Multi-Task Learning for Space Weather

2021-07-21

Supervisor: Max Berrendorf, Artem Smirnov
Group BigFive42: Yongxin, Jisen, Qi, Laura, Yue

Outline

- Motivation
- Encoder-decoder architecture
- Experimental settings
- Experiment results (Point2Point, Seq2Point, Seq2Seq)
- Qualitative analysis (visualization)

Motivation & Task

Prior Knowledge

upper atmosphere, ionosphere → propagation of GNSS signals (delay)
ionospheric delays → electron density
electron density distribution

- D-layer, E-layer, F-layer
- F1-layer, F2-layer

electron density profile → peak of F2-layer (dominant contribution)

- topside ionosphere
- smooth transition into the plasmasphere

Previous models + time history (solar irradiation or geomagnetic indices)

Sketch of Layers

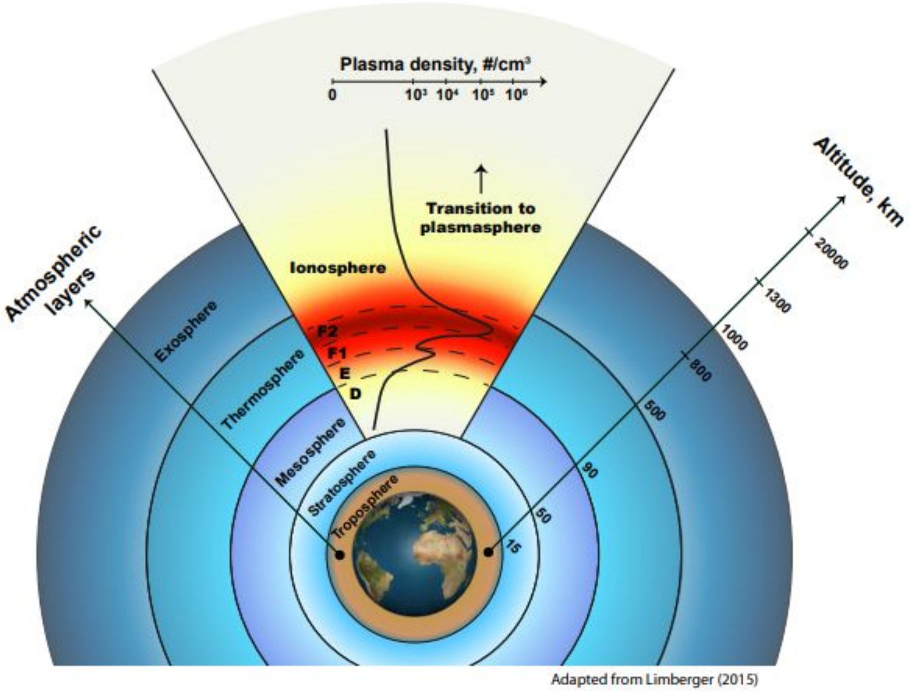


Figure 1: Sketch of the atmospheric layers (defined by temperature domain), and ionospheric layers based on ionisation.

predict the curve

- topside ionosphere
- transition of plasmasphere

Motivation & Task

Motivation

- The photons coming from the Sun ionize particles in the Earth's atmosphere
- The concentration of these charged particles is high enough to cause errors in GPS positioning of >100m
- Electron density profile

Task

- Predict 4 shape parameters of the ionospheric density profiles:
  - NmF2: F2-layer peak density
  - hmF2: F2-layer peak height
  - grad Hs and Hs: compute *h*, giving the decay with increased altitude

Multi-task learning

- Two kinds of input:
  - Shared input for all 4 targets
  - Time & location of measurement for each target

Spatio-temporal model

Dataset

shared input

- **SYM-H**: index that shows strength of the geomagnetic storms
- **F10.7**: solar index
- **Kp**: planetary geomagnetic index

target specific input

- **GLat**: geographic latitude of the measurement
- **GLon**: geographic longitude of the measurement
- **LT**: local solar time
- **topalt**: top altitude of the profile
- **toplat**: top latitude of the profile
- **MLat**: magnetic latitude of the profile
- **MLon**: magnetic longitude of the profile
- **DOY**: day of year

target

- **NmF2**: F2-layer peak density
- **hmF2**: F2-layer peak height
- **Hs** and **grad Hs**

Dataset split

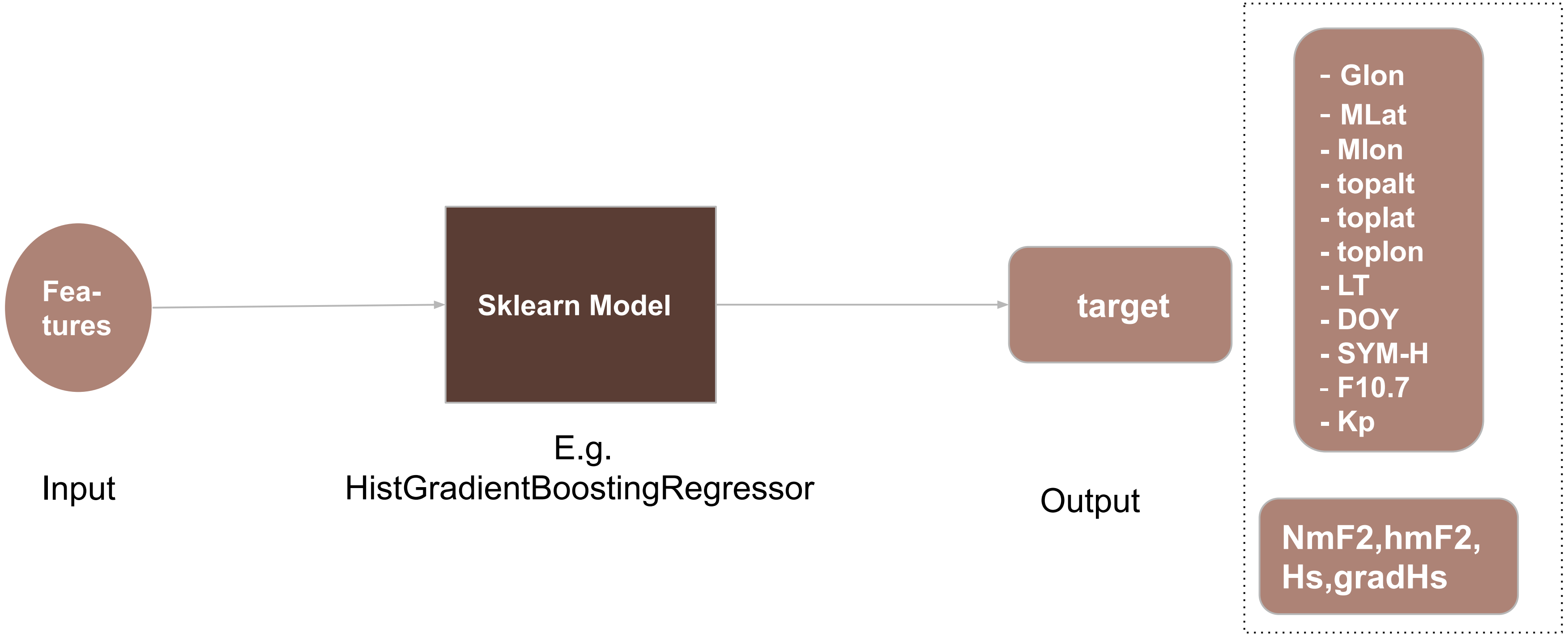
- Dataset sorted by time and evenly split into 5 folds
- Input measurement every 5 min
- Train - Validation - Test split:
    - [0] - [1] - [2]
    - [0, 1] - [2] - [3]
    - [0, 1, 2] - [3] - [4]

