TIME VERSUS SPACE: A SYSTEMATIC EMPIRICAL STUDY OF INFORMATION ALLOCATION IN LARGE-SCALE TRAFFIC FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatio-temporal traffic forecasting has largely focused on increasingly sophisticated graph-based models, implicitly assuming that spatial information is the primary source of predictive power. We revisit this assumption by asking a fundamental question: given a fixed observation budget, is forecasting accuracy improved more by allocating observations to temporal depth or spatial breadth? We answer this through a systematic empirical study of this trade-off in large-scale traffic forecasting, showing that predictive performance is determined primarily by whether the input covers one complete daily cycle of the target sensor, while additional spatial information provides only marginal benefit. We further explain this finding by showing that, under typical benchmark resolutions, congestion propagation between neighbouring sensors is faster than the observation interval, leaving little exploitable lead–lag information for graph-based models. On a high-resolution freeway corridor where propagation is resolved, physically relevant neighbourhoods instead become indispensable as traffic approaches capacity. These insights naturally lead to a simple graph-free temporal-only forecasting recipe that combines long-context temporal modelling with lightweight adaptive gating for atypical conditions, achieving state-of-the-art average accuracy on all four LargeST subsets. The principle is established with a lightweight model trained from scratch and transfers to an off-the-shelf pretrained time-series foundation model with only lightweight adaptation. Our findings indicate that predictive information is far more effectively captured through temporal depth than additional spatial observations, providing a new perspective on how spatio-temporal models should allocate their capacity.

1 INTRODUCTION

Traffic forecasting is a fundamental task in intelligent transportation systems, supporting a wide range of downstream applications, including route planning, congestion mitigation, and network management. Over the past decade, traffic forecasting has been widely regarded as a spatio-temporal prediction problem, where temporal evolution is closely coupled with spatial correlations (Yu et al., 2018; Li et al., 2018; Zhao et al., 2020). Motivated by the propagation of traffic conditions through connected road networks, nearly all recent advances have focused on improving spatial modelling through increasingly sophisticated graph representations (Zheng et al., 2020; Wu et al., 2019; Guo et al., 2019; Bai et al., 2020). This emphasis rests on an underlying assumption that has shaped most recent architectures: spatial information constitutes the dominant source of predictive power.

Despite its intuitive appeal, surprisingly little work has asked whether spatial information actually contains the dominant predictive signal. We therefore revisit traffic forecasting through the lens of observation budget allocation. Any forecasting model operates under a finite observation budget, jointly constrained by sensing and computational resources. Given a fixed observation budget, resources can be allocated in two ways: observing more sensors (spatial breadth) or observing longer histories (temporal depth). Together, these two dimensions define a unified design space for allocating predictive information in traffic forecasting (Figure 1). This leads to the central question of this paper: given the same observation budget, should predictive capacity be allocated to temporal depth or spatial breadth? Answering this question reveals where predictive information actually resides,

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.

providing a more fundamental understanding of spatio-temporal forecasting than comparing architectures. It also has practical weight: joint spatial modelling requires computation and memory that grow with the number of sensors, so if predictive information is concentrated in temporal depth, this burden can be substantially reduced.

To answer this question, we conduct an empirical study of predictive information allocation under a fixed observation budget. Rather than proposing yet another graph-based forecasting architecture, we evaluate the predictive value of different combinations of temporal depth and spatial breadth using both closed-form linear and nonlinear predictors. We then characterize where predictive information is concentrated, further decompose the contribution of spatial information, and explain our findings through traffic-flow analysis, testing on a high-resolution freeway corridor when physical neighbourhoods do become necessary. These insights lead to a simple graph-free temporal-only forecasting recipe. We first establish it with a lightweight mixer trained from scratch, and then show that it transfers to an off-the-shelf time-series foundation model adapted with calendar embeddings, historical-average priors, and a context gate. Our contributions are as follows.

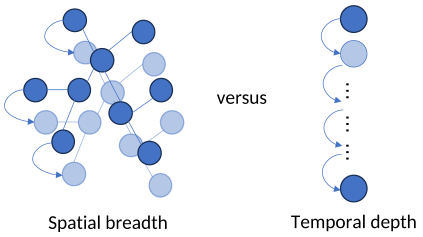


Figure 1: Two allocations of a fixed observation budget: spatial breadth (left) and temporal depth (right).

- We formulate predictive information allocation as a question in traffic forecasting and show that, in the benchmark settings studied, covering one complete daily cycle of the target's own history is more effective than expanding spatial input.
- We show that the remaining spatial gain is largely regional context, which random sensors recover, and that congestion propagation is typically resolved within one interval.
- We identify where spatial information shows measurable value: in sudden-onset events and, when the sampling resolves propagation, near capacity, where topology's predictive value becomes regime-dependent: exact graph connectivity is often unnecessary, but physically relevant local neighbourhoods become indispensable.
- We show that the principle supports strong temporal-only forecasting: a lightweight mixer and an adapted foundation model outperform graph-based baselines, and a context gate that mitigates historical-average mismatch gives state-of-the-art average accuracy on all four LargeST subsets.

2 Related Work

2.1 Evolution of Spatio-Temporal Modelling in Traffic Forecasting

After statistical models such as ARIMA (Williams & Hoel, 2003) and VAR (Chandra & Al-Deek, 2009) and recurrent networks (Ma et al., 2015), spatio-temporal graph neural networks (STGNNs), represented by STGCN (Yu et al., 2018), DCRNN (Li et al., 2018), and TGCN (Zhao et al., 2020), established graph-based forecasting as the dominant paradigm by jointly modelling spatial and temporal dependencies. Since then, research has focused on increasingly expressive spatial representations (Fang et al., 2021; Shao et al., 2022c; Wei et al., 2024). Adaptive and dynamic graph methods, such as Graph WaveNet (Wu et al., 2019), AGCRN (Bai et al., 2020), and DGCRN (Li et al., 2023), learned connectivity from data instead of relying solely on predefined graphs, while attention- and Transformer-based architectures, including ASTGCN (Guo et al., 2019), GMAN (Zheng et al., 2020), and STTN (Xu et al., 2020), captured more flexible global and heterogeneous inter-sensor relationships. Despite their architectural differences, these methods all seek more effective ways to model spatial information, rather than questioning how much predictive information it contributes.

2.2 Predictive Information in Spatio-Temporal Forecasting

Existing studies have examined predictive information through feature analyses and architectural ablations, but these investigations remain confined to individual models. Meanwhile, several studies have questioned the necessity of explicit graph modelling. STID (Shao et al., 2022a) matches STGNNs using per-sensor identity embeddings without graph propagation; SimST (Liu et al., 2023b) replaces message passing with simple local and global summaries; STAEformer (Liu et al., 2023a) attributes much of Transformer performance to adaptive embeddings rather than graph structure; and PatchSTG (Fang et al., 2025) further achieves SOTA performance through efficient spatial partitioning. Other studies extend the temporal context of spatial models. STEP (Shao et al., 2022b) and STD-MAE (Gao et al., 2024) pre-train on weeks of per-sensor history to enhance short-window STGNNs, and RPMixer (Yeh et al., 2024) mixes one day of all sensors through random projections, achieving strong results on LargeST. In parallel, long-context forecasting models such as PatchTST (Nie et al., 2022) and TimesFM (Das et al., 2024) show that stronger temporal modelling can also achieve remarkable forecasting performance. However, these studies either analyse individual architectures or improve forecasting models, rather than investigating how predictive information is distributed between temporal depth and spatial breadth under a common budget.

2.3 Traffic-flow Theory

Traffic-flow theory provides the physical basis for spatial modelling. Classical theories of kinematic waves and shockwave propagation describe how congestion propagates between adjacent road segments at a finite speed (Lighthill & Whitham, 1955; Richards, 1956). Recent models such as PDFormer (Jiang et al., 2023) and STDDE (Long et al., 2024) build propagation delay into graph-based architectures. However, existing work treats propagation as a modelling problem rather than an information problem. The more fundamental question is whether traffic propagation contains enough observable predictive information to justify propagation-aware architectures.

3 Empirical Framework

3.1 Problem Formulation and Observation-Budget Framework

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ denote a road network, where $\mathcal{V}$ is the set of $|\mathcal{V}|$ sensors, $\mathcal{E}$ denotes the edges connecting sensors, and $A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the weighted adjacency matrix. At each time step t, the traffic state is represented as $\mathbf{x}_t \in \mathbb{R}^{|\mathcal{V}|}$. Given the previous T history steps, $\mathbf{X} = (\mathbf{x}_{t-T+1}, \ldots, \mathbf{x}_t) \in \mathbb{R}^{T\times|\mathcal{V}|}$, the task is to predict the next H horizons, $\hat{\mathbf{Y}} \in \mathbb{R}^{H\times|\mathcal{V}|}$.

Conditioning on A is a modelling choice rather than a requirement of the forecasting task. A graph-based predictor jointly models all sensors as $\hat{\mathbf{Y}} = f_\theta(\mathbf{X}, A)$. A graph-free predictor instead forms the same network prediction $\hat{\mathbf{Y}}$ by forecasting each sensor from a local observation window: the prediction for sensor i is $\hat{\mathbf{y}}_i = g_\theta\big(\mathbf{X}_i^{(T,N)}\big)$, where $\mathbf{X}_i^{(T,N)}$ collects the T most recent steps of sensor i together with the recent histories of its N nearest neighbours.[1] The pair (T, N) specifies the observation budget: T is the temporal depth (history length per sensor) and N is the spatial breadth (number of neighbouring sensors). Setting $N = 0$ recovers a purely temporal, univariate predictor $\hat{\mathbf{y}}_i = g_\theta(\mathbf{x}_i)$ with $\mathbf{x}_i \in \mathbb{R}^T$, while a graph-based model corresponds to the maximal breadth $N = |\mathcal{V}|-1$. The two formulations therefore differ only in how the observation budget is allocated between temporal depth T and spatial breadth N.

Table 1: Two observation regimes: wide and shallow for STGNNs; narrow and deep for univariate models.

	Temporal depth (steps)	Spatial breadth (sensors)	Obs./ prediction
STGNN	12	716	$12 \times 716 = 8592$
TimesFM	512	1	512

[1] The own-sensor history has length T, whereas each neighbour contributes only its 12 most recent steps, matching the per-sensor temporal budget of a standard STGNN. The observation budget is therefore $T + 12N$ values rather than $T(N+1)$, which is the budget axis $B = T + 12N$ used in Section 4.1 and Figure 2.

Our goal is to measure how much predictive information about future traffic states is encoded in a given observation budget. Since predictive information cannot be measured directly, we adopt a predictive modelling framework: we train predictive models using a fixed observation budget and treat their predictive performance as a proxy for the information encoded in that budget.

Table 1 illustrates the contrasting observation regimes of conventional graph-based and long-context univariate forecasting. Note that the observation count $T + 12N$ describes the input of each prediction rather than the computational cost, since a graph model forecasts all $|\mathcal{V}|$ sensors from one input block. The two regimes in Table 1 therefore use different budgets, and the comparison under an equal budget is carried out along the budget line in Section 4.1.

