DL-Corrector-Remapper: A grid-free bias-correction deep learning methodology for data-driven high-resolution global weather forecasting

Tao Ge[*]
Washington University in St. Louis
St. Louis, MO. 63130
getao@wustl.edu

Jaideep Pathak
NVIDIA
Santa Clara, CA. 95050
jpathak@nvidia.com

Akshay Subramaniam
NVIDIA
Santa Clara, CA. 95050
asubramaniam@nvidia.com

Karthik Kashinath
NVIDIA
Santa Clara, CA. 95050
kkashinath@nvidia.com

Abstract

Data-driven models, such as FourCastNet (FCN), have shown exemplary performance in high-resolution global weather forecasting. This performance, however, is based on supervision on mesh-gridded weather data without the utilization of raw climate observational data, the gold standard ground truth. In this work we develop a methodology to correct, remap, and fine-tune gridded uniform forecasts of FCN so it can be directly compared against observational ground truth, which is sparse and non-uniform in space and time. This is akin to bias-correction and post-processing of numerical weather prediction (NWP), a routine operation at meteorological and weather forecasting centers across the globe. The Adaptive Fourier Neural Operator (AFNO) architecture is used as the backbone to learn continuous representations of the atmosphere. The spatially and temporally non-uniform output is evaluated by the non-uniform discrete inverse Fourier transform (NUIDFT) given the output query locations. We call this network the Deep-Learning-Corrector-Remapper (DLCR). The improvement in DLCR's performance against the gold standard ground truth over the baseline's performance shows its potential to correct, remap, and fine-tune the mesh-gridded forecasts under the supervision of observations.

1 Introduction

Reliable weather forecasting models play a crucial role in preparing for the harsh consequences of climate change by providing early warnings of extreme weather events for disaster mitigation. Numerous deep learning (DL)-based weather prediction models have been developed to forecast global weather under the supervision of the reanalysis mesh-gridded data [1–5]. FCN [6, 7] has shown exemplary performance in high-resolution data-driven global weather forecasting measured by the similarity of its forecasts against mesh-gridded reanalysis data, ERA5 [8]. However, FCN's mesh-gridded forecasts could not be directly compared against the raw, sparse, non-uniform observations. Yet, the gold standard for weather and climate model performance is their ability to match observations. Further, because FCN is trained on reanalysis data, it is likely to have biases with respect to direct observations. To address these issues, we develop a DL-based *grid-free* model that remaps and

[*] Worked performed during an internship at NVIDIA

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2022.

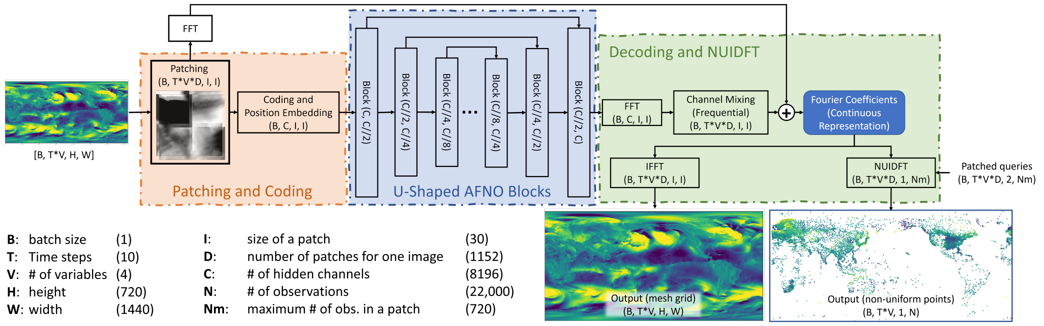

Figure 1: Flowchart of DL Corrector-Remapper (DLCR). DLCR consists of three parts: (i) image patching and coding; (ii) U-shaped AFNO blocks; and (iii) decoding and NUIDFT. The typical size of the output tensor is denoted under the name of each block.

corrects the mesh-gridded forecasts to arbitrary locations in space and time. This methodology is sufficiently general that it can be applied to *any* gridded forecast, not just from FCN.

