Extraction of fissile isotope antineutrino spectra using deep learning

Jun Wang

Institut Franco-Chinois de!’ Energie Nucléaire (IFCEN), Sun Yat-sen University

Qingdao, August 16, 2024

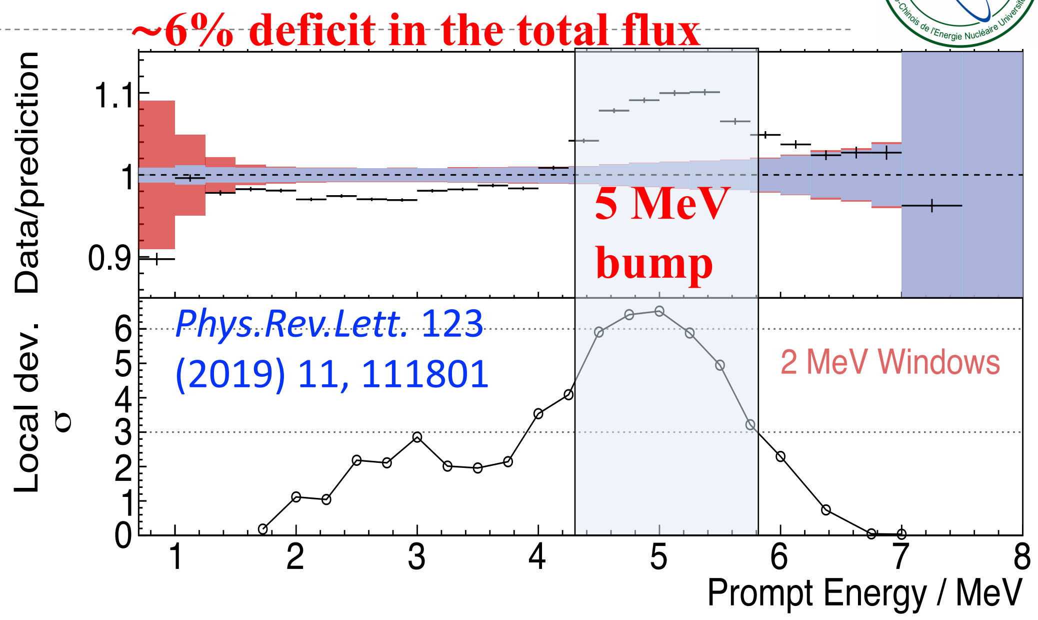
Introduction

- Background:

Double Chooz, RENO, Daya Bay, and NEOS confirmed reactor antineutrino anomaly (RAA) and the 5 MeV bump.

◎ New physics such as sterile neutrinos (rejected by STEREO at 95% CL, Nature 613 (2023) 7943, 257-261).

◉ Model predictions are inaccurate.

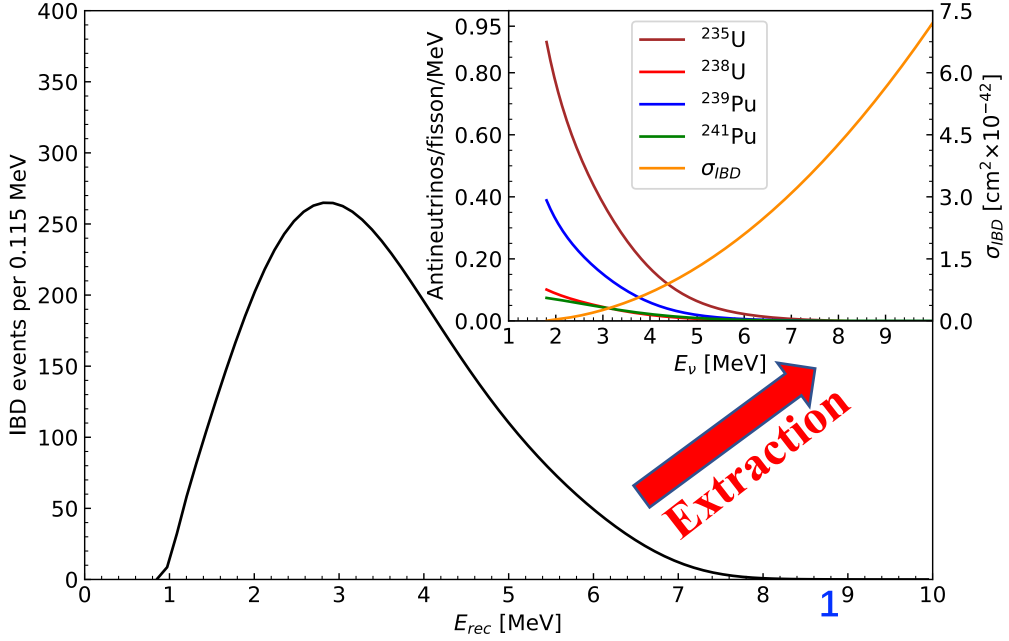


- Motivation:

The precise fissile isotope antineutrino spectra are crucial for **addressing the RAA problem, studying neutrino physics, and refining test nuclear databases**.

