FCDC: FORECAST-CONDITIONED DELAY CORRECTION FOR TRAFFIC FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Traffic changes can be propagated between connected locations with a delay, so observations available at the forecasting origin may remain useful during the prediction interval. When these observations are used for correction, the already-predicted trajectory must be taken into account. In Forecast-Conditioned Delay Correction (FCDC), the complete multistep forecast is supplied directly to a compact signed readout, alongside local history and delayed evidence. The base forecast is formed by adding a global backbone prediction and a shared linear own-history prediction. Historical values are routed into an extended arrival window through learned edge-lag distributions; available lags are combined by a history-dependent gate, and messages are normalized over valid sources. All branches are trained jointly, without a cache of realized forecast errors. With local evidence fixed, correction signs can be reversed by changing the forecast input; such reversals are excluded by the corresponding unconditioned readout. FCDC and seven controlled variants are evaluated with five seeds on four PEMS datasets. FCDC's mean MAE is 0.62–2.29% lower than the best recorded external baseline on each dataset. Its average relative MAE reductions against variants without forecast input and without delay correction are 0.63% and 1.38%, respectively, with lower overall MAE in every matched-seed comparison.

1 INTRODUCTION

Recurring network patterns, individual sensor trajectories, and changes propagated between locations must be represented in traffic forecasts. Spatial dependence is modeled by graph recurrent and convolutional networks (Li et al., 2018; Wu et al., 2019), while broader spatiotemporal context is captured through attention (Liu et al., 2023; Jiang et al., 2023a). When delayed observations are used to correct a forecast, they can be assessed in relation to the prediction already made.

Under a candidate lag, a neighbor's observation available at the forecasting origin can be assigned to an arrival time within the prediction interval. In the base forecast, a similar change, persistence at the current level, or a change in the opposite direction may already be predicted. The sign and magnitude of the correction should therefore be determined with access to that forecast.

Delay-aware historical features are incorporated into attention in PDFormer (Jiang et al., 2023a), and delays are learned within continuous graph dynamics in STDDE (Long et al., 2024). In ResCAL, forecasts are corrected using previous errors and graph signals (Kim et al., 2022); in MURECAST, residual retrieval is combined with utility estimation and calibrated activation (Jin et al., 2026). The correction in FCDC is constructed from current observations and trained jointly with the predictor.

The signed correction readout in FCDC (Figure 1) is conditioned directly on the complete multistep forecast being revised. This forecast is formed by adding the global prediction to a shared linear prediction of each node's own history. The sum, local history, and delayed observations are then passed to the readout. All branches are trained jointly, with gradients propagated through both the additive forecast and its conditioning path. With local evidence fixed, correction-sign changes are permitted by explicit forecast input but excluded by the corresponding unconditioned head (Proposition 4).

Historical observations are routed through positive candidate lags over an arrival window that can be extended into the prediction interval. The routed observations are combined into delayed evidence

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.

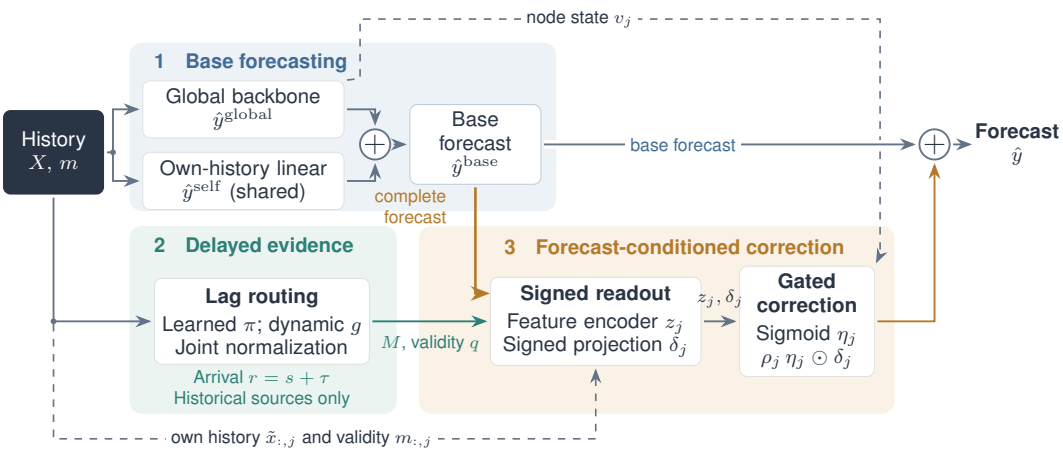


Figure 1: Overview of FCDC. The base forecast is the sum of global and shared own-history predictions. Historical observations are routed to delayed arrival slots and normalized over valid sources. The complete forecast (amber), local history, messages, and masks are supplied to the signed readout. Its correction is gated, masked by message availability, and added to the base forecast. Auxiliary inputs are indicated by dashed arrows.

using edge-specific lag priors and a history-dependent lag gate. Every source is restricted to observed history; no new observations are used for correction at any horizon.

Messages are jointly normalized over valid source contributions, and correction is disabled without incoming messages. The global forecast is recovered exactly by zero initialization of the own-history and correction output projections.

FCDC and seven controlled variants are evaluated with five seeds on PEMS03, PEMS04, PEMS07, and PEMS08. On each dataset, FCDC's mean MAE is below the eleven recorded external baseline values, and overall MAE is lower than each variant at every matched seed. Forecast conditioning, arrival coverage, local residual capacity, and fixed-uniform lag weights are examined through these controls. For four fixed models, correction signs are reversed by changing only the readout's forecast input. Input sensitivity and correction outcomes are analyzed in Sections 4.5 and 4.6, respectively; lag weights are examined in Appendix I.5.

2 Related Work

Spatiotemporal traffic representations. Diffusion convolution is combined with recurrent prediction in DCRNN (Li et al., 2018); adaptive adjacency and dilated temporal convolutions are used in Graph WaveNet (Wu et al., 2019). Diffusion and inherent components are separated through learned estimation and residual decomposition in D2STGNN (Shao et al., 2022b), while wavelet decomposition and spectral graph attention are used in STWave (Fang et al., 2023). Spatiotemporal embeddings for Transformer forecasting are examined in STAEformer (Liu et al., 2023). For large networks, spatial organization is used in PatchSTG, while adaptive graph convolution is regularized in RAGC (Fang et al., 2025; Wu et al., 2026).

Propagation delays. Historical traffic patterns are transformed into delay-aware attention features in PDFormer (Jiang et al., 2023a). In STDDE, delays are introduced into spatial-temporal differential equations, with predictions defined at continuous output times (Long et al., 2024). Scalar historical observations in FCDC are routed through a finite set of positive edge lags to historical and future arrival slots for forecast correction. The learned lag probabilities are interpreted as predictive routing preferences; physical travel-time identification is not assumed. These distinctions are summarized in Table 3.

Forecast correction. Subsequent errors are estimated from observed past forecast errors and graph signals in ResCAL (Kim et al., 2022). In MURECAST, candidates from a time-valid residual memory

are ranked by correction utility, and activation is calibrated (Jin et al., 2026). Delayed messages for the FCDC readout are computed from the current historical window and combined with the complete base forecast. Neither realized residuals from previous forecasts nor a retrieval memory are required.

Reusable forecasting components. The base predictor is built from shared linear temporal mappings (Zeng et al., 2023), bidirectional cross-attention with shared affinities (Hiller et al., 2024), and random-walk structural encodings (Rampášek et al., 2022). Its complete forecast is supplied directly to the signed correction readout.

3 Forecast-conditioned delay correction

3.1 Problem and information boundary

The sensor network is represented by a fixed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ with N sensors and nonnegative edge weights. An edge from source i to target j has weight $a_{ij} > 0$. At a forecasting origin, $Y \in \mathbb{R}^{H \times N}$ is predicted from the observed tensor $X \in \mathbb{R}^{T \times N \times C}$, containing the target signal and historical time-of-day and day-of-week features. History is indexed by $s \in \{0, \dots, T-1\}$ and prediction horizons by $h \in \{1, \dots, H\}$, where time $T + h - 1$ corresponds to horizon h. Batch indices are omitted. The constraint $H = T$ is required to align a learned output-position addition with temporal attention values.

The normalized target signal is denoted by $x_{s,j}$ and its observation-validity mask by $m_{s,j} \in \{0, 1\}$. In the local branches, $\tilde{x}_{s,j}$ is set to $x_{s,j}$ when $m_{s,j} = 1$ and zero otherwise. Validity is determined before normalization, with nonfinite values excluded; a normalized zero is not automatically missing. Unavailable observations outside the historical interval are marked by extending both m and $\tilde{x}$ with zeros. Future measurements are not imputed, and future target or calendar values are not used. Preprocessing and the global backbone's separate input assumptions are described in Appendix E.

3.2 Global and own-history base forecast

The historical tensor is mapped by two projections to time tokens $U^{(0)} \in \mathbb{R}^{T \times d}$ and node tokens $V^{(0)} \in \mathbb{R}^{N \times d}$:

$$U_s^{(0)} = \mathrm{Linear}_t(\mathrm{vec}\, X_{s,:,:}), \qquad V_j^{(0)} = \mathrm{Linear}_n(\mathrm{vec}\, X_{:,j,:}). \tag{1}$$

Historical calendar embeddings are added once at the input. Information is exchanged between the token sets through paired cross-attention layers. For one of n_h heads with width $d_h = d/n_h$, omitting projection details and learned additions,

$$S = Q_n K_t^\top / \sqrt{d_h}, \qquad O_n = \mathrm{softmax}_T(S) V_t, \qquad O_t = \mathrm{softmax}_N(S^\top) V_n. \tag{2}$$

The scores are normalized separately in the two directions, and both token sets are updated simultaneously, following the shared-affinity principle of bidirectional cross-attention (Hiller et al., 2024). In the final attention layer, node states $v_j \in \mathbb{R}^d$ are computed from node queries and time keys and values. The global prediction is $\hat{y}_j^{\mathrm{global}} = W_o v_j + b_o$. Layer-specific node additions are formed from learned node embeddings and random-walk return probabilities (Rampášek et al., 2022). The additions, residual updates, and flattening order are specified in Appendix C.

This forecast is supplemented by a direct own-history mapping:

$$\hat{y}_j^{\mathrm{self}} = W_s[\tilde{x}_{:,j}; m_{:,j}], \qquad \hat{y}_j^{\mathrm{base}} = \hat{y}_j^{\mathrm{global}} + \hat{y}_j^{\mathrm{self}}, \quad W_s \in \mathbb{R}^{H \times 2T}. \tag{3}$$

W_s is bias-free and shared across nodes. Valid standardized zeros are distinguished from unavailable observations by the mask channels. A direct temporal path is provided alongside the globally mixed representation.

3.3 Routing historical observations to delayed arrivals

Edge lag priors. The retained edge set is $\mathcal{E}_d = \{(i, j) : i \neq j,\ a_{ij} > 0\}$, with incoming weights normalized as $\bar{a}_{ij} = a_{ij} / \sum_{i':(i',j)\in\mathcal{E}_d} a_{i'j}$. No messages are assigned to nodes with an

empty incoming neighborhood. Each retained edge has learned logits $\theta_{ij} \in \mathbb{R}^K$ over positive lags $\tau \in \{1, \dots, K\}$:

$$\pi_{ij\tau} = \frac{\exp \theta_{ij\tau}}{\sum_{k=1}^{K} \exp \theta_{ijk}}. \tag{4}$$

After training, each edge's lag prior is fixed across samples, arrival positions, and forecast horizons; it does not depend on the current forecast. The lag support is constrained to $1 \le K < T$ and the arrival extension to $0 \le F \le \min(K, H)$. Arrival indices are $r \in \{0, \dots, T + F - 1\}$.

For each arrival and lag, a weighted numerator and its matching valid weight mass are computed as

$$\nu_{rj\tau} = \sum_{i:(i,j)\in\mathcal{E}_d} \bar{a}_{ij} \pi_{ij\tau} m_{r-\tau,i} \tilde{x}_{r-\tau,i}, \tag{5}$$

$$\omega_{rj\tau} = \sum_{i:(i,j)\in\mathcal{E}_d} \bar{a}_{ij} \pi_{ij\tau} m_{r-\tau,i}. \tag{6}$$

Only source times $0 \le r - \tau < T$ are included. For example, the last historical observation at time $T - 1$ is routed to arrival $T + 1$ under lag two when that lag and arrival slot are retained. Future arrival slots are filled by shifting available history; future source observations are neither read nor generated.

History-dependent lag selection. The available lags are $\mathcal{A}_{rj} = \{\tau : \omega_{rj\tau} > 0\}$. The local target context and *unnormalized* lag numerators are supplied to a shared two-layer MLP, $f_g : \mathbb{R}^{K+1} \to \mathbb{R}^K$:

$$\ell_{rj} = f_g([c_{rj}; \nu_{rj1}; \dots; \nu_{rjK}]), \qquad c_{rj} = \begin{cases} \tilde{x}_{rj}, & r < T, \\ \tilde{x}_{T-1,j}, & r \ge T. \end{cases} \tag{7}$$

At future arrival positions, the last local value is repeated only as context for the dynamic lag gate; no additional propagated observations are created. The normalized lag gate is

$$g_{rj\tau} = \begin{cases} \exp \ell_{rj\tau} / \sum_{k\in\mathcal{A}_{rj}} \exp \ell_{rjk}, & \tau \in \mathcal{A}_{rj}, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

All entries are zero when $\mathcal{A}_{rj}$ is empty. At each node and arrival time, the dynamic lag gate is conditioned on observed history and shared across incoming edges at a given lag. The forecast is passed separately to the correction readout.

Joint normalization. The delayed scalar message M_{rj} and its validity q_{rj} are

$$D_{rj} = \sum_{\tau=1}^{K} g_{rj\tau} \omega_{rj\tau}, \qquad M_{rj} = \begin{cases} \dfrac{\sum_{\tau=1}^{K} g_{rj\tau} \nu_{rj\tau}}{D_{rj}}, & D_{rj} > 0, \\ 0, & D_{rj} = 0, \end{cases} \qquad q_{rj} = \mathbb{I}[D_{rj} > 0]. \tag{9}$$

The edge–lag mixture is normalized over available observations with the same effective weights in the numerator and denominator. Lag signals are not normalized independently before averaging. A valid message can be zero because of standardization or cancellation; its validity is therefore stored separately in q and cannot be inferred from M.

