Data Forecasting of Gyro-Landau Extended Fluid Code Using Neural Networks

Wang Lian[1)], Li Hui[1,2)], Wang Tianbo[1)], Li Jiquan[1)]

1) Southwestern Institute of Physics, Center for Fusion Science

2) Dalian University of Technology, Physics College

4th IAEA Technical Meeting on Fusion Data Processing, Validation and Analysis

CONTENTS

- Background
  - What is Extended Fluid Code
  - Motivation: Why we need to use neuron network
- Flux Prediction Model
  - Model structure and evaluation
- Surrogate Model
  - Data processing
  - Model structure, traning, evaluation
  - Self-iteration
- Summary

Part Ⅰ: Background

Extended Fluid Code

- Extended Fluid Code (ExFC)
  - Based on multi-scale turbulence physics on Tokamaks
  - Using finite difference method to sovle 5-field electrostatic equation
  - Able to simulate distabilities including TM, ITG, TEM and KBM

$$\frac{dn_e}{dt} = -\omega_{dte}(n_0\phi - T_{e0}n_e - n_0T_e) + D_n\nabla_\perp^2 n_e$$

$$\frac{dT_e}{dt} = -T_{e0}\omega_{dte}\left[(\Gamma-1)\left(\phi + \frac{T_{e0}}{n_0}n_e\right) + (2\Gamma-1)T_e\right] - (\Gamma-1)\sqrt{\frac{8m_eT_{e0}}{m_i\pi}}|\nabla_\parallel|T_e + D_{Te}\nabla_\perp^2T_e$$

$$\frac{d\Omega}{dt} = aT_{i0}\left(\frac{\nabla_r n_0}{n_0} + \frac{\nabla_r T_{i0}}{T_{i0}}\right)\nabla_\theta\nabla_\perp^2\phi + af_c\frac{\nabla_r n_0}{n_0}\nabla_\theta\phi - \nabla_\parallel v_\parallel + f_t\omega_{dte}\left(\phi - T_e - \frac{T_{i0}}{n_0}n_e\right)$$
$$+\omega_d\left((1+f_c)\phi + T_i + f_t\frac{T_{i0}}{n_0}n_e\right) + D_U\nabla_\perp^2\Omega$$

$$\frac{dv_\parallel}{dt} = -\nabla_\parallel T_i - f_t\frac{T_{i0}}{n_0}\nabla_\parallel n - (1+f_c)\nabla_\parallel\phi + D_v\nabla_\perp^2 v_\parallel$$

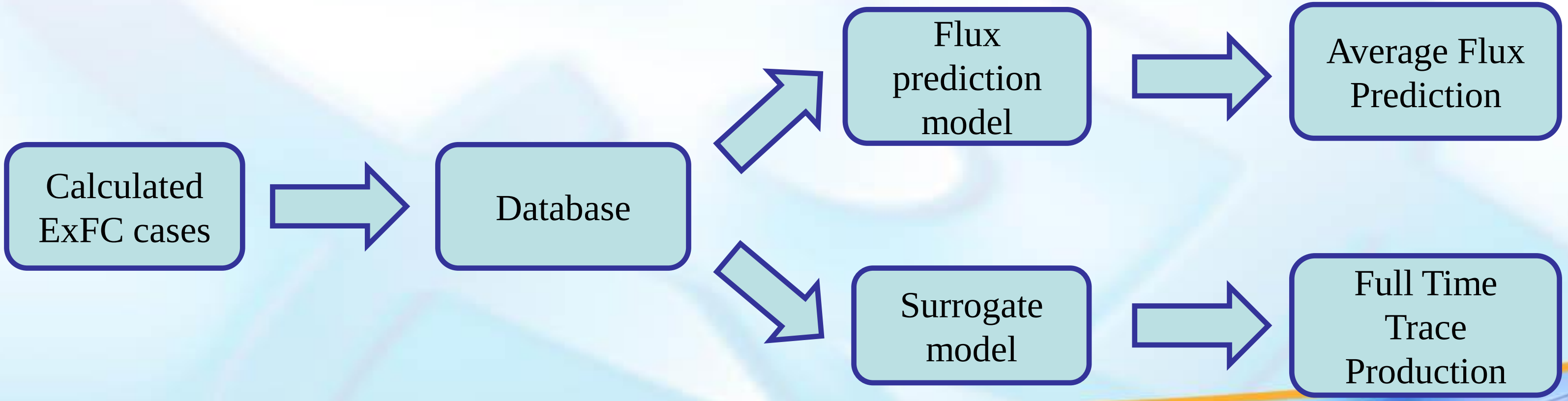
$$\frac{dT_i}{dt} = -(\Gamma-1)\nabla_\parallel v_\parallel + T_{i0}\omega_{di}\left[(\Gamma-1)\left(f_c\phi + f_t\frac{T_{i0}}{n_0}n\right) + (2\Gamma-1)T_i\right] - (\Gamma-1)\sqrt{\frac{8T_{i0}}{\pi}}|\nabla_\parallel|T_i + D_{Ti}\nabla_\perp^2T_i$$

