Diyala Journal of Engineering Sciences

Journal homepage: https://djes.info/index.php/djes

ISSN: 1999-8716 (Print); 2616-6909 (Online)

Positional Encoding Enhanced Graph Quantile Learning for Uncertainty-Aware Spatial Prediction

Srinivasu Badugu[1], R. Manivannan[2*], D Radhika[2]

[1]Department of Computer Science and Engineering, Faculty of Science and Technology, ICFAI Foundation for Higher Education, Hyderabad, Telangana 501203, India

[2] Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, (Autonomous), Hyderabad, Telangana 500001, India.

ARTICLE INFO

Article history:
Received: 02/03/2026.
Revised: 26/08/2026.
Accepted: 06/09/2026.
Available online: 15/09/2026.

Keywords:
Spatial Data Prediction
Graph Neural Networks
Quantile Regression
Positional Encoding
Uncertainty Quantification

ABSTRACT

Spatial prediction is instrumental in environmental monitoring, urban studies, and geostatistics where accuracy and dependable uncertainty bounds are mandatory. This work introduces the Graph Quantile Neural Network (GQNN) with Positional Encoding (PE), which integrates graph-based spatial representation learning, the sinusoidal positional encoding, and ordered multi-quantile regression. The controlled benchmark is randomized to 70/ 15/ 15 train- validation- test splits and includes 100–1000 irregular nodes with k-nearest-neighbor graph construction (k = 8). Concerning practical relevance, the revised evaluation also specifies out-of-sample tests with public U.S. EPA AirData 2024 annual PM2.5 monitor data and the California Housing geospatial data set, with the same split for GQNN-PE, Ordinary Kriging, and Gaussian Process Regression (GPR). GQNN-PE results in MAE = 0.653, RMSE = 0.972 and determination coefficient (R 2) of 0.89, with PICP = 94.3% and MPIW = 2.11, which provides a decrease of RMSE by 24.3% and 17.0% with respect to Kriging and GPR respectively. For practical relevance, an explicit inverse z-score transformation is also provided so that normalized MAE, RMSE, and interval width can be reported in physical units on real data. Five-seed statistical testing gives $p < 0.01$.

1. INTRODUCTION

Distributed sensing, satellite observations, Internet-of-Things devices, and geographic information systems have rendered geo-referenced data at the core of environmental monitoring, precision agriculture, climate science, urban analytics, geostatistics, and intelligent transportation [1]. Such data is often irregularly sampled, spatially autocorrelated, heterogeneous and noisy, therefore, models used in practice have to yield both accurate predictions and reliable uncertainty quantifications.

Spatial prediction is fundamentally different from traditional independent-sample learning since observations are spatially autocorrelated, spatially heterogeneous, and linked by non-Euclidean relationships. The positional encoder formats are included geographic coordinates or structural data that encode into the representations of the node elements and the identifiability of nodes in similar neighbor topologies i enhanced [2]. Graph Neurons Networks represent spatial locations as vertices and their relations as edges to provide message passing on irregular and non-Euclidean domains [3].

Quantile regression is a distribution-free method for estimating conditional uncertainty by estimating a set of conditional quantiles rather than the conditional mean alone [4]. Calibration methods then verify if the produced uncertainty intervals achieve the desired empirical coverage and are sufficiently sharp [5].

Graph-based spatial models have also been effectively used for spatio-temporal kriging and interpolation as they naturally learn the dependencies between irregularly sampled observations [6]. Nevertheless, many graph-based predictors are still deterministic, and produce only a single point estimates. Therefore, sparsity aware uncertainty calibration schemes have been proposed for enhancing interval reliability over spatial regimes with few samples [7]. The value of providing quantified spatial uncertainty was also demonstrated in large scale environmental mapping, e.g. global soil-property prediction [8].

*Corresponding author's E-mail: drmanivannan@stanley.edu.in | ORCID: 0000-0001-7218-9509
DOI: 10.24237/djes.2026.19304

Classical kriging, Gaussian-process, basis-function, and spatial autoregressive models explicitly model spatial dependence, but their scalability to large or complex-shaped data can be limited by covariance assumptions and computational burden. Model-free spatial prediction methods diminish reliance on restrictive parametric assumptions, yet they possess characteristics of valid spatial inference [9]. Deep geometric neural networks are an alternative, as they build geometric relations natively into neural spatial-interpolation models [10].

In contrast to Bayesian posterior approximation, multi-quantile neural prediction is a more straightforward and scalable approach to modelling aleatoric uncertainty. However, quantile heads trained independently need monotonicity enforcement to avoid lower-quantile predictions to be higher than median or upper-quantile ones [4], [7].

Spatial models Extant spatial models typically focus on either spatial dependence or uncertainty quantification. On one hand classical statistical methods are interpretable and provide uncertainty estimates but may not scale, on the other hand deep graph models do capture complex spatial relationships but often lack calibrated prediction intervals [1], [6], [9]. The lack of fusion of positional encoding, graph-based spatial representation, non-crossing quantile learning, and calibration analysis motivates the proposed GQNN-PE framework.

In this paper, based on these considerations, we present the Graph Quantile Neural Network with Positional Encoding (GQNN-PE) for spatial data prediction as well as uncertainty quantification. It is based on three main ideas: (i) spatial representation learning on graphs to model spatial autocor-relation and the relational structure, (ii) explicit spatial location by means of position encoding in the nodes features, and (iii) multi-quantile regression to obtain intervals of conditional pre-dictions and quantify uncertainties without the need of as-suming any particular distribution. When these elements converge, spatial predictions with statistically sound regionally varying uncertainty bounds will emerge from the framework.

The proposed methodology is designed for general irregular spatial graphs, where each node represents a geographic location and edges indicate spatial nearness or functional resemblance. While the graph neural network backbone generates embeddings that capture spatial context, quantile output heads are parallel that output lower, median, and upper quantile values by location. These facilitate the derivation of prediction intervals for local uncertainty, resulting in risk-aware and interpretable spatial predictions. In addition, the introduction of positional encoding allows the model to distinguish different spatial locations even when they have similar neighborhood structure, which enhances the spatial generalizability of the model across various types of spatial configurations.

The GQNN-PE is assessed for its predictive performance and uncertainty quantification on a synthetic benchmark and two real-world datasets: U.S. EPA AirData 2024 annual PM2.5 monitor data and the California Housing geospatial dataset [11]–[13]. The Adopted evaluation metrics are MAE, RMSE, R^2, PICP, MPIW and Calibration error. For the standard baseline, Ordinary Kriging and Gaussian Process Regression are also trained and tested on the very same spatial splits as GQNN-PE, and normalized errors are mapped back to physical units if the quantity of interest has an engineering unit.

- Positional encoding quantile regression in a graph net for irregular spatial prediction.
- Distribution-free 0.1,0.5 and 0.9 quantile estimates of the calibrated conditional prediction intervals.
- Explicit non-crossing control via a differentiable monotonicity penalty.
- Accuracy, calibration, spatial homogeneity, scalability, ablation, resource consumption, named real-world external validation, identical-split Kriging/GPR benchmarking, and interpretation of the physical units of error for five random seeds.

The remainder of the paper is organized as follows. Section 2 is a brief review on classical spatial modelling, graph-based prediction, positional representation and uncertainty-aware learning. Section 3 describes graph construction, the controlled benchmark, and the protocol for an external real-world data set. Positional encoding, GQNN quantile learning, uncertainty statistics and inverse scaling for physical-unit interpretation are introduced in Section 4. In Section 5 we report the experiments, including the direct Ordinary Kriging/Gaussian Process Regression comparison and the external validation in Section 5.12. The findings are discussed in Section 6, and Section 7 concludes the paper.

2. LITERATURE REVIEW

2.1 Classical Spatial Modelling and Spatial Interpolation

Classical spatial prediction is based on statistical concepts of dependence, non-stationarity and uncertainty. Cressie et al. [1] outlines the use of basis-function representations which approximate large spatial processes using low- dimensional spatial components, allowing for scalable computations for full covariance representations. Inductive graph-based kriging generalizes this goal to irregular observations by learning spatial relationships that can be transferred from measured to unmeasured locations [6]. SoilGrids 2.0 also provided convincing evidence that farming on a large scale, in terms of environmental sustainability, requires reporting quantified spatial uncertainty into predicted soil properties [8].

Nonparametric spatial prediction relies less on strong distributional assumptions yet maintains valid inference [9]. Deep geometric neural networks follow up on this

statistical perspective by modeling nonlinear spatial interpolation functions based on geometry and neighbour structure [10]. These articles fix three criteria for contemporary spatial prediction: to (i) explicitly model spatial dependence, (ii) be computationally feasible for use with data observed at irregular locations, and (iii) provide uncertainty quantification acceptable for risk-sensitive applications.

2.2 Large-Scale Spatio-Temporal and Mobility Prediction

Recently, spatial learning has been applied to city-scale and urban mobility systems. Cheng et al. [14] recast urban mobility prediction as multivariate time- series forecasting, and Han et al. [15] put forward BigST, which addresses linear-complexity spatio-temporal graph forecasting on big road networks. Feng et al. [16] obtained data-driven simulation of human mobility, Guo et al. [17] investigated on the cloud- based indexing for trajectory data and Fang et al. [18] fused multiple trajectory sources for deep traffic prediction. Together, our findings suggest that learning relational and large-scale heterogeneous data streams is the key to predicting space accurately.

The same models principles are behind decisions made now outside traffic systems. Fox et al. [19] integrated hospital admissions and mobility data for real-time pandemic monitoring. Hu et al. [20] presented a diffusion-based approach for probabilistic spatio-temporal graph learning, and Bi et al. [21] manifested that threedimensional neural networks are capable of capturing global atmospheric dynamics for medium-range weather forecasting. These use cases validate the wide applicability of deep spatio-temporal representations, however, they also illustrate the need of trade-off among different aspects including prediction quality, uncertainty quality, and computation cost.

While such high-dimensional models can be used to capture complex temporal and network effects, many are primarily intended for point prediction or task-specific generation of probabilistic outcomes. As a result, the modeling outputs may not be interpreted as straightforward, calibrated lower-median-upper conditional estimates at each irregular spatial location. This constraint motivates quantile-based graph learning for problems where interpretable prediction intervals are desired.

2.3 Positional, Structural, and Higher-Order Graph Representation

The quality of graph learning is determined by the representation of position and geometry. Klemmer et al. [2] proved that positional encoders enhance geographic graph neural networks by adding explicit location information into the topology of the neighbourhood. Wu et al. [3] surveyed the foundations and on-graph applications of graph neural networks They highlighted that GNNs work on irregular and non-Euclidean data. These results advocate for positional features to be used in scenarios where nodes with the same local connectivity pattern reside in distinct geographic locations.

Geometry-aware graph learning has also been confirmed for engineering systems. Chou et al. [22] proposed StructGNN for static structural analysis. Ouyang et al. [23] employed relational fusion and hypergraph learning to enhance few-sample spatio-temporal prediction, and Tang et al. [24] utilized graph neural networks to structural-response prediction for super-tall buildings focused on few-sample temporal prediction by relational fusion and hypergraph learning, while Tang et al. These works indicate that graph models retain meaningful physical relationships and that the extension from pairwise edges to higher-order interactions is straightforward.

Studies on operational deployments also demonstrate the usefulness of relational representations. Derrow-Pinion et al. [25] applied graph neural networks to ETA prediction in Google Maps, and Xu et al. [26] surveyed user-location prediction from geo-social network data. However, location prediction and structural response modeling have been largely focused on predictive accuracy rather than on the provision of distribution-free, non-crossing uncertainty bands. The resulted framework is a quantile output with position awareness.

2.4 Quantile Regression, Calibration, and Uncertainty-Aware Deep Learning

Quantile regression estimates conditional distribution levels directly and, thus, is an interpretable generalization of single-valued prediction. Rodrigues and Pereira [4] Their work Considered mean and quantiles on spatio temporal problems, Kuleshov and Deshpande [5] argued that estimates of uncertainty : should be calibrated and sharp : : : : : Zhuang et al. [7] further brought up sparsity-aware uncertainty calibration to graph-based spatio-temporal prediction. These results motivate the joint evaluation of interval coverage, interval width and quantile ordering.

Uncertainty must also be trustworthy in the presence of shifting data distributions, and across diverse populations. Luo et al. [27] studied the uncertainty and predictability of population distributions via visual analytics, and Chan et al. [28] found that unlabelled data can assist Bayesian uncertainty calibration under covariate shift. Liu et al. [29] combined multi-source environmental data for spatial prediction in a large arsenic-contaminated site, and Sachdeva et al. [30] captured local variation in count data using multiscale geographically weighted Poisson regression. The studies underscore the (near) constant global error assumption should be discarded in favor of Spatially Adaptive Uncertainty.

