

Quark Mass Models and Reinforcement Learning

T. R. Harvey and A. Lukas

*Rudolf Peierls Centre for Theoretical Physics, Oxford University,
1 Keble Road, Oxford, OX1 3NP, U.K.*

E-mail: thomas.harvey@physics.ox.ac.uk, andre.lukas@physics.ox.ac.uk

ABSTRACT: In this paper, we apply reinforcement learning to the problem of constructing models in particle physics. As an example environment, we use the space of Froggatt-Nielsen type models for quark masses. Using a basic policy-based algorithm we show that neural networks can be successfully trained to construct Froggatt-Nielsen models which are consistent with the observed quark masses and mixing. The trained policy networks lead from random to phenomenologically acceptable models for over 90% of episodes and after an average episode length of about 20 steps. We also show that the networks are capable of finding models proposed in the literature when starting at nearby configurations.

KEYWORDS: Beyond Standard Model, Quark Masses and SM Parameters

ARXIV EPRINT: [2103.04759](https://arxiv.org/abs/2103.04759)

Contents

1	Introduction	1
2	Background	3
2.1	Reinforcement learning	3
2.2	Froggatt-Nielsen models	4
3	Mass models and reinforcement learning	7
3.1	The environment	8
3.2	Neural network	9
3.3	Agent	9
4	Results	10
4.1	One $U(1)$ symmetry	10
4.2	Two $U(1)$ symmetries	13
5	Conclusion and outlook	15
A	Example models for one $U(1)$ symmetry	17
B	Example models for two $U(1)$ symmetries	18

1 Introduction

Machine learning in particle and string theory has developed into a fruitful and growing area of interdisciplinary research, triggered by the work in refs. [1, 2]. (For a review and a comprehensive list of references see ref. [3].) Much of the activity to date has been in the context of supervised learning (see, for example, refs. [4–11]), where data sets which arise in physics or related areas of mathematics have been used to train neural networks. However, there has also been some interesting work using reinforcement learning (RL), particular in relation to string model building [12, 13].

In the present paper, we are interested in reinforcement learning with environments which consist of classes of particle physics models. More precisely, we would like to address the following question. Can techniques of reinforcement learning be used to train a neural network to construct particle physics models with certain prescribed properties? At its most ambitious, such a network might be used to explore large classes of quantum field theories in view of their consistency with experimental data, thereby facilitating the search for physical theories beyond the standard model of particle physics. However, such a wide-ranging approach would require considerably conceptual work as well as computing

resources and does not seem feasible for a first exploration. (For a different approach to quantum field theory via methods of machine learning see ref. [14].)

For this reason, we will focus on a much more limited arena of particle physics models which can be relatively easily described and where extracting relevant physics properties is straightforward. Specifically, we will consider Froggatt-Nielsen (FN) models of fermion masses [15, 21–25], focusing on the quark sector. (For related early work on mass model building with horizontal U(1) symmetries see also refs. [16–20].)

The standard model of particle physics contains the up and down quark Yukawa couplings Y_{ij}^u and Y_{ij}^d , where $i, j, \dots = 1, 2, 3$ label the three families. Within the standard model, these couplings are mere parameters inserted “by hand”. Upon diagonalisation, they determine the masses $(m_{u,i}) = (m_u, m_c, m_t)$ and $(m_{d,i}) = (m_d, m_s, m_b)$ of the up and down type quarks as well as the CKM mixing matrix V_{CKM} .

FN models attempt to explain the values of Y_{ij}^u and Y_{ij}^d by introducing $U_a(1)$ symmetries, where $a = 1, \dots, r$, and singlet fields ϕ_α , where $\alpha = 1, \dots, \nu$, in addition to the structure present in the standard model. The idea is that the Yukawa couplings are either zero, if forbidden by the $U_a(1)$ symmetries, or given in terms of the vacuum expectation values (VEVs) $\langle \phi \rangle$ of the scalar fields, such that $Y_{ij}^u \sim \langle \phi \rangle^{n_{ij}}$ and $Y_{ij}^d \sim \langle \phi \rangle^{m_{ij}}$. Here, n_{ij} and m_{ij} are (non-negative) integers whose values are determined by $U_a(1)$ invariance of the associated operator. A FN model is easily described by its charge matrix $(\mathcal{Q}_a^I) = (q_a(Q^i), q_a(u^i), q_a(d^i), q_a(H), q_a(\phi))$, where q_a denotes the charge with respect to $U_a(1)$, Q_i are the left-handed quark-doublets, u_i and d_i are the right-handed up and down quarks and H is the Higgs doublet. (As we will discuss, the VEVs $\langle \phi_\alpha \rangle$, which may also be considered as part of the definition of a FN models, will be fixed to certain optimal values for a given charge assignment.) We can, therefore, think of the space of FN models as the space of charge matrices \mathcal{Q} . For practical reasons, we will impose limits, $q_{\min} \leq \mathcal{Q}_{aI} \leq q_{\max}$, on the entries of this matrix, so that the space of models becomes finite. However, note that, even for one U(1) symmetry ($r = 1$), one singlet ($\nu = 1$) and a modest charge range $-q_{\min} = q_{\max} = 9$ we have of the order of 10^{13} models. For two U(1) symmetries, two singlets and for the same charge range this number rises to roughly 10^{28} . This is quite sizeable, even though it is small compared to typical model numbers which arise in string theory. At any rate, given these numbers, systematic scanning of all or a significant fraction of the state space is clearly not practical or even feasible. Exploring such large environments requires different methods and this is where RL comes into play.

The idea of RL is to train a neural network with data obtained by exploring an environment, subject to a goal defined by a reward function. (See, for example, ref. [26] for an introduction.) It has been shown that RL can lead to impressive performance, even for very large environments, where systematic scanning is impossible [27]. It is, therefore, natural to ask whether RL can help explore the large model environments realised by quantum field theory and string theory. In the present paper, we will use RL to explore the space of FN models for the quark sector. More specifically, our environment consists of the set $\{\mathcal{Q}\}$ of all FN charge matrices for a given number, r , of U(1) symmetries, a given number, ν , of singlets ϕ_α and charges constrained by $q_{\min} \leq \mathcal{Q}_{aI} \leq q_{\max}$. An action within this environment simply amounts to adding or decreasing one of the charges \mathcal{Q}_a^I by one and

a reward is computed based on how well the models reproduce the experimental quark masses and mixings. A terminal state is one that reproduces the experimental masses and mixing to a given degree of accuracy.

We use a simple policy-based RL algorithm, with a single policy network whose input is, essentially, the charge matrix \mathcal{Q} and whose output is an action. The hope is that a successfully trained policy network of this kind will produce episodes starting from arbitrary (and typically physically unacceptable) FN models and efficiently lead to phenomenologically viable FN models.

The plan of the paper is as follows. In the next section, we briefly review the theoretical background of this work, namely RL and FN model building, mainly to set the scene and fix notation. In section 3 we describe our RL set-up and section 4 presents the results we obtained for the cases of one singlet and one U(1) symmetry and two singlets and two U(1) symmetries. The appendices contain a number of interesting FN models found by the neural network.

2 Background

2.1 Reinforcement learning

We start with a quick overview of RL, focusing on the aspects needed for this paper. For a comprehensive review see, for example, ref. [26] and [3].

