International Journal of Modern Physics A
Vol. 40, No. 28 (2025) 2550097 (17 pages)

DOI: 10.1142/S0217751X25500976

World Scientific
www.worldscientific.com

Machine learning Kreuzer–Skarke Calabi–Yau threefolds

Per Berglund [*,‡], Ben Campbell [*,§] and Vishnu Jejjala [†,¶]

[*]*Department of Physics and Astronomy,*
University of New Hampshire, Durham, NH 03824, USA

[†]*Mandelstam Institute for Theoretical Physics,*
School of Physics, NITheCS, and CoE-MaSS,
University of the Witwatersrand,
Johannesburg, WITS 2050, South Africa

[‡]*Per.Berglund@unh.edu*
[§]*Ben.Campbell@unh.edu*
[¶]*v.jejjala@wits.ac.za*

Received 9 April 2025
Revised 26 May 2025
Accepted 27 May 2025
Published 21 August 2025

By using a fully connected feedforward neural network, we study topological invariants of a class of Calabi–Yau manifolds constructed as hypersurfaces in toric varieties associated with reflexive polytopes from the Kreuzer–Skarke database. In particular, we find the existence of a simple expression for the Euler number that can be learned in terms of limited data extracted from the polytope and its dual.

Keywords: Calabi–Yau manifolds; reflexive polytopes; machine learning.

1. Introduction

Superstring theory proposes that we inhabit a 10-dimensional space–time. The Universe we observe, however, is plainly four-dimensional. One way to reconcile the mathematical consistency of quantum gravity with empirical facts about the world is for the predicted extra dimensions to be compact with a size determined by the string length. The energy scale needed to resolve such distances precludes experimental measurements of the extra dimensions. Compactification of the $E_8 \times E_8$ heterotic string on a Calabi–Yau threefold then presents a straightforward route to particle physics in the real world, or at the very least to a four-dimensional quantum field theory which has a non-Abelian gauge group and low energy chiral matter.[1,2] In the simplest models, the number of generations of

[¶]Corresponding author.

particles in the spectrum is $\frac{1}{2}|\chi(M)|$, with $\chi(M)$ being the Euler number of the compact Calabi–Yau manifold, M.

A Calabi–Yau space is a complex, Kähler manifold that admits a Ricci flat metric. (See Refs. 3 and 4 for pedagogical reviews.) We must know the flat metric in order to compute features of the particle physics spectrum such as the Yukawa couplings and patterns of supersymmetry breaking. However, topology by itself carries us far, and following Refs. 5 and 6, there are now thousands of semi-realistic constructions of the supersymmetric Standard Model from string compactification. The taxonomy of these spaces is involved. Some Calabi–Yau geometries arise as complete intersection hypersurfaces determined as the zero locus of polynomials in products of complex projective spaces, while others are hypersurfaces in toric varieties or are elliptically fibered. These different species of Calabi–Yau manifolds have a nontrivial Venn diagram. In what follows we will focus on the hypersurface case.

More specifically, we study a class of n complex dimensional spaces in which the Calabi–Yau manifold is described as an anticanonical divisor within an $(n+1)$ complex dimensional toric variety arising from a reflexive polytope, Δ, following the work of Batyrev.[7] Kreuzer and Skarke classified the lower dimensional cases, finding 4319 three-dimensional reflexive polyhedra[8] that give K3 (the Calabi–Yau twofold) and $473,800,776$ four-dimensional reflexive polyhedra[9,10] leading to an unknown number of Calabi–Yau threefolds, while to date, there is only a partial list of the five-dimensional reflexive polyhedra[11] that give elliptically fibered Calabi–Yau fourfolds suitable for F-theory model building. We will investigate the $n = 3$ dimensional Calabi–Yau spaces and the associated topological data, constructed from the class of reflexive four-dimensional Δ. Starting from a given Δ, we may algorithmically obtain a Calabi–Yau manifold from a consistent triangulation of the dual polytope, Δ^*; see Refs. 12–14 for recent work detailing the explicit constructions of Calabi–Yau threefolds from polytopes. In particular, the Hodge numbers $h^{1,1}$ and $h^{2,1}$, which count, respectively, Kähler and complex structure deformations, as well as the Euler number, $\chi = 2(h^{1,1} - h^{2,1})$, are all explicitly given in terms of the combinatorial properties of Δ and its dual, Δ^*.[7,a]

Since the Kreuzer–Skarke catalogue of four-dimensional reflexive polytopes has nearly half a billion entries, it is securely within the realm of Big Data. Machine learning has emerged as a tool for analyzing such large data sets in string theory, in particular focusing on the topology of Calabi–Yau manifolds.[16–19] (See Ref. 20 for a review.) Indeed, some of the initial investigations in this direction looked at the complete intersection Calabi–Yau threefolds. There are 7890 such geometries,[21] each of which is characterized by a configuration matrix that encodes the polynomial equations describing the complete intersection algebraic variety, M. Knowledge of this matrix is sufficient to predict the topological invariants $h^{1,1}$ and $\chi(M)$.[22–25] Orientifold threefolds are studied using machine learning in Refs. 26 and 27.

[a] The Hodge data do not by any means uniquely define the Calabi–Yau manifold. For example, there are nearly one million polyhedra with $(h^{1,1}, h^{2,1}) = (27, 27)$. The statistics of these polytopes is discussed in Ref. 15.

Similar investigations have been carried out for the complete intersection Calabi–Yau fourfolds.[28–30] Since the calculations scale polynomially with the size of the configuration matrix instead of doubly exponentially as would be expected from sequence chasing or Gröbner basis methods in computational algebraic geometry, one of the conclusions of Refs. 22 and 23 is that there may be more efficient ways to calculate the Hodge numbers. In this paper, we machine learn topological invariants of the Calabi–Yau threefolds constructed as hypersurfaces in toric varieties from a limited amount of data describing four-dimensional polytopes and dual polytopes taken from the Kreuzer–Skarke list.[10,31,b]

Finally, it is well known that artificial intelligence methods excel at finding associations between features in data. In certain cases, neural network-based curve fitting has facilitated the search for new analytic formulæ. Instances of this occur in examining line bundle cohomology on surfaces and on Calabi–Yau threefolds[34–37] and in analyzing the topological invariants of knots.[38–43] Here, we use the machine learning results concerning reflexive polytopes to deduce new analytic expressions for topological invariants.

The organization of this paper is as follows. In Sec. 2, we review the Kreuzer–Skarke data set. In Sec. 3, we detail the machine learning methods we employ. In Sec. 4, we describe the machine learning predictions of toric Calabi–Yau threefold Hodge numbers from four-dimensional reflexive polytope data. In Sec. 5, we present a new analytic expression for the Euler number. In Sec. 6, we provide a prospectus for future work in this direction.