3.2 Predictive Information Measurement

We employ two complementary measurement models, one linear and one nonlinear, so that results consistent across both are less likely to reflect the characteristics of a single model.

The linear model is ridge regression on the historical-average residuals, fitted for each sensor and horizon. Its closed-form solution is optimal for the regularized training objective, ruling out optimization failure, though not limited capacity. Separate regularization strengths for the target-history and spatial features keep a shared penalty from over-shrinking the spatial block (Appendix B.1).

The nonlinear model is a lightweight patch-based mixer (1.15M parameters) built from MLP-Mixer blocks (Tolstikhin et al., 2021) with per-window normalization similar to RevIN (Kim et al., 2021) (Appendix B.2). It is retrained independently for every (T, N) allocation, with hyperparameters selected once on the San Diego validation split. Neighbours enter through a zero-initialized spatial gate, so each configuration starts with no spatial contribution, though successful optimization is not guaranteed. Both models are complementary empirical probes of predictive utility within the tested model classes.

3.3 Evaluation Protocol

We use the LargeST benchmark (Liu et al., 2023c), the largest open-source benchmark for large-scale traffic flow forecasting. It comprises four California subsets ranging from 716 to 8,600 sensors: San Diego (SD), Greater Bay Area (GBA), Greater Los Angeles (GLA), and California (CA). Following the LargeST protocol, we use the 2019 traffic flow data, aggregate the 5-minute measurements into 15-minute intervals, adopt a chronological 60/20/20 train/validation/test split, and evaluate horizons of 3, 6, and 12 steps (45, 90, and 180 minutes).

Traffic flow values are z-score normalized using statistics computed from the training set only. Missing observations are retained as NaN and excluded from both training and evaluation through masked losses and masked metrics. Unless stated otherwise, we evaluate on test windows placed every 8 steps over all sensors (811 windows per subset); using every test window lowers the errors of all models by about 0.1 MAE and does not change the comparison (Appendix E.2). We report MAE, RMSE, and MAPE for each horizon and their average. Baseline results are taken from their publications, and two baselines are also retrained in our pipeline to check for protocol effects (Section 5.2).

4 Analysis

We analyse how predictive information is distributed under a fixed observation budget through three research questions (RQs): **RQ1:** How is predictive information distributed under a fixed observation budget? **RQ2:** What information remains in the spatial dimension? **RQ3:** Why does graph propagation contribute so little predictive value?

4.1 Information Distribution (RQ1)

We measure how the remaining predictive information is distributed by measuring forecast error over the full $T \times N$ allocation plane, where $T \in \{12, 24, 48, 96, 192, 384, 512\}$ denotes the target history length and $N \in \{0, \ldots, 512\}$ the number of additional neighbour sensors. Each additional sensor contributes its most recent 12 observations, matching existing STGNNs.

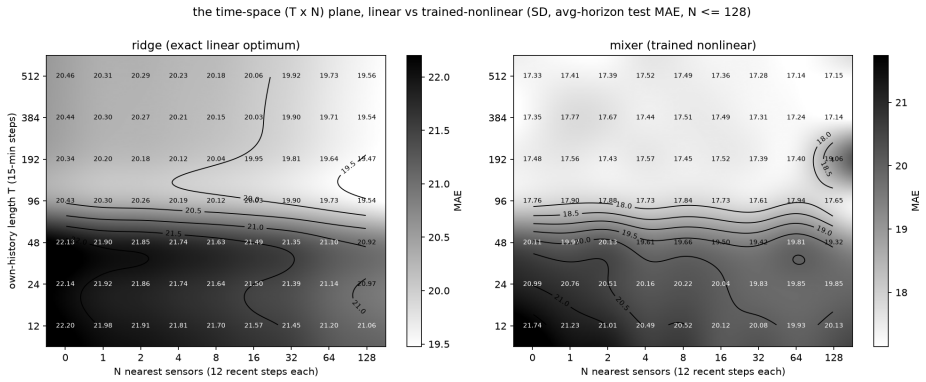


Figure 2: The time-space ($T \times N$) plane on the SD subset (average-horizon test MAE, $N \leq 128$), comparing the closed-form linear predictor (ridge, left) with the trained nonlinear mixer (right).

Figure 2 shows a consistent allocation pattern across both measurement models. Both exhibit a clear transition at $T = 96$, when the input covers a full daily cycle. Extending the history from $T = 48$ to $T = 96$ gives the largest MAE reduction across the allocation plane, with only marginal gains beyond one day. Adding spatial context is less effective than extending temporal history. For ridge regression, increasing T from 12 to 96 reduces MAE from 22.20 to 20.43 (-1.77), while increasing N from 0 to 128 at $T = 12$ reduces it to 21.06 (-1.14). Under a fixed budget $B = T + 12N = 96$, allocating all observations to temporal history ($T = 96, N = 0$) outperforms the mixed allocation $(48, 4)$ for both models. With a larger budget, a few neighbours can help once the history exceeds one day: with $B = 512$, 128 steps of history plus three neighbours improves on pure history by 0.29 MAE for ridge regression, but not for the mixer (Appendix B.4). These results indicate that predictive information is concentrated primarily in temporal depth, with spatial context providing comparatively limited additional benefit.

When the history is shorter than one day, the mixer gains more from neighbours than ridge regression (1.8 versus 1.0 MAE at $T = 12$). Once $T \geq 96$, however, the mixer gains almost nothing from neighbours across $N \leq 128$, while ridge still gains 0.5–0.9 MAE, indicating that nonlinear temporal modelling recovers from the target's own history what the linear model obtains from neighbours. Giving neighbours longer histories does not change this picture (Appendix B.4), and cross-sensor attention on top of the adapted TimesFM recovers at most 0.08 MAE (Appendix B.5). On GBA and GLA, ridge regression shows similar history gains but almost no spatial gain.

Table 2: Random sensors recover 85% of the gain from nearest sensors; mean ± s.e. over five paired seeds.

(T=48, N=4), 5 seeds	test MAE	paired gain
no spatial input	20.14 ± 0.05	—
4 *random* sensors	19.68 ± 0.08	$+0.458 \pm 0.046$
4 *nearest* sensors	19.60 ± 0.06	$+0.540 \pm 0.028$
nearest − random		$+0.082 \pm 0.026$

4.2 Understanding Spatial Information (RQ2)

Where spatial input does help (history shorter than one day), we ask what the additional sensors provide. If this gain arises from road-network connectivity, replacing the N nearest sensors with an equal number of random sensors should substantially reduce accuracy. Across the grid, random sensors give essentially the same benefit as the nearest ones along the entire N axis and at every history length. At $(T = 48, N = 4)$, the nearest sensors improve the mixer by 0.54 MAE and random sensors by 0.46, leaving only 0.08 attributable to proximity (Table 2). Road-network proximity thus explains only a small fraction of the spatial gain; most of it reflects broader regional traffic conditions shared across many sensors, not graph-specific propagation.

4.3 Graph Propagation Analysis (RQ3)

This seemingly contradicts congestion propagation in traffic flow theory. We therefore examine whether adjacent sensors provide useful lead–lag information before it is already contained in the target sensor's own history. Using the data's native 5-minute resolution, we compute the peak cross-correlation lag for all 16,930 adjacent sensor pairs. If congestion propagation carried substantial predictive information, adjacent sensors would exhibit consistent positive lead–lag relationships. Instead, Figure 3 shows that 53.9% of adjacent pairs peak at lag zero, 76.5% within one benchmark interval (15 minutes), and 91.2% within the shortest forecasting horizon (45 minutes). Before forecasting, propagated information is almost always present in the target sensor's own recent history.

The sensor geometry explains why. Consecutive sensors on the same freeway in LargeST are separated by a median distance of just 0.91 km. According to kinematic wave theory (Lighthill & Whitham, 1955; Richards, 1956), a congestion shockwave propagating upstream at 15–20 km/h traverses 76.5% of adjacent-sensor gaps in under 5 minutes and 99.4% within 15 minutes. The propagation delay between neighbouring sensors is thus typically shorter than a single aggregation interval, leaving little usable lead–lag information. The waves are nevertheless observable: conditioning on downstream congestion, where shockwaves are expected, raises the downstream-leading share from 6.8% to 12.4% (implied speed ≈11 km/h), yet even then 84.7% of pairs still peak within ±5 minutes.

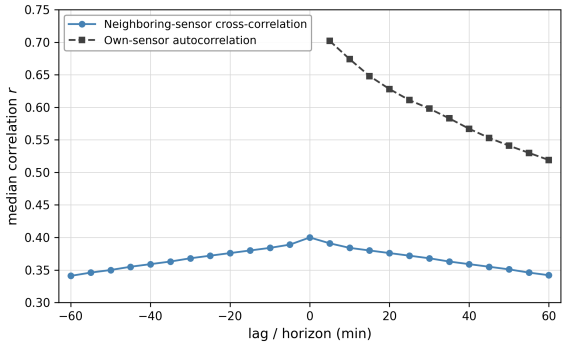


Figure 3: HA residual correlations at 5-minute resolution: medians over 15,642 adjacent sensor pairs with peak $r \geq 0.2$ on the training split.

Conditional predictability. A near-zero peak lag shows *when* a neighbour's signal arrives, not whether it adds predictive value beyond the target's own past. We therefore ask how much of the recipe's residual the k nearest neighbours predict out of sample, using both ridge regression and a nonlinear MLP. The aggregate MLP gain on SD (0.35 MAE, CI $[-0.08, +0.69]$) is statistically unresolved, not shown to be zero. Statistically supported MLP gains (≈2.8 MAE) appear in *sudden-onset* cells, where the target is extreme but its own context is quiet. These cells, about 2% of the total (Appendix C), mark a boundary of the temporal-only view.

4.4 When Physical Neighbourhoods Matter

The analysis above suggests that spatial information becomes useful only when the data resolve propagation. We test this on FT-AED (Coursey et al., 2024), a freeway corridor with 196 lane-level sensors recording speed every 30 seconds, where most adjacent pairs show a clear upstream-to-downstream lag. A frozen temporal forecaster is augmented with a spatial residual that averages a fixed window of same-lane neighbours without reading any graph. On 29 unseen weekdays, with predictions fixed in advance, the window reduces error by 0.50 mph relative to an equally sized residual that reads only the target's own history (29/29 days). Table 3 shows that this gain is strongly regime dependent: about 0.2 mph in free flow but about 1.5 mph near capacity and in recurrent congestion. Exact connectivity is not required, since the window reads no edges and outperforms a graph-based model, whose error grows by 0.40 mph under full graph corruption. Physically relevant neighbours, however, cannot be replaced: matched distant sensors cost

Table 3: FT-AED, 30-second speed: MAE (mph) at 4 min on 29 unseen weekdays, by traffic regime.