3.4 Conditioning a signed correction on the complete forecast

Messages and masks are split into historical arrivals, $(M^{\text{hist}}_{:,j}, q^{\text{hist}}_{:,j}) \in \mathbb{R}^T \times \{0,1\}^T$, and future arrivals, $(M^{\text{future}}_{:,j}, q^{\text{future}}_{:,j}) \in \mathbb{R}^F \times \{0,1\}^F$. The readout input is ordered as

$$u_j = [\tilde{x}_{:,j}; M^{\text{hist}}_{:,j}; m_{:,j}; q^{\text{hist}}_{:,j}; M^{\text{future}}_{:,j}; q^{\text{future}}_{:,j}; \hat{y}^{\text{base}}_j] \in \mathbb{R}^{4T+2F+H}. \tag{10}$$

The last block is the full prediction from Equation (3), including the own-history component, and is neither detached nor treated as a masked observation. The correction is computed through a two-layer encoder, a signed output projection, and a horizon-specific sigmoid correction gate:

$$z_j = W_2 \,\mathrm{ReLU}(W_1 u_j + b_1) + b_2 \in \mathbb{R}^{d_c}, \tag{11}$$

$$\delta_j = W_\Delta z_j, \qquad \eta_j = \mathrm{sigmoid}(W_v v_j + b_v + W_z z_j), \tag{12}$$

$$\hat{y}_j = \hat{y}^{\text{base}}_j + \rho_j \, \eta_j \odot \delta_j, \qquad \rho_j = \mathbb{I}\left[\sum_{r=0}^{T+F-1} q_{rj} > 0\right]. \tag{13}$$

Here $W_\Delta, W_z \in \mathbb{R}^{H\times d_c}$ and $W_v \in \mathbb{R}^{H\times d}$. The prediction can be increased or decreased by the correction at each horizon. Message availability is checked per node through ρ_j, so all H outputs can be adjusted when any message is valid. Explicit future arrival slots are limited by F; corrections for later horizons are inferred from the whole feature vector.

Delayed evidence and the prediction being modified are combined in the readout input in Equation (10). Without the forecast block, the sign of δ_{jh} is fixed when local features are fixed; only its magnitude can be rescaled by changes in v_j through the positive sigmoid correction gate. Once the forecast is included in the encoder input, different correction signs can be produced for different base predictions (Proposition 4). The sign restriction imposed by using hidden states only to gate a local adjustment is thus removed by conditioning on the decoded forecast, including its direct temporal component.

3.5 Learning and architectural properties

All parameters are optimized jointly using masked mean absolute error. The set of valid batch–horizon–node target indices is denoted by $\mathcal{I}$, and inverse target scaling by $\mathcal{S}^{-1}$. For $|\mathcal{I}| > 0$,

$$\mathcal{L} = \frac{1}{|\mathcal{I}|} \sum_{(b,h,j)\in\mathcal{I}} \left| \mathcal{S}^{-1}(\hat{y}_{bhj}) - y^{\mathrm{raw}}_{bhj} \right|. \tag{14}$$

An entirely masked target batch is assigned zero loss. Ground truth is used only in the loss and is excluded from forward features. No separate residual-target dataset or pretrained correction stage is used. Gradients are propagated to the parameters of $\hat{y}^{\mathrm{base}}$ through its direct addition and, once active, its use in u_j.

W_s and W_Δ are initialized to zero, as are the columns of W_1 corresponding to $\hat{y}^{\mathrm{base}}$ and the final layer of f_g. The initial forecast is exactly the global forecast, and the initial dynamic lag gate is uniform over available lags. Gradients can be propagated through the internal paths of the correction branch only after the output projection becomes nonzero.

All message sources, including those assigned to future arrival indices, are drawn from history before the forecasting origin. Whenever $D_{rj} > 0$, M_{rj} is a convex combination of valid source values, so a common constant is preserved. The correction is exactly zero for a node with no valid delayed messages. Zero is also returned by the bias-free direct path for an entirely missing own-history vector. These properties are proved under finite-input assumptions in Appendix D. The forward computation is specified in Algorithm 1, and the computational costs are given in Appendix F.

4 Experiments

FCDC and seven variants are evaluated on four datasets with five seeds each (160 final test results). Five primary variants are used to examine the correction readout and its inputs. In two further variants, the lag prior or lag gate is fixed to uniform weights. External comparisons are based on 44 recorded results from eleven methods in BasicTS (Shao et al., 2024). Full protocols are given in Appendix G.

4.1 Evaluation protocol

PEMS03, PEMS04, PEMS07, and PEMS08 contain 358, 307, 883, and 170 traffic-flow sensors, respectively. For each model, 12 future steps are predicted from 12 historical steps at five-minute intervals (one hour each). The data are split chronologically in a $60/20/20$ training/validation/test ratio. Windows are drawn within each split, and target normalization is fitted to training data only. Flow, historical time of day, and historical day of week are supplied as inputs. The graph and global backbone are shared across variants. Each run is assigned one NVIDIA GeForce RTX 3090 GPU. Dataset sizes, preprocessing details, and parameter counts are given in Appendix G.

Training is performed for 100 epochs with Adam, masked MAE loss, weight decay 10^{-4}, and batch size 16. The initial learning rate is 10^{-4} for PEMS03 and 10^{-3} otherwise. It is multiplied by 0.10 at epochs 20, 25, and 30 for PEMS03 and at epochs 20 and 25 otherwise. A batch size of 64 is used for testing. Means and sample standard deviations are calculated from the final test results for seeds 42, 2024, 2025, 2026, and 2027. Only independent training seeds are treated as repeated runs. MAE,

Table 1: Comparison with eleven external methods. FCDC's five-seed means and sample standard deviations (SDs) are reported on separate rows; point values are retained for external methods. Means are rounded to two decimals and SDs to three. Lower is better, and MAPE is in percent. Best and second-best distinct rounded values are bold and underlined, respectively; ties are ranked equally. Protocols are listed in Appendix G.2.

	PEMS03			PEMS04			PEMS07			PEMS08		
Method	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MAPE ↓
GWNet	14.59	<u>25.24</u>	15.52	18.80	30.14	13.19	20.44	33.38	8.71	14.67	23.55	9.46
AGCRN	15.24	26.65	15.89	19.28	31.02	13.18	20.68	34.45	8.77	15.78	24.76	10.42
STID	15.33	27.40	16.40	18.35	<u>29.86</u>	12.50	19.61	32.69	8.31	14.21	23.35	9.32
D2STGNN	14.63	26.31	15.32	18.32	29.89	12.51	19.49	<u>32.59</u>	8.09	14.10	23.36	9.33
STAEformer	15.35	27.55	15.18	<u>18.22</u>	30.18	**11.98**	<u>19.14</u>	32.60	8.01	13.46	23.25	<u>8.88</u>
PDFormer	14.94	25.39	15.82	18.36	30.03	<u>12.00</u>	19.97	32.95	8.55	13.58	23.41	9.05
MegaCRN	<u>14.57</u>	25.87	<u>14.75</u>	18.72	30.53	12.77	19.83	32.91	8.36	14.75	23.73	9.48
STWave	14.94	26.98	15.17	18.61	30.53	12.53	19.88	33.74	8.30	13.53	23.66	8.96
HimNet	15.10	26.84	15.87	18.27	30.05	12.47	19.22	32.65	8.03	13.56	23.19	8.99
STDDE	16.87	27.39	17.20	21.79	33.54	15.55	25.27	38.50	11.27	16.79	25.91	11.50
PM-DMNet(R)	15.30	26.80	16.04	18.34	30.76	12.01	<u>19.14</u>	33.28	<u>7.98</u>	<u>13.39</u>	<u>23.17</u>	8.89
FCDC mean	**14.48**	**24.80**	**14.47**	**17.88**	**29.63**	**11.98**	**18.76**	**32.33**	**7.85**	**13.08**	**23.01**	**8.80**
FCDC SD	0.014	0.218	0.071	0.036	0.115	0.032	0.036	0.068	0.094	0.045	0.114	0.181

RMSE, and percentage MAPE are computed after inverse scaling, with raw zero targets masked. In ranked tables, the best and next distinct two-decimal means are marked in bold and underlined, respectively.

4.2 Comparison with external methods

The external methods are GWNet, AGCRN, STID, and D2STGNN (Wu et al., 2019; Bai et al., 2020; Shao et al., 2022a;b); Transformers STAEformer and PDFormer (Liu et al., 2023; Jiang et al., 2023a); and HimNet, MegaCRN, and recursive PM-DMNet(R) (Dong et al., 2024; Jiang et al., 2023b; Weng et al., 2025). STDDE is included for its explicit delay dynamics, and STWave for its wavelet decomposition and spectral attention (Long et al., 2024; Fang et al., 2023). Both are trained for 200 epochs with their respective losses and optimization settings; checkpoints are selected by validation MAE. The adaptations and protocols are described in Appendix G.2.

FCDC's mean MAE is lower than the recorded value for each baseline on every dataset (Table 1). Relative to the best baseline, the reductions are 0.62%, 1.88%, 1.97%, and 2.29% on PEMS03/04/07/08. MAE and RMSE are below the best baseline point values at all five seeds. The mean MAE reductions against STDDE and STWave are 20.00% and 3.97%, respectively, with lower mean RMSE and MAPE on every dataset as well. On PEMS04, FCDC's mean MAPE and STAEformer's recorded value are tied at the displayed precision. Recorded point values are used for the external methods, which were not repeated across these seeds.

4.3 Controlled variants

The backbone and direct own-history path are retained in every variant, with common parameters and initial forecasts matched. In *No delay correction*, the adapter and complete correction head are removed. In *No forecast input*, $\hat{y}^{\text{base}}$ is removed from the encoder; delayed messages and the sigmoid correction gate are retained, as in Proposition 4. In *Hidden-state input*, the 12-dimensional forecast is replaced with the 256-dimensional node state v_j, while the correction gate is unchanged. In *Historical arrivals only*, F is set to zero, while $K = 3$ and the other delay operations are retained. The latest neighbor observations are excluded from delayed messages because arrivals are restricted to $r < T$; the full history remains available to the backbone and own-history path. In *Parameter-matched MLP*, the delay branch is replaced with a shared two-hidden-layer ReLU MLP with input $[\tilde{x}_{:,j}; m_{:,j}; \hat{y}_j^{\text{base}}]$ and a zero-initialized output projection. Its parameter count is identical to that of the removed branch except for two parameters on PEMS04. Dimensions and parameter differences are given in Appendix G.

Table 2: Overall test MAE for FCDC and seven variants across five seeds (mean ± sample standard deviation). Means are rounded to two decimals and SDs to three. Lower is better. Best and second-best distinct rounded means are bold and underlined, respectively; ties are ranked equally. All three metrics are reported in Appendix Tables 8 and 10.

Variant	PEMS03	PEMS04	PEMS07	PEMS08
FCDC	**14.48** ± 0.014	**17.88** ± 0.036	**18.76** ± 0.036	**13.08** ± 0.045
No delay correction	14.67 ± 0.011	18.14 ± 0.006	18.96 ± 0.016	13.32 ± 0.019
No forecast input	14.57 ± 0.007	17.97 ± 0.004	<u>18.86</u> ± 0.007	13.20 ± 0.005
Hidden-state input	14.61 ± 0.010	<u>17.93</u> ± 0.009	18.95 ± 0.018	13.23 ± 0.013
Historical arrivals only	<u>14.54</u> ± 0.009	18.01 ± 0.009	18.99 ± 0.012	13.25 ± 0.004
Parameter-matched MLP	14.55 ± 0.007	18.02 ± 0.012	18.95 ± 0.013	<u>13.18</u> ± 0.012
Uniform lag prior	14.60 ± 0.010	18.09 ± 0.007	18.91 ± 0.010	13.23 ± 0.009
Uniform lag gate	14.61 ± 0.003	18.04 ± 0.007	19.00 ± 0.006	13.27 ± 0.004

In *Uniform lag prior*, each edge's lag distribution is fixed to $1/K$, while the dynamic lag gate is trained. In *Uniform lag gate*, equal weights are assigned to available lags, while the learned edge priors are retained. The graph, message masks, joint normalization, and forecast-conditioned readout are unchanged in both controls, including the sigmoid correction gate. Parameter differences and results for all three metrics are given in Appendix G.1.

4.4 Results and Interpretation

Explicit forecast input. Overall MAE is lower with FCDC than with every variant at all five matched seeds on each dataset. A mean MAE increase of 0.091–0.112 is observed when forecast input is removed (Table 2). The relative reductions with FCDC range from 0.51–0.85%, averaging 0.63%. For each dataset, the reduction is computed as $100(\mathrm{MAE}_{\mathrm{variant}} - \mathrm{MAE}_{\mathrm{FCDC}})/\mathrm{MAE}_{\mathrm{variant}}$ from the unrounded seed means; the four reductions are then averaged. Mean MAPE is also lower with forecast input on all four datasets, while PEMS03 RMSE is lower for the unconditioned variant (Table 8).

Forecast versus hidden-state input. When the decoded forecast is replaced by the full node state, mean MAE is higher on every dataset despite the addition of 3,904 parameters. The mean relative reduction with FCDC is 0.81%, with lower MAE at all matched seeds, although one PEMS04 difference is only 0.0023 MAE. On three datasets, mean MAE is also higher with hidden-state input than without forecast input. The lowest PEMS08 MAPE among the six primary models is obtained with hidden-state input.

Future arrivals and local capacity. Relative to historical arrivals only, a mean MAE reduction of 0.89% is obtained with FCDC, with the largest reduction on PEMS08 (1.23%). PEMS07 MAE is slightly higher for historical arrivals only than for no delay correction. At approximately the same correction parameter budget, mean MAE is lower with FCDC than with the local MLP by 0.75%. At all 20 matched dataset and seed pairs, lower MAE is obtained with the MLP than without delay correction; a further reduction is obtained with FCDC at all 20. Some of the gain is therefore retained with a local residual readout. Relative to no delay correction, FCDC's mean MAE reduction is 1.38%.

Uniform lag prior and lag gate. Overall MAE is lower with FCDC in all 20 matched dataset and seed pairs for each fixed-uniform control (Table 2). The mean relative reductions are 0.99% for the lag prior and 1.11% for the lag gate. With only the four added seeds included, the reductions are 1.02% and 1.14%, respectively. All three metrics are reported in Appendix G.1.