Several data-driven approaches have been developed to refine mesh-gridded forecasts using sparse, non-uniform observational data [9–11]. However, our objective is different from existing work in the following distinct ways: (1) The desired model should accept arbitrary output query locations, *i.e.* it should be grid-free (existing works are only suitable for fixed locations); (2) Some observations may be missing randomly over space and time, so the desired model should handle random missing data seamlessly to fully utilize the available data at all locations and time steps; (3) The desired model should be capable of processing large spatio-temporal data from long forecasts, including all relevant variables, at high resolution, which can be highly memory-intensive.

To address these challenges, we develop a Fourier-based approach to learn continuous representations of the input data under the supervision of sparse, non-uniform observations. The model is grid-free since the continuous data can be reconstructed from the Fourier coefficients at any location. Moreover, because the network itself is independent of the output query, the model can naturally process missing data. To the best of our knowledge, this is the first time that a *grid-free* DL model has been developed to re-map and correct mesh-gridded high-resolution weather forecasts to sparse, non-uniform observations.

2 Methodology and Data

2.1 Type II Non-Uniform DFT

We use Fourier coefficients for continuous representations of spatio-temporal data. Therefore, the projection of the continuous representation to spatial positions is the real part of type II non-uniform inverse discrete Fourier transform (NUIDFT), which can be written in matrix form as:

$$\cos(2\pi Q M^T) F_{real} - \sin(2\pi Q M^T) F_{img}, \tag{1}$$

where $Q \in \mathbb{R}^{N\times 2}$ denotes the query matrix, $M \in \mathbb{R}^{WH\times 2}$ is the frequency basis, F_{real} and $F_{img} \in \mathbb{R}^{WH\times 1}$ denote the real and imaginary parts of the Fourier coefficients, respectively.

The sparse NUIDFT is developed to reduce the memory footprint by dividing the queries into different groups with respect to their actual length and truncating zero elements. For very high-resolution data, query locations can be randomly sub-sampled in each epoch to further reduce memory usage.

2.2 DL Corrector-Remapper and Baselines

We develop DL-Corrector-Remapper (DLCR), a Fourier-based network that incorporates NUIDFT into the backbone of Adaptive Fourier Neural Operators (AFNO) [12], to remap and bias-correct mesh-gridded forecasts to arbitrary locations in space and time, under the supervision of the gold standard – sparse, non-uniform observational data.

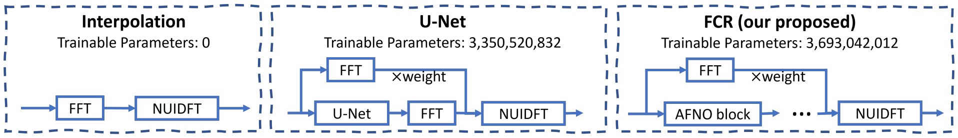

Figure 2: Model comparison of the interpolation, U-Net, and DLCR. These three methods utilize NUIDFT to reconstruct the spatio-temporal data from the continuous representation. The interpolation method has no trainable parameters, while U-Net and FCR both have ∼3 billion trainable parameters.

The overall structure of the proposed network is shown in Figure 1. Essentially, the output of the model is the Fourier coefficient, so one can use arbitrary queries to get arbitrary spatial outputs. Applying the inverse fast Fourier transform (IFFT) yields mesh-gridded output, while applying the NUIDFT with the query yields sparse observational output (see bottom right in Figure 1).

In the *patching process* , non-overlapping patches of the image are stacked in the channel dimension to form a token sequence [12]. Then, the channel coder compresses the channel size to reduce the memory footprint by accounting for the spatio-temporal and inter-variable dependencies. The *U-shaped AFNO blocks* are the backbone of the proposed model. The channel dimension is halved for each subsequent block k if $k < (K+1)//2$, and doubled if $k \geq (K+1)//2$, where K denotes the total number of blocks. This U-shaped structure reduces the memory footprint without sacrificing the depth of the network. Moreover, *skip connections* allow feature maps with more channels to flow in the network, so the narrowest block in the network does not reduce the complexity of the output.

