

UC Irvine

UC Irvine Previously Published Works

Title

Towards AI-assisted neutrino flavor theory design

Permalink

<https://escholarship.org/uc/item/7n99q295>

Journal

Communications Physics, 9(1)

ISSN

2399-3650

Authors

Baretz, Jason Benjamin

Fieg, Max

Ganesh, Vijay

et al.

Publication Date

2026-05-01

DOI

10.1038/s42005-026-02627-2

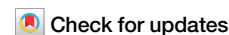
Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

<https://doi.org/10.1038/s42005-026-02627-2>

Towards AI-assisted neutrino flavor theory design



Jason Benjamin Baretz¹, Max Fieg^{1,2}, Vijay Ganesh³, Aishik Ghosh^{1,3,4}✉, V. Knapp-Pérez¹, Jake Rudolph¹ & Daniel Whiteson¹

Particle physics theories, such as those which explain neutrino flavor mixing, arise from a vast landscape of model-building possibilities. A model's construction typically relies on the intuition of theorists. It also requires considerable effort to identify appropriate symmetry groups, assign field representations, and extract predictions for comparison with experimental data. We develop Autonomous Model Builder (AMBer), a framework in which a reinforcement learning agent interacts with a streamlined physics software pipeline to search these spaces efficiently. AMBer selects symmetry groups, particle content, and group representation assignments to construct models while minimizing the number of free parameters introduced. We validate our approach in well-studied regions of theory space and extend the exploration to a previously unexamined symmetry group. While demonstrated in the context of neutrino flavor theories, this approach of reinforcement learning with physics software feedback may be extended to other theoretical model-building problems in the future.

Particle physics seeks answers to basic questions about the nature of matter and its interactions. Answers are not in the form of raw experimental results, but are expressed as a *model*¹ of particles and their interactions that match observations and make predictions for previously unexplained phenomena. In the case of neutrino oscillations, for example, a successful model should explain the observed neutrino masses and oscillations and minimize additional theoretical elements to maximize predictive power^{2–4}.

Building such a model is far from trivial, as the space of theoretical options is vast and high-dimensional, but constrained by experimental results. At the same time, the twin goals of achieving agreement with experiment and minimizing degrees of freedom are often in tension. Complicating matters further, theory spaces themselves are not conducive to systematic searches, as small changes in model content or structure can yield dramatic changes in quality. Finally, there is the computational challenge that an individual model takes significant work to evaluate: proposing a set of symmetries and fields, assigning representations of these symmetries to the fields, and performing intricate calculations to determine the predictions of a model.

Typical approaches rely on theorists to use their experience and inspiration to select and evaluate tailored theories. In the context of neutrino models, the result is a large space that is expensive to explore, containing vast regions which have not been thoroughly considered. Within these spaces may lie valuable, unexplored theories.

Neutrinos in the Standard Model (SM) are purely left-handed massless states. Thus, due to the constraints of renormalizability and gauge invariance, they cannot acquire mass through the Higgs mechanism within the SM. However, the discovery of neutrino oscillations^{5–8} demonstrates that neutrinos must have small, nonzero masses, in direct conflict with the SM. This discovery points to physics beyond the SM to explain neutrino masses, which may also explain the flavor structure observed in the SM. Neutrino flavor models seek to explain neutrino masses and the structure of lepton mixing through beyond the SM mechanisms, symmetries, and particles.

This paper introduces a general-purpose strategy to efficiently search the space of neutrino flavor theories. An optimized scientific software pipeline is developed and interfaced with machine learning to efficiently explore the space and evaluate individual models, reminiscent of automated theorem proving work in mathematics^{9–11}. For the context of neutrino flavor model-building, the scientific software constructs the Lagrangian, extracts the mass matrices and fits model parameters to data. We link this pipeline to a reinforcement learning (RL) strategy to intelligently search the space, hunting for models which satisfy the conditions of minimal additional parameters and accurate description of observations. There have been previous applications of RL in particle physics^{12–17}, which focus on minimizing the RL agent's interaction with scientific software. In contrast, we embrace the need to integrate scientific software into the learning loop, giving the agent the same tools as a theorist so it can focus on what it does

¹Department of Physics and Astronomy, University of California, Irvine, CA, USA. ²Particle Theory Department, Fermilab, Batavia, IL, USA. ³Georgia Institute of Technology, Atlanta, GA, USA. ⁴Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA. ✉e-mail: aishikghosh@physics.gatech.edu

best: finding complex relationships in high-dimensional spaces. Our approach is summarized in Fig. 1.

While our approach may scale effectively to many domains, we demonstrate it in the context of finding Minimal Supersymmetric Standard Model (MSSM)-like seesaw models to describe observed neutrino oscillations and masses using non-Abelian flavor symmetries broken via a scalar acquiring a vacuum expectation value (VEV)¹⁸. To verify our framework, it is first applied to the well-explored space of the non-Abelian discrete group A_4 ^{19,20}, where it rediscovers model patterns previously identified in the literature. We also unleash it on the larger, unexplored space of the group $T_{19} = \mathbb{Z}_{19}\mathbb{Z}_3$, identifying multiple promising models. Both searches illustrate AMBer's ability to find compelling models worthy of further study.

Within the models found by AMBer, there could be a dark matter candidate, a mechanism for leptogenesis, or a component of a UV complete theory. AMBer's value is its ability to filter through an unexplored space of flavor symmetries to provide a first pass of patterns and particle representations that reproduce observables. This eliminates the time consuming process of manually tuning these particle representations and fitting parameters to current measurements. We will therefore refer to the models found by AMBer that fit observations with sufficiently few free parameters as “filtered models”.

Methods

Neutrino flavor model-building

In this subsection, we outline a common method for neutrino flavor model-building, beginning with the choice of a mass mechanism. A flavor symmetry group is selected to increase predictivity. Next, the particle content is defined and the irreducible representation assignment is chosen. The Lagrangian is constructed following the constraints imposed by the symmetry group. Scalar fields, the so-called flavons, may be included to break the flavor symmetry through VEVs. It is assumed that the VEVs are acquired from an appropriate scalar potential, which we do not specify. Finally, the mass matrices are extracted and a fit of the free parameters to experimental data is performed.

A common mechanism to explain neutrino mass is the type-1 seesaw mechanism^{21–24}, which hypothesizes at least two right-handed neutrinos N_i which have no SM gauge interactions. The Lagrangian is given by

$$-\mathcal{L}_{\text{see-saw}} = g^{ij} \tilde{H} L^i N^j + M_M^{ij} \bar{N}^{i,c} N^j + h.c., \quad (1)$$

where g^{ij} is a 3×3 dimensionless coupling matrix, H is the SM Higgs, L^i are the $SU(2)_L$ SM lepton doublets and M_M^{ij} is the Majorana mass matrix for the heavy right-handed neutrinos N^i . Once the Higgs acquires a VEV $\langle h \rangle = v$,

the neutrinos acquire a Dirac mass $M_D^{ij} = v g^{ij}$. Integrating out the heavy degrees of freedom N^i gives the SM neutrinos an effective mass of

$$M_\nu = -M_D^T M_M^{-1} M_D. \quad (2)$$

The three mass matrices M_D , M_M and M_ν are 3×3 complex matrices resulting in a total of 54 real parameters. Their values are not predicted but must be determined by comparison to experimental observables. However, there are only 12 physical observables: three charged lepton masses, three neutrino masses, three neutrino mixing angles, one CP violating phase, and two Majorana phases, under the assumption that neutrinos are Majorana. Of these parameters, only nine have been experimentally measured and seven remain after factoring out an overall mass scale; see the “ A_4 Properties” subsection in the “Supplementary Methods”. Models with more independent parameters than observables are not sufficiently constrained and are thus more descriptive rather than predictive. The number of parameters can be reduced by applying a symmetry requirement, such as a discrete non-Abelian flavor symmetry^{25,26}, which imposes constraints to reduce the number of independent parameters. However, such a symmetry must be broken at some scale as there are 3 observed generations of leptons with distinct masses and mixings. This breaking is typically mediated by flavon fields, scalars that are charged under the flavor symmetry and acquire a non-zero VEV. In this work, it is assumed that the VEVs are acquired from an appropriate scalar potential which will remain unspecified.

One of the most well-studied flavor symmetry groups is the alternating group of order four, A_4 ^{19,20,27}; see “ A_4 Properties” subsection in the “Supplementary Methods”. It is the smallest finite group with a triplet irreducible representation. As a result, the three generations of leptons can be naturally accommodated within a single multiplet. Here, A_4 -type models are considered, together with models based on the group T_{19} ; see Supplementary Tab. 2. The latter is the smallest finite subgroup of $U(3)$ ²⁸ which has not yet been explored in the context of flavor model-building. The symmetry is usually cast within an MSSM-like model, which includes two Higgs $SU(2)_L$ doublets H_u and H_d . There are also three right-handed neutrinos, and up to six flavons; see Table 1a.

