
DiffLiG: Diffusion-enhanced Liquid Graph with Attention Propagation for Grid-to-Station Precipitation Correction

Yuxiang Li¹, Yang Zhang¹, Guowen Li¹, Mengxuan Chen², Meng Jin⁴, Fang Wang⁵,
Haohuan Fu^{2,3}, Juepeng Zheng^{1,3,†*}

¹Sun Yat-sen University ²Tsinghua University

³National Supercomputing Center in Shenzhen ⁴Huawei Technologies Co., Ltd

⁵CMA Earth System Modeling and Prediction Centre

Abstract

Modern precipitation forecasting systems, including reanalysis datasets, numerical models, and AI-based approaches, typically produce coarse-resolution gridded outputs. The process of converting these outputs to station-level predictions often introduces substantial spatial biases relative to station-level observations, especially in complex terrains or under extreme conditions. These biases stem from two core challenges: (i) **station-level heterogeneity**, with site-specific temporal and spatial dynamics; and (ii) **oversmoothing**, which blurs fine-scale variability in graph-based models. To address these issues, we propose **DiffLiG** (**Diffusion-enhanced Liquid Graph with Attention Propagation**), a graph neural network designed for precise spatial correction from gridded forecasts to station observations. DiffLiG integrates a GeoLiquidNet that adapts temporal encoding via site-aware OU dynamics, a graph neural network with a dynamic edge modulator that learns spatially adaptive connectivity, and a Probabilistic Diffusion Selector that generates and refines ensemble forecasts to mitigate oversmoothing. Experiments across multiple datasets show that DiffLiG consistently outperforms other methods, delivering more accurate and robust corrections across diverse geographic and climatic settings. Moreover, it achieves notable gains on other key meteorological variables, underscoring its generalizability and practical utility.

1 Introduction

Weather forecasting has gained increasing attention in both research and society, driven by rapid advances in forecast accuracy and temporal resolution. On one hand, traditional Numerical Weather Prediction (NWP) systems, such as the European Centre for Medium-Range Weather Forecasts (ECMWF)’s Integrated Forecasting System - High Resolution (IFS-HRES) and Integrated Forecasting System - Ensemble (IFS-ENS), have achieved remarkable success in multiscale forecasts through high-resolution physical modeling, data assimilation, and ensemble strategies. On the other hand, the recent surge in AI-based weather models has introduced a data-driven paradigm that bypasses explicit physical parameterizations [27, 32, 34], offering superior scalability and efficiency. For instance, FourCastNet [30] leverages Fourier neural operators for spatio-temporal coupling, Pangu-Weather [5] employs 3D attention to improve mid-range accuracy, and GraphCast [22] utilizes spherical GNNs for 10-day forecasts. FuXi [10] demonstrates a strong potential to extend the forecast horizons through a hierarchical transformer framework tailored for long-range weather prediction.

*[†]Corresponding author: zhengjp8@mail.sysu.edu.cn

Despite these advances, both the NWP and AI models primarily operate on regular-resolution grid outputs, which are insufficient for real-world applications requiring high-resolution, station-level forecasts. In practice, operational users (such as urban planners, airport operators, or hydrological managers) require precise local forecasts at specific stations. However, model outputs often fail to generalize across diverse topographies or under sparse observational coverage, leading to spatial biases and degraded local accuracy. We provide a detailed empirical analysis of such interpolation errors in the Appendix C.

A further concern arises from the evaluation process itself: most AI-based forecasting models are trained and validated on reanalysis data such as ERA5 [17], which integrates historical observations with simulated fields. As illustrated in Figure 6, this integration can introduce systematic biases, particularly in regions with sparse observations or complex terrain. While temporally consistent and spatially smooth, these products are not equivalent to real-world station observations. As noted by Ramavajjala & Mitra [33], models like FourCastNet may outperform NWP systems in ERA5, but fail to retain such advantages when evaluated in ground-truth datasets (*e.g.*, NOAA’s MADIS), exposing a considerable gap between simulation data and realism data in current AI weather systems.

Therefore, to enable accurate and reliable station-level forecasting, it is crucial to move beyond uniform, coarse-to-fine interpolation strategies and develop models that can adapt to the heterogeneous nature of stations while preserving local spatial structures. This calls for a dedicated interpolation framework that is both station-aware and uncertainty-informed, capable of correcting the structural mismatch between gridded forecasts and real-world observations.

Station-level interpolation faces two fundamental challenges: **station heterogeneity** and **oversmoothing**. The former refers to the diverse geographic, climatic, and observational characteristics across stations. However, most existing interpolation methods adopt uniform architectures or fixed graph structures, without explicitly adapting to the localized properties of individual stations, which often leads to suboptimal generalization and increased local errors, particularly in regions with complex terrain or sparse observations. Oversmoothing in interpolation is another widely observed issue, where predicted values lack spatial variability and fail to preserve local extremes, limiting their usefulness in tasks such as extreme event detection or fine-grained forecast correction. To better understand the origin of these challenges, we conduct a detailed empirical analysis in Appendix D. To address these issues, we propose **DiffLiG** (*Diffusion-enhanced Liquid Graph with Attention Propagation*), a graph neural network framework specifically designed to model station heterogeneity and alleviate oversmoothing in station-level interpolation.

To overcome these challenges, DiffLiG introduces a unified, uncertainty-aware interpolation framework composed of three key components: a GeoLiquidNet, which combines temporal attention over gridded inputs with a site-aware liquid neural network to capture station-specific temporal dynamics; a Dynamic Edge Modulator, which adaptively configures spatial graphs by jointly learning edge existence, directional attention, and distance-aware influence; and a Probabilistic Diffusion Selector, which generates multiple candidate corrections through noise-perturbed graph propagation and selects or aggregates them via a context-aware scoring mechanism, thereby enhancing both structural variability and predictive reliability [29, 23, 13].

In summary, our contributions are as follows:

- We propose **DiffLiG**, a novel graph neural network framework designed to bridge the gap between gridded forecasts and station-level observations, targeting station heterogeneity and oversmoothing.

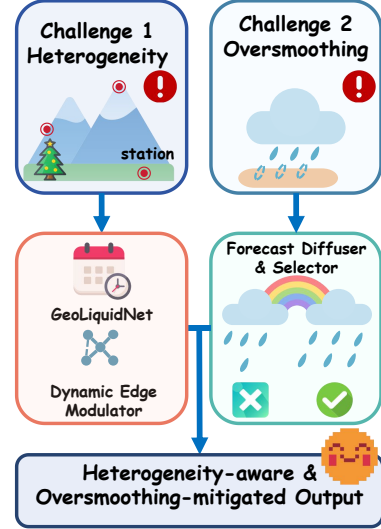


Figure 1: **Two core challenges in station-level interpolation.** **Heterogeneity** across stations is addressed by the GeoLiquidNet, which models site-specific temporal dynamics, and the Dynamic Edge Modulator, which learns spatially adaptive graph structures. **Oversmoothing** is mitigated by the Probabilistic Diffusion Selector, which generates and selects diverse correction candidates to preserve structural variability.

- We design three specialized modules to address station heterogeneity and prediction oversmoothing: a GeoLiquidNet for adaptive temporal encoding based on attention and site-aware OU dynamics; a Dynamic Edge Modulator for configuring spatially adaptive graphs through directional attention and learnable edge selection; and a Probabilistic Diffusion Selector for generating diverse correction candidates and selecting reliable outputs under uncertainty.
- Extensive experiments across multiple forecasting sources, variables, and geographic regions demonstrate that DiffLiG consistently outperforms traditional interpolation and GNN baselines in both accuracy and robustness at the station level.

2 Related Work

Grid-based forecasting. Grid-structured modeling is the backbone of both numerical and AI-driven weather prediction systems. Operational centers such as ECMWF deploy deterministic high-resolution forecasts (IFS-HRES) and ensemble-based systems (IFS-ENS) by solving physical equations over structured grids. Recent AI models have inherited this gridded formulation: FourCastNet [30] employs Fourier neural operators for global fields; GraphCast [22] uses spherical GNNs for medium-range forecasts; Pangu-Weather [5] introduces 3D convolutional architectures for volumetric prediction. Other large-scale efforts include FuXi [10] and Fengwu [9], which are trained on global reanalysis datasets and have demonstrated competitive performance in medium-range forecasting tasks. Foundation models such as AtmoRep [24], Aurora [6], and ClimaX [28] further decouple forecasting from supervision by pretraining over multi-variable grid datasets. However, these models produce outputs aligned to fixed grid structures and often struggle to adapt to sparse, topography-sensitive station layouts. This mismatch limits their direct applicability in real-world settings where localized accuracy is critical. Our work bypasses the grid altogether, targeting station-level prediction with geometry-aware spatial adaptation.

From grids to stations. Traditional spatial interpolation methods, such as inverse distance weighting (IDW), bilinear interpolation, kriging, and spline interpolation, efficiently generate point forecasts from gridded fields, require no training data, and offer clear interpretability with low computational cost. More recently, learning-based approaches have sought to improve accuracy and capture complex dependencies: Bentsen et al. [44] apply a GNN to station time series for wind prediction, MetNet-3 [2] fuses satellite and station inputs via a transformer-style U-Net (albeit on a regridded 4 km mesh), Zhao et al. [4] further highlight the smoothing bias in ERA5 during extreme tropical cyclone rainfall events and propose a deep learning-based correction framework, and MGNN [39] combines ERA5 and MADIS in a multimodal graph for deterministic forecasts. However, all these methods either depend on grid-aligned inputs or outputs, fail to adapt to station-specific heterogeneity, and lack mechanisms for uncertainty quantification or structural diversity. In contrast, our framework operates natively off-grid, leveraging a geometry-aware graph with adaptive edge modulation and geo-liquid temporal dynamics, together with diffusion-based ensemble generation to preserve fine-scale variability and capture predictive uncertainty.

3 Method

3.1 Problem Formulation

We address the problem of spatial interpolation from structured gridded meteorological forecasts to irregularly distributed observation stations. Our goal is to develop a lightweight and scalable graph neural network (GNN) framework that ensures physical consistency, strong spatial generalization, and the ability to model predictive uncertainty.

Formally, the model takes as input a triplet $(\mathcal{X}, \mathcal{P}, \mathcal{H})$:

- $\mathcal{X} = \{\mathbf{X}^{(t)} \in \mathbb{R}^{C \times H \times W} \mid t = -T_1, \dots, 0, \dots, +T_2\}$ denotes the gridded forecast data over $T = T_1 + T_2 + 1$ time steps. Each $\mathbf{X}^{(t)}$ is a multichannel spatial field (e.g., precipitation, wind) with C variables over a regular grid of size $H \times W$ (e.g., 721×1440 for 0.25° global resolution).
- $\mathcal{P} = \{p_j \in \mathbb{R}^2 \mid j = 1, \dots, M\}$ represents the spatial coordinates (latitude and longitude) of M observation stations.

- $\mathcal{H} = \left\{ h_j^{(t)} \in \mathbb{R} \mid j = 1, \dots, M; t = -T_1, \dots, 0 \right\}$ contains the historical observations at each station over $T_1 + 1$ time steps (from $t = -T_1$ to $t = 0$).

The objective is to learn a mapping function:

$$\mathcal{F} : (\mathcal{X}, \mathcal{P}, \mathcal{H}) \longrightarrow \hat{\mathcal{Y}}, \quad \text{where} \quad \hat{\mathcal{Y}} = \{\hat{y}_j \in \mathbb{R} \mid j = 1, \dots, M\}$$

that predicts the target variable at each station for the future time step $t = +T_2$. The model is trained to minimize the prediction error with respect to the ground truth:

$$\hat{y}_j \approx y_j^{(+T_2)}, \quad \forall j = 1, \dots, M.$$

3.2 DiffLiG Overview

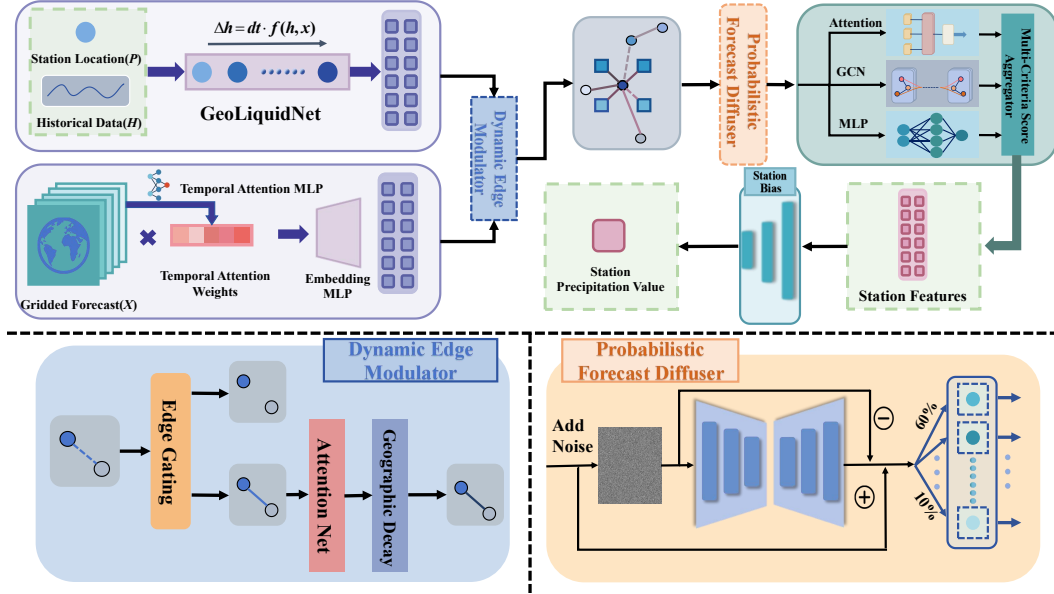


Figure 2: **Overview of DiffLiG.** The model takes gridded meteorological forecasts $\mathcal{X}$, station coordinates $\mathcal{P}$, and historical station observations $\mathcal{H}$ as input. The **GeoLiquidNet** module encodes temporal features for each station based on $\mathcal{H}$ and $\mathcal{P}$. The **Forecast Encoder** assigns temporal attention weights across different time steps of $\mathcal{X}$, followed by an embedding MLP to extract forecast representations. These representations are integrated by the **Dynamic Edge Modulator**, which constructs adaptive graph edges through edge gating, attention net, and geographic decay. The **Probabilistic Forecast Diffuser** generates multiple perturbed predictions to capture epistemic uncertainty. A **Multi-Criteria Forecast Selector** evaluates and selects the most reliable prediction among them. Finally, the selected prediction is refined by the **Station Bias Module**, which incorporates site-specific adjustments through an output MLP, producing the final forecast $\hat{\mathcal{Y}} = \{\hat{y}_j\}_{j=1}^M$, where $\hat{y}_j \approx y_j^{(+T_2)}$.

