Calabi-Yau four-, five-, sixfolds as $\mathbb{P}^n_{\mathbf{w}}$ hypersurfaces: Machine learning, approximation, and generation

Edward Hirst[*] and Tancredi Schettini Gherardini[†]

Centre for Theoretical Physics, Queen Mary University of London, E1 4NS, United Kingdom

(Received 18 January 2024; accepted 21 March 2024; published 2 May 2024)

Calabi-Yau fourfolds may be constructed as hypersurfaces in weighted projective spaces of complex dimension five defined via weight systems of six weights. In this work, neural networks were implemented to learn the Calabi-Yau Hodge numbers from the weight systems, where gradient saliency and symbolic regression then inspired a truncation of the Landau-Ginzburg model formula for the Hodge numbers of any dimensional Calabi-Yau constructed in this way. The approximation always provides a tight lower bound, is shown to be dramatically quicker to compute (with computation times reduced by up to 4 orders of magnitude), and gives remarkably accurate results for systems with large weights. Additionally, complementary datasets of weight systems satisfying the necessary but insufficient conditions for transversality were constructed, including considerations of the interior point, reflexivity, and intra-divisibility properties, overall producing a classification of this weight system landscape, further confirmed with machine learning methods. Using the knowledge of this classification and the properties of the presented approximation, a novel dataset of transverse weight systems consisting of seven weights was generated for a sum of weights ≤ 200, producing a new database of Calabi-Yau fivefolds, with their respective topological properties computed. Furthermore, an equivalent database of candidate Calabi-Yau sixfolds was generated with approximated Hodge numbers.

DOI: 10.1103/PhysRevD.109.106006

I. INTRODUCTION

Calabi-Yau manifolds have been an epicenter for academic breakthroughs since their conception by Calabi [1], some 80 years ago. Amplified by the awarding of a Fields medal for the proof of their existence by Yau [2], their importance within mathematics and to the mathematical community has since been firmly substantiated. However, beyond their interest in mathematics, these geometries have received notable acclaim within the physics community as well. For self-consistency, in superstring theory, the space-time within which we live must be ten-dimensional in nature; to ensure compatibility with the four-dimensional space-time we observe, the remaining six dimensions must form some compact geometry, of which Calabi-Yau manifolds and their orbifolds are the most prudent and popular candidates [3].

A selection of the defining features of Calabi-Yau manifolds are what makes them so appropriate for string compactification. Beyond being compact, their Kähler SU(n) holonomy allows them to support the appropriate fields, as fluxes, which can reduce to those seen in the standard model. Moreover, being Ricci flat in nature, they manifestly satisfy the vacuum Einstein equations desired to incorporate gravity. Under dimensional reduction of a string theory via Calabi-Yau compactification, many properties of the subsequent four-dimensional theory become directly dependent on the used Calabi-Yau geometry, and thus choosing the correct Calabi-Yau becomes paramount to producing a theory that well models the Universe.

Unfortunately, the landscape of these geometries is enormous, and its structure is largely unknown [4]. Through a variety of construction methods, billions of these geometries have so far been enumerated [5,6], and with numbers at this scale brute-force analysis of the corresponding theories becomes computationally infeasible [7]. Databases of this size hence require statistical methods of analysis to extract meaningful insight, and, inspired by a multitude of successes in other fields, academics have been recently experimenting with the application of techniques from machine learning.

Machine learning is a broadly used umbrella term for techniques in computational statistics; loosely separated into three subfields: supervised, unsupervised, and reinforcement learning [8–10]. The first subfield of supervised learning can be considered as advanced techniques in

[*]e.hirst@qmul.ac.uk
[†]t.schettinigherardini@qmul.ac.uk

function fitting, requiring both input and output data to fit. The second unsupervised learning subfield includes more general feature analysis and dimensional reduction, looking at input data on its own. The final subfield is reinforcement learning, which trains an agent to search a space of potential solutions for an optimum.

With use notably initiated for the string community in the simultaneous works [11–14], successes inspired applications in a broader range of contexts. Beyond many excellent programs of work seeking to numerically construct the elusive Ricci-flat Calabi-Yau metrics [15–25] enabling further steps in string phenomenology [26,27], there has been a variety of papers finding real efficacy of machine learning in predicting Calabi-Yau topological properties.

In particular, supervised methods have been especially amenable to the prediction of Hodge numbers, where expensive and difficult computations can be avoided if statistically confident predictions indicate a candidate geometry is highly unlikely to be relevant for one's desired application. This has been shown in the Calabi-Yau "threefold" (i.e., three complex dimensional) construction cases of weighted projective spaces [11,28], complete intersections [11,29–34], their generalized cases [35], and via toric varieties [36,37].

While Calabi-Yau threefolds are excellent candidates for superstring compactification from ten dimensions, superstring theory also has interpretations within its parent theories of M theory and F theory, which are 11 and 12 dimensional, respectively. Therefore, to compactify these higher-dimensional theories down to four dimensions, higher-dimensional geometries are needed. M-theory compactification requires seven-dimensional manifolds [38], notably G2 manifolds exhibiting ever-more elusive constructions; machine learning in this area has been initiated by recent work considering the related G2-structure geometries with success predicting their equivalent Hodge numbers [39]. Alternatively, F-theory compactification requires Calabi-Yau "fourfolds," where machine learning methods have been effective for the complete intersection construction[1] [41,42], for which an exhaustive list has been determined [43,44]; however, machine learning methods have not yet been tested for the other constructions.

While the database of Calabi-Yau fourfolds from weighted projective spaces has been constructed [45–47], the toric variety construction method is too large to be enumerated in full [48], despite new work showing machine learning methods can help search this intractable space [49]. Therefore, inspired by an array of successes in machine learning Calabi-Yau threefolds, this work looks to examine the suitability of these methods to the yet untouched database of Calabi-Yau fourfolds built from weighted projective spaces, while developing on techniques inaugurated in [28].

This paper begins by detailing the Calabi-Yau construction of interest in Sec. II, followed by analysis of the weight system data and respective topological invariants in Sec. III, with detail on generated complementary datasets. In Sec. IV, the machine learning methods used are introduced, followed by their application, results, and interpretation. Central to this work, in Sec. V, is the presentation of an approximation formula for computation of Hodge numbers of Calabi-Yau manifolds constructed via weighted projective spaces, providing a tight lower bound and enormous improvements in computation time. In Sec. VI, this approximation, as well as the other related properties, is used to construct candidate transverse weight systems of seven and eight weights with sums of weights up to 200, along with the topological properties of the subsequent Calabi-Yau five- and sixfolds, respectively. Finally, in Sec. VII, results are summarized and outlook applications discussed.

The code for this work was completed in PYTHON, with use of machine learning libraries SCIKIT-LEARN [50] and TensorFlow [51]; datasets and scripts are made available at this paper's respective repository on GitHub [52].

As a final comment, we leave some references to exciting applications of machine learning across alternative subfields in mathematical physics, which have included work on amoeba [53–55], branes [56–59], conformal theories [60–64], quivers [65–68], phenomenology [69–76], and other related geometry [77–85]. With an abundance of mathematical objects used throughout physics, the age of application of machine learning to uncover new physical understanding is alluring, just at its beginning.

II. BACKGROUND

The most natural appearance of Calabi-Yau fourfolds is in the context of $N = 1$ compactification of F theory to four dimensions, which was studied in the seminal works of [86–88], among others.

F theory emerges upon geometrization of the axiodilaton present in type IIB superstring theory, resulting in a 12-dimensional theory. It was first developed in the seminal work of Vafa [86]. A Calabi-Yau fourfold, which is elliptically fibered, serves as the internal space for the compactification of F theory to an $N = 1$ supersymmetric theory in four dimensions (see [89], for instance). As usual, the moduli space is determined by the possible deformation families, encoded in the cohomological data, which are the main subject of this work. Moreover, Calabi-Yau fourfolds can also appear in the compactification of M theory to three dimensions, leading to an $N = 2$ supersymmetric theory. The two reductions are linked when the fourfold X is elliptically fibered, as shown in [87]. For a given fibration

[1]We note recent successes in generating and machine learning Calabi-Yau fivefolds [40], where the first systematic construction of complete intersection Calabi-Yau fivefolds is presented, producing a large but inexhaustive list of new spaces.

$\mathcal{E} \to X \xrightarrow{\pi} B$, a compactification of M theory on X coincides with a compactification of F theory on $X \times S^1$. The set of Calabi-Yau fourfolds studied in this paper also includes spaces with negative Euler number, a feature that allows for supersymmetry breaking in M-theory compactification.

A. The construction

The Calabi-Yau fourfolds considered in this work are constructed as codimension-one hypersurfaces in compact complex five-dimensional weighted projective spaces $\mathbb{P}^5_{\mathbf{w}}$. A general n-dimensional weighted projective space $\mathbb{P}^n_{\mathbf{w}}$ is defined by considering $\mathbb{C}^{n+1}$, spanned by coordinates $\{z_0, \ldots, z_n\}$, removing the origin to form $\mathbb{C}^{n+1}/\{0\} = (\mathbb{C}/\{0\})^{n+1} = (\mathbb{C}^*)^{n+1}$, then subjecting it to an identification given by

$$(z_0, \ldots, z_n) \sim (\lambda^{w_0} z_0, \ldots, \lambda^{w_n} z_n), \tag{2.1}$$

for all nonzero complex numbers $\lambda \in \mathbb{C}^*$. The integer numbers w_i's are called "weights" (hence the name of the construction), and the vector of weights $(w_0, w_1, \ldots, w_n)$ is known as a "weight system" of $n+1$ weights. For weight systems to uniquely define weighted projective spaces, the set of weights needs to be coprime, removing redundancy introduced by rescaling of the identification parameter λ. There are infinitely many coprime weight systems which thus each uniquely define a weighted projective space. However, not all of these weighted projective spaces will admit Calabi-Yau hypersurfaces; in the cases they do, the weight system is defined as transverse. For transverse weight systems, these Calabi-Yau hypersurfaces are then homogeneous functions of specific degree, as required to satisfy the defining vanishing first Chern class property necessary to produce a Calabi-Yau.

To briefly introduce this, start by assuming we have a transverse weight system such that the hypersurface can avoid the singularities of the ambient $\mathbb{P}^n_{\mathbf{w}}$. This defining hypersurface equation $p = 0$ therefore has no common solutions with its derivative $dp = 0$. Defining $\mathcal{T}_{\mathbb{P}}$ as the tangent bundle of the ambient $\mathbb{P}^n_{\mathbf{w}}$, such that the hypersurface submanifold $\mathcal{M}$ has respective normal bundle $\mathcal{N}$, then $\mathcal{T}_{\mathbb{P}} = \mathcal{T}_{\mathcal{M}} \oplus \mathcal{N}$, allowing the computation of the Chern polynomial for the hypersurface submanifold from that of the ambient space and normal bundle [90]. A tangent space in $\mathcal{T}_{\mathbb{P}}$ is a space of vectors $v = v^i \frac{\partial}{\partial z^i}$ that act on functions in the ambient space (i.e., functions of the ambient space's homogeneous coordinates z^i). The homogeneous nature of the functions, however, leads to an identification in this space of vectors $v^i \sim v^i + w_i z^i$, since $\sum_i w_i z^i \frac{\partial}{\partial z^i} f = mf$ for generic homogeneous function f of degree m, reducing the space dimension by 1 as required. The independence of the vectors (except for this identification) leads to a decomposition of the tangent bundle into line bundles $\mathcal{T}_{\mathbb{P}} = (\mathcal{O}(w_0) \oplus \mathcal{O}(w_1) \oplus \ldots \oplus \mathcal{O}(w_n))/\mathcal{O}$, with the trivial bundle in the denominator. The Chern polynomial of these one-dimensional line bundles is then $c(\mathcal{O}(w_i)) = 1 + w_i\omega$, for ω the Kähler form of the ambient space, leading to

$$c(\mathcal{T}_{\mathbb{P}}) = \Pi_i (1 + w_i \omega). \tag{2.2}$$

Whereas, since the degree $\mathfrak{d}$ hypersurface equation is codimension 1, it can be viewed as a fiber coordinate for $\mathcal{N}$, such that $\mathcal{N} = \mathcal{O}(\mathfrak{d})$, and hence $c(\mathcal{O}(\mathfrak{d})) = 1 + \mathfrak{d}\omega$. Therefore, the overall Chern polynomial for the hypersurface submanifold tangent space is

$$c(\mathcal{T}_{\mathcal{M}}) = \frac{\Pi_i (1 + w_i \omega)}{1 + \mathfrak{d}\omega}, \tag{2.3}$$

where the first Chern class c_1 can be extracted by expansion of the above, leading to the condition $c_1(\mathcal{T}_{\mathcal{M}}) = (\sum_i (w_i) - \mathfrak{d})\mathrm{Tr}(\omega)$, which for the hypersurface to define a Calabi-Yau manifold requires $c_1 = 0$, causing

$$\mathfrak{d} = \sum_i w_i. \tag{2.4}$$

Therefore, Calabi-Yau manifolds can be constructed as hypersurfaces in weighted projective spaces, where the weights form a transverse weight system and the hypersurface is defined by a homogeneous equation of degree equal to the sum of the weight system weights, $w_{\mathrm{tot}} := \sum_i w_i$.

