DL-Corrector-Remapper

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

Tao Ge[1,2], Jaideep Pathak[1], Akshay Subramaniam[1], Karthik Kashinath[1]

NeurIPS 2022 Workshop

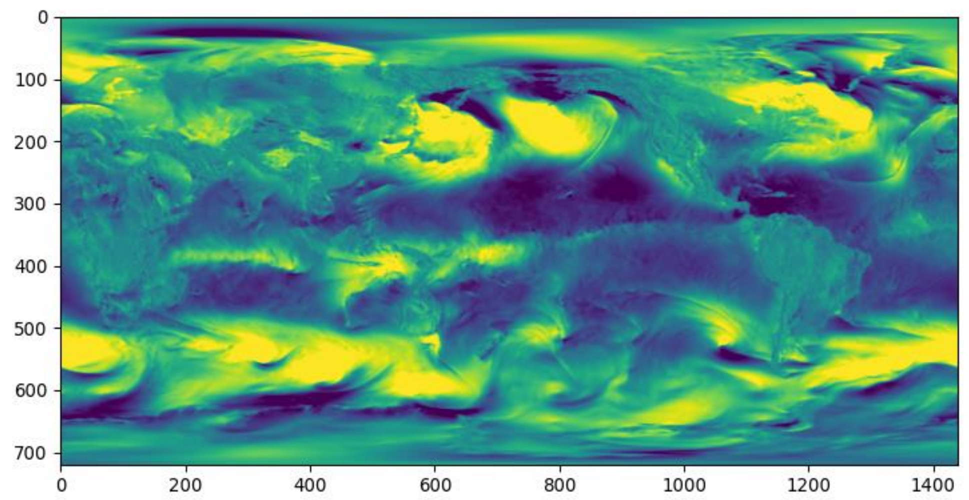
Tackling Climate Change with Machine Learning

1 NVIDIA

2 Washington University in St. Louis

Introduction

DL-based mesh-gridded forecast model



Mesh-gridded forecast: wind velocity

Deep-learning(DL)-based mesh-gridded forecast model

Under the supervision of the reanalysis mesh-gridded data

FourCastNet

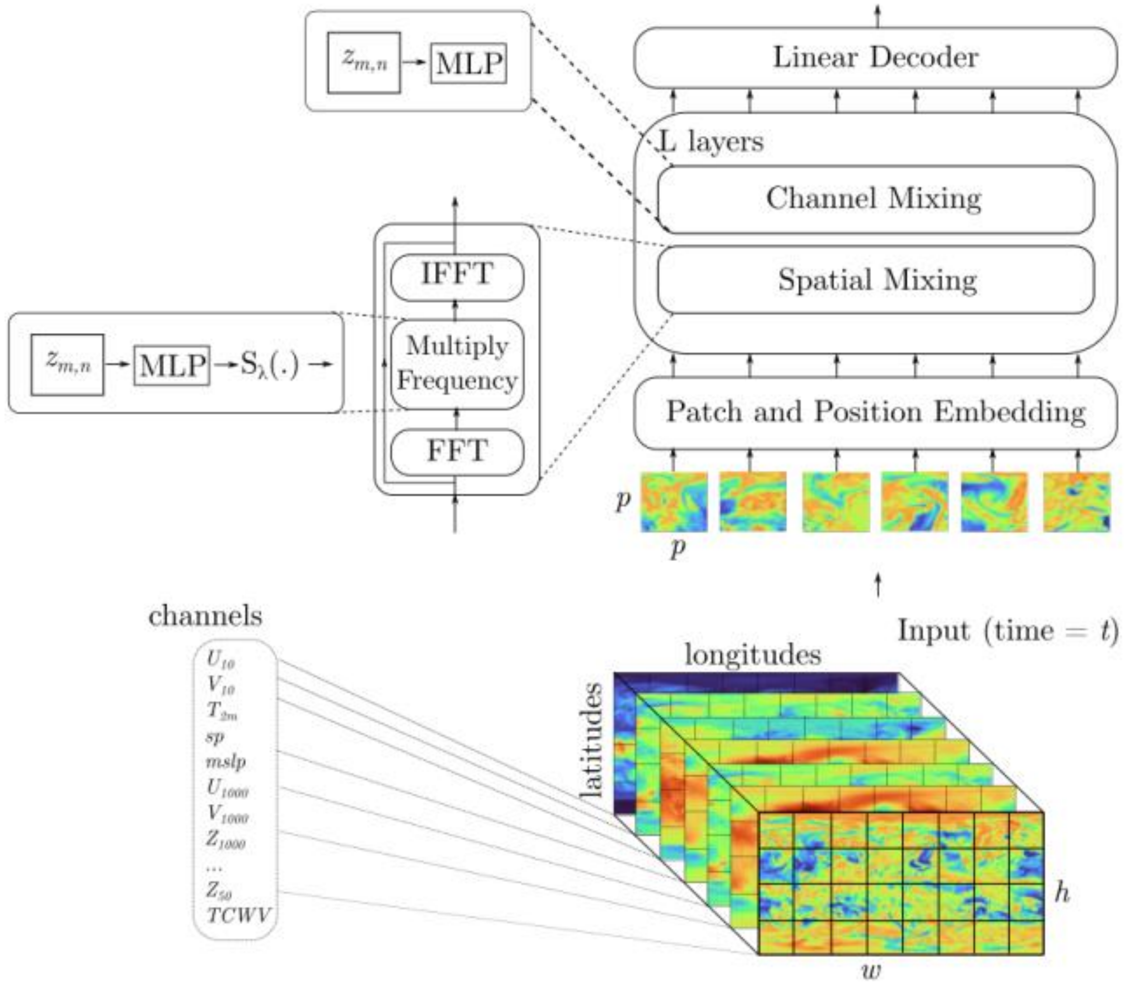
Backbone: Adaptive Fourier Neural Operator (AFNO)

Ground Truth: ERA5

Highlights:

- $10^4 \sim 10^5$ × speedup compared to state-of-the-art numerical weather predictions (NWP)
- Comparable accuracy to NWP