3.3 GeoLiquidNet

To handle the non-stationary nature of precipitation sequences, where relevant timescales vary across weather systems[26], regions, and station density, we propose **GeoLiquidOU**, a module that learns spatially-conditioned time constants to adaptively control hidden state transitions. Inspired by Liquid Time-constant Networks (LTC) [15], which model time constants as input-dependent variables to capture nonlinear dynamics, GeoLiquidOU inherits this property to adaptively modulate temporal sensitivity based on each site’s historical inputs. Neuroscientific insights further support this design: the diversity of timescales in the brain reflects the dynamic nature of encoded information [41], analogous to varying temporal patterns in meteorological data. GeoLiquidOU thus integrates dynamic temporal encoding with location-aware adaptability, enhancing both interpretability and flexibility in modeling site-specific temporal patterns. Further analytical discussions on the OU-based dynamics are deferred to the Appendix E.

Formally, for a station with input sequence $h = \{h^{(1)}, h^{(2)}, \dots, h^{(T)}\}$ and location $p \in \mathbb{R}^2$, the module first embeds each timestep via $z^{(t)} = \phi_{\text{proj}}(h^{(t)})$, where $\phi_{\text{proj}}(\cdot)$ is a learnable projection MLP that maps the raw observation $h^{(t)}$ into a latent feature space.

A learnable location encoder $\tau(p)$ then generates the time-step controller as:

$$\tau(p) = \text{Sigmoid}(W_2 \cdot \tanh(W_1 \cdot p)) + 0.5,$$

where W_1 and W_2 are linear transformation matrices. The output $\tau(p)$ serves as a positive location-dependent scaling factor that defines a site-specific time interval $\Delta t = \tau(p) \cdot \Delta t_{\text{base}}$.

The latent state $\mathbf{z}_t \in \mathbb{R}^d$ evolves through a parameterized Ornstein–Uhlenbeck (OU) process:

$$\Delta \mathbf{z}_t = \Delta t \cdot \left[\gamma \cdot (\Theta(\mathbf{z}^{(t)}) - \mathbf{z}_t) + \sigma \cdot \varepsilon \right], \quad (1)$$

where $\Theta(\cdot)$ is a learnable MLP that predicts the target equilibrium state, $\gamma, \sigma \in \mathbb{R}^d$ are trainable dynamic coefficients controlling the deterministic and stochastic components, and $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ denotes Gaussian noise.

The process is initialized with $\mathbf{z}_0 = \phi_{\text{init}}(h^{(1)})$, and iteratively updated over T steps. The final state $\mathbf{z}_T$ thus encodes the historical evolution of each site, capturing temporal dynamics modulated by its geographic location. This continuous-time formulation allows GeoLiquidNet to flexibly adapt to diverse temporal rhythms across stations, enhancing generalization and interpretability in sequence modeling.

3.4 Spatial Message Passing Network

To model the spatial heterogeneity and directional propagation of meteorological variables, we introduce a **Spatial Message Passing Network (SMPN)**. It adaptively learns graph structure and edge importance through four components: edge selection, distance-aware decay, directional attention, and residual node update.

Edge Selection Mechanism. We employ a neural discriminator $\psi(\cdot)$ to predict the existence probability of an edge based on node coordinates and their Haversine distance: $g_{ij} = \mathcal{G}(\psi([\mathbf{p}_i, \mathbf{p}_j, d_{ij}]))$, where $\mathcal{G}(\cdot)$ denotes a learnable gating function that outputs an edge existence probability. During training, a binary gate $\tilde{g}_{ij} \in \{0, 1\}$ is sampled from g_{ij} to enable data-driven, spatially adaptive edge pruning [12, 19].

Distance-aware Decay. To reflect spatial attenuation, each edge is weighted by a decay factor $\kappa_{ij} = \exp(-\beta d_{ij}^2)$, where d_{ij} is the Haversine distance. The decay rate β is learned via a bounded sigmoid function, allowing smooth, interpretable modulation of long-range influence.

Multi-head Attention. To capture directional dependencies, we design an interaction-aware attention mechanism [37, 40, 8]. For each edge, the input vector $\mathbf{e}_{ij} = [\mathbf{h}_i \parallel \mathbf{h}_j \parallel (\mathbf{p}_i - \mathbf{p}_j)]$ is processed by M attention heads. Normalized scores $\alpha_{ij}^{(m)}$ determine the contribution of neighbor j to i , enabling asymmetric and context-sensitive aggregation [42].

Residual Update. The final propagation weight is $w_{ij} = \tilde{g}_{ij} \cdot \kappa_{ij} \cdot \alpha_{ij}$. Messages m_{ij} are generated via ϕ_{msg} and aggregated with normalization. Each node is updated via a residual MLP [7, 16]:

$$\mathbf{h}'_i = \mathbf{h}_i + \phi_{\text{upd}}([\mathbf{h}_i, \bar{\mathbf{m}}_i])$$

This enables stable, fine-grained updates across two stages: first from gridded fields to stations, then among stations for local refinement.

Algorithm 1: Spatial Message Passing Network

Input: Node features $\mathbf{h}_i$, positions $\mathbf{p}_i$, number of spatial neighbors k , edge gates $\tilde{g}_{ij}$, decay factors κ_{ij} , attentions α_{ij}

1. For each node i , identify the k nearest neighbors $\mathcal{N}_k(i)$ based on Haversine distance
2. Initialize $\bar{\mathbf{m}}_i \leftarrow \mathbf{0}$, count $\leftarrow 0$
3. For each $j \in \mathcal{N}_k(i)$:
 - If $\tilde{g}_{ij} = 1$, compute propagation weight $w_{ij} = \tilde{g}_{ij} \cdot \kappa_{ij} \cdot \alpha_{ij}$
 - Compute message $m_{ij} = \phi_{\text{msg}}([\mathbf{h}_i, \mathbf{h}_j, \mathbf{p}_i - \mathbf{p}_j])$
 - Accumulate: $\bar{\mathbf{m}}_i += w_{ij} \cdot m_{ij}$; increment count
4. Normalize: $\bar{\mathbf{m}}_i \leftarrow \bar{\mathbf{m}}_i / \text{count}$
5. Update node state: $\mathbf{h}'_i = \mathbf{h}_i + \phi_{\text{upd}}([\mathbf{h}_i, \bar{\mathbf{m}}_i])$

3.5 Probabilistic Forecast Diffuser

To capture the inherent uncertainty of meteorological predictions and generate diverse forecast candidates, we propose a **Probabilistic Forecast Diffuser**. This module injects stochastic perturbations into node features and processes them through a graph-based denoising network, yielding a set of sample-level forecasts for subsequent evaluation and selection. A concise mathematical analysis explaining why the diffusion process enables recovery and amplification of extreme precipitation values is provided in Appendix F.

Sample Generation via Diffusion. Given node features x and position encodings $\phi(\mathbf{p})$, we generate N samples by adding Gaussian noise $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 I)$ and passing the perturbed inputs through a graph U-Net denoiser f_{UNet} [18, 35, 1]. Each sample is constructed via residual correction:

$$s_i = \lambda \cdot x + (1 - \lambda) \cdot (x + \varepsilon_i - f_{\text{UNet}}(x + \varepsilon_i + \phi(\mathbf{p}))) \quad (2)$$

Here, $\lambda \in [0, 1]$ is a blending coefficient that controls the contribution of original versus corrected features. The result is an ensemble of spatially coherent yet diverse forecasts $\{s_1, \dots, s_N\}$.

Uncertainty Estimation. We quantify prediction uncertainty for each node by computing the sample variance:

$$u_j = \frac{1}{N} \sum_{i=1}^N (s_{i,j} - \bar{s}_j)^2, \quad \bar{s}_j = \frac{1}{N} \sum_{i=1}^N s_{i,j} \quad (3)$$

This serves as a natural metric for identifying regions of disagreement across the ensemble and guiding adaptive selection.

3.6 Multi-Criteria Score Selector

To extract the most plausible forecast from the generated ensemble, or combine multiple outputs into a robust final prediction, we introduce the **Multi-Criteria Score Selector**. This module evaluates each candidate using three scoring perspectives: semantic consensus, spatial coherence, and physical plausibility [21, 25].

1. Semantic Consensus. A self-attention mechanism is applied across all N samples to estimate their mutual agreement. Each sample is assigned an attention-based consensus score, measuring its alignment with the ensemble majority.

2. Spatial Coherence. To enforce geophysical smoothness, we pass each sample through a lightweight graph convolution network that estimates the degree of spatial consistency across neighboring stations. This captures the natural continuity of meteorological variables in space.

3. Extreme Value Detection. We implement a lightweight MLP-based scoring head that focuses on detecting physically implausible or statistically extreme predictions. It penalizes samples that contain abnormal spikes, unrealistic magnitudes, or values significantly deviating from historical ranges, thereby reducing their influence during ensemble selection.

Final Selection. The above three scores are aggregated using a two-layer MLP to compute a unified quality score q_i for each sample. We apply a softmax over all scores to produce sample weights. The final output is selected via either hard picking (the best-scoring q_i) or weighted fusion of the ensemble.

4 Experimental

4.1 Experimental Setup

To comprehensively evaluate the generalization and robustness of the proposed method under various meteorological models and complex geographical conditions, we design a station-level precipitation interpolation task based on multiple forecast products and ground truth observations.

Data Sources. We utilize 24-hour accumulated precipitation forecasts from five representative meteorological models, including both traditional numerical systems and emerging AI-based models. Specifically, the selected inputs cover: ECMWF’s high-resolution deterministic forecast (IFS-HRES), ensemble forecast system (IFS-ENS), and reanalysis data (ERA5), as well as models such as FourCast-Net, and GraphCast. As ground-truth reference, we employ in-situ measurements from 2288 surface meteorological stations provided by the China Meteorological Administration. These stations are distributed across diverse terrains and climatic regions, introducing strong spatial heterogeneity. All observational data are resampled to daily resolution and temporally aligned with forecast inputs. The resulting dataset forms a 24-hour accumulated precipitation interpolation benchmark for evaluating spatial reasoning and correction capabilities under complex landscapes.

Task Definition. The task is formulated as a grid-to-station spatial mapping problem. Each sample consists of coarse-resolution gridded inputs and historical observations at stations, and the model is required to predict the target variable (e.g., 24-hour precipitation) at all station locations.

Extended Experiments. To examine the model’s variable adaptability and temporal generalization, we also conduct extended experiments on 2-meter temperature prediction using Pangu-Weather temperature products. Detailed results of this auxiliary experiment are included in the Appendix G. We report additional results in the Appendix H, including a sensitivity analysis under varying forecast horizons and structural parameter settings.

Category	Methods
Traditional Interpolation	Nearest Neighbor
	Linear Interpolation
	Inverse Distance Weighting
	Ordinary Kriging
Graph Neural Networks	GCN [19]
	GAT [38]
	GraphSAGE [14]
	KCN [3]
Physics-Inspired Architectures	MeshGraphNet [31]
	MGNN [39]
Spatial & Generative	ViT [11]
	Diffusion Networks [18]

Table 1: Classification of Baseline Models

Model	P (#)	S (MB)
DiffLiG (ours)	49,361	0.19
LiGAP	39,397	0.15
GAP	33,650	0.13
Graph-bias	29,506	0.11
Graph	24,930	0.10
GCN	29,031	0.11
GAT	29,287	0.11
GraphSAGE	33,127	0.13
KCN	30,520	0.12
ViT	51,538	0.20
Diffusion	54,014	0.21
MeshGraphNet	41,863	0.16
MGNN	40,741	0.16

Table 2: Model Complexity Comparison

Baseline Models. To assess model performance, we compare against a broad range of interpolation and learning-based baselines, as shown in Table 1. This diverse benchmark suite allows us to evaluate accuracy, structural consistency, and generalization ability across multiple paradigms. A more detailed description of the baseline methods, including their modeling assumptions and limitations, is provided in the Appendix K.

