# A deep-learning code of beam orbit correction on CIADS superconducting linac

Peiyong Jiang

Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou

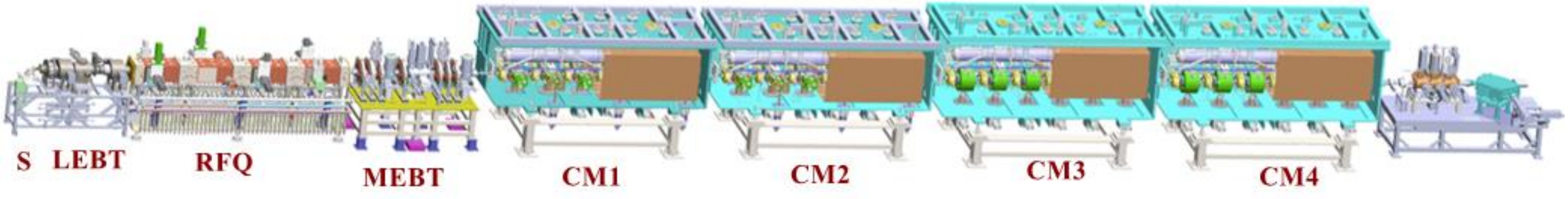
# Outline

- Introduction to CIADS superconducting linac
- Layout
- Beam orbit correction
- Thought source
- Code

# Introduction to CIADS superconducting linac

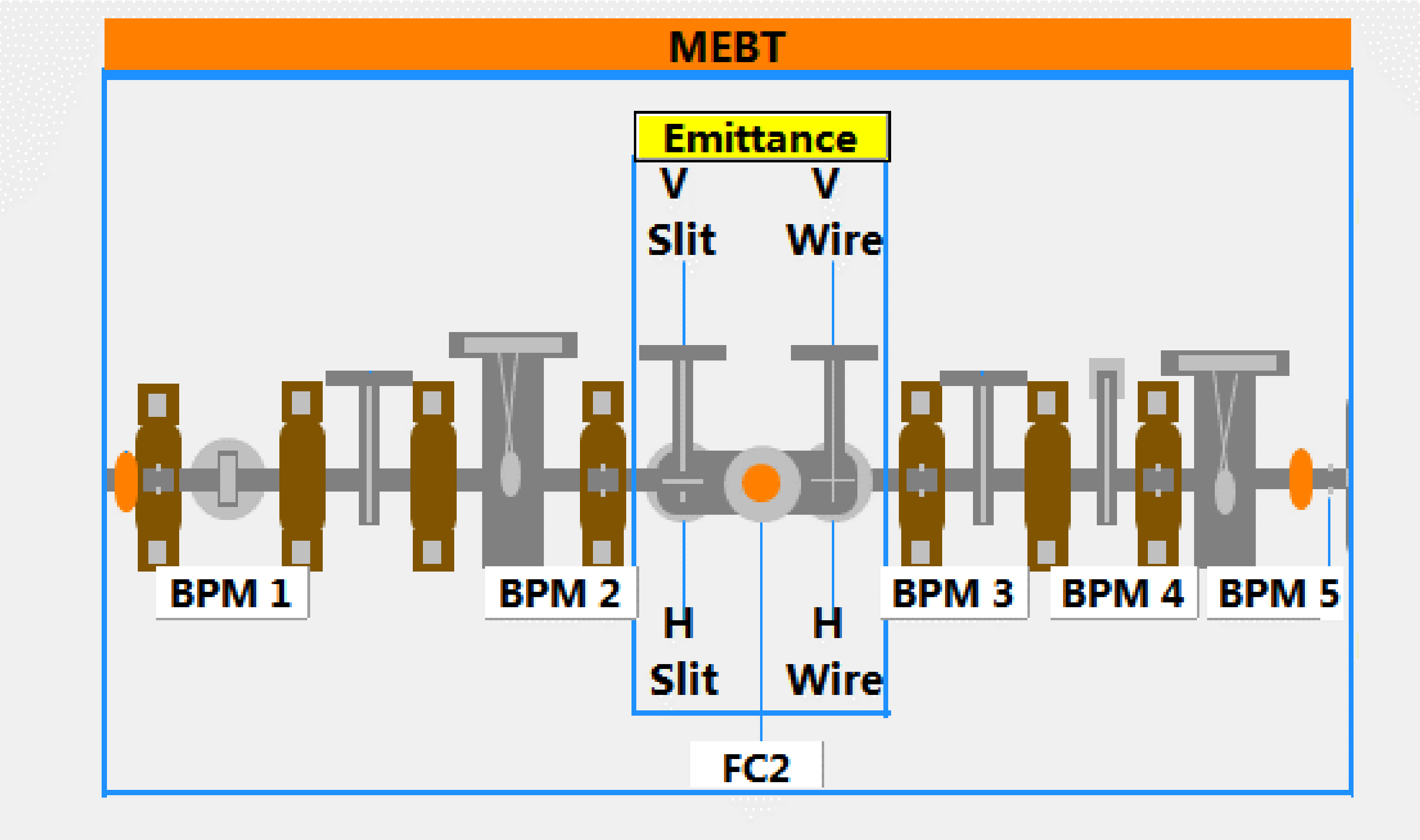
- Structure
  - Source-LEBT-RFQ-MEBT-CM1-CM2-CM3-CM4-Dumper
- Commissioning
  - Main Problem: Heat
  - Solution：
    - Lattice configuration
    - Beam orbit correction ☆☆☆☆☆

MEBT CM1



# Layout-MEBT

- 7 Quads
- 7 sets of correctors
  - 7 horizontal correctors
  - 7 vertical corrector
- 5 sets of BPMs
  - 5 horizontal BPMs
  - 5 vertical BPMs
- 5 sets of BPMs are arranged in the center of 1st , 4th , 5th , 7th quads and the end of MEBT respectively
- The 1st set of BPMs can not be corrected