Forecast horizons and other metrics. After averaging over seeds, MAE is lower with FCDC in all 240 horizon comparisons with the five primary variants and all 96 with the uniform controls. At individual seeds, the corresponding counts are 1190/1200 and 480/480. The horizon means are reported in Appendix H. Lower mean RMSE on PEMS03 is obtained with no forecast input, historical

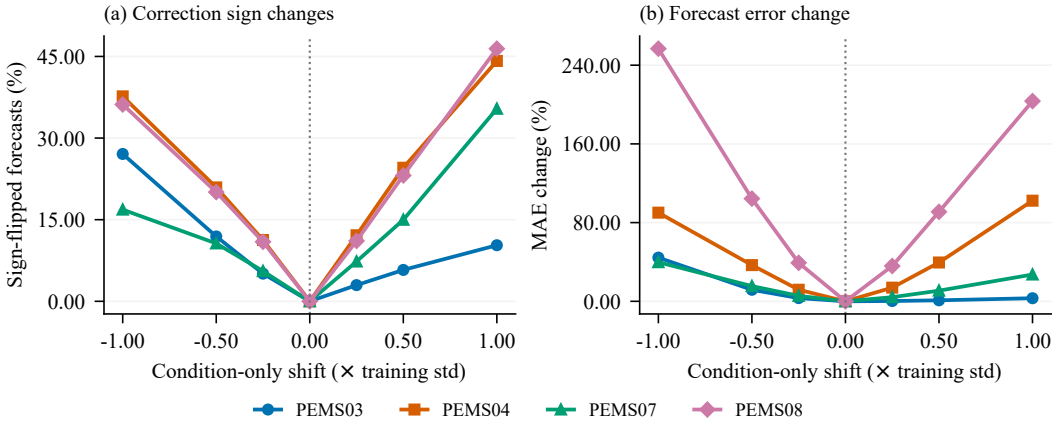


Figure 2: Frozen-readout sensitivity on complete test sets. Only the forecast supplied to the correction head is shifted; the additive prediction is held fixed. Left: fraction of valid forecasts with a correction-sign reversal relative to zero shift. Right: MAE increase relative to zero shift.

arrivals, the MLP, and uniform priors. PEMS04 MAPE is lower for historical arrivals, and PEMS08 MAPE is lower for hidden-state input and the MLP.

Some factors are changed together in these controls. With historical arrivals, arrival coverage and access to recent observations are both restricted; with the MLP, message access and readout structure are both changed. Prior learning and differences across lags and edges are changed together under uniform priors, as are average lag preference and input dependence under uniform gates.

4.5 Is the forecast used by the trained readout?

For each dataset, one validation-selected FCDC checkpoint is evaluated on every test window with its weights fixed. The conditioning forecast is shifted by the same amount at all 12 horizons; the additive base forecast, history, messages, and node states are held fixed. Correction-sign changes and forecast errors are reported in Figure 2. Across datasets, opposite correction signs are observed under the -1.00 and $+1.00$ training-standard-deviation shifts at 17.61–82.60% of valid forecast points, consistent with the sign changes permitted by Proposition 4. On every dataset, the lowest MAE among the tested shifts is recorded with unmodified input.

With the additive base forecast held fixed, MAE is increased by 0.57–19.43 when the forecast supplied to the readout is permuted across sensors, compared with 0.02–0.09 when delayed messages and their masks are permuted together. An increase of 0.44–1.75 is recorded when future-arrival values and masks are zeroed. Under these permutations, greater sensitivity is observed to the forecast input than to the sensor assignment of delayed messages. The procedures and their scope are described in Appendix I.3; subgroup results, successful and failed cases, and edge-level lag preferences are given in Appendix I.

4.6 Correction outcomes on unmodified inputs

Correction outcomes are evaluated on unmodified test inputs using the four fixed models from Section 4.5 and grouped into five categories in Figure 3. Across datasets, pointwise absolute error is reduced on 65.43–78.97% of valid forecasts and increased on 21.03–34.56%. Among all valid forecasts, 6.87–9.96% are worsened by overshoot and 12.75–24.60% by correction in the wrong direction. Category definitions, exact counts, and results at each horizon are provided in Appendix I.2. Computational costs are reported in Appendix J.

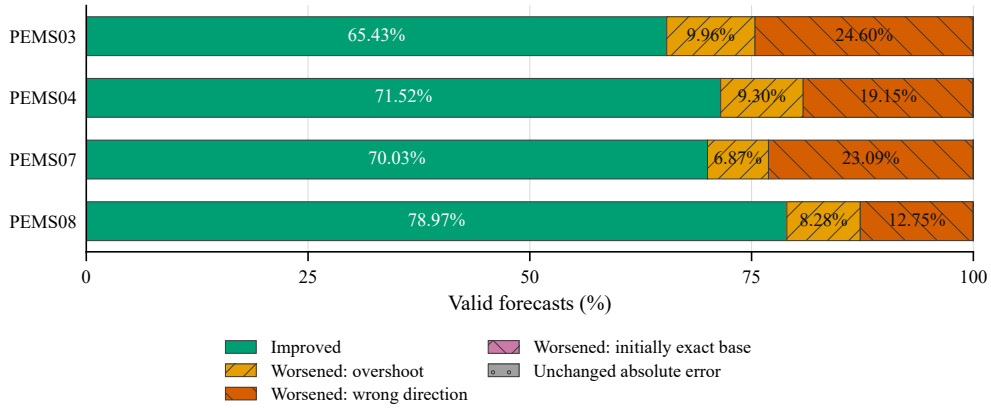


Figure 3: Correction outcomes on unmodified test inputs. Final and internal base forecasts from the same jointly trained FCDC are compared, with tolerance $\epsilon = 10^{-6}$ in raw flow units. Results for the independently trained no-correction model are reported in Table 2. Bar widths are computed from the unrounded counts in Table 16.

5 Conclusion

The signed correction in FCDC is conditioned on the complete multistep forecast and on historical observations routed through an extended arrival window. The readout and base predictor are trained jointly. Across four PEMS datasets, five-seed mean MAE is below the recorded values of all eleven external baselines. Overall MAE is also lower than for each of the seven variants at every matched seed, including the local MLP, the two fixed-uniform controls, and the variant without forecast input. With weights fixed, correction signs are reversed by changing only the readout's forecast input.

AI use statement

OpenAI Codex was used to assist with manuscript editing and formatting, including improvements to grammar, sentence structure, readability, and consistency of terminology, notation, and descriptions across sections. It was also used to assist with plotting scripts and the refinement of figure layouts, labels, and visual presentation. In addition, Codex was used to help identify potentially relevant recent studies and to assist with debugging code. All cited literature was independently verified by the authors, and AI-assisted code changes and manuscript edits were reviewed and validated by the authors. The authors take responsibility for the final content of the paper, including all text, claims, figures, and code produced with the assistance of generative AI.

Ethics statement

FCDC uses aggregate road-sensor time series and does not require person-level mobility records. Forecast errors may affect traffic management decisions, particularly in regions where sensor coverage is sparse or observations are missing. Before deployment, performance should be evaluated under local conditions and across regions with varying levels of sensor coverage. Appropriate data-use permissions should be verified, and forecast errors, distribution shifts, and sensor failures should be monitored during deployment.

Reproducibility statement

The forward computation and training objective are specified in Section 3 and Algorithm 1. Implementation details are given in Appendices C and E, with proofs and computational cost assumptions in Appendices D and F. Training configurations, evaluation metrics, and baseline settings are documented in Appendix G, and diagnostic procedures in Appendix I. In the supplementary materials, 44 individual efficiency measurements are provided alongside code for offline regeneration of tables and figures.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17804–17815, 2020. URL https://dl.acm.org/doi/abs/10.5555/3495724.3497218.

Zheng Dong, Renhe Jiang, Haotian Gao, Hangchen Liu, Jinliang Deng, Qingsong Wen, and Xuan Song. Heterogeneity-Informed Meta-Parameter Learning for Spatiotemporal Time Series Forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 631–641, 2024. doi: 10.1145/3637528.3671961.

Yuchen Fang, Yanjun Qin, Haiyong Luo, Fang Zhao, Bingbing Xu, Liang Zeng, and Chenxing Wang. When Spatio-Temporal Meet Wavelets: Disentangled Traffic Forecasting via Efficient Spectral Graph Attention Networks. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 517–529, 2023. doi: 10.1109/icde55515.2023.00046.

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. doi: 10.1145/3690624.3709177.

Markus Hiller, Krista A. Ehinger, and Tom Drummond. Perceiving Longer Sequences With Bi-Directional Cross-Attention Transformers. In *Advances in Neural Information Processing Systems*, volume 37, pp. 94097–94129, 2024. doi: 10.52202/079017-2985.

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. doi: 10.1609/aaai.v37i4.25556.

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. doi: 10.1609/aaai.v37i7.25976.

Xueting Jin, Jinfeng Xu, Qianxin Xie, and Yuxuan Zhang. MURECAST: Memory-Based Utility-Aligned Residual Evidence with Calibrated Activation for Selective Correction of Traffic-Flow Forecasts. *Symmetry*, 18(9):1485, 2026. doi: 10.3390/sym18091485.

Daejin Kim, Youngin Cho, Dongmin Kim, Cheonbok Park, and Jaegul Choo. Residual Correction in Real-Time Traffic Forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 962–971, 2022. doi: 10.1145/3511808.3557432.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

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, 2023. doi: 10.1145/3583780.3615160.

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. doi: 10.1145/3589334.3645688.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515, 2022. doi: 10.52202/068431-1054.

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. doi: 10.1145/3511808.3557702.

Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S. Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment*, 15(11):2733–2746, 2022b. doi: 10.14778/3551793.3551827.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):291–305, 2024. doi: 10.1109/TKDE.2024.3484454.

Wenchao Weng, Mei Wu, Hanyu Jiang, Wanzeng Kong, Xiangjie Kong, and Feng Xia. Pattern-Matching Dynamic Memory Network for Dual-Mode Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 26(9):13603–13617, 2025. doi: 10.1109/tits.2025.3564564.

Kaiqi Wu, Weiyang Kong, Sen Zhang, Zitong Chen, and Yubao Liu. Efficient Traffic Forecasting on Large-Scale Road Network by Regularized Adaptive Graph Convolution. In *2026 IEEE 42nd International Conference on Data Engineering (ICDE)*, pp. 835–848, 2026. doi: 10.1109/ICDE65706.2026.00068.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019. doi: 10.24963/ijcai.2019/264.

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. doi: 10.1609/aaai.v37i9.26317.

A Comparison of information flow

Table 3: Information flow in closely related methods. The comparison is limited to mechanisms. "Current forecast" is the prediction being modified at the present forecasting origin.

Method	Evidence and timing	Role in prediction
PDFormer	Delay-aware attention features are derived from historical traffic patterns.	Delay information is incorporated into representation learning.
STDDE	Learned delayed states are supplied to continuous graph dynamics.	Future outputs are obtained by evolving the graph dynamics.
ResCAL	Past observed residuals and graph signals.	An existing forecast is adjusted by a residual estimator.
MURECAST	Time-valid residual memory and query/base-context cues.	Retrieved corrections are selected using utility and calibrated activation.
FCDC	Current-window observations routed to historical and future arrival slots.	A jointly trained signed correction is conditioned on the complete current forecast.

B Complete forward algorithm

A single-sample forward pass is given in Algorithm 1; node operations are batched in the implementation. Before the forward pass, positive off-diagonal edges are retained and incoming weights are normalized as in Section 3.3. Masked numerators are supplied to the dynamic lag gate, and the combined base forecast is passed separately to the readout.

Algorithm 1 FCDC forward computation

Require: Historical tensor X, raw-derived mask m, graph A, lengths $T = H$, lag support K, extension F
Ensure: Forecast $\hat{Y} \in \mathbb{R}^{H \times N}$

Form time/node tokens and historical calendar additions from X
Run $L-1$ paired cross-attention blocks and one final spatial block
$\hat{y}_j^{\text{global}} \leftarrow W_o v_j + b_o$ for all nodes j
$\tilde{x} \leftarrow \text{where}(m \wedge \text{finite}(x), x, 0)$; update m accordingly
$\hat{y}_j^{\text{self}} \leftarrow W_s[\tilde{x}_{:,j}; m_{:,j}]$; $\hat{y}_j^{\text{base}} \leftarrow \hat{y}_j^{\text{global}} + \hat{y}_j^{\text{self}}$
$\pi_{ij:} \leftarrow \text{softmax}(\theta_{ij:})$ on positive off-diagonal edges
Initialize $\nu, \omega \in \mathbb{R}^{(T+F)\times N \times K}$ to zero
for $\tau = 1, \ldots, K$ **do**
  for $s = 0, \ldots, \min(T-1, T+F-1-\tau)$ **do**
    $r \leftarrow s + \tau$
    for each incoming edge (i, j) with $m_{s,i} = 1$ **do**
      $\nu_{rj\tau} \leftarrow \nu_{rj\tau} + \bar{a}_{ij}\pi_{ij\tau}\tilde{x}_{s,i}$
      $\omega_{rj\tau} \leftarrow \omega_{rj\tau} + \bar{a}_{ij}\pi_{ij\tau}$
    end for
  end for
end for
Compute local contexts c and gate logits ℓ using Equation (7); set $\mathcal{A}_{rj} \leftarrow \{\tau : \omega_{rj\tau} > 0\}$
Normalize g over available lags; set $g = 0$ if $\mathcal{A}$ is empty
Compute M, q with joint normalization in Equation (9)
for each node j **do**
  $u_j \leftarrow [\tilde{x}_{:,j}; M_{:,j}^{\text{hist}}; m_{:,j}; q_{:,j}^{\text{hist}}; M_{:,j}^{\text{future}}; q_{:,j}^{\text{future}}; \hat{y}_j^{\text{base}}]$
  Compute z_j, δ_j, η_j using Equations (11)–(12)
  $\rho_j \leftarrow \mathbb{I}[\exists r : q_{rj} = 1]$
  $\hat{y}_j \leftarrow \hat{y}_j^{\text{base}} + \rho_j \eta_j \odot \delta_j$
end for
return $\hat{Y} = [\hat{y}_1, \ldots, \hat{y}_N]$

C EXACT GLOBAL BACKBONE SPECIFICATION

Input ordering and calendar features. For time token s, $X_{s,:,:}$ is flattened with the channel index outermost and the node index innermost. For node token j, $X_{:,j,:}$ is flattened with the channel index outermost and the time index innermost. The two affine projections have input widths NC and TC, respectively. If p is the number of time slots per day, normalized calendar values are decoded as

$$\text{slot} = \text{round}(p \cdot \text{tod}) \bmod p, \qquad \text{week} = \text{round}(7 \cdot \text{dow}) \bmod 7. \tag{15}$$

A shared embedding table of size $7p \times d$ is indexed by $p \cdot \text{week} + \text{slot}$. Calendar features for each time token are taken at that time from the final node index; those for each node token are taken at the final historical time. Under the usual input convention, calendar features are identical across nodes. The time state is initialized as $\text{LN}(\text{Dropout}(U^{(0)} + E_t))$ and the node state as $\text{LN}(V^{(0)} + E_n)$, without input dropout on the node branch. Calendar embeddings are added only at the input.

