Evaluating Spatio-Temporal Forecasting Trade-offs Between Graph Neural Networks and Foundation Models

Ragini Gupta*, Naman Raina*, Bo Chen*, Li Chen´, Claudiu Danilov+, Josh Eckhardt+, Keyshla Bernard+, Klara Nahrstedt*
*University of Illinois Urbana-Champaign, USA, ´University of Louisiana at Lafayette, USA, +Boeing Research and Technology, USA
{raginig2, namanr2, boc2, klara}@illinois.edu, {li.chen}@louisiana.edu, {cdanilov, josh.d.eckhardt, keyshla.j.bernard}@boeing.com

Abstract

Modern IoT deployments for environmental sensing produce high volume spatiotemporal data to support downstream tasks such as forecasting, typically powered by machine learning models. While existing filtering and strategic deployment techniques optimize collected data volume at the edge, they overlook how variations in sampling frequencies and spatial coverage affect downstream model performance. In many forecasting models, incorporating data from additional sensors denoise predictions by providing broader spatial contexts. This interplay between sampling frequency, spatial coverage and different forecasting model architectures remain underexplored. This work presents a systematic study of forecasting models - classical models (VAR), neural networks (GRU, Transformer), spatio-temporal graph neural networks (STGNNs), and time series foundation models (TSFMs: Chronos Moirai, TimesFM) under varying spatial sensor nodes density and sampling intervals using real-world temperature data in a wireless sensor network. Our results show that STGNNs are effective when sensor deployments are sparse and sampling rate is moderate, leveraging spatial correlations via encoded graph structure to compensate for limited coverage. In contrast, TSFMs perform competitively at high frequencies but degrade when spatial coverage from neighboring sensors is reduced. Crucially, the multivariate TSFM Moirai outperforms all models by natively learning cross-sensor dependencies. These findings offer actionable insights for building efficient forecasting pipelines in spatio-temporal systems. All code for model configurations, training, dataset, and logs are open-sourced for reproducibility.[1]

CCS Concepts

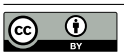
• **Information systems** → **Spatial-temporal systems**; • **Computing methodologies** → **Machine learning**.

Keywords

Spatio-temporal forecasting, Graph Neural Networks, Time Series Foundation Models

ACM Reference Format:
Ragini Gupta*, Naman Raina*, Bo Chen*, Li Chen´, Claudiu Danilov+, Josh Eckhardt+, Keyshla Bernard+, Klara Nahrstedt*, *University of Illinois Urbana-Champaign, USA, ´University of Louisiana at Lafayette, USA, +Boeing Research and Technology, USA, {raginig2, namanr2, boc2, klara}@illinois.edu, {li.chen}@louisiana.edu, {cdanilov, josh.d.eckhardt, keyshla.j.bernard}@boeing.com . 2025. Evaluating Spatio-Temporal Forecasting Trade-offs Between Graph Neural Networks and Foundation Models. In *The 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BUILDSYS '25), November 19–21, 2025, Golden, CO, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3736425.3771958

[1]https://github.com/UIUC-MONET-Projects/Benchmarking-Spatiotemporal-Forecast-Models



BUILDSYS '25, Golden, CO, USA

ACM ISBN 979-8-4007-1945-5/2025/11
https://doi.org/10.1145/3736425.3771958

1 Introduction

Urban IoT systems such as environmental monitoring platforms rely on short-term weather forecasting (2–6h ahead) techniques to support decision making in mobility planning and environmental control. Densely deployed meteorological sensors such as temperature, pressure or humidity in a wireless sensor network offer rich spatial and temporal coverage but raises questions on how sampling frequency and inter-sensor (node) correlations affect downstream forecasting, an aspect often overlooked by edge-side data acquisition methods like adaptive sampling and filtering [8]. Forecasting approaches range from statistical models (Autoregressive, Recurrent Neural Network-RNN) to modern architectures (Transformers, Spatio-Temporal Graph Neural Networks- STGNN), each with different sensitivities: Transformers excel at high-frequency data, while RNNs may overfit or underfit depending on resolution [23]. STGNNs explicitly model graph structures to capture spatiotemporal dependencies [4, 13]. Most recently, Time Series Foundation Models (TSFMs) like Chronos [1], Moirai [22], TimesFM [5] have emerged, leveraging large-scale pretraining to enable strong zero-shot forecasting across diverse domains. Unlike language, time-series data is highly heterogeneous, making forecasting complex in domains like temperature where dynamic, confounding variables are inherent. TSFMs eliminate ad-hoc modeling by providing a single pre-trained model that works across tasks with zero-shot inference or minimal fine-tuning, demonstrating strong single-sensor univariate forecasting [14], but their effectiveness in multi-sensor spatiotemporal settings remains an open question. This leads to a critical question: **How do explicit graph-based structures (STGNNs) compare against pretrained Time Series Foundation Models, for short-term temperature forecasting under different spatio-temporal conditions?** To answer this, we compare STGNNs with VAR, RNNs, Transformers, and two state of the art TSFM models —Moirai [22], Chronos [1], TimesFM [5]—using a 25-node IoT meterological dataset from New Mexico [3, 10]. We vary **(1)**: Temporal resolution through downsampling (5-60 minute intervals) to assess model sensitivity to sampling rates, and **(2)**: Number of nodes (8, 16, 25) to examine robustness across spatial densities. This setup enables comparison of graph vs. non-graph models under varied spatiotemporal conditions. Our findings show that STGNNs outperform classical and Transformer based models when spatial correlations are strong or data is sparse, while TSFMs perform competitively in uniform, high-frequency settings with low spatial variability. This highlights the need to align model choice with data topology and sensing strategy.

2 Related Works

Graph Neural Networks for time-series forecasting. Time-series forecasting progressed from statistical ARIMA to deep learning, with early sequence models such as Long Short-Term Memory (LSTM) [19] and Gated Recurrent Units (GRU) (an RNN variant) [6] captured temporal patterns but largely ignored spatial relationships. Spatio-Temporal GNNs (STGNNs) [21] model spatial–temporal dependencies by treating sensors as graph nodes, with graphs built from geographic proximity or cross-sensor similarity (modalities/meteorology). Refinements include STGCN [12] and Graph WaveNet [18], and later attention/Transformer-based STGNNs [24]. Despite these advances, such models rely heavily on training data and often generalize poorly to unseen settings.

