Exploring the Heterotic Landscape with Genetic Algorithms and Reinforcement Learning

Thomas Harvey

UNIVERSITY OF OXFORD

Based on work in collaboration with: Steve Abel, Andrei Constantin and Andre Lukas

2103.04759, 2108.07316, 2110.14029, 2111.07333

Outline

- Introduction - RL, GA and Monads
- Reinforcement Learning (RL) applied to Monads
- Genetic Algorithms (GA) applied to Monads
- Conclusion

Introduction

Introduction: ML (Specifically RL)

- There has been a focus on supervised ML in String Theory
    - For review see: F. Ruehle, Phys. Rep. 839, 1-117, 2020
- RL: “Mastering the game of Go with deep neural networks and tree search” - Silver et al.
- RL is more appropriate to search the string landscape
    - Halverson, Nelson, Ruehle, 1911.07835
    - Larfors, Schneider, 2003.04817
    - Constantin, Harvey, Lukas 2108.07316
    - Krippendorf, Kroepsch, Syvaeri, 2107.04039

Introduction: GA

- GA dates back to the 1960s
  - J. Holland, “Adaption in Natural and Artificial Systems”, 1975
- Has been used in our field to search large environments
  - Abel, Rizos, 1404.7359
  - Abel, Cerdeno, Robles 1805.03615,
  - Cole, Schachner, Shiu, 1907.10072,
  - Abel, Constantin, Harvey, Lukas 2110.14029
  - Cole, Krippendorf, Schachner, Shiu, 2111.11466

Introduction: GA and RL for String Model Building

- Our Aim: *“Can RL/GA construct string realisations of the standard model”*
- If so:
  - How do they compare?
  - Do they find any, until now, unknown models?
  - Do they give us model building insight?
- We focus heterotic models via Monad bundles
  - Very few promising models are known here!

Introduction: Monad Bundles

- Heterotic Models are specified by
  - Smooth Calabi-Yau manifold X
  - Vector Bundle V on X
    - Will be SU(4) bundle for us →SO(10) GUT Theory
- Monad bundles are constructed via a short exact sequence.

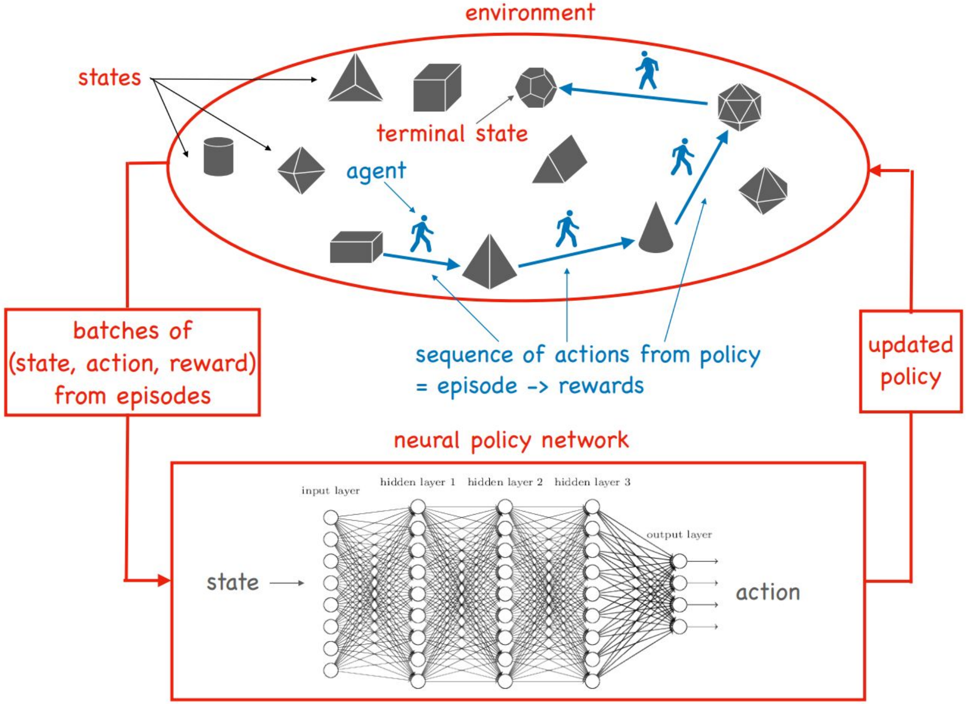
$$0 \to V \to B \xrightarrow{f} C \to 0 \qquad V \cong \mathrm{Ker}(f) \qquad B = \bigoplus_{a=1} \mathcal{O}_X(\mathbf{b}_a) \qquad C = \bigoplus_{\alpha=1} \mathcal{O}_X(\mathbf{c}_\alpha)$$