The main components of an RL system are the *environment*, the *agents* and the *neural network(s)*. The latter are set up to learn certain properties of the environment, based on data delivered as the agent explores the environment. The mathematical underpinning of RL is provided by a *Markov decision process* (MDP), defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, \mathcal{R})$. Here \mathcal{S} is a set which contains the environment's states, \mathcal{A} is a set of maps $\alpha : \mathcal{S} \rightarrow \mathcal{S}$ which represent the actions, \mathcal{P} provides a probability $\mathbb{P}(S = s' | S = s, A = \alpha)$ for a transition from state s to state s' via the action α , $\gamma \in [0, 1]$ is called the *discount factor* and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the *reward function*. Among the states in \mathcal{S} a subset of so-called *terminal states* is singled out which may, for example, consist of states with certain desirable properties. Within this set-up we can consider a sequence

$$s_0 \xrightarrow{\alpha_0, r_0} s_1 \xrightarrow{\alpha_1, r_1} s_2 \xrightarrow{\alpha_2, r_2} s_3 \cdots$$

of states s_t and actions α_t , producing rewards r_t , where $t = 0, 1, 2, \dots$, which is referred to as an *episode*. In principle, an episode can have infinite length, although in practice a finite maximal episode length, N_{ep} , is imposed. If an episode arrives at a terminal state before it reaches its maximal number of steps it is stopped. The return, G_t , of a state s_t in such an episode is defined as

$$G_t = \sum_{k \geq 0} \gamma^k r_{t+k}. \tag{2.1}$$

The discount factor γ can be dialled to small values in order to favour short-term rewards dominating the return, or to values close to one so that longer-term rewards affect the return as well. The choice of action in a MDP is guided by a *policy* π , which provides

probabilities $\pi(\alpha|s) = \mathbb{P}(A_t = \alpha|S_t = s)$ for applying a certain action α to a state s . Relative to such a policy, two important value functions, namely the *state value function* V_π and the *state-action value function* Q_π , can be defined as expectation values of the return.

$$V_\pi(s) = \mathbb{E}(G_t|S_t = s), \quad Q_\pi(s, \alpha) = \mathbb{E}(G_t|S_t = s, A_t = \alpha). \quad (2.2)$$

The purpose of an RL system is to maximise a value function (state or state-action) over the set of possible policies. In practice, this can be realised in a number of ways which differ by which of the functions π , V_π and Q_π are represented by neural networks and how precisely these neural networks are trained via exploration of the environment. Common to all algorithms is an iterative approach, where a batch of data, in the form of triplets (s_t, a_t, G_t) , is collected from episodes which are guided by the neural network(s) in their present state. This data is then used to update the neural network(s), followed by a further round of exploration and so on.

For our purposes, we will consider what is probably the simplest approach, a basic policy-based algorithm referred to as REINFORCE. This set-up contains a single neural network π_θ with weights θ which represents the policy π . Its inputs are states and the outputs are probabilities for actions. Exploration of the environment is guided by the policy, meaning the steps in an episode are selected based on π_θ , so

$$s_0 \xrightarrow{\pi_\theta} s_1 \xrightarrow{\pi_\theta} s_2 \xrightarrow{\pi_\theta} s_3 \cdots. \quad (2.3)$$

Data is collected by performing such episodes successively, so we can say that the system contains only one agent. According to the *policy-gradient theorem*, the neural network π_θ should be trained on the loss function

$$L(\theta) = Q_\pi(s, a) \ln(\pi_\theta(s, a)), \quad (2.4)$$

where $Q_\pi(s, a)$ can, in practice, be replaced by the return G of the state s . Schematically, the algorithm then proceeds as follows.

- (1) Initialise the policy network π_θ .
- (2) Collect a batch of data triplets (s_t, a_t, G_t) from as many episodes (2.3) as required. New episodes start at random states s_0 .
- (3) Use this batch to update the weights θ of the policy network π_θ , based on the loss (2.4).
- (4) Repeat from (2) until the loss is sufficiently small so that the policy has converged.

2.2 Froggatt-Nielsen models

Before we discuss Froggatt-Nielsen models, we quickly review fermion masses in the standard model of particle physics, in order to set up notation and present the experimental data.

m_u	m_d	m_c	m_s	m_t	m_b
$0.00216_{-0.00026}^{+0.00049}$	$0.00467_{-0.00017}^{+0.00048}$	1.27 ± 0.02	$0.093_{-0.005}^{+0.011}$	172.4 ± 0.07	$4.18_{-0.02}^{+0.03}$
v	s_{12}	s_{13}	s_{23}	δ	
~ 174	0.22650 ± 0.00048	$0.00361_{-0.00009}^{+0.00011}$	$0.04053_{-0.00061}^{+0.00083}$	$1.196_{-0.043}^{+0.045}$	

Table 1. Experimentally measured masses in GeV and mixing angles of quarks from ref. [28].

The standard model contains Yukawa interactions, which are responsible for generating the masses and mixing of quarks and leptons. In this paper, we focus on the quark sector for simplicity, although we expect that our work can be generalised to include the lepton sector. The quark Yukawa couplings in the standard model take the form

$$\mathcal{L}_{\text{Yuk}} = Y_{ij}^u \bar{Q}^i H^c u^j + Y_{ij}^d \bar{Q}^i H d^j + \text{h.c.}, \quad (2.5)$$

where Q_i are the left-handed quarks, u_i, d_i are the right-handed up and down type quarks and H is the Higgs doublet. We use indices $i, j, \dots = 1, 2, 3$ to label the three families. Within the standard model, the Yukawa matrices Y^u and Y^d are not subject to any theoretical constraints — their (generally complex) values are inserted “by hand” in order to fit the experimental results for masses and mixing.

Once the charge-neutral component H^0 in the Higgs doublet develops a VEV, $v = \langle H^0 \rangle$, the above Yukawa terms lead to Dirac mass terms with associated mass matrices

$$M_u = v Y^u, \quad M_d = v Y^d. \quad (2.6)$$

These matrices need to be diagonalised,

$$M_u = U_u \hat{M}_u V_u^\dagger, \quad M_d = U_d \hat{M}_d V_d^\dagger, \quad \text{where } \hat{M}_u = \text{diag}(m_u, m_c, m_t), \quad \hat{M}_d = \text{diag}(m_d, m_s, m_b), \quad (2.7)$$

by unitary matrices U_u, V_u, U_d, V_d in order to obtain the masses (m_u, m_c, m_t) for the up-type quarks and the masses (m_d, m_s, m_b) for the down-type quarks. The other observable quantity obtained from the quark Yukawa couplings is the Cabibbo-Kobayashi-Maskawa (CKM) matrix V_{CKM} , defined by

$$V_{\text{CKM}} = U_u^\dagger U_d = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix}. \quad (2.8)$$

The CKM matrix is unitary and can, hence, be written in terms of three angles $\theta_{12}, \theta_{13}, \theta_{23}$ and a phase δ , as in the above equation, where the abbreviations $s_{ij} = \sin(\theta_{ij})$ and $c_{ij} = \cos(\theta_{ij})$ have been used. The experimentally measured values for these quantities are given in table 1. Inserting the mixing angles and the phase from table 1 into the parametrisation (2.8) gives the numerical CKM matrix

$$|V_{\text{CKM}}| \approx \begin{pmatrix} 0.9740 & 0.2265 & 0.0036 \\ 0.2264 & 0.9732 & 0.0405 \\ 0.0085 & 0.0398 & 0.9992 \end{pmatrix}. \quad (2.9)$$

In the context of the standard model, the Yukawa matrices Y^u and Y^d in eq. (2.5) have to be chosen to fit these experimental values for masses and mixing but this still leaves considerable freedom. Only 10 real constraints are imposed on the 36 real parameters which determine Y^u and Y^d .

Froggatt-Nielsen (FN) models provide a framework for adding more structure to the Yukawa sector of the standard model, in an attempt to remove some of this ambiguity and provide a theoretical explanation for the observed masses and mixing. Two main ingredients are added to the picture: a number of global $U(1)$ symmetries $U_a(1)$, where $a = 1, \dots, r$, and a number of complex scalar fields ϕ_α , where $\alpha = 1, \dots, \nu$, which are singlets under the standard model gauge group. The standard model fields as well as the scalar singlets are assigned $U_a(1)$ charges which we denote by $q_a(Q^i)$, $q_a(u^i)$, $q_a(d^i)$, $q_a(H)$ and $q_a(\phi_\alpha)$. In fact, to simplify matters, we assume that we have the same number of $U(1)$ symmetries and singlet fields, $\nu = r$, and that the a^{th} singlet ϕ^a is only charged under $U_a(1)$. The resulting singlet charges are then denoted by $q_a(\phi)$.

Given this set-up, the standard model Yukawa couplings (2.5) are no longer in general consistent with the $U_a(1)$ symmetries and should be replaced by

$$\mathcal{L}_{\text{Yuk}} = \sum_{i,j} \left(a_{ij} \phi_1^{n_{1,ij}} \dots \phi_r^{n_{r,ij}} \bar{Q}^i H^c u^j + b_{ij} \phi_1^{m_{1,ij}} \dots \phi_r^{m_{r,ij}} \bar{Q}^i H d^j \right) + \text{h.c.} \quad (2.10)$$

where $n_{a,ij}$ and $m_{a,ij}$ are non-negative integers. For a term (ij) in the up-quark sector to be invariant under $U_a(1)$ we require the conditions

$$n_{a,ij} = - \frac{q_a(\bar{Q}^i H^c u^j)}{q_a(\phi)}. \quad (2.11)$$

Hence, the term (ij) in the u-quark sector is allowed if the $n_{a,ij}$ given by eq. (2.11) are non-negative integers for all $a = 1, \dots, r$. In this case, the coefficient a_{ij} is of order one, otherwise it is set to zero. An analogous rule applies to the terms for the down-type quarks. Once the scalars ϕ_a develop VEVs, $v_a = \langle \phi_a \rangle$, Yukawa couplings

$$Y_{ij}^u = a_{ij} v_1^{n_{1,ij}} \dots v_r^{n_{r,ij}}, \quad Y_{ij}^d = b_{ij} v_1^{m_{1,ij}} \dots v_r^{m_{r,ij}}. \quad (2.12)$$

are generated.¹ The main model building idea in this setting is that moderately small singlet VEVs v_a can generate the required large hierarchies in masses, in a way that is controlled by the integers $n_{a,ij}$ and $m_{a,ij}$ and, hence, ultimately, by the choices of $U_a(1)$ charges.

