Systematic mapping of $U(1)_{L_e-L_\mu-L_\tau}$ flavor model via reinforcement learning

Satsuki Nishimura (Kyushu University)

Davide Meloni (Roma Tre University)

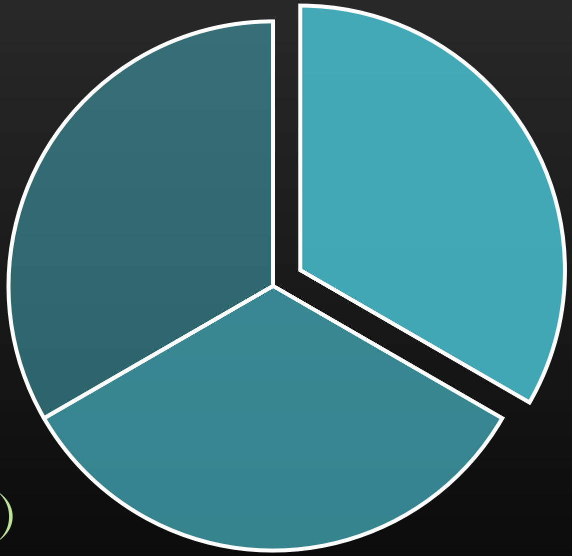
Hajime Otsuka (Kyushu University)

(Now in progress)

Machine Learning

- A technique in which a computer extracts hidden rules or patterns as it iteratively learns data.