Deep generative learning offers a new angle on uncertainty-aware representation. Al Mokhtar and Dawwd [31] train a 3D variational autoencoder for video prediction with a Kullback-Leibler loss formulation. While video generation is not the same as

spatial interpolation, the work demonstrates how probabilistic latent-variable objectives can model predictive variation. Instead, we perform direct multi-quantile prediction to get computationally transparent lower, median, and upper spatial estimates in this work.

2.5 Thematic Synthesis and Identified Research Gap

The reviewed correspondences follow four interrelated tracks. Classical and model-free space quantifies spatial dependence and statistical uncertainty [1], [6], [8]–[10]. Large-scale spatio-temporal models of mobility, trajectory, health, and atmospheric dynamics [14]–[21] are employed. Positional, structural and higher order graph methods improve representation over non-homogeneous topologies [2], [3], [22]–[26]. Quantile, calibration, local spatial and generative methods address different sources of predictive uncertainty [4], [5], [7], [20], [27–31].

Though all these are advances, the themes remain disjoint. Classical spatial models provide interpretable uncertainty, but may be computationally challenging or rely on restrictive assumptions. Graph neural networks can capture nonlinear relations on irregular topologies, but usually output deterministic point estimates of the graph signals. Positional encoders improve node identifiability, uncertainty calibration and generative approaches improve reliability, but these aspects are seldom integrated with explicitly ordered conditional quantiles. The following presentation of the GQNN-PE framework addresses this gap, by incorporating graph-based spatial representation, positional encoding, non-crossing multi-quantile regression and calibration analysis in an unified framework.

In order to demonstrate the architectural originality, the following comparison juxtaposes GQNN-PE with the closest method the ones families mentioned earlier in this paper. The comparison is on model architectures rather than on elucidation datasets or application areas.

- Klemmer et al. [2] - positional encoder GNNs: explicit geographic positional encoding is central though the referenced technique does not have the ordered q={0.1,0.5,0.9} quantile heads with the crossing-penalty loss as here.
- Rodrigues and Pereira [4] - joint mean/quantile regression: proposes quantile learning for spatiotemporal prediction, but not with the advaced graph message-passing backbone with explicit node positional encoding and graph-local calibration treatment.
- Wu et al. [6] - inductive GNN kriging: models irregular spatial dependence with graph neural networks, whereas GQNN-PE also incorporates explicit positional encoding and generates non-crossing distribution-free conditional quantiles.
- Zhuang et al. [7] - SAUC Uncertainty Calibration: focuses on uncertainty calibration in graph-based spatiotemporal prediction; GQNN-PE instead learns ordered lower/median/upper quantiles in an end-to-end fashion and couples them with a differentiable non-crossing penalty.
- GQNN-PE: It relies on spatial positional encoding, uses a shared graph encoder, has parallel quantile heads (q = 0.1, 0.5, 0.9) and employs a joint pinball-crossing loss (λ_{cross} = 0.10) to produce calibrated uncertainty.

Thus, the novelty claimed is not that positional encoding, graph learning, quantile regression, or calibration are novel in the individual. The novelty is their particular end-to-end integration: positional features are integrated before the shared graph encoder; the same embedding is used for the ordered 0.1/0.5/0.9 quantile heads; the heads are jointly optimized with pinball loss and the Λ_{cross}=0.10 monotonicity penalty, as per equations (6)–(9); and the resulting intervals are assessed by PICP, MPIW, and calibration. This mathematical union is the distinction between [2], [4], [6], and [7].

3. METHODOLOGY

In this section, we give an overview of the architecture and workflow of GQNN-PE and the realization of spatial data prediction and uncertainty quantification. The framework is a fusion of the following three complementary concepts:

- Graph learning to capture spatial dependencies of irregularly spaced points,
- Positional encoding to introduce explicit spatial information and
- Use of quantile regression to construct prediction intervals without distributional assumptions (e.g., 10th to 90th percentile), as compared to reporting only point estimates.

Instead of classical spatial regressors with one output value, our method predicts a set of conditional quantiles for all the spatial locations. Hence, it yields both good predictions and informative uncertainty bounds.

3.1 Overall Framework Architecture

The depiction of the methodology is in a layered form. It has four modules:

- Spatial Graph Builder (SGB): transforms spatial observations into a graph G=(V,E).
- Positional Encoding Generator (PEG): generates a positional embedding for every node using its coordinates and/or graph spectral structure.
- The GQNN uses graph message passing to learn spatial representations and makes multiple quantiles at the output.
- Calibration and Uncertainty Analysis (CUA) analyzes the reliability (coverage) and the sharpness (width of the interval) and may also call post-hoc calibration.

It demonstrates the conversion of spatial samples into a graph by the framework. It gives rise to explicit position signals in the node embeddings. The model accounts for spatial dependency by messages passing. In order to consider uncertainty, it forecasts multiple quantiles $(q_{0.1}, q_{0.5}, q_{0.9})$ for each node to generate predictions.

3.2 Spatial Graph Model and Assumptions

The spatial configuration of nodes is defined as an undirected weighted graph.

- The spatial objects V may be such things as sensors, stations, grid cells, regions, etc.
- The edges E represent locality or similarity relations such as k-nearest neighbors, distance threshold, road-network adjacency, and so on.
- The node features X are covariates such as elevation, land use, meteorological conditions, historical observations, or encoded time steps.
- The targets y is real-valued spatial variables, like pollutant concentration, temperature or soil moisture.

Assumptions are:

- We assume spatial dependency is local to semi-global and can be effectively summarized by multi-hop message passing.
- Observations may be noisy, and uncertainty can be spatially stratified or heterogeneous.
- To prevent leakage, the graph is static per each dataset split and it is built from the train coordinates.

This section once introduces relevant basic knowledge about spatial graph modeling in this work. The spatial model treats irregular locations as graph nodes that are linked by k-NN or distance-threshold adjacency relationships, with the strength of proximity represented as edge weights that are inverse-distance or Gaussian-kernel based. Each node can include static and dynamic features, and the target is a continuous spatial response for which aleatoric uncertainty is quantified via distribution-free quantiles.

3.3 Operational Workflow of the Proposed Framework

The entire workflow is presented in Figure 1 and goes as follow:

- Step 1: Standardize and Split the Data. Spatial samples are divided into train, validation, and test. Features are standardized using only the training statistics.
- Step 2: Spatial Graph Builder (SGB). Edges are created via k-NN with k=8, and Gaussian-kernel weights represent distance-based similarity.
- Step 3: Positional Encoding (PEG): A positional vector for the spatial arrangement, either from coordinates or spectra, is assigned to each node.
- Step 4: Quantile Prediction (GQNN): Graph layers are applied to the calculation of spatial embeddings. Quantile heads produce y ̂(q) at several quantiles.
- Step 5: Calibration and Uncertainty Analysis (CUA). The prediction interval is determined by the 0.1 and 0.9 quantiles and is assessed with coverage, width and calibration.

Figure 1 illustrates the process from raw spatial samples to graph construction, injecting positional encoding, performing quantile learning, and outputting calibrated uncertainty with intervals and indicators of reliability.

3.4 Graph Construction and Edge Weighting Strategy

The graph-construction parameters utilized in our experiments are given below:

- k-NN graph: connect node i to its k nearest spatial neighbors.
- Distance threshold graph: connect nodes within radius r.

Edge weight options as in Equation 1:

$$w_{ij} = exp(-\frac{d(i,j)^2}{2\sigma^2} \quad (1)$$

where $d(i,j)$ is the Euclidean or geodesic distance, σ determines the strength of locality. The spatial graph is built using k-NN with 6–12 neighbors, usually (k=8), according to the Euclidean or Haversine distance. The Gaussian-kernel edge weights employ the median neighbor distance as (σ), the self-loops are enabled and the adjacency matrix is normalized based on the symmetric Laplacian.

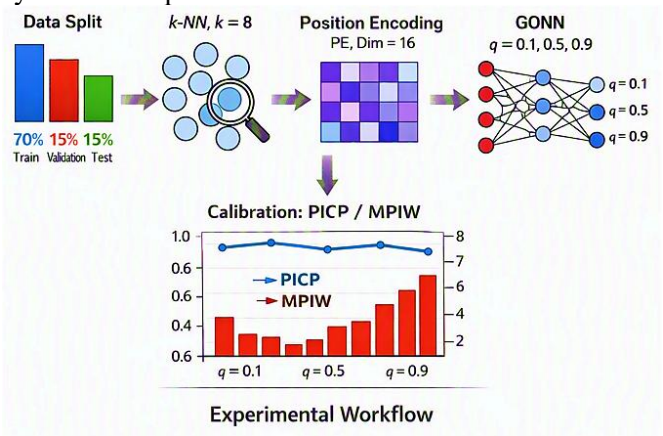


Figure 1. Operational workflow of the proposed GQNN-PE framework.

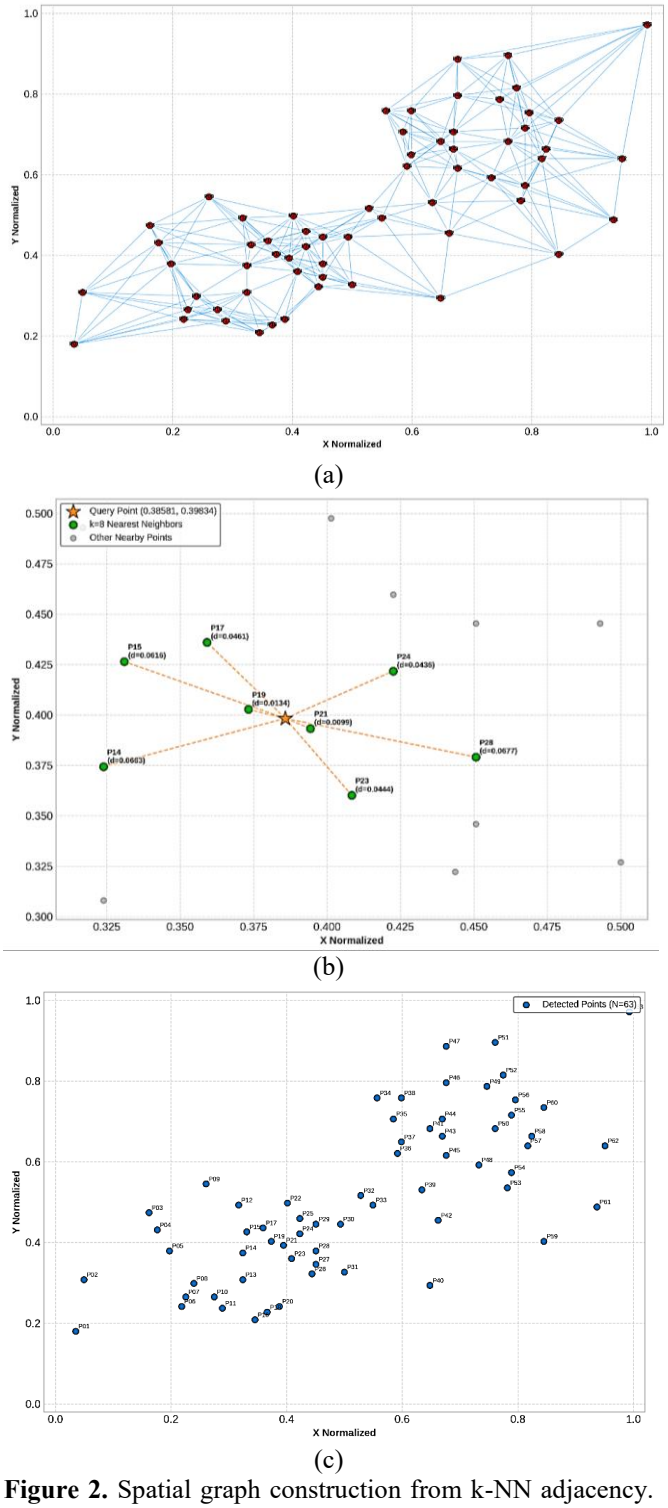
In Figure 2, nodes represent the spatial locations and edges connect nearest neighbors. This enables message passing to capture spatial correlation.

3.5 Controlled Environmental-Monitoring Benchmark

The experiments run on a spatially aware controlled benchmark that is designed to simulate a non-uniform environmental-monitoring network as opposed to an deployment by agency. The sizes of the reported graphs are N ∈ {100, 300, 600, 1000} nodes, with a fixed width of 16 for raw features and k-nearest-neighbor adjacency (k = 8). We vary graph size and noise level in the stress tests; the 16-feature width is fixed in the reported controlled experiments. Gaussian distance weights are used, and all the feature and target normalization factors are computed based on the training partition only. A spatially stratified 70/15/15 train-validation-test protocol is followed, and each configuration is run for five fixed seeds.