B. Weight system properties

Consequently, for the central focus of this work, which is Calabi-Yau fourfolds, we are interested in weight systems consisting of six weights. For the hypersurface to be Calabi-Yau, the weight system must be transverse, which synonymously in the mathematics literature may also be referred to as quasismooth, in that the hypersurface has no additional singularities other than those inherited from the ambient space. The complete list of transverse weight systems of six weights was classified in [46], totaling 1100055. Transverse weight systems were first bulk generated in [90] for the threefold case of five-weight weight systems, later extended to the full finite list of 7555 in [91,92]. In general, the number of transverse weight systems was proved to be finite for any dimension in [93]. These constructions, as well as those for fourfolds in [46], relied heavily on the use of these hypersurfaces as potentials in Landau-Ginzburg theories [94,95], with limited direct interpretation in terms of the weights. However, in the original construction in [90], a necessary but insufficient condition for transversality was introduced in terms of the weights exclusively.

This necessary but insufficient condition for a general dimensional weight system to be transverse is based on divisibility between these weights. We thus dub this

property "intradivisibility." A weight system is intradivisible if and only if

$$\forall\, w_i\ \exists\, w_j \text{ such that } \frac{w_{\text{tot}} - w_j}{w_i} \in \mathbb{Z}^+, \tag{2.5}$$

such that each weight can be subtracted from the sum of the weights and the result will be divisible by a weight in the weight system.[2] This property can be computed from the weights alone and provides a means of identifying weight systems that are certainly not transverse—where this condition does not hold.

In addition to intradivisibility, another property of a weight system is required for it to be transverse, and this property comes from the more general toric interpretation of the weighted projective spaces.[3] In this interpretation, the ambient $\mathbb{P}^n_{\mathbf{w}}$ are toric varieties, defined by fans in $\mathbb{R}^n$, which can be built from convex lattice polytopes in $\mathbb{Z}^n$ centered on the origin.

A polytope [79] is itself defined by a collection of d hyperplane inequalities such that all points $x \in \mathbb{R}^n$ are in the polytope if $H \cdot x \geq b$ for some defining $d \times n$ matrix H and constant d vector b. The constituent parts of the polytope as defined by the intersection of the hyperplanes are the polytope faces, where 0 faces are vertices, 1 faces are edges, and so on up to $(n-1)$ faces, which are facets. In the case where the vertices' coordinates are all integers across the polytope, the polytope is a lattice polytope.[4] If the polytope contains a single lattice point in its strict interior, then the polytope is called the interior point (IP); due to the affine symmetry of the lattice, this point can always be shifted to be the origin.[5] The respective toric fan for a lattice polytope is defined by constructing 1 cones, which are lines connecting the origin to each vertex, then extending each line infinitely. The remaining higher cones of the fan are then defined by the intersections of the polytope hyperplanes.[6] The toric variety [99] is then constructed from the fan through consideration of its respective dual fan; each cone in the fan has a dual cone that is the set of all points whose inner product with points in the cone produces a non-negative number. The union of all dual cones is the dual fan. Finally, the toric variety is defined as the maximal spectrum of the generators of the dual fan's 1 cones, i.e., taking the dual fan one-cone generators (vectors in $\mathbb{Z}^n$) and treating their entries as exponents of the coordinates in some $\mathbb{C}^n$, each generator providing a condition on the coordinate ring $\mathbb{C}^n$, and the resulting spectrum of maximal ideals of this quotient ring defines the toric variety.

From this construction it has been shown how polytopes lead to toric varieties. Despite the series of steps needed to go from the polytope to the variety, a surprising amount about the variety can be deduced from the polytope information alone. An example of this is that, for the variety to be compact, the dimension of the polytope must equal the dimension of the lattice it is defined on. In fact, in this vein there is a more direct construction method for the toric variety from the polytope and one more similar to the weighted projective space construction. Whereas, where weighted projective spaces are defined through one identification of $\mathbb{C}^{n+1}$ using one weight system as in (2.1), this can be generalized to k identifications of $\mathbb{C}^{n+k}$ using k weight systems. If these weights are selected to be vectors spanning the kernel of the lattice polytope's vertex matrix,[7] then the generated variety is the same toric variety as that constructed via the dual fan method above.

In this way, the weight systems of consideration for weighted projective spaces can be generalized to include combined weight systems (with many weight systems) for toric varieties. The hypersurface equation defining the potential Calabi-Yau in the weighted projective space then becomes a generic hypersurface in the toric variety's anticanonical divisor class [100], with alternative interpretations as nontransverse hypersurfaces in weighted projective spaces [101]. Inverting the process of extracting weights from a polytope, polytopes can also be constructed from (combined) weight systems. However, before introducing this, the definition of a polytope's dual is needed.

In a similar way to how a polytope's fan has a dual fan, a polytope has a dual polytope, defined as the set of points such that the inner product between any point in the polytope and any point in the dual polytope ≥ -1. By definition, the dual of an IP polytope is hence also IP [102]. However, the dual of a lattice polytope is not necessarily lattice, and in the special cases where both a polytope and its dual are lattice, the polytopes are denoted as a reflexive pair—both satisfying the reflexivity property. In fact, it is

[2]We note a nomenclature subtlety in [28] where "transverse" was used to depict a weight system satisfying this property, and "Calabi-Yau" was used to depict a weight system that can admit a Calabi-Yau hypersurface. In this work, we reserve transverse for the weight systems with Calabi-Yau hypersurfaces where the solutions to the hypersurface equation and its derivative are transverse, and we introduce intradivisibility for weight systems satisfying the property of (2.5).

[3]These can be alternatively generalized to fake weighted projective spaces [96], but this is another story.

[4]Lattice polytopes can also be physically interpreted as toric diagrams of quiver gauge theories [97].

[5]In the mathematics literature, lattice polytopes with exclusively the origin in the strict interior are called "Fano," since they lead to Fano varieties. Where the boundary lattice points are only the polytope's vertices, the polytope is terminal Fano, while where there are extra boundary lattice points, the polytope is canonical Fano [98].

[6]To ensure a smooth toric variety, the polytope can first be triangulated to resolve the singularities arising from the interior points of the facets.

[7]We note that the $GL(k,\mathbb{Z})$ symmetry in the kernel basis manifests itself as the redundancy from the linear addition of weight systems before identification. Additionally, on more general lattices there are further identifications from the lattice gradings to account for where the vertices are not primitive.

one of the astounding beauties of the toric construction of Calabi-Yau that the hypersurfaces in toric varieties from dual lattice polytopes are in fact mirror symmetric [100]. A weight system is thus "reflexive" if the lattice polytope constructed from it is reflexive.

Let us return to the construction of an IP polytope from an IP weight system. Here, the hyperplane equations defining the polytope include an equality for each weight system such that $\sum_i w_i x_i = w_{\text{tot}}$, and inequalities defined by $x_i \geq 0 \ \forall i$ [103]. Through this construction, the point $x_i = 1 \ \forall i$ is manifestly contained within the polytope, since it naturally satisfies the equalities and inequalities, noting that an affine transformation can set this point to be the origin. This general polytope is hence always IP, however, it may be rational and we are often more interested in its restriction to a lattice.

To be able to then define and check the IP property of a given weight system, it suffices to construct the respective polytope and consider it as existing on the crudest lattice it can (that generated by the polytope's vertices). The dual polytope can then be generated from this and, respectively, the dual lattice (all real points that dot product with all points in the polytope's lattice to integers), however, the vertices of the dual polytope may not lie on the dual lattice. Thus, a restriction is required by taking the convex hull of dual lattice points that lie within/on this dual polytope. This restriction may slice parts of the dual polytope off, producing a smaller dual polytope, which when taking the dual again will produce an new version of the original polytope, which we define to be the integer polytope of interest, and in doing this the new boundaries may intersect the origin. Therefore, the new restricted polytope may no longer contain the origin in its interior and would thus not be IP [48]. In the cases where the origin does remain in the strict interior, the respective lattice polytope is IP, and we define the weight system to be IP too.

All real polytopes constructed from weight systems are simplexes, since there are as many intersections of the single defining equality with the inequalities as there are lattice dimensions (and also weights); equivalently, those from the larger combined weight systems are the union of simplexes. However, the restriction to the relevant lattice as described above may generalize the polytope, causing the lattice polytopes to be unions of simplexes also. For the weight system to be transverse, there must be no further unavoidable singularities than the origin, and this translates to having no interior points on the polytope facets. It is where this occurs that weight systems can be IP but not transverse.

The importance of the IP property for weight systems comes from [104], where it was shown that any transverse weight system is by *necessity* IP for any size weight system. However, the converse is not true, and thus overall we have two independent necessary but insufficient conditions for a weight system to be transverse: intradivisibility and IP. Beyond this, we have another weight system property: reflexivity; where its interrelation with transversality depends on the construction dimension in question [45,103]. Denoting the sets of IP, reflexive, and transverse weight systems of n weights by $IP(n)$, $R(n)$, and $T(n)$, respectively, with their respective sizes as $|IP(n)|$, $|R(n)|$, and $|T(n)|$, the relations between, and frequencies of, weight systems with each property are shown in Table I.

Because of the need for an identification, weight systems are not defined for one weight, and since the transverse property requires taking a codimension-one hypersurface, this property is also not defined for weight systems of two weights. For two weights, the single weight system is (1,1), which is equivalent to the single one-dimensional IP and reflexive polytope with vertices a distance 1 either side of the origin. For three weights, there are three IP weight systems $\{(1,1,1),(1,1,2),(1,2,3)\}$, which are all both reflexive and transverse, corresponding to three of the five reflexive triangles. Stepping to four weights, the number grows to 95 [104], whereas for five weights there is no longer an equality between all these sets of weights, with set sizes computed in [91,92,104]. The weight systems of central focus in this work have six weights, and it is at this stage that each set of weight systems becomes distinct, where the $IP(6)$ and $R(6)$ sets were computed in [48], and the $T(6)$ set was addressed in [46]. Beyond systems with six weights, the set sizes are unknown, as constructions

TABLE I. Known relations between the sets of weight systems satisfying the IP $[IP(n)]$, reflexive $[R(n)]$, and transverse $[T(n)]$ properties as the number of weights in the weight system increases. Additionally are shown the sizes of the sets of weight systems satisfying those properties (denoted with $|\cdot|$) at each weight system size. The weight systems for dimensions ≥ 7 have not been fully computed yet. "?" indicate that those numbers have not been computed yet.

		Property		
Number of weights	Relations	$\lvert IP(n)\rvert$	$\lvert R(n)\rvert$	$\lvert T(n)\rvert$
2	$IP(2) = R(2)$	1	1	⋯
3	$IP(3) = R(3) = T(3)$	3	3	3
4	$IP(4) = R(4) = T(4)$	95	95	95
5	$IP(5) = R(5) \supset T(5)$	184026	184026	7555
6	$IP(6) \supset R(6)$ $IP(6) \supset T(6)$	322383760930	185269499015	1100055
≥ 7	$IP(\geq 7) \supset R(\geq 7)$ $IP(\geq 7) \supset T(\geq 7)$	?	?	?

have not yet been attempted (until this work, as detailed in Sec. VI). The intradivisibility property is not believed to have a finiteness bound, which is why it is not included in these count considerations. The interrelation of this property with the others is discussed in more detail in Sec. III B.

