, Figure 1(c) shows that traffic flows on different dates share similar overall daily profiles but differ markedly in local peak magnitudes, indicating that fixed periodic patterns

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

cannot fully characterize date-specific local variations. Overall, these observations reveal three key characteristics of multi-day traffic evolution: delayed interactions among nodes and their accumulation over time; continuous evolution across phases and days with changing inter-node dependencies; and shared overall patterns coexisting with date-specific peak and trough variations. Details of the references, case selection, and local correlation calculations are provided in Appendix B.

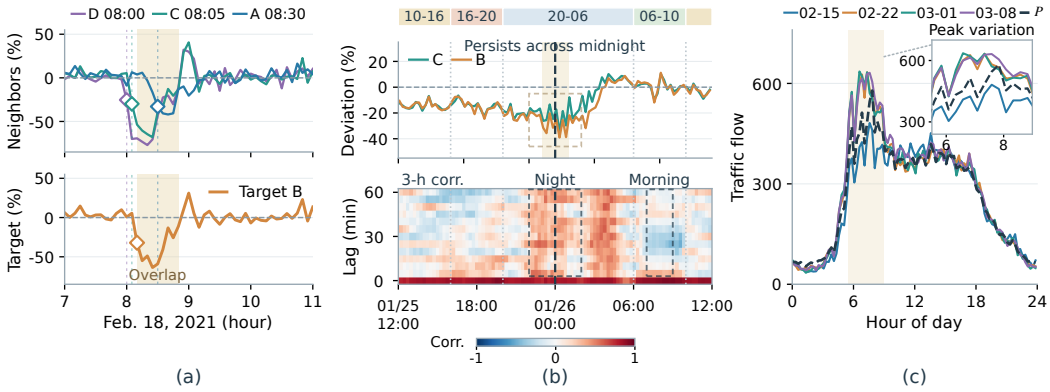

Figure 1: Analysis of real-world traffic data. **(a)** Staggered and overlapping deviations across neighboring nodes. **(b)** Persistent deviations across phases and days with time-varying lagged correlations. **(c)** Shared daily patterns with date-specific local peak variations.

These observations motivate three interrelated modeling challenges in learning multi-day urban traffic evolution. (1) Delayed interactions and temporal accumulation. Neighboring nodes may change asynchronously, while deviations persist into subsequent periods. Models therefore need to capture interactions at different delays and their accumulated effects over time. (2) Evolving relationships and state continuity across phases and days. Multi-day evolution spans different intraday phases and day boundaries, where inter-node dependencies may change while earlier states continue to influence subsequent evolution. Models must therefore adapt interactions to changing temporal conditions while preserving state continuity. (3) Joint representation of global periodic patterns and local peak and trough variations. Different dates often share similar overall profiles but differ substantially in local peaks and troughs. Models must preserve stable periodic structure while retaining sufficient flexibility for date-specific local variations. Addressing these challenges within a unified framework is essential for modeling urban evolution over extended horizons.

In this work, we propose DeCoDy (Delay-Coupled Neural Dynamics), a graph dynamics framework with time delays for modeling multi-day urban traffic evolution. To capture delayed interactions among nodes and their temporal accumulation, we introduce the Multi-Delay Interaction and Accumulation module, which integrates state persistence with neighboring responses at multiple delays to model future evolution. To accommodate evolving relationships across phases and days, we introduce the Phase Evolution and Adaptive Coupling module, which combines historical conditions, future calendar information, and background profiles to generate time-varying local driving inputs and node couplings. To jointly represent global periodic patterns and local peak and trough variations, we design the Global and Local Residual Representation module, which uses a periodic background as a long-term reference and models dynamic deviations from it. Finally, a constrained full-horizon response solver generates continuous multi-day trajectories while controlling numerical solution error. Our main contributions are summarized as follows:

1. We analyze real-world urban traffic data and identify three key characteristics of multi-day urban traffic evolution: delayed interactions among nodes and their temporal accumulation, evolving relationships and state continuity across phases and days, and the coexistence of global periodic patterns with local peak and trough variations.
2. We propose DeCoDy, a graph dynamics framework with time delays for modeling multi-day urban traffic evolution:
   (a) Multi-Delay Interaction and Accumulation models delayed interactions among nodes and their temporal accumulation;

(b) Phase Evolution and Adaptive Coupling captures evolving relationships while preserving state continuity across phases and days;

(c) Global and Local Residual Representation jointly represents global periodic patterns and local peak and trough variations.

3. We conduct extensive evaluations on four real-world urban traffic datasets with forecast horizons ranging from 8 hours to 7 days. DeCoDy achieves the lowest MAE in 14 of 16 dataset–horizon settings and the lowest RMSE on all four datasets for 7-day forecasting.

2 Related Work

2.1 Delayed Interactions and Temporal Accumulation

Existing spatiotemporal forecasting methods widely use graph structures to characterize dependencies among spatial nodesLi et al. (2017). A common line of work learns spatial relationships from data and combines them with temporal modeling to capture interactions among nodesJiang et al. (2023b). More fine-grained approaches further construct localized spatiotemporal relations to distinguish dependencies that vary across temporal contextsHu et al. (2025). From a dynamical perspective, delayed neighboring states can also be incorporated into continuous state evolutionLong et al. (2024), allowing interactions to occur at different temporal offsetsJiang et al. (2023a).

These studies suggest that spatial dependencies involve not only whether nodes are related, but also when neighboring states influence one another. For multi-day forecasting, however, a single spatial relationship or local delay alone is insufficient to describe how interactions at multiple delays and a node's own persistent response jointly accumulate in future trajectories. Integrating multi-delay interactions among nodes with state persistence therefore remains an important problem in modeling long-horizon urban traffic evolution.

2.2 Evolving Interactions and State Continuity Across Phases and Days

Traffic relationships do not remain constant throughout the forecast horizon. Existing approaches have explored adaptive spatial relationshipsJiang et al. (2023b); Wu et al. (2019); Bai et al. (2020), time-dependent interaction structuresJiang et al. (2023a); Hu et al. (2025), and condition-specific dependencies to account for changes in traffic dynamics over timeLiu et al. (2023a). Such studies indicate that interactions among nodes may vary with temporal conditions rather than remaining fixed throughout the forecasting process.

Multi-day evolution further spans multiple intraday phases and day boundaries, requiring models to adapt inter-node relationships to changing temporal conditions while preserving the influence of preceding states. A key question is therefore how to adjust local driving inputs and node interactions over time without treating different phases as independent segments, thereby maintaining continuous evolution across phases and days.

2.3 Global Periodic Patterns and Local Variations

Long-term traffic sequences typically contain recurring periodic structure alongside substantial date-specific local deviationsWu et al. (2022). Existing studies have addressed this property through temporal decompositionZhou et al. (2021; 2022); He et al. (2025), explicit periodic modelingWu et al. (2022); Liu et al. (2021); Lin et al. (2024), and residual learning. These approaches generally separate relatively stable recurring patterns from time-varying components so that long-term regularity and local changes can be modeled with different representations.

Such studies suggest that separating recurring structure from local variations benefits long-horizon forecasting. However, decomposition alone does not specify how future local deviations evolve continuously over time. Multi-day urban traffic evolution therefore requires a unified representation that preserves a stable periodic reference while retaining sufficient flexibility for date-specific peak and trough variations, allowing global recurring structure and local deviations to jointly shape future trajectories.

