HODOS: A TRAFFIC FORECASTING FOUNDATION MODEL WITH IN-CONTEXT SPATIAL MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate traffic forecasting is essential for intelligent transportation. Since traffic conditions at one sensor are shaped by those nearby, spatial modeling lies at the core of the task. Prior methods usually bind spatial dependencies to predefined graphs, adaptive adjacency matrices, or node embeddings, which limits their transferability to new regions. Recent spatiotemporal foundation models (STFMs) pursue zero-shot transfer through pretraining on large-scale spatiotemporal data, yet their spatial modeling still hinges on target-domain spatial priors such as sensor coordinates or connectivity graphs and fails when these are missing or inaccessible. Cross-sensor self-attention needs none of these, but it compares sensors with shared projections and has no adaptive control over how much spatial information each sensor should receive. To fill this gap, we propose In-Context Spatial Modeling (ICSM), which models spatial interactions from the observed traffic context alone. It augments spatial self-attention with context-conditioned low-rank query–key modulation and a context gate that controls the strength of spatial update. Building upon this, we develop **Hodos**, a traffic forecasting foundation model, and construct **JaRTI**, a pretraining corpus of about 31 billion real traffic observations from 51 Japanese regions over eight years. Pretrained on JaRTI and evaluated zero-shot on six widely used US traffic benchmarks, Hodos reduces MAE and RMSE by 26.9% and 26.2% without spatial priors on average over the strongest baselines. Visualizations reveal transferable cross-region spatial patterns and show how Hodos responds as congestion emerges and propagates. Code and a subset of the dataset are available at `https://anonymous.4open.science/r/Hodos-C85E`.

1 INTRODUCTION

Accurate traffic forecasting supports traffic management, route planning, and incident response. Learning spatiotemporal patterns to anticipate changes in traffic flow and speed has long been a central focus of research (Li et al., 2018; Yu et al., 2018; Wu et al., 2019; Guo et al., 2019; Wu et al., 2020; Deng et al., 2021; Jiang et al., 2021; Shao et al., 2022b; Deng et al., 2023b; Li et al., 2023; Jiang et al., 2023a;b; Deng et al., 2023a; Gao et al., 2024; Lee & Ko, 2024; Deng et al., 2024; Su et al., 2025; Gao et al., 2025; Fang et al., 2025; Zou et al., 2026; Yu et al., 2026; Fan et al., 2026).

Since observations at one location usually inform predictions at others, spatial modeling is central to traffic forecasting task. Existing end-to-end models capture these dependencies through predefined graphs (Li et al., 2018; Yu et al., 2018), adaptive adjacency matrices (Wu et al., 2019; Bai et al., 2020), or sensor embeddings (Shao et al., 2022a; Liu et al., 2023a). However, their dependence on network structure or sensor-specific representations limits their transferability. To learn general traffic patterns, spatiotemporal foundation models (STFMs) pretrain backbones on diverse datasets, learning knowledge across regions to be applied to new datasets. But their spatial components still require target graphs, grids, or coordinates for zero-shot inference (Li et al., 2024; Yuan et al., 2024; Chen et al., 2026), or target-network tuning of spatial adapters (Zhong et al., 2025), as summarized in Table 1. These requirements constrain spatial transfer when target-domain spatial priors are unavailable or adaptation is impractical, motivating spatial modeling without these metadata.

To solve this issue, vanilla cross-series attention provides a natural starting point by computing input-dependent interactions across series without predefined spatial structure. A few general time-

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.

Table 1: Qualitative comparison with representative spatiotemporal foundation models. ✓: satisfied; △: partially satisfied; ✗: not satisfied. Real/Synth denote real/synthetic data.

Aspect	Joint STFMs			Factorized STFMs		Hodos (Ours)
	OpenCity	UniST	UrbanFM	FactoST	FactoST-v2	
Pretraining Scale	∼151M Real	∼100M Real	>1B Real	∼13M Real	∼0.5B Real + 11B Synth	**∼31B Real**
Probabilistic Forecasting	✗	✗	✗	✗	✓	✓
Spatial Relation Source	Graph	Grid	Coordinates	Node embed.	Node embed.	**Context only**
Zero-shot Forecasting without Target-domain Spatial Priors	✗	✗	✗	△[a]	△[a]	✓

[a] Only the pretrained temporal backbone is used; the spatial adapters require second-stage target-network tuning (Zhong et al., 2025; 2026).

series foundation models such as Moirai (Woo et al., 2024) and Chronos-2 (Ansari et al., 2025) follow this route through any-variate and group attention, respectively. In traffic forecasting, the cross-sensor dependence is dynamic and local: which neighbors matter and how much a sensor should rely on them depend on its current state. Under light traffic, small disturbances are carried downstream with the vehicles and fade out, whereas near capacity a local slowdown grows into a queue that propagates upstream against the flow. For a sensor at the tail of a forming queue, the most informative neighbor is the downstream sensor that has already slowed down, even though the two observations are different. These observations lead to our key challenge: *how to explicitly generate sensor-specific matching adjustments and spatial update scales from the observed context window without target spatial priors or second-stage tuning.*

To address this challenge, we propose Hodos, a traffic forecasting foundation model that derives transferable spatial interactions from the input context. Its core ICSM module explicitly parameterizes the influence of recent traffic state on spatial matching and update strength through two complementary designs. First, *low-rank context modulation* maps each sensor's window-level context summary to query–key adjustments. Shared low-rank bases provide reusable matching directions, while context-dependent features adjust their contributions for each sensor. Second, a *context gate* uses the same summary to scale the spatial residual, controlling how strongly the aggregated information updates each sensor's representation. Specifically, Hodos first embeds traffic time series into patches with calendar features and appends future tokens. The resulting sequence passes through alternating RoPE-based temporal attention and ICSM blocks, after which the future token representations are decoded into quantile forecasts. To pretrain Hodos across diverse traffic conditions, we construct JaRTI, which contains approximately 31 billion real traffic observations from 51 Japanese regions between 2018 and 2025.

Our contributions are summarized as follows:

- We propose Hodos, a traffic forecasting foundation model that learns transferable spatial interactions from the observed context. Its ICSM operator adapts cross-sensor matching and the contribution of spatial information through low-rank context modulation and a context gate.
- We construct JaRTI, a large-scale pretraining corpus containing approximately 31 billion real traffic observations across 51 Japanese regions over eight years, providing diverse regional and temporal patterns for traffic forecasting pretraining.
- We validate Hodos on six US traffic flow and speed benchmarks under zero-shot transfer, achieving average MAE and RMSE reductions of 26.9% and 26.2% relative to the strongest baseline in each setting. Qualitative analyses further illustrate how the model adjusts spatial interactions to changing traffic conditions.

2 Related Work

2.1 End-to-End Traffic Forecasting Models

Early univariate statistical methods, such as ARIMA (Box & Jenkins, 1970), forecast each series independently and do not capture cross-sensor dependencies. To capture these dependencies, spa-

tiotemporal graph neural networks (STGNNs) model interactions over sensor graphs. DCRNN (Li et al., 2018) and STGCN (Yu et al., 2018) combine temporal modeling with diffusion or convolution over predefined sensor graphs. To reduce reliance on predefined connectivity, Graph WaveNet (Wu et al., 2019) learns adaptive adjacency from node embeddings, while AGCRN (Bai et al., 2020), HimNet (Dong et al., 2024), and MetaDG (Zou et al., 2026) further generate node-specific parameters. Beyond graph construction, STID (Shao et al., 2022a) and STAEformer (Liu et al., 2023a) encode sensor information through identity embeddings and adaptive embeddings, respectively. These approaches have achieved great success by capturing spatial dependencies with network structures or learned embeddings associated with specified sensors. *Nevertheless, their dependence on dataset-specific graphs or embeddings limits the direct reuse of spatial modeling in unseen regions.*

2.2 Foundation Models for Traffic Forecasting

To reuse forecasting knowledge across regions, recent spatiotemporal foundation models (STFMs) pretrain shared backbones on diverse datasets. However, their spatial components still require target-specific information or adaptation. OpenCity (Li et al., 2024), UniST (Yuan et al., 2024), and UrbanFM (Chen et al., 2026) incorporate spatial structure through target graphs, grids, and coordinates, respectively. FactoST (Zhong et al., 2025) and FactoST-v2 (Zhong et al., 2026) separate spatial-agnostic backbones from spatial adapters fitted to target networks. General time-series foundation models (TSFMs) provide a complementary route by learning from heterogeneous series without requiring target spatial metadata. TimesFM (Das et al., 2024) and Chronos (Ansari et al., 2024) forecast individual series, while Moirai (Woo et al., 2024) and Chronos-2 (Ansari et al., 2025) enable cross-series interactions through any-variate and group attention. *We design a cross-sensor attention operation by introducing a window-level conditioning path that maps each sensor's context summary to low-rank query–key adjustments and an adaptive spatial context gate. This design explicitly parameterizes the influence of recent traffic state on both matching and residual scaling.*

3 Problem Formulation

3.1 Traffic Forecasting

Traffic forecasting is a multivariate time series forecasting task. At each time step t, the observations from N sensors are represented by $\boldsymbol{X}_t \in \mathbb{R}^{N\times C}$, where C is the number of features, such as traffic flow or speed. In our study, C is equal to 1 in all used benchmarks. Given the historical T_{in} observations $\boldsymbol{X} = \boldsymbol{X}_{t-T_{\text{in}}+1:t} \in \mathbb{R}^{T_{\text{in}}\times N\times C}$, the goal is to predict the next T_{out} steps $\boldsymbol{Y} = \boldsymbol{X}_{t+1:t+T_{\text{out}}}$. Formally, a forecasting model parameterized by θ learns the mapping

$$\hat{\boldsymbol{Y}} = f_\theta(\boldsymbol{X}), \qquad f_\theta : \mathbb{R}^{T_{\text{in}}\times N\times C} \to \mathbb{R}^{T_{\text{out}}\times N\times C}. \tag{1}$$

In the zero-shot setting, the model is pre-trained on a source dataset $\mathcal{D}_{\text{src}}$ and then applied to a target dataset $\mathcal{D}_{\text{tgt}}$ from a different region. The two datasets should be disjoint, such that $\mathcal{D}_{\text{src}} \cap \mathcal{D}_{\text{tgt}} = \varnothing$.

4 Methodology

4.1 Overview

As illustrated in Figure 1, Hodos first normalizes traffic observations and combines patch embeddings with calendar information. Future tokens are then appended. This joint sequence passes through L layers of RoPE-based temporal attention and ICSM. The former captures temporal evolution, while the latter conditions cross-sensor interactions on the current traffic state. Finally, a quantile head decodes the future tokens into probabilistic forecasting.

4.2 Input Tokenization

To accommodate regional differences in traffic magnitudes, we first normalize observed values as $\tilde{X} = (X - \mu)/s$, where μ and s are the mean and stabilized standard deviation computed jointly over all time steps and sensors within the input window. Following the normalize–restore principle of reversible instance normalization (Kim et al., 2022), these statistics also restore forecasts to their original units. Sharing them across sensors preserves relative traffic levels within the window.