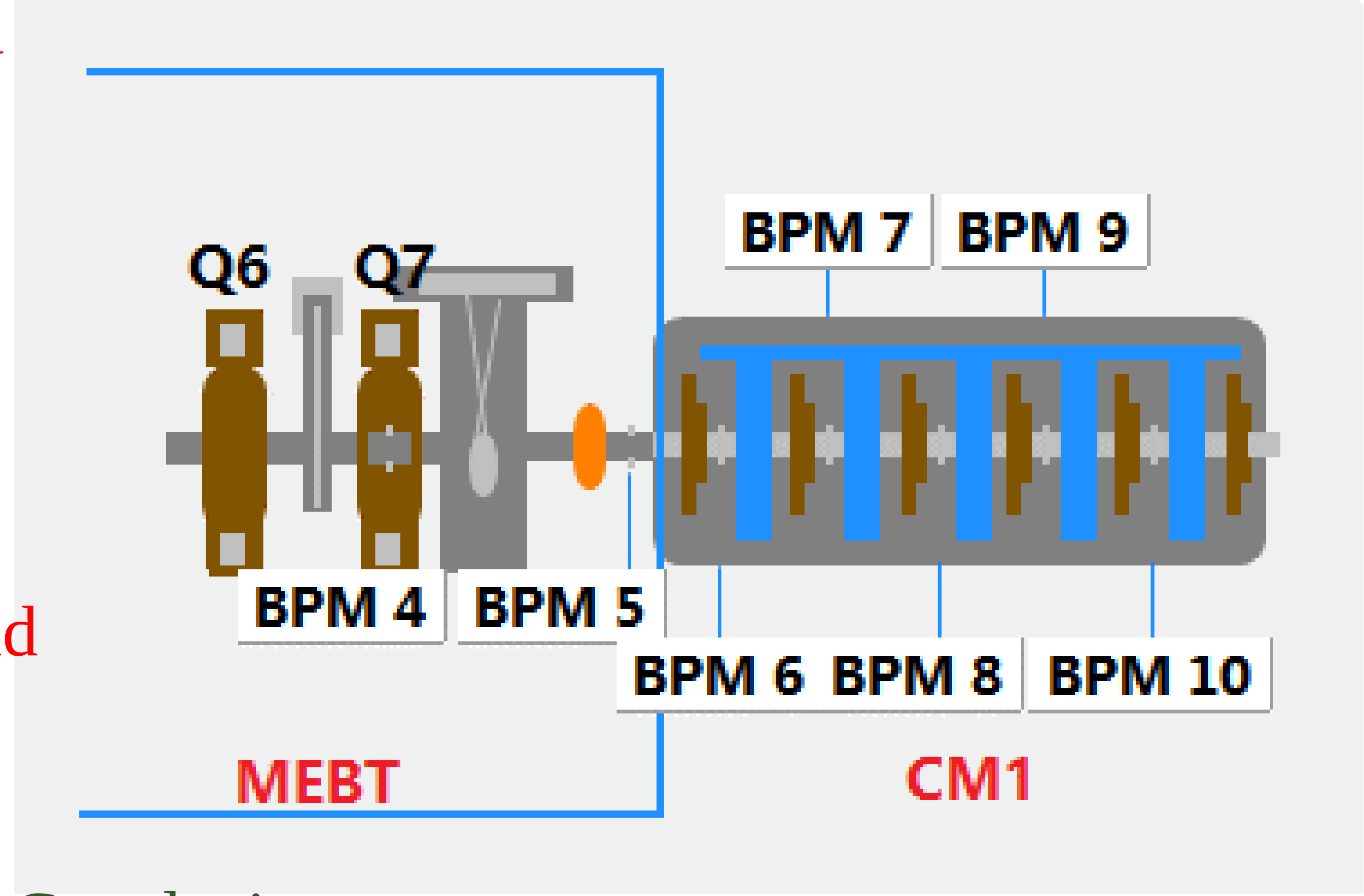

Conclusion:

7 horizontal correctors, 7 vertical correctors,

5 horizontal BMPs, 5 vertical BMPs.

# Layout-CM1

- 6 sets of solenoid and cavity
- 6 sets of correctors
  - 6 horizontal correctors
  - 6 vertical corrector
- 5 sets of BPMs
  - 5 horizontal BPMs
  - 5 vertical BPMs
- five sets of BMPs are behind the first five cavities
- Last set of correctors in MEBT is used to correct BPM6
- Last set of correctors in CM1 is not used




Conclusion:

6 horizontal correctors, 6 vertical correctors,

5 horizontal BMPs, 5 vertical BMPs.