3 METHOD

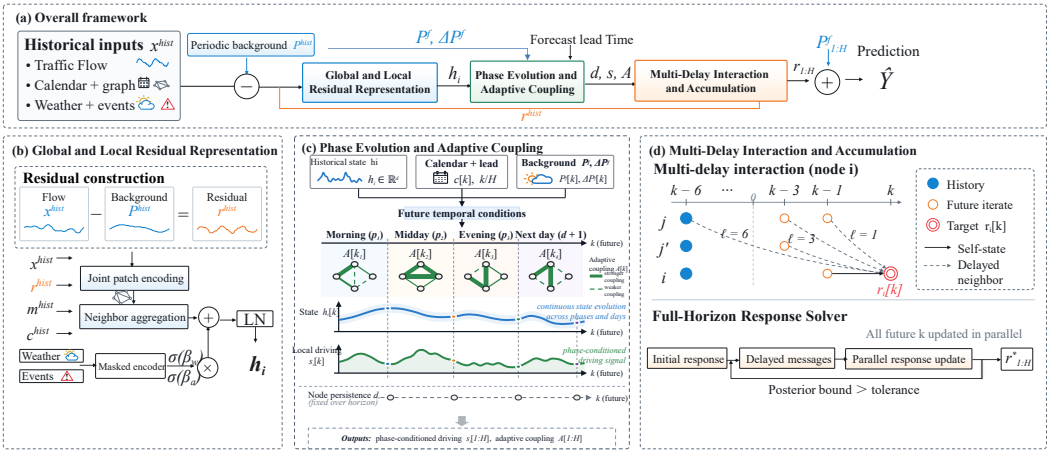

Figure 2: Overview of DeCoDy, comprising Global and Local Residual Representation, Phase Evolution and Adaptive Coupling, and Multi-Delay Interaction and Accumulation.

We consider a road graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes. Given historical traffic $\mathbf{X} \in \mathbb{R}^{L \times N}$, historical calendar features $\mathbf{C}^h$, external conditions $\mathbf{E}^h$, observation masks $\mathbf{M}^h$, and known future calendar information $\mathbf{C}^f$, DeCoDy predicts $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times N}$ over a multi-day horizon. It consists of Global and Local Residual Representation, Phase Evolution and Adaptive Coupling, and Multi-Delay Interaction and Accumulation.

3.1 GLOBAL AND LOCAL RESIDUAL REPRESENTATION

This module separates stable periodic structure from date-specific deviations. For node i at weekly slot τ, the periodic background is

$$P_i(\tau) = \alpha_{i,\tau}\mu^{\text{week}}_{i,\tau} + (1-\alpha_{i,\tau})\mu^{\text{day}}_{i,\tau \bmod 288}, \qquad \alpha_{i,\tau} = \frac{n_{i,\tau}}{n_{i,\tau}+\lambda}, \tag{1}$$

where $n_{i,\tau}$ is the number of valid training observations and λ controls shrinkage toward the daily profile. The background is computed only from training data and remains fixed.

Historical traffic is represented relative to this background:

$$r_i[t] = x_i[t] - P_i(\tau(t)), \qquad t = 1-L, \ldots, 0, \tag{2}$$

providing both date-specific deviations and the historical boundary for subsequent dynamics.

For each node, historical flow, residuals, masks, and calendar features are patched and encoded into $\mathbf{e}_i$. Road-neighbor information is aggregated as

$$\mathbf{g}_i = \text{LN}\left(\mathbf{e}_i + \phi\left(W_g\left[\mathbf{e}_i, \sum_{j\in\mathcal{N}_i}\bar{w}_{ji}\mathbf{e}_j\right] + b_g\right)\right), \tag{3}$$

where $\bar{w}_{ji}$ is the normalized road weight. Historical weather and event encoders then yield

$$\mathbf{h}_i = \text{LN}\left(\mathbf{g}_i + \sigma(\beta_w)\mathbf{e}^w_i + \sigma(\beta_a)\mathbf{e}^a_i\right), \tag{4}$$

which summarizes historical traffic and operating conditions.

3.2 PHASE EVOLUTION AND ADAPTIVE COUPLING

This module generates future local driving and node interactions conditioned on historical state and future temporal conditions. For future step k,

$$\mathbf{q}_i[k] = \phi(W_h\mathbf{h}_i + W_c[\mathbf{c}[k], k/H] + W_p[P_i[k], P_i[k]-P_i[k-1]] + \mathbf{b}_q), \tag{5}$$

where $\mathbf{c}[k]$ contains known daily and weekly calendar features.

Node persistence and time-varying local driving are defined by

$$d_i = \mathrm{softplus}(\mathbf{w}_d^\top \mathbf{h}_i + b_d) + \epsilon, \qquad s_i[k] = d_i(\mathbf{w}_s^\top \mathbf{q}_i[k] + b_s),$$

where $d_i > 0$ is fixed within a forecast horizon while $s_i[k]$ varies over time.

For road edge $j \to i$ and delay $\ell \in \mathcal{D}$, the raw adaptive coupling is

$$\widetilde{A}_{ij\ell}[k] = B_{ij\ell} + \eta \tanh\left(W_r \mathbf{q}_i[k] + W_s \mathbf{q}_j[k] + W_e w_{ji} + \mathbf{b}_e\right)_\ell, \tag{6}$$

where $B_{ij\ell}$ is an edge–delay baseline initialized from training-set lagged correlations.

We bound the total coupling gain using

$$\gamma_i = \rho d_i \sigma(\mathbf{w}_\gamma^\top \mathbf{h}_i + b_\gamma), \qquad A_{ij\ell}[k] = \frac{\gamma_i \widetilde{A}_{ij\ell}[k]}{1 + \sum_{j' \in \mathcal{N}_i} \sum_{\ell' \in \mathcal{D}} |\widetilde{A}_{ij'\ell'}[k]|},$$

which guarantees

$$\frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \sum_{\ell \in \mathcal{D}} |A_{ij\ell}[k]| < \rho < 1. \tag{7}$$

Proposition 1. Horizon-Uniform Boundedness. Let

$$R_0 = \max_{i, t \le 0} |r_i[t]|, \qquad U = \max_{i, 1 \le k \le H} \left| \frac{s_i[k]}{d_i} \right|.$$

Under Equation (7),

$$\max_{i, 1 \le k \le H} |r_i[k]| \le \max\left\{ R_0, \frac{U}{1 - \rho} \right\}. \tag{8}$$

Thus, response magnitude remains bounded without explicit growth with H. The proof is given in Appendix C.1.

3.3 Multi-Delay Interaction and Accumulation

Given d_i, $s_i[k]$, and $A_{ij\ell}[k]$, future residual dynamics are

$$r_i[k] = a_i r_i[k-1] + b_i \left(s_i[k] + \sum_{j \in \mathcal{N}_i} \sum_{\ell \in \mathcal{D}} A_{ij\ell}[k] r_j[k - \ell] \right), \tag{9}$$

where

$$a_i = e^{-d_i}, \qquad b_i = \frac{1 - e^{-d_i}}{d_i}. \tag{10}$$

The terms respectively describe self-state persistence, local driving, and delayed neighboring influence. We use $\mathcal{D} = \{1, 2, 3, 6, 12\}$, corresponding to 5–60 minutes.

Because d_i is fixed within a horizon, define the causal local response operator

$$(\mathcal{H}_i f_i)[k] = \sum_{u=1}^{k} b_i a_i^{k-u} f_i[u],$$

and the base trajectory

$$r_i^{\mathrm{base}}[k] = a_i^k r_i[0] + \mathcal{H}_i(s_i + m_i^{\mathrm{hist}})[k].$$

Let $\mathcal{M}$ denote delayed interactions among future trajectories. Then

$$\mathbf{r} = \mathbf{r}^{\mathrm{base}} + \mathcal{K}(\mathbf{r}), \qquad \mathcal{K} = \mathcal{H} \circ \mathcal{M}. \tag{11}$$

Starting from $\mathbf{r}^{(0)} = \mathbf{r}^{\mathrm{base}}$, we update

$$\mathbf{r}^{(m+1)} = \mathbf{r}^{\mathrm{base}} + \mathcal{K}(\mathbf{r}^{(m)}), \tag{12}$$

simultaneously over all future positions. Local responses are computed with zero-padded FFTs.