Spatial input	Free flow	Near cap.	Recurrent
None (temporal only)	2.49	6.84	6.26
Local window	**2.28**	**5.38**	**4.79**
without upstream	2.29	5.64	4.96
Matched distant	2.38	6.59	6.06
Random	2.38	6.70	6.09

0.10 mph in free flow but 1.21 mph near capacity. On LargeST San Diego the same design shows no dependence on proximity, so physical neighbourhoods become indispensable only when traffic approaches capacity and sampling resolves propagation (Appendix H). The temporal-only recipe is therefore suited to data whose sampling interval exceeds the propagation time; at finer resolution, a temporal backbone should be complemented by a protected local neighbourhood.

5 Temporal-only Forecasting and Evaluation

Motivated by the preceding analysis, we develop a temporal-only forecasting recipe that exploits long history without any spatial information. The mixer of Section 3 provides the main evidence that the allocation, not model scale, drives accuracy; the foundation model shows that the principle transfers to an off-the-shelf pretrained backbone. We then investigate: **RQ4:** Can a temporal-only recipe outperform graph-based forecasting models? **RQ5:** Where does the proposed recipe fail, and can the failure be repaired? **RQ6:** Does the proposed recipe generalize and scale?

5.1 Forecasting Recipe

Backbone and adaptation. Our recipe builds on TimesFM 2.5 (Das et al., 2024), a pretrained univariate time-series foundation model. We keep it frozen and forecast each sensor separately from $T = 512$ steps (5.3 days) of that sensor's own history, producing all $H = 12$ horizon steps in one forward pass. We adapt the backbone with three lightweight modules instead of full fine-tuning; all of them operate on each sensor independently, without a road graph or cross-sensor computation.

(i) Low-rank adaptation. We employ Low-rank Adaptation (LoRA) (Hu et al., 2021) to adapt the frozen backbone to the traffic distribution. Each adapted projection matrix is updated as $W' = W + BA$, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$ are trainable low-rank matrices with $r \ll d$. We apply rank-8 LoRA to all linear projections of the last six layers, adding about 0.6M parameters.

(ii) Calendar embeddings. We encode daily and weekly periodicity through learnable time-of-day and day-of-week embeddings, updating each input token as $\tilde{\mathbf{e}}_t = \mathbf{e}_t + E^{\mathrm{tod}}[\mathrm{tod}(t)] + E^{\mathrm{dow}}[\mathrm{dow}(t)]$, where E^{tod} and E^{dow} denote the corresponding embedding tables. Both tables are zero-initialized, so training starts from the LoRA-adapted backbone.

(iii) Historical-average residualization. Following seasonal decomposition (Cleveland et al., 1990), the backbone models only the residual component. A historical average (HA) is computed for each sensor as the training-set mean value for every combination of time-of-day and day-of-week, yielding a 7×96 lookup table. For each prediction window, the HA values of the H target steps form $\mathbf{y}_i^{\mathrm{ha}} \in \mathbb{R}^H$, and the backbone is trained to predict the residual sequence, from which the final forecast is recovered:

$$\mathbf{r}_i = \mathbf{y}_i - \mathbf{y}_i^{\mathrm{ha}}, \qquad \hat{\mathbf{r}}_i = g_\theta(\mathbf{x}_i), \qquad \hat{\mathbf{y}}_i = \hat{\mathbf{r}}_i + \mathbf{y}_i^{\mathrm{ha}}. \tag{1}$$

5.2 Performance Evaluation (RQ4)

Baselines. We compare the proposed method against two categories of representative baselines: (1) **Graph-based STGNNs**, represented by the strongest graph-based model on each subset. (2) **Existing graph-free methods**, including STID (Shao et al., 2022a), PatchSTG (Fang et al., 2025), and RPMixer (Yeh et al., 2024), which remove explicit graph propagation while retaining dedicated architectures; RPMixer also reads one day of every sensor.

Performance. We build the recipe incrementally on San Diego (Table 13, Appendix D.1). The zero-shot foundation model reaches 24.18 MAE, indicating that it has learned diurnal structure during pretraining but is far from the trained baselines. LoRA provides the largest improvement (24.18 to 18.34), calendar cues further reduce the average MAE to 18.08, and historical-average residualization yields the final temporal-only recipe with an average MAE of 17.43, outperforming the strongest graph-based baseline, Graph WaveNet (18.18 average MAE).

Part of this gap could come from our evaluation pipeline or from the HA prior, which the baselines do not use. We therefore retrain Graph WaveNet and STID in our pipeline, with and without HA

Table 4: Performance on LargeST. *: strongest graph-based baseline. # Param: trainable parameters (our recipes add a frozen 200M backbone). Bold/underline: best/second best.

Dataset	Method	# Param	Horizon 3			Horizon 6			Horizon 12			Average		
			MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SD	Graph WaveNet*	311K	15.24	25.13	9.86%	17.74	29.51	11.70%	21.56	36.82	15.13%	18.18	30.49	12.23%
	STID	258K	15.14	25.07	9.84%	17.63	29.16	11.46%	21.02	36.72	15.02%	17.93	30.32	12.11%
	PatchSTG	2.3M	14.53	24.34	9.22%	16.86	28.63	11.11%	20.66	36.27	14.72%	17.35	29.75	11.68%
	RPMixer	–	15.12	–	–	17.04	–	–	19.60	–	–	17.25	–	–
	Temporal-only recipe	746K	14.84	25.01	9.44%	17.40	30.24	11.94%	20.05	36.92	14.08%	17.43	30.72	11.82%
	Recipe + gate	1.5M	14.50	24.57	9.13%	16.86	29.18	11.38%	19.23	35.16	13.16%	**16.87**	**29.55**	**11.23%**
GBA	D^2STGNN*	446K	17.54	28.94	12.12%	20.92	33.92	14.89%	25.48	40.99	19.83%	21.31	34.62	15.61%
	STID	711K	17.30	29.02	13.56%	20.47	33.34	16.60%	24.35	39.87	20.60%	20.71	34.08	16.92%
	PatchSTG	3.1M	16.81	28.71	12.25%	19.68	33.09	14.51%	23.49	39.23	18.93%	19.99	33.68	**15.23%**
	RPMixer	–	17.35	–	–	19.44	–	–	21.65	–	–	19.48	–	–
	Temporal-only recipe	746K	16.37	28.91	14.17%	19.20	34.04	17.49%	21.49	38.63	19.53%	19.02	33.86	17.06%
	Recipe + gate	1.5M	15.96	28.37	13.72%	18.55	33.10	16.81%	20.69	37.25	18.67%	**18.40**	**32.91**	16.40%
GLA	AGCRN*	792K	17.27	29.70	10.78%	20.38	34.82	12.70%	24.59	42.59	16.03%	20.75	35.70	13.17%
	STID	901K	16.35	27.12	9.82%	19.81	33.26	12.14%	24.51	42.39	16.07%	20.22	34.26	12.68%
	PatchSTG	1.7M	15.84	26.34	9.27%	19.06	31.85	11.30%	23.32	39.64	14.60%	19.41	32.61	**11.72%**
	RPMixer	–	16.49	–	–	18.82	–	–	21.18	–	–	18.83	–	–
	Temporal-only recipe	746K	15.70	26.51	10.15%	18.41	31.67	12.85%	21.08	37.65	15.38%	18.40	31.94	12.79%
	Recipe + gate	1.5M	15.33	25.94	9.62%	17.81	30.50	12.06%	20.40	36.17	14.51%	**17.85**	**30.87**	12.06%
CA	STGODE*	1.0M	17.57	29.91	13.91%	20.98	36.62	16.88%	25.46	45.99	21.00%	21.34	37.51	17.26%
	STID	1.5M	15.47	26.11	11.21%	18.46	31.34	13.70%	22.55	39.05	17.78%	18.83	32.17	14.23%
	PatchSTG	1.8M	14.69	24.82	10.51%	17.41	29.43	12.83%	21.20	36.13	16.00%	17.77	30.13	**13.11%**
	RPMixer	–	15.90	–	–	17.79	–	–	19.93	–	–	17.87	–	–
	Temporal-only recipe	746K	14.64	25.37	11.38%	17.14	30.12	14.16%	19.51	35.39	16.53%	17.09	30.29	14.02%
	Recipe + gate	1.5M	14.30	24.89	10.90%	16.62	29.19	13.49%	18.88	34.23	15.76%	**16.60**	**29.44**	13.38%

(Appendix A.2). The retrained Graph WaveNet reaches 18.71 MAE, close to its published 18.18, and HA reduces its error by only 0.20 (0.15 for STID), compared with 0.62–0.65 for our 512-step models. With HA, Graph WaveNet reaches 18.51, still well above the 17.43 of our recipe. We also train the mixer from scratch on the same 512-step input. It reaches 17.31 MAE without any pretraining, whereas restricting it to 12 steps raises the error to 21.20, close to DCRNN (Appendix D.3). Without re-tuning, the same mixer reaches 18.57, 17.92, and 16.64 on GBA, GLA, and CA, below every baseline and below the adapted TimesFM. The improvement therefore comes from the longer temporal context rather than from the 200M parameters of TimesFM.

Table 4 further compares the proposed temporal-only recipe against all baselines on the four LargeST subsets. The recipe consistently outperforms the strongest graph-based STGNN and STID across all four subsets. Although it does not surpass PatchSTG and RPMixer on SD, adding the context gate introduced in Section 5.3 gives the lowest average MAE on every subset. On SD, the gain over PatchSTG at 45 and 90 minutes is within noise, and PatchSTG still has the lowest MAPE on GBA, GLA, and CA. The margin over the strongest graph-based baseline increases with network size, from 0.75 MAE on SD (716 sensors) to 4.25 MAE on CA (8,600 sensors). This trend is partly confounded because the strongest graph models cannot be trained on CA, but the margin over PatchSTG and RPMixer also grows, from about 0.4 MAE on SD to 1.0–1.6 MAE elsewhere.

5.3 Failure Analysis and Repair (RQ5)

We next ask whether the recipe's errors concentrate in specific regimes and can be repaired.

Failure analysis. We divide the SD test set into deciles of target deviation from the historical average. The resulting error profile in Figure 4 is strongly regime dependent. The HA variant outperforms the no-HA variant by 1.5–2.5 MAE across deciles 1–8, but underperforms it by about 10 MAE in the most atypical decile. The recipe therefore fails primarily under strongly atypical conditions, rather than under ordinary traffic.

Context-gated repair. When an anomaly is already under way, this failure can be anticipated from the recent traffic context. We therefore introduce a context gate that selects between the two graph-free variants according to the context deviation, defined as the mean absolute difference between the last 12 observed steps and the historical average. If the context deviation exceeds the 90th percentile of its training-set distribution, the model without HA is used; otherwise, the HA variant is selected. It needs no extra training or test-set tuning and is insensitive to the threshold.