2. Kreuzer–Skarke Data

Kreuzer and Skarke tabulated all $473,800,776$ reflexive polyhedra in four dimensions.[9,10] Starting from any one of these polytopes, Δ, we can construct a four-dimensional toric variety X_Δ in which the anticanonical hypersurface is a (possibly) singular Calabi–Yau variety realized as a generic section of the anticanonical bundle of X_Δ.[7] In general, each such hypersurface may admit several maximal projective crepant partial desingularizations each associated to a given triangulation of the dual polytope, Δ^*, with distinct Stanley–Reisner rings corresponding to different geometries. Furthermore, the same Calabi–Yau threefold could as well arise from triangulating different four-dimensional reflexive polytopes. In one lower dimension, this is what happens for K3: whereas there are 4319 reflexive polytopes in three dimensions, any two complex analytic K3 surfaces are diffeomorphic as smooth four manifolds.[44] The Calabi–Yau twofold is essentially unique. In contrast, we do not have even a rough enumeration of the corresponding Calabi–Yau threefolds. It is, however, likely that this number is well in excess of the half a billion reflexive polytopes.

[b] See also Ref. 32. As well, polytopes were studied with machine learning in Ref. 33.

We focus our attention in this paper on a subset of the topological properties of the Calabi–Yau spaces obtained from the polytope data. For the class of four-dimensional reflexive polytopes there are 30,108 unique pairs of Hodge numbers $(h^{1,1}, h^{2,1})$ of the associated Calabi–Yau manifold. This sets a lower bound on the number of Calabi–Yau threefolds. Our goal is to compute these topological invariants of a Calabi–Yau threefold using only minimal information about the polytopes.

Let us briefly review what the polytope data provide. For toric geometry, we must specify a reflexive polytope Δ and its dual Δ^*. A polytope in $\mathbb{R}^4$ is the convex hull of finitely many lattice points, $m \in \Delta$, the number of which is denoted by $l(\Delta)$. That is to say, we specify the vertices with integer valued vectors. Pairs of neighboring vertices define an edge, collections of edges define a face, etc. Adopting the nomenclature of Ref. 8, a polytope is said to have the *interior point (IP)* property when it contains the origin as its sole interior point with integer coordinates. The *dual (polar) polytope* is defined as

$$\Delta^* = \{v \in \mathbb{R}^4 | \langle m, v\rangle \geq -1 \quad \forall m \in \Delta\}, \tag{2.1}$$

where the inner product $\langle m, v\rangle$ is calculated in $\mathbb{R}^4$. The polytope is *reflexive* when the vertices of Δ^* are specified by integer vectors, and Δ^* shares the IP property, from which it follows that $(\Delta^*)^* = \Delta$. Furthermore, the convex hull of Δ^* consists of $l(\Delta^*)$ points. The Calabi–Yau hypersurface M is constructed as a generic section of the anticanonical bundle, $-K_X$, on X_Δ, and explicitly given by the following vanishing condition:

$$0 = \sum_{m\in\Delta} a_m \prod_i x_i^{\langle m, v_i^*\rangle + 1}, \tag{2.2}$$

where v_i^* are vertices of Δ^*, x_i are coordinates on the toric variety, X_Δ, and a_m are coefficients that parameterize the complex structure of M. Exchanging Δ and Δ^* provides an analogous construction of the mirror Calabi–Yau W, for which $h^{1,1}$ and $h^{2,1}$ are interchanged:

$$0 = \sum_{n\in\Delta^*} b_n \prod_i y_i^{\langle n, v_i\rangle + 1}, \tag{2.3}$$

where now the v_i are vertices of Δ, y_i are coordinates on the "mirror" toric variety, X_{Δ^*}, and b_m are coefficients that parameterize the complex structure of the mirror manifold W.[c]

In what follows, we as well introduce the scaled up polytopes $k\Delta$ and $k\Delta^*$, where $k = 2$ is the integer scale factor. Note that neither $k\Delta$ nor $k\Delta^*$ are reflexive polyhedra in their own right. To define the scaled up objects, we simply multiply the vectors that define the vertices of Δ and Δ^* by $k = 2$, with $l(2\Delta)$ and $l(2\Delta^*)$ the number of points in the convex hull of the scaled up polytopes, respectively. Remarkably, we find that specifying $(l(\Delta), l(\Delta^*), l(2\Delta), l(2\Delta^*))$, i.e. the number of lattice points on

[c]Here, $n \in \Delta^*$ includes all the lattice points in Δ^*, just like $m \in \Delta$ containing all the lattice points in Δ.

Δ, and the corresponding data for Δ^*, 2Δ and $2\Delta^*$, is sufficient to machine learn topological invariants of the Calabi–Yau threefold.

Our analysis contrasts with machine learning efforts on complete intersection Calabi–Yau manifolds.[24,25,30,39,45] There, the configuration matrix provides the complete information about how the geometry is realized. Here, we are not even providing the full polytope data, viz., the vectors that identify the vertices of Δ (and by duality, the vertices of Δ^*). To specify the desingularized Calabi–Yau geometry, we must go even further by triangulating the polytope.

3. Machine Learning Methods

A fully connected feedforward neural network accomplishes a nonlinear regression. To an input vector $\boldsymbol{v}$, we associate the output

$$f_\theta(\boldsymbol{v}) = \|L^{(n)}(\sigma_{n-1}(L^{(n-1)}(\ldots\sigma_1(L^{(1)}(\boldsymbol{v}))))\|_1, \tag{3.1}$$

where

$$\boldsymbol{v}^{(k)} = \sigma_k(L^{(k)}(\boldsymbol{v}^{(k-1)})), \quad L^{(k)}(\boldsymbol{v}^{(k-1)}) = W^{(k)}\cdot\boldsymbol{v}^{(k-1)} + \boldsymbol{b}^{(k)}, \tag{3.2}$$

with $\boldsymbol{v}^{(0)} = \boldsymbol{v}$. This is an n layer neural network. The θ collectively denotes a set of hyperparameters, which are the matrix elements of the weight matrices $W^{(k)}$ and the components of the bias vectors $\boldsymbol{b}^{(k)}$. If $W^{(k)}$ is an $m_k \times m_{k-1}$ matrix, there are m_k neurons in the kth hidden layer. The functions σ_k implement the nonlinearity elementwise on $L^{(k)}(\boldsymbol{v}^{(k-1)})$. Suppose we know that we should ascribe the result ϕ_i to an input vector $\boldsymbol{v}_i$. The hyperparameters are randomly initialized and set in training by optimizing a loss function

$$\Upsilon(\theta) = \sum_i \rho(\boldsymbol{v}_i) d(f_\theta(\boldsymbol{v}_i), \phi_i) \tag{3.3}$$

over a collection of input vectors, where the summands in (3.3) are a suitable measure of the difference d between the neural network prediction $f_\theta(\boldsymbol{v}_i)$ and the true answer ϕ_i, weighted by some density ρ.