The next steps are the assignment of irreducible representations to each particle and the construction of the Lagrangian. For instance, in the A_4 group, a common choice is to assign the $SU(2)$ lepton doublets in a triplet irreducible representation²⁷. To construct the Lagrangian, it requires that the invariant contractions under the flavor symmetry be computed, for example, using *Discrete*²⁹ in Mathematica. The generalized superpotential

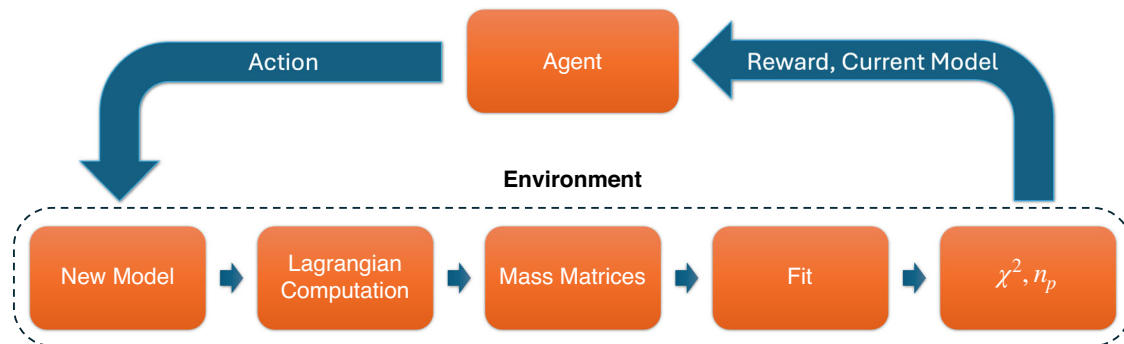


Fig. 1 | Diagram illustrating the reinforcement learning agent, AMBer. AMBer searches the space of models, taking actions to modify the model. Each model is then evaluated using a pipeline of physics software, which produces a reward depending on the χ^2 of the fit to data and the number of model parameters. The reward and model inform the agent's selection of the next action. This structure could be

generalized to any model-building task by defining the space of models, providing the agent with the physics software necessary to calculate predictions, and designing a reward parameterizing the user's preferences for output models. The result of this learning process is a set of models matching those preferences, which a physicist can use as a basis for further study.

Table 1 | Example model found by AMBer

(a)																
	L_i			E_i			N_i			H_u		H_d		ϕ_1		ϕ_2
A_4	3			3			3			1 [*]		1		3		3
$\mathbb{Z}_4$	2			3			3			3		3		0		3
(b)																
	L_1	L_2	L_3	E_1	E_2	E_3	N_1	N_2	N_3	H_u	H_d	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5
$A_4^{(1)}$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
$A_4^{(1')}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$A_4^{(1'')}$	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
$A_4^{(3)}$	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0
$\mathbb{Z}_4$	3	3	3	4	4	4	4	4	4	4	4	1	4	0	0	0

(a) Field assignments under A_4 and $\mathbb{Z}_4$ symmetries. The bold number in the A_4 column represents the respective irreducible representation for that particle, while the $\mathbb{Z}_4$ charges range from 0 to 3. The model is written using the notation commonly found in the literature. (b) Model representation written as presented to the agent. Non-Abelian symmetries are one-hot encoded while $\mathbb{Z}_4$ charges range from 1 to 4. The last two columns with zero entries indicate to the agent that those two fields are not present.

(from which the Lagrangian is extracted) up to dimension five, is then

$$\begin{aligned} \mathcal{W} = & \alpha_{(C)}^{ij} (L_i E_j H_d) + \frac{\alpha_{(C)}^{ijk}}{\Lambda} (L_i E_j \phi_k H_d) \\ & + \alpha_{(D)}^{ij} (L_i N_j H_u) + \frac{\alpha_{(D)}^{ijk}}{\Lambda} (L_i N_j \phi_k H_u) \\ & + \Lambda \alpha_{(M)}^{ij} (N_i N_j) + \alpha_{(M)}^{ijk} (N_i N_j \phi_k) + \frac{\alpha_{(M)}^{ijkl}}{\Lambda} (N_i N_j \phi_k \phi_l), \end{aligned} \quad (3)$$

where the α coefficients are dimensionless constants, ϕ_i are the flavons, and Λ is both the cut-off scale and the right-handed neutrino mass scale. The first line in Equation (3) is the contribution to the charged lepton mass matrix. The second line is the contribution to the neutrino Dirac mass matrix. Finally, the last line is the contribution to the Majorana mass matrix. The exact form of the superpotential for a given model immediately follows from the particle content and corresponding charge assignments, from which the α parameters can be fit to the experimental observables in Supplementary Table 1.

The flavons are assumed to acquire a VEV scale equal to 0.1Λ , and thus influence the overall scale of the Yukawa couplings. On the other hand, the different VEV alignment configurations for the flavon triplets determine the mass matrix structure, and hence the neutrino mixing observables. Here, the flavons are chosen to acquire one of the following typical VEV configurations^{20,27}:

$$\begin{aligned} \langle \phi \rangle = & 0.1\Lambda(1, 0, 0), \quad \langle \phi \rangle = 0.1\Lambda(1, 1, 1), \quad \langle \phi \rangle = 0.1\Lambda(0, 1, 0), \\ \langle \phi \rangle = & 0.1\Lambda(0, 0, 1), \quad \langle \phi \rangle = 0.1\Lambda(0, 1, -1), \quad \langle \phi \rangle = 0.1\Lambda(1, \omega^2, \omega), \\ \langle \phi \rangle = & 0.1\Lambda(1, \omega, \omega^2), \end{aligned} \quad (4)$$

where $\omega = \exp(\frac{2\pi i}{3})$. The complex Clebsch-Gordan coefficients (see the supplementary subsections “ A_4 properties” and “ T_{19} properties” in the “Supplementary Methods”) along with the complex VEVs in equation (4) are the sources of CP violation, as the dimensionless coefficients α in equation (3) are assumed to be real. Once the VEVs are chosen, the mass matrices can be constructed, and the values of the effective parameters are obtained numerically through χ^2 minimization using the standard Pontecorvo-Maki-Nakagawa-Sakata (PMNS) parameterization³⁰.

The number of free parameters, n_p , of the model is a strong heuristic estimator of its capacity to generalize to future experiments rather than simply describe current results, and will be a vital metric in the RL approach. Dimensionful parameters are factored out by removing an overall scale and expressing related parameters as ratios where possible. For instance, one can

always factor out one parameter $\hat{\alpha}_{(C)}$ in the first line in Equation (3), $\hat{\alpha}_{(D)}$ in the second line, and $\hat{\alpha}_{(M)}$ in the third line. The mass scales are then given by $\hat{\alpha}_{(C)} v_d$ for the charged leptons and $\hat{\alpha}_{(D)} v_u^2 / \Lambda \hat{\alpha}_{(M)}$ for the active neutrinos. This reduces the number of observables by one for each of the neutrino sectors and the charged lepton sector, and allows the fit to focus on the reduced parameter space of the mass ratios. Thus, the number of effective parameters is given by the number of dimensionless couplings $\alpha^{(i)}$ minus three. These effective parameters are fit to the observables in Supplementary Tab. 1 with some minimization software. Here, the FlavorPy package^{31,32} is chosen to perform the fits.

A viable neutrino mass model must also satisfy constraints related to the absolute mass scale of neutrinos, namely the effective electron-neutrino mass, $m_{\nu_e}^{\text{eff}}$, the effective majorana mass, m_{ee} , and the sum of neutrino masses, Σm_ν . Only dimensionless quantities are used for the fit, so to compare the models found with these dimensionful quantities, an overall active neutrino mass scale M_ν must be chosen. This scale is chosen to be

$$M_\nu^2 = \frac{1}{2} \left(\frac{|\Delta m_{21}^2|_{\text{exp}}}{|\Delta \tilde{m}_{21}^2|_{\text{fit}}} + \frac{|\Delta m_{31}^2|_{\text{exp}}}{|\Delta \tilde{m}_{31}^2|_{\text{fit}}} \right), \quad (5)$$

where the numerator is the experimentally measured value, and the denominator $\Delta \tilde{m}_{ij}^2$ is the dimensionless mass-squared difference that is evaluated from the dimensionless neutrino mass matrix at the fitted parameters. Fits to the $m_{\nu_e}^{\text{eff}}$, m_{ee} , or Σm_ν are not included when searching for models. Instead, these quantities are calculated for filtered models after training. The number of filtered models satisfying bounds set by current experiments will be reported. Several important aspects of neutrino flavor model-building have not been discussed in full here. For instance, the construction of the flavon potential, lepton flavor violating processes^{33,34}, the Renormalization Group Equations (RGE) running of the model parameters³⁵, and the corrections to the Kähler potential and the superpotential, some of which can induce theoretical uncertainties^{3,36}. To start with these, the decay width of lepton flavor violating processes, such as $\mu \rightarrow e\gamma$, has been previously computed in A_4 models^{33,34}, where these decays are generated by higher dimensional operators suppressed by a mass scale M . The specification of such a mass scale M would render the results sensitive to experimental constraints. However, the mass scale M generally depends on the masses of the heavy particles, which are left unspecified in this work. Furthermore, the value of the charged-lepton mass ratios at the GUT scale obtained from RGE is included in Supplementary Tab. 1 assuming only MSSM particle content. While it is possible that additional contributions from flavons and right-handed neutrinos could be important, they are generally expected to be negligible if their masses are near the GUT scale. In any case, the neutrino flavor models found here will ultimately

require a detailed study of these effects to be considered viable. Such considerations are deferred to future work.

Model-building with RL

This subsection discusses how neutrino flavor model-building has been cast as an RL task, and describes the software optimization, the basics of RL, and the specifics of how the RL agent searches the space and constructs models. Additionally, an ML-based visualization and diagnostic tool is detailed.

Software optimization. In the model-building recipe outlined in subsection “Neutrino Flavor Model-Building” in the “Methods” section, several steps can be computationally intensive. Specifically, the calculation of the superpotential, the mass matrices and the parameter fit are bottlenecks that prevent rapid evaluation of a specific model. To enable an efficient search, an optimized software pipeline is developed to allow automated end-to-end evaluation of a model.

The computation of the superpotential from the particle assignments requires knowledge of the Clebsch-Gordan coefficients of the chosen flavor symmetry group. Existing scientific software such as *Discrete*²⁹ is often used for these calculations. This package extracts the matrix representations of the given group elements through the library *GAP*³⁷, and computes the Clebsch-Gordan coefficients for the product between irreducible representations³⁸. However, the *Discrete* package is written in Mathematica, which can be slow and cumbersome to integrate into a Python-based RL loop. To allow seamless compatibility and computational efficiency within the RL environment, *PyDiscrete*, a Python-based translation of some of the key features of *Discrete*, was developed. This implementation offers significant speed improvements depending on the task. For example, the Clebsch-Gordan coefficients were computed for all of the different products of three triplets in T_{19} (see Supplementary Tab. 2) and the average computation time was calculated to be 590 times faster than *Discrete*.