Model Complexity. To further examine architectural efficiency, we report the parameter counts and storage sizes of all compared models in Table 2. Although DiffLiG is not the smallest in scale, it introduces only a moderate increase in parameters while achieving significant gains in accuracy and generalization, reflecting an efficient use of model capacity. Its ablation variants reveal the role of individual modules: LiGAP removes the diffusion-based ensemble component from DiffLiG, GAP further discards the Temporal Module to isolate temporal contributions, Graph-bias omits the Spatial

Message Passing Network, and Graph represents the most reduced configuration without the Station Bias Module.

Evaluation Metrics. We employ five quantitative metrics to comprehensively assess model performance: Root Mean Squared Error (RMSE), Anomaly Correlation Coefficient (ACC), and Threat Scores (TS) at 2 mm, 20 mm, and 100 mm thresholds. RMSE measures the overall magnitude deviation between predictions and observations, ACC evaluates spatial correlation and pattern consistency across stations, while TS metrics quantify the model’s capability to detect precipitation events of increasing intensity, reflecting its sensitivity to light, moderate, and extreme rainfall. Together, these metrics provide a balanced evaluation of numerical accuracy, spatial coherence, and event-level detection skill.

Implementation and Training Details. To ensure fair comparison, all baseline models are trained under identical hardware environments, using the same loss function and optimization settings. Further experimental details, including training strategies, configurations, and runtime statistics, are provided in Appendix I.

4.2 Experimental Results

Method	IFS-HRES			ERA5			IFS-ENS			FourCastNet		
	RMSE	ACC	TS@2mm	RMSE	ACC	TS@2mm	RMSE	ACC	TS@2mm	RMSE	ACC	TS@2mm
DiffLiG (ours)	5.0883	0.7395	0.4621	4.8021	0.7677	0.5121	4.7208	0.7721	0.5286	5.1014	0.7303	0.4937
Nearest	6.0998	0.6511	0.4121	5.2145	0.7265	0.4693	4.9717	0.7468	0.4801	5.3871	0.6795	0.4865
Linear	5.9033	0.6667	0.4011	5.1435	0.7313	0.4808	4.9395	0.7504	0.4757	5.3721	0.6948	0.4858
OK	5.7423	0.6577	0.3931	5.1011	0.7387	0.4891	4.8932	0.7581	0.4434	5.3088	0.7052	0.4811
IDW	5.8274	0.6742	0.3995	5.1058	0.7368	0.4558	4.9299	0.7515	0.4716	5.3683	0.6831	0.4837
GCN	5.4832	0.6927	0.4139	5.0680	0.7417	0.4385	4.8731	0.7610	0.5052	5.2313	0.7016	0.4970
GAT	5.5444	0.6667	0.3202	5.0401	0.7449	0.4388	4.9137	0.7572	0.4987	5.2551	0.6964	0.4921
GraphSAGE	5.6401	0.6495	0.3754	5.1154	0.7243	0.4634	4.9343	0.7423	0.4922	5.2511	0.7001	0.4936
KCN	5.3931	0.6962	0.4158	5.0837	0.7468	0.4498	4.8897	0.7572	0.4925	5.2264	0.7122	0.4923
ViT	5.7848	0.6338	0.3626	5.1232	0.7121	0.4428	5.2365	0.7034	0.4832	5.3912	0.7080	0.4812
Diffusion	5.8480	0.5963	0.3342	5.1000	0.7284	0.4537	5.1742	0.7183	0.4952	5.3781	0.6932	0.4921
MeshGraphNet	5.5322	0.6767	0.4374	5.1728	0.7283	0.4903	5.0132	0.7332	0.5143	5.2619	0.6931	0.4987
MGN	5.4428	0.6858	0.3640	4.9904	0.7501	0.5000	4.8542	0.7657	0.4357	5.1878	0.7164	0.4423

Table 3: **Performance Comparison of Interpolation Methods across Multi-Source Forecast Inputs.** Top-3 values in each metric are highlighted with increasing color intensity. Bold values indicate the best performance, while underlined values represent the second-best results for each metric.

To evaluate the model’s capability in extracting reliable station-level information from coarse-resolution gridded forecasts, we conduct comprehensive experiments across four representative forecast sources: IFS-HRES, ERA5, IFS-ENS, and FourCastNet. Results are summarized in Table 3.

DiffLiG demonstrates consistent superiority across all three metrics. In terms of TS@2mm, which measures the model’s ability to identify precipitation events, DiffLiG achieves the highest scores on IFS-HRES (0.4621), ERA5 (0.5121), and IFS-ENS (0.5286). On FourCastNet, it reaches 0.4937, closely following the best-performing baseline (0.4987), underscoring its strong event detection capability. For RMSE, DiffLiG achieves the lowest error across all four sources, indicating more accurate magnitude reconstruction. In terms of ACC, DiffLiG again attains the highest accuracy on each dataset, reflecting its consistent predictive correctness and classification reliability.

Furthermore, the consistent improvements of DiffLiG over all baselines across multiple evaluation metrics suggest its strong capacity to mitigate station-level **heterogeneity**. Unlike traditional methods that often suffer from degraded performance in regions with complex topography or sparse observations, DiffLiG achieves lower RMSE (more accurate magnitude), higher ACC (stronger spatial correlation with observations), and higher TS@2mm (better event detection skill). These results indicate that the model effectively adapts to diverse spatial characteristics and observational conditions, demonstrating its practical utility for heterogeneous station-level interpolation in real-world settings.

In addition, DiffLiG exhibits strong cross-source generalization. It maintains either top or second-best performance across all forecast sources, regardless of the underlying modeling paradigm: be it physically based numerical weather prediction (e.g., IFS-HRES, IFS-ENS) or AI-driven models (e.g., FourCastNet). This cross-domain consistency validates the robustness of its architectural design under varying input distributions and error characteristics.

4.3 Evaluation of Oversmoothing Mitigation on IFS-HRES

To evaluate the capability of different models in mitigating spatial oversmoothing, we compute TS at 20mm and 100mm thresholds on the IFS-HRES dataset. As shown in Table 4, DiffLiG achieves the highest TS values across both thresholds, demonstrating its effectiveness in alleviating the oversmoothing problem. LiGAP, which removes the diffusion module from DiffLiG while keeping all other components identical, shows a noticeable degradation in both TS@20mm and TS@100mm, highlighting the effectiveness of the probabilistic diffusion mechanism in mitigating the oversmoothing effect inherent in graph propagation.

Model	TS@20mm	TS@100mm
DiffLiG (ours)	0.3122	0.0202
LiGAP	0.2819	0.0147
Linear	0.2764	0.0156
GAT	0.2986	0.0162
GCN	0.2836	0.0131
GraphSAGE	0.2648	0.0157
KCN	0.2862	0.0136
MeshGraphNet	0.2925	0.0144
MGNN	0.2611	0.0153

Table 4: Oversmoothing Mitigation on IFS-HRES.

4.4 Visual Evidence of Oversmoothing Mitigation

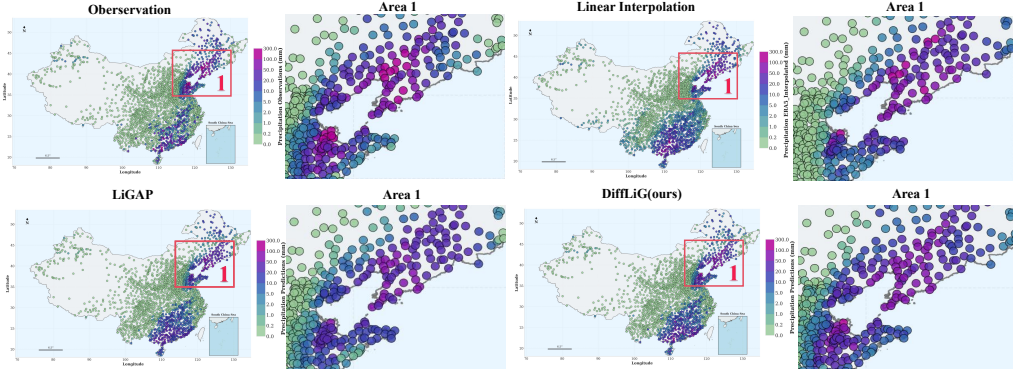


Figure 3: Visual comparison of spatial interpolation results. Predicted daily accumulated precipitation for July 5, 2022.

To assess oversmoothing mitigation, we visualize predicted station-level precipitation on July 5, 2022—a day with high spatial variability (Figure 3). Compared to baseline methods, DiffLiG produces sharper and more spatially coherent outputs, capturing localized extremes more effectively. LiGAP, which removes the diffusion module from DiffLiG, shows improved spatial structure but remains conservative due to its deterministic nature. The full DiffLiG model leverages ensemble diversity and selective refinement to better recover fine-scale variability aligned with observations. A more detailed visualization of the results is provided in the Appendix N for further reference.

4.5 Cross-Dataset Evaluation on GraphCast for Robustness Validation

GraphCast			
Method	RMSE ↓	ACC ↑	TS@2mm ↑
DiffLiG (ours)	5.5172	0.7345	0.5218
Nearest	5.7036	0.7133	0.3520
Linear	5.6829	0.7153	0.3515
OK	5.5923	0.7132	0.3612
IDW	5.6772	0.7158	0.3506
GCN	5.5517	0.7302	0.4985
GAT	5.5773	0.7271	0.5245
GraphSAGE	5.8757	0.6811	0.5001
KCN	5.5980	0.7256	0.5181
ViT	5.8321	0.6844	0.4212
Diffusion	5.7342	0.6923	0.4354
MeshGraphNet	6.2677	0.6246	0.5171
MGNN	5.6537	0.7288	0.3526

Table 5: Performance of different methods on GraphCast.

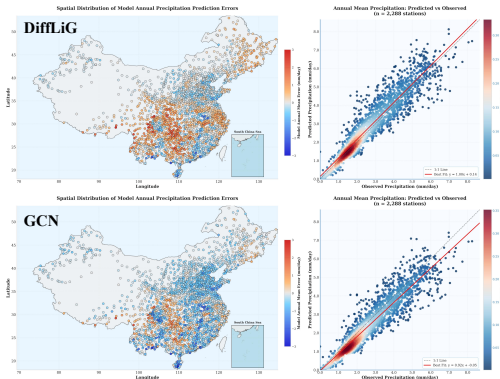


Figure 4: Spatial distribution (left) and scatter-plot (right) for DiffLiG vs. GCN.

To evaluate the robustness and generalization capability of our heterogeneity-aware forecasting model, we conduct a cross-dataset inference experiment. Specifically, the model is trained entirely on the **IFS-HRES** dataset and directly applied to the **GraphCast** dataset without any fine-tuning or domain adaptation. The performance comparison is reported in Table 5, and the spatial error distribution alongside station-wise scatter-plot of DiffLiG versus GCN is visualized in Figure 4.

4.6 Ablation Studies

Structural modeling. Starting from the base model with static edges and uniform aggregation, performance is limited across all metrics. Adding the *Dynamic Edge Modulator* leads to a significant improvement in TS@2mm, highlighting the benefit of learning spatially adaptive connections. The *Station Bias Correction* further improves ACC by capturing site-specific deviations, especially where grid forecasts systematically diverge from observations. These modules enhance the model’s ability to reflect spatial heterogeneity, but temporal dynamics remain unmodeled.

Temporal encoding. Introducing the *Temporal Module* leads to a substantial improvement in TS@2mm (+16.3%), indicating enhanced sensitivity to localized rainfall events. By jointly encoding gridded sequences and station histories, the model better captures evolving precipitation patterns, leading to concurrent improvements in both ACC and RMSE.

Uncertainty modeling and selection. The *Probabilistic Forecast Diffuser* reduces RMSE by modeling ensemble uncertainty, though basic aggregation methods show limited effect on TS and ACC. The final addition of the *Multi-Criteria Score Selector* brings all metrics to their best levels by prioritizing forecasts with spatial consistency and physical plausibility. This confirms the importance of informed selection in ensemble-based interpolation.

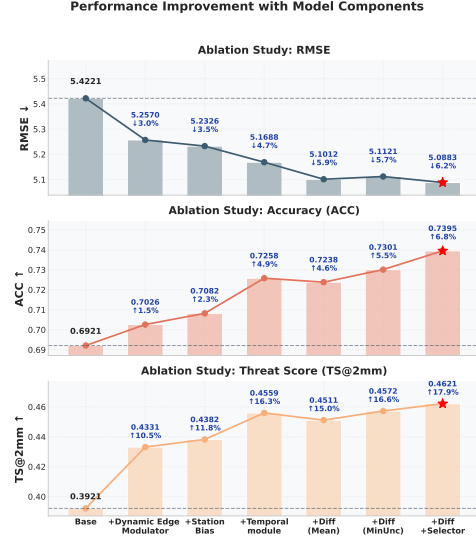


Figure 5: **Ablation trend.** Module-wise improvements across RMSE, ACC, and TS@2mm.