However……



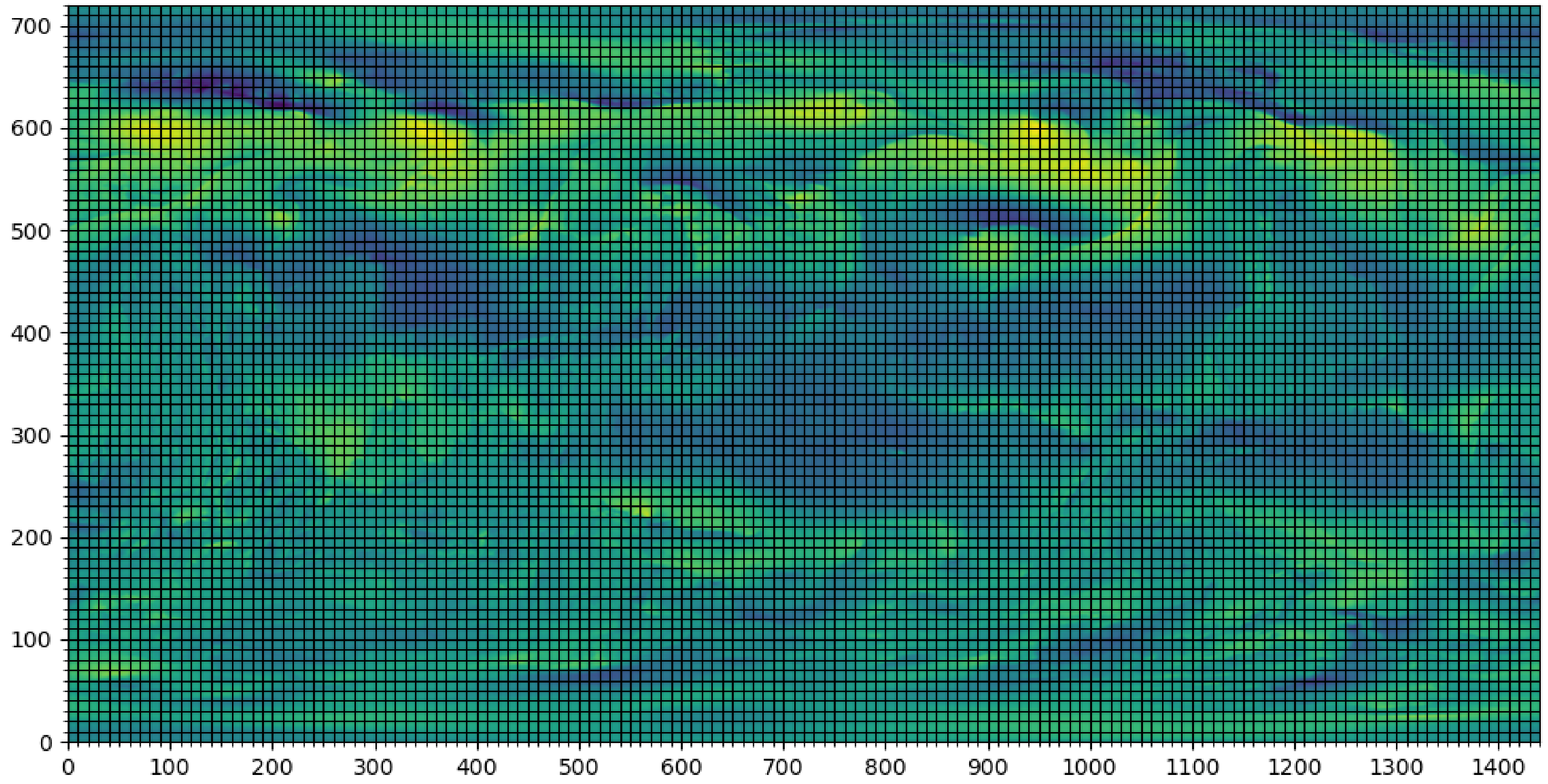
Jaideep Pathak et al., FourCastNet: A Global Data-driven High-resolution Weather Model Using Adaptive Fourier Neural Operators, 2022. https://arxiv.org/abs/2202.11214
John Guibas et al., Adaptive Fourier Neural Operators: Efficient Token Mixers for Transformers, 2021. https://arxiv.org/abs/2111.13587.

Motivation and Objective

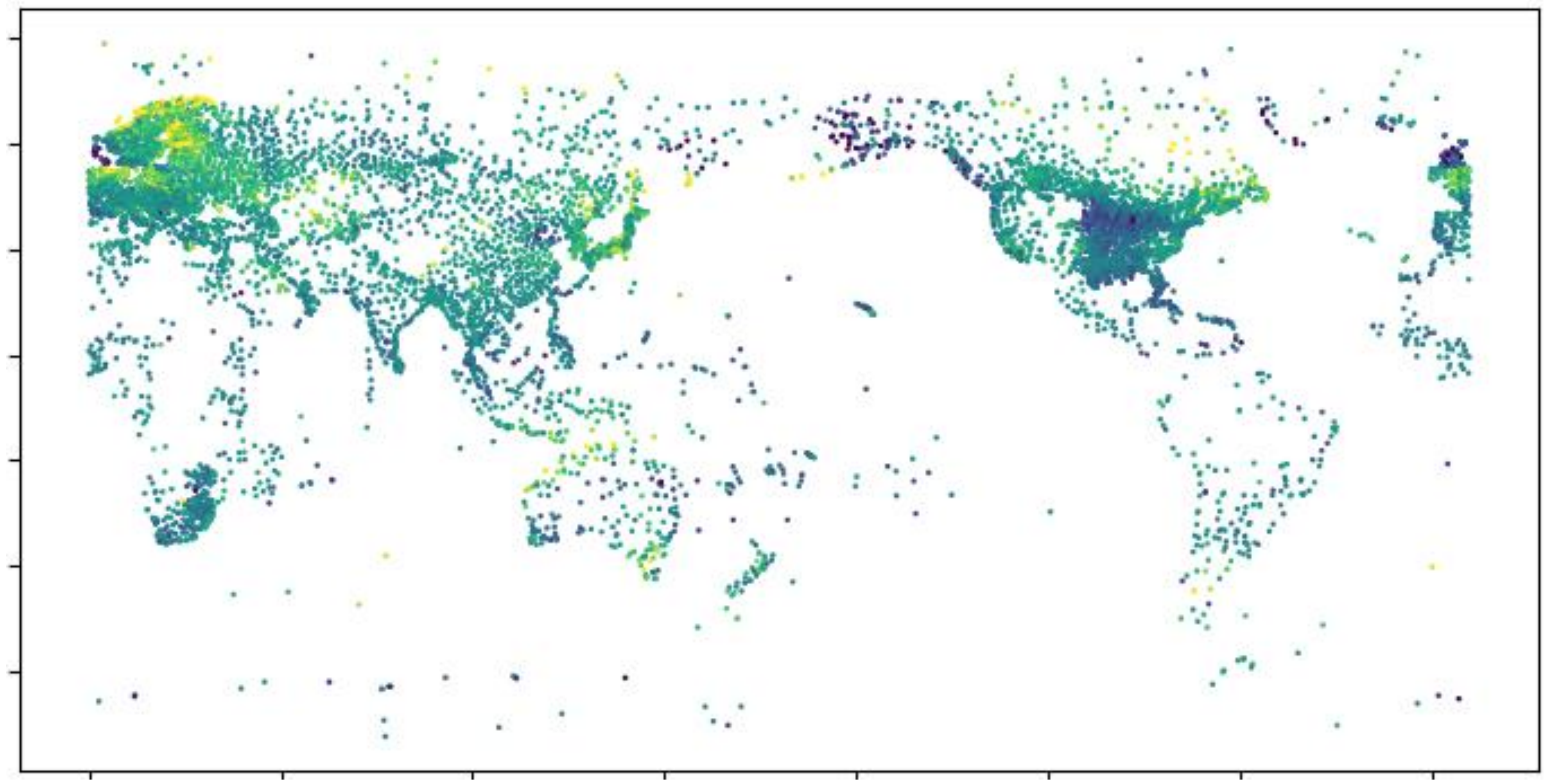
Remap and bias-correct FourCastNet to Gold standard: Sparse, Non-Uniform Observational Data