# Beam orbit correction- Response Matrix Method

$$\Delta X = A \cdot \Delta I$$

$$\Delta X = (\Delta X_1, \Delta X_2, \dots, \Delta X_m)$$

$$\Delta I = (\Delta I_1, \Delta I_2, \dots, \Delta I_n)$$

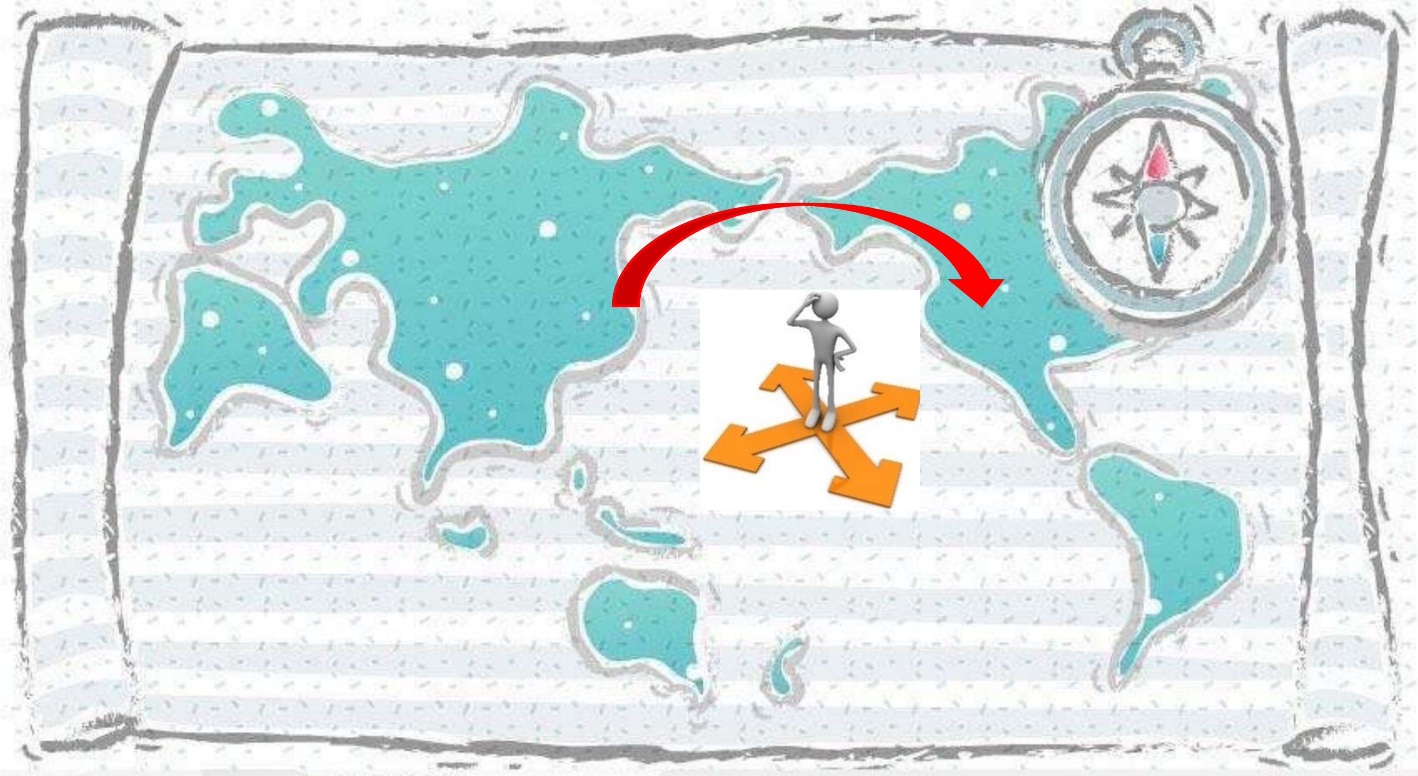
$$A = U \cdot \begin{pmatrix} \omega_1 & 0 & \cdots & 0 \\ 0 & \omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \omega_n \end{pmatrix} \cdot V^t$$

$$\Delta I = A^{-1} \cdot \Delta X = V \cdot \begin{pmatrix} 1/\omega_1 & 0 & \cdots & 0 \\ 0 & 1/\omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1/\omega_n \end{pmatrix} \cdot U^t \cdot \Delta X$$

- ΔX describes the desired correction at m BPMs
- ΔI is the excitation currents of n correctors, that we want to determine.
- SVD condition: m≥n

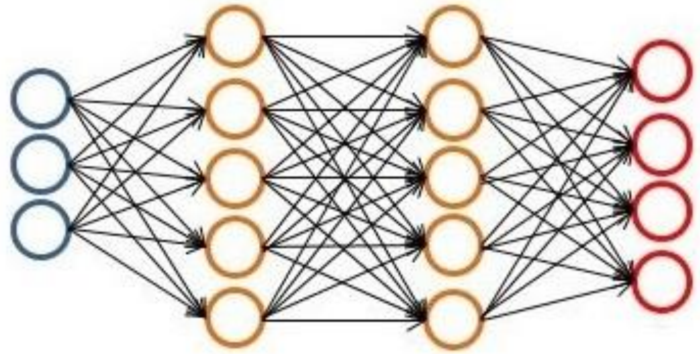
- w= $A^{-1}$
- ΔI =w· ΔX
  - ΔX: m×1
  - ΔI: n×1
  - B: n ×m
- To find matrix w