Structural and position additions. The original supplied adjacency A^{rw} is used for structural encoding, with its directions, weights, and any diagonal entries retained. The transition matrix is $P = D_{\text{out}}^{-1} A^{\text{rw}}$, with zero rows for zero out-degree. The structural features are

$$R_j = [P_{jj}, (P^2)_{jj}, \ldots, (P^{d_{\text{rw}}})_{jj}]. \tag{16}$$

Self-loop removal and incoming-weight normalization for delay routing are performed separately from this preprocessing. At each layer, R is projected through

$$\psi(R)_j = \text{LN}_d(W_b \,\text{ReLU}(\text{LN}_{64}(W_a R_j + b_a)) + b_b)\,, \tag{17}$$

where the affine widths are $d_{\text{rw}} \to 64 \to d$. Softmax is applied over the *node axis*, separately for each feature, to obtain $\widetilde{R} = \text{softmax}_N(\psi(R))$. With a learned node matrix $E_N \in \mathbb{R}^{N \times d}$, the node addition is $e_N = \text{LN}(E_N \odot \widetilde{R})$. The other two additions are $e_T = \text{LN}(E_T)$ and $e_F = \text{LN}(E_F)$, where $E_T \in \mathbb{R}^{T \times d}$ and $E_F \in \mathbb{R}^{H \times d}$. Independent parameters are used at each layer. The learned sequences are deterministic position parameters.

Paired attention update. For each of the first $L - 1$ layers, the incoming time and node states are denoted by U, V. The queries, keys, and values are formed by affine projections and additions:

$$\begin{aligned} Q_n &= \text{Linear}_q(V) + e_N, & K_t &= \text{Linear}_k(U) + e_T, \\ V_t &= \text{Linear}_{vt}(U) + e_F, & V_n &= \text{Linear}_{vn}(V) + e_N. \end{aligned} \tag{18}$$

The temporal key and value projections are implemented by a single affine map with $2d$ outputs. After the projections are split into heads, Equation (2) is applied with attention-probability dropout sampled separately in the two directions. Head outputs are concatenated, then passed through separate affine projections and output dropout to obtain A_t, A_n. For either branch with input B and attention output A, the residual update is

$$B_{\text{res}} = B + A, \qquad B' = \text{LN}_{\text{out}}(B_{\text{res}} + \text{FFN}(\text{LN}_{\text{attn}}(B_{\text{res}})))\,, \tag{19}$$

where FFN consists of an affine $d \to 4d$ projection, ReLU, an affine $4d \to d$ projection, and dropout. The unnormalized B_{res} is retained in the skip connection. Both branches are updated from the incoming states, without using the other's updated state.

Final spatial output block. In the final layer, only attention from node queries to time keys and values is computed, with independent node, temporal-position, and output-position additions. The final v_j is obtained from the node version of Equation (19). No node–node self-attention layer is included. All historical time tokens are accessible to paired cross-attention; no triangular within-history mask is applied. Future observations are not accessed. With $H = T$, e_F and V_t are aligned along the time dimension.

D ARCHITECTURAL PROPERTIES AND PROOFS

Finite parameters and finite backbone and readout inputs are assumed in the proofs. Invalid observations are conditionally removed from local arithmetic. The proofs are stated in real arithmetic; implementation results are subject to floating-point rounding.

Proposition 1 (Historical information boundary)**.** *Parameters, graph, historical inputs, and the historical validity mask are held fixed. Every source value with nonzero effective weight in a delayed message is drawn from time $s < T$. In evaluation mode, the FCDC forecast cannot be changed by modifying future target or calendar values while the required tensor shapes are preserved.*

Proof. In Equations (5)–(6), $s = r - \tau$ is used with zero mask outside $0 \le s < T$. Future-arrival gate contexts are also derived from history. Backbone outputs are computed only from historical inputs and calendars, graph encodings, and learned parameters; future values are not read by output-position additions. The own-history branch and readout are computed from available inputs. Future inputs are used solely for shape and batch checks, so the forecast is independent of their values. In training mode, this independence is preserved when dropout randomness is fixed. Training labels are used in the loss, which is excluded from this forward-pass statement. □

Proposition 2 (Convex combination and constant preservation)**.** *If $D_{rj} > 0$, the message M_{rj} is a convex combination of its available historical source values. Its value is between their minimum and maximum. If all contributing values equal c, then $M_{rj} = c$. If $D_{rj} = 0$, the message and its validity are zero.*

Proof. For each incoming edge and lag, a nonnegative weight is defined by

$$w_{ij\tau}^{(r)} = \bar{a}_{ij}\pi_{ij\tau}g_{rj\tau}m_{r-\tau,i} \ge 0. \tag{20}$$

By substitution into Equation (9), $D_{rj} = \sum_{i,\tau} w_{ij\tau}^{(r)}$ and

$$M_{rj} = \sum_{i,\tau} \alpha_{ij\tau}^{(r)}\tilde{x}_{r-\tau,i}, \qquad \alpha_{ij\tau}^{(r)} = w_{ij\tau}^{(r)}/D_{rj}. \tag{21}$$

For positive D_{rj}, the coefficients are nonnegative, sum to one, and are nonzero only for valid historical sources. The convex-hull and constant-preservation results follow. When the denominator is zero, the message is set to zero and $q_{rj} = 0$ by the piecewise definition. The input-dependent lag gate is evaluated pointwise in this argument. □

Proposition 3 (Initialization and missing-evidence fallback)**.** *At initialization, $\hat{y}_j = \hat{y}_j^{\mathrm{global}}$ for every node. At any finite parameter setting, zero correction is assigned to a node without valid delayed messages. For an entirely invalid own-history window, zero is returned by the direct own-history branch.*

Proof. At initialization, $\hat{y}^{\mathrm{self}} = 0$ because $W_s = 0$, and $\delta = 0$ because $W_\Delta = 0$. Hence $\hat{y} = \hat{y}^{\mathrm{global}}$ by Equations (3) and (13). If $q_{rj} = 0$ for every arrival, then $\rho_j = 0$, and zero correction is enforced by conditional selection regardless of the finite encoder outputs. For an entirely invalid own history, both $\tilde{x}_{:,j}$ and $m_{:,j}$ are zero, so zero is returned by the bias-free mapping in Equation (3) for every W_s. □

Initial gradient flow. At $W_\Delta = 0$, the correction derivative with respect to W_Δ can be nonzero. Derivatives through that branch to the encoder, adapter, and gates are zero. After W_Δ is updated, gradients can be propagated through these paths, including the initially zero forecast-input columns of W_1. Gradients are also propagated to W_s and the backbone through the additive base forecast.

Proposition 4 (Sign restriction of the unconditioned readout)**.** *The local feature vector in Equation (10) before the forecast is appended is denoted by u_j^0 and held fixed with $\rho_j = 1$. In the readout obtained by omitting the forecast input, the correction sign at horizon h cannot be changed by varying the node state. With $d_c \ge 2$, corrections of opposite signs can be represented by the forecast-conditioned readout for base predictions on opposite sides of any fixed scalar a at that horizon.*

Proof. Without the forecast input, $z_j = f(u_j^0)$ and $\delta_{jh} = (W_\Delta z_j)_h$ are fixed. For finite logits, $0 < \eta_{jh} < 1$, so $\mathrm{sign}(\eta_{jh}\delta_{jh}) = \mathrm{sign}(\delta_{jh})$, independently of v_j. For the conditioned head, $\mathrm{ReLU}(\hat{y}_{jh}^{\mathrm{base}} - a)$ and $\mathrm{ReLU}(a - \hat{y}_{jh}^{\mathrm{base}})$ can be formed by two encoder units, with these coordinates preserved by W_2. The signed output is set to their second-minus-first difference, $\delta_{jh} = a - \hat{y}_{jh}^{\mathrm{base}}$,

and all sigmoid correction gate logits are set to zero. The correction is then $(a - \hat{y}_{jh}^{\text{base}})/2$, with opposite signs on either side of a. □

This proposition applies to the stated readout architecture rather than unrestricted forecasters.

E Implementation details and reference dimensions

Masks and scaling. The historical mask is computed before scaling, with nonfinite values excluded. Values matching a finite raw null sentinel are excluded using absolute tolerance 5×10^{-5} and zero relative tolerance. For a NaN sentinel, validity is determined by finiteness alone. In the local branches, finiteness is checked again and invalid values are conditionally replaced to avoid multiplying NaNs by zero. Valid standardized zeros are retained. The preprocessed tensor is supplied to the backbone without this observation-validity mask; finite inputs are required.

Normalization and numerical precision. Invalid source times and observations are excluded from both the value numerator and its mass. A zero denominator is replaced by one for division, and a zero numerator is returned at those entries. Positive denominators are left unclamped, so constants are preserved even for small positive masses. For half-precision and bfloat16 histories, edge sums, edge lag probabilities, and masses are accumulated in float32. The scalar message is then converted back to the history's numerical type. Validity is determined by positive mass and stored separately.

Initialization. Edge logits are initialized to zero. In the dynamic lag gate, first-layer weights are drawn from $\mathcal{N}(0, 0.02^2)$, while biases and final-layer weights are initialized to zero. The correction encoder's forecast-input columns are initialized to zero; ordinary linear-layer initialization is retained for other columns. The signed projection, sigmoid correction gate bias, and bias-free own-history weight are initialized to zero, while ordinary initialization is retained for the gate's node-state and encoded-feature projections. Calendar embeddings are drawn from $\mathcal{N}(0, 0.02^2)$; Xavier-normal initialization is used for learned node and position arrays. Attention and feed-forward weights are drawn from $\mathcal{N}(0, 0.02^2)$ with zero biases. Epsilon 10^{-5} is used in layer normalization.

Table 4: Architectural settings of the full FCDC model used in the four PEMS experiments. Only the components specified in Section 4.3 are changed in the variants.

Quantity	Configured value
Historical length T / forecast length H	$12/12$
Input channels C	3: target, time of day, day of week
Hidden width d / attention heads	$256/4$
Backbone depth L	5: four paired blocks, one spatial block
Random-walk encoding width d_{rw}	32
Candidate lags K / future arrival slots F	$3/3$
Dynamic lag gate width d_g / correction width d_c	$16/16$
Readout feature width $4T + 2F + H$	66
Dropout / time slots per day p	$0.10/288$

Tensor layouts and graph convention. Input and output tensors are arranged as $[B, T, N, C]$ and $[B, H, N, 1]$, respectively. A future tensor with shape $[B, H, N, C]$ is required for shape checks, but its values are not used. Delay edges are interpreted under the supplied source-row, target-column convention; symmetrization is not applied, and reverse edges are not added. At least one positive off-diagonal edge is required in the supplied graph; individual nodes with no incoming edges are allowed. Graph inputs are not used by the own-history path.

Limitations. General missing-data support for the backbone is not provided by local masking alone. Useful observations can be excluded from delayed messages by incorrect or missing graph edges, and lags beyond K are not explicitly modeled. Node-specific parameters, a fixed node ordering, and $H = T$ are assumed by the backbone. Architectural changes are therefore required for unseen nodes or unequal input and forecast lengths.

F COMPUTATIONAL COSTS

For batch size B and E retained edges, each paired block has attention cost $O(BNTd)$ and projection/feed-forward cost $O(B(N+T)d^2)$. Including input projections and all L layers, the dominant backbone cost is

$$O\big(BNTCd + BL\left[NTd + (N+T)d^2\right]\big)\,. \tag{22}$$

The interaction tensors have $O(Bn_hNT)$ elements per paired layer, where n_h is the number of heads. Structural projection and storage of node-specific parameters add to these costs.

Edge sums are accumulated in $O(BKTE)$, and dynamic lag gates are evaluated in $O(BN(T+F)Kd_g)$. Numerator and mass tensors with $O(BN(T+F)K)$ entries are materialized, together with edge-indexed intermediate tensors. Additional intermediates across lags may be retained by automatic differentiation. The own-history projection is computed in $O(BNTH)$. With $D_u = 4T + 2F + H$, the readout cost is

$$O\big(BN[D_u d_c + d_c^2 + Hd_c + Hd]\big)\,. \tag{23}$$

The cost of the global output projection is $O(BNHd)$. Random-walk powers are precomputed once, with cost dependent on sparse-matrix fill-in.

Additional parameter counts are $2TH$ for the own-history path, $EK + (2K+2)d_g + K$ for dynamic routing, and $D_u d_c + d_c^2 + 2d_c + 2Hd_c + Hd + H$ for the readout. Under the settings in Table 4, the respective counts are 288, $3E + 131$, and 4,812, with $Hd_c = 192$ added by forecast conditioning. The backbone is excluded from these counts; measured execution costs are reported in Appendix J.

G EXPERIMENTAL PROTOCOL

Data and scaling. Dataset lengths and window counts are reported in Table 5. For a series of S steps, validation and test are each allocated $\lfloor 0.2S \rfloor$ steps, with the remainder used for training. Contiguous 24-step windows are drawn at stride one within each split. Only flow is scaled, using a single mean and standard deviation computed across sensors over the first $\lfloor 0.6S \rfloor$ time steps. Calendar features are kept at their configured normalized values. Raw zeros are treated as missing within an absolute tolerance of 5×10^{-5}. The historical mask is recorded by the local branch before scaling (Appendix E). Graph processing differs for random-walk encoding and delayed routing, as described in Appendix C and Section 3.3.

Table 5: Dataset sizes and window counts under chronological splitting. Each window is restricted to one split, with 12 historical and 12 future steps.

Dataset	Sensors	Time steps	Train windows	Validation	Test
PEMS03	358	26,208	15,703	5,218	5,218
PEMS04	307	16,992	10,173	3,375	3,375
PEMS07	883	28,224	16,913	5,621	5,621
PEMS08	170	17,856	10,691	3,548	3,548

Metric aggregation. Let $\mathcal{I}_b$ be the valid target indices in evaluation batch b, $n_b = |\mathcal{I}_b|$, and e_i the prediction error after inverse scaling. The valid-target mean is obtained, up to floating-point error, by weighting each batch MAE or MAPE by n_b. For RMSE, averaging batch values differs from pooling squared errors:

$$R_{\text{batch}} = \frac{\sum_b n_b \sqrt{n_b^{-1} \sum_{i \in \mathcal{I}_b} e_i^2}}{\sum_b n_b}, \qquad R_{\text{pool}} = \sqrt{\frac{\sum_b \sum_{i \in \mathcal{I}_b} e_i^2}{\sum_b n_b}}. \tag{24}$$

In BasicTS (Shao et al., 2024), R_{pool} is used for overall and horizon-specific RMSE. The contribution $n_b r_b^2$ is accumulated for each batch RMSE r_b, and $\sqrt{\sum_b n_b r_b^2 / \sum_b n_b}$ is returned after all batches. Empty batches are assigned zero weight.

Means and sample standard deviations are computed from each run's overall metrics across five seeds. Horizon-wise summaries are computed separately across the same seeds. Relative MAE reductions are calculated from the two models' unrounded means on each dataset; the four dataset-wise reductions are weighted equally in the reported average.