C. Topological properties

As previously mentioned, the cohomological data of the Calabi-Yau used in compactification determines the moduli space of the resulting compactified supersymmetric theory. Since Calabi-Yau manifolds are manifestly complex and Kähler [102], the complexity allows use of the decomposition of the complexified cotangent bundle into holomorphic and antiholomorphic parts via eigenspaces of the complex structure: $\mathcal{T}^*_{\mathcal{M}} = \mathcal{T}^{*(1,0)}_{\mathcal{M}} \oplus \mathcal{T}^{*(0,1)}_{\mathcal{M}}$. This, in turn, causes a decomposition of the differential forms as exterior products of these holomorphic and antiholomorphic cotangent bundles $\Lambda^k \mathcal{T}^*_{\mathcal{M}} = \bigoplus_{p+q=k} \Lambda^p \mathcal{T}^{*(1,0)}_{\mathcal{M}} \otimes \Lambda^q \mathcal{T}^{*(0,1)}_{\mathcal{M}}$, such that the (p,q)-forms are sections of each sum component with vector space $\Omega^{p,q}(\mathcal{M})$. From here, the cohomology arises using the decomposition of the exterior derivative operator $\mathrm{d} = \partial + \bar{\partial}$, allowing definition of the Dolbeault cohomology groups

$$H^{p,q}_{\bar{\partial}}(\mathcal{M}) = \frac{\mathrm{Ker}(\bar{\partial}: \Omega^{p,q}(\mathcal{M}) \mapsto \Omega^{p,q+1}(\mathcal{M}))}{\mathrm{Im}(\bar{\partial}: \Omega^{p,q-1}(\mathcal{M}) \mapsto \Omega^{p,q}(\mathcal{M}))}. \tag{2.6}$$

These are defined for all p (arbitrarily they may instead be defined for all q using ∂), and the dimension of these groups defines the "Hodge numbers,"

$$h^{p,q} = \dim H^{p,q}_{\bar{\partial}}(\mathcal{M}). \tag{2.7}$$

These Hodge numbers may be arranged into a Hodge diamond, where the symmetries of complex conjugation ($h^{p,q} = h^{q,p}$) and Serre duality ($h^{p,q} = h^{n-p,n-q}$ for $\dim_{\mathbb{C}} \mathcal{M} = n$) become clear. The Kählerity of the Calabi-Yau manifolds relates these Hodge numbers to the complexified de Rham real cohomological Betti numbers b_k, since

$$\begin{aligned} H^k_{\mathrm{d}}(\mathcal{M}) &= \bigoplus_{p+q=k} H^{p,q}_{\bar{\partial}}(\mathcal{M}), \\ \Rightarrow b_k &= \sum_{p+q=k} h^{p,q}, \end{aligned} \tag{2.8}$$

then also allowing for the manifolds' Euler number χ to be computed from these Hodge numbers via $\chi = \sum_k (-1)^k b_k$.

Furthermore, for the specialized Kähler case of Calabi-Yau manifolds, there are even further restrictions on these Hodge numbers. One of the defining properties of a Calabi-Yau manifold is a unique holomorphic top form, which then sets $h^{n,0} = 1$, hence also setting to 1 the other Hodge diamond corners (via the conjugation and duality) [105]. Additionally, as the Calabi-Yau's are simply connected, they have trivial first fundamental group and therefore also trivial first homology group, setting $h^{1,0} = 0$ and, respectively, the remaining boundary components of the Hodge diamond [106]. The final Hodge diamond therefore takes the form

$$\begin{array}{ccccccccccc} & & & & 1 & & & & & & \\ & & & 0 & & 0 & & & & & \\ & & 0 & & h^{1,1} & & 0 & & & & \\ & 0 & & h^{1,2} & & h^{1,2} & & 0 & & & \\ \iddots & & \iddots & & & & \ddots & & \ddots & & \\ 1 & & \cdots & & \cdots & & \cdots & & 1 & & \\ \ddots & & \ddots & & & & \iddots & & \iddots & & \\ & 0 & & h^{1,2} & & h^{1,2} & & 0 & & & \\ & & 0 & & h^{1,1} & & 0 & & & & \\ & & & 0 & & 0 & & & & & \\ & & & & 1 & & & & & & \end{array} \tag{2.9}$$

showing that there remain few nontrivial components for consideration in the subsequent string compactification. For Calabi-Yau fourfolds, the nontrivial Hodge numbers are $\{h^{1,1}, h^{1,2}, h^{1,3}, h^{2,2}\}$. It is noted that in this dimension there exists a further constraint on the Hodge numbers [107], which reads

$$-4h^{1,1} + 2h^{1,2} - 4h^{1,3} + h^{2,2} = 44, \tag{2.10}$$

allowing $h^{2,2}$ to be eliminated from the above list.

These Hodge numbers, as well as the Euler number, are of particular interest to physicists, and work characterizing and classifying Calabi-Yau's beyond these topological properties has seen insightful early progress [108,109]. For the weighted projective space construction of Calabi-Yau manifolds, there are direct formulas for these topological properties from the weights alone [110,111]. Specifically, these are

$$\begin{aligned} \chi &= \frac{1}{w_{\mathrm{tot}}} \sum_{l,r=0}^{w_{\mathrm{tot}}-1} \left[\prod_{i | l q_i \& r q_i \in \mathbb{Z}} \left(1 - \frac{1}{q_i}\right) \right], \\ Q(u,v) &= \frac{1}{uv} \sum_{l=0}^{w_{\mathrm{tot}}} \left[\prod_{\tilde{\theta}_i(l) \in \mathbb{Z}} \frac{(uv)^{q_i} - uv}{1 - (uv)^{q_i}} \right]_{\mathrm{int}} \\ &\quad \times \left(v^{\mathrm{size}(l)} \left(\frac{u}{v}\right)^{\mathrm{age}(l)} \right), \end{aligned} \tag{2.11}$$

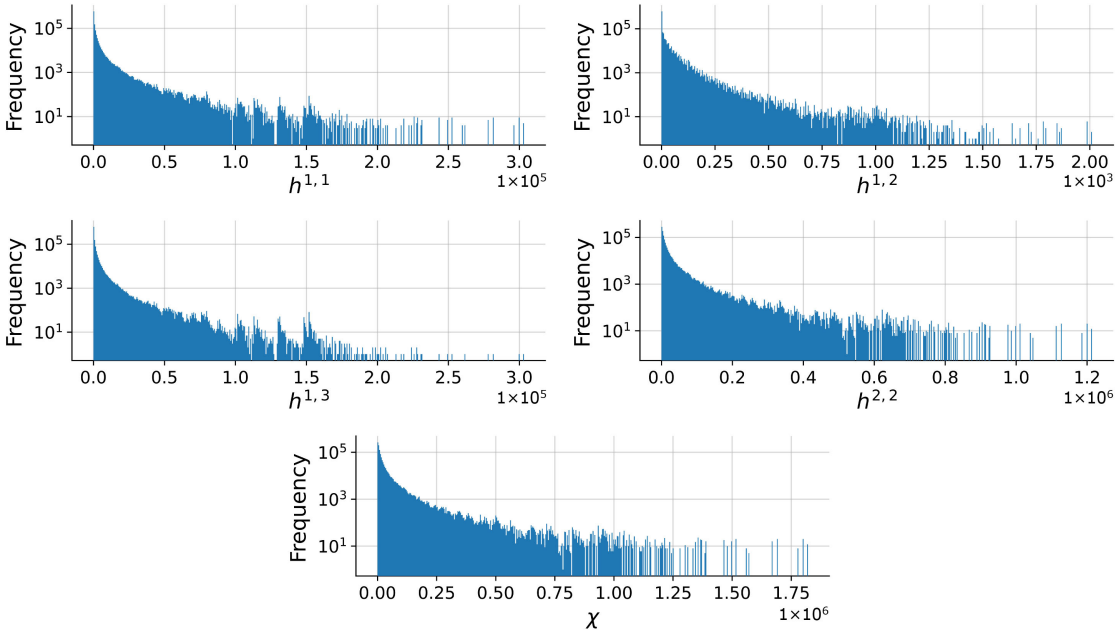


FIG. 1. These plots illustrate the distribution of each of the invariants for Calabi-Yau fourfolds in weighted projective spaces. Each bin contains 500 samples.

for weights w_i, normalized weights $q_i = w_i/w_{\text{tot}}$, and u, v as dummy variables of the Poincaré polynomial $Q(u,v) \coloneqq \sum_{p,q} h^{p,q} u^p v^q$. For $Q(u,v)$, $\tilde{\theta}_i(l)$ is the canonical representative of lq_i in $(\mathbb{R}/\mathbb{Z})^5$, $\text{age}(l) = \sum_{i=0}^{4} \tilde{\theta}_i(l)$, and $\text{size}(l) = \text{age}(l) + \text{age}(w_{\text{tot}} - l)$. Note also for χ, where $\forall\ ilq_i$ or $rq_i \notin \mathbb{Z}$ then the product takes value 1. These components are reintroduced and explained in more detail in Sec. V.

These formulas, as can be seen from Eq. (2.11), are especially complicated, requiring factorially many integer divisibility checks as the weights in the weight system increase in value, as well as numerous extremely expensive polynomial divisions. It is with this in mind that this work is motivated to investigate the efficacy of machine learning methods at approximating these formulas in Sec. IV, with the aim of distilling physical insight to form a suitable approximation, as discussed in Sec. V, focusing on the more demanding Poincaré polynomial formula for Hodge numbers—from which the Euler number can be computed.

III. DATA ANALYSIS

In this section, the database of transverse six-vector weight systems, used to construct Calabi-Yau (CY) fourfolds via $\mathbb{P}^5_{\mathbf{w}}$ spaces, is analyzed from a general data science perspective. Databases of weight systems satisfying different combinations of the considered properties for transversality are then generated and discussed, with further data analysis.

A. The fourfolds dataset

Here, the global properties of the primary dataset under investigation are summarized. It was first presented in [46], where some patterns in the Hodge numbers arrangement were discussed and illustrated by scattered plots, with further preliminary plots available in [112]. Our work, on the other hand, is a natural extension of the investigations on the analogous manifolds in three complex dimensions, performed in [28]. As such, we focus on the features that are most relevant for machine learning purposes, and we start from the distribution of the invariants, which is shown in Fig. 1. We observe that, by using the logarithmic scale for the frequency, all histograms display a similar behavior. The majority of samples is always concentrated around low values, and the ranges span several order of magnitudes. The key features of the distributions in Fig. 1 can be summarized as

$$\langle h^{1,1} \rangle = 2933.8^{303148}_{1}, \qquad \langle h^{1,2} \rangle = 24.1^{2010}_{0}, \qquad \langle h^{1,3} \rangle = 2300.3^{303148}_{1},$$
$$\langle h^{2,2} \rangle = 20932.0^{1213644}_{82}, \qquad \langle \chi \rangle = 31307.5^{1820448}_{-252}, \tag{3.1}$$

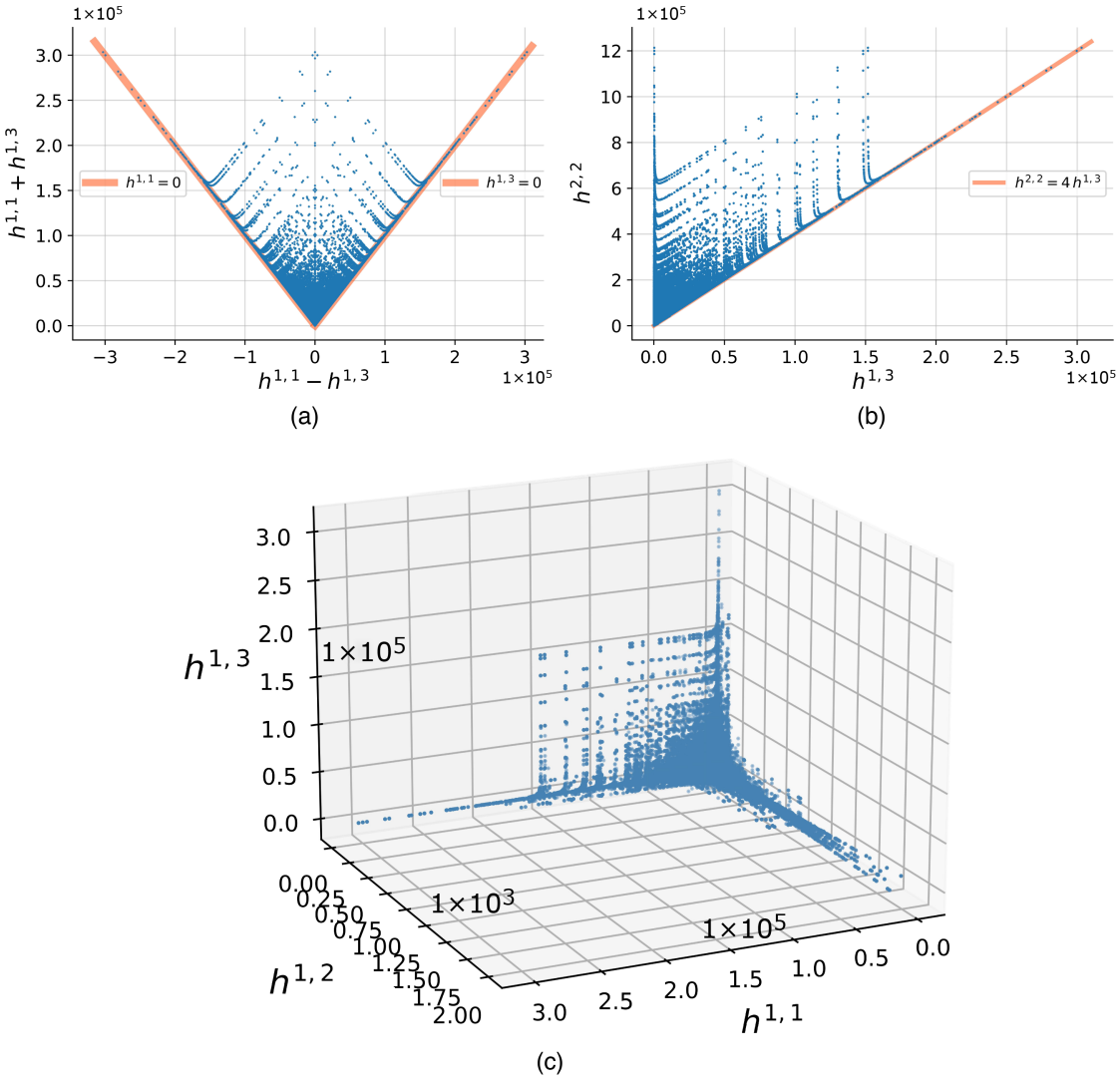


FIG. 2. Scattered plots of the Hodge numbers of Calabi-Yau fourfolds in weighted projective spaces. (a) Illustrates that the spaces are mirror symmetric to a high degree. (b) Shows the relation between the two highest Hodge numbers, also compared with the constraint (2.10). Finally, the 3D plot (c) illustrates the relation between the three independent Hodge numbers.