Supervised Learning



Reinforcement Learning

S. Nishimura, C. Miyao, H. Otsuka,
- JHEP12(2023)021 (2304.14176 [hep-ph])
- JHEP10(2025)043 (2409.10023 [hep-ph])

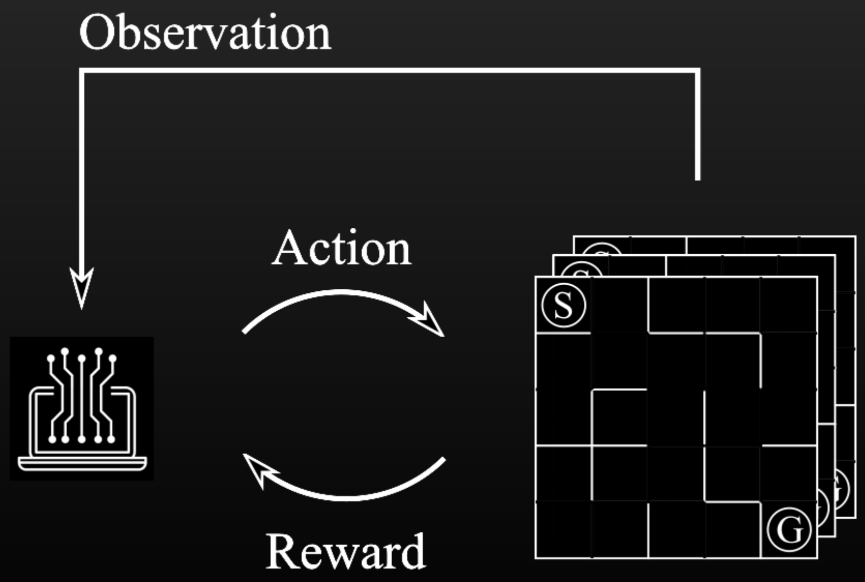
Unsupervised Learning

K. Ishiguro, S. Nishimura, H. Otsuka,
- JHEP08(2024)133 (2312.07181 [hep-th])

S. Nishimura, H. Otsuka, H. Uchiyama,
- arXiv:2503.21432 [hep-ph]
- arXiv:2504.00944 [hep-ph]

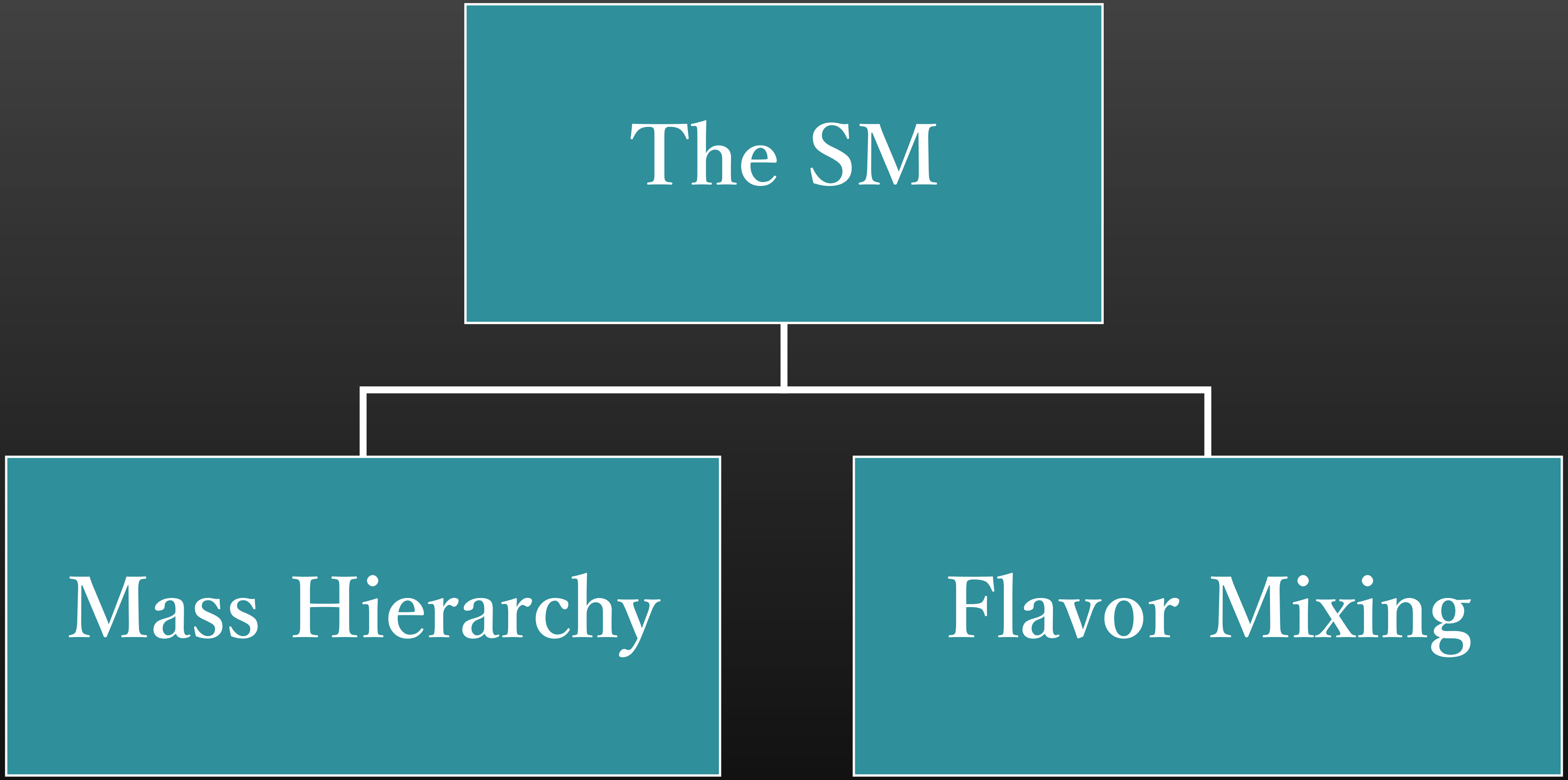
Reinforcement Learning (RL)

- Reinforcement learning can find optimal solutions even from a small amount of reference data by repeatedly trying to solve problems to be solved.



Can we utilize and apply the unique feature of RL to searching for flavor models?

Contents


The SM
Mass Hierarchy
Flavor Mixing

$L_e - L_\mu - L_\tau$ sym. and neutrino (1)

- $U(1)_{L_e-L_\mu-L_\tau}$ flavor model assigns charges as follows:

$$\left(Q_e, Q_\mu, Q_\tau\right) = (+1, -1, -1), \qquad \left(Q_q, Q_H\right) = (0,0)$$

- Under this symmetry, the neutrino mass matrix naturally derives an inverted hierarchy, explains large atmospheric mixing angles.

