29 April 2026

S2G-DI: A Deep Learning Framework for Generating High-Resolution Wind Gust Fields from Sparse Mesonet Observations

Harish Baki[1], Maximilian Pierzyna[2], Sukanta Basu[1,3]

1. Atmospheric Sciences Research Center University at Albany
2. Department of Geoscience and Remote Sensing Delft University of Technology
3. Department of Environmental & Sustainable Engineering University at Albany

Abstract

The growing deployment of surface observational networks (so-called Mesonets) has the potential to transform many sectors, such as agriculture, water resource management, renewable energy assessment, disaster risk reduction, power outage prediction, and high-impact weather monitoring. However, network observations are sparse, whereas these downstream applications typically require gridded data. Traditional interpolation methods, such as Barnes interpolation, do not perform well in complex terrain. Therefore, in this study, we propose a deep learning–based framework, called Sparse-to-Gridded Deep Interpolation (S2G-DI), to generate high-resolution spatially continuous fields from sparse surface observations. We focus on wind gusts, which are challenging to interpolate due to their highly localized and transient nature, but are critical for assessing weather-related hazards. We experimented with three deep learning architectures, employing convolutional neural networks and transformers with increasing complexity, and our sensitivity analysis shows that a U-Net architecture with meteorological and topographic inputs significantly outperforms Barnes interpolation, reducing RMSE by over 30% and better preserving fine-scale features. The model remains robust to missing data and generalizes well even with limited training data. Evaluations during extreme wind gust events confirm that the framework captures both spatial structure and intensity more reliably than traditional methods. However, for some of these cases, there is still room for improvement for the S2G-DI framework.

https://doi.org/10.22541/essoar.176031304.41826798/v3

Keywords

atmospheric sciences, barnes interpolation, deep learning, mesonet, meteorology, s2g-di, wind gust

S2G Deep Interpolation: A Deep Learning Framework for Generating High-Resolution Wind Gust Fields from Sparse Mesonet Observations

Harish Baki[a] , Maximilian Pierzyna[b] , Sukanta Basu[a c]

[a] *Atmospheric Sciences Research Center, University at Albany, New York, USA*

[b] *Department of Geoscience and Remote Sensing, Delft University of Technology, Delft, Netherlands*

[c] *Department of Environmental & Sustainable Engineering, University at Albany, New York, USA*

Corresponding author: Harish Baki, hbaki@albany.edu

ABSTRACT: The growing deployment of surface observational networks (so-called Mesonets) has the potential to transform many sectors, such as agriculture, water resource management, renewable energy assessment, disaster risk reduction, power outage prediction, and high-impact weather monitoring. However, network observations are sparse, whereas these downstream applications typically require gridded data. Traditional interpolation methods, such as Barnes interpolation, do not perform well in complex terrain. Therefore, in this study, we propose a deep learning–based framework, called Sparse-to-Gridded Deep Interpolation (S2G-DI), to generate high-resolution spatially continuous fields from sparse surface observations. We focus on wind gusts, which are challenging to interpolate due to their highly localized and transient nature, but are critical for assessing weather-related hazards. We experimented with three deep learning architectures, employing convolutional neural networks and transformers with increasing complexity, and our sensitivity analysis shows that a U-Net architecture with meteorological and topographic inputs significantly outperforms Barnes interpolation, reducing RMSE by over 30% and better preserving fine-scale features. The model remains robust to missing data and generalizes well even with limited training data. Evaluations during extreme wind gust events confirm that the framework captures both spatial structure and intensity more reliably than traditional methods. However, for some of these cases, there is still room for improvement for the S2G-DI framework.

1. Introduction

High-resolution gridded meteorological data is essential for a wide range of applications across science and industry. It supports land surface and agricultural studies, including agricultural decision-making (Subedi et al. 2025), soil moisture modeling (Peraza et al. 2025), and snowpack distribution modeling (Le Toumelin et al. 2023). It plays a critical role in natural hazard and disaster management, such as forest fire monitoring and prediction (Chen et al. 2023), and in hydrological extremes, including floods (Amponsah et al. 2018) and droughts (Peng et al. 2020). In atmospheric science, it is used for monitoring and characterization of extreme weather events, such as cyclones (Knapp and Kossin 2007), heatwaves (Hu et al. 2023), and thunderstorms (Houston et al. 2015). Other applications include precipitation forecasting (An et al. 2025), evapotranspiration estimation (Blankenau et al. 2020), and air pollution analysis (Venter et al. 2024). High-resolution data is also crucial for wind resource assessment (Christiansen et al. 2006), as well as for water resource management (Müller et al. 2021). The majority of these applications rely on key meteorological variables, including precipitation, temperature, wind speed and direction, wind gust, humidity, solar radiation, and soil moisture.

In the Continental United States (CONUS), the expansion of mesoscale observational networks (mesonets) has been a key step toward addressing the demand for dense and continuous meteorological measurements, with a total of 27 statewide mesonets established to date (Campbell Scientific, Inc. 2023). The mesonets, including the New York State Mesonet (NYSM), typically have a station spacing of about 30 km and are designed to capture mesoscale atmospheric phenomena (Mahmood et al. 2017). These networks collect raw measurements of variables at 2-3 s intervals, with quality-controlled products made available to the public at 5-10 min resolution. However, despite their high temporal fidelity, the observations are inherently point-based (ungridded) and spatially sparse. In particular, the NYSM network provides measurements at discrete locations, whereas the target analysis fields aim to resolve variability at kilometer-scale resolutions (e.g., 2–3 km) in modern high-resolution systems such as Real-Time Mesoscale Analysis (RTMA) and High-Resolution Rapid Refresh (HRRR). Addressing this spatial sparsity requires transforming point observations into gridded fields, also known as *objective analysis*, which is a longstanding challenge in atmospheric sciences.

The objective analysis involves interpolating irregularly spaced measurements onto a structured grid. Many empirical and statistical techniques have been developed for this purpose, including the Cressman scheme (Cressman 1959), Barnes interpolation (Barnes 1964), kriging (Matheron 1967), multiquadric interpolation (Hardy 1971), Shepard's method (Shepard 1968), and Gandin-based statistical approaches (Gandin 1963). Although most of these techniques were originally proposed in the mid 20th century, they are still employed in several recent studies, such as the Barnes interpolation in the MicroMet framework (Liston and Elder 2006), kriging in the creation of the E-OBS dataset (Haylock et al. 2008), inverse distance-weighted interpolation in deep learning-based wind gust nowcasting (Xiao et al. 2023), and Gandin-based techniques for precipitation analysis (Chen et al. 2008). While effective for some applications, these techniques make simplifying assumptions and often fail to capture the complex geographical and topographical effects present in real-world data.

These early objective analysis techniques laid the foundation for sophisticated data assimilation (DA) frameworks, which integrate observations with numerical weather prediction (NWP) models to produce physically and dynamically consistent gridded fields of meteorological variables (Kalnay 2003). Modern DA frameworks, such as the two/three/four-dimensional variational assimilation (2D/3D/4D-Var), the ensemble Kalman filter (EnKF), and hybrid approaches can handle a wide range of heterogeneous observations, including conventional surface and upper-air observations from weather balloons, radiosondes, surface stations, aircraft, and buoys, as well as remotely sensed data from radar, satellite radiances, and GPS-based retrievals (Barker et al. 2012). These frameworks operate in close association with specific NWP models, such as the Weather Research and Forecasting (WRF), Integrated Forecast System (IFS), Global Forecast System (GFS), since they require model-based background fields, ensemble forecasts, and error covariance information for continuous operation. Advances in NWP-DA systems have enabled the development of state-of-the-art gridded meteorological products (Mankin et al. 2025), including the 5th generation European Center for Medium-range Weather Forecast (ECMWF) ReAnalysis (ERA5) (Hersbach et al. 2020), Copernicus European Regional ReAnalysis (CERRA) (Ridal et al. 2024), CONUS404 (Rasmussen et al. 2023), and Real-Time Mesoscale Analysis (RTMA) (De Pondeca et al. 2011). Despite their demonstrated accuracy, however, these systems are computationally expensive due to their reliance on large-scale integrations and ensemble techniques, which restrict operational

update frequencies to hourly or longer intervals. This trade-off between accuracy and computational expense motivates the exploration of alternative, data-driven approaches, without the reliance on complex DA and NWP model integrations to transform station-based observations into gridded fields.

Deep learning (DL) methods have recently emerged as a promising alternative for generating gridded meteorological fields from sparse observations. A detailed review of relevant DL-based approaches is presented in Appendix A. In this work, we introduce the Sparse to Gridded Deep Learning-based Interpolation (S2G-DI) framework for directly reconstructing high-resolution gridded fields from sparse NYSM observations, covering the entire state of New York. Among the key observed meteorological variables, we demonstrate the capabilities of the S2G-DI framework with wind gusts as an illustrative example. Wind gusts are chosen because they are high-impact weather events leading to severe societal and economic consequences (Kahl 2020) and are complex to represent in NWP-DA systems (Sheridan 2018).

The proposed S2G-DI framework is designed to produce gridded wind gust fields directly from sparse mesonet observations in a computationally efficient manner. It follows a decoupled training–inference strategy: during training, the model learns the mapping between sparse and gridded fields using a gridded NWP-based product (RTMA in this study) as reference, while inference requires only real-time station observations. This design eliminates amplitude and displacement errors (Gilleland et al. 2009) between NWP products and observations, enabling gridded fields to be generated at the native frequency of the mesonet (5 min for NYSM) with fine spatial resolution (2.5 km) and minimal computational cost. This approach eliminates the need for NWP-DA-based products during inference, supporting near real-time analysis. Such analysis fields enable rapid forecast updates, which are crucial for wind gusts. This study has three main objectives: (i) to develop the S2G-DI framework for producing gridded wind gust fields from sparse mesonet observations, (ii) to assess its sensitivity to model architecture, input configurations, data availability, and applicability to other meteorological variables, and (iii) to evaluate its ability to capture extreme gust events in real-world conditions. Performance is benchmarked against the Barnes interpolation method (Barnes 1964), a widely used operational objective analysis technique.

The remainder of the paper is organized as follows. Section 2 describes the S2G-DI framework. Section 3 presents the study area, observational networks, and simulated and observed data used

for model training and inference. Section 4 presents deep learning methodology and model performance evaluation methods. Section 5 present results for the sensitivity analysis on identifying optimized framework, while Sections 6, 7 and 8 present the evaluation of the optimized framework with real observations as well as an independent source data, and framework's generalization across meteorological variables, respectively. Section 9 concludes the paper and discusses directions for future work.

2. S2G-DI framework

The S2G-DI framework is illustrated in Fig. 1, which follows a decoupled training–inference approach through two phases. The first phase is the *training/validation/testing phase* (Fig. 1(a–d)), where a deep learning (DL) model is trained on a reference high-resolution gridded meteorological dataset (acting as a proxy, RTMA in this study), to learn the reconstruction of continuous fields from sparsified versions of themselves, assuming spatial similarity between the gridded and observational data. This phase involves four main steps:

(a) A high-fidelity 2D gridded field of the target variable of interest at a time instance is selected from the reference proxy dataset $X_{ref}^{(hf)} \in \mathbb{R}^{n \times m}$ (Fig. 1(a)).

(b) Using a set of known station coordinates, values from the $X_{ref}^{(hf)}$ are extracted at those station locations, resulting in a sparse 1D vector $X^{(s)} \in \mathbb{R}^{s}$ (Fig. 1(b)). This mimics the observed values from sparse stations.

(c) The $X^{(s)}$ is then nearest-neighbor interpolated onto the full latitude–longitude grid of the $X_{ref}^{(hf)}$. This produces a low-fidelity 2D gridded field $X^{(lf)} \in \mathbb{R}^{n \times m}$ with the same shape and coordinates as the $X_{ref}^{(hf)}$, but introduces discontinuities and artifacts that degrade fidelity (Fig. 1(c)). This intermediate gridding step enables the formulation of the problem within an image-to-image learning framework. Although the original observations are spatially sparse point measurements, representing them on a structured grid allows the use of convolutional architectures, which are well suited for learning spatial dependencies.

(d) This $X^{(lf)}$ is provided as input to a deep learning model, which outputs a high-fidelity 2D prediction $X_{DL}^{(hf)} \in \mathbb{R}^{n \times m}$, trained by minimizing the loss relative to the reference field $X_{ref}^{(hf)}$ (Fig. 1(d)).

For clarity, the framework is described in a variable-agnostic manner so that X can represent any variable of interest. Here, m and n represent the grid points along latitude-longitude directions, and s represents the number of stations. The pipeline is $X_{ref}^{(hf)} \rightarrow X^{(s)} \rightarrow X^{(lf)} \rightarrow X_{DL}^{(hf)} \Leftrightarrow X_{ref}^{(hf)}$. This procedure enables the DL model to learn a mapping from sparse, discontinuous inputs to spatially continuous and physically consistent outputs.

The second phase is the *inference phase*, during which the real sparse observations from mesonet (e.g., NYSM) stations are used to obtain a continuous gridded field of the same, utilizing the trained DL model from *training phase* (Fig. 1(b*–d*)). This involves three steps (b*–d*), which are consistent with the steps (b–d) from the *training phase*:

(b*) The observed values of the selected variable are obtained from the NYSM stations, resulting in a 1D vector $X^{(s)}$ (Fig. 1(b*)).

(c*) The $X^{(s)}$ is then nearest-neighbor interpolated onto the full latitude–longitude grid, resulting in a low-fidelity 2D gridded field $X^{(lf)}$ (Fig. 1(c)).

(d*) Now, this $X^{(lf)}$ is provided as input to the trained DL model from *training phase*, which outputs a high-fidelity gridded field $X_{DL}^{(hf)}$.

The pipeline is $X^{(s)} \rightarrow X^{(lf)} \rightarrow X_{DL}^{(hf)}$. Since the DL model learned to interpolate the sparse field (c) to the full spatially consistent field (d), we assume that the same interpolation applies when using the real sparse observations as input (c* to d*). In other words, based on spatial correlations and features learned from proxy data during training, the model can interpolate real sparse observations to a full spatially consistent analysis field.

The key strength of the framework is its decoupled design, which reduces mismatches between the temporal resolution of gridded mesoscale datasets and station observations. Each timestamp is treated as a stand-alone sample, so the method does not depend on aligned time steps across datasets. This flexibility allows gridded fields to be generated at the observation frequency (5 min for the NYSM) even when the training dataset has a coarser resolution (1 hr for the RTMA). The method can be trained with alternative mesoscale products, such as HRRR (Dowell et al. 2022) or CONUS404 (Rasmussen et al. 2023), and can be adapted to other regions, observational networks, and variables. Additionally, the NWP-based proxy dataset is only required during training, whereas the inference phase relies solely on station observations. This design removes

dependence on operational NWP models, enabling gridded fields to be generated directly from sparse observations with minimal computational cost. As a result, the framework supports near real-time updates, a significant advantage over conventional DA–NWP systems.

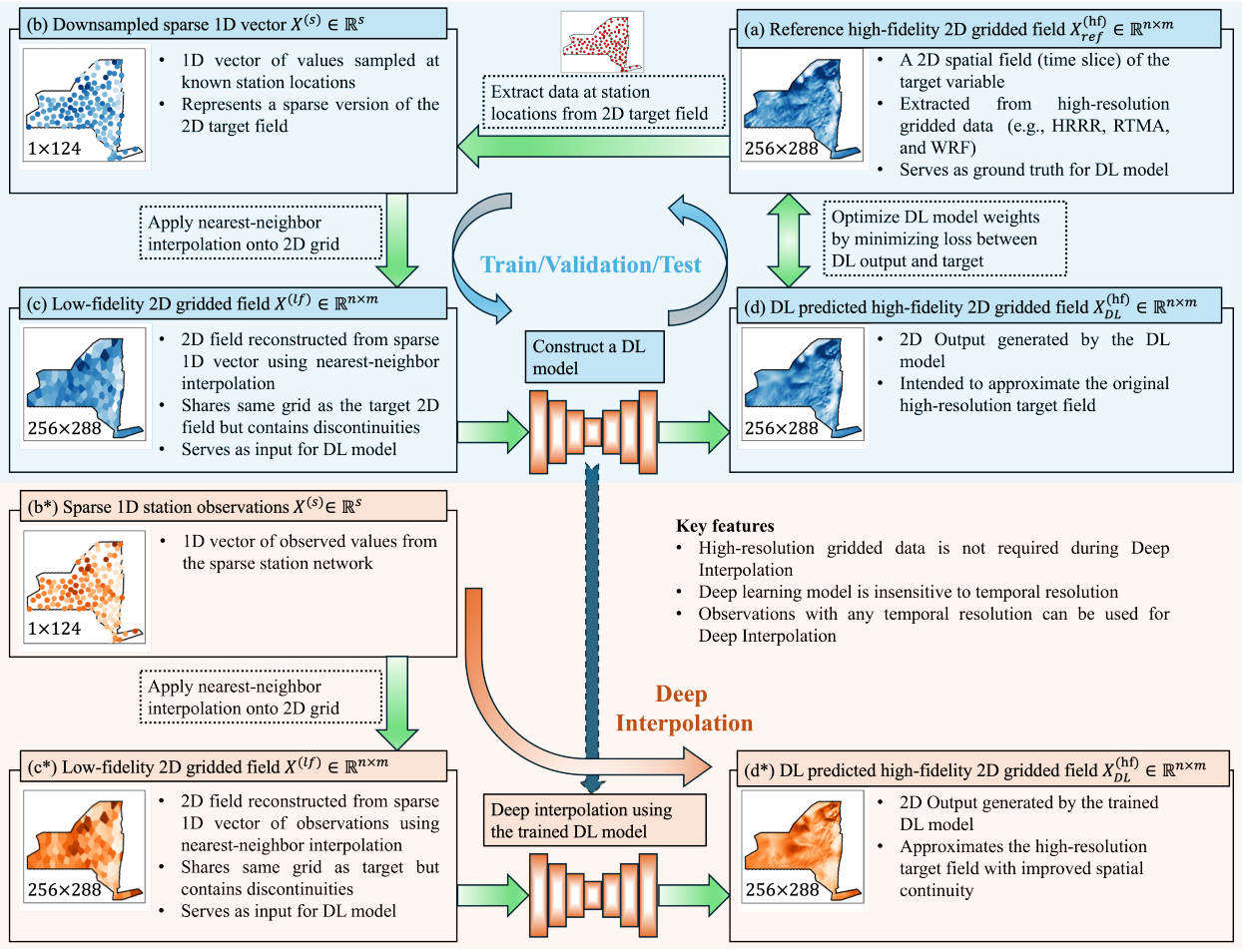


FIG. 1: Flowchart of the S2G-DI framework developed in this study.