J. Weiland, et al. NF 29 (1989)

Extended Fluid Code

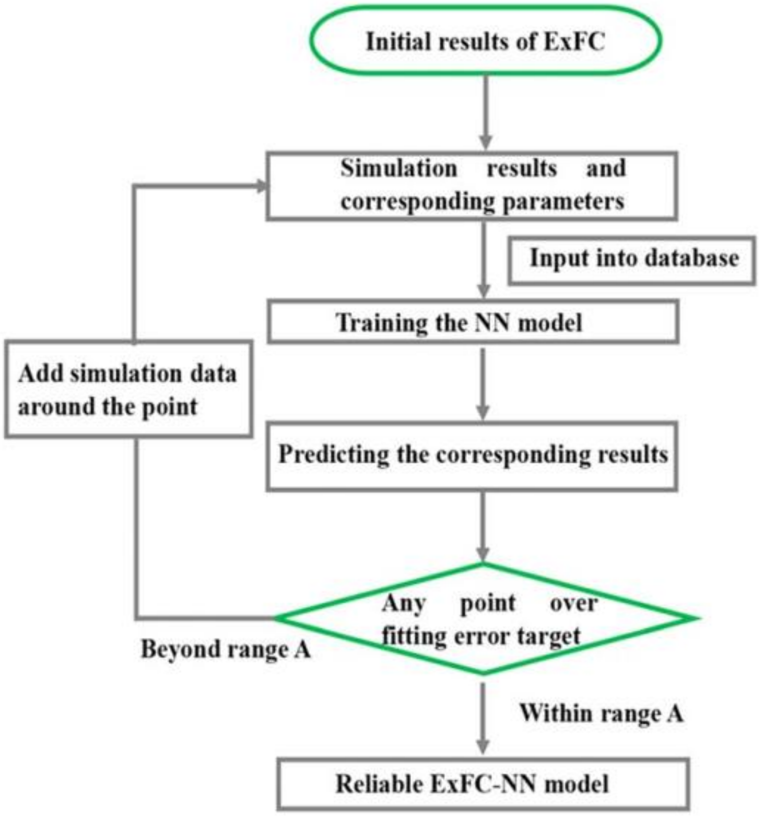
- Time cost
  - Grid size: $128 \times 256 \times 128$
  - A time slice cost: 2 mins
  - A case cost: around 4 hours
  - Parameter scanning and threshold determination: A few days

Using neuron networks to shorten the time cost!