The symbolic calculation of the lepton mass matrices was automated with the Python subpackage, *Model2Mass*. This subpackage uses *PyDiscrete* and *SymPy* to construct the superpotential in equation (3) for any finite non-Abelian group, for a given particle content and its respective irreducible representation assignments; see Table 1a. It also extracts the charged-lepton mass matrix, the neutrino Dirac mass matrix and the Majorana mass matrix. The *PyDiscrete* and *Model2Mass* subpackages are publicly released together in the *FlavorBuilder* package, installable via the Python Package Index (PyPI), and are available at this [link](#).

Fitting the free parameters to observables requires minimizing the χ^2 over a large parameter space that features many local minima. Searches for a global minimum often employ heuristics which are computationally expensive and unpredictable. To balance the need to limit computational cost and avoid local minima, *FlavorPy* is set to perform 100 brief searches from unique initial conditions, with the best resulting χ^2 selected.

Reinforcement learning. In RL³⁹, an agent learns a policy to select from available actions to maximize value (roughly, the expected reward) within an environment. Traditional applications include learning policies to make decisions about gameplay⁴⁰ or navigation⁴¹. RL has also been applied to search theoretical physics spaces, including string theory landscapes¹², Froggatt-Nielsen models of quark and lepton masses^{13–15}, dark matter models with vector-like leptons¹⁶, and supersymmetric gauge theories¹⁷. In previous applications, an agent has been able to find regions of quality models within large theory spaces without requiring an exhaustive search. These approaches have sought to limit the need to interface the agent with complex scientific software, either by fixing the available symmetries and couplings^{13–15}, using embedding strategies¹⁶, or avoiding the statistical fits. However, it is essential to overcome these challenges and interface the agent with scientific software in order to extend this approach to larger theoretical spaces. The approach taken

here is to design a framework for integrating existing scientific software with RL, in a similar spirit to previous efforts in interfacing physics software with probabilistic programming tools⁴².

Here, the search for a neutrino flavor model which matches observations with few free parameters is cast as an RL task. The agent’s environment can be considered the space of possible models, with its state the current model being evaluated. The available actions include changes to the particle content, changes to the particle representations, or in some cases, changes to the symmetries imposed. In addition to fixed intervals for model evaluation, the agent can choose to evaluate the model at intermittent steps. The value function is the agent’s expected return, which is the discounted sum of future rewards, with more distant rewards weighted less than near-term rewards. The reward for a model is generally parameterized by its χ^2 and number of parameters. The agent’s goal is therefore to maximize its expected return, encouraging it to take actions that ultimately produce high-quality models even if the immediate reward is low. The training occurs over a series of episodes where each episode starts with an initial state and continues either until a maximum number of steps are taken or a terminal state is reached. A terminal state is one for which the reward exceeds a certain threshold. When a terminal state is achieved, the agent is given a large reward and is reset to a random point in the theory space, starting a new episode.

RL algorithms develop a policy to select actions, a task often enhanced by simultaneously learning to predict the expected return given the current state. Proximal Policy Optimization (PPO)⁴³ optimizes two neural networks, one parameterizing the policy and one for predicting the expected return. During training, the agent utilizes the policy network to select actions and receives rewards based on the new state produced by each action. Periodically, the value network and policy network are updated in order to maximize the accuracy of the value network’s predictions and the advantage of the actions chosen by the policy network. The networks utilize a loss function composed of three terms:

1. Value loss, the mean squared error between the predicted return and the actual return.
2. Policy loss, the advantage of action taken over average of all possible actions, weighted by ratio of the probability of given action from old and new policies. The policy loss is clipped to prevent excessively large updates.
3. Entropy loss, the Shannon entropy of the policy, encourages exploration of a broad distribution across the action space.

Although both networks share the loss function, the value loss is a function only of the value network parameters, while the policy and entropy losses are functions of only the policy network parameters. With this method, the value network learns to improve its predictions of the expected return, and the policy network learns to determine which actions will lead to the highest future return⁴³. While one could experiment with alternative RL algorithms, PPO implemented with *Stable-Baselines3*⁴⁴ worked sufficiently well for this task.

Autonomous model builder. The Autonomous Model Builder (AMBer) described here is an RL agent with the task to learn a policy that will change a neutrino flavor model to make it more accurate while minimizing the number of free parameters, beginning from a random initial model.

Environment. The model being considered is the input to the policy and value networks. Each particle in a model is represented by a one-hot encoded vector to indicate its representation under the non-Abelian symmetry, with an additional element indicating the particle’s charge under the Abelian symmetry. Several rules are incorporated into the environment to ensure that changes to the model are mathematically consistent. These include having a representation for all leptons and Higgses (not required for flavons), consistent representation of particles (for example, if one right-handed charged lepton is in a triplet, all three must be), and consistent Abelian charges (i.e., all members of a triplet must have the same charge).

The environment's current state is represented as a matrix; see Table 1b for an example.

Additional components of the model provide important context, including a vector representing which leptons are associated together in a triplet, a vector representing the VEV configuration for each flavon, and a vector indicating the order of the present Abelian symmetry (i.e. the number of elements in the group).

Finally, the agent only evaluates the reward every n th step, so the number of steps until the next reward evaluation is also included in the environment. At each step, AMBer observes the entire state of the environment before selecting an action.

Actions. The agent can change the representation of a particle (or multiplet), its charge under an Abelian symmetry, the VEV configuration of a flavon triplet, the number of flavons, and in some cases the order of the Abelian symmetry. Each representation of the non-Abelian group can be changed in one action from any other representation. For the $\mathbb{Z}_N$ charge, two types of actions are defined. The first shifts the $\mathbb{Z}_N$ charge of a single field by ± 1 . For the second action, all fields, excluding the flavon fields, with the same SM representation have their $\mathbb{Z}_N$ charge changed to ± 1 of the first particle (e.g., L^1 , E^1 , or N^1). Under this choice, constructing mass matrices of rank 3 becomes easier than achieving ranks of 2 or 1. Finally, the agent can choose to evaluate the model between its set evaluation steps in order to gain more information on recent actions. Dynamic decisions to run physics software are a challenge for traditional search algorithms, but can be naturally incorporated in RL. All possible actions for each sector are listed in Supplementary Tab. 3.

Reward. The reward function R for AMBer is defined as

$$R(\chi^2, n_p, N) := \begin{cases} -c_{\text{inv}}, & \text{if invalid action} \\ -c_{\text{rank}} - \frac{9 - \sum_i \text{rank}(M_i)}{9}, & \text{if invalid model} \\ c_1 R_\chi(\chi^2) + c_2 R_p(n_p) + c_3 R_Z(N), & \text{otherwise} \end{cases} \quad (6)$$

where $c_{\text{rank}}, c_{\text{inv}} > 0$ quantify the penalty (i.e. a negative reward) for an invalid action or model respectively, c_j for $j = 1, 2, 3$, are positive coefficients and N is the order of the Abelian group $\mathbb{Z}_N$.

Invalid actions include those that do not change the model, to avoid stagnation, or actions which change the Abelian charge of an absent flavon. These invalid actions must be learned in this implementation as the action space is fixed and cannot dynamically exclude invalid actions. Invalid models are those that are obviously in conflict with experiment, such as predicting massless leptons, as in the case where M_C has a rank less than 3. Another possibility of invalid model is the case with 2 or more massless neutrinos. Assuming three right-handed neutrinos, this requires a valid model to have a rank ≥ 2 for M_D and rank 3 for M_M . Since this is known prior to performing a fit, the invalid model penalty allows the agent to avoid unnecessary computations. The invalid model penalty includes a term associated with the sum of the rank of the three mass matrices, which encourages the agent to increase the rank in order to reduce the penalty, leading the agent to consider models with more mass terms.

In case of a valid action and model, the coefficients c_j scale the relative importance of the terms

$$R_\chi(\chi^2) := -c_{\text{scale}} \tanh\left[\frac{\log_{10}(\chi^2 - \chi_{\text{target}}^2)}{\log_{10}(\chi_{\text{target}}^2)} - 1\right] - c_{\text{shift}}, \quad (7)$$

$$R_p(n_p) := \log_{10} \frac{n_p^{\text{target}}}{n_p}, \quad (8)$$

$$R_Z(N) := \frac{(N - N_{\min} - N_{\text{target}})^3}{N_{\max} - N_{\min}}. \quad (9)$$

The R_χ term encourages the agent to minimize the χ^2 fit. The χ_{target}^2 is a parameter which varies over the run to modify the strictness of the R_χ term, while c_{scale} and c_{shift} are constants which set the range of the reward and are fixed over the run. The R_p term encourages the agent to minimize the number of parameters in the model, n_p . The parameter n_p^{target} shifts over the course of the run to control the strictness of this term. The R_Z term in equation (9) encourages exploration by including a reward for higher order models. This term is used only when the agent is allowed to alter the order of the Abelian symmetry. In these circumstances, the agent can choose $N = 2, \dots, 10$, so $N_{\min} = 2$ and $N_{\max} = 10$ are fixed. N_{target} determines the degree of preference towards higher order Abelian groups.

On the steps where the agent does not evaluate the model, it receives a reward of 0. If it selects the action to evaluate the model on an intermittent step, the agent receives a reward based on whether it has improved the model since its last evaluation. If the reward has improved, instead of receiving the actual reward for that model the agent is given a reward just below the threshold. If the reward does not improve from its last value, the agent receives a reward equal to the invalid model case, $-c_{\text{rank}}$. This encourages the agent to only use this intermittent evaluation action when it thinks the model has improved.

Finally, the reward is normalized to maintain values between -1 and 1. When the reward exceeds 0.1, it is boosted to 1 to emphasize that state. For more details on the behavior of the reward function, see the "Reward Function" subsection in the "Supplementary Methods".