3. Data

a. Study area and observation network

Our study focuses on the State of New York (NYS), located between 40.5°–45°N latitude and 72°–80°W longitude (Fig. 2). NYS exhibits a wide range of topographical features, including elevated terrains such as the Adirondack Mountains in the northeast and the Catskill Mountains in the southeast-central region; major water bodies such as Lake Ontario (northwestern NY), Lake Erie (far western NY), the Hudson River, and the Atlantic Ocean; and lower terrain regions,

including the lake plains surrounding Lake Ontario and Lake Erie, the St. Lawrence River Valley (north of the Adirondacks), the Hudson Valley (east of the Catskills), the Mohawk Valley (extending west–east between the Adirondacks and Catskills), Long Island, and the coastal plain.

This topographic diversity results in substantial spatiotemporal variability in meteorological conditions, contributing to a wide range of climatological phenomena across the state. For instance, the lake plains frequently experience snowstorms driven by lake-effect snow processes (Niziol et al. 1995; Niziol 1987). Mountain-valley interactions influence the distribution and intensity of severe weather events such as wind, hail, and tornadoes (Wasula et al. 2002). Additionally, Long Island is subject to coastal meteorological effects from the Atlantic Ocean, including sea-breeze circulations and low-level jet formations (Colle and Novak 2010).

Considering these diverse topographical features and associated climatology in the state of New York, it is of paramount importance to have a denser observational network. For this purpose, the NYSM has been in operation since 2018, with the main goal of monitoring high-impact weather events across the state (Brotzge et al. 2020). It consists of 126 weather stations, out of which 124 are used in this study (Fig. 2) located across nearly all 62 counties, with additional stations in larger or geographically diverse areas to capture local conditions more accurately. The network has an average spacing of 30 km to observe mesoscale weather features. It plays a key role in detecting and tracking events such as hurricanes, tornadoes, severe thunderstorms, flooding, blizzards, snow squalls, ice storms, and extreme temperatures. Each station records a variety of near-surface and subsurface variables, including air temperature (at two heights), dew point temperature, relative humidity, wind speed and direction, pressure, precipitation, solar radiation, snow depth, soil temperature and moisture (at three depths), and visible images.

b. Data for Phase I: Training

In this study, we used the National Centers for Environmental Prediction's RTMA (De Pondeca et al. 2011) as a gridded meteorological reference dataset for the deep learning training. The RTMA is considered a proxy for the observations, since we expect this dataset to capture spatial correlations reliably. The RTMA provides hourly analyses at a horizontal resolution of 2.5 km, has been available since 2006, and is being continuously developed. It incorporates data from both satellite and ground-based observations through data assimilation to attain near-real-time analysis.

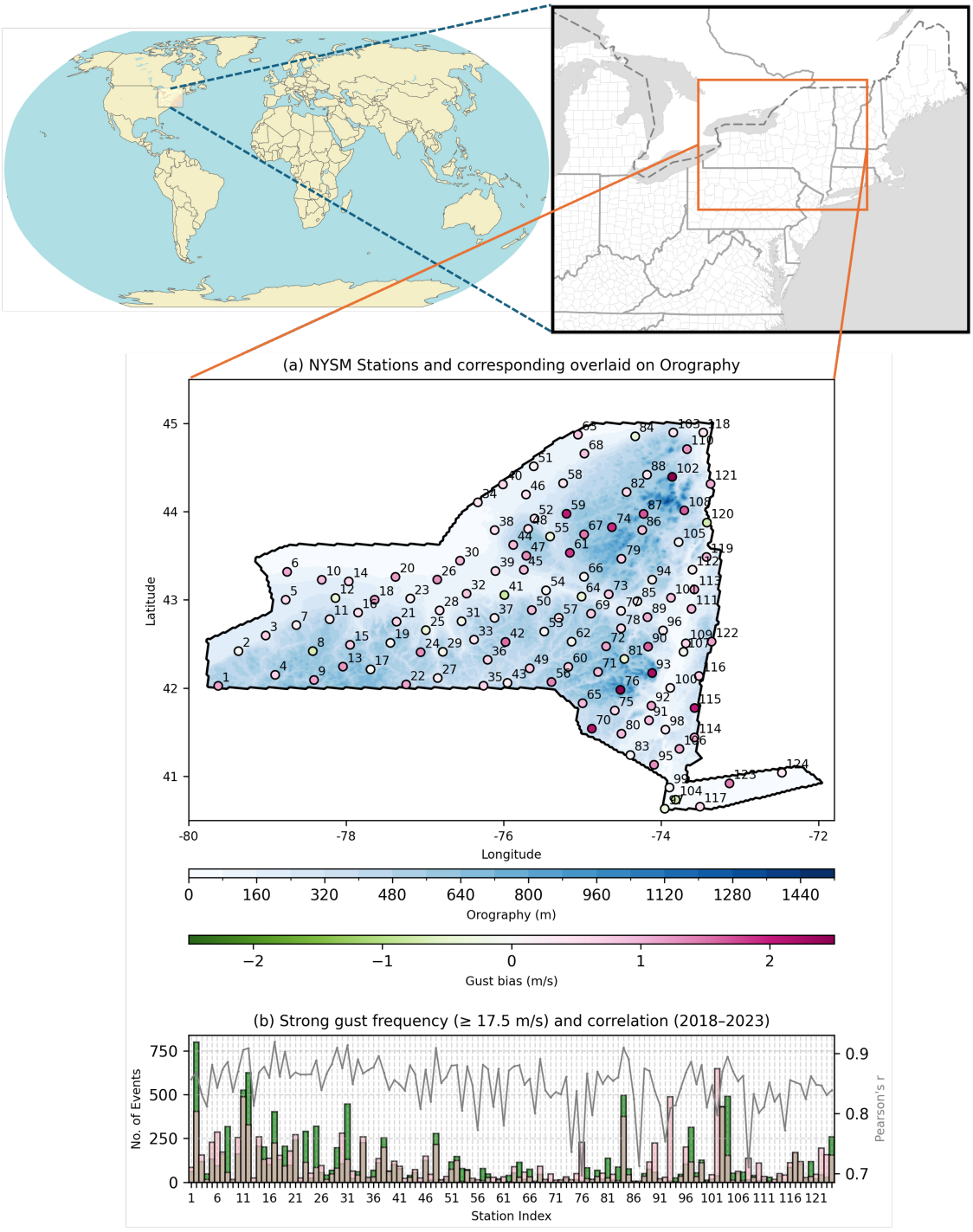


FIG. 2: An illustration of the study area and the NYSM observational network. (a) Spatial distribution of the NYSM stations, numbered in ascending order), overlaid on the orography. The station markers are colored according to the overall mean bias in 10 m wind gust between the RTMA analysis and NYSM observations, computed during 2018–2023 across individual stations. (b) The bar plot (left y-axis) shows the stationwise frequency of strong wind gust events (≥ 17.5 m s^{-1}) seen in NYSM and RTMA datasets; the line plot (right y-axis) indicates the stationwise Pearson's correlation coefficient of wind gust between RTMA and NYSM datasets.

From the RTMA, we used 10 m wind speed (M_{10}), 10 m wind gust (G_{10}), 2 m air temperature (T_2), and 2 m specific humidity (Q_2). These meteorological variables are time-varying. In contrast, the geopotential height at the lowest model level is also used as static topographic information. Further details of how the RTMA analysis is generated, the full list of variables, and their spatial coverage can be found at De Pondeca et al. (2011). We used RTMA data for the period 2018 to 2023, at hourly frequency. To generate spatially continuous fields within New York State, where the NYSM stations are located, a rectangular domain of 256×288 was extracted from the RTMA dataset that covers the entire state. This domain was then masked using the New York State boundary, keeping only the data within the state and setting values outside the boundary to zero. This masking approach was implemented to improve the accuracy of the deep learning models during training.

c. Data for Phase II: Inference

From the 124 NYSM stations, we obtained the same variables as the RTMA, which are M_{10}, G_{10}, T_2, and Q_2, at a frequency of 5 minutes. Since the NYSM records observations at a frequency of 3 s, the wind gust is computed as the maximum within a 5-minute time window.

d. Comparison of observed and reference data

To evaluate the RTMA data with the observations, we selected wind gusts, and the NYSM wind gusts were resampled by taking the maximum within a ±10-minute window centered on each hour to match RTMA's hourly temporal resolution. The RTMA wind gusts exhibit a positive mean bias relative to NYSM observations (Fig. 2(a)), indicating a general tendency to overestimate gust magnitudes. Figure 2(b, left y-axis) presents the frequency of strong wind gust events (≥ 17.5 m s^{-1}), showing a mixed pattern: while several stations report more frequent events in NYSM data, the majority display a higher frequency in RTMA. The Pearson correlation coefficients between RTMA and NYSM gusts (Fig. 2(b), right y-axis) are above 0.85 at most stations, though only a few exceed 0.90. Collectively, these results suggest that RTMA generally overestimates both the magnitude and frequency of wind gusts compared to NYSM observations. Nonetheless, our S2G-DI framework is not vulnerable to this bias, due to the decoupled design.

e. Independent evaluation data

Apart from observational datasets, we also utilize meteorological data from the High-Resolution Rapid Refresh (HRRR) model (Dowell et al. 2022) as an independent validation source. HRRR is a convection-allowing implementation of the Weather Research and Forecasting (WRF) model, operating at a horizontal resolution of 3 km with an hourly data assimilation cycle over the CONUS. From HRRR, we obtained the same variables as in RTMA, namely M_{10}, G_{10}, T_2, and Q_2, at hourly frequency for the year 2023. HRRR data are used exclusively during independent evaluation experiments and are not involved in any stage of model training.

4. Deep Learning methodology

a. Model architectures and training strategy

We chose three types of deep learning (DL) model architectures that vary in parameter count and complexity (Appendices B(a–c)). First, we selected a simple deep convolutional neural network (DCNN) architecture (Fig. B1) with minimal complexity and a low parameter count. Next, we implemented a U-Net architecture (Fig. B2), which is widely used in image processing and related tasks due to its encoder–decoder structure with skip connections. Finally, we increased the model complexity by adopting a U-Net-style architecture with a Swin Transformer backbone (SwinT2UNet; Fig. B3). All three architectures are well established and have been used in a variety of related studies (Fukami et al. 2021; Sunderhaft et al. 2024; Wang et al. 2022). To maintain focus and avoid cluttering the main text with standard model descriptions, we provide detailed explanations of these architectures in Appendices B(a–c).

Following the framework in Fig. 1, the target tensor corresponds to the high-fidelity gridded field of the variable of interest, taken from the reference RTMA dataset. For the 10 m wind gust as the target variable ($X = G_{10}$), the reference field is denoted as $X_{ref}^{(hf)} = G_{10,ref}^{(hf)}$, and is represented as a single-channel tensor of shape batch $(B) \times 1 \times 256 \times 288$. On the other lahd, the DL models take input tensors of shape $\mathrm{B} \times C_{\mathrm{in}} \times 256 \times 288$, where C_{in} denotes the number of input channels. The first input channel corresponds to the low-fidelity (lf) interpolated field $G_{10}^{(lf)}$, as shown in Fig. 1(c). The last input channel is a station mask, which has the same spatial dimensions as the domain (256×288). This binary mask assigns a value of 1 to grid points corresponding to NYSM station locations and 0 elsewhere, explicitly informing the model of the station locations within

the domain. The DL models' output is a single-channel tensor of the same shape as the target, representing the predicted high-fidelity (lf) gridded field $G^{(hf)}_{10,DL}$.

The RTMA data from 2018 to 2021, at an hourly frequency, are used for model training. The years 2022 and 2023 are used for validation and testing, respectively. While Fig. 1 illustrates the framework using a single time instance for clarity, in practice training is performed with mini-batches of samples. The models were trained using the `Charbonnier` loss (Charbonnier et al. 2002), computed between the predicted and target fields:

$$\mathcal{L}(y, \hat{y}) = \sqrt{||y - \hat{y}||^2 + \epsilon^2}, \tag{1}$$

where $y = G^{(hf)}_{10,ref}$ denotes the reference field, $\hat{y} = G^{(hf)}_{10,DL}$ the model prediction, and $\epsilon = 0.001$. When $\epsilon = 0$, the loss reduces to the mean absolute error. Among conventional loss functions, mean squared error (MSE) tends to over-penalize large errors, while mean absolute error (MAE) is more robust but not differentiable at zero. The Charbonnier loss combines the advantages of both: it does not excessively penalize outliers like MSE, yet provides a smooth, continuously differentiable approximation to MAE. It has also been shown to achieve better performance than both MSE and MAE in super-resolution applications (Anagun et al. 2019; Cai et al. 2024).

We aim to predict the optimal continuous spatial field within the NYS. However, the selected domain covers the vast majority of land outside NYS, where the NYSM stations will have little observability. Thus, to improve model predictability and avoid nonphysical predictions, we only considered points lying inside the NY State boundary by masking outside points with zeros in the input and target tensors. In general, one would not expect to predict the target variable values at the station locations, since they are in fact already known by the DL models through the first input channel (interpolated field). Thus, the target variable values at the station locations are preserved by copying them directly from the first input tensor to the output tensor. To reflect this setup in the loss function, a masking strategy was employed: the loss was computed only over the grid points inside the New York State boundary, excluding those corresponding to NYSM station locations. This ensured that model updates were applied only within the masked boundary region, while preserving the NYSM station values unchanged in the output tensor.

The models are optimized using the Adam optimizer with the following hyperparameters: an initial learning rate of 0.003, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and no weight decay. Training

is carried out with an early stopping criterion: if the validation loss does not decrease for 20 consecutive epochs, training is halted. The training continues for a maximum of 200 epochs if early stopping is not triggered. Additionally, an exponential learning rate decay with a decay factor of $\gamma = 0.9$ is applied during training. Each run is trained using two NVIDIA A100 GPUs, each with 80 GB of memory, with a batch size of 16. The models converge after approximately 60–70 epochs, corresponding to a total training time of about 3 hours. Once trained, the inference time for generating a single analysis field is approximately 5 ms on a single A100 GPU, and less than 400 ms when executed on 32 CPU threads on an AMD EPYC 7742 processor using PyTorch. This enables near real-time application even in CPU-only environments.

To assess how various design choices of our framework affect the quality of the S2G-DI analysis fields in comparison to Barnes baseline, we present several sensitivity studies in section 5, with RTMA acting as training data. Further, the validity of the framework is evaluated with true observations from NYSM (section 6) and with HRRR acting as an independent evaluation dataset (section 7).

b. Quantitative evaluation metrics

For qualitative evaluation, spatial maps of predicted and observed fields were visually compared. To quantitatively assess model accuracy, we adopted three widely used image-based metrics: Root Mean Square Error (RMSE) (Eq. 2), Peak Signal-to-Noise Ratio (PSNR) (Eq. 3), and Structural Similarity Index Measure (SSIM) (Eq. 4). These metrics treat both the predictions and targets as 2D spatial fields (images), allowing a pixel-wise comparison. RMSE measures the average magnitude of error, with larger errors penalized more heavily. PSNR quantifies the ratio between the maximum possible signal power and the power of the error, indicating reconstruction fidelity. SSIM evaluates the similarity in luminance, contrast, and structure between the prediction and target fields, and is particularly effective for assessing perceptual similarity.

$$\begin{aligned} \mathrm{RMSE}(y, \hat{y}) &= \sqrt{\mathrm{MSE}(y, \hat{y})} \\ &= \sqrt{\frac{1}{|\Omega|} \sum_{(i,j,t)\in\Omega} (y_{i,j,t} - \hat{y}_{i,j,t})^2} \end{aligned} \tag{2}$$

$$\mathrm{PSNR}(y, \hat{y}) = 10 \cdot \log_{10}\left(\frac{(y_{\max} - y_{\min})^2}{\mathrm{MSE}(y, \hat{y})}\right) \tag{3}$$

$$\mathrm{SSIM}(y, \hat{y}) = \frac{1}{|T|}\sum_{t \in T}\frac{(2\mu_{y_t}\mu_{\hat{y}_t} + C_1)(2\sigma_{y_t\hat{y}_t} + C_2)}{(\mu_{y_t}^2 + \mu_{\hat{y}_t}^2 + C_1)(\sigma_{y_t}^2 + \sigma_{\hat{y}_t}^2 + C_2)} \tag{4}$$

Here, $|\Omega|$ represents the total number of valid time instances and grid points lying inside the domain (excluding the station locations), $|T|$ is the total number of time instances (samples), t is individual time instance (sample), μ_y denotes pixel mean of the target tensor, $\mu_{\hat{y}}$ denotes mean of the predicted tensor, σ_y^2 denotes variance of the target tensor, $\sigma_{\hat{y}}^2$ denotes variance of the prediction tensor, $\sigma_{y\hat{y}}^2$ is the covariance of target and predicted tensors, and $[C_1, C_2]$ are the two constants to stabilize the division with weak denominator.