Optimization. Results are reported for seeds 42, 2024, 2025, 2026, and 2027 for FCDC and all seven variants on each dataset: 160 runs in total. Models are trained for 100 epochs without early stopping. Adam is configured with $(\beta_1, \beta_2) = (0.90, 999/1000)$ and $\epsilon = 10^{-8}$; learning rates, weight decay, schedules, and batch sizes are given in Section 4.1. Training windows are shuffled, while validation and test windows are kept in chronological order. Validation MAE is evaluated after each epoch.

Historical-arrival control. With $F = 0$, arrival indices satisfy $r \le T - 1$. Since $r = s + \tau$ and $\tau \ge 1$, the retained source times satisfy $s \le T - 1 - \tau \le T - 2$. The latest neighbor observation at $s = T - 1$ is therefore excluded from every delayed-source route; for each lag τ, the final τ historical source times are excluded. With $F = K = 3$ in the full model, every historical source–lag pair has an arrival within the extended window, subject to graph connectivity and observation validity. Only routed source values are restricted; the full historical window is retained by the backbone and local own-history inputs.

Shared initialization and parameter control. For each ablation, the reference model is constructed before the designated module is replaced or removed. Common parameter values and initial forecasts are preserved; subsequent training trajectories may differ. The own-history and final correction projections are initialized to zero. When forecast inputs are removed, 192 encoder weights are dropped. When node-state inputs are substituted, 3,904 weights are added. With the future-arrival

Table 6: Total trainable parameter counts. The MLP and full-model counts are identical on three datasets and differ by two parameters on PEMS04.

Variant	PEMS03	PEMS04	PEMS07	PEMS08
FCDC	7,988,487	7,882,803	9,065,607	7,601,831
No delay correction	7,980,268	7,875,820	9,055,468	7,595,244
No forecast input	7,988,295	7,882,611	9,065,415	7,601,639
Hidden-state input	7,992,391	7,886,707	9,069,511	7,605,735
Historical arrivals only	7,988,391	7,882,707	9,065,511	7,601,735
Parameter-matched MLP	7,988,487	7,882,801	9,065,607	7,601,831

value and mask columns removed, the count is reduced by 96 weights. Total parameter counts are reported in Table 6.

The local MLP is configured with input width $D = 2T + H = 36$, layer widths $D \to h_1 \to h_2 \to H$, and a bias-free output layer, giving $(D+1)h_1+(h_1+1+H)h_2$ parameters. Integer widths are selected solely by parameter count to match the removed branch's budget, subject to $h_1/2 \le h_2 \le 2h_1$; balanced widths are preferred in ties (Table 7). Local history, its validity mask, and the base forecast are supplied as inputs. Network information from the shared graph-aware backbone is included through the forecast; delayed-neighbor inputs and incoming-message fallback are excluded.

Table 7: Local correction capacity control. The target budget is that of the delay adapter and gated readout, excluding the shared backbone and own-history path. Widths are selected using parameter counts alone.

Dataset	Delay branch	MLP	h_1	h_2
PEMS03	8,219	8,219	74	63
PEMS04	6,983	6,981	69	54
PEMS07	10,139	10,139	77	81
PEMS08	6,587	6,587	63	56

All three metrics for the five primary variants and FCDC are reported in Table 8.

Table 8: Overall test metrics for FCDC and the five primary controlled variants across five seeds (mean ± sample standard deviation). Means are rounded to two decimals and SDs to three. MAPE is in percent. Within each dataset, best and second-best distinct rounded means are bold and underlined, respectively; ties are ranked equally.

Dataset	Variant	MAE ↓	RMSE ↓	MAPE ↓
PEMS03	FCDC	**14.48** ± 0.014	24.80 ± 0.218	**14.47** ± 0.071
	No delay correction	14.67 ± 0.011	25.20 ± 0.018	<u>14.51</u> ± 0.007
	No forecast input	14.57 ± 0.007	24.70 ± 0.011	14.61 ± 0.004
	Hidden-state input	14.61 ± 0.010	24.91 ± 0.025	14.61 ± 0.006
	Historical arrivals only	<u>14.54</u> ± 0.009	**24.54** ± 0.019	14.55 ± 0.011
	Parameter-matched MLP	14.55 ± 0.007	<u>24.57</u> ± 0.012	14.75 ± 0.007
PEMS04	FCDC	**17.88** ± 0.036	**29.63** ± 0.115	<u>11.98</u> ± 0.032
	No delay correction	18.14 ± 0.006	30.14 ± 0.011	12.44 ± 0.006
	No forecast input	17.97 ± 0.004	<u>29.75</u> ± 0.011	12.07 ± 0.005
	Hidden-state input	<u>17.93</u> ± 0.009	29.77 ± 0.023	12.05 ± 0.004
	Historical arrivals only	18.01 ± 0.009	29.86 ± 0.026	**11.94** ± 0.006
	Parameter-matched MLP	18.02 ± 0.012	29.93 ± 0.011	12.11 ± 0.003
PEMS07	FCDC	**18.76** ± 0.036	**32.33** ± 0.068	**7.85** ± 0.094
	No delay correction	18.96 ± 0.016	32.70 ± 0.014	8.97 ± 0.006
	No forecast input	<u>18.86</u> ± 0.007	32.55 ± 0.014	8.73 ± 0.007
	Hidden-state input	18.95 ± 0.018	32.74 ± 0.024	8.13 ± 0.005
	Historical arrivals only	18.99 ± 0.012	32.88 ± 0.023	7.97 ± 0.004
	Parameter-matched MLP	18.95 ± 0.013	<u>32.40</u> ± 0.007	<u>7.87</u> ± 0.003
PEMS08	FCDC	**13.08** ± 0.045	**23.01** ± 0.114	8.80 ± 0.181
	No delay correction	13.32 ± 0.019	23.38 ± 0.013	11.89 ± 0.008
	No forecast input	13.20 ± 0.005	23.19 ± 0.003	9.66 ± 0.003
	Hidden-state input	13.23 ± 0.013	23.37 ± 0.024	**8.67** ± 0.004
	Historical arrivals only	13.25 ± 0.004	23.24 ± 0.010	9.17 ± 0.003
	Parameter-matched MLP	<u>13.18</u> ± 0.012	<u>23.18</u> ± 0.010	<u>8.73</u> ± 0.004

Table 9: Comparison of four- and five-seed summaries. Relative MAE reductions are calculated from dataset-level seed means, then averaged equally across the four datasets. A win is counted when lower overall MAE is obtained by FCDC at a matched dataset and seed. Seeds 2024–2027 are used in the four-seed column.

	Five seeds		Four seeds	
Variant	Reduction (%)	Wins	Reduction (%)	Wins
No delay correction	1.38	20/20	1.41	16/16
No forecast input	0.63	20/20	0.66	16/16
Hidden-state input	0.81	20/20	0.83	16/16
Historical arrivals only	0.89	20/20	0.93	16/16
Parameter-matched MLP	0.75	20/20	0.78	16/16
Uniform lag prior	0.99	20/20	1.02	16/16
Uniform lag gate	1.11	20/20	1.14	16/16

G.1 FIXED-UNIFORM LAG PRIOR AND LAG GATE

For both fixed-uniform controls, the backbone, own-history path, forecast-conditioned readout, graph edge weights, masks, and joint normalization are retained. In *Uniform lag prior*, each edge's prior is fixed to $\pi_{ij\tau} = 1/K$, while the dynamic lag gate is trained. In *Uniform lag gate*, edge-specific lag priors remain learned, while $g_{rj\tau} = 1/|\mathcal{A}_{rj}|$ is fixed for available lags and zero otherwise, with all entries zero for an empty set. Availability can still vary across positions. The readout's sigmoid correction gate is unchanged in this control. With the prior fixed, EK trainable parameters are removed; with the lag-gate MLP replaced, 131 are removed under the reported configuration.

Table 10: FCDC and the fixed-uniform controls across five seeds (mean ± sample standard deviation). Means are rounded to two decimals and SDs to three. MAPE is in percent. Within each dataset, best and second-best distinct rounded means are bold and underlined, respectively; ties are ranked equally.

Dataset	Variant	MAE ↓	RMSE ↓	MAPE ↓
PEMS03	FCDC	**14.48** ± 0.014	<u>24.80</u> ± 0.218	**14.47** ± 0.071
	Uniform lag prior	<u>14.60</u> ± 0.010	**24.66** ± 0.026	<u>14.60</u> ± 0.014
	Uniform lag gate	14.61 ± 0.003	24.83 ± 0.012	14.61 ± 0.004
PEMS04	FCDC	**17.88** ± 0.036	**29.63** ± 0.115	**11.98** ± 0.032
	Uniform lag prior	18.09 ± 0.007	30.04 ± 0.016	<u>12.07</u> ± 0.002
	Uniform lag gate	<u>18.04</u> ± 0.007	<u>29.90</u> ± 0.015	12.08 ± 0.002
PEMS07	FCDC	**18.76** ± 0.036	**32.33** ± 0.068	**7.85** ± 0.094
	Uniform lag prior	<u>18.91</u> ± 0.010	<u>32.65</u> ± 0.016	<u>8.01</u> ± 0.005
	Uniform lag gate	19.00 ± 0.006	32.76 ± 0.014	8.14 ± 0.003
PEMS08	FCDC	**13.08** ± 0.045	**23.01** ± 0.114	**8.80** ± 0.181
	Uniform lag prior	<u>13.23</u> ± 0.009	<u>23.23</u> ± 0.015	<u>9.03</u> ± 0.006
	Uniform lag gate	13.27 ± 0.004	23.36 ± 0.006	9.06 ± 0.001

Mean MAE and MAPE are lower with FCDC than with both fixed-uniform variants on all four datasets. The same holds for mean RMSE except on PEMS03, where the values are 24.6635 for uniform priors and 24.7995 for FCDC. The mean relative MAE reductions are 0.99% against uniform priors and 1.11% against uniform gates. Lower MAE is obtained with FCDC in all 96 dataset–variant–horizon comparisons of seed means and in 480 out of 480 comparisons at matched seeds.

Five-seed summaries and summaries over the four added seeds are compared in Table 9. Mean relative MAE reductions remain positive against all seven variants: 0.66–1.41% for four runs and 0.63–1.38% for five.

G.2 External baseline protocol

The eleven external baselines in Table 1 are compared using 44 model–dataset evaluations recorded in BasicTS (Shao et al., 2024) across PEMS03/04/07/08. The same dataset definitions, sensor counts, five-minute sampling, $60/20/20$ split ratio, and 12-step forecasting setting are used. Model-specific losses and optimization settings are retained, including the original STAEformer loss.

PEMS03 contains 26,208 time steps. RMSE is computed as R_{pool} in Equation (24).

The five-seed FCDC mean and standard deviation are reported alongside the external-baseline records in Table 1. The external methods were not repeated across these five seeds. Values are ranked at the displayed precision of two decimals, with ties given the same emphasis and the next distinct value ranked second. Relative reductions are computed at the available precision before rounding, then averaged with equal weight for each of the four datasets.

For STDDE and STWave, the dataset splits, training-only Z-score scaling, raw-zero target mask, 12-step inputs and outputs, and BasicTS metrics are as described above. Both are trained for 200 epochs with validation after each epoch; the final test checkpoint is selected by the lowest validation MAE. Model-specific losses and optimization schedules are retained.

STDDE configuration. Spline controls, two controlled delay differential equations, and historical reconstruction supervision are retained in the BasicTS adaptation based on the STDDE paper (Long et al., 2024). A three-hop graph is constructed, and delays are initialized from correlations in the training data. Learnable peak/off-peak edge delays are applied to interpolated historical states and constrained by a sigmoid to $[0, 3]$ sampling intervals; peak hours are defined as 07:00–09:00 and 17:00–19:00. The implementation therefore differs from the official method. Hidden width is set to 64, and one sampling interval is used as the Euler step. AdamW is configured with a learning rate of 10^{-3} and weight decay of 0.01; the rate is multiplied by 0.70 every 30 epochs. The batch size is 16 for training, validation, and test. Normalized forecast and historical-reconstruction SmoothL1 losses are summed with equal weights.

STWave configuration. The released wavelet/graph-attention core and auxiliary low-frequency supervision are retained in the BasicTS adaptation (Fang et al., 2023). Eight heads of width 16 and two layers are used. The wavelet decomposition has one level, with Symlet 2 for PEMS03, Daubechies 1 for PEMS04/07, and Coiflet 1 for PEMS08. The batch size is 64 in all phases except on PEMS07, where it is 32. Adam is initialized with a learning rate of 10^{-3}. The rate is adjusted by a validation-MAE plateau scheduler with patience 20, relative threshold 10^{-3}, reduction factor 0.10, and minimum rate 2×10^{-6}. Gradient norm is clipped at 5. Raw-unit forecast and low-frequency masked MAEs are summed with equal weights. Graph preprocessing is fitted to training data, and wavelet normalization and calendar rollover are applied consistently across splits.

H MAE AT EVERY FORECAST HORIZON

Five-seed mean MAE at all 12 horizons is reported for FCDC and the five primary variants in Tables 11–14. Delay correction is removed in "No delay"; explicit forecast input is removed in "No forecast"; hidden-state input is substituted in "State"; arrivals are restricted to history in "Hist. only"; and a parameter-matched local readout is used in "MLP". The displayed means are ranked by bold and underlined values. At saved precision, mean MAE is lower with FCDC in all 240 dataset–variant–horizon comparisons. Lower MAE is obtained in 1,190 out of 1,200 comparisons at matched seeds. For the two fixed-uniform variants, the corresponding counts are 96 of 96 for means and 480 of 480 within seeds. All horizon metrics for the fixed-uniform comparisons in Appendix G.1 are retained in the supporting snapshot.

Table 11: PEMS03 test MAE at all 12 horizons (5–60 minutes), averaged over five seeds. Lower is better. Best and second-best distinct two-decimal means are bold and underlined, respectively; ties are ranked equally. Standard deviations for overall metrics are reported in Table 8. Column abbreviations are defined in Appendix H.

Horizon	FCDC	No delay	No forecast	State	Hist. only	MLP
1	**11.99**	12.17	12.06	12.10	12.04	12.05
2	**12.71**	12.88	12.79	12.82	12.76	12.76
3	**13.31**	13.47	13.39	13.42	13.36	13.37
4	**13.77**	13.94	13.85	13.88	13.82	13.84
5	**14.15**	14.34	14.24	14.27	14.20	14.23
6	**14.50**	14.67	14.58	14.62	14.55	14.57
7	**14.83**	15.00	14.92	14.95	14.88	14.90
8	**15.15**	15.33	15.24	15.28	15.20	15.21
9	**15.43**	15.63	15.54	15.57	15.50	15.51
10	**15.69**	15.90	15.80	15.86	15.77	15.78
11	**15.96**	16.18	16.08	16.13	16.05	16.06
12	**16.27**	16.49	16.40	16.44	16.35	16.36

Table 12: PEMS04 test MAE at all 12 horizons (5–60 minutes), averaged over five seeds. Lower is better. Best and second-best distinct two-decimal means are bold and underlined, respectively; ties are ranked equally. Standard deviations for overall metrics are reported in Table 8. Column abbreviations are defined in Appendix H.