- Solution:

Decomposing the neutrino spectrum to extract the fissile isotope antineutrino spectra.



Our works:

the CNN method (2023), Nucl.Sci.Tech. 34 (2023) 5, 79

the FNN method (2024), arxiv: 2407.05834

Setup:

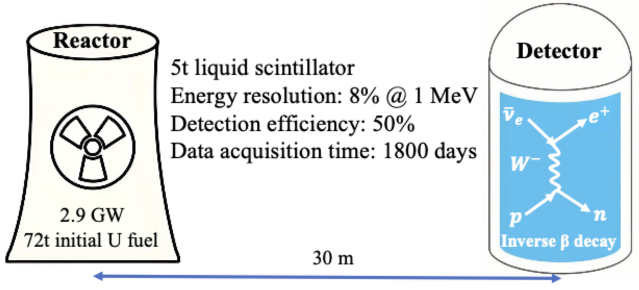


Figure 1. Configuration of a virtual experiment.

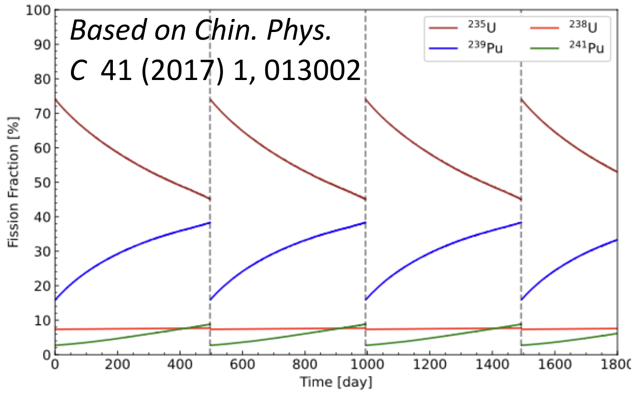


Figure 2. The evolution of fissile fractions for the four main isotopes in a reactor core as a function of operation day.

Inverse β decay (IBD) yield:

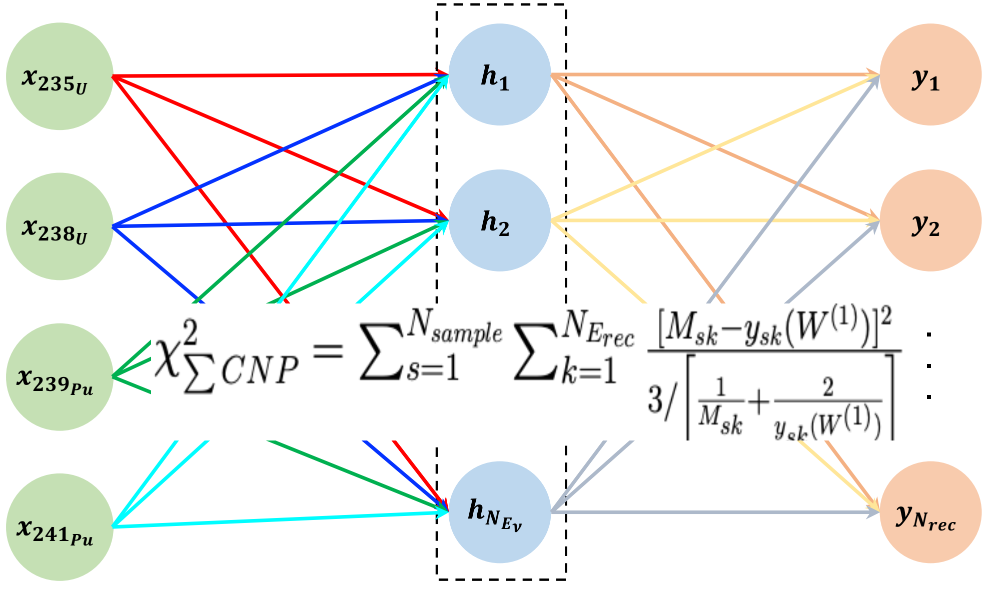
The expected reconstructed prompt energy spectrum $[M_k]$ can be expressed as

$$\mathrm{M}_k = \frac{N_p \varepsilon}{4\pi L^2} \int_{E_{rec}^{k}}^{E_{rec}^{k+1}} dE_{rec} \int_{T_{DAQ}} dt \int_{E_{thr}} dE_\nu \frac{W(t)}{\sum_l f_l(t)\epsilon_l} \phi(E_\nu, t) P_{ee}(L, E_\nu) \sigma_{IBD}(E_\nu) R(E_\nu, E_{rec}).$$