The visual inspection or the metrics alone may fail to explain differences across spatial scales. Therefore, to provide a scale-aware quantitative assessment, we computed the power spectral density (PSD) of each prediction and target using the two-dimensional Fast Fourier Transform (2D FFT) (Skamarock 2004). The PSD characterizes how the variance of the spatial field is distributed across frequencies (scales). Since the resulting PSD is symmetric across quadrants, we extracted and analyzed only the diagonal of one quadrant to avoid redundancy.

c. Cross-validation strategy

We implemented a random-folding cross-validation (CV) strategy to evaluate model performance under a varying number of inference stations. This approach, illustrated in Fig. 3, simulates scenarios of random station dropout by selectively excluding groups of stations during inference. The 124 stations are divided into four spatially diverse groups, each with comparable intra-group distance characteristics. Two inference configurations are tested: one using three groups for inference and one for testing ($n_{\text{inference}} = 93$), and another using two groups for inference and two for testing ($n_{\text{inference}} = 62$). In the 93 inference stations setup, four folds (F1–F4) are generated by rotating which group is held out as unseen, while the remaining three groups are used for inference. The union of the held-out groups across all four folds ensures that every station is exactly unseen once, providing a single complete set of unseen stations. Similarly, in the 62 inference station configuration, six folds (F1–F6) are created by considering all possible combinations of two groups used for inference and the remaining two as unseen. This results in three disjoint sets of

entirely unseen stations, which can be considered as three ensemble members of the full set of unseen stations.

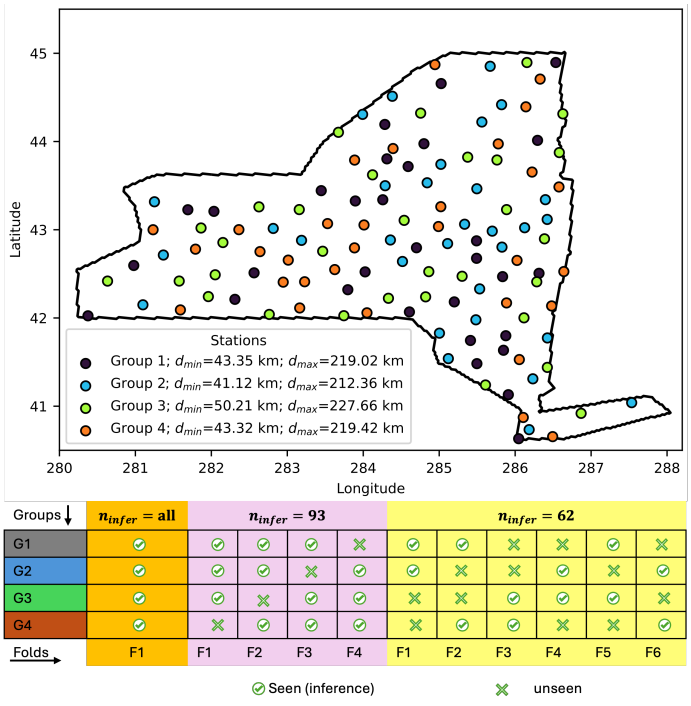


FIG. 3: Station grouping and fold-wise inference strategy for model evaluation. The map (top) shows the spatial distribution of 124 stations across New York, divided into four groups (G1–G4) with the same number of stations (31) and similar intra-group distance statistics. Each group maintains comparable minimum and maximum pairwise distances. The matrix (bottom) illustrates the folding strategies used during inference, with green checkmarks (✓) indicating groups used during inference (seen), and green crosses (×) indicating groups excluded during inference (unseen).

5. Results: Sensitivity analysis with RTMA data

We conduct a sensitivity study to illustrate the influence of various design choices on the deep interpolation performance of our approach, i.e., to generate high-resolution gridded analysis fields from sparse inputs (Fig. 1(a)-(d)). The sensitivity study is conducted using gridded RTMA data only to determine a model configuration that achieves the best possible sparse-to-gridded reconstruction. We employ wind gust as an illustrative target variable for these sensitivity experiments because it is governed by complex processes and, thus, is challenging to model (Kwon and Kareem 2009).

a. Sensitivity to model architecture and input variables

We first aim to optimize the DL model architecture and the subset of available variables to be used as inputs. More specifically, we compare a low-complexity deep convolutional neural network (DCNN, ca. 0.7M parameters, cf. Appendix B(a)), a standard medium-complexity UNet (ca. 10M parameters, cf. Appendix B(b)), and a high-complexity UNet with SWin transformer (ca. 15M parameters, cf. Appendix B(c)). All three architectures are trained on different subsets of input features, with the high-resolution wind gust G_{10} from RTMA as the target in all cases. The different configurations are listed in Table 1. The simplest experiment takes only nearest-neighbor-interpolated G_{10}^{lf} and the station masks as input. More input features are added successively, beginning with the terrain height H_0 from the RTMA dataset, which is provided as a continuous spatial field to inform the DL models about the underlying topography.

Other meteorological variables are the 10 m wind speed (M_{10}), 2 m temperature (T_2), and 2 m specific humidity (Q_2), which are all provided as nearest-neighbor-interpolated fields as explained in Fig. 1(c). Note that the same variables should be available from both RTMA and NYSM observations, as generating analysis fields from only observations during inference is otherwise not possible (cf. Fig. 1(b*–d*)).

The three different architectures and five different subsets of input variables yield 15 distinct model configurations. The model weights of each combination are randomly initialized using Xavier initialization (Glorot and Bengio 2010). To obtain an ensemble and examine the robustness of the performance to different architectures and variables, we repeat each of the 15 experiments five times with different random initializations, resulting in a total of 75 runs. Additionally, Barnes interpolation is employed to generate baseline wind gust analysis fields from the same sparse RTMA wind gust inputs used by the DL models.

The performance of the different configurations with respect to RMSE, PSNR, and SSMI is presented in Fig. 4. All scores are normalized by the scores achieved by Barnes interpolation to illustrate the benefit of our DL approach. To get a qualitative understanding of the performance, examples of analysis fields generated by different models are presented in Fig. 5 as spatial maps. The traditional Barnes interpolation fields are also displayed together with bias maps of the estimated fields compared to the RTMA ground truth. Finally, the average power spectral density (PSD)

TABLE 1: List of experiments conducted to assess the sensitivity of deep interpolation performance to different input variable configurations. Here, lf indicates low-fidility and hf indicates high-fidility.

	Inputs						Target
ID	$G_{10}^{(\mathrm{lf})}$	$M_{10}^{(\mathrm{lf})}$	$T_2^{(\mathrm{lf})}$	$Q_2^{(\mathrm{lf})}$	Terrain height $H_0^{(\mathrm{hf})}$	Station mask	$G_{10}^{(\mathrm{hf})}$
G_{10}	✓					✓	✓
G_{10}-H_0	✓				✓	✓	✓
G_{10}-H_0-M_{10}	✓	✓			✓	✓	✓
G_{10}-H_0-M_{10}-T_2	✓	✓	✓		✓	✓	✓
G_{10}-H_0-M_{10}-T_2-Q_2	✓	✓	✓	✓	✓	✓	✓

displayed in Fig. 6 presents how well the different model configurations can capture patterns of various sizes compared to the RTMA training data.

From Fig. 4 we see that the UNet performs overall best in all three metrics and all variable configurations, followed by SwinT2UNet and DCNN. Error bars indicate that UNet and DCNN are insensitive to random weight initialization, as their performance varies only slightly. This is not the case for SwinT2UNet, where different initializations results in different scores, as indicated by larger error bars. As the SWinT2UNet has 50% more parameters to tune than the UNet, the increased variance of the transformer UNet scores may also indicate too large a model capacity with respect to our training data. The spatial maps (first row) reveal significant qualitative differences between Barnes interpolation (b*) and the three DL methods (c* – e*). All three DL methods yield significantly better performance than Barnes interpolation, as is evident qualitatively from the maps and quantitatively from the statistics discussed before. The DL methods recover more spatial details related to topography compared with Barnes interpolation, as shown in the bias maps in the second row. That is also visible from the PSD in Fig. 6 where the curves of the three deep learning models overlap. All three DL models are close to the RTMA ground truth (black), indicating that spatial patterns of all scales are matched well. The PSD of Barnes interpolation (blue), on the other hand, deviates significantly from the ground truth and the DL models at wavelengths smaller than 40 km, indicating that finer structures, such as terrain influences, are missed. This difference is expected, as Barnes interpolation does not consider orography, whereas our framework can take it as input.

[t]

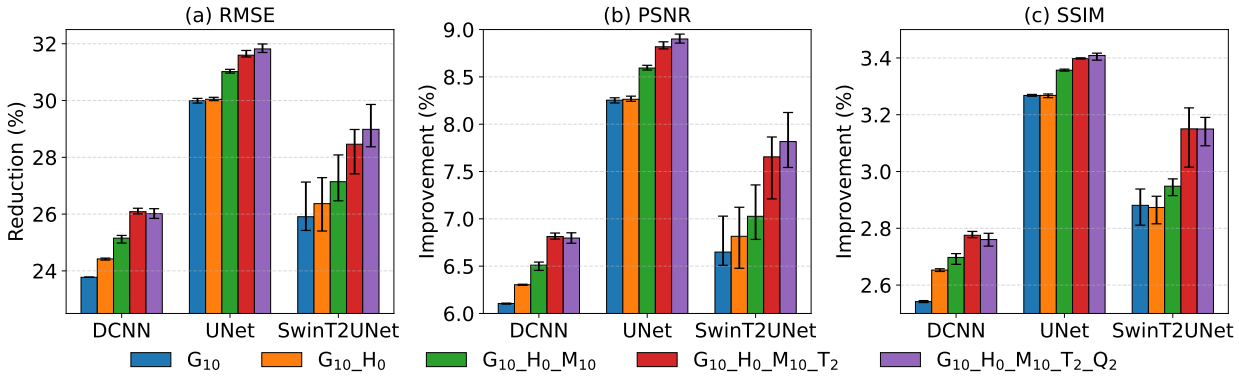


FIG. 4: Change of relative performance of our deep interpolation framework depending on the complexity of the deep learning model (DCNN < UNet < SwinT2UNet) and the number of input variables used (cf. Tab. 1). Statistics are given relative to traditional Barnes interpolation. Error bars indicate sensitivity to random weight initialization.

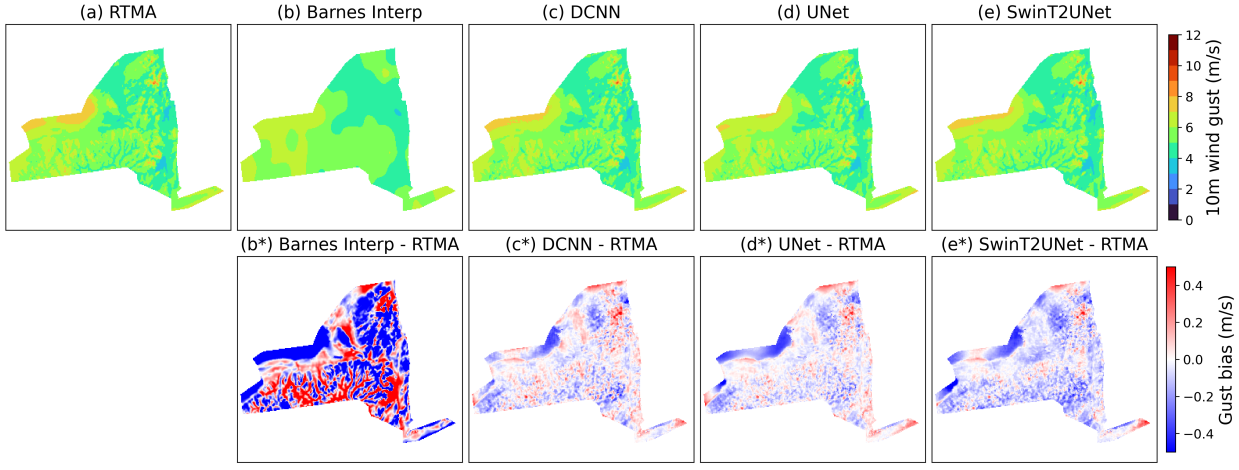


FIG. 5: (First row) Comparison of the annual spatial mean 10 m wind gust from (a) RTMA, (b) Barnes interpolation, (c) `DCNN`, (d) `UNet`, and (e) `SwinT2UNet`. The deep learning models are configured with the best input configuration (G_{10}_H_0_M_{10}_T_2_Q_2). (Second row) Spatial bias of the Barnes interpolation (b*) and deep learning models (c* – e*) relative to the RTMA annual mean 10 m wind gust.

Considering also the different input configurations presented in Fig 4, it is clear that adding more variables improves the performance for all DL architectures. Adding orography does not seem to improve the performance of the two UNets much compared to DCNN, but providing M_{10} and T_2 yields better scores across all metrics and models. Including also the specific humidity Q_2, only yields a minor performance improvement, suggesting that it does not contain much extra information for gust prediction compared to the previously added variables. Nevertheless, we keep Q_2 to achieve maximum performance. The overall performance increase due to extra variables is

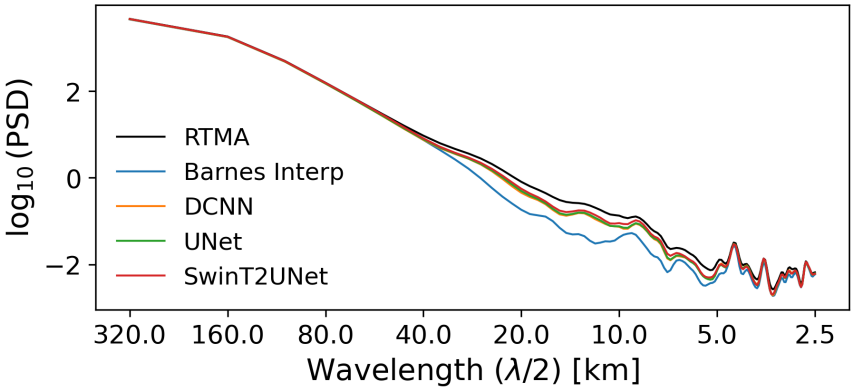


FIG. 6: Annual mean power spectral density (PSD) of RTMA (black) compared with interpolated fields using Barnes's method (blue) and different deep learning architectures provided the best input configuration (G_{10}_H_0_M_{10}_T_2_Q_2).

attributed to physical dependencies between variables, which improve the predictability of wind gusts.

Intermediate result: While all DL architectures are close to each other in performance, we select the UNet because of its overall high skill, its insensitivity to random initialization, and its balanced complexity. As adding more variables improves performance, we continue the study with the UNet trained on all inputs.

b. Sensitivity of Barnes interpolation: bias correction

To further investigate the origin of the performance gains observed for the deep interpolation framework, we design a modified Barnes configuration that explicitly tests whether the improvements of the U-Net can be attributed to distributional bias correction alone.

We apply a grid-point quantile mapping (QM) correction (Themeßl et al. 2011) to the Barnes-interpolated fields (hereafter Barnes_QM), using RTMA as the reference dataset. The QM correction is constructed during the training period (2018–2021) by first generating Barnes-interpolated fields from RTMA data using the same station sampling strategy as in the main experiments. For each grid point, empirical cumulative distribution functions are estimated for both the Barnes-interpolated values and the corresponding RTMA values using 1000 quantile intervals to adequately resolve the full distribution, including its upper tail.

The quantile mapping function is defined at each grid point by pairing corresponding quantiles of the Barnes and RTMA distributions, thereby establishing a transfer function that maps Barnes

values to RTMA-consistent values. During the evaluation period (2023), this precomputed mapping is applied to Barnes-interpolated fields generated from RTMA-based station inputs, without further adjustment. In this way, Barnes_QM preserves the spatial structure imposed by the interpolation while enforcing the marginal distribution of the reference field at each grid point.

This experiment isolates the role of distributional alignment in the observed performance gains and assesses whether correcting marginal distributions alone is sufficient to reproduce the spatial characteristics captured by the deep learning approach. As shown in Fig. 7, the annual mean gust field from Barnes_QM is visually closer to RTMA in terms of small-scale texture compared to the Barnes baseline, which is expected given the imposed RTMA-consistent distributions. The correction drastically reduced the overall bias compared to Barnes baseline, but not as better compared to the UNet. Further, the correction only aligns the mean characteristics, but fails at temporal consistency.