Time-Series Foundation Models for forecasting. Inspired by the transformative success of Large Language Models (LLMs), Time-Series Foundation Models (TSFMs) have emerged as a promising path toward generalizable time-series analysis, demonstrating strong few-shot and zero-shot generalization across domains like cyber-physical systems [11], building energy analytics [14, 15], and human activity recognition [17]. These models, based on adapted transformer architectures, are pre-trained on large, diverse corpora of time-series data empowering them to perform core tasks like forecasting and anomaly detection. Contemporary TSFMs like MOMENT [9] and TimesFM [5] aim to generalize across datasets and modalities, but remain limited to a fixed set of tasks. Although these models support covariate inputs as exogenous variables, their integration methods such as flattening multivariate sequences through variate embeddings or using similarity-based attention, have failed to show consistent performance gains as the transformer encoder treats these covriates as extra input tokens alongside the past target history to form context representation, thus, failing to capture the inter-variable relationship. This limitation becomes critical in spatio-temporal environments such as urban sensor networks, where data heterogeneity across sensor types, locations, and sampling resolutions creates complex inter-dependencies that their design is ill-equipped to handle. In contrast, newer generation of true multivariate TSFMs, such as Moirai [22], are built on the core assumption of the transformer that all channels in a multivariate series are inherently correlated and interdependent, however, their performanc not studied. Prior work has examined TSFMs, STGNNs [4, 13], and sequential models [23] in isolation, leaving a critical need for a controlled, cross-paradigm comparison. We rigorously quantify the performance of TSFMs against spatio-temporal Graph Neural Networks, revealing their trade-offs under varied sampling rates and spatial node coverage (or node densities). This analysis establishes a critical empirical baseline, providing essential insight into the practical challenges of data heterogeneity and precisely defining the necessity for effective multi-variate integration in future TSFMs.

3 Overview of Time Series Forecasting Models

We group models into 3 categories: Classical, TSFMs, and STGNNs.

A) Classical baseline models: We include baseline models: Vector AutoRegression (VAR)—a multivariate AR model, Gated Recurrent Unit (GRU)—a sequential RNN variant, and Transformer [20]—which models temporal dependencies using self-attention.

B) Time Series Foundation Models (TSFMs): TSFMs are large pretrained models designed for zero-shot or transfer forecasting. We evaluate three TSFM models: Moirai, TimesFM, and Chronos. Table 3.1 summarizes their key attributes. Moirai is an encoder-only Transformer trained on 27B observations using mixture log-likelihood loss. It supports multivariate input and models cross-input dependencies via attention, using multi-resolution tokenization (at multi-time scales) over fixed-length patches of a sequence. In contrast, TimesFM is a decoder-only Transformer trained on 100B+ time points (including LOTSA dataset) across energy, weather, and traffic domains. It is univariate but incorporates covariates (auxiliary inputs) and models TS patches via residual blocks, creating tokens to forecast. Similarly, Chronos [1] is a univariate, pretrained TSFM that tokenizes continuous values into discrete bins and uses a transformer to predict future sequences autoregressively. Its Chronos-X[2] extension incorporates exogenous covariates through cross-attention mechanisms allowing the model to leverage external variables while maintaining strong zero-shot forecasting capabilities.

C) Spatio-Temporal Graph Neural Networks (STGNNs): STGNN models forecast time series by capturing temporal patterns and spatial dependencies using a graph structure, where an adjacency matrix encodes node interactions. STGNNs operate in two learning stages: (i) Temporal learning, where each node processes its historical data using sequential neural network models like Gated Recurrent Units (GRU) or Convolutional Neural Network (CNN) to capture local temporal patterns and (ii) Spatial aggregation, where nodes exchange information with neighbors via Graph Convolutions Network (GCN), with edge weights reflecting inter-node relation. Each node updates its state by combining local and neighbor features. We use 2 STGNNs: **(a) GRUGCN [7]-** first processes each node's input through GRU units to capture temporal patterns, then applies graph convolutions to aggregate spatial information from neighbor nodes. **(b) TGCN [16]-** applies a two-layer GCN to the raw input features at each time step to model spatial dependencies, and then uses a GRU to capture temporal patterns. Unlike GRUGCN, the GCN in T-GCN operates on input features rather than GRU outputs. This allows earlier incorporation of neighborhood information. The choice between them depends on whether spatial dependencies should inform temporal processing (TGCN) or follow it (GRUGCN).

Table 3.1: Time Series Foundation Models attributes

Attribute	Chronos (Amazon)	TimesFM (Google)	Moirai-Small (Salesforce)
Zero-shot Capability	✓	✓	✓
Multivariate Forecasting	✗ (univariate)	✗ (univariate, adapted)	✓(any-variate)
Covariate Support	✓(via ChronosX)	✓	✓
Forecasting Type	Probabilistic (quantile)	Point-wise	Probabilistic
Irregular TS Handling	✗	✗	✓
Pretraining Corpus Size	4B time-points	100B time-points	27B observations (LOTSA)
Primary Domains	General purpose	Finance, Climate	Finance, Climate, Transport
Architecture	Encoder–Decoder (T5)	Decoder-only Transformer	Encoder-only Transformer
Tokenization Strategy	Quantile-based binning	Patch-wise (fixed)	Patching + multi-resolution
Temporal Shift Robustness	Moderate	✗	✓(via normalization)
Computational Cost	Quadratic (self-attn)	Quadratic (self-attn)	Quadratic (self-attn)

4 Methodology

We evaluate forecasting performance on meteorological sensor network data, where each model predicts future temperature readings given historical inputs. Our framework maintains fixed durations (8h context window W, 4h prediction horizon P) while varying: (1) sampling rates $f_s \in \{5, 15, 30, 45, 60\}$ minutes, and (2) network density with node counts $K \in \{8, 16, 25\}$ representing progressively larger spatial coverage - 8 nodes form the most tightly clustered subgroup, 16 nodes cover intermediate proximity, and 25 nodes encompass the full network. We calculate the context length in samples as $C = 60 \times W / f_s = 480 / f_s$, yielding a 96 context samples at the finest

5-minute resolution ($f_s = 5$), decreasing to 8 samples at the coarsest 60-minute resolution ($f_s = 60$). The prediction length follows similarly as $H = 60 \times P/f_s = 240/f_s$, producing forecast sequences ranging from 48 steps (5-minute) down to 4 steps (60-minute) per sensor. This creates a continuous temporal resolution spectrum from 5-minute to 60-minute sampling. Models process spatial relationships differently: Moirai handles all nodes as multivariate inputs, while TimesFM and Chronos treat each sensor as an independent univariate series with forecasts aggregated via ensemble averaging. STGNNs construct graphs using absolute Pearson correlations ($|\rho_{ij}|$) to capture micro-climatic patterns beyond physical proximity. This design enables comprehensive evaluation across temporal granularities, spatial coverage densities across model types.

