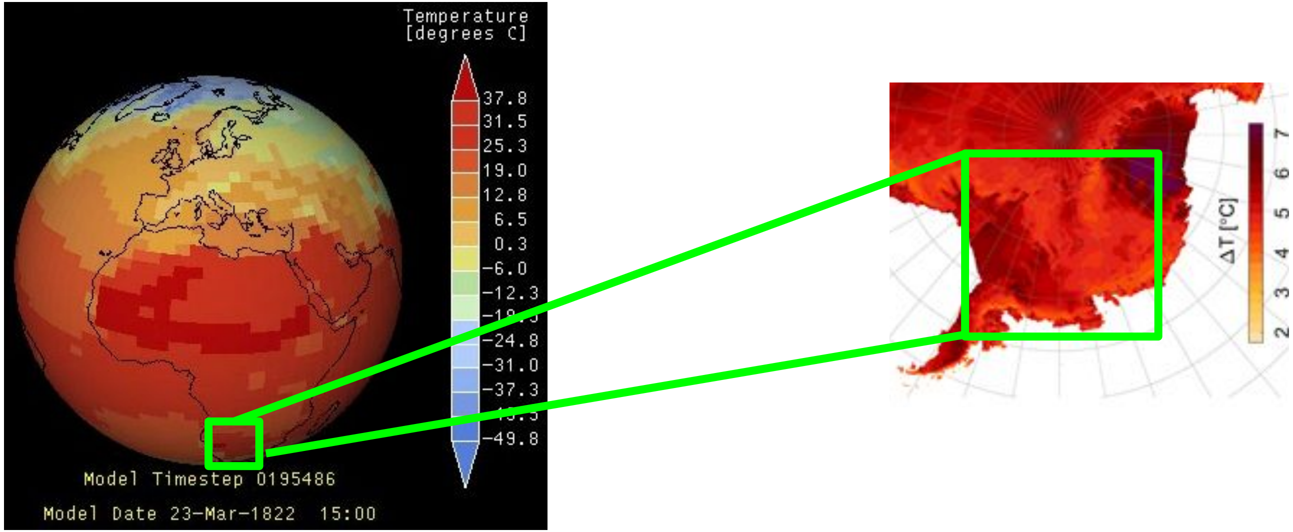
Low Cost Climate Predictions

Turing AI Fellows and Teams Community Hackathon

Climate modelling



Global:

- Computationally **cheap**
- **Low** resolution

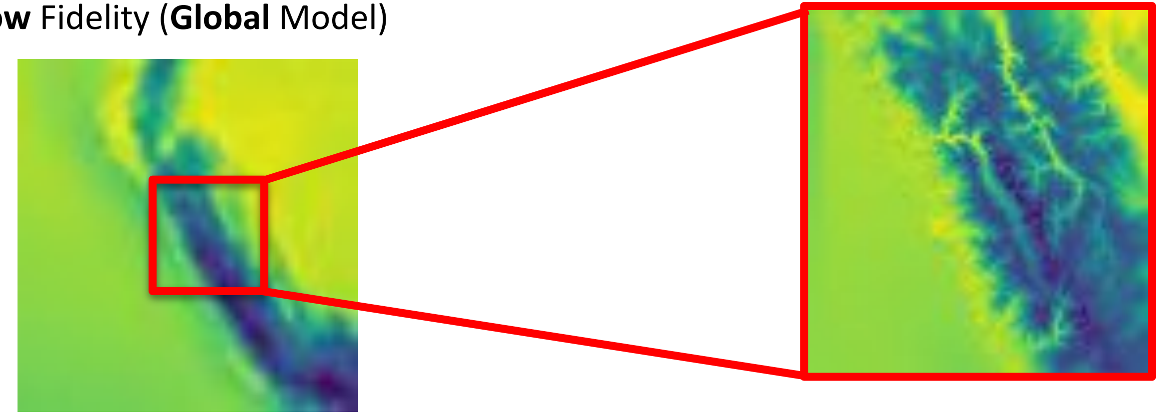
Regional:

- Computationally **expensive**
- **High** resolution

Our Data - Temperature

High Fidelity (**Regional** Model)

Low Fidelity (**Global** Model)



Dimensions (93, 87)