# Thought source - From Q-Learning to DQN

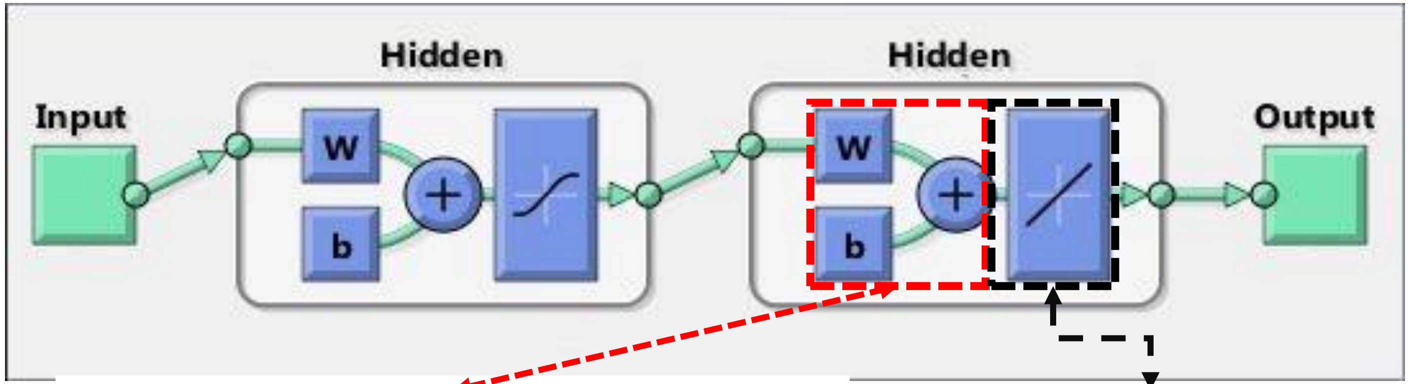


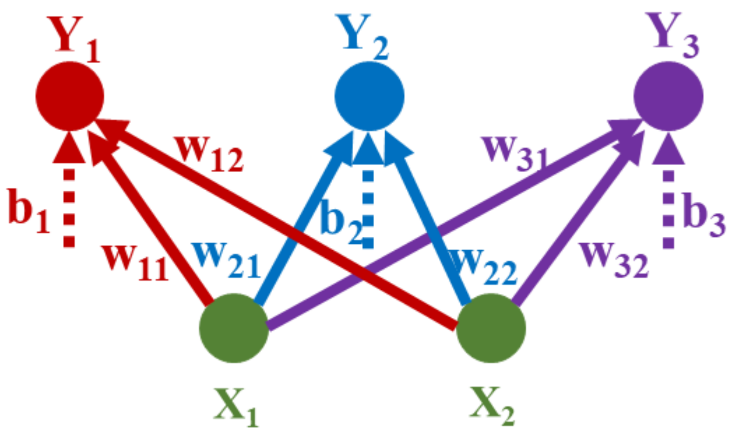
- Q-Learning
  - States: Crossing indexs
  - Actions: Moving directions
  - Q=Q(state,action)
  - Q-Table
- DQN:
  - Q-Table → NN ☆☆☆☆☆
  - Fixed weights
  - memory recall

| States | ← | → | ↑ | ↓ |
|---|---|---|---|---|
| Crossing1 | Q11 | Q12 | Q13 | Q14 |
| Crossing2 | Q21 | Q22 | Q23 | Q24 |
| … | … | | | |
| CrossingN | QN1 | QN2 | QN3 | QN4 |



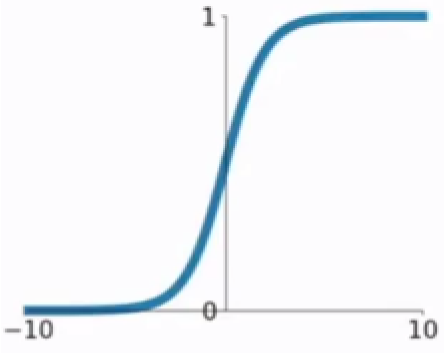
# Thought source - Neural Network(NN)



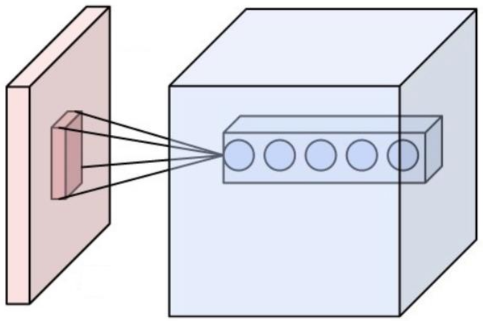

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix} \cdot \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

Sigmoid

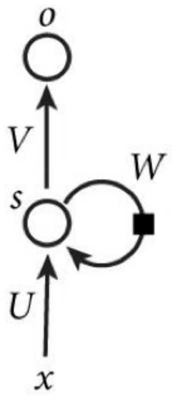



$$\sigma(x) = \frac{1}{1+e^{-x}}$$

- Input - Hidden * n – Output
- Y=w·X+b
- Z=ActivationFunction(Y)
- Popular NN:
  - FC
  - CNN: shared-weights in space
  - RNN: shared-weights in time



CNN




RNN

# Thought source – From response matrix to NN

**What would happen in beam orbit correction**
**if we replace response matrix with neural network？**

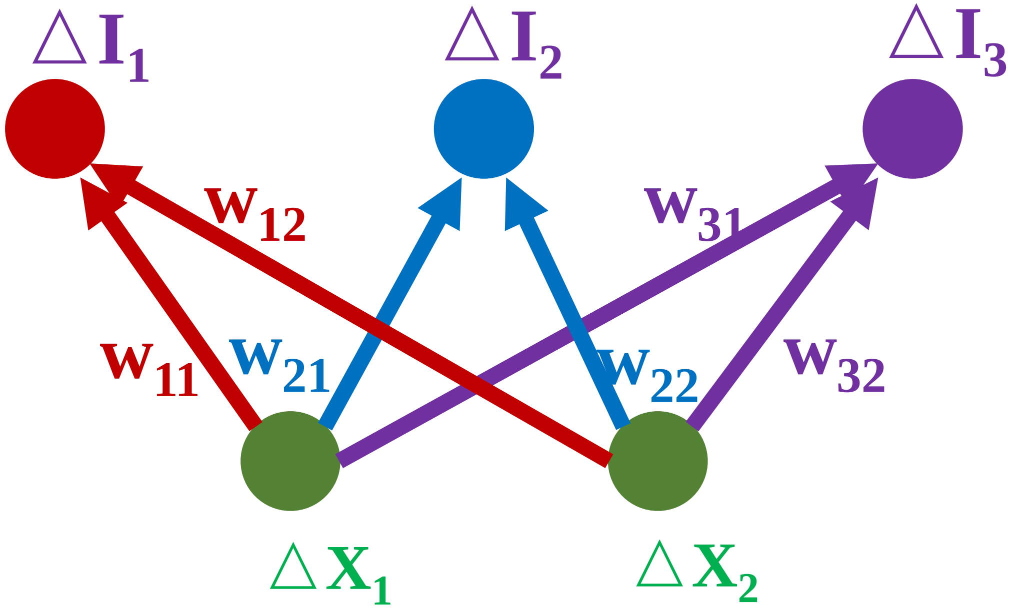
# Code - Response matrix from NN's perspective