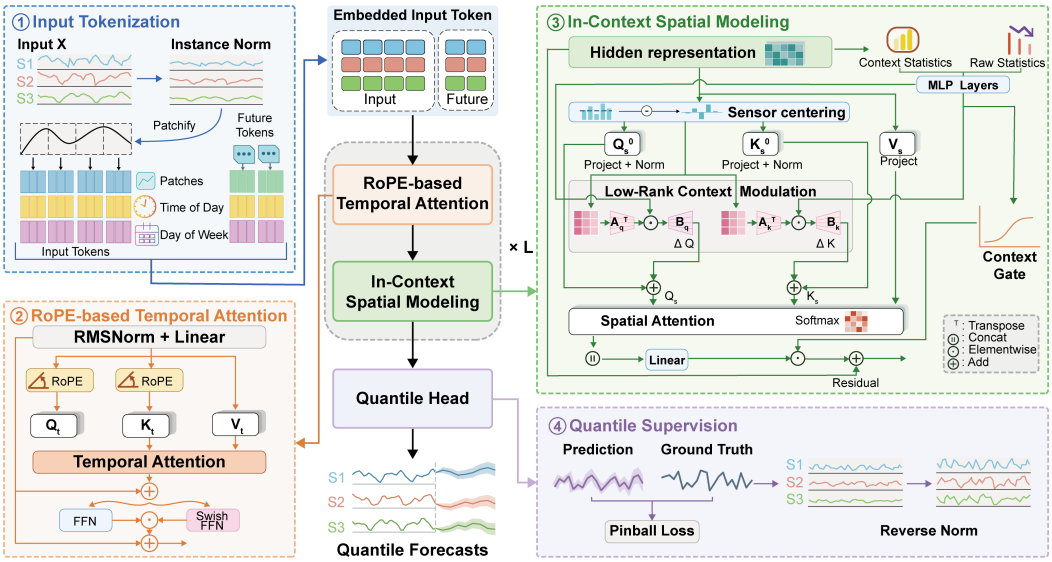


Figure 1: Overview of Hodos. Normalized inputs are patchified, concatenated with calendar embeddings, and future tokens are appended. RoPE-based temporal attention and ICSM alternate for L layers; ICSM uses the observed context to modulate query–key matching and gate the spatial residual. Quantile outputs are supervised by the pinball loss before reverse normalization.

After normalization, the T_{in} input steps are split into $P_{\text{in}} = T_{\text{in}}/T_p$ non-overlapping patches of length T_p, each linearly projected into a patch embedding (Nie et al., 2023; Dosovitskiy et al., 2021). Each patch embedding is then concatenated with learned time-of-day and day-of-week embeddings. To predict the future T_{out} steps, we append $P_{\text{out}} = \lceil T_{\text{out}}/T_p \rceil$ learnable future tokens $\boldsymbol{F} \in \mathbb{R}^{P_{\text{out}} \times D_f}$ with a hidden dimension of D_f, shared across all sensors, each concatenated with the calendar embeddings of its target time slot. This gives the tokenized input representation $\boldsymbol{H}_{\text{in}} \in \mathbb{R}^{P \times N \times D}$, where $P = P_{\text{in}} + P_{\text{out}}$.

4.3 RoPE-based Temporal Attention

Given the patchified input representation $\boldsymbol{H}_{\text{in}}$, we first model the temporal evolution of each sensor with self-attention along the patch axis. Although calendar embeddings already carry absolute time information, they do not encode the relative order within the sequence. Therefore, we encode relative positions with rotary position embedding (RoPE) (Su et al., 2024). Specifically, for each sensor, the normalized input $\bar{\boldsymbol{H}} = \text{RMSNorm}(\boldsymbol{H}_{\text{in}})$ (Zhang & Sennrich, 2019) is projected into

$$\boldsymbol{Q}_{\text{t}} = \text{RoPE}\left(\text{RMSNorm}(\bar{\boldsymbol{H}}\boldsymbol{W}_Q)\right), \quad \boldsymbol{K}_{\text{t}} = \text{RoPE}\left(\text{RMSNorm}(\bar{\boldsymbol{H}}\boldsymbol{W}_K)\right), \quad \boldsymbol{V}_{\text{t}} = \bar{\boldsymbol{H}}\boldsymbol{W}_V, \tag{2}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{D \times d_h}$ are learnable projections and $d_h = D/M$. The inner RMSNorm stabilizes the attention logits (Henry et al., 2020; Dehghani et al., 2023), and RoPE makes each query–key product depend on position through the relative offset. Then, attention and a SwiGLU feed-forward network (Shazeer, 2020) update the tokens through residual connections:

$$\begin{aligned} \boldsymbol{H}_{\text{a}} &= \boldsymbol{H}_{\text{in}} + \text{Linear}\left(\text{Concat}_h\left(\text{softmax}(\boldsymbol{Q}_{\text{t}}\boldsymbol{K}_{\text{t}}^{\top}/\sqrt{d_h})\,\boldsymbol{V}_{\text{t}}\right)\right), \\ \boldsymbol{H}_{\text{t}} &= \boldsymbol{H}_{\text{a}} + \text{SwiGLU}\left(\text{RMSNorm}(\boldsymbol{H}_{\text{a}})\right), \end{aligned} \tag{3}$$

where Concat_h joins the M head outputs and $\boldsymbol{H}_{\text{t}} \in \mathbb{R}^{P \times N \times D}$ is the output temporal representation. Here, attention is bidirectional over all P patches, so future tokens could aggregate information from observed patches. The block is applied independently to all sensors with shared parameters. Bias and dropout terms are omitted for brevity. The complete equations are given in Appendix H.

4.4 In-Context Spatial Modeling

To complement temporal modeling, we then capture spatial dependencies across sensors. Prior traffic models generate spatial or attention parameters from geographic attributes (Pan et al., 2019),

sensor embeddings (Bai et al., 2020; Dong et al., 2024), or sensor-specific latent variables (Cirstea et al., 2022). In contrast, we compute both from the input context so that ICSM can be applied directly to unseen networks. Vanilla attention produces input-dependent weights, but its query and key projections use the same learned parameters across windows. ICSM extends this mechanism by introducing two complementary forms. Inspired by feature-wise conditioning in FiLM (Perez et al., 2018), *low-rank context modulation* adjusts the query–key matching transformation through shared low-rank bases, enabling the matching criterion to vary with each sensor's current state. A *context gate* scales the contribution of the aggregated spatial information. These mechanisms explicitly control how cross-sensor information is calculated and how strongly it influences the representation.

Low-rank Context Modulation. Starting from $\boldsymbol{Z} = \mathrm{RMSNorm}(\boldsymbol{H}_{\mathrm{t}})$, we summarize the P_{in} input patches of each sensor and project the summary with an MLP layer:

$$\boldsymbol{C} = \big[\mathrm{Mean}_{\mathrm{in}}(\boldsymbol{Z}); \mathrm{Last}_{\mathrm{in}}(\boldsymbol{Z}); \mathrm{Std}_{\mathrm{in}}(\boldsymbol{Z}); \mathrm{Stat}(\boldsymbol{X})\big], \quad [\boldsymbol{\Gamma}_q, \boldsymbol{\Gamma}_k, \tilde{\boldsymbol{g}}] = \mathrm{Linear}\big(\mathrm{MLP}(\boldsymbol{C})\big). \tag{4}$$

The first three terms pool the first P_{in} patches to describe each sensor's average state, latest state, and variability. The two columns of $\mathrm{Stat}(\boldsymbol{X})$ measure (i) the standardized recent residual relative to the window's time-of-day profile, and (ii) the recent volatility relative to the whole window, computed as the ratio of the standard deviation of changes in the recent segment to that over the full window. Both are detailed in Appendix I. These two features compare the recent segment against the whole window, indicating whether a sensor has recently deviated from its typical level. The MLP layer yields modulation vectors $\boldsymbol{\Gamma}_q, \boldsymbol{\Gamma}_k \in \mathbb{R}^{N\times M\times r}$ with rank r of M heads, and gate logits $\tilde{\boldsymbol{g}} \in \mathbb{R}^N$.

Before modulation, the base query–key matching should reflect local differences, but traffic representations share a network-wide component that obscures them. We therefore center each patch across sensors, $\boldsymbol{Z}^c = \boldsymbol{Z} - \mathrm{Mean}_n(\boldsymbol{Z})$, and compute unit-norm base queries $\boldsymbol{Q}_{\mathrm{s}}^0 = \ell_2(\boldsymbol{Z}^c\boldsymbol{W}_q)$ and keys $\boldsymbol{K}_{\mathrm{s}}^0 = \ell_2(\boldsymbol{Z}^c\boldsymbol{W}_k)$, whereas values $\boldsymbol{V}_{\mathrm{s}} = \boldsymbol{Z}\boldsymbol{W}_v$ keep the uncentered features. Sensors are thus compared by their local departures, while the exchanged information retains the full representation.

Instead of generating full query and key projections, which would require $2Dd_h$ values per head and sensor, the modulation vectors rescale r directions of shared low-rank bases:

$$\Delta\boldsymbol{Q} = \Big(\big[(\boldsymbol{Z}^c\boldsymbol{A}_q^\top) \odot \boldsymbol{\Gamma}_q\big]\boldsymbol{B}_q\Big), \Delta\boldsymbol{K} \quad = \Big(\big[(\boldsymbol{Z}^c\boldsymbol{A}_k^\top) \odot \boldsymbol{\Gamma}_k\big]\boldsymbol{B}_k\Big). \tag{5}$$

where, for each head, $\boldsymbol{A}_q, \boldsymbol{A}_k \in \mathbb{R}^{r\times D}$ and $\boldsymbol{B}_q, \boldsymbol{B}_k \in \mathbb{R}^{r\times d_h}$ are learned bases shared across sensors and patches, $\boldsymbol{\Gamma}_q, \boldsymbol{\Gamma}_k$ take the $N \times r$ slice of that head, and $\odot$ is element-wise multiplication with $\boldsymbol{\Gamma}$ broadcast over patches. Besides, we bound the norm of each increment as $\delta_{\max}$ so that modulation adjusts rather than replaces the unit-norm base. Before clipping, the query modulation multiplies the centered features of each sensor by $\boldsymbol{A}_q^\top \mathrm{diag}(\boldsymbol{\gamma})\boldsymbol{B}_q$, a matrix of rank at most r, where $\boldsymbol{\gamma}$ is the sensor's row of $\boldsymbol{\Gamma}_q$. Thus, the bases provide reusable matching directions, and the context sets their weights for each sensor and window.

Each head then attends attention over sensors at every patch p with the modulated queries and keys:

$$\alpha_p(n, n') = \mathrm{softmax}_{n'}\left(s_h\langle \boldsymbol{q}_{\mathrm{s},p,n}, \boldsymbol{k}_{\mathrm{s},p,n'}\rangle\right), \quad \boldsymbol{o}_{p,n} = \sum\nolimits_{n'} \alpha_p(n, n')\, \boldsymbol{v}_{\mathrm{s},p,n'}, \tag{6}$$

where vectors are rows of the corresponding matrices and s_h is a learned per-head scale as in cosine attention (Liu et al., 2022). As the query carries the context of sensor n and the key that of n', query modulation changes how a sensor weighs others, and key modulation how others weigh it.

Context Gate. The spatial representation produced by context-modulated attention may differ in usefulness across sensors and traffic conditions. We complement adaptive query–key matching with explicit control over its contribution. A context gate, therefore, is designed to scale the overall spatial update for each sensor:

$$\boldsymbol{S} = \mathrm{Linear}\big(\mathrm{Concat}_h(\boldsymbol{O}^h)\big), \qquad \boldsymbol{g} = \sigma(\tilde{\boldsymbol{g}} + b_0), \qquad \boldsymbol{H}_{\mathrm{s}} = \boldsymbol{H}_{\mathrm{t}} + \boldsymbol{g} \odot \boldsymbol{S}, \tag{7}$$

where $\boldsymbol{O}^h$ is the collection of $\boldsymbol{o}_{p,n}$ with head h. σ is the sigmoid function with fixed bias b_0 and $\odot$ applies each sensor's gate to all its patches. This per sensor and window gate varies with the traffic state. The output $\boldsymbol{H}_{\mathrm{s}} \in \mathbb{R}^{P\times N\times D}$ is the spatially aggregated representation.