Proposition 2. Finite-Horizon Expansion and Error Control. With positive minimum delay $\ell_{\min}$,

$$\mathcal{K}^{Q} = 0, \qquad Q = \left\lceil \frac{H}{\ell_{\min}} \right\rceil,$$

and the exact response is

$$\mathbf{r}^{\star} = \sum_{p=0}^{Q-1} \mathcal{K}^{p} \mathbf{r}^{\mathrm{base}}. \tag{13}$$

For

$$q = \max_{i,k} \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \sum_{\ell \in \mathcal{D}} |A_{ij\ell}[k]| < \rho,$$

the m-th update satisfies

$$\left\| \mathbf{r}^{(m)} - \mathbf{r}^{\star} \right\|_{\infty} \leq \frac{q}{1-q} \left\| \mathbf{r}^{(m)} - \mathbf{r}^{(m-1)} \right\|_{\infty}. \tag{14}$$

Hence, positive delays yield finite propagation depth and the gain constraint provides an a posteriori error certificate. The proof is given in Appendix C.2.

The bound in Equation (14) is checked every four iterations after at least four updates. We stop when the error in original flow units is at most 0.05, with a default maximum of 64 updates.

3.4 Prediction and Training Objective

The final forecast is

$$\hat{y}_i[k] = \max\left\{0,\, \sigma_x(P_i[k] + r_i[k]) + \mu_x\right\}, \tag{15}$$

where μ_x and σ_x are training-set statistics.

The model is optimized end to end using masked MAE:

$$\mathcal{L} = \frac{\sum_{n,i,k} M_{n,i,k} |\hat{y}_{n,i}[k] - y_{n,i}[k]|}{\sum_{n,i,k} M_{n,i,k}}. \tag{16}$$

Gradients propagate through the executed full-horizon updates, while the periodic background and normalization statistics remain fixed.

Table 1: Traffic flow forecasting results on four real-world traffic datasets. Forecast horizons $H \in \{96, 288, 864, 2016\}$ correspond to 8 hours, 1 day, 3 days, and 7 days, respectively. MAE and RMSE are reported for each setting. The best results are boldfaced and the second-best results are underlined.

Dataset	Pred.	DeCoDy		Extralonger		VisiFold		DST-Mamba		TimeFilter		iTransformer		FaST		CycleNet		STID	
	steps	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
$PEMS04_{266}$	96	**25.11**	40.52	25.36	**40.43**	29.02	45.03	33.48	48.99	34.84	50.94	37.96	56.11	34.13	51.66	36.78	54.40	27.40	42.75
	288	**26.71**	44.27	27.29	45.63	28.73	45.34	28.09	44.32	31.00	48.45	30.79	50.15	27.16	**43.89**	34.08	52.85	27.88	44.19
	864	**27.20**	45.45	28.11	**45.26**	30.71	48.10	31.51	48.73	34.44	52.65	35.40	54.76	28.36	45.82	37.61	56.31	29.50	46.53
	2016	**26.77**	**44.16**	29.77	47.41	31.62	49.35	34.60	54.39	36.31	55.84	35.15	54.20	29.38	47.06	35.82	55.07	29.88	47.31
$PEMS08_{112}$	96	**22.42**	**35.06**	22.78	35.40	25.24	38.14	32.16	47.39	37.60	56.21	33.00	48.06	29.95	44.87	34.17	49.25	24.00	36.43
	288	**23.15**	36.22	23.52	36.32	24.43	36.92	24.82	37.81	27.11	41.73	26.60	40.73	23.34	36.01	29.80	44.06	23.55	**35.85**
	864	**24.74**	38.46	27.13	40.87	26.76	39.93	30.78	46.81	31.18	46.20	31.08	46.58	24.99	**38.29**	34.06	49.98	25.88	39.02
	2016	**23.94**	**37.70**	29.66	44.77	28.27	41.92	29.22	44.01	32.07	47.02	30.88	46.54	25.50	39.28	32.60	48.14	26.53	40.11
SD_{673}	96	**22.09**	**34.52**	27.75	42.11	30.43	47.42	33.32	50.18	38.35	58.48	37.01	57.20	35.62	53.81	39.61	58.25	26.25	40.11
	288	23.05	35.92	**22.87**	36.65	25.18	39.23	28.90	44.38	32.64	50.91	29.88	48.61	24.77	38.71	23.23	**35.73**	23.16	35.97
	864	**24.23**	**37.72**	26.10	40.34	27.55	42.64	38.03	57.75	37.47	55.94	34.93	54.19	26.65	41.51	24.64	38.41	26.11	40.22
	2016	24.96	**38.62**	30.72	46.85	28.04	43.51	33.25	52.25	34.77	52.16	33.81	53.17	26.63	41.81	**24.60**	38.77	25.96	40.33
$Seattle_{196}$	96	**8.33**	**13.67**	9.14	14.25	9.69	15.04	11.27	16.37	11.91	17.55	16.92	22.90	12.25	18.42	11.34	16.34	9.05	14.00
	288	**8.89**	14.90	9.12	14.99	9.49	15.01	10.51	16.46	10.27	16.33	14.75	20.77	9.47	15.33	11.02	16.78	9.23	**14.56**
	864	**9.33**	**15.76**	10.33	16.50	10.62	16.80	12.16	18.83	12.00	18.95	11.95	18.41	10.72	17.39	12.57	19.31	10.23	16.25
	2016	**9.63**	**16.46**	11.76	18.40	11.05	17.62	12.63	20.00	12.33	19.68	12.15	19.01	10.89	17.90	12.54	19.44	10.68	17.04

4 EXPERIMENTS

4.1 DATASETS AND EXPERIMENTAL SETTINGS

Datasets. We evaluate DeCoDy on four real-world traffic datasets: PEMS04 and PEMS08 Guo et al. (2019), SD Liu et al. (2023b), and Seattle Cui et al. (2019). All datasets are sampled at 5-minute intervals and chronologically split into training, validation, and test sets in a $6:2:2$ ratio.

Baselines and Experimental Settings. We compare DeCoDy with eight representative methods: iTransformer Liu et al. (2024), Extralonger Zhang et al. (2024), TimeFilter Hu et al. (2025), VisiFold Zhang et al. (2026), CycleNet Lin et al. (2024), STID Shao et al. (2022a), DST-Mamba He et al. (2025), and FaST Fang et al. (2025). Forecast horizons are $H \in \{96, 288, 864, 2016\}$, corresponding to 8 hours, 1 day, 3 days, and 7 days. The 8-hour task uses $L = 96$, while the remaining tasks use $L = 288$. A separate model is trained for each horizon. All methods use the same data splits, node sets, and test windows while retaining their native input configurations. Models are selected by validation MAE and evaluated using MAE and RMSE, with random seed 42. Further details are provided in Appendix A.

4.2 MAIN RESULTS

Overall Forecasting Performance. Table 1 compares DeCoDy with eight baselines across four datasets and four horizons. DeCoDy achieves the lowest MAE in 14 of 16 settings and the lowest RMSE in 9 settings. For 7-day forecasting, it obtains the lowest RMSE on all four datasets and the lowest MAE on three. On these three datasets, DeCoDy reduces MAE over the strongest baseline by 8.90%, 6.10%, and 9.77% on PEMS04, PEMS08, and Seattle, respectively, demonstrating strong performance over extremely long horizons.

Performance Across Consecutive Forecast Days. Figure 3 divides each 7-day forecast into consecutive 24-hour intervals. DeCoDy achieves the lowest MAE in 22 of 28 dataset–day combinations, including days 2–7 on PEMS04, days 1–6 on PEMS08, days 3–5 on SD, and all seven days on Seattle. Its advantage therefore extends beyond the early forecast period and persists across multiple future days.

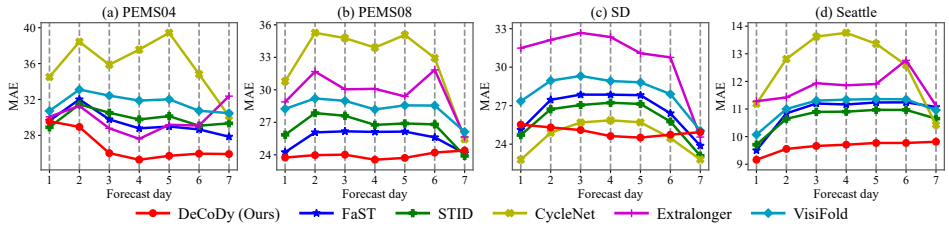

Figure 3: Daily MAE for 7-day forecasting. Each point aggregates valid targets across all nodes and forecast origins within the corresponding 24-hour interval. Lower values are better. FaST and STID use a maximum of 15 training rounds; CycleNet uses 10 on SD and 40 on the other datasets. All results use validation-selected checkpoints.

