Physics Letters B 795 (2019) 700–706

Contents lists available at ScienceDirect

Physics Letters B

www.elsevier.com/locate/physletb

# Getting CICY high

Kieran Bull [a,b], Yang-Hui He [c,b,d,e], Vishnu Jejjala [f,g], Challenger Mishra [h,i,j,*]

[a] School of Physics and Astronomy, University of Leeds, LS2 9JT, UK
[b] Rudolf Peierls Centre for Theoretical Physics, Clarendon Laboratory, Parks Rd, University of Oxford, OX1 3PU, UK
[c] Department of Mathematics, City University, London, UK
[d] Merton College, University of Oxford, UK
[e] School of Physics, NanKai University, Tianjin, PR China
[f] Mandelstam Institute for Theoretical Physics, NITheP, CoE-MaSS, and School of Physics, University of the Witwatersrand, South Africa
[g] David Rittenhouse Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA
[h] The Alan Turing Institute, London, UK
[i] Department of Computer Science, University of Oxford, UK
[j] Instituto de Ciencias Matemáticas, (ICMAT), Madrid, Spain

ARTICLE INFO

*Article history:*
Received 13 March 2019
Received in revised form 7 June 2019
Accepted 17 June 2019
Available online 8 July 2019
Editor: N. Lambert

*Keywords:*
Machine learning
Neural network
Support Vector Machine
Calabi–Yau
String compactifications

ABSTRACT

Supervised machine learning can be used to predict properties of string geometries with previously unknown features. Using the complete intersection Calabi–Yau (CICY) threefold dataset as a theoretical laboratory for this investigation, we use low $h^{1,1}$ geometries for training and validate on geometries with large $h^{1,1}$. Neural networks and Support Vector Machines successfully predict trends in the number of Kähler parameters of CICY threefolds. The numerical accuracy of machine learning improves upon seeding the training set with a small number of samples at higher $h^{1,1}$.



## 1. Introduction

Ever since Kaluza and Klein extended the original insight of Einstein, we regard the fundamental forces as having an intrinsically geometric origin. The modern realization of this paradigm is the compactification of superstring theory down to four dimensions in order to recover the particle physics probed in experiments and inferred from astrophysical observations. In the most straightforward approach consistent with low energy supersymmetry, the six extra dimensions predicted by string theory comprise a compact *Calabi–Yau threefold*. Geometric and topological properties of the Calabi–Yau threefold determine features of the four dimensional effective action. For example, the Euler character of the geometry fixes the number of generations of light particles. Starting from the work of [1] and [2], numerous constructions of this type replicate the matter spectrum and gauge symmetries that we observe in Nature [3–10]. Naïve extrapolation of even the simplest class of models suggests that there are $10^{23}$ (nearly a mole's worth) of superstring derived Standard Models [11].

The vacuum selection problem, to find a principle that explicates which solution of the fundamental theory constitutes our world and how and why this came to be, remains an outstanding puzzle. It is also unknown what the typical string compactification looks like and how closely this solution resembles the one we actually inhabit. There are 7890 complete intersection Calabi–Yau (CICY) threefolds realized as the zero locus of polynomials in complex projective space. There are an unknown number of toric Calabi–Yau threefolds obtained from triangulation [12,13] of the 473 800 776 reflexive polytopes in $\mathbb{R}^4$ tabulated by Kreuzer and Skarke [14]. Other Calabi–Yau spaces are neither CICY nor toric. The largest available database [15,16] describes only the toric Calabi–Yau geometries with Hodge number $h^{1,1} \leq 6$. While [17] explores the shape of the full Kreuzer–Skarke dataset, it suffices to notice that the distribution peaks sharply, and 910 113 of the polytopes sit at $(h^{1,1}, h^{2,1}) = (27, 27)$. The explicit Standard Model constructions to date meanwhile correspond to geometries whose

* Corresponding author.
*E-mail addresses:* pykb@leeds.ac.uk (K. Bull), hey@maths.ox.ac.uk (Y.-H. He), vishnu@neo.phys.wits.ac.za (V. Jejjala), challenger.mishra@gmail.com (C. Mishra).

https://doi.org/10.1016/j.physletb.2019.06.067

Hodge numbers are $O(1)$ rather than $O(10)$. These are atypical as manifolds with small Hodge numbers are sparse.

Recently, a promising new approach to studying the vacuum selection problem has emerged. The development of Big Data techniques in computer science and the broad applicability of these methods to such disparate fields as art, finance, chess and go, linguistics, medicine, music, experimental particle physics, and zoology invites us to also use these tools to investigate aspects of string phenomenology and string mathematics. In particular, the paradigm of machine learning the landscape by using neural networks to study algebraic geometry, potentially bypassing expensive computations such as Gröbner bases, was proposed in [18,19] (*cf.* a pedagogical introduction in [20]). Already, there has been a significant amount of work in this direction, ranging from the studies of CICY geometries to the computation of line bundle cohomologies of toric hypersurfaces [18–33]. These studies have relied upon a multitude of algebro-geometric databases collected over the past few decades. A large fraction of such machine learning aided studies of the string landscape has been through the lens of neural networks with a variety of architectures [18,19,21–28]. A host of other techniques such as linear and logistic regression, Support Vector Machines (SVMs), and random forests, to name a few, have also been used, sometimes in conjunction with neural networks [21,25, 28–31,33].