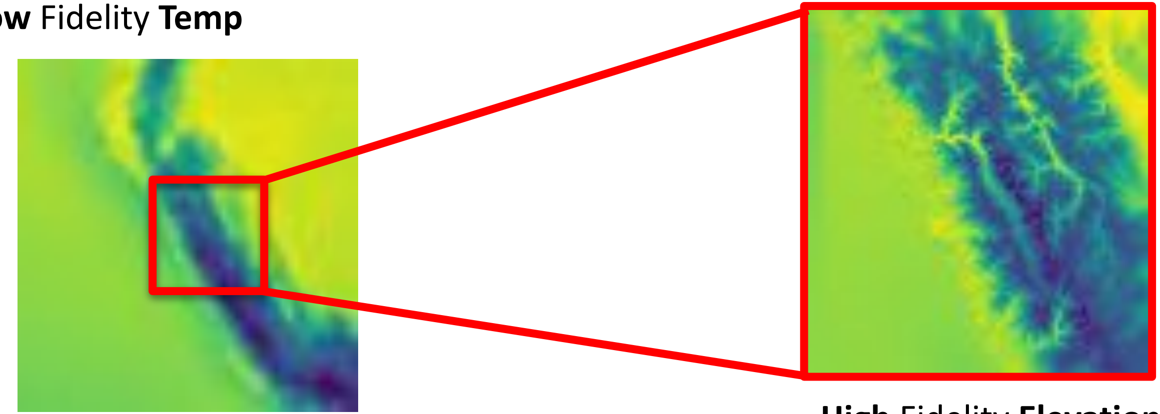
Dimensions (54, 52)

Both start/end date = **1980**-01-31 / **2018**-12-31 (**468 months**)

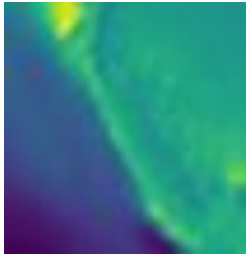
Our Data - Extras

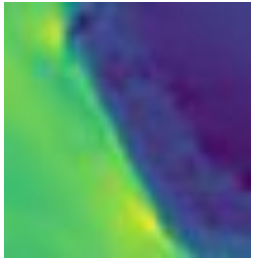
High Fidelity **Temp**

Low Fidelity **Temp**

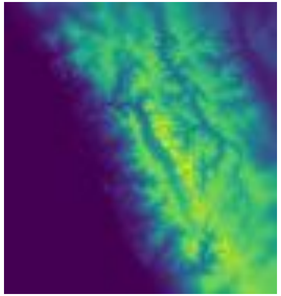


Low Fidelity West-East and South-North **Wind**





High Fidelity **Elevation**



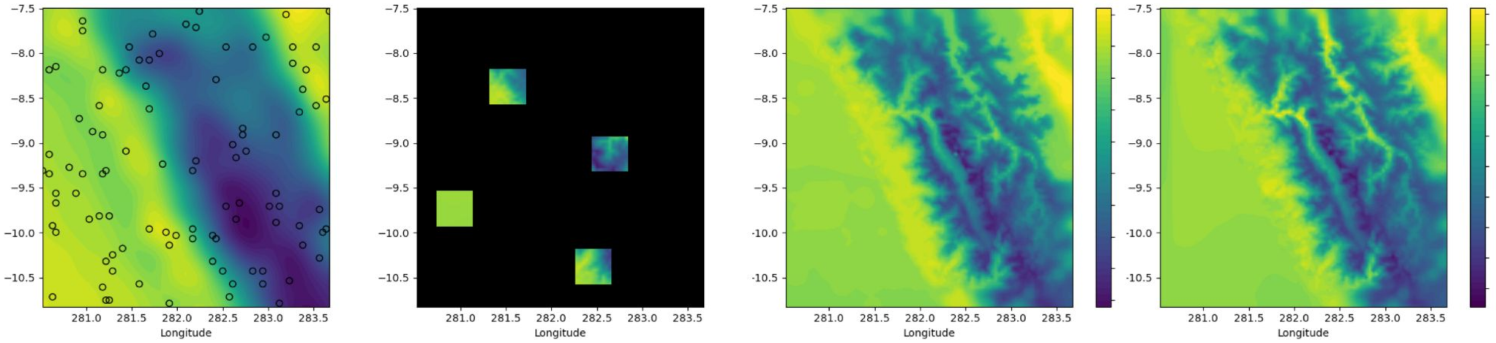
Task

Predict High Fidelity Temp from Low Fidelity Data

Previous Methods

Model	Input	Output
HF	Latitude, Longitude, Altitude	HF Temp
$LF \rightarrow HF$	Latitude, Longitude, Altitude, LF Wind, LF Temp	HF Temp
MF	Latitude, Longitude, Altitude	LF/HF Temp