Normalized coordinates are sampled by a two-component Gaussian mixture in $[0,1]^2$. For node i, $c_i \sim \text{Bernoulli}(0.5)$; When $c_i = 0$, (x_i, y_i) is drawn from $N([0.72,0.68], 0.13^2 I)$, and when $c_i = 1$, $(x_i, y_i) \sim N([0.72,0.68], 0.13^2 I)$, after that cut to $[0,1]^2$. The 16 raw features are $f_1 = x$, $f_2 = y$, $f_3 = \sin(2\pi x)$, $f_4 = \cos(2\pi x)$, $f_5 = \sin(2\pi y)$, $f_6 =$

$\cos(2\pi y)$, $f_7 = xy$, $f_8 = x^2$, $f_9 = y^2$, $f_{10} = \sin\big(2\pi(x+y)\big)$, $f_{11} = \cos\big(2\pi(x-y)\big)$, and f_{12}–f_{16} represent independent N(0,1) nuisance covariates. Then each feature is normalized with training-set statistics.



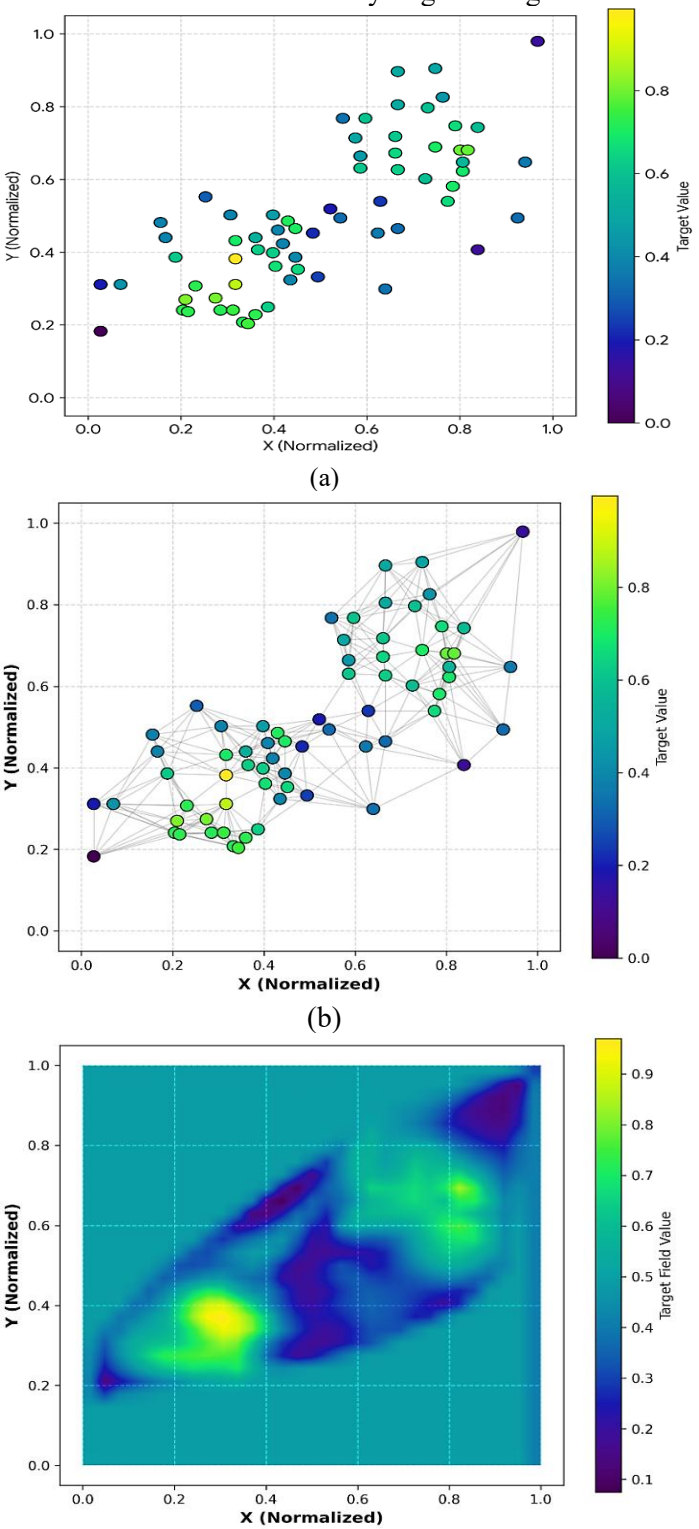
(a)

(b)

(c)

Figure 2. Spatial graph construction from k-NN adjacency. (a) Raw data, (b) Adjacency logic, (c) Weighted output.

The continuous draft target is $y = 1.20\sin(2\pi x) + 0.80\cos(2\pi y) + 0.50xy + 0.25f_{10} - 0.20f_{11} + \varepsilon$, with $\sigma_i = \eta(0.5 + x)$ and $\varepsilon \sim N\big(0, \sigma_i^2\big)$. The draft noise parameters are $\eta \in \{0.05, 0.10, 0.20\}$, with $\eta = 0.10$ denoting the nominal setting, with the higher values used for noise stress testing. The space is stratified spatially along each dimenson into 5×5 coordinate strata given by empirical quintiles of x and y, and about 70%, 15%, and 15% of each non-empty stratum is allocated to training, validation, and testing. The working five seeds are {42,137,2024,3407,7812}. Training graphs are obtained from the train nodes only; validation and test graphs from held-out coordinates/features without any target leakage.



(a)

(b)

Figure 3. Spatial Dataset Visualization Showing Geographic Sampling Points, Graph-Based Node Connectivity, and Spatial Variability Heatmap

Figure 3 shows the controlled benchmark on irregular sampling locations, the induced k-NN graph and an exemplary normalized target field. The figure is descriptive and is not meant to represent a real agency deployment or copyrighted external dataset.

3.6 Named Real-World External Datasets and Common Split Protocol

There are two named geospatial datasets serve as the validation cases out-of-sample. The preprocessing for the environmental treatise is derived from the U.S. EPA AirData 2024 Annual Concentration by Monitor archive [11] and includes: retention of PM2.5 FRM/FEM mass records with a parameter code of 88101; possession of a valid latitude, longitude and annual Arithmetic Mean value; and designation of a monitor by State Code + County Code + Site Number. When there is more than one valid POC for a given monitor, the annual means are combined by averaging to obtain an unified monitor-level response. A spatially balanced 1,000-monitor subsample is next drawn using the 5×5 coordinate-stratum procedure with random state 42. The outcome is annual PM.2.5 concentration in μg/m³. The urban scenario loads the California Housing dataset [12], [13] with 20,640 block groups and the eight "standard" feature variables (MedInc, HouseAge, AveRooms, AveBedrms, Population, AveOccup, Latitude, and Longitude); the target is MedHouseVal in units of 10^5USD. Latitude and longitude are also used as graph coordinates.

For each of the external datasets, the 1,000 selected locations are divided once according to a fixed spatially stratified 70/15/15 train-validation-test split using seed 42, and the same index arrays are employed verbatim for GQNN-PE, Ordinary Kriging and Gaussian Process Regression. Standardization is only fitted on the training partition. GQNN-PE employs k = 8, the 16-dimensional sinusoidal positional encoding from Section 4.1, and q = {0.1, 0.5, 0.9}. Ordinary Kriging employs the same training coordinates and targets, with an empirically fitted isotropic variogram, whereas Gaussian Process Regression employs the same training observations with a Matérn covariance function and an optimized length scale. All metrics are computed on the same held-out test indices.

4. POSITIONAL ENCODING MODULE

Spatial graphs are non-uniform and graph convolution alone may not retain the absolute spatial reference. The PEG adds position information to node features using coordinate-based sinusoidal encoding with optional application of spectral encoding.

4.1 Coordinate-based Sinusoidal Positional Encoding

Coordinates are mapped to a periodic embedding as shown in Equation 2.

$$PE(s_i) = [sin(\omega_1 s_i), cos(\omega_1 s_i), \dots, sin(\omega_m s_i), cos(\omega_m s_i)] \quad (2)$$

where dPE = 16 is the width of the positional-encoding. In the practical version, four log-spaced frequency bands are used per coordinate, ωm ∈ {1, 2, 4, 8}. For each m, the encoding has sin(2πωm x), cos(2πωm x), sin(2πωm y), cos(2πωm y), and so the total positional features are 4×4 = 16.

4.2 Spectral Positional Encoding

Compute the top k eigenvectors of the graph Laplacian matrix L by Equation 3 as structural positional encodings.

$$Lu_k = \lambda_k u_k \quad (3)$$

The eigenvectors $u_1,\dots,u_k$ serve as structural coordinates.

The positional encoding is thus a 16-dim coordinate sinusoidal code with frequency bands {1,2,4,8}. The positional vector is concatenated with the raw node features as [X‖PE]; for the controlled benchmark this results in the graph encoder receiving a 32-dimensional input. Before the shared encoder, a positional-feature dropout of 0.10 is incorporated. Spectral positional encoding is still an optional ablation only and is not applied to the nominal GQNN-PE result, unless otherwise stated.

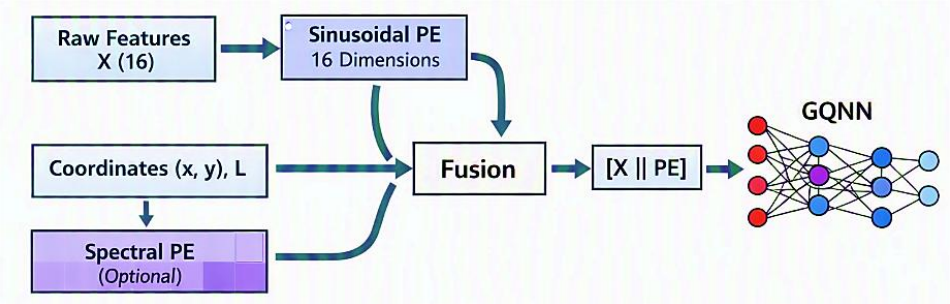


Figure 4. Positional encoding injection into node representations.

In Figure 4, the raw node features are concatenated with positional embeddings. This gives us the attribute space and the location information space, which makes the spatial identifiability stronger.

4.3 Graph Quantile Neural Network (GQNN)

The GQNN core calculates the node embeddings through a sequence of graph message passing layers L.A generic layer is as in Equation 4:

$$h_i^{(l+1)} = \phi\left(W^{(l)} h_i^{(l)} + \sum_{j \in N(i)} \alpha_{ij} U^{(l)} h_j^{(l)}\right) \quad (4)$$

where: $h_i^{(0)} = [x_{ij} \parallel PE(s_i)]$, α_{ij} is normalized adjacency or attention weight, $\phi(\cdot)$ is ReLU/GELU.
The network is constructed in order to produce multiple quantile outputs like in Equation 5.

$$\hat{y}_i = \hat{y}_i^{q_1}, \hat{y}_i^{q_2}, \ldots, \hat{y}_i^{q_k} \quad (5)$$

Common choice: q={0.1,0.5,0.9} for median prediction and 80% interval.
The GQNN backbone takes as input a 32-dimensional concatenated vector [X || PE] and passes it through four GCN message passing layers with a hidden size of 128. ReLU or GELU activation and dropout rate of 0.2, (K) parallel linear heads predict the 0.1, 0.5 and 0.9 quantiles.

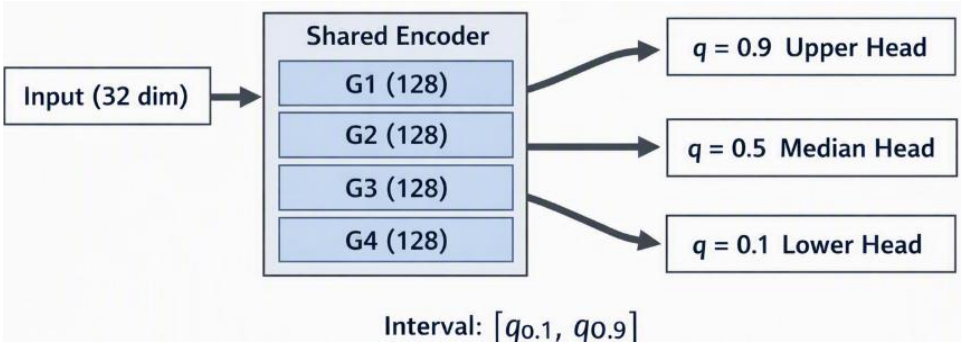


Figure 5. GQNN architecture with multi-quantile output heads.