4.3 ABLATION STUDY

Ablation Study. We evaluate five variants on PEMS04, PEMS08, and Seattle: C-PB removes the periodic background and its associated conditions; C-SC fixes future coupling; C-FGC removes future cross-node coupling; C-SD retains only a single five-minute delay; and C-EI removes historical weather and event encoding. As shown in Table 2, DeCoDy achieves the lowest MAE and RMSE in all 12 dataset–horizon settings. The degradation of C-FGC and C-SD supports the effectiveness of cross-node interactions and multi-delay modeling, with C-SD increasing seven-day MAE on Seattle by 9.86%. The increased errors of C-SC support adapting coupling relationships to future temporal conditions for cross-phase forecasting. Removing the periodic background increases seven-day MAE by 6.78%, 4.65%, and 6.18% on the three datasets, respectively, while removing external in-

formation increases seven-day MAE on Seattle by 9.48%, demonstrating the contributions of the periodic reference and historical external conditions.

Table 2: Performance comparisons on ablative variants.

Datasets		PEMS04				PEMS08				Seattle			
Predict Length		96	288	864	2016	96	288	864	2016	96	288	864	2016
C-PB	MAE	26.180	27.212	28.313	28.584	22.667	23.780	25.482	25.059	8.637	9.224	9.979	10.229
	RMSE	41.854	44.485	46.197	46.567	35.171	36.516	39.014	38.963	13.806	14.908	16.097	16.862
C-SC	MAE	26.069	27.335	27.893	27.563	23.183	23.152	24.862	24.030	8.411	8.962	9.386	9.657
	RMSE	40.651	44.614	45.607	45.104	35.314	36.254	38.797	38.550	13.806	14.970	15.940	16.641
C-FGC	MAE	25.953	26.799	27.278	26.936	23.127	23.906	25.294	24.222	8.974	8.940	9.344	9.732
	RMSE	40.570	44.320	45.738	44.429	35.487	36.838	38.670	37.743	14.301	15.746	16.649	17.430
C-SD	MAE	25.131	26.755	27.787	26.846	22.429	23.214	24.823	23.973	8.485	9.051	9.427	10.584
	RMSE	40.803	44.927	45.982	44.418	35.329	36.238	38.521	37.724	13.873	14.963	15.833	17.425
C-EI	MAE	25.119	26.959	27.281	27.218	22.469	23.888	24.943	24.119	9.059	9.452	10.102	10.547
	RMSE	40.569	44.658	45.683	44.239	35.553	37.033	38.832	37.993	14.189	15.271	16.588	16.499
DeCoDy	MAE	**25.109**	**26.707**	**27.198**	**26.769**	**22.424**	**23.151**	**24.736**	**23.945**	**8.329**	**8.885**	**9.329**	**9.634**
	RMSE	**40.518**	**44.266**	**45.447**	**44.164**	**35.062**	**36.220**	**38.460**	**37.703**	**13.666**	**14.903**	**15.756**	**16.462**

4.4 Hyperparameter Analysis

Hyperparameter Analysis. We study two structural parameters for 7-day forecasting: the coupling gain bound ρ and periodic-background shrinkage strength λ, using the complete validation sets of PEMS08 and Seattle. We vary $\rho \in \{0.2, 0.4, 0.6, 0.8, 0.9\}$ and $\lambda \in \{1, 2.5, 5, 7.5, 10\}$.

As shown in Figure 4, PEMS08 errors generally decrease as ρ increases to 0.8 and rise at 0.9, illustrating the trade-off between coupling strength and stability margin. For λ, the default value of 5 performs best on PEMS08, while both smaller and larger values increase error. Seattle is comparatively insensitive to both parameters. Overall, DeCoDy remains stable over the examined ranges.

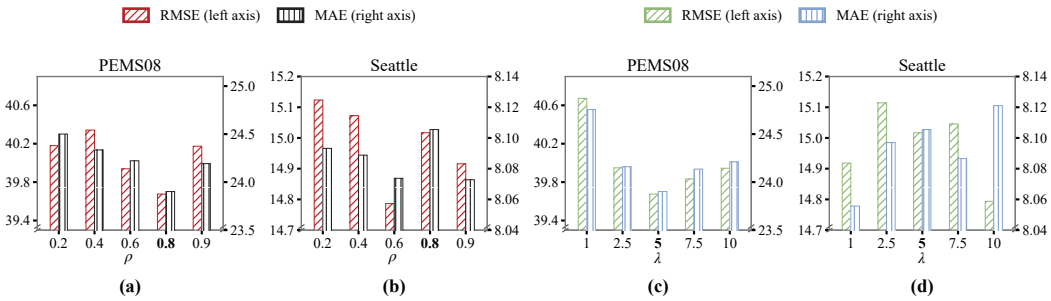

Figure 4: Hyperparameter sensitivity on PEMS08 and Seattle with $L = 288$, $H = 2016$. (a), (b) Varying ρ with $\lambda = 5$. (c), (d) Varying λ with $\rho = 0.8$. RMSE (left axis) and MAE (right axis) are evaluated on the complete validation set. Bold ticks mark defaults; the vertical axes are truncated. The solver iteration cap is 128 for $\rho = 0.9$ and 64 otherwise, with an error tolerance of 0.05.

4.5 Case Study

To examine local deviations at long forecast lead times, we analyze a 7-day forecast for an I-405 node in Seattle. The forecast originates at 04:00 on November 8, 2015. On November 13, traffic retains the overall periodic profile but shows persistent negative deviations from 06:00 to 20:00. DeCoDy captures these deviations at lead times of approximately 122–136 hours. Predicted cumulative flow is 14.8% below the periodic background, compared with an observed reduction of 17.2%, a difference of 2.4 percentage points. The predicted decreases are also consistent with observations across three consecutive periods. This case shows that DeCoDy can preserve global periodic structure while capturing date-specific deviations far into the forecast horizon. Statistical details are provided in Appendix B.4.

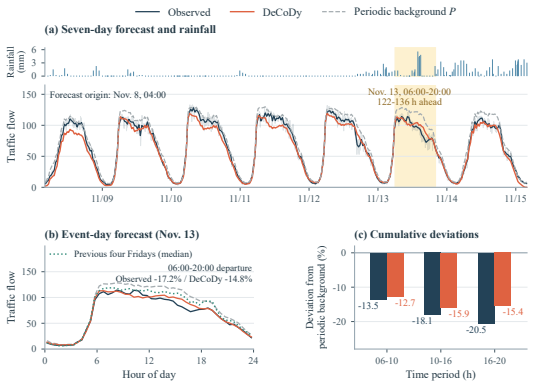

Figure 5: Seven-day forecasting case on I-405 in Seattle. (a) Full forecast and rainfall context. (b) Event-day trajectories. (c) Cumulative deviations from the periodic background in three periods.

5 Conclusion

This work studies extremely long-horizon urban traffic evolution and analyzes three key characteristics of real-world traffic data, including delayed interactions among nodes and their temporal accumulation, evolving relationships and state continuity across phases and days, and the coexistence of global periodic patterns with local peak and trough variations. To address these challenges, we propose DeCoDy, whose Multi-Delay Interaction and Accumulation, Phase Evolution and Adaptive Coupling, and Global and Local Residual Representation modules respectively model delayed interactions over time, evolving inter-node dependencies across temporal phases, and global periodic structure together with local variations. Constrained coupling parameterization and full-horizon solution further support stable response evolution and controlled numerical solution error over extended horizons. Experiments on four real-world urban traffic datasets demonstrate strong performance from 8 hours to 7 days, with DeCoDy achieving the lowest MAE in 14 of 16 dataset–horizon settings and the lowest RMSE on all four datasets for 7-day forecasting.