4.5 Quantile Decoding and Training Objective

Finally, a linear head decodes the future tokens $\boldsymbol{H}_{\mathrm{s}}^{\mathrm{out}} \in \mathbb{R}^{P_{\mathrm{out}}\times N\times D}$ of the final representation. For each future token, the output head produces a $T_p \times |\mathcal{Q}|$ block of values, i.e., one forecast per quantile level in $\mathcal{Q}$ for each of the T_p steps covered by its patch. This gives the normalized forecasts

$\hat{\tilde{\boldsymbol{Y}}} \in \mathbb{R}^{T_{\text{out}} \times N \times |\mathcal{Q}|}$. To learn these quantiles, we minimize the pinball loss (Wen et al., 2017):

$$\mathcal{L} = \frac{1}{T_{\text{out}} N |\mathcal{Q}|} \sum_{t=1}^{T_{\text{out}}} \sum_{n=1}^{N} \sum_{\tau \in \mathcal{Q}} \max\left\{ \tau e_{t,n}^{(\tau)},\, (\tau - 1) e_{t,n}^{(\tau)} \right\}, e_{t,n}^{(\tau)} \quad = \tilde{Y}_{t,n} - \hat{\tilde{Y}}_{t,n}^{(\tau)}, \tag{8}$$

where $\tilde{Y}_{t,n}$ is the ground truth normalized using the input statistics at future step t and sensor n, and $\hat{\tilde{Y}}_{t,n}^{(\tau)}$ is the corresponding prediction at the quantile level $\tau \in \mathcal{Q}$. The loss weights under-prediction by τ and over-prediction by $1 - \tau$. Forecasts are restored as $\hat{\boldsymbol{Y}}^{(\tau)} = \mu + s\, \hat{\tilde{\boldsymbol{Y}}}^{(\tau)}$. The median of the sorted quantiles is used for the point forecasts, as detailed in Appendix G.

5 Experiments

5.1 Pretraining Dataset: Japan Road Traffic Information(JaRTI)

JaRTI contains cross-section traffic flow from 38,942 detectors across 51 Japanese regions, spanning January 2018 to December 2025 at a minimum temporal resolution of five minutes. The dataset comprises approximately 31 billion real observations, about 204 times the pretraining dataset of OpenCity (151 million, Li et al., 2024) and 6.8 times the public traffic benchmark LargeST (4.52 billion raw points, Liu et al., 2023b). In comparison, FactoST-v2 uses about 11 billion synthetic points and 0.56 billion real observations (Zhong et al., 2026). The scale and temporal coverage of JaRTI provide diverse real traffic patterns for pretraining. Data processing details and per-region statistics are provided in Appendix A and Table 5. Since pretraining uses Japanese data only and evaluation uses US networks, the two datasets do not share any sensors or regions.

5.2 Experimental Setup

Benchmarks, protocols, and metrics. We evaluate on six widely used US traffic benchmarks: METR-LA and PEMS-BAY contain traffic speed (Li et al., 2018), while PEMS03, PEMS04, PEMS07, and PEMS08 contain traffic flow (Guo et al., 2019; Song et al., 2020). We adopt the standard chronological splits of 7:1:2 for METR-LA and PEMS-BAY (Li et al., 2018) and 6:2:2 for PEMS03/04/07/08 (Song et al., 2020), applying the pretrained model directly to each test split for zero-shot forecasting. We consider $12 \to 12$ and $96 \to 96$ input–output settings and report MAE and RMSE averaged over all forecast steps following previous works (Zhong et al., 2025; 2026). More dataset details are provided in Appendix C.

Baselines. We compare Hodos with spatiotemporal foundation models (STFMs): FactoST-v2 (Zhong et al., 2026), FactoST (Zhong et al., 2025), OpenCity (Li et al., 2024), and UniST (Yuan et al., 2024); and general time-series foundation models (General TSFMs): TimesFM (Das et al., 2024), Moirai (Woo et al., 2024), and multivariate Chronos-2 (Ansari et al., 2025). Results follow Table IX of Zhong et al. (2026), except for our evaluations of TimesFM on PEMS-BAY and Chronos-2 on all six benchmarks. We average MAE ranks over each model's unstarred settings. Here stars denote pretraining overlap. Appendices D and E provide additional baseline information.

Implementation. Hodos embeds patches of length $T_p = 6$ into 64-dimensional vectors and concatenates them with time-of-day and day-of-week embeddings to form tokens of width $D = 192$. These tokens pass through four blocks of temporal attention and ICSM, each with eight attention heads. The temporal feed-forward network uses SwiGLU with a hidden dimension of 688, and ICSM uses modulation rank $r = 8$. We predict the quantiles $\mathcal{Q} = \{0.1, 0.2, \ldots, 0.9\}$ and pretrain the model on JaRTI for 100,000 steps using AdamW (Loshchilov & Hutter, 2019) on NVIDIA RTX Pro 6000 GPUs. Additional architecture and training settings are provided in Appendix C.

5.3 Zero-shot Performance

Table 2 presents the zero-shot forecasting results on six benchmarks. Hodos achieves the best MAE and RMSE in 11 of the 12 settings, with average reductions of 26.9% and 26.2% over the strongest eligible baseline in each setting, and attains the best average MAE rank of 1.08 across all settings. The only exception is PEMS-BAY at horizon 12, where FactoST performs better. The gains are smaller on speed benchmarks than on flow benchmarks, which likely reflects the composition of the

Table 2: Zero-shot performance of MAE and RMSE for 12→12 and 96→96 horizons. **Bold**: best; underline: second best; *: benchmark included in the model's pretraining corpus, excluded from ranking. Impr.: relative error reductions in MAE / RMSE of Hodos over the best eligible baseline.

		Ours	STFMs				General TSFMs			
Dataset	Len.	**Hodos**	FactoST-v2	FactoST	OpenCity	UniST	TimesFM	Moirai	Chronos-2	**Impr.**
PEMS03	12	**19.53 / 31.43**	25.70 / 40.75	30.12 / 46.92	30.37 / 47.49	101.87 / 129.94	29.71 / 49.21	28.23* / 46.36*	25.73 / 41.85	+24.0% / +22.9%
	96	**39.99 / 63.95**	86.90 / 124.62	113.62 / 144.80	125.18 / 159.23	102.87 / 137.90	108.59 / 152.74	74.85* / 110.78*	93.99 / 137.10	+54.0% / +48.7%
PEMS04	12	**25.42 / 39.68**	33.13 / 49.92	38.65 / 56.59	39.34* / 58.50*	67.91 / 87.63	35.00 / 53.27	34.65* / 52.13*	32.44 / 49.71	+21.6% / +20.2%
	96	**52.36 / 79.82**	107.15 / 147.60	142.11 / 175.99	153.18* / 188.95*	115.76 / 153.58	127.39 / 171.13	105.95* / 141.40*	109.37 / 153.88	+51.1% / +45.9%
PEMS07	12	**29.80 / 45.63**	37.85 / 57.48	45.47 / 66.68	45.18 / 67.15	104.93 / 133.65	42.10 / 64.59	35.64* / 50.25*	37.14 / 57.55	+19.8% / +20.6%
	96	**60.94 / 89.84**	126.83 / 173.31	156.07 / 190.59	172.20 / 211.25	144.67 / 184.62	151.99 / 205.18	125.91* / 169.43*	128.75 / 179.19	+52.0% / +48.2%
PEMS08	12	**21.23 / 33.23**	27.11 / 41.18	32.14 / 47.27	32.45* / 48.42*	73.46 / 93.14	29.68 / 45.18	38.23* / 53.12*	26.92 / 41.65	+21.1% / +19.3%
	96	**44.78 / 67.36**	91.09 / 127.28	121.43 / 151.59	128.48* / 161.75*	104.77 / 136.21	92.72 / 126.10	119.10* / 151.83*	95.61 / 137.58	+50.8% / +46.6%
PEMS-BAY	12	2.20 / 5.01	2.27 / 5.46	**2.02 / 4.59**	3.23* / 6.91*	14.89 / 16.92	2.53 / 6.10	1.97* / 4.69*	2.35 / 5.56	−8.9% / −9.2%
	96	**4.99 / 9.30**	6.11 / 11.56	6.12 / 10.47	6.92* / 11.79*	9.29 / 12.72	6.27 / 11.73	6.51* / 11.70*	5.03 / 10.13	+0.8% / +8.2%
METR-LA	12	**4.44 / 9.25**	5.32 / 12.82	5.98 / 14.26	4.30* / 8.37*	24.33 / 29.31	6.59 / 14.99	5.55 / 13.79	5.26 / 11.44	+15.6% / +19.1%
	96	**8.35 / 14.27**	10.87 / 21.49	12.68 / 18.82	10.85* / 17.60*	25.88 / 30.19	14.16 / 24.01	12.87 / 22.19	10.57 / 18.95	+21.0% / +24.2%
Avg. Rank / Impr.		**1.08**	2.58	4.92	6.25	5.75	4.67	4.50	2.67	+26.9% / +26.2%

pretraining corpus: JaRTI contains only traffic flow observations. Nevertheless, even without any speed data in pretraining, Hodos still outperforms the baselines in most speed settings, suggesting that the traffic patterns it learns from flow transfer across measurement types.

Comparison with STFMs. Hodos consistently outperforms the STFM baselines on all four flow benchmarks. The advantage becomes more pronounced at the longer horizon, where it reduces MAE by over 50% relative to FactoST-v2. Furthermore, the improvements extend to both METR-LA settings and long-horizon PEMS-BAY forecasting, despite the change from traffic flow pretraining to speed prediction. These results demonstrate that Hodos learns traffic representations that transfer across regions and measurement types, with strong performance at both forecasting horizons.

Comparison with General TSFMs. Hodos outperforms TimesFM and multivariate Chronos-2 on both metrics in every setting and also beats Moirai in 11 of 12 settings, although Moirai was pretrained on five of the six evaluation networks (Woo et al., 2024). Chronos-2 (Ansari et al., 2025) was also pretrained on LargeST (Liu et al., 2023b), which shares the California PeMS source with our benchmarks. Nevertheless, on the four flow benchmarks at 96→96, MAE and RMSE reductions reach 46.6%–62.4% and 42.3%–55.6% relative to Moirai and 52.1%–57.5% and 48.1%–53.4% relative to Chronos-2. These gains confirm spatial modeling beyond univariate mode and support the context-conditioned mechanism of ICSM beyond vanilla cross-series attention.

5.4 Ablation Study

To examine the contributions of each component, we evaluate several variants on METR-LA and PEMS04, with all variants pretrained on JaRTI using the same training schedule. As shown in Table 3, the full model achieves the lowest error across all metrics. Removing either low-rank modulation or the context gate consistently degrades performance, indicating that adapting the spatial matching function and controlling the strength of spatial updates both contribute to accuracy. Notably, the variant with only the context gate is sometimes worse than vanilla attention; this suggests that the two components are complementary.

Table 3: Ablation study on METR-LA and PEMS04 (MAE / RMSE).

Variant	**METR-LA** (MAE / RMSE)		**PEMS04** (MAE / RMSE)	
	12→12	96→96	12→12	96→96
Hodos (full)	**4.44 / 9.25**	**8.35 / 14.27**	**25.42 / 39.68**	**52.36 / 79.82**
w/o low-rank modulation	4.65 / 9.47	8.64 / 14.51	26.36 / 41.04	55.89 / 81.39
w/o context gate	4.58 / 9.35	8.48 / 14.50	25.65 / 40.19	53.99 / 80.10
replace ICSM with vanilla self-attention	4.70 / 9.49	9.52 / 15.29	26.77 / 43.30	56.55 / 83.82
temporal-only backbone (w/o spatial modeling)	4.81 / 9.60	9.51 / 15.23	28.81 / 46.44	61.43 / 91.77

Furthermore, replacing spatial modeling with vanilla spatial self-attention increases error across all four settings. Compared with the temporal-only backbone, vanilla spatial attention brings only marginal gains overall and sometimes even hurts accuracy, as in long-horizon forecasting on METR-

Table 4: Patch-length sensitivity (MAE / RMSE). Tokens: number of input tokens for a 96-step window. Time: wall-clock of one full evaluation pass over the METR-LA 96→96 test set.

