Benchmarking Deep Learning for $PM_{2.5}$ Forecasting in IoT-Based Smart Cities: GCN Spatial Encoding and Transformer Temporal Modeling

Abdessamad BADOUCH[1], Kaoutar BELHOUCINE[2]
Faculty of Sciences, Ibn Zohr University, Agadir, Morocco[1]
Faculty of Sciences, Ibn Tofail University, Kénitra, Morocco[2]

***Abstract*—Graph convolutional networks are widely used in air quality forecasting, yet their benefit over simpler approaches remains insufficiently validated. This study presents a fully reproducible benchmark comparing five model families (ARIMA, XGBoost, LSTM, CNN-LSTM, and a GCN+Transformer model) on the public Beijing Multi-site Air Quality Dataset. All experiments share identical preprocessing, three random seeds (confidence intervals reported as exploratory), and a chronological split, with full code availability. ARIMA outperforms deep learning on aggregated data, due to strong temporal autocorrelation. On individual stations, Transformer-based models achieve the best performance through improved temporal modeling. A systematic multi-node analysis reveals that graph-based spatial aggregation can degrade performance under highly homogeneous conditions: when monitoring stations exhibit strong correlations, a simple linear baseline outperforms the GCN topologies tested here. Alternative spatial encoders (e.g., GAT, learnable adjacency) may behave differently and remain an open question. These findings define a practical regime in which GCN-based spatial aggregation provides no benefit over linear baselines. A preliminary compression experiment on Apple M1 reports a 42% inference-latency reduction via 8-bit quantization; this is indicative only, and validation on Raspberry Pi 4 or NVIDIA Jetson Nano is identified as required follow-up.**

***Keywords*—Air quality forecasting; graph neural networks; IoT smart cities; transformer; $PM_{2.5}$ prediction**

I. Introduction

Urban air pollution is one of the most serious environmental health challenges of our time. $PM_{2.5}$ particles, with aerodynamic diameter below 2.5 micrometres, penetrate deep into lung tissue and are associated with cardiovascular and respiratory disease, premature mortality, and impaired cognitive development [2], [3]. In cities across China, India, and the Middle East, daily $PM_{2.5}$ concentrations regularly exceed WHO guidelines by factors of five to ten. Forecasting these concentrations accurately, and doing so in real time, is a prerequisite for any early warning or emergency response system.

Smart city infrastructure changes what is technically possible here. Networks of low-cost IoT sensors now generate continuous, high-resolution spatiotemporal streams from dozens of monitoring stations within a single city. This spatial dimension is what makes graph neural networks attractive: if stations at different locations carry different information, encoding their spatial relationships should help the model. Whether inter-station diversity is high enough for graph aggregation to help, rather than simply average away useful local variation, depends on the specific city and meteorological regime. That question motivates this study.

This study does not propose a new architecture. It provides a reproducible benchmark on public data, with a clear empirical result: on Beijing's monitoring network, graph spatial encoding does not improve forecasting performance over a linear baseline. Published claims about GCN superiority in air quality forecasting consistently appear on proprietary, spatially heterogeneous datasets [20], [22], often multi-city or multi-provincial, without testing whether the spatial component would remain useful if correlations were high and uniform. Our benchmark fills that gap by providing the baseline case. Negative results on public reproducible benchmarks are an undervalued but essential contribution to the field: without them, the community cannot know which spatial modeling assumptions are safe to import from traffic forecasting, where GCN consistently helps, to air quality forecasting, where the spatial precondition may not hold.

Showing that GCN does not help in one spatially homogeneous setting does not by itself establish exactly when it does help. Testing that requires a second dataset from a heterogeneous network, which we identify as the most important direction for follow-up work. The contribution here is a documented, reproducible reference case, grounded in the meteorology of the Beijing basin.

The contributions of this study are as follows. First, a reproducible benchmark comparing five model families on UCI #501 is provided (anonymized repository; URL withheld during review and to be disclosed upon acceptance), with fixed seeds and public code, addressing the reproducibility gap in prior work [20], [21], [22]. Second, a multi-signal analysis identifies when ARIMA is competitive (aggregated, smooth signals) and when it is not (per-station noisy data), with the Transformer and GCN components disentangled in both settings. Third, a reproducible negative result on GCN spatial encoding is established for networks with near-uniform correlations, with a physical explanation rooted in Beijing's meteorological dynamics. Fourth, a preliminary edge deployment analysis reports a 42% latency reduction via 8-bit quantization on Apple M1 hardware, offered as an initial feasibility indicator for IoT deployment.

II. Related Work

A. Classical and Statistical Methods

ARIMA and its seasonal variant SARIMA remain the standard baselines in time series forecasting. They are well understood, interpretable, and require no training data beyond the target series itself. Their main limitation is the linearity assumption: ARIMA captures autocorrelation structure well but cannot model abrupt nonlinear events such as dust storms or holiday traffic changes. Recent work has also explored statistical process control approaches such as modified EWMA charts for autocorrelated $PM_{2.5}$ monitoring [23], which complement forecasting models for anomaly detection use cases. Beyond $PM_{2.5}$, related assessment frameworks have been applied to other pollutants such as SO_2 [24], confirming the general relevance of statistical baselines in air quality studies. Our benchmark confirms R^2=0.973 for ARIMA on the aggregated signal, consistent with prior comparisons [18]. This result is a product of the aggregation process, as we discuss in Section IV.

XGBoost and gradient boosting methods have also been applied to air quality forecasting with good results on tabular feature sets [1]. Hybrid pipelines combining LSTM, Random Forest, and XGBoost have also been proposed for real-time AQI prediction [25], combining the complementary strengths of recurrent and tree-based models. They handle nonlinearity and work well on moderate-size datasets without tuning recurrent architectures. Their limitation is the same as ARIMA's: they treat each time step independently unless explicit lag features are engineered, and they have no native mechanism for spatial reasoning.