The models in these experiments are implemented using `Julia 1.6.2`[46] and the `Flux` package.[47,48] Each model consists of a neural network with five hidden layers with 300 neurons per layer. We use rectified linear unit (ReLU) activation functions so that $\sigma(x) = \max(0, x)$ with the output layer given by a sigmoid function. The sigmoid ensures that outputs are always in the interval $[0, 1]$. As we are treating this as a classification problem, this is compatible with the boundedness of the Hodge numbers: $h^{1,1}$ and $h^{2,1}$ range from $[1, 491]$, and similarly the Euler number, χ, ranges from $[-960, 960]$. Optimization is enacted via the Adam algorithm[49] and a logit cross entropy loss function. The initial learning rate is $\eta = 10^{-3}$, and a learning rate schedule is set such that after eight epochs with no improvement the learning rate is halved until it is less than $\eta = 10^{-5}$, after which η is fixed. (The learning rate is a parameter of the neural network that determines how much to change the model in

response to the estimated error at each iteration.) The maximum number of training epochs, or passes through the entire training data, is set at 50 as longer training shows no improvement. There is no minimum number of training epochs required and the models are allowed to stop training early. No dropout layers were used in the models as their inclusion showed no improvement. We have not attempted to optimize the architecture of the neural networks; for the purposes of learning Hodge numbers from polytope data, a network of this type is sufficient.

Of the roughly half-billion four-dimensional polytopes from the Kreuzer–Skarke database, a random sample of one million polytopes was selected, as shown in Fig. 1. The model was trained on 80% of the polytopes and tested on the complementary 20% of the data set. This procedure was then repeated 100 times, where the total sample is shuffled and split before each model is trained; thus each model was trained and tested on a different selected subset of the originally sampled one million polytopes. The input data are given by the input vector $\boldsymbol{v} = (l(\Delta), l(\Delta^*), l(2\Delta), l(2\Delta^*))$, that is the number of lattice points on Δ, its dual Δ^*, as well as the twice scaled up polytopes, 2Δ and $2\Delta^*$, respectively, data which are calculated from any given Kreuzer–Skarke polytope Δ. The output data are given by a one-hot encoded vector with 961 entries when learning to predict χ and 491 entries when learning to predict $h^{1,1}$, $h^{2,1}$ or $h^{1,1} + h^{2,1}$, due to the allowed finite range of values for these topological invariants. As the topological data of the Calabi–Yau threefold are integers, it is natural to treat this as a classification problem. The set of hyperparameters is chosen to maximize accuracy in training $\chi = 2(h^{1,1} - h^{2,1})$. No appreciable difference in accuracy was found when different sets of optimal hyperparameters were obtained from training to predict $h^{1,1}$, $h^{2,1}$ and $h^{1,1} + h^{2,1}$.

The mirror symmetric plot of $h^{1,1} + h^{2,1}$ versus $h^{1,1} - h^{2,1}$ is densely populated in the center. As a result, the majority of the randomly selected data comes from this region.

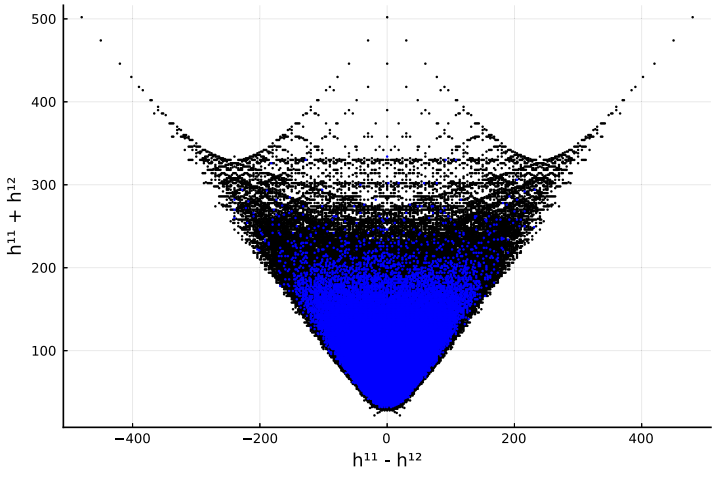

Fig. 1. (Color online) Distribution of randomly sampled data as a plot of $h^{1,1} - h^{2,1}$ versus $h^{1,1} + h^{2,1}$. Blue represents an $(h^{1,1}, h^{2,1})$ pair for which a polytope has been randomly selected.

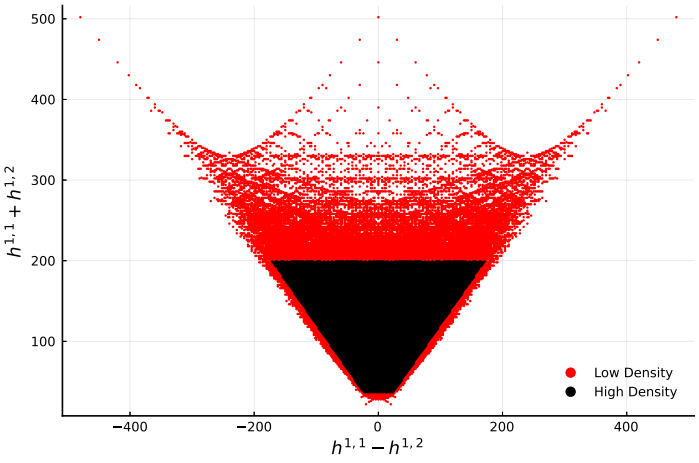

Fig. 2. (Color online) Distribution of boundary data as a plot of $h^{1,1} - h^{2,1}$ versus $h^{1,1} + h^{2,1}$. Red represents an $(h^{1,1}, h^{2,1})$ pair for which all polytopes with this data are sampled.

The machine learning trials were therefore repeated using only data on the boundary, where the boundary is approximated by four linear conditions for ease, see Fig. 2. Models were trained to predict χ, $h^{1,1}$, $h^{2,1}$ and $h^{1,1} + h^{2,1}$ with the same 80% training to 20% testing ratio as in the original set of experiments.[d] The data are again split and shuffled before each model was trained. The set of hyperparameters was varied, though no considerable change in performance was obtained compared to the values used for the original randomly selected data set. However, the maximum number of training epochs was increased to 100 as this allowed for improvement in accuracy as opposed to the randomly sampled data.