Experimental design. We used the Internet-of-Battlefield Things (IoBT) data set [10], containing meteorological data (temperature, humidity, rainfall, etc.) from 25 sensor towers covering a 10km x 10km area in New Mexico, sampled every 5 minutes over 9 days. The high sampling frequency provides substantial data volume for short-term forecasting evaluation, with each sensor contributing $9 \times 24 \times 12 = 2,592$ samples, totaling 64,800 time steps across all sensors. This volume is well-suited for short-term forecasting where models rely on recent temporal contexts (predicting 4H from 8H histories), and the effective sample size is further amplified through sliding window processing, generating thousands of training sequences for hour-ahead prediction. The dataset is split into 5 training, 2 validation, and 2 testing days, consistent with standard practices for short-term forecasting evaluation [4, 5]. Temperature forecasting performance is reported as MAE and RMSE averaged across nodes. The sensor network layout and spatial node coverage for $K \in \{8, 16, 25\}$ nodes are provided in Appendix A.

Enhancing TimesFM and Chronos with spatial interdependency modeling. To capture cross-sensor dependencies, we extend Chronos X and TimesFM with a post-hoc ensemble similarity method. For each target sensor, we select its top-k most correlated neighbors (using Pearson similarity) and blend the target's forecast with neighbors' predictions using similarity-based weights, effectively treating neighboring forecasts as exogenous covariates. This adaptive blending exploits spatial relationships to improve robustness under heterogeneous sensor conditions while preserving both models' zero-/few-shot behavior.

4.1 Model Building and Implementation

4.1.1 Baseline models and TSFMs. We evaluate 3 classical models as baselines— VAR, GRU, and Transformer— using the window and horizon configurations aforementioned. VAR operates directly on raw multivariate data without training, whereas GRU and Transformer employ trained 2-layer encoders (hidden size=64) on fixed windows to match zero-shot evaluation conditions. For TSFMs, we use Moirai, TimesFM, and Chronos, all evaluated in zero-shot inference mode without any fine-tuning. Moirai processes all nodes jointly as a $K \times W$ matrix, using self-attention to capture cross-sensor dependencies and output probabilistic forecasts. In contrast, TimesFM and Chronos employ an ensemble similarity approach: each node is forecasted independently using its historical context, then predictions are blended via correlation-weighted averaging of the top-k most similar neighbor forecasts (with $k = 3$ and $\alpha = 0.6$ weight for the target node).

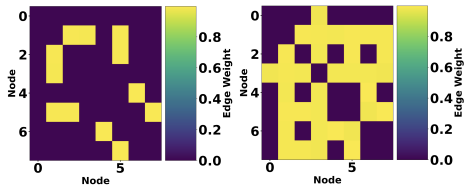


Figure 4.1: Adjacency matrix heatmaps for $p = 20\%$ (left) and $p = 60\%$ (right) in STGNNs.

4.1.2 STGNNs. Graph construction is critical in STGNNs, as it defines the structure ($\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ represent the sets of sensor nodes and weighted connections between them, respectively). The goal is to model relationships between nodes (sensors) based on their time series data, accounting for spatial correlations due to shared micro-climate factors (e.g., altitude, local events) and temporal variations due to multi-frequency sampling (e.g., 5-min vs. 60-min intervals). In order to achieve a head-on-head comparison with other forecasting models, we aggregate raw time series data sequences from sensors onto a common timeline, build $\mathcal{G}$ and then run a standard *window* $\rightarrow$ *horizon* forecasting pipeline. We construct the graphs using a Pearson correlation for building an adjacency matrix under varying redundancy levels (p%). Here, redundancy is the degree of shared information between sensors due to spatial correlations. Controlling redundancy balances noise reduction (via correlated signals) and information diversity (via unique signals), optimizing the graph's ability to capture meaningful spatial relationships without over-fitting or introducing overly sparse connections missing important details. To construct the adjacency matrix, for N sensors with time series $X_i = [X_{1,i}, ..., X_{T,i}]^\top$, we compute the absolute Pearson correlation $\rho_{ij} = |\text{corr}(X_i, X_j)|$ between each pair (i, j). This produces a symmetric $N \times N$ matrix where each entry quantifies the strength of data correlation between sensor pairs. To optimize the graph's predictive utility, we threshold these correlations by retaining only the top p% of connections. The threshold θ is adaptively determined based on the desired redundancy level p:

$$\theta = \begin{cases} \max \mathcal{U} & \text{if } p = 0\% \text{ (no edges)} \\ \text{percentile}(\mathcal{U}, 100 - p) & \text{if } 0 < p < 100\% \\ \min \mathcal{U} & \text{if } p = 100\% \text{ (full connectivity)} \end{cases} \quad (1)$$

where $\mathcal{U} = \{|\rho_{ij}| : i < j\}$ is the set of unique correlations and p is the desired redundancy level (working example provided in Appendix). This approach preserves the most significant micro-climatic relationships while avoiding over-connection, as visualized by heatmaps for adjacency matrices in Fig. 1 for $p = 20\%$ and $p = 60\%$ redundancy levels, showing how increasing redundancy levels gives denser graph network connectivity.

4.1.3 Implementation details. GRUGCN and TGCN are implemented using the Torch Spatiotemporal (TSL) framework with PyTorch Lightning. Both use one GRU and one Graph Convolutional Network (GCN) layer. GRUGCN processes input tensors of shape $(B, W, N, 1)$, where B is batch size, W is window length, and N is the number of nodes (same as K). Models are trained with the Adam optimizer (learning rate 1×10^{-3}) and a batch size of 64.

5 Experimental Results

A. Baseline models and TSFMs. Forecast performance for baseline models and TSFMs appears in Tables 2 and 3, respectively.