B. Deep Learning Models

LSTM networks [16] were among the first deep learning architectures applied to air quality forecasting, and they remain strong baselines. The gated mechanism handles long-range temporal dependencies better than vanilla RNNs, but training is sequential and parallelism is limited. Several studies have combined convolutional layers with LSTM to capture both local spatial patterns and temporal dynamics [10], [11], [12], with reported improvements over LSTM-only baselines in spatially structured settings.

Transformer architectures [17] arrived later in this application domain. The self-attention mechanism processes all timesteps in parallel, is more efficient on GPU hardware, and handles long-range dependencies without the vanishing gradient issues that affect LSTMs. AirFormer [22] demonstrated the Transformer's capacity to scale to 1,085 stations across China, though on proprietary data and without systematic baselines. Our ablation confirms that the Transformer is the decisive component in our framework, contributing ΔR^2=0.111 relative to an equivalent GCN+LSTM architecture.

C. Graph Neural Networks in Air Quality

GNNs have become popular for air quality forecasting because monitoring stations are naturally modeled as graph nodes. Zhang et al. [10] combined temporal difference features with graph Transformer networks and reported improvements over CNN-LSTM on multi-station data in China. Liang et al. [20] developed a Res-GCN model incorporating satellite imagery on 34 Beijing and 27 Tianjin stations, using a dynamic GCN structure to capture evolving spatial patterns. These results are encouraging but rely on multi-city or multi-provincial settings where inter-station spatial diversity is inherently higher. Ghose et al. [6] applied a graph-like neighbor influence model in an Indian smart city context, showing that nearby station information helps when pollution sources are spatially heterogeneous.

D. Structured Taxonomy: Reproducibility and Ablation Gaps

Table I surveys representative works spanning classical statistical methods to graph-based and Transformer architectures, making clear where this study sits within the literature. Eight evaluation criteria are assessed: inclusion of a classical baseline (ARIMA or similar), machine learning baselines (XGBoost/RF), recurrent DL (LSTM/GRU), CNN-based spatial features, graph neural network component, Transformer component, controlled GCN ablation against a linear baseline, and full reproducibility (public data and code). Symbols: ✓ present; x absent; $\sim$ partial.

Several patterns stand out in Table I. Among surveyed works, none simultaneously provides all eight features. The most common gap is the absence of GCN ablation: GCN components are commonly added without testing whether GCN outperforms a linear baseline on the same data. Similarly, full reproducibility on public data with code remains rare. Among the works using GCN [6], [10], [20] or similar graph structures, none ablate the spatial component against a linear baseline, leaving open the question of whether the reported gains come from the graph aggregation itself or from the other components of the architecture. Table II summarises method-by-method gaps and how this study addresses each of them.

III. Methodology

A. Problem Formulation and Graph Setup

The problem is framed as multivariate time series regression. At each hourly timestep t, a station i reports a vector of six measurements: $PM_{2.5}$ concentration (the target), NO_2, temperature (TEMP), atmospheric pressure (PRES), dew-point temperature (DEWP), and wind speed (WSPM). The model receives a sequence of 24 consecutive hourly observations (window length $T = 24$) and predicts $PM_{2.5}$ at timestep $t+1$.

In the multi-node setting, the 12 stations are represented as graph nodes. Two graph construction strategies are compared. In the *distance-based* topology, edges connect each station to its $k = 5$ nearest neighbors by Haversine distance, with edge weights $w_{ij} = 1/d_{ij}$ inversely proportional to distance. In the *correlation-based* topology, edges connect the $k = 5$ stations with the highest Pearson correlation of $PM_{2.5}$ time series, computed exclusively on the training split to prevent data leakage. In the aggregated benchmark, the 12 station signals are averaged into a single hourly series before model input; in that setting the GCN has only one node and therefore performs a linear projection of the feature vector rather than any neighborhood aggregation.

The choice $k = 5$ was validated by a sensitivity analysis over $k \in \{3, 5, 7, 10\}$: performance under the distance-based topology varies by less than ΔR^2=0.004 across all values

TABLE I. COMPREHENSIVE TAXONOMY OF AIR QUALITY FORECASTING STUDIES (2019–2025). ARIMA = CLASSICAL STATISTICAL BASELINE; ML = TREE/ENSEMBLE ML; LSTM = RECURRENT DL; CNN = CONVOLUTIONAL SPATIAL FEATURES; GCN = GRAPH NEURAL NETWORK; TRANS. = TRANSFORMER; ABLAT. = CONTROLLED GCN VS. LINEAR ABLATION; REPRO. = PUBLIC DATA AND CODE

Study	Dataset	ARIMA	ML	LSTM	CNN	GCN	Trans.	Ablat.	Repro.
Kow et al. [5] 2022	Taiwan	✓	x	x	✓	x	x	x	x
Zhang et al. [10] 2022	Beijing	x	x	✓	x	✓	✓	x	x
Zhang et al. [11] 2022	Chinese cities	x	x	✓	✓	x	x	x	x
Ghose et al. [6] 2022	India, smart city	x	x	✓	x	∼	x	x	x
Krishan et al. [16],[3]	Multi-dataset	✓	✓	✓	x	x	x	x	x
Cican et al. [8] 2023	Bucharest, Romania	✓	✓	✓	x	x	x	x	x
Srivastava [12] 2023	Multi-station	x	x	✓	x	x	x	x	x
Soh et al. [21] 2023	Beijing+Taizhou	✓	x	x	x	x	✓	x	x
Hameed et al. [2] 2023	Doha, Qatar	x	x	✓	✓	x	x	x	x
Ansari & Quaff [13] 2025	Azamgarh, India	✓	✓	✓	x	x	x	x	x
Ansari & Quaff [14] 2025	Azamgarh, India	✓	✓	✓	x	x	x	x	x
Tello-Leal et al. [7] 2024	Mexico, urban	x	✓	✓	x	x	x	x	x
Liang et al. [20] 2024	Beijing+Tianjin	x	x	x	✓	✓	x	x	x
Rahmani et al. [9] 2024	France, multi-city	x	x	✓	x	x	x	x	x
Shahbazi et al. [1] 2024	Multi-region	✓	✓	✓	x	x	x	x	x
Alsabagh et al. [19] 2025	Beijing UCI #501	✓	x	✓	✓	x	x	x	x
This work (2025)	**Beijing, UCI #501**	✓	✓	✓	✓	✓	✓	✓	✓