At this stage the environment of FN models consists of $U_a(1)$ charges for all fields, the singlet VEVs v_a and the coefficients a_{ij} , b_{ij} . In principle, the singlet VEVs are meant to be fixed by a scalar potential but implementing this in detail adds another layer of model building. Instead, for a given choice of charges and coefficients a_{ij} , b_{ij} , we will fix the VEVs v_a such that the model provides an optimal fit to the experimental masses and mixing. Note this does not imply that the VEVs are inserted “by hand”. Rather, for each

¹If these Yukawa couplings are generated at a high energy scale they have to be renormalised down to the electro-weak scale, in order to facilitate comparison with the experimental values. Since this typically leads to order one coefficients which have already been included via a_{ij} , b_{ij} we will not consider this explicitly.

state, that is, for each set of charges, the system determines the best choices for these VEVs in view of matching the data. This means the RL system returns both the charges as well as the VEVs of a model. The non-zero coefficients a_{ij} , b_{ij} might be considered as part of the environment definition but, to keep things simply, we will fix those to specific numerical values of order one. While, in general, a_{ij} and b_{ij} can be complex, we simplify this scenario by only allowing them to take real values. Consequently, we will not attempt to fit the CP violating phase δ in the CKM matrix. As a further simplification, we require that the top Yukawa term $\bar{Q}^3 H^c u^3$ is present without any singlet insertions, a condition which seems reasonable given the size of the top Yukawa coupling. This requirement can be used to fixed the $U_a(1)$ charges of the Higgs multiplet as

$$q_a(H) = q_a(u^3) - q_a(Q^3). \tag{2.13}$$

Altogether, this means a FN model within our set-up is specified by the charges choices

$$\left(\mathcal{Q}_a^I\right) = \left(q_a(Q^i), q_a(u^i), q_a(d^i), q_a(\phi)\right), \tag{2.14}$$

which we have assembled into the $r \times 10$ integer charge matrix \mathcal{Q} . In practice, the charges in \mathcal{Q} will be restricted to a certain range

$$q_{\min} \leq \mathcal{Q}_a^i \leq q_{\max}, \tag{2.15}$$

with q_{\min} and q_{\max} to be specified later. While this leads to a finite space of charge matrices and associated FN models, numbers can be considerable. For example, for $-q_{\min} = q_{\max} = 9$ we have $\sim 10^{13}$ models in the case of a single $U(1)$ symmetry and $\sim 10^{26}$ models for the case of two $U(1)$ symmetries.

The environment (2.14) of FN models has a number of permutation degeneracies, since the assignment of charges to families and the order of $U_a(1)$ symmetries does not carry physical meaning, although part of this symmetry is broken by designating Y_{33}^u the top Yukawa coupling. This means there is a permutation degeneracy isomorphic to

$$S_2 \times S_2 \times S_3 \times S_r \tag{2.16}$$

in the environment (2.14). For the purpose of RL we will not attempt to remove this redundancy, as this would complicate the constraints on the charges in \mathcal{Q} .

From the viewpoint of particle physics the task is now to investigate the model landscape defined by eq. (2.14) and extract the phenomenologically promising cases. Considerable effort has been invested into this, since the original proposal of Froggatt and Nielsen [15]. It is precisely this task we wish to carry out using reinforcement learning.

3 Mass models and reinforcement learning

We now explain how we propose to map the problem of FN model building onto the structure of reinforcement learning. We begin by describing the set-up of the RL environment.

3.1 The environment

We need to identify how the various ingredients of a MDP are realised in our context. We take the set \mathcal{S} of states to consist of all FN models for a fixed number, r , of U(1) symmetries and the same number of singlet fields. These models are represented by the $r \times 10$ integer charge matrices \mathcal{Q} in eq. (2.14), with entries restricted as in eq. (2.15). The set \mathcal{A} of actions α consists of the basic operations

$$\mathcal{Q}_a^I \xrightarrow{\alpha} \mathcal{Q}_a^I \pm 1, \tag{3.1}$$

that is, increasing or decreasing a single charge \mathcal{Q}_a^I by one while keeping all other charges unchanged. These are deterministic actions so we do not need to introduce transition probabilities \mathcal{P} . The number of different actions is $2 \times r \times 10 = 20r$. For the discount factor γ we choose the value $\gamma = 0.98$.

Defining the reward function \mathcal{R} requires a bit more effort. We start by defining the intrinsic value for a state \mathcal{Q} as

$$\mathcal{V}(\mathcal{Q}) = - \min_{|v_a| \in I} \sum_{\mu} \left| \log_{10} \left(\frac{|\mu_{\mathcal{Q}, v_a}|}{|\mu_{\text{exp}}|} \right) \right|. \tag{3.2}$$

Here, μ runs over the six quark masses as well as the entries of the CKM matrix, $\mu_{\mathcal{Q}, v_a}$ is the value for one of these quantities predicted by the model with charge matrix \mathcal{Q} and scalar fields VEVs v_a , computed from eqs. (2.12), (2.6), (2.7), (2.8) (using fixed random values of the order-one coefficients a_{ij} , b_{ij}), and μ_{exp} is its experimental value as given in table 1 and eq. (2.9). The minimisation is carried out over the scalar field VEVs v_a , in a certain range $I = [v_{\text{min}}, v_{\text{max}}]$, with typical values $v_{\text{min}} = 0.01$ and $v_{\text{max}} = 0.3$. From this definition, the intrinsic value of a state \mathcal{Q} is simply the (negative) total order of magnitude by which predicted masses and mixings deviate from the experimental ones, for optimal choices of the scalar field VEVs.

We have deliberately chosen a value function which checks order of magnitude agreement, rather than one which measures the quality of a state relative to the experimental errors of the masses and mixings. This is because the information from U(1) charges and resulting powers of VEVs which make up our environment are only expected to get to the correct order of magnitude. Finer adjustments can be made by choosing the order one coefficients a_{ij} and b_{ij} which are not fixed by the U(1) symmetries. However, for simplicity we have opted to fix these coefficient, rather than make them part of the environment. A check based on experimental error would, therefore, be too sensitive and miss many models which may become acceptable after a suitable adjustment of these order one coefficients.

A terminal state \mathcal{Q} in our environment is one which is phenomenologically promising, that is, a state which gives rise to (roughly) the correct masses and mixings. More specifically, we call a state terminal if its intrinsic value $\mathcal{V}(\mathcal{Q})$ is larger than a certain threshold value \mathcal{V}_0 and if each individual deviation $-\log_{10}(|\mu_{\mathcal{Q}}|/|\mu_{\text{exp}}|)$ (computed for the scalar field VEVs which minimise eq. (3.2)) is larger than a threshold value \mathcal{V}_1 . Since we have fixed our order-one parameters a_{ij} , b_{ij} these threshold values are chosen relatively generously, so as to not miss any promising models. For our computations, we have used $\mathcal{V}_0 = -10$ and $\mathcal{V}_1 = -1$.

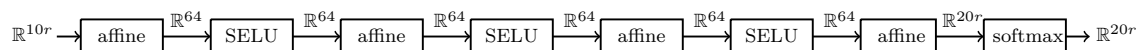
Based on this intrinsic value, the reward $\mathcal{R}(\mathcal{Q}, \alpha)$ for an action $\mathcal{Q} \xrightarrow{\alpha} \mathcal{Q}'$ of the form (3.1), connecting two states \mathcal{Q} and \mathcal{Q}' , is defined by

$$\mathcal{R}(\mathcal{Q}, \alpha) = \begin{cases} \mathcal{V}(\mathcal{Q}') - \mathcal{V}(\mathcal{Q}) & \text{if } \mathcal{V}(\mathcal{Q}') - \mathcal{V}(\mathcal{Q}) > 0 \\ \mathcal{R}_{\text{offset}} & \text{if } \mathcal{V}(\mathcal{Q}') - \mathcal{V}(\mathcal{Q}) \leq 0 \end{cases} . \quad (3.3)$$

Here, $\mathcal{R}_{\text{offset}}$ is a fixed (negative) value which penalises a decrease of the intrinsic value, typically chosen as $\mathcal{R}_{\text{offset}} = -10$. In addition, if the new state \mathcal{Q}' is terminal a terminal bonus $\mathcal{R}_{\text{term}}$, typically chosen as $\mathcal{R}_{\text{term}} = 100$, is added to the reward (3.3).