T_p	Tokens	#Params	METR-LA		PEMS04		Time (s)
			12→12	96→96	12→12	96→96	
1	96	3.33M	4.54 / 9.36	9.62 / 14.99	25.99 / 40.47	54.59 / 81.45	75.6
2	48	3.33M	4.46 / 9.26	8.74 / 14.46	25.81 / 40.22	52.99 / 80.38	33.4
4	24	3.33M	4.49 / 9.32	8.65 / 14.47	25.51 / 39.71	52.37 / **79.09**	18.56
6 (default)	16	3.33M	**4.44 / 9.25**	**8.35 / 14.27**	**25.42 / 39.68**	52.36 / 79.82	13.97
12	8	3.35M	4.58 / 9.29	9.45 / 14.96	26.19 / 42.04	**52.13** / 79.85	**10.18**

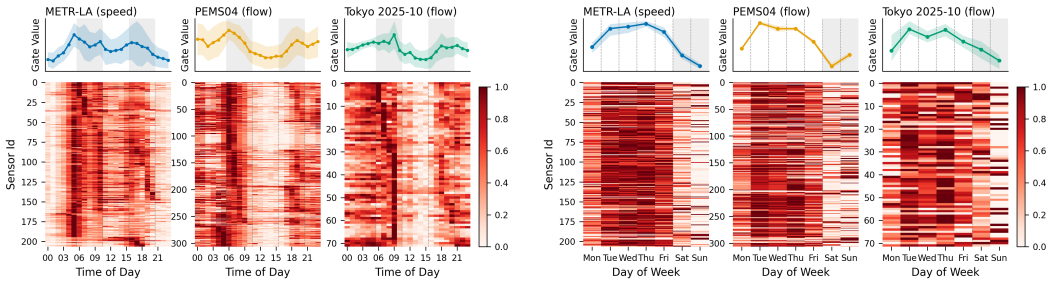
LA. In contrast, Hodos yields clear improvements across settings. This suggests that how spatial information is used matters, not only whether it is used.

5.5 Hyper-parameter Study

We study the effect of patch length T_p on forecasting accuracy and inference efficiency. For a fixed input length, a smaller T_p produces finer temporal patches and more tokens. As shown in Table 4, increasing T_p substantially reduces the number of tokens and inference time, while the parameter count remains nearly unchanged. Short-horizon performance is relatively stable across the tested values, whereas long-horizon METR-LA forecasting benefits from increasing T_p up to 6. However, further increasing it to 12 degrades accuracy on METR-LA and short-horizon PEMS04, despite a small additional saving in inference time. This suggests that overly large patches may obscure local traffic variations by reducing temporal granularity. Long-horizon PEMS04 is less sensitive among the larger patch sizes, with different values yielding the best MAE and RMSE. Overall, our default $T_p = 6$ performs best in three of the four settings while retaining a low inference cost.

5.6 Model Analysis

We examine how the context gate adjusts the strength of spatial updates across time and regions. Figure 2 visualizes its daily and weekly gate patterns on METR-LA, PEMS04, and Tokyo datasets. For clarity, we show a random subset of sensors from the Tokyo dataset. Despite differences in geographic location and number of traffic sensors, all three exhibit recurring increases around the morning and evening peak periods highlighted in Figure 2a. Moreover, the gate is generally higher on weekdays than on weekends in Figure 2b. These similar gate patterns suggest recurring spatial interactions across regions, providing a basis for learning transferable traffic representations through foundation-model pretraining. A possible explanation is that concentrated travel demand brings



(a) Context gate across the day.

(b) Context gate across the week.

Figure 2: Context-gate patterns across the day and week on METR-LA, PEMS04, and Tokyo. Heatmaps show normalized per-sensor spatial context gate values and curves show network means.

connected road segments closer to capacity, so a local slowdown can affect neighboring segments and make their observations more informative for forecasting. In contrast, more dispersed demand may reduce this coupling. The similar patterns across Japanese and US networks therefore suggest that the context gate captures a transferable tendency to use spatial information more strongly during these recurring traffic conditions. A supplementary spatial visualization is provided in Appendix B.

5.7 Case Study

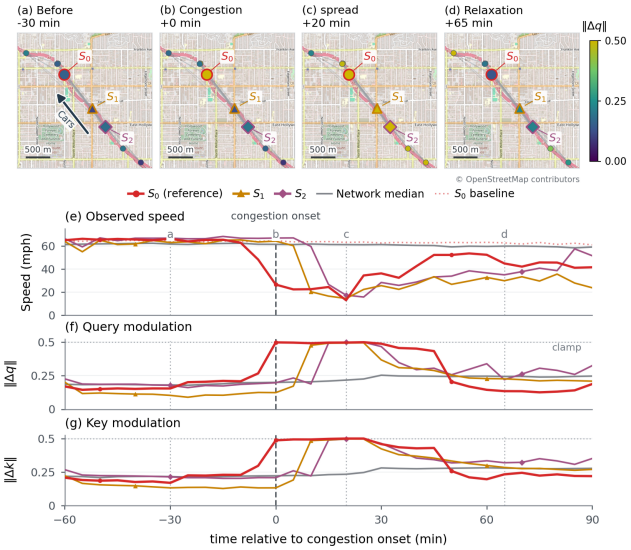


Figure 3: Modulated query and key norms during a congestion event on METR-LA. Maps show query-increment norms at successive stages and curves show traffic speed and the norms of $\Delta \boldsymbol{q}$ and $\Delta \boldsymbol{k}$ at three neighboring sensors.

We further conduct a case study to examine how low-rank context modulation responds to a congestion event on the METR-LA dataset. Figure 3 follows three neighboring sensors, with cars moving from S_2 through S_1 to S_0. Before congestion, traffic speeds are stable, and the query and key increments remain small. When the speed at S_0 drops sharply, both $\|\Delta \boldsymbol{q}\|$ and $\|\Delta \boldsymbol{k}\|$ increase, indicating a stronger adjustment to its spatial attention computation. Subsequently, similar responses appear at S_1 and S_2 as their speeds decline, consistent with congestion spreading upstream. During the later relaxation stage, the increments decrease toward background levels as speeds partially recover. Moreover, the network-median modulation changes much less than that of the affected sensors, indicating that the response is concentrated around the local event. By adapting both queries and keys, ICSM can adjust how an affected sensor draws on other sensors.

5.8 Efficiency Analysis

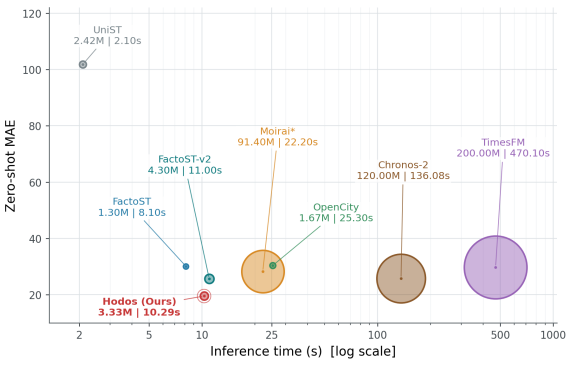


Figure 4: Zero-shot accuracy, inference time, and model size on PEMS03. Bubble area represents parameter count.

Figure 4 compares accuracy, runtime, and model size on PEMS03 using an NVIDIA A100, with some baseline efficiency results taken from Feng et al. (2026) using the same GPU. Among these models, Hodos achieves the best forecasting accuracy with only 3.33M parameters. UniST is fastest but has the highest MAE. Hodos has a runtime comparable to that of FactoST and FactoST-v2 and is faster than OpenCity and Moirai. Chronos-2 and TimesFM require substantially longer inference times, reflecting their much larger parameter counts. Overall, Hodos offers a promising balance between accuracy, model size, and inference efficiency.

6 Conclusion

In this paper, we propose Hodos, a traffic forecasting foundation model that learns transferable spatial interactions from traffic context alone through ICSM, without requiring target-domain spatial priors. Specifically, low-rank context modulation adapts cross-sensor query–key matching, while a context gate controls spatial update strength. Pretrained on JaRTI from 51 Japanese regions, Hodos achieves state-of-the-art performance on six US benchmarks, reducing average MAE and RMSE by 26.9% and 26.2%. Component ablation results support the effectiveness of context-conditioned spatial modeling for cross-region traffic forecasting. Future work will expand the pretraining corpus to cover more regions and traffic measurement types like speed and occupancy, and extend Hodos to integrate multimodal data, allowing contextual signals beyond traffic observations to inform ICSM.

AI USE STATEMENT

Generative AI tools were used in this work to polish the language of author-written drafts and to assist in analyzing experimental results during experiments. All AI-assisted text and analyses were checked by the authors and the authors take responsibility for the final content of this work.

ETHICS STATEMENT

JaRTI is an author-processed compilation of cross-section traffic-volume records published by the Japan Road Traffic Information Center (JARTIC) through its open-data website.[1] JARTIC's official terms of use[2] permit copying, redistribution, and modification, including commercial use, subject to the stated conditions and applicable third-party rights, and explicitly state compatibility with Creative Commons Attribution 4.0 International (CC BY 4.0). We attribute the original records to JARTIC and document our processing in Appendix A. JaRTI contains no vehicle identifiers, trajectories, images, or personal information.

REPRODUCIBILITY STATEMENT

Section 5.2 states the benchmarks, protocols, metrics, baseline sources, and model configuration. Appendix C summarizes the architecture, optimization, augmentation, and hardware settings. Appendix A documents the corpus and its cleaning rules. Appendix D summarizes the baseline models. The code and a subset of the dataset are available at `https://anonymous.4open.science/r/Hodos-C85E`. Due to file size limitations, the complete dataset and model checkpoints will be publicly released upon acceptance of the paper.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 17804–17815, 2020.

George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

Wei Chen, Yuqian Wu, Junle Chen, Xiaofang Zhou, and Yuxuan Liang. UrbanFM: Scaling urban spatio-temporal foundation models. *arXiv preprint arXiv:2602.20677*, 2026.

Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2900–2913, 2022.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[1] `https://www.jartic.or.jp/service/opendata/`

[2] `https://www.jartic.or.jp/d/opendata/riyou_kiyaku.pdf`

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Jiewen Deng, Jinliang Deng, Du Yin, Renhe Jiang, and Xuan Song. Tts-norm: Forecasting tensor time series via multi-way normalization. *ACM Transactions on Knowledge Discovery from Data*, pp. 1–25, 2023a.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang. ST-Norm: Spatial and temporal normalization for multi-variate time series forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 269–278, 2021.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W. Tsang. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, pp. 7665–7680, 2023b.

Jinliang Deng, Xiusi Chen, Renhe Jiang, Du Yin, Yi Yang, Xuan Song, and Ivor W Tsang. Disentangling structured components: Towards adaptive, interpretable and scalable time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, pp. 3783–3800, 2024.

Zheng Dong, Renhe Jiang, Haotian Gao, Hangchen Liu, Jinliang Deng, Qingsong Wen, and Xuan Song. Heterogeneity-informed meta-parameter learning for spatiotemporal time series forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 631–641, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Lixiang Fan, Bohao Li, Tao Zou, Junchen Ye, and Bowen Du. Incident-guided spatiotemporal traffic forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 243–254, 2026.

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 & Data Mining (KDD)*, pp. 307–317, 2025.

Yutong Feng, Shiyuan Piao, Yutong Xia, Xu Liu, Wenqi Fan, Fugee Tsung, See-Kiong Ng, and Yuxuan Liang. Learning spatio-temporal foundation models from pure synthetic data. *arXiv preprint arXiv:2607.16251*, 2026.

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 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3998–4006, 2024.

Haotian Gao, Zheng Dong, Jiawei Yong, Shintaro Fukushima, Kenjiro Taura, and Renhe Jiang. How different from the past? spatio-temporal time series forecasting with self-supervised deviation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

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

Renhe Jiang, Du Yin, Zhaonan Wang, Yizhuo Wang, Jiewen Deng, Hangchen Liu, Zekun Cai, Jinliang Deng, Xuan Song, and Ryosuke Shibasaki. DL-Traff: Survey and benchmark of deep learning models for urban traffic prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)*, pp. 4515–4525, 2021.