TABLE II. COMPARISON OF RELATED WORK: GAPS ADDRESSED BY THIS STUDY

Study	Dataset	Method	Key Limitation	Addressed
Du et al. [4]	Beijing UCI	CNN-LSTM	No Transformer; no GCN ablation; no edge deployment	✓
Kow et al. [5]	Taiwan	Deep CNN	Image input only; single modality	✓
Hameed et al. [2]	Doha	Multimodal DL	Proprietary data; no GNN; no edge	✓
Zhang et al. [10]	Beijing	Graph Transformer	No ARIMA/XGBoost; no GCN ablation; no code	✓
Zhang et al. [11]	Chinese cities	CNN-LSTM	No GNN; no edge deployment	✓
Lin et al. [15]	Multi-city	Ensemble DL	No GCN; no reproducibility	✓
Ghose et al. [6]	India	DL + neighbors	No Transformer; no GCN ablation	✓
Ansari & Quaff [13], [14]	Azamgarh	Deep learning	Single city; no graph model	✓
AirFormer [22]	1,085 stations	Transformer	Proprietary data; no GCN ablation; no code	Partial
Liang et al. [20]	Beijing+Tianjin	Res-GCN	Satellite imagery; no GCN spatial ablation	Partial
Soh et al. [21]	Beijing+Taizhou	Sparse Transformer	Single-station only; no GNN; no code	✓
Alsabagh et al. [19]	Beijing, UCI	DNN+CNN, cyclic encoding	No GNN or Transformer; no edge analysis	✓
This work	**Beijing, UCI #501 (public)**	**Benchmark + GCN ablation**	Full reproducible benchmark + negative spatial result	—

(best at $k = 5$: R^2=0.932; worst at $k = 10$: R^2=0.928), confirming that the negative result is not sensitive to the specific neighborhood size and holds for all tested values of k.

B. GCN Feature Encoder

The spatial encoder uses two layers of standard graph convolution (`GCNConv` from `torch_geometric`). For a node i with hidden state h_i and neighbor set $\mathcal{N}(i)$, the update rule at layer l is:

$$h_i^{(l+1)} = \mathrm{ReLU}\Big(W^{(l)} \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{w_{ij}}{\sqrt{d_i d_j}} h_j^{(l)}\Big), \qquad (1)$$

where, d_i and d_j are the degrees of nodes i and j and $W^{(l)}$ is a learned weight matrix. The renormalization by node degrees follows the standard GCN formulation. When applied to the aggregated signal (a single node), the neighbor sum reduces to the identity, and the GCN layer becomes a plain linear transformation, a property that is central to interpreting the per-station results in Section IV.

C. Transformer Temporal Encoder

The output of the GCN encoder is passed to a Transformer encoder with multi-head self-attention [17]. Given the sequence of GCN-encoded feature vectors, the attention mechanism computes a weighted sum of value vectors, where the weights are determined by the scaled dot-product similarity between query and key projections. This allows each position in the sequence to attend to any other position regardless of distance, capturing long-range temporal dependencies that LSTM's sequential processing handles less reliably.

We use sinusoidal positional encoding, as in the original Transformer [17], rather than learned positional embeddings,

for two reasons: it generalizes to sequence lengths not seen during training, and it introduces no additional parameters. The architecture has 2 layers, 4 attention heads, a model dimension of 64, and dropout of 0.1. Fig. 1 gives an overview of the full pipeline.

D. Training and Reproducibility

Training uses mean squared error loss: $\mathcal{L} = (1/N)\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, where y_i is the true $PM_{2.5}$ concentration and $\hat{y}_i$ is the model prediction. The Adam optimizer is used with learning rate 10^{-3} and weight decay 10^{-5}. Early stopping with patience 7 is applied on validation loss. All experiments use a fixed random seed (42 for the primary run) and are replicated with seeds 123 and 777 to estimate variability. All results are reported as mean $\pm$ standard deviation across 3 seeds, with 95% confidence intervals computed as mean $\pm t_{0.025,2} \cdot \mathrm{SD}/\sqrt{3}$. Because only three seeds are used, the confidence intervals reported throughout this study should be interpreted as exploratory rather than definitive estimates of model variability. The t-distribution with two degrees of freedom yields a critical value of $t \approx 4.303$, so reported intervals are wide and sensitive to single-seed outliers; this is particularly visible for the CNN-LSTM variant, which exhibits high seed-to-seed variance. We retain this exploratory protocol because all results, including the negative GCN spatial-encoding finding, hold well beyond the magnitude of these intervals; re-estimation with 10 or more seeds is identified as a refinement priority for follow-up work.

Data are split chronologically (70% train, 15% validation, 15% test) with no shuffling, to prevent any leakage of future information. All input features are MinMax normalized using statistics computed exclusively on the training split; missing values are imputed by linear interpolation before normalization. Table III summarizes the hyperparameters.