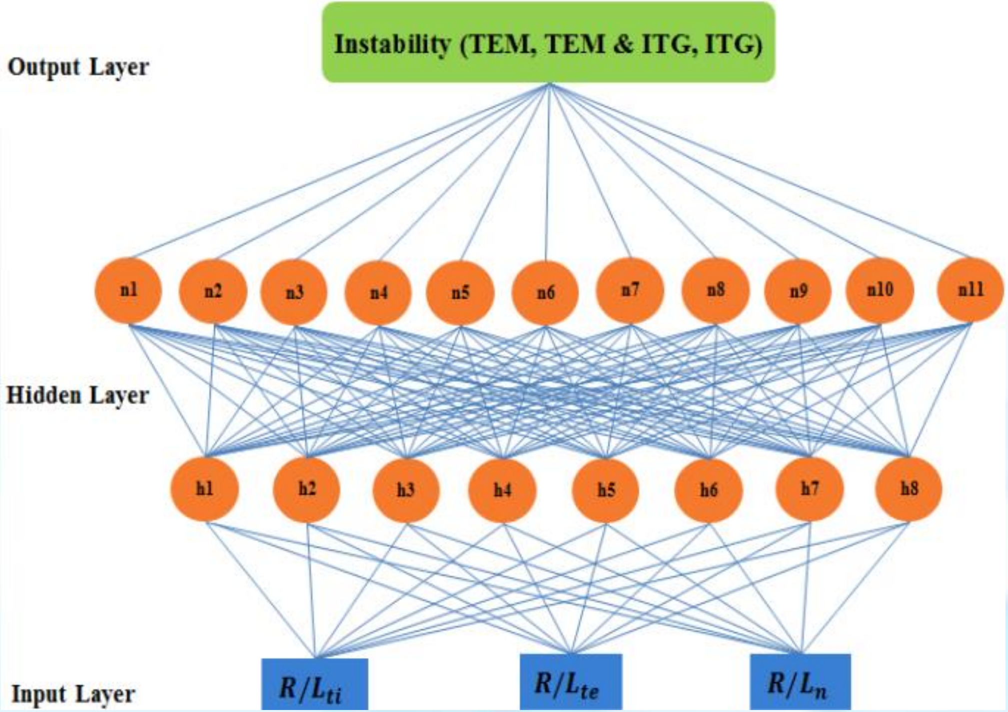
Part II: Flux Prediction Model

Model Structure

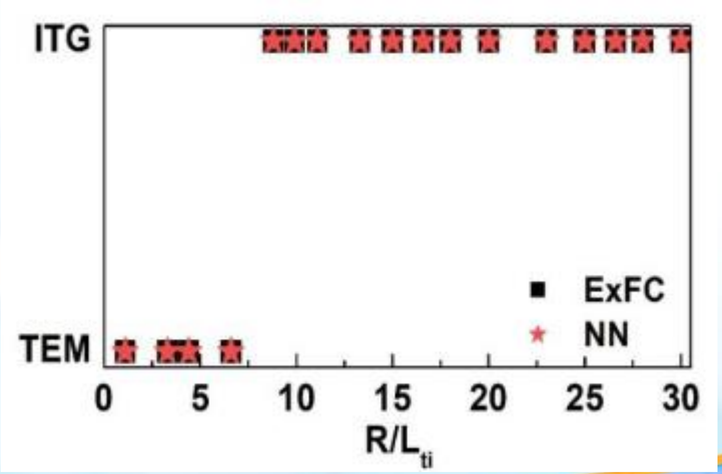


Model	layer	Number of neurons	Purpose
1	Input	3	Predict tuebulence type
	Dense	8	
	Dense	11	
	Output	1	
2	Input	7	Predict radial density profile
	Dense	40	
	Dense	40	
	Output	1	

