This document is the code instruction manual for the paper "**RA-ConvLSTM: Recurrent-architecture attentional ConvLSTM networks for prediction of global Total Electron Content.**"

1. **Data**

The data used in this study is the CODG global ionospheric VTEC data published by the CODE analysis center, covering the period from 2008 to 2020. In October 2014, the temporal resolution of this data was changed from 2 hours to 1 hour. Therefore, after downloading the data, linear interpolation should be used to unify the temporal resolution to 1 hour.

Data download address：<http://ftp.aiub.unibe.ch/CODE/>

1. **Attentional ConvLSTM cell and RA-ConvLSTM model**

The **Model** folder contains two files:

1. **SAConvLSTM.py**: This code of the attentional ConvLSTM cell as described in the study.
2. **Encode2Decode.py:** This file contains the RA-ConvLSTM model architecture.

When running the program, ensure both Python files from the **Model** folder and the **main.py** file are placed in the same directory.



1. **Mian.py**

This folder contains two programs:

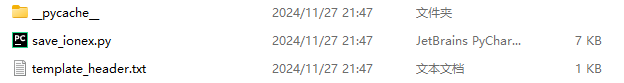
1. **main\_2015.py**: The main function for generating the test set for the year 2015.
2. **main\_2020.py**: The main function for generating the test set for the year 2020.

When running these programs, ensure that both functions are placed in the same directory as **SAConvLSTM.py** and **Encode2Decode.py**.



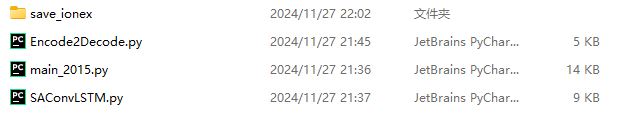
1. **Output final results**

The **save\_ionex** folder contains the program for generating predicted GIMs. When running the program, ensure that this folder is placed in the same directory as the aforementioned files.

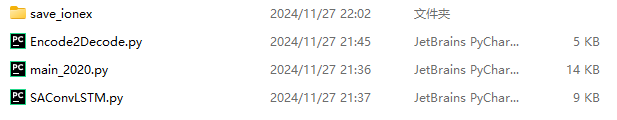


1. **The final run configuration**

1）test 2015 Year：

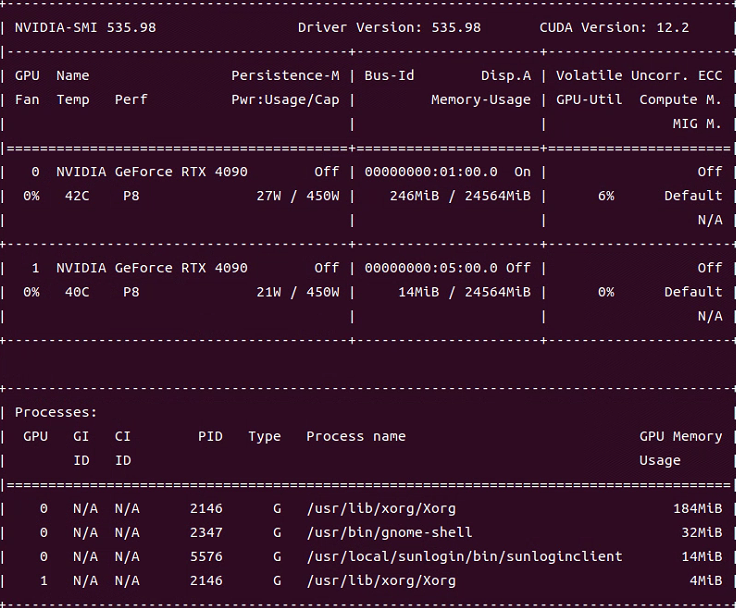


2）Test 2020 Year



1. **Others**
2. **Software and Development Environment**:
   1. This study is implemented using **PyTorch**, which is a popular deep learning framework for building and training neural networks.
   2. Other related functions are written using standard **Python syntax** to handle various data processing and utility tasks.
   3. The development platform for running and testing the models is **PyCharm**, a widely-used integrated development environment (IDE) for Python, which offers robust features for code editing, debugging, and project management.
3. **Hardware Configuration**:

Due to the large size of the model, the experiments in this study are conducted on **two 4090 GPUs**, this setup utilizes **dual-GPU parallel training**.



3）You can find all additional dependencies required for the setup in the **requirements.txt** file located in this folder.