- Such a bundle is called a “Monad”
  - Key Point: Specified by a large number of integers

Reinforcement Learning (RL)

RL: The Basics

- Agent explores environment collecting rewards
- We will specify rewards based on agreement with experiment
- Model building with RL:
  - String Theory - Here
  - Particle Physics - Harvey and Lukas 2103.04759
- Realised REINFORCE and Actor-Critic in Mathematica



RL: Monads

RL	**Physics**
Environment	All Monads (B,C) on a fixed Calabi-Yau with entries: $b_{min} \leq b_a^i \leq b_{max}$ and $c_{min} \leq c_a^i \leq c_{max}$
Actions	For fixed (i, j, a): $b_a^i \rightarrow b_a^i \pm 1$ and $c_a^j \rightarrow c_b^j \pm 1$ simultaneously
Reward	Increase in State Value State Values ~ -(Deviation from MSSM*) + (Big Bonus if Terminal)
Terminal State	The Standard Model spectrum*

*All checks requiring cohomology calculations are done after training

RL: Mo

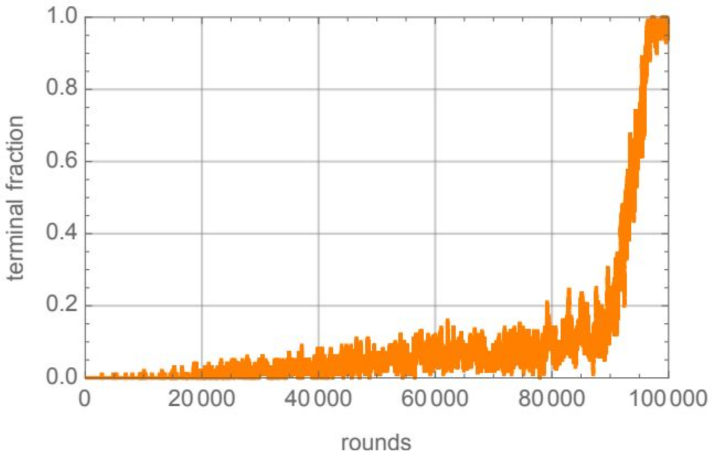
RL	
Environment	
Actions	
Reward	
Terminal Stat	

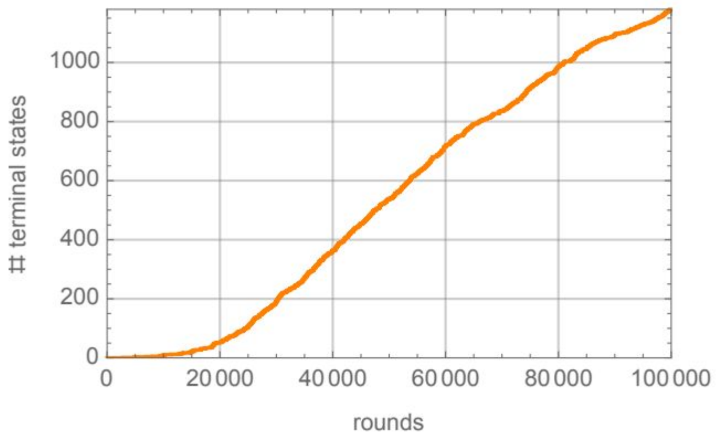
property	term in $v(B,C)$	comment
index match	$-\frac{2\lvert\mathrm{ind}(V)-\tau\rvert}{hM^3}$	$\tau=-3\lvert\Gamma\rvert$ is the target index, $\mathrm{ind}(V)$ computed from Eq. (2.20)
anomaly	$\frac{1}{hM^2}\sum_{i=1}^{h}\min\left(c_{2i}(TX)-c_{2i}(V),0\right)$	no penalty if anomaly condition satisfied, $c_{2i}(V)$ computed from Eq. (2.20)
bundleness	$-(d_{\mathrm{deg}}+1)$	d_{deg} = dimension of degeneracy locus as discussed in Sec. 2.4; if the degeneracy locus is empty, d_{deg} is to be taken as -1
split bundle	$-n_{\mathrm{split}}$	n_{split} = number of splits in V
equivariance	$-\sum_{U\subset B,C}\mathrm{mod}(\mathrm{ind}(U),\lvert\Gamma\rvert)$	U runs over all line bundles in B,C or blocks of same line bundles, as discussed in Sec. 2.4
trivial bundle	$-n_{\mathrm{trivial}}$	n_{trivial} = number of trivial line bundles
stability V	$-\frac{\max(0,h^0(X,B)-h^0(X,C))}{hM^3}$	tests Hoppe's criterion for V, cohomologies from formulae in Sec. 2.3
stability V^*	$-\frac{\max(0,h^0(X,B^*)-h^0(X,C^*))}{hM^3}$	tests Hoppe's criterion for V^*, cohomologies from formulae in Sec. 2.3