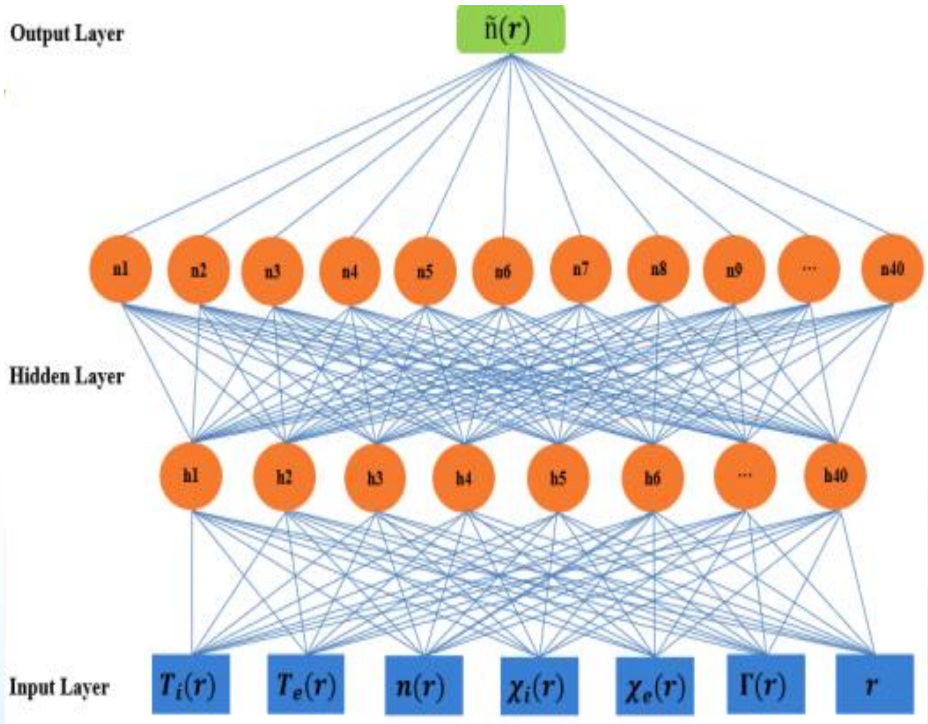
Predict Turbulence Type

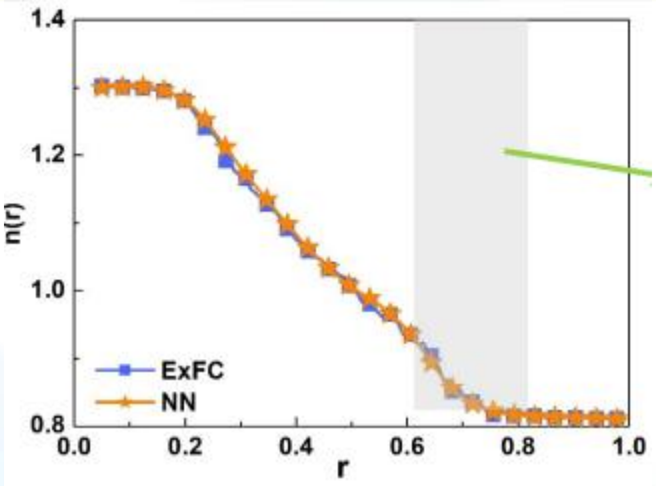


Main Turbulence Type	TEM	ITG&TEM	ITG
Accuracy	0.994	0.910	0.997



Predict Radial Density Profile





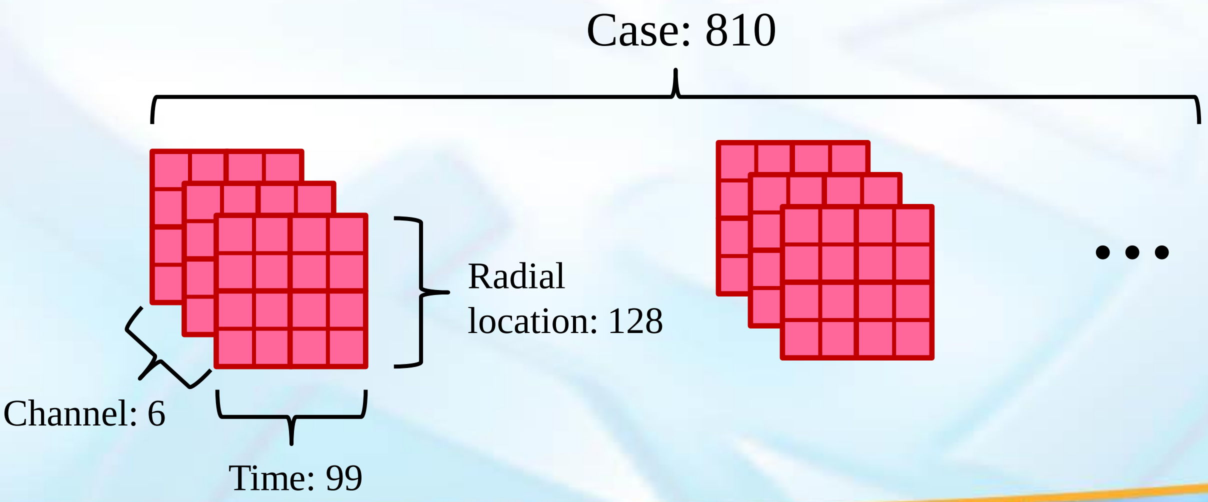
Succefully predicted the structure changed by fluctuation

- $RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(n_n^{NN} - n_n^{ExFC})^2} = 0.0088$
- Relative error less than 0.7%

Part III: Surrogate Model

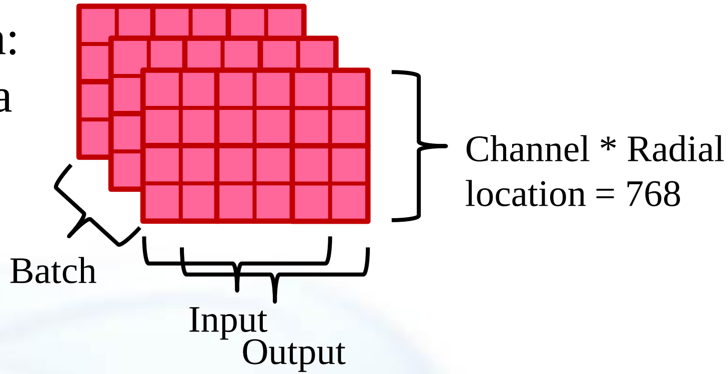
Data Processing

- Data structure: [810, 99, 6, 128]
- [case, time, channel, radial location]
- Density flux, electron temperature flux, ion temperature flux, ion temperature, electron temperature, density



Data Processing

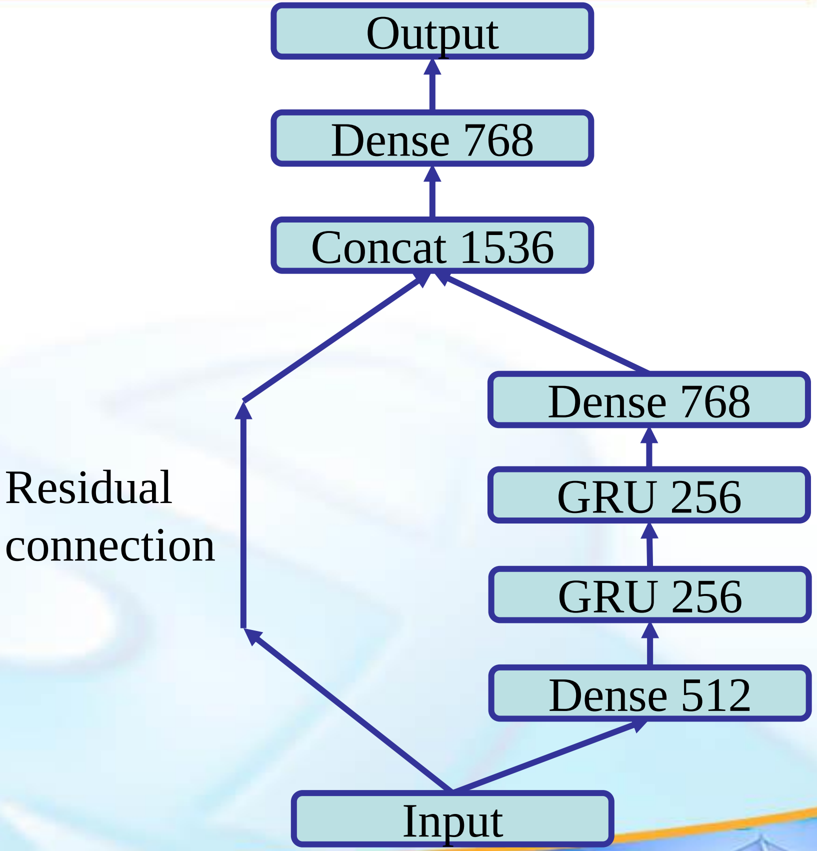
- Problem analysis
  - Multivariate time-relevant regression problem: using data of former time slices to predict data of next time slices
  - Using data of 5 time slices to predict data of next 5 time slices shfited by 1 time slice
- Recurrent neuron network: Gated Recurrent Unit
- Database rearrange
  - Merging the channel dimensionn and radial location dimension into a single dimension of size 768
  - Using a window of size 5 to cut the time dimension: [76140, 5, 768]
  - Training set: [60912, 5, 768]
  - Validation set: [15228, 5, 768]