Horizon	FCDC	No delay	No forecast	State	Hist. only	MLP
1	**16.25**	16.55	16.38	16.26	16.43	16.38
2	**16.79**	17.04	16.90	16.84	16.91	16.88
3	**17.19**	17.45	17.29	17.23	17.31	17.29
4	**17.48**	17.74	17.58	17.52	17.61	17.60
5	**17.72**	17.99	17.81	17.77	17.85	17.86
6	**17.92**	18.20	18.01	17.98	18.06	18.07
7	**18.11**	18.39	18.20	18.20	18.27	18.28
8	**18.29**	18.55	18.38	18.35	18.41	18.46
9	**18.44**	18.69	18.53	18.51	18.57	18.62
10	**18.60**	18.86	18.67	18.67	18.72	18.77
11	**18.76**	19.01	18.83	18.82	18.89	18.94
12	**18.96**	19.21	19.04	19.00	19.08	19.12

Table 13: PEMS07 test MAE at all 12 horizons (5–60 minutes), averaged over five seeds. Lower is better. Best and second-best distinct two-decimal means are bold and underlined, respectively; ties are ranked equally. Standard deviations for overall metrics are reported in Table 8. Column abbreviations are defined in Appendix H.

Horizon	FCDC	No delay	No forecast	State	Hist. only	MLP
1	**16.08**	16.26	16.21	16.22	16.27	16.23
2	**16.97**	17.15	17.08	17.13	17.15	17.11
3	**17.61**	17.80	17.72	17.78	17.80	17.76
4	**18.10**	18.29	18.21	18.28	18.31	18.26
5	**18.51**	18.70	18.61	18.69	18.74	18.66
6	**18.86**	19.07	18.97	19.06	19.09	19.03
7	**19.18**	19.39	19.28	19.38	19.41	19.36
8	**19.47**	19.68	19.55	19.66	19.71	19.66
9	**19.73**	19.94	19.82	19.93	19.98	19.92
10	**19.97**	20.18	20.05	20.16	20.21	20.18
11	**20.22**	20.43	20.30	20.41	20.46	20.46
12	**20.47**	20.66	20.54	20.65	20.71	20.72

Table 14: PEMS08 test MAE at all 12 horizons (5–60 minutes), averaged over five seeds. Lower is better. Best and second-best distinct two-decimal means are bold and underlined, respectively; ties are ranked equally. Standard deviations for overall metrics are reported in Table 8. Column abbreviations are defined in Appendix H.

Horizon	FCDC	No delay	No forecast	State	Hist. only	MLP
1	**11.40**	11.59	11.50	11.46	11.55	11.46
2	**11.87**	12.05	11.99	11.96	12.00	11.93
3	**12.27**	12.47	12.39	12.38	12.41	12.35
4	**12.60**	12.81	12.71	12.72	12.73	12.68
5	**12.87**	13.10	12.98	13.01	13.02	12.97
6	**13.12**	13.38	13.23	13.27	13.28	13.23
7	**13.35**	13.61	13.47	13.51	13.52	13.47
8	**13.56**	13.82	13.67	13.72	13.73	13.66
9	**13.74**	14.00	13.85	13.90	13.93	13.85
10	**13.90**	14.17	14.01	14.07	14.09	14.01
11	**14.06**	14.31	14.17	14.24	14.25	14.18
12	**14.26**	14.50	14.37	14.45	14.45	14.38

I FROZEN-CHECKPOINT CORRECTION ANALYSIS

I.1 POPULATION AND COMPONENT EXTRACTION

One FCDC checkpoint per dataset is analyzed, selected by validation MAE at epochs 91, 61, 70, and 29 for PEMS03, PEMS04, PEMS07, and PEMS08, respectively. Weights, graph buffers, and historical inputs are held fixed. All combinations of test window, horizon, and sensor with a finite, nonzero target are included, using the null tolerance in Appendix G. The valid forecast counts are 22,251,604; 12,272,795; 59,236,542; and 7,213,341 on the four datasets, respectively.

During the forward pass, the base forecast from Equation (3) and the gated correction $\Delta_j = \rho_j \eta_j \odot \delta_j$ from Equation (13) are recorded; the normalized model output is recovered exactly by their sum. With training mean μ and standard deviation σ, the base forecast is converted to raw units as $\sigma\hat{y}^{\text{base}} + \mu$ and the correction as $\sigma\Delta$. Any float32 rounding difference is measured by the maximum discrepancy between their sum and the separately inverse-scaled final output. In every batch, the correction is reproduced exactly by the zero-shift readout, without fitting any parameters.

For a valid forecast point, let $e^{\text{base}} = |\hat{y}^{\text{base}} - y|$ and $e^{\text{final}} = |\hat{y} - y|$ in raw flow units. A point is improved if $e^{\text{final}} < e^{\text{base}} - \epsilon$, worsened if $e^{\text{final}} > e^{\text{base}} + \epsilon$, and tied otherwise, with $\epsilon = 10^{-6}$. Overall MAE and these outcome frequencies are reported in Table 15, with a finer decomposition in Appendix I.2.

Table 15: Within-checkpoint correction behavior. Base and final forecasts are taken from the same jointly trained FCDC. Their difference is not an estimate of the gain over a separately trained no-correction model. Improved/worse are percentages of valid forecast points, with ties at the remaining points. Sign crossing is measured between condition shifts of -1.00 and $+1.00$ training standard deviations.

Dataset	Base MAE	Final MAE	Improved (%)	Worse (%)	Sign crossing (%)
PEMS03	20.21	14.50	65.43	34.56	17.61
PEMS04	29.02	17.88	71.52	28.45	81.83
PEMS07	25.80	18.78	70.03	29.97	52.23
PEMS08	31.72	13.12	78.97	21.03	82.60

The internal base and correction in Table 15 are jointly trained; the separately trained No delay correction variant in Table 2 is not recovered by removing correction at inference.

I.2 CORRECTION OUTCOMES ON UNMODIFIED TEST INPUTS

Pointwise outcomes are summarized in Figure 3, with exact counts in Table 16. For each valid forecast, the residual, applied correction, and change in absolute error are defined in raw flow units as $r = y - \hat{y}^{\text{base}}$, $c = \hat{y} - \hat{y}^{\text{base}}$, and $d = e^{\text{final}} - e^{\text{base}}$. Absolute error is classified as reduced when $d < -\epsilon$ and unchanged when $|d| \leq \epsilon$. Error increases, where $d > \epsilon$, are classified as overshoot if c and r have the same nonzero sign, wrong direction if their signs are opposite, and an initially exact base if $r = 0$. A correction that crosses the target is counted as an improvement if absolute error is reduced. Absolute error can also be left unchanged by a nonzero correction.

The recorded float32 predictions are converted to float64 before differences are computed. The applied correction is calculated from the final and base forecasts to exclude the small inverse-scaling discrepancy in the separately recorded branch output. Each valid forecast is assigned to exactly one category. After the three types of error increase are combined, the overall and per-horizon counts are reproduced from the three-way classification above. The overshoot counts are also reproduced when the recorded branch correction is used.

Table 16: Exact counts for the five correction outcomes in Figure 3. Each valid forecast point is assigned to one category. Overshoot, wrong direction, and an initially exact base are counted only when absolute error is increased.

Dataset	Valid forecasts	Improved	Overshoot	Wrong direction	Initially exact base	Unchanged
PEMS03	22,251,604	14,559,354	2,216,081	5,474,153	11	2,005
PEMS04	12,272,795	8,777,010	1,141,631	2,350,043	1	4,110
PEMS07	59,236,542	41,483,673	4,071,971	13,679,968	23	907
PEMS08	7,213,341	5,696,671	597,163	919,503	1	3

Absolute error is reduced by correction on 65.43–78.97% of valid forecasts. Among all valid forecasts, 6.87–9.96% are worsened by overshoot and 12.75–24.60% by correction in the wrong direction. Of the worsened forecasts, 22.94–39.37% are classified as overshoots: error is increased despite a correction directed toward the target. Error reduction is therefore not guaranteed by the positive sigmoid correction gate η_j. Corrections away from an initially exact base are recorded at 11, 1, 23, and 1 points on PEMS03, PEMS04, PEMS07, and PEMS08, respectively.

The same decomposition is reported at each forecast horizon in Figure 4. On PEMS03, the improved share falls from 72.85% at 5 minutes to 61.43% at 60 minutes. It remains within 69.61–70.24% on PEMS07. These frequencies are measured under unmodified inputs; sensitivity to the conditioning forecast is examined separately through the interventions below.

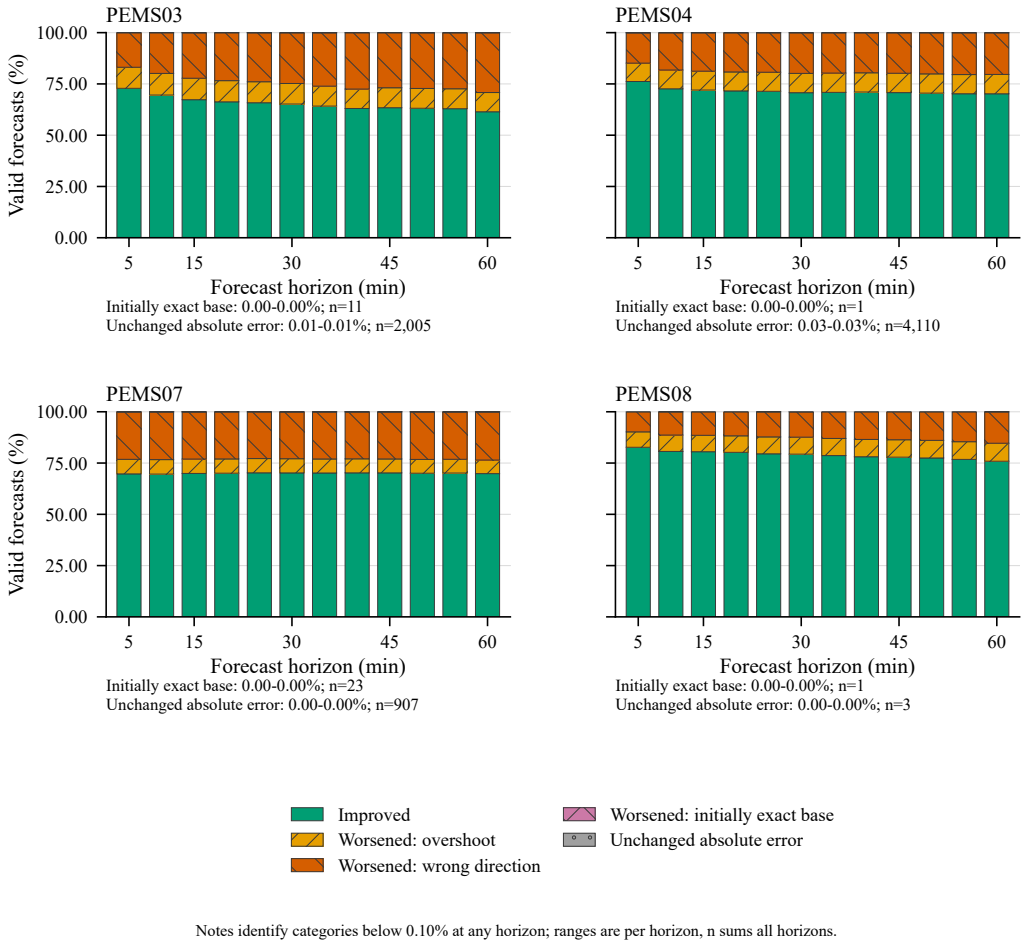


Figure 4: Correction outcomes over the 12 forecast horizons, using the same five categories and error tolerance as Figure 3. Each bar is normalized by the valid forecast count at that horizon. For categories with a share below 0.10% at any horizon, the per-horizon range and total count over all horizons are shown below each panel. A displayed value of 0.00% can represent a nonzero count.

I.3 INTERVENTIONS ON READOUT INPUTS

The gated correction with conditioning forecast b, fixed local/message features ξ_j, and fixed node state v_j is denoted by $\Delta_j(b; \xi_j, v_j)$. In normalized coordinates the uniform-shift diagnostic is

$$\hat{y}_j^{(\alpha)} = \hat{y}_j^{\text{base}} + \Delta_j(\hat{y}_j^{\text{base}} + \alpha \mathbf{1}_H; \xi_j, v_j), \tag{25}$$

where $\alpha \in \{-1.00, -0.50, -0.25, 0.00, 0.25, 0.50, 1.00\}$. One unit of α corresponds to one training standard deviation in raw flow units. The additive $\hat{y}_j^{\text{base}}$ is held fixed. Sign reversals relative to $\alpha = 0.00$ are counted in Figure 2; the two endpoint conditions, $\alpha = -1.00$ and $+1.00$, are compared in Table 15. In both comparisons, opposite signs and correction magnitudes above ϵ in raw units are required. The endpoint crossing rate is between 17.61% on PEMS03 and 82.60% on PEMS08. On all four datasets, MAE is lowest at zero shift among the seven tested offsets.

Three further controls are applied to the frozen readout. In *Permuted forecast input*, another sensor's complete 12-step forecast is substituted only in the conditioning channels. In *Future arrivals removed*, the $F = 3$ future message values and their validity masks are zeroed, while historical arrivals are retained. In *Permuted delayed messages*, all $T + F = 15$ arrival values and their masks are reassigned together across sensors. A fixed derangement, formed by rotating a fixed random ordering by one position, is used for both permutation controls. Other inputs remain at their original sensors. Targets are used only for evaluation and are unavailable through readout inputs.

Table 17: MAE under frozen-readout interventions. The additive base forecast is held fixed, and readout inputs are perturbed without retraining. Best and second-best distinct displayed values are marked in bold and underlined.

Readout input	PEMS03	PEMS04	PEMS07	PEMS08
FCDC (unchanged)	**14.50**	**17.88**	**18.78**	**13.12**
Permuted forecast input	15.07	29.34	23.35	32.55
Future arrivals removed	14.94	18.65	19.26	14.87
Permuted delayed messages	<u>14.55</u>	<u>17.98</u>	<u>18.80</u>	<u>13.19</u>

The largest error increase on every dataset is observed under forecast permutation (Table 17). Under message permutation, MAE is increased by 0.02–0.09; the main correction gains cannot be attributed to sensor-specific delayed content on the basis of this small response. When future arrivals are removed, MAE is increased by 0.44–1.75. Message values and their validity masks are removed together, so content and missingness effects are combined in this control.