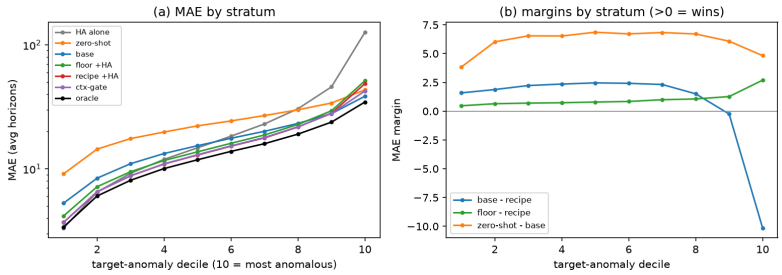


Figure 4: Evaluation by traffic regime on San Diego (deciles of target deviation from the historical average). (a) MAE per decile. (b) MAE of the no-HA variant (base) minus the HA variant (recipe) falls from about $+2.5$ in typical deciles to -10 in the most atypical one; the gating rule removes most of this trade-off.

As shown in Table 4, the gated model consistently improves the temporal-only recipe across all four LargeST subsets. On San Diego, it mitigates HA-related errors when the anomaly is already visible in the observed context (45.41 to 38.48 MAE; Appendix D.2) and reaches the lowest overall MAE (16.87), better than both variants at every horizon. It cannot anticipate sudden-onset events, whose context is quiet (MAE unchanged at 75.49); these remain unresolved and are where spatial inputs show added value (Section 4.3). Similar improvements on GBA, GLA, and CA show that the repair generalizes. Although results come from single runs, seed variation (0.03–0.05 MAE) is much smaller than the gate's improvement, which remains significant under a day-block bootstrap on all subsets (Appendix E.4); the gate also has the lowest RMSE on every dataset (Appendix E.3).

5.4 Generalization and Scalability (RQ6)

Within San Diego, a model trained on half the sensors retains nearly the same gain on the held-out half, suggesting that its parameters are not tied to individual sensor identities. Trained on San Diego and evaluated directly on the unseen GBA network, the model incurs only 0.12 higher MAE than its GBA-trained counterpart and outperforms the strongest GBA-trained STGNN at every horizon (Appendix F.2). The recipe also scales: sensors are forecast independently, with no graph construction or message passing, whereas several STGNNs exceed GPU memory on CA.

6 Conclusion and Limitations

We asked whether, under a fixed observation budget, predictive capacity is better spent on temporal depth or spatial breadth. A systematic measurement of predictive information shows that, in the benchmark settings studied, temporal depth dominates: once the target's history spans one daily cycle, additional spatial observations add little unique information, and the small remainder is broad regional context rather than road-network propagation, because congestion reaches neighbouring sensors within one aggregation interval. Spatial inputs show statistically supported value in sudden-onset events and, when the sampling resolves propagation, near capacity, where physical neighbourhoods become indispensable even though exact connectivity is not required. Guided by this, our temporal-only recipe outperforms graph-based baselines on all LargeST subsets; a lightweight mixer trained from scratch matches the adapted foundation model, showing that accuracy follows the allocation rather than model scale, and a zero-training context gate gives the best overall performance. The takeaway is a shift in framing: how predictive information is distributed under a fixed budget, rather than ever more sophisticated spatial modelling.

Our benchmark results use 15-minute flow data, with the timing and linear analyses repeated at 5-minute resolution (Appendix B.6), and the high-resolution speed evidence comes from a single corridor. The nonlinear allocation plane is measured only on SD, and long-input graph baselines are not evaluated. The context gate uses a fixed threshold and cannot anticipate sudden-onset events; learned blending of the two variants leaves further headroom (Appendix F.1). Network-level speed datasets such as METR-LA and PEMS-BAY are a natural next test.

AI USE STATEMENT

In this work, we used generative AI tools primarily to assist with manuscript preparation and language polishing, as well as coding and debugging tasks supporting experimental execution. All AI-assisted content was reviewed by the authors, who take full responsibility for the final manuscript, implementation, and reported results.

REPRODUCIBILITY STATEMENT

All experiments use the publicly available LargeST benchmark (Liu et al., 2023c) under its standard protocol; data preprocessing and the evaluation protocol are described in Section 3.3 and Appendix A.1. Baseline sources, training configurations, and compute resources are given in Appendices A.2–A.4, and the measurement models, forecasting recipe, and context gate are specified in Appendices B.1–B.3, D.1, and D.2. Robustness checks across seeds, evaluation strides, and bootstrap resampling are reported in the validation appendix (Appendix E.1). Anonymized code is included as supplementary material, with instructions for environment setup and data preparation and a mapping from each table and figure to the command that produces it (Appendix A.5).

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. *Advances in Neural Information Processing Systems*, 33:17804–17815, 2020.

Srinivasa Ravi Chandra and Haitham Al-Deek. Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, 13(2):53–72, 2009.

Robert B Cleveland, William S Cleveland, Jean E McRae, Irma Terpenning, et al. Stl: A seasonal-trend decomposition. *J. off. Stat*, 6(1):3–73, 1990.

Austin Coursey, Junyi Ji, Marcos Quinones-Grueiro, William Barbour, Yuhang Zhang, Tyler Derr, Gautam Biswas, and Daniel B. Work. FT-AED: Benchmark dataset for early freeway traffic anomalous event detection. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 37, 2024.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, April 2024.

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*, KDD '25, pp. 307–317, New York, NY, USA, July 2025. Association for Computing Machinery. ISBN 979-8-4007-1245-6. doi: 10.1145/3690624.3709177.

Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 364–373, 2021.

Haotian Gao, Renhe Jiang, Zheng Dong, Jinliang Deng, Yuxin Ma, and Xuan Song. Spatial-temporal-decoupled masked pre-training for spatiotemporal forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3998–4006, 2024.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):922–929, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.3301922.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021.

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, 2023.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution. *ACM Trans. Knowl. Discov. Data*, 17(1):9:1–9:21, February 2023. ISSN 1556-4681. doi: 10.1145/3532611.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting, February 2018.

Michael James Lighthill and Gerald Beresford Whitham. On kinematic waves ii. a theory of traffic flow on long crowded roads. *Proceedings of the royal society of london. series a. mathematical and physical sciences*, 229(1178):317–345, 1955.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-Temporal Adaptive Embedding Makes Vanilla Transformer SOTA for Traffic Forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pp. 4125–4129, New York, NY, USA, 2023a. Association for Computing Machinery. ISBN 979-8-4007-0124-5. doi: 10.1145/3583780.3615160. URL https://dl.acm.org/doi/10.1145/3583780.3615160.

Xu Liu, Yuxuan Liang, Chao Huang, Hengchang Hu, Yushi Cao, Bryan Hooi, and Roger Zimmermann. Do we really need graph neural networks for traffic forecasting? *arXiv preprint arXiv:2301.12603*, 2023b.

Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. *Advances in Neural Information Processing Systems*, 36:75354–75371, December 2023c.

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.

Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation research part C: emerging technologies*, 54:187–197, 2015.

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

Paul I Richards. Shock waves on the highway. *Operations research*, 4(1):42–51, 1956.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-Temporal Identity: A Simple yet Effective Baseline for Multivariate Time Series Forecasting. https://arxiv.org/abs/2208.05233v2, August 2022a.

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

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. *arXiv preprint arXiv:2206.09112*, 2022c.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

Ye Wei, He Haitao, Kunhao Yuan, Gerald Schaefer, Zhigang Ji, and Hui Fang. A Memory-augmented Conditional Neural Process model for traffic prediction. *Knowledge-Based Systems*, 304:112578, November 2024. ISSN 0950-7051. doi: 10.1016/j.knosys.2024.112578.

Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for Deep Spatial-Temporal Graph Modeling, May 2019.

Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

Chin-Chia Michael Yeh, Yujie Fan, Xin Dai, Uday Singh Saini, Vivian Lai, Prince Osei Aboagye, Junpeng Wang, Huiyuan Chen, Yan Zheng, Zhongfang Zhuang, Liang Wang, and Wei Zhang. RPMixer: Shaking up time series forecasting with random projections for large spatial-temporal data. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3634–3640, July 2018. doi: 10.24963/ijcai.2018/505.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, September 2020. ISSN 1558-0016. doi: 10.1109/TITS.2019.2935152.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. GMAN: A Graph Multi-Attention Network for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (01):1234–1241, April 2020. ISSN 2374-3468. doi: 10.1609/aaai.v34i01.5477.

A Experimental Details

A.1 Dataset

Table 5: The four LargeST subsets used in this work.

Dataset	#Points	Timespan	% missing
SD	716	01/01/2019–12/31/2019	2.16
GBA	2,352	01/01/2019–12/31/2019	0.68
GLA	3,834	01/01/2019–12/31/2019	1.05
CA	8,600	01/01/2019–12/31/2019	1.51

All experiments in the main paper use the LargeST benchmark (Liu et al., 2023c); this section summarizes the dataset, preprocessing and evaluation settings used throughout the study. LargeST comprises four California subsets of increasing spatial scale (SD, GBA, GLA, and CA), spanning 2017–2021 at a native 5-minute temporal resolution and providing comprehensive metadata for constructing spatial topologies. Table 5 summarizes the four subsets used in this work. Following the official LargeST protocol, we use the 2019 data only, aggregate the native 5-minute observations into 15-minute intervals by averaging groups of three, and split each subset chronologically into 60%/20%/20% train/validation/test sets. In addition, Appendices E.3 and E.4 report the context gate on the Seattle Loop dataset (323 detectors, full-year 2015, 5-minute speeds), which tests the gate outside California and on a speed variable.

Preprocessing. We z-score normalize each series using training-split statistics only. The mixer additionally applies a per-window normalization similar to RevIN (Kim et al., 2021) inside its own architecture (Appendix B.2); this complements rather than replaces the shared z-score normalization. Missing values are kept as NaN throughout and excluded from every loss and metric via masking. Forecast horizons are 3, 6, and 12 steps (45, 90, and 180 minutes). We report results for each horizon and their mean ("avg"), computed over H3/H6/H12 (unlike the LargeST benchmark, which averages over all 12 forecasting horizons). Unless stated otherwise, evaluation uses a dense set of test windows at stride 8 over all sensors, yielding exactly 811 windows on every subset.

A.2 Baselines

Graph-based baselines. We compare against six representative graph-based traffic forecasting models. **STGCN** (Yu et al., 2018) introduces graph convolutions for spatial dependency modelling. **DCRNN** (Li et al., 2018) models traffic propagation through diffusion graph convolutions and recurrent networks. **Graph WaveNet** (Wu et al., 2019) combines adaptive graph learning with dilated temporal convolutions. **AGCRN** (Bai et al., 2020) learns node-specific adaptive graph representations. **D^{2}STGNN** (Shao et al., 2022c) decouples dynamic spatial and temporal dependencies using adaptive graph modelling. **STGODE** (Fang et al., 2021) models traffic dynamics through neural ordinary differential equations on graphs.