The proposed network (DLCR) is compared to the NUIDFT-based interpolation and the widely used standard U-Net. Figure 2 shows a schematic of these models and the corresponding number of trainable parameters.

2.3 Loss function

To correct biases of mesh-gridded forecasts whilst preserving their spatio-temporal structure, shapes, and patterns, we develop a novel loss function with two terms: (i) a magnitude difference term; and (ii) a structural similarity term,

$$\underset{\theta}{\operatorname{argmin}} ||y_{sp} - \Gamma_\theta(x, q_{sp})||_2^2 + \lambda(1 - LCC(y_{gd}, \Gamma_\theta(x, q_{gd}))) \tag{2}$$

where y_{sp} denotes sparse, non-uniform observational data, q_{sp} is the corresponding sparse output query, y_{gd} denotes the reanalysis mesh-gridded data (ERA5), q_{gd} is the mesh-gridded query, Γ_θ denotes the trainable model, x denotes the input data, λ denotes a scalar that controls the weight of the structural similarity term, and LCC denotes the local cross-correlation [13] (see Appendix A).

2.4 Datasets and Training

The input dataset is FCN inference data from 2000 to 2018 at 0.25°resolution [6]. We use the global observational data acquired from National Centers for Environmental Information as the ground truth reference [14]. Four overlapping variables are selected: surface wind velocity components (U and V), surface temperature (t2m), and mean sea level pressure (MSLP). A data sample is shown in Figure 3. The input and reference data are normalized by the global mean and variance of each variable.

Each forecast is five-days-long, with a 6-hourly time step, generated across 19 years, which yields 27360 samples in total. We use 23040 samples from 2000 to 2015 to train the model and 1440 samples from the year 2017 for validation. Moreover, 2% of the observational sites are removed from the training set, so these positions are 'unobserved' or *hidden*. 'Unobserved' time is used to test the reliability of future forecasts, while 'unobserved' positions are used to test the model's ability to produce observation-quality forecasts for locations that it has never seen before. Since the reference data is extremely sparse, the output query is perturbed by Gaussian noise $\mathcal{N}(0, 0.04)$ to address spatial over-fitting.

The networks are trained on 64 A100 NVIDIA GPUs with 80 GB memory on the Selene supercomputer [15]. Each training epoch takes 3000 seconds. The optimizer is ADAM, and the start learning rate is 10^{-3}. We train the model for 10 epochs with a scheduler that reduces the learning rate by a factor of 0.7 every 3 epochs.

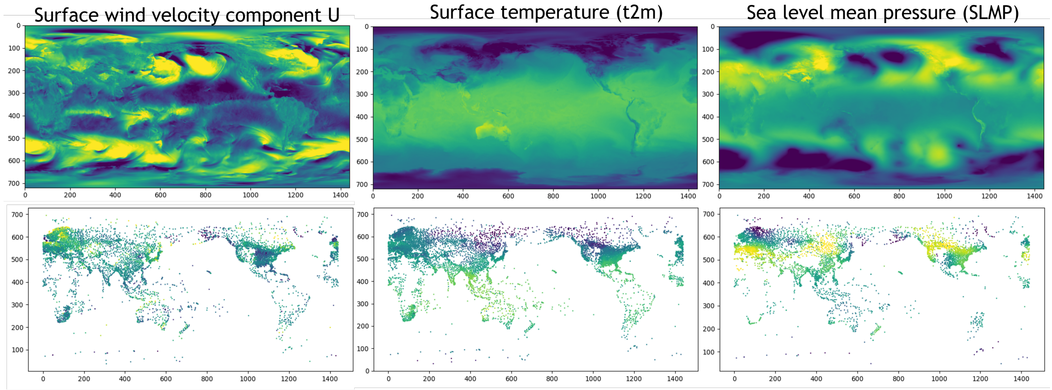
Surface wind velocity component U
Surface temperature (t2m)
Sea level mean pressure (SLMP)
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200
1400

Figure 3: The mesh-gridded input (top) and non-uniform reference (bottom) data. Wind velocity V is omitted in this figure. The reference data is extremely sparse and missing randomly.