- Input layer: ΔX
- Output Layer: ΔI=w· ΔX
- Label: ΔI_
- Loss=MeanSquaredError (ΔI, ΔI_)

- No hidden layer
- No bias
- No activation function

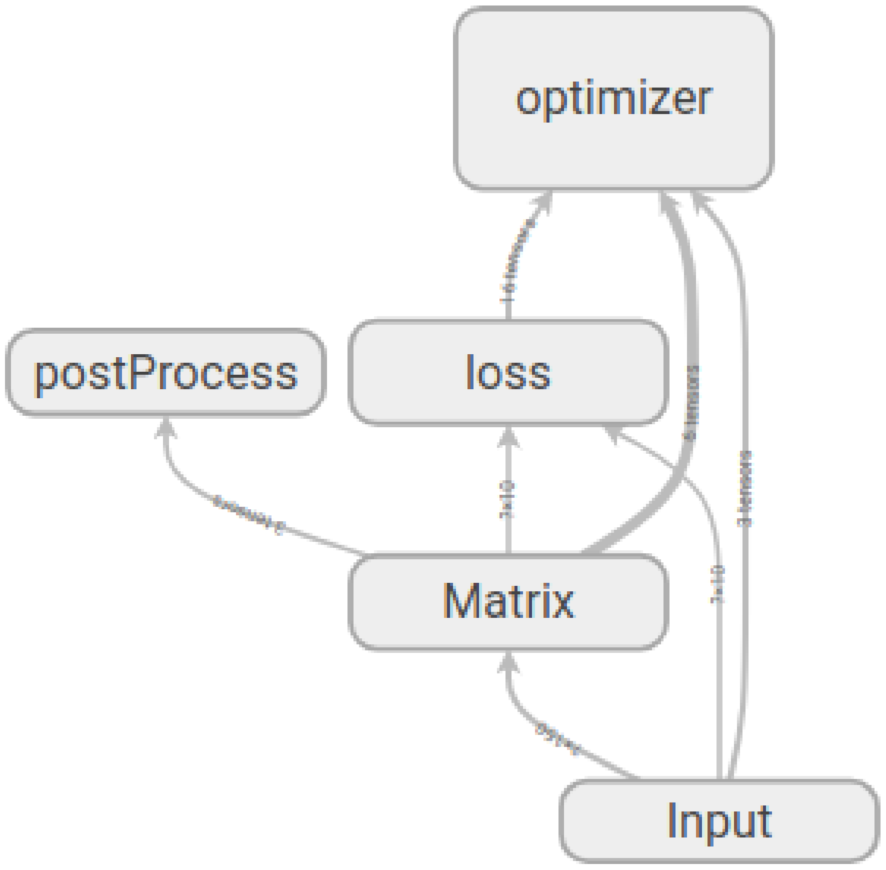
- Data collection:
  - Simulation
  - Experiment