Baseline configurations are specified as follows. **ARIMA**: fixed order $(5, 1, 2)$, with no seasonal component (i.e., non-seasonal ARIMA rather than SARIMA), implemented via `statsmodels`. Forecasts use a walk-forward / rolling-refit protocol: at each step of the 500-point test horizon, the ARIMA$(5, 1, 2)$ model is re-fit on the most recent 200 $PM_{2.5}$ observations and produces a one-step-ahead forecast. The order $(5, 1, 2)$ was selected as a fixed, replicable configuration rather than via automatic AIC/BIC search, so that the comparison is reproducible without dependence on a search procedure. **XGBoost**: 500 estimators, max_depth=6, learning_rate=0.05, subsample=0.8, colsample_bytree=0.8, early stopping with 20 rounds on the validation set (`eval_metric="rmse"`); features are the lag-1 target value concatenated with the exogenous variables at time t. **LSTM**: 2 stacked layers, hidden size 128, dropout 0.2; trained for up to 50 epochs with batch size 64 and early stopping (patience 7) on validation MSE. **CNN-LSTM**: two stacked 1D-convolutional layers (64 filters each, kernel size 3, padding 1, ReLU activations) followed by a 2-layer LSTM with hidden size 128 and dropout 0.2; trained under the same optimizer and early-stopping protocol as the LSTM. All deep learning baselines share the GCN+Transformer's optimizer (Adam, lr 10^{-3}, weight decay 10^{-5}), MSE loss, batch size (64), input window (24 h), and 3-seed protocol (seeds 42, 123, 777). The full configuration is available in the public repository.

TABLE III. HYPERPARAMETERS OF THE GCN+TRANSFORMER FRAMEWORK

Module	Parameter	Value
GCN encoder	Architecture / input dim.	2-layer GCNConv / 6 features per node
	Graph topology	Haversine k-NN (k=5) or Pearson $PM_{2.5}$ (k=5)
Transformer	Layers / heads / d_{model} / dropout	2 / 4 / 64 / 0.1
Training	Optimizer / lr / wd	Adam / 10^{-3} / 10^{-5}
	Loss / batch / seq. len.	MSE / 64 / 24 h
	Early stop / seeds	7 / {42, 123, 777}
Data split	Train / val / test	70% / 15% / 15% (chronological)
Edge (prelim.)	Quantization / HW	qint8 / Apple M1

IV. EXPERIMENTAL RESULTS

A. Aggregated Benchmark

Table IV shows performance on the station-averaged signal. ARIMA achieves R^2=0.9726, the highest value in the table. The gap between ARIMA and the deep learning models is real and is reported without qualification. It also requires context. Averaging 12 stations acts as a spatial low-pass filter: it removes short-lived, spatially localized events and produces a smooth, strongly autocorrelated univariate series, the conditions under which ARIMA is near-optimal [3], [18]. Confirming this interpretation, ARIMA's performance drops substantially on the non-averaged Dongsi signal (R^2: $0.973 \rightarrow 0.899$), which we discuss in Section IV-A.1 below.

Among deep learning models, GCN+Transformer achieves R^2=0.939, outperforming LSTM (R^2=0.847) by ΔR^2=0.092. The confidence intervals are non-overlapping, which indicates the improvement is consistent across initialization conditions. CNN-LSTM performs considerably worse (R^2=0.698), with a wide confidence interval that suggests instability in training. Fig. 2 visualizes the performance comparison under both evaluation settings.

TABLE IV. BENCHMARK ON AGGREGATED SIGNAL, BEIJING MULTI-SITE (UCI #501). ⋆ BEST OVERALL; † BEST DL MODEL. ARIMA ADVANTAGE IS AN AGGREGATION ARTIFACT; SEE TEXT

Category	Model	MAE	RMSE	R^2 (95%CI)	Note
Stat.	ARIMA⋆	4.17±1.2	6.51±0.8	0.9726	Best on smooth agg. signal
	XGBoost	6.35±.22	11.67±.35	0.963 [.94–.99]	
DL	LSTM	12.0±.10	23.81±.48	0.847 [.60–.89]	
	CNN-LSTM	21.1±4.8	33.04±5.4	0.698 [.04–1.0]	High variance across seeds
	GCN+T†	8.17±1.0	15.07±.67	0.939 [.88–.99]	Best DL on aggregated signal

1) Per-station results at dongsi: Dongsi is selected as the per-station validation site. It is one of the twelve stations in UCI #501 and was chosen because it has higher intra-series variance than the station average, a property consistent with its mixed residential and traffic environment in the Dongcheng district. Evaluating on a single station, without any cross-station averaging, removes the smoothing effect and tests the model under conditions closer to a real IoT sensor deployment.