Graph-free baselines. We additionally compare against two representative graph-free methods. **STID** (Shao et al., 2022a) removes explicit graph propagation by relying on learnable spatial-temporal identity embeddings. **PatchSTG** (Fang et al., 2025) achieves state-of-the-art performance through efficient spatial partitioning with Transformer-based modelling.

Baseline results are taken from the publicly available results in the camera-ready papers and official GitHub repositories. Following our evaluation protocol, the reported "avg" is computed as the mean over the three forecasting horizons (H3/H6/H12) for all methods, rather than the average over all 12 forecast steps, ensuring a consistent and conservative comparison across methods.

In-pipeline reproduction. To verify that our evaluation pipeline (NaN masking, dense stride-8 evaluation windows, and data splits) does not artificially favor our method, we retrained Graph WaveNet and STID within the same pipeline using the hyperparameters summarized in Table 7. Table 6 compares the reproduced and published results.

Table 6: In-pipeline reproduction of Graph WaveNet and STID.

SD, in-pipeline	45m	90m	180m	avg
Graph WaveNet, raw	15.48	18.15	22.51	18.71
Graph WaveNet, +HA	15.25	18.32	21.94	18.51
STID, raw	15.97	18.48	22.41	18.95
STID, +HA	15.54	18.51	22.36	18.80
Graph WaveNet, published	15.24	17.74	21.56	18.18
STID, published	15.14	17.63	21.02	17.93

The reproduced Graph WaveNet remains close to the published result (18.71 vs. 18.18 MAE), indicating that the improvements of our recipe are not an artifact of our evaluation pipeline. Adding the historical-average (HA) prior reduces the MAE of Graph WaveNet and STID by only 0.20 and 0.15 MAE, respectively, compared with 0.62–0.65 MAE for our 512-step recipe and mixer. This difference is expected: the HA prior mainly accounts for predictable daily periodicity, which short-context models cannot effectively exploit from only a 12-step input window. In contrast, our long-context recipe explicitly models multi-day temporal patterns, making residualization considerably more beneficial.

A.3 Training

Table 7 summarizes the core training hyperparameters for the three model families used in this work. The recipe is fine-tuned with a single fixed configuration across all datasets. The mixer's configuration was chosen once via a 24-configuration sweep (width $\times$ blocks $\times$ learning rate $\times$ dropout) scored on San Diego's dense validation split, and this same configuration is then reused, unswept, for every other region and dataset in the paper. The in-pipeline graph baselines are trained with LargeST's own published hyperparameters, verbatim, using dense-validation early stopping (patience 30) rather than a fixed epoch budget.

A fixed random seed is set before model construction, together with a seeded batch sampler; the mixer is seeded 5 ways, the recipe 4 ways (Appendix E.1).

A.4 Computation

Device selection is automatic (cuda¿mps¿cpu). Multi-region experiments, in-environment baseline reproduction, and seed studies were run on cloud A10G GPUs (the CA recipe on an A100 because of its 8,600-sensor evaluation batches). The 101-cell (T, N) mixer grid (Appendix B.3) cost approximately $15 in total with per-cell checkpointing for idempotent resume, while the four-region replication pass (recipe, base, mixer, and the linear oracle on GBA, GLA, and CA) completed in about 2 hours of wall-clock time across four containers. At full network scale (CA, 8,600 sensors), several representative graph-based baselines (AGCRN, DGCRN, and D^{2}STGNN) could not be trained because they ran out of memory even with batch size 4 on a 48GB NVIDIA RTX A6000, whereas our graph-free recipe forecasts all 8,600 sensors on a laptop without evaluation out-of-memory errors, since sensors are processed independently. Code and exact commands for reproducing every table and figure are provided in the supplementary material.

A.5 Code and Reproducibility

The supplementary code package includes a `README.md` file with instructions for environment setup, data download, and the three main entry points. It also provides `REPRODUCIBILITY.md`, which maps every table and figure in the paper to the exact script and command-line arguments used to generate it.

Table 7: Training configurations for the recipe, from-scratch mixer, and in-pipeline graph baselines.

Parameter	Value	Description
Recipe (adapted TimesFM 2.5)		
optimizer	Adam	Optimizer
steps	3,000	Training steps
learning_rate	3×10^{-4}	Peak learning rate
lr_scheduler	cosine	Decay schedule
batch_size	64	Batch size
context_length	512	Input history length
horizon	12	Forecast steps per pass
Mixer (from-scratch)		
width	256	Hidden dimension
blocks	4	Mixer residual blocks
learning_rate	10^{-3}	Adam learning rate
dropout	0.0	Dropout rate
steps	6,000	Training steps
Graph baselines (in-pipeline)		
optimizer	Adam	Optimizer
learning_rate	10^{-3}	Initial learning rate
weight_decay	10^{-4}	L2 regularization
gradient_clip	5	Max gradient norm
dropout	0.3	Dropout rate
lr_scheduler	none	No schedule
batch_size	64	Batch size
max_epochs	100	Max training epochs
patience	30	Early-stopping patience

B Measurement Models

B.1 Closed-Form Linear Predictor (Ridge)

The linear measurement model is ridge regression on the HA-residual anomalies, with a separate regularization coefficient selected on the validation split for each (sensor, horizon) pair. For spatial variants, we use a two-block formulation with independent regularization coefficients for the target-history block and the spatial block, jointly selected on the validation set. In this design the own-history model is recovered in the limit of an infinite spatial penalty, so the spatial fit is not forced to trade off the two blocks under a single shared penalty. The closed-form solution is optimal for this regularized training objective; it does not guarantee the lowest attainable test MAE, and the linear class has limited capacity, so the measured spatial gain is an empirical estimate of predictive utility within this class rather than a bound on the spatial information in the data. In contrast, a single isotropic regularization coefficient consistently over-shrinks the high-dimensional spatial block, causing own+neighbours to perform 1.9 MAE worse than own-only; this configuration is retained only as a documented pitfall and is not used for any reported results. Training rows whose targets are NaN are assigned the training-set mean uniformly across all variants, introducing no measurable bias (0.000 MAE difference when refitting the own-only model using only complete rows). The "own+all" variant uses a seeded 16-sensor subset on SD, GBA, and GLA for computational tractability; on CA, the corresponding $12S \times 12S$ Gram matrix (approximately 85 GB) is prohibitively large, so this variant is omitted.

B.2 Trained Nonlinear Plane (Mixer)

The nonlinear measurement model is a lightweight patch-mixer trained from scratch and retrained independently for every (T, N) allocation.

Architecture. Input sequences are partitioned into non-overlapping 32-step patches, linearly embedded together with a validity mask, augmented with zero-initialized calendar embeddings, pro-

cessed by four MLP-Mixer (Tolstikhin et al., 2021) residual blocks with RevIN-style normalization (Kim et al., 2021), and decoded by a linear prediction head. For spatial variants, each of the N nearest neighbours contributes a token containing its 12 most recent anomaly values, a validity mask, and the log-haversine distance to the target sensor; a zero-initialized scalar gate merges this neighbour information with the target sensor's own-history representation.

Design choices. The zero-initialized gate ensures every spatial model starts with no spatial contribution, i.e., from its own-history counterpart. It does not guarantee that optimization will then use neighbour information successfully, or that doing so will improve test error. Replacing the zero-initialized gate with random initialization increases the error of a representative probe cell by 1.9 MAE, confirming that the initialization is essential rather than cosmetic.

Training. Hyperparameters are selected once on the San Diego validation split and reused unchanged across all datasets and allocation settings. All (T, N) cells share the same random seed and the same 811 evaluation windows, enabling paired comparisons across the entire allocation plane.

B.3 Full Allocation Planes

Table 8: Closed-form linear predictor (ridge): average-horizon test MAE over the (T, N) plane, SD (count-weighted).

$T\backslash N$	0	1	2	4	8	16	32	64	128	256	512
12	22.20	21.98	21.91	21.81	21.70	21.57	21.45	21.20	21.06	21.07	21.01
24	22.14	21.92	21.86	21.74	21.64	21.50	21.39	21.14	20.97	20.97	20.93
48	22.13	21.90	21.85	21.74	21.63	21.49	21.35	21.10	20.92	20.86	20.82
96	20.43	20.30	20.26	20.19	20.12	20.03	19.90	19.73	19.54	19.50	19.60
192	20.34	20.20	20.18	20.12	20.04	19.95	19.81	19.64	19.47	19.41	19.52
384	20.44	20.30	20.27	20.21	20.15	20.03	19.90	19.71	19.54	19.47	19.60
512	20.46	20.31	20.29	20.23	20.18	20.06	19.92	19.73	19.56	19.49	19.62

Table 9: Trained nonlinear predictor (mixer): average-horizon test MAE over the (T, N) plane, SD (seed 0, main cells).

$T\backslash N$	0	1	2	4	8	16	32	64	128	256	512
12	21.74	21.23	21.01	20.49	20.52	20.12	20.08	19.93	20.13	21.11	21.84
24	20.99	20.76	20.51	20.16	20.22	20.04	19.83	19.85	19.85	21.84	20.05
48	20.11	19.97	20.13	19.61	19.66	19.50	19.42	19.81	19.32	20.97	21.07
96	17.76	17.90	17.88	17.73	17.84	17.73	17.61	17.94	17.65	19.28	19.18
192	17.48	17.56	17.43	17.57	17.45	17.52	17.39	17.40	19.06	19.17	18.26
384	17.35	17.77	17.67	17.44	17.51	17.49	17.31	17.24	17.14	19.19	19.72
512	17.33	17.41	17.39	17.52	17.49	17.36	17.28	17.14	17.15	17.36	19.09

The complete allocation planes confirm the qualitative conclusions reported in Figure 2. The observation-budget grid varies own-history length $T \in \{12, 24, 48, 96, 192, 384, 512\}$ (15-minute steps) against spatial breadth $N \in \{0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$ nearest sensors, where each neighbour contributes its most recent 12 steps, giving a budget axis $B = T + 12N$. Neighbours are selected by haversine k-NN, providing a graph-free spatial axis that extends beyond the road graph's typical ≈25–29-neighbour degree. Tables 8 and 9 report the complete 7×11 grids summarized by Figure 2. For the ridge grid, wide-N columns use a Schur-complement reduction to remain tractable at full sensor population; a standalone N=512 run reproduces the N=256 column to machine precision, and the $(N$=32) cell reproduces the road-graph own+neighbour fit (19.92 vs. 19.95). For the mixer grid, one model is trained for every (T, N) cell (77 main cells, 4 calibration replicates, and 20 random-sensor control cells; 101 in total), all sharing the same seed and 811 test windows to enable paired comparisons.