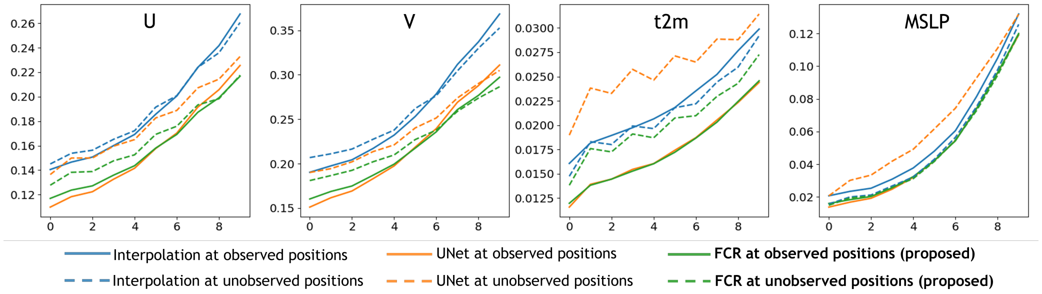

Figure 4: MSE of the output against the observation vs. prediction time step (lower is better). Our proposed model (DLCR) outperforms baselines across observed and unobserved locations.

3 Results

Figure 4 shows the plots of mean square error (MSE) that are averaged over 80 instances across the year 2017 (validation). The horizontal axis denotes the time step, where each step corresponds to 12 hours. For observed positions, the performance of DLCR is close to the performance of U-Net, and they both outperform the interpolation baseline. For unobserved positions, DLCR outperforms the interpolation baseline and U-Net, and it performs better on more complicated variables (U and V), whereas the performance of the U-Net is even worse than the performance of the interpolation in estimating t2m and MSLP. Besides, the difference between the solid and dashed blue curves may indicate a small systematic bias in the unobserved locations. (see appendix for additional information)

4 Conclusion

We propose a grid-free DL model (DLCR) to correct, remap, and fine-tune gridded uniform high-resolution global weather forecasts under the supervision of the gold standard ground truth – observations – which are sparse and non-uniform in space and time, can be noisy, and have gaps and missing data. DLCR outperforms baselines even when the output positions are unobserved. In this paper DLCR is developed for correcting and remapping FourCastNet's output, but the method itself can be applied to any forecast data. This work has implications for improving a broad range of data-driven and traditional NWP forecasting approaches by making their outputs comparable to and trainable on observational data. Importantly, DLCR allows arbitrary query locations and is trained to produce *observation-quality* forecasts in 'unobserved' locations, therefore it provides a way to improve the reliability of data-driven and NWP forecasts in locations on the globe and for time instances where observations do not exist.

References

[1] P. D. Dueben and P. Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009, 2018.

[2] Jonathan A. Weyn, Dale R. Durran, and Rich Caruana. Can Machines Learn to Predict Weather? Using Deep Learning to Predict Gridded 500-hPa Geopotential Height From Historical Weather Data. *Journal of Advances in Modeling Earth Systems*, 11(8):2680–2693, 2019.

[3] Jonathan A. Weyn, Dale R. Durran, and Rich Caruana. Improving Data-Driven Global Weather Prediction Using Deep Convolutional Neural Networks on a Cubed Sphere. *Journal of Advances in Modeling Earth Systems*, 12(9):e2020MS002109, 2020. e2020MS002109 10.1029/2020MS002109.