In our previous work [21], using the CICY threefolds as a testbed, we answered the following questions. Given the configuration matrix which defines a CICY threefold, can machine learning techniques compute the Hodge numbers of the geometry? Can the machine deduce whether the geometry is favorable, *viz.*, does the number of projective space factors in the ambient space equal $h^{1,1}$? This property is important because such geometries accommodate the construction of stable vector bundles for string model building. Can the machine determine which geometries enjoy discrete symmetries, which are crucial for introducing Wilson lines that break the GUT symmetry to the Standard Model group? We find that even with 50% of the data for training, neural network classifiers identify the Hodge numbers at better than 80% accuracy. We select favorability with SVMs with more than 90% accuracy. Because CICYs with discrete symmetries are relatively rare ($\sim 2.5\%$ of all cases) [34], correctly isolating only these geometries is a comparatively less successful effort.

Heuristically, all of these investigations unfold as follows. We segregate the dataset into two disjoint parts: a *training set* $T$ and its complement $T^c$, used for *validation*. The machine is taught the associations

$$\{a_1, a_2, \ldots, a_n\} \quad \longrightarrow \quad \{b_1, b_2, \ldots, b_n\} \tag{1}$$

for elements $a_i \in T$. Based on what it has learned about the training set, the machine then tries to determine the $b_j$ corresponding to the unseen elements $a_j \in T^c$. The selection of the elements in $T$ is performed at random at the outset. Since the CICY threefolds have been studied for decades starting from the work of [35], we know what the answers are and can check how frequently the algorithms arrive at the correct result. Choosing different training sets and repeating the experiment allows us to assign error bars to the results obtained from validation. By increasing the size of the training set incrementally, we examine the machine's learning curve.

While this provides an unexpectedly good proof of concept, the methodology is not realistic for addressing the fundamental challenge in studying Calabi–Yau compactifications: *the difficulty of a calculation increases with the Hodge numbers and the dimension*. This, after all, is why explicit Standard Model constructions are on manifolds with Hodge numbers of $O(1)$ and why triangulating polytopes to populate the toric Calabi–Yau database [16] stopped at $h^{1,1} = 6$. One estimate of the total number of triangulations of the Kreuzer–Skarke dataset is $10^{10505}$ [33]. While there are $10^8$ reflexive polytopes associated to toric Calabi–Yau threefolds, the best guess in the literature is that there are $10^{18}$ reflexive polytopes whose triangulations yield toric Calabi–Yau fourfolds relevant for F-theory model building [36]. There may be $10^{3000}$ distinctly resolvable base geometries [37]. The scale of these numbers renders any systematic survey of the string landscape unfeasible. We would therefore like to develop techniques such that the training and validation sets are different in character. We aim to train with the easy cases and use the machine to predict solutions to harder problems for which the calculations are more intricate or where the answers could be unknown. We want as well to measure how reliable the results are when we segregate the data in this imbalanced way. By organizing the CICY dataset into a low $h^{1,1}$ training set and a high $h^{1,1}$ validation set, we report on progress in this effort.

The structure of this letter is as follows. In Section 2, we review the CICY threefolds. In Section 3, we describe the machine learning architectures we employ. In Section 4, we present the results of our investigation, which focuses on determining $h^{1,1}$ starting from the configuration matrix as the input. In Section 5, we provide a brief discussion and a prospectus for future work.

## 2. Complete intersection Calabi–Yau threefolds

For completeness, we briefly recall the relevant geometry. We refer the reader to [38] for a pedagogical review and references therein to the original literature.

A Calabi–Yau manifold admits a Ricci flat Kähler metric. We enforce this requirement by ensuring that the first Chern class vanishes. The simplest example of a compact Calabi–Yau threefold is the Fermat quintic in $\mathbb{P}^4$:

$$\sum_{\alpha=1}^{5} z_\alpha^5 = 0\,, \tag{2}$$

where $(z_1, \ldots, z_5) \sim \lambda(z_1, \ldots, z_5)$ are coordinates on projective space and $\lambda \in \mathbb{C}^\star$. As (2) is a homogeneous equation, we designate this geometry $\mathbb{P}^4(5)_{-200}$. The subscript denotes the Euler character $\chi = 2(h^{1,1} - h^{2,1})$. This is the prototype example of a class of geometries. Consider the *configuration matrix*

$$X = \begin{matrix} \mathbb{P}^{n_1} \\ \vdots \\ \mathbb{P}^{n_\ell} \end{matrix} \begin{pmatrix} q_1^1 & \cdots & q_K^1 \\ \vdots & \ddots & \vdots \\ q_1^\ell & \cdots & q_K^\ell \end{pmatrix}_\chi . \tag{3}$$

The zero locus of a set of homogeneous polynomials defined by the given matrix over the combined set of coordinates in the product of the projective spaces $\mathbb{P}^{n_i}$ is a *complete intersection Calabi–Yau* (CICY) threefold when

$$\sum_{i=1}^{\ell} n_i - K = 3\,, \tag{4}$$

$$\sum_{a=1}^{K} q_a^i = n_i + 1\,, \quad \forall\, i \in \{1, \ldots, \ell\}\,. \tag{5}$$

The former condition imposes the requirement that the manifold is a complete intersection threefold while the latter guarantees that $c_1 = 0$. The simplest geometries obtained in this manner are

$$\mathbb{P}^5(3,3)_{-144}\,, \quad \mathbb{P}^5(4,2)_{-176}\,,$$
$$\mathbb{P}^6(3,2,2)_{-144}\,, \quad \mathbb{P}^7(2,2,2,2)_{-128}\,. \tag{6}$$

The Tian–Yau manifold is another example of a CICY threefold:

$$\begin{matrix}\mathbb{P}^3\\ \mathbb{P}^3\end{matrix}\begin{pmatrix}3 & 0 & 1\\ 0 & 3 & 1\end{pmatrix}_{-18} \quad\Longleftrightarrow\quad \begin{aligned} a^{\alpha\beta\gamma} w_\alpha w_\beta w_\gamma &= 0\,,\\ b^{\alpha\beta\gamma} z_\alpha z_\beta z_\gamma &= 0\,,\\ c^{\alpha\beta} w_\alpha z_\beta &= 0\,, \end{aligned} \tag{7}$$

where $w$ and $z$ are homogeneous coordinates on each of the two $\mathbb{P}^3$s and $a$, $b$, $c$ are generic coefficients.