- DL methods, like FourCastNet, have excellent skill in high-resolution data-driven global weather forecasting, based on held-out test set from ERA5 reanalysis mesh-gridded data as the ground truth.
- However, the mesh-gridded forecasts cannot be directly compared against the gold standard ground truth, i.e., raw sparse, non-uniform climate data from observations.
- Further, because the model is trained on reanalysis data, it is likely to have biases w.r.t. observations
- **Goal: develop a model that can remap and correct mesh-gridded forecasts to arbitrary locations in space and time, under the supervision of sparse observations**

Mesh-gridded weather forecast

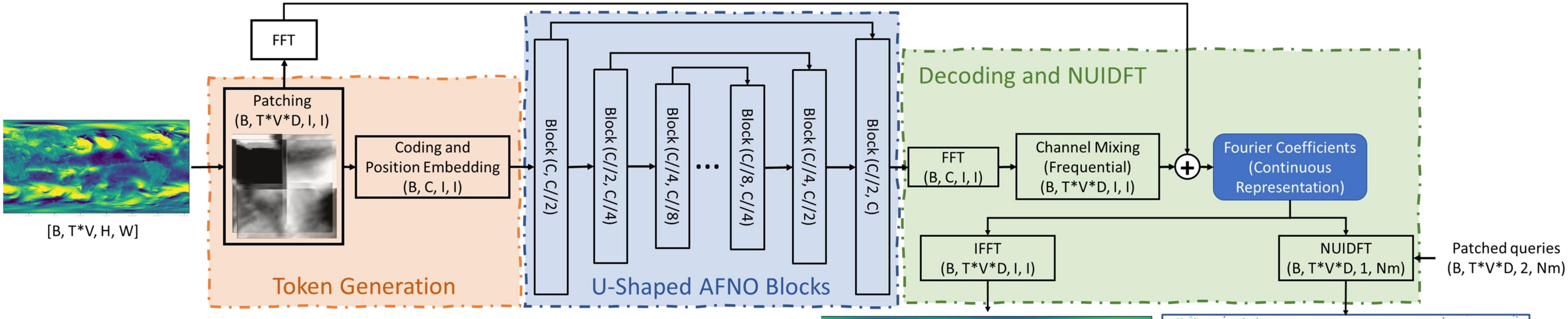


Data from observational sites

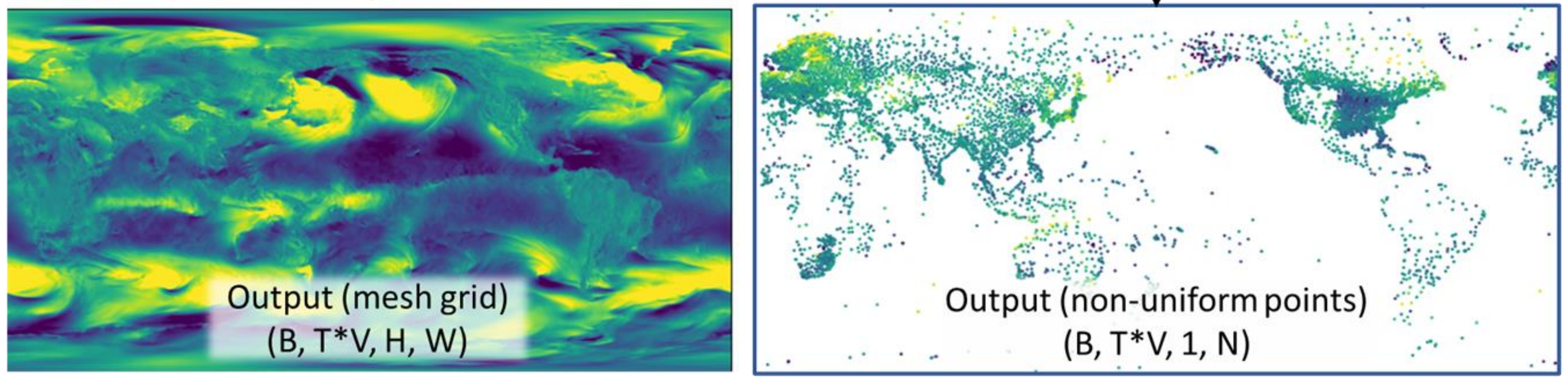


DL Corrector-Remapper (DLCR)