2006-06-14T09:11:24 ~ 2008-05-10T12:59:42
2008-05-10T12:59:42 ~ 2010-01-10T10:36:33
2010-01-10T10:36:33 ~ 2012-03-27T12:52:08
2012-03-27T12:52:08 ~ 2014-05-15T09:08:09
2014-05-15T09:08:09 ~ 2019-12-10T11:28:34

from 2006 to 2019, 5 parts, each part 3 years
time series splitting versus random splitting

Encoder-Decoder Architecture

Model types: Sklearn

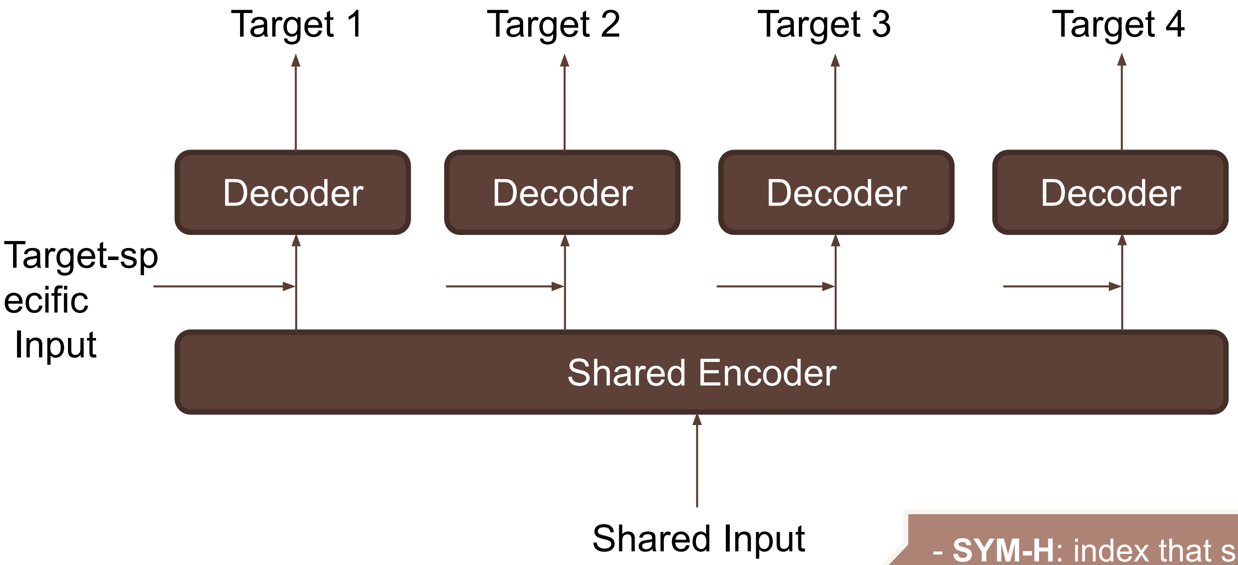


Multi-task Encoder-decoder architecture

4 targets are:
- **NmF2**
- **hmF2**
- **Hs**
- **grad Hs**

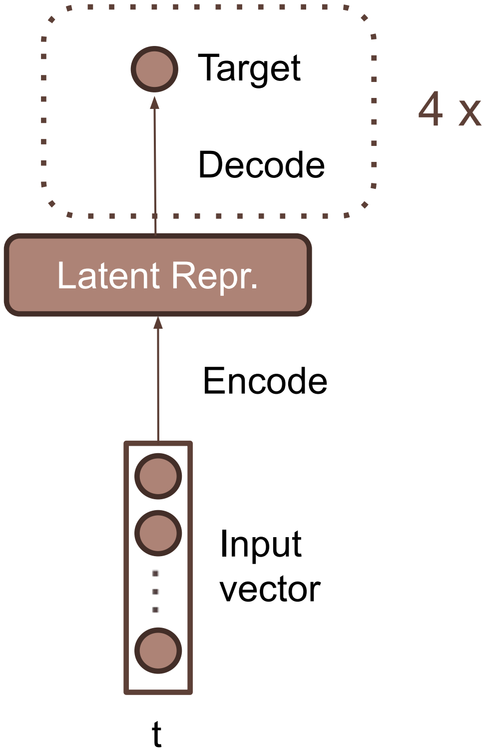
Location & time of target measurement:

- **GLat**
- **GLon**
- **MLat**
- **Mlon**
- **topalt**
- **toplat**
- **toplon**
- **LT**
- **DOY**