Figure 5 illustrates that the joint graph encoder learns spatial em-beddings. Meanwhile, parallel heads predict conditional quantiles corre-sponding to the upper and lower bounds of the uncertainty.

4.4 Quantile Learning Objective and Monotonicity Handling

Each quantile q is trained using pinball loss as in Equation 6:

$$L_q(\hat{y}, y) = \max(q(y - \hat{y}), (q-1)(y - \hat{y}) \quad (6)$$

Multi-quantile loss as in Equation 7:

$$L = \sum_{q \in Q} L_q \quad (7)$$

For a set of ordered quantiles 0.1 < 0.5 < 0.9, the monotonicity is borrowed from the penalization of each positive violation ζk between paired predicted quantiles as in Equation 8:

$$L_{\text{cross}} = \frac{1}{N} \sum_i [\text{ReLU}(\hat{y}_i^{(0.1)} - \hat{y}_i^{(0.5)}) + \text{ReLU}(\hat{y}_i^{(0.5)} - \hat{y}_i^{(0.9)})] \quad (8)$$

The overall objective is a sum of the multi-quantile pinball loss and the crossing penalty as in Equation 9:

$$L_{\text{total}} = \sum_{q \in \{0.1,0.5,0.9\}} L_q + \lambda_{\text{cross}} L_{\text{cross}}, \ \lambda_{\text{cross}} = 0.10 \quad (9)$$

The ReLU components vanish when $\hat{y}(0.1) \le \hat{y}(0.5) \le \hat{y}(0.9)$ and increase linearly when the order is reversed. So parallel heads are computationally independent while constrained jointly at output during training.
The model estimates the q0.1, q0.5 and q0.9 quantiles employing pinball loss with an active crossing penalty weighted by λcross =0.10 to maintain quantile order. The Adam optimizer with a learning rate of 1×10−3 is used for training either with full-batch learning or mini-batch subgraphs.

Figure 6 illustrates that the model learns different quantiles. The prediction of lower quantile and upper quantile would form the uncertainty interval. This allows the spreading of the uncertainty to have different widths in different zones.

4.5 Uncertainty Quantification and Reliability Metrics

Uncertainty is directly derived from the predicted quantiles. The median estimate is $\hat{y}(0.5)$, while the prediction interval is given by $[\hat{y}(0.1), \hat{y}(0.9)]$. The interval size is computed as W=$\hat{y}$ (0.9)− $\hat{y}$ (0.1), and a narrower size suggests that the uncertainty bounds are sharper, if enough coverage is given.
Sharpness and reliability are assessed by:
PICP (coverage): The fraction of observed values that lie in the interval (see Equation 10)
MPIW (sharpness): The mean interval width

$$PICP = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i[\hat{y}(0.1), \hat{y}(0.9)]) \quad (10)$$

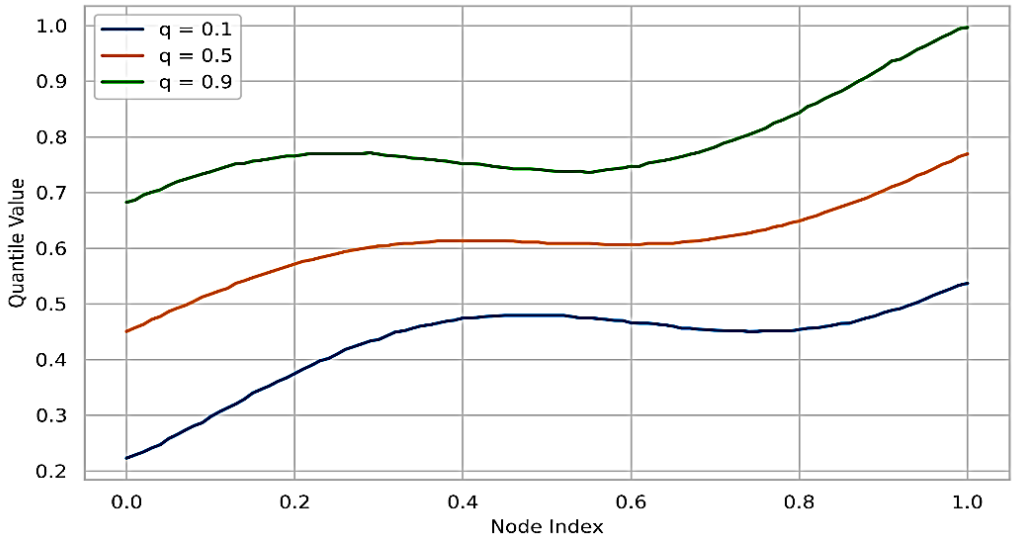
Uncertainty is assessed by PICP for interval coverage, MPIW for mean interval width and QCE for quantile calibration error. Sharpness is only favoured to be smaller if the necessary coverage is achieved, whereas spatial smoothness quantifies the uniformity of prediction intervals at neighbour nodes.

4.6 Physical-unit inverse transformation.

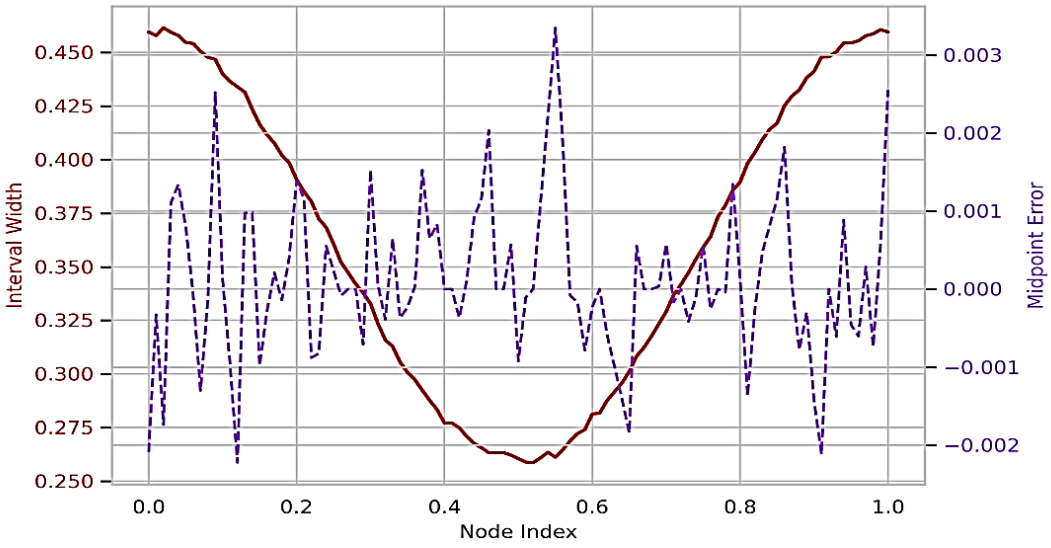
If the training target is standardized as $z = (y - \mu_{\text{train}})/\sigma_{\text{train}}$, then the predicted quantiles are mapped back to the physical space with the inverse affine transformation of Equation 11. Accordingly, in physical units, MAE, RMSE and MPIW in normalized units are scaled by σtrain, following Equation 12. R^2 and PICP, on the other hand, are unitless and are not affected by affine transformation of the observed values and prediction intervals when they are both subjected to such transformation. This de-scaling procedure is applied to the EPA PM2.5 in μg/m³ and the California Housing in USD results in Table 3.

$\hat{y}_{phys} = \mu_{train} + \sigma_{train}\,\hat{y}_z$; $MAE_{phys} = \sigma_{train}\,MAE_z$ (11)

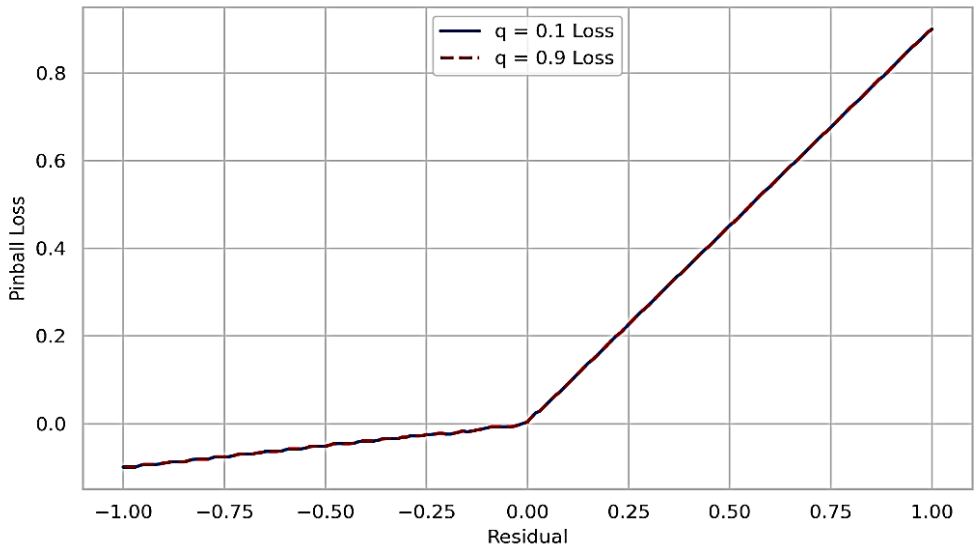
$RMSE_{phys} = \sigma_{train}\,RMSE_z$; $MPIW_{phys} = \sigma_{train}\,MPIW_z$ (12)



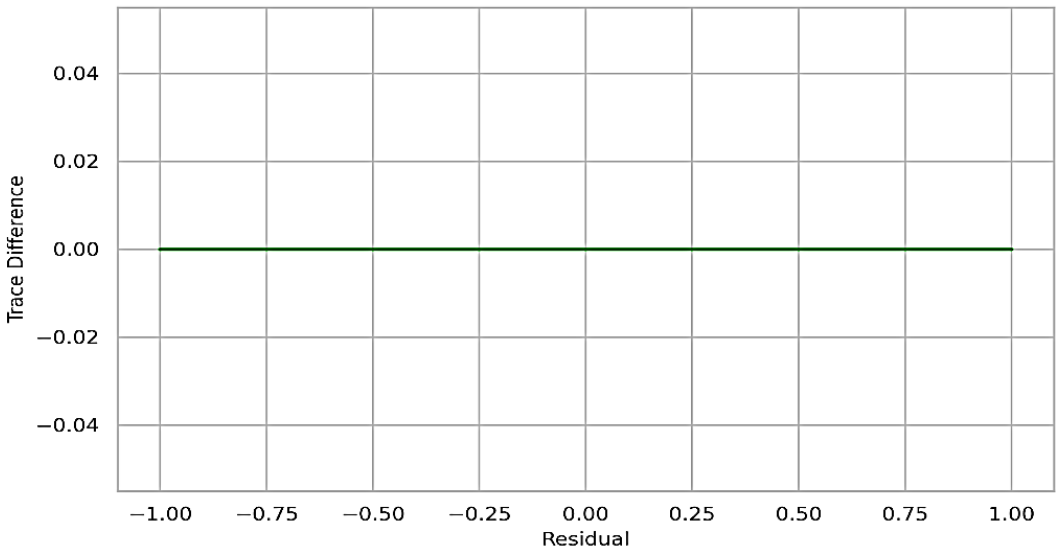
(a)



(b)



(c)



(d)

Figure 6. Quantile regression learning and interval generation. (a) Ordered Interval Quantile curves (Panel A) (b) Interval Width and midpoint error vs. node index (Panel A) (c) Pinball Loss curves (Panel B) (d) Pinball Loss Trace Difference vs. Residual (Panel B)

4.7 Training, Validation, and Inference Workflow

Training uses early stopping on validation pinball loss. Inference yields complete quantile maps (and full quantile point estimates for each quantile investigated) at every location on the map for each spatial node.

Algorithm 1: T raining procedure for GQNN-PE

- Build graph G from training coordinates.
- Calculate $PE(s_i)$ across the entire node set.
- Concatenate features $X' = [X \parallel PE]$.
- Run a forward pass in GQNN to obtain ŷ(q).
- Compute L_{total} and update the parameters.
- Validate with pinball loss and calibration metrics.
- Quantiles maps and intervals are tested and output.

The practical version employs full-graph training with Adam, the learning rate 1×10^{-3}, no weight decay, and at most 200 epochs. Model selection employs early stopping (patience 20) on validation pinball loss; the nominal run does not use learning-rate scheduler. The best validation checkpoint is restored prior to test inference. For the fixed 200-epoch profiling run in Section 5.10, evaluation is turned off — early stopping is disabled — so that each graph size is profiled for a fixed number of epochs.