5 Conclusion

We presented DiffLiG, a diffusion-enhanced liquid graph framework for high-resolution spatial correction from gridded forecasts to station-level observations. To address the challenge of station heterogeneity, DiffLiG integrates a Geo-Liquid Net that captures site-specific temporal dynamics, and a Dynamic Edge Modulator that independently learns spatially adaptive graph structures tailored to each target station. To mitigate oversmoothing and model predictive uncertainty, a unified Probabilistic Diffusion Selector generates diverse correction candidates via noise-perturbed sampling and refines them through multi-criteria ensemble selection.

Extensive experiments across multiple meteorological datasets and forecast products demonstrate that DiffLiG delivers consistent improvements in both accuracy and event sensitivity over classical interpolation methods and strong graph-based baselines. Its effectiveness extends beyond precipitation, achieving strong performance on other key variables such as temperature, highlighting its flexibility and generalizability.

DiffLiG offers a unified and efficient framework for correcting coarse meteorological forecasts under real-world spatial complexity, with strong potential for application in post-processing, data assimilation, and multi-source fusion tasks. Looking forward, DiffLiG provides a foundation for exploring graph-based spatial correction under more complex scenarios, such as cross-regional generalization, extreme weather adaptation, and integration with multi-modal inputs (e.g., satellite imagery or radar). Future work may also incorporate causal or physical priors to further enhance the interpretability and resilience of correction under evolving climate dynamics.

Acknowledgements

This work was supported in part by the National Key Research and Development Plan of China under Grant 2023YFB3002400; in part by the National Natural Science Foundation of China under Grant T2125006 and Grant 42401415; in part by Shenzhen Science and Technology Program under Grant KCXFZ20240903093759004 and Grant KJZD20230923115106012; in part by the Open Research Fund of Pengcheng Laboratory under No.2025KF1B0040; and in part by Jiangsu Innovation Capacity Building Program under Project BM2022028.

References

- [1] Martin Andrae, Tomas Landelius, Joel Oskarsson, and Fredrik Lindsten. Continuous ensemble weather forecasting with diffusion models, 2025.
- [2] Marcin Andrychowicz, Lasse Espeholt, Di Li, Samier Merchant, Alexander Merose, Fred Zyda, Shreya Agrawal, and Nal Kalchbrenner. Deep learning for day forecasts from sparse observations, 2023.
- [3] Gabriel Appleby, Linfeng Liu, and Li-Ping Liu. Kriging convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3187–3194, April 2020.
- [4] Guido Ascenso, Andrea Ficchi, Matteo Giuliani, Enrico Scoccimarro, and Andrea Castelletti. Downscaling, bias correction, and spatial adjustment of extreme tropical cyclone rainfall in era5 using deep learning. *Weather and Climate Extremes*, 46:100724, 2024.
- [5] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast, 2022.
- [6] Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Vaughan, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, Jayesh K. Gupta, Kit Thambiratnam, Alexander T. Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. A foundation model for the earth system, 2024.
- [7] Xavier Bresson and Thomas Laurent. Residual gated graph convnets, 2018.
- [8] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks?, 2022.
- [9] Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, Yuanzheng Ci, Bin Li, Xiaokang Yang, and Wanli Ouyang. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead, 2023.
- [10] Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. Fuxi: A cascade machine learning forecasting system for 15-day global weather forecast, 2023.
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.
- [12] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks, 2020.
- [13] Peter Grönquist, Chengyuan Yao, Tal Ben-Nun, Nikoli Dryden, Peter Dueben, Shigang Li, and Torsten Hoefler. Deep learning for post-processing ensemble weather forecasts. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194):20200092, February 2021.
- [14] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.
- [15] Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks, 2020.

- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [17] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.
- [19] Bo Jiang, Ziyang Zhang, Doudou Lin, and Jin Tang. Graph learning-convolutional networks, 2018.
- [20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [21] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.
- [22] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Meroze, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Graphcast: Learning skillful medium-range global weather forecasting, 2023.
- [23] Erik Larsson, Joel Oskarsson, Tomas Landelius, and Fredrik Lindsten. Diffusion-lam: Probabilistic limited area weather forecasting with diffusion. In *ICLR 2025 Workshop on Tackling Climate Change with Machine Learning*, 2025.
- [24] Christian Lessig, Ilaria Luise, Bing Gong, Michael Langguth, Scarlet Stadtler, and Martin Schultz. Atmorep: A stochastic model of atmosphere dynamics using large scale representation learning, 2023.
- [25] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2018.
- [26] Siqi Liu and Andreas Lehrmann. Dynaconf: Dynamic forecasting of non-stationary time series, 2024.
- [27] S. Karthik Mulkavilli, Daniel Salles Civitarese, Johannes Schmude, Johannes Jakubik, Anne Jones, Nam Nguyen, Christopher Phillips, Sujit Roy, Shraddha Singh, Campbell Watson, Raghu Ganti, Hendrik Hamann, Udaysankar Nair, Rahul Ramachandran, and Kommy Weldemariam. Ai foundation models for weather and climate: Applications, design, and implementation, 2023.
- [28] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. Climax: A foundation model for weather and climate, 2023.
- [29] Joel Oskarsson, Tomas Landelius, Marc Peter Deisenroth, and Fredrik Lindsten. Probabilistic weather forecasting with hierarchical graph neural networks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 41577–41648. Curran Associates, Inc., 2024.
- [30] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators, 2022.
- [31] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks, 2021.

- [32] Thomas Rackow, Nikolay Koldunov, Christian Lessig, Irina Sandu, Mihai Alexe, Matthew Chantry, Mariana Clare, Jesper Dramsch, Florian Pappenberger, Xabier Pedruzo-Bagazgoitia, Steffen Tietsche, and Thomas Jung. Robustness of ai-based weather forecasts in a changing climate, 2024.
- [33] Vivek Ramavajjala and Peetak P. Mitra. Verification against in-situ observations for data-driven weather prediction, 2023.
- [34] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, and Nuno Carvalhais. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.
- [35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.
- [36] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [38] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.
- [39] Qidong Yang, Jonathan Giezendanner, Daniel Salles Civitarese, Johannes Jakubik, Eric Schmitt, Anirban Chandra, Jeremy Vila, Detlef Hohl, Chris Hill, Campbell Watson, and Sherrie Wang. Local off-grid weather forecasting with multi-modal earth observation data, 2025.
- [40] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. Graph transformer networks, 2020.
- [41] Roxana Zeraati, Anna Levina, Jakob H. Macke, and Richard Gao. Neural timescales from a computational perspective, 2024.
- [42] Kai Zhang, Yaokang Zhu, Jun Wang, and Jie Zhang. Adaptive structural fingerprints for graph attention networks. In *International Conference on Learning Representations*, 2020.
- [43] Jiachen Zhu, Xinlei Chen, Kaiming He, Yann LeCun, and Zhuang Liu. Transformers without normalization, 2025.
- [44] Lars Ødegaard Bentsen, Narada Dilp Warakagoda, Roy Stenbro, and Paal Engelstad. Spatio-temporal wind speed forecasting using graph networks and novel transformer architectures. *Applied Energy*, 333:120565, 2023.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We state our main claims in the abstract, elaborate on the research scope and contributions in the introduction 1, summarize the paper's contributions at the end of the introduction 1, and validate them through comprehensive experiments 4.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We include a dedicated discussion of the paper's limitations in the Appendix B.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We include key assumptions and theoretical proofs either in the main text or in the appendix, with clear pointers provided in the main paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Our method includes a detailed description of the model architecture in the Section 3, specifies the data sources used in the Section 4, and provides implementation details and hyperparameters in the Appendix I.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the data sources used in the Section 4.1, and we plan to release the code upon paper acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide detailed experimental settings and configurations in the Appendix I.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: While we do not report formal statistical error bars due to the significant time and computational resources required, we conducted multiple independent runs and observed that the results consistently converge with minimal variance, indicating stable and reliable performance.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report the compute resources required for training and evaluation in the Appendix I.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in this paper fully complies with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss both the potential positive and negative societal impacts of our work in the Appendix A.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not involve any models or data with a high risk of misuse or safety concerns.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The relevant license and attribution information is provided in the Appendix M.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: We plan to release our newly developed model along with documentation upon acceptance. The asset will include descriptions of model structure, training procedure, and license, and will be anonymized at submission time.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[NA\]](#)

Justification: This paper does not involve human subjects and does not require IRB approval.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Large language models were not used as part of the core methodology in this research. Minor usage was limited to writing assistance only.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendices Contents

A	Broader Impact	22
B	Limitations	22
C	Empirical Analysis of ERA5 Precipitation Interpolation Errors	23
D	Origins of Station Heterogeneity and Oversmoothing Biases	25
E	Theoretical Analysis of OU-Based Temporal Encoding	25
E.1	Continuous-Time OU Dynamics	25
E.2	Discrete-Time Approximation via Euler Method	26
E.3	Explicit Recursion and Temporal Weighting	26
E.4	Frequency-Domain Interpretation: A Low-Pass Filter	26
E.5	Interpretation for GeoLiquidOU	27
F	Latent-Space Noise Injection and UNet Denoising with Residual Perturbation	27
G	Cross-Variable Forecasting: Temperature as Input	28
H	Sensitivity to Forecast Horizon and Graph Connectivity	30
H.1	Sensitivity to Forecast Lead Time	30
H.2	Impact of Backtracking Window Size	31
H.3	Impact of ERA5-to-Station Edge Connectivity	31
H.4	Impact of Station-to-Station Connectivity	33
I	Implementation and Training Details	33
I.1	Training Objective and Loss Function	33
I.2	Task Configuration	33
I.3	Training Configuration	33
I.4	Staged Training Strategy	33
J	Haversine Distance Computation	33
K	Baseline Method Descriptions	34
L	Evaluation Metrics	36
M	License and Attribution Details	36
N	Visual comparison of spatial interpolation results	37

A Broader Impact

This study proposes a lightweight and modular graph-based framework for station-level precipitation correction, designed to enhance the adaptability and spatial resolution of existing weather forecasting systems. Without requiring the retraining of large-scale models, our approach enables flexible post-processing of numerical or AI-generated forecasts, producing locally refined predictions at arbitrary coordinates using only location metadata and limited historical observations. This capability supports fine-grained, adaptive forecasting across diverse geographies, including remote, mountainous, or under-instrumented regions, and has the potential to broaden the accessibility of reliable weather services worldwide.

Our method improves the spatial precision and reliability of extreme precipitation forecasts by explicitly modeling site-specific spatial structure, incorporating dynamic edge modulation, and introducing a diffusion-based ensemble generator with a multi-criteria output selector. This allows more accurate localization of high-risk areas and robust identification of extreme-value signals, thus offering decision-level support for early warning systems related to flooding, torrential rainfall, or hydrological disasters. Unlike coarse-resolution grid-based outputs, the proposed framework allows targeted forecast refinement at critical sites, contributing to improved disaster preparedness and risk mitigation.

In terms of sustainability, the proposed architecture emphasizes post-hoc spatial adaptation rather than end-to-end retraining, reducing the demand for high-performance computing and enabling deployment in low-resource environments. As climate modeling and forecasting scale to global applications, energy-efficient and scalable forecasting solutions are urgently needed. Our model contributes to this trend by providing an extensible correction module that balances accuracy with computational economy.

Beyond meteorology, the modular design and uncertainty-aware structure of our model may benefit other spatiotemporal inference tasks with sparse measurements, such as air quality estimation, ecological monitoring, or environmental hazard detection. The integration of spatial message passing, site heterogeneity handling, and ensemble-based confidence evaluation offers a transferable paradigm for building robust, site-adaptive models across scientific domains.

While we have not identified direct negative societal impacts associated with the proposed method, potential concerns may arise depending on downstream applications. For instance, if the model is deployed without adequate validation in regions with limited ground truth data, it may introduce misleading corrections or overconfident forecasts, especially in high-stakes scenarios. Moreover, localized correction systems, if misused or misinterpreted by non-experts, could inadvertently lead to false alarms or reduced trust in public forecasting services. These risks highlight the importance of responsible deployment and continual calibration against local observations when adapting the model to new domains.

B Limitations

While the proposed station-level correction framework demonstrates strong adaptability and predictive performance, several limitations remain that warrant further attention and refinement.

Our current implementation primarily utilizes precipitation and temperature variables from ERA5 as model inputs. Other meteorological factors that are physically relevant, such as wind speed, humidity, surface pressure, or convective potential, have not yet been incorporated. The exclusion of these features may limit the model’s ability to capture complex atmospheric processes, particularly in scenarios driven by multi-variable interactions. Integrating additional variables, such as wind field structures or vertical profiles, into both the graph construction and node representation may enhance the model’s responsiveness to high-impact weather systems.

Furthermore, although the model is generalizable to arbitrary geographic locations, our experiments have so far focused mainly on precipitation and temperature. Key atmospheric variables such as wind vectors or specific humidity have not yet been tested, and the current results do not fully assess the framework’s applicability in broader multi-modal forecasting scenarios. Different atmospheric variables exhibit distinct spatial dependencies and temporal dynamics, and whether the current modeling paradigm generalizes across such diversity remains an open question.