$\Delta I = w \cdot \Delta X$

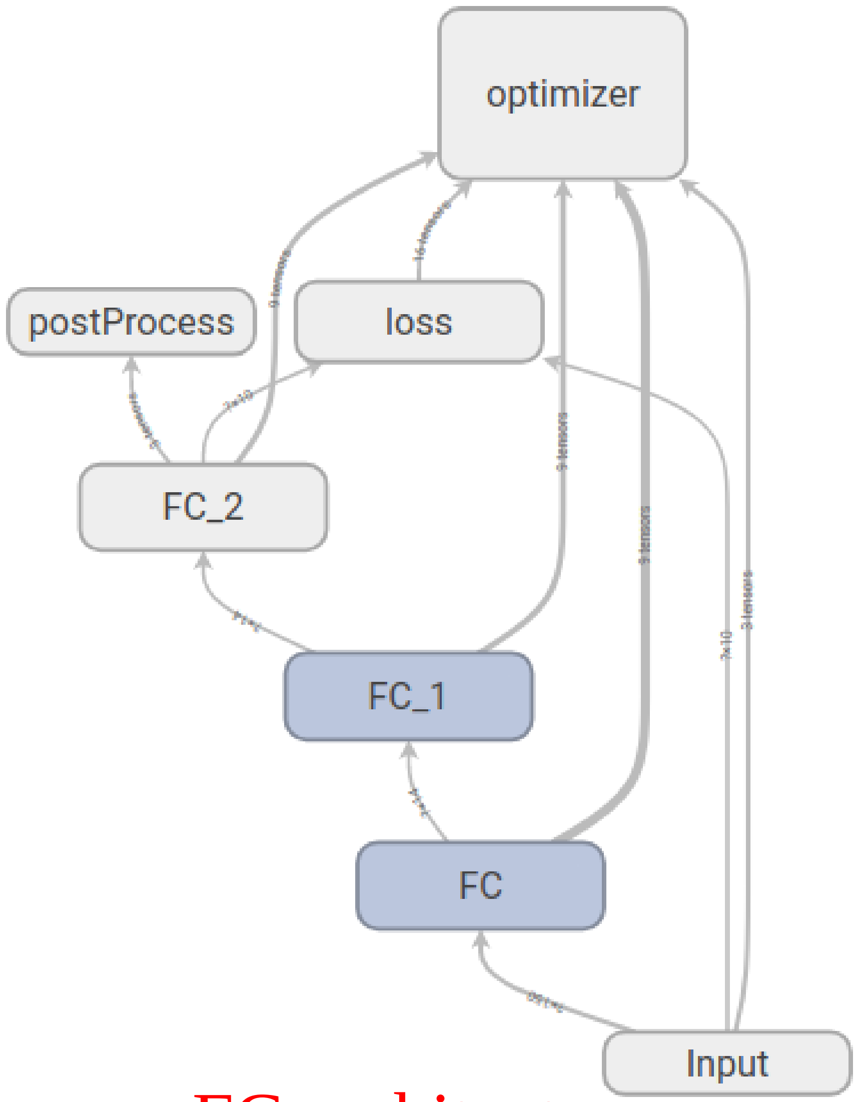



**Conclusion: Response matrix is the simplest NN**

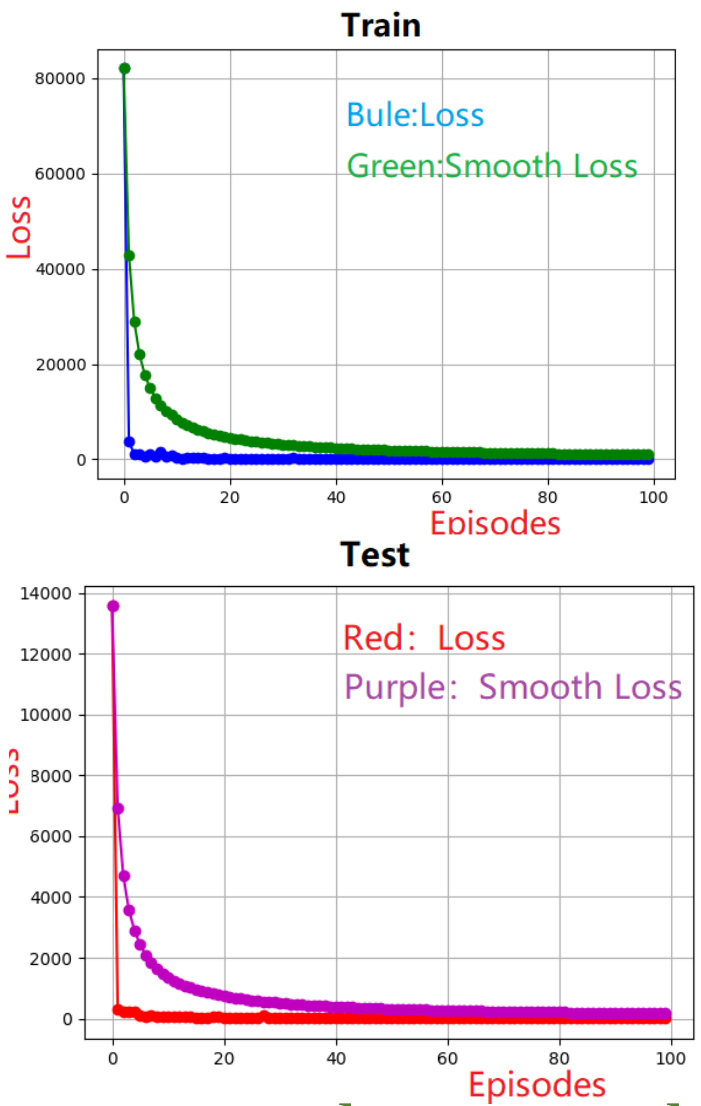
# Code– Response Matrix & FC architectures