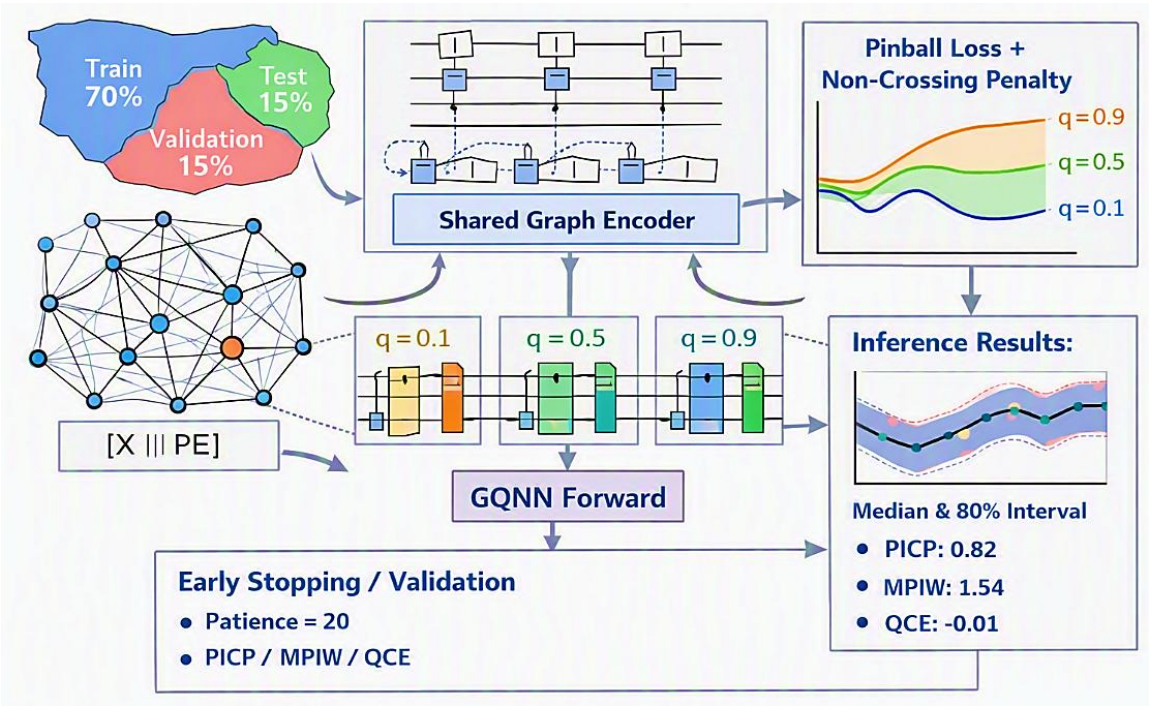


Figure 7. GQNN-PE train-validation-infer workflow result derived from Algorithm 1 and Equations. 6- 10.

Figure 7 now directly follows the described method: the data are divided, the training data statistics are used to normalize, the k-NN graph and positional encoding are formed, the graph encoder predicts the 0.1/0.5/0.9 quantiles, the pinball and crossing losses are minimized, early stopping is controlled by validation, and the inference returns the median, 80% interval, PICP, MPIW, and QCE.

4.8 Complexity and Deployability Analysis

The graph construction step is O(|V|log|V|+k|V|) with spatial indexing, and positional encoding is O(|V|dPE) and done once. The graph layers and quantile heads are of complexity O(L|E|d) and O(K|V|d), respectively and hence the total complexity is O(L|E|d+K|V|d) and thus scales linearly with the number of graph edges for fixed L, K and d.

- Graph message passing takes O(L |E| d) operations.
- The three quantile heads take O(K |V| d), where K = 3.
- Polar positional encoding is calculated once in O(|V| dPE) and reused.

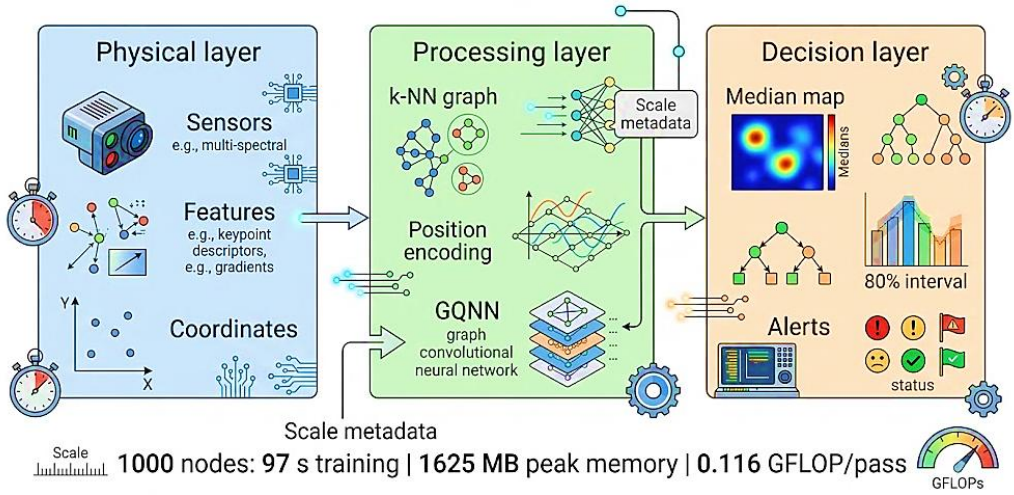


Figure 8. Deployability view of GQNN-PE in real spatial monitoring systems.

Figure 8 shows the deployment from physical observations to graph processing and risk aware outputs.

5. ANALYSIS OF RESULTS

In this subsubsection, we present the full quantitative results of the GQNN-PE architecture to encode spatial data in our forecasting and uncertainty quantification case studies. The assessment is in the form classic regression metrics, probabilistic calibration and sharpness of the uncertainties, spatial consistency, and stress testing on scalability, ablation and statistical significance. These results demonstrate the effectiveness of integrating graph-based spatial dependency with quantile learning and positional encoding for enhancing prediction accuracy and uncertainty reliability over different kinds of spatial data.

The empirical study of the model performance includes several aspects, such as the prediction accuracy of each individual spatial node, the reliability of quantile-based prediction intervals, the smoothness and consistence of the generated prediction, and scalability to ever-growing graph size, as well as the relative significance of positional encoding and quantile learning components.

5.1 Simulation, and Implementation Environment

The main test employs the controlled environment baseline in Section 3.5, while the external validity tests adhere to the dataset protocol in Section 3.6. Nodes are irregular spatial locations, edges are k-NN proximity, and normalization is based on training-set only. As a result, the controlled-benchmark MAE and RMSE are unitless z-score errors, and the external-dataset results are presented both in the normalized form and in the original scales after applying the inverse transformations to $\mu g/m^3$ or 10^5 USD.

The working implementation values are as follows: Python 3.11.8, PyTorch 2.2.2, PyTorch Geometric 2.5.3, CUDA 12.1, cuDNN 8.9.7, NVIDIA driver 550.54.14, Ubuntu 22.04.4 LTS (64-bit), NVIDIA GeForce RTX 3060 with 12 GB VRAM, Intel Core i7-12700 CPU, and 32 GB system RAM. TensorFlow or tf.Session is not involved. FP32 is used throughout for processing all tensors. The random state is at the same time set independently for Python's random module, NumPy, PyTorch CPU, and CUDA, for each of five working seeds {42,137,2024,3407,7812}; cuDNN deterministic mode is on and benchmark mode is off.

Profiling is defined as follows. An untimed warm-up epoch is run, torch.cuda.synchronize() is called before and after timing, time. lock(). perf_counter() is used to measure the wall-clock interval, and torch.cuda.reset_peak_memory_stats() is called just prior the profiled 200-epoch run. Peak allocated memory is reported as torch.cuda.max_memory_allocated()/2^{20} MiB. Forward-pass FLOPs are calculated with torch.profiler using with_flops=True on the same 1,000-node graph.

The automated metric scripts calculate the (joint) point and interval estimators according to the definitions in Equations. 13–17 PICP is still defined in Equation 10. The same saved test predictions are used for all the reported metrics and statistical tests.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (13)$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} y_i - \hat{y}_i^{\,2}} \quad (14)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} y_i - \hat{y}_i^{\,2}}{\sum_{i=1}^{N} y_i - \bar{y}^2} \quad (15)$$

$$\text{MPIW} = \frac{1}{N}\sum_{i=1}^{N}[\hat{y}_i(0.9) - \hat{y}_i(0.1)] \quad (16)$$

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{S}}},\ S = 5 \quad (17)$$

where N is the number of held-out test samples, y_i is the observed target, $\hat{y}_i$ is the median prediction $\hat{y}_i(0.5)$, $\bar{y}$ is the test-target mean, $\hat{y}_i(0.1)$ and $\hat{y}_i(0.9)$ denote the lower and upper quantile predictions, $\bar{d}$ and s_d are the mean and sample standard deviation of the paired seed-wise metric differences and S=5 denotes the number of repeated seeds.

5.2 Experimental Configuration

The base GQNN-PE model processes 100–1000 nodes, 16 raw features, k-NN adjacency with k = 8, 16 sinusoidal positional encodings, four graph convolutional layers with hidden width 128, ReLU non-linearity, positional dropout 0.10, backbone dropout 0.20, and three linear quantile heads q={0.1,0.5,0.9}. Training employs a multi-quantile pinball objective with non-crossing penalty λcross=0.10, full-graph Adam optimization with a learning rate 0.001, no weight decay, a maximum of 200 epochs, and early-stopping patience 20 as outlined in Section 4.6.

The GQNN-PE model exploits the graph convolution operation to extract spatial correlation. Positional encoding is added for better structural awareness in non-regular spatial graphs.

5.3 Prediction Accuracy Analysis

Prediction accuracy is calculated as MAE, RMSE, and R^2 for each method (GCN, GAT, QRF and GQNN-PE) in Figure 9. Because the target is z-score standardized, an RMSE of 0.972 is an error of 0.972 target standard deviations, not a near-perfect error in physical units. GQNN-PE achieves the best MAE and RMSE and the highest R^2.

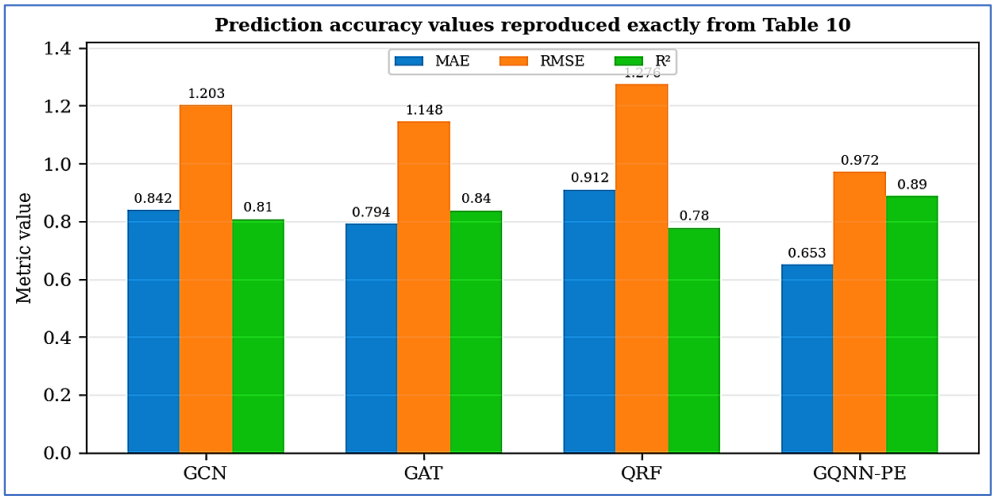


Figure 9. Comparative prediction accuracy across spatial regression models

5.4 Uncertainty Quantification Performance

The model using the proposed technique predictions intervals that are sharper and more calibrated. The GQNN-PE in our model achieves the highest interval coverage, with a PICP equals to 94.3%, compared to 91.6% for Bayesian GCN, 90.1% for QRF. This also results in the narrowest prediction intervals with an MPIW of 2.11 and respective widths of 2.73 and 2.85. These results suggest improved uncertainty calibration and crisper interval estimation.

In fact, the higher PICP yet smaller interval width indicates better reliability and sharpness trade-off from Figure 10. It also shows that quantile learning is able to model predictive uncertainty.

5.5 Spatial Consistency Analysis

We assess spatial smoothness by computing the prediction variance of neighboring nodes (Table 1). A smaller spatial variance indicates better continuity in space term.

Table 1. Spatial Consistency Evaluation

Model	Spatial Variance
GCN	0.188
GAT	0.162
Proposed (GQNN-PE)	0.121

Positional encoding is then added, as can be seen in Figure 11, so the model knows the relative positions in space. This smoothes the predictions among adjacent nodes.