Overall Structure – Grid-Free Network

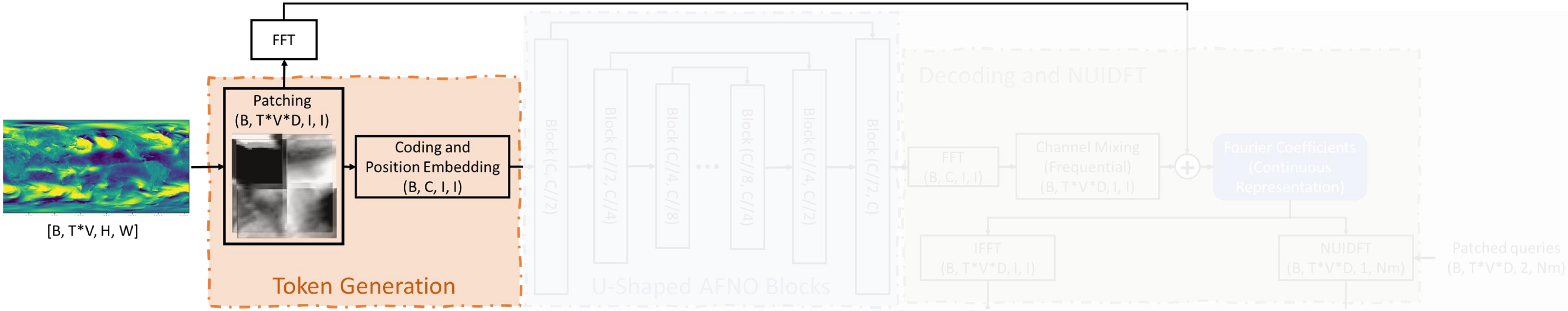


B:	batch size	(1)	**I**:	size of a patch	(30)
T:	Time steps	(10)	**D**:	number of patches for one image	(1152)
V:	# of variables	(4)	**C**:	# of hidden channels	(8196)
H:	height	(720)	**N**:	# of observations	(22,000)
W:	width	(1440)	**Nm**:	maximum # of obs. in a patch	(720)



DL Corrector-Remapper (DLCR)

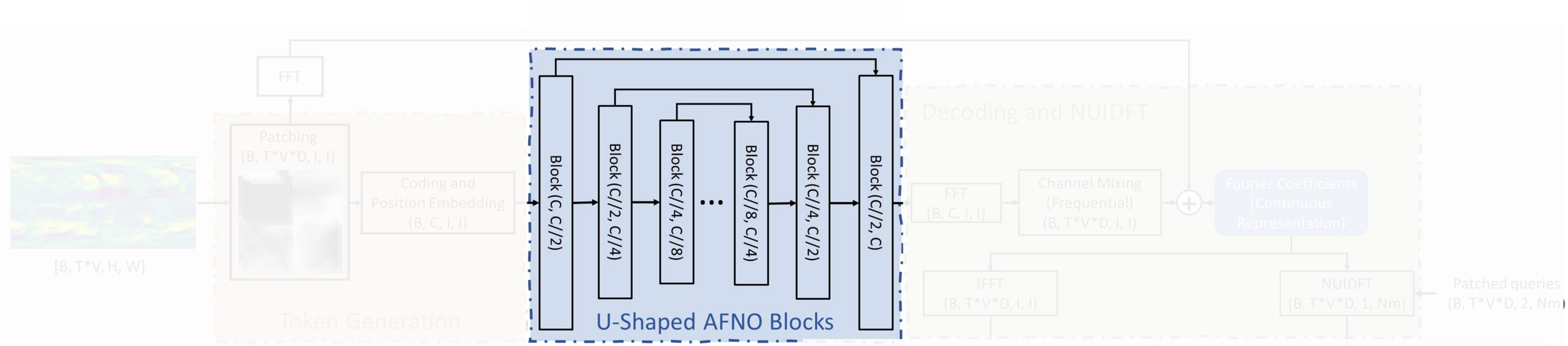
Overall Structure – Grid-Free Network



B:	batch size	(1)	**I**:	size of a patch	(30)
T:	Time steps	(10)	**D**:	number of patches for one image	(1152)
V:	# of variables	(4)	**C**:	# of hidden channels	(8196)
H:	height	(720)	**N**:	# of observations	(22,000)
W:	width	(1440)	**Nm**:	maximum # of obs. in a patch	(720)

DL Corrector-Remapper (DLCR)

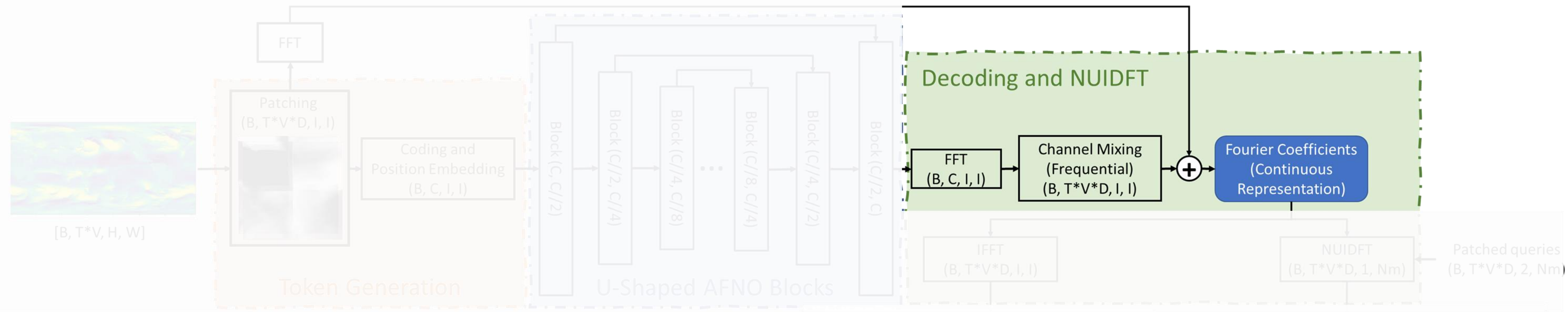
Overall Structure – Grid-Free Network



B:	batch size	(1)	**I**:	size of a patch	(30)
T:	Time steps	(10)	**D**:	number of patches for one image	(1152)
V:	# of variables	(4)	**C**:	# of hidden channels	(8196)
H:	height	(720)	**N**:	# of observations	(22,000)
W:	width	(1440)	**Nm**:	maximum # of obs. in a patch	(720)