The fitted readout is tested at fixed weights; input combinations outside the training distribution may be introduced by these perturbations.

I.4 HISTORICAL TRENDS AND CLOCK-TIME GROUPS

Each window–sensor pair is assigned a history class before its forecast error is examined. Let d_j be the raw mean of the last three historical observations minus the mean of the first three. The threshold c is set to the 75th percentile of $|d_j|$ over training-history windows with all six endpoint observations valid: 33.67, 40.67, 50.67, and 33.67 on the four datasets. A test history is classified as rising for $d_j > c$, falling for $d_j < -c$, and steady otherwise. A history with an invalid endpoint observation is classified as unknown. The labels are assigned from observed history alone, excluding future changes; actual congestion events are not identified.

A separate clock-time partition is defined by the first forecast slot. The intervals 07:00–09:00 and 17:00–19:00 on all days are used as a rush-hour proxy, with all other slots in its complement. Realized congestion is not estimated by this fixed schedule. Valid forecast counts and within-checkpoint errors for both partitions are reported in Table 18; the three known history classes are shown in Figure 5.

Table 18: Correction behavior by historical trend and origin clock time. Counts are given for valid window–horizon–sensor forecasts. The test set is partitioned separately by trend and clock time. A trend is marked unknown when a historical endpoint observation is missing.

Dataset	Group	Valid forecasts	Base MAE	Final MAE	Improved (%)
PEMS03	Rising	2,507,116	31.39	22.94	63.02
	Steady	16,793,669	17.31	12.58	65.15
	Falling	2,588,990	28.67	20.22	65.85
	Unknown	361,829	16.58	4.03	92.28
	Rush-hour proxy	3,705,842	26.52	19.36	63.44
	Other hours	18,545,762	18.95	13.52	65.83
PEMS04	Rising	1,549,304	42.25	27.28	70.31
	Steady	9,068,578	25.98	15.28	72.47
	Falling	1,546,688	34.32	23.85	67.96
	Unknown	108,225	18.80	15.98	59.68
	Rush-hour proxy	2,117,016	36.04	22.47	71.42
	Other hours	10,155,779	27.56	16.93	71.54
PEMS07	Rising	6,988,697	32.37	25.92	66.48
	Steady	45,038,295	24.43	16.80	71.73
	Falling	7,093,872	27.21	24.47	62.35
	Unknown	115,678	75.51	7.23	95.70
	Rush-hour proxy	9,885,862	28.64	22.30	69.22
	Other hours	49,350,680	25.23	18.07	70.19
PEMS08	Rising	877,914	38.55	18.95	76.78
	Steady	5,340,632	30.09	11.69	79.95
	Falling	955,293	35.60	16.04	76.37
	Unknown	39,502	7.25	5.82	59.13
	Rush-hour proxy	1,223,536	40.08	15.67	80.32
	Other hours	5,989,805	30.01	12.60	78.70

Aggregate MAE is reduced with correction in every reported history and clock-time group, although the largest benefits are not consistently observed at rush hours or during rapid historical changes. On PEMS07, MAE is reduced by 6.35 at rush hours and 7.16 at other times, and by 2.74 for falling histories and 7.63 for steady histories. On PEMS03, reductions of 8.46 and 8.45 are recorded for rising and falling histories, compared with 4.74 for steady histories.

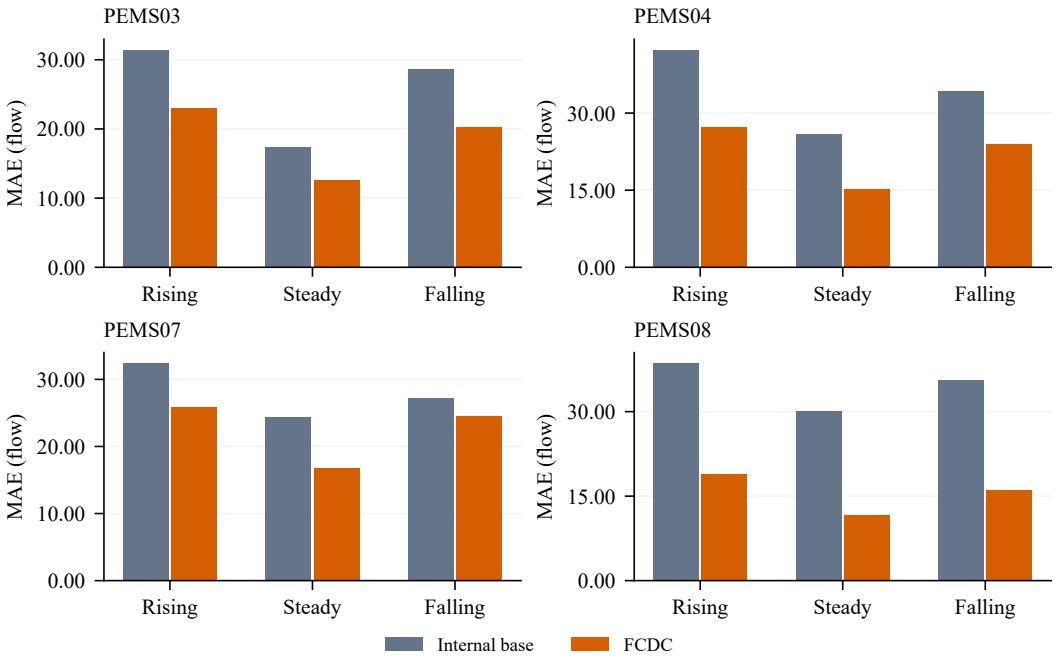


Figure 5: Internal base and corrected MAE grouped by historical trend, with thresholds computed from training history only. Unknown histories and the independent clock-time partition are reported in Table 18.

I.5 LEARNED LAG WEIGHTS

A prior $\pi_{ij\tau}$ over the three candidate lags is learned for each retained edge and held fixed across samples, arrival positions, and forecast horizons at evaluation. Effective routing weights are modulated by historical context and source availability through the dynamic lag gate $g_{rj\tau}$ and the mask $m_{r-\tau,i}$. Correction magnitude is separately modulated by the sigmoid gate η_j in the readout. The priors are summarized by their mean normalized entropy deficit and the distribution of $q_{ij} = \max_\tau \pi_{ij\tau}$ across edges (Table 19). Entropy is computed for each edge before averaging; concentration is measured by q_{ij} regardless of which lag is preferred.

For the dynamic lag gates, only historical arrival positions with positive mass for all $K = 3$ lags are included. Boundary and future positions are excluded because a single lag can be forced by availability. Gate standard deviations are computed across the included positions and nodes.

Table 19: Lag diagnostics. Mean entropy deficit is calculated as $100(1 - \mathcal{H}(\pi_{ij})/\log K)$ averaged over edges. Variation across edges is summarized by the median, 95th percentile and maximum of $q_{ij} = \max_\tau \pi_{ij\tau}$; $q_{ij} = 1/3$ for a uniform prior. Dynamic lag-gate statistics are computed at historical arrival positions with all three lags available; SD is taken across positions and nodes.

Dataset	Mean entropy	Largest edge lag probability (%)			Lag-1 gate (%)	
	deficit (%)	Median	95th pct.	Max.	Mean	SD
PEMS03	0.11	34.32	37.39	47.08	71.35	18.68
PEMS04	0.84	36.80	44.49	64.51	84.04	0.51
PEMS07	0.12	34.29	37.82	44.93	93.97	3.22
PEMS08	1.25	38.08	45.98	59.24	91.41	0.77

Mean edge entropy deficits are 0.11–1.25%. Across datasets, the median q_{ij} ranges from 34.29% to 38.08%, compared with 33.33% for a uniform prior. Weak lag preferences are observed on most edges, with more pronounced preferences in a smaller subset (Figure 6). On PEMS04 and PEMS08, the 95th percentiles are 44.49% and 45.98%, and the maxima are 64.51% and 59.24%.

On average, 71.35–93.97% of the dynamic lag-gate mass is assigned to the first lag (five minutes). First-lag standard deviations are 18.68, 0.51, 3.22, and 0.77 percentage points on PEMS03, PEMS04, PEMS07, and PEMS08, with small variation on PEMS04 and PEMS08 in the included positions.

In two retrained controls, the edge priors or the lag gate are replaced with fixed uniform weights (Table 10). Lower five-seed mean MAE is obtained with FCDC on every dataset, with mean relative reductions of 0.99% and 1.11%, respectively. The changes to lag weighting and full results are given in Appendix G.1.

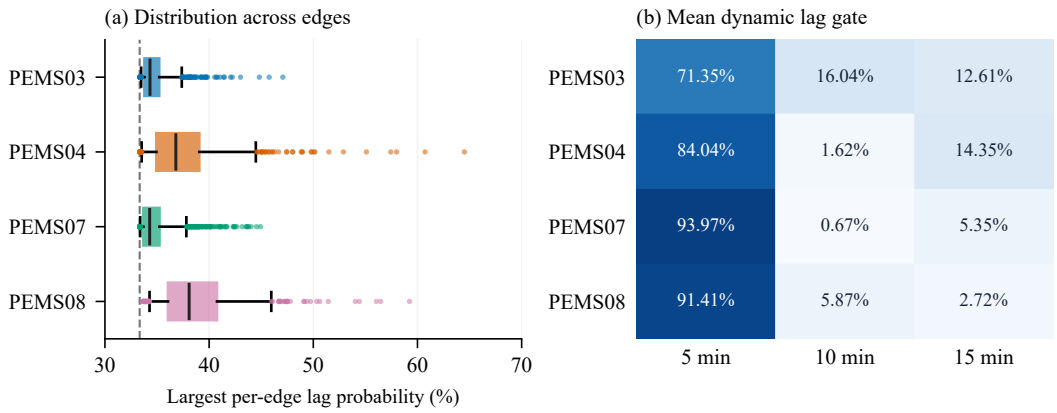


Figure 6: Learned lag weights for one fixed model per dataset. Left: the largest lag probability for each edge. The interquartile range is shown by each box, with a center line at the median; whiskers are drawn to the outermost observations within the 5th–95th percentiles, and the remaining edges are plotted as points. The uniform value $1/3$ is marked by the dashed line. Right: mean dynamic lag-gate probabilities on a 0.00–100.00% color scale, using historical positions with all three lags available.

I.6 SUCCESSFUL AND FAILED CORRECTIONS

Four cases per dataset are presented in Figures 7–10, chosen by a rule fixed before the analysis run. Window–sensor pairs are eligible when all 12 historical observations and all 12 future targets are valid. Within each of the rising, falling, and steady history classes, the pair nearest the median positive 12-horizon MAE gain is selected. The fourth case is nearest the median negative gain across all eligible histories. Ties are broken by the earliest flattened window–sensor index. Case selection is conditioned on outcome; population frequencies are reported in Table 15 and Figure 3. Sensor and window indices in the figures are zero-based.

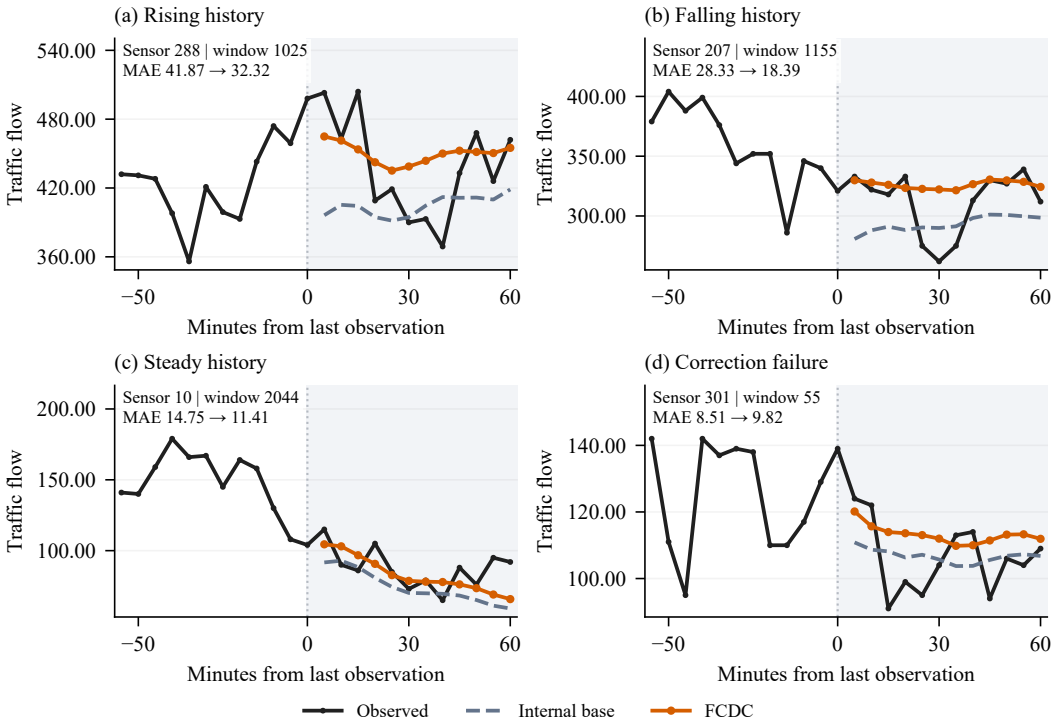


Figure 7: PEMS03 correction cases selected near the median positive gain in each historical trend class and the median negative gain across classes. The forecast interval is shaded; internal-base and final 12-step MAE are reported in the annotation.