- S.T. Petcov, W. Rodejohann,
  PRD 71, 073002 (2005), arXiv:hep-ph/0409135
- M. Lindner, A. Merle, V. Niro,
  JCAP 01 (2011) 034, JCAP 07 (2014) E01, arXiv:1011.4950
- F. Feruglio, A. Romanino,
  Rev. Mod. Phys. 93, 015007 (2021), arXiv:1912.06028

$L_e - L_\mu - L_\tau$ sym. and neutrino（2）

- The following Weinberg operator generates neutrino mass.

$$L_{\mathrm{LO}} = \frac{1}{M}\left(x_{12} l_e l_\mu + x_{13} l_e l_\tau\right) H_u H_u$$

$$m_\nu = \frac{v_u^2}{M}\begin{pmatrix} 0 & x_{12} & x_{13} \\ x_{12} & 0 & 0 \\ x_{13} & 0 & 0 \end{pmatrix}$$

$L_e - L_\mu - L_\tau$ sym. and neutrino（3）

- This predicts $(\theta_{12}, \theta_{13}) = (\pi/4, 0)$ with degenerated eigenvalues, so corrections due to symmetry breaking are required.

$$r = \Delta m^2_{\rm sol} \,/\, \Delta m^2_{\rm atm} = 0$$

- Symmetry breaking is introduced by the VEV of two flavons. We assume $\langle \phi \rangle / M = \langle \theta \rangle / M = \lambda$.

NLO corrections to neutrino (1)

- Sym. breaking introduces next to leading (NLO) corrections.

$$(m_\nu)_{ij} = \frac{1}{M} x_{ij} v_u^2 l_i l_j \lambda^{\gamma_{ij}+\delta_{ij}}$$

- From the symmetry conditions, the charge structure of the correction term is characterized as follows.

$$Q_\nu = \begin{pmatrix} 2 + Q_\phi \gamma_{11} + Q_\theta \delta_{11} & Q_\phi \gamma_{12} + Q_\theta \delta_{12} & Q_\phi \gamma_{13} + Q_\theta \delta_{13} \\ & -2 + Q_\phi \gamma_{22} + Q_\theta \delta_{22} & -2 + Q_\phi \gamma_{23} + Q_\theta \delta_{23} \\ & & -2 + Q_\phi \gamma_{33} + Q_\theta \delta_{33} \end{pmatrix}$$

NLO corrections to neutrino（2）

- The final neutrino mass matrix is given below, enabling corrections for the solar mass difference and mixing angles.

$$m_\nu = \frac{v_u^2}{M} x_{12} \begin{pmatrix} {x'}_{11}\lambda^{\gamma_{11}+\delta_{11}} & 1 + {x'}_{12}\lambda^{\gamma_{12}+\delta_{12}} & x_{13} + {x'}_{13}\lambda^{\gamma_{13}+\delta_{13}} \\ & {x'}_{22}\lambda^{\gamma_{22}+\delta_{22}} & {x'}_{23}\lambda^{\gamma_{23}+\delta_{23}} \\ & & {x'}_{33}\lambda^{\gamma_{33}+\delta_{33}} \end{pmatrix}$$

NLO corrections to charged lepton

- The mass matrix of charged leptons is given as follows.

$$m_\ell = m_\tau \begin{pmatrix} a_{11}\lambda^{\alpha_1+\beta_1} & a_{12}\lambda^{\alpha_2+\beta_2} & a_{13}\lambda^{\alpha_3+\beta_3} \\ a_{21}\lambda^{\alpha_4+\beta_4} & a_{22}\lambda^{\alpha_5+\beta_5} & 1 \\ a_{31}\lambda^{\alpha_6+\beta_6} & a_{32}\lambda^{\alpha_7+\beta_7} & 1 \end{pmatrix}$$

- The entire structure of charges is specified by 26-dimensional vector $s = \{Q_e, Q_\mu, Q_\phi, Q_\theta, \alpha_1, \ldots, \delta_4\}$

Motivation of machine learning

- The parameter space is significantly large:

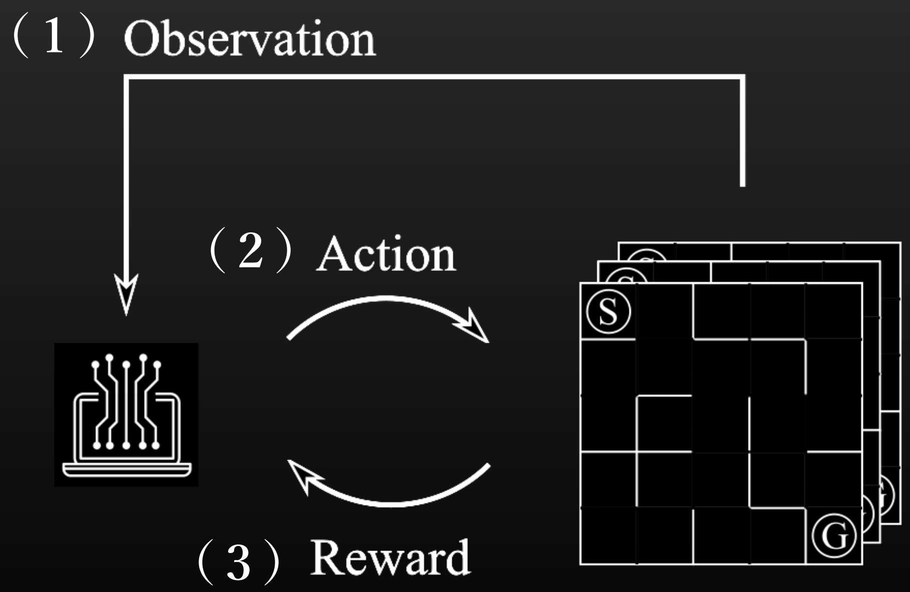
$$\{Q_e, Q_\mu, Q_\phi, Q_\theta\} \in [-10,10], \qquad \{\alpha_1, \ldots, \delta_4\} \in [0,10]$$

$$\rightarrow 21^4 \times 11^{24} \sim 2 \times 10^{30} \text{ patterns}$$

- We apply reinforcement learning to identify the charge structure which reproduce experimental observables.

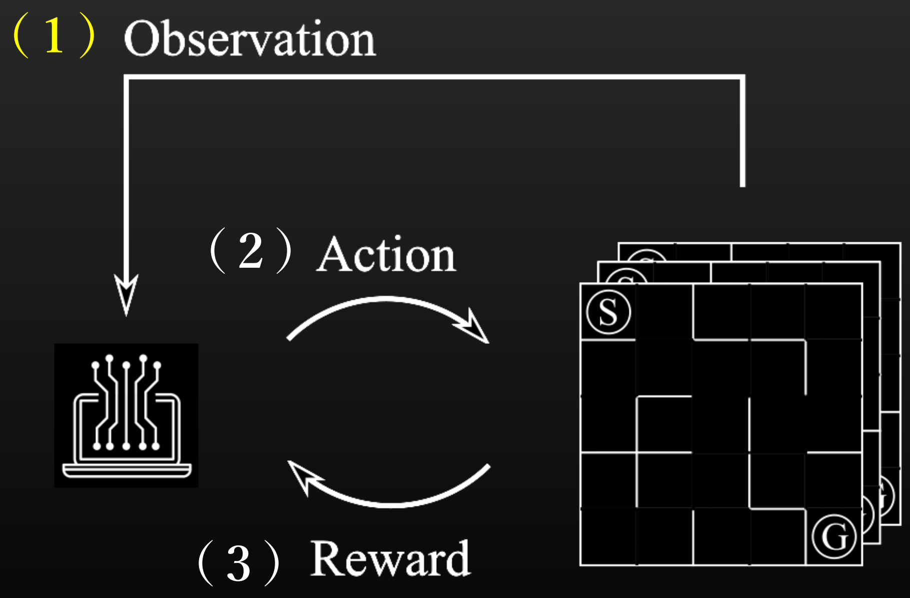
Reinforcement Learning (RL)

- Subject of learning : Agent
- Problem to be solved : Environment



Reinforcement Learning (RL)