Training. AMBer trains with 250 parallel environments to maximize CPU utilization. Each environment has a maximum episode length of 1000 steps a full run continues until 32000 steps are taken in each environment. The neural networks of the agent use the default Stable-Baselines3 PPO hyperparameters, with the exception of the entropy coefficient, which is set to 0.05. The networks have 2 fully connected hidden layers with 64 nodes each, and are randomly initialized, as are the initial state models. Training occurs every 32 steps through the environments. In addition to when the agent chooses to evaluate, models are evaluated every n steps, where n is a tunable parameter. This allows several actions to be taken without any reward or penalty, which permits the agent to pass through poor models without reducing its reward. The A_4 runs use $n = 10$, while the T_{19} runs use $n = 5$.

The performance goals, χ_{target}^2 , n_p^{target} and N_{target} shift over the training, encouraging the agent to build upon lessons it learns early in the run in order to find increasingly high quality models. The values of these targets are detailed in Supplementary Tab. 4. The reward function is plotted in Supplementary Fig. 2. Each training run consists of multiple independent agents with identical hyperparameters and environments in order to identify general trends. Training each agent takes approximately 10 hours running on a single dedicated CPU. Here, training runs are performed in three theory spaces:

1. $A_4 \times \mathbb{Z}_4$ with $n_\phi \leq 5$, where n_ϕ is the number of flavons.
2. $A_4 \times \mathbb{Z}_N$ where the agent can alter N in the range $[2, 10]$ and $n_\phi \leq 5$.
3. $T_{19} \times \mathbb{Z}_4$ with $n_\phi \leq 6$.

The upper bound on the number of flavons n_ϕ is motivated by models in the literature²⁷. For T_{19} , $n_\phi \leq 6$ is chosen such that AMBer has the ability to place one flavon in each triplet representation, if it so chooses. All hyperparameters are manually tuned.

Diagnostics for AMBer runs. Performance during training can be evaluated by tracking the χ^2 and n_p metrics, the number of valid actions and models, and network-level quantities.

The left column of Fig. 2 shows how the χ^2 and number of parameters n_p evolve over the course of a run. In panel (a), corresponding to an $A_4 \times \mathbb{Z}_4$ run, the average number of parameters per model decreases over time, while the median χ^2 initially decreases before returning to a value of ~ 50 . This demonstrates the tradeoff between the competing goals of simplicity and fitness. In panel (c), corresponding to an $A_4 \times \mathbb{Z}_N$ run, the average number of parameters changes little, while the χ^2 steadily decreases. In panel (e),

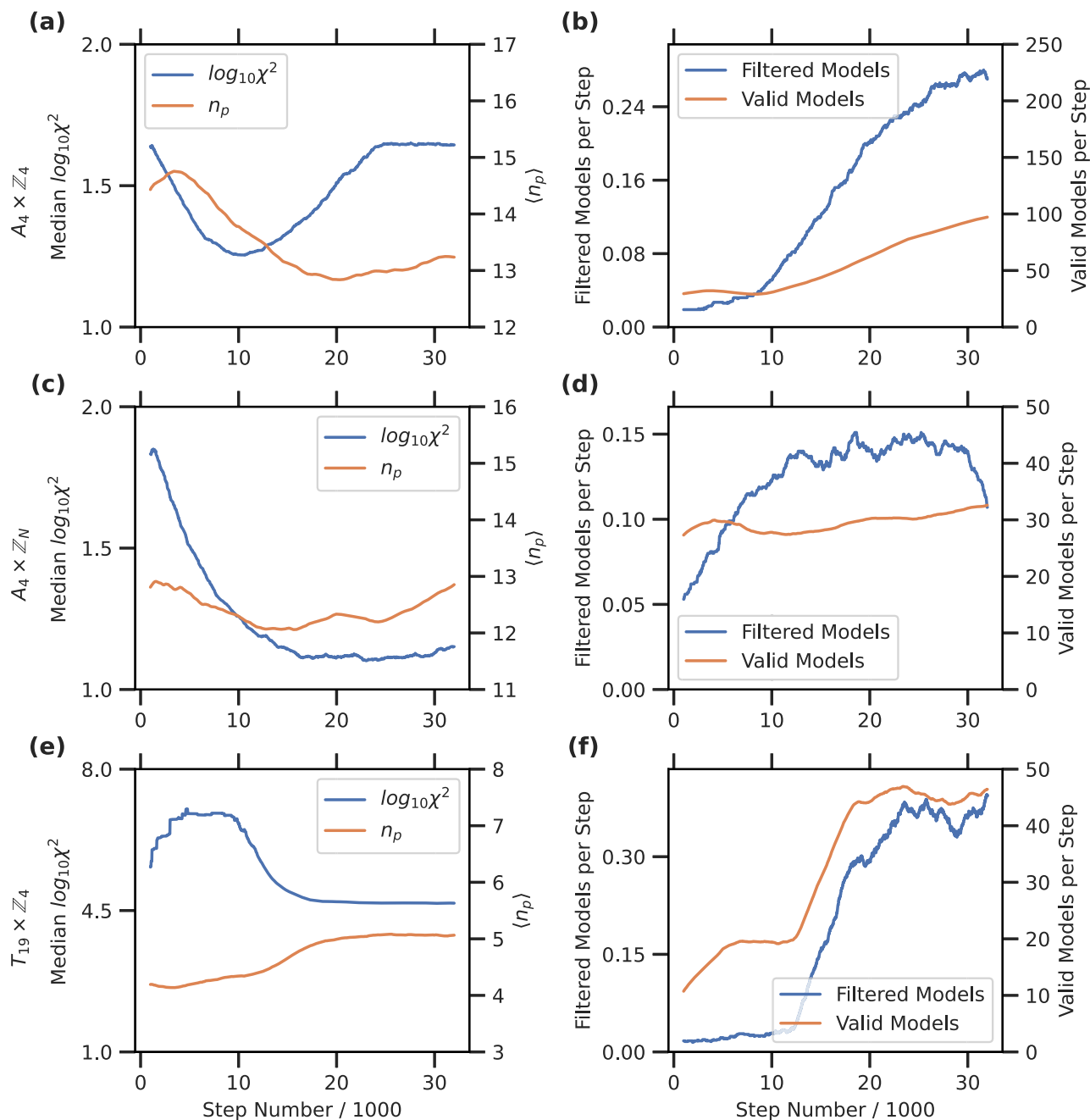


Fig. 2 | Evolution of variables of interest over training. Searches are performed in three spaces: $A_4 \times \mathbb{Z}_4$ (a and b), $A_4 \times \mathbb{Z}_N$ (c and d), and $T_{19} \times \mathbb{Z}_4$ (e and f). a, c and e show the evolution of χ^2 in blue (where the curve indicates the median $\log_{10}\chi^2$ over all environments) and the mean number of parameters ($\langle n_p \rangle$) as training progresses in orange for the different theory spaces. b, d and f show the number of valid models in

orange and filtered ($\chi^2 \leq 10$ and $n_p \leq 7$) models in blue for the three theory spaces. These are smoothed to more clearly see the relevant trends. There are two y-axes for each panel due to the differences in scale of each variable. Supplementary Tab. 4 describes how the reward targets change over time.

corresponding to a $T_{19} \times \mathbb{Z}_4$ run, there is a more complex trade-off between the two training goals. The average number of parameters starts much lower in this space, meaning the agent must learn to increase the number of parameters. As it does this, the median χ^2 decreases as it becomes easier to find a good fit with more parameters.

The right column demonstrates how the number of valid models and filtered models (those with $\chi^2 \leq 10$, $n_p \leq 7$) evolves over time. In all three theory spaces, the agent learns to build valid models over time due to the rank penalty, and increases its rate of filtered models due to the R_χ and R_p terms in the reward. In the $A_4 \times \mathbb{Z}_N$ run shown, the increase in valid models per step is more modest as the agent learns to build invariant terms with each different Abelian symmetry. Additionally, the number of filtered models per

step decreases at the end, likely indicating the agent has become influenced too heavily by one reward term over the others. This hypothesis is supported by the late increase in the number of parameters in panel (c).

Table 2 details the number of filtered models found in random scans versus AMBer training runs. In the $A_4 \times \mathbb{Z}_4$ and $A_4 \times \mathbb{Z}_N$ searches, 5 independent agents are trained, while in the $T_{19} \times \mathbb{Z}_4$ run 20 independent agents are trained in order to capture greater run-to-run variation. Since T_{19} has more representations, and therefore more plausible permutations of those representations, it is more susceptible to mode collapse, over-preferring a particular permutation. Additionally included are the number of inequivalent models and the number of models that satisfy dimensionful experimental constraints. A model is considered inequivalent if it does not

Table 2 | Performance comparison of AMBer and random scans

Space	Models Seen	Random scan NO	Random Scan IO	AMBer NO	AMBer IO
$A_4 \times \mathbb{Z}_4$	4×10^6	28/28/16	20/20/3	733/683/402	518/463/26
$A_4 \times \mathbb{Z}_N$	4×10^6	21/21/19	21/21/6	1404/1394/837	1179/1170/87
$T_{19} \times \mathbb{Z}_4$	24×10^6	409/409/309	244/244/44	6471/6439/4535	8678/8291/688

Each method evaluates the same number of models; results are shown separately for Normal Ordering (NO) and Inverted Ordering (IO). Reported as x/y/z: total number of filtered models, the number of inequivalent filtered models, and the number of inequivalent filtered models satisfying constraints from beta decay, neutrinoless double beta decay, and cosmological neutrino mass limits. The fewer inequivalent filtered models satisfying these constraints for inverted versus normal ordering reflect the generally heavier sum of neutrino masses in IO, which makes the bounds more restrictive.