AI Use Statement

In this work, we used generative AI tools to assist with language polishing, grammatical correction, and improving the clarity and presentation of the manuscript, as well as limited assistance in code debugging and figure refinement. We did not use generative AI to generate experimental data, fabricate results, or independently determine the scientific conclusions of this work. All AI-assisted content was carefully reviewed, revised, and verified by the authors. Technical descriptions and claims were checked against our implementation and experimental results, and references were manually verified against the original publications. The authors take full responsibility for the final content of this work, including all text, claims, code, figures, and experimental results produced with the aid of generative AI.

References

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2019.

Yuchen Fang, Yuxuan Liang, Bo Hui, Zezhi Shao, Liwei Deng, Xu Liu, Xinke Jiang, and Kai Zheng. Efficient large-scale traffic forecasting with transformers: A spatial data management perspective. In *Proceedings of the 31st ACM SIGKDD conference on knowledge discovery and data mining v. 1*, pp. 307–317, 2025.

Haotian Gao, Renhe Jiang, Zheng Dong, Jinliang Deng, Yuxin Ma, and Xuan Song. Spatial-temporal-decoupled masked pre-training for spatiotemporal forecasting. *arXiv preprint arXiv:2312.00516*, 2023.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

Sicheng He, Junzhong Ji, and Minglong Lei. Decomposed spatio-temporal mamba for long-term traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11772–11780, 2025.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. *arXiv preprint arXiv:2501.13041*, 2025.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4365–4373, 2023a.

Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 8078–8086, 2023b.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37:106315–106345, 2024.

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*, pp. 4125–4129, 2023a.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in neural information processing systems*, 36:75354–75371, 2023b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Qingqing Long, Zheng Fang, Chen Fang, Chong Chen, Pengfei Wang, and Yuanchun Zhou. Unveiling delay effects in traffic forecasting: a perspective from spatial-temporal delay differential equations. In *Proceedings of the ACM web conference 2024*, pp. 1035–1044, 2024.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

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*, pp. 4454–4458, 2022a.

Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1567–1577, 2022b.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 914–921, 2020.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. arxiv 2022. *arXiv preprint arXiv:2210.02186*, 2022.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Zhiwei Zhang, Fandong Meng, Jie Zhou, Wenjuan Han, et al. Extralonger: Toward a unified perspective of spatial-temporal factors for extra-long-term traffic forecasting. *arXiv preprint arXiv:2411.00844*, 2024.

Zhiwei Zhang, Xinyi Du, Weihao Wang, Xuanchi Guo, and Wenjuan Han. Visifold: Long-term traffic forecasting via temporal folding graph and node visibility. *arXiv preprint arXiv:2603.11816*, 2026.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

A Experimental Details

A.1 Datasets

We evaluate the proposed method on four real-world traffic datasets: PEMS04, PEMS08, SD, and Seattle. All datasets are sampled at five-minute intervals, with traffic flow as the forecasting target. Table 3 summarizes the data actually used for training and evaluation in this study.

Table 3: Statistics of the datasets used in this study.

Dataset	Nodes used	Time steps	Sampling interval
PEMS04	266	16,992	5 minutes
PEMS08	112	17,856	5 minutes
SD	673	34,560	5 minutes
Seattle	196	105,120	5 minutes

PEMS04 and PEMS08. To associate traffic nodes with historical external conditions, we retain 266 and 112 nodes, respectively, whose station identities, geographic locations, and road attributes can be verified. Node selection applies consistently to historical inputs, road graphs, normalization statistics, training losses, and evaluation targets. The road graphs retain existing connections among selected nodes and reindex their endpoints, without adding substitute connections for excluded nodes. Our results therefore correspond to these node subsets and cannot be compared numerically with published results on the full-node versions directly.

SD. This dataset contains traffic observations from January to April 2021, and we use all 673 nodes. External conditions are aligned under a fixed PST (UTC-8) traffic-clock assumption. This interpretation is part of our data-processing protocol and has not been independently confirmed by the traffic data provider.

Seattle. We use observations from the full year of 2015 and retain 196 nodes based on the availability of original observations within the training interval. Forecasting targets use the original released flow values without lane-count scaling. External data are aligned using local civil time in America/Los_Angeles, supported by analyses of daily patterns and daylight-saving consistency within the training interval. Traffic targets are retained at daylight-saving time slots that cannot be uniquely mapped to physical time, while the corresponding external conditions are marked as missing.

All datasets are split chronologically into training, validation, and test sets in a $6:2:2$ ratio. Target windows do not cross their respective split boundaries. Traffic normalization statistics, periodic backgrounds, and lagged correlation statistics for coupling initialization are computed exclusively from the training interval.

A.2 Historical External Conditions

DeCoDy uses historical weather and incident records to supplement traffic representations. Weather inputs include temperature, humidity, wind speed, wind direction, precipitation, visibility, and pressure, together with observation age and validity information. Incident inputs summarize the number of matched incidents within a historical interval and the time since the most recent matched incident. Node-level external conditions are constructed according to the data sources and spatial matching protocols of each dataset.

All external inputs are restricted to the historical window before the forecast origin. Future inputs contain only known calendar information, without actual weather or incidents from the forecast interval. Missing external observations are handled using explicit masks, and their normalization parameters are estimated only from valid observations in the training interval.

Incident data come from retrospectively compiled records, with historical conditions constructed according to occurrence times. This setting is therefore a retrospective forecasting experiment based on historical occurrence times, rather than a complete reconstruction of incident reports actually received at the time of forecasting. In Seattle, incident matching represents event context within 500 meters of a node, without implying verified incidents on the same road and in the same direction.

A.3 Baseline Methods

We select eight comparison methods covering general multivariate forecasting, periodic modeling, lightweight spatiotemporal representations, and long-horizon traffic forecasting.

1. **iTransformer** represents variable sequences as tokens and learns relationships among variables through attention.
2. **Extralonger** jointly models spatial and temporal factors for extra-long-term traffic forecasting.
3. **TimeFilter** constructs spatiotemporal relationships at the temporal-patch level and filters fine-grained dependencies relevant to forecasting.
4. **VisiFold** organizes cross-time relationships through a temporal folding graph and uses node visibility to handle larger traffic networks.
5. **CycleNet** explicitly learns recurring periodic patterns and predicts residuals after removing them.
6. **STID** incorporates spatial and temporal identity embeddings into multilayer perceptrons to construct a lightweight forecasting model.
7. **DST-Mamba** decomposes time series into seasonal and trend components and combines Mamba with linear mappings to model long-range dependencies.
8. **FaST** uses a mixture-of-experts mechanism for long-horizon forecasting on large-scale spatiotemporal graphs.

All baselines are trained and evaluated using our data splits, node sets, and forecasting tasks, while retaining their native input configurations and corresponding training objectives. DeCoDy additionally uses historical weather and incident conditions. The main comparison therefore reflects overall forecasting performance under the current input configurations and cannot by itself attribute gains separately to external information and model architecture.

A.4 Implementation and Training Settings

For eight-hour forecasting, both the look-back window and forecast horizon are set to 96. For one-day, three-day, and seven-day forecasting, the look-back window is 288 and the forecast horizons are 288, 864, and 2016, respectively. An independent model is trained for each dataset–horizon combination, with the random seed fixed at 42. The main DeCoDy settings are listed in Table 4.

A training round is defined by a fixed sample-presentation budget rather than a complete pass through the training set. Training uses a predetermined stream of sampled windows, with progress monitored on a fixed sparse set of validation windows. The first four rounds are excluded from early-stopping patience accounting, and at least five rounds are completed. After training, candidate checkpoints are compared using MAE on the complete validation set, and the selected checkpoint is evaluated on the complete test set.

DeCoDy uses FP32 computation. Larger tasks use microbatches and gradient accumulation to control memory consumption while preserving the effective batch size. Background statistics remain fixed during training, while the remaining forecasting parameters are jointly optimized using masked MAE in original flow units.

A.5 Evaluation Metrics

Let Ω denote the set of all valid sample, forecast-step, and node triples in the evaluation set. We compute the following metrics in original flow units:

$$\mathrm{MAE} = \frac{1}{|\Omega|} \sum_{(n,k,i)\in\Omega} \left|\hat{y}_{n,i}[k] - y_{n,i}[k]\right|, \tag{17}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\Omega|} \sum_{(n,k,i)\in\Omega} \left(\hat{y}_{n,i}[k] - y_{n,i}[k]\right)^2}. \tag{18}$$

Table 4: Default implementation and training settings of DeCoDy.