The model also assumes the availability of some historical station-level observations. Although we avoid retraining large-scale backbones, the quality and availability of local data can still affect the performance of post-hoc corrections. In regions with highly sparse or irregular observations, such as new or mobile stations, the reliability of predictions may degrade. Enhancing model robustness through techniques such as transfer learning or structural priors could be a promising future direction.

Lastly, while the diffusion-based ensemble generation and selection mechanisms provide a degree of uncertainty modeling, the current framework does not explicitly incorporate physical consistency constraints. Soft constraints based on hydrological or thermodynamic conservation laws (e.g., moisture or energy balance) could improve both the interpretability and reliability of outputs, especially under extreme or anomalous conditions.

C Empirical Analysis of ERA5 Precipitation Interpolation Errors

To quantitatively motivate the need for high-resolution station-level correction, we conduct a systematic assessment of precipitation biases present in the ERA5 reanalysis dataset after standard bilinear interpolation onto surface meteorological stations.

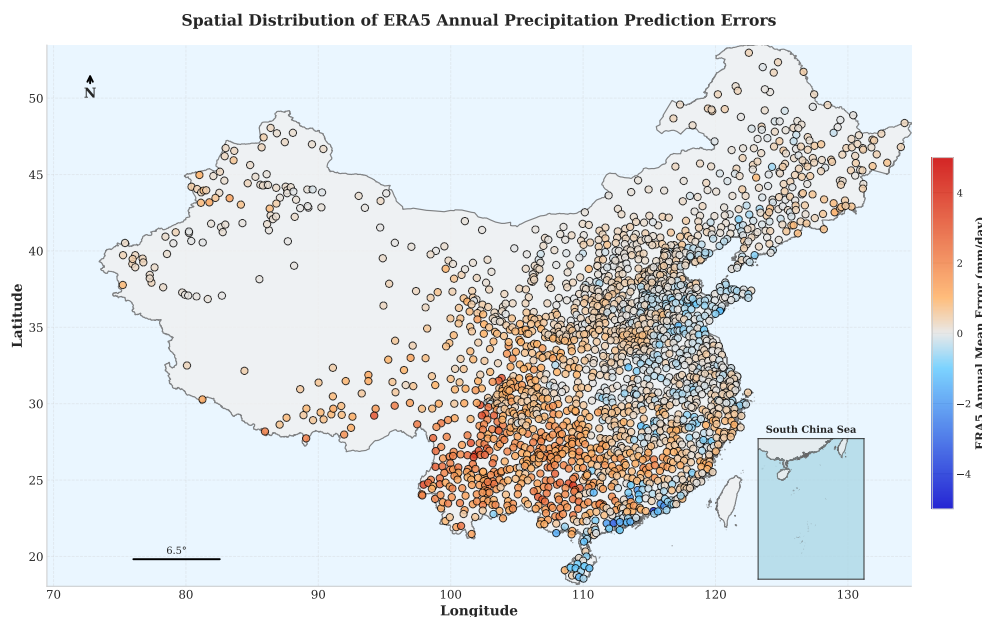


Figure 6: **Spatial Distribution of ERA5 Annual Precipitation Prediction Errors.** Each dot represents a station. Color denotes the signed mean error (in mm/day) of bilinearly interpolated ERA5 daily precipitation compared with ground observations, averaged over the entire year.

As shown in Figure 6, large spatial biases emerge across the region, with systematic overestimations in southwestern and southern China, and underestimations in southeastern coastal regions. These regional inconsistencies reflect the inability of grid-based interpolation to adapt to complex terrains and localized rainfall dynamics.

Figure 7 further illustrates the severity of these discrepancies in relative terms. In regions such as western Sichuan and Yunnan, the relative errors exceed 75%, suggesting that the interpolated ERA5 data may drastically misrepresent local annual precipitation totals. This disproportionate bias is particularly problematic in hydrologically sensitive regions or areas with sparse gauge density.

Figure 8 illustrates the temporal variation of interpolation errors in ERA5 precipitation data across the year. The top panel shows that both RMSE and mean error rise significantly during the summer months, indicating reduced accuracy during the rainy season. The middle panel reveals that the maximum and absolute maximum daily errors also spike during this period, with values occasionally exceeding 200 mm/day. The bottom panel shows that although many extreme rainfall events are

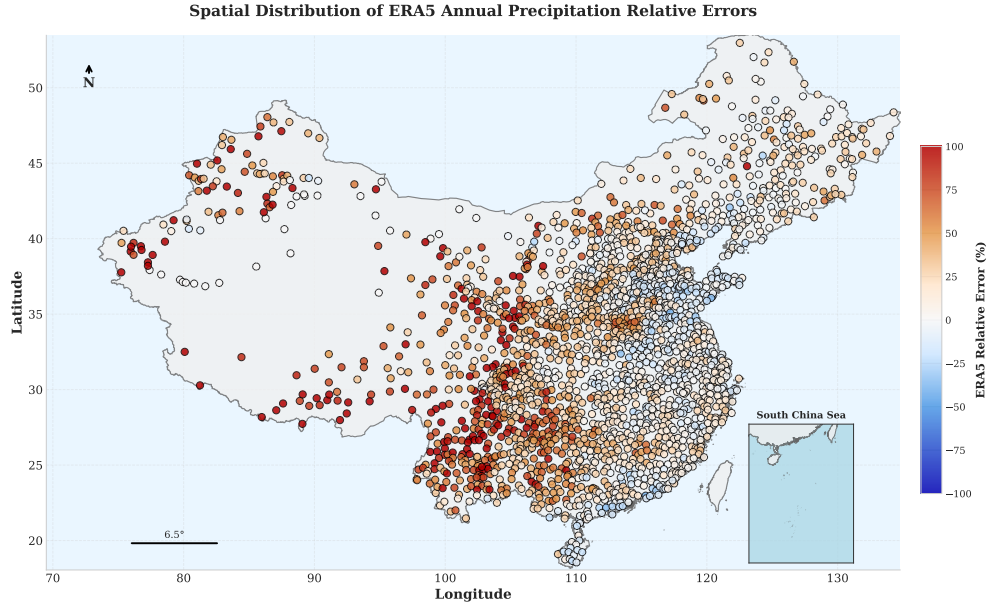


Figure 7: **Spatial Distribution of ERA5 Annual Precipitation Relative Errors.** The relative error is computed as the ratio between absolute bias and station-observed annual precipitation. High values indicate severe misrepresentation of total rainfall.

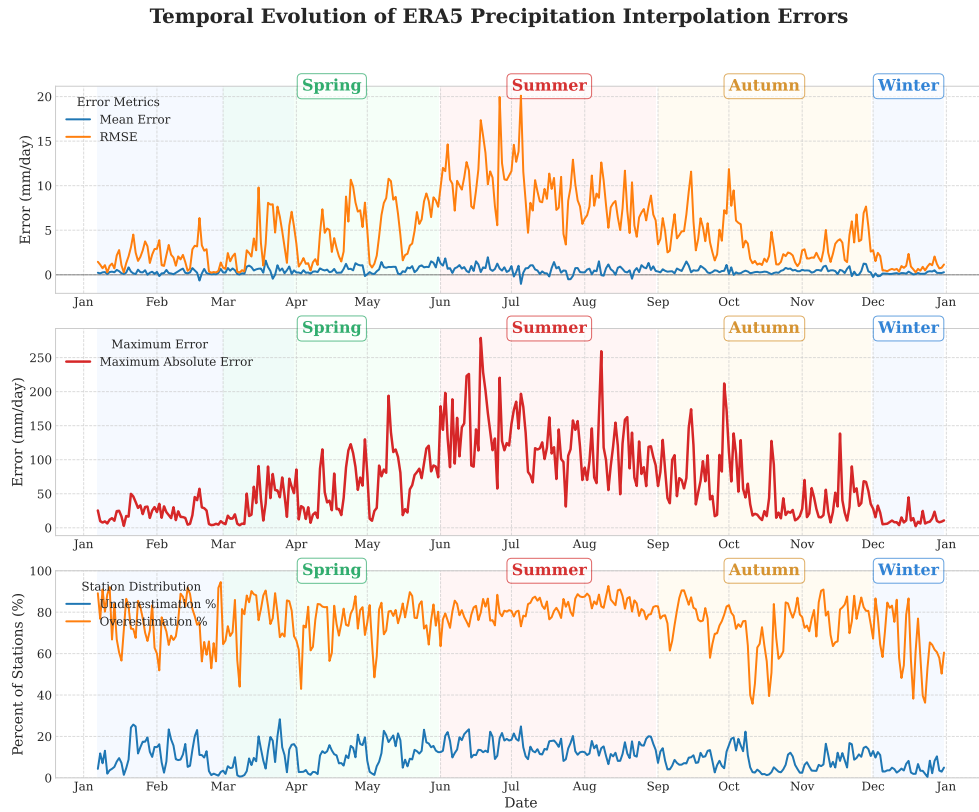


Figure 8: **Temporal Evolution of ERA5 Precipitation Interpolation Errors.** Top: daily root mean square error (RMSE) and mean error; Middle: maximum and maximum absolute errors; Bottom: percentage of stations with overestimation or underestimation. Shaded regions mark seasonal transitions.

underestimated in magnitude, a large proportion of stations still experience consistent overestimation throughout the year.

D Origins of Station Heterogeneity and Oversmoothing Biases

Despite the increasing availability of gridded reanalysis products such as ERA5, their direct interpolation to station locations often introduces non-trivial biases, particularly in regions with complex terrain or sparse observational coverage. These systematic errors give rise to two fundamental challenges in station-level precipitation modeling: *station heterogeneity* and *oversmoothing*.

Station Heterogeneity. As illustrated in Figures 6 and 7, bilinearly interpolated ERA5 precipitation fields exhibit highly uneven error distributions across space. Some stations experience persistent overestimation, while others are subject to substantial underestimation, particularly during periods of intense rainfall. This spatially inconsistent performance indicates that interpolation biases are inherently location-dependent and cannot be effectively corrected using globally shared model parameters. Importantly, this heterogeneity extends beyond spatial variability. As shown in Figure 8, interpolation accuracy also varies across seasons: errors are generally lower during dry periods, but escalate significantly during the rainy season. This seasonal inconsistency underscores the *temporal heterogeneity* of the interpolation task, where the statistical properties of precipitation—and the associated error patterns—change dynamically over time. Therefore, models that rely on static assumptions or uniform temporal behavior are likely to underperform in real-world applications. Capturing both spatial and temporal heterogeneity is essential for robust and generalizable interpolation.

Oversmoothing of Extremes. The temporal error profile shown in Figure 8 reveals that, although the overall interpolation error increases during the rainy season, the more critical failure lies in the underestimation of extreme rainfall events. In particular, the maximum error metrics in summer often exceed 200 mm/day, while the corresponding observed values are significantly higher. This pattern indicates that coarse-resolution gridded models tend to suppress sharp gradients and attenuate local extremes through spatial averaging. Such smoothing leads to the loss of physically meaningful high-magnitude signals—especially relevant for hydrological risk and extreme event forecasting. Consequently, downstream learning models trained on these smoothed inputs may underrepresent rare yet impactful weather patterns, limiting their utility in high-stakes applications.

Together, these findings highlight the empirical origins of the two central challenges tackled in this work. They motivate the design of interpolation frameworks that are not only spatially adaptive but also capable of preserving localized extremes. Subsequent sections build upon these insights to introduce model components that explicitly address heterogeneity and oversmoothing in a unified manner.

E Theoretical Analysis of OU-Based Temporal Encoding

To support the design of the GeoLiquidOU module, we provide a theoretical derivation of its Ornstein–Uhlenbeck (OU)-inspired update mechanism. This section analyzes how the model integrates historical input over time via exponential smoothing, and how its behavior corresponds to a first-order low-pass filter in the frequency domain.

E.1 Continuous-Time OU Dynamics

We begin with the continuous-time OU process, excluding noise terms for clarity:

$$\frac{dh(t)}{dt} = \kappa [\theta(x(t)) - h(t)], \quad (4)$$

where $h(t)$ is the hidden state, $\theta(x(t))$ denotes an input-driven target signal, and $\kappa > 0$ is the decay rate.

Using an integrating factor $e^{\kappa t}$ and solving the ODE, we obtain:

$$h(t) = h(0)e^{-\kappa t} + \int_0^t \kappa e^{-\kappa(t-s)} \theta(x(s)) ds, \quad (5)$$

which reveals that the current state is a decayed memory of initial state plus an exponentially weighted integration of past inputs. The kernel $K(t) = \kappa e^{-\kappa t}$ acts as a memory filter with decay rate κ .

E.2 Discrete-Time Approximation via Euler Method

To implement the update in discrete time with step size Δt , we apply Euler's method:

$$\begin{aligned} h[n+1] &= h[n] + \Delta t \cdot \kappa [\theta[n] - h[n]] \\ &= (1 - \alpha)h[n] + \alpha\theta[n], \end{aligned} \quad (6)$$

where $\alpha = \kappa\Delta t \in (0, 1)$. This update rule corresponds exactly to an Exponential Weighted Moving Average (EWMA), widely used in both time series filtering and biologically inspired computing.