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

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 (ICLR)*, 2022.

Hyunwook Lee and Sungahn Ko. TESTAM: A time-enhanced spatio-temporal attention model with mixture of experts. In *The Twelfth International Conference on Learning Representations*, 2024.

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 Transactions on Knowledge Discovery from Data*, 17(1):1–21, 2023.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

Zhonghang Li, Long Xia, Lei Shi, Yong Xu, Dawei Yin, and Chao Huang. OpenCity: Open spatio-temporal foundation models for traffic prediction. *arXiv preprint arXiv:2408.10269*, 2024.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pp. 4125–4129, 2023a.

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. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023b.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer V2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12009–12019, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1720–1730, 2019.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*, pp. 4454–4458, 2022a.

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

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

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

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Xinyu Su, Feng Liu, Yanchuan Chang, Egemen Tanin, Majid Sarvi, and Jianzhong Qi. Dualcast: a model to disentangle aperiodic events from traffic series. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3290–3298, 2025.

Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1907–1913, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 753–763, 2020.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3634–3640, 2018.

JiaJun Yu, Fang Yuan, Guang-Yong Chen, and Min Gan. Active arbitration: Decoupling spatio-temporal duality for efficient traffic forecasting. In *Proceedings of the 35th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 5236–5244, 8 2026.

Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. UniST: A prompt-empowered universal model for urban spatio-temporal prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 4095–4106, 2024.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Siru Zhong, Junjie Qiu, Yangyu Wu, Xingchen Zou, Zhongwen Rao, Bin Yang, Chenjuan Guo, Hao Xu, and Yuxuan Liang. Learning to factorize spatio-temporal foundation models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Siru Zhong, Junjie Qiu, Yangyu Wu, Yiqiu Liu, Yuanpeng He, Zhongwen Rao, Bin Yang, Chenjuan Guo, Hao Xu, and Yuxuan Liang. Learning to factorize and adapt: A versatile approach toward universal spatio-temporal foundation models. *arXiv preprint arXiv:2601.12083*, 2026.

Yiqing Zou, Hanning Yuan, Qianyu Yang, Ziqiang Yuan, Shuliang Wang, and Sijie Ruan. Meta dynamic graph for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16584–16592, 2026.

A JaRTI Datasheet

Source and variable. JaRTI is constructed from cross-section traffic-volume records published by the Japan Road Traffic Information Center (JARTIC) through its open-data service (`https://www.jartic.or.jp/service/opendata/`). Each observation records the number of vehicles passing a roadside detector at a minimum five-minute interval. Region files are organized by prefecture, with Hokkaido split into the Sapporo, Hakodate, Asahikawa, Kushiro, and Kitami sub-regions, giving 51 regions. The source data are provided under JARTIC's open-data terms of use[3], which state compatibility with CC BY 4.0. JaRTI is a processed compilation of these records.

Spatial layout. Two of the 38,942 detectors (one each in Akita and Saitama) have no coordinates, so Figure 5 shows the other 38,940. The detectors cover all 47 prefectures from Hokkaido to Okinawa and follow the road network rather than a regular grid. The large metropolitan areas carry dense road meshes. The Greater Tokyo and Osaka–Kyoto–Kobe views in panels (b) and (c) contain 6,914 and 5,185 detectors, together about 31% of the corpus. Elsewhere, detectors cluster around regional cities and along the main roads between them.

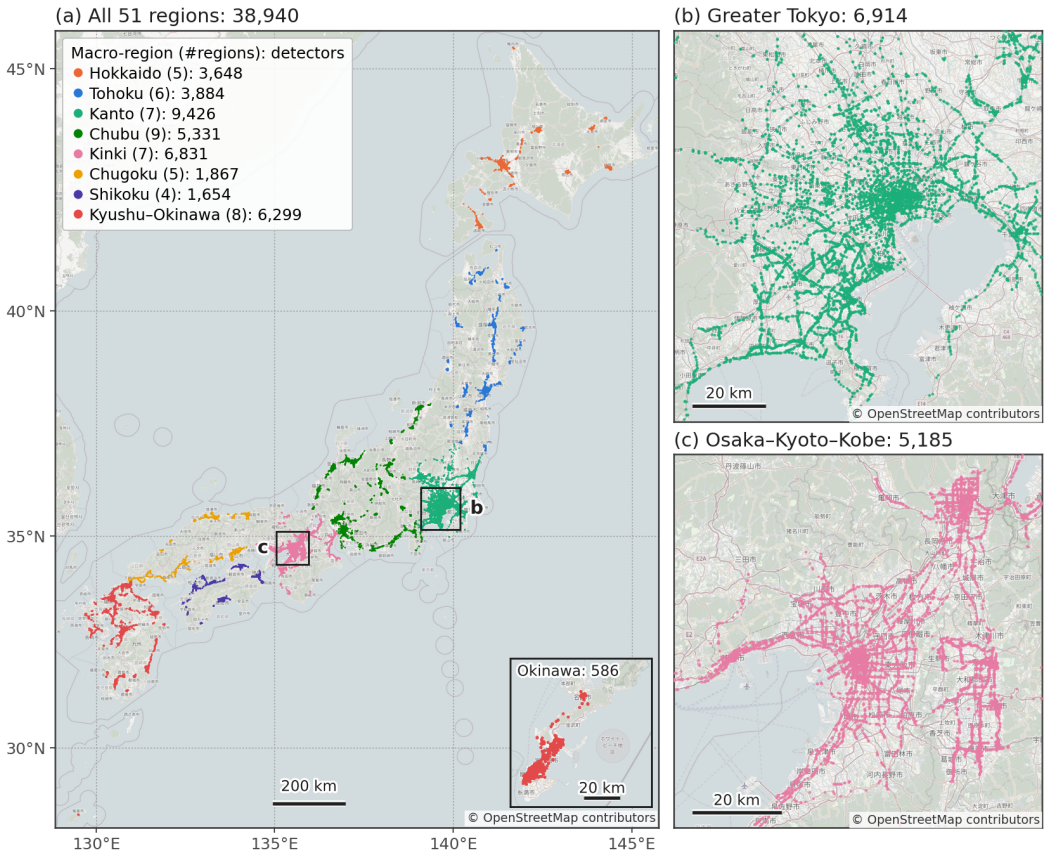


Figure 5: Spatial distribution of the JaRTI detectors on an OpenStreetMap basemap. (a) All 38,940 detectors with coordinates, colored by the eight traditional regions of Japan. The legend gives the number of JaRTI regions and detectors in each macro-region, and the inset shows Okinawa (586 detectors). The boxes mark the zoomed views of (b) Greater Tokyo and (c) Osaka–Kyoto–Kobe, whose titles give the number of detectors inside each box.

Coverage. Table 5 lists every region with its detector count, time span, and number of observations. All 51 regions span eight years in total from January 2018 to December 2025. A few region-months are absent, giving 4,853 region-month files. The corpus totals 38,942 detectors and 30,881,960,902 observations at minimum five-minute resolution. The largest regions by detectors are Tokyo (2,585),

[3] `https://www.jartic.or.jp/d/opendata/riyou_kiyaku.pdf`.

Kanagawa (2,332), Osaka (2,203), Fukuoka (2,072), and Sapporo (1,889) and the smallest are Akita (155) and Yamanashi (166).

Cleaning rules. Rows are pivoted to a regular grid per region-month and duplicate (timestamp, detector) rows are averaged. Missing values are masked out of the loss and metrics.

Scale relative to other corpora. OpenCity pretrains on about 151 million observations (Li et al., 2024). FactoST pretrains on about 13 million points and FactoST-v2 pretrains on about 11 billion points, most of which are synthetic (Zhong et al., 2025; 2026). LargeST (Liu et al., 2023b), the largest public traffic benchmark, covers 8,600 sensors from 2017 to 2021 at a five-minute raw interval (525,888 time frames), about 4.52 billion raw points. JaRTI is therefore about 200 times OpenCity's corpus, about 6.8 times LargeST, while consisting of real traffic observations only.

Region disjointness and privacy. Pretraining uses Japanese data only and all evaluation benchmarks are US networks. So no sensor, network, or region is shared. The corpus contains no vehicle identifiers, trajectories, or personal information.

Table 5: JaRTI: full regional breakdown of all 51 administrative regions. Detector counts are unique detectors reporting in each region's most recent month; observation counts are raw (timestamp, detector) rows summed over all region-month files. Totals: 38,942 detectors, 30,881,960,902 observations.

Region	#Det.	Span	#Obs.
Aichi	515	2018/01–2025/12	415,164,100
Akita	155	2018/01–2025/12	125,306,941
Aomori	648	2018/01–2025/12	548,842,793
Asahikawa	477	2018/01–2025/12	391,987,966
Chiba	983	2018/01–2025/12	823,399,583
Ehime	697	2018/01–2025/12	534,160,883
Fukui	416	2018/01–2025/12	331,076,735
Fukuoka	2,072	2018/01–2025/12	1,713,838,390
Fukushima	479	2018/01–2025/12	383,082,469
Gifu	543	2018/01–2025/12	236,543,243
Gumma	731	2018/01–2025/12	610,926,301
Hakodate	414	2018/01–2025/12	338,905,334
Hiroshima	292	2018/01–2025/12	244,217,286
Hyogo	1,027	2018/01–2025/12	730,935,708
Ibaraki	1,446	2018/01–2025/12	1,154,049,559
Ishikawa	612	2018/01–2025/12	447,397,398
Iwate	1,071	2018/01–2025/12	906,270,650
Kagawa	290	2018/01–2025/12	235,016,692
Kagoshima	753	2018/01–2025/12	619,243,581
Kanagawa	2,332	2018/01–2025/12	1,909,090,904
Kitami	310	2018/01–2025/12	249,155,095
Kochi	395	2018/01–2025/12	317,985,963
Kumamoto	1,089	2018/01–2025/12	814,250,073
Kushiro	558	2018/01–2025/12	436,492,321
Kyoto	1,008	2018/01–2025/12	821,723,208
Mie	359	2018/01–2025/12	280,312,216
Miyagi	1,004	2018/01–2025/12	805,204,814
Miyazaki	680	2018/01–2025/12	524,536,714
Nagano	1,095	2018/01–2025/12	974,568,204
Nagasaki	248	2018/01–2025/12	206,393,380
Nara	1,233	2018/01–2025/12	978,729,707
Niigata	749	2018/01–2025/12	641,608,101
Oita	593	2018/01–2025/12	492,282,544
Okayama	403	2018/01–2025/12	326,874,457
Okinawa	586	2018/01–2025/12	481,425,250
Osaka	2,203	2018/01–2025/12	1,842,538,353
Saga	278	2018/01–2025/12	235,310,116
Saitama	1,078	2018/01–2025/12	829,958,066
Sapporo	1,889	2018/01–2025/12	1,428,643,803
Shiga	471	2018/01–2025/12	402,716,389
Shimane	461	2018/01–2025/12	325,880,799
Shizuoka	513	2018/01–2025/12	446,037,947
Tochigi	272	2018/01–2025/12	209,719,833
Tokushima	272	2018/01–2025/12	222,803,072
Tokyo	2,585	2018/01–2025/12	1,678,232,049
Tottori	443	2018/01–2025/12	373,160,000
Toyama	722	2018/01–2025/12	554,236,494
Wakayama	530	2018/01–2025/12	455,057,751
Yamagata	528	2018/01–2025/12	452,882,662
Yamaguchi	268	2018/01–2025/12	243,444,149
Yamanashi	166	2018/01–2025/12	130,340,856
Total (51)	**38,942**	**2018/01–2025/12**	**31B**

B Extended Analysis

Gate on the map. Figure 6 places the METR-LA gate of Figure 2a (weekday non-holiday windows) on the sensor map at 03, 06, 09, 12, 17, and 21 h. The gate is high across the whole network at 06 and 17 h and low at 03 and 21 h, so the daily profile of Figure 2a is a network-wide movement rather than the behavior of a few sensors.