4. Numerical Results

A plot of $h^{1,1} - h^{2,1}$ versus $h^{1,1} + h^{2,1}$ for the randomly selected data and the chosen boundary data is shown in Figs. 1 and 2, respectively. Models are trained to predict χ, $h^{1,1}$, $h^{2,1}$ and $h^{1,1} + h^{2,1}$ for each data set. In a given trial, the mean absolute error for $h^{1,1}$ is defined by taking the expectation value

$$\delta_{h^{1,1}} = \langle |h^{1,1}_{\text{predicted}} - h^{1,1}_{\text{true}}| \rangle \tag{4.1}$$

over the test data set, which is complementary to the collection of vectors used in training. In other words, the one-hot vector is used to make a prediction of the Hodge number, and we are comparing this to the known true value. Similar expressions compute $\delta_{h^{2,1}}$, $\delta_{h^{1,1}+h^{2,1}}$ and δ_χ. The mean relative absolute error $\hat{\delta}$ normalizes by dividing the difference between the predicted and true values by the true value.

[d]The case $h^{1,1} + h^{2,1}$ was included as the natural "dual" expression to χ, though in fact the testing accuracy was the lowest of the different labels used.

When we compute the mean absolute relative error for χ, we normalize by $|\chi_{\text{true}}|$ and exclude the cases where this vanishes.

For predicting χ, the models averaged an accuracy of $97.36\% \pm 1.42\%$ and an absolute error of 0.1746 ± 0.1941 when trained and tested on the randomly sampled data and an accuracy of $96.02\% \pm 1.24\%$ and absolute error of 0.2560 ± 0.0702 on the boundary data. Models trained and tested on the boundary data did better overall for the remaining trials. For $h^{1,1}$, models trained on the randomly sampled data averaged an accuracy of $46.89\% \pm 0.91\%$ and an absolute error of 0.7099 ± 0.0966 while models trained on the boundary data averaged an accuracy of $75.35\% \pm 1.08\%$ and absolute error of 0.3142 ± 0.0494. Similar results hold for $h^{2,1}$ with models averaging an accuracy of $46.74\% \pm 1.03\%$ and an absolute error of 0.7262 ± 0.0896 when trained on the randomly sampled data while averaging and accuracy of $75.48\% \pm 0.78\%$ and absolute error of 0.3131 ± 0.0632. Finally, for predicting $h^{1,1} + h^{2,1}$, models averaged an accuracy of $32.64\% \pm 0.27\%$ and absolute error of 1.464 ± 0.046 for the randomly sampled data and an accuracy of $67.77\% \pm 1.60\%$ and absolute error of 0.7122 ± 0.0649. The accuracies are sometimes low, but this comparison queries whether the prediction exactly matches the true value. The absolute errors indicate that the wrong predictions of the neural network are not very wrong. This may suggest that the minimal data about the reflexive polytope we have provided to the network supplies the leading term of an exact expression for the Hodge numbers, which is then corrected by subleading terms that are some function of data not provided. The results for models trained on randomly sampled and boundary data are enumerated in Tables 1 and 2, respectively. The standard deviations are obtained from computing the variance over 100 runs. The results of other experiments are summarized in Tables 3–6.

Confusion matrices for models trained and tested on the randomly sampled data and trained and tested on the boundary are shown in Figs. 3 and 4, respectively. The model used to generate the confusion matrix for the randomly sampled data achieved an accuracy of 99.25% and mean absolute error of 0.0123 on the randomly sampled testing data while the model used to generate the confusion matrix for the boundary data had an accuracy of 96.32% and mean absolute error of 0.1339. Models trained on the randomly selected data are also evaluated on the boundary data to see how

Table 1. Mean accuracy, absolute error and relative absolute error for model predictions on each label trained and tested on the randomly sampled data. Averages and standard deviations are taken over 100 models for each label. The total data are shuffled before being split into 80% training and 20% testing sets for each model.

Label	Accuracy (%)	Absolute error	Relative absolute error
χ	97.36 ± 1.42	0.1746 ± 0.1941	0.0049 ± 0.0099
$h^{1,1}$	46.89 ± 0.91	0.7099 ± 0.0966	0.0222 ± 0.0029
$h^{2,1}$	46.74 ± 1.03	0.7262 ± 0.0896	0.0227 ± 0.0026
$h^{1,1} + h^{2,1}$	32.64 ± 0.27	1.464 ± 0.046	0.0214 ± 0.0006

Table 2. Mean accuracy, absolute error and relative absolute error for model predictions on each label trained and tested on the boundary data. Averages and standard deviations are taken over 100 models for each label. The total data is shuffled before being split into 80% training and 20% testing sets for each model.

Label	Accuracy (%)	Absolute error	Relative absolute error
χ	96.02 ± 1.24	0.2560 ± 0.0702	0.0035 ± 0.0018
$h^{1,1}$	75.35 ± 1.08	0.3142 ± 0.0494	0.0090 ± 0.0008
$h^{2,1}$	75.48 ± 0.78	0.3131 ± 0.0632	0.0089 ± 0.0009
$h^{1,1} + h^{2,1}$	67.77 ± 1.60	0.7122 ± 0.0649	0.0075 ± 0.0007

Table 3. Fivefold cross-validation for models trained on the Euler number. Each table entry represents a different model with initially randomized parameters and denotes the accuracy of that model. For each row a random sample of 1,000,000 polytopes from the entire KS database is taken and partitioned into five groups. Each column represents which of the five groups is used for testing; the model associated with that entry is trained on the remaining four groups. The average accuracy for each random sample is given in the last column.

χ	1	2	3	4	5	Average
1	0.9424	0.9424	0.9421	0.957	0.9458	0.94594
2	0.9286	0.9358	0.9489	0.9326	0.9538	0.93994
3	0.9467	0.9236	0.9048	0.9518	0.9171	0.9288
4	0.9493	0.9462	0.9444	0.9319	0.9446	0.94328
5	0.9462	0.9203	0.9401	0.9483	0.9409	0.93916

well they extrapolate outside of the central region of Fig. 1. Predicting χ from boundary data with a model trained on randomly sampled data resulted in an accuracy of 78.43% and average absolute error of 1.19.