TABLE I: Summary of the models evaluated.



(a) Low-fidelity training samples

(b) High-fidelity training samples

(c) Inferred high-fidelity output

(d) True high-fidelity output

Fig. 3: A sample of the multi-fidelity model using the proposed batch-wise acquisition function, $a_{IVR,B,\max}$, and acquiring small sub-regions of high-fidelity data, showing (a) the low-fidelity training samples from the GCM (b) the high-fidelity training sub-regions from the RCM (c) the inferred high-fidelity temperature predictions for the entire region of interest, and (d) the target high-fidelity temperature predictions from the RCM.

Previous Methods - Details

Model as a Gaussian process

$$f_{high}(x) = f_{err}(x) + \rho\, f_{low}(x)$$

$$\begin{bmatrix} f_{low}(h) \\ f_{high}(h) \end{bmatrix} \sim GP\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_{low} & \rho k_{low} \\ \rho k_{low} & \rho^2 k_{low} + k_{err} \end{bmatrix} \right)$$

Need to set up an augmented matrix of inputs with their fidelity level and a corresponding matrix of the target variable.

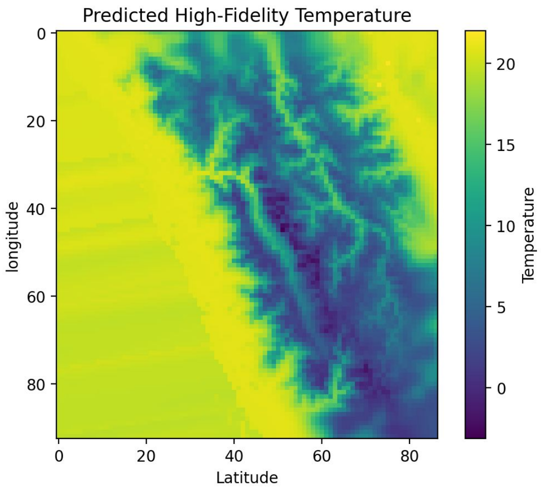
Issue: cannot include covariates in high fidelity that is not included in low fidelity.

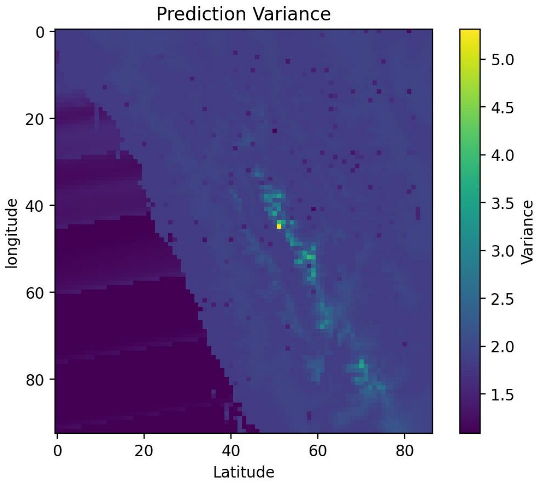
$$X = \begin{pmatrix} x^0_{low;0} & x^1_{low;0} & x^2_{low;0} & 0 \\ x^0_{low;1} & x^1_{low;1} & x^2_{low;1} & 0 \\ x^0_{low;2} & x^1_{low;2} & x^2_{low;2} & 0 \\ x^0_{high;0} & x^1_{high;0} & x^2_{high;0} & 1 \\ x^0_{high;1} & x^1_{high;1} & x^2_{high;1} & 1 \end{pmatrix} \quad Y = \begin{pmatrix} y_{low;0} \\ y_{low;1} \\ y_{low;2} \\ y_{high;0} \\ y_{high;1} \end{pmatrix}$$

Previous Methods - Choosing which high fidelity data to use

- Consider spatial grid in batches (square subset of full grid).
- Consider the sum of variance of all points within given batches $$a_{MV,B,\Sigma} = \sum_{b \in B} \sigma^2(b)$$
- Batch with greatest variance is added to high fidelity data used in modeling.