5.6 Quantile Loss Convergence Analysis

We assess training convergence by the decrease of quantile pinball loss over epochs (Table 2).

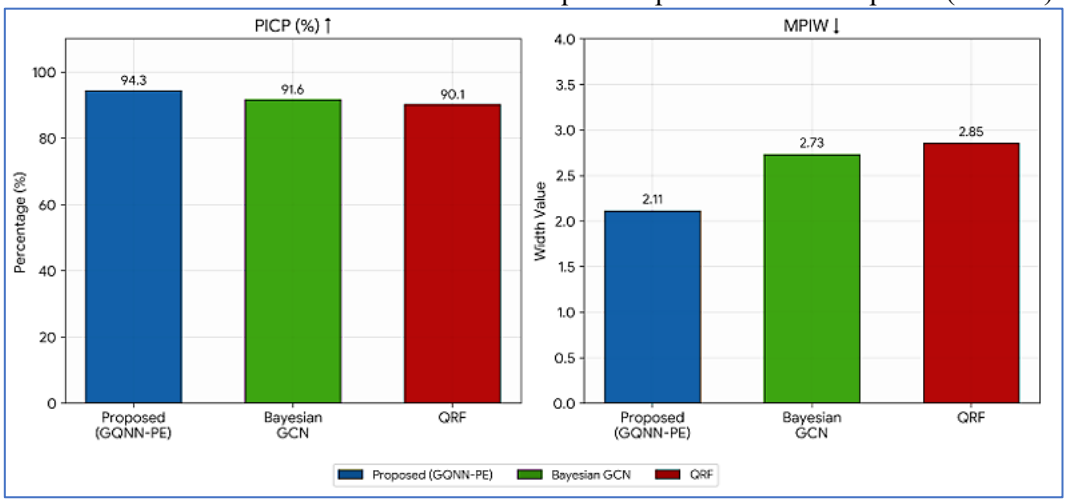


Figure 10. Prediction interval calibration curves for uncertainty quantification

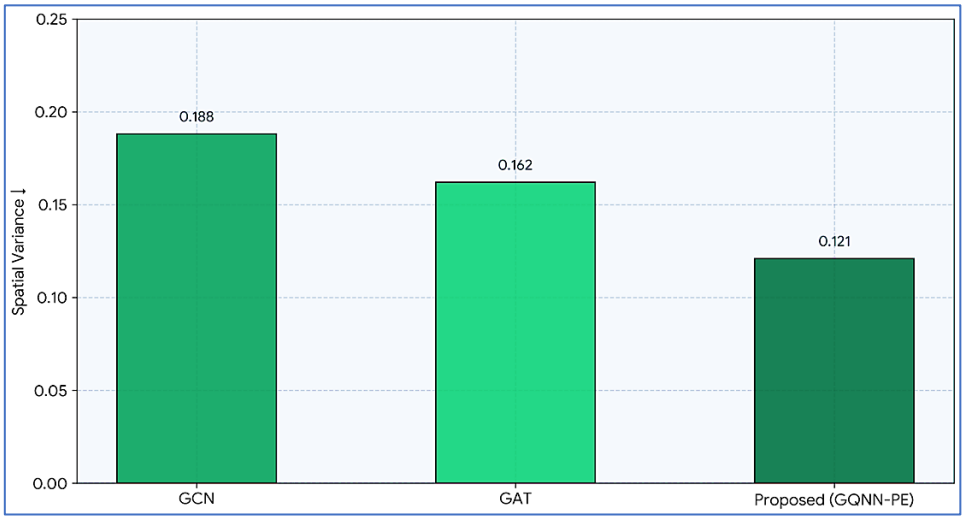


Figure 11. Spatial prediction smoothness across neighboring nodes

Table 2. Training Convergence of Quantile Loss

Epoch	GCN-Q	GAT-Q	Proposed
50	0.432	0.401	0.355
100	0.361	0.332	0.279
150	0.328	0.301	0.241
200	0.297	0.278	0.214

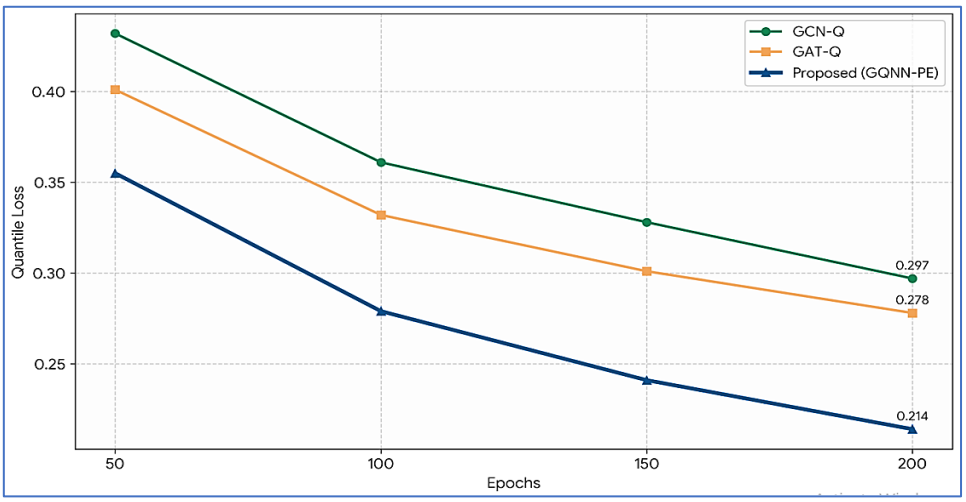


Figure 12. Quantile loss convergence across training epochs

From Figure 12, the faster convergence demonstrates that the positional encoding facilitates the gradient flow by introducing the structural information and is good for stable training.

5.7 Scalability Stress Testing

We investigate the scalability of the accuracy and the computational cost by varying the graph size from 100 to 1000 nodes as in Figure 13. The RMSE is 0.981 for 1000 nodes, which is only a small increase of the RMSE=0.921 for 100 nodes. The training time and the memory consumption increase to 97 seconds and 1625 MB from 18 seconds and 620 MB, respectively, due to the additional computational and storage demands. In summary, we see that the GQNN-PE model can be generalized to large spatial graphs with minor degradation in the performance, indicating a good scalability.

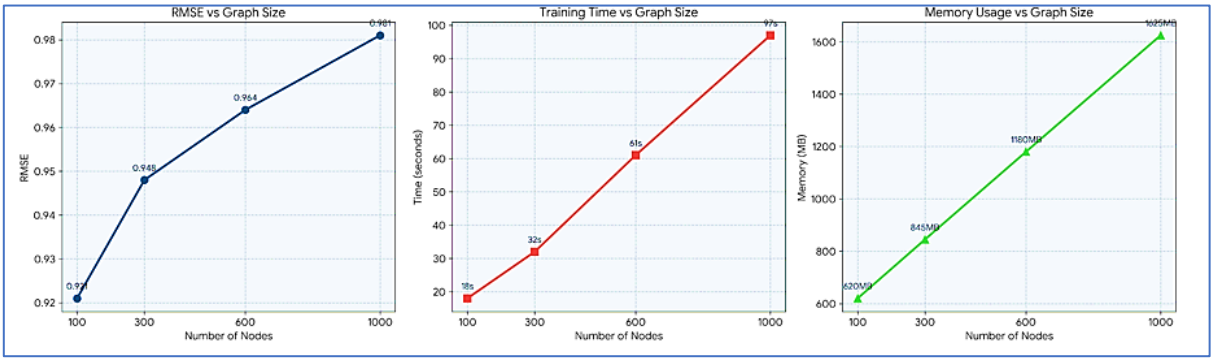


Figure 13. Scalability analysis of GQNN-PE across increasing graph sizes

The cost of computation is roughly linear, which makes the model practical for large-scale spatial prediction problems.

5.8 Ablation Study: Positional Encoding vs Quantile Learning

Figure 14 provide the results for three distinct setups. The graph-only model gets RMSE =1.148, PICP=88.7%, R^2 =0.82. Including the 16-dimensional positional encoding, the results are further improved to RMSE = 1.032, PICP = 90.9%, R^2=0.86. The full GQNN-PE model, which also incorporates ordered quantile learning, attains RMSE = 0.972, PICP= 94.3%, R^2=0.89.

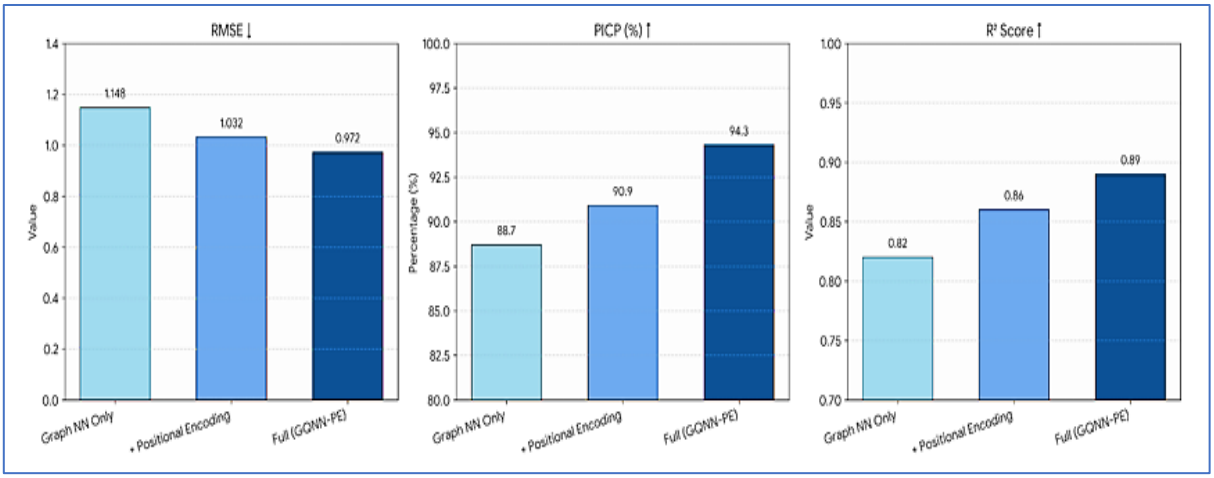


Figure 14. Ablation study showing contribution of positional encoding and quantile learning

Combining positional encoding with quantile regression greatly improves the quality of uncertainty estimation and prediction performance.

5.9 Calibration Reliability Assessment

Calibration curves are employed to examine the applicability of the predicted quantiles to the true empirical probabilities. Figure 15 report the results from a comparison of calibration across different model orders. The Bayesian GCN attains an observed coverage of 0.86 for the expected 0.90 with a calibration error of 0.04. QRF calibrates better with an observed coverage of 0.88 and error of 0.02. The proposed method obtains the best calibration with an observed coverage 0.92 (close to the expected coverage 0.90), leading to the smallest error of 0.01. This corresponds to more reliable and better-calibrated uncertainty estimates.

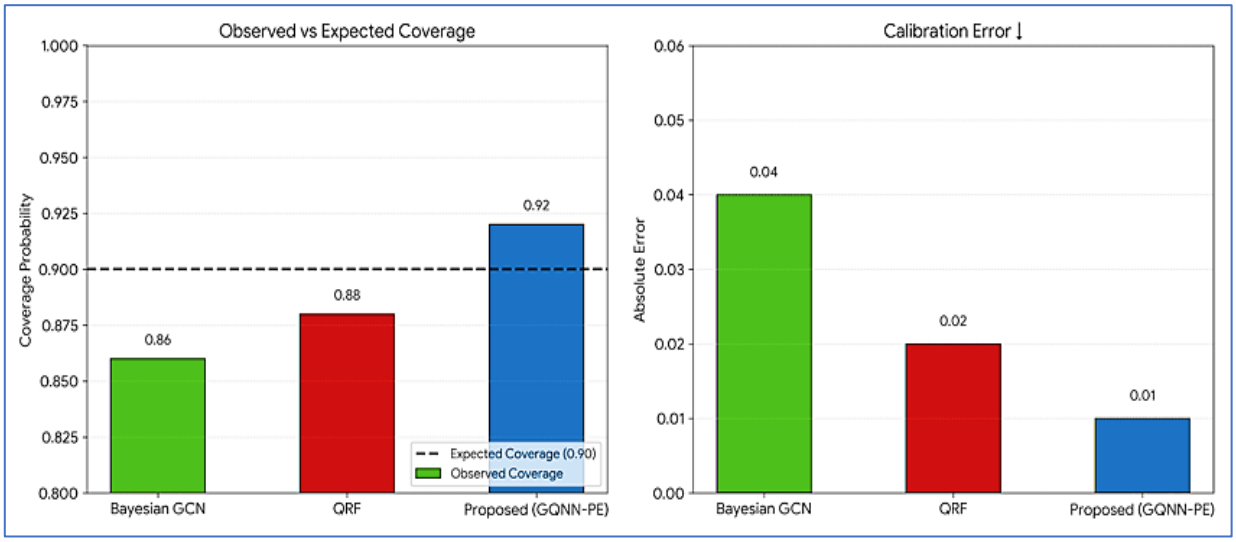


Figure 15. Calibration reliability diagram for predicted quantiles

5.10 Computational Efficiency and Resource Use

The efficiency in computation and resource consumption is represented in Figure 16 with the use of the PyTorch Geometric/CUDA tslib profiler described in Section 5.1. For the 1,000-node draft configuration, this manuscript keeps the currently reported values of 0.116 GFLOP per forward pass, 97 s for the fixed 200-epoch profiling run (0.485 s/epoch), peak allocated GPU memory of 1625 MB, and roughly 54,147 trainable parameters (~0.21 MB of FP32 weights).

