README File - Instructions for Running the Codes

This package contains the following elements:

1. A code to generate the training and test data based on the ETAS model for earthquake seismicity. This code generates the nowcasting time series that has been used on real data.

2. A science transformer deep learning code that was written with the help of ChatGPT using the typical PyTorch methods. We are calling it QuakeGPT

To run these codes do the following.

1. To generate the training and test data, type at the prompt:

```
>> python Write_ETAS_Data.py
```

You will get choices 1-3, generate training data only, test only, or both. You may wish to keep a test set of data and generate new training sets to continue improving the model.

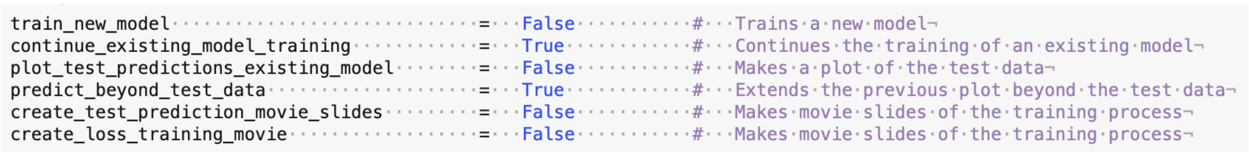
Note that running these codes will result not only in the data files being written, but also 4 matplotlib plots for 1) the nowcast curves; 2) the magnitude-frequency relation; 3) the magnitude-time plot: and 4) an Omori aftershock plot.

Note that these plots must be manually closed in order to finish the run. You may wish to change this, or eliminate them altogether.

2. To run the transformer code, type at the prompt:

```
>>python QuakeGPT.py
```

This begins the training or testing cycle. You can choose what you want the code to do:



```
train_new_model                         =   False          #   Trains a new model
continue_existing_model_training        =   True           #   Continues the training of an existing model
plot_test_predictions_existing_model    =   False          #   Makes a plot of the test data
predict_beyond_test_data                =   True           #   Extends the previous plot beyond the test data
create_test_prediction_movie_slides     =   False          #   Makes movie slides of the training process
create_loss_training_movie              =   False          #   Makes movie slides of the training process
```

Note that for the test plots, the green dot is the first predicted value in the prediction window, the red dot is the last predicted value in the window, and the light blue line connects the two.

Note that this is a sliding window predictor. Given the recent history prior to the current data point, ("window_size"), the transformer predicts the future data in the future interval ("future_window_size"). At the moment, the code does not use batching, although this could be added later.

I also plan to add the USGS data soon as test data. And also another idea for an indicator of approaching large earthquakes that I have in mind.