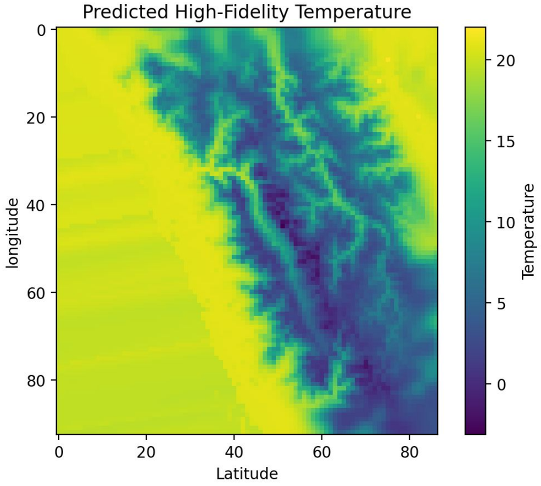
Previous Methods - Our Results

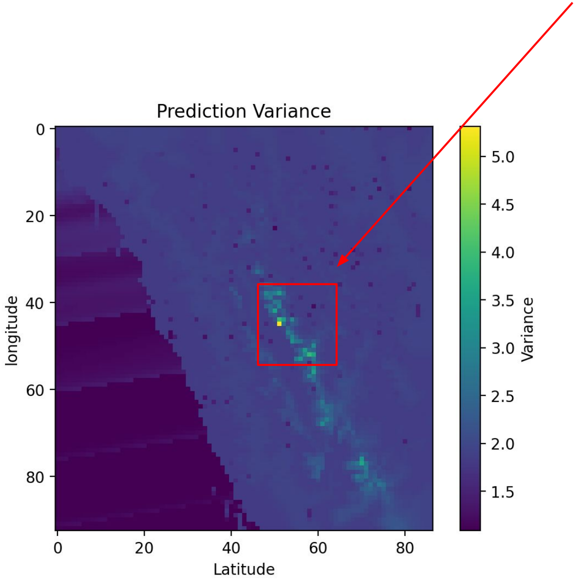




Previous Methods - Our Results

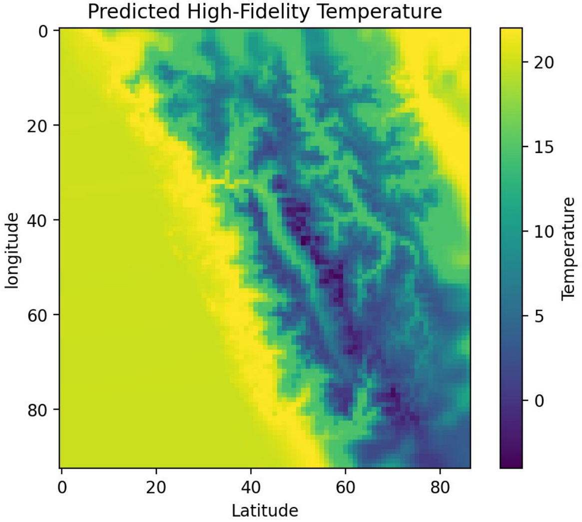
New data to include

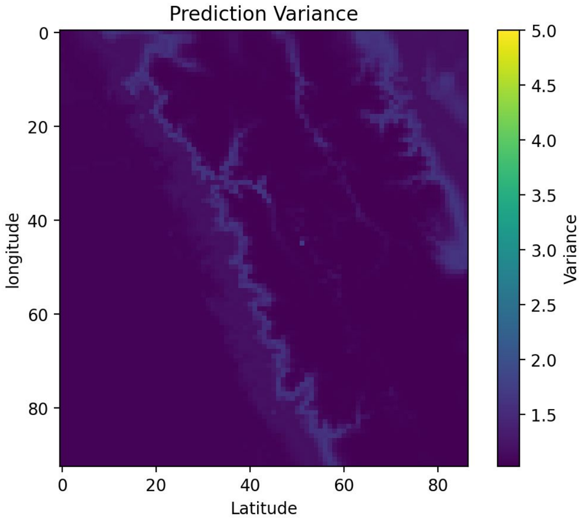




Previous Methods - Our Results

Observe reduction in variance…





Linear Models

Problem formulation: Given **historic** low and high fidelity pairs can we generalise into the **future**