Both grids exhibit the same dominant pattern: a pronounced daily-cycle transition between $T = 48$ and $T = 96$, after which increasing spatial breadth provides only limited additional benefit. The ridge predictor shows a small neighbour gain that peaks around $N \approx 256$ before slightly reversing,

whereas the trained mixer exploits neighbours only when temporal depth is insufficient (primarily the $T = 12$ row) and becomes unstable in the widest-N columns ($\pm$1–2 MAE; 20–40$\times$ seed variance). Along the fixed-budget frontier, temporal depth consistently dominates mixed allocations (e.g., $(96, 0)$ outperforms every mixed configuration under $B = 96$). Even the best cell on the ridge grid (19.41 MAE at $(192, 256)$) remains substantially above both the temporal-only recipe (17.43) and the gated recipe (16.87).

B.4 Longer Neighbour Histories and Wide Neighbourhoods

Longer neighbour histories. The grid gives each additional sensor only its 12 most recent steps, following the STGNN convention, and one could object that this design prevents spatial input from showing its value. We therefore repeated the linear analysis with long neighbour histories. Given the full 512-step target history, four nearest sensors contributing 512 steps each improve the ridge fit by 0.07 ± 0.07 MAE (paired standard error over a 128-sensor subset), less than the same four sensors contributing 96 steps each (0.33 ± 0.06). A neighbouring sensor's useful content is its current deviation from its own weekly average; its longer history duplicates periodic structure that the HA residualization has already removed, and carrying the extra features costs estimation accuracy. The benefit of neighbour history saturates by about 96 steps. Fixed-budget splits behave consistently with the grid: with 512 observations in total, 128 steps of target history plus three neighbours at 128 steps each improves on all-history by 0.29 MAE for the linear model (an advantage that the trained model does not show, per Figure 2), while 64 steps of target history plus seven neighbours at 64 steps each is 0.97 MAE worse. Any allocation that reduces the target history below one daily cycle increases error by more than the added sensors recover.

Reading many sensors degrades the trained model. At $N \geq 256$ the trained model's error varies by 1–2 MAE across configurations, 20–40 times the seed-to-seed noise, and these failures recur under independent seeds, so they are properties of the configuration rather than chance. The one such cell inside $N \leq 128$, (T=192, N=128) at 19.06, reproduces under a second seed (19.17) while all its neighbours on the grid sit at 17.1–17.6. We interpret the instability as the estimation and optimization cost of many undifferentiated input features rather than as an information-theoretic limit. A differently regularized spatial model might avoid it, and our results do not bound what such a model could gain. The ridge predictor shows a related pattern within the linear class: its N=512 column is worse than smaller N at every $T \geq 96$ (Table 8).

B.5 Cross-Sensor Attention Modules on the Adapted TimesFM

Before the linear and trained-model measurements, we tried to recover the adapted TimesFM's short-horizon deficit with spatial correctors, which are the strongest spatial readers in this paper because they sit on top of the strongest temporal model. Each corrector keeps the adapted TimesFM (without HA) frozen and predicts the residual of its forecast from the recent readings of other sensors: a per-sensor encoder of the last 12 steps of every sensor, a multi-head attention layer across sensors, and a per-horizon readout whose output is added to the frozen forecast through a gate initialized to zero, so training starts from the unadapted model. Correctors are trained on cached forecasts with a chronological validation split, weight decay, and early stopping, over a sweep of width, depth, and regularization. Four variants were evaluated on the dense San Diego windows (average MAE; the frozen model scores 18.08, and 15.50 at 45 minutes):

- unrestricted attention over all 716 sensors, raw flow as input: 18.13 (no improvement; held-out short-horizon error floors at about 14.5 regardless of capacity or regularization);
- attention restricted to the road graph (about 25 neighbours per sensor), in the upstream, downstream, or both directions, with or without an affinity bias: 18.09, with the same held-out floor as the unrestricted version, so the graph structure is not the bottleneck;
- unrestricted attention with the input replaced by each sensor's deviation from its historical average: 18.05, with the 45-minute error falling from 15.50 to 15.42, the only variant that helps; tripling the number of training windows leaves it at 18.05, so the gain is saturated rather than data-limited;
- the deviation input with the graph restriction: 18.09, i.e. the restriction removes the gain.

The recoverable spatial signal on top of the adapted TimesFM is thus about 0.08 MAE at 45 minutes and 0.03 on average, and did not increase under any of the capacity, regularization, graph-structure, direction, or data-volume variations we tried. The one feature that helped, deviation from the sensor's own historical average, applied univariately is the HA residualization of Section 5.1, which gains 0.65 MAE; the temporal use of the feature is worth twenty times its spatial use.

B.6 Native 5-Minute Analysis

To rule out the possibility that 15-minute aggregation obscures useful spatial information, we repeat the linear oracle at the native 5-minute resolution with corresponding forecast horizons (Table 10). If graph propagation were the dominant source of spatial information, the spatial gain would be expected to peak at short horizons and decay thereafter. Instead, the gain remains nearly constant (0.29–0.39 MAE) from 5 to 60 minutes. At the shared 15-minute horizon, temporal aggregation reduces the gain by only 0.11 MAE (0.29 $\rightarrow$ 0.18), indicating that 15-minute aggregation is not masking a strong propagation signal.

Table 10: Linear spatial gain (own $\rightarrow$ own+neighbours, MAE) by forecast horizon at native 5-minute and benchmark 15-minute resolutions.

resolution	5m	15m	30m	45m	60m	90m	180m
5-min	+0.36	+0.29	+0.33	–	+0.39	–	–
15-min	–	+0.18	–	+0.43	–	+0.46	+0.56

C Conditional Predictability

This section quantifies how much predictive information remains in neighbouring sensors after the target sensor's own history has been fully exploited, supporting the analysis in Section 4.3. To isolate this residual information, each sensor is first residualized using the full recipe, and the remaining residual is predicted from the N nearest neighbours' recent anomaly histories. Models are fitted on the earlier held-out test windows and evaluated on the later ones. To avoid relying on a single model class, we evaluate two complementary probes: a closed-form linear ridge model (pooled across sensors, with λ selected on the validation split and a zero model always available) and a nonlinear MLP (256 hidden units, early-stopped on validation). Their results measure predictive utility within these model classes and do not bound the residual spatial information in the data. Confidence intervals are estimated using day-block bootstraps over the test period.

C.1 Regime Definitions

We stratify evaluated cells (w, s, h) (window w, sensor s, and horizon h) using two observable statistics of the target sensor's own signal, both defined as the mean absolute deviation from its historical-average (HA) table. The *target anomaly*

$$a^{\text{tgt}}_{w,s} = \frac{1}{H} \sum_{h=1}^{H} |y_{w,s,h} - \text{HA}_{w,s,h}| \tag{2}$$

measures how far the realized future is from climatology, while the *context anomaly*

$$a^{\text{ctx}}_{w,s} = \frac{1}{12} \sum_{i=1}^{12} |x_{w,s,-i} - \text{HA}| \tag{3}$$

measures the deviation of the previous 12 observations. Top-decile target cells are those whose a^{tgt} lies in the top 10% of the test distribution. Sudden-onset cells are the subset whose context remains quiet, i.e., a^{ctx} is at or below its median. The complementary subset, in-progress cells, consists of top-decile target cells with an already anomalous context.

Intuitively, sudden-onset cells correspond to cases where the future departs sharply from the historical pattern while the recent history still appears normal. A purely temporal model therefore has no

in-context warning, and information from other sensors is one plausible source of advance warning. This is the most challenging regime for a graph-free forecaster and the most favourable regime for a graph model, motivating its separate analysis. It is also the regime that the context gate of Section 5.3 cannot address: the gate reacts to anomalies that are already visible in the observed context, whereas sudden-onset cells have a quiet context by definition. Sudden-onset cells account for only a small fraction of the evaluation set (2.5% on SD and 1.5% on both GBA and GLA), so even substantial improvements within this regime contribute only a few hundredths of an MAE point to the overall average.

C.2 Results

Table 11 shows that the linear predictor extracts almost no residual spatial signal: its gains are at most 0.13 MAE ($R^2 = 0.001$), statistically distinguishable from zero but small in magnitude. The nonlinear predictor reduces the residual by 0.35 MAE on the full evaluation set, but the day-block confidence interval $[-0.08, +0.69]$ includes zero, so this aggregate gain is not statistically resolved: the data neither establish an improvement nor show that the gain is zero or practically negligible. Statistically supported MLP gains are observed in the top-decile target cells (2.48 MAE, day-block CI [+0.53, +3.49]) and in their sudden-onset subset (about 2.8 MAE, day-block CI [+1.9, +3.5]), which accounts for only about 2% of all cells. We therefore conclude only that, within the two probes tested, the nonlinear probe found statistically supported spatial gains in the most anomalous cells, including the sudden-onset regime that the context gate cannot address, while its aggregate gain was not resolved; this is not evidence that spatial information is absent elsewhere.

Table 11: Out-of-sample reduction in the recipe's residual from the $k = 8$ nearest neighbours (robust to $k = 16$ and seed).

	ridge (linear)		MLP (nonlinear)	
regime (SD, out-of-sample)	ΔMAE	R^2	ΔMAE	R^2
all cells	+0.02	0.001	+0.35 [−0.08, +0.69]	0.056
top-decile target	+0.12	0.001	+2.48	0.084
sudden-onset	+0.13	0.001	+2.76	0.044

C.3 Replication across Flow Regions

The same pattern holds on GBA and GLA (Table 12). The linear predictor consistently extracts almost no residual signal, and the aggregate MLP reduction is not statistically resolved on any region: its day-block confidence interval includes zero in every case, which leaves the aggregate effect uncertain rather than establishing that it is zero. Statistically supported gains appear in sudden-onset cells, increasing modestly with network size (2.8, 3.1, and 3.8 MAE on SD, GBA, and GLA, respectively). Because sudden-onset cells represent only about 2% of the evaluation set, the gains within these cells alone contribute only a few hundredths of a point to the overall MAE across all regions.

Table 12: Conditional predictability across flow regions (ΔMAE on the recipe residual, day-block CIs). All MLP all-cells intervals include zero ("CI $\ni$ 0"): these aggregate gains are statistically unresolved, not evidence of zero gain.

	ridge	MLP (nonlinear)	
region	sudden-onset	sudden-onset (95% CI)	all cells (95% CI)
SD	+0.13	+2.76 [+1.93, +3.53]	+0.35 [−0.08, +0.69]
GBA	+0.05	+3.13 [+1.70, +4.14]	−0.11 (CI $\ni$ 0)
GLA	+0.28	+3.78 [+1.71, +5.25]	+0.27 (CI $\ni$ 0)

D RECIPE DETAILS

D.1 ADDITIONAL DETAILS OF THE FORECASTING RECIPE

The recipe adapts a frozen TimesFM 2.5 backbone (200M parameters) with three lightweight modules, supporting the recipe construction described in Section 5.1. First, rank-8 LoRA is applied to all linear projections in the last 6 of the 20 transformer layers (≈0.6M trainable parameters). Second, zero-initialized time-of-day (96 slots) and day-of-week (7 slots) embeddings are added to the input tokens (≈0.13M parameters). Third, a per-sensor historical-average (HA) prior is constructed as a 7×96 (day-of-week $\times$ time-of-day) lookup table of training-split means. The model predicts the anomaly (flow $-$ HA), and the corresponding HA value is added back to obtain the final forecast. The backbone takes $T{=}512$ historical steps (5.3 days) from a single sensor and predicts all $H{=}12$ forecast horizons in one non-autoregressive pass.