Convert it into the following matrix language form

$$\begin{aligned} M_{1\times N_{E_{rec}}} &= X_{1\times 4} \cdot S_{4\times N_{E_\nu}} \cdot P_{N_{E_\nu}\times N_{E_\nu}} \cdot \sigma_{N_{E_\nu}\times N_{E_\nu}} \cdot R_{N_{E_\nu}\times N_{E_{rec}}} \\ &= X_{1\times 4} \cdot S_{4\times N_{E_\nu}} \cdot P\sigma R_{N_{E_\nu}\times N_{E_{rec}}}, \end{aligned}$$

FNN architecture: a white-box model.



$W^{(1)} = S^T_{4\times N_{E\nu}}$ $W^{(2)} = P\sigma R^T_{N_{E_\nu}\times N_{E_{rec}}}$

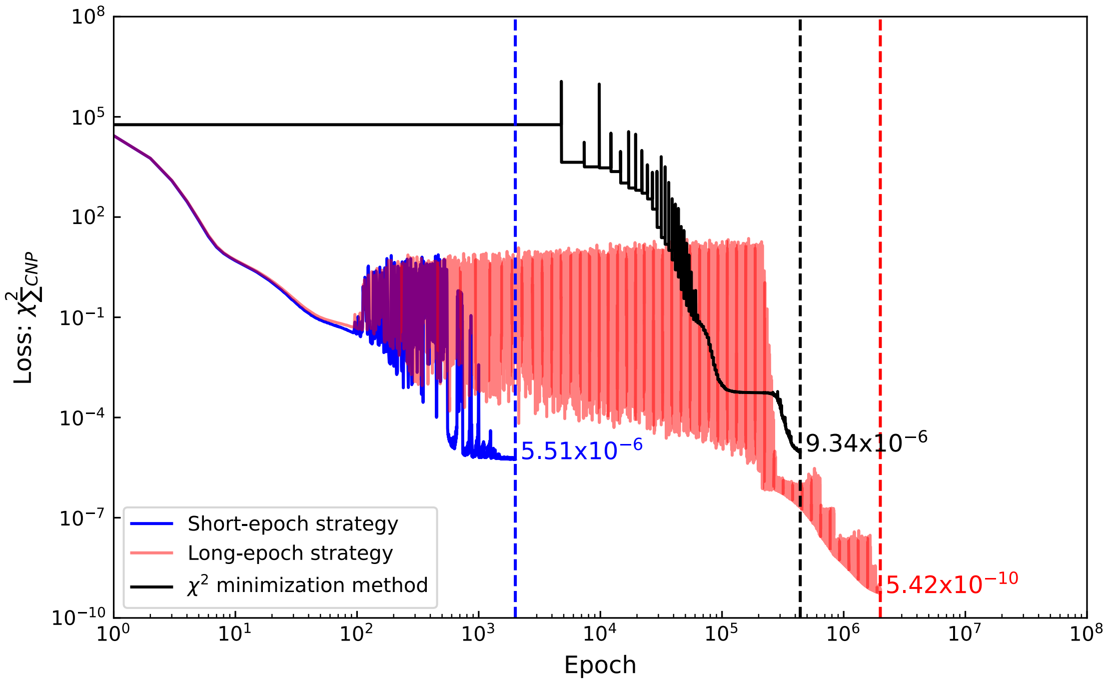
Input layer Hidden layer Output layer

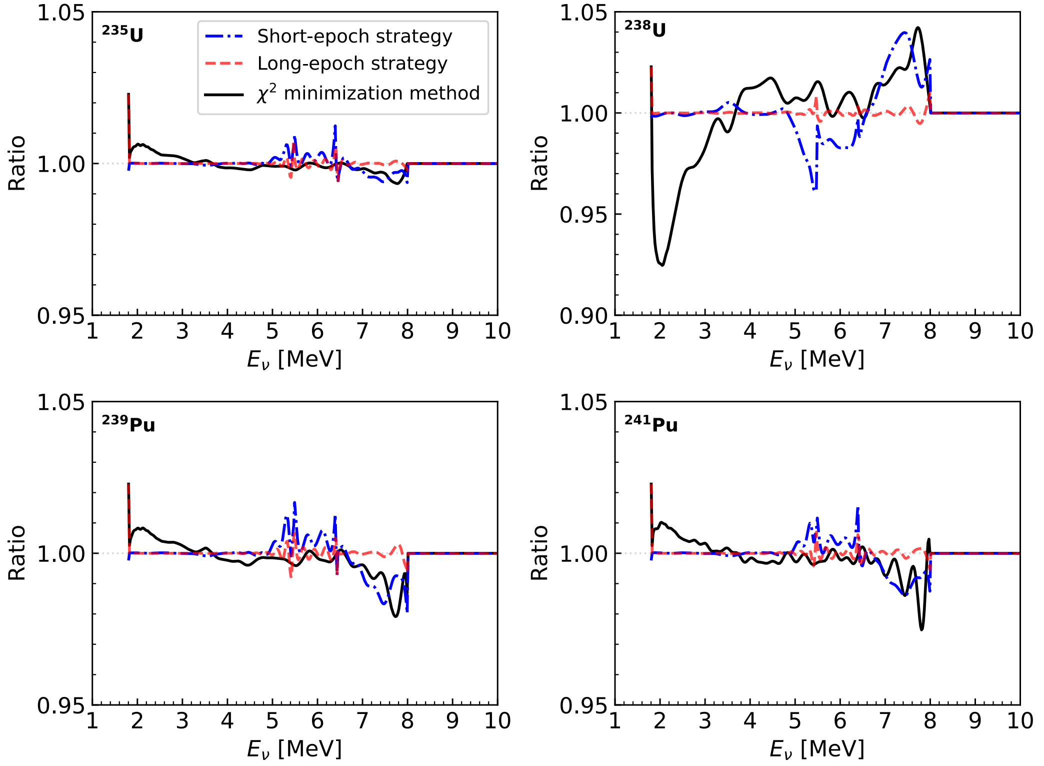
where

$$X_l = \sum_{u}^{u+N_t} \frac{N_p \varepsilon W(t^u) f_l(t^u)}{4\pi L^2 \sum_l f_l(t^u)\epsilon_l} \Delta t \Delta E_\nu \Delta E_{rec}$$

- $M_{1\times N_{E_{rec}}}$: the binned reconstructed prompt energy spectrum;
- $X_{1\times 4}$: the scaled reactor evolution information;
- $S_{4\times N_{E_\nu}}$: the binned antineutrino neutrino spectra for the four isotopes;
- $P\sigma R_{N_{E_\nu}\times N_{E_{rec}}}$: the composed of neutrino oscillation probability, IBD reaction cross-section, and detector effects.

Results, conclusions, and outlooks.





Primary Conclusions:

- The FNN model not only converges **faster** but also **better** than the χ^2 minimization method.
- The FNN model constrained the relative errors of **all** isotopic antineutrino spectra in the $2-8$ MeV range to **within 1%**.

Outlooks:

- Implementing this model in high-precision experiments like **TAO** would be an excellent match.
- We can **further investigate a broader range of physics topics** using this model and its variants, including unfolding, neutrino oscillation parameter measurements, sterile neutrino searches, etc.

THANK YOU FOR YOUR LISTEN.

谢谢！

Table 1. Key fitting settings of the FNN model and the traditional method.

Fitting approach	FNN model		χ^2 minimization method
Software platform	PyTorch		ROOT
Hareware platform	NVIDIA GeForce RTX 3060 Ti		A server with two 28-core Intel(R) Xeon(R) Gold 6330 CPUs @ 2.00 GHz
Training strategy	Short-epoch	Long-epoch	
Epoch	2e3	2e6	Auto
Optimizer	AdamW	Adam	TMinuit2 library
Loss function	$\chi^2_{\sum CNP} = \sum_{s=1}^{N_{sample}} \sum_{k=1}^{N_{E_{rec}}} \frac{[M_{sk}-y_{sk}(W^{(1)})]^2}{3/\left[\frac{1}{M_{sk}}+\frac{2}{y_{sk}(W^{(1)})}\right]}$		
Initial fitting values	Huber-Mueller model		

The partitions of the hidden layer	[1], (1, 180], (180, 225], (225, 303], (303, 401]	
The learning rates for the hidden layer	[3.4892e-4, 9.9485e-4, 2.754e-4, 1.8272e-4, 0]	
The weight decay rates for the hidden layer	[7.418e-3, 7.748e-3, 4.155e-3, 9.999e-3, 0]	[0, 0, 0, 0, 0]
The learning rate for the output layer	0	
The weight decay for the output layer	0	
Learning rate scheduler	ReduceLROnPlateau (factor = 0.32, patience = 1e2)	ReduceLROnPlateau (factor = 0.32, patience = 1e4) & epoch $\geq$ 2e5
Batch size	30	