Response Matrix architecture

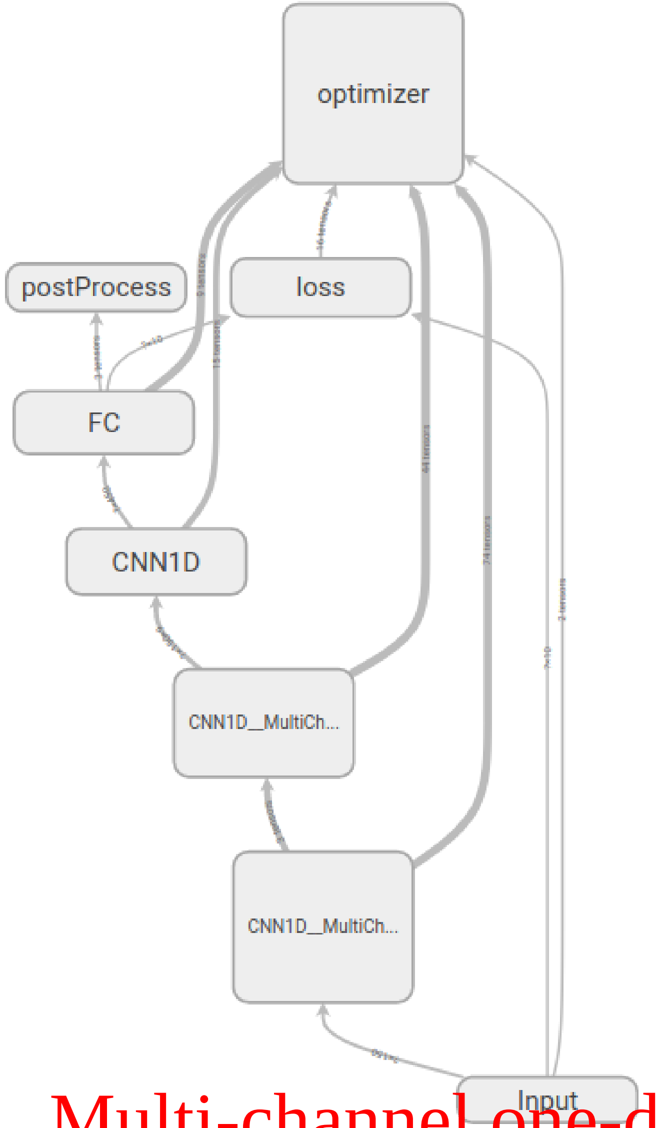


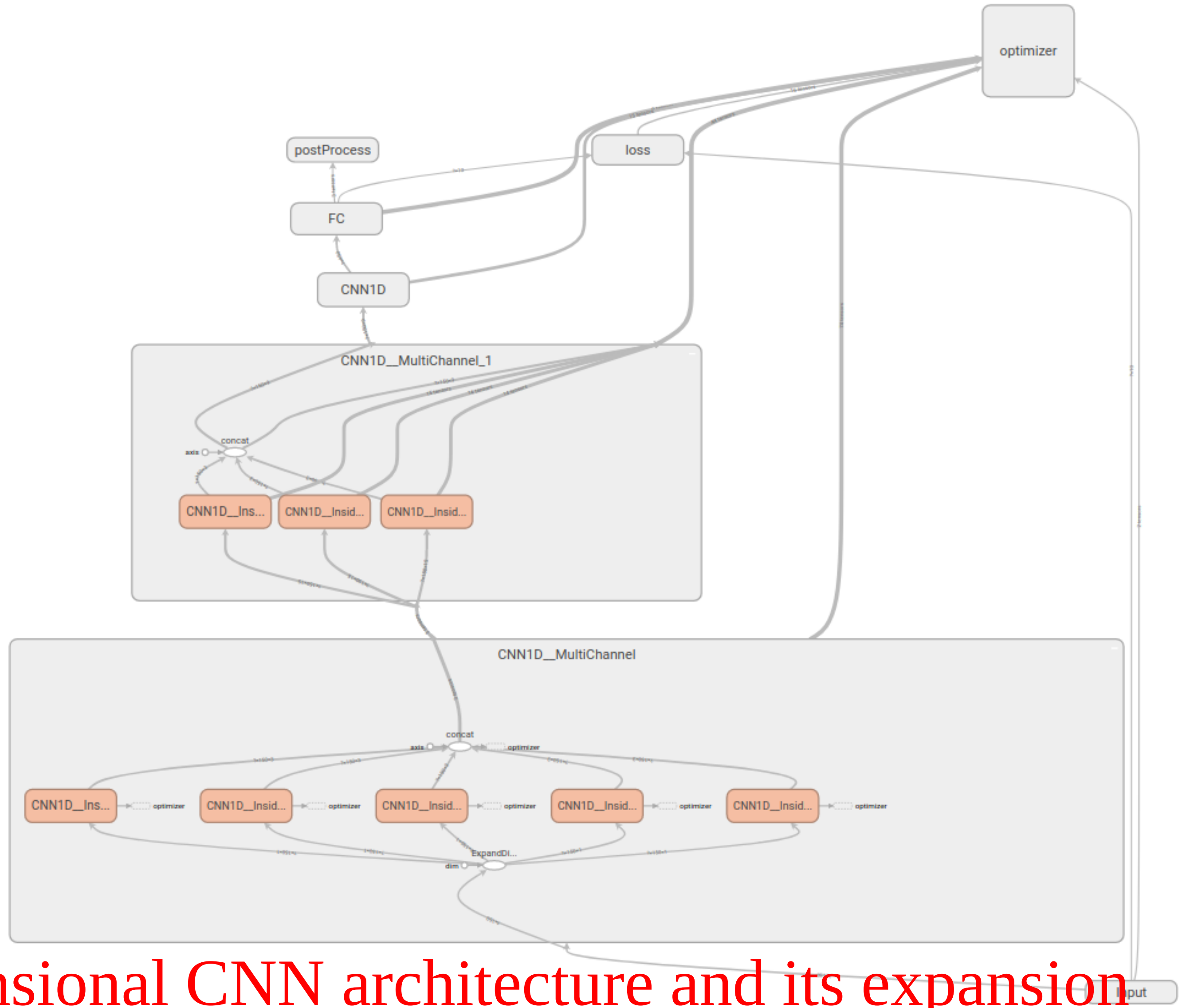

FC architecture




Losses along episodes

# Code – CNN architecture




Multi-channel one-dimensional CNN architecture and its expansion

# Code – Tricks & Experience

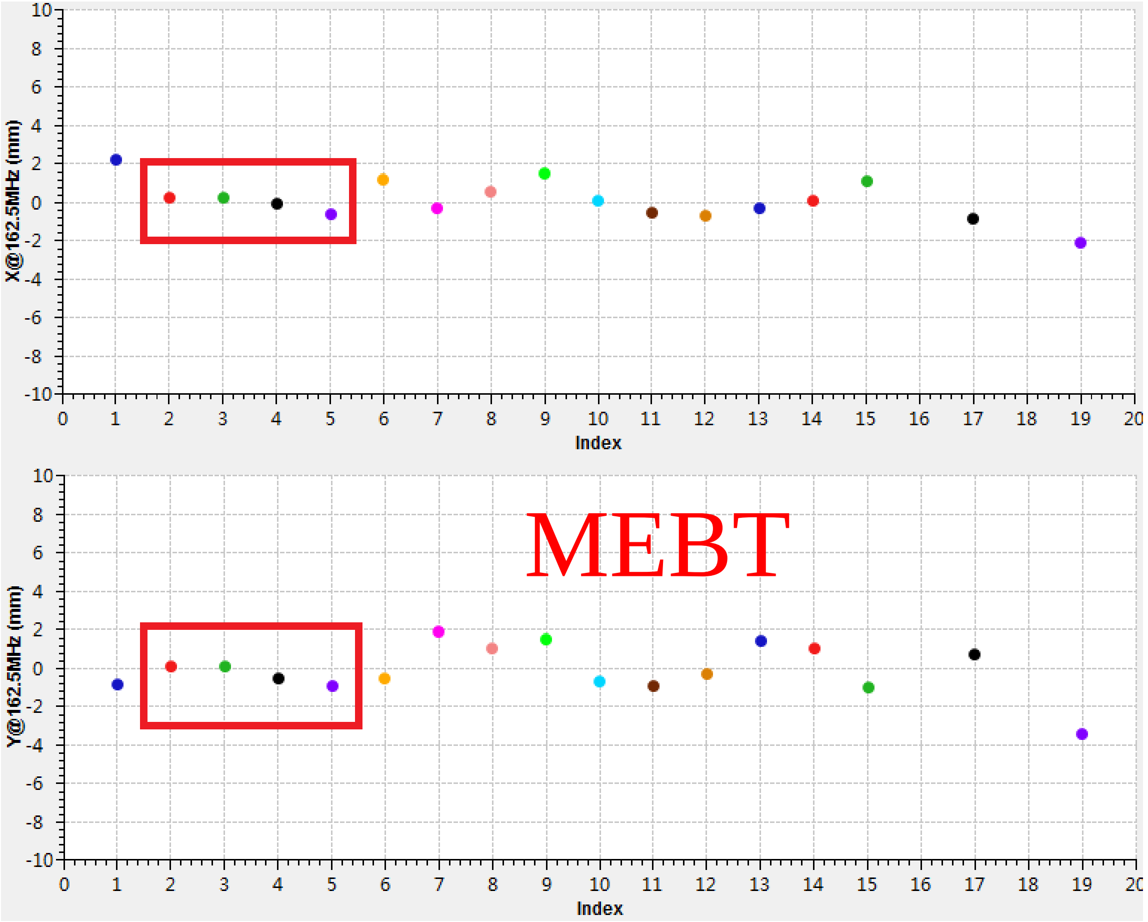
- Input ： Output
  - ΔX : ΔI
  - [ΔX, I] : ΔI
  - [X, I] : I
- Loss:
  - MSE
  - Cross entropy
- Experience:
  - Loss would be pretty larger than 0