DL Corrector-Remapper (DLCR)

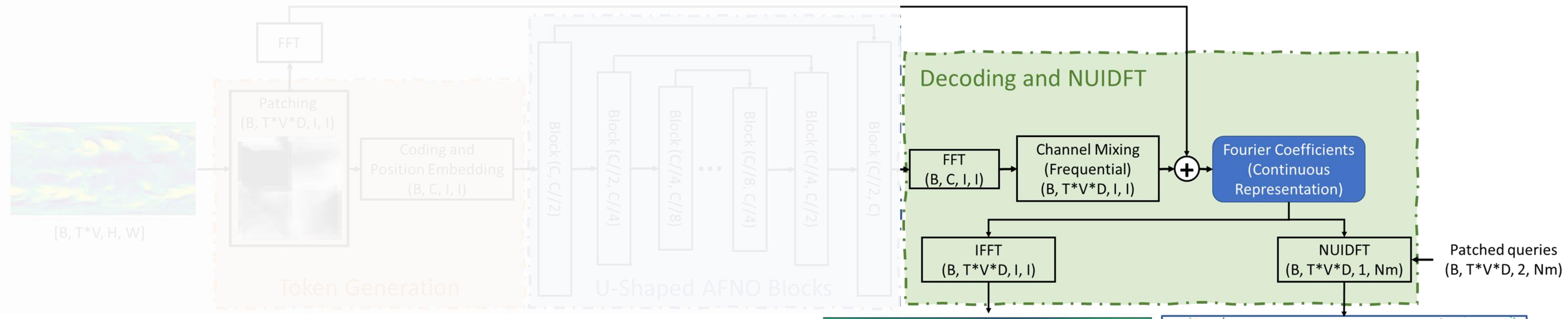
Overall Structure – Grid-Free Network

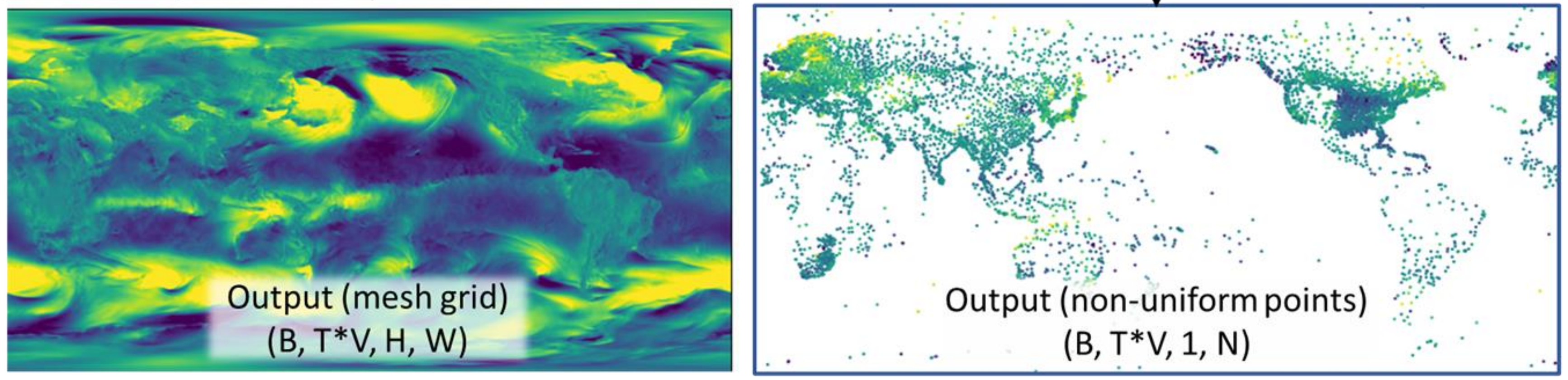


B:	batch size	(1)	**I**:	size of a patch	(30)
T:	Time steps	(10)	**D**:	number of patches for one image	(1152)
V:	# of variables	(4)	**C**:	# of hidden channels	(8196)
H:	height	(720)	**N**:	# of observations	(22,000)
W:	width	(1440)	**Nm**:	maximum # of obs. in a patch	(720)

DL Corrector-Remapper (DLCR)