(1) Impact of sampling rate (temporal granularity). VAR excels at slow sampling (15–60 min) because linear cross-series structure dominates, but adds little at 5 min where dense temporal signals already suffice. GRU worsens at mid-range (15–45 min) as it struggles to trade off temporal vs. spatial cues, yet improves at 60 min when spatial correlations become clearer. Transformer peaks at the extremes—5 min (rich temporal detail) and 60 min (strong global patterns)—but weakens in the mid-range where neither signal is dominant. When compared to TSFMs, the advantages of pretrained architectures become clear. At high frequency (5 min), Moirai moderately outperforms both baselines and other TSFMs, while at low frequency (60 min) its strength is pronounced (e.g., MAE = 0.93 with 8 nodes vs. 2.98 for GRU and 2.53 for Transformer), showing the benefit of multi-scale temporal representations that exploit long-range dependencies when temporal data is sparse. TimesFM degrades at coarser rates (e.g., RMSE $\geq$ 25 at 60 min/8 nodes) because its autoregressive token generation accumulates errors without global context. In contrast, Chronos is consistently more robust—e.g., MAE = 3.810 at 15 min/16 nodes (65% lower than TimesFM's 11.034) and remains stable at coarser rates. Post-hoc ensemble similarity with covariates narrows this gap where it reduces TimesFM errors by 11–12% at 15–30 min/16 nodes but yields only 1.4% reduction for Chronos. This is attributed to TimesFM's decoder-only architecture that generates tokens autoregressively, which struggles to capture extended temporal dependencies leading to performance degradation at longer horizons (30-60 minutes). Therefore, neighbor-based smoothing corrects these predictions significantly. Conversely, Chronos's predictions are already spatially coherent (given it's encoder-decoder tokenization). It is worth noting, while ensembling can stabilize TimesFM, the most sampling-robust performance comes from models that natively learn spatial dependencies (in their core transformer architecture) with Moirai. TSFM forecast graphs at the lowest (5 min) and highest (60 min) sampling rates are shown in Appendix A.

(2) Effect of cluster size (spatial node coverage). Model rankings shift with the number of nodes. For small clusters (8 nodes), lightweight baselines can still be competitive; for example, at 15-min/8 nodes a plain Transformer slightly outperforms GRU and VAR (MAE 2.02 vs. 2.51 and 2.38), showing that attention can capture short-range structure even in small systems. As clusters grow to 25 nodes, architectures that better exploit cross-series interactions gain an edge, wherein the Transformer improves or holds steady (e.g., 5-min MAE $\sim$ 2.15 $\rightarrow\sim$ 2.07 from 8 $\rightarrow$ 25 nodes), while GRU and VAR either plateaus or worsen (GRU 2.16 $\rightarrow$ 2.46; VAR 2.48 $\rightarrow$ 2.57). A notable exception is VAR's strong performance at 45-min/25 nodes (MAE 1.95), where shared linear structure across many series is efficiently exploited. When compared with TSFMs, *Moirai* is the strongest overall and even improves with scale (e.g., 5-min MAE 2.04 $\rightarrow$ 1.85 from 8 $\rightarrow$ 25 nodes). Its a multi-resolution architecture pretrained on diverse LOTSA corpa benefits model predictions with increased node count adding rich spatial context. Its masked encoder excels at coarser sampling (30–60 min), where temporal signals are sparse, by extracting shared spatial patterns across sensors. In contrast, *TimesFM* struggles on longer horizons (30–60 min) and is largely insensitive to node count, because its decoder-only, channel-independent architecture processes series separately and fuses late, limiting learned spatial interactions and extended temporal dependencies. Chronos is largely insensitive to node count where it is often slightly better at 25 nodes (e.g., improvements at 30/45/60-min). Ensemble similarity adds only 1–2% further gain unlike that for TimesFM

Table 5.1: Forecast performance (MAE / RMSE) for baseline models.

Sampling Rate (f_s)	Nodes (K)	GRU	Transf.	VAR
5	8	2.16/3.10	2.20/3.41	2.48/3.48
5	16	2.35/3.59	2.20/3.46	2.53/3.44
5	25	2.46/3.52	2.15/3.31	2.57/3.45
15	8	2.51/3.72	2.02/3.12	2.38/3.29
15	16	2.23/3.40	2.15/3.36	2.15/3.04
15	25	2.31/3.39	2.07/3.31	2.14/3.07
30	8	2.18/3.31	2.15/3.25	2.60/3.48
30	16	2.12/3.15	2.13/3.30	2.12/2.97
30	25	2.11/3.19	2.15/3.10	2.07/2.96
45	8	2.16/3.14	2.12/2.96	2.59/3.43
45	16	2.06/2.96	2.61/3.57	2.02/2.79
45	25	2.14/3.11	2.09/2.92	1.95/2.74
60	8	2.98/4.00	2.53/3.18	3.37/4.21
60	16	2.71/3.35	2.64/3.53	2.53/3.33
60	25	2.71/3.41	2.08/2.82	2.23/2.91

Note: Values are in °C.

Table 5.2: Forecast performance (MAE/RMSE) for TSFMs.

f_s	K	Chronos	Chronos (w/ Covariates)	TimesFM	TimesFM (w/ Covariates)	Moirai
5	8	3.842 / 4.420	3.768 / 4.186	6.389 / 8.663	3.753 / 5.054	2.039 / 2.955
5	16	3.731 / 4.331	3.728 / 4.280	6.859 / 9.271	4.301 / 5.735	1.868 / 2.304
5	25	3.437 / 4.004	3.413 / 3.953	6.957 / 9.441	5.143 / 6.506	1.854 / 2.308
15	8	3.699 / 4.208	3.768 / 4.186	9.348 / 11.276	6.776 / 8.594	1.570 / 2.079
15	16	3.810 / 4.376	3.756 / 4.247	11.034 / 13.272	9.694 / 11.570	1.475 / 1.730
15	25	3.446 / 3.996	3.400 / 3.892	9.816 / 12.118	8.814 / 10.662	1.743 / 2.265
30	8	3.507 / 4.121	3.565 / 4.053	22.931 / 24.410	17.762 / 20.022	1.384 / 1.772
30	16	3.566 / 4.129	3.503 / 3.992	22.445 / 23.772	20.009 / 21.622	1.386 / 1.765
30	25	3.322 / 3.865	3.275 / 3.770	21.038 / 22.961	19.134 / 20.977	1.393 / 1.725
45	8	3.159 / 3.781	3.046 / 3.614	8.224 / 10.207	5.991 / 7.066	0.861 / 1.130
45	16	3.173 / 3.762	3.105 / 3.641	7.788 / 9.473	6.669 / 7.603	1.221 / 1.424
45	25	2.975 / 3.581	2.940 / 3.501	7.301 / 8.998	6.455 / 7.517	0.957 / 1.252
60	8	4.042 / 4.505	4.145 / 4.478	22.382 / 25.823	22.051 / 25.513	0.932 / 1.196
60	16	4.154 / 4.581	4.273 / 4.611	21.468 / 25.045	21.111 / 24.592	0.862 / 1.263
60	25	3.922 / 4.405	3.987 / 4.410	21.587 / 25.146	21.354 / 24.802	0.881 / 1.168

Note: Values are in °C.