E.3 Explicit Recursion and Temporal Weighting

Unrolling the recursion across N steps yields:

$$h[N] = (1 - \alpha)^N h[0] + \sum_{i=0}^{N-1} \alpha(1 - \alpha)^{N-1-i} \theta[i], \quad (7)$$

where the contribution of past inputs decays exponentially. The unnormalized weight for timestep i is:

$$w_i = \alpha(1 - \alpha)^{N-1-i}, \quad (8)$$

and the normalized weight becomes:

$$\bar{w}_i = \frac{w_i}{\sum_{j=0}^{N-1} w_j} = \frac{\alpha(1 - \alpha)^{N-1-i}}{1 - (1 - \alpha)^N}. \quad (9)$$

This shows that the discrete update implicitly applies a form of exponential memory over past θ values, where smaller α values yield longer memory horizons.

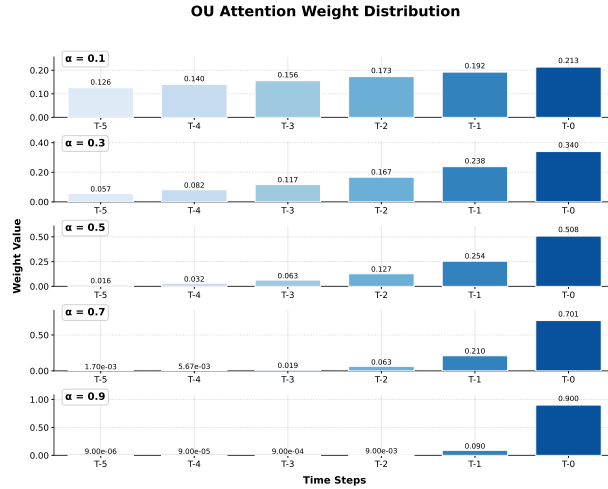


Figure 9: Visualization of OU attention weights.

E.4 Frequency-Domain Interpretation: A Low-Pass Filter

We now analyze the above recursion as a linear time-invariant (LTI) system with input $\theta[n]$ and output $h[n]$. Applying the z -transform yields the transfer function:

$$H(z) = \frac{\alpha z^{-1}}{1 - (1 - \alpha)z^{-1}}. \quad (10)$$

Evaluating on the unit circle $z = e^{i\omega}$ gives the magnitude response:

$$|H(e^{i\omega})| = \frac{\alpha}{\sqrt{1 - 2(1 - \alpha) \cos \omega + (1 - \alpha)^2}}. \quad (11)$$

This confirms that the system behaves as a classic first-order low-pass filter. Its key frequency characteristics are:

- **Zero frequency** ($\omega = 0$):

$$|H(e^{i0})| = \frac{\alpha}{1 - (1 - \alpha)} = 1.$$

This indicates full preservation of low-frequency (DC) components, allowing long-term trends to pass through unchanged.

- **Highest frequency** ($\omega = \pi$):

$$|H(e^{i\pi})| = \frac{\alpha}{\sqrt{(2 - \alpha)^2}} = \frac{\alpha}{2 - \alpha} \ll 1.$$

This shows that high-frequency signals (rapid temporal variations) are strongly attenuated, effectively denoising short-term fluctuations.

- **General behavior:** The magnitude response $|H(e^{i\omega})|$ decreases monotonically with increasing ω . The parameter α controls the cutoff: smaller values of α lead to longer memory and stronger smoothing.

E.5 Interpretation for GeoLiquidOU

This analysis shows that the GeoLiquidOU module dynamically adjusts the temporal sensitivity of each station by learning a site-specific decay factor $\alpha = \tau(p) \cdot \kappa$, where $\tau(p)$ is a location-aware time constant. This enables the model to function as a learnable, spatially-conditioned low-pass filter, selectively integrating temporal context based on local variability.

F Latent-Space Noise Injection and UNet Denoising with Residual Perturbation

Let $z_0 = E(x_0) \in \mathbb{R}^d$ be the latent-space representation of the original interpolated value x_0 . Under the DDPM framework [18, 36], the forward diffusion (noise injection) at step t is defined by

$$z_t = \sqrt{\alpha_t} z_0 + \sqrt{1 - \alpha_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I_d),$$

so that

$$z_t \sim \mathcal{N}(\sqrt{\alpha_t} z_0, (1 - \alpha_t) I_d).$$

Here, $\alpha_t \in (0, 1)$ denotes the precomputed noise-schedule coefficient.

In the reverse (denoising) step, a UNet model $\epsilon_\theta(z_t, t)$ is trained to predict the injected noise. Denote the prediction error by

$$\delta = \epsilon_\theta(z_t, t) - \epsilon,$$

which can be modeled as a zero-mean Gaussian random vector with covariance $\sigma_\delta^2 I_d$. The corresponding one-step reconstruction in latent space is then

$$\tilde{z}_0 = \frac{1}{\sqrt{\alpha_t}} \left(z_t - \sqrt{1 - \alpha_t} \epsilon_\theta(z_t, t) \right) = z_0 - \underbrace{\frac{\sqrt{1 - \alpha_t}}{\sqrt{\alpha_t}} \delta}_{\Delta z}.$$

Since $\delta \sim \mathcal{N}(0, \sigma_\delta^2 I_d)$, it follows that

$$\tilde{z}_0 \sim \mathcal{N}\left(z_0, \frac{1 - \alpha_t}{\alpha_t} \sigma_\delta^2 I_d\right).$$

Because the residual covariance $\frac{1 - \alpha_t}{\alpha_t} \sigma_\delta^2 I_d$ is positive definite, the reconstructed latent vector $\tilde{z}_0$ admits nonzero variance around z_0 . In other words, despite denoising, the model retains a controlled stochastic perturbation in latent space, which—when subsequently decoded—enables recovery or amplification of extreme values beyond the initial interpolated range.

G Cross-Variable Forecasting: Temperature as Input

To verify the adaptability of our model architecture to alternative meteorological variables, we conduct a parallel experiment using 2-meter temperature (T2M) fields from **Pangu-Weather**. While the entire model is retrained from scratch on the T2M dataset, the network structure remains unchanged.

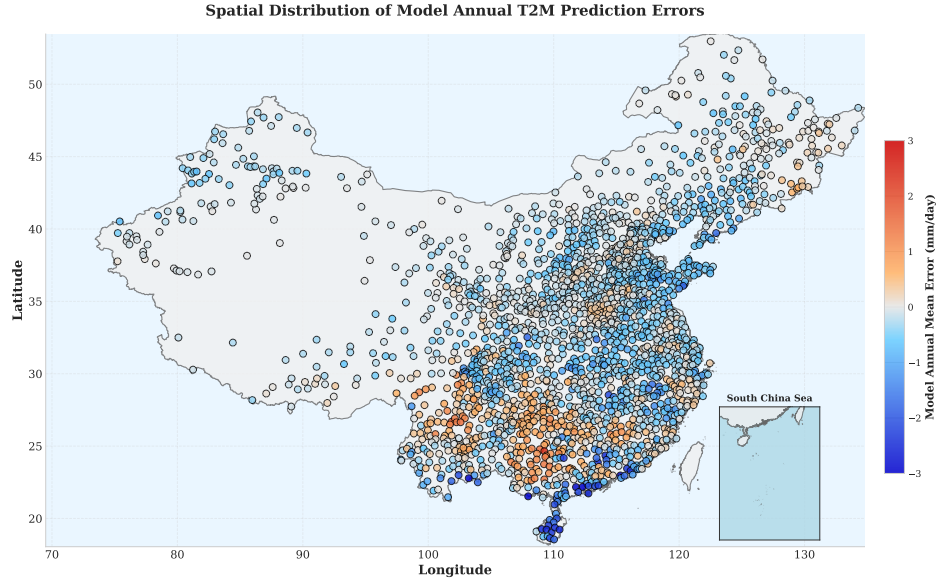


Figure 10: **Spatial Distribution of Absolute Temperature Errors.** The annual mean absolute error (MAE) of 2-meter temperature predictions across China using our method **DiffLiG**.

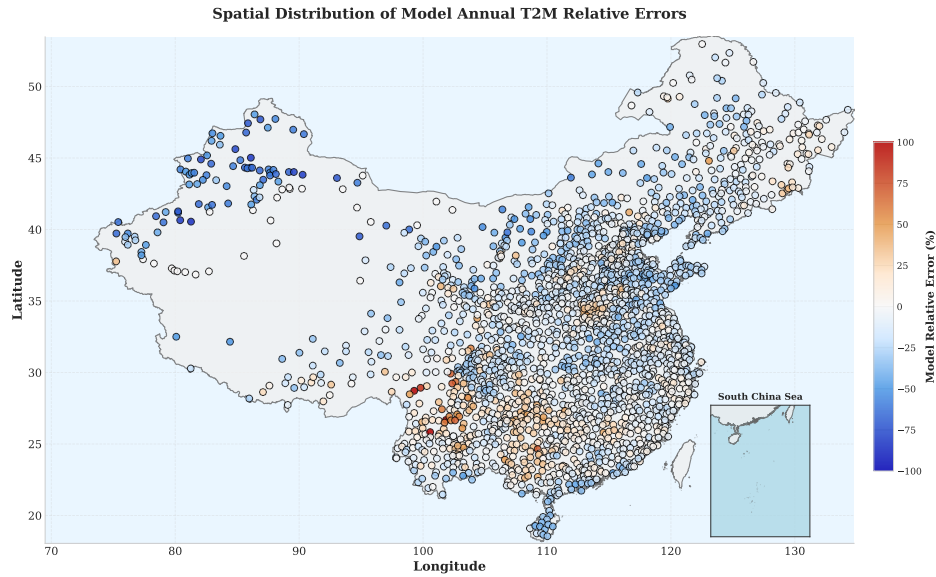


Figure 11: **Spatial Distribution of Relative Temperature Errors.** The relative error percentage (%) of 2-meter temperature predictions, highlighting spatial performance differences across varied terrain and climate zones.

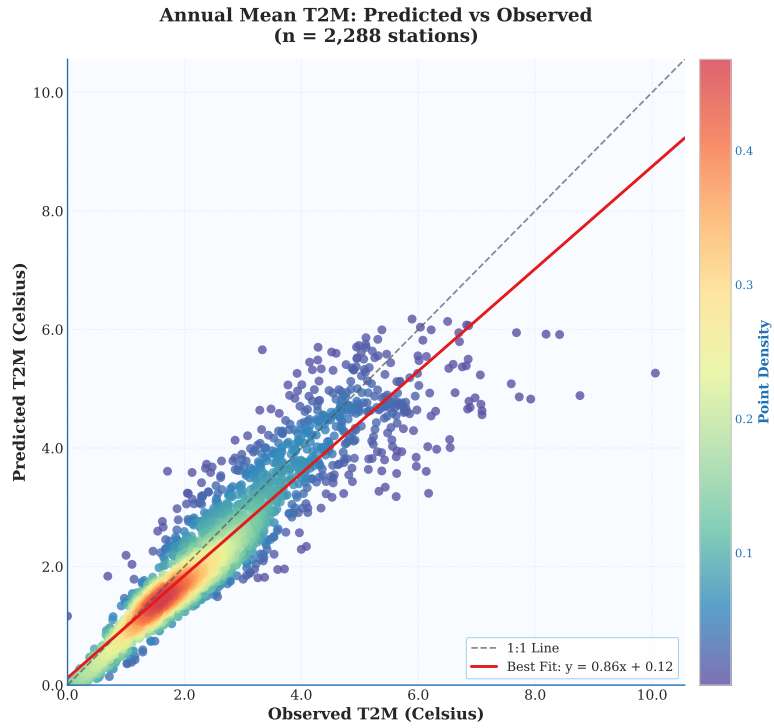


Figure 12: **Scatter Plot of Predicted vs Observed Annual Mean T2M.** This plot shows station-level comparisons between predicted and observed annual average temperatures. A best-fit line ($y = 0.86x + 0.12$) is provided alongside the identity line.

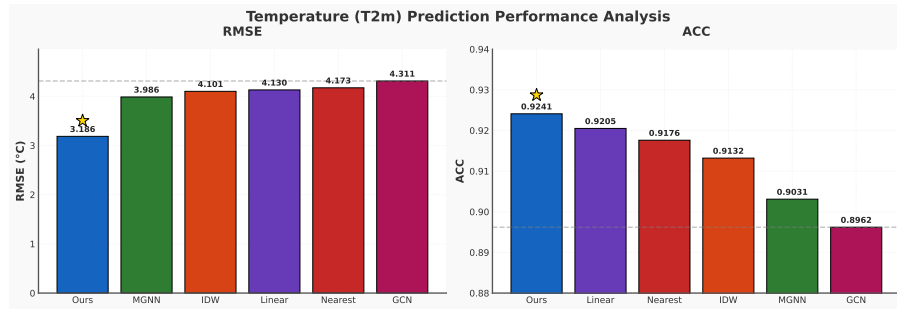


Figure 13: **Performance Comparison on 2-meter Temperature Prediction.** Our method DiffLiG, retrained using Pangu-Weather T2M inputs, achieves the best RMSE and ACC scores among baseline methods including MGNN, GCN, and classical interpolation techniques.