Table 2: Contributions to the intrinsic value for the monad environment. The intrinsic value $v(B,C)$ is the sum of all eight terms and $M=\max(b_{\max},c_{\max})$.

RL: Bicubic

- SO(10) GUT From Monad on Bicubic, with Z3 X Z3 Wilson line
- $b_{min} = -3$, $b_{max} = 5$, $c_{min} = 0$, $c_{max} = 5$, $r_B = 6$, $r_c = 2$
  - 10^12 states in total!
- Training ~1 hour on single CPU - Find ~15 models after extra checks

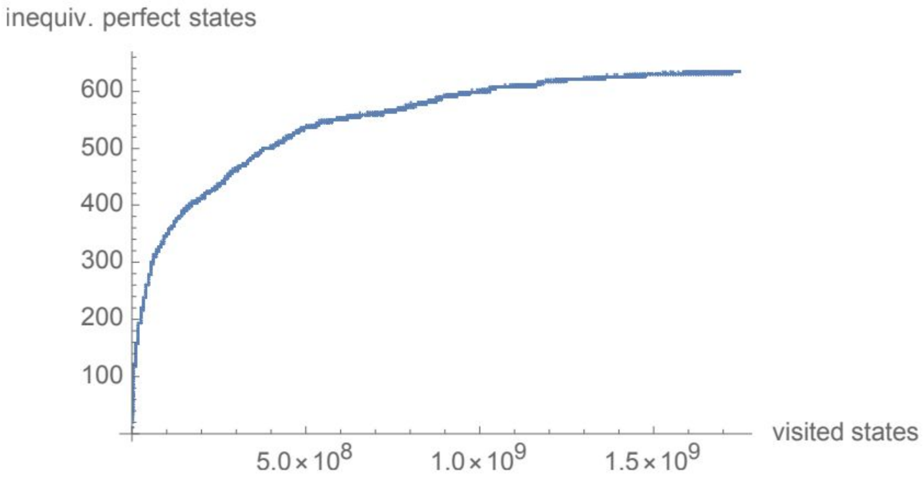




RL: Bicubic

- Keep searching to saturation (35 core days)
- Suggests almost all models found
- Contains 27 genuine new (6,2) models, and the one known model!
    - Anderson, Gray, He, Lukas - 0911.1569
- Also have O(500) models on triple tri-linear

$$X \sim \left(\begin{array}{c|ccc} \mathbb{P}^2 & 1 & 1 & 1 \\ \mathbb{P}^2 & 1 & 1 & 1 \\ \mathbb{P}^2 & 1 & 1 & 1 \end{array}\right)$$