For CICY threefolds, the size of the configuration matrix $X$ is constrained. We find that

$$K \leq N_1 + N_a + 3\,, \qquad N_1 \leq 9\,, \qquad N_a \leq 6\,. \tag{8}$$

Here, $N_1$ counts the number of $\mathbb{P}^1$ factors and $N_a$ counts the number of other projective space factors. There are 7890 configuration matrices ranging in size from $1\times 1$ (the quintic) to $12\times 15$ with elements $q_a^i \in [0,5]$. In this dataset available at [39], we find 70 distinct Euler characters $\chi \in [-200, 0]$ and 266 distinct Hodge pairs $(h^{1,1}, h^{2,1})$. The topological invariant $h^{1,1}$ counts the number of two cycles and four cycles and accounts for the Kähler deformations, whereas $h^{2,1}$ counts the number of three cycles and accounts for the complex structure deformations. These are, respectively, the size and shape parameters of the geometry. Within the set of CICY threefolds,

$$0 \leq h^{1,1} \leq 19\,, \qquad 0 \leq h^{2,1} \leq 101\,. \tag{9}$$

Mirror symmetry — invariance under the interchange $h^{1,1} \leftrightarrow h^{2,1}$ — is not a property of the dataset. As $\chi$ is always negative, $h^{1,1} \leq h^{2,1}$ for any given CICY threefold. The Euler character is a cubic expression in the elements of the configuration matrix. Calculating $h^{1,1}$ and $h^{2,1}$ is conceptually straightforward but requires some care [40–45]. One of the goals of applying machine learning to this dataset is to circumvent the necessity of studious sequence chasing. Of the CICY threefolds, 195 possess freely acting symmetries; 37 different finite groups appear, ranging from $\mathbb{Z}_2$ to $\mathbb{Z}_8 \rtimes H_8$ [34]. A number of CICY threefolds also admit non-freely acting symmetries [46,47].

We tabulate the number of geometries with each value of $h^{1,1}$ in Table 1. Among the CICY threefolds, 4874 out of the 7890 are favorable, *i.e.*, for these manifolds, the second cohomology class descends from that of the ambient space $\mathbb{P}^{n_1}\times\ldots\times\mathbb{P}^{n_\ell}$ [10]. Favorability, however, is not a topological invariant. It turns out that all but 48 CICY threefolds can be given a favorable representation by splitting the configuration matrices, while the remaining can be described favorably in products of almost del Pezzo surfaces [48], resulting in the dataset of maximally favorable CICY threefolds available at [49]. We will however work with the original CICY threefold dataset [39] in line with our work in the previous publication [21]. We finally note that $h^{2,1}$ ranges over a larger interval than $h^{1,1}$. Fig. 1 plots the number of geometries at a given $h^{2,1}$. Knowing $h^{1,1}$, once we compute $\chi$, the Hodge number $h^{2,1}$ is

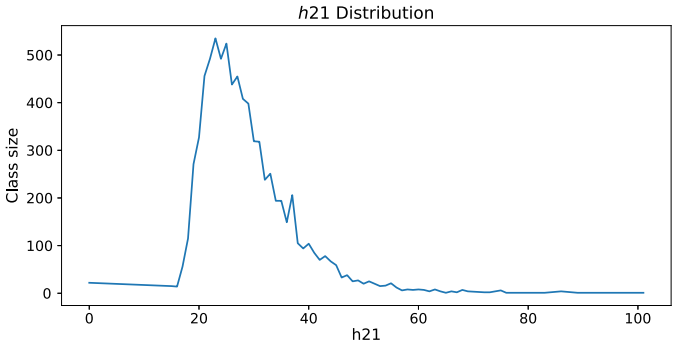


**Fig. 1.** Multiplicity of $h^{2,1}$ in CICY threefold dataset.

of course redundant information. The goal of machine learning is to determine topological invariants and properties like favorability using the configuration matrix as an input.

The CICY fourfolds are catalogued in [50]. (There are 921 497 configuration matrices most of which correspond to elliptically fibered Calabi–Yau spaces.) Fourfolds have four non-trivial Hodge numbers of which three are independent:

$$4h^{1,1} - 2h^{2,1} + 4h^{3,1} - h^{2,2} + 44 = 0\,. \tag{10}$$

Identifying all of the discrete symmetries in this dataset has not been accomplished. Thus, there is a potential benefit to applying machine learning to this effort as well. This is work in progress.