H Sensitivity to Forecast Horizon and Graph Connectivity

To evaluate the robustness and stability of our model across different forecasting and structural configurations, we conduct a series of sensitivity experiments. Specifically, we examine the impact of four critical parameters:

- **Forecast lead time,**
- **Retrospective window size,**
- **Maximum number of edges from ERA5 grid points to stations**
- **Maximum number of inter-station connections.**

Unless otherwise specified, the model is trained with the following default configuration: forecast lead time of **1 day**, retrospective window size of **5 days**, a maximum of **4 connections** from each ERA5 grid point to nearby stations, and up to **3 connections** between observation stations. The goal of this analysis is to assess the sensitivity of model performance with respect to these parameters, thereby providing insight into the design choices behind spatial and temporal coupling strategies.

H.1 Sensitivity to Forecast Lead Time

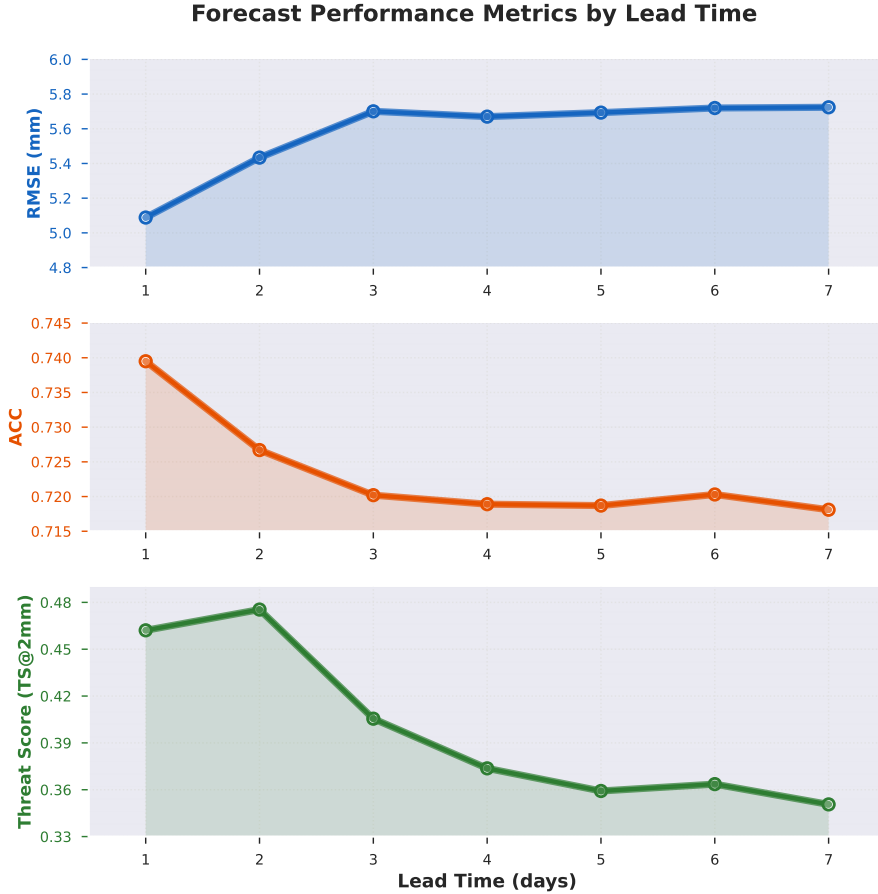


Figure 14: **Impact of Lead Time on Forecast Accuracy.**

To assess how prediction horizon affects model performance, we conduct a sensitivity analysis by varying the *lead time* from 1 to 7 days. As shown in Figure 14, the model retains relatively high accuracy and low RMSE at short-term forecasts. However, performance deteriorates as the prediction window extends, particularly in terms of event detection (TS@2mm). This suggests that while the

model can extrapolate coarse patterns over longer horizons, precise localization of extreme events becomes more challenging.

H.2 Impact of Backtracking Window Size

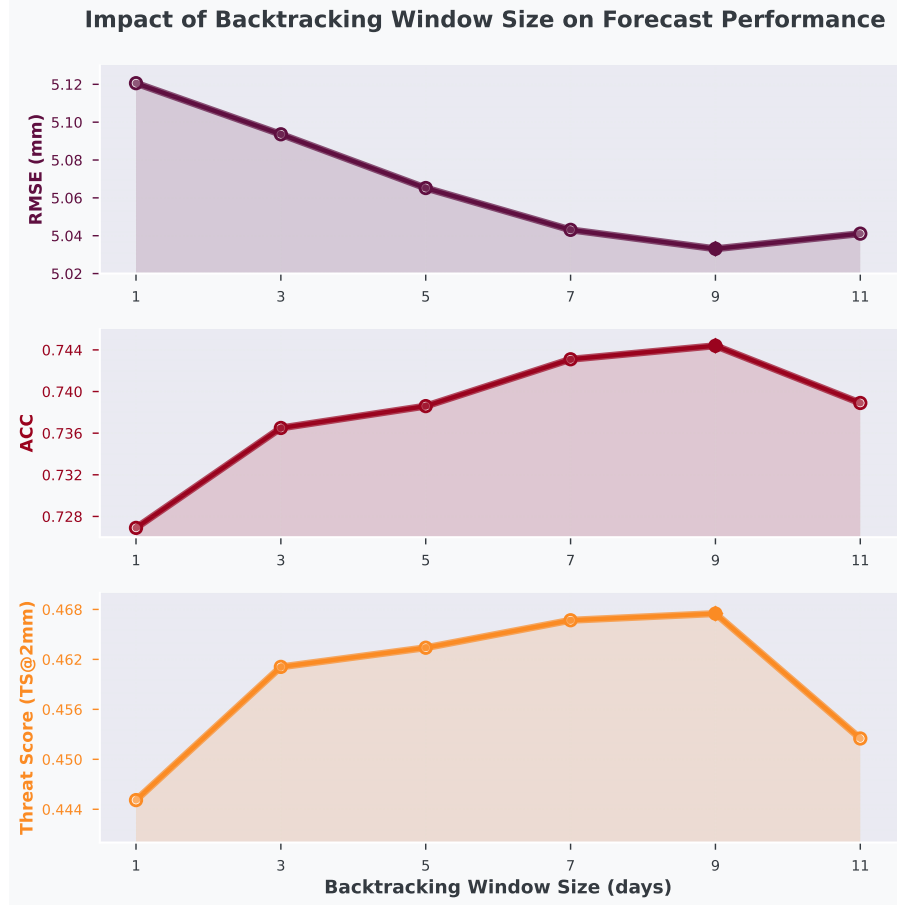


Figure 15: Forecast performance under different temporal backtracking window sizes.

Figure 15 illustrates how the backtracking window size—i.e., the number of past days used for station-level temporal encoding—affects forecast performance. As the window length increases, performance steadily improves across all metrics, suggesting that a longer temporal context helps capture the heterogeneous evolution of local precipitation patterns. To balance accuracy and computational efficiency, we choose a 5-day window as the default configuration.

H.3 Impact of ERA5-to-Station Edge Connectivity

Figure 16 illustrates the impact of varying the maximum number of ERA5-to-station edges on prediction performance. As the number of edges increases, we observe consistent improvements across RMSE. However, the performance gain plateaus beyond 4 edges, indicating diminishing returns with denser spatial connections. Based on this trend, we select 4 as the default edge limit in our model configuration to balance accuracy and computational efficiency.

Impact of ERA5-Station Edge Connectivity on Forecast Performance

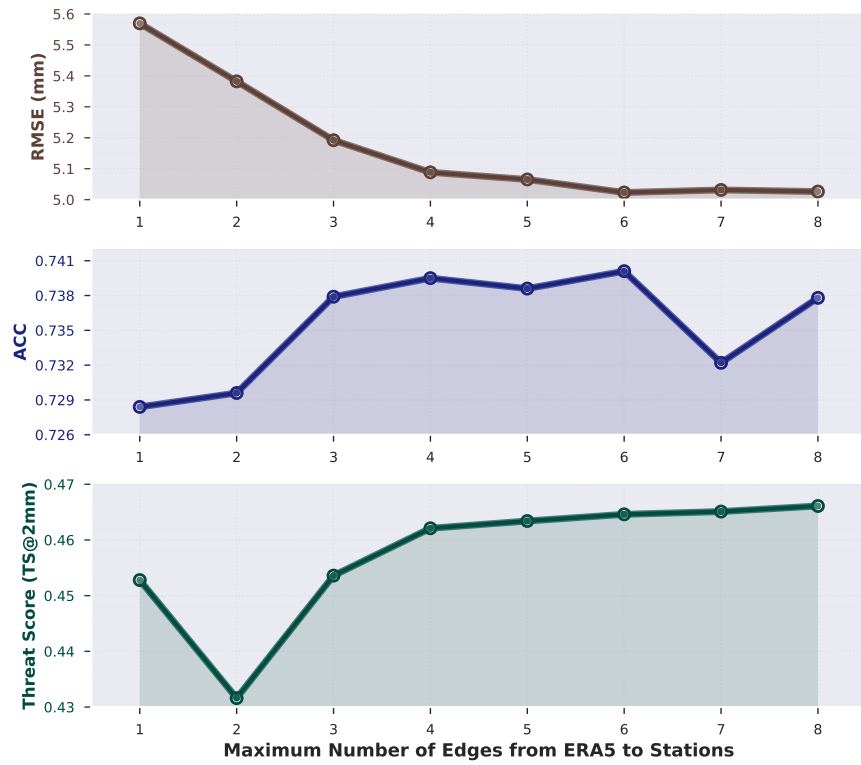


Figure 16: Forecast performance across different ERA5-to-station edge limits.

Impact of Maximum Station-to-Station Connectivity on Forecast Performance

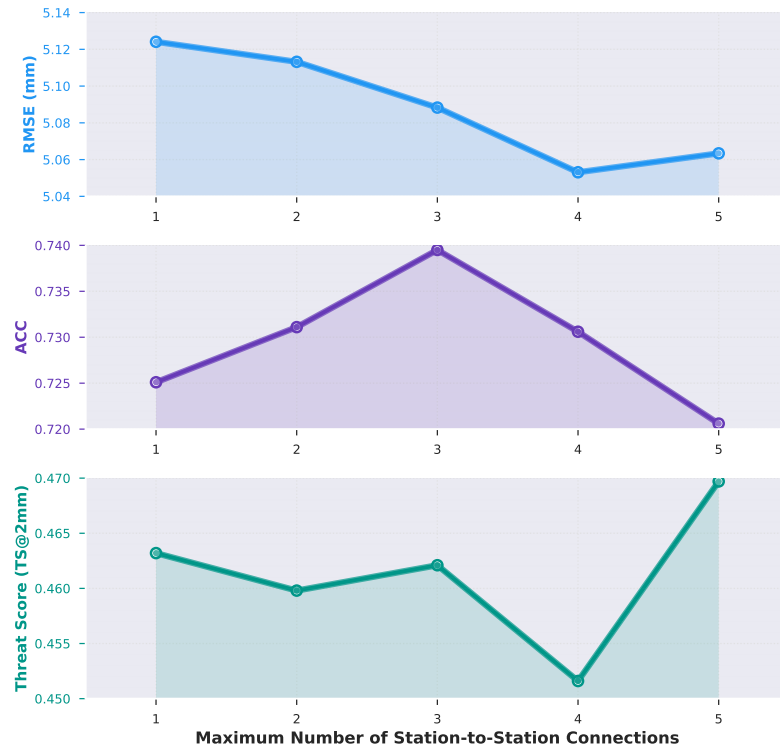


Figure 17: Forecast performance under different inter-station connectivity levels.

H.4 Impact of Station-to-Station Connectivity

Figure 17 shows the effect of varying the number of station-to-station connections on forecast performance. Unlike the ERA5-to-station connectivity, increasing inter-station edge counts leads to only minor fluctuations in RMSE, ACC, and TS@2mm. The performance remains relatively stable across different settings, indicating that the model is less sensitive to this parameter. We therefore fix the maximum number of inter-station edges at 3 for subsequent experiments.

I Implementation and Training Details

I.1 Training Objective and Loss Function

The proposed graph-based forecasting framework is trained in a supervised regression setting. The objective is to minimize the discrepancy between the predicted values and ground-truth observations at meteorological stations, thereby enabling accurate station-level interpolation.

Let $\hat{y}_i$ denote the predicted value and y_i the corresponding ground-truth observation for the i -th sample across time and space. The loss function is defined as the mean squared error (MSE) over all station samples:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N \|\hat{y}_i - y_i\|^2 \quad (12)$$

where N is the total number of training samples. This loss directly supervises the end-to-end prediction pipeline, encouraging the model to align with observed precipitation patterns across both spatial and temporal dimensions.

I.2 Task Configuration

We formulate the task as a station-level spatial interpolation problem, where the model observes past $T_1 = 5$ days of gridded data to predict the station value at the next day ($T_2 = 1$). The temporal granularity is set to daily. For each target station, the maximum number of spatial edges is constrained to ensure computational efficiency and physical relevance:

- **Grid-to-station connections:** at most 4 ERA5 grid nodes;
- **Station-to-station connections:** at most 3 surrounding stations.