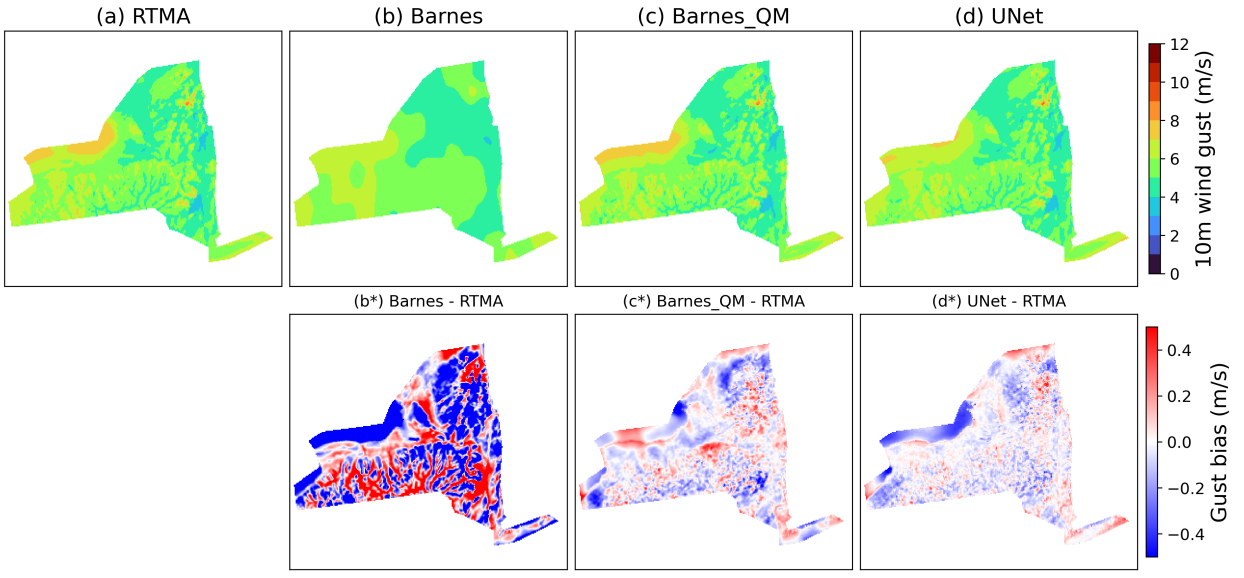


FIG. 7: Comparison of annual mean 10 m wind gust fields over New York State using (a) RTMA, (b) Barnes interpolation, (c) grid-point quantile-mapped Barnes, and (d) UNet. Panels (b*)–(d*) show the corresponding bias fields relative to RTMA. All fields are aggregated over the evaluation period 2023. Barnes and UNet are included for reference.

Table 2 supports these findings quantitatively. QM-based correction yields only modest improvements over Barnes (7.5% RMSE reduction, 1.1% PSNR gain, and 1.1% SSIM gain), indicating that aligning the marginal distribution alone does not substantially improve the reconstructed fields.

This reflects the locality of the QM correction, which does not enforce temporal frame-by-frame matching.

In contrast, the U-Net yields the strongest gains across all metrics, including a 31.7% reduction in RMSE and a 3.4% increase in SSIM. These improvements demonstrate that the superior performance of the deep interpolation framework cannot be explained by bias correction alone, but instead arises from its ability to learn spatial and temporal coherent relationships between meteorological variables and underlying characteristics.

TABLE 2: Relative performance of different methods compared to baseline Barnes interpolation using RTMA as reference.

Method	RMSE reduction (%)	PSNR improvement (%)	SSIM improvement (%)
Barnes_QM	7.5	1.1	1.1
UNet	31.7	8.9	3.4

c. Sensitivity to amount of training data

The next step of the sensitivity study is to successively reduce the training data, one year at a time. Specifically, the models are trained on data from 2018–2021 (4 years, same as before), 2018–2020 (3 years), 2018–2019 (2 years), and 2018 (1 year). These experiments aim to assess the influence of training data size on model performance. In all cases, 2022 and 2023 are kept for validation and testing, respectively, and the optimal UNet architecture with all inputs is used. The motivation behind this step is to assess the applicability of our framework for regions where no extensive proxy dataset like RTMA is available. Generating a similar high-resolution dataset from scratch using numerical weather models is expensive, so running as few simulations as possible for training data generation is desirable.

Tab. 3 presents the improvement of RMSE, PSNR, and SSIM over Barnes interpolation for four models trained on different ranges of training data. As expected, performance decreases across all metrics when fewer training data are used. We attribute this to the variability of weather across different years, so datasets of only a few years are expected to contain less diverse weather, resulting in less generalized models. Even then, the UNet trained on only one year (2018) achieves a significantly improved statistics (reduced RSME, and increased PSNR and SSIM) compared

TABLE 3: Change of relative performance of our deep interpolation technique compared to Barnes interpolation when one, two, three, or four years of gridded RTMA data are used for training. The marked (*) configuration is used in other parts of the sensitivity study.

Train years	RMSE reduction (%)	PSNR improvement (%)	SSIM improvement (%)
2018-2021(*)	31.7	8.9	3.4
2018-2020	30.5	8.4	3.2
2018-2019	29.2	7.9	3.1
2018	26.4	6.9	2.8

to the traditional Barnes method. This result is very promising for applications in regions with fewer available proxy data. Comparing the change in scores from one year of training data to four years also indicates a diminishing return on using more and more data. For RMSE, for example, training on two years compared to one year yields an improvement of 2.8 percent points, whereas going from three to four years only improves the scores by 1.2 percent points. Similar patterns are observed in the other metrics. We expect this trend to continue with performance saturating at some point.

Intermediate result: In conclusion, we demonstrate that a single year of training data yields a model that outperforms the traditional Barnes interpolation, and that adding more years of data has a positive but gradually diminishing effect. For the rest of the study, we continue with all four training years and the all-variable UNet to achieve the best possible performance.

d. Random dropout of training stations

The final step of the sensitivity analysis assesses the effect of data missing from some stations during inference, i.e., when the analysis field is generated from incomplete sparse inputs. In an extensive observational network, such as NYSM with 100+ stations, it is common that data is missing at a few locations for certain periods due to sensor failures, maintenance, or communication problems. In general, our framework is robust against that, as the nearest-neighbor interpolation (cf. Fig 1b* and c*) can always generate a 2D field, which can be refined by the DL model. However, having fewer sparse inputs results in different polygon patterns of the interpolated fields. In the following, we aim to assess the impact of these different patterns on the final output and stress that dropout during training is, in fact, crucial to make the model robust against different polygon patterns, and thus, missing data during deep interpolation.

To assess the impact of missing stations on model performance, we drop stations during training and during inference. As training happens on synthetic gridded RTMA data, missing values do not typically appear at this stage. The motivation for still dropping stations is to simulate different network configurations during training and to assess the robustness of the training. We begin by utilizing all 124 NYSM stations during training and successively reduce the number of stations to $n = 100$, $n = 75$, and $n = 50$. More specifically, during each epoch of the training, we randomly select only n stations of each sample and interpolate them using nearest neighbors, resulting in different polygon patterns for each sample and epoch. The resulting four models are called $\mathrm{DL}_{n_{all}}$, $\mathrm{DL}_{n_{100}}$, $\mathrm{DL}_{n_{75}}$, and $\mathrm{DL}_{n_{50}}$ in the following. Note that these training stations are virtual stations, where RTMA data are extracted at the NYSM locations and the DL models are trained to recover the full field (training phase, cf. Fig 1). Next, missing data during inference, i.e., deep interpolation, is simulated following the cross-validation strategy explained in Section 4(c). The strategy provides dataset versions with $m = 124$ stations, $m = 93$ stations, or only $m = 62$ stations for inference. Evaluating all four trained models with all three station subsets yields a total of twelve experimental combinations, which are discussed below. The CV strategy enables us to evaluate the performance of all experiments on the full 124 stations, as we always make predictions for the left-out folds, making results comparable between experiments. Barnes interpolation is again employed as the baseline. As performance is naturally expected to drop when fewer observations are available, we compare DL-interpolated fields and Barnes-interpolated fields in the same situations.

For example, when only $m = 93$ stations are selected for inference for the DL models, the same stations are used for Barnes interpolation. That way, we can compare the advantage of our models over Barnes interpolation fairly. For an absolute picture, we also compare all our missing data experiments against the best-performing Barnes-interpolated fields, utilizing all 124 stations, to demonstrate the overall advantage of our framework.

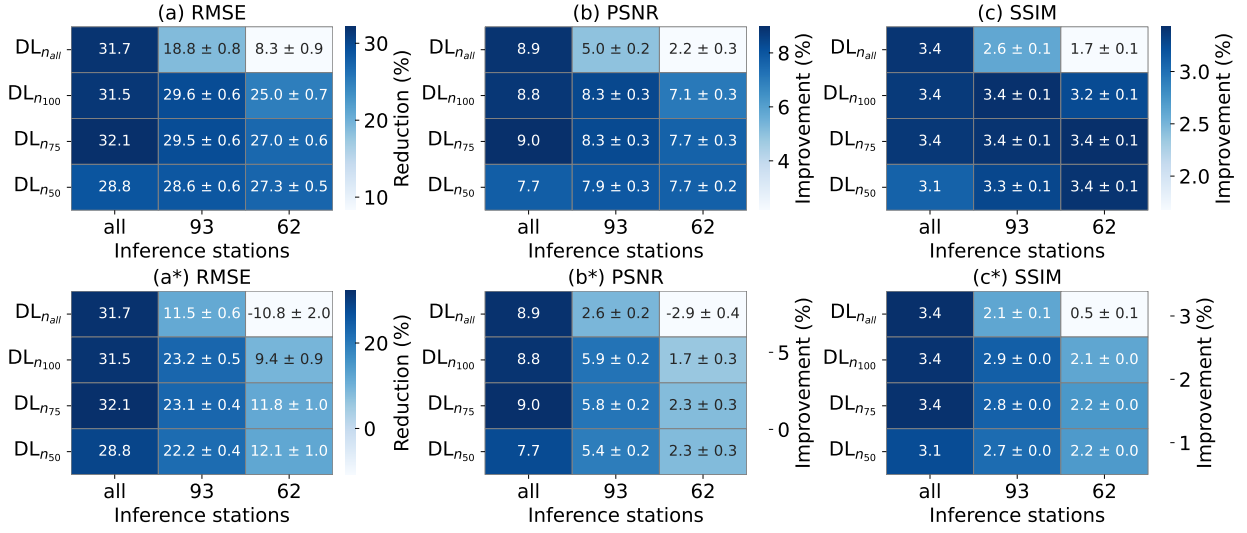


FIG. 8: Performance of deep learning models relative to Barnes interpolation across different numbers of training stations (rows) and inference stations (columns), simulating missing data in the observational network. Colors show improvement in (a) RMSE, (b) PSNR, and (c) SSIM metrics. Scores in the first row are normalized column-wise, i.e., with scores achieved by Barnes interpolation with the same number of missing stations. Second row scores (a* – c*) are all relative to the best-performing Barnes interpolated fields obtained when using observations from all stations ($m = 124$). The rows of each matrix present how the performance changes when one specific trained model (e.g., $\mathrm{DL}_{n_{all}}$ trained on all stations) is presented with different fractions of missing data during inference. Similarly, the columns illustrate how randomly dropping different ratios of stations already during training affects the performance when a fixed number of stations (e.g., 93) is available during inference.

The results are presented in Fig. 8. Qualitatively, the patterns of changing performance in Fig. 8 are the same across all panels. Unsurprisingly, the best extrapolation performance is achieved when all stations are available during inference ("all" columns). However, except for the $\mathrm{DL}_{n_{all}}$, the relative performance (first row) barely changes when 93 or 62 stations are used during inference. These station numbers correspond to 25% and 50% of missing data, which are significant amounts. Even in absolute terms (second row), our framework retains a significant advantage over the all-station Barnes interpolation results in these situations, as deep interpolation with 50% of missing values still outperforms all-station Barnes interpolation by more than 10%.

Only the $DL_{n_{all}}$ experiments show a significant drop in performance for an increasing number of missing stations. This behavior is attributed to a lack of generalization of the trained model. If all station locations are used during training, i.e., no randomization happens, the model always sees the same polygon pattern, only with different cell values. When fewer stations are available during inference, this pattern changes, which the model cannot handle well. In contrast, the models trained on random station subsets exhibited diverse polygon patterns during training and are consequently robust against changes in the number of stations during inference, emphasizing that training station randomization is crucial.

Intermediate result: Our framework maintains a consistent advantage over the Barnes interpolation, even when data is missing, *if* station locations are already randomized during training to make the model more robust. Based on the metrics presented in Fig. 8, the $n = 75$ model is selected for the rest of this work and is treated as the final S2G-DI configuration. The results also suggest that our framework can be used to fill gaps in observations across the network.

6. Results: Evaluation based on observational data (NYSM)

The S2G-DI model optimized in the sensitivity study is used here to assess the quality of analysis fields generated from real observations of the New York State Mesonet (NYSM). In this section, NYSM observations are provided as inputs at stage b* (Fig. 1) to generate the analysis fields. So far, all evaluations of the deep learning framework have been conducted using synthetic RTMA proxy data, which also served as input for analysis generation. By replacing these proxy inputs with real observations, this evaluation tests whether the model produces realistic analysis fields when applied to data that differ from the training distribution.

a. Performance in extreme wind gust conditions

Three example cases of extreme wind gust ($G_{10} > 17.5$ m/s) are selected from our held-out testing year 2023 to provide a comprehensive assessment across different types of extreme wind phenomena, from synoptic-scale frontal systems to mesoscale severe weather outbreaks. Focusing on extreme gusts allows us to evaluate our approach under challenging conditions. The events are listed in Tab. 4 and visualized in Fig. 9 as time series at two stations affected by the respective events. The curves presented for our approach (blue) are always taken from the out-of-fold

predictions following our CV strategy (cf. Section c). In other words, a subset of stations *not containing* the presented stations is used for the deep interpolation to obtain a gridded analysis field. The presented time series are extracted from these analysis fields at the locations of the respective stations (blue lines) and compared against the unseen true observations (black lines). The Barnes interpolation curves (red) are obtained following the same strategy, and RTMA data (cyan diamonds) are displayed for reference. The uncertainty intervals around the DL curves and Barnes curves stem from the ensemble prediction generated during cross-validation.

TABLE 4: Extreme wind gust events ($G10 > 17.5\,\mathrm{m\,s^{-1}}$) selected to assess the extreme prediction capability of the DL models.

#	Period	Synoptic conditions	Peak Wind Gust	Affected Regions
1	Feb 3–4, 2023	Powerful arctic cold front, leading to strong gusts and gradient winds.	$20–29\,\mathrm{m\,s^{-1}}$	North-West and South-East of NYS
2	Mar 25–26, 2023	Strong low-pressure system, producing widespread gusty winds before and after frontal passage (CNY Central 2023a,b).	$20–27\,\mathrm{m\,s^{-1}}$	Central NYS
3	Apr 1–2, 2023	Rare early-spring tornado outbreak, leading to severe thunderstorms, damaging wind gusts, and multiple confirmed tornadoes (National Weather Service Philadelphia/Mt. Holly 2023; New York Post 2023a,b).	$25–30\,\mathrm{m\,s^{-1}}$	Central and South-Eastern NYS

On first sight, our DL approach matches the observed gusts well in magnitude and timing, although very strong gust magnitudes are often underestimated. That is visible, for example, in panel (a) between 9:00 and 13:00 or panel (c) around 15:00 during the first day. It seems that our approach particularly struggles with very dynamic gusts with short peak gusts in a short time interval, while less dynamic, more sustained strong gust events, as depicted in panels (b) and (d), are captured better. Barnes interpolation, on the other hand, generally captures trends but exhibits more underestimation and less dynamic behaviour. In some cases, such as from 15:00 to 18:00 in panel (f), the Barnes interpolation completely misses the extreme gust. This performance advantage over Barnes interpolation is attributed to the auxiliary variables, such as M_{10}, T_2, and the terrain information, which our approach can utilize. However, extreme gust peaks are very rare events with less than 0.5% of yearly samples satisfying $G_{10} > 17.5$ m/s. That makes the extreme gust regression a very imbalanced regression problem, which is inherently challenging to handle (He and Garcia 2009). Machine learning models typically favor the core of the data

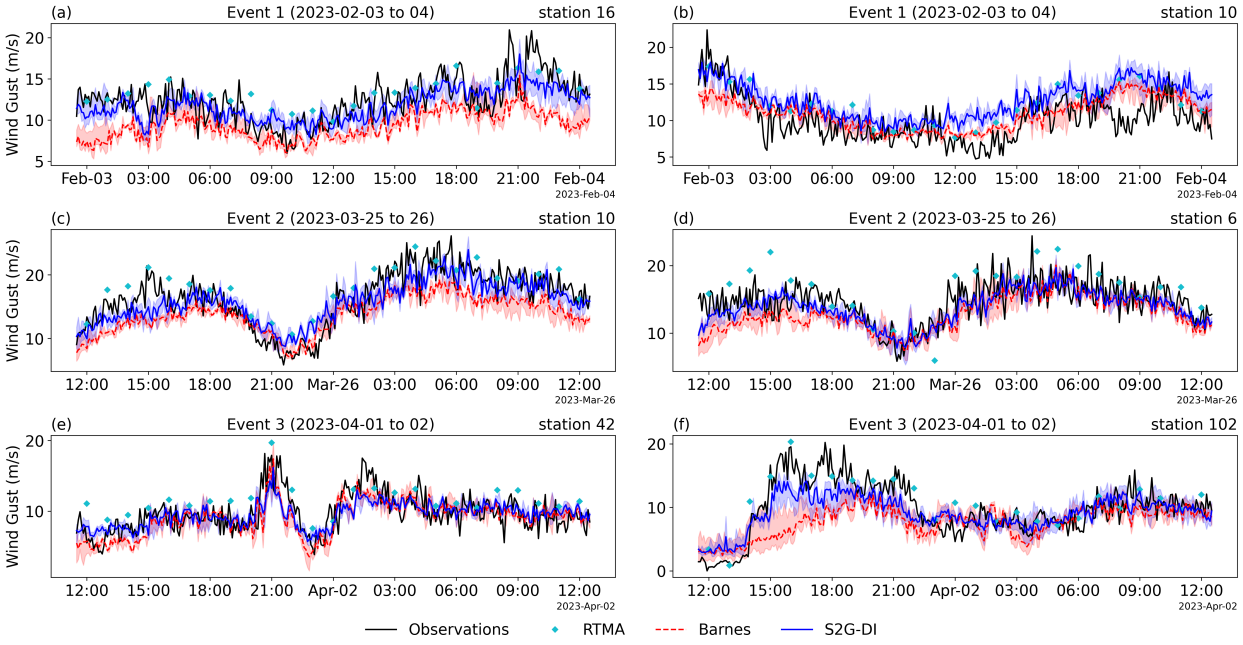


FIG. 9: Time series comparison of 10 m wind gusts during three extreme events (rows), with two representative (randomly selected) out-of-sample stations shown per event (columns). The curves for Barnes interpolation (red) and our S2G-DI (blue) represent ensemble medians, with shaded areas indicating the minima and maxima of the ensemble. RTMA (cyan diamonds) and observations (black) are shown for reference.

distribution, not its tails, as tails could be outliers, and because missing a few extreme values does not result in a significant penalty during optimization. While we recognize that further improvements in representing extreme gust magnitudes are needed, the current performance of the deep interpolation framework is encouraging, as it reliably captures the temporal evolution of gust events at unseen locations.