## 3. Neural networks and Support Vector Machines

Our analysis in this paper involves a neural network regressor as well as a classifier, and an SVM regressor. We begin by specifying the input representations of the neural networks. For the regressor, we use the flattened form of the CICY configuration matrix, whereas for the classifier, we use the configuration matrix itself. To ensure uniformity of input dimensions, we pad with zeroes. While we have not conducted a systematic study, preliminary investigations suggest that different representations of the input related by permutation of the configuration matrices, or padding do not significantly alter the performance of the neural networks. We also find that using adjacency matrices of the bipartite graph representations of the CICY matrices as input, does not yield significantly different results. The architectures for the regressors are similar to that in [21]. We use the `Keras` Python package with `TensorFlow` backend to implement the neural network. The neural network consists of a 1000 neuron layer, ReLU (rectified linear unit) activation layer, 1 neuron summation and sigmoid activation. We use the quadratic programming Python package `Cvxopt` to solve the SVM optimization problem. The hyperparameters are selected by hand. We employ a Gaussian kernel

**Table 1**
The frequency row gives the multiplicity of $h^{1,1}$ in the CICY threefold dataset. $N(h^{1,1})$ counts the number of CICY threefolds with Hodge number less than or equal to $h^{1,1}$. The favorable row counts the number of favorable CICYs with a given $h^{1,1}$.

| $h^{1,1}$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| frequency | 22 | 5 | 36 | 155 | 425 | 856 | 1257 | 1463 | 1328 | 1036 |
| $N(h^{1,1})$ | 22 | 27 | 63 | 218 | 643 | 1499 | 2756 | 4219 | 5547 | 6583 |
| favorable | 0 | 5 | 36 | 155 | 425 | 835 | 1139 | 0 | 0 | 0 |

| $h^{1,1}$ | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|
| frequency | 648 | 372 | 161 | 72 | 22 | 16 | 1 | 0 | 0 | 15 |
| $N(h^{1,1})$ | 7231 | 7603 | 7764 | 7836 | 7858 | 7874 | 7875 | 7875 | 7875 | 7890 |
| favorable | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

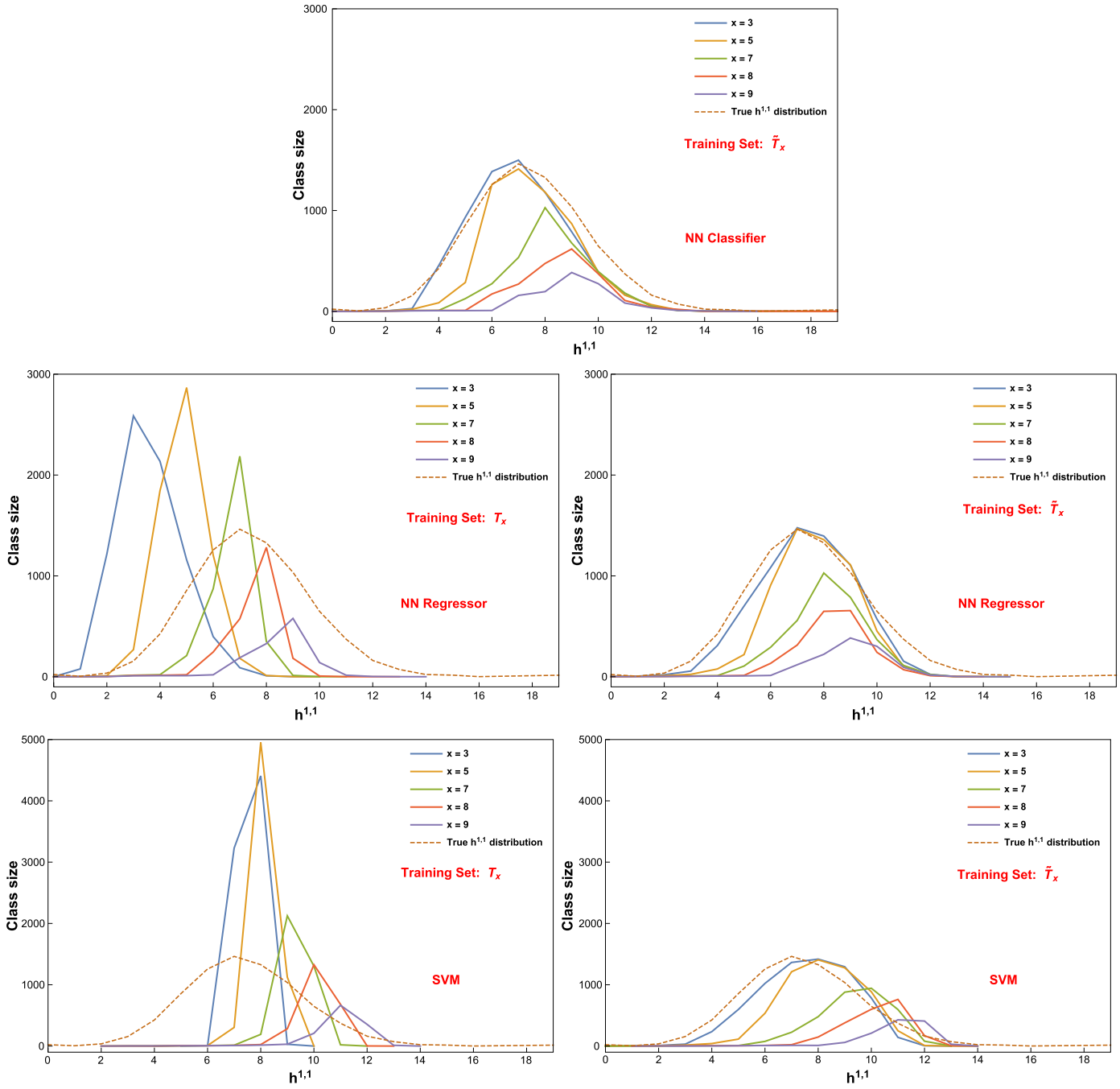


**Fig. 2.** Neural network and SVM predictions of $h^{1,1}$ for CICY threefolds. The first row shows predictions by the neural network classifier using the training set $\widetilde{T}_x$. The second row shows the neural network regressor predictions using training sets $T_x$ (left) and $\widetilde{T}_x$ (right). The third row shows the same for the SVM.