Setting	Value
Historical condition dimension	48
Future conditional query dimension	16
Historical patch length	6 time steps
Fixed periodic background length	2016 time steps
Periodic-background shrinkage λ	5
Delay set $\mathcal{D}$	$\{1, 2, 3, 6, 12\}$
Coupling gain bound ρ	0.8
Optimizer	Adam
Maximum OneCycle learning rate	0.002
Weight decay	0
Gradient clipping threshold	5
Effective training batch size	16
Sample presentations per round	2048
Optimizer updates per round	128
Default maximum training rounds	40
Early-stopping patience	5
Relative improvement threshold	0.001
Minimum solver updates	4
Solver error-check interval	4 updates
Default maximum solver updates	64
Solver error tolerance in original flow units	0.05

Overall metrics aggregate valid targets over the complete forecast horizon. Daily metrics for seven-day forecasting instead divide the same seven-day forecast into seven consecutive 24-hour intervals and aggregate errors separately within each interval; these results do not correspond to seven independently trained forecasting tasks.

Hyperparameter analyses report metrics on the complete validation set. Current results use a single random seed and do not support estimating statistical significance across seeds.

B Details of Data Observations and Case Analysis

B.1 Temporally Staggered and Persistent Node Deviations

Figure 1(a) uses observations from 07:00 to 11:00 on February 18, 2021, in SD. Target node B has station ID 1108441; its three graph neighbors D, C, and A have IDs 1122928, 1126553, and 1125457, respectively.

For each node, the median flow at corresponding times on other dates with the same day of the week provides a descriptive reference. The figure displays percentage deviations of observed flow from this reference. A persistent-drop onset is defined as the first point in a sequence of three consecutive five-minute observations below 75% of the reference. The onsets for D, C, B, and A are 08:00, 08:05, 08:10, and 08:30, respectively.

Shading illustrates the temporal overlap among earlier deviations, target changes, and subsequent neighbor changes. This analysis describes the ordering and persistence of observed changes; differences between threshold-defined onset times are not interpreted as physical propagation times or used to determine causal directions between nodes. This median reference differs from the training-set shrinkage-based weekly background used by the model.

B.2 Evolving Inter-Node Associations Across Intraday Periods and Days

Figure 1(b) uses the fixed node pair C/B and displays continuous observations from 12:00 on January 25 to 12:00 on January 26, 2021. Deviations are obtained by subtracting the periodic background constructed exclusively from the training interval from observed flows.

The case is selected before comparing local correlations. We examine midnight boundaries within the training interval, requiring both nodes to maintain deviations of the same sign in all four con-

secutive half-hour intervals spanning one hour before and after midnight, with absolute deviations of at least 10% relative to the background. Negative deviations are prioritized, and cases are ranked by the minimum deviation magnitude across both nodes and these intervals. The resulting display window is an exploratory case rather than a randomly sampled observation of the population.

Local lagged correlations are computed as

$$\mathrm{Corr}(e_C(t-\ell), e_B(t)), \tag{19}$$

where e_C, e_B are observed deviations in original flow units and ℓ ranges from 0 to 60 minutes in five-minute increments. Each correlation uses a three-hour window centered at the displayed time, with centers advancing every 15 minutes. Each window requires at least 24 valid observation pairs and a standard deviation of at least 2 original flow units for both deviation sequences.

Centered windows may contain observations on both sides of an intraday transition or day boundary. The heatmap therefore describes local association changes retrospectively; it is neither an online input available at the forecast origin nor evidence of an instantaneous relationship switch at a boundary.

B.3 Shared Periodic Profiles and Local Peak and Trough Variations

Figure 1(c) uses traffic observations at SD station 1118020 on February 15, February 22, March 1, and March 8, 2021. All four dates are Mondays within the training interval.

The reference curve is the median flow at corresponding five-minute positions over six Mondays preceding the displayed dates, with no overlap between reference and displayed dates. Curves use nonoverlapping 15-minute means, and the local peak interval from 05:30 to 09:00 is enlarged. The figure compares overall daily profiles and peak differences across dates and contains no model predictions.

B.4 Statistical Protocol for the Seattle Case

The main-text case uses Seattle I-405 node d405es01612, with a seven-day forecast originating at 04:00 on November 8, 2015. The analysis focuses on 06:00–20:00 on November 13, corresponding to forecast lead times of approximately 122–136 hours.

Displayed curves use 30-minute means, whereas cumulative flow deviations are calculated from the original five-minute data. For an analysis interval $\mathcal{I}$, the cumulative deviation relative to the periodic background is defined as

$$D(\mathcal{I}) = \frac{\sum_{k\in\mathcal{I}} z[k] - \sum_{k\in\mathcal{I}} P^{\mathrm{raw}}[k]}{\sum_{k\in\mathcal{I}} P^{\mathrm{raw}}[k]} \times 100\%, \tag{20}$$

where $z[k]$ denotes either the observed or predicted flow, and $P^{\mathrm{raw}}[k]$ is the model's periodic background restored to original flow units.

The historical Friday median curve uses corresponding times on the four Fridays preceding the forecast origin and is displayed separately from the model's periodic background. Rainfall comes from retrospective observations at BFI station and is used only to describe the case context. Reports contain precipitation accumulated since the hourly reset; multiple reports within the same hour are not summed. This case illustrates correspondence between predicted trajectories and observed deviations, rather than establishing a causal relationship between rainfall and flow changes.

C Theoretical Proofs

C.1 Proof of Proposition 1

Consider the multi-delay response in Equation (9). Since $d_i > 0$, the coefficients in Equation (10) satisfy $0 < a_i = e^{-d_i} < 1$ and $b_i = (1-a_i)/d_i > 0$. Let

$$B = \max\left\{R_0, \frac{U}{1-\rho}\right\}.$$

The residual magnitudes on the historical boundary are bounded by $R_0 \le B$. We proceed by induction over future time steps. Suppose that all response magnitudes before time k are at most B. Because every delay $\ell \in \mathcal{D}$ is a positive integer, each $r_j[k-\ell]$ belongs either to the historical boundary or to a future response covered by the induction hypothesis. By the triangle inequality and Equation (7),

$$\begin{aligned}
|r_i[k]| &\le a_i |r_i[k-1]| + b_i \left(|s_i[k]| + \sum_{j \in \mathcal{N}_i} \sum_{\ell \in \mathcal{D}} |A_{ij\ell}[k]|\, |r_j[k-\ell]| \right) \\
&\le a_i B + \frac{1-a_i}{d_i} \left(d_i U + \rho d_i B \right) \\
&= a_i B + (1-a_i)(U + \rho B) \\
&\le a_i B + (1-a_i) B = B,
\end{aligned}$$

where the final inequality uses $U \le (1-\rho)B$. Induction therefore bounds all node response magnitudes by B over any finite forecast horizon, proving Equation (8). The proof requires only a bound on the absolute coupling row sums; it does not require nonnegative or time-invariant coupling coefficients.

Here, R_0 need only cover the historical boundary accessed by the solver. For a given horizon, the response bound depends on the normalized driving bound U over that horizon. If driving inputs across different horizons share a common finite bound, the response bound also holds uniformly across those horizons. This proposition concerns the exact response of the dynamical equations; numerical error from finitely many parallel iterations is controlled separately by Equation (14).