3.2 Neural network

To represent the policy π , we use a fully connected network f_{θ} with the following structure.



Here, “affine” refers to an affine layer performing the transformation $\mathbf{x} \rightarrow W\mathbf{x} + \mathbf{b}$ with weight W and bias \mathbf{b} , “SELU” is the standard scaled exponential linear unit activation function and “softmax” is a softmax layer which ensures that the output can be interpreted as a vector of probabilities which sum to one. The input of this network is the charge matrix \mathcal{Q} , in line with the input dimension of $10r$ while the output is a probability vector whose dimension, $20r$, equals the number of different actions (3.1).

Training data is provided in batches which consist of triplets $(\mathcal{Q}_t, \alpha_t, G_t)$, where the actions α_t are represented by a standard unit vector in \mathbb{R}^{20r} . The probability of an action can then be written as $\pi_{\theta}(\mathcal{Q}_t, \alpha_t) = \alpha_t \cdot f_{\theta}(\mathcal{Q}_t)$ and the loss (2.4) takes the form

$$L(\theta) = G_t \ln(\alpha_t \cdot f_{\theta}(\mathcal{Q}_t)) . \quad (3.4)$$

Based on this loss function, the above network is trained with the ADAM optimiser, using batch sizes of 32 and a typical learning rate of $\lambda = 1/4000$.

3.3 Agent

The FN environment will be explored by a single agent, following episodes (2.3) of maximal length $N_{\text{ep}} = 32$, and guided by the policy network π_{θ} . Each new episode is started from a random state, to improve exploration of the environment. Terminal states which are encountered during training are stored for later analysis.

The FN environment and the REINFORCE algorithm are realised as MATHEMATICA [29] packages, the latter based on the MATHEMATICA suite of machine learning modules. For terminal states found during training or by applying the trained network we perform a further Monte Carlo analysis in the space of order one coefficients a_{ij}, b_{ij} (which were held fixed during training) in order to optimise their intrinsic value $\mathcal{V}(\mathcal{Q})$.

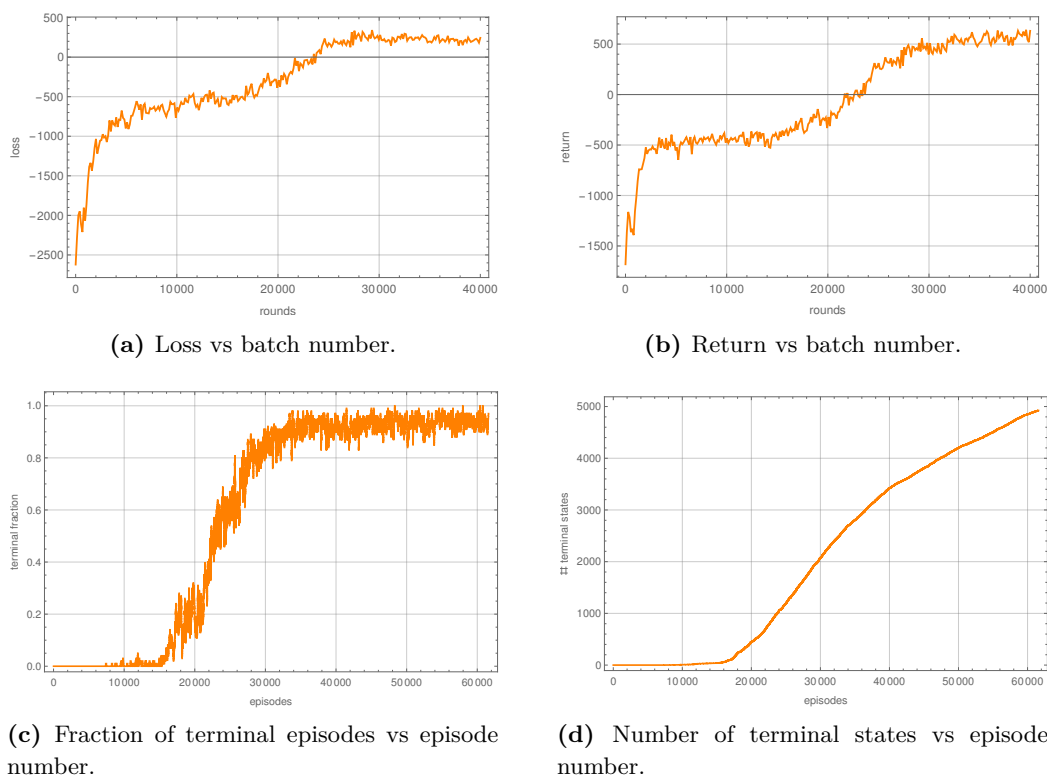


Figure 1. Training measurements for the case of a single U(1) and $-q_{\min} = q_{\max} = 9$.

4 Results

In this section, we present the results we have obtained by applying the REINFORCE algorithm to the FN environment, as described in the previous section. We focus on the two cases of one U(1) symmetry with one singlet scalars and two U(1) symmetries with two singlet scalars, starting with the former.

4.1 One U(1) symmetry

The entries of the 1×10 charge matrix \mathcal{Q} are restricted as in eq. (2.15), with $-q_{\min} = q_{\max} = 9$, so the environment contains $19^{10} \sim 10^{13}$ states. Training of the network in section 3.2 takes about an hour on a single CPU and the measurements made during training are shown in figure 1. After an initial phase of exploration, lasting for about 15000 rounds, the network learns rapidly and the fraction of episodes which end in terminal states (plot (c) in the figure 1) rises to $> 90\%$ within 10000 rounds or so. This pattern is quite characteristic and persists under variation of the various pieces of meta data, including the depth and width of the network, the constants which enter the definition (3.3) of the reward and the definition of a terminal state. The result is also stable under modest variations of the learning rate $\lambda = 1/4000$, although too large learning rates ($\lambda > 1/1000$) suppress exploration and lead to convergence to the “wrong” policy. The residual positive loss in figure 1(a) can

be attributed to the fact that frequently more than one efficient path to a terminal state exists. In other words, there are several very similar optimal policies.

During training, 4924 terminal states are found, which are reduced to 4630 after the redundancies due to the permutations (2.16) are removed. Episodes guided by the trained network, starting at a random state and with maximal length 32, lead to terminal states in 93% of cases, and the average episode length is 16.4.

As figure 1 shows, training has lasted for about 50000 episodes, each with a maximal length of 32 (and actual average episode length decreasing to about 16 during training). This means that the network has explored of the order of 10^6 states during training. We emphasise that this is a tiny fraction, $\sim 10^{-7}$, of the size of the environment. Hence, we are not performing a systematic scan, but rather, the network learns based on a relatively small sample. It is instructive to compare the efficiency of this learning process with random sampling. If we randomly generate 10^6 states from the environment it turns out about 40 of them are terminal states. This should be compared with the 4924 terminal states the network has found based on sampling a similar number of states.