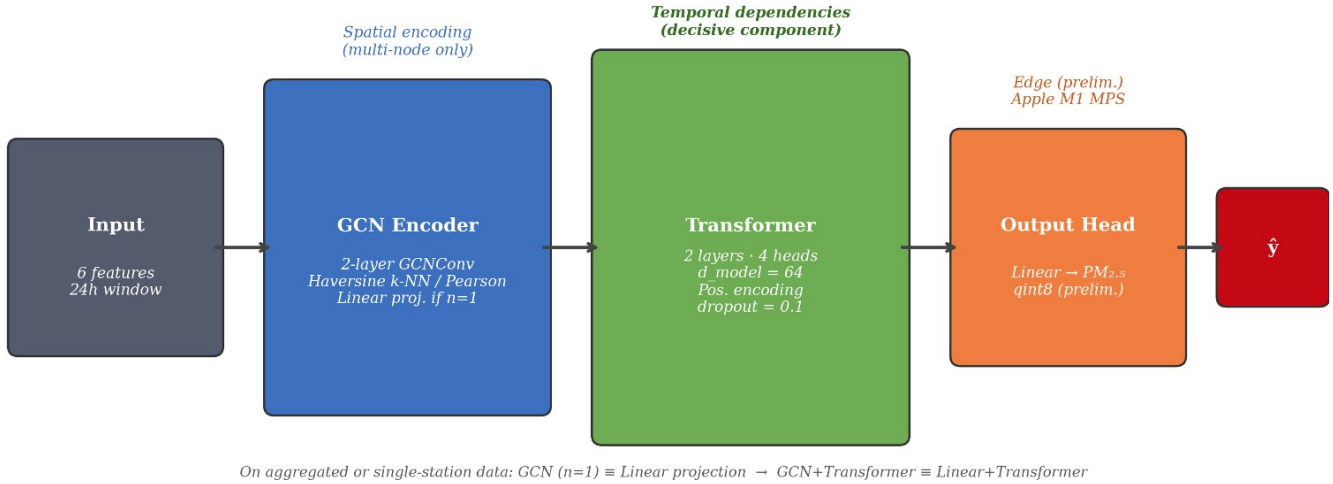

Fig. 1. GCN+Transformer pipeline. Input features (6 variables, 24 h window) pass through the GCN encoder and then the Transformer. On aggregated or single-station data ($n = 1$ node), the GCN reduces algebraically to a linear projection, making GCN+Transformer ≡ Linear+Transformer.

Table V includes both GCN+Transformer and Linear+Transformer. Including Linear+Transformer is important because, on a single node, GCN reduces algebraically to a linear projection: with no neighboring nodes to aggregate, the GCN convolution $h = \mathrm{ReLU}(W \cdot h_{\mathrm{self}})$ is mathematically identical to a linear layer. The two models therefore produce identical outputs on Dongsi, confirming that the improvement over ARIMA is driven entirely by the Transformer temporal encoder. Implementation note: the Linear+Transformer variant directly replaces the single-node `GCNConv` with its algebraically equivalent linear operation (a single `Linear` layer followed by ReLU, with the same weight initialization scheme and the same downstream Transformer encoder). The reported metrics for the two rows are therefore numerically identical and not approximations.

TABLE V. PER-STATION RESULTS: DONGSI (NON-AGGREGATED HOURLY DATA). † ON A SINGLE STATION ($n = 1$ NODE), GCN ≡ LINEAR PROJECTION. ALL DL MODELS: MEAN ± SD OVER 3 SEEDS

Model	MAE	RMSE	R^2	Signal type
ARIMA	10.11	15.70	0.8986 (↓0.074)	Non-aggregated
LSTM	15.54±0.10	25.08±0.58	0.855	Non-aggregated
Linear+Trans†	13.46±3.86	20.31±3.39	0.902	GCN = Linear (n=1)
GCN+Trans†	13.46±3.86	20.31±3.39	**0.902 (best)**	Identical to Linear+T

The ΔR^2=+0.003 improvement of GCN+Transformer over ARIMA on Dongsi is therefore produced by the Transformer component. This does not show that GCN helps; it shows instead that long-range temporal modeling helps under noisy, station-level conditions where ARIMA's linear autoregression assumption no longer holds. The primary result of this experiment is not the magnitude of the aggregate metric improvement, but the disentanglement: on a single noisy station, both GCN+Transformer and Linear+Transformer produce identical outputs, confirming that the Transformer encoder is the decisive component.

B. Component Ablation

Table VI quantifies each component's contribution using three model variants on the aggregated benchmark. Replacing the GCN encoder with a standard linear layer produces no change in performance: both variants achieve R^2=0.9386 with identical RMSE. This is consistent with the single-node analysis above: when spatial aggregation is absent, the GCN degenerates to a linear transformation. Replacing the Transformer with an LSTM, on the other hand, produces a large drop: R^2 falls from 0.939 to 0.828 and RMSE nearly doubles. The $4.3\times$ increase in inference latency for the LSTM variant (4.74 ms vs 1.11 ms) reflects the sequential nature of recurrent processing compared to the parallel self-attention of the Transformer on MPS hardware.

TABLE VI. COMPONENT ABLATION: AGGREGATED BENCHMARK (UCI #501)

Variant	R^2	RMSE	Latency (M1)
Full GCN+Transformer	0.9386	15.07±0.67	1.11 ms/batch
No-GCN (Linear+Trans.)	0.9386 (=)	15.07±0.67	1.03 ms/batch
No-Trans. (GCN+LSTM)	0.8275 (↓0.111)	25.25±1.41	4.74 ms/batch

C. Multi-Node Graph Topology Ablation

The multi-node experiment treats all 12 stations as independent graph nodes, comparing both graph construction strategies against a Linear+Transformer baseline (Table VII). The inter-station $PM_{2.5}$ Pearson correlation matrix was computed on the training split before any model training. Every pair of stations has a correlation between 0.78 and 0.96, with a mean of 0.88. This is a consequence of Beijing's geography and climate: the city sits in a semi-enclosed basin and experiences large-scale pollution events driven by regional meteorology such as Siberian anticyclones, southerly wind reversals, and thermal inversions that affect the entire network simultaneously.

Under these conditions, a neighboring station's $PM_{2.5}$ series is nearly identical to the target station's, so adding its features contributes nothing useful. GCN aggregation in this regime produces a weighted average of near-identical inputs,