Setup: Train a linear model for the **first 300** months then evaluate on the **final 168** months

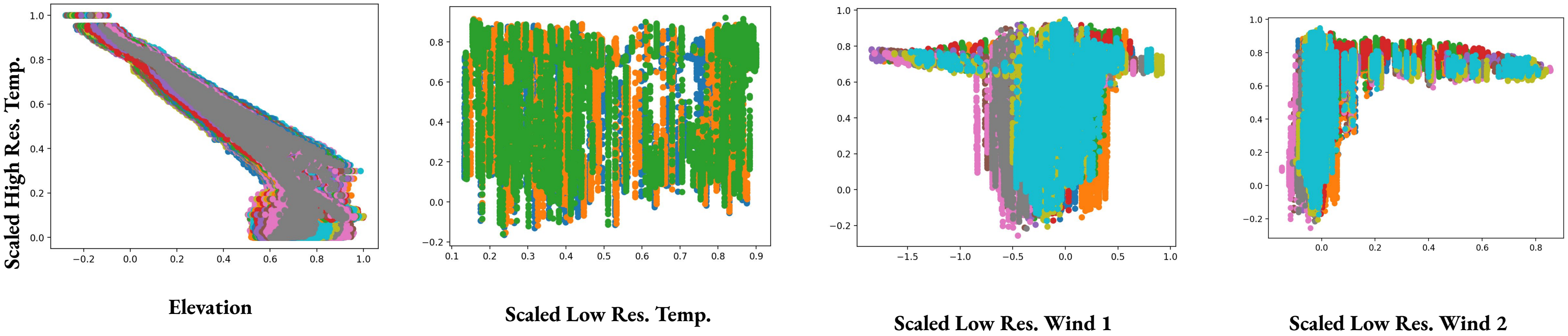
Linear Model - Single Model

Predict high resolution temperature as function of elevation, low resolution temperature and low resolution wind speeds

Single Model for entire ‘image’ with assumption longitude and latitude information encapsulated in elevation and low resolution data

The model achieved an **MSE** of **3.33**

Linear Model - Relationship between features and output



Linear Model - Spatial and Temporal Generalisation

Splitting our data spatially and linearly we can examine the spatial and temporal generalisability of our model

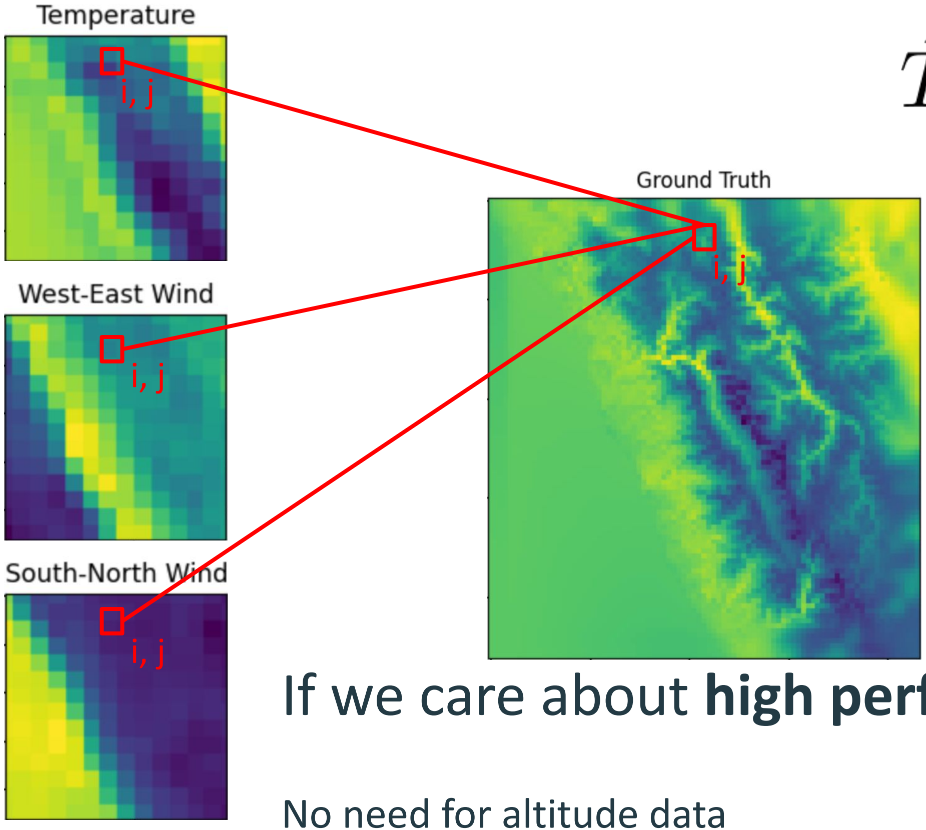
	Train Months	Test Months
Train Region	3.022	3.149
Test Region	3.766	4.144