The intrinsic values of the terminal states found during training are optimised by performing a Monte-Carlo search over the order one coefficients a_{ij} , b_{ij} . In this way, we find 89 models \mathcal{Q} with an intrinsic value $\mathcal{V}(\mathcal{Q}) > -1$. From these, the model with the highest intrinsic value is given by²

$$\mathcal{Q} = \left(\begin{array}{ccc|ccc|ccc|c|c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline 6 & 4 & 3 & -2 & 2 & 4 & -3 & -1 & -1 & 1 & 1 \end{array} \right). \quad (4.1)$$

For a scalar VEV $v_1 \simeq 0.224$ and the order one coefficients

$$(a_{ij}) \simeq \begin{pmatrix} -1.975 & 1.284 & -1.219 \\ 1.875 & -1.802 & -0.639 \\ 0.592 & 1.772 & 0.982 \end{pmatrix}, \quad (b_{ij}) \simeq \begin{pmatrix} -1.349 & 1.042 & 1.200 \\ 1.632 & 0.830 & -1.758 \\ -1.259 & -1.085 & 1.949 \end{pmatrix}, \quad (4.2)$$

this model has an intrinsic value $\mathcal{V}(\mathcal{Q}) \simeq -0.598$ and leads to the mass matrices

$$M_u \simeq \begin{pmatrix} 0.000 & 0.126 & -2.380 \\ 0.009 & -3.517 & -24.904 \\ 0.013 & 15.456 & 170.815 \end{pmatrix}, \quad M_d \simeq \begin{pmatrix} -0.001 & 0.023 & 0.026 \\ 0.036 & 0.363 & -0.768 \\ -0.123 & -2.119 & 3.806 \end{pmatrix}. \quad (4.3)$$

After diagonalisation, the resulting quark masses and mixings are

$$\begin{aligned} (m_u, m_c, m_t) &\simeq (0.003, 1.292, 173.358) \text{ GeV} \\ (m_d, m_s, m_b) &\simeq (0.005, 0.066, 4.439) \text{ GeV} \end{aligned}, \quad V_{\text{CKM}} = \begin{pmatrix} 0.969 & 0.247 & 0.003 \\ -0.247 & 0.968 & 0.050 \\ 0.009 & -0.049 & 0.999 \end{pmatrix}, \quad (4.4)$$

in reasonable agreement with the values in table 1 and eq. (2.9). Further examples of models with a high intrinsic value found during training are listed in appendix A.

²Even though the Higgs charge is not part of the environment we include it here for convenience.

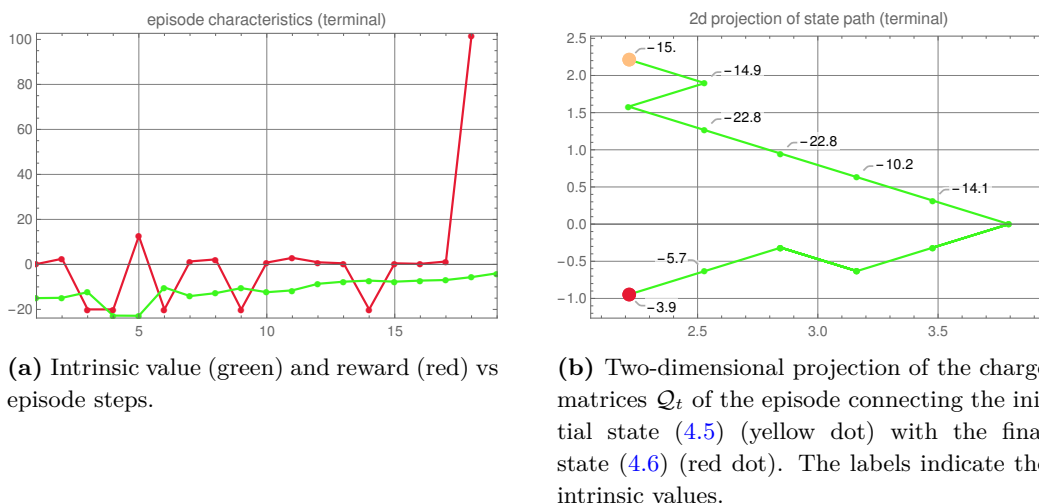


Figure 2. Characteristics of the episode connecting the states (4.5) and (4.6).

Of course, the trained network can be used to find new models. For example, consider starting with the initial state

$$\mathcal{Q} = \left(\begin{array}{ccc|ccc|ccc|c|c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline 0 & 2 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right). \quad (4.5)$$

The optimal intrinsic value for this state, achieved for a singlet VEV $v_1 \simeq 0.112$, is $\mathcal{V}(\mathcal{Q}) \simeq -15$, so this is definitely not a phenomenologically viable model. Using (4.5) as the initial state of an episode, guided by the trained network, it takes 18 steps to reach the terminal state

$$\mathcal{Q} = \left(\begin{array}{ccc|ccc|ccc|c|c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline 2 & 3 & 1 & 1 & 3 & 3 & -2 & -2 & -3 & 2 & 1 \end{array} \right), \quad (4.6)$$

with intrinsic value $\mathcal{V}(\mathcal{Q}) \simeq -3.94$ for a singlet VEV $v_1 \simeq 0.056$. The intrinsic value and the reward along this episode, as well as a two-dimensional projection of the path mapped out by the episode is shown in figure 2.

We can also test the trained network by checking whether it can guide us towards a model known in the literature, starting at a nearby state. For example, consider the model from ref. [22], given by the charge matrix

$$\mathcal{Q} = \left(\begin{array}{ccc|ccc|ccc|c|c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline 3 & 2 & 0 & -3 & -1 & 0 & -3 & -2 & -2 & 0 & 1 \end{array} \right). \quad (4.7)$$

which has an intrinsic value of $\mathcal{V}(\mathcal{Q}) \simeq -4.3$ for a singlet VEV $v_1 \simeq 0.159$. Suppose we use the initial state

$$\mathcal{Q} = \left(\begin{array}{ccc|ccc|ccc|c|c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline 4 & 2 & -3 & -3 & -1 & 0 & -3 & -2 & -2 & -3 & 2 \end{array} \right). \quad (4.8)$$

which is a perturbation of the literature model (4.7) but, as is, does not amount to a potentially viable model. Generating an episode starting at the state (4.8) then leads to the literature model (4.7) in four steps, as indicated in figure 3.

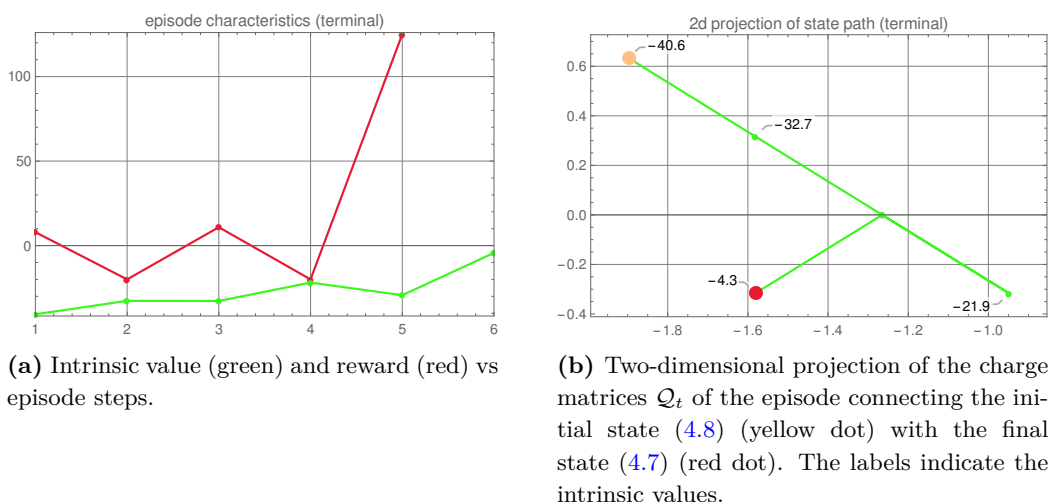


Figure 3. Characteristics of the episode connecting the states (4.8) and (4.7).

4.2 Two U(1) symmetries

Next, we present results for an environment with two U(1) symmetries and two singlet scalar fields. The entries of the 2×10 charge matrix \mathcal{Q} are constrained as in eq. (2.15) but we now consider a somewhat smaller range with $-q_{\min} = q_{\max} = 5$. This still leads to a considerably larger environment than previously, with a total of $11^{20} \sim 10^{21}$ states.

Training for this environment on a single CPU takes about 25 hours and leads to the measurements shown in figure 4. The networks finds 60686 terminal states which reduce to 57807 once the permutation redundancies (2.16) are removed. Episodes guided by the trained network and with maximal length 32 lead to terminal states in 95% of cases and the average episode length is 19.9 steps.

As with the single U(1) case, the network has sampled of the order of 10^6 states during training which is a tiny fraction of about 10^{-14} of the total. Generating 10^6 states randomly produces a few terminal states while the network finds over 60000, based on a similar sample size.