We use a 7-year dataset ranging from 2016 to 2022. Specifically, we assign the years 2016–2020 for training, 2021 for validation, and 2022 for final testing. All models are implemented using the PyTorch deep learning framework.

I.3 Training Configuration

The entire training process is conducted on a single NVIDIA RTX 4090D GPU with 24GB of memory. Each training epoch requires approximately 3 minutes. The total training spans 150 epochs and completes in about 7.5 hours. This setup demonstrates strong computational efficiency and scalability.

I.4 Staged Training Strategy

We progressively activate key modules to stabilize training dynamics and ease inter-module adaptation. The training process follows the table 7:

J Haversine Distance Computation

To measure the great-circle distance between two stations on the Earth’s surface, we employ the Haversine formula, which accounts for the spherical geometry of the planet.

Let the two geographic locations be denoted as $\mathbf{p}_i = (\lambda_i, \phi_i)$ and $\mathbf{p}_j = (\lambda_j, \phi_j)$, where λ and ϕ represent the longitude and latitude (in degrees), respectively. These coordinates are first converted into radians:

Table 6: Training Hyperparameters

Parameter	Value
Framework	PyTorch
Optimizer	Adam [20]
Initial Learning Rate	1e-4
Weight Decay	1e-4
Batch Size	16
Epochs	150
Time Window	$T_1 = 5$ (past), $T_2 = 1$ (future)
Grid Edges (max)	4
Station Edges (max)	3
Loss Function	MSE Loss
Dropout	0.1
Activation	LeakyReLU / DynamicTanh [43]
Training Years	2016–2020
Validation Year	2021
Test Year	2022

Table 7: Staged Training Strategy

Stage	Epoch Range	Activated Modules
Warm-up	0–20	Basic GNN backbone with fixed topology
GNN Training	20–40	Activate Dynamic Edge Modulator
Forecast Diversity	40–60	Enable Probabilistic Forecast Diffuser
Selector Adaptation	60–80	Introduce Multi-Criteria Score Selector
Station-wise Bias	80–100	Learn station-specific bias correction
Joint Training	100–150	Fine-tune all modules jointly for synergy

$$\lambda_i^{\text{rad}} = \lambda_i \cdot \frac{\pi}{180}, \quad \phi_i^{\text{rad}} = \phi_i \cdot \frac{\pi}{180} \quad (13)$$

We then compute the difference in latitude and longitude:

$$\Delta\phi = \phi_j^{\text{rad}} - \phi_i^{\text{rad}}, \quad \Delta\lambda = \lambda_j^{\text{rad}} - \lambda_i^{\text{rad}} \quad (14)$$

The Haversine formula estimates the spherical distance d_{ij} between the two points as follows:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_i^{\text{rad}}) \cdot \cos(\phi_j^{\text{rad}}) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (15)$$

$$c = 2 \cdot \arctan 2(\sqrt{a}, \sqrt{1-a}) \quad (16)$$

$$d_{ij} = R \cdot c \quad (17)$$

Here, $R = 6371$ km denotes the average radius of the Earth.

K Baseline Method Descriptions

To evaluate the effectiveness and generalizability of our proposed framework, we compare against a broad set of baseline models, categorized into four methodological groups. These baselines cover a range of paradigms from classical statistical interpolation to advanced neural forecasting models.

(1) Traditional Interpolation Methods. We include four classic spatial interpolation techniques widely used in geostatistics and meteorology:

- **Nearest Neighbor** assigns each station the value of the closest grid point without smoothing, often leading to abrupt spatial transitions.

- **Linear Interpolation** assumes uniform variation between nearby points, but does not capture spatial correlation or terrain effects.
- **Inverse Distance Weighting (IDW)** assigns weights inversely proportional to distance, introducing smooth decay but lacking adaptivity to local density or directional features.
- **Ordinary Kriging (OK)** estimates station values using spatial covariance models (variograms), providing statistically grounded predictions but requiring stationarity assumptions and manual variogram fitting.

While these methods are computationally efficient and interpretable, they rely on static spatial assumptions, lack learnable parameters, and do not account for temporal context or spatial heterogeneity, resulting in degraded performance over complex terrains or data-sparse regions.

(2) Graph Neural Network (GNN) Models. GNNs provide a flexible framework for modeling spatial relationships among stations by learning representations through message passing:

- **GCN** [19] applies localized spectral convolutions on fixed station graphs, but tends to oversmooth signals across the network.
- **GAT** [38] introduces attention mechanisms to weigh neighbor importance, improving expressiveness but still relying on static edges.
- **GraphSAGE** [14] supports inductive learning via sampled aggregation, yet assumes homogeneous information flow across all regions.
- **KCN** [3] combines kernel-based weighting with graph convolutions to capture geostatistical structure, but uses fixed kernels and non-adaptive connectivity.

Despite their spatial modeling strengths, these GNNs often employ fixed topologies and uniform propagation rules, which limit their ability to adjust to station-specific characteristics and introduce oversmoothing in high-resolution prediction tasks.

(3) Physics-Inspired and Multimodal Graphs. We also include two architectures that incorporate physical priors or multimodal inputs:

- **MeshGraphNet** [31] learns graph-based simulators aligned with physical PDEs, capturing spatial continuity but requiring large data and stable mesh structures.
- **MGNN** [39] constructs heterogeneous graphs combining reanalysis grids and observational stations, enabling multimodal fusion but still relying on predefined graph rules and lacking dynamic edge adaptation.

These models offer better spatial fidelity than standard GNNs but often inherit rigid structures or handcrafted designs, limiting flexibility in highly heterogeneous or evolving forecast settings.

(4) Generative and Transformer-Based Spatial Models. To assess uncertainty modeling and spatial diversity, we include:

- **Diffusion Models** [18] generate samples via iterative noise reversal, allowing diverse outputs but often requiring long sampling chains and lacking explicit graph structure.
- **Vision Transformer (ViT)** [11] applies global self-attention over spatial patches, capturing long-range dependencies but prone to global averaging, which may obscure local extremes or abrupt transitions.

These models demonstrate strong generative capacity, yet often lack station-aware structure and may smooth over localized variations, making them less suitable for fine-grained, heterogeneous interpolation tasks.

Together, these baselines form a representative and challenging benchmark suite, allowing us to assess not only interpolation accuracy but also spatial adaptivity, generalization capacity, and robustness to heterogeneity.

L Evaluation Metrics

We adopt three widely used metrics to evaluate interpolation and prediction accuracy for station-level precipitation: Anomaly Correlation Coefficient (ACC), Root Mean Square Error (RMSE), and Threat Score (TS) under a 2 mm threshold. The formal definitions are provided below.

Given the true values y_i and predicted values $\hat{y}_i$ at each location i , we define their respective climatological means $\bar{y}_i$ and $\bar{\hat{y}}_i$ as the multi-year averages over a reference period for the observed and predicted data, respectively. The anomaly correlation coefficient (ACC) is then computed as:

$$\text{ACC} = \frac{\sum_i (y_i - \bar{y}_i)(\hat{y}_i - \bar{\hat{y}}_i)}{\sqrt{\sum_i (y_i - \bar{y}_i)^2} \cdot \sqrt{\sum_i (\hat{y}_i - \bar{\hat{y}}_i)^2}} \quad (18)$$

ACC measures the pattern similarity between predicted and observed anomalies, accounting for inter-location variance.

Root Mean Square Error (RMSE). RMSE evaluates the overall deviation between predictions and ground truth, and is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (19)$$

where N is the total number of evaluated samples. RMSE is sensitive to large errors and commonly used in regression tasks.

Threat Score at 2mm (TS@2mm). To assess the model’s ability to detect light to moderate precipitation events, we calculate the Threat Score (TS) under a fixed threshold of 2 mm/day:

$$\text{TS@2mm} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \quad (20)$$

where TP (true positives), FN (false negatives), and FP (false positives) are determined based on thresholded binary classification:

$$\text{TP} = \sum_i \mathbf{1}[y_i \geq 2 \wedge \hat{y}_i \geq 2], \quad \text{FN} = \sum_i \mathbf{1}[y_i \geq 2 \wedge \hat{y}_i < 2], \quad \text{FP} = \sum_i \mathbf{1}[y_i < 2 \wedge \hat{y}_i \geq 2]$$

TS reflects the model’s event-level accuracy in a binary detection task and is widely used in precipitation forecast verification.

M License and Attribution Details

This study uses several existing assets whose licenses and usage terms are properly acknowledged and respected:

- **Model License.** The proposed model will be released under the CC-BY 4.0 license upon publication.
- **Baselines.** All baseline models used in our experiments are implemented based on open-source releases from their original papers. We cite the corresponding works in our main text, and all implementations respect the original licenses provided by their authors.
- **ERA5 Data.** We use the hourly precipitation product from the ERA5 reanalysis dataset, provided by the European Centre for Medium-Range Weather Forecasts (ECMWF). This dataset is publicly accessible via the [Copernicus Climate Data Store](https://cds.climate.copernicus.eu/) and is licensed under CC-BY 4.0. We use the version released in 2022.

- **CMA Observation Data.** The ground truth observation data from the China Meteorological Administration (CMA) is not publicly available. Access to this dataset was granted to us through proper institutional authorization.

No scraped, re-packaged, or third-party datasets outside of these sources are used in this study.

N Visual comparison of spatial interpolation results

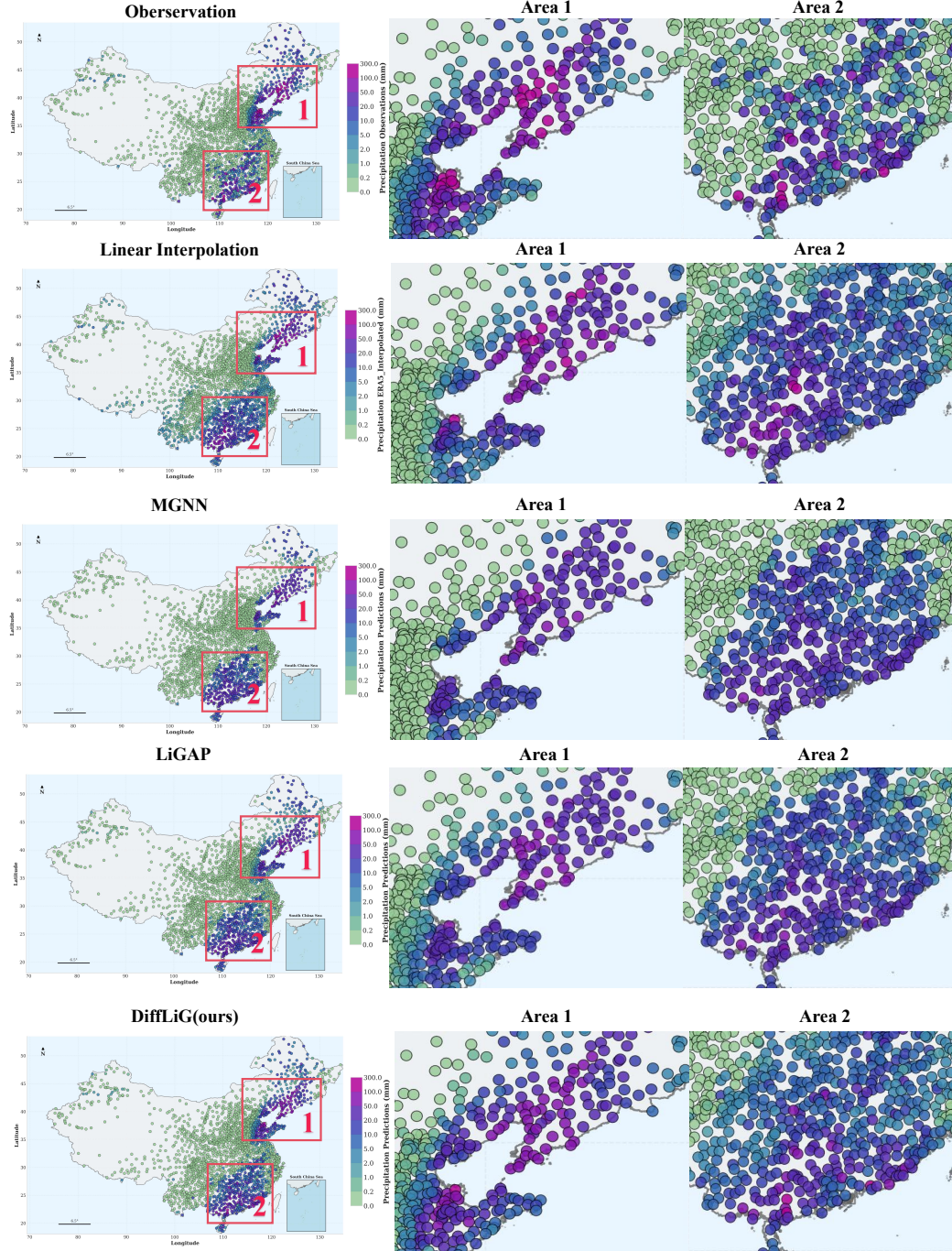


Figure 18: **Visual comparison of spatial interpolation results.** Predicted daily accumulated precipitation for July 5, 2022.