- Input
  - [None, 5, 768]
  - [batch_size, time_step, channel * location]
- Output
  - [None, 5, 768]
  - [batch_size, time_step, channel * location]

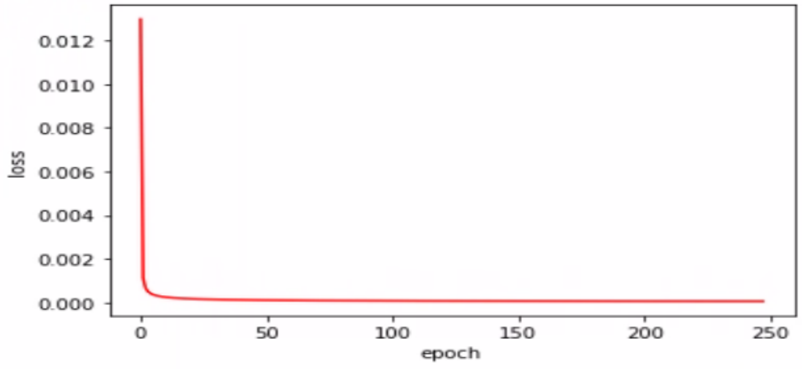
Model Structure

Layer	Number of neurons/units	Output Shape
Input	768	[None, 5. 768]
Dense	512	[None, 5, 512]
GRU	256	[None, 5, 256]
GRU	256	[None, 5, 256]
Dense	768	[None, 5, 768]
Concat	/	[None, 5, 1536]
Dense	768	[None, 5, 768]

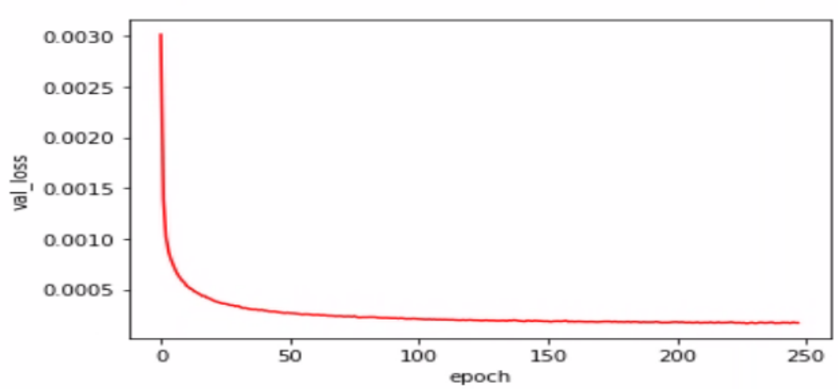


Training Process

- Batch size:32
- Optimizer: Nadam
- Learning rate: 1e-5
- Loss function: mean square error
- Callback: early stop



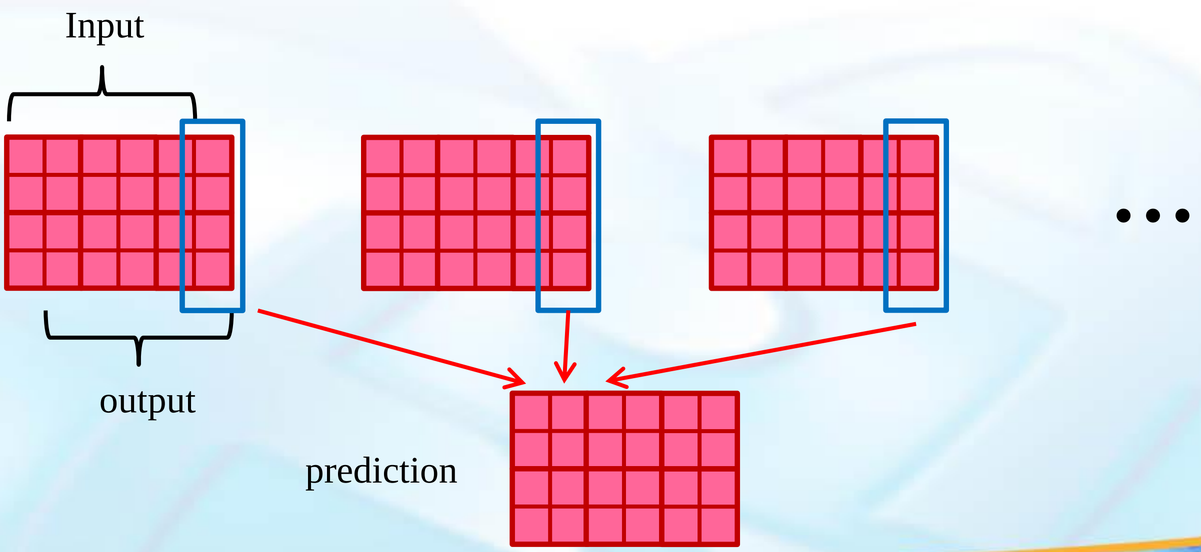
Epoch	Training loss	Validation loss
248	4.9652e-5	1.6971e-4