After a Monte-Carlo optimisation of the order one coefficients a_{ij} , b_{ij} we find 2019 from the 57807 models found during training have an intrinsic value $\mathcal{V}(\mathcal{Q}) > -1$. The best of these has charge allocation

$$\mathcal{Q} = \left(\begin{array}{ccc|ccc|ccc|cc} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi_1 & \phi_2 \\ \hline 2 & 2 & 1 & -2 & 0 & 1 & -1 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -2 & 0 & 0 & 1 \end{array} \right), \quad (4.9)$$

and an intrinsic value $\mathcal{V}(\mathcal{Q}) \simeq -0.390$, provided we choose singlet VEVs $v_1 \simeq 0.079$, $v_2 \simeq 0.112$ and order one coefficients

$$(a_{ij}) \simeq \begin{pmatrix} -1.898 & 0.834 & -0.587 \\ -0.575 & -0.592 & 1.324 \\ -1.123 & -1.265 & 0.982 \end{pmatrix}, \quad (b_{ij}) \simeq \begin{pmatrix} -1.759 & 1.358 & 1.013 \\ -1.267 & 1.897 & -1.196 \\ 1.771 & 1.386 & -1.785 \end{pmatrix}. \quad (4.10)$$

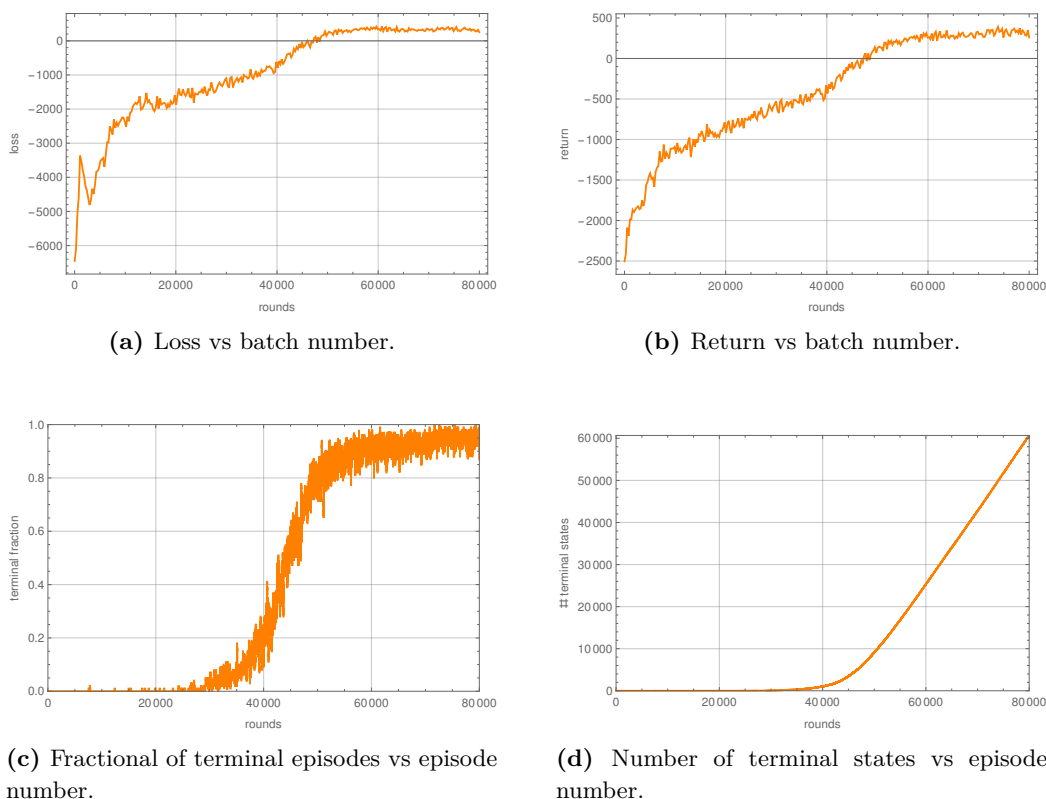


Figure 4. Training measurements for the case of a two U(1) symmetries and $-q_{\min} = q_{\max} = 5$.

This results in the mass matrices

$$M_u \simeq \begin{pmatrix} -0.001 & 0.103 & -0.910 \\ -0.004 & -0.650 & 18.297 \\ -0.098 & -17.489 & 170.815 \end{pmatrix}, \quad M_d \simeq \begin{pmatrix} -0.002 & 0.019 & 0.020 \\ -0.012 & 0.234 & -0.208 \\ 0.218 & 2.149 & -3.910 \end{pmatrix}, \quad (4.11)$$

and the masses and mixing

$$\begin{aligned} (m_u, m_c, m_t) &\simeq (0.002, 1.210, 172.679) \text{ GeV} \\ (m_d, m_s, m_b) &\simeq (0.005, 0.111, 4.476) \text{ GeV} \end{aligned}, \quad V_{\text{CKM}} = \begin{pmatrix} 0.975 & -0.223 & 0.004 \\ 0.223 & 0.974 & 0.040 \\ -0.013 & -0.038 & 0.999 \end{pmatrix}, \quad (4.12)$$

in rough agreement with the values in eq. (2.9). More examples of promising models found by the network are listed in appendix B.

We can also demonstrate that the trained network is capable of finding models which have been constructed in the literature. Consider the model from ref. [22] which is described by the charge matrix

$$Q = \left(\begin{array}{ccc|ccc|ccc|cc} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi_1 & \phi_2 \\ 3 & 0 & 0 & 1 & -1 & 0 & 1 & -4 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -2 & 0 & 0 & -2 & 1 & -1 & 0 & 0 & 1 \end{array} \right). \quad (4.13)$$

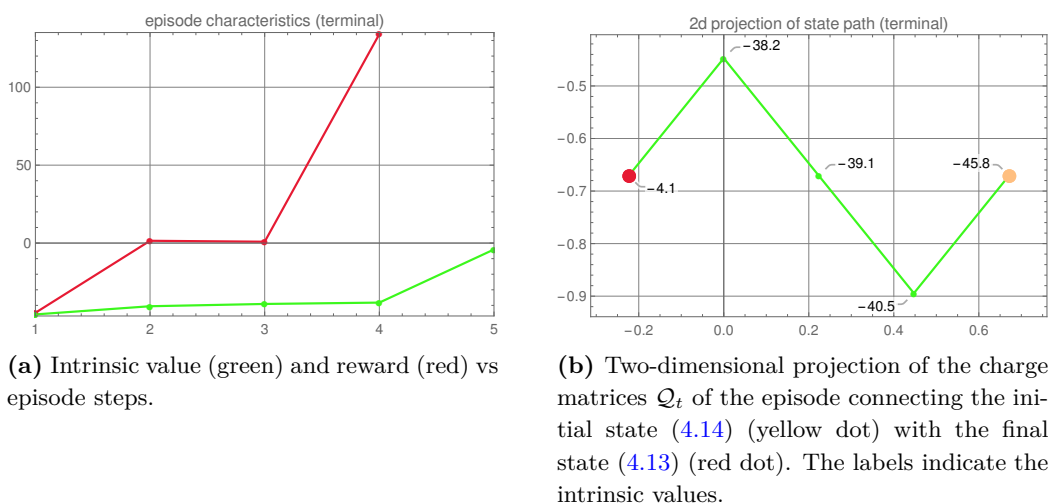


Figure 5. Characteristics of the episode connecting the states (4.14) and (4.13).

For singlet VEVs $v_1 \simeq 0.158$ and $v_2 \simeq 0.028$ it is a terminal state with intrinsic value $\mathcal{V}(\mathcal{Q}) \simeq -4.1$ which, however, has not been found during training. To see that this model can be obtained we start an episode at a nearby state with charge matrix

$$\mathcal{Q} = \left(\begin{array}{ccc|ccc|ccc|cc} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi_1 & \phi_2 \\ 5 & 0 & 0 & 1 & -1 & 0 & 1 & -4 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -2 & 0 & 0 & -2 & 1 & -1 & 0 & 0 & 3 \end{array} \right). \quad (4.14)$$

The trained network then takes us from this state to the literature model (4.13) in three steps, as can be seen in figure 5.

5 Conclusion and outlook

In this paper, we have explored particle physics models with reinforcement learning (RL). We have focused on a simple framework — Froggatt-Nielsen (FN) models for quark masses and mixing — and the simplest policy-based RL algorithm. Our results show that the space of these models can be efficiently explored in this way. For both cases we consider, that is, for FN models with one U(1) symmetries and two U(1) symmetries, the network can be trained to settle on a highly efficient policy which leads to terminal states in $> 90\%$ of all cases and in an average number of < 20 steps. Training is accomplished based on sampling about 10^6 states, which is a tiny fraction, of the order of 10^{-7} and 10^{-14} for the two cases, of the total number of states. Therefore, training does not amount to systematic scanning but rather a guided exploration of the environment. At the same time, the network is significantly more efficient, by factors of the order of 10^2 and 10^4 for the two cases, at finding terminal states than simple random sampling. This shows that reinforcement learning is a powerful method to explore large environments of particle physics models, which defy systematic scanning. The trained networks can be used to find

promising models from random initial states and it is capable of finding literature models, provided it is started at a near-by state.