- Procedure : The agent observe the environment,

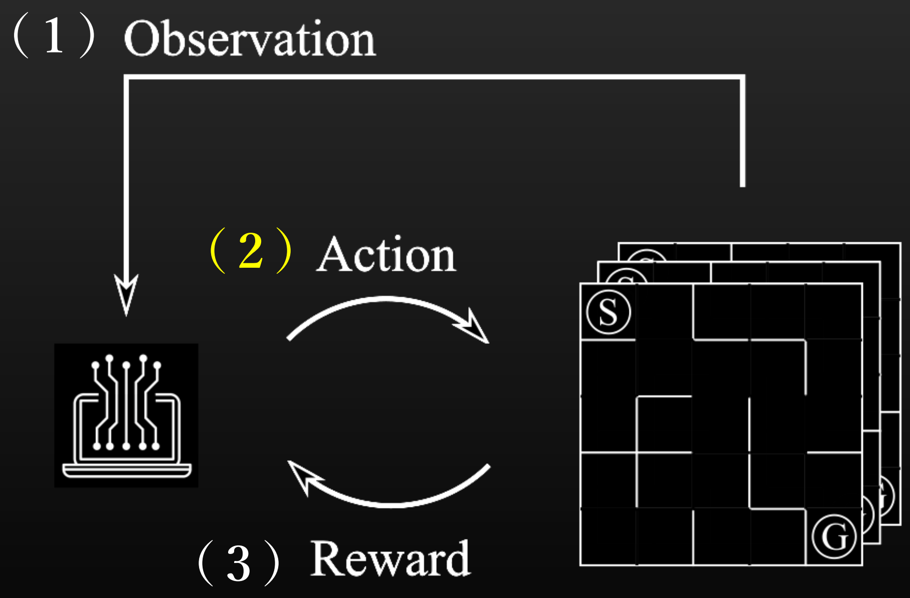


(Example of mazes)

（1） observe walls

Reinforcement Learning (RL)

- Procedure：The agent observe the environment, choose an action,



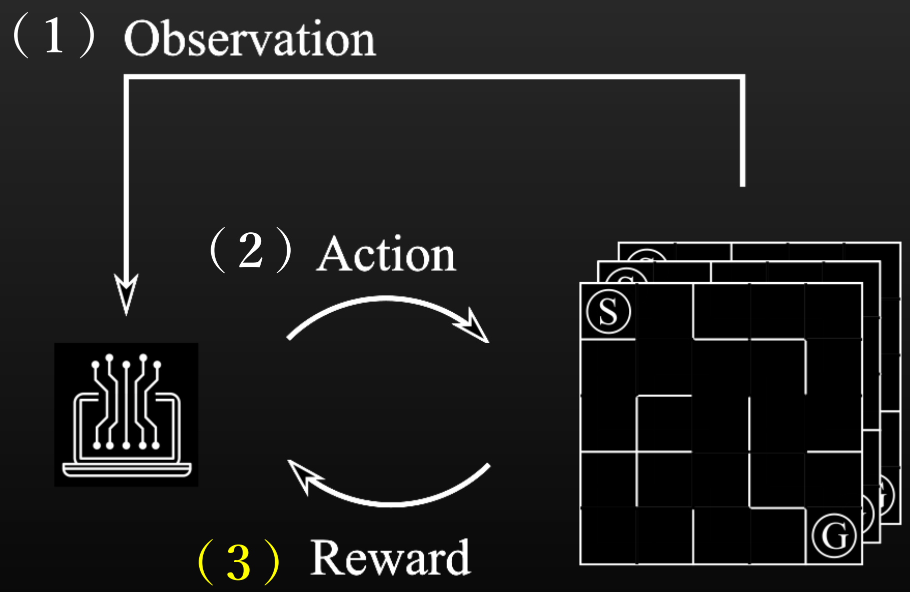
(Example of mazes)

（1） observe walls

（2） take one step

Reinforcement Learning (RL)

- Procedure：The agent observe the environment, choose an action, and get rewards depending on the action.



(Example of mazes)

（1） observe walls

（2） take one step

（3） get points as closing the goal

Reinforcement Learning (RL)

- Procedure：The agent observe the environment, choose an action,
  and get rewards depending on the action.

- The agent autonomously acquires a principle of action
  that maximizes the sum of rewards.

(Examples of mazes) By turning back upon reaching a dead-end,
the agent can solve mazes correctly.

Relating works

- T. R. Harvey, A. Lukas,
  JHEP, 2021, 161 (2021), arXiv:2103.04759.
- S. Nishimura, C. Miyao, H. Otsuka,
  JHEP, 2023, 21 (2023), arXiv:2304.14176.
  JHEP, 2025, 43 (2025), arXiv:2409.10023.

- RL was applied to the FN model, and it was indicated that RL can reproduce the experimental flavor structures efficiently and give discussion regarding neutrino sector or axion physics.