B. Performance analysis of STGNNs. Table 4 reports STGNN performance along two dimensions: (i) the effect of graph redundancy among inter-dependent nodes, and (ii) the interplay between sampling rate and node coverage (number of nodes). Both GRUGCN and T-GCN show non-linear sensitivity to graph redundancy, with optimal performance at 20–60% connectivity. Both GRUGCN and T-GCN show a non-linear response to graph redundancy because of a classic bias–variance / oversmoothing trade-off in message passing. With too few edges, the model underuses spatial signal (high variance); with too many edges (100% redundancy), repeated aggregation amplifies noise and oversmooths node features, making neighbors indistinguishable and introducing spurious correlations—hence the large MAE increases (GRUGCN +32%, T-GCN +68%). The sweet spot (with 20–60% connectivity) retains informative neighbors while limiting noise propagation, which is why GRUGCN hits MAE 2.088 at 30min/8 nodes (60%) and T-GCN peaks at MAE 1.976 at 45min/25 nodes (20%). A 16-node graph is consistently robust where it balances spatial coverage with manageable redundancy so the receptive field is rich but not saturated. GRUGCN, with its temporal-first design, performs best at mid-range sampling rates (15–45 min), where there is enough temporal resolution for the GRU to extract sequential patterns before spatial aggregation; at coarse rates (60 min), the weaker temporal signal reduces its effectiveness and makes it more sensitive to graph redundancy. In contrast, T-GCN's spatial-first

Table 5.3: Performance evaluation for STGNNs (GRUGCN and TGCN)

Hyperparameters		GRUGCN										TGCN									
Red.(p)	Nodes(K)	5 min		15 min		30 min		45 min		60 min		5 min		15 min		30 min		45 min		60 min	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
0%																					
	8	2.686	3.748	2.130	3.015	2.159	3.019	2.469	3.279	2.673	3.482	2.568	3.700	2.289	3.505	2.008	2.975	2.013	2.848	3.246	4.156
	16	2.254	3.424	2.128	2.916	2.278	3.109	2.204	3.000	2.749	3.676	2.684	3.700	1.829	2.638	2.171	2.992	2.164	2.899	3.033	3.881
	25	2.343	3.461	2.309	3.196	2.283	3.020	2.306	3.075	2.886	3.590	2.646	3.676	2.114	3.281	2.236	3.004	2.274	2.898	3.168	4.048
20%																					
	8	2.328	3.442	2.209	3.069	2.296	3.075	2.262	2.987	2.668	3.306	2.733	3.753	2.415	3.697	2.023	3.017	2.004	2.793	2.890	3.773
	16	2.290	3.504	2.333	3.250	2.144	2.899	2.253	2.987	2.519	3.227	2.388	3.601	1.994	2.919	2.113	2.951	2.056	2.768	2.351	3.030
	25	2.658	3.600	2.218	3.105	2.253	3.086	2.308	3.117	2.528	3.263	2.297	3.259	2.038	3.104	2.201	3.117	1.976	2.704	2.784	3.676
60%																					
	8	2.614	3.653	2.366	3.244	2.088	2.889	2.286	3.132	2.803	3.578	2.801	3.823	2.191	3.279	2.125	2.904	2.286	3.035	3.185	4.085
	16	2.579	3.585	2.089	2.993	2.064	2.802	2.168	2.934	2.582	3.279	2.701	3.693	2.082	3.403	1.956	2.792	2.136	2.867	2.969	3.863
	25	2.557	3.461	2.014	2.725	2.144	2.873	2.212	2.896	2.479	3.165	2.665	3.623	2.037	3.008	2.017	2.722	2.209	2.842	3.022	3.830
100%																					
	8	2.758	3.770	2.244	3.117	2.404	3.272	2.046	2.752	2.585	3.211	2.679	3.727	2.255	3.601	2.071	3.027	2.180	2.902	2.952	3.818
	16	2.601	3.558	2.082	2.948	2.193	2.993	2.175	2.961	2.441	3.186	2.687	3.672	1.970	2.871	2.335	3.451	2.233	2.992	2.470	3.264
	25	2.684	3.739	2.111	3.013	2.192	2.999	2.019	2.707	2.373	3.018	2.466	3.758	1.844	2.623	2.080	2.867	2.165	2.872	3.312	4.157

architecture excels at the extremes: at 5 min, early spatial aggregation smooths noisy fine-grained fluctuations, while at 60 min it compensates for sparse temporal context by exploiting inter-node correlations. Redundancy further interacts with graph size—helping smaller graphs (8 nodes) by enriching limited spatial input, but leading to oversmoothing and diminishing returns in dense graphs (25 nodes). Overall, performance is maximized when the model's processing order (temporal-first vs. spatial-first) and graph connectivity are aligned with the temporal granularity of the data.

When comparing across model classes, Moirai delivers the best sampling-robust performance, even in zero-shot settings, due to its any-variate attention mechanism that helps modeling multivariate dependencies without model fine-training or extra context via covariates. In terms of univariate TSFMs tested under zero-shot setting, Chronos is the most reliable baseline. Post-hoc correlation-weighted ensembling with neighboring sensors can move TimesFM closer to Chronos, but gains are modest and architecture-dependent. STGNNs can surpass univariate TSFMs when redundancy is tuned and GNN design is aligned with sampling rates and node coverage. Moreoever, Chronos-X and TimesFM with covariates show only marginal improvements over their base versions, underscoring that simple covariate injection is less effective than Moirai.

6 Discussion

A. No-one-size fits all- model performance is conditional. Each model class excels under specific conditions. STGNNs (GRUGCN, T-GCN perform well at mid-to-high sampling rates and moderate redundancy. They leverage spatial graphs effectively, especially with 16 nodes where other models do not perform well (is this right). However, at low sampling or extreme redundancy, they can overfit or oversmooth. VAR model improves with more sensor nodes by capturing linear correlations but cannot model nonlinear patterns. TSFMs (e.g., Moirai, TimesFM) excel on dense, high-frequency, uniform data but struggle with low spatial diversity; STGNNs are stronger in sparse or highly correlated deployments.

B. Redundancy and graph structure must be tuned, not maximized. In case of STGNN, we observe that moderate spatial redundancy (40-60%) optimizes STGNN performance, while extreme sparsity (0-20%) or density (80-100%) degrades accuracy through underfitting or oversmoothing. This effect is most pronounced in small networks (8 nodes), where redundancy compensates for limited spatial coverage, but becomes counterproductive in large networks (25 nodes) where excessive connections dilute local patterns. The architectural differences between GRUGCN and T-GCN further reveal that optimal redundancy depends on model design - GRUGCN's temporal-first approach benefits from moderate redundancy at medium sampling rates, while T-GCN's spatial-first architecture struggles with sparse temporal signals at low frequencies. These findings emphasize that effective graph construction requires balancing connectivity with specificity. Neither maximally sparse nor fully dense graphs consistently outperform; instead, there exists a sweet spot in redundancy, typically around 40–60%, that must be carefully matched to both network size and model architecture to achieve optimal performance.