where we borrow the notation $\text{mean}_{\text{min}}^{\text{max}}$ from [44]. The same range and similar mean values of $h^{1,1}$ and $h^{1,3}$ are a hint of mirror symmetry, which is indeed present in this dataset, as noted in [46]; Fig. 2(a) provides an illustration of it. This plot should be compared with the famous threefold version, where $h^{1,1} + h^{1,2}$ is plotted against the Euler number χ [90]. Quantitatively, the degree of mirror symmetry is around 70%, as reported in [46]. This feature was discovered in generating the set of Calabi-Yau's constructed as hypersurfaces in weighted projective spaces, derived from their embedding within toric varieties and notably does not apply to the complete intersection Calabi-Yau's (CICY) construction [44,113].

Figure 2(b) shows the relation between the two highest Hodge numbers (note that, due to mirror symmetry, this would look almost identical if we were to plot $h^{1,1}$ instead of $h^{1,3}$). The orange line corresponds to $h^{1,1}, h^{1,2} \ll h^{1,3}$, as can be seen from the relation (2.10) and a good amount of data clusters along this line. This feature was noted in [44], where they analyzed the less symmetric set of complete intersection Calabi-Yau fourfold Hodge numbers and found that the data only showed the linear behavior depicted in the plot (in orange). The distribution of the nontrivial Hodge numbers $h^{1,\bullet}$ is shown in Fig. 2(c). By virtue of (2.10), this contains all of the cohomological information of the manifold. As we expect, the $h^{1,1} - h^{1,3}$ plane at $h^{1,2} \approx 0$ displays the mirror-symmetric behavior shown in Fig. 2(a) (with a 45° rotation).

Another interesting feature of this dataset is that, analogous to what was observed in [28], an evident linear forking behavior in the plot of $h^{1,1}$ vs highest weight of the system can be observed. It is shown in Fig. 3, where the dataset was also partitioned into reflexive and nonreflexive weight systems. This partitioning is discussed in more

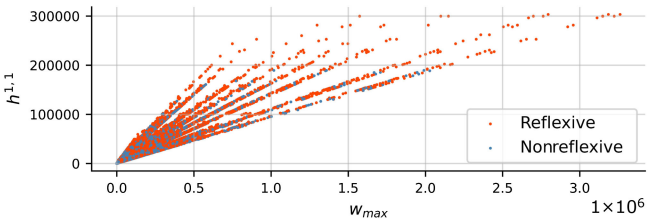


FIG. 3. This plot of $h^{1,1}$ as a function of the highest weight $w_{\max}$ shows that the linear clustering observed in [28] for Calabi-Yau threefolds in weighted projective spaces is also manifest in the fourfolds dataset. Both reflexive and nonreflexive systems display the same behavior, although the regime $h^{1,1} > 200000$ is dominated by reflexive ones. This is confirmed by the principal component analysis shown in Fig. 6.

detail and put in a broader context in Sec. III B. For now, we just note that, for highest weights larger than $\sim 5 \times 10^4$, the $h^{1,1}$ values fall neatly into linear clusters. Motivated by the findings presented in [28], we explore this behavior of the dataset at hand with similar techniques. As it is evident from Fig. 3, for large weights, there are eight peaks in the $h^{1,1}/w_{\max}$ distribution (where the largest weight is the final weight in the weight system, such that $w_{\max} = w_5$). These are linear clusters in the $h^{1,1}$ vs $w_{\max}$ plane, as shown in Fig. 4(a). To neatly illustrate the clusters, we only considered systems with largest weights $w_{\max} \gtrapprox 3 \times 10^5$, which can be seen from Fig. 4(b). The gray lines are the clusters obtained via the K-Means algorithm as used in [28], which is now described. The statistical confidence of a clustering behavior can be quantified by the inertia measure. Running the K-Means algorithm on an input dataset with a prespecified number of clusters, the cluster centers/means $\mu_{\mathcal{C}}$ are randomly initialized and the data points are allocated to the clusters to which they are closest. In each cluster, the center is then updated to the mean of the data points allocated to it, from which all data points are then reallocated to the clusters they are closest to with respect to these new means. This process is iterated until convergence. Given a final set of clusters $\mathcal{C}$, with associated means $\mu_{\mathcal{C}}$, across the clustered dataset, which here is on inputs $r_i = h^{1,1}/w_{\max}$, then the inertia is defined as

$$\mathcal{I} = \sum_{\mathcal{C}} \sum_{r_i \in \mathcal{C}} (\mu_{\mathcal{C}} - r_i)^2. \tag{3.2}$$

We are implicitly assuming that any r_i belongs to the cluster whose mean to which it is closest. The number of clusters for the problem at hand was found to be eight, deduced by eye from Fig. 4. Furthermore, two normalized versions of (3.2) may also be introduced, which have a nice statistical interpretation,

$$\hat{\mathcal{I}} = \frac{\mathcal{I}}{n_{\text{samples}}} \quad \text{and} \quad \hat{\hat{\mathcal{I}}} = \frac{\hat{\mathcal{I}}}{\max(r_i) - \min(r_i)}. \tag{3.3}$$

They are normalized with respect to the number of samples and with respect to both the number of samples and the range of the samples, respectively. For the clustering analysis of the $h^{1,1}, w_{\max}$ data reported in Fig. 4, we find

$$\mathcal{I} = 0.050, \qquad \hat{\mathcal{I}} = 9.3 \times 10^{-6}, \qquad \hat{\hat{\mathcal{I}}} = 3.8 \times 10^{-5}. \tag{3.4}$$

In words, these value show that, on average, the ratios $h^{1,1}/w_{\max}$ that we considered are 0.0038% of the total range from their nearest cluster [for range shown in Fig. 4(a) to be ≈ 0.25]. These results strongly corroborate the liner clustering behavior observed.

B. Additional weight datasets

The exact conditions for transversality of a weight system are derived from the use of the transverse polynomials as potentials of Landau-Ginzburg string vacua [91]. These conditions arise from the necessity for the central charge of these theories to be nine and a subtle application of Bertini's theorem allowing deformation of polynomials to reduce the singularity structure to exclusively an isolated singularity.

The direct combinatoric interpretation of these conditions in terms of exclusively the weights is unclear, and as demonstrated in [90] a first step toward a complete list of necessary and sufficient conditions is provided by the property we dub intradivisibility. Because of the necessary but insufficient nature of this property, while all transverse weight systems will satisfy it, there are many examples of weight systems satisfying it that induce further singularities on their subsequent hypersurfaces, preventing them from

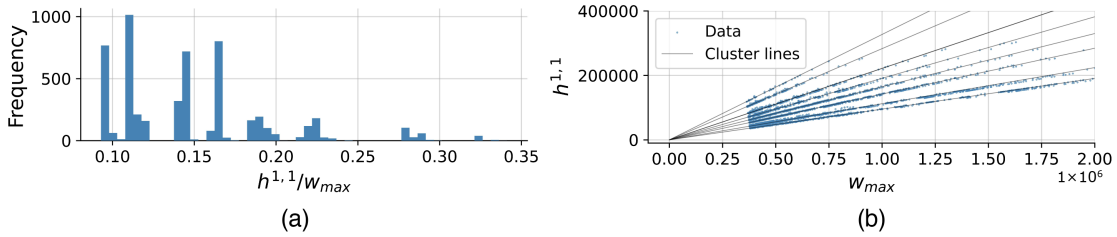


FIG. 4. These plots focus on the clustering observed in Fig. 3 for large weights and show that the clustering analysis correctly reproduces the multilinear behavior. The clusters are shown as peaks in the $h^{1,1}/w_{\max}$ histogram in (a) and as lines in the $h^{1,1}$ vs $w_{\max}$ plane in (b).

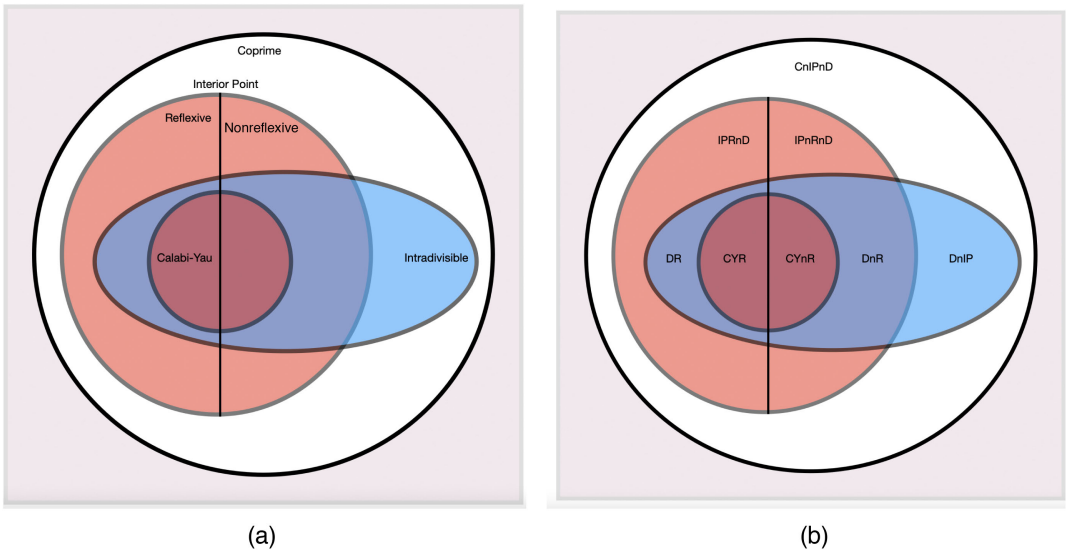


FIG. 5. Venn diagrams displaying (a) the conditional dependencies of the considered six-weight weight system properties and (b) the partition of the weight system data into nonoverlapping subdatasets.

being Calabi-Yau in nature and hence the weight system transverse.

As well as the intradivisibility property, via works in [100,114], there is a further necessary but insufficient property required for a weight system to exhibit a Calabi-Yau hypersurface. This property comes from the interpretation of a weight system as a lattice polytope and, respectively, the weighted projective space as a compact toric variety. As described in Sec. II, the respective lattice polytope must hence have a single interior point (denoted as the IP property) for the subsequent toric variety to exhibit a Calabi-Yau hypersurface. Additionally, at this dimensionality ($n > 4$) the IP polytope no longer needs to be reflexive to exhibit a Calabi-Yau hypersurface, relaxing the necessity for this condition that is essential for the Calabi-Yau construction in lower dimensions.

As demonstrated, for higher-dimensional Calabi-Yau constructions, the relative importance of the previous essential properties becomes less clear, as well as their interrelations. Therefore, to graphically represent the dependencies of these properties, a Venn diagram is presented in Fig. 5(a). This in essence classifies the relevant ambient weighted projective spaces, which are defined uniquely by coprime weight systems.[8] Principally, for a six-vector weight system to be transverse and hence exhibit a Calabi-Yau fourfold hypersurface, it must be *both* IP and intradivisible.[9]

It is therefore interesting to probe this relative importance among the necessary conditions using equivalent datasets of weight systems satisfying different combinations of these properties. Specifically, a dataset of weight systems is constructed for *every* combination of these properties. The partition of these weight systems is described by starting with coprime weight systems satisfying neither IP nor intradivisibility (CnIPnD); then weight systems satisfying either intradivisibility (DnIP), or IP, which can then be nonreflexive (IPnRnD) or reflexive (IPRnD); then weight systems satisfying both intradivisibility and IP, but still not transverse, hence still split into nonreflexive (DnR) and reflexive (DR); and then, finally, the transverse weight systems exhibiting Calabi-Yau hypersurfaces, which are again either nonreflexive (CYnR) or reflexive (CYR). This becomes a full partition of weight systems into subdatasets with respect to these properties, and these subdataset labels are chosen as acronyms to reflect the satisfied properties of coprime $\mapsto$ C, IP $\mapsto$ IP, intradivisible $\mapsto$ D, reflexive $\mapsto$ R, and where relevant the absence of a property is denoted with an "n" before the respective property notation (i.e., nonreflexive $\mapsto$ nR). This partition is represented on the property interrelation Venn diagram in Fig. 5(b), spanning all unique parts of it.