with $\sigma = 2.74$, $C = 10$, and $\epsilon = 0.01$ for predicting $h^{1,1}$, and $\sigma = 3$, and no slack for the remaining experiments. Calculations for the regressors are performed on a Lenovo Y50 laptop, i7-4700HQ, 2.4 GHz quad core with 16 GB RAM. The architecture of the neural network classifier, implemented on `Mathematica` (version 11.3), consists of two Long Short-Term Memory layers with a dropout of 0.2, each followed by a tanh and ReLU activation in sequence, and batch normalization. This is connected to two linear layers with dropout of 0.2, each (again) followed by a tanh and ReLU activation in sequence. The final components are a linear layer, followed by a tanh and ReLU activation in sequence. Each layer has 120 nodes. The penultimate layer of the neural network is a softmax layer.

## 4. Predicting $h^{1,1}$

We use machine learning to compute the Hodge number $h^{1,1}$ of CICY threefolds. Training on the configuration matrices at low $h^{1,1}$, the algorithms successfully predict trends in the distributions of Hodge numbers at higher $h^{1,1}$, but do not provide accuracy comparable to the random sampling previously studied in [21]. This is corrected by including a small selection of samples at higher $h^{1,1}$.

We set up the experiment in two parts. In the first part, we train with configuration matrices with $h^{1,1} \leq x$, and test with configuration matrices with $h^{1,1} > x$. In the second part, we repeat the experiment by augmenting the training set above with 10% of the configuration matrices with $h^{1,1} > x$, randomly sampled, and

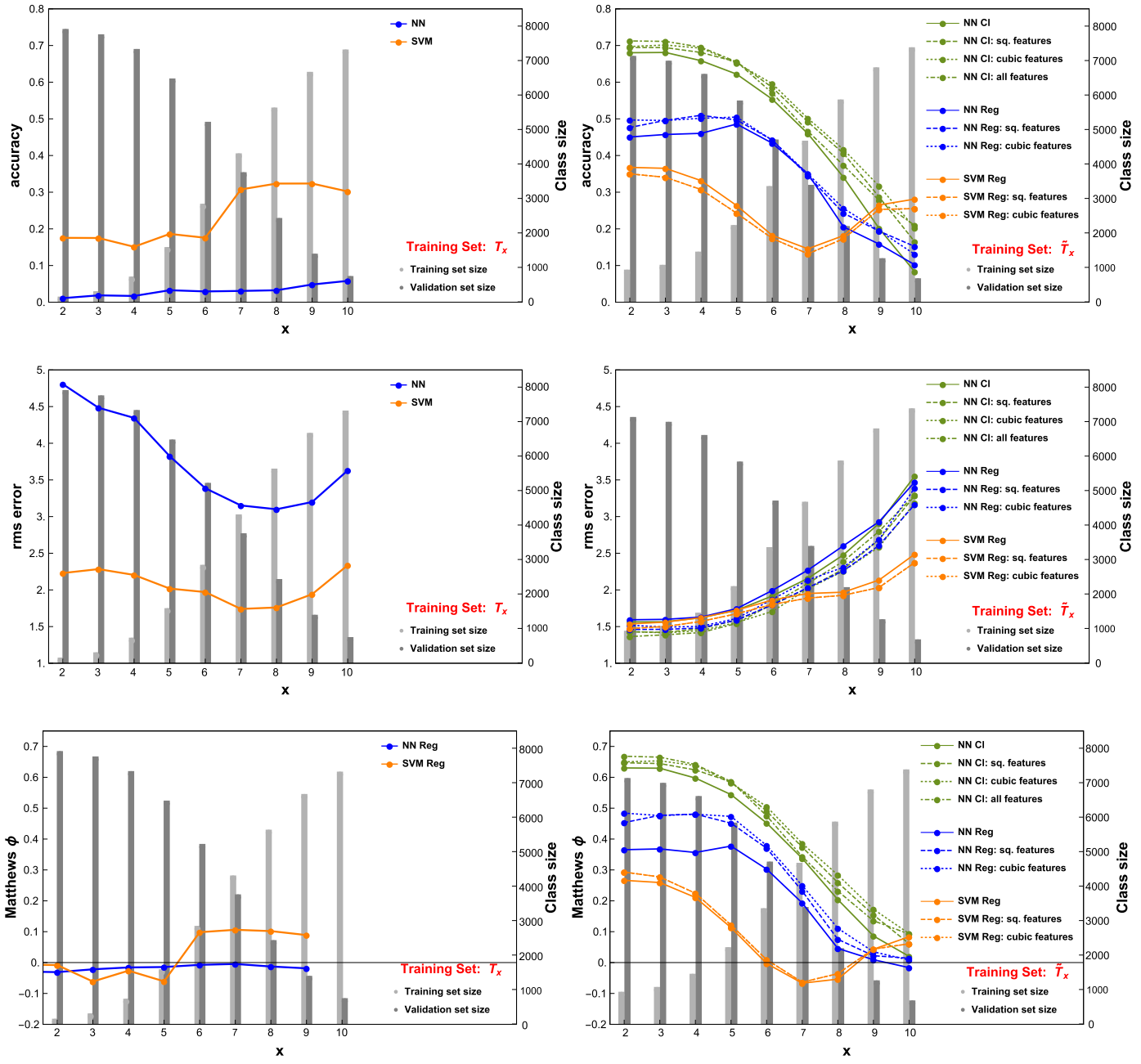


**Fig. 3.** Accuracy, rms error and Matthews correlation coefficient ($\phi$) of $h^{1,1}$ predictions for CICY threefolds, by the neural networks and SVM. Lighter bars represent the training set size, and darker bars, the validation set size. Figures on the left in each row correspond to experiments using the training set $T_x$, and the figures on the right correspond to experiments using the training set $\tilde{T}_x$. For the experiments involving $\tilde{T}_x$, we also show the effect of using the squares and cubes of the elements of the CICY configuration matrices (3) as input features. For the neural network classifier we also show the effect of including square and cubic features in addition to the original feature, the CICY matrices.