RL: Bicubic

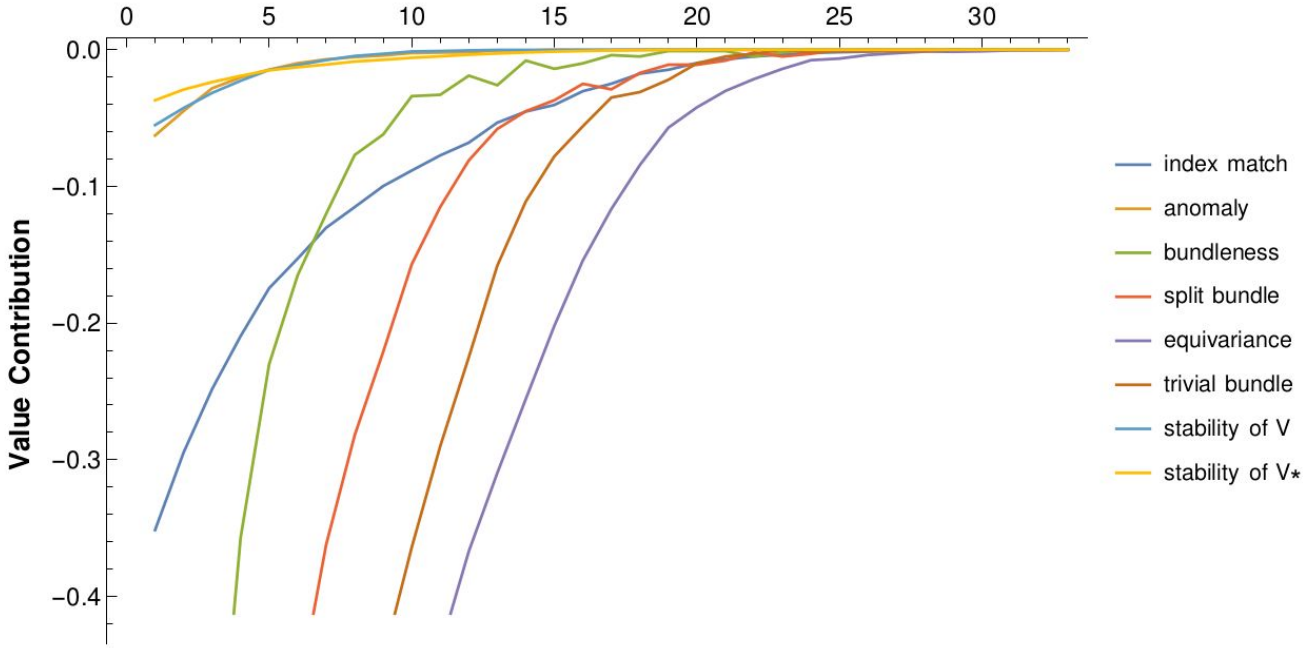


Figure 7: The different contribtutions to the intrinsic value for $(r_b, r_c) = (6, 2)$ bicubic models. This data is averaged over 1000 termianl states using the trained network.

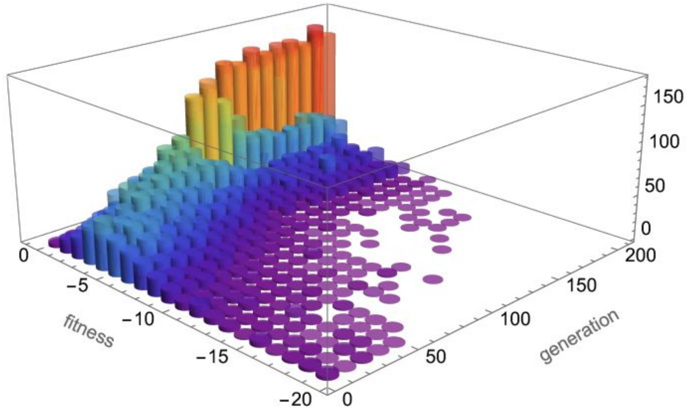
Genetic Algorithms (GA)