C. TSFMs show promise but need contextual grounding. Moirai's superior forecasting performance even under zero-shot settings, stems from its native multivariate design. Its any-variate attention mechanism flattens all sensor data into a single sequence, processed simultaneously by a masked encoder. This allows the model to directly learn cross-sensor dependencies through self-attention, using rotary position embeddings to preserve the temporal order of patches (of time series sequence) within this flattened sequence. This integrated approach to spatiotemporal learning is fundamentally more powerful than the late-stage covariate fusion or post-hoc ensembling used by univariate models, which merely append spatial context rather than learning it. Going forward, univariate TSFMs should become context-aware either by accepting natural-language prompts that describe situational factors or by adding architectural support for true multivariate inputs that learn cross time-series relationship.

7 Conclusion

Multivariate TSFMs achieve the best spatio-temporal forecasts by natively modeling cross-sensor relationships, surpassing GNNs. While contextual covariates offer slight gains for univariate TSFMs, well-tuned STGNNs still lead, highlighting the need for architectural improvements towards multivariate, context-aware foundation models.

Acknowledgment

This work was supported by The Boeing Company under the University Master Research Agreement #2021-STU-PA-404.

References

[1] 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. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815* (2024).

[2] Sebastian Pineda Arango, Pedro Mercado, Shubham Kapoor, Abdul Fatir Ansari, Lorenzo Stella, Huibin Shen, Hugo Senetaire, Caner Turkmen, Oleksandr Shchur, Danielle C. Maddix, Michael Bohlke-Schneider, Yuyang Wang, and Syama Sundar Rangapuram. 2025. ChronosX: Adapting Pretrained Time Series Models with Exogenous Variables. arXiv:2503.12107 [cs.LG] https://arxiv.org/abs/2503.12107

[3] Manish Bhurtel and Danda B. Rawat. 2025. FL-MBS: Federated Learning and Mantissa Bits Steganography for Resource Constrained Autonomous Uncrewed Vehicles. *IEEE Trans. Aerospace Electron. Systems* 61, 4 (2025), 8939–8952. doi:10.1109/TAES.2025.3544593

[4] W. Chuyuan, P. Dechang, P. Mingtian, and Z. Haopeng. 2023. Short-term load forecasting using spatial-temporal embedding graph neural network. *Electric Power Systems Research* 225 (2023), 109873.

[5] A. Das, W. Kong, R. Sen, and Y. Zhou. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=jn2iTJas6h

[6] Jianfei Gao and Bruno Ribeiro. 2022. On the Equivalence Between Temporal and Static Equivariant Graph Representations. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 7052–7076. https://proceedings.mlr.press/v162/gao22e.html

[7] J. Gao and B. Ribeiro. 2022. On the equivalence between temporal and static equivariant graph representations. In *International Conference on Machine Learning*. PMLR, 7052–7076.

[8] Dimitrios Giouroukis, Alexander Dadiani, Jonas Traub, Steffen Zeuch, and Volker Markl. 2020. A survey of adaptive sampling and filtering algorithms for the internet of things. In *Proceedings of the 14th ACM International Conference on Distributed and Event-Based Systems* (Montreal, Quebec, Canada) *(DEBS '20)*. Association for Computing Machinery, New York, NY, USA, 27–38. doi:10.1145/3401025.3403777

[9] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. MOMENT: a family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org, Article 642, 38 pages.

[10] R. Gupta, K. Nahrstedt, N. Suri, and J. Smith. 2021. SVAD: End-to-End Sensory Data Analysis for IoBT-Driven Platforms. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*. 903–908.

[11] Liying Han, Gaofeng Dong, Xiaomin Ouyang, Lance Kaplan, Federico Cerutti, and Mani Srivastava. 2025. Toward Foundation Models for Online Complex Event Detection in CPS-IoT: A Case Study. In *Proceedings of the 2nd International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things* (Irvine, CA, USA) *(FMSys)*. Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/3722565.3727198

[12] Nantian Huang, Shengyuan Wang, Rijun Wang, Guowei Cai, Yang Liu, and Qianbin Dai. 2023. Gated spatial-temporal graph neural network based short-term load forecasting for wide-area multiple buses. *International Journal of Electrical Power & Energy Systems* 145 (2023), 108651. doi:10.1016/j.ijepes.2022.108651

[13] G. Jin, Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, and Y. Zheng. 2023. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *IEEE Trans. on knowledge and data engineering* 36, 10 (2023), 5388–5408.

[14] O. B Mulayim, P. Quan, L. Han, X. Ouyang, D. Hong, M. Bergés, and M. Srivastava. 2024. Are Time Series Foundation Models Ready to Revolutionize Predictive Building Analytics?. In *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (Hangzhou, China) *(BuildSys '24)*. ACM, 169–173.

[15] Ozan Baris Mulayim, Pengrui Quan, Liying Han, Xiaomin Ouyang, Dezhi Hong, Mario Bergés, and Mani Srivastava. 2025. Can Time-Series Foundation Models Perform Building Energy Management Tasks? arXiv:2506.11250 [cs.LG] https://arxiv.org/abs/2506.11250

[16] H. Nantian, W. Shengyuan, W. Rijun, C. Guowei, L. Yang, and Q. Dai. 2023. Gated spatial-temporal graph neural network based short-term load forecasting for wide-area multiple buses. *International Journal of Electrical Power & Energy Systems* 145 (2023), 108651.

[17] Iris Nguyen, Liying Han, Burke Dambly, Alireza Kazemi, Marina Kogan, Cory Inman, Mani Srivastava, and Luis Garcia. 2025. *Detecting Context Shifts in the Human Experience Using Multimodal Foundation Models.* Association for Computing Machinery, New York, NY, USA, 620–621. https://doi.org/10.1145/3715014.3724037

[18] Arian Prabowo, Wei Shao, Hao Xue, Piotr Koniusz, and Flora D. Salim. 2023. Because Every Sensor Is Unique, so Is Every Pair: Handling Dynamicity in Traffic Forecasting. In *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation* (San Antonio, TX, USA) *(IoTDI '23)*. Association for Computing Machinery, New York, NY, USA, 93–104. doi:10.1145/3576842.3582362

[19] Siddhi Jayesh Tandel and Suja Sreejith Panickar. 2025. Time Series-Based Weather Forecasting System Using LSTM. In *Trends in Sustainable Computing and Machine Intelligence*, Surekha Lanka, Antonio Sarasa Cabezuelo, and Chandrasekar Vuppalapati (Eds.). Springer Nature Singapore, Singapore, 229–241.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is All you Need. In *NeurIPS*, Vol. 30.