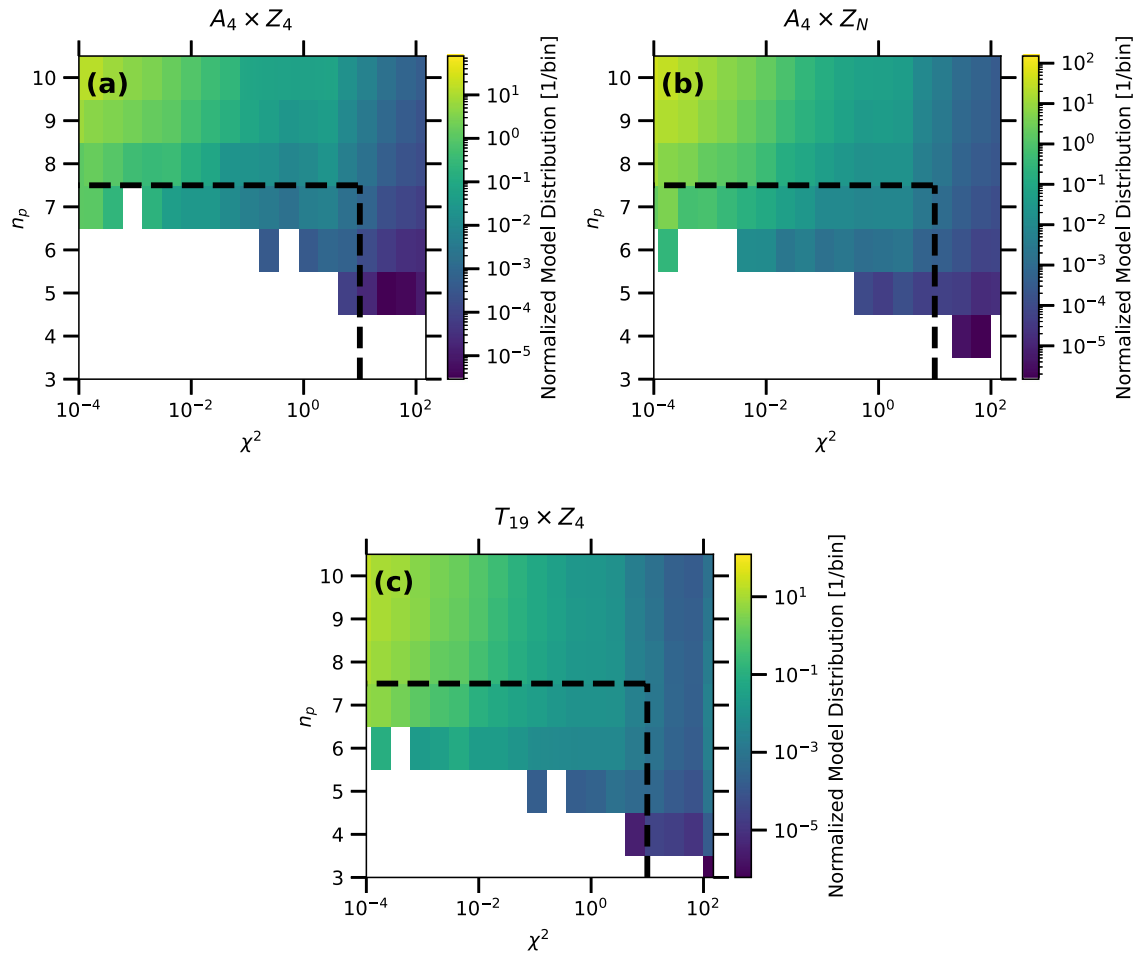


Fig. 3 | Number of parameters n_p and χ^2 for a representative distribution of found models. **a** shows the distribution for $A_4 \times \mathbb{Z}_4$, **b** for $A_4 \times \mathbb{Z}_N$, and **c** for $T_{19} \times \mathbb{Z}_4$. The region within the dashed black lines contains models with ≤ 7 parameters, and good fits, $\chi^2 \leq 10$.

share with any other model identical analytical expressions for their mass matrices, and if their irreducible representations under the flavor groups differ, even after accounting for possible field permutations. For instance, exchanging the irreducible representations of the E_1 and E_2 fields yields the same physical model, since this operation merely permutes the columns of the charged lepton mass matrix. The same reasoning applies to permutations among the L_i , N_i , and flavon fields ϕ_i (taking into account their respective VEVs). Hence, the numbers quoted in Table 2 indicate that AMBer is not settling into a specific set of neutrino flavor models given our definition of inequivalent models. A fully exhaustive analysis would require checking whether any two models are related by a unitary transformation.

For the filtered models, the observables related to the absolute mass scale of neutrinos, $m_{\nu_\nu}^{\text{eff}}$, m_{ee} and Σm_ν , are calculated. The experimental constraints on these are given by $m_{\nu_\nu}^{\text{eff}} < 0.45$ eV from KATRIN⁴⁵, $m_{ee} < 36$ meV from KamLAND-ZEN⁴⁶ and $\Sigma m_\nu < 0.12$ eV from Planck⁴⁷.

The number of inequivalent filtered models that satisfy these constraints is reported in Table 2.

AMBer outperforms a random scan in all spaces. To verify this efficiency gain, the number of CPU-hours for a random scan and a training run must be compared. On average, a training run takes approximately 70 percent longer to complete than a random scan. This can be attributed to the fact that AMBer identifies many more valid models, requiring more expensive computations than the random scan. Nonetheless, the number of filtered models per CPU-hour is still significantly higher for AMBer than for a random scan. Additional neural network level diagnostics are shown in Supplementary Fig. 3.

Results

This section describes the models found in each theory space. AMBer finds too many models to describe each one in detail, so broad features are

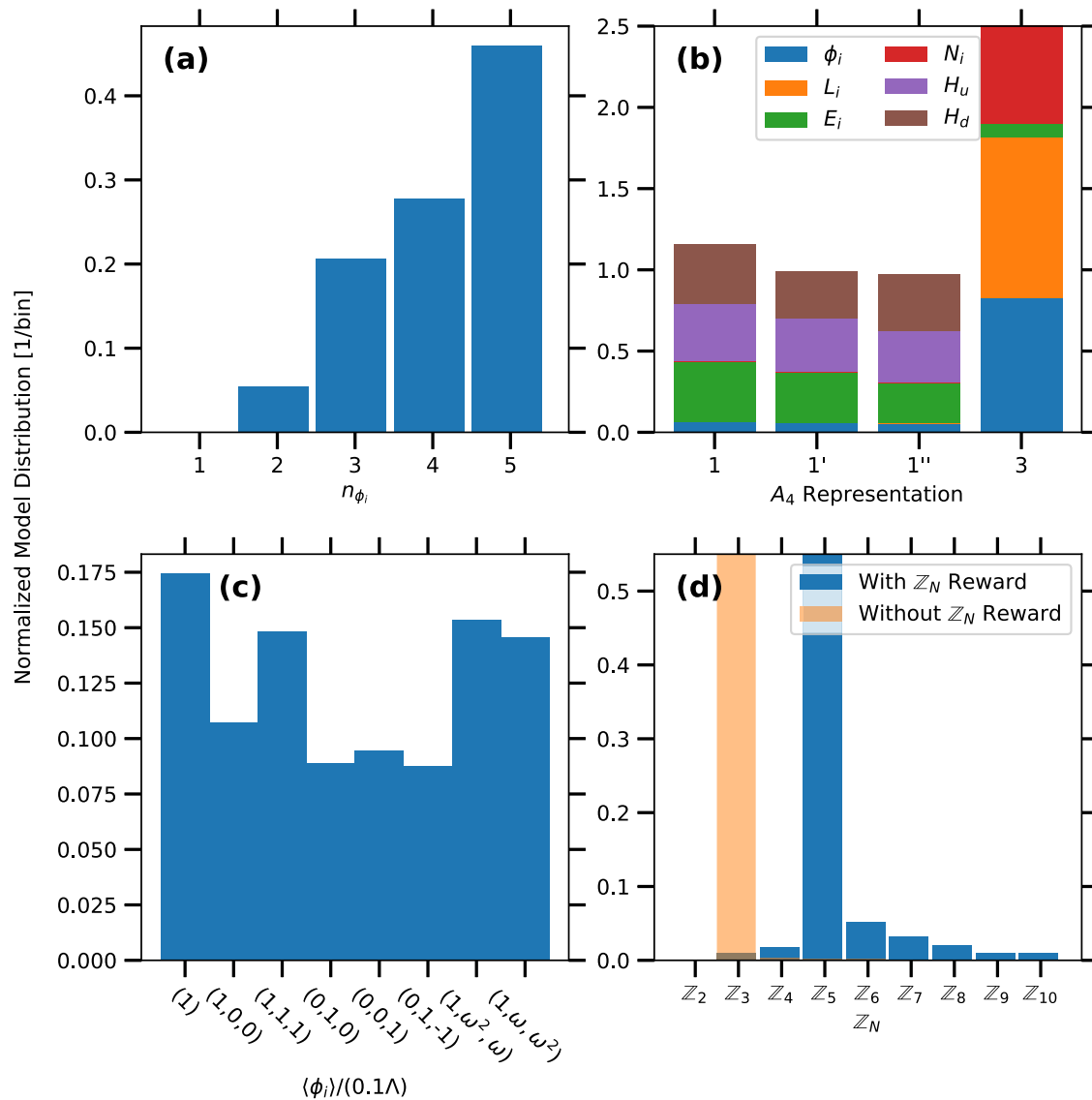


Fig. 5 | $A_4 \times \mathbb{Z}_N$ model space distributions of filtered models. **a** shows the distribution of the number of flavon fields, **b** the A_4 representations, **c** the distribution of vacuum alignments and **d** the $\mathbb{Z}_N$ symmetry. For comparison, **d** shows a configuration with and without the $R_{\mathbb{Z}}(N)$ reward in equation (9), and AMBer is found to

be more efficient at exploring large N with this reward included. The distribution fraction for $\mathbb{Z}_3$ models is ~ 0.9 without the $R_{\mathbb{Z}}(N)$ reward term, but is cut off to better display the distribution of higher order models. The vacuum alignment is normalized by the flavon breaking scale, 0.1Λ , i.e. $\langle \vec{\phi} \rangle / (0.1\Lambda)$ as in equation (4).