After discussing the interpolation performance at individual locations, Fig. 10 presents the spatial extents of the three events as captured by (a) our deep interpolation and by (b) traditional Barnes interpolation. Comparing deep interpolation and Barnes interpolation reveals that the Barnes interpolated fields suffer much more from missing stations (going from columns (i) to (iii)) than the S2G-DI approach. Particularly for events two and three, it is evident that the estimated area affected by wind gust (orange and red) increases drastically for Barnes interpolation when compared to deep interpolation. Our approach, however, retains a remarkably consistent spatial pattern compared to the full network (a, i), even when observations are available at only half of the stations (a, iii). We view this spatial consistency and the similarity of the deep interpolated analysis

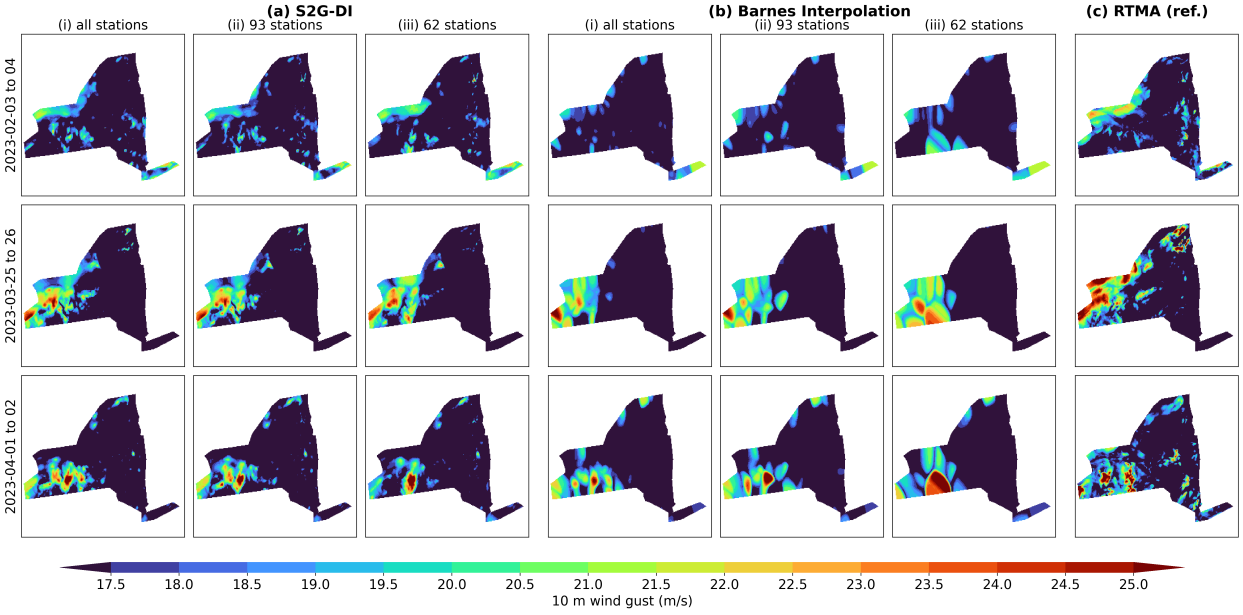


FIG. 10: Comparison of 10 m maximum wind gust analysis fields for three extreme events (rows) using (a) our deep interpolation framework and (b) traditional Barnes interpolation. The three columns per interpolation method (i – iii) present the change of the spatial pattern if observations are only available at a subset of stations for interpolation. In the absence of an absolute 2D ground truth, (c) RTMA is given for reference. The maps show the maximum wind gust per grid point for the entire event, thereby illustrating the aggregated geographical extent affected by the event. The colorbar is set to only show areas where gust exceeds the extreme threshold of 17.5 m/s.

fields to RTMA as a strong indication of the correctness and robustness of our method. The spatial findings are consistent with the underestimations discussed for the time series (cf. Fig. 9) and the missing-station experiments presented in the sensitivity study.

The spatial behavior of the analysis fields is explored in more detail in Fig. 11 using PSDs. The displayed curves are the time averages of the PSDs obtained for each snapshot of the interpolated analysis field from deep interpolation and Barnes interpolation, respectively. As such, the PSDs reflect the average detail with which an event is represented by either method. Again, RTMA is shown in black for reference. As observed during the sensitivity study, the observation-based analysis fields produced by Barnes interpolation also lose details for wavelengths of 40 km and smaller. The S2G-DI fields, on the other hand, retain more details, corroborating the advantage and value of our method.

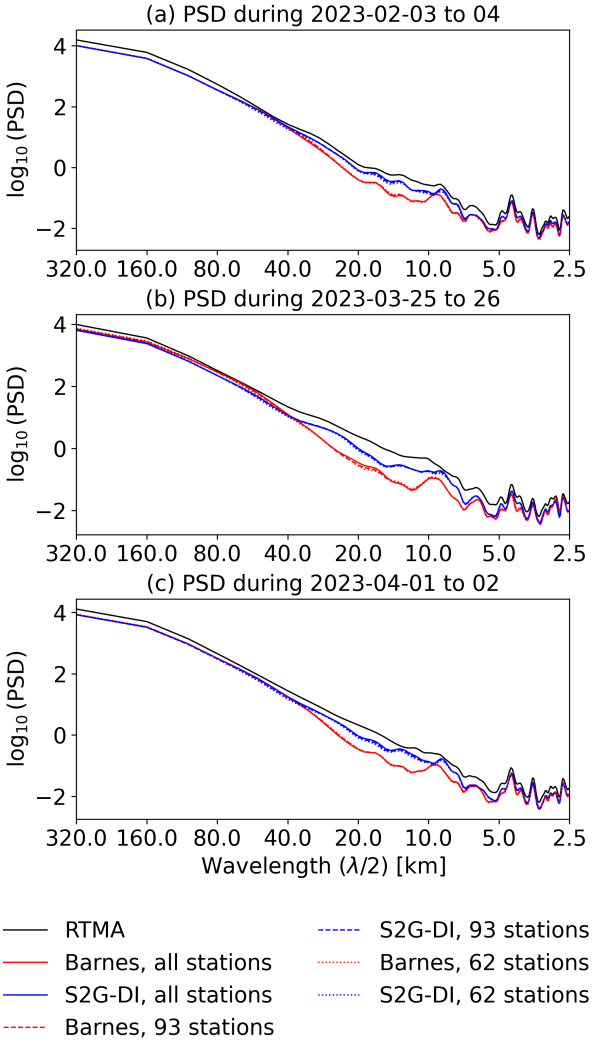


FIG. 11: Comparison of average power spectral density (PSD) for deep interpolated wind gust fields (blue) and fields from Barnes interpolation (red). Solid, dashed, and dotted lines indicate generation of analysis fields using all stations, 93 stations, and 62 stations, respectively. RTMA data is shown for reference (black), but cannot be viewed as absolute ground truth.

b. General performance across all wind gust conditions

While the selected extreme events provide detailed insight into the temporal and spatial behavior of the interpolation methods, it is important to assess whether these findings generalize across the full evaluation period. To this end, we aggregate the results over the entire year 2023 and evaluate the performance at held-out stations using complementary statistical metrics (Fig. 12).

The evaluation is performed at two temporal resolutions: native 5-minute observations (Fig. 12(a–b)) and hourly maximum gusts aggregated from the same data (Fig. 12(a*–b*)).

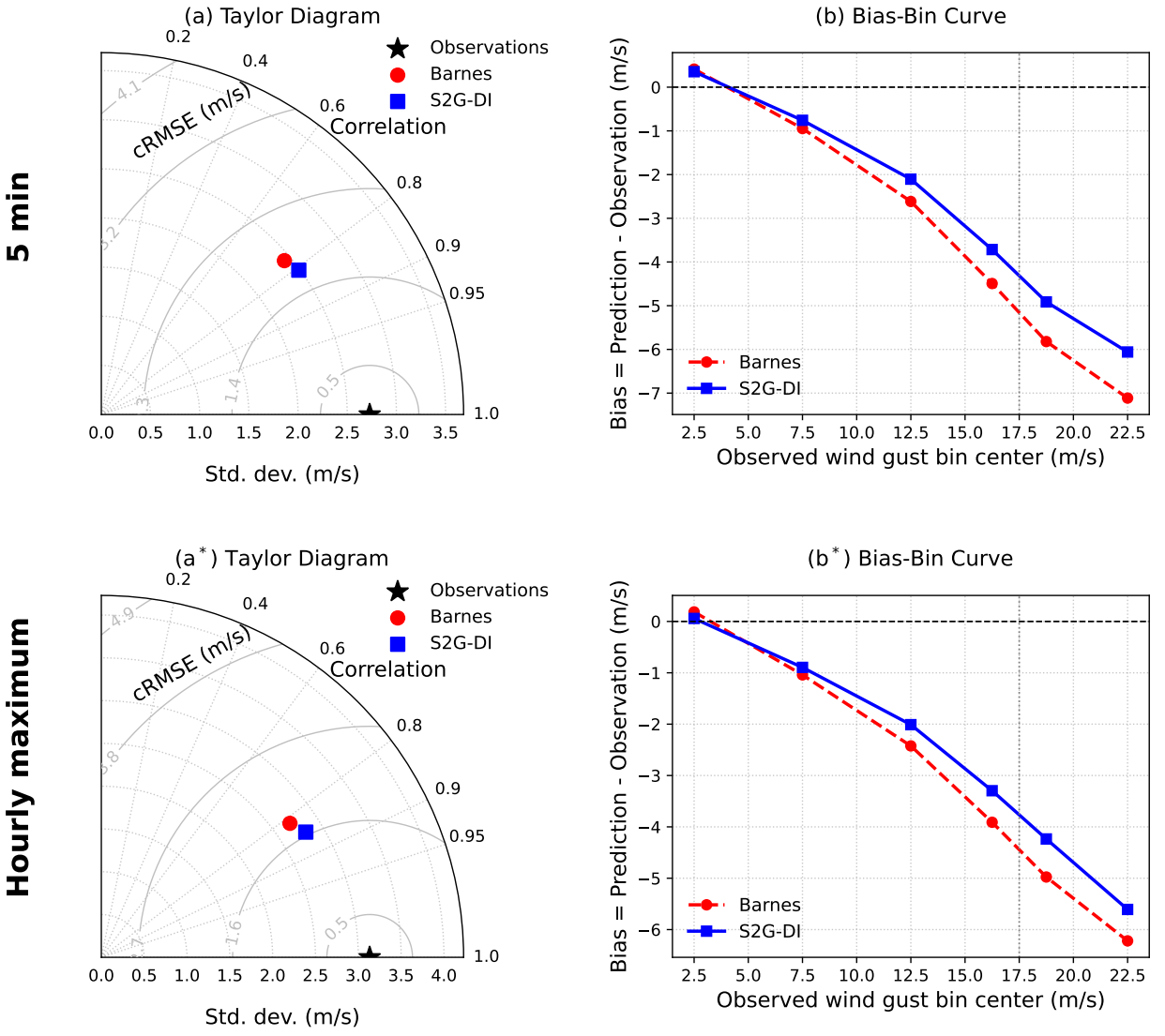


FIG. 12: Statistical evaluation of interpolation performance at held-out NYSM stations aggregated over the full year 2023 for two temporal resolutions. The top row shows results at native 5-minute resolution, while the bottom row shows results after aggregation to hourly maximum gusts. Panels (a, a*) display Taylor diagrams summarizing correlation, standard deviation, and centered RMSE, while (b, b*) show bias as a function of observed wind gust magnitude. All results are computed at held-out stations using out-of-fold predictions following the cross-validation strategy.

Considering the native 5-minute evaluation (Fig. 12(a–b)), the Taylor diagram summarizes the overall statistical performance, showing that S2G-DI achieves higher correlation with observations, a standard deviation closer to the observed variability, and a reduced centered RMSE compared to Barnes interpolation. The bias-bin analysis reveals systematic differences between the S2G-DI approach and the Barnes interpolation. While both approaches exhibit increasing underestimation

with higher wind gust magnitudes, S2G-DI consistently reduces the magnitude of this bias across all bins, particularly in the extreme range beyond 17.5 m/s. These results collectively demonstrate that the improvements of the deep interpolation framework are not limited to specific events but persist across the full temporal domain, reinforcing its robustness and generalization capability when applied to real observational data.

While both methods exhibit improved absolute performance at hourly resolution (Fig. 12(a*-b*)) due to the smoothing of short-lived gust variability, the relative performance between S2G-DI and Barnes interpolation remains consistent across all metrics. In particular, the Taylor diagram indicates unchanged ranking in terms of correlation and error characteristics and the bias-bin structure is preserved. This consistency demonstrates that the spatial relationships learned from hourly RTMA training data generalize well to sub-hourly inference, and that the performance gains of the deep interpolation framework are robust to the temporal mismatch between training and inference resolutions.

7. Results: Evaluation against independent HRRR analysis

To assess the dependence of the S2G-DI framework on the choice of reference dataset, we perform an additional experiment using the HRRR dataset. In this experiment, HRRR is used to as inference dataset, following the same experimental protocol as in the NYSM-based setup. Specifically, sparse samples are extracted from HRRR fields to mimic station observations, and both Barnes interpolation and deep interpolation are applied to reconstruct the full fields. The generated fields are then compared with source HRRR dataset. Because HRRR is not used during training, this setup provides an independent test of whether the relative performance of the methods generalizes across datasets.

Figure 13 shows the annual mean wind gust fields and corresponding bias distributions of Barnes interpolation and the S2G-DI model optimized relative to HRRR. Compared to the RTMA-based evaluation in Fig. 5, both methods again reproduce the large-scale spatial patterns of the reference field, indicating consistent behavior across datasets. However, as in the RTMA-based case, Barnes interpolation produces spatially smooth fields with limited variability, while the S2G-DI fields retain enhanced spatial structure that more closely resembles the variability present in the reference.

The bias maps further reveal systematic differences between the two approaches. Similar to the RTMA-based results, Barnes interpolation exhibits spatially coherent regions of over- and underestimation, particularly aligned with orographic features. In contrast, the S2G-DI reduces the magnitude of these structured biases and distributes residual errors more evenly across the domain. Notably, while the spatial patterns of bias differ between RTMA and HRRR—reflecting differences between the two reference products, the relative behavior of the methods remains consistent.

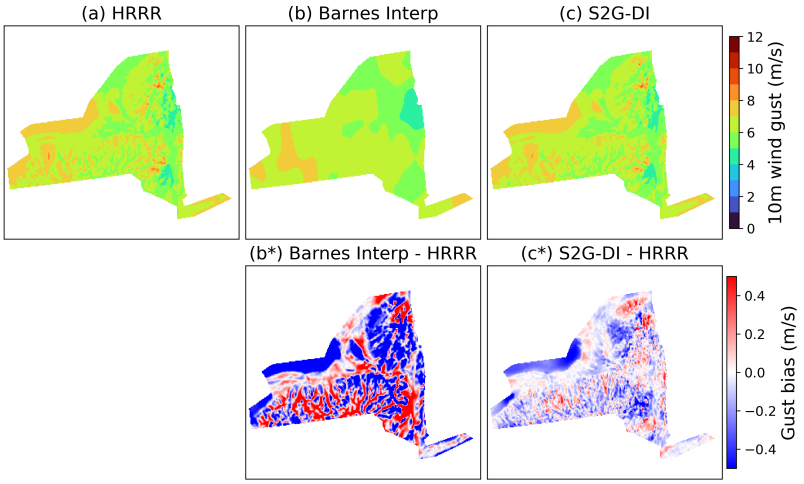


FIG. 13: Comparison of annual mean 10 m wind gust fields over New York State using (a) HRRR, (b) Barnes interpolation, and (c) S2G-DI. Panels (b*) and (c*) show the corresponding bias fields relative to HRRR. All fields are aggregated over the evaluation period 2023.

Table 5 quantifies these differences, showing that S2G-DI achieves substantial improvements over Barnes interpolation, including a 27.9% reduction in RMSE, a 7.3% increase in PSNR, and a 2.7% increase in SSIM. These improvements are comparable in magnitude to those observed in the RTMA-based evaluation, indicating that the performance gains of the deep interpolation framework persist when the entire evaluation is conducted on an independent high-resolution dataset.