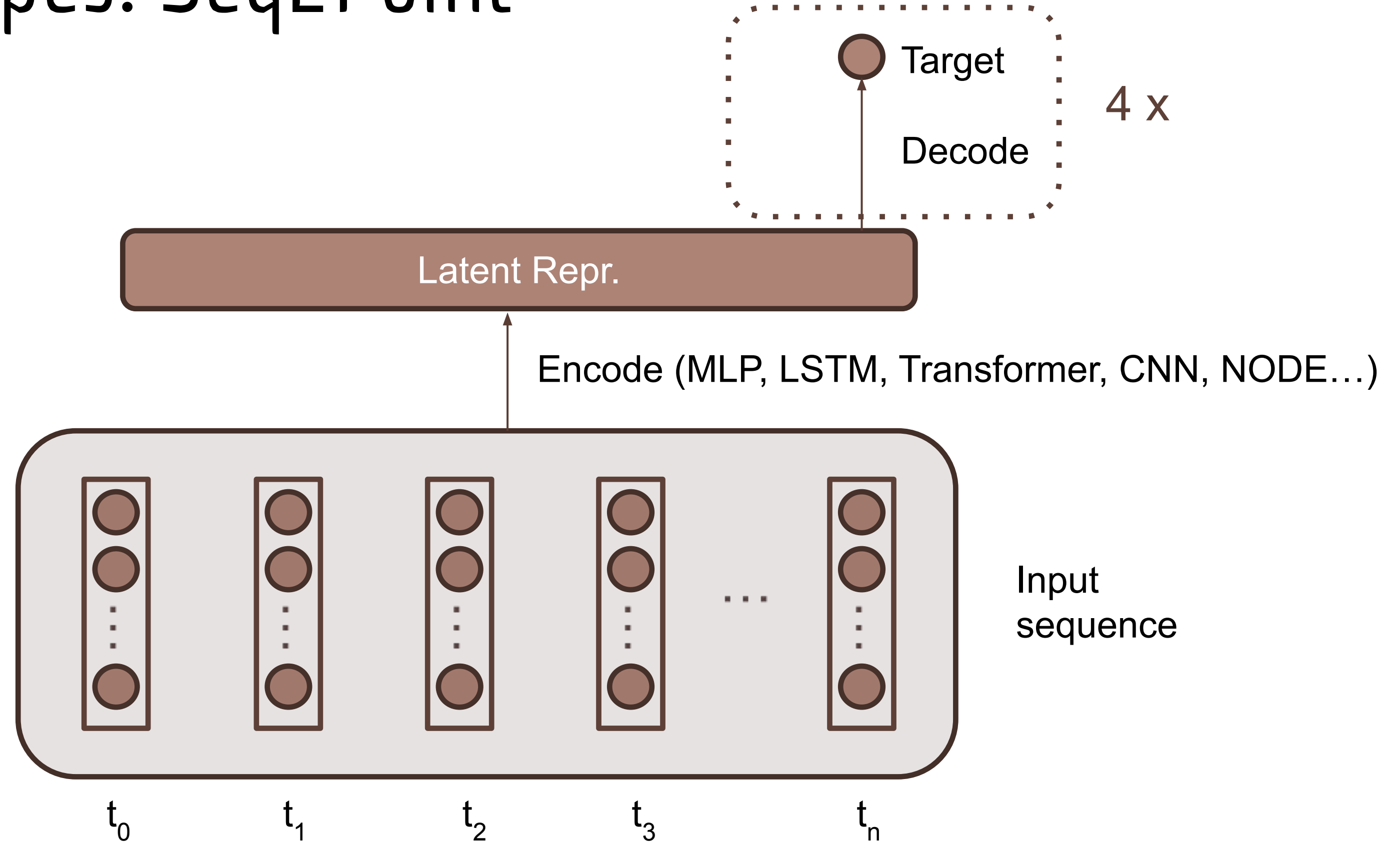


- **SYM-H**: index that shows strength of the geomagnetic storms
- **F10.7**: solar index
- **Kp**: planetary geomagnetic index

Model types: Point2Point

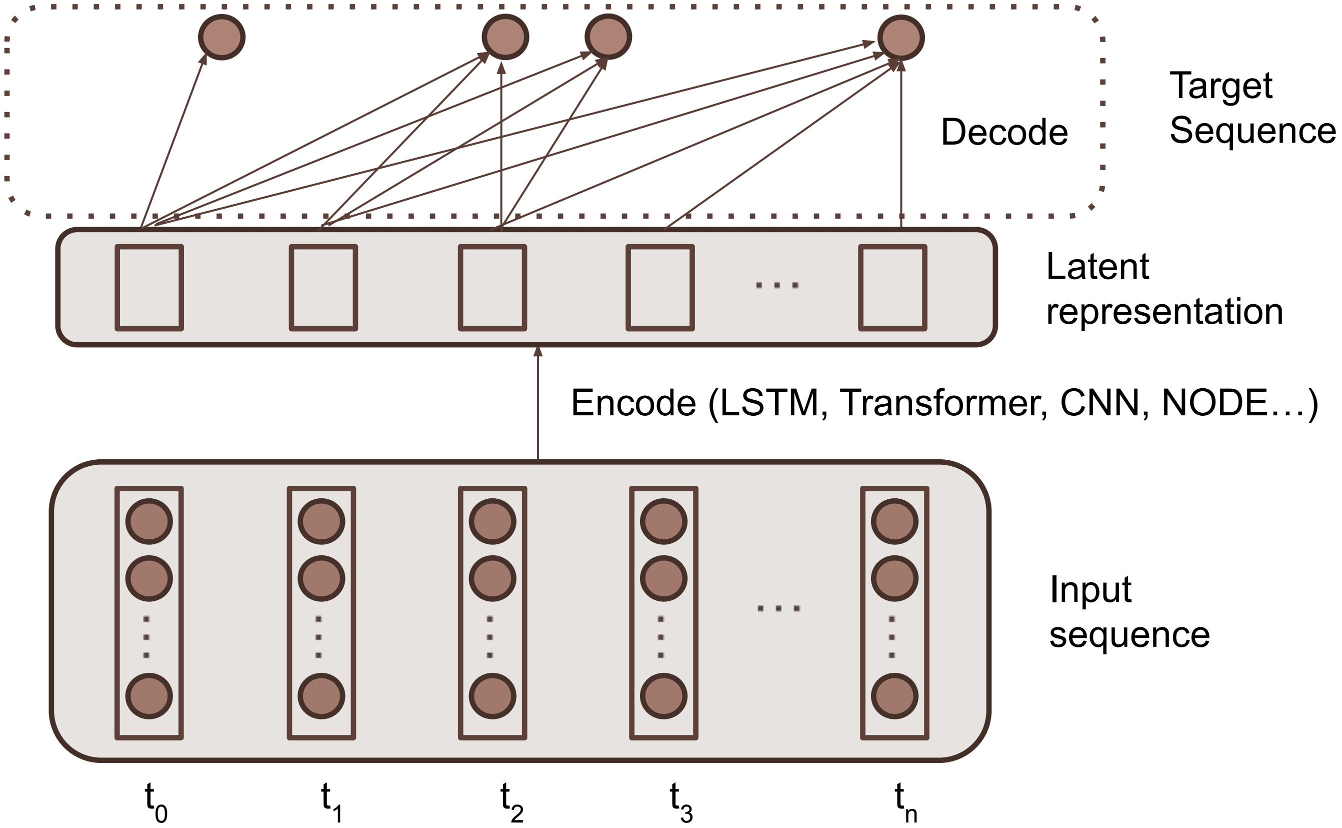


Model types: Seq2Point



Model types: Seq2Seq

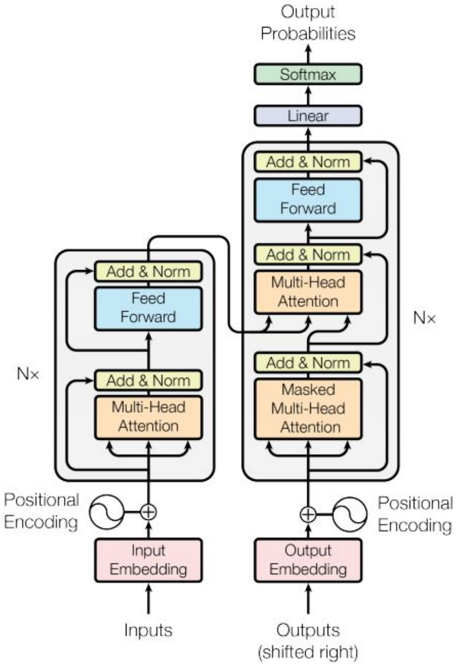
X 4
But different target timestamps for each target
(LSTM, Transformer, CNN, NODE…, here: LSTM)