Overall Structure – Grid-Free Network

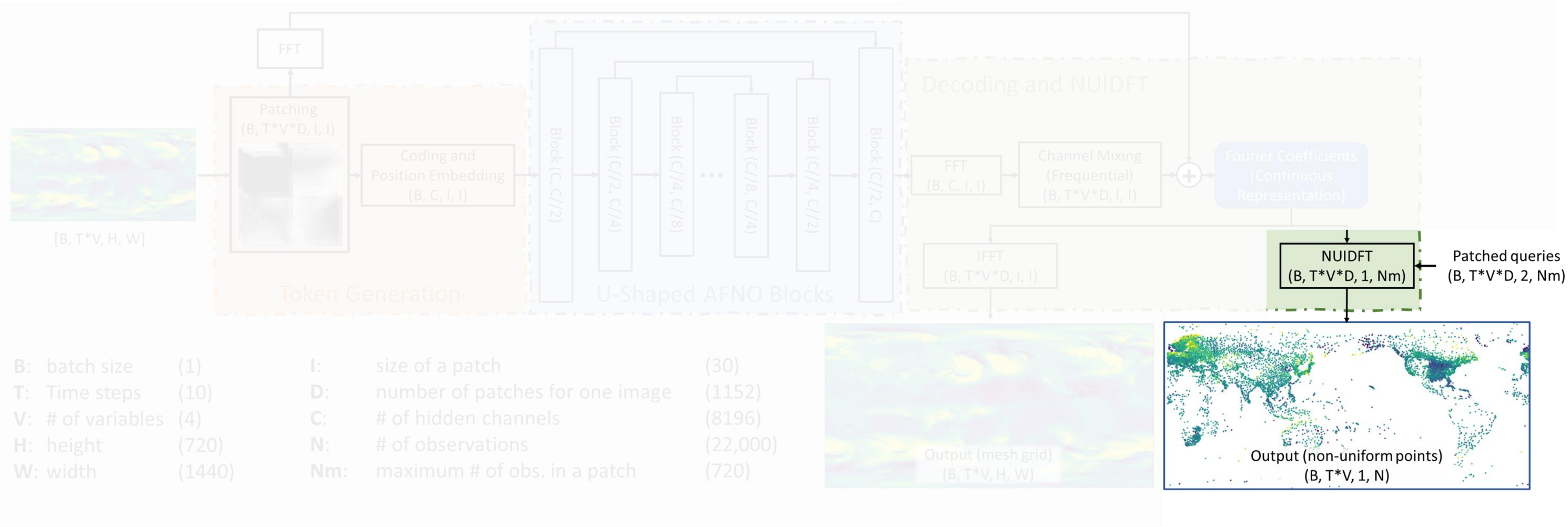




B:	batch size	(1)	**I**:	size of a patch	(30)
T:	Time steps	(10)	**D**:	number of patches for one image	(1152)
V:	# of variables	(4)	**C**:	# of hidden channels	(8196)
H:	height	(720)	**N**:	# of observations	(22,000)
W:	width	(1440)	**Nm**:	maximum # of obs. in a patch	(720)

DL Corrector-Remapper (DLCR)

Overall Structure – Grid-Free Network



NUIDFT: $\frac{1}{\sqrt{WH}}\{\cos(2\pi\, Q^T \cdot M^T)F_{real}^T - \sin(2\pi\, Q^T \cdot M^T)F_{img}^T\}$

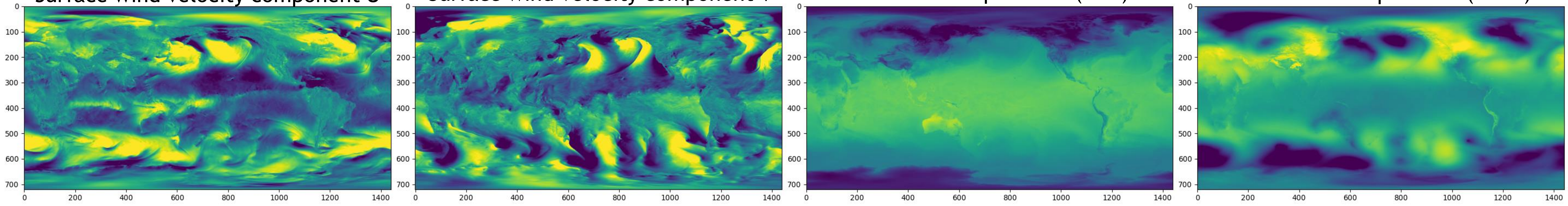
Q:	query matrix
M:	frequency basis
F_{real}, F_{img}:	real/img. Fourier coefficients

Model Training

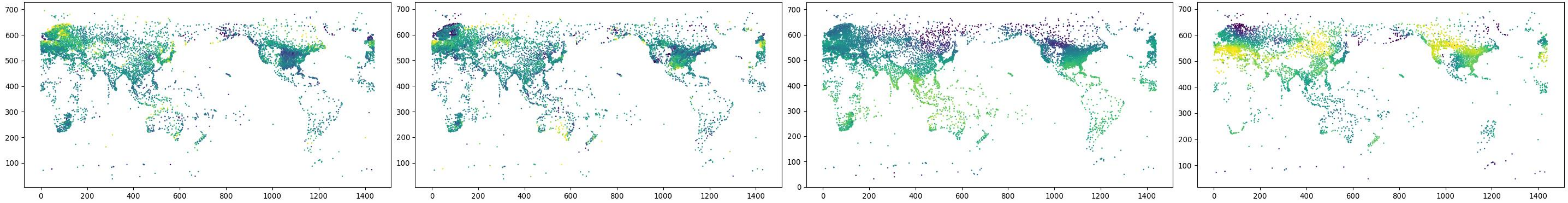
Dataset

Input: Inference Data 2000-2018 from FourCastNet

Surface wind velocity component U | Surface wind velocity component V | Surface temperature (t2m) | Sea level mean pressure (SLMP)