The ability of the neural network to predict the topological invariants with some accuracy suggests the existence of approximate analytic formulæ. To test this hypothesis,

Table 4. Fivefold cross-validation for models trained on the $h^{1,1}$. Each table entry represents a different model with initially randomized parameters and denotes the accuracy of that model. For each row a random sample of 1,000,000 polytopes from the entire KS database is taken and partitioned into five groups. Each column represents which of the five groups is used for testing; the model associated with that entry is trained on the remaining four groups. The average accuracy for each random sample is given in the last column.

$h^{1,1}$	1	2	3	4	5	Average
1	0.4754	0.4767	0.4758	0.4836	0.4791	0.47812
2	0.4731	0.483	0.4754	0.4719	0.477	0.47608
3	0.4696	0.4747	0.4792	0.4768	0.4755	0.47516
4	0.4719	0.4788	0.4776	0.46	0.468	0.47126
5	0.4615	0.4772	0.4672	0.469	0.4645	0.46788

Table 5. Fivefold cross-validation for models trained on the $h^{2,1}$. Each table entry represents a different model with initially randomized parameters and denotes the accuracy of that model. For each row a random sample of 1,000,000 polytopes from the entire KS database is taken and partitioned into five groups. Each column represents which of the five groups is used for testing; the model associated with that entry is trained on the remaining five groups. The average accuracy for each random sample is given in the last column.

$h^{2,1}$	1	2	3	4	5	Average
1	0.4711	0.4809	0.4782	0.4774	0.476	0.47672
2	0.4753	0.4739	0.4776	0.4714	0.471	0.47384
3	0.4703	0.4695	0.4766	0.4769	0.4734	0.47334
4	0.477	0.4771	0.4716	0.4701	0.4772	0.4746
5	0.4747	0.4714	0.47	0.4714	0.473	0.4721

Table 6. Fivefold cross-validation for models trained on the topological invariants χ, $h^{1,1}$ and $h^{2,1}$. A random sample of 1,000,000 polytopes from the boundary is taken and partitioned into five groups. Each column represents which of the five groups is used for testing; the model associated with that entry is trained on the remaining four groups. The average accuracy for each random sample is given in the last column.

Label	1	2	3	4	5	Average
χ	0.9183	0.8827	0.9	0.8901	0.9214	0.9025
$h^{1,1}$	0.7077	0.7099	0.7169	0.7018	0.7223	0.71172
$h^{2,1}$	0.7169	0.7183	0.7096	0.7117	0.7177	0.71484

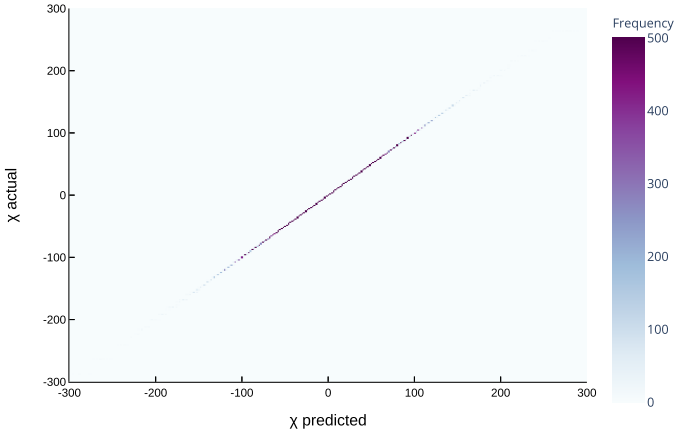

Fig. 3. Confusion matrix for model trained on randomly sampled data evaluated on randomly sampled data. This model achieved an accuracy of 99.25% and mean absolute error of 0.0123. The range has been cropped to $\chi \in [-300, 300]$ as the majority of the data is in this interval.

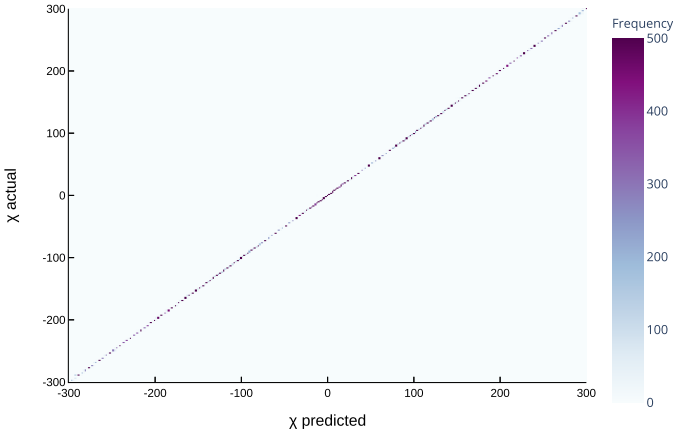

Fig. 4. Confusion matrix for model trained on boundary data evaluated on boundary data. This model achieved an accuracy of 96.43% and a mean absolute error of 0.1339. The range has been cropped to $\chi \in [-300, 300]$ for comparison with Fig. 3.

we considered a random sample of 200,000 four-dimensional reflexive polytopes and their mirrors from the Kreuzer–Skarke list. Taking the number of points in $l(\Delta)$, the number of points in $l(\Delta^*)$, the number of points in $l(2\Delta)$, and the number of points in $l(2\Delta^*)$, we performed a linear regression on these data to fit expressions for $h^{1,1}$ and $h^{2,1}$ using `Mathematica 12.3.1`.[50] Selecting half of these polytopes and their mirrors at random and repeating the regression 100 times, we find

$$\begin{aligned} h^{1,1} = &-(3.33 \pm 0.05) - (3.471 \pm 0.005) l(\Delta) + (5.529 \pm 0.005) l(\Delta^*) \\ &+ (0.4176 \pm 0.0007) l(2\Delta) - (0.5824 \pm 0.0007) l(2\Delta^*), \end{aligned} \tag{4.2}$$

$$\begin{aligned} h^{2,1} = &-(3.33 \pm 0.05) + (5.529 \pm 0.005) l(\Delta) - (3.471 \pm 0.005) l(\Delta^*) \\ &- (0.5824 \pm 0.0007) l(2\Delta) + (0.4176 \pm 0.0007) l(2\Delta^*). \end{aligned} \tag{4.3}$$

Plugging in the values of $l(\Delta)$, $l(\Delta^*)$, $l(2\Delta)$ and $l(2\Delta^*)$ into (4.2) and rounding to the nearest integer predicts $h^{1,1}$ correctly 44% of the time. The prediction is off by one another 44% of the time. In total, the mean of the absolute value of the error in the prediction is 0.70 ± 0.85 on the full data set. Similarly, the prediction of $h^{2,1}$ in (4.3) is correct 46% of the time and off by one 42% of the time. In total, the mean of the absolute value of the error in the prediction is 0.70 ± 1.09.