In generating these datasets, first the database of transverse weight systems of [46] was partitioned into reflexive and nonreflexive to produce the CYR and CYnR subdatasets, respectively. Then the publicly available sample of 10^6 five-dimensional six-vector IP weight systems [48] was partitioned into reflexive and nonreflexive as well as intradivisible and nonintradivisible to initiate the respective DR, DnR, IPRnD, IPnRnD subdatasets, explicitly ensuring no overlap with the CYR and CYnR datasets. New to this work, we generate all intradivisible six-vector weight

[8]Any common factor can be removed by redefinition of the identification parameter λ, making the coprime case the natural unique representative of each weighted projective space.

[9]We tested that, as expected, weight systems without the Calabi-Yau property are incompatible with the formula obtained via the Landau-Ginzburg model, described in Sec. V. Specifically, the polynomial divisions involved in such an expression are not well defined, i.e., the result contains a reminder.

TABLE II. The sizes of the subdatasets of weight systems of six weights in each part of the partition, along with the means and ranges across all weight values in each subdataset.

Subdataset	CnIPnD	DnIP	IPnRnD	IPRnD	DnR	DR	CYnR	CYR
Size	408124	9614	999975	988436	172462	81215	847122	252933
$\text{mean}^{\text{max}}_{\text{min}}$	198^{3294}_{1}	39^{198}_{1}	323^{7875}_{1}	59^{858}_{1}	42^{200}_{1}	40^{200}_{1}	$3969^{2028138}_{1}$	$9080^{3260733}_{1}$

systems with a sum of weights with a maximum value of 400 (using functionality in this paper's GitHub), partitioning off those that are not in the CY property subdatasets and then checking the IP and reflexivity properties (using PALP functionality [115]) to supplement the DR, DnR, and DnIP subdatasets. Finally, coprime weight systems were generated stochastically[10] and checked for intradivisibility and IP. These coprime weight systems were then partitioned into IPRnD, IPnRnD, and CnIPnD subdatasets, omitting any that were intradivisible to keep the sum of weights maximum value fixed for the DR, DnR, and DnIP datasets. These datasets were then combined with the above and any repetitions of weight systems removed. This substantially increased the subdataset sizes, producing a final partition with class sizes as shown in Table II.

As can be seen in Table II, the subdatasets are not balanced in size. In some cases this is particularly natural, where the CY subdatasets are exhaustive in their partition between CYnR and CYR, including the entire finite list of possibilities. Moreover, there are finitely many IP weight systems that can be split among the appropriate properties, of which only a sample is publicly available and which we supplement with statistical searches. Conversely, the intradivisibility property is not expected to enforce finiteness on the dataset of satisfying weight systems. Therefore, this set has not been generated exhaustively in previous work and is completed exhaustively here for a sum of weights up to 400.[11] These class sizes are hence well motivated from a viewpoint of exhaustive consideration and analysis, as well as due to computational limitations. Conversely, there are infinitely many coprime weight systems satisfying neither intradivisibility nor IP, which we hence sample stochastically until a suitable order of magnitude matching the other class sizes was achieved.

The difference in subdataset sizes provides concrete stochastic information about the overlap of these weight system properties, and one could then crudely infer probabilities of a generic coprime weight system satisfying each property combination using these dataset sizes. The later machine learning architectures implemented have generic adaptability to accommodate variable class sizes, as described in Sec. IV B, and appropriate performance measures are used to avoid bias misinterpretations of learning.

C. Principal component analysis

Linear behavior in distributions can be analyzed through principal component analysis (PCA). This unsupervised machine learning technique extracts an orthonormal basis for the dataset in question, with basis vectors ranked according to their degree of contribution toward the variance in the data's distribution. The basis is computed as eigenvectors of the dataset's covariance matrix, where the symmetric nature of the matrix ensures real eigenvalues that can be ordered decreasingly and then used to rank the basis. The normalized eigenvalues are named the explained variance and provide a measure of relative importance of each eigenvector (the larger the explained variance, the more important the respective eigenvector). For a prespecified desired degree of representation, a dataset can be projected onto the first i eigenvectors in the ranked basis such that the sum of the respective first i normalized eigenvalues exceeds the desired proportion of representation. In this sense, PCA is often used as a dimensionality reduction technique.

In this work, the union of all subdatasets of six-vector weight systems was analyzed with PCA, as one large dataset, to probe the capacity of linear structure being used for simple classification between the subdatasets of the partition. In this PCA, the explained variances were

$$(0.999999498, 0.000885369, 0.000405971, 0.000233533, 0.000010095, 0.000001597),$$

demonstrating a clear dominance in the first principal component. Because of the nature of representation of the weight systems, where the entries are sorted in increasing size, it is expected that the latter parts of the vector will dominate the most significant principal components.[12]

[10] In a similar vein to [28], an exponential distribution was fitted to the Calabi-Yau weight systems and used to generate trial weight systems, which were checked to be coprime.

[11] The sum of the weight limit was selected as the limit of computation reached at high power computer timeout of 240 core hours.

[12] Within the method of PCA it is often typical to center and scale the data components prior to analysis. Centering has no physical effect on the features since the covariance is relative to the mean, and so it is not implemented here; while scaling is typically important where each component is a different measure with different units—not applicable here where there are no units and the relative sizes of the weights are inherently important to the weight system definition.

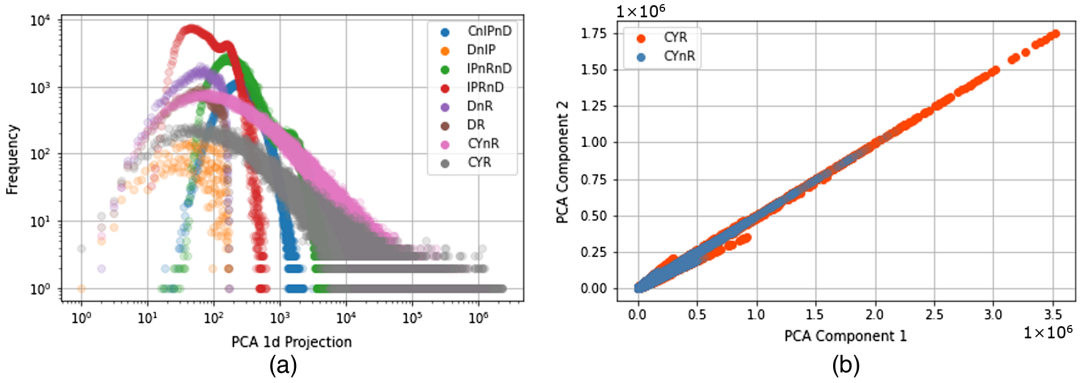


FIG. 6. PCA of the classification partition of weight system data; explained variance demonstrated a single dominant principal component. (a) The frequency distribution for the one-dimensional projections of the weight systems are shown for each part of the partition (on a log-log scale). Equivalently, (b) shows a two-dimensional projection of the Calabi-Yau data, corroborating the forking behavior observed.

This is the case, as shown by the components of this eigenvector for the first principal component,

$$(0.000220159, -0.002178435, 0.007423314, \\ -0.006743665, -0.342499831, 0.939461808).$$

However, the final two components are still of the same magnitude such that the projection is not trivial. The dominance of the first principal component motivates a one-dimensional projection of the data, using the above eigenvector. The $\text{mean}_{\text{min}}^{\text{max}}$ values for the one-dimensional projections of each subdataset were

$$\begin{aligned} \langle \text{CnIPnD} \rangle &= 356_{17}^{3041}, & \langle \text{DnIP} \rangle &= 68_{1}^{167}, \\ \langle \text{IPnRnD} \rangle &= 592_{16}^{6313}, & \langle \text{IPRnD} \rangle &= 122_{11}^{614}, \\ \langle \text{DnR} \rangle &= 78_{2}^{171}, & \langle \text{DR} \rangle &= 75_{2}^{169}, \\ \langle \text{CYnR} \rangle &= 8318_{2}^{1439056}, & \langle \text{CYR} \rangle &= 19116_{1}^{2313640}, \end{aligned} \tag{3.5}$$

given to the nearest integer. They display similar lower bounds throughout, while higher mean and maximum values for nonreflexive subdatasets, and substantially larger ranges for the CY data.

To explore the distributions of these projections, their nearest integer values of the one-dimensional projections for each subdataset in the partition were plotted according to their frequencies of occurrence in the histogram of Fig. 6(a).

Plotted according to a log-log scale, the distributions show a surprising approximate continuity of the lines. The projection distributions all experience significant overlap in the values they can take, and the overlap of the distribution lines shows that each subdataset experiences regions of the data space where the constituent vectors are distributed similar to another subdataset (since frequencies are the same in that range of the projections), making classification difficult in each case of comparison.

Interestingly, the distributions of the CY subdataset projection interpolates between the IP and D datasets in the region of highest frequency, respecting this overlap behavior in the full weight system generation where CY weight systems must be both D and IP. Alternatively, in each case when a property's subdataset is split into R and nR, the subsequent subdatasets exhibit distributions of a similar shape. The nonreflexive cases then have a higher density of high frequencies matching their usually more populous subdatasets distributed over smaller ranges, as demonstrated in (3.5).

The similarity between the R and nR subdatasets' PCA projections indicates classification architectures will likely find identification of this property harder. The higher skewed values of PCA projection, as well as the far larger maximum values, for the CY datasets will perhaps be used by the architectures to aid learning.

In addition, PCA is performed independently for the dataset of transverse CY weight systems (i.e., union of the CYR and CYnR subdatasets), exhibiting comparable explained variances and dominant normalized eigenvector. The two-dimensional projection of these data is presented in Fig. 6(b) and shows a similar forking structure to Fig. 3, equivalently seen for CY threefolds in [28]. Furthermore, as seen in the plots against $h^{1,1}$, the reflexive weight systems dominate the tails of the forks. All these comparisons corroborate the suggested intimately linear relationship between the weights and $h^{1,1}$, priming the data for machine learning application.

IV. MACHINE LEARNING

In this section, we present the results of various investigations performed through supervised machine learning (ML). Neural networks (NNs) are employed to predict the Hodge numbers of Calabi-Yau fourfolds and to identify

weight systems with specific properties. These two applications are different in nature, and for this reason, despite using the same NN architecture, some of the metadata choices differ.

NNs are high-dimensional nonlinear function fitters; they are built from constituent neurons that receive a vector input, act linearly on that vector to produce a number, then act nonlinearly on that number with an activation function: $\underline{x} \mapsto act(\underline{\mathfrak{w}} \cdot \underline{x} + \mathfrak{b})$, for NN weights $\mathfrak{w}$ (not to be confused with weight system weights w_i), bias $\mathfrak{b}$, and activation $act(\cdot)$. The neurons are organized into layers, such that the output numbers of each neuron in a layer are concatenated into a vector to pass to all the neurons in the next layer. Overtraining the "optimizer" compares output predictions of the NN function to true values of training data through a *loss*, updating the $(\mathfrak{w}, \mathfrak{b})$ parameters to optimize the fitting. After training is complete, the trained NN is used to predict output values on independent test data, from which performance measures can then be calculated [8].

For the prediction of cohomological data, which has a very wide range of possible values, a NN regressor was used. Since the input data are small (just six integers), a simple architecture with few layers was enough for this problem. Specifically, we used the built-in multilayer perception regressor from SCIKIT-LEARN, with the following features: (16, 32, 16) layer structure, rectified linear unit activation, mean squared error (MSE) loss, and Adam optimizer. We chose a training-test split of 80:20 and performed a fivefold cross-validation for each investigation. The batch size was set to 200, and we imposed an upper bound of 250 epochs (the network could stop before that if it reached convergence). Regarding the performance measures, we focused on the following three:

$$\begin{aligned} \text{MSE} &= \frac{1}{n}\sum (y_{\text{pred}} - y_{\text{true}})^2 && \in [\mathbf{0}, \infty), \\ \text{MAPE} &= \frac{1}{n}\sum \left| \frac{y_{\text{pred}} - y_{\text{true}}}{y_{\text{true}}} \right| && \in [\mathbf{0}, \infty), \\ R^2 &= 1 - \frac{\sum (y_{\text{true}} - y_{\text{pred}})^2}{\sum (y_{\text{true}} - y_{\text{truemean}})^2} && \in (-\infty, \mathbf{1}], \end{aligned} \quad (4.1)$$

for outputs y, where the bold numbers indicate the optimal values, i.e., those corresponding to perfect prediction, and MAPE stands for mean absolute percentage error.