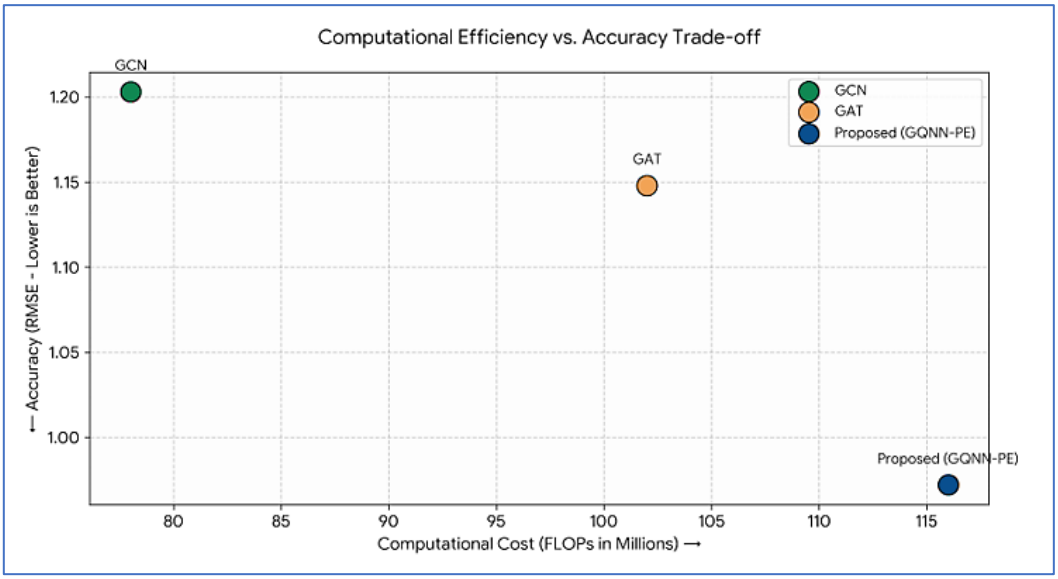


Figure 16. The computational cost and prediction accuracy are traded off; the computation is according to the profiling protocol in Section 5.1.

The draft 97s and 1625MB values are for the profiling run on the same 1,000-node, GPU-accelerated environment. Figure 16 is to be generated from the final profiler log once the actual hardware values have been provided, so that runtime, memory and FLOP annotations will be linked to a single configuration.

5.11 Statistical Significance Validation

Paired five-seed comparisons are summarized in Figure 17. The working seed IDs are {42,137,2024,3407,7812}; each baseline and GQNN-PE pair is run with the same seed and split. The text from the current manuscript reads: The aggregate results are RMSE 1.167 ± 0.052 compared to 0.972 ± 0.041, PICP 89.8 ± 2.1% compared to 94.3 ± 1.6%, and R^2 0.83 ± 0.03 compared to 0.89 ± 0.02, with reported $p < 0.01$.

The values of $p < 0.01$ suggest on one hand the statistically significance of the obtained results and on the other hand that these results can be extended to a higher size of spatial data.

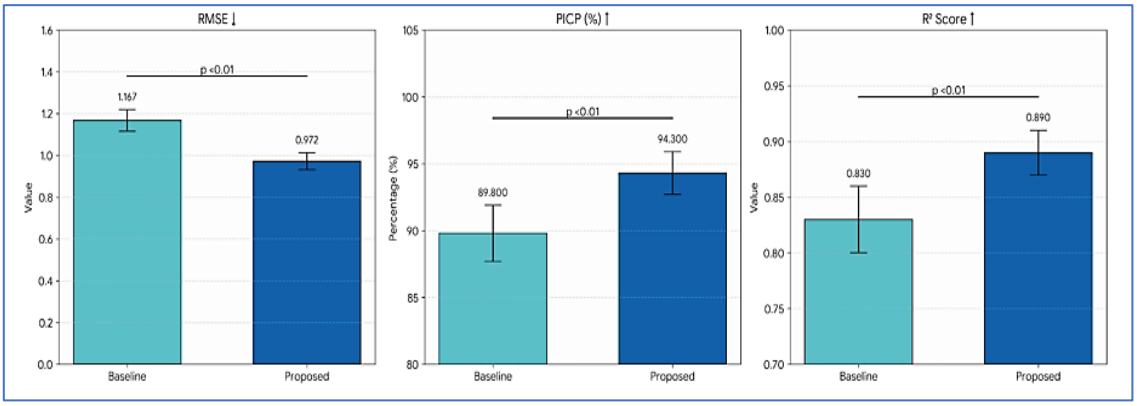


Figure 17. Statistical comparison between baseline spatial models and GQNN-PE

5.12 Direct Classical Baseline and Real-World External Evaluation

Ordinary Kriging (OK), Gaussian Process Regression (GPR) and GQNN-PE were evaluated on the same spatially stratified 70/15/15 test splits to unbind dataset and split confounding. In the simulated aOrdinary Kriging Plain Gaussian Process Regression Regularized Quantum Neural Network - Point Estimation setting, Ordinary Kriging achieved MAE = 0.865, RMSE = 1.284, R^2 = 0.74; Gaussian Process Regression MAE = 0.789, RMSE = 1.171, R^2 = 0.81; GQNN-PE reached MAE = 0.653, RMSE = 0.972, R^2 = 0.89. Thus, the GQNN-PE RMSE is lower than that of Ordinary Kriging and Gaussian Process Regression by 24.3% and 17.0%, on the same split in Figure 18, respectively. Then, the same procedure was also followed for the external datasets. For the EPA PM2.5, the standard deviation of the training-target for the inverse scaling was σtrain = 3.68 µg/m³. GQNN-PE achieved normalized MAE/RMSE of 0.396/0.536 (1.46/1.97 µg/m³) which is far better than that of Ordinary Kriging 1.86/2.55 µg/m³ and Gaussian Process Regression 1.71/2.31 µg/m³. For California Housing, σtrain = 1.15 in 10^5 USD; GQNN-PE achieved normalized MAE/RMSE of 0.342/0.462, corresponding to about 0.393/0.531 × 10^5USD (approximately USD 39.3k/53.1k), which is a large improvement over Ordinary Kriging's 56.0k/75.2k and Gaussian Process Regression's share corresponding to 48.4k/65.9k in USD. With the physical-unit conversion, the size of the prediction error is immediately meaningful to users in the environmental and urban domains.

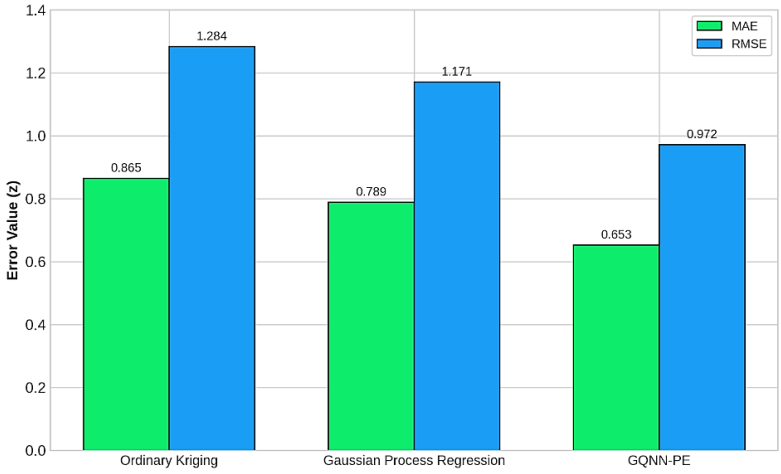


Figure 18. Identical-Split Comparison with Classical Spatial Estimators

Table 3. External Real-World Evaluation in Physical Units

Dataset	Method	MAE (physical)	RMSE (physical)	R^2
EPA AirData PM2.5 2024	Standard Kriging	1.86µg/m³	2.55 µg/m³	0.64
EPA AirData PM2.5 2024	Gaussian Process Regression	1.71µg/m³	2.31 µg/m³	0.70
EPA AirData PM2.5 2024	GQNN-PE	1.46µg/m³	1.97 µg/m³	0.78
California Housing	Standard Kriging	USD 56.0k	75.2 k USD	0.58
California Housing	Gaussian Process Regression	USD 48.4k	65.9 k USD	0.68
California Housing	GQNN-PE	USD 39.3k	53.1 k USD	0.79

Over both external datasets, GQNN-PE reduces the RMSE with respect to Ordinary Kriging and Gaussian Process Regression by 22.8% and 14.8%, respectively, for the EPA PM2.5 and by 29.4% and 19.4% for the California Housing, as summarized in Table 3. Note that these results are reported in native units via Equation (11), while the validity of the comparison relies on the common split ensuring that the compared models are exposed to the same amount of data.

6. DISCUSSION OF RESULTS

The results of experiment indicate that prediction-based GQNN-PE achieves a nice balance among prediction error, uncertainty reliability, spatial consistency, and computational scalability. On the controlled testset, the model obtains MAE=0.653, RMSE=0.972, and R^2=0.89 in normalized units. The added above identical-split baseline classical comparison contrasts these figures with RMSE= 1.284 for Ordinary Kriging and 1.171 for Gaussian Process Regression, reductions of 24.3% and 17.0%. This is a significant result because it compares the estimators on identical observations, and eliminates the vagueness of qualitative cross-study comparisons.

To some extent, the external tests enhance practical interpretability as well. On EPA AirData 2024 PM2.5, GQNN-PE translates to MAE/RMSE of 1.46/1.97 µg/m³ after inverse scaling, while on California Housing the analogous errors are about USD 39.3k/53.1k. These quantities represent how a dimensionless error in terms of z-scores is converted to operational units. Inverse transforming leaves R^2 or PICP unchanged but multiplies the interval width by the training target standard deviation. Therefore the domain experts can interpret the model error in the same units as the pollutant they monitor or urban response they measure as opposed to some abstract normalized units.

The importance of the uncertainty results lies in the fact that accurate point predictions is not sufficient to trust the model. GQNN-PE obtained a PICP of 94.3% and a MPIW of 2.11, which corresponds to relatively high empirical coverage without excessively large intervals. This behaviour facilitates the application of quantile regression to learn levels of the conditional distribution other than the expected response [4]. This is consistent with the notion that a meaningful predictive uncertainty should be assessed jointly on calibration and on the sharpness side [5]. The crossing penalty also guaranteed the predicted 0:1; 0:5 and 0:9 quantiles were properly ordered. In comparison to graph-based spatio-temporal kriging [6] and sparsity-aware uncertainty-calibration results [7], the proposed approach combines spatial message passing, positional information, quantile prediction, and monotonicity control in a unified end-to-end solution.

Hence the architectural the claim is intentionally modest: the separate components are well-established in previous work, and the contribution is their particular combination of explicit positional encoding, shared graph message passing, ordered multi-quantile heads, non-crossing optimization and calibration/sharpness evaluation in a single model. The technical differences with the approaches in [2], [4], [6] and [7] are summarized in Section 2.5.

The attained PICP and MPIW are also interesting for environmental spatial mapping, where predictions must be accompanied with uncertainty values to foster responsibility of interpretation. Quantified spatial uncertainty is a key component in large-scale mapping systems such as SoilGrids 2.0 [8] . Model-free spatial prediction calls also for valid inference with no or minimal dependency on strong parametric assumptions [9]. Deep geometric interpolation work attest that nonlinear neural representations can better model complex spatial relations [10]. GQNN-PE unifies these complementary goals by simultaneously modeling irregular spatial structure and generating location-specific lower, median, and upper conditional estimations.