If we fit instead to $h^{1,1} + h^{2,1}$ and $h^{1,1} - h^{2,1}$, we find

$$\begin{aligned} h^{1,1} + h^{2,1} = &-(6.64 \pm 0.01) + (2.06 \pm 0.01)(l(\Delta) + l(\Delta^*)) \\ &- (0.165 \pm 0.001)(l(2\Delta) + l(2\Delta^*)), \end{aligned} \tag{4.4}$$

$$\frac{1}{2}\chi = h^{1,1} - h^{2,1} = 9(l(\Delta^*) - l(\Delta)) + (l(2\Delta) - l(2\Delta^*)). \tag{4.5}$$

The formula for the sum of the Hodge numbers (4.4) is exactly correct 23% of the time and off by one 38% of the time. The mean of the absolute value of the error in

the prediction is 1.45 ± 1.39. The formula for the difference of the Hodge numbers (4.5), on the other hand, is exact. The coefficients in the fit are integer and have zero error. The prediction is perfect for every polytope. We will now derive this expression as a new analytic formula for the Euler number.

Ab initio, we could have just used linear regression to achieve these results. However, we did not know that the appropriate features of the reflexive polytope to look at are $l(\Delta)$, $l(\Delta^*)$, $l(2\Delta)$ and $l(2\Delta^*)$. The number of points in the scaled up polytope does not have an obvious geometric or physical meaning. Performing a number of machine learning experiments whose results we do not report here precisely because they are uninteresting or produce null results enabled us to determine that these particular features can approximately reproduce Calabi–Yau Hodge numbers. Thus, we have performed a regression *a posteriori* once we knew which features to fit. Since similar formulæ for the Euler character of complete intersection Calabi–Yau geometries are cubic[4] in the entries of the configuration matrix, the fact that linear functions do so well here is a surprise.

5. Analytic Formulæ

In what follows, we will focus on the so-called stringy topological data as originally introduced by Batyrev in the context of reflexive polytopes, Δ, and the associated toric varieties, X_Δ.[7,51,52,e] The main idea is that certain topological invariants, including the Hodge numbers and in particular the Euler number, are independent of the choice of the so-called crepant desingularization of X_Δ.[54,55]

Starting from the Ehrhart series,[f]

$$P_\Delta(t) = \sum_{k\geq 0} l(k\Delta)t^k, \tag{5.1}$$

we use that we can rewrite this as

$$P_\Delta(t) = \frac{\Phi(t)}{(1-t)^{n+1}}, \tag{5.2}$$

where $n = \dim \Delta$. Here, we have introduced the Ehrhart polynomial,

$$\Phi(t) = \sum_{i=0}^{n} \psi_i(\Delta)t^i, \tag{5.3}$$

where $\psi_i(\Delta)$ encode topological information about the toric variety X_Δ. For a reflexive polytope, $\psi_i(\Delta)$ satisfies $\psi_{n-i}(\Delta) = \psi_i(\Delta)$, where we in addition know that $\psi_0(\Delta) = 1$ since $l(0\Delta) = 1$. By expanding the geometric series, we can determine the

[e]Related work on calculating the stringy Chern classes, $c_i(X)$, in terms of the combinatorial data for more general polytopes Δ can be found in Ref. 53.

[f]Ehrhart series and Hilbert series are examined through a machine learning lens in Ref. 56.

$\psi_i(\Delta)$ in terms of the $l(k\Delta)$. In particular, from the above, it then follows that $\psi_1(\Delta) = l(\Delta) - n - 1$ and $\psi_2(\Delta) = l(2\Delta) - (n+1)l(\Delta) + n(n+1)/2$.

Libgober and Wood showed that there exists an identity relating $c_1(X)c_{n-1}(X)$ and the Hodge numbers, $h^{p,q}(X)$ for X a smooth projective variety.[57] This was generalized by Batyrev *et al.* to any toric variety X_Δ associated to a reflexive polytope Δ in terms of the combinatorial data of Δ,[7,51,52]

$$\sum_{i=0}^{n} \psi_i \left(i - \frac{n}{2}\right)^2 = \frac{n}{12} d(\Delta) + \frac{1}{6} \sum_{\substack{\theta \in \Delta \\ \dim\theta = n-2}} d(\theta) d(\theta^*). \tag{5.4}$$

Here, $d(\theta)$ refers to the so-called degree of the face $\theta \in \Delta$ and is given by $d(\theta) = (n-2)!\mathrm{Vol}(\theta)$. In particular, $d(\Delta)$ is the volume of Δ, up to the factor of $n!$. For the case $n = 3$, this leads to the statement that all K3 surfaces have $\chi = 24$:

$$\chi(M) = \sum_{\substack{\theta \in \Delta \\ \dim\theta = 1}} d(\theta) d(\theta^*) = 24. \tag{5.5}$$

In writing this expression, we have used the fact that the Euler number χ for the case of a Calabi–Yau hypersurface M defined by the anticanonical divisor $-K_X$ associated to the anticanonical bundle $\mathcal{O}(-K_X)$ of the toric variety X_Δ is given by[52]

$$\chi(M) = \int_X c_1(X)c_2(X) = \sum_{\substack{\theta \in \Delta \\ \dim\theta = 1}} d(\theta) d(\theta^*), \tag{5.6}$$

generalizing the standard result for the smooth case, cf. Ref. 58.

In $n = 4$ dimensions, the generalized Libgober–Wood identity for X_Δ can be shown to be given by[52]

$$12(l(\Delta) - 1) = 2d(\Delta) + \sum_{\substack{\theta \in \Delta \\ \dim\theta = 2}} d(\theta) d(\theta^*), \tag{5.7}$$

and similarly for Δ^*,

$$12(l(\Delta^*) - 1) = 2d(\Delta^*) + \sum_{\substack{\theta \in \Delta^* \\ \dim\theta = 2}} d(\theta) d(\theta^*). \tag{5.8}$$

Analogously to the $n = 3$ case, we can write

$$\begin{aligned} \chi(M) &= \int_X [c_1(X)c_3(X) - c_1^2(X)c_2(X)] \\ &= \sum_{\substack{\theta \in \Delta \\ \dim\theta = 1}} d(\theta) d(\theta^*) - \sum_{\substack{\theta \in \Delta \\ \dim\theta = 2}} d(\theta) d(\theta^*). \end{aligned} \tag{5.9}$$