Prediction during congestion propagation. To complement the modulation analysis in Section 5.7, Figure 7 compares the forecasting behavior of Hodos and two ablation variants during the same congestion event. Before congestion, all three models closely follow the stable free-flow speeds. As congestion develops at S_0 and spreads upstream to S_1 and S_2, their predictions begin to diverge. The vanilla spatial self-attention and temporal-only variants produce shallower speed reductions, overestimating traffic speeds during the pronounced slowdowns. In contrast, Hodos captures a deeper decline at S_0 and follows the low-speed periods at S_1 and S_2 more closely, indicating a better response to the evolving traffic conditions. The blue uncertainty band also widens during congestion, reflecting greater predictive uncertainty around these changes. Nevertheless, Hodos still responds

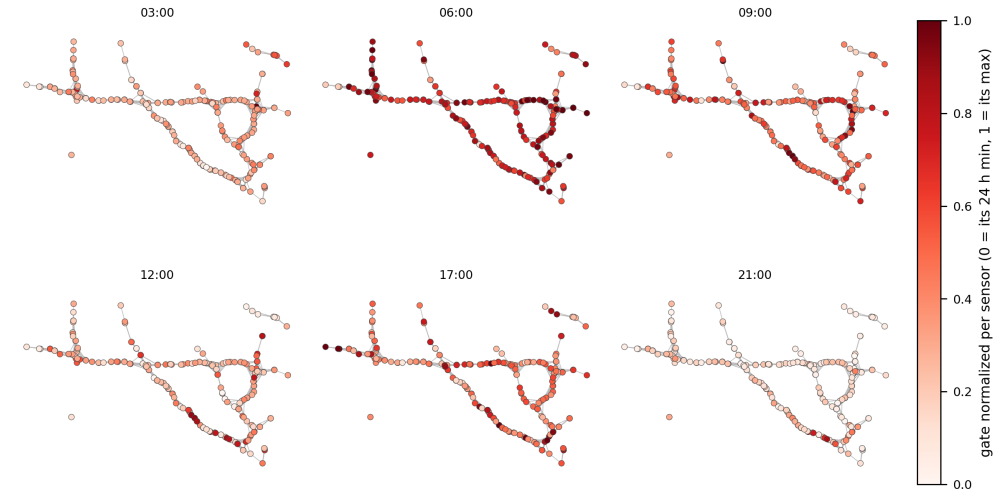


Figure 6: Per-sensor gate of ICSM on the METR-LA sensor map at six hours of the day. The METR-LA setting follows Figure 2a and each sensor is normalized to its own 24-hour range.

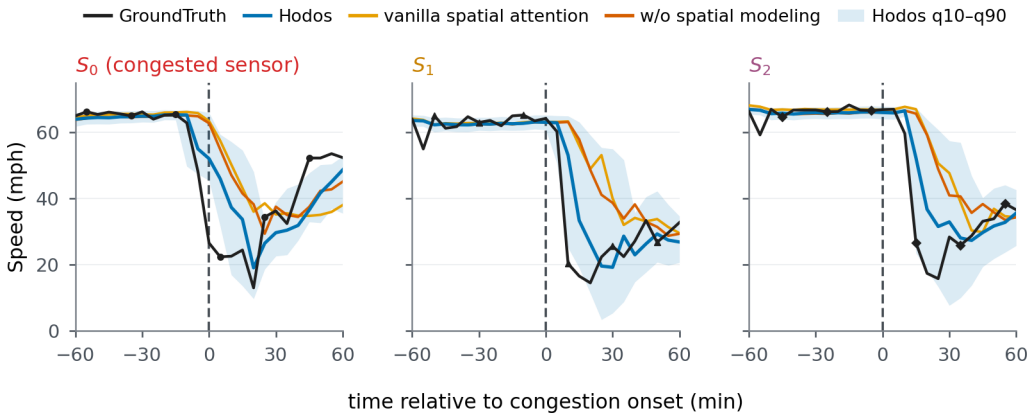


Figure 7: Traffic speed predictions at the three neighbouring METR-LA sensors in Figure 3. Colored curves show median predictions and blue shading indicates Hodos's predictive uncertainty. The dashed lines mark congestion onset at S_0.

late to the initial drop and underestimates the later recovery at S_0, so abrupt transitions remain challenging. Figure 8 further shows that the attention weights among the three sensors stay close to the uniform level before the onset, whereas 15–20 minutes after the onset all six rise and peak, so the affected sensors draw more strongly on one another as congestion propagates. These observations support the benefit of adapting spatial interactions to the current traffic context.

C Experimental Settings

Table 6 summarizes the model architecture, Table 7 reports optimization, augmentation, and hardware settings, and Table 8 lists the benchmark characteristics and chronological splits. The hyperparameters in Tables 6 and 7 were selected from preliminary runs on a small subset of the Japanese corpus JaRTI. Table 4 reports US benchmark results for other patch lengths for analysis only, and the default $T_p = 6$ was not selected post hoc from these results.

Ablation and patch-length runs. Every variant in Tables 3 and 4 is trained from scratch on JaRTI for 100,000 steps with the optimizer, schedule, and augmentation reported in Table 7. We evaluate the final checkpoint of each run. The variant without low-rank modulation removes the context-conditioned query and key increments while retaining the context gate. The variant without the

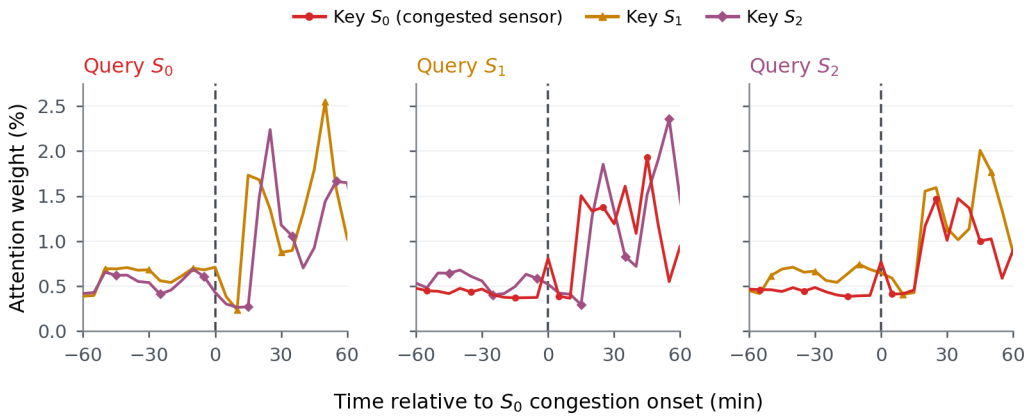


Figure 8: Spatial attention weights of Hodos among the three METR-LA sensors in Figure 3. Each panel takes one sensor as the query and line colors indicate the key sensor. The dashed lines mark congestion onset at S_0.

Table 6: Architecture details of Hodos.

Component	**Setting**
Blocks L	4 (each: temporal block followed by one ICSM block)
Width D	192 (value embedding 64, time-of-day embedding 64, day-of-week embedding 64)
Heads M	8
Temporal block	RMSNorm, RoPE attention on the patch axis, SwiGLU, dropout 0.1
Patch length T_p	6 steps, input and output
Time-of-day encoding	8 Fourier frequency pairs, linearly projected
Day-of-week encoding	Embedding table of size 7
ICSM $\delta_{\max}$	0.5
ICSM rank r	8
Quantile levels $\mathcal{Q}$	9 levels: $\{0.1, 0.2, \ldots, 0.9\}$. Output head $T_p \times \lvert\mathcal{Q}\rvert$ per future token
Instance normalization	joint time-and-sensor statistics per window with minimum scale 1.0

context gate retains low-rank modulation and replaces the input-dependent gate with a fixed scalar, $g_n = g_{\text{const}}$, shared across sensors and windows. The vanilla spatial self-attention variant replaces every ICSM block by a standard multi-head self-attention block over sensors. The temporal-only backbone removes the spatial attention blocks. The evaluation time of different patch-size variants in Table 4 is one full pass over the METR-LA 96→96 test set on one NVIDIA RTX PRO 6000 GPU in bf16, using an identical evaluation harness for all rows.

D Baseline Details and Transfer Settings

Some baseline results in Table 2 are quoted from Table IX of Zhong et al. (2026). All entries use the input and output lengths of Section 5.2 without parameter updates.

OpenCity (Li et al., 2024) integrates a timeShift transformer with graph neural networks and spatial context is encoded from the normalised Laplacian of the region graph (eigenvectors as region embeddings) and graph convolution layers, so an adjacency matrix of the target network is consumed at inference. Its pretraining corpus has 151,089,924 observations over 10,110 regions and includes METR-LA, PEMS-BAY, PEMS04, PEMS08, CAD4, CAD7, CAD8, CAD12, NYC taxi, and four Chinese city datasets. The corresponding entries are starred. OpenCity reports zero-shot, fine-tuning, and efficient fine-tuning performance. Its zero-shot mode consumes the target adjacency. Variants: mini (2M), base (5M), plus (26M).

Table 7: Optimization, augmentation, and hardware settings of the main pretraining run.

Item	Setting
Optimizer steps	100,000
Objective	masked pinball loss averaged over the 9 levels and over valid targets
Optimizer	AdamW, learning rate 10^{-3}, betas $(0.9, 0.95)$, weight decay 0.1, $\epsilon = 10^{-8}$, gradient clipping 1.0
Schedule	warmup-stable-decay (Hu et al., 2024): 5,000 warmup steps, constant, decay over the last 50% of steps to 5% of the peak
Precision / memory	bf16 autocast / gradient checkpointing
Augmentation	polarity flip of the normalized window with probability 0.3
Hardware	NVIDIA RTX Pro 6000 GPUs
Inference	window statistics only and no target statistics, graph, coordinates, or identities

Table 8: Evaluation benchmark characteristics and chronological splits. Only the test portion is used for evaluation.

Dataset	Sensors	Variable	Area	Split
METR-LA	207	speed	Los Angeles	7:1:2
PEMS-BAY	325	speed	Bay Area	7:1:2
PEMS03	358	flow	California	6:2:2
PEMS04	307	flow	California	6:2:2
PEMS07	883	flow	California	6:2:2
PEMS08	170	flow	California	6:2:2

UniST (Yuan et al., 2024) operates on grid-partitioned city data with tensors of shape $T \times C \times H \times W$, is pretrained on more than 20 datasets from multiple cities, and uses knowledge-guided prompts with spatial and temporal memory pools. It reports zero-shot prediction on grids and prompt tuning or fine-tuning for adaptation. Variants range from 2M to 30M parameters. UniST does not report a total observation count and the estimate in Table 1 is quoted from Zhong et al. (2026).

FactoST (Zhong et al., 2025) factorizes the model: stage one pretrains a space-agnostic backbone with multi-frequency reconstruction and domain-aware prompting and stage two freezes or fine-tunes the backbone and attaches an adapter that fuses spatial metadata, sparsifies interactions, and aligns domains with continual memory replay. Its headline results are few-shot and its pretraining corpus is about 13 million points from six domains. FactoST reports zero-shot results with the backbone alone, without the adapter.

FactoST-v2 (Zhong et al., 2026) uses an encoder-only backbone with randomized sequence masking, probabilistic quantile prediction across variable horizons, and an adapter with meta-adaptive learning and prompting that binds a spatial bank of node embeddings and calendar-aware temporal embeddings to the target network through fine-tuning. Its zero-shot mode is the backbone alone with no parameter updates, and the corpus has about 11 billion points from eight domains, real plus KernelSynth synthetic. The default Tiny variant has 4,378,928 parameters and other variants have 2.5M, 12.2M, and 30.4M.