- Traning loss is 3 times less than validation loss
- Overfitting

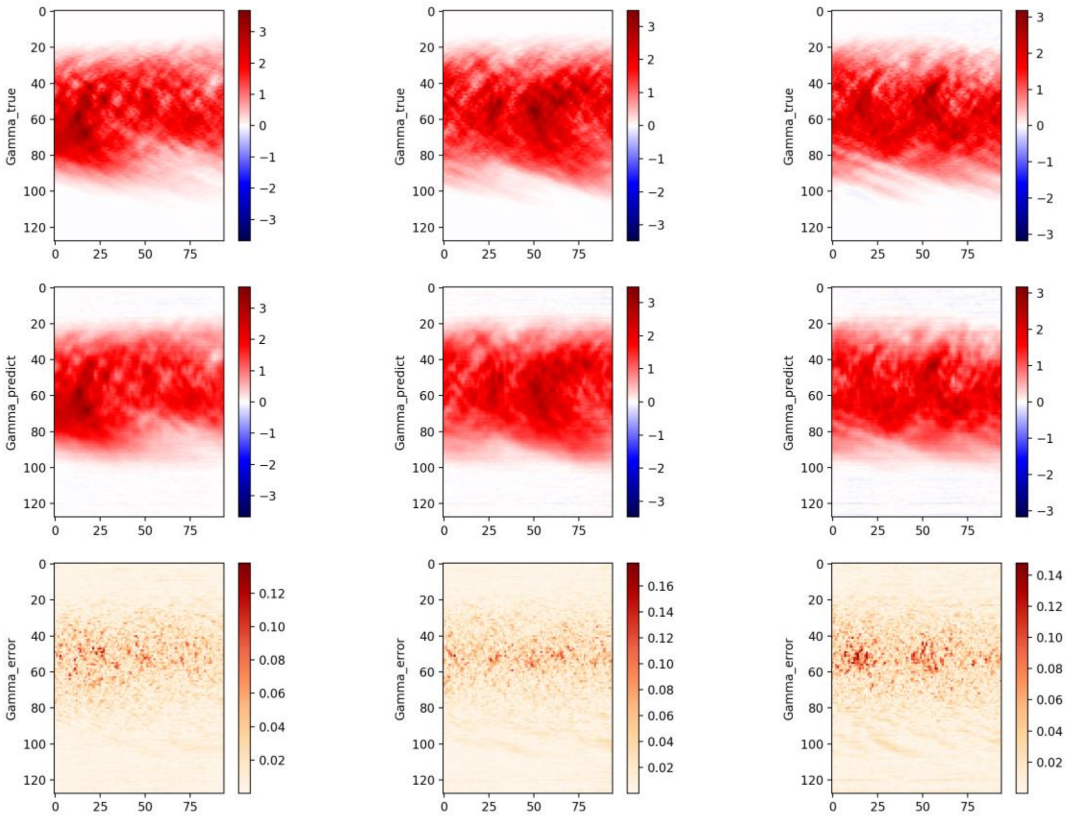
Model Evaluation

- Evaluation method
  - Using the trained model to predict adjacent samples
  - Merge the last time slice of each predicted results
- Relative error: $\left|\frac{n_{NN}-n_{ExFC}}{n_{ExFC}}\right|$