Using that $\dim\theta^* = n - 1 - \dim\theta$, it then follows that for a reflexive polytope in $n = 4$ dimensions, we can rewrite the right-hand side of the expression for $\chi(M)$ in terms of the difference of the Libgober–Wood identities for Δ and Δ^*, respectively,

$$\chi(M) = 12(l(\Delta^*) - l(\Delta)) + 2(d(\Delta) - d(\Delta^*)). \tag{5.10}$$

Finally, we can express this in terms of the number of points, $l(2\Delta)$, $l(2\Delta^*)$, of the twice scaled up polytopes 2Δ and $2\Delta^*$, respectively, using that

$$d(\Delta) = 2 + l(2\Delta) - 3l(\Delta), \quad d(\Delta^*) = 2 + l(2\Delta^*) - 3l(\Delta^*), \tag{5.11}$$

and hence,

$$\chi(M) = 2(l(2\Delta) - l(2\Delta^*)) + 18(l(\Delta^*) - l(\Delta)). \tag{5.12}$$

This agrees with the expression (4.5) found by linear regression. As a simple example, let us consider Δ describing the Newton polyhedron associated with the degree five hypersurface M in $X = \mathbb{P}^4$, for which we know $(h^{1,1}, h^{2,1}) = (1, 101)$. Using the Kreuzer–Skarke database one finds that $l(\Delta) = 126$, $l(\Delta^*) = 6$,[10] and furthermore calculates that $l(2\Delta) = 1001$ and $l(2\Delta^*) = 21$. The above expression for the Euler number correctly gives $\chi(M) = -200$.

6. Discussion and Prospectus

Neural networks are universal approximators.[59,60] This means that the output of the first layer of a finite width neural network can approximate any suitably well behaved function on a compact subset of $\mathbb{R}^n$. We have obtained an analytic expression for the Euler number $\chi(M)$ from data about points on the reflexive polytope Δ, its dual and the scaled up versions of these, respectively, where M is given by the anticanonical divisor $-K_X$ of the toric variety X_Δ associated to Δ. Since the data, $l(\Delta)$, $l(\Delta^*)$, $l(2\Delta)$ and $l(2\Delta^*)$, are integer valued, there are many suitable functions. With a simple architecture, the neural network approximates one of these. The better than 96% accuracy further suggests that these results should be analyzed from the perspective of probably approximately correct learning.[61] The lower accuracy of the predictions of $h^{1,1}(M)$, $h^{2,1}(M)$, and the sum $h^{1,1} + h^{2,1}(M)$ indicates that there probably is not a similarly simple formula for these topological invariants based on the minimal data we have provided.

The accuracy for predicting $h^{1,1}$ and $h^{2,1}$ is considerably higher for data along the boundary. This may be explained by the fact that there do exist exact expressions for the individual Hodge numbers for special classes of reflexive polytopes, Δ, such that $h^{2,1} = l(\Delta) - 5$, and similarly for the dual polytopes, Δ^*, with $h^{1,1} = l(\Delta^*) - 5$.[g] These classes of polytopes do indeed occur to a larger degree along the boundary.

It would be interesting to determine what other data we can add to get more precise predictions of the individual Hodge numbers. Clearly, knowing the lattice

[g]The latter case corresponds to the statement that $h^{1,1}(M) = \dim \mathrm{Pic}(X_\Delta)$, the Picard number of the ambient space.

points of the polytope is enough to compute these quantities, but perhaps we can get away with supplying less information in the input. It would also be interesting to test whether quantities like the Kähler cone and the Mori cone can be machine learned.

Calabi–Yau manifolds are an important testbed for applying machine learning to problems in string theory. We have focused our attention in this paper on the Hodge numbers. The success of machine learning methods in this endeavor may point to structural features in the data set taken as a whole. Based on Ref. 62, Reid famously conjectured that the moduli space of Calabi–Yau threefolds is connected through conifold transitions and that any Calabi–Yau geometry may be obtained from the small resolution of degenerations within this family.[63] Perhaps this insight is at the heart of the machine learned relationships. (Similar speculations appear in Ref. 64.)

Finally, we would like to go beyond topology to geometry. First steps in this direction have constructed numerical Ricci flat metrics on Calabi–Yau threefolds via machine learning.[45,65–69] In line with Reid's fantasy, we would as well like to understand geometric transitions from the perspective of machine learning.

Acknowledgments

We thank Damián Mayorga Peña, Challenger Mishra, and in particular Tristan Hübsch for discussions, and Philip Candelas for `Mathematica` code used to analyze reflexive polytopes. PB and BC are supported in part by the Department of Energy grant DE-SC0020220. VJ is supported by the South African Research Chairs Initiative of the Department of Science and Technology and the National Research Foundation and by the Simons Foundation Mathematics and Physical Sciences Targeted Grant, 509116.

ORCID

Per Berglund https://orcid.org/0000-0003-1675-4133
Ben Campbell https://orcid.org/0009-0004-2331-1802
Vishnu Jejjala https://orcid.org/0000-0003-2603-6717

References

1. P. Candelas, G. T. Horowitz, A. Strominger and E. Witten, *Nucl. Phys. B* **258**, 46 (1985).
2. B. R. Greene, K. H. Kirklin, P. J. Miron and G. G. Ross, *Nucl. Phys. B* **278**, 667 (1986).
3. P. Candelas, Lectures on complex manifolds, in *Superstrings and Grand Unification* (World Scientific, 1988), pp. 1–93.
4. T. Hübsch, *Calabi-Yau Manifolds: A Bestiary for Physicists* (World Scientific, 1992).
5. V. Braun, Y.-H. He, B. A. Ovrut and T. Pantev, *Phys. Lett. B* **618**, 252 (2005), arXiv: hep-th/0501070.
6. V. Bouchard and R. Donagi, *Phys. Lett. B* **633**, 783 (2006), arXiv:hep-th/0512149.
7. V. V. Batyrev, *J. Algebr. Geom.* **3**, 493 (1994), arXiv:alg-geom/9310003.
8. M. Kreuzer and H. Skarke, *Adv. Theor. Math. Phys.* **2**, 853 (1998), arXiv:hep-th/9805190.
9. M. Kreuzer and H. Skarke, *Adv. Theor. Math. Phys.* **4**, 1209 (2002), arXiv:hep-th/0002240.