test using the remaining configuration matrices. We denote these two training sets by $T_x$ and $\tilde{T}_x$ respectively. The integer bound $x$ is a tuneable parameter. In our experiments we choose $2 \leq x \leq 10$. With reference to Table 1, the size of the first training set $T_x$ is $N(x)$, and the size of the validation set is $7890 - N(x)$. Similarly, the size of the second training set $\tilde{T}_x$ is $\tilde{N}(x) := N(x) + \lfloor \frac{7890-N(x)}{10} \rfloor$, and the size of the test set is $7890 - \tilde{N}(x)$. Using the training set $T_x$ ($\tilde{T}_x$), at $h^{1,1} = 7$, we train with $\sim 53\%$ (58%) of the dataset while at $h^{1,1} = 9$, we train with $\sim 83\%$ (85%) of the dataset.

The true distribution of CICY threefolds peaks at the value $h^{1,1} = 7$. Fig. 2 shows neural network and SVM predictions of this distribution. Fig. 3 shows the accuracy, root-mean-squared (rms) errors and Matthews correlation coefficient ($\phi$) for the predictions. The left and right panels of these figures correspond to the use of the two training sets $T_x$ and $\tilde{T}_x$ respectively, which were defined above. The neural network classifier performs rather poorly, when trained using the set $T_x$, and we exclude its predictions from Figs. 2 and 3.

Focusing first on the experiment using the training set $T_x$, wherein we use the neural network and SVM regressors, we note that the algorithms predict a peak in the $h^{1,1}$ distribution for each value of $x$, though the position of the peak is slightly incorrect. Both the algorithms consistently overpredict the number of manifolds with low $h^{1,1}$, regardless of the parameter $x$. This is not surprising since the only data the machine has seen for training are

those geometries with $h^{1,1} \leq x$. This stagnates the neural network, with it eventually predicting most of the manifolds with $h^{1,1} > x$ to have $h^{1,1} \leq x$, causing the growth in the rms error after the initial dip (Fig. 3). The dip itself corresponds to the better predictions as seen in the neural network plot (Fig. 2). From the accuracy and rms error plots (left panels in Fig. 3), we note that the SVM performs significantly better than the neural network, though the overall predictive powers of both the algorithms are limited. This analysis shows that the regressors are capable of predicting trends in the distribution of Hodge numbers from the limited data.

We now compare the results above with those from the experiment using the modified training set $\widetilde{T}_x$. The right panels in Fig. 2 show the level of agreement of the predictions with the true $h^{1,1}$ distribution, demonstrating a marked improvement in the machines' predictive ability, from above. This is further evidenced by the higher accuracies and Matthews coefficient, and lower rms errors (in the right panels of Fig. 3). This significant enhancement of predictive ability is seemingly disproportionate to the expected gain of these algorithms (especially the neural networks) from the use of an increased number of training examples. This indicates that adding a small fraction of randomly sampled data from the list of manifolds with $h^{1,1} > x$ to the training set results in significantly improved predictions. Finally, we note that the neural networks perform better than the SVM in the domain of low $x$, and the SVM performs marginally better in the domain of high $x$. The accuracy, which is lower than what we report in [21], corresponds to an exactly correct identification of a manifold's $h^{1,1}$ based on an imbalanced training set. The misidentifications follow a Gaussian profile: a prediction is more likely to be off by a little than by a lot. Even with a simple `Mathematica` implementation, the algorithm is much better at distinguishing large from larger $h^{1,1}$.

As we have noted in Section 2, the Euler character is cubic in the elements of the configuration matrix. It is also proportional to the difference between $h^{1,1}$ and $h^{2,1}$. Instead of training with the elements $m_{ij}$ of the CICY configuration matrix, suppose we use $m_{ij}^2$ or $m_{ij}^3$ as inputs.[1] We can nudge the performance slightly. The square and cubic inputs both yield nearly the same results (Fig. 3). The neural networks respond more favorably to the alternative input than the SVM.

## 5. Discussion

The difficulty of exploring the string landscape and characterizing the vacuum space of solutions is technical. We cannot perform detailed calculations, for instance, in Standard Model building, when the Hodge numbers are large. Indeed, even finding all triangulations of a reflexive polytope at $h^{1,1} \geq 7$ to determine the full set of toric Calabi–Yau threefolds that are candidate geometries for superstring compactification has not been accomplished [16]. A similar systematic effort for fourfolds in F-theory has not even been attempted. As a result, we do not know how many string vacua there are and what fraction of these resemble the real world.

Supervised machine learning provides a structure to attack this class of problems in the face of incomplete data. Studying CICY threefold geometries, this letter suggests that the strategy to employ is to compute simple examples and a representative smattering of the harder cases. This supplies the information that the machine requires to predict trends in the data and achieve results roughly comparable to sampling from the entire dataset. The same methodology should be extensible to other large datasets of Calabi-Yau manifolds including the Kreuzer-Skarke dataset [14], CICY fourfolds [50], generalised CICYs [51] and Calabi-Yau manifolds in weighted projective spaces [52]. We hope to return to these cases in the future. Something similar happens when neural networks learn the hyperbolic volume of knot complements from Jones polynomials [53]. The answers we obtain offer a starting point by flagging geometries that a string phenomenologist or a string theorist might find interesting. Because the answers are not always error-free, we view this as an example of probably approximately correct learning [54].