Model Evaluation

Case 1 Case 2 Case 3

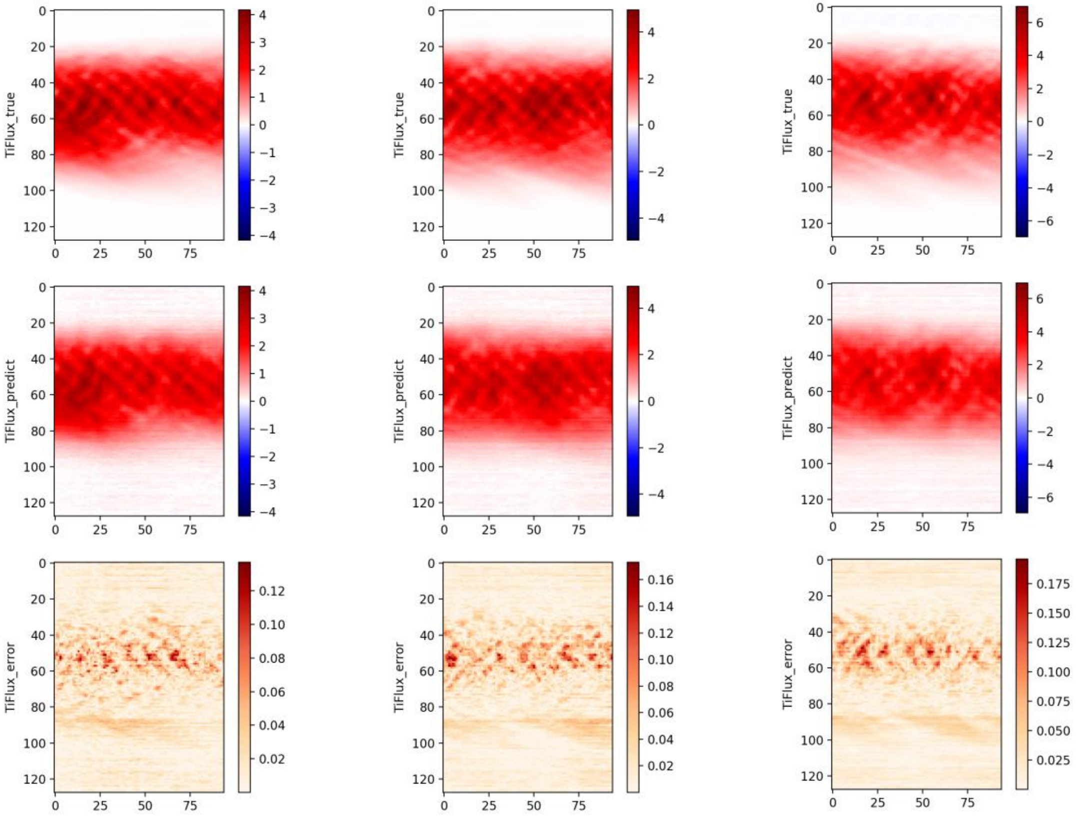


- First row: density flux given by ExFC
- Second row: density flux given by neuron networks
- Third row: relative error

- A total relative error less than 16%
- Errors are mainly centered at areas where fine structures of turbulence exist

Model Evaluation

Case 1 Case 2 Case 3



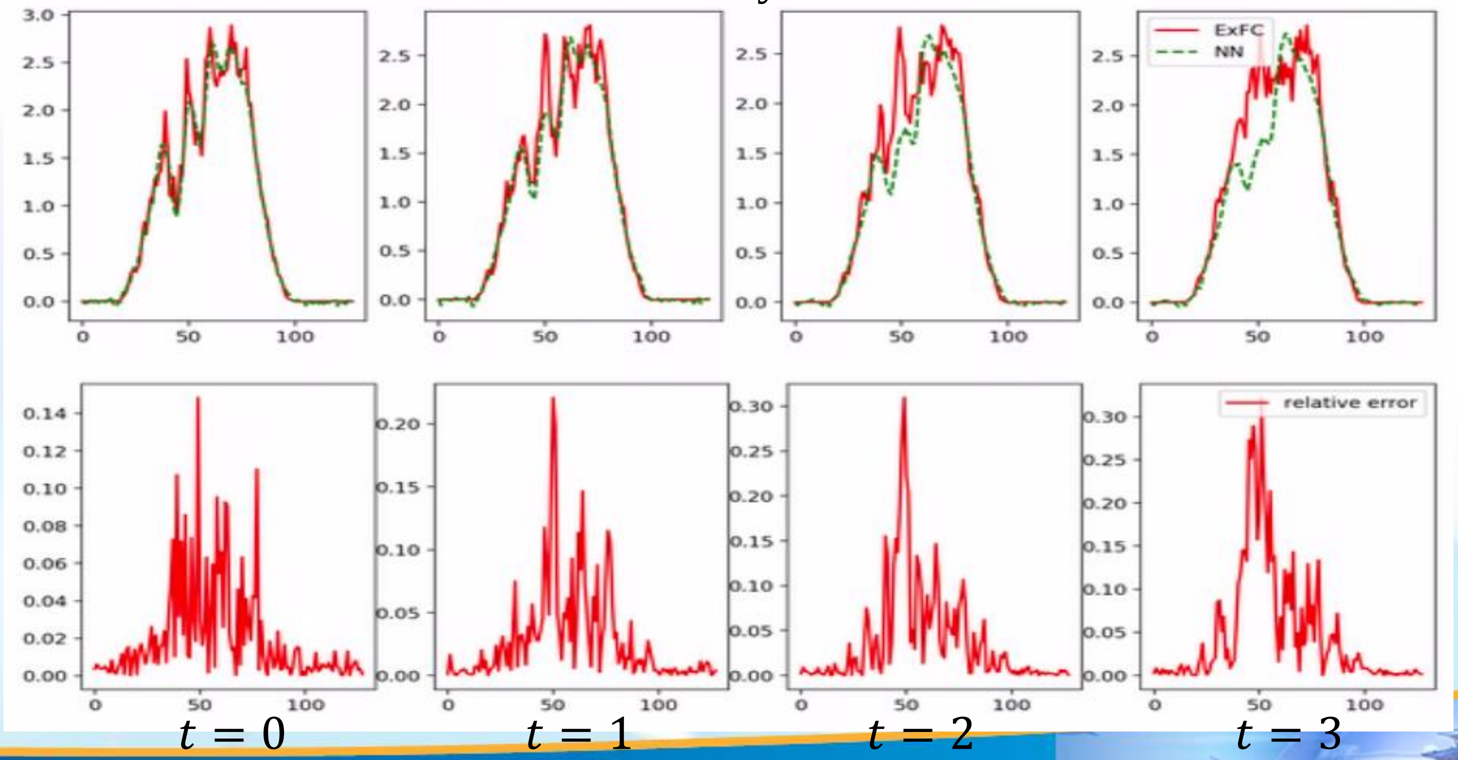
- First row: ion temperature flux given by ExFC
- Second row: ion temperature flux given by neuron networks
- Third row: relative error