TABLE 5: Relative performance of S2G-DI compared to Barnes interpolation using HRRR as reference.

RMSE reduction (%)	PSNR improvement (%)	SSIM improvement (%)
27.9	7.3	2.7

8. Results: Robustness across meteorological variables

Finally, four additional runs are conducted for target variables other than wind gust to assess the generalization capability of the best-performing DL configuration. More specifically, the UNet with all input variables, all training years, and $n = 75$ random training stations is used to generate analysis fields of M_{10}, T_2, and Q_2. The performance compared to Barnes interpolation, with respect to RMSE, PSNR, and SSIM, is presented in Tab. 6. The difference in spatial patterns (not shown here) between S2G-DI and Barnes is qualitatively similar to the wind gust case discussed previously: Barnes interpolation misses terrain details and fine-scale features, which our framework captures, leading to better performance. S2G-DI achieves the highest performance for the T_2 fields, possibly because temperature strongly depends on terrain slopes and elevation, which S2G-DI captures well compared to Barnes interpolation. Specific humidity, on the other hand, exhibits a very fine granular structure, which is captured less well by S2G-DI, resulting in only 25% improvement in RMSE over Barnes. The deep interpolation performance for mean wind M_{10} is a bit better than for wind gust G_{10}. This performance difference is attributed to M_{10} fields behaving qualitatively similar to G_{10} fields while exhibiting less complex patterns and dynamics, which we consider easier to predict. All in all, we conclude that our S2G-DI is well-suited to generate analysis fields of various variables.

TABLE 6: Deep interpolation performance of S2G-DI compared to traditional Barnes interpolation for 10 m wind speed (M_{10}), 2 m temperature (T_2), and 2 m specific humidity (Q_2). G_{10} scores are repeated for reference.

Variable	Method	RMSE (lower better)	Reduction (%)	PSNR (higher better)	Improvement (%)	SSIM (higher better)	Improvement (%)
M_{10} (m/s)	Barnes	1.269		31.362		0.893	
	S2G-DI	0.710	44.0%	35.818	14.2%	0.951	6.5%
T_2 (K)	Barnes	1.271		53.764		0.999	
	S2G-DI	0.548	56.9%	60.828	13.1%	1.000	0.1%
Q_2 (kg/kg)	Barnes	4e-4		42.090		0.985	
	S2G-DI	3e-4	29.8%	45.240	7.5%	0.993	0.8%
G_{10} (m/s)	Barnes	1.309		34.471		0.934	
	S2G-DI	0.888	32.1%	37.562	9.0%	0.966	3.4%

9. Conclusions and Future Scope

In this study, we presented the *Sparse-to-Gridded Deep Interpolation (S2G-DI)* framework, which aims to obtain high-quality gridded analysis fields of meteorological variables from sparse observations coming from a network of weather stations. The primary focus of our study was on 10 m wind gusts. Wind gusts are rapid changes in wind speed or short-duration maxima, typically defined using the three-second average wind speed maxima (World Meteorological Organization (WMO) 2023). These are associated with diverse high-impact weather phenomena, including tornadoes, thunderstorms, cyclones, and other high-impact weather systems (Sheridan 2018). Gusts can severely damage infrastructure, energy systems, transportation, and fire management operations. They may topple transmission lines, damage wind and solar farms, and bring down trees or branches onto power infrastructure (Mitchell 2013). Such failures can spark wildfires, which strong gusts can then spread rapidly over large areas (Mass and Ovens 2021). Wind gusts also pose hazards to aviation (Gultepe et al. 2019). While cyclones can sustain damaging winds for extended periods, convective systems often produce the most extreme short-duration gusts, sometimes exceeding 25 $\mathrm{m\,s^{-1}}$ (El Rafei et al. 2023). Capturing these short-lived and spatially localized events, therefore, remains a major challenge for both observation networks and modeling systems. We also demonstrated good performance for 2 m temperature, 2 m specific humidity, and 10 m wind speed.

Our S2G-DI is a two-step framework: first, sparse observations from weather networks, such as the New York State Mesonet (Brotzge et al. 2020), are nearest-neighbor-interpolated to gridded but low-fidelity 2D fields. Next, these low-fidelity inputs are passed through a deep learning (DL) model, which produces refined analysis fields by considering interactions between meteorological variables and accounting for terrain effects. The DL model for refinement is obtained through a decoupled precursor step, where it is trained on a gridded high-resolution proxy dataset covering the area of the observational network. We make the core assumption that the local meteorology is captured sufficiently well in the proxy dataset, such that later inference with a different dataset, i.e., the refinement of the observed data, produces realistic spatial fields. This assumption was confirmed throughout the study, with the help of HRRR analysis acting as an independent evaluation dataset. Our deep interpolation is also shown to be robust against missing data from a subset of weather stations, which is common due to maintenance or instrument malfunctions.

Decoupling the training of the DL model on proxy data from the deep interpolation (inference) on observations means the model never uses both datasets at the same time. Therefore, the gridded training data and observations do not require a temporal overlap and can have different temporal resolutions. For example, the DL model can be trained on archive data with a 1 h time step (common for gridded numerical data), while later S2G-DI can use real-time observations at the common 5-10 min time steps. Additionally, the common spatiotemporal misalignment between numerical and observational datasets is irrelevant, as only observations are used during operational deep interpolation. We believe these distinct features are highly valuable for the community.

In a thorough sensitivity study, we found that a medium-complexity UNet performs best compared to a lower-complexity deep convolutional neural network or a high-complexity transformer-based UNet. The quality of the analysis fields also increased when auxiliary variables, such as temperature and wind speed, were added as inputs. We attribute this to meteorological interactions between inputs, resulting in a performance increase in predicting the analysis target, such as wind gust. To demonstrate how powerful our deep interpolation is, we benchmarked it against the commonly used traditional Barnes interpolation (Barnes 1964). In comparison with Barnes interpolation, our framework not only achieved higher scores but also yielded more spatially detailed analysis fields because it accounts for terrain.

As gridded proxy data are crucial for training the DL model, we assessed how performance depends on the available amount of training data. We found that even just one year of gridded data yields a model that significantly outperforms Barnes interpolation and that adding more data has a positive but slowly diminishing effect. This result stresses that our framework is also applicable for regions where large gridded datasets are not readily available. While generating at least one year of training data using numerical weather prediction models is still a notable computational effort, we view it as generally feasible for the community. As previously mentioned, the decoupling of training and inference/deep interpolation enables training of the DL model on gridded data from archives, provided that the distributions of the gridded data match those of the observations.

Similar to the performance dependency on the temporal amount of training data, it is crucial to check the quality of the generated analysis fields when observations are only available at a subset of stations. Missing data are common in real-world networks due to sensor malfunctions or unavailability during maintenance. From a technical perspective, the S2G-DI framework inherently

allows deep interpolation from varying numbers of stations, as the nearest-neighbor interpolation (first step) can always generate low-fidelity gridded fields as input for the DL model. As part of the sensitivity study, we demonstrated that S2G-DI retains a significant performance advantage over Barnes interpolation at varying degrees of missing stations. More specifically, we outperformed Barnes interpolation in the same missing data situations, but also when Barnes interpolation had access to *all* observations, while S2G-DI faced missing data. We found that the crucial step to achieve this level of robustness is to randomly drop stations already during training. Dropping stations at training time was found to be crucial, as it ensures that the DL model sees a variety of network configurations and learns to robustly generate accurate results. Overall, we believe that the high performance of deep interpolation even under missing data makes our framework practical and attractive for operational use.

To assess how our optimized S2G-DI model handles challenging situations and real-world observations, we analyzed three observed extreme wind gust events that occurred in New York State in 2023. Extreme gusts are defined as gust magnitudes exceeding 17.5 m/s. Evaluating the time series at held-out stations showed that our model outperformed the traditional Barnes interpolation in many cases. We attribute this to the fact that the deep learning model can capture spatial meteorological patterns and variable interactions in contrast to the recursive weighted interpolation used by Barnes.

However, while our S2G-DI framework captures the timings and general trends of the events well, it misses the magnitude of the extreme gust peaks in many cases. As less than 0.5% of observed yearly gust samples at each station have gust magnitudes larger than our threshold, the extreme gust estimation is a highly imbalanced regression problem. Imbalanced data pose a common challenge in machine learning (He and Garcia 2009; Branco et al. 2016), as standard models and training procedures tend to favor the center of data distributions rather than the tails where extreme events are located. One strategy is to adjust the modeling procedure to account for the data imbalance, for example, through over- or undersampling techniques, by assigning higher weights to rarer samples, or by specifically adjusting the model architecture (Branco et al. 2016). But the imbalance issue can also be addressed explicitly from a physics perspective. In future work, we aim to incorporate weather radar data, such as from the Multi-Radar/Multi-Sensor system (Smith et al. 2016). Radar data would provide crucial information about thunderstorms, which are directly

associated with the convective processes that generate extreme convective wind gusts (Sheridan 2018). In other words, strong radar echoes are expected to appear only during extreme gust events, which form the tail of the gust distribution. Consequently, we expect to enhance the model's ability to capture the distribution tails and, thus, its ability to capture extreme gust events, by utilizing this complementary dataset.

Overall, our S2G-DI was shown to outperform the traditional Barnes interpolation in different tests for wind gust, wind magnitude, 2 m temperature, and 2 m specific humidity. Further, the setup of the S2G-DI framework is generic, so we expect it can be trained with alternative mesoscale products and geographical regions. For example, preliminary results Martinez-Roig (2025) showed the successful application of S2G-DI to interpolate observed 10m wind speed in Spain based on the New European Wind Atlas Dörenkämper et al. (2020). We, therefore, do not see any fundamental limitations that would prevent the application of our approach to other regions, datasets, or observations.

The framework possesses several practical features, including decoupling between training and interpolation, efficiency with respect to the volume of gridded training data required, and the capability to robustly handle missing observations while maintaining high performance. We believe that our approach to obtaining analysis fields is a valuable addition for the community, as many countries and states operate networks of weather stations. In the future, we plan to expand this work to capture extreme events more reliably, and we view the S2G-DI framework as a first step toward purely observation-driven forecasting, which we also actively research.

Acknowledgments. The authors acknowledge support from the Weather Innovation and Smart Energy and Resilience (WISER) center, funded by the U.S. National Science Foundation Industry–University Cooperative Research Centers (IUCRC) program. We thank the New York State Mesonet (NYSM) for providing the observational data used in this research. Original funding for the NYSM buildup was provided by Federal Emergency Management Agency grant FEMA-4085-DR-NY. The continued operation and maintenance of the NYSM is supported by National Mesonet Program, University at Albany, Federal and private grants, and others. The authors are grateful to Daniel Kirk-Davidoff and Evan Duffey for useful feedback on an earlier version of this work.

Data availability statement. The RTMA dataset is publicly available on Amazon Web Services (AWS) and can be accessed using the AWS CLI: `s3://noaa-urma-pds/`. The HRRR hourly analysis is publicly available on AWS and can be accessed using the AWS CLI: `s3://noaa-hrrr-bdp-pds/`. The NYSM observations are hosted at `https://nysmesonet.org` and can be requested through their data request page at `https://nysmesonet.org/weather/requestdata`. The deep learning model training and inference scripts are available in the GitLab project **S2G-DI** at `https://gitlab.com/HarishBaki/S2G-DI.git`, which can be freely accessed.

APPENDIX A

Additional Background on Deep Learning Methods

In recent years, data-driven approaches, encompassing both classical machine learning (ML) and deep learning (DL), have gained increasing popularity as a promising alternative for generating high-resolution gridded meteorological fields from a combination of observations and model simulations (Franco et al. 2020; Boomgard-Zagrodnik and Brown 2022; Chen et al. 2024; Jeong et al. 2025; Dujardin and Lehning 2022). Depending on the input–output structure, these approaches can be categorized into *point-to-point* learning, *image-to-point* learning, and *image-to-image* learning frameworks.

In a *point-to-point* learning framework, each input is expressed as a feature vector of station-level attributes such as spatial coordinates, distances between stations, or other predictors. This setup allows interpolation at any location and has been applied to several meteorological variables. For temperature and precipitation, studies have used a variety of methods, including artificial neural net-

works in the Virtual Weather Station framework (Franco et al. 2020), random forests with mesonet data (Boomgard-Zagrodnik and Brown 2022), and more recent deep neural network–based kriging approaches (Chen et al. 2024). Hybrid designs have also appeared, for example, combining sparse and dense station networks before training LSTM models to reconstruct daily maximum/minimum temperatures and precipitation (Jeong et al. 2025). Similar ideas have been explored for wind gusts: early work used ANNs with station variables (Sallis et al. 2011), followed by the use of reanalysis predictors (Coburn and Pryor 2022), and later decision trees, ensemble methods, and neural network–Gaussian process combinations (Kartal et al. 2023; Jahan et al. 2024; Zanetta et al. 2025). While these approaches work well for filling gaps at specific stations, they are less effective at representing spatial correlations in areas with complex terrain.

In an *image-to-point* learning framework, each input sample is represented by a gridded image containing inherent coordinate information, while predictions are made for a single grid point. For example, Dujardin and Lehning (2022) used a convolutional neural network (CNN) to predict u and v wind components at target locations, given COSMO1 model outputs within a 19×19 pixel window and high-resolution topography from a digital elevation model (DEM), demonstrating the added value of topography inputs for capturing spatial correlations. Their approach, however, focused on downscaling model-simulated fields rather than mapping station observations to grids. While effective for representing local spatial correlations, the limited spatial context restricts the framework's ability to capture mesoscale phenomena such as cyclones, fronts, and thunderstorms.

In contrast, the *image-to-image* learning framework treats both inputs and targets as structured images, enabling explicit modeling of spatial correlations and interactions between neighboring locations, thus producing spatially consistent interpolations. Fukami et al. (2021) demonstrated this approach by downsampling a two-dimensional (2D) target field to sensor locations, projecting it onto a structured grid via Voronoi tessellation, and training a CNN to map the low-quality field to the original. Applications included wake flow reconstruction, sea surface temperature (SST) estimation, and turbulent channel flow modeling. Subsequent studies extended this concept: Zhao et al. (2024) replaced the CNN with a Fourier neural operator, Wang et al. (2024) used convolutional LSTMs for joint SST reconstruction and forecasting, and Sunderhaft et al. (2024) applied it to Antarctic surface temperatures from reanalysis. Similarly, Güemes et al. (2022) replaced Voronoi tessellation with cubic binning and employed a GAN for SST reconstruction.

These works highlight the suitability of DL-based *image-to-image* learning for mapping sparse observations to gridded fields and its potential for real-time use, as shown by Achermann et al. (2024) in volumetric wind speed prediction over complex terrain (1.5×1.5 km^2) from sparse uncrewed aerial vehicle observations.

Building on the *image-to-image* framework proposed by Fukami et al. (2021), which projects sparse observations onto a structured grid before refining them with a deep learning model, we adapt and extend this approach for the real-time reconstruction of high-resolution gridded wind gust fields. While prior applications have focused on variables such as sea surface temperature (Güemes et al. 2022; Wang et al. 2024) and wind speed over small complex-terrain domains (Achermann et al. 2024), they have not addressed the unique challenges of generating high-quality gridded wind gust analyses across a large, topographically complex region such as New York State. Our S2G-DI framework bridges this gap through a decoupled training–inference design, enabling real-time gridded wind gust field generation from sparse mesonet observations at a 5 min temporal resolution, while remaining adaptable to other meteorological variables, observational networks, and geographical regions.

APPENDIX B

Deep Learning Models

a. DCNN

The DCNN architecture, illustrated in Fig. B1, is relatively simple and takes the same input and target tensors as the UNet model. It comprises seven sequential layers, each consisting of a 2D convolutional layer with a 7×7 filter and 48 feature channels, followed by a `GeLU` activation. A final 2D convolutional layer with a 3×3 filter and a single output channel, without any activation function, is applied to produce the final output. The architecture has been adopted from the studies of Fukami et al. (2021) and Sunderhaft et al. (2024), with little modifications made. Since our objective has been to predict at locations other than the stations, the values at the station locations in the predictions are replaced with those of the input tensor. This is to make sure that the values at the station locations will always be the same as what we provide as inputs. The entire architecture contains approximately 0.69 million trainable parameters.

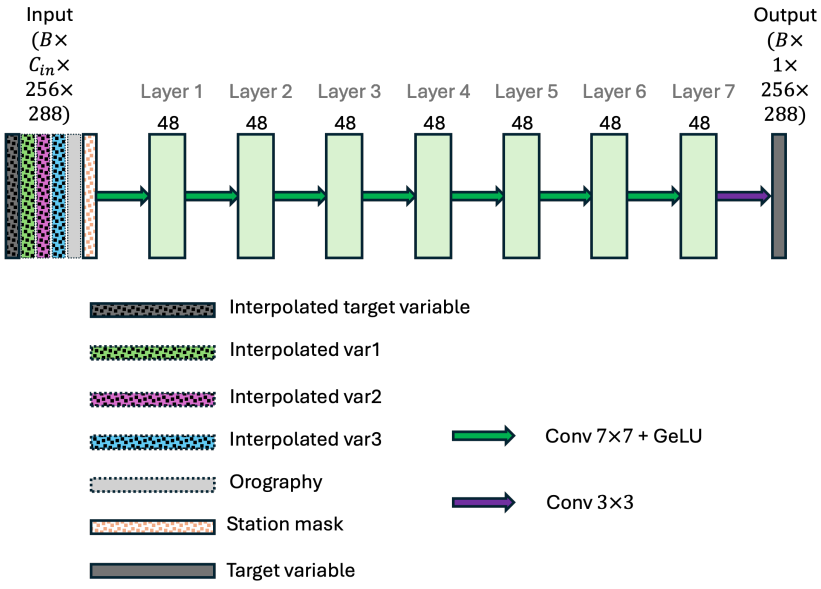


FIG. B1: An illustration of the Deep convolutional Neural Network (DCNN) model architecture.

b. UNet

The U-Net architecture is shown in Fig. B2, has been adopted from Wang et al. (2022) with little modifications. It consists of five hierarchical layers and follows an encoder–decoder structure. The encoder progressively extracts features by reducing the spatial resolution and increasing the feature dimensionality through a series of `ConvBlocks` and `DownBlocks`. In contrast, the decoder reconstructs the full-resolution output by gradually increasing the spatial dimensions and reducing the feature depth via a series of `UpBlocks` and `ConvBlocks`.