Model only sees data within the temporal and spatial training window

*best GP model gets **MSE** of **15.62** but uses less high fidelity data*

Linear Model - ‘Pixel’ Wise

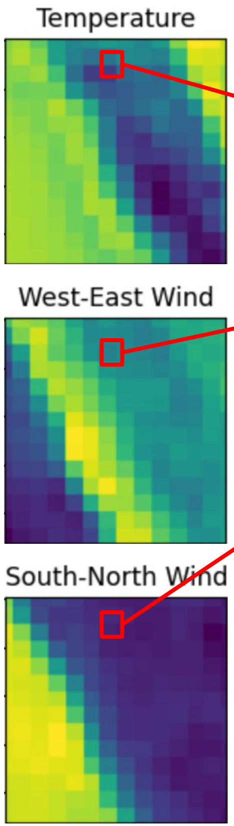
$$\hat{T}_{H\,i,j} = \begin{bmatrix} T_{L\,i,j} \\ W_{L\,i,j} \\ S_{L\,i,j} \\ 1 \end{bmatrix} \cdot \mathbf{w}_{i,j}$$



If we care about **high performance** in a **specific region**

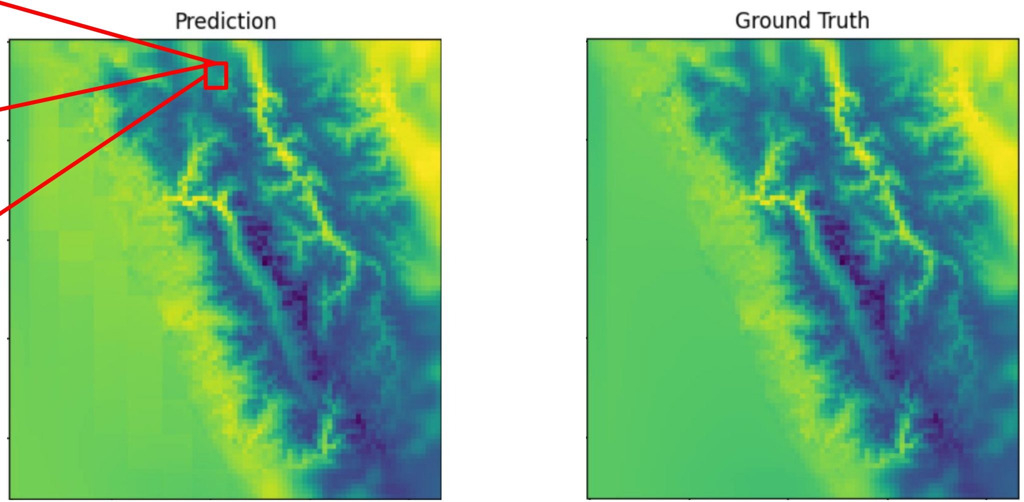
No need for altitude data

Linear Model - 'Pixel' Wise - Results



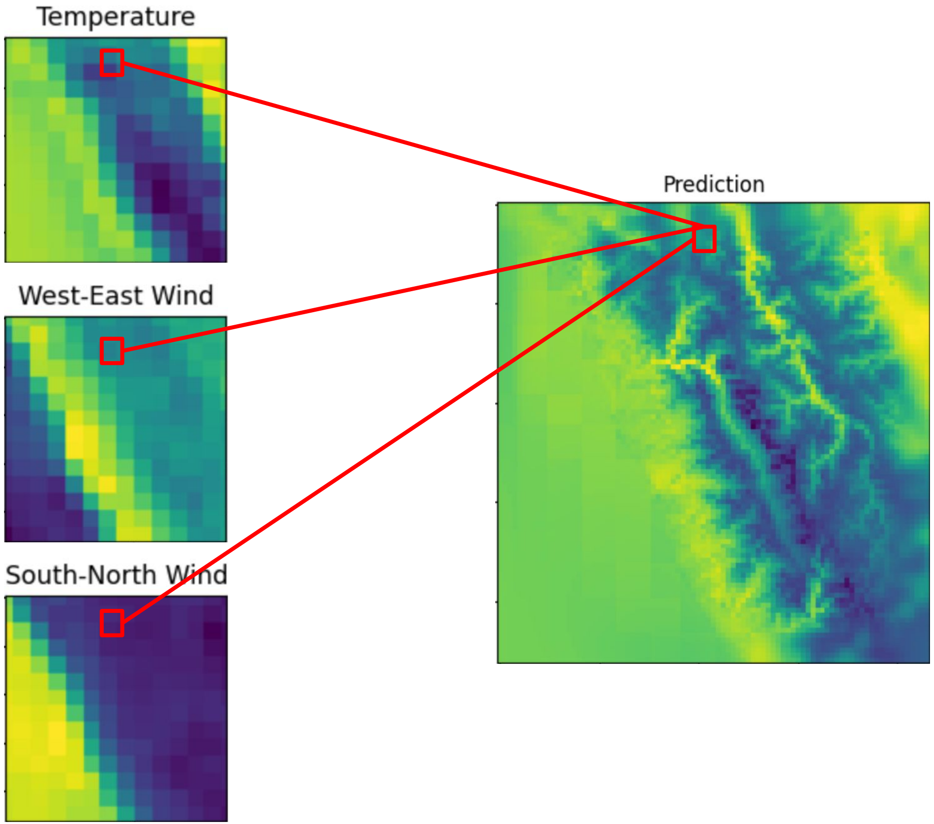
Test **MSE** of **0.336**

(not pixel wise -> **3.33**)



Solution for a **region specific** model using **historical** data

Linear Model - ‘Pixel’ Wise - Limitations



- Specific to the **trained region**
- Will not **generalise** to other areas