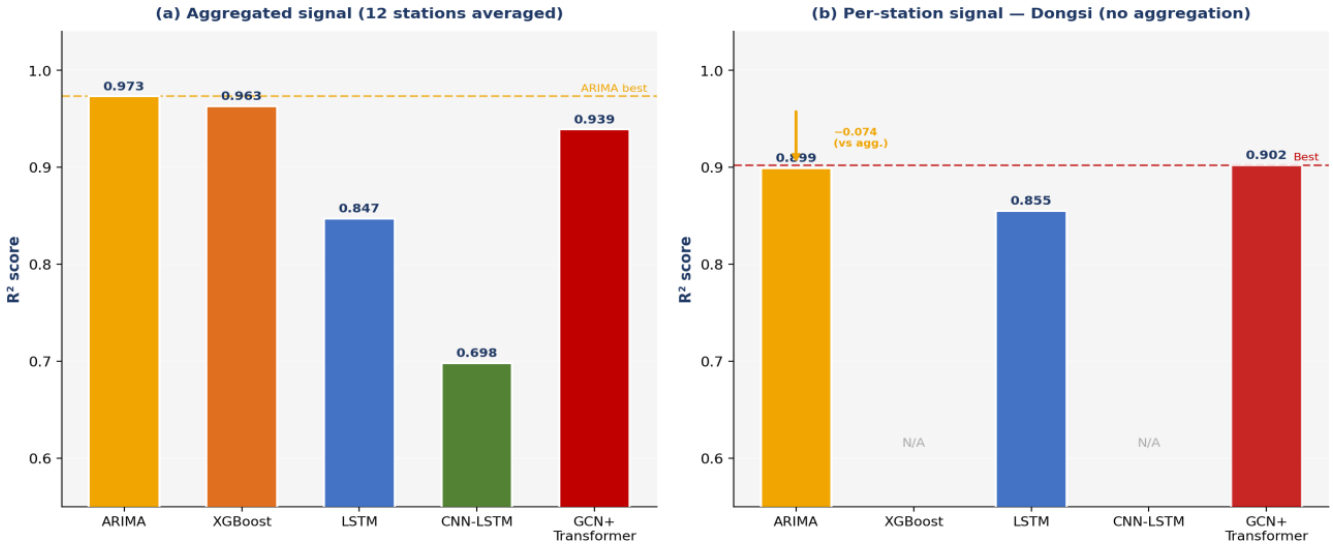

Fig. 2. Model performance (R^2) under two evaluation settings. (a) Aggregated signal: ARIMA achieves the best score due to the smoothing effect of station averaging. (b) Per-station signal (Dongsi): ARIMA degrades sharply (-0.074), while the Transformer-based model remains best. XGBoost and CNN-LSTM were not evaluated on Dongsi and are shown as N/A.

TABLE VII. MULTI-NODE TOPOLOGY ABLATION (12 STATIONS, 3 SEEDS). REFERENCE: LINEAR+TRANSFORMER R^2=0.952 (INTER-STATION CORRELATION: 0.78–0.96)

Model / Topology	R^2 (±SD)	RMSE	ΔR^2	Verdict
Linear+Trans. (ref.)	0.9516±0.0001	20.36	—	Reference
GCN+T, distance (k=5)	0.9319±0.0008	24.17	−0.020	x Inferior
GCN+T, corr. (k=3)	0.9211±0.0009	26.01	−0.030	x Inferior
GCN+T, corr. (k=5)	0.9114±0.0009	27.56	−0.040	x Inferior
GCN+T, corr. (k=7)	0.8947±0.0023	30.04	−0.057	x Inferior
GCN+T, corr. (k=10)	0.8811±0.0004	31.93	−0.070	x Inferior

which adds noise without contributing signal. The results confirm this: both GCN topology variants underperform the linear baseline, with the correlation-based topology performing worse than the distance-based one (R^2 0.911 vs 0.932). The correlation-based result is inferior because uniform high correlations produce near-identical edge weights, making the graph structure almost fully degenerate.

Concretely, with pairwise correlations clustered in the narrow band $[0.78, 0.96]$, the top-$k = 5$ neighbor selection becomes effectively arbitrary (any station could be a neighbor of any other with similar weight), and the resulting adjacency matrix carries little discriminative information. The GCN message-passing operation then averages near-identical signals weighted by near-identical coefficients, which is mathematically close to no spatial operation at all, but with the added cost of injecting estimation noise from the empirical correlation matrix into the model. This explains why the correlation-based topology underperforms even the distance-based one, which at least encodes a fixed geographic prior.

To further verify that this finding is not an artefact of the default $k = 5$ setting, we ran a sensitivity analysis over $k \in \{3, 5, 7, 10\}$ on the correlation-based topology. The result holds: GCN+Transformer underperforms the Linear+Transformer baseline (R^2=0.951) at every tested value of k, with R^2 monotonically decreasing as k grows: 0.921 ($k = 3$), 0.911 ($k = 5$), 0.895 ($k = 7$), and 0.881 ($k = 10$), a spread of 0.040 R^2 units between the best and worst configurations. The monotonic degradation as k increases is the empirical signature of the degeneracy described above: adding more neighbours injects more averaging of near-identical signals together with more noise from the estimated correlation matrix, without contributing additional spatial information. Standard deviations across the three seeds are small (≤ 0.002 R^2), so the ordering is statistically stable. This sweep complements the distance-based k-sensitivity already reported (R^2 spread < 0.004) and shows that the negative GCN finding holds across both graph-construction strategies and across reasonable neighbourhood sizes. This monotonic degradation directly supports the "degenerate graph" interpretation: in the Beijing network, increasing the neighbourhood size adds redundant signals and estimation noise but no new spatial information, so the larger the graph, the worse the performance.

V. DISCUSSION

A. Reading the ARIMA Result Correctly

It may seem surprising that a classical statistical model outperforms deep learning on a spatial forecasting task. The explanation becomes clear when the evaluation setup is examined closely. The primary benchmark in Table IV uses a signal that has been averaged across 12 stations. That averaging step smooths out everything that makes the prediction problem hard for ARIMA: short-lived local events, sensor noise, and spatial heterogeneity. What remains is a smooth, slowly varying time series with strong seasonal and diurnal autocorrelation, exactly the type of signal ARIMA was built for.

The practical implication is direct. A system that receives one aggregated city-level reading per hour might reasonably use ARIMA. A system receiving data from 12 individual IoT sensors, each with its own noise and local dynamics, faces a different problem where deep learning's advantages become

real, as the Dongsi results show. The choice of benchmark should reflect the actual deployment context.