- In contrast, this work aims to
  - demonstrate the capacity of RL for other flavor symmetry
  - construct systematical strategy to obtain natural solutions

RL configuration (env & action)

- The environment is a 26-dimensional integer variable space, which determines the flavor structure of leptons.

$$s = \{Q_e, Q_\mu, Q_\phi, Q_\theta, \alpha_1, \dots, \delta_4\}$$

- An action is defined as an operation that changes any one of these variables by ± 1.

RL configuration (reward: 1)

- The quality of state s is evaluated based on six observables.

$$O = \left(\frac{m_e}{m_\mu}, \frac{m_\mu}{m_\tau}, \sin^2\theta_{13}, \sin^2\theta_{23}, \sin^2\theta_{12}, r \right)$$

- The χ^2 value is defined as the evaluation function.

$$\chi^2 = \sum \frac{(E_i - O_i)^2}{\sigma_i^2}$$

RL configuration (reward: 2)

- The agent obtains positive reward R_{base} when χ^2 decrease.

$$R_{\text{base}} = -0.25 \times [\log_{10} \chi^2(s') - \log_{10} \chi^2(s)]$$

- If the agent realize $\chi^2 < 10^4$, we give the bonus reward R_{bonus}.

$$R_{\text{bonus}} = \frac{4 - \log_{10} \chi^2(s')}{8}$$

RL configuration (NN architecture)

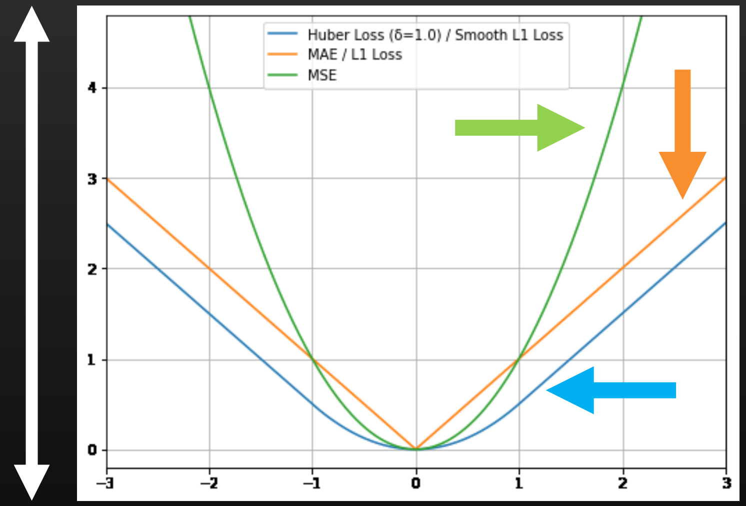
- A neural network is used to determine an action at each step, and there are several hyperparameters. (e.g., learning rate)

- We try 6 architectures with transformer structure in advance. Using the best architecture, we conduct grid search for 216 combinations of hyperparameter settings.

Loss Function

- An indicator to estimate the learning status of a neural network.

Bad Neural Network



Good Neural Network

Mean Squared Error (MSE)

$$L = (x - \bar{x})^2$$

Mean Absolute Error (MAE)

Huber function：MSE + MAE

$$L \sim \begin{cases} (x - \bar{x})^2 & \text{if } \; x - \bar{x} \leq \delta \\ |x - \bar{x}| & \text{if } \; x - \bar{x} > \delta \end{cases}$$

Result (hyperparameter search: 1)

- The table indicates the top 10 settings in terms of χ^2 value, and the best solution has $\chi^2 = 1.0 \times 10^3$. Each run needs 47 min. If we generate charges randomly, the median is $\chi^2 = 5.1 \times 10^5$.

