The FTIM program consists of three key modules: Encoding_Module, Imputation_Module, and Reconstruction_Module. The Encoding_Module is responsible for encoding the data, converting raw latitude and longitude information into a form of changing distance, and transforming absolute altitude values into relative altitude values. The Imputation_Module seeks to precisely impute missing values in trajectory data, ultimately outputting complete and continuous trajectory data in encoded format. The Reconstruction_Module then feeds on the output from the Imputation_Module, restoring it to the original trajectory format. The main operational environment of the FTIM program encompasses the following aspects.

Python	3.10
json5	0.9.25
h5py	3.11.0
pytorch-cuda	11.6

(1) Encoding_Module

This module primarily extracts information from raw JSON - formatted data and converts it into a format compatible with the Imputation_Module. Executing main.py accomplishes the data processing. The OpenSky data is stored in the **Opensky_Original_Data** folder. Note that due to storage limitations, only partial data samples are provided here; the complete dataset is accessible on the OpenSky official website.

To enhance readers' understanding of the data processing procedure, several intermediate datasets are preserved during the model's operation. The **Data** folder contains the raw data directly extracted from JSON files, while the **Data_Diff** folder holds the preliminarily transformed data. Subsequently, the processed data is divided into training and testing sets, which are stored in the **Opensky_Original_Data** and **Opensky_Test_Data** folders, respectively. Finally, the training and testing sets, saved in the **Train_Test_H5_Data** folder in the h5 format, are ready for use by the Imputation_Module.

(2) Imputation_Module

This module focuses on imputing missing values in encoded data to generate complete coded trajectory data. Before starting, move the data generated by the Encoding_Module to the **Train_Test_H5_Data** folder. Then, run main.py to initiate model training and testing. The trained model is saved in the **Model_Saved** folder, and test results are stored in the **Result_Write** folder for subsequent use by the Reconstruction_Module.

(3) Reconstruction_Module

This module specializes in restoring the imputed results from the Imputation_Module to trajectory data format. First, copy the data generated by the Imputation_Module to the **Result_Write** folder. Then, execute main.py to transform the imputed data into trajectory format, with the final trajectory data saved in the **Final_Data** folder.