We also stress that the aggregated and per-station evaluations are not directly comparable as a single ranked benchmark: they reflect two structurally different forecasting problems with different intrinsic difficulty. Aggregation across 12 stations acts as a spatial low-pass filter that mechanically increases the autocorrelation of the target series and therefore raises the achievable R^2 ceiling for any reasonable autoregressive model, including ARIMA. Conversely, per-station forecasting at Dongsi preserves short-lived local events and sensor noise, and the achievable R^2 is correspondingly lower for every model in the comparison (ARIMA drops from 0.973 to 0.899; Transformer-based models drop from 0.939 to 0.902). The two settings should therefore be read jointly: the aggregated benchmark answers "which model best fits a smoothed city-level signal?" (answer: ARIMA, by construction of the aggregation), while the per-station benchmark answers "which model best handles noisy, locally driven station data?" (answer: Transformer-based, with the GCN component contributing nothing additional in this single-node configuration). Reporting both prevents either result from being over-extrapolated.

B. The GCN Result and Its Scope

The negative result on GCN spatial encoding is reproducible and physically interpretable. When a city's pollution dynamics are dominated by large-scale meteorological forcing, all stations move together, and knowing your neighbor's $PM_{2.5}$ concentration tells you almost nothing beyond what your own sensor already reports. Under those conditions, a GCN that aggregates neighboring states introduces noise rather than signal.

We are careful about what this finding does and does not say. We have demonstrated a case where GCN does not contribute additional signal, an upper-bound condition. For urban networks where inter-station correlations are uniformly high (around 0.78–0.96 as measured here), spatial graph encoding can be replaced by a linear layer without loss. What we have not shown is what happens when spatial heterogeneity is present, for example in a coastal city where prevailing winds create a sustained upwind-downwind pollution gradient, or in an industrial zone where emission sources are spatially concentrated. Those settings are where prior GCN-based work has found its advantages [10], [20], [22]. Our result does not contradict those findings; it contextualizes them.

C. IoT Edge Deployment: Scope of the Current Analysis

The 42% latency reduction from 8-bit quantization is an indicative result, not a deployment claim. The figure was measured on Apple M1 hardware, which is not representative of the edge devices typically deployed in smart city IoT networks. The corresponding effect of dynamic 8-bit quantization on predictive accuracy is left to follow-up work: at the time of this revision, the available PyTorch quantization backend (qnnpack on Apple Silicon) does not currently support the Transformer attention modules of our architecture in a form that allows a reliable end-to-end accuracy comparison. We therefore do not report a partial or unverified accuracy number, and identify the joint accuracy/latency trade-off, measured end-to-end on the actual target device, as a required follow-up. A meaningful edge deployment analysis would require testing on at least one resource-constrained device such as a Raspberry Pi 4 or NVIDIA Jetson Nano, covering inference latency, peak and average power draw, and behavior under thermally limited sustained operation. We have not done this, and we list it as a concrete next step rather than implying coverage that does not exist.

D. Limitations

The most significant limitation is that the study rests on a single real dataset. The negative GCN result is a property of Beijing's specific meteorological dynamics, and it would be overstepping to present it as a general finding without cross-city replication. The next logical step, evaluation on a dataset with documented spatial heterogeneity, is identified in the future work section and is not a matter of polishing; it is about completing the empirical argument.

To partially address this concern within the constraints of the present study, we compute the inter-station Pearson correlation for the 12 UCI Beijing stations (mean $r = 0.88$, range 0.78–0.96) and identify it as the operative condition explaining the absence of GCN gain. This provides a falsifiable prediction: we tentatively hypothesize, as a falsifiable conjecture to be tested in follow-up work and not as an established rule, that on networks where mean inter-station $PM_{2.5}$ correlation falls below approximately 0.70, GCN spatial encoding may recover its benefit. The specific value of 0.70 is illustrative and drawn from a single-dataset extrapolation; the actual threshold (if a sharp transition exists at all) would require cross-network validation to be characterized empirically. Three candidate datasets satisfying this criterion are identified for follow-up validation: 1) OpenAQ multi-city data (Beijing–Shanghai–Guangzhou, where inter-city correlations are near zero by construction); 2) the KDD Cup 2018 dataset covering Beijing and London simultaneously; 3) any coastal industrial network exhibiting directional wind-driven pollution gradients. The regime transition hypothesis is thus stated precisely enough to be refuted, which is appropriate for a single-dataset empirical result.

Second, the primary benchmark is built on aggregated data, which as we have argued disadvantages deep learning by design. A fully per-station multi-station benchmark would be more representative of IoT deployment conditions. Third, no uncertainty quantification beyond multi-seed confidence intervals is provided; MC dropout and ensemble methods remain as extensions.

VI. Conclusion and Future Work

The study makes three contributions: a fully reproducible benchmark on a public dataset, an analysis of when ARIMA is and is not competitive with deep learning, and a controlled negative result on GCN spatial encoding with a physical explanation. On Beijing's monitoring network, where inter-station $PM_{2.5}$ correlations range from 0.78 to 0.96 due to city-wide meteorological forcing, the two GCN-based spatial encoders tested here (distance-based and correlation-based k-NN topologies) do not improve forecasting performance over a Linear+Transformer baseline; replacing the GCN with a linear

layer improves R^2 by 0.020 to 0.040. This result is specific to the standard GCN formulation; alternative spatial encoders such as graph attention networks or learnable adjacency models may behave differently and remain to be tested. The Transformer encoder is the component that matters: removing it reduces R^2 by 0.111 and increases inference latency by a factor of four.