[21] Fei Wang, Dawei Lin, Baojun Chen, Guodong Jing, Yi Geng, Xudong Ge, Daoming Wei, and Ning Zhang. 2025. HyMePre: A Spatial–Temporal Pretraining Framework with Hypergraph Neural Networks for Short-Term Weather Forecasting. *Applied Sciences* 15, 15 (2025). doi:10.3390/app15158324

[22] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. 2024. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*.

[23] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. 2021. A Transformer-based Framework for Multivariate Time Series Representation Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Singapore) *(KDD '21)*. ACM, 2114–2124.

[24] Hongshan Zhao, Yuchen Wu, Libo Ma, and Sichao Pan. 2023. Spatial and Temporal Attention-Enabled Transformer Network for Multivariate Short-Term Residential Load Forecasting. *IEEE Transactions on Instrumentation and Measurement* 72 (2023), 1–11. doi:10.1109/TIM.2023.3305655

Appendix

A Sensor layout

Sensors in our IoT network record temperature, humidity, rainfall, and pressure at fixed sampling rates, producing timestamped time series per sensor. Multiple sensors yield multivariate, spatiotemporal data. Figure A.1 shows the geographical topology of the sensors used in this experiment, whereas Figure A.2 is important in motivating the key concept of redundancy where we see the overlap of temperature data as recorded by two colocated (belonging to the same cluster) sensors.

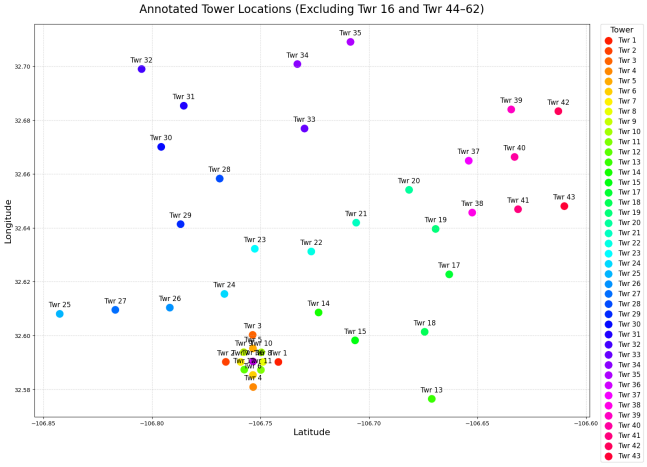


Figure A.1: Annotated tower locations (lat–lon).

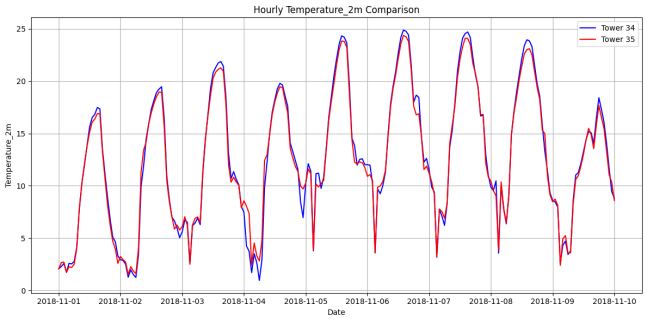
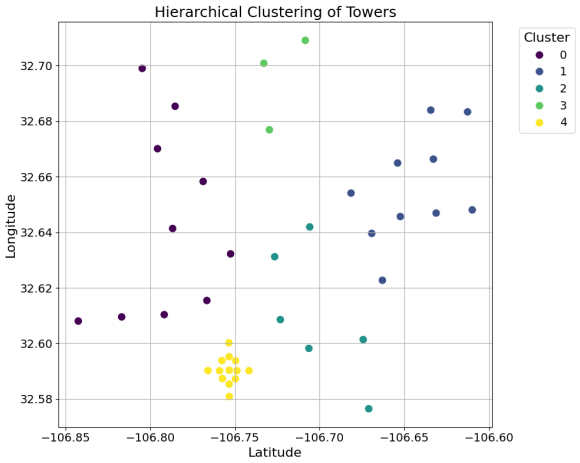


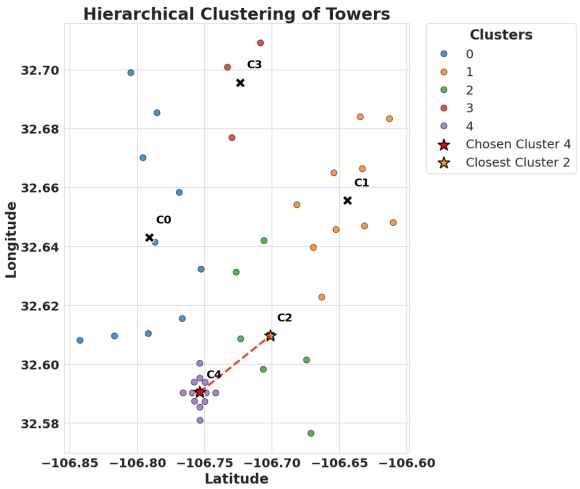
Figure A.2: Hourly temperature overlap for colocated Towers 34 and 35.

B Sensor selection for spatial node coverage

To select representative sensor subsets from the full 25-node network, we employ agglomerative clustering to ensure geographic representativeness. We partition all sensor locations into $k = 5$ clusters (treating each cluster as a Region of Interest- ROI), then select sensors for our subsets ($K = 8, 16, 25$) through a systematic expansion process. Starting with the $K = 8$ subset, we choose sensors nearest to cluster centroids supplemented with geographically dispersed nodes. We then expand to $K = 16$ and $K = 25$ by iteratively merging neighboring clusters and incorporating their centroid sensors, maintaining representative spatial coverage while varying node density to evaluate model performance under different spatial configurations.