The core building block of the network is the `ConvBlock`, which serves as the backbone of the architecture. Each `ConvBlock` consists of a 2D convolutional layer with a 3×3 kernel, followed by a GeLU activation, another 3×3 convolution, and a second GeLU activation. To mitigate overfitting, a dropout layer with a rate of 0.2 and a drop-path layer with a rate of 0.2 are included as regularization components. A residual skip connection is used, which is added after a final convolutional layer of 1×1 filter. This residual skip connection facilitates improved gradient flow during training and helps alleviate vanishing gradient problems, following the approach introduced in the original Residual Network (ResNet) architecture (He et al. 2015).

The `DownBlock` reduces the spatial resolution by half while preserving the feature dimensionality. It consists of a 2D convolutional layer with a 4×4 kernel and a stride of 2, using the same number of

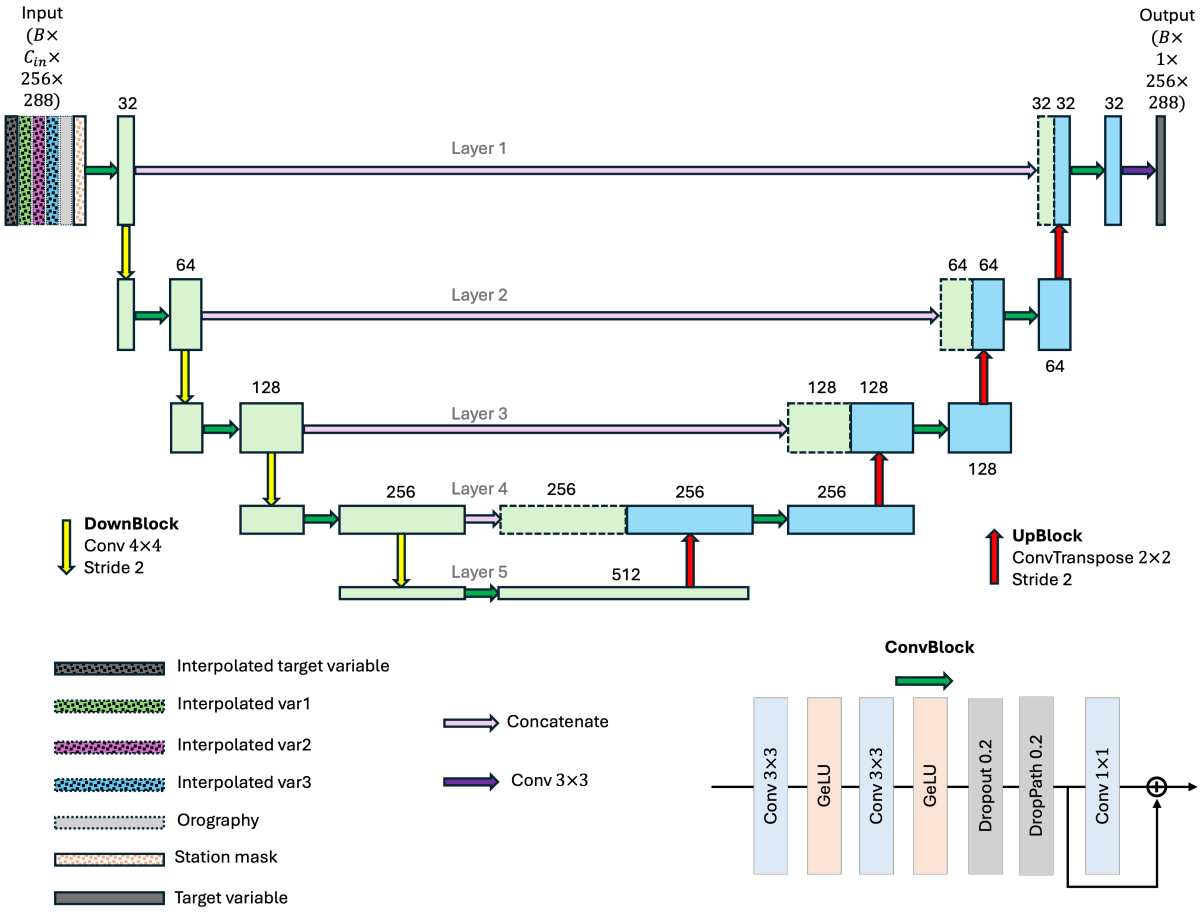


FIG. B2: An illustration of the UNet model architecture employed in this study.

input and output channels. This approach retains more contextual features compared to commonly used pooling techniques (He et al. 2015). Conversely, the `UpBlock` increases the spatial resolution by a factor of two while reducing the feature dimension by half. It is implemented using a 2D transposed convolutional layer with a 2×2 kernel and a stride of 2, where the number of output channels is half the number of input channels. While the `DownBlock` operates solely on the spatial dimensions and maintains feature dimensionality, the `UpBlock` simultaneously modifies both spatial and feature dimensions.

In the first layer, the input is passed through a `ConvBlock` with 32 filters, producing an output with the same spatial dimensions as the input but with 32 channels. The output of this layer is retained for a future skip connection. Simultaneously, it is passed through a `DownBlock`, which reduces the spatial resolution by half. This sequence of applying a `ConvBlock` followed by a

`DownBlock` is repeated up to layer 5, with the number of filters doubled at each successive layer. By layer 5, the feature map reaches a channel depth of 512, while the spatial resolution is reduced to 16×18. Together, these operations constitute the encoder portion of the U-Net architecture.

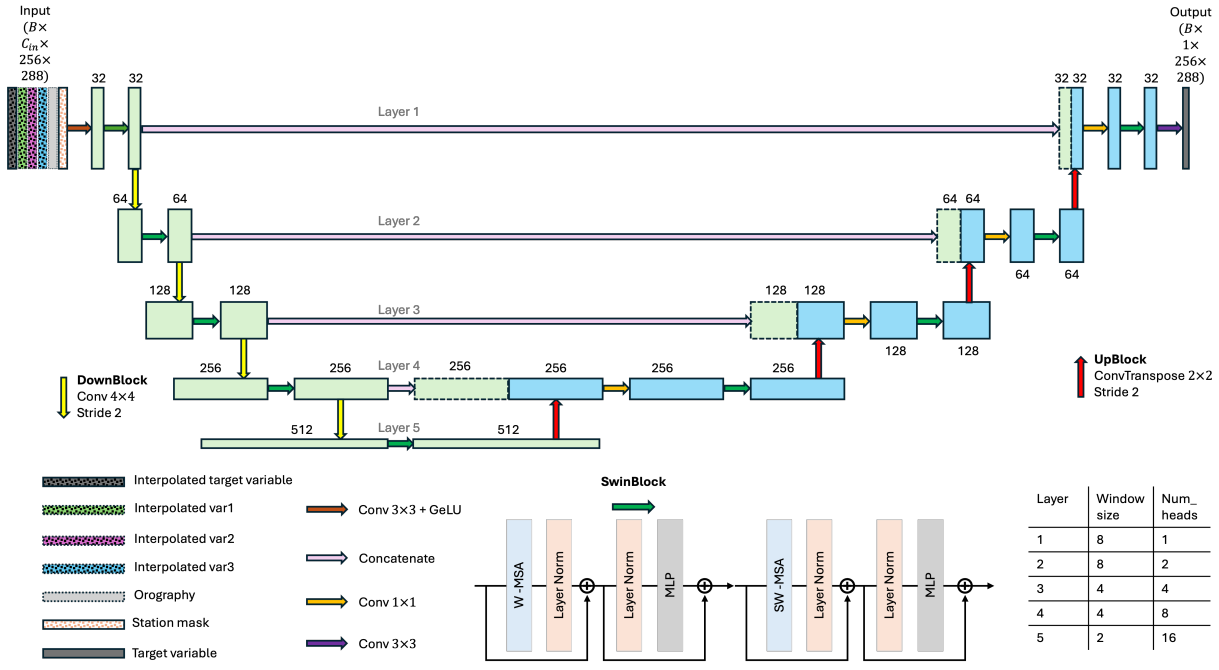
The output of the encoder portion is passed through an `UpBlock`, which increases the spatial resolution by a factor of 2 and reduces the feature dimension by half. The upsampled output is then concatenated along the feature dimension with the corresponding `ConvBlock` output from layer 4 of the encoder (via a skip connection), resulting in a feature map with doubled channel depth at that layer. The skip connection helps in extracting high-level localized contextual information along with mitigating vanishing gradients in deeper models. This concatenated output is subsequently passed through a `ConvBlock`, which reduces the feature dimensionality by half (e.g., to 256 channels). This sequence of applying an `UpBlock`, concatenation, and `ConvBlock` is repeated up to layer 1, with the number of features halving at each successive layer. By the time the decoder reaches layer 1, the feature map has a channel depth of 32, and the spatial resolution is restored to the original input dimensions of 256×288. Together, these series of operations constitute the decoder portion of the U-Net.

Finally, a 2D convolutional layer with a 3×3 kernel, a single output feature, and no activation function is applied to produce the final output, which consists of one channel corresponding to the target variable. Same as the DCNN, the values at the station locations are replaced from the input tensor. This UNet architecture consists of around 9.5 million parameters in total.

c. SwinT2UNet

Among all the models, the `SwinT2UNet` (shown in Fig. B3) is the most complex architecture, has been adopted from Wang et al. (2022) with considerable modifications. It follows the same encoder–decoder structure as the UNet; however, the backbone is replaced with a `SwinBlock`, along with several additional architectural modifications.

The `SwinBlock` is, in fact, the `SwinTransformerV2Block` from , which sequentially applies a non-shifted window and a shifted-window mechanism. Each block consists of a window-based multi-headed self-attention (W-MSA) module, followed by layer normalization, a residual connection, another layer normalization, a multilayer perceptron (MLP), and a second residual connection. This is followed by a shifted-window multi-headed self-attention (SW-MSA) module,



Layer	Window size	Num_ heads
1	8	1
2	8	2
3	4	4
4	4	8
5	2	16

FIG. B3: An illustration of the UNet with Sifted window multiheaded self-attention transformer block (Swintransformer) as backbone (SwinT2UNet) model architecture.

again followed by layer normalization, a residual connection, another layer normalization, an MLP, and a final residual connection.

Unlike in the UNet architecture, the `DownBlock` reduces the spatial resolution by half while doubling the feature dimensionality. It consists of a 2D convolutional layer with a 4×4 kernel and a stride of 2, where the number of output channels is twice that of the input channels. On the other hand, the `UpBlock` is identical to that of the UNet, it increases the spatial resolution by a factor of two while reducing the feature dimension by half.

In the first layer, the input is passed through an input projection block, which consists of a 2D convolutional layer with a 3×3 kernel and 32 feature channels, followed by a `GeLU` activation. This block transforms the input into the required number of feature representations for subsequent operations. The output from the input projection block is then passed through a `SwinBlock` with a window size of 8 and one attention head, which preserves the spatial and feature dimensions while extracting key hierarchical representations. The output from this layer is retained for a future skip connection. Simultaneously, it is passed through a `DownBlock`, which reduces the spatial resolution by half and doubles the number of feature channels. Similar to the UNet encoder, this

sequence consisting of a SwinBlock followed by a DownBlock is repeated up to layer 5, with the number of filters doubling at each successive layer. The specific window sizes and number of attention heads used in each SwinBlock are summarized in the table shown in Fig. B3. By layer 5, the spatial resolution is reduced to 16×18, and the feature dimensionality increases to 512.

Similar to the UNet, the output of the encoder portion is passed through an UpBlock and followed by a concatenation of SwinBlock output from encoder layer 4. The resulting output having a feature dimension of 512 goes through a 2D convolutional layer with 1×1 filter, to reduced the features by half, which is needed since the following SwinBlock takes and gives same number of (256) features. This sequence of applying UpBlock, concatenation, feature reduction through 1×1 convolution, and SwinBlock is repeated up to layer 1, with the number of features halving at each successive layer. By the time the decoder reaches layer 1, the feature map has a channel depth of 32, and the spatial resolution is restored to the original input dimensions of 256×288. Together, these series of operations constitute the decoder portion of the SwinT2UNet.

Similar to the UNet, the output of the encoder portion is passed through an UpBlock, followed by a concatenation with the corresponding SwinBlock output from encoder layer 4. The resulting feature map, with a dimensionality of 512, is then passed through a 2D convolutional layer with a 1×1 kernel to reduce the number of features by half. This step is necessary because the subsequent SwinBlock expects and produces 256 features. This sequence of operations, i.e., UpBlock, concatenation, feature reduction via 1×1 convolution, and SwinBlock, is repeated up to layer 1, with the number of features halving at each successive layer. By the time the decoder reaches layer 1, the feature map has a channel depth of 32, and the spatial resolution is restored to the original input dimensions of 256×288. Together, these operations constitute the decoder portion of the SwinT2UNet architecture.

Finally, a 2D convolutional layer with a 3×3 kernel, a single output channel, and no activation function is applied to the output of the decoder to produce the final output, which consists of one channel corresponding to the target variable. Here as well, the values at the station locations are replaced from the input tensor. This SwinT2UNet architecture consists of around 14.24 million parameters in total.

APPENDIX C

Barnes interpolation

Barnes Objective Analysis is a widely used two-pass interpolation method that applies Gaussian distance weighting to observations surrounding each grid point (Barnes 1964; Koch et al. 1983). In the first pass, the interpolated value at a grid point (i, j) is obtained by a weighted average of the observations $f(x_m, y_m)$, where each observation is weighted based on its distance r_m from the grid point. This is expressed as:

$$g_0(i,j) = \frac{\sum_{m=1}^{M} w_m f(x_m, y_m)}{\sum_{m=1}^{M} w_m}, \tag{C1}$$

where the weight is given by $w_m = \exp\left(-\frac{r_m^2}{\kappa}\right)$. Here, κ is a smoothing parameter with units of length squared, defined as $\kappa = \kappa^* \left(\frac{2\Delta n}{\pi}\right)^2$, where κ^* is a dimensionless smoothing coefficient and Δn is a representative station spacing.

The second pass refines the field by correcting for the local bias between the observed values and the first-pass estimates. This correction is expressed as:

$$g_1(i,j) = g_0(i,j) + \frac{\sum_{m=1}^{M} w'_m \left[f(x_m, y_m) - g_0(x_m, y_m)\right]}{\sum_{m=1}^{M} w'_m}, \tag{C2}$$

where the corrected weights are defined as $w'_m = \exp\left(-\frac{r_m^2}{\gamma\kappa}\right)$, and γ is a convergence (or damping) parameter, typically set between 0.2 and 1.0, to sharpen the influence of nearby residuals in the correction pass.

We adopted the Barnes interpolation implementation provided by MetPy (available at: `https://unidata.github.io/MetPy/latest/api/generated/metpy.interpolate.interpolate_to_points.html`), which accepts the convergence parameter (γ) and the dimensionless smoothing parameter (κ^*) as input arguments. The default values of γ and κ^* are 0.25 and 5.052, respectively. However, the values of these two parameters have been shown to influence the

interpolation accuracy considerably (Sun and Crook 2001). Thus, it is of paramount importance to first identify the optimal values of these parameters. For the purpose, we sampled (γ, κ^*) in the ranges of [0.05,0.4] and [3,7] for γ and κ^*, respectively, using Sobol sampling design. Using the sampled parameter values in the Barnes interpolation, we interpolated G_{10} for the period of 2023, and computed RMSE, PSNR, and SSIM scores, in comparison to the RTMA data. From the scores, we found a new optimal value for the parameters as $\gamma = 0.264$ and $\kappa^* = 6.566$, as shown in Fig. C1.

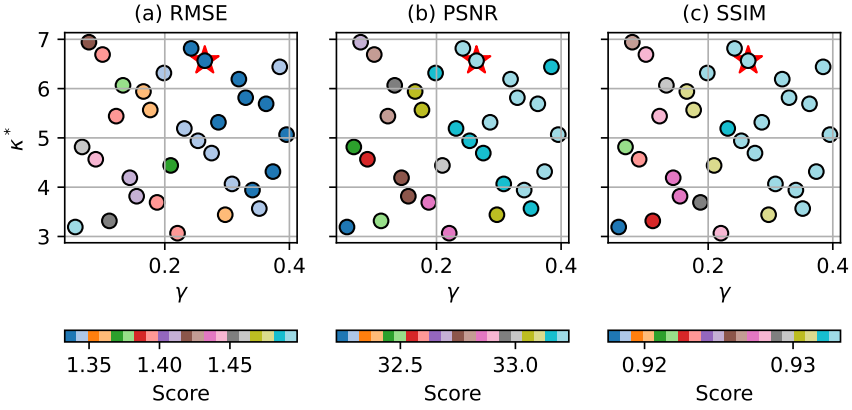


FIG. C1: Performance evaluation of Barnes interpolation accuracy across different values of the smoothing parameter κ^* and convergence parameter γ, sampled using a Sobol sequence. The subplots show the variation of (a) RMSE, (b) PSNR, and (c) SSIM scores between interpolated G_{10} and RTMA data for the year 2023. The red star in each panel denotes the location of the optimal parameter combination $(\gamma = 0.264176, \kappa^* = 6.566287)$.