TimesFM (Das et al., 2024) is a decoder-only, patched time-series foundation model pretrained on about 380 billion mixed real and synthetic time points and forecasts each series independently. We reevaluate TimesFM 1.0 (200M) on PEMS-BAY using the checkpoint `timesfm-1.0-200m-pytorch` in fp32, with each sensor supplied only the prescribed input window. We use the predicted median as the point forecast and compute masked MAE and RMSE over all forecast steps, excluding zero-valued targets. The remaining TimesFM entries are quoted from Zhong et al. (2026).

Moirai 1.0 (Woo et al., 2024) is a masked-encoder universal forecaster with multi-patch projections and any-variate attention, pretrained on LOTSA (27 billion observations across nine domains, including several PEMS traffic datasets, which are starred). The FactoST baseline setup specifies

the checkpoint `Salesforce/moirai-1.0-R-base` (Zhong et al., 2025) and we quote the zero-shot results from Table IX of Zhong et al. (2026).

Chronos-2 (Ansari et al., 2025) uses group attention to share information across series. We evaluate its `amazon/chronos-2` checkpoint with all sensors in each window grouped into one multivariate task. More details are discussed in Appendix E.

E Cross-sensor Information in a General Time-series Foundation Model

To test whether cross-sensor information helps a general time-series foundation model, we ran Chronos-2 (Ansari et al., 2025) zero-shot on the six benchmarks. Two modes receive identical inputs: univariate, in which every sensor is an independent task, and multivariate, in which all sensors of a window form one task so that the group attention of Chronos-2 can share information across sensors. Table 9 reports both.

Table 9: Chronos-2 zero-shot forecasting, univariate versus multivariate inference (MAE / RMSE). Both rows use the same `amazon/chronos-2` checkpoint and the same inputs; they differ only in whether the sensors of a window are grouped into one multivariate task, so that Chronos-2's group attention can share information across sensors, or forecast as independent univariate tasks. Masked metrics on the test split, ground-truth zeros excluded, pooled over all horizon steps. Δ is the relative change of multivariate w.r.t. univariate; negative favors multivariate.

Len.	Model	PEMS03	PEMS04	PEMS07	PEMS08	PEMS-BAY	METR-LA
12→12	Chronos-2 (univariate)	25.85 / 42.27	32.96 / 50.90	37.76 / 58.81	27.20 / 42.36	2.38 / 5.71	5.40 / 11.72
	Chronos-2 (multivariate)	**25.73 / 41.85**	**32.44 / 49.71**	**37.14 / 57.55**	**26.92 / 41.65**	**2.35 / 5.56**	**5.26 / 11.44**
	Δ (%)	-0.5 / -1.0	-1.6 / -2.3	-1.6 / -2.1	-1.0 / -1.7	-1.5 / -2.6	-2.6 / -2.3
96→96	Chronos-2 (univariate)	99.64 / 156.03	119.85 / 177.87	146.52 / 213.37	108.16 / 161.79	6.02 / 11.57	11.55 / 19.93
	Chronos-2 (multivariate)	**93.99 / 137.10**	**109.37 / 153.88**	**128.75 / 179.19**	**95.61 / 137.58**	**5.03 / 10.13**	**10.57 / 18.95**
	Δ (%)	-5.7 / -12.1	-8.7 / -13.5	-12.1 / -16.0	-11.6 / -15.0	-16.5 / -12.4	-8.5 / -5.0

Multivariate inference is better in all 24 cells. MAE improves by 0.5% to 2.6% at 12→12 and by 5.7% to 16.5% at 96→96. At the longer horizon, the relative RMSE reduction exceeds the MAE reduction on the four flow benchmarks. Hodos achieves lower MAE than the multivariate mode across the evaluated settings, with particularly large differences on the flow benchmarks and similar performance on long-horizon PEMS-BAY forecasting (Table 2). The stronger benefit of cross-sensor information at the longer horizon also appears in Table 3, where adding ICSM to the temporal-only backbone reduces MAE by 7.7% to 11.8% at one hour and 12.2% to 14.8% at eight hours. Together, these comparisons support the value of cross-sensor information for traffic forecasting.

F Limitations

External contextual information. Hodos conditions on historical traffic observations and calendar features, without explicitly incorporating weather forecasts, roadwork schedules, or event information. These external signals could provide advance context for changes that have not yet appeared in the observed traffic series. Incorporating such optional inputs while preserving transfer across networks is a useful direction for future work.

From forecasting to traffic management. The evaluation focuses on cross-region traffic forecasting. The effects of using these forecasts for signal control or route guidance are outside the scope of the present study. Assessing these applications would require coupling the forecaster with a decision policy and evaluating outcomes such as travel time and congestion in a simulation or operational setting. This would establish how forecasting improvements translate into benefits for transportation systems when management decisions also influence subsequent traffic conditions.

G Point Forecasting via Quantile Rearrangement

From quantile outputs to point forecasts. Hodos predicts nine quantile levels $\mathcal{Q} = \{q_j = j/10\}_{j=1}^{9}$ using the head in Section 4.5. The pinball objective trains each output toward its corresponding quantile, but the unconstrained head can produce quantile crossings. We therefore apply *increasing rearrangement* at inference. For each forecast step t and sensor n, we first restore the predictions to their original units, $\hat{y}_{t,n}^{(q_j)} = \mu + s\hat{\hat{y}}_{t,n}^{(q_j)}$, and then sort the nine predicted values in ascending order:

$$\big(\hat{y}_{t,n}^{\uparrow(q_1)}, \dots, \hat{y}_{t,n}^{\uparrow(q_9)}\big) = \mathrm{sort}_{\uparrow}\big(\hat{y}_{t,n}^{(q_1)}, \dots, \hat{y}_{t,n}^{(q_9)}\big). \tag{9}$$

The point forecast is the median of this rearranged quantile vector:

$$\hat{y}_{t,n}^{\mathrm{point}} = \hat{y}_{t,n}^{\uparrow(0.5)} = \mathrm{median}\big\{\hat{y}_{t,n}^{(q_j)} : j = 1, \dots, 9\big\}. \tag{10}$$

Thus, we use the fifth smallest predicted value, which can differ from the raw output channel labeled $q = 0.5$ when quantiles cross. Sorting is performed independently for each step and sensor.

H RoPE-based Temporal Block

This appendix specifies the temporal block in Section 4.3, including normalization, rotary embeddings, multi-head attention, and the SwiGLU feed-forward network. We write the computation for one sensor, suppressing the batch and sensor indices, so that $\boldsymbol{H}_{\mathrm{in}} \in \mathbb{R}^{P \times D}$. There are M attention heads of even width $d_h = D/M$. In the first layer, $\boldsymbol{H}_{\mathrm{in}}$ contains the input and future tokens. In subsequent layers, it denotes the preceding spatial block's output. Parameters are shared across sensors within each layer.

Normalization and projections. For a row vector $\boldsymbol{x} \in \mathbb{R}^d$, RMSNorm (Zhang & Sennrich, 2019) acts along the feature dimension:

$$\mathrm{RMSNorm}(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\sqrt{d^{-1}\sum_{i=1}^{d} x_i^2 + \epsilon}} \odot \boldsymbol{\gamma}, \qquad \epsilon = 10^{-6}, \tag{11}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^d$ is a learned gain. The two token-wise norms below have separate gains of width D. Query and key norms have separate gains of width d_h, shared across heads. With $\boldsymbol{Z}_1 = \mathrm{RMSNorm}_1(\boldsymbol{H}_{\mathrm{in}})$, head h computes

$$\begin{aligned} \overline{\boldsymbol{Q}}_{\mathrm{t}}^{h} &= \mathrm{RMSNorm}_q(\boldsymbol{Z}_1 \boldsymbol{W}_Q^h + \boldsymbol{b}_Q^h), \\ \overline{\boldsymbol{K}}_{\mathrm{t}}^{h} &= \mathrm{RMSNorm}_k(\boldsymbol{Z}_1 \boldsymbol{W}_K^h + \boldsymbol{b}_K^h), \\ \boldsymbol{V}_{\mathrm{t}}^{h} &= \boldsymbol{Z}_1 \boldsymbol{W}_V^h + \boldsymbol{b}_V^h, \end{aligned} \tag{12}$$

where $\boldsymbol{W}_Q^h, \boldsymbol{W}_K^h, \boldsymbol{W}_V^h \in \mathbb{R}^{D \times d_h}$ and $\boldsymbol{b}_Q^h, \boldsymbol{b}_K^h, \boldsymbol{b}_V^h \in \mathbb{R}^{d_h}$. Biases are broadcast over patches.

Rotary position embedding. For patch positions $p = 0, \dots, P-1$, RoPE (Su et al., 2024) rotates adjacent feature pairs using

$$\begin{aligned} \boldsymbol{R}_p &= \mathrm{diag}\big(R(p\omega_0), \dots, R(p\omega_{d_h/2-1})\big), \qquad \omega_j = 10000^{-2j/d_h}, \\ R(\varphi) &= \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}. \end{aligned} \tag{13}$$

Using row-vector matrix notation, the rotated queries and keys are

$$\boldsymbol{Q}_{\mathrm{t}}^{h}[p,:] = \overline{\boldsymbol{Q}}_{\mathrm{t}}^{h}[p,:]\boldsymbol{R}_p^{\top}, \qquad \boldsymbol{K}_{\mathrm{t}}^{h}[p,:] = \overline{\boldsymbol{K}}_{\mathrm{t}}^{h}[p,:]\boldsymbol{R}_p^{\top}. \tag{14}$$

Values are not rotated. Since $\boldsymbol{R}_p^{\top}\boldsymbol{R}_{p'} = \boldsymbol{R}_{p'-p}$, each query–key dot product incorporates the relative patch offset:

$$\boldsymbol{Q}_{\mathrm{t}}^{h}[p,:]\big(\boldsymbol{K}_{\mathrm{t}}^{h}[p',:]\big)^{\top} = \overline{\boldsymbol{Q}}_{\mathrm{t}}^{h}[p,:]\boldsymbol{R}_{p'-p}\big(\overline{\boldsymbol{K}}_{\mathrm{t}}^{h}[p',:]\big)^{\top}. \tag{15}$$

Positions continue through the future patches in the joint sequence. No rotary embedding is applied along the sensor axis.

Attention and the first residual. Each head computes attention over all P patches, and the concatenated outputs are projected back to width D:

$$\begin{aligned} \boldsymbol{O}_{\mathrm{t}}^{h} &= \mathrm{softmax}\left(\frac{\boldsymbol{Q}_{\mathrm{t}}^{h}(\boldsymbol{K}_{\mathrm{t}}^{h})^{\top}}{\sqrt{d_h}}\right)\boldsymbol{V}_{\mathrm{t}}^{h}, \\ \boldsymbol{A}_{\mathrm{t}} &= \mathrm{Concat}_{h=1}^{M}(\boldsymbol{O}_{\mathrm{t}}^{h})\boldsymbol{W}_{O} + \boldsymbol{b}_{O}, \\ \boldsymbol{H}_{\mathrm{a}} &= \boldsymbol{H}_{\mathrm{in}} + \mathrm{Dropout}(\boldsymbol{A}_{\mathrm{t}}), \end{aligned} \tag{16}$$

where $\boldsymbol{W}_O \in \mathbb{R}^{D\times D}$ and $\boldsymbol{b}_O \in \mathbb{R}^{D}$. Softmax is applied row-wise over key patches, without a causal mask or dropout on the attention weights. In the compact main-text notation, $\mathrm{TA}(\boldsymbol{Q}_{\mathrm{t}}, \boldsymbol{K}_{\mathrm{t}}, \boldsymbol{V}_{\mathrm{t}})$ denotes this attention update, including head concatenation and output projection, with head and sensor indices suppressed.