Attention-LSTM model

- We have a well performing LSTM Seq2Seq model
- Each target timestamp takes the last encoder timestamp before it, which is a summarization of previous input timestamps
- We tried attention as another way of summarization
- Calculate the attention weights for input-target timestamp pair from a vector consisting:
  - difference of two timestamps
  - target-specific inputs at the target timestamp
- This allows more flexible summarization of input sequence
- Similar performance as LSTM Seq2Seq, faster since we don’t have LSTM encoder

Transformer based models



Encoder block + Decoder block

positional encoding

self-attention mechanism

linear layer (mapping)

positional encoding

Transformer encoder layer

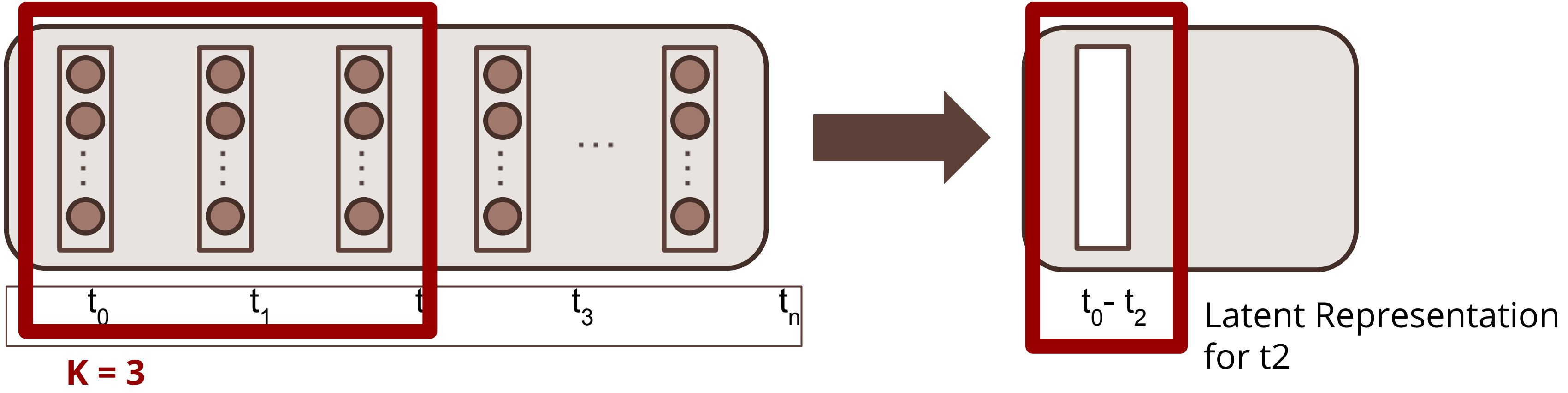
decoders (target numbers: 4)

model trained with:
- transformer encoder + lstm decoder
- transformer encoder + transformer encoder

figure source: Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

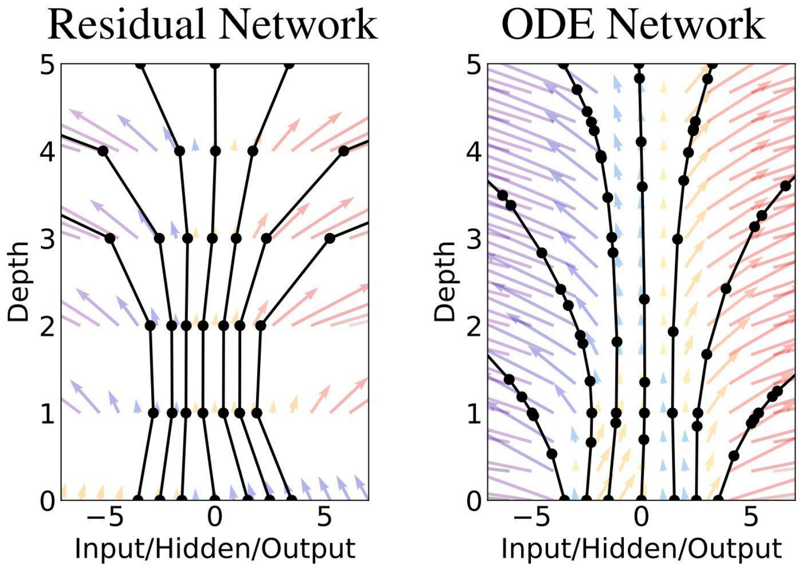
Convolutional Neural Network

- Usually used for computer vision applications
- Convolution operations can be done in 1D, 2D or 3D space
- 1D convolution used for sequential data



Neural Ordinary Differential Equation

- Continuous modeling of hidden state



- Traditional Residual Network:

$$h_{t+1} = h_t + f\left(h_t, \theta_t\right)$$

- Neural ODE:

$$\frac{d\mathbf{h}(\mathrm{t})}{dt} = f\left(\mathbf{h}(t), t, \theta\right)$$

- Motivation: model continuous evolution of hidden state over non-uniform time sequence

Figure source: Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud, Neural Ordinary Differential Equations, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Experimental Settings

Sklearn Model

- Baseline:
  DummyRegressor
- Tree - Based Model:
  Randomforest
  HistGradientBoosting
- Non -Tree Based Model:
  LinearRegression
  Lasso
  Ridge
  KNeighborsRegression

Experimental Setting

Evaluation Metric

- MAPE(Mean Absolute Percentage Error)

→ Statistical measure to define the accuracy of a machine learning algorithm

$$\mathrm{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

where A_t - actual value

F_t - forecast value

n - number of times the summation iteration happens

→ The lower the MAPE, the better fit the model

Experimental Setting

Evaluation Metric

- Spearman Correlation

→ Statistical measurement of the strength of a monotonic relationship

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}.$$

where x_i - values of the x-variable in a sample

$\bar{x}$ - mean of the values of the x-variable

y_i - values of the y-variable in a sample

$\bar{y}$ - mean of the values of the y-variable

→ The closer to ±1, the stronger the monotonic relation

Experimental Results

	MAPE				SC			
Model	NmF2	hmF2	Hs	gradHs	NmF2	hmF2	Hs	gradHs
DummyRegressor	0.851	0.131	0.236	0.367	NaN	NaN	NaN	NaN
HistGradientBoosting	0.4	0.1	0.184	0.293	0.772	0.719	0.638	0.589
PointToPoint	0.205	0.054	0.109	0.204	0.946	0.9	0.855	0.744
SequenceToPoint	0.202	0.052	0.112	0.198	0.947	0.906	0.859	0.748