There are numerous extensions of this work. At a basic level, there are various steps to extend the system within the context of fermion mass models, by enlarging the environment to cover more general classes of theories. (i) The lepton sector can be included, that is, the lepton charges become part of the environment. (ii) The order one coefficients, suitably discretised, are included in the environment. (iii) A class of scalar field potentials is added to the environment. The scalar field VEVs which are determined by an optimal fit to the data in our present system would then be fixed by minimising these potentials. Adding all three components to our environment is feasible and would only require modest computing resources, such as a single machine with a GPU. Our present results strongly suggest that this is likely to produce a successful RL system which finds suitable charge assignments for all fermions as well as scalar potentials which produce the required VEVs. Getting all these elements right simultaneously is not necessarily an easy task for a model builder and we believe such an RL system could provide valuable assistance in finding promising models of fermion masses.

Looking further ahead, we can ask if other classes of particle physics models, such as, for example, supersymmetric or dark matter extensions of the standard model, can be explored in this way. As its most ambitious, this line of thought suggests an RL environment which consists of large classes of quantum field theories, extending the standard model of particle physics. The actions available to the agent would allow for changes of the symmetry, the particle content and the interaction terms in the Lagrangian. The intrinsic value of such models might be determined by comparing their predictions with a wide range of experimental data. Realising such an environment would require significantly more theoretical preparation than was necessary for the FN environment. All required observables have to be readily computable for the entire class of quantum field theories considered. With rapid progress in amplitude computations over the past years this may well be in reach. Of course substantially more computing power will also be required in order to facilitate a fast evaluation of each model against the data. It is conceivable this could be achieved by a small cluster where the computation of a large number of observables can be parallelised. The benefits of such a system might be considerable — it would allow exploring large classes of standard model extensions and their consistency with experimental data and might help to find the correct path for physics beyond the standard model.

Acknowledgments

T. R. H. is supported by an STFC studentship. A. L. would like to thank Andrei Constantin for useful discussions.

A Example models for one U(1) symmetry

In this appendix we list some models with two U(1) symmetries with a high intrinsic value $\mathcal{V}(\mathcal{Q})$, found during training.

charges	$\mathcal{Q} = \left(\begin{array}{ccc ccc ccc c c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline q & 6 & 4 & 3 & -2 & 2 & 4 & -3 & -1 & -1 & 1 & 1 \end{array} \right)$	
$\mathcal{O}(1)$ coeff.	$(a_{ij}) \simeq \begin{pmatrix} -1.975 & 1.284 & -1.219 \\ 1.875 & -1.802 & -0.639 \\ 0.592 & 1.772 & 0.982 \end{pmatrix}$	$(b_{ij}) \simeq \begin{pmatrix} -1.349 & 1.042 & 1.200 \\ 1.632 & 0.830 & -1.758 \\ -1.259 & -1.085 & 1.949 \end{pmatrix}$
VEV, Value	$v_1 \simeq 0.224, \quad \mathcal{V}(\mathcal{Q}) \simeq -0.598$	
charges	$\mathcal{Q} = \left(\begin{array}{ccc ccc ccc c c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline 1 & 2 & 0 & -1 & -3 & 1 & -3 & -5 & -4 & 1 & 1 \end{array} \right)$	
$\mathcal{O}(1)$ coeff.	$(a_{ij}) \simeq \begin{pmatrix} -0.601 & 1.996 & 0.537 \\ -0.976 & -1.498 & -1.156 \\ 1.513 & 1.565 & 0.982 \end{pmatrix}$	$(b_{ij}) \simeq \begin{pmatrix} 0.740 & -1.581 & -1.664 \\ -1.199 & -1.383 & 0.542 \\ 0.968 & 0.679 & -1.153 \end{pmatrix}$
VEV, value	$v_1 \simeq 0.158, \quad \mathcal{V}(\mathcal{Q}) \simeq -0.621$	
charges	$\mathcal{Q} = \left(\begin{array}{ccc ccc ccc c c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline 7 & 8 & 4 & 2 & -1 & 4 & 0 & -1 & 0 & 0 & 1 \end{array} \right)$	
$\mathcal{O}(1)$ coeff.	$(a_{ij}) \simeq \begin{pmatrix} -1.027 & -1.234 & 1.914 \\ 1.027 & -0.525 & -0.921 \\ -1.995 & 1.317 & 0.982 \end{pmatrix}$	$(b_{ij}) \simeq \begin{pmatrix} -1.223 & -1.322 & 1.121 \\ 1.004 & 0.945 & 1.851 \\ -1.930 & -1.071 & -1.720 \end{pmatrix}$
VEV, value	$v_1 \simeq 0.316, \quad \mathcal{V}(\mathcal{Q}) \simeq -0.642$	
charges	$\mathcal{Q} = \left(\begin{array}{ccc ccc ccc c c} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi \\ \hline 3 & 4 & 1 & -3 & -6 & -1 & 1 & -1 & -1 & -2 & 1 \end{array} \right)$	
$\mathcal{O}(1)$ coefficients	$(a_{ij}) \simeq \begin{pmatrix} 1.226 & -0.747 & 1.017 \\ 1.473 & 1.350 & 1.776 \\ -1.575 & 0.988 & 0.982 \end{pmatrix}$	$(b_{ij}) \simeq \begin{pmatrix} -1.012 & 1.947 & -1.941 \\ -1.834 & -1.410 & 1.170 \\ -0.614 & -0.872 & 1.405 \end{pmatrix}$
VEV, value	$v_1 \simeq 0.224, \quad \mathcal{V}(\mathcal{Q}) \simeq -0.721$	

Table 2. Models with high intrinsic value for a single U(1) symmetry.

B Example models for two U(1) symmetries

In this appendix we list some models with a single U(1) symmetry with a high intrinsic value $\mathcal{V}(\mathcal{Q})$, found during training.

charges	$\mathcal{Q} = \left(\begin{array}{ccc ccc c cc} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi_1 & \phi_2 \\ \hline 2 & 2 & 1 & -2 & 0 & 1 & -1 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -2 & 0 & 0 & 1 \end{array} \right)$	
$\mathcal{O}(1)$ coeff.	$(a_{ij}) \simeq \begin{pmatrix} -1.898 & 0.834 & -0.587 \\ -0.575 & -0.592 & 1.324 \\ -1.123 & -1.265 & 0.982 \end{pmatrix}$	$(b_{ij}) \simeq \begin{pmatrix} -1.759 & 1.358 & 1.013 \\ -1.267 & 1.897 & -1.196 \\ 1.771 & 1.386 & -1.785 \end{pmatrix}$
VEVs, value	$(v_1, v_2) \simeq (0.079, 0.112), \quad \mathcal{V}(\mathcal{Q}) \simeq -0.390$	
charges	$\mathcal{Q} = \left(\begin{array}{ccc ccc c cc} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi_1 & \phi_2 \\ \hline 3 & 3 & 0 & 1 & 0 & 1 & -2 & -1 & -1 & 1 & 1 & 0 \\ \hline 1 & 2 & 2 & 1 & -2 & 2 & 0 & -2 & -1 & 0 & 0 & 1 \end{array} \right)$	
$\mathcal{O}(1)$ coeff.	$(a_{ij}) \simeq \begin{pmatrix} 0.715 & -1.366 & -1.988 \\ 1.005 & 1.497 & 0.576 \\ 1.767 & -1.194 & 0.982 \end{pmatrix}$	$(b_{ij}) \simeq \begin{pmatrix} 1.195 & -1.352 & -1.410 \\ -0.979 & -1.860 & -0.521 \\ -1.932 & -0.946 & 0.912 \end{pmatrix}$
VEVs, value	$(v_1, v_2) \simeq (0.224, 0.224), \quad \mathcal{V}(\mathcal{Q}) \simeq -0.425$	
charges	$\mathcal{Q} = \left(\begin{array}{ccc ccc c cc} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi_1 & \phi_2 \\ \hline 2 & 3 & 1 & 1 & -1 & 1 & -1 & -1 & -2 & 0 & 1 & 0 \\ \hline 2 & 2 & 1 & -1 & -2 & 1 & 0 & -1 & 0 & 0 & 0 & 1 \end{array} \right)$	
$\mathcal{O}(1)$ coeff.	$(a_{ij}) \simeq \begin{pmatrix} 1.603 & 1.793 & 1.589 \\ 0.653 & -1.887 & -0.869 \\ -1.609 & -0.679 & 0.982 \end{pmatrix}$	$(b_{ij}) \simeq \begin{pmatrix} 1.369 & -1.107 & 1.151 \\ -1.658 & -1.831 & 1.868 \\ 1.819 & 1.395 & -0.813 \end{pmatrix}$
VEVs, Value	$(v_1, v_2) \simeq (0.316, 0.158), \quad \mathcal{V}(\mathcal{Q}) \simeq -0.480$	
charges	$\mathcal{Q} = \left(\begin{array}{ccc ccc c cc} Q_1 & Q_2 & Q_3 & u_1 & u_2 & u_3 & d_1 & d_2 & d_3 & H & \phi_1 & \phi_2 \\ \hline 1 & 3 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & 1 & 0 \\ \hline 3 & 1 & 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right)$	
$\mathcal{O}(1)$ coeff.	$(a_{ij}) \simeq \begin{pmatrix} -1.231 & 1.029 & 0.753 \\ 0.955 & -1.997 & 1.507 \\ -1.265 & 1.447 & 0.982 \end{pmatrix}$	$(b_{ij}) \simeq \begin{pmatrix} 1.685 & 1.397 & -0.842 \\ 0.877 & 0.777 & -1.451 \\ -1.544 & 1.425 & 1.359 \end{pmatrix}$
VEVs, Value	$(v_1, v_2) \simeq (0.158, 0.079), \quad \mathcal{V}(\mathcal{Q}) \simeq -0.488$	