References

Achermann, F., T. Stastny, B. Danciu, A. Kolobov, J. J. Chung, R. Siegwart, and N. Lawrance, 2024: WindSeer: real-time volumetric wind prediction over complex terrain aboard a small uncrewed aerial vehicle. *Nat. Commun.*, **15 (1)**, 3507.

Amponsah, W., P. A. Ayral, B. Boudevillain, C. Bouvier, I. Braud, P. Brunet, and Borga, 2018: Integrated high-resolution dataset of high-intensity european and mediterranean flash floods. *Earth System Science Data*, **10 (4)**, 1783–1794.

An, S., T.-J. Oh, E. Sohn, and D. Kim, 2025: Deep learning for precipitation nowcasting: A survey from the perspective of time series forecasting. *Expert Syst. Appl.*, **268 (126301)**, 126 301.

Anagun, Y., S. Isik, and E. Seke, 2019: Srlibrary: Comparing different loss functions for super-resolution over various convolutional architectures. *Journal of Visual Communication and Image Representation*, **61**, 178–187.

Barker, D., and Coauthors, 2012: The weather research and forecasting model's community variational/ensemble data assimilation system: Wrfda. *Bulletin of the American Meteorological Society*, **93 (6)**, 831–843.

Barnes, S. L., 1964: A technique for maximizing details in numerical weather map analysis. *Journal of Applied Meteorology*, **3 (4)**, 396–409.

Blankenau, P. A., A. Kilic, and R. Allen, 2020: An evaluation of gridded weather data sets for the purpose of estimating reference evapotranspiration in the united states. *Agric. Water Manag.*, **242 (106376)**, 106 376.

Boomgard-Zagrodnik, J. P., and D. J. Brown, 2022: Machine learning imputation of missing mesonet temperature observations. *Comput. Electron. Agric.*, **192 (106580)**, 106 580.

Branco, P., L. Torgo, and R. P. Ribeiro, 2016: A Survey of Predictive Modeling on Imbalanced Domains. *ACM Comput. Surv.*, **49 (2)**, 31:1–31:50, https://doi.org/10.1145/2907070.

Brotzge, J. A., and Coauthors, 2020: A technical overview of the new york state mesonet standard network. *J. Atmos. Ocean. Technol.*, **37 (10)**, 1827–1845.

Cai, Y., H. He, and Z. He, 2024: Loss functions analysis of performance improvements in single-image super-resolution. *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 3138–3141.

Campbell Scientific, Inc., 2023: Mesonets. Accessed: 7 August 2025, https://www.campbellsci.com.br/mesonets.

Charbonnier, P., L. Blanc-Feraud, G. Aubert, and M. Barlaud, 2002: Two deterministic half-quadratic regularization algorithms for computed imaging. *Proceedings of 1st International Conference on Image Processing*, IEEE Comput. Soc. Press, Vol. 2, 168–172 vol.2.

Chen, K., E. Liu, M. Deng, X. Tan, J. Wang, Y. Shi, and Z. Wang, 2024: DKNN: deep kriging neural network for interpretable geospatial interpolation. *Geogr. Inf. Syst.*, **38 (8)**, 1486–1530.

Chen, M., W. Shi, P. Xie, V. B. S. Silva, V. E. Kousky, R. Wayne Higgins, and J. E. Janowiak, 2008: Assessing objective techniques for gauge-based analyses of global daily precipitation. *J. Geophys. Res.*, **113 (D4)**.

Chen, Y., J. Hall, W. D. Van, N. Andela, S. Hantson, L. Giglio, and Randerson J, 2023: Multi-decadal trends and variability in burned area from the 5th version of the global fire emissions database (GFED5). *Earth System Science Data Discussions*, **2023**, 1–52.

Christiansen, M. B., W. Koch, J. Horstmann, C. B. Hasager, and M. Nielsen, 2006: Wind resource assessment from c-band sar. *Remote Sensing of Environment*, **105 (1)**, 68–81.

CNY Central, 2023a: Gusty wind causes minor tree damage around area, no widespread power outages. Accessed: 2025-07-28, https://cnycentral.com/news/local/gusty-wind-causes-minor-tree-damage-around-area-no-widespread-power-outages.

CNY Central, 2023b: Strong wind gusts continue for cny tonight with rain changing to snow showers into sunday. Accessed: 2025-07-28, https://cnycentral.com/news/instagram/strong-wind-gusts-continue-for-cny-tonight-with-rain-changing-to-snow-showers-into-sunday.

Coburn, J., and S. C. Pryor, 2022: Do machine learning approaches offer skill improvement for short-term forecasting of wind gust occurrence and magnitude? *Weather Forecast.*, **37 (5)**, 525–543.

Colle, B. A., and D. R. Novak, 2010: The new york bight jet: Climatology and dynamical evolution. *Mon. Weather Rev.*, **138 (6)**, 2385–2404.

Cressman, G. P., 1959: An operational objective analysis system. *Mon. Weather Rev.*, **87 (10)**, 367–374.

De Pondeca, M. S., and Coauthors, 2011: The real-time mesoscale analysis at noaa's national centers for environmental prediction: current status and development. *Weather and Forecasting*, **26 (5)**, 593–612.

Dörenkämper, M., and Coauthors, 2020: The Making of the New European Wind Atlas – Part 2: Production and evaluation. *Geoscientific Model Development*, **13 (10)**, 5079–5102, https://doi.org/10.5194/gmd-13-5079-2020.

Dowell, D. C., and Coauthors, 2022: The high-resolution rapid refresh (hrrr): An hourly updating convection-allowing forecast model. part i: Motivation and system description. *Weather and Forecasting*, **37 (8)**, 1371–1395.

Dujardin, J., and M. Lehning, 2022: Wind-Topo: Downscaling near-surface wind fields to high-resolution topography in highly complex terrain with deep learning. *Q. J. R. Meteorol. Soc.*, **148 (744)**, 1368–1388.

El Rafei, M., S. Sherwood, J. Evans, and A. Dowdy, 2023: Analysis and characterisation of extreme wind gust hazards in new south wales, australia. *Nat. Hazards (Dordr.)*, **117 (1)**, 875–895.

Franco, B. M., L. Hernández-Callejo, and L. M. Navas-Gracia, 2020: Virtual weather stations for meteorological data estimations. *Neural Comput. Appl.*, **32 (16)**, 12 801–12 812.

Fukami, K., R. Maulik, N. Ramachandra, K. Fukagata, and K. Taira, 2021: Global field reconstruction from sparse sensors with voronoi tessellation-assisted deep learning. *Nat. Mach. Intell.*, **3 (11)**, 945–951.

Gandin, L. S., 1963: Objective analysis of meteorological fields. *Israel program for scientific translations*, **242**.

Gilleland, E., D. Ahijevych, B. G. Brown, B. Casati, and E. E. Ebert, 2009: Intercomparison of spatial forecast verification methods. *Weather and forecasting*, **24 (5)**, 1416–1430.

Glorot, X., and Y. Bengio, 2010: Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 249–256.

Gultepe, I., and Coauthors, 2019: A review of high impact weather for aviation meteorology. *Pure and applied geophysics*, **176 (5)**, 1869–1921.

Güemes, A., C. Sanmiguel Vila, and S. Discetti, 2022: Super-resolution generative adversarial networks of randomly-seeded fields. *Nat. Mach. Intell.*, **4 (12)**, 1165–1173.

Hardy, R. L., 1971: Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, **76 (8)**, 1905–1915.

Haylock, M. R., N. Hofstra, A. M. G. Klein Tank, E. J. Klok, P. D. Jones, and M. New, 2008: A european daily high-resolution gridded data set of surface temperature and precipitation for 1950–2006. *J. Geophys. Res.*, **113 (D20)**.

He, H., and E. A. Garcia, 2009: Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, **21 (9)**, 1263–1284, https://doi.org/10.1109/TKDE.2008.239.

He, K., X. Zhang, S. Ren, and J. Sun, 2015: Deep residual learning for image recognition. *arXiv [cs.CV]*.

Hersbach, H., and Coauthors, 2020: The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, **146 (730)**, 1999–2049.

Houston, A. L., N. A. Lock, J. Lahowetz, B. L. Barjenbruch, G. Limpert, and C. Oppermann, 2015: Thunderstorm observation by radar (thor): An algorithm to develop a climatology of thunderstorms. *Journal of Atmospheric and Oceanic Technology*, **32 (5)**, 961–981.

Hu, Y., G. Jia, H. Gao, Y. Li, M. Hou, J. Li, and C. Miao, 2023: Spatial characterization of global heat waves using satellite-based land surface temperature. *International Journal of Applied Earth Observation and Geoinformation*, **125**, 103 604.

Jahan, I., D. Cerrai, and M. Astitha, 2024: Storm gust prediction with the integration of machine learning algorithms and WRF model variables for the northeast united states. *Artificial Intelligence for the Earth Systems*, **3 (3)**.

Jeong, Y., D. Kim, and K. Byun, 2025: A novel deep learning-based approach for reconstruction of historical long-term high-quality gridded meteorological dataset. *J. Hydrol. (Amst.)*, **654 (132850)**, 132 850.

Kahl, J. D. W., 2020: Forecasting peak wind gusts using meteorologically stratified gust factors and MOS guidance. *Weather Forecast.*, **35 (3)**, 1129–1143.

Kalnay, E., 2003: *Atmospheric modeling, data assimilation and predictability*. Cambridge university press.

Kartal, S., S. Basu, and S. J. Watson, 2023: A decision-tree-based measure–correlate–predict approach for peak wind gust estimation from a global reanalysis dataset. *Wind Energy Sci.*, **8 (10)**, 1533–1551.

Knapp, K. R., and J. P. Kossin, 2007: New global tropical cyclone data set from isccp b1 geostationary satellite observations. *Journal of Applied Remote Sensing*, **1 (1)**, 013 505.

Koch, S. E., M. desJardins, and P. J. Kocin, 1983: An interactive barnes objective map analysis scheme for use with satellite and conventional data. *J. Clim. Appl. Meteorol.*, **22 (9)**, 1487–1503.

Kwon, D.-K., and A. Kareem, 2009: Gust-front factor: New framework for wind load effects on structures. *Journal of structural engineering*, **135 (6)**, 717–732.

Le Toumelin, L., I. Gouttevin, N. Helbig, C. Galiez, M. Roux, and F. Karbou, 2023: Emulating the adaptation of wind fields to complex terrain with deep learning. *Artif. Intell. Earth Syst.*, **2 (1)**, 1–39.

Liston, G. E., and K. Elder, 2006: A meteorological distribution system for high-resolution terrestrial modeling (MicroMet). *J. Hydrometeorol.*, **7 (2)**, 217–234.

Mahmood, R., and Coauthors, 2017: Mesonets: Mesoscale weather and climate observations for the united states. *Bull. Am. Meteorol. Soc.*, **98 (7)**, 1349–1361.

Mankin, K. R., S. Mehan, T. R. Green, and D. M. Barnard, 2025: Review of gridded climate products and their use in hydrological analyses reveals overlaps, gaps, and the need for a more objective approach to selecting model forcing datasets. *Hydrol. Earth Syst. Sci.*, **29 (1)**, 85–108.

Martinez-Roig, M., 2025: Comparative Evaluation of Deep Learning Architectures for Spatial Infilling of Meteorological Data. Geosphere Austria.

Mass, C. F., and D. Ovens, 2021: The synoptic and mesoscale evolution accompanying the 2018 camp fire of northern california. *Bull. Am. Meteorol. Soc.*, **102 (1)**, E168–E192.

Matheron, G., 1967: Kriging or polynomial interpolation procedures. *CIMM Transactions*, **70 (1)**, 240–244.

Mitchell, J. W., 2013: Power line failures and catastrophic wildfires under extreme weather conditions. *Eng. Fail. Anal.*, **35**, 726–735.

Müller, S. H., D. Cáceres, S. Eisner, M. Flörke, C. Herbert, C. Niemann, and Döll, 2021: The global water resources and use model WaterGAP v2. 2d: Model description and evaluation. *Geoscientific Model Development*, **14 (2)**, 1037–1079.

National Weather Service Philadelphia/Mt. Holly, 2023: Event review: April 1, 2023 tornado outbreak. Accessed: 2025-07-28, https://www.weather.gov/phi/eventreview2023401.

New York Post, 2023a: New york city facing threat of 60 mph winds, tornadoes. Accessed: 2025-07-28, https://nypost.com/2023/04/01/new-york-city-facing-threat-of-60-mph-winds-tornadoes/.

New York Post, 2023b: Northeast hit with tornadoes, damaging wind gusts. Accessed: 2025-07-28, https://nypost.com/2023/04/02/northeast-hit-with-tornadoes-damaging-wind-gusts/.

Niziol, T., 1987: Operational forecasting of lake effect snowfall in western and central new york. *Weather and Forecasting*, **2**, 310–321.

Niziol, T. A., W. R. Snyder, and J. S. Waldstreicher, 1995: Winter weather forecasting throughout the eastern united states. part IV: Lake effect snow. *Weather Forecast.*, **10 (1)**, 61–77.

Peng, J., S. Dadson, F. Hirpa, E. Dyer, T. Lees, D. G. Miralles, and Funk, 2020: A pan-african high-resolution drought index dataset. *Earth System Science Data*, **12 (1)**, 753–769.

Peraza, J., P. R. Rossini, and A. Patrignani, 2025: Mapping mesoscale soil moisture using a model-data fusion approach. *J. Hydrol. (Amst.)*, **654 (132768)**, 132 768.

Rasmussen, R., and Coauthors, 2023: Conus404: The ncar–usgs 4-km long-term regional hydroclimate reanalysis over the conus. *Bulletin of the American Meteorological Society*, **104 (8)**, E1382–E1408.

Ridal, M., and Coauthors, 2024: Cerra, the copernicus european regional reanalysis system. *Quarterly Journal of the Royal Meteorological Society*, **150 (763)**, 3385–3411.

Sallis, P. J., W. Claster, and S. Hernández, 2011: A machine-learning algorithm for wind gust prediction. *Comput. Geosci.*, **37 (9)**, 1337–1344.

Shepard, D., 1968: A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference*, 517–524.

Sheridan, P., 2018: Current gust forecasting techniques, developments and challenges. *Adv. Sci. Res.*, **15**, 159–172.

Skamarock, W. C., 2004: Evaluating mesoscale nwp models using kinetic energy spectra. *Monthly weather review*, **132 (12)**, 3019–3032.

Smith, T. M., and Coauthors, 2016: Multi-radar multi-sensor (mrms) severe weather and aviation products: Initial operating capabilities. *Bulletin of the American Meteorological Society*, **97 (9)**, 1617–1630.

Subedi, S., A. Kechchour, M. Kantar, V. Sharma, and B. C. Runck, 2025: Can gridded real-time weather data match direct ground observations for irrigation decision-support? *Agrosyst. Geosci. Environ.*, **8 (2)**.

Sun, J., and N. A. Crook, 2001: Real-time low-level wind and temperature analysis using single WSR-88D data. *Weather Forecast.*, **16 (1)**, 117–132.

Sunderhaft, R., L. Frank, and J. Davis, 2024: Deep learning improvements for sparse spatial field reconstruction. *arXiv [cs.CV]*.

Themeßl, M. J., A. Gobiet, and A. Leuprecht, 2011: Empirical-statistical downscaling and error correction of daily precipitation from regional climate models. *International Journal of Climatology*, **31 (10)**, 1530–1544.

Venter, Z. S., A. Hassani, E. Stange, P. Schneider, and N. Castell, 2024: Reassessing the role of urban green space in air pollution control. *Proc. Natl. Acad. Sci. U. S. A.*, **121 (6)**, e2306200 121.

Wang, H., H. Zhou, and S. Cheng, 2024: Dynamical system prediction from sparse observations using deep neural networks with voronoi tessellation and physics constraint. *Comput. Methods Appl. Mech. Eng.*, **432 (117339)**, 117 339.

Wang, Z., X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, 2022: Uformer: A general U-shaped transformer for image restoration. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE.

Wasula, A. C., L. Bosart, and K. D. LaPenta, 2002: The influence of terrain on the severe weather distribution across interior eastern new york and western new england. *Weather and Forecasting*, **17**, 1277–1289.

World Meteorological Organization (WMO), 2023: *Guide to Instruments and Methods of Observation*. World Meteorological Organization, Geneva, Switzerland, URL https://library.wmo.int/idurl/4/41650, wMO-No. 8.

Xiao, H., Y. Wang, Y. Zheng, Y. Zheng, X. Zhuang, H. Wang, and M. Gao, 2023: Convective-gust nowcasting based on radar reflectivity and a deep learning algorithm. *Geosci. Model Dev.*, **16 (12)**, 3611–3628.

Zanetta, F., D. Nerini, M. Buzzi, and H. Moss, 2025: Efficient modeling of sub-kilometer surface wind with gaussian processes and neural networks. *Artif. Intell. Earth Syst.*

Zhao, X., X. Chen, Z. Gong, W. Zhou, W. Yao, and Y. Zhang, 2024: RecFNO: A resolution-invariant flow and heat field reconstruction method from sparse observations via fourier neural operator. *Int. J. Therm. Sci.*, **195 (108619)**, 108 619.