Further Details

Comparison between Seq2Seq Models

- Results from test set, fold id [4]

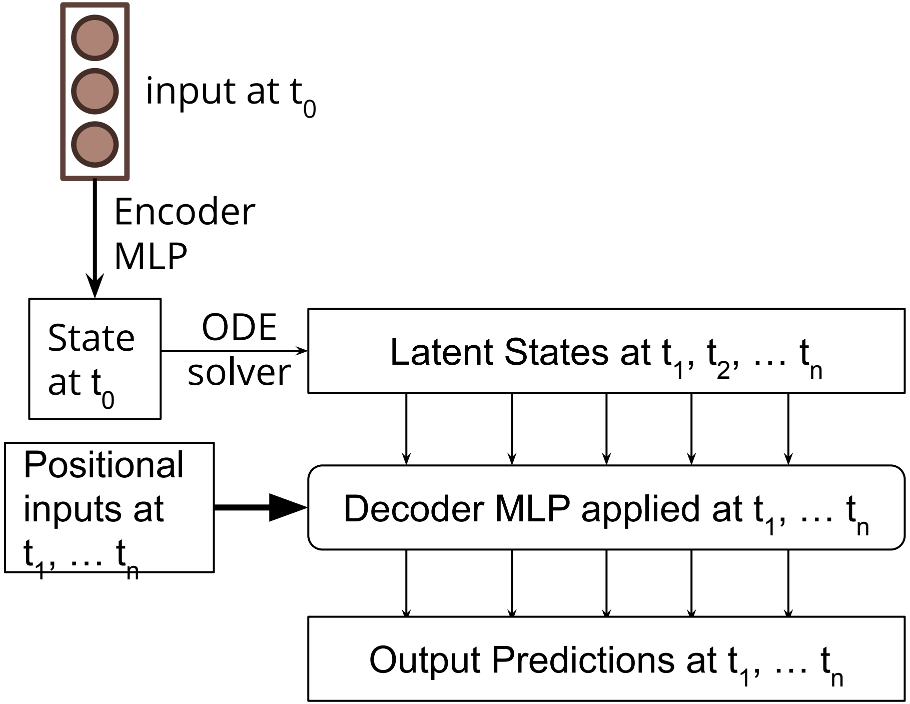
Target	LSTM encoder + LSTM decoder		Attention encoder + LSTM decoder	
	MAE	S. corr.	MAE	S. corr.
Hs	5.438	0.849	5.569	0.839
grad Hs	0.014	0.738	0.014	0.736
NmF2	93436	0.945	94665	0.944
hmF2	14.77	0.898	14.94	0.895

Comparison between Seq2Seq Models (cont.)

- Results from test set, fold id [4]

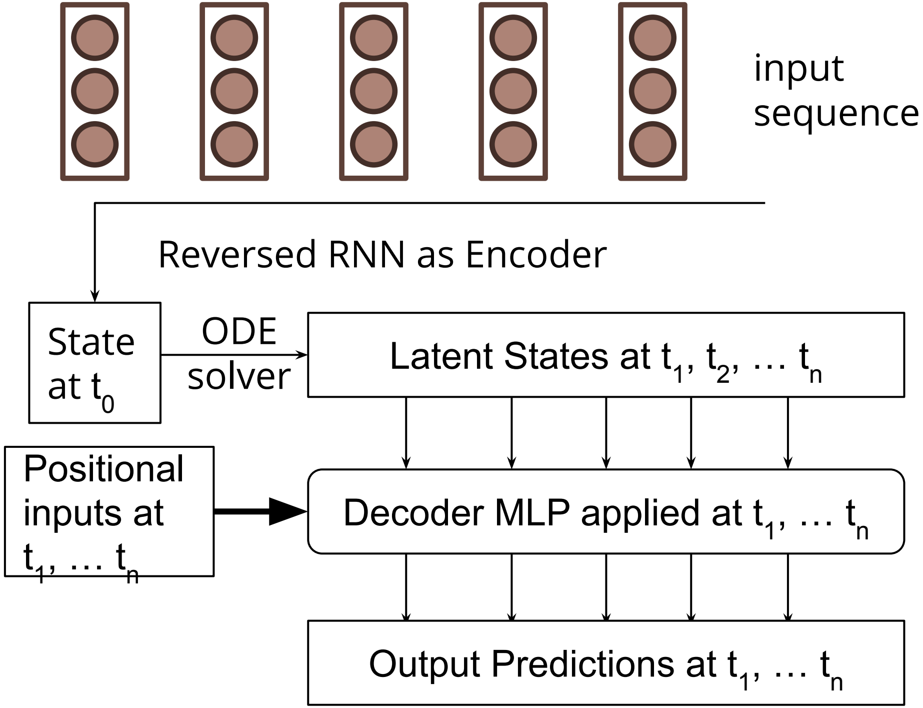
Target	Transformer encoder + LSTM decoder		Transformer encoder + Transformer	
	MAE	S. corr.	MAE	S. corr.
Hs	6.585	0.786	3.664	0.897
grad Hs	0.016	0.670	0.008	0.882
NmF2	127200	0.882	72823	0.906
hmF2	19.07	0.835	10.69	0.904

Neural ODE: Point2Seq attempt



- Result: even the best result has no improvement compared to the training mean, just predicting some constants.
- Reason: one input vector does not have enough information for predicting the whole sequence.