Table 13: Incremental construction of the temporal-only recipe on SD using LoRA, calendar cues, and a per-sensor HA prior. The final recipe achieves an average MAE of 17.43, outperforming the strongest graph-based baseline (Graph WaveNet, 18.18) without spatial information.

Model (MAE)	45 min	90 min	180 min	average
Frozen TimesFM (zero-shot)	18.55	24.01	29.99	24.18
+ LoRA	15.72	18.26	21.05	18.34
+ Calendar	15.50	17.97	20.78	18.08
+ HA (Recipe)	**14.84**	**17.40**	**20.05**	**17.43**

Context-length sweep. Table 14 shows that the HA prior consistently improves performance across all context lengths, rather than only at the full 512-step budget. Moreover, adding the HA prior allows substantially shorter contexts to match or exceed much longer HA-free ones: with HA, a 2-day context (17.78) already outperforms a 5-day context without HA (18.08).

Table 14: HA improves the recipe at every context length; "raw control" repeats the same short context without the HA table, isolating the table's value (-3.3 MAE at cc=96).

config (SD)	45m	90m	180m	avg
+HA, cc=48 (12h)	16.25	20.17	25.35	20.59
+HA, cc=96 (1 day)	15.41	18.17	21.11	18.23
+HA, cc=192 (2 days)	15.15	17.75	20.43	17.78
+HA, cc=288 (3 days)	15.06	17.70	20.48	17.74
+HA, cc=384 (4 days)	14.95	17.59	20.33	17.62
+HA, cc=512 (recipe)	**14.83**	**17.40**	**20.05**	**17.43**
no-HA, cc=512 (base)	15.50	17.97	20.78	18.08
raw control, cc=96	17.77	21.30	25.51	21.53
raw control, cc=192	16.58	19.71	23.30	19.86

Calendar/HA redundancy. With the HA prior in place, the contribution of the calendar embeddings becomes much smaller but remains consistently positive. LoRA+HA without calendar achieves an average MAE of 17.56 (14.91/17.54/20.22), compared with 17.43 when calendar embeddings are included. Both variants already outperform Graph WaveNet, indicating that the calendar embeddings are no longer the primary source of improvement once HA is available, but they remain essentially free and provide a small, consistent gain.

D.2 GATE MECHANISM

The context gate switches between the recipe (+HA) and its no-HA base using only information available at forecast time. The decision variable is the *context deviation*, defined as the mean absolute difference between the last 12 observed steps and their corresponding HA values. If the deviation exceeds the 90th percentile of its training-split distribution, the no-HA prediction is used; otherwise the +HA prediction is retained. The threshold is fitted once on the training split (no test-set tuning)

and is insensitive to its exact value: p80/p90/p95 yield 16.94/16.87/16.87 MAE on SD, while the gate switches on approximately 18% of cells.

Table 15 explains when the gate helps. The HA prior consistently improves the periodic bulk (deciles 1–8) but degrades the most anomalous decile. Splitting this top decile into *in-progress* and *sudden-onset* cases further isolates the remaining failure mode: the gate recovers most of the error on in-progress anomalies (45.41 to 38.48 MAE), which are already visible in the observed context, whereas sudden-onset events remain at 75.49 MAE with and without the gate. Because their observed context is quiet, the context-deviation signal gives no warning, and these events remain an unresolved limitation of the recipe.

Table 15: MAE by |target−HA| decile. The gate recovers most of the degradation introduced by the HA prior, leaving only the rare sudden-onset cases unresolved.

stratum (SD)	cells	base	+HA	gate	oracle
deciles 1–8	80%	5.3–23.3	3.7–21.8	≈+HA	–
decile 9	10%	27.91	28.16	27.99	23.84
decile 10	10%	38.57	48.71	**42.54**	34.63
in-progress	8.9%	36.57	45.41	**38.48**	32.56
sudden-onset	1.1%	54.49	75.49	75.49	51.24
all	100%	18.09	17.45	**16.87**	14.67

To test whether the gate generalizes beyond SD, we apply exactly the same gate to the other three LargeST subsets without any retraining, using only the +HA and -HA prediction outputs from each region. It consistently outperforms both parent models on every dataset, achieving 16.87/18.40/17.85/16.60 MAE on SD/GBA/GLA/CA, compared with 17.43/19.02/18.39/17.09 for the ungated recipe and 18.18/21.31/20.75/21.34 for the strongest graph baseline.

D.3 Control Experiment: Observation Allocation vs. Model Scale

This control experiment isolates whether the performance gain arises from observation allocation rather than model scale.

Table 16: Control experiment isolating the effect of observation allocation from model scale.

condition	ctx	HA	45m	90m	180m	avg
mixer	512	on	15.03	17.31	19.60	**17.31**
mixer	512	off	15.50	17.74	20.55	17.93
mixer	96	on	15.19	17.50	20.01	17.57
mixer	12	on	16.42	20.63	26.54	21.20
recipe +HA (200M frozen + 0.75M)	512	on	14.83	17.40	20.05	17.43
base (no HA)	512	off	15.50	17.97	20.78	18.08
Graph WaveNet (published)	12	–	15.24	17.74	21.56	18.18
DCRNN (published)	12	–	17.14	20.99	26.99	21.71

Table 16 reports the results, training the nonlinear mixer from scratch under the same observation allocations as the recipe. At matched context (T=512), the 1.15M-parameter mixer matches the adapted 200M-parameter foundation model (17.31 vs. 17.43 MAE) and, even without the HA prior, outperforms the strongest published graph baseline (17.93 vs. 18.18). Restricting the mixer to the STGNNs' standard 12-step temporal context degrades performance to 21.20 MAE, closely matching DCRNN at 21.71. These results suggest that the improvement primarily arises from observation allocation rather than model scale or graph propagation.

Table 17: Multi-seed results for the recipe and the nonlinear mixer on SD.

model	per-seed MAE	mean ± s.d.
mixer (5 seeds)	17.38/17.33/17.40/17.37/17.35	17.37 ± 0.03
recipe (4 seeds)	17.43/17.46/17.51/17.55	17.49 ± 0.05

E VALIDATION

E.1 MULTI-SEED

Table 17 reports the multi-seed results of the recipe and the nonlinear mixer on SD. Every seed of both models outperforms the strongest published Graph WaveNet baseline (18.18 MAE). The mixer achieves slightly lower error and is approximately twice as seed-stable as the recipe ($\sigma = 0.03$ vs. 0.05), supporting rather than weakening the "allocation, not scale" claim. The single-seed headline results reported in the main paper (17.31 for the mixer and 17.43 for the recipe) lie within these distributions; the reported recipe result happens to be its best-performing seed.

E.2 CALIBRATION

We perform three calibration checks to bound potential sources of systematic error in the evaluation pipeline.

Missing-value handling. Refitting the linear oracle using only non-imputed training rows changes its score by 0.000 MAE, indicating no measurable bias from missing-value handling.

Cross-hardware reproducibility. The same configuration reproduces within 0.03 MAE across hardware (17.33 vs. 17.31 for $(T{=}512, N{=}0)$).

Evaluation protocol. The dense evaluation is not favorable. Table 18 re-scores the saved SD adapters on every test window (stride 1; 6,484 windows) rather than the dense stride-8 subset (811 windows) used throughout the paper. Every model improves by approximately 0.1 MAE (base 17.97 vs. 18.09; recipe 17.35 vs. 17.45; gate 16.78 vs. 16.87), while the performance margins remain essentially unchanged. The reported stride-8 subset is therefore representative rather than flattering.

Table 18: Performance under stride-1 and stride-8 evaluation on SD.

model	stride-1 (6,484 win)	stride-8 (811 win)	Δ
base (no HA)	17.97	18.09	−0.12
recipe +HA	17.35	17.45	−0.10
context gate	16.78	16.87	−0.09

E.3 ADDITIONAL METRICS

Table 19: Average RMSE and WAPE for the base model, recipe, and gated recipe across datasets.

	avg RMSE			avg WAPE (%)		
dataset	base	recipe	gate	base	recipe	gate
SD	30.11	30.70	**29.55**	7.50	7.24	**7.00**
GBA	33.00	33.90	**32.89**	7.95	7.95	**7.69**
GLA	31.38	31.91	**30.86**	6.78	6.65	**6.45**
CA	29.93	30.31	**29.42**	7.39	7.20	**6.99**
Seattle	7.02	6.83	**6.78**	6.73	6.79	**6.61**

Table 19 reports the average RMSE and WAPE across all datasets. Consistent with the decile analysis in Appendix D.2, the HA prior alone increases RMSE on the four traffic-flow datasets despite

improving MAE, reflecting the larger penalty that RMSE assigns to the rare tail failures. The context gate restores the lowest RMSE on every dataset, including Seattle, by recovering these failures while preserving the bulk improvement. WAPE follows the same ordering as MAE throughout.

E.4 Uncertainty (Autocorrelation-Aware Bootstrap)

Forecast errors are correlated across nearby windows and sensors, so uncertainty is estimated using a day-block bootstrap (2,000 resamples), with the more conservative interval between day-block and sensor-resampling bootstrap reported. Table 20 shows that the gate consistently improves over the ungated recipe under the conservative day-block bootstrap across all datasets, with every 95% confidence interval excluding zero. By contrast, the average gain of the HA prior alone is not consistently resolved under the same analysis, indicating that day-to-day variability dominates its average effect. This is consistent with the gate acting as a per-window regime switch rather than a fixed global correction.

Table 20: Bootstrap confidence intervals for the gate gain over the ungated recipe.

dataset	recipe−gate	day-block 95% CI	sensor 95% CI
SD	+0.585	[+0.11, +1.18]	[+0.54, +0.64]
GBA	+0.620	[+0.34, +0.95]	[+0.58, +0.66]
GLA	+0.544	[+0.24, +0.94]	[+0.51, +0.58]
CA	+0.494	[+0.23, +0.83]	[+0.48, +0.52]
Seattle	+0.101	[+0.03, +0.19]	[+0.09, +0.11]

F Further Analysis

F.1 Gate Alternatives

We compare the proposed zero-parameter hard gate (Appendix D.2) with several increasingly flexible alternatives, all fitted on the first half of the test period and evaluated on the held-out second half. Re-tuning the threshold never improves over the a priori training-split p90 threshold out of sample (on SD, the re-tuned model selects p99 and performs worse). Allowing a constant blend of the two parent models already outperforms the hard gate. A two-parameter logistic blend based on the context-deviation statistic performs better still, reducing MAE from 18.24 to 17.54 on SD, 19.79 to 18.92 on GBA, 18.96 to 18.25 on GLA, and 17.75 to 17.06 on CA. A per-cell oracle blend provides a further 1.5–2.0 MAE reduction, indicating that additional headroom remains. Despite these gains, we use the hard gate as the main method because it introduces no fitted parameters, requires no held-out calibration data, and generalizes unchanged across all datasets. The learned blends are therefore reported only as an estimate of the remaining headroom rather than as competing methods.