[4] Stephan Rasp, Peter D. Dueben, Sebastian Scher, Jonathan A. Weyn, Soukayna Mouatadid, and Nils Thuerey. WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

[5] Ryan Keisler. Forecasting Global Weather with Graph Neural Networks, 2022.

[6] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators, 2022.

[7] Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Animashree Anandkumar. FourCastNet: Accelerating Global High-Resolution Weather Forecasting using Adaptive Fourier Neural Operators, 2022.

[8] H. Hersbach, Berrisford Bell, B., Biavati P., Horányi G., J. A., Muñoz Sabater, J. Nicolas, Peubey, R. C., Radu, I. Rozum, D. Schepers, A. Simmons, C. Soci, D. Dee, and J-N Thépaut. ERA5 hourly data on single levels from 1959 to present. N Copernicus Climate Change Service (C3S) Climate Data Store (CDS). Accessed: 2022-07.

[9] Stephan Rasp and Sebastian Lerch. Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 146(11):3885 – 3900, 2018.

[10] Federico Amato, Fabian Guignard, Sylvain Robert, and Mikhail Kanevski. A novel framework for spatio-temporal prediction of environmental data using deep learning. *Scientific Reports*, 10(22243), 2020.

[11] Hui Zhang, Yaqiang Wang, Dandan Chen, Dian Feng, Xiaoxiong You, and Weichen Wu. Temperature Forecasting Correction Based on Operational GRAPES-3km Model Using Machine Learning Methods. *Atmosphere*, 13(2), 2022.

[12] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive Fourier Neural Operators: Efficient Token Mixers for Transformers. *CoRR*, abs/2111.13587, 2021.

[13] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Med Image Anal.*, 12(1):26–41, 2008.

[14] NOAA National Centers for Environmental Information (2001): Global Surface Hourly. NOAA National Centers for Environmental Information: https://www.ncei.noaa.gov/metadata/geoportal/rest/metadata/item/gov.noaa.ncdc:C00532/html. Accessed: 2022-07.

[15] NVIDIA Selene: Leadership-Class Supercomputing Infrastructure. https://www.nvidia.com/en-us/on-demand/session/supercomputing2020-sc2019/.

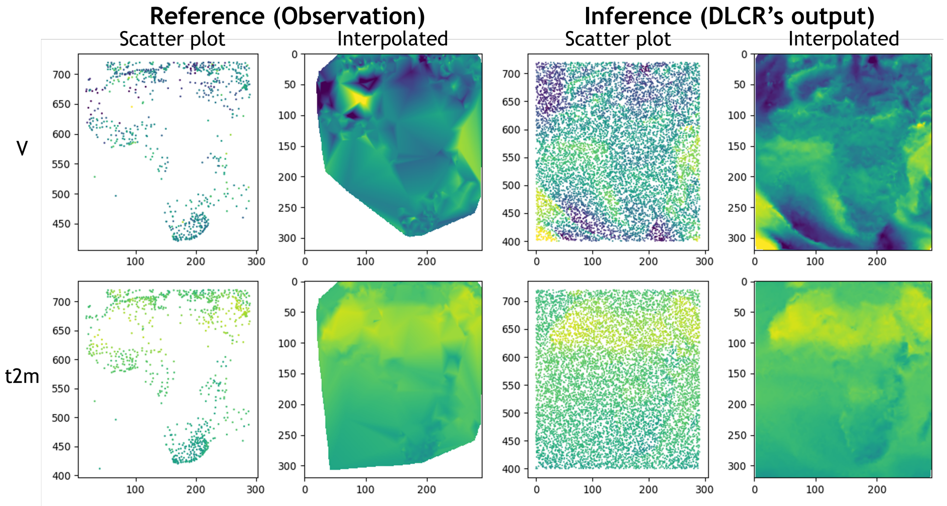

Figure 5: Figures of the reference and DLCR inference output for the region of Africa. The DLCR inference output is generated via a random query unseen by the model. It is clear that interpolation of DLCR's output, after querying DLCR densely over the African continent, has significantly better qualitative performance in regions of Africa with very sparse observations, whereas interpolation of the raw observations produces unphysical results in these regions. Thus the benefit of a grid-free model that can be queried at arbitrary locations is evident.