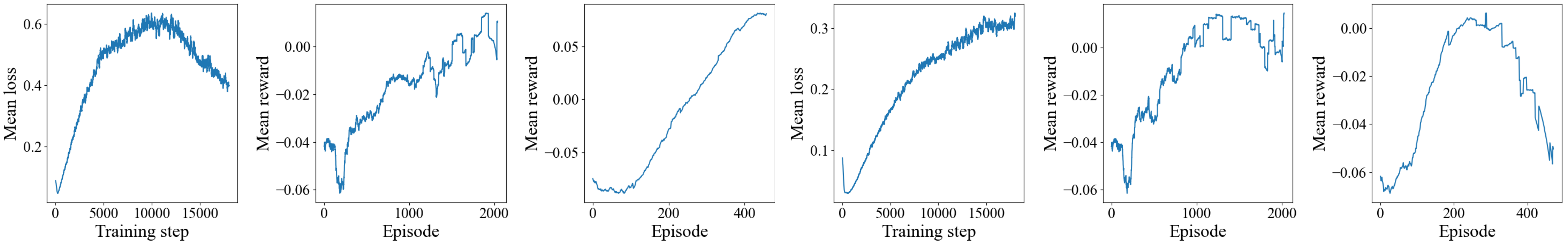
→ We obviously show that RL can decrease χ^2 effectively various flavor model.

ID	$\log_{10} \chi^2_{\rm RL}$	α	R	Batch size	γ	τ	update freq.
0	3.000	1×10^{-4}	0.9	512	0.99	0.01	1
1	3.122	5×10^{-5}	0.5	512	0.99	0.001	1
2	3.200	1×10^{-4}	0.9	512	0.99	0.001	10
3	3.224	1×10^{-4}	0.9	512	0.99	0.001	5
4	**3.224**	$\mathbf{1 \times 10^{-4}}$	**0.9**	**512**	**0.95**	**0.001**	**1**
5	3.283	5×10^{-5}	0.5	512	0.99	0.01	10
6	3.347	5×10^{-5}	0.5	1024	0.95	0.01	5
7	3.349	1×10^{-5}	0.5	1024	0.95	0.01	5
8	3.366	5×10^{-5}	0.5	1024	0.99	0.01	5
9	3.417	1×10^{-4}	0.9	512	0.95	0.001	10

Result (hyperparameter search: 2)

- The graphs show the transition of loss & reward for ID = 0, 216. They reveal that solution with low χ^2 can be realized when the loss does not diverge and the reward increases until the end.

ID = 0 ID = 216



Result (naturalness of solutions: 1)

- To evaluate naturalness of solutions found by RL, we define a threshold F_N for each model as the proportion of random coefficients for which the condition $\chi^2 \leq N$ is satisfied.

- C. Cornella, D. Curtin, E. T. Neil, J. O. Thompson, PRD 111, 015042 (2025), arXiv:2306.08026.

- 100,000 sets of $O(1)$ coefficients are generated for each model, and then, F_N is calculated. Each run needs 15 min.

Result (naturalness of solutions: 2)

- Although only 13 patterns realize nonzero F_{1000} among 216 settings, many solutions that are accurate before the MC calculation remain.

→ RL successfully find natural candidates even before MC.

ID	$\log_{10}\chi^2_{\text{RL}}$	$\log_{10}\chi^2_{\text{MC}}$	$\mathcal{F}_{100}$	$\mathcal{F}_{1000}$
144	3.635	1.755	5.000×10^{-5}	2.025×10^{-2}
2	3.200	1.945	3.000×10^{-5}	2.904×10^{-2}
104	3.582	1.979	1.000×10^{-5}	1.156×10^{-2}
6	3.347	1.919	1.000×10^{-5}	2.430×10^{-3}
7	3.349	1.958	1.000×10^{-5}	2.280×10^{-3}
0	3.000	2.151	0.000	1.710×10^{-2}
89	3.516	2.272	0.000	1.490×10^{-3}
5	3.283	2.477	0.000	1.450×10^{-3}
88	3.516	2.337	0.000	1.430×10^{-3}
3	3.224	2.929	0.000	5.005×10^{-4}
4	**3.224**	**2.900**	**0.000**	$\mathbf{3.257\times10^{-4}}$
1	3.122	2.933	0.000	1.627×10^{-4}
192	3.697	2.680	0.000	2.000×10^{-5}

Summary（1）

- We applied reinforcement learning (RL) to the search for charge assignment in the $U(1)_{L_e-L_\mu-L_\tau}$ flavor model.
  We tested various hyperparameters and Monte Carlo search to evaluate naturalness of solutions found by RL.

Summary (2)

- While RL with MC calculation reproduces experimental results, we discuss learning behavior in terms of loss & reward and show that RL can be used to derive "natural" solutions.

- This discussion provides the systematic strategy for RL-based parameter search in various models.