Conversely, for the identification of weight system properties, a NN classifier was employed. This was built and implemented with the same architecture as the regressor (using TensorFlow [51]), however, changing the loss and performance measures to match the classification problem style. The loss function was categorical cross-entropy, and performance measures were functions of the confusion matrix. A confusion matrix, CM_{ij}, counts the number of test data inputs in class i that the trained NN classifies into class j; this can then be normalized and the performance measures defined,

$$\begin{aligned} \text{Accuracy} &= \sum_i CM_{ii} && \in [0, \mathbf{1}], \\ \text{MCC} &= \frac{\sum_{ijk}\left(CM_{ii}CM_{jk} - CM_{ij}CM_{ki}\right)}{\sqrt{\left(\sum_i\left(\sum_j CM_{ij}\right)\left(\sum_{k\neq i,l} CM_{kl}\right)\right)\left(\sum_i\left(\sum_j CM_{ji}\right)\left(\sum_{k\neq i,l} CM_{lk}\right)\right)}} && \in [-1, \mathbf{1}], \end{aligned} \quad (4.2)$$

again where bold values indicate optimal values for perfect learning. Note that where accuracy is very interpretable as the proportion of correctly classified inputs, the Matthews correlation coefficient (MCC) is, in general, a more representative measure as it accounts for off-diagonal terms and hence generalized type I and II errors.

A. Regressing Hodge numbers

Following the promising performances presented in [28], we employ a supervised ML technique on the Calabi-Yau weight system dataset under investigation. While for the threefolds case both $h^{1,1}$ and $h^{1,2}$ could be learned to high levels of precision, we find that the same is not true for fourfolds. This does not come as a surprise, since the underlying geometric structure becomes richer and more complicated by going up in complex dimensions. In fact, $h^{1,1}$ is still learned with very high precision and accuracy, while the architecture described above proves less adequate for $h^{1,2}$ and $h^{1,3}$. This is partially shown in Table III, where we also observe a trend that is common to all of our findings. We note that the small-weights regime is essentially different from the large-weights regime in terms of ML performance. The neural networks yield consistently better results when restricted to the first half of the dataset, compared to the second half. This suggests that there are some features, associated with the large-weights behaviors, which are harder to learn with our architecture.[13] Moreover, we observe another drop in accuracy when investigating the

[13]One might think that this is motivated by the fact that the second half of the dataset contains a wider range of cohomological numbers, since it has a wider range of weights. However, this is not the case, as shown in Table IV.

TABLE III. This table shows the performances of the fully connected neural network on $h^{1,1}$ and $h^{1,3}$ for the full dataset, the lower half (which contains smaller weights), and the upper half (containing larger weights).

	$h^{1,1}$			$h^{1,3}$		
Data Investigated	First half	Second half	Whole	First half	Second half	Whole
R^2	0.9261 ± 0.0018	0.9114 ± 0.0024	0.9101 ± 0.0049	0.9378 ± 0.0086	0.279 ± 0.091	0.063 ± 0.076
MAPE	0.1578 ± 0.0056	0.2498 ± 0.0015	0.409 ± 0.045	1.68 ± 0.35	4.50 ± 0.72	3.03 ± 0.63
MSE	2072 ± 55	14519431 ± 496316	8066404 ± 550014	1188821 ± 90490	61416783 ± 6406298	48040860 ± 3496383

TABLE IV. This table shows the ranges of the topological quantities under investigation for the two halves of the dataset and their mean value. The dataset is ordered according to the sum of weights (see bottom right) and not according to any of the cohomological properties. Hence, we see that the range of the invariants does not split into two disjoint sets among the two halves.

Subset		First half	Second half
$h^{1,1}$	Min	1	212
	Max	1173	303148
	Mean	204.1	5663.5
$h^{1,2}$	Min	0	0
	Max	1989	2010
	Mean	21.9	26.4
$h^{1,3}$	Min	1	1
	Max	303148	227486
	Mean	1475.9	3124.7
$h^{2,2}$	Min	82	1062
	Max	1213644	1213644
	Mean	6720.0	35143.9
χ	Min	−252	720
	Max	1820448	1820448
	Mean	9996.2	52618.8
w_{tot}	Min	6	4480
	Max	4480	6521466
	Mean	1561.6	60170.4

whole dataset, showing that the NN struggles to deal with these two regimes at once. For reference, we report the properties of the two halves in Table IV. In order to probe the performance of ML on the full problem, i.e., determining the complete Hodge diamond, we also focused on learning $h^{2,2}$, both on its own and together with the two Hodge numbers above. As shown in Sec. II, such a triple is enough to contain all the cohomological information. The results of these investigations are shown in Table V. We again see that the accuracy drops from left to right, according to the chosen subset. Finally, for completeness, we also present our results on $h^{1,2}$ and χ in Table VI. Since both of them can be zero, the MAPE measure does not apply to these cases, and therefore we omit it.

The fact that higher cohomologies in Calabi-Yau fourfolds are harder to learn with neural networks has already appeared in the literature, in [41]. Although they analyzed a different construction of fourfolds, i.e., CICY, their results also indicate that $h^{1,1}$ is the only Hodge number that can be successfully learned to high levels of precision with fully connected networks. Convolutional neural network variants have exhibited the highest accuracies on the CICY matrix inputs [32]; however, due to the permutation symmetry of the configuration matrices, as well as the weight system vectors for the construction considered here, the benefits of the convolutional architecture's focus on local properties is lost. We therefore stick to the more general dense feed-forward architectures.

TABLE V. This table shows the performances of the fully connected neural network on $h^{2,2}$ and on the triple $(h^{1,1}, h^{1,3}, h^{2,2})$, which specifies all the cohomological information. Again, we report results associated with the full dataset, to the lower half only (which contains smaller weights), and to the upper half only (containing larger weights).

	$h^{2,2}$			$(h^{1,1}, h^{1,3}, h^{2,2})$		
Data investigated	First half	Second half	Whole	First half	Second half	Whole
R^2	0.944 $\pm$ 0.015	0.714 $\pm$ 0.022	0.6228 $\pm$ 0.0082	0.670 $\pm$ 0.093	0.529 $\pm$ 0.016	0.528 $\pm$ 0.015
MAPE	0.497 $\pm$ 0.060	0.60 $\pm$ 0.09	0.67 $\pm$ 0.04	1.9 $\pm$ 0.4	3.3 $\pm$ 0.4	2.0 $\pm$ 0.2
MSE	19287530 $\pm$ 5619861	1295749477 $\pm$ 109548762	1006910414 $\pm$ 36669624	37962157 $\pm$ 15616868	578685604 $\pm$ 25936516	348268647 $\pm$ 10385842

TABLE VI. This table shows the performances of the fully connected neural network on χ and on the triple $h^{1,2}$. Both invariants can be zero, so we omit the MAPE measure, which is not well defined.

Data investigated	χ			$h^{1,2}$		
	First half	Second half	Whole	First half	Second half	Whole
R^2	0.9400 $\pm$ 0.0033	0.653 $\pm$ 0.015	0.616 $\pm$ 0.010	0.0715 $\pm$ 0.0069	0.0554 $\pm$ 0.0064	0.0436 $\pm$ 0.0038
MSE	46999083 $\pm$ 2865647	3551553463 $\pm$ 143894491	2309837118 $\pm$ 77130429	2520 $\pm$ 80	5100 $\pm$ 103	3834 $\pm$ 120

1. NN gradient saliency

Some first steps toward interpretability of these NN results start with gradient saliency analysis. The trained NNs are (highly nonlinear) functions from inputs to outputs, and differentiating these functions with respect to each of the inputs can give some indication of the dependency of the output classification on each part of the weight system.

In the saliency analysis performed here, each NN is differentiated with respect to each of the inputs and the differential evaluated at each of the test data inputs. The absolute values of these gradient components are then averaged over the test dataset, as well as averaged over the run repetitions—here repeating the investigation with randomized 80:20 train:test splits for 100 independent NNs of the same architecture. Since function scales can vary through the NN layers, the relative saliency values are the features of interest; they are represented, for the NNs predicting $h^{1,1}$, in Fig. 7. The six weights of the input weight systems are represented by six boxes, where lighter colors indicate higher saliency values and larger relative importance. These results show that the NNs focus on the weights in each system according to their size. They prioritize the information encoded in the lower weights, while the largest weights seem not to play an as important role. This implies that the networks are not exploiting the clustering behavior shown in Fig. 4, previously discussed. Perhaps to be expected, if we consider that the vast majority of weights actually lie in the "bulk" of the scatter plots, while the linear behavior is only evident for systems with extremely large weights.

FIG. 7. NN gradient saliency scores for the $h^{1,1}$ supervised learning on input weight systems. The lighter colors indicate a larger normalized absolute gradient for that weight in the input six-vector weight systems, where the saliency scores are averaged over the full test sets of each investigation and each of the 100 repetitions of the investigations.

2. Symbolic regression

While NNs have limited interpretability due to the large number of constituent functions being concatenated, there are other methods of supervised learning that are more directly interpretable for extracting mathematical insight.

With the knowledge that NNs can well predict $h^{1,1}$ values of Calabi-Yau fourfolds from the ambient weighted projective space weights alone, there is hence experimental evidence for approximate formulas connecting directly these integers. Motivated by this, in this section, techniques of symbolic regression are implemented via the GPLEARN library to search for candidate approximation formulas.

Symbolic regression is a method of supervised learning implemented via a genetic algorithm. Initially, a basis of functions is provided to the agent; here we will restrict ourselves to the standard normal division algebra basis: $\{+, -, \times, \div\}$. Then, a population of candidate expressions is randomly initialized as expression trees; where expression trees diagrammatically represent formulas as demonstrated in Fig. 8. The population of expressions is then

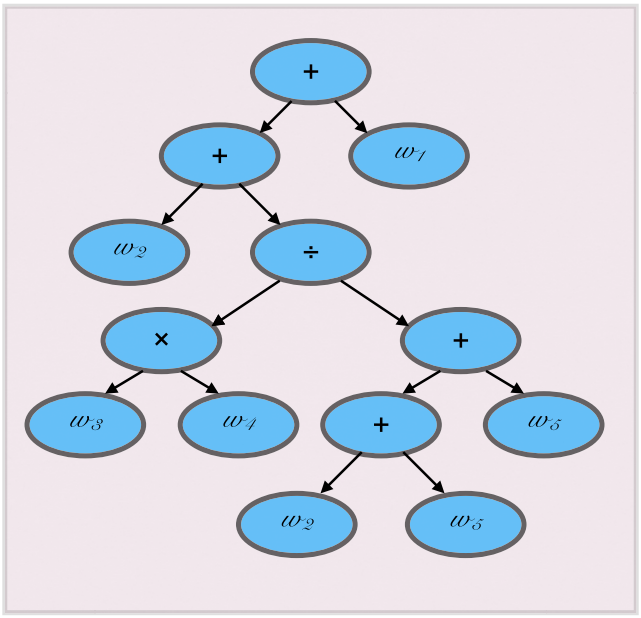


FIG. 8. An expression tree representing a candidate learned formula: $w_1 + w_2 + \frac{w_3 w_4}{w_2 + 2w_5}$, via symbolic regression.

TABLE VII. Candidate expressions for $h^{1,1}$ as functions of the six weights $(w_0, w_1, w_2, w_3, w_4, w_5)$ in the input transverse weight systems from independent symbolic regression runs, with respective performance measures.

Expression	R^2	MAPE
$w_1 + w_2 + \frac{w_3 w_4}{w_2 + 2w_5}$	0.884	0.322
$\frac{3}{4}w_1 + w_2 + \frac{3}{8}w_3$	0.872	0.332
$w_1 + w_2 + \frac{1}{6}w_4$	0.860	0.339

evaluated on the training data, noting a parsimony factor rewarding simpler expressions, and many of the best performing expressions are selected for breeding by swapping randomly selected subtrees. The output of the breeding is a new population of expressions that are then randomly mutated in a variety of ways to produce the next generation. This process of evaluating, breeding, and mutation is then iterated for a fixed number of generations, where the best expression is then selected from the final generation's population as the output. This output expression is then tested on the test data to produce the final performance measures, here using the same as for the NNs.

After 50 generations of 1000 expressions, with the GPLEARN recommended breeding and mutation factors and a parsimony of 0.8, the final output candidate expressions as well as performance measures for three independent runs were as given in Table VII.