  - Overfitting VS Nonlinear performance
  - Response matrix can basically meet the needs

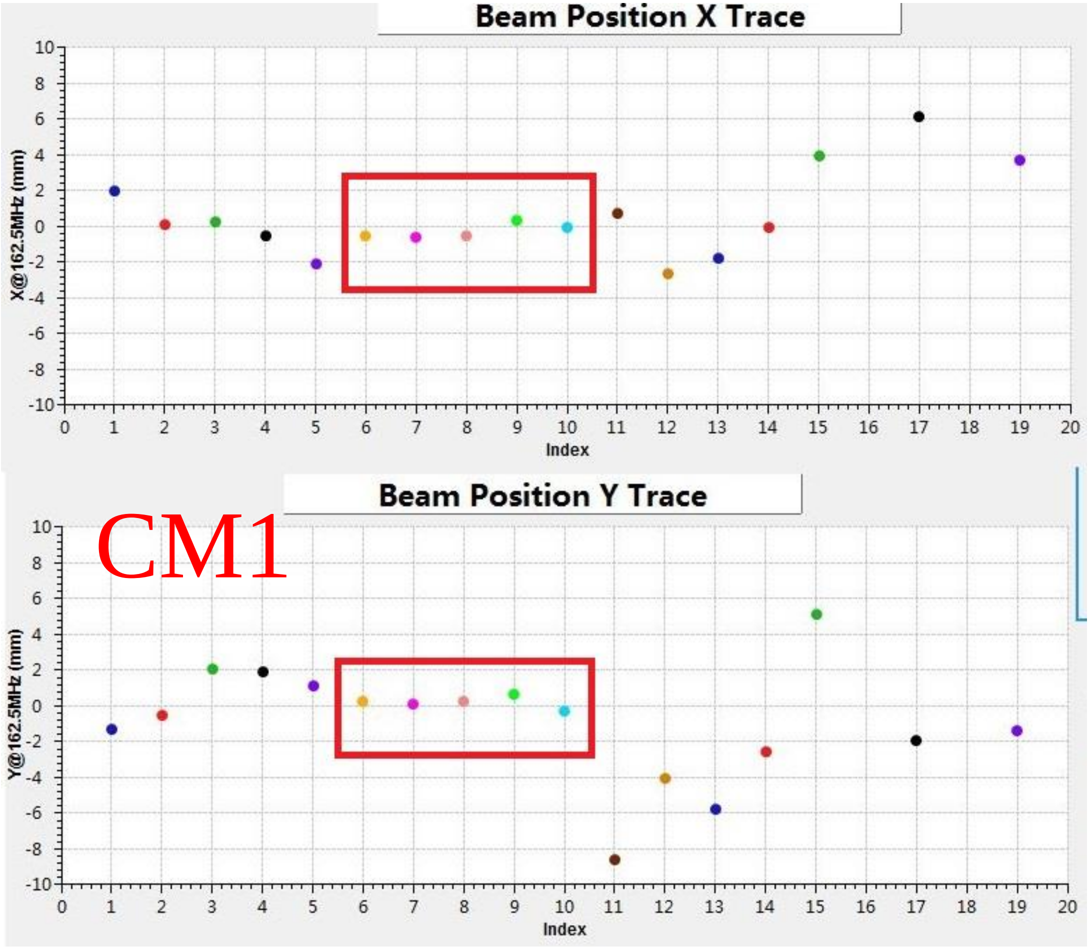
OS system: CentOS7
Python: 3.5
Toolkit: TF1.5

Train Set:
150k events (Simulation)
1.6k events (Experiment)

Test Set:
1.5k events(Simulation)
0.1k events(Experiment)

# Code – Proof from CIADS




Experimental results based on MEBT and CM1 of CIADS with deep-learning

# Thank you very much for your attention!

jiangpeiyong@impcas.ac.cn