On per-station data at Dongsi, where there is no smoothing from station averaging, the Transformer-based model (R^2=0.902) narrowly beats ARIMA (R^2=0.899). We show algebraically that this improvement is driven by the Transformer, not the GCN, given that with a single graph node, the GCN reduces to a plain linear projection. For a compact urban IoT network with meteorologically homogeneous pollution dynamics, a simpler Linear+Transformer outperforms GCN+Transformer with lower computational cost.

The next step is to test on a second real dataset with documented spatial heterogeneity, for example an OpenAQ multi-city dataset or a coastal industrial network where spatial gradients are physically present. Without that, the regime transition we hypothesize remains unconfirmed. Other directions include: a full per-station multi-station benchmark as primary evaluation metric; dedicated edge hardware tests on Raspberry Pi 4 and Jetson Nano; dynamic or learnable graph topology construction; XAI analysis via attention maps and SHAP values for policy interpretability; multi-step and multi-pollutant prediction horizons; and structured pruning for further compression.

REFERENCES

[1] Z. Shahbazi et al., "Enhancing air quality forecasting using machine learning techniques," *IEEE Access*, vol. 12, pp. 197290–197299, 2024.

[2] S. Hameed et al., "Deep learning-based multimodal urban air quality prediction and traffic analytics," *Scientific Reports*, vol. 13, p. 22181, 2023.

[3] Q. Liao et al., "Deep learning for air quality forecasts: A review," *Current Pollution Reports*, vol. 6, pp. 399–409, 2020.

[4] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Deep air quality forecasting using hybrid deep learning framework," *IEEE Trans. Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2412–2424, 2021.

[5] P. Y. Kow et al., "Real-time image-based air quality estimation by deep learning neural networks," *J. Environmental Management*, vol. 307, p. 114560, 2022.

[6] B. Ghose, Z. Rehena, and L. Anthopoulos, "A deep learning-based air quality prediction using influencing pollutants in smart city," *J. Universal Computer Science*, vol. 28, no. 8, 2022.

[7] E. Tello-Leal et al., "Deep learning models for air pollutant concentration prediction in urban environments," *Sustainability*, vol. 16, no. 16, p. 7062, 2024.

[8] G. Cican et al., "Machine learning techniques in air quality prediction — Bucharest case study," *Sustainability*, vol. 15, no. 11, p. 8445, 2023.

[9] M. Rahmani et al., "PMForecast: Leveraging temporal LSTM for in situ air quality predictions," *Environmental Science and Pollution Research*, vol. 31, pp. 51760–51773, 2024.

[10] Z. Zhang et al., "Temporal difference-based graph transformer networks for PM2.5 prediction," *Frontiers in Environmental Science*, vol. 10, 2022.

[11] Q. Zhang et al., "Deep-air: A hybrid CNN-LSTM for fine-grained air pollution estimation," *IEEE Access*, vol. 10, pp. 55818–55841, 2022.

[12] H. Srivastava and S. K. Das, "Air pollution prediction using XRSTH-LSTM algorithm," *Environmental Science and Pollution Research*, vol. 30, no. 60, pp. 125313–125327, 2023.

[13] A. Ansari and A. R. Quaff, "Data-driven predictive modelling of hourly AQI using deep learning: Azamgarh, India," *Theoretical and Applied Climatology*, vol. 156, p. 74, 2025.

[14] A. Ansari and A. R. Quaff, "Advanced machine learning for precise hourly AQI prediction," *International J. Environmental Research*, vol. 19, 2025.

[15] C. Y. Lin, Y. S. Chang, and S. Abimannan, "Ensemble multifeatured deep learning models for air quality forecasting," *Atmospheric Pollution Research*, vol. 12, no. 5, 2021.

[16] M. Krishan et al., "Air quality modelling using LSTM over NCT-Delhi, India," *Air Quality, Atmosphere & Health*, vol. 12, pp. 899–908, 2019.

[17] A. Vaswani et al., "Attention is all you need," in *Proc. NeurIPS*, 2017.

[18] E. Chianese et al., "Spatio-temporal learning in predicting ambient particulate matter," *Ecological Informatics*, vol. 49, pp. 54–61, 2019.

[19] A. S. Alsabagh et al., "Deep learning framework for hourly air pollutants forecasting using encoding cyclical features across multiple monitoring sites in Beijing," *Scientific Reports*, vol. 15, 2025, doi: 10.1038/s41598-025-05472-5.

[20] Y. Liang et al., "A multi-modal deep-learning air quality prediction method: Beijing and Tianjin case study," *Entropy*, vol. 26, no. 3, p. 237, 2024.

[21] P. W. Soh, J. W. Chang, and J. W. Huang, "Modeling $PM_{2.5}$ using deep sparse attention-based transformer networks," *International J. Environmental Science and Technology*, vol. 20, pp. 1383–1396, 2023.

[22] Y. Liao et al., "AirFormer: Predicting nationwide air quality in China with Transformers," in *Proc. AAAI*, vol. 37, no. 12, 2023.

[23] Y. Supharakonsakun and Y. Areepong, "A Novel Statistical Process Control Approach for $PM_{2.5}$ Monitoring Using Time Series Modeling," *Emerging Science Journal*, vol. 9, no. 6, Dec. 2025, doi: 10.28991/ESJ-2025-09-06-09.

[24] A. Drygval, P. Drygval, and V. Tabunshchik, "Assessment of Sulfur Dioxide Levels in Atmospheric Air Over the Period 2019–2024," *Civil Engineering Journal*, vol. 11, no. 11, pp. 4724–4744, 2025, doi: 10.28991/CEJ-2025-011-11-016.

[25] J. Jayapradha, S.-C. Haw, N. Palanichamy, V. Arunesh, S. Pranav, and T. Senthil Kumar, "LRX: A Hybrid-based Real-Time Air Quality Index Prediction and Visualization Model," *Emerging Science Journal*, vol. 9, no. 5, pp. 2454–2470, 2025, doi: 10.28991/ESJ-2025-09-05-010.