AMBer is found to focus on $\mathbb{Z}_3$. $\mathbb{Z}_2$ is very rarely employed. This is likely due to the fact that for $\mathbb{Z}_2$, it would be difficult to accommodate a well-fit model with a small number of parameters.

Search in $T_{19} \times \mathbb{Z}_4$ with $n_{\phi} \leq 6$

In the $T_{19} \times \mathbb{Z}_4$ space with up to six flavon multiplets, AMBer explores a subspace of models that has not been explored in the literature. This search space is more complicated due to the larger T_{19} group. In this group, there are six irreducible triplet representations which can be multiplied to form a number of singlet or triplet representations, as shown in Supplementary Tab. 2. Figure 6 shows the distribution of found models for this search. As compared to the A_4 run, a more uniform distribution of models across the different representations is found, although there is a slight preference to place E_i in singlets, L_i , N_i in respective triplets

This flavor symmetry allows AMBer to pack all lepton flavors, L_i , E_i , N_i , into their respective triplets, and still obtain the observed hierarchy. For example, consider the Dirac mass term for the

charged-lepton sector, $\bar{L}E\phi_i H_d$ where L and E are in the 3_1 and 3_2 representation, respectively. Their tensor product decomposes into a $3_2 \oplus 3_3 \oplus \bar{3}_3$, which can subsequently be made into a singlet if there is a flavon in any of the conjugate representations present in the decomposition. This also points to a certain flexibility in this search space, as there are generally many ways to form group invariant interactions, particularly for Majorana masses of the right-handed neutrinos. In order to reduce parameters, the $\mathbb{Z}_4$ charges are likely being tuned to reduce the free parameters for models.

A particularly simple model is shown in Table 3. The superpotential is given by

$$\begin{aligned} \mathcal{W} = & \hat{\alpha}_{(C)}[(L E_1 H_d \phi_2) + \alpha_1(L E_3 H_d \phi_1) + \alpha_2(L E_2 H_d \phi_1)] \\ & + \hat{\alpha}_{(M)}[(N N \phi_1 \phi_1) + \alpha_3(N N \phi_2 \phi_2)] + \hat{\alpha}_{(D)}[(N L H_u \phi_2) \\ & + \alpha_4(N L H_u \phi_3)]. \end{aligned} \quad (10)$$

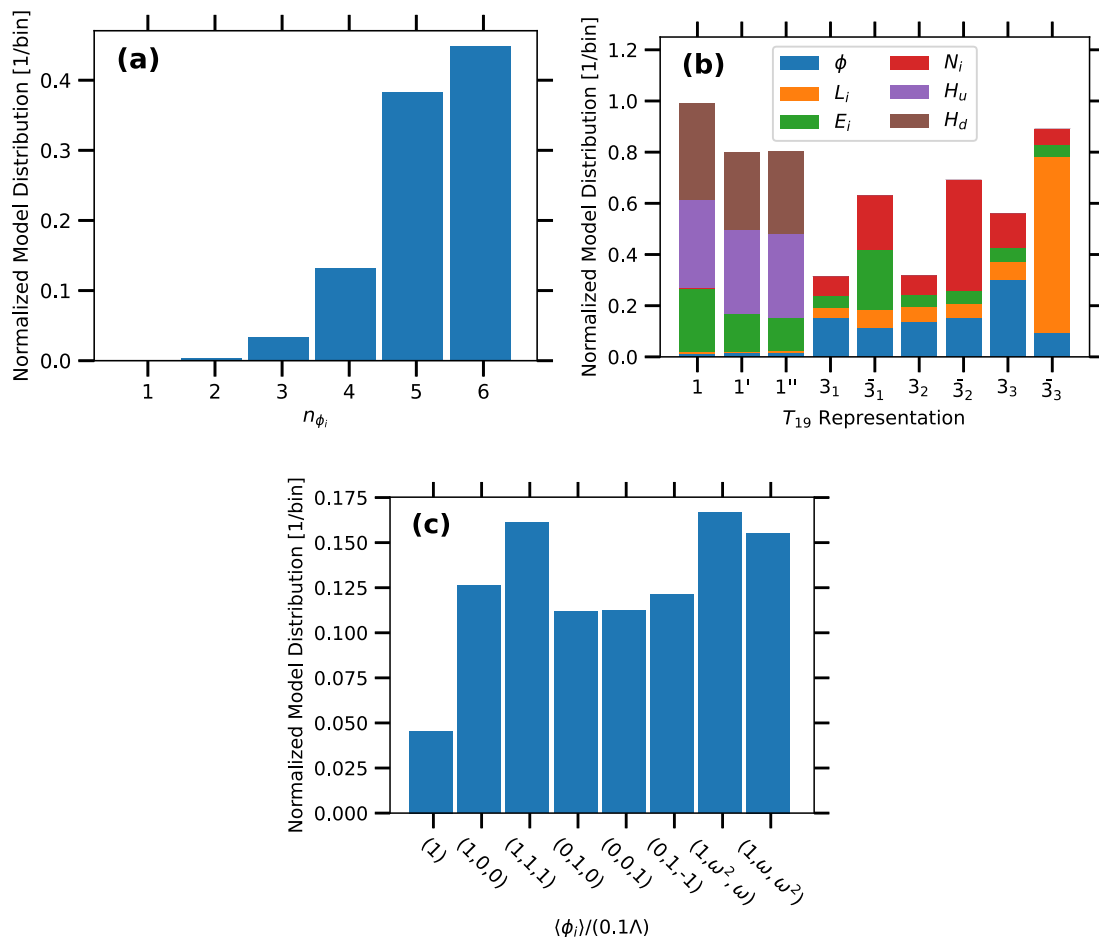


Fig. 6 | $T_{19} \times \mathbb{Z}_4$ model space distributions of filtered models. **a** shows the distribution of the number of flavon fields, **b** the T_{19} representations, and **c** the distribution of vacuum alignments. The vacuum alignment is normalized by the flavon breaking scale, 0.1Λ , i.e. $\langle \vec{\phi} \rangle / (0.1\Lambda)$ as in equation (4).

Table 3 | Example model found by AMBer in the $T_{19} \times \mathbb{Z}_4$ search

	L	E_1	E_2	E_3	N	H_u	H_d	ϕ_1	ϕ_2	ϕ_3
T_{19}	$\bar{3}_1$	$1'$	1	$1'$	3_3	$1'$	$1'$	3_1	3_1	3_3
$\mathbb{Z}_4$	0	1	3	3	0	2	1	0	2	2

It is discussed in further detail in the main text. Here, $\langle \phi_1 \rangle / (0.1\Lambda) = (1, \omega^2, \omega)$, $\langle \phi_2 \rangle / (0.1\Lambda) = (1, \omega, \omega^2)$, and $\langle \phi_3 \rangle / (0.1\Lambda) = (0, 0, 1)$.

The best-fit point in the parameter space of this model is at the values given by

$$\alpha_1 = 0.000271, \alpha_2 = 0.0565, \alpha_3 = -0.343, \alpha_4 = 0.915. \quad (11)$$

Hence, with only 4 free parameters, it yields $\chi^2 \approx 10$. This particular model places the charged lepton doublets and the right-handed neutrinos in separate triplets, and the right-handed charged leptons in singlets—an alternative model is shown in Supplementary Tab. 7 that has them all in triplets and has comparable performance. After the flavons acquire a VEV, the mass matrices can be written as

$$m_D = v_u \hat{\alpha}_{(D)} \begin{bmatrix} 0 & -\frac{1}{2} - i\frac{\sqrt{3}}{2} & \alpha_4 \\ 0 & 0 & -\frac{1}{2} + i\frac{\sqrt{3}}{2} \\ 1 & 0 & 0 \end{bmatrix},$$

$$m_C = v_d \hat{\alpha}_{(C)} \begin{bmatrix} -1/2 + i\sqrt{3}/2 & -\alpha_2(1/2 + i\sqrt{3}/2) & -\alpha_1(1/2 + i\sqrt{3}/2) \\ 1 & \alpha_2 & \sqrt{3}\alpha_1(-\sqrt{3}/6 - i/2) \\ (0.5 - i\sqrt{3}/2)^2 & \sqrt{3}\alpha_2(-\sqrt{3}/6 + i/2) & \alpha_1(1/2 - i\sqrt{3}/2)^2 \end{bmatrix}, \quad (12)$$

$$m_M = \Lambda \hat{\alpha}_{(M)} \begin{bmatrix} (-\alpha_3 - 1) + i\sqrt{3}(\alpha_3 - 1) & 0 & 0 \\ 0 & 2(\alpha_3 + 1) & 0 \\ 0 & 0 & (-\alpha_3 - 1) + i\sqrt{3}(-\alpha_3 + 1) \end{bmatrix},$$

where $\hat{\alpha}_{(C)}$, $\hat{\alpha}_{(M)}$ and $\hat{\alpha}_{(D)}$ are constants that can be factored out, because only dimensionless mass ratios are fit (see discussion at the end of Section “Neutrino flavor model-building”). The predicted values for the lepton observables are given by

$$\begin{aligned} m_e/m_\mu &= 0.0048, & m_\mu/m_\tau &= 0.0565, & \delta_{CP}^\ell/\pi &= 1.606 \text{ rad}, \\ \sin^2\theta_{12} &= 0.341, & \sin^2\theta_{13} &= 0.0222, & \sin^2\theta_{23} &= 0.467, \\ m_1 &= 28 \text{ meV}, & m_2 &= 30 \text{ meV}, & m_3 &= 57 \text{ meV}, \\ \eta_1/\pi &= 1.08 \text{ rad}, & \eta_2 &= 0.15 \text{ rad}. \end{aligned} \quad (13)$$