A Appendix: Local Cross-Correlation

Local cross-correlation (LCC) is a similarity metric that focuses on the similarity of patterns, structures, and high contrast boundaries between signals, written as

$$LCC(A,B) = \frac{\Big(\sum_i \sum_j \big(A[i,j]-\bar{A}[i,j]\big)\big(B[i,j]-\bar{B}[i,j]\big)\Big)^2}{\sum_i \sum_j \Big(A[i,j]-\bar{A}[i,j]\Big)^2 \sum_i \sum_j \Big(B[i,j]-\bar{B}[i,j]\Big)^2}, \tag{3}$$

where A and B denote two arbitrary 2-dimensional images, and $\bar{A}$ denotes the local mean that is calculated by the convolution of the image A and an averaging kernel V, *i.e.*,

$$\bar{A}[i,j] = \sum_n \sum_m A[i-n, j-m]V[n,m]. \tag{4}$$

In this work, V is chosen to be a 5×5 uniform matrix whose sum is normalized to 1. The size of the kernel controls the scale of the detectability. LCC with a large kernel only captures large structures and ignores subtle changes, while LCC with a small kernel captures subtle variations yet is more vulnerable to noise.

B Appendix: Additional Figures

Figure 5 shows the scatter plot and the interpolated mesh-gridded image of the reference observation data, as well as the scatter plot and the interpolated output inference with a random query (6000 locations). The display region covers -40°to 40°of latitude and -22.5°to 50°of longitude, which corresponds to the continent of Africa. These query positions are not necessary to be on the mesh grid or observed. The randomly queried output of the proposed method still looks realistic in the region with a limited number of observational sites.

Figure 6 shows the observation positions in this work, where hidden positions are labelled as orange triangles. The world map is evenly divided into 48 patches, and these hidden positions are randomly selected in each patch.

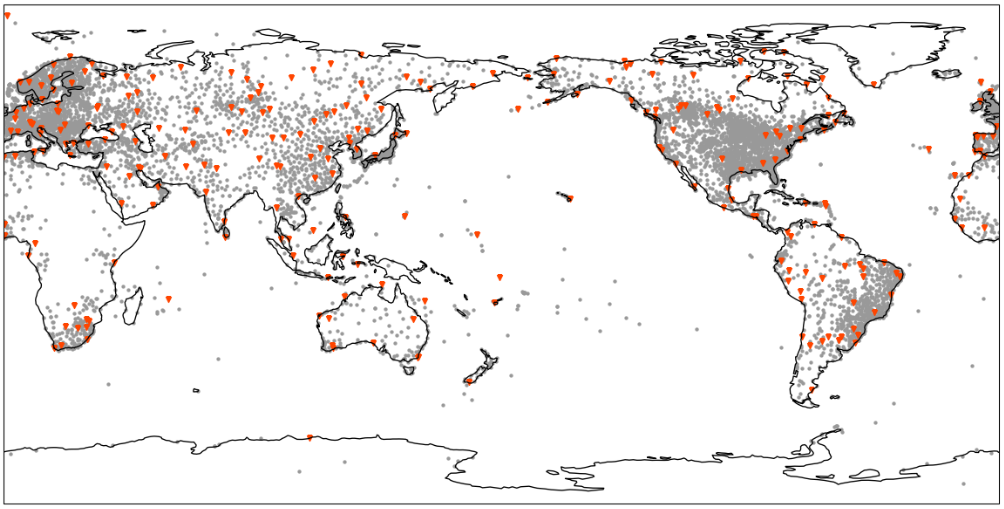

Figure 6: Figure showing the locations of the surface stations. Observed positions are indicated by grey dots, and unobserved positions are indicated by orange pins.