10. M. Kreuzer and H. Skarke, Calabi–Yau data, https://hep.itp.tuwien.ac.at/~kreuzer/CY/.
11. F. Schöller and H. Skarke, *Commun. Math. Phys.* **372**, 657 (2019), arXiv:1808.02422.
12. R. Altman, J. Gray, Y.-H. He, V. Jejjala and B. D. Nelson, *J. High Energy Phys.* **02**, 158 (2015), arXiv:1411.1418.
13. R. Altman, J. Gray, Y.-H. He, V. Jejjala and B. D. Nelson, Toric Calabi–Yau database, https://www.rossealtman.com/toriccy/.
14. M. Demirtas, L. McAllister and A. Rios-Tascon, arXiv:2008.01730.
15. Y.-H. He, V. Jejjala and L. Pontiggia, *Commun. Math. Phys.* **354**, 477 (2017), arXiv:1512.01579.
16. Y.-H. He, *Phys. Lett. B* **774**, 564 (2017), arXiv:1706.02714.
17. D. Krefl and R.-K. Seong, *Phys. Rev. D* **96**, 066014 (2017), arXiv:1706.03346.
18. F. Ruehle, *J. High Energy Phys.* **08**, 038 (2017), arXiv:1706.07024.
19. J. Carifio, J. Halverson, D. Krioukov and B. D. Nelson, *J. High Energy Phys.* **09**, 157 (2017), arXiv:1707.00655.
20. F. Ruehle, *Phys. Rep.* **839**, 1 (2020).
21. P. Candelas, A. M. Dale, C. A. Lutken and R. Schimmrigk, *Nucl. Phys. B* **298**, 493 (1988).
22. K. Bull, Y.-H. He, V. Jejjala and C. Mishra, *Phys. Lett. B* **785**, 65 (2018), arXiv:1806.03121.
23. K. Bull, Y.-H. He, V. Jejjala and C. Mishra, *Phys. Lett. B* **795**, 700 (2019), arXiv:1903.03113.
24. H. Erbin and R. Finotello, arXiv:2007.13379.
25. H. Erbin and R. Finotello, arXiv:2007.15706.
26. R. Altman, J. Carifio, X. Gao and B. Nelson, arXiv:2111.03078.
27. X. Gao and H. Zou, arXiv:2112.04950.
28. J. Gray, A. S. Haupt and A. Lukas, *J. High Energy Phys.* **07**, 070 (2013), arXiv:1303.1832.
29. Y.-H. He and A. Lukas, *Phys. Lett. B* **815**, 136139 (2021), arXiv:2009.02544.
30. H. Erbin, R. Finotello, R. Schneider and M. Tamaazousti, arXiv:2108.02221.
31. M. Kreuzer and H. Skarke, *Comput. Phys. Commun.* **157**, 87 (2004), arXiv:math/0204356.
32. D. S. Berman, Y.-H. He and E. Hirst, arXiv:2112.06350.
33. J. Bao, Y.-H. He, E. Hirst, J. Hofscheier, A. Kasprzyk and S. Majumder, arXiv:2109.09602.
34. D. Klaewer and L. Schlechter, *Phys. Lett. B* **789**, 438 (2019), arXiv:1809.02547.
35. C. R. Brodie, A. Constantin, R. Deen and A. Lukas, *Fortschr. Phys.* **68**, 1900087 (2020), arXiv:1906.08730.
36. C. R. Brodie, A. Constantin, R. Deen and A. Lukas, *Fortschr. Phys.* **68**, 1900086 (2020), arXiv:1906.08769.
37. M. Larfors and R. Schneider, *Fortschr. Phys.* **68**, 2000034 (2020), arXiv:2003.04817.
38. M. C. Hughes, *J. Knot Theory Ramifications* **29**, 2050005 (2020), arXiv:1610.05744.
39. V. Jejjala, A. Kar and O. Parrikar, *Phys. Lett. B* **799**, 135033 (2019), arXiv:1902.05547.
40. S. Gukov, J. Halverson, F. Ruehle and P. Sułkowski, arXiv:2010.16263.
41. J. Craven, V. Jejjala and A. Kar, *J. High Energy Phys.* **06**, 040 (2021), arXiv:2012.03955.
42. A. Davies *et al.*, *Nature* **600**, 70 (2021).
43. J. Craven, M. Hughes, V. Jejjala and A. Kar, arXiv:2112.00016.
44. K. Kodaira, *Am. J. Math.* **86**, 751 (1964).
45. V. Jejjala, D. K. Mayorga Pena and C. Mishra, arXiv:2012.15821.
46. J. Bezanson, A. Edelman, S. Karpinski and V. B. Shah, *SIAM Rev.* **59**, 65 (2017), arXiv:1411.1607.
47. M. Innes *et al.*, arXiv:abs/1811.01457.
48. M. Innes, *J. Open Source Softw.* **3**, 602 (2018).
49. D. P. Kingma and J. Ba, arXiv:1412.6980.
50. W. R. Inc., Mathematica, Version 12.3.1.
51. V. V. Batyrev and D. I. Dais, arXiv:alg-geom/9410001.

52. V. Batyrev and K. Schaller, *Commun. Num. Theor. Phys.* **11**, 1 (2017), arXiv:1607.04135.
53. P. Berglund and T. Hübsch, Work in progress, https://doi.org/10.48550/arXiv.2403.07139..
54. V. V. Batyrev, *Math. Res. Lett.* **7**, 155 (2000), arXiv:alg-geom/9711019.
55. P. Aluffi, *Int. Math. Res. Not.* **2004**, 3367 (2004), arXiv:math/0401167.
56. J. Bao, Y.-H. He, E. Hirst, J. Hofscheier, A. Kasprzyk and S. Majumder, arXiv:2103.13436.
57. A. S. Libgober and J. W. Wood, *J. Differ. Geom.* **32**, 139 (1990).
58. W. Fulton, *Intersection Theory* (Springer, 1998).
59. G. Cybenko, *Math. Control Signals Syst.* **2**, 303 (1989).
60. K. Hornik, *Neural Netw.* **4**, 251 (1991).
61. L. G. Valiant, *Commun. ACM* **27**, 1134 (1984).
62. R. Friedman, *Math. Ann.* **274**, 671 (1986).
63. M. Reid, *Math. Ann.* **278**, 329 (1987).
64. J. Halverson and C. Long, *Fortschr. Phys.* **68**, 2000005 (2020), arXiv:2001.00555.
65. A. Ashmore, Y.-H. He and B. A. Ovrut, *Fortschr. Phys.* **68**, 2000068 (2020), arXiv:1910.08605.
66. L. B. Anderson, M. Gerdes, J. Gray, S. Krippendorf, N. Raghuram and F. Ruehle, arXiv:2012.04656.
67. M. R. Douglas, S. Lakshminarasimhan and Y. Qi, arXiv:2012.04797.
68. A. Ashmore, R. Deen, Y.-H. He and B. A. Ovrut, arXiv:2110.12483.
69. M. Larfors, A. Lukas, F. Ruehle and R. Schneider, arXiv:2111.01436.