Neural ODE: Seq2Seq attempt



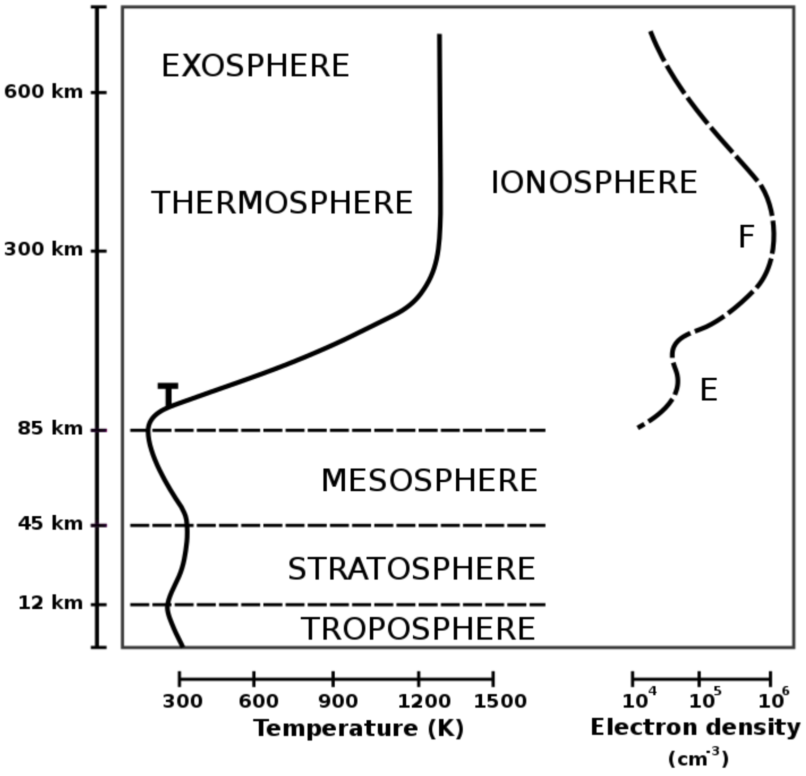
- Result: MSELoss with StandardScaler:

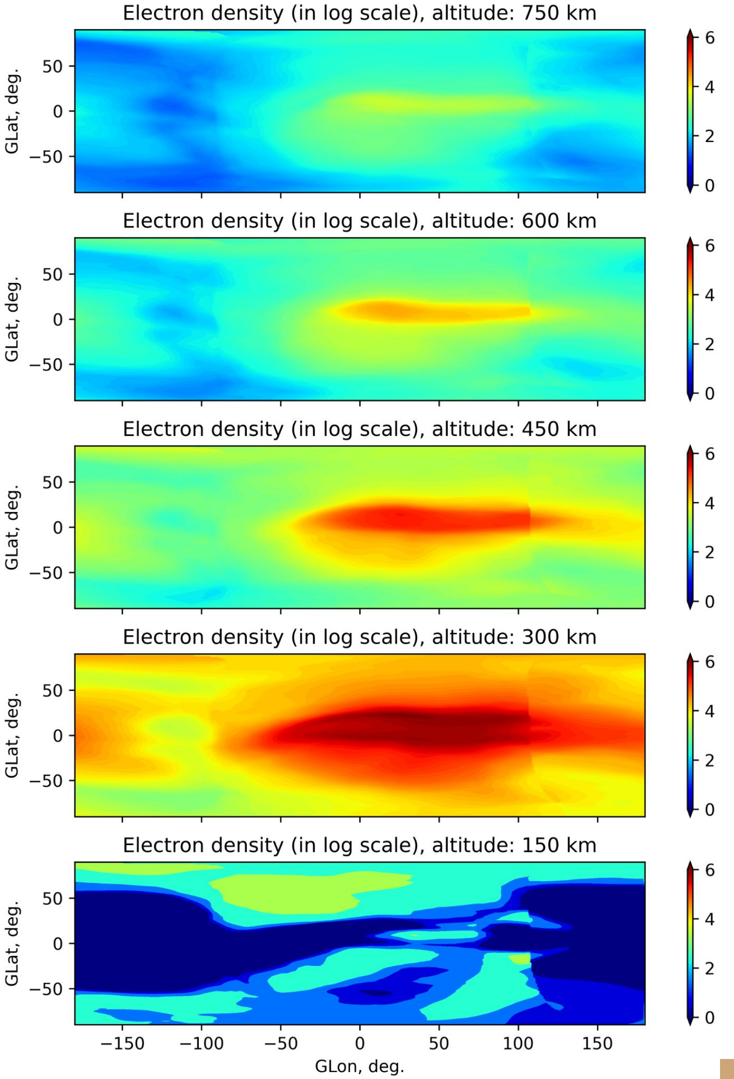
Target	Train Loss	Validate Loss
NmF2	0.17	0.31
hmF2	0.26	0.39
Hs	0.40	0.53
grad Hs	0.54	0.77

- Above are some preliminary results; further investigation needed.