VisionTransformer Autoencoder

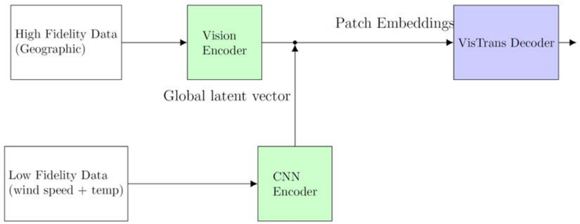
The whole structure is a VisionTransformer based autoencoder.

The low fidelity data is encoded by a CNN-based encoder and the output latent vector is used as the global latent variable everywhere for the high fidelity prediction.

To apply VisionTransformer, the high fidelity data is divided into patches.

The high fidelity data is used as the meta-positional embedding, together with the actual position embedding (patch-wise embeddings).

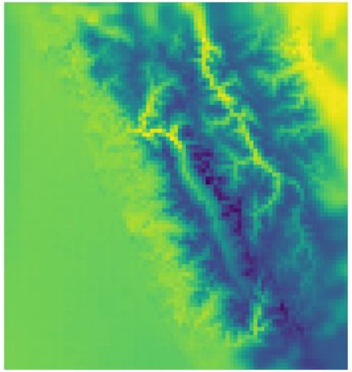
The latent extraction from the low fidelity is concatenated with every patch embeddings and the attention mechanism is applied on the all the patch embeddings to decode them back to the 2D prediction



VisionTransformer Autoencoder

MSE = 1.27

Prediction



Ground Truth