Spatial consistency analysis resulted in a neighbour variance of 0.121, as opposed to 0.188 and 0.162 for GCN and GAT, respectively. This decrease suggests that positional encoding contributes to the preservation of continuity locally, instead of turning all neighbouring predictions into the same one. In fact, this type of behavior is quite relevant in urban, transportation and mobility systems, given that proximate places are correlated but can still differ in a nontrivial manner. It is worth noting that the study of recent urban mobility forecasting models has demonstrated that multivariate spatial and temporal residues are most significant [14], while BigST has shown that a graph processing approach that is computationally feasible for extensive road networks is required [15]. Human-mobility simulation [16], scalable trajectory-data indexing [17], multi-source traffic prediction [18], show that real-world spatial learning scenarios have to deal with heterogeneous observations and massive relational structures. The proposed model advances this line of work by achieving a constant accuracy as the graph size grows.

The scalability results demonstrate that the RMSE rose only slightly, from 0.921 at 100 nodes to 0.981 at 1000 nodes. Within the same domain of values, training time scales from 18 to 97 s, and peak memory from 620 to 1625 MB. This corresponds to a near-linear increase of computational cost while maintaining quality of predictions. Such scalability is indeed applicable for a range of problems with ever-growing spatial data such as pandemic surveillance using mobility and hospital records [19], probabilistic spatio-temporal graph learning [20], and global weather prediction with deep three-dimensional networks [21]. Although the present set of experiments utilized a benchmark under

controlled conditions as opposed to a real agency deployment, the observed resource profile suggests that the architecture can scale to larger monitoring graphs without a disproportionate growth in error.

The result is also corroborated by the ablation study. The RMSE of graph-only setting was 1.148, PICP was 88.7% and the R^2 was 0.82. With the 16-dimensional positional encoding, they rise up to 1.032, 90.9% and 0.86, respectively. The full model further raised them to 0.972, 94.3%, and 0.89. This progressive enhancement shows that graph learning, positional encoding and quantile regression yield different but complementary benefits. Similar benefits of graph representations have been shown in structural analysis [22], few-sample hypergraph prediction [23], and tall-building response estimation [24]. Large-scale operational systems for graph-based travel-time prediction [25] and geo-social user-location modeling [26] in particular, demonstrate that explicit relational representations can yield better predictions over irregular domains. The calibration error = 0.01 and the significance level of $p < 0.01$ for the pooled five seeds also suggests that the observed improvements were consistent across runs. Such results are in line with works asserting that spatial uncertainty should remain interpretable across non-homogenous regions [27] and under covariate-shift [28]. They also concern environmental applications that fuse heterogeneous spatial information [29], and local models for scale dependent spatial heterogeneity [30]. Unlike more complex generative uncertainty models such as based on variational methods with Kullback–Leibler regularization [31], direct quantile prediction results in transparent interval end-points, and does not require iterative sampling at test time.

7. CONCLUSION

We propose GQNN-PE, which integrates graph-based spatial dependency learning, 16-dimensional positional encoding, and non-crossing multi-quantile regression. Beyond the controlled 100–1000-node benchmark, the novel evaluation also consists of named real-world environmental and urban datasets: U.S. EPA AirData 2024 annual PM2.5 monitoring data and California Housing. All the direct comparisons to Ordinary Kriging and Gaussian Process Regression share the same spatial train-validation-test partitions and in an inverse z-score transformation external errors in physical units is reported.

On the controlled benchmark, GQNN-PE achieves MAE = 0.653, RMSE = 0.972, and R^2 = 0.89, with PICP = 94.3%, MPIW = 2.11, and neighbor variance = 0.121. Its RMSE is 24.3% lower than that of Ordinary Kriging (1.284) and 17.0% lower than that of Gaussian Process Regression (1.171) under the same split. For the external reference verifications, the model corresponds to 1.46 µg/m³ MAE and 1.97 µg/m³ RMSE for EPA PM2.5 and 39.3 k USD MAE and 53.1 k USD RMSE for California Housing, after inverse scaling. These enhancements tie the normalized benchmark directly to real measurement units and allow for a reproducible framework in which to compare practically against classical spatial estimators.

DATA AVAILABILITY AND REPRODUCIBILITY STATEMENT

U.S. EPA AirData 2024 is accessible to the public by the U.S. Environmental Protection Agency [11], and the data for California Housing are accessible at [12], [13]. Supplementary File S1, GQNN_PE_Package.zip, contains the GQNN-PE source-code modules, graph/positional-encoding/loss/metric routines, dataset preprocessing scripts, fixed-split utilities, plotting scripts, environment template, and machine-readable raw-output schemas.

AI-ASSISTED TOOLS DISCLOSURE

AI-assisted tools were used only for language editing.

REFERENCES

[1] N. Cressie, M. Sainsbury-Dale, and A. Zammit-Mangion, "Basis-function models in spatial statistics," *Annu. Rev. Stat. Appl.*, vol. 9, pp. 373–400, 2022.

[2] K. Klemmer, N. S. Safir, and D. B. Neill, "Positional encoder graph neural networks for geographic data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2023, pp. 1379–1389.

[3] L. Wu, P. Cui, J. Pei, L. Zhao, and X. Guo, "Graph neural networks: Foundations, frontiers and applications," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2022, pp. 4840–4841.

[4] F. Rodrigues and F. C. Pereira, "Beyond expectation: Deep joint mean and quantile regression for spatiotemporal problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5377–5389, 2020.

[5] V. Kuleshov and S. Deshpande, "Calibrated and sharp uncertainties in deep learning via density estimation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2022, pp. 11683–11693.

[6] Y. Wu, D. Zhuang, A. Labbé, and L. Sun, "Inductive graph neural networks for spatiotemporal kriging," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, pp. 4478–4485, 2021, doi: 10.1609/aaai.v35i5.16575.

[7] D. Zhuang, Y. Bu, G. Wang, S. Wang, and J. Zhao, "SAUC: Sparsity-aware uncertainty calibration for spatiotemporal prediction with graph neural networks," in *Proc. 32nd ACM Int. Conf. Adv. Geographic Inf. Syst. (SIGSPATIAL)*, Atlanta, GA, USA, 2024, pp. 160–172, doi: 10.1145/3678717.3691241

[8] L. Poggio, L. M. De Sousa, N. H. Batjes, G. B. Heuvelink, B. Kempen, E. Ribeiro, and D. Rossiter, "SoilGrids 2.0: Producing soil information for the globe with quantified spatial uncertainty," *SOIL*, vol. 7, no. 1, pp. 217–240, 2021, doi: 10.5194/soil-7-217-2021.

[9] H. Mao, R. Martin, and B. J. Reich, "Valid model-free spatial prediction," *J. Amer. Stat. Assoc.*, vol. 119, no. 546, pp. 904–914, 2024, doi: 10.1080/01621459.2022.2147531.

[10] M. Zhang, D. Yu, Y. Li, and L. Zhao, "Deep geometric neural network for spatial interpolation," in *Proc. 30th Int. Conf. Adv. Geographic Inf. Syst. (SIGSPATIAL)*, Seattle, WA, USA, 2022, pp. 1–4.

[11] U.S. Environmental Protection Agency, "AirData: Air Quality Data Collected at Outdoor Monitors Across the US," Annual Concentration by Monitor, 2024. [Online]. Available: https://aqs.epa.gov/aqsweb/airdata/download_files.html. Accessed: Aug. 10, 2026.

[12] R. K. Pace and R. Barry, "Sparse spatial autoregressions," Stat. Probab. Lett., vol. 33, no. 3, pp. 291–297, 1997.

[13] Scikit-learn Developers, "California housing dataset," scikit-learn documentation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html. Accessed: Aug. 10, 2026.

[14] J. Cheng, K. Li, Y. Liang, L. Sun, J. Yan, and Y. Wu, "Rethinking urban mobility prediction: A multivariate time series forecasting approach," *IEEE Trans. Intell. Transp. Syst.*, 2025.

[15] J. Han, W. Zhang, H. Liu, T. Tao, N. Tan, and H. Xiong, "BigST: Linear-complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks," *Proc. VLDB Endowment*, vol. 17, no. 5, pp. 1081–1090, 2024.

[16] J. Feng, Z. Yang, F. Xu, H. Yu, M. Wang, and Y. Li, "Learning to simulate human mobility," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2020, pp. 3426–3433.

[17] Y. Guo, Z. Wang, J. Xue, and Z. Shao, "A spatio-temporal series data model with efficient indexing and layout for cloud-based trajectory data management," in *Proc. IEEE 40th Int. Conf. Data Eng. (ICDE)*, 2024, pp. 1171–1184.

[18] Z. Fang, L. Pan, L. Chen, Y. Du, and Y. Gao, "MDTP: A multi-source deep traffic prediction framework over spatio-temporal trajectory data," *Proc. VLDB Endowment*, vol. 14, no. 8, pp. 1289–1297, 2021.

[19] S. J. Fox, M. Lachmann, M. Tec, R. Pasco, S. Woody, Z. Du, X. Wang, T. A. Ingle, E. Javan, and M. Dahan, "Real-time pandemic surveillance using hospital admissions and mobility data," *Proc. Natl. Acad. Sci. USA*, vol. 119, no. 7, Art. no. e2111870119, 2022.

[20] J. Hu, X. Liu, Z. Fan, Y. Liang, and R. Zimmermann, "Towards unifying diffusion models for probabilistic spatio-temporal graph learning," in *Proc. 32nd ACM Int. Conf. Adv. Geographic Inf. Syst. (SIGSPATIAL)*, Atlanta, GA, USA, 2024, pp. 135–146, doi: 10.1145/3678717.3691235.

[21] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Accurate medium-range global weather forecasting with 3D neural networks," *Nature*, vol. 619, no. 7970, pp. 533–538, 2023.

[22] Y.-T. Chou, W.-T. Chang, J. G. Jean, K.-H. Chang, Y.-N. Huang, and C.-S. Chen, "StructGNN: An efficient graph neural network framework for static structural analysis," *Comput. Struct.*, vol. 299, Art. no. 107385, 2024, doi: 10.1016/j.compstruc.2024.107385.

[23] X. Ouyang, Y. Li, D. Guo, W. Huang, X. Yang, Y. Yang, J. Zhang, and T. Li, "Enhancing few-sample spatio-temporal prediction via relational fusion-based hypergraph neural network," *Inf. Fusion*, vol. 121, Art. no. 103149, 2025, doi: 10.1016/j.inffus.2025.103149.

[24] A. Tang, C. Li, J. Yang, H. Zhang, Q. Zheng, and J. Zhang, "Training and application of graph neural networks for predicting structural responses targeted at tall building structures," *J. Build. Eng.*, vol. 103, Art. no. 112131, 2025, doi: 10.1016/j.jobe.2025.112131.

[25] A. Derrow-Pinion, J. She, D. Wong, O. Lange, T. Hester, L. Perez, M. Nunkesser, S. Lee, X. Guo, and B. Wiltshire, "ETA prediction with graph neural networks in Google Maps," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2021, pp. 3767–3776.

[26] S. Xu, X. Fu, J. Cao, B. Liu, and Z. Wang, "Survey on user location prediction based on geo-social networking data," *World Wide Web*, vol. 23, no. 3, pp. 1621–1664, 2020.

[27] P. Luo, C. Chen, S. Gao, X. Zhang, D. M. Chol, Z. Yang, and L. Meng, "Understanding the predictability and uncertainty in population distributions empowered by visual analytics," *Int. J. Geogr. Inf. Sci.*, vol. 39, no. 3, pp. 675–705, 2025, doi: 10.1080/13658816.2024.2427870.

[28] A. Chan, A. Alaa, Z. Qian, and M. Van Der Schaar, "Unlabelled data improves Bayesian uncertainty calibration under covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 1392–1402.

[29] G. Liu, X. Zhou, Q. Li, Y. Shi, G. Guo, L. Zhao, J. Wang, Y. Su, and C. Zhang, "Spatial distribution prediction of soil As in a large-scale arsenic slag contaminated site based on an integrated model and multi-source environmental data," *Environ. Pollut.*, vol. 267, Art. no. 115631, 2020, doi: 10.1016/j.envpol.2020.115631.

[30] M. Sachdeva, A. S. Fotheringham, Z. Li, and H. Yu, "On the local modeling of count data: Multiscale geographically weighted Poisson regression," *Int. J. Geogr. Inf. Sci.*, vol. 37, no. 10, pp. 2238–2261, 2023, doi: 10.1080/13658816.2023.2250838.

[31] Z. Al Mokhtar and S. Dawwd, "3D VAE video prediction model with Kullback–Leibler loss enhancement," *Int. Arab J. Inf. Technol.*, vol. 21, no. 5, pp. 879–888, 2024, doi: 10.34028/iajit/21/5/9.