- A total relative error less than 17.5%
- Errors are mainly centered at areas where fine structures of turbulence exist

Self-iteration

- Using output as input to call the predict method iteratively

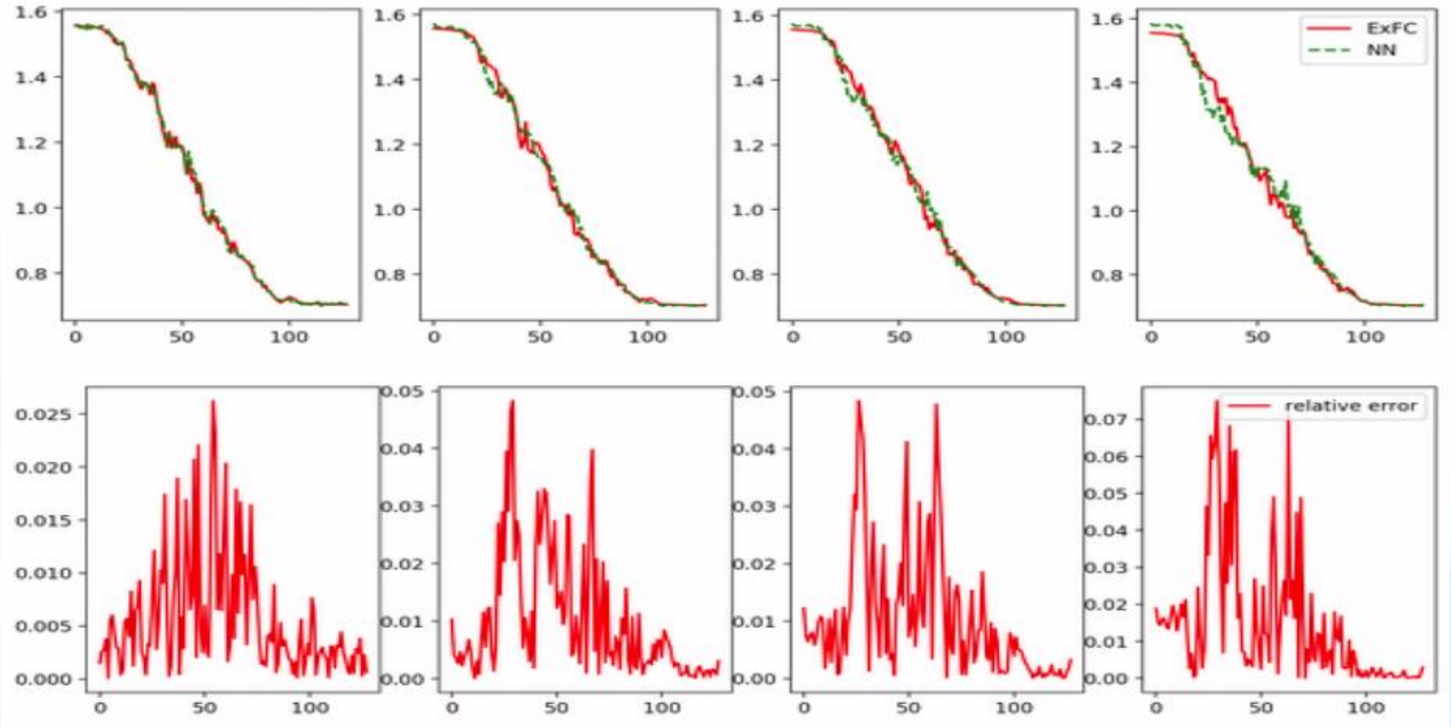
Density Flux



China National Nuclear Corporation

Self-iteration

Density



$t = 0$ $t = 5$ $t = 10$ $t = 20$

Self-iteration

- Flux
  - NN model is only able to forecast data of 1-3 time slices ahead within a relative error of about 20%

- Density/Temperature
  - NN model is able to forecast data of 10-15 time slices ahead within a relative error of about 5%

- NN model is suitable for density/temperature data forcasting

- Time cost for NN model
  - Seraval microseconds for a time slice
  - If run NN model and ExFC alternatively, about 50-75% less time will be consumed for a single case (if we neglect the error caused by NN model)

Part IV: Summary

Summary

- Flux prediction model
  - Sevaral flux prediction neuron networks have been developed to predict the type of turbulence and different macro parameters based on the database of ExFC.
  - These models achieve a high performance on prediction with a relative error of less than 5%.

- Surrogate model
  - A recurrent neuron network has been established using the Gated Recurrent Unit.
  - This model is able to predict data of one step ahead with a relative error less than about 20%.
  - Some parts of turbulent structures are preserved, while other parts are smoothed, leading to information loss in the predicted data.
  - At this moment, the model can only reproduce 1 to 3 time slices of the original data given by ExFC through self-iteration with an acceptable relative error.

Thank you for your attention!