With the goal of extracting just the first order behavior for the NN approximate formula, the high parsimony and simple function basis used has limited this regression performance, leading to expected lower performance scores relative to the NNs. Generalization to a broader basis with lower parsimony can provide expressions with higher performance, well demonstrated by a run using the full GPLEARN basis,

$$\left\{+, -, \times, \div, -(\cdot), \sqrt{\cdot}, \frac{1}{\cdot}, |\cdot|, \log(\cdot), \max(\cdot), \min(\cdot), \sin(\cdot), \cos(\cdot), \tan(\cdot)\right\}, \tag{4.3}$$

producing the expression

$$\frac{\sqrt{w_3 + 3\sqrt{w_4} + \min\left(\frac{w_4}{\log(\cos(w_0) - |w_1|)}, 2w_1 + \min\left((w_3 - w_0), \ldots\right.\right.}}{\sqrt{w_2\ \min\left(\sqrt{w_2^2 \log(\cos(w_0) - |w_1|)}, \min\left((w_3 - w_0), \min\left(\sqrt{w_2 w_3}, \sqrt{\frac{w_2^2 w_5}{w_3 - w_0}}\right)\right)\right)\right)}}, \tag{4.4}$$

with R^2 score 0.896. However, due to the far higher equation complexity and this relatively minimal increase in performance, the lack of interpretability puts motivation on consideration of the initially specified simple basis. Candidate expressions are quoted in Table VII, where these three independent expressions have similar performance and some similar structure.

The first thing to note is that in each equation there are three summed terms, which are each positive functions of weights. More specifically, each has a term equal to w_2 and another term either equal or proportional to w_1. The occurrence of these earlier weights somewhat corroborates the importance of earlier parts of the weight system seen in Sec. IV A 1, however, without the w_0 factor—which may be related to w_0 having a significantly smaller range. In each case, there is one further term involving higher weights, and across these expressions all additional weights do occur in this term. Overall, it is quite surprising how well such simple expressions can perform at predicting the $h^{1,1}$ values, and the simple linear sum behavior does support there being an approximate linear relationship as observed in Fig. 3.

B. Classifying CY property

The generation of weight system subdatasets for each property combination, as described in Sec. III B, enables the design of ML experiments to distinguish these properties in weight systems. In these cases, the problem is set up as supervised classification, again using the same NN architecture throughout these subinvestigations for consistency and ease of comparison.[14]

To investigate the stability of the partition, a multiclassification investigation is carried out between all eight subdatasets. Subsequently, a binary classification investigation is then carried out to probe the ability of ML architectures to identify each considered property: IP, intradivisibility, reflexivity, transversality (i.e., CY); for each of these, the datasets in each of the two classes were formed by taking appropriate unions of the partition

[14]We note that tuning hyperparameters leads to improved learning performance, however, here we are only focused on showing the existence of good learning, maintaining consistent architecture hyperparameters to compare between investigations.

TABLE VIII. Classification results for various partitions of the weight system data. The table shows the mean accuracy and MCC scores, to three decimal places, with standard error, across the five cross-validation runs, for the respective investigations labeled by the property being distinguished. The first investigation is multiclassification between all eight partitions of the weight system data: {CnIPnD, DnIP, IPnRnD, IPRnD, DnR, DR, CYnR, CYR}; the remaining investigations are binary classifications between unions of these nonoverlapping datasets as labeled by the index in the stated list of weight system partitions. The class sizes are also given for reference (where the second class exhibits the investigated property); many are approximately balanced classifications, but where they are not the MCC is a more appropriate nonbiased measure.

Investigation	Data partition	Class sizes	Accuracy	MCC
Multiclassification	$\{0\},\{1\},\{2\},\{3\},$ $\{4\},\{5\},\{6\},\{7\}$	[408124, 9614, 999975, 988436, 172462, 81215, 847122, 252933]	0.796 ± 0.007	0.740 ± 0.009
IP	$\{0,1\},\{2,3,4,5,6,7\}$	[417738, 3342143]	0.963 ± 0.001	0.808 ± 0.008
Intradivisible	$\{0,2,3\},\{1,4,5,6,7\}$	[2396535, 1363346]	0.906 ± 0.001	0.795 ± 0.003
Reflexive	$\{2,4,6\},\{3,5,7\}$	[2019559, 1322584]	0.848 ± 0.002	0.681 ± 0.003
Calabi-Yau	$\{0,1,2,3,4,5\},\{6,7\}$	[2659826, 1100055]	0.940 ± 0.002	0.852 ± 0.002
Calabi-Yau reflexive	$\{6\},\{7\}$	[252933, 847122]	0.774 ± 0.001	0.132 ± 0.009

subdatasets. To avoid problems caused by unbalanced datasets, during training class weights were fed into the NN such that it is proportionally more rewarded for correctly classifying weight systems in smaller classes; furthermore, the MCC performance measure was used, which is known to be unaffected by unbalanced class sizes—in this sense, the MCC is the more appropriate measure of learning.

The investigations, with the appropriate partitions of the partition subdatasets, as well as class sizes, and finally the averaged learning results over the fivefold cross-validation are presented in Table VIII.

These classification results are all considerably strong. For the multiclassification problem, an untrained NN would have null performance expressed by an accuracy ~ 0.125 and MCC ~ 0; however, both performance measures are substantially higher than these scores. Therefore, despite the weight systems being generally indistinguishable by eye, the NNs can learn to extract the appropriate property information sufficiently enough to classify well. Examining further the classification output, the averaged normalized confusion matrix for this multiclassification investigation is given by

$$\begin{pmatrix} 0.084 & 0.000 & 0.012 & 0.004 & 0.000 & 0.000 & 0.009 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.001 & 0.001 & 0.000 & 0.000 & 0.000 \\ 0.003 & 0.000 & 0.263 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.001 & 0.000 & 0.000 & 0.249 & 0.006 & 0.000 & 0.005 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.032 & 0.012 & 0.000 & 0.001 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.015 & 0.006 & 0.000 & 0.001 & 0.000 \\ 0.009 & 0.000 & 0.001 & 0.024 & 0.005 & 0.000 & 0.185 & 0.002 \\ 0.003 & 0.000 & 0.000 & 0.007 & 0.001 & 0.000 & 0.054 & 0.002 \end{pmatrix}, \tag{4.5}$$

to three decimal places, where the row is the true class and column is the predicted class. The matrix diagonal represents correctly classified weight systems. As can be seen, the NNs prioritize the first, third, fourth, and sixth classes such that the coprime weight systems with none of the properties are well distinguished from IP and CY subdatasets—these are also the most populous classes. The off-diagonal terms are mostly zero, indicating good learning. However, the demands of this multiclassification problem are high; the architectures must learn to identify many quite different properties simultaneously. Despite this, the surprising success motivates the binary classification of each property individually.

Each of the binary classification investigations exhibits higher performance measures than multiclassification, indicating that the architectures unsurprisingly perform better when learning one weight system property at a time. Theoretically, these trained NNs could then each be used in turn to identify the properties of a new candidate weight system and which part of the partition it probably lies in. The benefit of this is the computation of, particularly, IP and reflexivity becomes especially expensive for larger

weight systems, where the respective polytope is large, and then calculating the dual polytope to check these properties takes increasingly more memory and time. With the trained NNs, candidate weight systems could be fed into these NNs allowing quick elimination of weight systems unlikely to satisfy these properties. Then the expensive analytic checks can be performed for the filtered weight system database, producing a far higher proportion of weight systems with the desired properties.

Focusing on the MCC scores, the architectures struggle most with identifying reflexivity, especially in the CY case. Considering the number of steps required to compute this analytically, via construction of the respective lattice polytope, taking the dual polytope, and then performing many integer checks of the corresponding vertices, this is perhaps not surprising. Conversely, the performance for identifying the IP property is then surprisingly high, which still requires generating the polytope. Therefore, it is likely the NNs can approximate the polytopes within their architectures but struggle with the integer checks of vertices—a property notoriously evasive for ML [116]. Respectively, the NNs can well learn the intradivisibility property, a more direct computation with the weight data, however, still with a number of necessary checks. Finally, and most pleasingly, the CY property can be well learned for fourfold weight systems, a result observed for threefolds in [28]. The ability to so successfully predict the existence of further singularity structure in the respective hypersurfaces beyond that of the ambient weighted projective space remains astounding, for a method that is still unclear how to perform directly without using the Landau-Ginzburg string interpretation.

1. NN gradient saliency

The relative saliency values for each of the classification tasks are represented in Fig. 9. For these investigations, the saliency scores only show significant dependence on the input features for the reflexivity identification, where the earlier weights in the sorted weight systems (and hence smaller weights) are more important in determining the classification. This is accentuated for the CY reflexivity investigation. This is likely related to the distribution in Fig. 3, where reflexive weight systems appear to be skewed toward lower weights.

V. THE APPROXIMATION

The computation of the Hodge numbers for Calabi-Yau fourfolds as hypersurfaces in weighted projective spaces was performed in [46] via the Landau-Ginzburg model. Such a calculation involves constructing a number of Poincaré-type polynomials and summing their contributions. To do so, one has to perform polynomial multiplications and divisions, which become computationally expensive when the sum of the weights takes large values. This is also the regime where the linear clustering behavior is more manifest. In the present section, we introduce an approximation for the Hodge numbers, which is well defined for all Calabi-Yau weight systems, always provides a lower bound, and is significantly faster to compute.

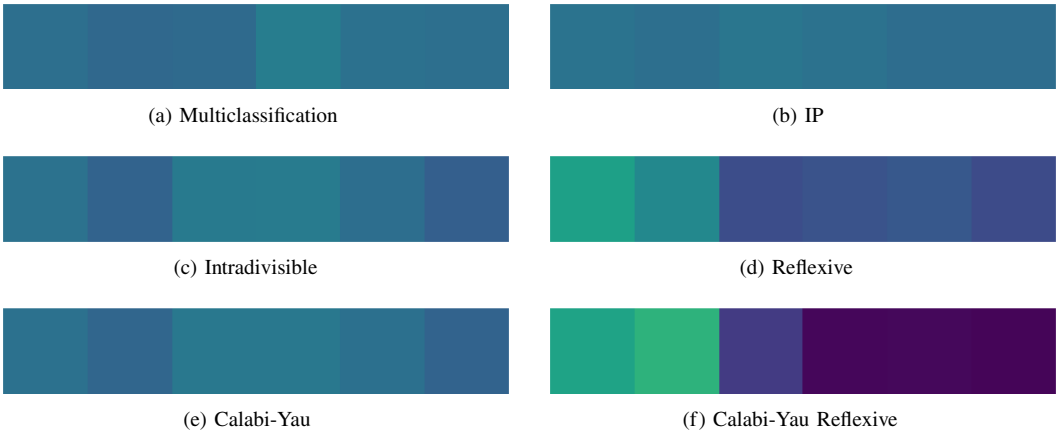
A. Presenting the formula

We first review the standard calculation by following [111]. Given a weight system $(w_0, \ldots, w_n)$, we define

$$(q_0, q_1, \ldots, q_n) = \left(\frac{w_0}{w}, \frac{w_1}{w}, \ldots, \frac{w_n}{w}\right) \in \mathbb{Q}_{>0}^{n+1}. \tag{5.1}$$

Moreover, for $0 \leq l < w$, we further define

$$\begin{aligned} \theta(l) &= (\theta_0(l), \theta_1(l), \ldots, \theta_n(l)) = (lq_0, lq_1, \ldots, lq_n), \\ \tilde{\theta}(l) &= \left(\tilde{\theta}_0(l), \tilde{\theta}_1(l), \ldots, \tilde{\theta}_n(l)\right) \in [0, 1)^{n+1}, \end{aligned} \tag{5.2}$$



(a) Multiclassification (b) IP (c) Intradivisible (d) Reflexive (e) Calabi-Yau (f) Calabi-Yau Reflexive

FIG. 9. NN gradient saliency scores for the property classification supervised learning on input weight systems. The lighter colors indicate a larger normalized absolute gradient for that weight in the input six-vector weight systems, where the saliency scores are averaged over the full test sets of each investigation and each of the 100 repetitions of the investigations.