F.2 Cross-Sensor Transfer

The recipe contains no sensor- or city-specific trainable parameters: LoRA and the calendar embeddings are shared globally, while the historical-average prior is computed independently for each sensor. This design should therefore generalize naturally to unseen sensors. Table 21 evaluates this by splitting SD's 716 sensors into two disjoint halves, training on subset A and evaluating on both A and the held-out subset B. Although the held-out sensors are intrinsically more difficult (floor MAE 18.70 on B versus 18.25 on A), the recipe achieves nearly identical improvements over the respective floors (0.94 versus 0.91 MAE), indicating essentially no transfer loss.

G Interpretability

This section investigates why long temporal context is effective by probing what information the frozen backbone actually uses. We apply three complementary analyses to the 16 context patches of a $P{=}32$, $T{=}512$ input. Because all analyses operate at the patch level, we first identify which patches correspond to the same time of day in previous days. One day corresponds to 96

Table 21: Cross-sensor transfer on San Diego using disjoint train/test sensor splits. The floor denotes the model without LoRA or calendar adaptation.

setting	45m	90m	180m	avg	Δ vs. floor
floor A	15.19	18.22	21.33	18.25	–
floor B	15.58	18.64	21.87	18.70	–
recipe A (in-dist.)	14.71	17.33	19.98	17.34	-0.91
recipe B (transfer)	15.09	17.71	20.47	17.76	-0.94

Table 22: Cross-city transfer to the Bay Area (GBA, 2,352 sensors). Our SD-trained recipe (SD→GBA) is within 0.12 MAE of direct GBA training and outperforms the strongest GBA-trained baseline (D^{2}STGNN) at every horizon.

MAE on GBA	45 min	90 min	180 min	average
Ours, SD→GBA	**16.43**	**19.32**	**21.68**	**19.14**
Ours (GBA)	16.37	19.20	21.49	19.02
D^{2}STGNN (GBA)	17.54	20.92	25.48	21.31

time steps (three patches), so the daily analogue of the forecast target appears at patch indices $\lfloor (512 - 96d)/32 \rfloor \in \{13, 10, 7, 4, 1\}$, one patch more recent than the naive origin-aligned indices $\{12, 9, 6, 3\}$. Using the origin-aligned indices initially obscured the daily structure.

G.1 Attention

This section investigates whether the frozen backbone explicitly attends to daily-periodic information. We capture the exact attention weights by instrumenting the model's attention function, verified against a forecast-naive reimplementation to within 10^{-6}. Attention is dominated by recency: the most recent 24 hours receive about 49% of the total attention mass, while the remaining five days receive diffuse attention. Neither the zero-shot nor the LoRA-adapted model exhibits a distinct attention peak at the daily-phase patches $\{13, 10, 7, 4, 1\}$. We do not use attention rollout because, under the causal mask, patch 0 becomes an absorbing state that accumulates rollout mass regardless of the input content. The resulting attribution is therefore an artifact of the rollout procedure rather than a meaningful signal.

G.2 Permutation

Permutation importance is used to measure each context patch's functional contribution to the forecast, beyond what can be inferred from attention weights. Permutation importance (each patch is permuted across the 716 sensors; 40 origins $\times$ 5 permutations; mean $\pm$ SEM reported) shows that the frozen backbone relies primarily on recent observations, while also exhibiting a clear daily-periodic preference. The most recent patch (patch 15) has by far the largest importance (+73 $\pm$ 6), approximately four times that of a typical non-daily patch. In addition, each daily-phase patch $\{13, 10, 7, 4, 1\}$ is consistently 2–6 times more important than its neighbouring non-daily patches in both the zero-shot and LoRA-adapted models. Among them, patch 13 (one day before the forecast target) ranks second overall (+22 $\pm$ 3), behind only the most recent patch.

G.3 Shapley

Shapley values are used to attribute the forecast to individual context patches while accounting for interactions between them. Monte-Carlo Shapley values (same permute-across-sensors baseline; 15 origins $\times$ 8 orderings; additivity verified) reinforce the same conclusion. Importance is concentrated almost entirely on the daily-phase patches, while most non-daily patches have values close to zero or even negative. In particular, patches 12, 9, and 6 have significantly negative Shapley values, indicating that the model specifically prefers the exact daily-phase patch rather than nearby observations shifted by eight hours. Overall, the five daily-phase patches account for 42% of the total Shapley value despite representing only 5 of the 16 input patches (40% after LoRA adaptation).

Table 23: Per-patch Shapley value (MAE reduction credited to each patch; SD, mean over 15 origins).

patch (role)	zero-shot	adapted
15 (now)	+58.20	+61.53
14	+3.40	+2.01
13 (1 day)	**+23.23**	**+20.35**
12	−3.49	−4.39
11	−0.76	+0.25
10 (2 days)	**+5.89**	**+5.80**
9	−2.20	−1.87
8	−0.31	+0.35
7 (3 days)	**+3.19**	**+3.89**
6	−1.27	−0.91
5	−0.10	+0.35
4 (4 days)	**+4.85**	**+4.77**
3	−0.79	−0.64
2	+0.40	+0.33
1 (5 days)	**+1.90**	**+2.85**
0	+0.12	−0.35
daily-phase share	42.3%	39.9%

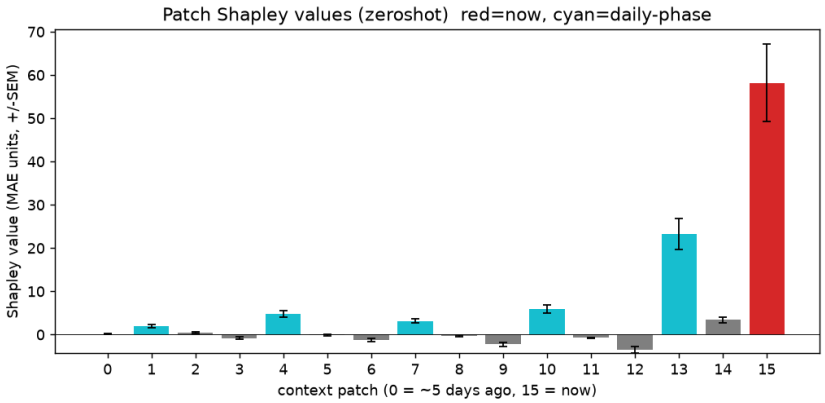


Figure 5: Shapley attribution over the 16 context patches (zero-shot backbone). Spikes fall exactly at the daily-analogue patches (1–5 days back); intervening patches contribute nothing.

G.4 Attention versus Functional Importance

Attention weights show no distinct daily-phase peak, whereas both permutation importance and Shapley values indicate that the model relies strongly on the daily-phase patches. This suggests that daily information is carried primarily through the value/content pathway rather than the attention weights themselves, explaining why attention alone fails to reveal the daily structure.

Whether LoRA-adapted calendar cues reduce this reliance remains unclear. The total importance assigned to the daily-phase patches decreases from 41.7 to 35.8 after LoRA adaptation, but the difference is not statistically significant ($+6.0 \pm 4.8$, 1.2σ). Shapley analysis shows the same trend (patch-13 importance decreases from 23.2 to 20.3, $< 1\sigma$). The current evidence therefore suggests that intrinsic daily reliance and learned calendar cues coexist rather than substitute for one another.

H Physical Neighbourhoods on a High-Resolution Corridor

Data and regimes. FT-AED (Coursey et al., 2024) covers an 18-mile stretch of Interstate 24 towards Nashville with 196 lane-level radar sensors (49 milemarkers $\times$ 4 lanes) recording speed, volume, and occupancy every 30 seconds on weekday mornings. The development block uses 20

weekdays of October 2023 (14 for training, 6 for testing). The confirmation block uses 29 weekdays from January to June 2024, with models trained on July to December 2023; all contrasts and their expected signs were fixed from the development results before any model was trained on this block. Regimes are defined from training data only: *recurrent* when the historical-average speed at that time of day is already well below free-flow speed, *near capacity* when occupancy exceeds the training near-capacity threshold while speed has started to drop, and *free flow* otherwise.

Models. All models share a frozen temporal forecaster T that reads only the target's own history, and add a zero-initialized spatial residual on top of it, so that each spatial model starts from T. The *local window* averages the histories of the same-lane sensors up to three milemarkers upstream and downstream, using lane and milemarker order only. The graph-based model G reads the same sources through a supplied adjacency graph. The capacity-matched control R_{own} gives T a residual of the same size that reads only the target's own history. Candidate-set variants retrain the window model with the immediate upstream sensor removed, with the upstream sensor or all six sensors replaced by statistically matched sensors outside the propagation range, or with six random sensors. Results are averaged over five seeds, and intervals are crossed day-by-seed bootstrap 95% CIs.

Table 24: FT-AED confirmation block (29 weekdays, five seeds): change in MAE (mph) at a 4-minute horizon relative to the local window, by regime. Positive values mean the variant is worse.

Candidate set	Free flow	Near capacity	Recurrent
Upstream sensor removed	$+0.011$	$+0.259$	$+0.174$
Upstream replaced by matched distant sensor	$+0.005$	$+0.320$	$+0.178$
All six replaced by matched distant sensors	$+0.103$	$+1.207$	$+1.267$
Six random sensors	$+0.101$	$+1.315$	$+1.299$
Downstream sensors only	$+0.086$	$+1.345$	$+1.421$

Results. On the confirmation block the local window improves on the temporal forecaster by 0.56 mph and on the capacity-matched control by 0.50 mph (95% CI $[0.45, 0.55]$), in favour of the window on all 29 days. Under full corruption of the supplied graph, G loses 0.40 mph on average across four corruption families, whereas the window model is unaffected because it never reads the graph. Table 24 reports the regime-wise cost of changing the candidate set: every substitution is nearly free in free flow but costly near capacity and in recurrent congestion, and all nine regime-wise contrasts shown for the upstream, matched-distant, and random sets were confirmed with the predicted sign. Overall, 28 of the 29 directional predictions fixed in advance were confirmed on this block. The development block shows the same pattern, with a spatial gain about six times larger in congestion than in free flow.

Contrast with LargeST. Applying the same design to LargeST San Diego (15-minute flow, 30-minute horizon), the window again improves on the capacity-matched control, by 0.20 vehicles per 15 minutes, but proximity no longer matters: eight matched distant sensors give slightly lower error than the eight nearest (by 0.16), and eight random sensors by 0.13. This agrees with the random-sensor control of Section 4.2. The two settings differ in network, variable, resolution, and horizon at the same time, so we read the contrast as a boundary condition rather than as a controlled effect of resolution alone.