The topological invariants of CICY geometries are by now extremely well studied. We have therefore not learned anything new about these manifolds as a result of this investigation. The work of [18–21] and what we report here nevertheless teaches us something profound. The traditional methods for computing topological features of Calabi–Yau geometries — sequence chasing, doubly exponential Gröbner basis algorithms, etc. — may not be the most efficient way to proceed. Machine learning responds to these queries in polynomial time. We therefore conclude that there are better ways to calculate.

How does a machine learn? At the most basic level, the problems we confront in computational algebraic geometry reduce to finding the (co-)kernels of integer matrices. We have a black box that applies this process to land on useful semantics without knowing any syntax. The central open question is to dissect the black box and translate these algorithms into something a human can understand and implement. We aim to report progress in this endeavor in future work.

[1] We thank Andre Lukas for suggesting this experiment.

## Acknowledgements

YHH thanks the Science and Technology Facilities Council, UK, for grant ST/J00037X/1, the Chinese Ministry of Education, for a Chang-Jiang Chair Professorship at NanKai University and the City of Tian-Jin for a Qian-Ren Scholarship, as well as Merton College, Oxford, for her enduring support. VJ is supported by the South Africa Research Chairs Initiative of the DST/NRF through the grant SARChI 78554. CM was supported by a Severo Ochoa Fellowship at ICMAT during the preparation of this manuscript. We thank Dustin Cartwright, Mario Garcia-Fernandez, and Michele Cicoli for insightful comments. We also thank participants at the ICERM workshop on "Non-linear algebra" at Brown University and the workshop "Machine Learning Landscape" at ICTP. We are especially grateful to Andre Lukas for discussions. Some of the computations in this letter were carried out using the LOVELACE computing cluster at ICMAT.

## References

[1] P. Candelas, G.T. Horowitz, A. Strominger, E. Witten, Vacuum configurations for superstrings, Nucl. Phys. B 258 (1985) 46–74.
[2] B.R. Greene, K.H. Kirklin, P.J. Miron, G.G. Ross, A three generation superstring model. 1. Compactification and discrete symmetries, Nucl. Phys. B 278 (1986) 667–693.
[3] V. Braun, Y.-H. He, B.A. Ovrut, T. Pantev, A Heterotic standard model, Phys. Lett. B 618 (2005) 252–258, arXiv:hep-th/0501070.
[4] V. Braun, Y.-H. He, B.A. Ovrut, T. Pantev, The exact MSSM spectrum from string theory, J. High Energy Phys. 05 (2006) 043, arXiv:hep-th/0512177.
[5] V. Bouchard, R. Donagi, An SU(5) heterotic standard model, Phys. Lett. B 633 (2006) 783–791, arXiv:hep-th/0512149.
[6] L.B. Anderson, Y.-H. He, A. Lukas, Heterotic compactification, an algorithmic approach, J. High Energy Phys. 0707 (2007) 049, arXiv:hep-th/0702210.
[7] L.B. Anderson, Y.-H. He, A. Lukas, Monad bundles in heterotic string compactifications, J. High Energy Phys. 0807 (2008) 104, arXiv:0805.2875.
[8] L.B. Anderson, J. Gray, A. Lukas, E. Palti, Two hundred heterotic standard models on smooth Calabi-Yau threefolds, Phys. Rev. D 84 (2011) 106005, arXiv:1106.4804.
[9] L.B. Anderson, J. Gray, A. Lukas, E. Palti, Heterotic line bundle standard models, J. High Energy Phys. 1206 (2012) 113, arXiv:1202.1757.