Ground Truth: Global Observation Data 2000-2018

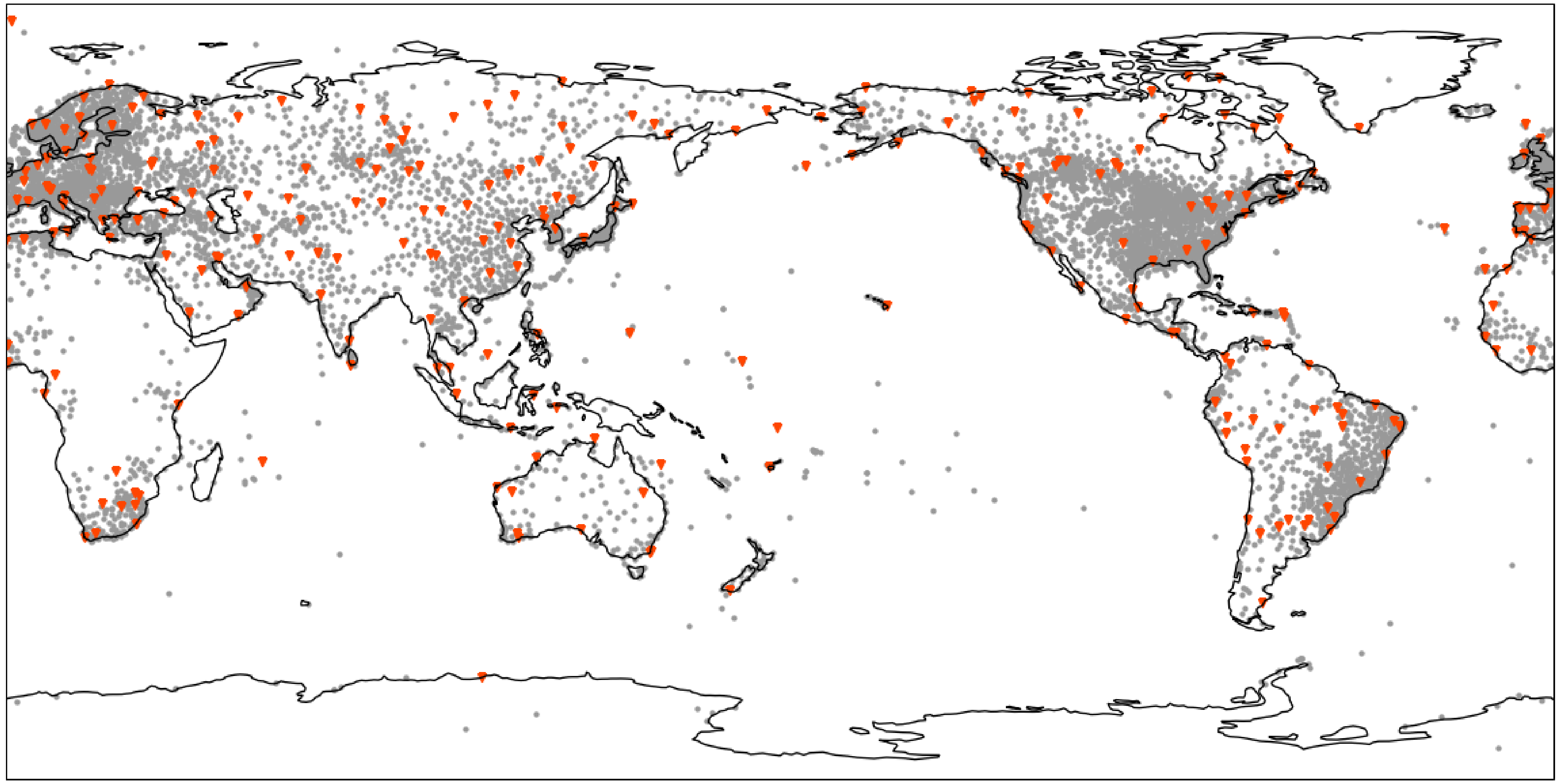


- 0.25° resolution
- 720x1440 image size
- 4 variables
- 10 timesteps (120 hours)
- 5 days lead time

Global observation data from National Centers for Environmental Information: https://www.ncei.noaa.gov/metadata/geoportal/rest/metadata/item/gov.noaa.ncdc:C00532/html

Model Training

	Dataset	
Total instances:	27,360	2000-2018
Training (observed time):	23,040	2000-2015
Time gap	1,440	2016
Test (unobserved time):	1,440	2017
Total locations:	~22,000	(100%)
Training (observed locations):	~21,500	(98%)
Test (unobserved locations):	~500	(2%)



Observed locations by model   Unobserved locations

MODELING GOALS:

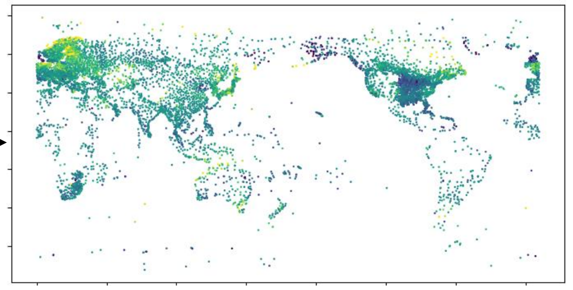
Unobserved time: produce reliable future forecasts

Unobserved locations:
Produce observation-quality data for locations that do not have observations. Much harder.

Model Training

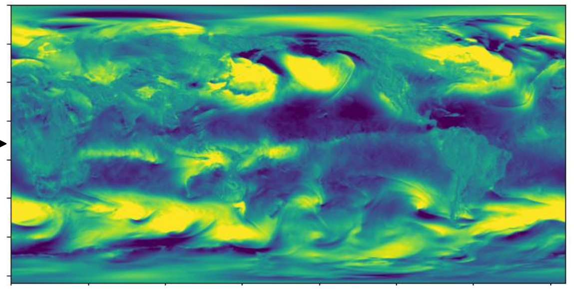
Loss Function

q_{sparse} → **Model** →



$$\text{Output}_{sparse} = \Gamma(\text{Input}, q_{sparse})$$

q_{mesh} → **Model** →



$$\text{Output}_{mesh} = \Gamma(\text{Input}, q_{mesh})$$

$$loss = \left\| \text{Output}_{sparse}, \text{observation} \right\|_2^2 + \lambda \cdot (1 - LCC(\text{Output}_{mesh}, \text{ERA5}))$$

Real Ground Truth

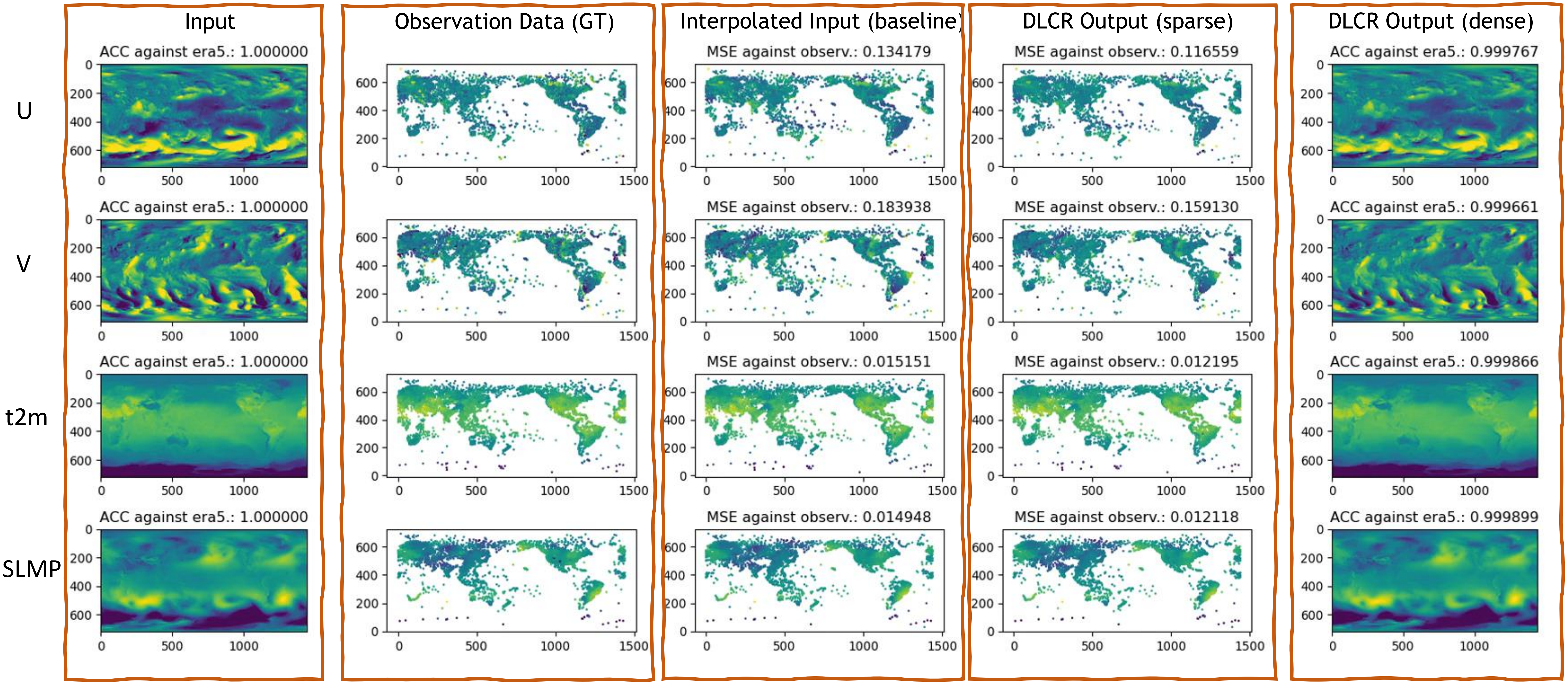
"Ground Truth"