C.2 Proof of Proposition 2

Finite delay-path expansion. Coupling coefficients are generated before solving and remain given during a single solve, so both $\mathcal{M}$ and $\mathcal{K}$ are linear operators on future responses. For any future trajectory $\mathbf{z}$, future neighboring influences are

$$(\mathcal{M}\mathbf{z})_i[u] = \sum_{j \in \mathcal{N}_i} \sum_{\substack{\ell \in \mathcal{D} \\ u-\ell \ge 1}} A_{ij\ell}[u] z_j[u-\ell].$$

Historical boundary contributions are already included in $\mathbf{r}^{\mathrm{base}}$ and are not part of this linear operator. By the definition of the causal convolution,

$$(\mathcal{K}\mathbf{z})_i[k] = \sum_{u=1}^{k} b_i a_i^{k-u} \sum_{j \in \mathcal{N}_i} \sum_{\substack{\ell \in \mathcal{D} \\ u-\ell \ge 1}} A_{ij\ell}[u] z_j[u-\ell].$$

The input time index of every nonzero term satisfies $u - \ell \le k - \ell_{\min}$. Thus, each application of $\mathcal{K}$ advances the time index from input to output by at least $\ell_{\min}$: $\mathcal{K}^p$ can map an input at time t only to outputs at times no earlier than $t + p\ell_{\min}$. For $p = Q = \lceil H/\ell_{\min} \rceil$, we have $t + Q\ell_{\min} \ge 1 + H$, which lies beyond the future interval $1, \dots, H$. Hence $\mathcal{K}^Q = 0$.

It follows that

$$(I - \mathcal{K}) \sum_{p=0}^{Q-1} \mathcal{K}^p = I - \mathcal{K}^Q = I,$$

so $I - \mathcal{K}$ is invertible and Equation (11) has the unique solution

$$\mathbf{r}^\star = (I - \mathcal{K})^{-1} \mathbf{r}^{\mathrm{base}} = \sum_{p=0}^{Q-1} \mathcal{K}^p \mathbf{r}^{\mathrm{base}}.$$

This finite expansion follows from positive delays and a finite forecast horizon and does not itself require a contraction condition.

Computable contraction bound. To obtain an error guarantee for early stopping, define

$$q = \max_{i, 1 \le k \le H} \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \sum_{\ell \in \mathcal{D}} |A_{ij\ell}[k]| < \rho < 1. \tag{21}$$

Using $b_i = (1 - a_i)/d_i$, any trajectory $\mathbf{z}$ satisfies

$$\begin{aligned}|(\mathcal{K}\mathbf{z})_i[k]| &\le \sum_{u=1}^{k} b_i a_i^{k-u}\, q d_i \|\mathbf{z}\|_\infty \\ &= q(1 - a_i^k)\|\mathbf{z}\|_\infty \le q\|\mathbf{z}\|_\infty.\end{aligned}$$

Therefore, $\|\mathcal{K}\|_\infty \le q < 1$, so the full-horizon fixed-point mapping is a contraction.

A posteriori error certificate. Let $\boldsymbol{\delta}^{(m)} = \mathbf{r}^{(m)} - \mathbf{r}^{(m-1)}$ for $m \ge 1$. The fixed-point update gives $\boldsymbol{\delta}^{(m+1)} = \mathcal{K}\boldsymbol{\delta}^{(m)}$, and hence

$$\|\boldsymbol{\delta}^{(m+t)}\|_\infty \le q^t \|\boldsymbol{\delta}^{(m)}\|_\infty, \qquad t \ge 1.$$

Summing subsequent iteration differences yields

$$\begin{aligned}\|\mathbf{r}^\star - \mathbf{r}^{(m)}\|_\infty &\le \sum_{t=1}^{\infty} \|\boldsymbol{\delta}^{(m+t)}\|_\infty \\ &\le \sum_{t=1}^{\infty} q^t \|\boldsymbol{\delta}^{(m)}\|_\infty = \frac{q}{1-q}\|\mathbf{r}^{(m)} - \mathbf{r}^{(m-1)}\|_\infty.\end{aligned}$$

This proves Equation (14). The bound applies to fixed-point iteration with the exact operator for given parameters; it does not include rounding errors from finite-precision FFT computations.

D Solver Complexity Analysis

Let N denote the number of nodes, E the number of effective directed edges, $K = |\mathcal{D}|$ the number of delays, H the forecast horizon, and J the number of full-horizon updates actually executed. The following analysis concerns a single sample and treats conditional parameter generation separately from response solution.

Graph message computation. Each update computes neighboring influences for every effective edge, delay, and future time position, with computational complexity

$$O(EKH). \tag{22}$$

Local response computation. Node-local exponential responses are computed using zero-padded FFTs. Across all nodes, the computational complexity per update is

$$O(NH \log H). \tag{23}$$

Thus, after future driving inputs and coupling parameters have been generated, the response solver executes J full-horizon updates at a computational cost of

$$O(J(EKH + NH \log H)). \tag{24}$$

This expression explicitly retains the iteration count J. The actual count depends jointly on the error tolerance and coupling strength; finite-horizon solvability alone does not imply that it is a constant independent of the forecast horizon.

Explicitly storing edge–delay coupling coefficients over the entire forecast horizon requires $O(EKH)$ memory, while node responses and FFT work arrays require $O(NH)$ memory. Training additionally stores conditional encodings and intermediate quantities for backpropagation through the iterations. Actual memory consumption depends on batch size, iteration count, and the activation-checkpointing strategy.

Full-horizon updates jointly compute future positions through graph messages and convolutions, but their theoretical operation count does not directly establish a wall-clock advantage. Appendix E reports solver timings and numerical errors for the same response equation under fixed hardware, precision, and batch size; these measurements do not constitute end-to-end speed comparisons between different forecasting models.

E Numerical Validation and Computational Efficiency of the Solver

To examine the computational cost and numerical accuracy of full-horizon parallel solution, we use the 112-node PEMS08 subset, fix the trained seven-day forecasting model and its parameters, and compare parallel solution with sequential time-step recurrence for the same response equation.

Measurements are performed on an NVIDIA GeForce RTX 4060 Laptop GPU with a batch size of 4 and FP32 computation. Inputs are fixed windows from the training interval, with no additional parameter training. Future driving inputs and coupling coefficients are generated beforehand. Timing covers only response solution, excluding data preprocessing, CPU-to-GPU transfer, and coefficient generation. Reported times are medians of five repeated measurements after warmup.

E.1 Solver Efficiency at Different Horizon Lengths

We use the first 288, 864, and 2016 steps of coefficients generated by the fixed seven-day model to compare the two solution methods. Shorter intervals measure solver scaling and do not correspond to separately trained one-day or three-day forecasting models. Results are reported in Table 5. Under the measured configurations, parallel solution substantially reduces response computation time while maintaining small numerical differences from sequential recurrence. The time ratios (sequential divided by parallel) concern the current hardware and the two solver implementations; they are not end-to-end speedup factors of the complete forecasting model over other models.

Table 5: Solver efficiency for the same response equation. Maximum output differences are in original flow units.

Horizon	Parallel / ms	Sequential / ms	Time ratio	Max. difference
288	13.50	163.15	12.09	0.01260
864	13.70	492.86	35.98	0.01257
2016	32.54	1136.20	34.92	0.01262

E.2 Solver Tolerance and Measured Numerical Error

To further examine the stopping condition, we fix the same checkpoint and seven-day response coefficients and set the tolerance in original flow units to 0.01, 0.05, and 0.2. The numerical reference uses FP64 sequential recurrence to reduce rounding error in the reference computation itself. Results are reported in Table 6. On this fixed batch, tightening the tolerance increases the update count and reduces measured numerical error. Tolerances of 0.05 and 0.2 both stop at update 12 and therefore produce the same response. Since the stopping condition is checked every four updates, different tolerances need not yield different iteration counts.

Table 6: Update counts, numerical errors, and times for different solver tolerances. Maximum errors are measured against the FP64 reference.

Tolerance	Updates	Posterior bound	Max. error	Time / ms
0.01	16	0.001423	0.000491	42.07
0.05	12	0.033589	0.012615	32.46
0.20	12	0.033589	0.012615	32.39

These results evaluate numerical accuracy for a given response equation. They do not guarantee forecasting error relative to actual traffic flow or establish universal timing conclusions across datasets.

F Robustness Checks for Observational Analysis

To examine whether the observed local lagged associations depend on a particular statistical window, we use existing observations to analyze window length, date grouping, and temporal splits. All analyses use data from the training interval.

F.1 Local Window Length

Keeping the node pair, observations, and periodic background of Figure 1(b) unchanged, we compute local lagged correlations using centered windows of two, three, and four hours. Table 7 averages correlations over positive lags of 5–60 minutes and the specified time intervals. All three settings

Table 7: Mean lagged correlations under different local windows.