Table 3. Models with high intrinsic value for two U(1) symmetries.

Open Access. This article is distributed under the terms of the Creative Commons Attribution License ([CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited.

References

- [1] Y. He *Machine-learning the string landscape*, *Phys. Lett. B* **227** (2017) 564.
- [2] F. Ruehle, *Evolving neural networks with genetic algorithms to study the String Landscape*, *JHEP* **08** (2017) 038 [[arXiv:1706.07024](https://arxiv.org/abs/1706.07024)] [[INSPIRE](#)].
- [3] F. Ruehle, *Data science applications to string theory*, *Phys. Rept.* **839** (2020) 1 [[INSPIRE](#)].
- [4] D. Klaewer and L. Schlechter, *Machine Learning Line Bundle Cohomologies of Hypersurfaces in Toric Varieties*, *Phys. Lett. B* **789** (2019) 438 [[arXiv:1809.02547](https://arxiv.org/abs/1809.02547)] [[INSPIRE](#)].
- [5] K. Bull, Y.-H. He, V. Jejjala and C. Mishra, *Machine Learning CICY Threefolds*, *Phys. Lett. B* **785** (2018) 65 [[arXiv:1806.03121](https://arxiv.org/abs/1806.03121)] [[INSPIRE](#)].
- [6] Y.-N. Wang and Z. Zhang, *Learning non-Higgsable gauge groups in 4D F-theory*, *JHEP* **08** (2018) 009 [[arXiv:1804.07296](https://arxiv.org/abs/1804.07296)] [[INSPIRE](#)].
- [7] C.R. Brodie, A. Constantin, R. Deen and A. Lukas, *Machine Learning Line Bundle Cohomology*, *Fortsch. Phys.* **68** (2020) 1900087 [[arXiv:1906.08730](https://arxiv.org/abs/1906.08730)] [[INSPIRE](#)].
- [8] F. Day and S. Krippendorf, *Accelerating the search for Axion-Like Particles with machine learning*, *JCAP* **03** (2020) 046 [[arXiv:1907.07642](https://arxiv.org/abs/1907.07642)] [[INSPIRE](#)].
- [9] S. Krippendorf and M. Syvaeri, *Detecting Symmetries with Neural Networks*, [arXiv:2003.13679](https://arxiv.org/abs/2003.13679) [[INSPIRE](#)].
- [10] S. Gukov, J. Halverson, F. Ruehle and P. Sułkowski, *Learning to Unknot*, *Mach. Learn. Sci. Tech.* **2** (2021) 025035 [[arXiv:2010.16263](https://arxiv.org/abs/2010.16263)] [[INSPIRE](#)].
- [11] L.B. Anderson, M. Gerdes, J. Gray, S. Krippendorf, N. Raghuram and F. Ruehle, *Moduli-dependent Calabi-Yau and SU(3)-structure metrics from Machine Learning*, *JHEP* **05** (2021) 013 [[arXiv:2012.04656](https://arxiv.org/abs/2012.04656)] [[INSPIRE](#)].
- [12] J. Halverson, B. Nelson and F. Ruehle, *Branes with Brains: Exploring String Vacua with Deep Reinforcement Learning*, *JHEP* **06** (2019) 003 [[arXiv:1903.11616](https://arxiv.org/abs/1903.11616)] [[INSPIRE](#)].
- [13] M. Larfors and R. Schneider, *Explore and Exploit with Heterotic Line Bundle Models*, *Fortsch. Phys.* **68** (2020) 2000034 [[arXiv:2003.04817](https://arxiv.org/abs/2003.04817)] [[INSPIRE](#)].
- [14] J. Halverson, A. Maiti and K. Stoner, *Neural Networks and Quantum Field Theory*, *Mach. Learn. Sci. Tech.* **2** (2021) 035002 [[arXiv:2008.08601](https://arxiv.org/abs/2008.08601)] [[INSPIRE](#)].
- [15] C.D. Froggatt and H.B. Nielsen, *Hierarchy of Quark Masses, Cabibbo Angles and CP-violation*, *Nucl. Phys. B* **147** (1979) 277 [[INSPIRE](#)].
- [16] A. Davidson, V.P. Nair and K.C. Wali, *Peccei-Quinn Symmetry as Flavor Symmetry and Grand Unification*, *Phys. Rev. D* **29** (1984) 1504 [[INSPIRE](#)].
- [17] A. Davidson and K.C. Wali, *Family Mass Hierarchy From Universal Seesaw Mechanism*, *Phys. Rev. Lett.* **60** (1988) 1813 [[INSPIRE](#)].
- [18] A. Davidson, S. Ranfone and K.C. Wali, *Quark Masses and Mixing Angles From Universal Seesaw Mechanism*, *Phys. Rev. D* **41** (1990) 208 [[INSPIRE](#)].

- [19] A. Davidson, M. Koca and K.C. Wali, *U(1) as the Minimal Horizontal Gauge Symmetry*, *Phys. Rev. Lett.* **43** (1979) 92 [INSPIRE].
- [20] A. Davidson and K.C. Wali, *Minimal flavor unification via multigenerational Peccei-Quinn symmetry*, *Phys. Rev. Lett.* **48** (1982) 11 [INSPIRE].
- [21] M. Leurer, Y. Nir and N. Seiberg, *Mass matrix models*, *Nucl. Phys. B* **398** (1993) 319 [[hep-ph/9212278](#)] [INSPIRE].
- [22] M. Leurer, Y. Nir and N. Seiberg, *Mass matrix models: The Sequel*, *Nucl. Phys. B* **420** (1994) 468 [[hep-ph/9310320](#)] [INSPIRE].
- [23] E. Dudas, S. Pokorski and C.A. Savoy, *Yukawa matrices from a spontaneously broken Abelian symmetry*, *Phys. Lett. B* **356** (1995) 45 [[hep-ph/9504292](#)] [INSPIRE].
- [24] E. Dudas, C. Grojean, S. Pokorski and C.A. Savoy, *Abelian flavor symmetries in supersymmetric models*, *Nucl. Phys. B* **481** (1996) 85 [[hep-ph/9606383](#)] [INSPIRE].
- [25] L.E. Ibáñez and G.G. Ross, *Fermion masses and mixing angles from gauge symmetries*, *Phys. Lett. B* **332** (1994) 100 [[hep-ph/9403338](#)] [INSPIRE].
- [26] S.R. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge U.S.A. (2018).
- [27] D. Silver et al., *Mastering the game of Go without human knowledge*, *Nature* **550** (2017) 354.
- [28] PARTICLE DATA GROUP collaboration, *Review of Particle Physics*, *PTEP* **2020** (2020) 083C01 [INSPIRE].
- [29] Wolfram Research, Inc., *Mathematica*, Version 12.2, Champaign, IL (2020) <https://www.wolfram.com/mathematica/>.