$$LCC(A, B) = \frac{\left(\sum_x (A_x - A_x * K)(B_x - B_x * K)\right)^2}{\sum_x (A_x - A_x * K)^2 \sum_n (B_x - B_x * K)^2}$$

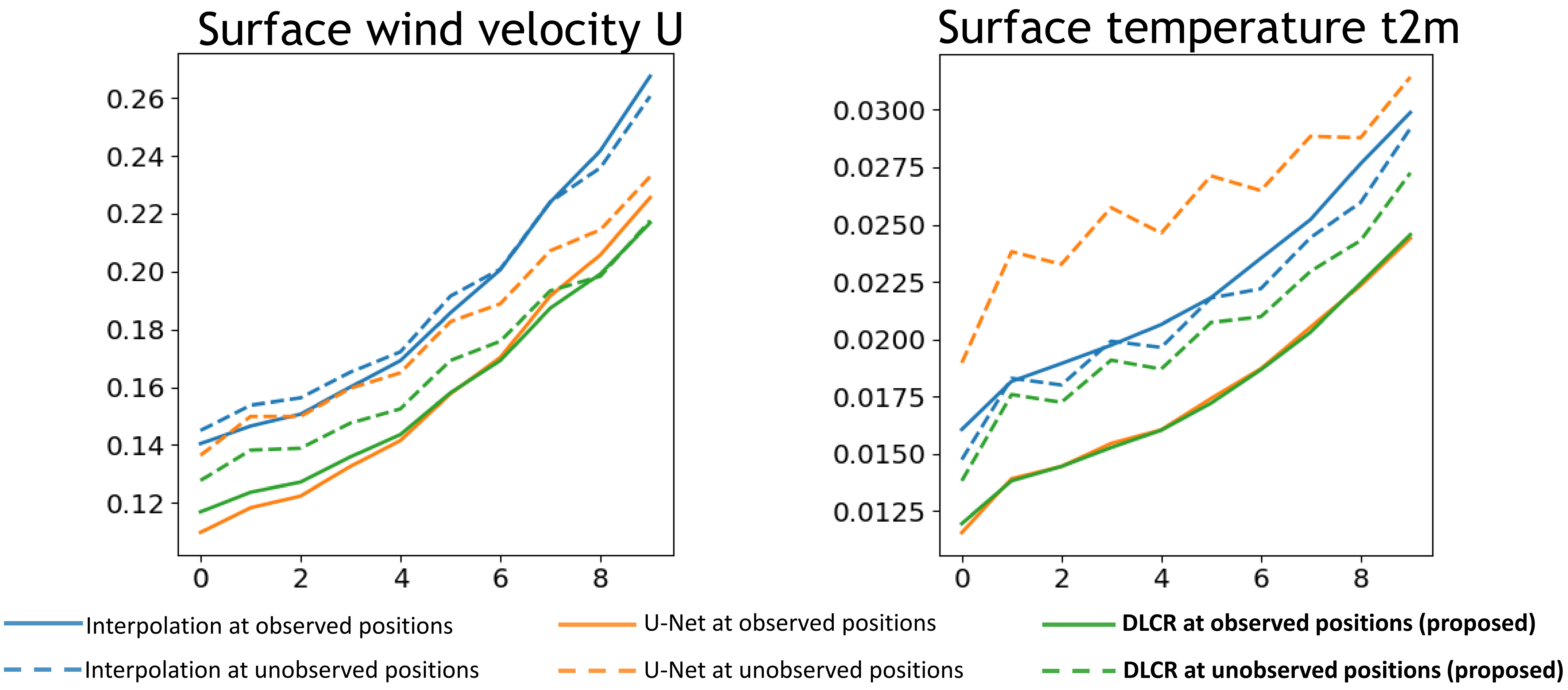
λ hyperparameter
$*$ convolution
K kernel

Results

Out-of-sample timestep, Observed Locations



Results



- Plots of mean square error (MSE) that are averaged over 80 instances across the year 2017 (out of sample).
- The proposed network improves over baselines for both observed and unobserved positions for out of sample timesteps
- Observed positions: the performance of DLCR is close to the performance of U-Net, and they both outperform the interpolation baseline.
- Unobserved positions: DLCR outperforms the interpolation baseline and U-Net, and it performs better on more complicated variables (wind velocity U), whereas the performance of the U-Net is even worse than the performance of the interpolation in estimating t2m.

DL-Corrector-Remapper

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

getao@wustl.edu

https://arxiv.org/abs/2210.12293