Window	Jan. 25, 17:30–18:30	Jan. 25, 21:30–Jan. 26, 04:30
2 hours	-0.0133	0.1670
3 hours	-0.0298	0.2846
4 hours	-0.0059	0.3790

yield a higher average lagged correlation at night than in the preceding interval, indicating that this direction of difference is not specific to the three-hour window. Correlation magnitudes vary with window length, so we use them to describe local association patterns rather than interpreting them as a unique interaction strength.

F.2 Date Types and Intraday Periods

Beyond the continuous case in the main text, we group observations for the same fixed node pair over the complete training prefix. The analysis interval contains 72 days: 51 Mondays through Fridays and 21 weekend days. Intraday periods are consistently defined as morning (06:00–10:00), midday (10:00–16:00), and evening (16:00–20:00).

We first construct lagged observation pairs on the complete timeline and then group them by the date type and period of the target timestamp, avoiding artificial lag correspondences caused by concatenating different dates. Each group's association profile is summarized by the mean of 13 correlations at lags from 0 to 60 minutes in five-minute increments. We perform 2000 resampling iterations using calendar weeks as blocks. Results are reported in Table 8. The directions of these

Table 8: Mean correlation differences between date types and periods, with week-block resampling intervals.

Comparison	Mean difference	95% interval
Mon.–Fri. morning − weekend morning	-0.3278	$[-0.4249, -0.1626]$
Weekend morning − weekend midday	0.1832	$[0.1026, 0.3456]$
Weekend morning − weekend evening	0.0919	$[0.0484, 0.1190]$
Weekend midday − weekend evening	-0.0914	$[-0.2103, -0.0173]$

differences remain consistent in the first and second 36-day segments and in leave-one-week-out checks, supporting the observation that residual associations for the fixed node pair vary with date type and intraday period. Other group comparisons do not show equally stable directions, so we do not generalize this conclusion to stable differences between all periods.

These intervals are exploratory descriptions of uncertainty conditional on a fixed periodic background, without multiple-comparison correction. The analysis concerns a fixed node pair and a limited observation interval. Statistical associations are not interpreted as causal propagation or used to infer population-level conclusions for the entire road network.

G Algorithm and Supplementary Implementation Details

Algorithm 1 presents the complete DeCoDy prediction procedure. Driving and coupling coefficients for all future positions are generated once and held fixed during response solving. We omit the batch dimension; E is the number of effective directed edges and $K = |\mathcal{D}|$ is the number of delays.

Table 9: Main symbols in the algorithm.

Symbol	Meaning or shape
L, H, N	History length, forecast horizon, and number of nodes
P	Training-estimated periodic background, $2016 \times N$
$\mathbf{r}^h$	Historical residuals, $L \times N$
$\mathbf{h}_i, \mathbf{q}_i[k]$	Historical representation $\mathbb{R}^{48}$, future query $\mathbb{R}^{16}$
$d_i, s_i[k]$	Positive decay rate and node driving term
$A_{ij\ell}[k]$	Sparse edge–delay coupling, $E \times K \times H$
$\mathcal{H}, \mathcal{M}$	Causal response convolution and future multi-delay messaging
$\mathcal{K} = \mathcal{H} \circ \mathcal{M}$	Full-horizon linear coupling operator
$q \in [0, 1)$	Upper bound on the contraction factor
$\mathbf{r}^{(m)}$	Future residuals at iteration m, $H \times N$
ε_{raw}	Solver tolerance in raw flow units, default 0.05

Algorithm 1: DeCoDy Full-Horizon Prediction

Input: Historical traffic X and mask M^h; historical external information E^h and masks; historical and future calendars C^h, C^f; graph G; training statistics P, μ_x, σ_x; parameters θ; horizon H; $\varepsilon_{\text{raw}}, J_{\min}, J_{\max}, J_{\text{check}}$.
Output: Forecast $\widehat{Y} \in \mathbb{R}^{H \times N}$.

Standardize historical traffic; fill missing entries with standardized zeros and retain masks.
Retrieve historical background P^h and future background P^f; compute $r_i[t] = x_i[t] - P_i^h[t]$.
Encode historical traffic, residuals, masks, and calendars, and aggregate graph-neighborhood, weather, and incident information into $\mathbf{h}_i$.
Combine $\mathbf{h}_i$, future calendars, k/H, $P_i^f[k]$, and its differences to construct $\mathbf{q}_i[k]$.
Generate all $d_i, s_i[k], A_{ij\ell}[k]$ once and hold them fixed during solving.
Compute $a_i = e^{-d_i}$ and $b_i = (1 - a_i)/d_i$; construct causal response convolution $\mathcal{H}_i$ using zero-padded FFTs.
Compute historical messages $m_i^{\text{hist}}[k] = \sum_j \sum_{\ell \in \mathcal{D}: k-\ell \le 0} A_{ij\ell}[k] r_j[k-\ell]$.
Compute $r_i^{\text{base}}[k] = a_i^k r_i[0] + \mathcal{H}_i(s_i + m_i^{\text{hist}})[k]$.
Define $(\mathcal{M}z)_i[k] = \sum_j \sum_{\ell \in \mathcal{D}: k-\ell \ge 1} A_{ij\ell}[k] z_j[k-\ell]$ and $\mathcal{K} = \mathcal{H} \circ \mathcal{M}$.
Compute $q = \max_{i,k} d_i^{-1} \sum_{j,\ell} |A_{ij\ell}[k]|$.
Initialize $\mathbf{r}^{(0)} = \mathbf{r}^{\text{base}}$ and converged $=$ false.
for $m = 1, \ldots, J_{\max}$ **do**
  $\mathbf{r}^{(m)} = \mathbf{r}^{\text{base}} + \mathcal{K}(\mathbf{r}^{(m-1)})$.
  if $m \ge J_{\min}$ and $((m - J_{\min}) \bmod J_{\text{check}} = 0$ or $m = J_{\max})$ **then**
    $E_m = \sigma_x \frac{q}{1-q} \|\mathbf{r}^{(m)} - \mathbf{r}^{(m-1)}\|_\infty$.
    if $E_m \le \varepsilon_{\text{raw}}$ **then**
      converged $=$ true; **break**.
    end if
  end if
end for
If converged $=$ false, report nonconvergence and terminate.
Restore the flow scale: $\widehat{Y} = \max\{0, \sigma_x(P^f + \mathbf{r}^{(m)}) + \mu_x\}$.
return $\widehat{Y}$.

G.1 Graph Construction and Coupling Initialization

After removing self-loops, we form an undirected union by taking the maximum weight across the two directions, then expand each connection into two effective directions. These directions do not represent verified road travel directions. Coupling is initialized as

$$B_{ij\ell}^{(0)} = 0.1 w_{ji} \operatorname{Corr}(r_j[t-\ell], r_i[t]).$$

Correlations use valid pairs from the training interval only; entries with fewer than three valid pairs or a near-zero correlation denominator are set to zero. Couplings are learnable after initialization, and these correlations do not establish causality.

G.2 Missing Values and Periodic Background

Missing traffic is filled with the training mean, becoming zero after standardization, and masks distinguish it from observed zero flow. Historical boundary residuals subtract the periodic background from filled values and are not treated as observed data. The periodic background is estimated entirely from training data. Missing weekly-slot statistics fall back successively to the corresponding daily slot, the node mean, and the global mean.

G.3 Temporal Encoding and Response Solving

Historical calendars use sine and cosine encodings of daily and weekly phases, giving four dimensions. Future calendars use eight daily and three weekly Fourier harmonics, together with normalized lead time k/H and the periodic-background shape to construct future queries. The FFT length is the smallest power of two no smaller than $2H - 1$, yielding linear causal convolution.

Defaults are $J_{\min} = 4$, $J_{\max} = 64$, $J_{\text{check}} = 4$, and $\varepsilon_{\text{raw}} = 0.05$. Training backpropagates through the updates actually executed; the periodic background and standardization statistics remain fixed.