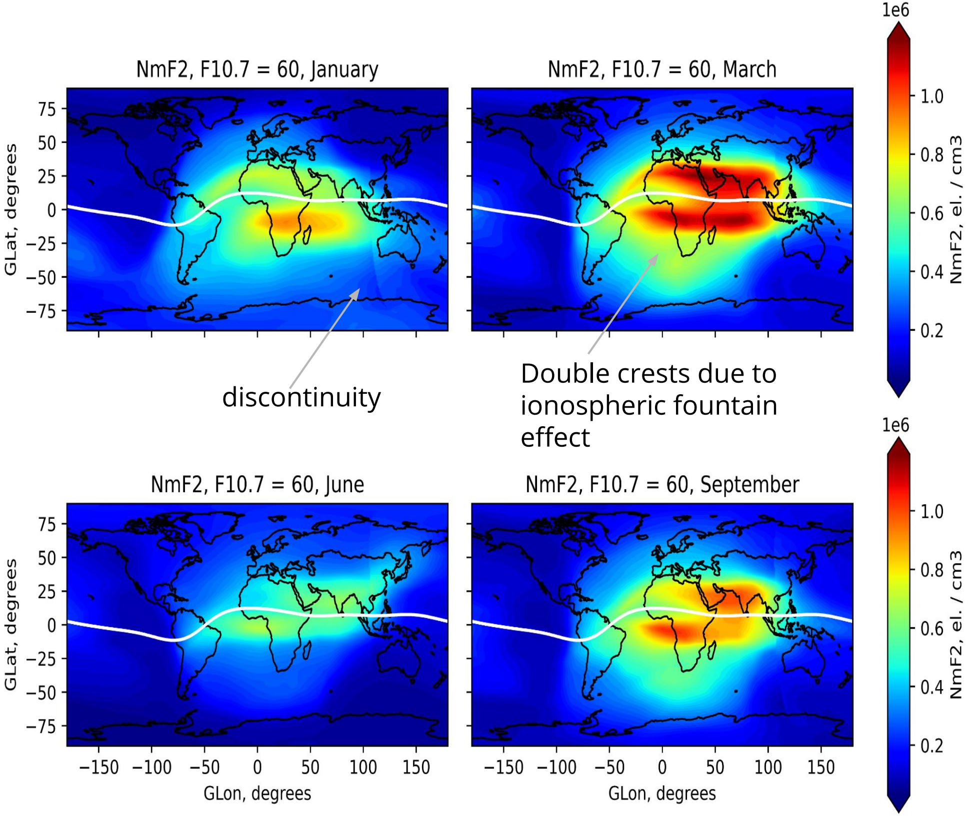
Qualitative Analysis

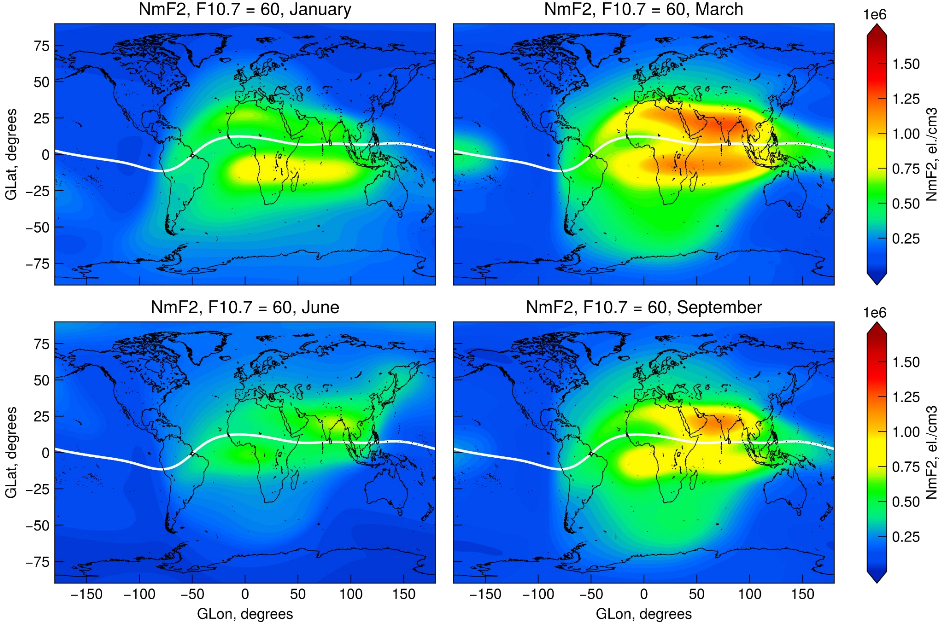
Altitudinal variation



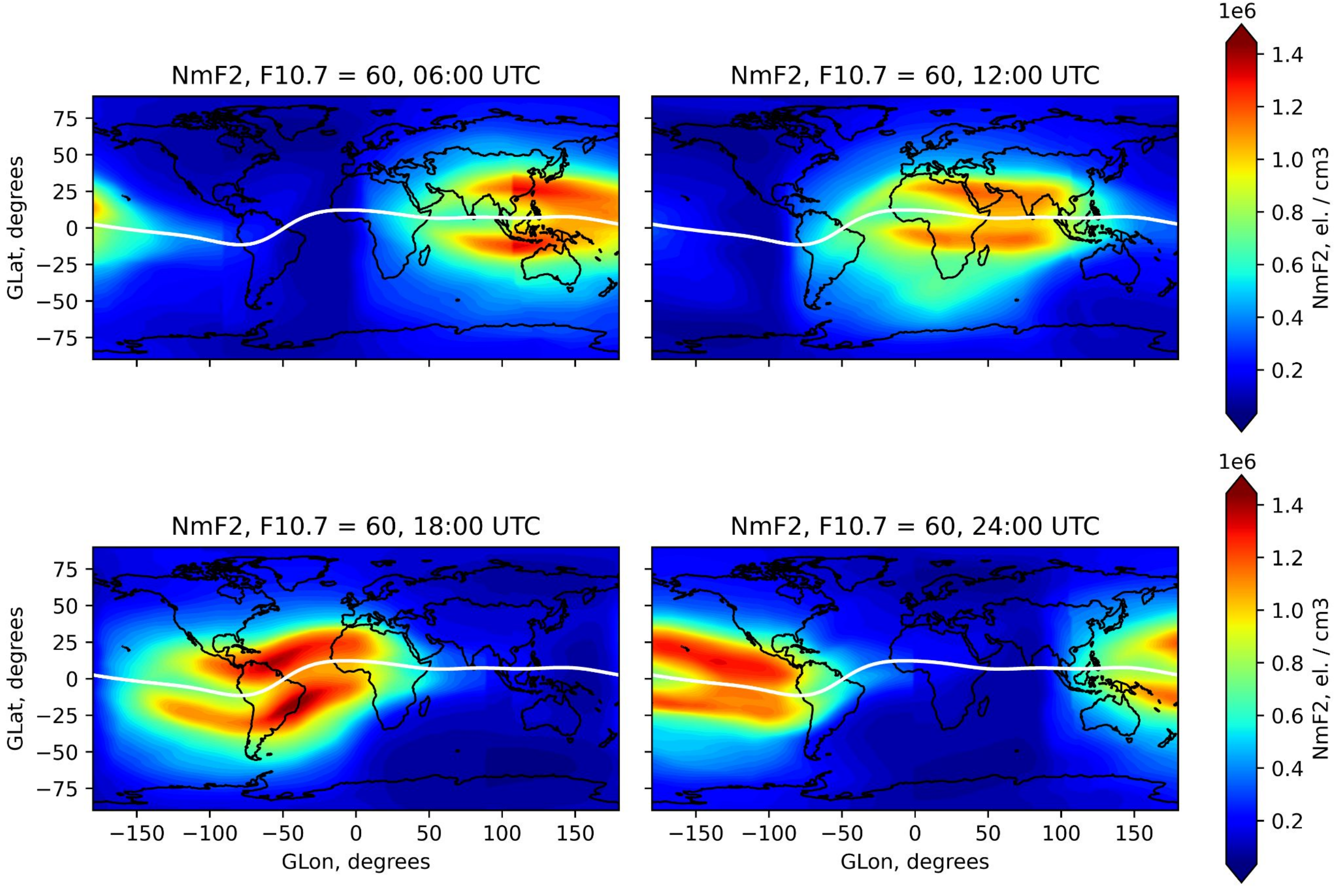


Seasonal variation of NmF2 (MLP-P2P)