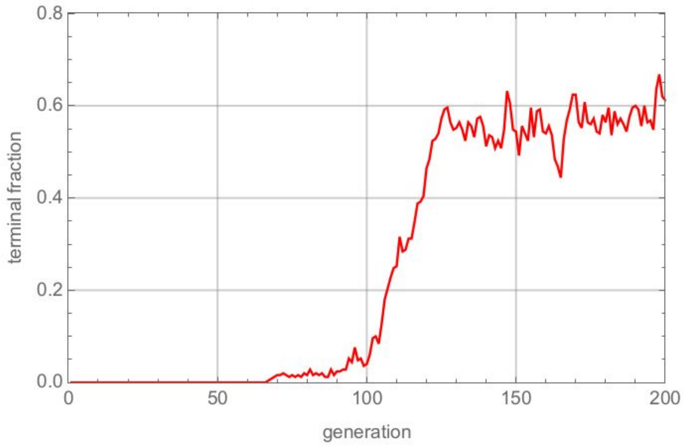
GA: The Basics

- Take integers specifying Monad and convert to binary
- Assign “fitness” (= state value in previous langage) to each
- Create population (250 in our case)
- Evolve the population via crossing and mutation many times over
- After a number of generations the population has many terminal states
- Code for GA available:
    - https://github.com/harveyThomas4692/GAMathematica

GA: Bicubic

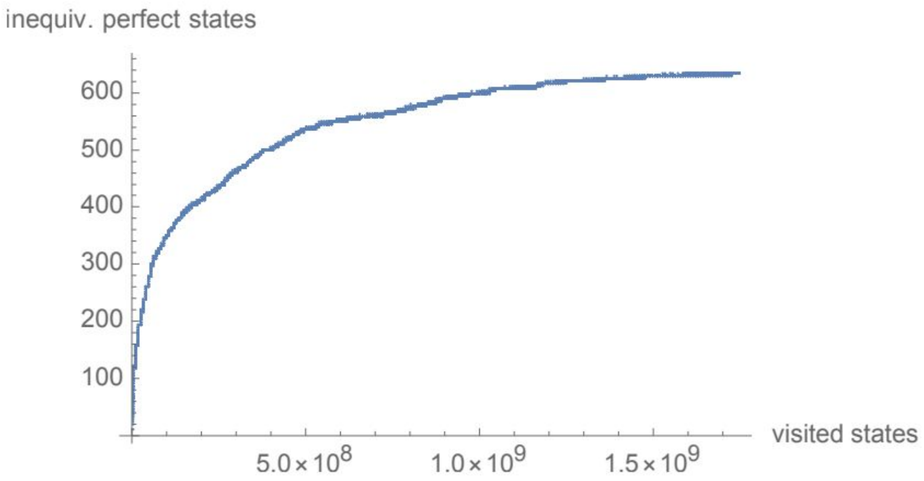
This produces terminal states in minutes



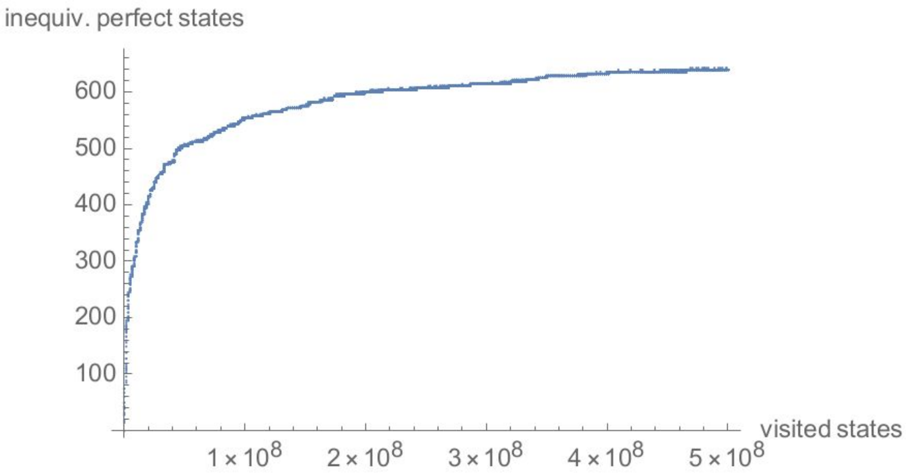


GA: Bicubic RL vs GA

RL (35 Core days)



GA (10 Core days)



- Largely the same terminal states as found with RL
- GA appears faster than RL for this problem and our implementation
- GA tends to find more permutations (expected from implementation)

Conclusion

- RL and GA are both efficient in engineering string models
- Many new models are discovered
  - Both methods give similar models, including many new models
- For our implementation GA was faster at finding terminal states
- Can this be extended to other manifolds? larger h^{11}? Extra Constraints?
- Can this be used in other string constructions?