Here, m_i denote the neutrino masses and η_i denote the Majorana phases. Moreover, the sum of the neutrino masses, the effective electron-neutrino mass and the effective majorana mass is given by

$$\sum m_\nu = 116 \text{ meV}, m_{\nu_e}^{\text{eff}} = 30 \text{ meV}, \text{ and } m_{ee} = 28 \text{ meV}. \quad (14)$$

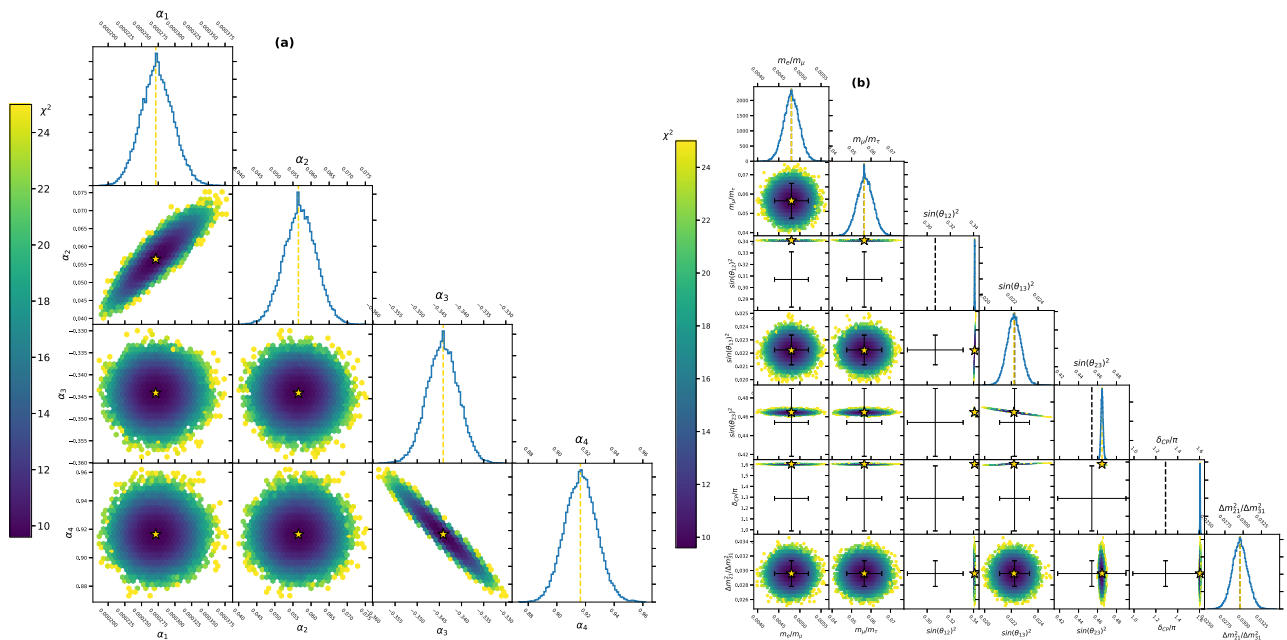


Fig. 7 | Corner plots for Markov-Chain Monte-Carlo exploration around the best-fit minimum for a particular model found in the $T_{19} \times \mathbb{Z}_4$. The best fit found the search is shown with the star indicating the best fit point. **a** shows the χ^2

distribution and correlations of the model's parameters, **b** shows the distribution of the model's predictions, along with the corresponding predictions with 2σ error bars.

We see that the constraints $m_{\nu}^{\text{eff}} < 0.45 \text{ eV}$ from KATRIN⁴⁵, $m_{ee} < 35 \text{ meV}$ from KamLAND-ZEN⁴⁶, and $\Sigma m_\nu < 0.12 \text{ eV}$ from Planck⁴⁷ are all satisfied for this model.

A Markov-Chain Monte-Carlo is performed around the minima, to explore correlations and map the χ^2 distribution. In Fig. 7a, the fitted parameters and their correlations are shown. While α_3 and α_4 are found to be $\mathcal{O}(1)$, α_1 and α_2 are not. These values are attributed to the requirement that α_1 and α_2 generate the hierarchical masses of the charged leptons. In Fig. 7b, the resulting fits to the observables are shown along with the experimentally measured values with 2σ error bars. While this model is simple and predictive, it struggles to fit θ_{12} and θ_{23} without eroding the goodness of fit of the remaining measurements. While not implemented here, it is possible that the diagnostics shown in this figure could be used as features for further explorations by AMBer. For example, one could penalize AMBer for fine-tuning parameters, or use correlations as feedback to find better models.

Conclusions

Particle physics model-building has traditionally relied on a physicist's intuition to explore fields and the symmetries they obey to describe the Universe we observe. However, the model space is vast, and a majority of it remains unexplored. AMBer is presented as an AI-assisted framework to explore the high-dimensional discrete space of neutrino flavor models. It uses reinforcement learning and high-performance computing to efficiently navigate the model space using an automated end-to-end pipeline. Existing scientific software was used and redesigned, and additional software was developed and released publicly. This pipeline constructs the superpotential, extracts the mass matrices, counts the number of free parameters, and fits the model to experimental data. It uncovers models that satisfy predictions which accurately match experimental observations and introduce a minimal amount of additional free parameters. These models serve as a starting point to evaluate the viability of a particular flavor symmetry to reproduce the lepton mixing parameters. Further investigations can be done on these models to incorporate the needed features of a full model. For instance, the calculation of theoretical error bars or an appropriate scalar potential could be studied, as discussed at the

end of subsection “Neutrino Flavor Model-Building” in the “Methods” section.

AMBer rediscovers known model patterns in the representation assignment of neutrino flavor models based on the symmetry group $A_4 \times \mathbb{Z}_N$. Additionally, it finds models based on the unexplored symmetry group $T_{19} \times \mathbb{Z}_4$. This allows study of the distributions of the particle content, representation assignments, preferred Abelian symmetry group, and flavon VEV alignment. To make this possible, AMBer is encouraged to find models with few free parameters and a good fit to data. At the same time, it is penalized for finding unrealistic models, such as those with zero electron mass. Several example models discovered by AMBer have been explicitly discussed in the main text and in the Appendix. In particular, a neutrino flavor model based on the group T_{19} with four free parameters, three flavons, and $\chi^2 \approx 10$ is presented. The complete list of inequivalent filtered models (i.e., models with $\chi^2 \leq 10$, $n_p \leq 7$) discovered has also been released in machine-readable format and is available on the GitHub repository [here](#). This list demonstrates AMBer's ability to find accurate models with few parameters. Thus, this tool can assist a physicist by filtering through a set of symmetries and field content to provide an initial set of models which can be further analyzed.

As the design of particle theory models becomes increasingly reliant on such assistants, there are interesting future research questions in how to quantitatively define the elegance of the theory, how to perform fast and accurate hypothesis tests when comparing models with different free parameters, and how to re-design scientific software to enable such searches in even more challenging model spaces.

From a machine learning perspective, it would be interesting to develop approaches that enable the agent to transfer knowledge acquired in one theory space to effectively explore another theory space. The ability for the agent to dynamically choose when to run physics software will enable scaling this approach to more complex software pipelines. Future work may also consider a more efficient representation of fields and symmetries to the agent, such that the agent is not given an option to choose invalid actions or generate equivalent models with seemingly different particle representation assignments. Computational aspects of the method can be improved to further leverage parallel computing capabilities at supercomputers,

increasing the rate at which AMBer can explore large theory spaces. With the growing use of Large Language Models (LLMs) for science, AMBer could be coupled with LLMs as reinforcement learning with physics feedback, with embedded software enhancing the reliability of scientific predictions.

From the physics perspective, it would be interesting to identify what properties make a flavor model realistic and whether this can be determined from the distribution of the particle content, representation assignment, or flavon VEV alignment. Another future direction is to run AMBer on a larger and more diverse set of non-Abelian flavor symmetry groups. Our approach is expected to be applicable to any finite non-Abelian flavor symmetry, provided that the corresponding mass matrices can be computed in a reasonable time. To this end, an automated method may be required to list out all of their irreducible representations. While the testability of models was not considered here, in future work, AMBer could also be explicitly incentivized to find models that can be ruled out by upcoming experimental results. At this stage, AMBer should be regarded as a proof of principle demonstrating that RL can assist model builders in performing more efficient and targeted searches. The impact of our work lies in our approach to interfacing an AI agent with physics software, while the models found in this work illustrate its potential. We envision that particle physicists in the future will be able to use AI assistants such as AMBer to rapidly explore alternative ideas for model building. Further, while this work focused on neutrino flavor model-building, the approach could be applied to other areas of flavor model-building, dark matter, or even cosmology, if the interfacing with scientific software can be scaled efficiently.

Data availability

Models found by AMBer are available in a Github repository at <https://github.com/jake-rudolph-1/models-by-AMBer>. They are stored as “.txt” files, each with a dictionary specifying all relevant information for a unique neutrino flavor model.

Code availability

The software developed for analyzing neutrino models within the RL pipeline has been publicly released as the FlavorBuilder package, installable via the Python Package Index (PyPI) at <https://pypi.org/project/FlavorBuilder/>. This code makes use of GAP version 4.14.0. The RL environments were developed using Gymnasium and run using Stable-Baselines3. They make further use of FlavorPy version 0.2.0.