[10] L.B. Anderson, A. Constantin, J. Gray, A. Lukas, E. Palti, A comprehensive scan for heterotic SU(5) GUT models, J. High Energy Phys. 01 (2014) 047, arXiv:1307.4787.
[11] A. Constantin, Y.-H. He, A. Lukas, Counting string theory standard models, arXiv:1810.00444.
[12] V.V. Batyrev, Dual polyhedra and mirror symmetry for Calabi-Yau hypersurfaces in toric varieties, J. Algebraic Geom. 3 (1994) 493–545, arXiv:alg-geom/9310003.
[13] V.V. Batyrev, L.A. Borisov, On Calabi-Yau complete intersections in toric varieties, arXiv:alg-geom/9412017.
[14] M. Kreuzer, H. Skarke, Complete classification of reflexive polyhedra in four-dimensions, Adv. Theor. Math. Phys. 4 (2002) 1209–1230.
[15] R. Altman, J. Gray, Y.-H. He, V. Jejjala, B.D. Nelson, A Calabi-Yau database: threefolds constructed from the Kreuzer-Skarke list, J. High Energy Phys. 02 (2015) 158, arXiv:1411.1418.
[16] R. Altman, J. Gray, Y.-H. He, V. Jejjala, B.D. Nelson, Toric Calabi–Yau database, http://rossealtman.com/.
[17] Y.-H. He, V. Jejjala, L. Pontiggia, Patterns in Calabi–Yau distributions, Commun. Math. Phys. 354 (2017) 477–524, arXiv:1512.01579.
[18] Y.-H. He, Deep-learning the landscape, arXiv preprint arXiv:1706.02714, 2017.
[19] Y.-H. He, Machine-learning the string landscape, Phys. Lett. B 774 (2017) 564–568.
[20] Y.-H. He, The Calabi-Yau landscape: from geometry, to physics, to machine-learning, arXiv:1812.02893.
[21] K. Bull, Y.-H. He, V. Jejjala, C. Mishra, Machine learning cicy threefolds, Phys. Lett. B 785 (2018) 65–72.
[22] T. Rudelius, Learning to inflate, arXiv preprint arXiv:1810.05159, 2018.
[23] D. Krefl, R.-K. Seong, Machine learning of Calabi-Yau volumes, Phys. Rev. D 96 (2017) 066014.
[24] D. Klaewer, L. Schlechter, Machine learning line bundle cohomologies of hypersurfaces in toric varieties, arXiv preprint arXiv:1809.02547, 2018.
[25] Y.-N. Wang, Z. Zhang, Learning non-higgsable gauge groups in 4d f-theory, arXiv preprint arXiv:1804.07296, 2018.
[26] F. Ruehle, Evolving neural networks with genetic algorithms to study the string landscape, J. High Energy Phys. 2017 (2017) 38.
[27] J. Liu, Artificial neural network in cosmic landscape, J. High Energy Phys. 2017 (2017) 149.
[28] A. Mütter, E. Parr, P.K. Vaudrevange, Deep learning in the heterotic orbifold landscape, arXiv preprint arXiv:1811.05993, 2018.
[29] J. Carifio, J. Halverson, D. Krioukov, B.D. Nelson, Machine learning in the string landscape, J. High Energy Phys. 2017 (2017) 157.
[30] J. Carifio, W.J. Cunningham, J. Halverson, D. Krioukov, C. Long, B.D. Nelson, Vacuum selection from cosmology on networks of string geometries, arXiv:1711.06685.
[31] S. Abel, J. Rizos, Genetic algorithms and the search for viable string vacua, J. High Energy Phys. 2014 (2014) 10.
[32] J. Halverson, F. Ruehle, Computational complexity of vacua and near-vacua in field and string theory, arXiv preprint arXiv:1809.08279, 2018.
[33] R. Altman, J. Carifio, J. Halverson, B.D. Nelson, Estimating Calabi-Yau hypersurface and triangulation counts with equation learners, arXiv preprint arXiv:1811.06490, 2018.
[34] V. Braun, On free quotients of complete intersection Calabi-Yau manifolds, J. High Energy Phys. 04 (2011) 005.
[35] P. Candelas, A.M. Dale, C.A. Lutken, R. Schimmrigk, Complete intersection Calabi-Yau manifolds, Nucl. Phys. B 298 (1988) 493.
[36] F. Schöller, H. Skarke, All weight systems for Calabi-Yau fourfolds from reflexive polyhedra, arXiv:1808.02422.
[37] W. Taylor, Y.-N. Wang, Scanning the skeleton of the 4D F-theory landscape, J. High Energy Phys. 01 (2018) 111, arXiv:1710.11235.
[38] T. Hubsch, Calabi-Yau Manifolds: A Bestiary for Physicists, World Scientific, Singapore, 1994.
[39] A. Lukas, L. Anderson, J. Gray, Y.-H. He, S.-J.L. Lee, The cicy list including Hodge number data, available at http://www-thphys.physics.ox.ac.uk/projects/calabiyau/cicylist/index.html, 2007.
[40] P.S. Green, C. Lütken, T. Hübsch, All Hodge numbers of all complete intersection Calabi-Yau manifolds, Class. Quantum Gravity 6 (1987) 105–124.
[41] P. Candelas, R. Davies, New Calabi-Yau manifolds with small Hodge numbers, Fortschr. Phys. 58 (2010) 383–466, arXiv:0809.4681.
[42] P. Candelas, A. Constantin, Completing the web of $Z_3$ – quotients of complete intersection Calabi-Yau manifolds, Fortschr. Phys. 60 (2012) 345–369, arXiv:1010.1878.
[43] P. Candelas, A. Constantin, C. Mishra, Hodge numbers for cicys with symmetries of order divisible by 4, Fortschr. Phys. 64 (2016) 463–509.
[44] A. Constantin, J. Gray, A. Lukas, Hodge numbers for all cicy quotients, J. High Energy Phys. 2017 (2017) 1.
[45] C. Mishra, Calabi-Yau Manifolds, Discrete Symmetries and String Theory, PhD thesis, 2017.
[46] A. Lukas, C. Mishra, Discrete symmetries of complete intersection Calabi-Yau manifolds, arXiv preprint arXiv:1708.08943, 2017.
[47] P. Candelas, C. Mishra, Highly symmetric quintic quotients, Fortschr. Phys. 66 (2018) 1800017.
[48] L.B. Anderson, X. Gao, J. Gray, S.-J. Lee, Fibrations in cicy threefolds, J. High Energy Phys. 2017 (2017) 77.
[49] The maximally favorable CICY list, http://www1.phys.vt.edu/cicydata/.
[50] J. Gray, A.S. Haupt, A. Lukas, All complete intersection Calabi-Yau four-folds, J. High Energy Phys. 07 (2013) 070, arXiv:1303.1832.
[51] L.B. Anderson, F. Apruzzi, X. Gao, J. Gray, S.-J. Lee, A new construction of Calabi–Yau manifolds: generalized cicys, Nucl. Phys. B 906 (2016) 441–496.
[52] P. Candelas, M. Lynker, R. Schimmrigk, Calabi-Yau manifolds in weighted p4, Nucl. Phys. B 341 (1990) 383–402.
[53] V. Jejjala, A. Kar, O. Parrikar, Deep learning the hyperbolic volume of a knot, arXiv:1902.05547.
[54] L.G. Valiant, A Theory of the Learnable, 1984.