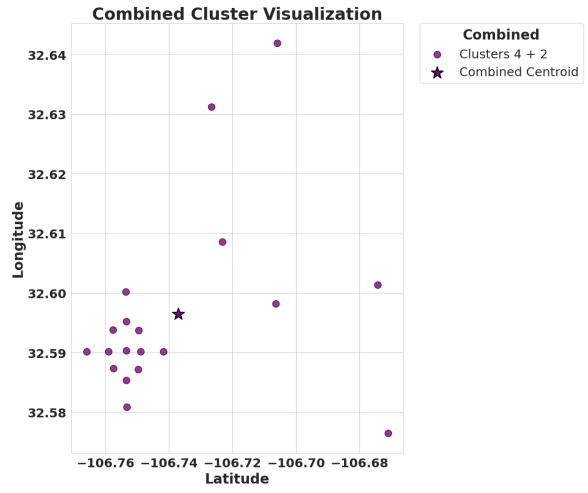


(a) k=5 spatial clusters (colors).

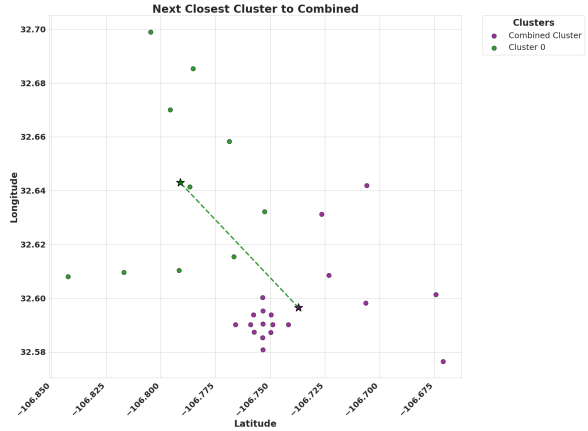


(b) Cluster centroids; ROI's nearest neighbor.

Figure B.1: (a,b) Spatial clustering and nearest-centroid selection.



(a) Merge ROI with nearest sensor tower (★ = merged centroid).



(b) Next closest cluster to the merged centroid.

Figure B.2: (c,d) Iterative merging of clusters by centroid distance.

C A working example for modeling spatio-temporal dependencies in sensor network using GNNs

To demonstrate how a Graph Neural Network (GNN) models spatial dependencies among colocated sensors, we construct a small example using a 5×5 Pearson correlation matrix.

Step 1: Pairwise correlation matrix

We begin with a simple correlation matrix ρ_{ij} representing the Pearson correlation coefficients between five sensors measuring temperature:

$$\rho_{ij} = \begin{pmatrix} 1.00 & 0.85 & 0.78 & 0.62 & 0.55 \\ 0.85 & 1.00 & 0.73 & 0.51 & 0.48 \\ 0.78 & 0.73 & 1.00 & 0.66 & 0.59 \\ 0.62 & 0.51 & 0.66 & 1.00 & 0.47 \\ 0.55 & 0.48 & 0.59 & 0.47 & 1.00 \end{pmatrix}$$

Here, the diagonal entries ($\rho_{ii} = 1$) represent perfect self-correlation and are ignored when constructing the adjacency matrix. The matrix is symmetric, so we only need to consider the upper-triangular values ($i < j$) for thresholding. Extracting the upper triangle (excluding diagonal) yields:

$$\mathcal{R} = \{0.85,\ 0.78,\ 0.62,\ 0.55,\ 0.73,\ 0.51,\ 0.48,\ 0.66,\ 0.59,\ 0.47\},$$

$$\mathcal{R}_{\text{sorted}} = \{0.47,\ 0.48,\ 0.51,\ 0.55,\ 0.59,\ 0.62,\ 0.66,\ 0.73,\ 0.78,\ 0.85\}.$$

Step 2: Thresholding by redundancy level

To control redundancy, we retain only the top $p\%$ of strongest correlations. This determines how densely connected the graph is.

For Redundancy $\rho = 80\%$. This implies the graph density to retain the top 80% of correlations, i.e., the eight largest out of 10 values. The threshold becomes:

$$\theta_{80} = \text{2nd-smallest value} = 0.48.$$

Thus:

$$A_{ij}^{(80\%)} = \begin{cases} \rho_{ij}, & \text{if } \rho_{ij} \geq 0.48, \\ 0, & \text{otherwise.} \end{cases}$$

The resulting adjacency matrix is shown in Table C.1.

Table C.1: Adjacency at 80% redundancy ($\theta = 0.48$) based on Pearson correlation.

	1	2	3	4	5
1	0	0.85	0.78	0.62	0.55
2	0.85	0	0.73	0.51	0.48
3	0.78	0.73	0	0.66	0.59
4	0.62	0.51	0.66	0	0.47
5	0.55	0.48	0.59	0.47	0

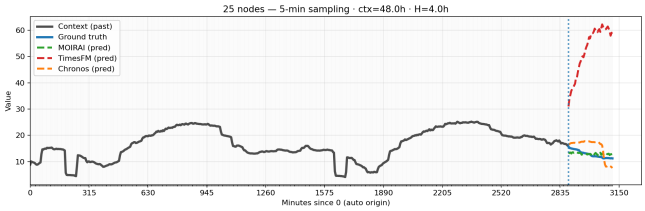
Here, eight of ten unique pairs remain connected ($8/10 = 80\%$), producing a dense spatial graph. The adjacency matrix A defines how information flows between sensors in the GNN. The GNN uses this adjacency to perform graph convolution:

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{A}\mathbf{H}^{(l)}W^{(l)}),$$

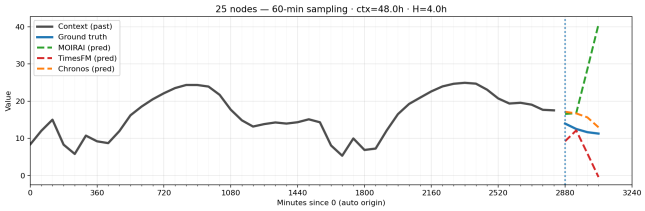
where $\tilde{A}$ is the normalized adjacency derived from A, ensuring that each node aggregates information proportionally to its correlations with neighbors. This process allows the GNN to learn spatial and temporal dependencies jointly: spatial through the correlation-based adjacency, and temporal through sequential updates across time windows. In our experiments, varying the threshold θ effectively varies redundancy allowing us to quantify how connectivity strength among colocated sensors affects downstream forecasting accuracy.

D Time Series Foundation Models performance (5 minute and 60 minute sampling rates)

An example visualization of our forecasting plots with context/history is shown as solid gray, ground truth as solid blue, and model predictions as dashed lines (green = MOIRAI, red = TimesFM, orange = Chronos). The vertical dotted line marks the forecast start (split).



(a) TSFM models forecasting at 5-minute sample rate.



(b) TSFM models forecasting at 60-minute sample rate.

Figure D.1: Comparing TSFM models' performance at different sampling intervals