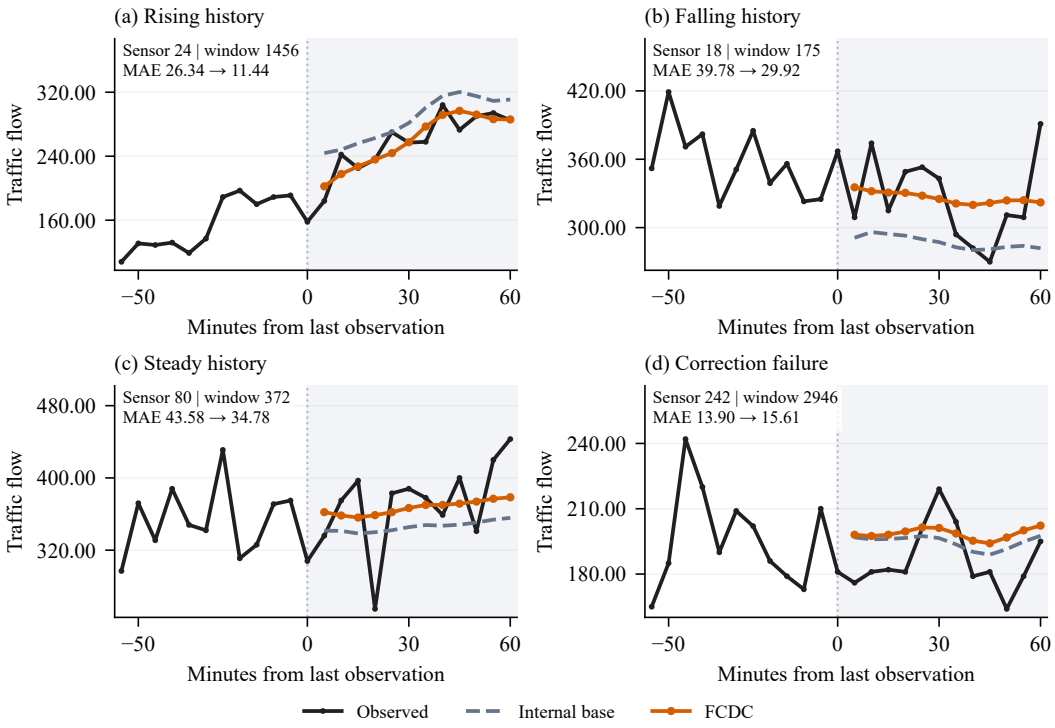


Figure 8: PEMS04 correction cases, using the selection rule and annotations of Figure 7.

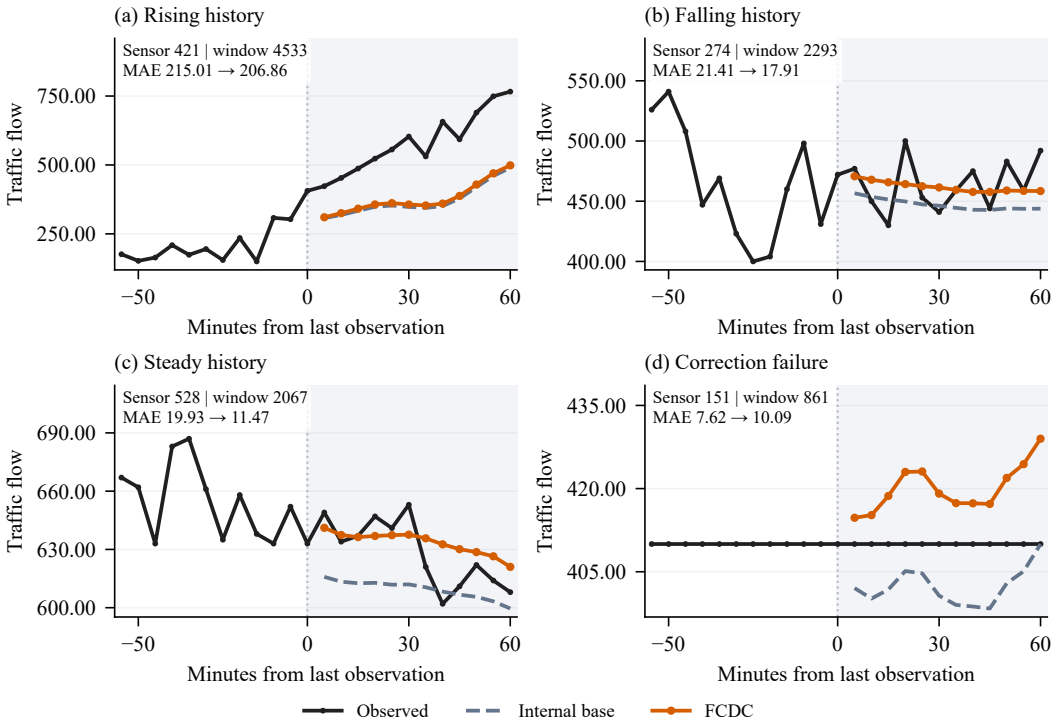


Figure 9: PEMS07 cases, including one with increased MAE after correction.

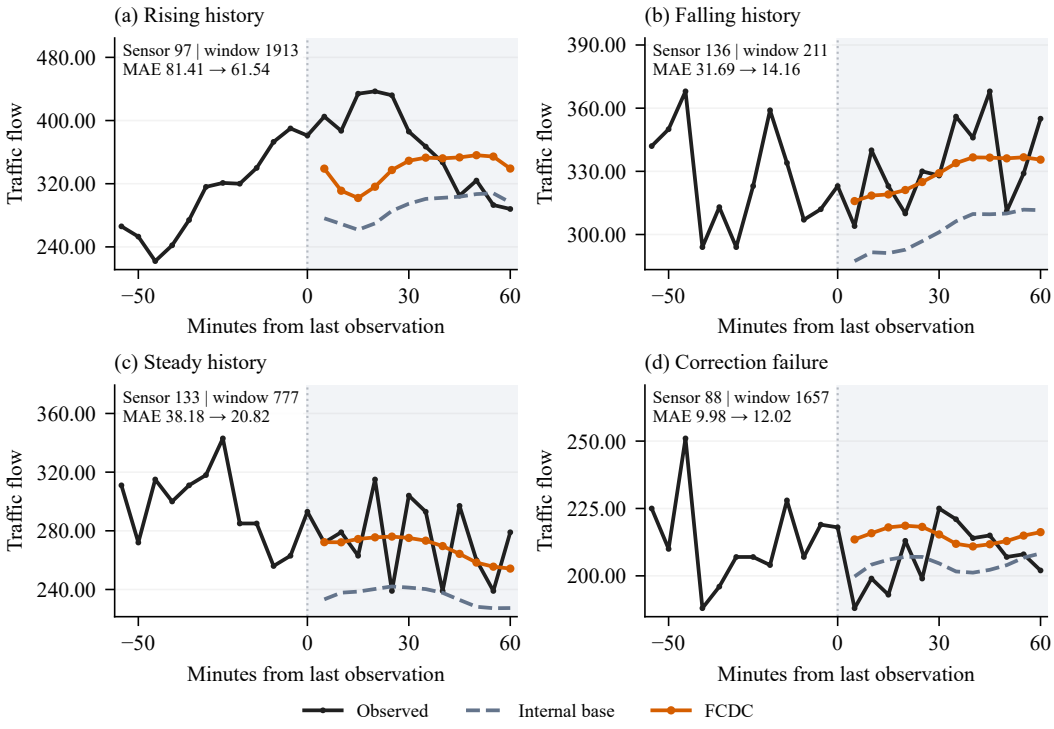


Figure 10: PEMS08 cases selected with the same history thresholds and median-gain rule.

J COMPUTATIONAL EFFICIENCY

J.1 MEASUREMENT PROTOCOL AND EXTERNAL COMPARISON

Each measurement was performed on a single NVIDIA GeForce RTX 3090 GPU with 24 GB memory. Jobs were distributed across two such devices, with one measurement worker per device. The presence of other GPU processes was checked throughout timing. The software versions were PyTorch 2.5.1+cu124, CUDA 12.4, and cuDNN 9.1.0. All models were executed in FP32 with one PyTorch CPU thread. Automatic mixed precision, TF32, compilation, and cuDNN benchmarking were disabled. For every model, batch size was set to 16 on PEMS03/04/08 and 1 on PEMS07, with 12 historical and 12 future steps. Within each dataset, the same four batches, spread across the relevant split, were supplied to every model. Only one raw batch was held on the GPU at a time.

Inference was timed in evaluation mode, including input normalization, mask and calendar preparation, model execution, and output rescaling. Training measurements included computation of the configured original loss, a backward pass, gradient clipping where configured, and an optimizer update. A fresh optimizer was warmed before measurement, and curriculum context was set to the full forecast horizon. For each mode, 20 warm-up iterations were followed by five blocks of 50 timed iterations. CUDA was synchronized before and after every measured call for wall-clock timing. Data loading, host-to-device transfer, metric calculation, logging, and one-time graph/cache preparation were excluded. Latency is reported as the median of all 250 calls, and peak memory as PyTorch's maximum allocated memory across those calls. Means, quartiles, 95th percentiles, reserved memory, and individual timings were also recorded. Throughput is $1000B/\operatorname{mean}(t_{\mathrm{ms}})$. Training time to convergence was not measured.

Table 20: External runtime comparison in FP32. Each measurement was conducted on a single RTX 3090 GPU. Latency is the median per batch; memory is peak allocated memory. Lower is better (↓). Best and second-best distinct displayed values are marked in bold and underlined within each dataset. *Only execution costs are reported for PDFormer, using synthetic preprocessing inputs and initialized weights.

Model	Infer. (ms)	Train (ms)	Infer. (MiB)	Train (MiB)
PEMS04 ($B = 16$)				
FCDC	**7.18**	**54.41**	<u>108.73</u>	**489.23**
STAEformer	56.73	255.27	722.96	3778.80
STDDE	<u>21.05</u>	<u>76.29</u>	**98.67**	<u>696.45</u>
STWave	39.86	145.96	674.73	2665.61
PDFormer*	61.73	235.47	754.24	4637.27
PEMS07 ($B = 1$)				
FCDC	**7.71**	<u>54.74</u>	<u>57.02</u>	<u>230.19</u>
STAEformer	11.18	56.67	329.26	1273.50
STDDE	19.12	73.63	**23.36**	**135.53**
STWave	<u>8.76</u>	**49.87**	140.63	510.75
PDFormer*	22.36	81.15	331.95	1925.79

Saved checkpoints were loaded for FCDC, its two core controls, STAEformer, STDDE, and STWave. For PDFormer, initialized weights and synthetic graph, Laplacian, and pattern inputs of the configured shapes were used. The full dense model was executed, with one-time preprocessing bypassed. This workload is marked separately in the tables and is used only to measure execution latency and GPU memory; accuracy and convergence cannot be assessed from it. The PDFormer accuracy in Table 1 was obtained from an independently trained model.

On PEMS04, the lowest median inference and training-step latencies in Table 20 are obtained with FCDC, with a training peak memory of 489.23 MiB. The inference speedups over STAEformer and STWave are 7.90× and 5.55×, respectively. On PEMS07, the lowest median inference latency is again obtained with FCDC. Its median training-step latency is close to that of STAEformer and higher than that of STWave; lower memory usage is obtained with STDDE.

J.2 COST OF THE CORRECTION BRANCH

FCDC, no delay correction, and the parameter-matched MLP are compared under the same batches and measurement settings in Table 21. The backbone and direct own-history path are retained in both controls. In Table 22, differences are measured relative to the control with the adapter and complete correction readout removed.

Table 21: Core-model execution costs under the same protocol. Latency is the median per batch; memory is peak allocated memory. Lower is better (↓). Best and second-best distinct displayed values are marked in bold and underlined.

Model	Infer. (ms)	Train (ms)	Infer. (MiB)	Train (MiB)
PEMS03 ($B = 16$)				
FCDC	9.35	54.82	117.49	541.74
No delay correction	**7.20**	<u>49.58</u>	**117.43**	**533.34**
Parameter-matched MLP	<u>7.32</u>	**41.08**	<u>117.46</u>	<u>533.47</u>
PEMS04 ($B = 16$)				
FCDC	7.18	54.41	108.73	489.23
No delay correction	**6.40**	**40.43**	**108.69**	**484.58**
Parameter-matched MLP	<u>6.47</u>	<u>50.23</u>	<u>108.71</u>	<u>484.70</u>
PEMS07 ($B = 1$)				
FCDC	7.71	54.74	57.02	230.19
No delay correction	**5.33**	**49.66**	**56.95**	**229.98**
Parameter-matched MLP	<u>5.79</u>	<u>50.42</u>	<u>56.99</u>	<u>230.14</u>
PEMS08 ($B = 16$)				
FCDC	7.39	**47.36**	75.09	328.64
No delay correction	<u>7.22</u>	<u>48.68</u>	**75.05**	**324.10**
Parameter-matched MLP	**5.87**	50.11	<u>75.08</u>	<u>324.20</u>

Table 22: FCDC overhead relative to no delay correction. Timing percentages are based on batch medians and memory percentages on peak allocation.

Dataset	Infer. (%)	Train (%)	Train mem. (%)
PEMS03	29.85	10.56	1.57
PEMS04	12.13	34.57	0.96
PEMS07	44.56	10.24	0.09
PEMS08	2.35	-2.71	1.40

With correction, training peak allocation is higher by 0.20–8.39 MiB (0.09–1.57%), and inference takes an additional 0.17–2.38 ms per batch (2.35–44.56%). FCDC inference is 0.70–2.03 ms slower than the parameter-matched MLP, in which message access and computation are both changed.

The timing percentages in Table 22 are calculated from recorded medians. The negative PEMS08 training difference is insufficient to establish an architectural speedup from correction. The larger PEMS07 inference percentage is measured relative to a 5.33 ms no-correction batch at $B = 1$. Because different batch sizes are used across datasets, the percentages cannot be interpreted as a scaling experiment.

J.3 ADDITIONAL MEASUREMENTS AND TIMING VARIATION

Table 23: Additional measurements on PEMS03 and PEMS08. Units, ranking, and measurement protocol are as in Table 20; PDFormer with synthetic preprocessing inputs and initialized weights is marked by *.

Model	Infer. (ms)	Train (ms)	Infer. (MiB)	Train (MiB)
PEMS03 ($B = 16$)				
FCDC	**9.35**	**54.82**	**117.49**	**541.74**
PDFormer*	70.60	284.90	985.88	6026.44
PEMS08 ($B = 16$)				
FCDC	**7.39**	**47.36**	**75.09**	**328.64**
PDFormer*	25.75	95.11	285.87	1796.89

Additional PDFormer measurements on PEMS03 and PEMS08 are reported in Table 23. The same synthetic preprocessing as in Table 20 is used, with matched batch sizes and input samples. Missing timing and accuracy values are not estimated from other datasets.

Timing varied across blocks despite per-device isolation. Inference block medians of 7.24–9.51 ms were recorded for FCDC on PEMS07 and 5.56–9.31 ms for no correction on PEMS08. On PEMS07, training-step medians are 54.74 and 56.67 ms for FCDC and STAEformer; the means are 54.92 and 54.65 ms, giving the reverse ordering. The five blocks were run sequentially, with workers assigned to different GPUs of the same model. Clock, temperature, power, and host-load telemetry were not recorded. Small timing differences under this protocol are insufficient to establish a stable ordering between architectures.

Measurement coverage and verification. Measurements were obtained for 22 model–dataset combinations in 44 inference/training records, covering 11,000 timed calls. Seventeen combinations were completed in the main run, one in a successful STDDE/PEMS04 retry, and four in separate PDFormer runs. Three failed records were excluded. Before the STDDE retry was timed, no mismatched elements were found in the corrected prediction comparison. Deterministic kernels were enabled only for available preflight operations and disabled during measurement. All latency summaries were checked against the individual samples.