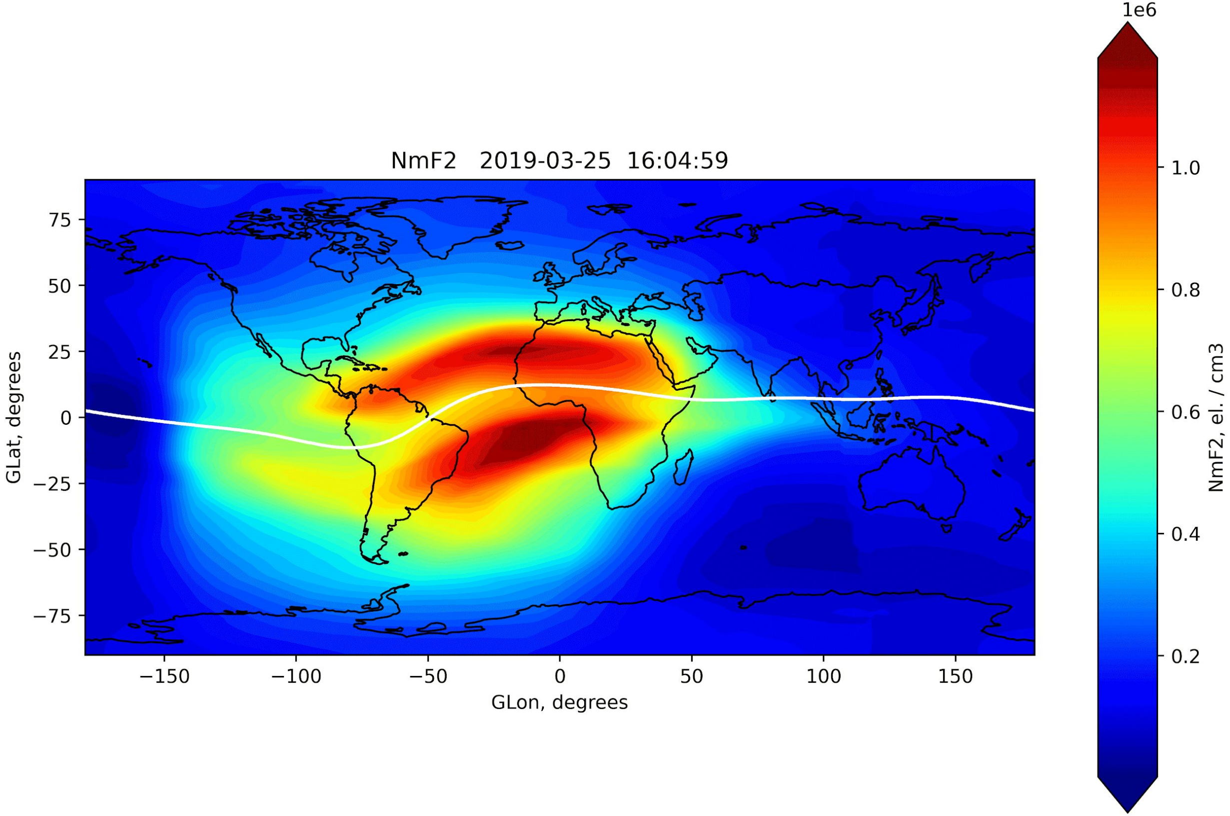


Diurnal variation of NmF2 (MLP-P2P)



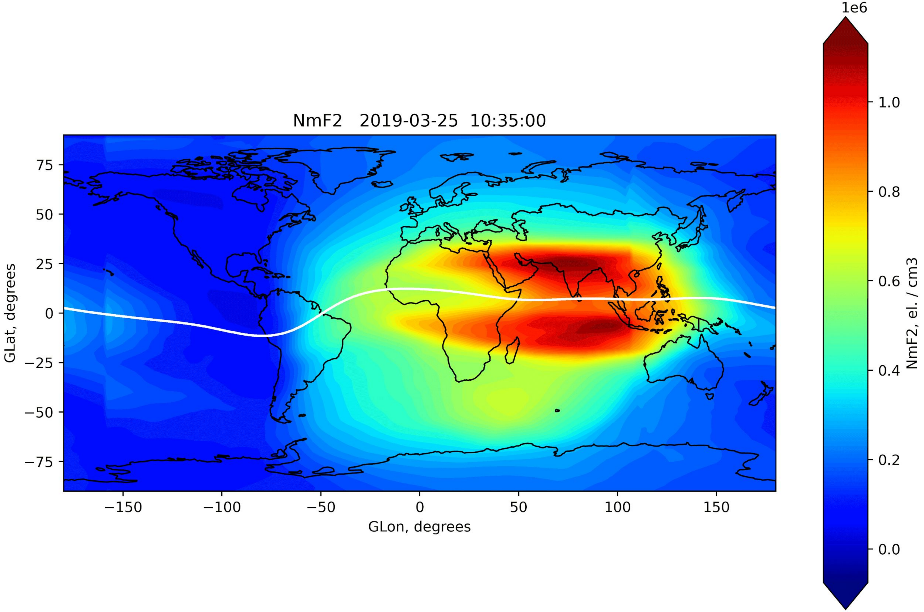
Variation within 2h (LSTM Seq2Seq)

From 16:00 to 18:00

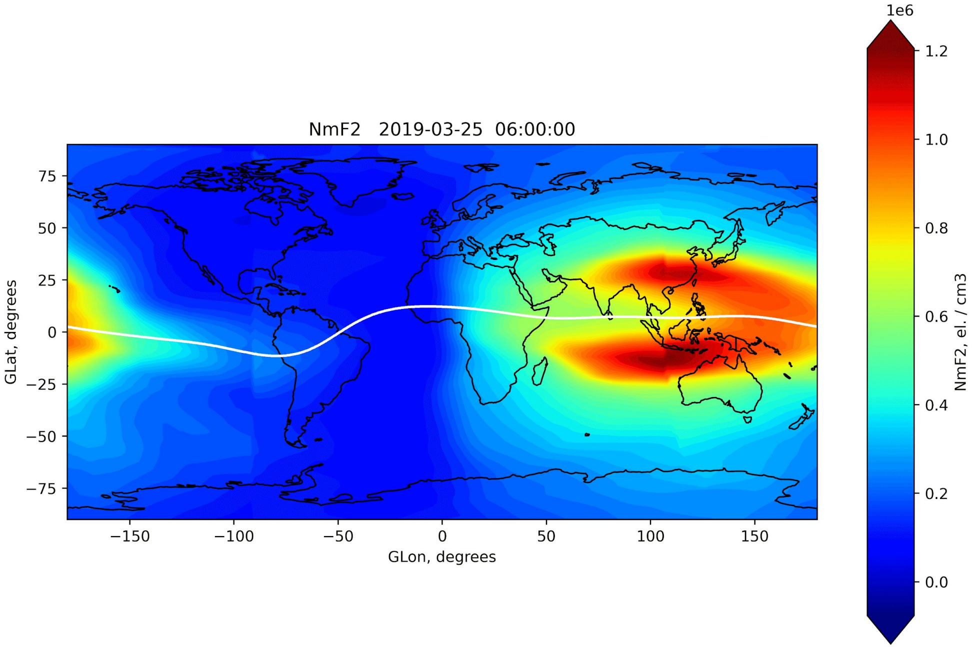


Variation within 2h (LSTM Seq2Seq)

From 10:30 to 12:30



From 6:00 to 8:00



Hypothesis: prediction improves within time series

Conclusion

- **Multi-task learning**: predict 4 ionospheric density parameters

- **Spatio-temporal models** to capture different variations

- **Encoder-decoder** architecture: Point2Point, Seq2Point, Seq2Seq
  - Sequence information in both input & target improves prediction
  - Best results achieved by Transformer model

Thanks for your attention!