SwiGLU and the second residual. A second token-wise RMSNorm precedes the SwiGLU feed-forward network (Shazeer, 2020):

$$\begin{aligned} \boldsymbol{Z}_2 &= \mathrm{RMSNorm}_2(\boldsymbol{H}_{\mathrm{a}}), \\ \boldsymbol{G} &= \mathrm{SiLU}(\boldsymbol{Z}_2\boldsymbol{W}_g + \boldsymbol{b}_g) \odot (\boldsymbol{Z}_2\boldsymbol{W}_u + \boldsymbol{b}_u), \\ \boldsymbol{F}_{\mathrm{t}} &= \mathrm{Dropout}(\boldsymbol{G})\boldsymbol{W}_d + \boldsymbol{b}_d, \\ \boldsymbol{H}_{\mathrm{t}} &= \boldsymbol{H}_{\mathrm{a}} + \mathrm{Dropout}(\boldsymbol{F}_{\mathrm{t}}). \end{aligned} \tag{17}$$

Here, $\mathrm{SiLU}(x) = x\sigma(x)$, σ is the sigmoid function, and $\odot$ denotes element-wise multiplication. The gate and feature projections have $\boldsymbol{W}_g, \boldsymbol{W}_u \in \mathbb{R}^{D\times D_{\mathrm{ff}}}$ and $\boldsymbol{b}_g, \boldsymbol{b}_u \in \mathbb{R}^{D_{\mathrm{ff}}}$. The output projection has $\boldsymbol{W}_d \in \mathbb{R}^{D_{\mathrm{ff}}\times D}$ and $\boldsymbol{b}_d \in \mathbb{R}^{D}$. Thus, SwiGLU includes both the gated product and the projection back to D. Dropout is applied to the gated product and to both residual branches during training, and is the identity at inference. Collecting the outputs for all sensors gives $\boldsymbol{H}_{\mathrm{t}} \in \mathbb{R}^{P\times N\times D}$, which enters the spatial block.

I ICSM Equations and Pseudocode

This appendix describes the computation of one ICSM block. Algorithm 1 summarizes the block given the raw-window statistics $\boldsymbol{u}_{1:N}$. Notation follows Section 4.4: $\boldsymbol{H} \in \mathbb{R}^{P\times N\times D}$ is the block input, P_{in} input patches precede P_{out} future tokens, M heads of width $d_h = D/M$, rank r, conditioning width d_c.

Normalization.

$$\boldsymbol{Z}_{p,n} = \mathrm{RMSNorm}(\boldsymbol{H}_{p,n}), \qquad \boldsymbol{Z}^{c}_{p,n} = \boldsymbol{Z}_{p,n} - \frac{1}{N}\sum_{n'=1}^{N}\boldsymbol{Z}_{p,n'}. \tag{18}$$

Context summary.

$$\boldsymbol{c}_n = \Big[\ \tfrac{1}{P_{\mathrm{in}}}\textstyle\sum_{p\le P_{\mathrm{in}}}\boldsymbol{Z}_{p,n};\ \ \boldsymbol{Z}_{P_{\mathrm{in}},n};\ \ \mathrm{std}_{p\le P_{\mathrm{in}}}(\boldsymbol{Z}_{p,n});\ \ \boldsymbol{u}_n\ \Big] \in \mathbb{R}^{3D+2}, \tag{19}$$

where $\boldsymbol{u}_n = [u_n^{\mathrm{res}}; u_n^{\mathrm{vol}}]$ is computed from the raw input window $\boldsymbol{X}$ of sensor n: u_n^{res} is the mean residual of the last w steps relative to the window's own time-of-day profile, divided by the standard deviation of the sensor's window, where the time-of-day profile of a step is a leave-one-out kernel-weighted average over the other steps of the same window with a Gaussian kernel in time-of-day distance. u_n^{vol} is the ratio of the standard deviation of first differences over the last w steps to that over the whole window. u_n^{vol} is informative only for longer windows.

Conditioning network.

$$\boldsymbol{\phi}_n = \boldsymbol{W}_2\,\mathrm{GELU}\big(\boldsymbol{W}_1\,\mathrm{RMSNorm}(\boldsymbol{c}_n)\big) \in \mathbb{R}^{d_c}, \qquad [\Gamma_n^{q};\ \Gamma_n^{k};\ \tilde{g}_n] = \boldsymbol{W}_{\psi}\boldsymbol{\phi}_n + \boldsymbol{b}_{\psi}, \tag{20}$$

with $\Gamma_n^{q}, \Gamma_n^{k} \in \mathbb{R}^{M\times r}$ and $\tilde{g}_n \in \mathbb{R}$; $\boldsymbol{W}_{\psi}$ and $\boldsymbol{b}_{\psi}$ are initialized to zero.

Base projections, low-rank increments, and clipping. For every head h,

$$\begin{aligned} \boldsymbol{q}^{0}_{\mathrm{s},p,n,h} &= \frac{\boldsymbol{W}_q^{h}\boldsymbol{Z}^{c}_{p,n}}{\|\boldsymbol{W}_q^{h}\boldsymbol{Z}^{c}_{p,n}\|_2}, \qquad \boldsymbol{k}^{0}_{\mathrm{s},p,n,h} = \frac{\boldsymbol{W}_k^{h}\boldsymbol{Z}^{c}_{p,n}}{\|\boldsymbol{W}_k^{h}\boldsymbol{Z}^{c}_{p,n}\|_2}, \qquad \boldsymbol{v}_{\mathrm{s},p,n,h} = \boldsymbol{W}_v^{h}\boldsymbol{Z}_{p,n}, \\ \Delta\boldsymbol{q}_{p,n,h} &= (\boldsymbol{B}_q^{h})^{\top}\big(\Gamma^{q}_{n,h} \odot \boldsymbol{A}_q^{h}\boldsymbol{Z}^{c}_{p,n}\big), \qquad \Delta\boldsymbol{k}_{p,n,h} = (\boldsymbol{B}_k^{h})^{\top}\big(\Gamma^{k}_{n,h} \odot \boldsymbol{A}_k^{h}\boldsymbol{Z}^{c}_{p,n}\big), \\ \Delta &\leftarrow \Delta\cdot\min\big(1,\ \delta_{\max}/\|\Delta\|_2\big) \quad \text{for each increment,} \end{aligned} \tag{21}$$

where $\boldsymbol{A}_q^h, \boldsymbol{A}_k^h \in \mathbb{R}^{r\times D}$ and $\boldsymbol{B}_q^h, \boldsymbol{B}_k^h \in \mathbb{R}^{r\times d_h}$ are shared across sensors and patches.

Modulated attention, gate, and residual.

$$\boldsymbol{q}_{\mathrm{s},p,n,h} = \boldsymbol{q}^0_{\mathrm{s},p,n,h} + \Delta\boldsymbol{q}_{p,n,h}, \qquad \boldsymbol{k}_{\mathrm{s},p,n,h} = \boldsymbol{k}^0_{\mathrm{s},p,n,h} + \Delta\boldsymbol{k}_{p,n,h},$$
$$\alpha_{p,h}(n,n') = \frac{\exp(s_h\, \boldsymbol{q}^\top_{\mathrm{s},p,n,h}\boldsymbol{k}_{\mathrm{s},p,n',h})}{\sum_{m=1}^{N}\exp(s_h\, \boldsymbol{q}^\top_{\mathrm{s},p,n,h}\boldsymbol{k}_{\mathrm{s},p,m,h})}, \qquad \boldsymbol{o}_{p,n,h} = \sum_{n'=1}^{N}\alpha_{p,h}(n,n')\,\boldsymbol{v}_{\mathrm{s},p,n',h}, \tag{22}$$
$$g_n = \sigma(\tilde{g}_n + b_0), \qquad \boldsymbol{H}'_{p,n} = \boldsymbol{H}_{p,n} + g_n\,\boldsymbol{W}_o\,[\boldsymbol{o}_{p,n,1};\ \ldots;\ \boldsymbol{o}_{p,n,M}].$$

The per-head scale s_h is a learned scalar, initialized to 10 and capped at 100, and plays the role of the softmax temperature scale of cosine attention.

Algorithm 1 One ICSM block.

Require: block input $\boldsymbol{H} \in \mathbb{R}^{P\times N\times D}$; raw window statistics $\boldsymbol{u}_{1:N}$; parameters $\boldsymbol{W}_q^h, \boldsymbol{W}_k^h, \boldsymbol{W}_v^h, \boldsymbol{W}_o, \boldsymbol{A}_q^h, \boldsymbol{A}_k^h, \boldsymbol{B}_q^h, \boldsymbol{B}_k^h, \boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_\psi, \boldsymbol{b}_\psi, s_h$; constants $\delta_{\max}, b_0$

$\boldsymbol{Z} \leftarrow \mathrm{RMSNorm}(\boldsymbol{H})$; $\boldsymbol{Z}^c \leftarrow \boldsymbol{Z} - \mathrm{mean}_n(\boldsymbol{Z})$ ▷ common-mode removal per patch
for $n = 1, \ldots, N$ **do**
  $\boldsymbol{c}_n \leftarrow [\mathrm{mean}_{p\le P_{\mathrm{in}}}\boldsymbol{Z}_{p,n};\ \boldsymbol{Z}_{P_{\mathrm{in}},n};\ \mathrm{std}_{p\le P_{\mathrm{in}}}\boldsymbol{Z}_{p,n};\ \boldsymbol{u}_n]$ ▷ input patches only
  $\boldsymbol{\phi}_n \leftarrow \boldsymbol{W}_2\,\mathrm{GELU}(\boldsymbol{W}_1\,\mathrm{RMSNorm}(\boldsymbol{c}_n))$
  $[\Gamma^q_n; \Gamma^k_n; \tilde{g}_n] \leftarrow \boldsymbol{W}_\psi\boldsymbol{\phi}_n + \boldsymbol{b}_\psi$; $g_n \leftarrow \sigma(\tilde{g}_n + b_0)$
end for
for each patch p and head h **do**
  $\boldsymbol{q}^0_{\mathrm{s}} \leftarrow \mathrm{norm}(\boldsymbol{W}_q^h\boldsymbol{Z}^c_{p,:})$; $\boldsymbol{k}^0_{\mathrm{s}} \leftarrow \mathrm{norm}(\boldsymbol{W}_k^h\boldsymbol{Z}^c_{p,:})$; $\boldsymbol{v}_{\mathrm{s}} \leftarrow \boldsymbol{W}_v^h\boldsymbol{Z}_{p,:}$
  $\Delta\boldsymbol{q} \leftarrow \mathrm{clip}_{\delta_{\max}}\big((\boldsymbol{B}_q^h)^\top(\Gamma^q_{:,h} \odot \boldsymbol{A}_q^h\boldsymbol{Z}^c_{p,:})\big)$; $\Delta\boldsymbol{k} \leftarrow \mathrm{clip}_{\delta_{\max}}\big((\boldsymbol{B}_k^h)^\top(\Gamma^k_{:,h} \odot \boldsymbol{A}_k^h\boldsymbol{Z}^c_{p,:})\big)$
  $\boldsymbol{q}_{\mathrm{s}} \leftarrow \boldsymbol{q}^0_{\mathrm{s}} + \Delta\boldsymbol{q}$; $\boldsymbol{k}_{\mathrm{s}} \leftarrow \boldsymbol{k}^0_{\mathrm{s}} + \Delta\boldsymbol{k}$
  $\alpha \leftarrow \mathrm{softmax}_{n'}(s_h\,\boldsymbol{q}_{\mathrm{s}}\boldsymbol{k}_{\mathrm{s}}^\top)$ ▷ $N\times N$ operator for this patch and head
  $\boldsymbol{o}_{p,:,h} \leftarrow \alpha\,\boldsymbol{v}_{\mathrm{s}}$
end for
$\boldsymbol{H}' \leftarrow \boldsymbol{H} + g \odot \boldsymbol{W}_o\,\mathrm{concat}_h(\boldsymbol{o}_{:,:,h})$ ▷ gated residual
return $\boldsymbol{H}'$