Received: 24 June 2025; Accepted: 31 March 2026;

Published online: 01 May 2026

References

1. Fox, P. J. et al. Tf08 snowmass report: Bsm model building (2022). <https://arxiv.org/abs/2210.03075>. 2210.03075.
2. Feruglio, F. & Romanino, A. Lepton flavour symmetries (2021). <https://arxiv.org/abs/1912.06028>. 1912.06028.
3. Almunin, Y. et al. Neutrino flavor model building and the origins of flavor and CP violation. *Universe* **9**, 512 (2023).
4. de Gouvêa, A. et al. Theory of neutrino physics—Snowmass TF11 (aka NF08) topical group report. <https://arxiv.org/abs/2209.07983>. 2209.07983 (2022).
5. Fukuda, Y. et al. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.* **81**, 1562–1567 (1998).
6. Ahmad, Q. R. et al. Direct evidence for neutrino flavor transformation from neutral current interactions in the Sudbury Neutrino Observatory. *Phys. Rev. Lett.* **89**, 011301 (2002).
7. Cleveland, B. T. et al. Measurement of the solar electron neutrino flux with the Homestake chlorine detector. *Astrophys. J.* **496**, 505–526 (1998).
8. Ahn, M. H. et al. Indications of neutrino oscillation in a 250 km long baseline experiment. *Phys. Rev. Lett.* **90**, 041801 (2003).
9. Urban, J., Sutcliffe, G., Pudlák, P. & Vyskočil, J. Malarea sg1 - machine learner for automated reasoning with semantic guidance. In Armando, A., Baumgartner, P. & Dowek, G. (eds.) *Automated Reasoning*, 441–456 (Springer Berlin Heidelberg, 2008).
10. Jakubuv, J. & Urban, J. Hammering Mizar by Learning Clause Guidance. In Harrison, J., O’Leary, J. & Tolmach, A. (eds.) *10th International Conference on Interactive Theorem Proving (ITP 2019)*, vol. 141 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 34:1–34:8 (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019). <https://doi.org/10.4230/LIPIcs.ITP.2019.34>.
11. Blaauwbroek, L. et al. *Learning Guided Automated Reasoning: A Brief Survey*, 54–83 (Springer Nature Switzerland, 2024). https://doi.org/10.1007/978-3-031-61716-4_4.
12. Halverson, J., Nelson, B. & Ruehle, F. Branes with brains: exploring string vacua with deep reinforcement learning. *J. High Energy Phys.* **2019**. [https://doi.org/10.1007/JHEP06\(2019\)003](https://doi.org/10.1007/JHEP06(2019)003). (2019).
13. Nishimura, S., Miyao, C. & Otsuka, H. Exploring the flavor structure of quarks and leptons with reinforcement learning. *JHEP* **23**, 021 (2020).
14. Harvey, T. R. & Lukas, A. Quark mass models and reinforcement learning. *JHEP* **08**, 161 (2021).
15. Nishimura, S., Miyao, C. & Otsuka, H. Reinforcement learning-based statistical search strategy for an axion model from flavor. 2409.10023 (2024).
16. Wojcik, G. N., Eu, S. T. & Everett, L. L. Graph Reinforcement Learning for Exploring BSM Model Spaces. 2407.07203 (2024).
17. Carta, F., Gauntlett, A., Griffin, F. & He, Y.-H. BPS spectroscopy with reinforcement learning. 2501.14863 (2025).
18. Altarelli, G. & Feruglio, F. Discrete flavor symmetries and models of neutrino mixing. *Rev. Mod. Phys.* **82**, 2701–2729 (2010).
19. Ma, E. & Rajasekaran, G. Softly broken A(4) symmetry for nearly degenerate neutrino masses. *Phys. Rev. D.* **64**, 113012 (2001).
20. Altarelli, G. & Feruglio, F. Tri-bimaximal neutrino mixing, A(4) and the modular symmetry. *Nucl. Phys. B* **741**, 215–235 (2006).
21. Minkowski, P. $\mu \rightarrow e\gamma$ at a Rate of One Out of 10^9 Muon Decays? *Phys. Lett. B* **67**, 421–428 (1977).
22. Yanagida, T. Horizontal gauge symmetry and masses of neutrinos. *Conf. Proc. C.* **7902131**, 95–99 (1979).
23. Glashow, S. L. The future of elementary particle physics. *NATO Sci. Ser. B* **61**, 687 (1980).
24. Gell-Mann, M., Ramond, P. & Slansky, R. Complex spinors and unified theories. *Conf. Proc. C.* **790927**, 315–321 (1979).
25. Ishimori, H. et al. Non-abelian discrete symmetries in particle physics. *Prog. Theor. Phys. Suppl.* **183**, 1–163 (2010).
26. Feruglio, F. & Romanino, A. Lepton flavor symmetries. *Rev. Mod. Phys.* **93**, 015007 (2021).
27. Ding, G.-J. & Meloni, D. A model for tri-bimaximal mixing from a completely broken A_4 . *Nucl. Phys. B* **855**, 21–45 (2012).
28. Ludl, P. O. On the finite subgroups of U(3) of order smaller than 512. *J. Phys. A* **43**, 395204 (2010). [Erratum: *J. Phys. A* **44**, 139501 (2011)].
29. Holthausen, M. & Schmidt, M. A. Natural vacuum alignment from group theory: the minimal case. *JHEP* **01**, 126 (2012).
30. Navas, S. et al. Review of particle physics. *Phys. Rev. D.* **110**, 030001 (2024).
31. Baur, A. FlavorPy v0.2.0. <https://doi.org/10.5281/zenodo.11060597>. (2024).
32. Baur, A., Nilles, H. P., Ramos-Sanchez, S., Trautner, A. & Vaudrevange, P. K. S. The first string-derived eclectic flavor model with realistic phenomenology. *JHEP* **09**, 224 (2022).
33. Feruglio, F., Hagedorn, C., Lin, Y. & Merlo, L. Lepton flavour violation in models with A(4) flavour symmetry. *Nucl. Phys. B* **809**, 218–243 (2009).
34. Feruglio, F., Hagedorn, C., Lin, Y. & Merlo, L. Lepton flavour violation in a supersymmetric model with A(4) flavour symmetry. *Nucl. Phys. B* **832**, 251–288 (2010).

35. Antusch, S., Kersten, J., Lindner, M., Ratz, M. & Schmidt, M. A. Running neutrino mass parameters in see-saw scenarios. *JHEP* **03**, 024 (2005).
36. Criado, J. C. & Feruglio, F. Modular invariance faces precision neutrino data. *SciPost Phys.* **5**, 042 (2018).
37. The GAP Group. *GAP—Groups, Algorithms, and Programming, Version 4.14.0*. <https://www.gap-system.org>. (2024).
38. van Den Broek, P. M. & Cornwell, J. F. Clebsch-gordan coefficients of symmetry groups. *Phys. status solidi (b)* **90**, 211–224 (1978).
39. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (A Bradford Book, Cambridge, MA, USA, 2018).
40. Silver, D. et al. Mastering the game of go without human knowledge. *Nature* **550**, 354–359 (2017).
41. Inamdar, R., Sundarr, S. K., Khandelwal, D., Sahu, V. D. & Katal, N. A comprehensive review on safe reinforcement learning for autonomous vehicle control in dynamic environments. *e-Prime - Adv. Electr. Eng., Electron. Energy* **10**, 100810 (2024).
42. Baydin, A. G. et al. Etalumis: bringing probabilistic programming to scientific simulators at scale. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '19, 1–24 (ACM, 2019). <https://doi.org/10.1145/3295500.3356180>.
43. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. *CoRRabs/1707.06347* (2017). <http://arxiv.org/abs/1707.06347>.
44. Raffin, A. et al. Stable-baselines3: Reliable reinforcement learning implementations. *J. Mach. Learn. Res.* **22**, 1–8 (2021).
45. Aker, M. et al. Direct neutrino-mass measurement based on 259 days of KATRIN data. *Science* **388**, adq9592 (2025).
46. Abe, S. et al. Search for the Majorana Nature of Neutrinos in the Inverted Mass Ordering Region with KamLAND-Zen. *Phys. Rev. Lett.* **130**, 051801 (2023).
47. Aghanim, N. et al. Planck 2018 results. VI. Cosmological parameters. *Astron. Astrophys.* **641**, A6 (2020).

Acknowledgements

AG and JR used resources from the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, for this research. We specifically thank Wahid Bhimji for the support with computing resources that were essential for this work. The work of MF was supported in part by U.S. National Science Foundation Grant PHY-2210283 and the National Science Foundation Graduate Research Fellowship Award No. DGE-1839285. The work of VKP was supported by the U.S. National Science Foundation under Grant No. PHY-2210283 and by UC-MEXUS-CONACyT grant No. CN-20-38. The work of JB was supported by DOE, High Energy Physics Research and Technology Division under grant DE-SC0023966, and the auto-encoder was trained using the Greenplanet cluster (NSF Grant No. CHE- 0840513). DW and AG are supported by the DOE Office of Science. AG thanks Josef Urban for the invitation to Prague in 2016 and fruitful discussions that seeded core ideas of this work, as well participants of the AI for theorem proving conference in 2018, such as Michael Douglas and David McAllester. We would also like to thank Mu-

Chun Chen, Jonathan Feng, Xueqi Li, Xiang-Gan Liu, Michael Ratz, and Miša Toman for fruitful discussions. We thank Wahid Bhimji, Jim Halverson, Alexander Shmakov, Jessica Howard, Michael Ratz, Tim Tait, and Shimon Whiteson for providing valuable comments on the manuscript.

Author contributions

Using the CASRAI CRediT Contributor Roles Taxonomy: Conceptualization: M.F., V.K.P., J.R., J.B., A.G.; Software: J.R., V.K.P.; Visualization: J.B., M.F., J.R.; Resources: A.G., D.W.; Project administration: J.R., A.G.; Supervision: A.G., D.W., Validation: J.R., V.K.P., M.F., A.G., D.W., J.B., V.G.; Data curation: J.R., J.B.; Investigation: J.R., M.F., J.B., V.K.P.; Formal analysis: M.F.; Methodology: A.G., J.R., V.K.P., M.F., J.B.; Writing—original draft: V.K.P., J.R., M.F., A.G., J.B.; Writing—review and editing: D.W., J.R., M.F., V.K.P., A.G.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42005-026-02627-2>.

Correspondence and requests for materials should be addressed to Aishik Ghosh.

Peer review information *Communications Physics* thanks the anonymous reviewers for their contribution to the peer review of this work.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026