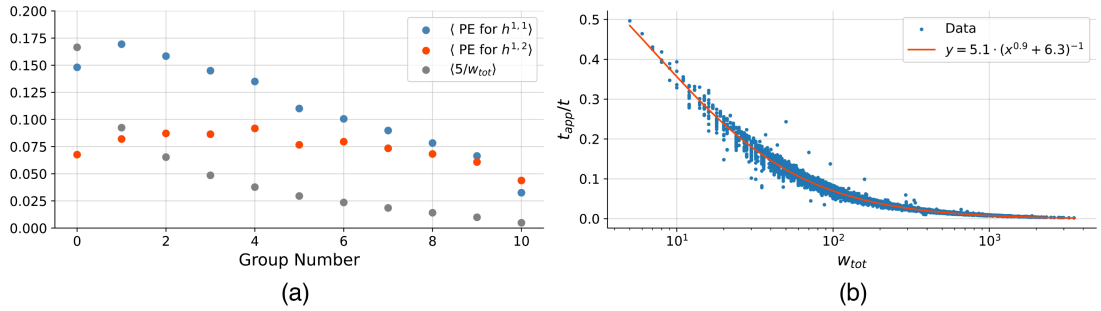


FIG. 10. These plots summarize the main features of the approximation (5.5), both in terms of accuracy and in terms of computational efficiency, compared with the exact formula (5.4). The data refer to Calabi-Yau threefolds as hypersurfaces in weighted projective spaces.

with $\tilde{\theta}$ being the canonical representative of $\theta(l)$ in $(\mathbb{R}/\mathbb{Z})^{n+1}$. To conclude, we reintroduce the last two quantities that appear in the formula for the Hodge numbers,

$$\begin{aligned} \mathrm{age}(l) &= \sum_{i=0}^{n} \tilde{\theta}_i(l) = \sum_{\tilde{\theta}_i(l)\neq 0} \tilde{\theta}_i(l), \\ \mathrm{size}(l) &= \mathrm{age}(l) + \mathrm{age}(w_{\mathrm{tot}} - l). \end{aligned} \tag{5.3}$$

Given these ingredients, then the full formula for the Hodge numbers $h^{p,q}$ reads

$$\sum_{p,q}(-1)^{p+q} h^{p,q} u^p v^q = \frac{1}{uv} \sum_{0\le l<w_{\mathrm{tot}}} \left[\prod_{\tilde{\theta}_i(l)=0} \frac{(uv)^{q_i} - uv}{1-(uv)^{q_i}}\right]_{\mathrm{int}} \times (-u)^{\mathrm{size}(l)} \left(\frac{v}{u}\right)^{\mathrm{age}(l)}. \tag{5.4}$$

It is evident that the polynomial products and divisions within the square brackets are what takes most of the computational resources. Just for reference, we note that, for high weights, the software SageMath cannot perform the calculation due to the high number of terms. For this reason, the algorithm had to be hard coded directly. One way to go about simplifying this formula is to identify for which values of l the terms in the sum contribute the most. Just by empirical observation, we find that the main contributions come from the element of zero age and the ones with maximal size.[15] As we are about to argue in detail, it turns out that including only terms of these types provides a very efficient approximation for the Hodge numbers. It is both very accurate and much faster to implement. Moreover, it bounds the exact results from below. Explicitly, the approximated Hodge numbers $h^{p,q}_A$ can be computed as

$$\sum_{p,q}(-1)^{p+q} h^{p,q}_A u^p v^q = \frac{1}{uv}\left\{\left[\prod_i \frac{(uv)^{q_i} - uv}{1-(uv)^{q_i}}\right]_{\mathrm{int}} + \sum_{\mathrm{size}(l)=n} (-u)^{w_{\mathrm{tot}}} \left(\frac{v}{u}\right)^{\mathrm{age}(l)}\right\}. \tag{5.5}$$

We tested this approximation against the two relevant datasets: the Calabi-Yau fourfolds considered in this article and the smaller set of Calabi-Yau threefolds. We start by presenting our findings for the latter case.[16]

B. Application to Calabi-Yau threefolds

As just mentioned, (5.4) becomes more and more involved to compute as the weights in the system become larger. Since the dataset for Calabi-Yau manifolds in weighted projective spaces are already ordered according to the sum of the weights, we conveniently divided the 7555 threefolds' weight systems (of five weights) into 11 groups, from lowest to highest. The plot in Fig. 10(a) shows the mean percentage error of the approximation formula (5.5) for each of those groups, both for $h^{1,1}$ and $h^{1,2}$, showing that the approximation becomes more precise as we go to higher weights. For reference, we also include a measure for the mean sum of weights in each of the groups $\langle 5/w_{\mathrm{tot}}\rangle$. We observe that the average percentage error for large weights is remarkably small, lying somewhere between 3% and 5% for both Hodge numbers.

The plot of Fig. 10(b) shows the ratio of the computational time against the sum of weights w_{tot}. As anticipated, the computational time needed to evaluate (5.5) is considerably smaller than the time taken by the full version (5.4). In fact, their ratio gets to values on the order of 10^{-2} for the largest weight systems in the dataset.[17]

[15]This is well exemplified by looking at examples 4.5 and 4.6 in [111].

[16]For completeness, we point out that the approximation also fails to be well defined for non-Calabi-Yau weight systems in general.

[17]One might argue that our implementation of the formula (5.4) could be further optimized, reducing the time needed to compute the Hodge numbers exactly. However, such an optimization would lead to a quicker implementation of (5.5) as well, so that we do not expect the ratio to change significantly.

TABLE IX. Performance measures for the approximation applied to Calabi-Yau threefolds. Specifically, the R^2 score, mean absolute percentage error, mean absolute error, and percentage of exact results where the approximation matched the true value.

	$h^{1,1}$	$h^{1,2}$
R^2 score	0.969	0.981
MAPE	0.113	0.075
MAE	7.1	3.3
Exact results	32%	56%

Some other useful figures for this approximation are reported in Table IX. These results show that the approximation, even though it excludes the vast majority of the terms that appear in (5.4), is still able to match the exact values a significant number of times. Moreover, we find another crucial feature of the truncated sum (5.5),

$$h_A^{1,1} \leq h^{1,1} \quad \text{and} \quad h_A^{1,2} \leq h^{1,2}. \tag{5.6}$$

Thus, since this is also the case for fourfolds, the approximation presented in this work offers a quickly accessible tool for extracting tight lower bounds of the Hodge numbers.

As a final feature, we note that the dataset built from the approximated Hodge numbers $h_A^{1,1/2}$ correctly reproduces the clustering behavior observed in [28]. This is best shown with a histogram plot, in Fig. 11, where we can clearly see various peaks in $h^{1,1}/w_{\max}$, corresponding to the slopes of the clustering lines. They overlap almost completely, showing that the clusters are essentially the same for both datasets: the exact Hodge numbers and the ones obtained via our approximation. Consistent with (5.6), the peaks

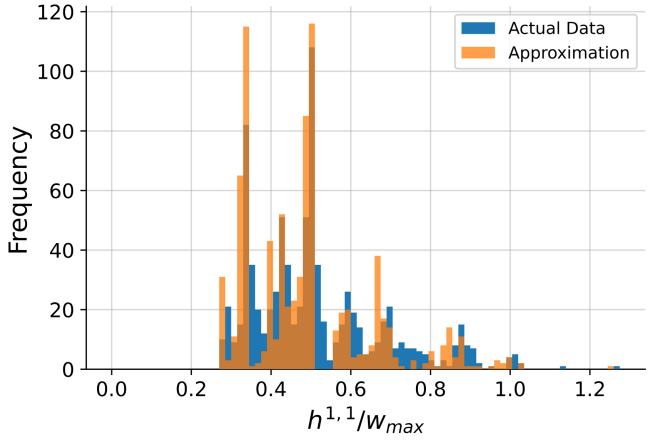


FIG. 11. This histogram plot shows that the approximated Hodge numbers also cluster for certain values of the ratio $h^{1,1}/w_{\max}$. Only weight systems with $w_{\max} > 250$ are plotted here, since this corresponds to the regime where the forking behavior is most visible. This plot should be compared with the one appearing in [28].

associated with $h_A^{1,1}$ are slightly shifted to the left. Thus, by narrowing down the full formula from the Landau-Ginzburg model (5.4) to a small number of terms, we obtained a much simpler expression, which still reproduces the same behavior for large weights. This might be a step forward toward the understanding of the linear clustering that characterizes the cohomological numbers of Calabi-Yau's in weighted projective spaces.

C. Application to Calabi-Yau fourfolds

We now move to the case of fourfolds in weighted projective spaces. This dataset is considerably bigger compared to threefolds, with 1100055 spaces, and it contains systems with very large weights. A natural consequence is that the computational times are much longer, which makes the advantages of the approximation even more evident. To give a concrete example, we focused on the millionth weight system in the dataset, which reads [45, 74, 2460, 12792, 17876, 33173]. Our implementation of (5.4) takes roughly 40 h, while the approximated version (5.5) is computed in 49 sec. The ratio between the two is 0.00034. Moreover, the approximated results are very accurate: the exact ones are $(h^{1,1} = 10718, h^{1,2} = 0, h^{1,3} = 986, h^{2,2} = 46860)$, while the approximated ones read (10683, 0, 986, 46500).[18]

Regarding the precision of the approximation, we illustrate it in Fig. 12(a), with a similar plot to the one used for threefolds. We omit $h^{1,2}$ from the picture because it can take the value of zero, making the percentage error not well defined. We sampled randomly and uniformly 20% of the dataset and then divided it into groups of 2000 samples. As before, all samples were ordered according to the sum of weights, then divided into the groups, as shown by the gray points. We observe once again that the percentage error gets smaller as the weights become larger, reaching roughly 1% for the samples with largest weights, for all three Hodge numbers. The plot in Fig. 12(b), on the other hand, shows the comparison between the computational resources employed by the exact expression from the Landau-Ginzburg model and by our approximation. It is evident from the example just discussed that systems with large weights necessitate a very long computation time for the full formula (5.4). Thus, we collected data within the first half of the dataset and then extrapolated our findings to the second half, containing very large weights. The red curve provides a good interpolation of the data, and it turns out to give very accurate predictions as well. This can be confirmed by plugging $w_{\text{tot}} = 66420$, which corresponds to the example weight system discussed above, into the expression for the best fit function. The result is $t_{\text{app}}/t = 0.00035$,

[18]To make this comparison, we had to match our conventions with the ones used for the existing datasets; this is discussed in the next section, after Eq. (6.2).

TABLE X. Performance measures for the approximation applied to Calabi-Yau fourfolds. Specifically, the R^2 score, mean absolute percentage error, mean absolute error, and percentage of exact results where the approximation matched the true value.

	$h^{1,1}$	$h^{1,3}$	$h^{2,2}$
R^2	0.999	0.999	0.999
MAPE	0.058	0.039	0.082
MAE	74.7	32.4	681.9
Exact results	26.9%	48.7%	7.1%

which is remarkably close to the actual ratio (0.00034) obtained from the explicit computation.

Finally, let us report the main properties of the approximation. They are shown in Table X, and they are extracted from the same data plotted in Fig. 12(a), i.e., from a set of roughly 220000 random Calabi-Yau weight systems. We end this section by making a final remark. The first one is that, analogous to the threefolds case, the approximation provides a lower bound also for fourfolds. However, this is trivially satisfied for $h^{1,2}$, since our approximation always yields zero for this case. Summarizing, we have that

$$h_A^{1,1} \leq h^{1,1} \quad h_A^{1,2} \equiv 0 \leq h^{1,2}, \quad h_A^{1,3} \leq h^{1,3}, \quad h_A^{2,2} \leq h^{2,2}. \tag{5.7}$$

Therefore, use of this approximation in practical computations of Hodge numbers not only provides a significant speed improvement, but also will always be a lower bound. Therefore, in designing string effective theories where the topology of the chosen Calabi-Yau manifold for compactification intrinsically sets many properties of the resulting theory, this approximation allows for incompatible manifolds to be confidently and quickly discarded where any $h_A^{p,q}$ is larger than the desired values for the desired theory being built. Additionally, it also provides a good approximation for the remaining candidates, allowing them to be sorted prior to search with the full formula, such that many less manifolds will need to be checked before finding the correct topology for the desired theory.

VI. HIGHER WEIGHT SYSTEMS

The approximation has been tested on both threefolds and fourfolds. These are the only two existing datasets of Calabi-Yau manifolds built as hypersurfaces in weighted projective spaces. Here, we present the first efforts toward the understanding of these spaces in higher dimensions, by generating a partial dataset of candidate transverse weight systems of seven weights, and the respective Calabi-Yau fivefolds' Hodge numbers. We discuss how the approximation can be used to quickly extract information about such a dataset, and we additionally generate a first partial dataset of candidate transverse weight systems of eight weights, then use (5.5) to construct an approximated list of sixfolds' Hodge numbers.

A. Calabi-Yau fivefolds

Calabi-Yau fivefolds appear in a number of dimensional reductions in the literature. For instance, it was found that M theory compactified on a Calabi-Yau fivefold results in an exotic $\mathcal{N} = 2$ supersymmetric quantum mechanics [117]. Moreover, Calabi-Yau fivefolds play a role in F theory, where upon compactification, they provide a way to systematically construct $N = (0, 2)$ conformal field theories (CFTs) [118,119], which may lead to their classification. Therefore, an extended dataset of such Calabi-Yau manifolds would make it possible to explore the landscape of such CFTs. Additionally, in [120], a three-dimensional string vacua with $\mathcal{N} = 1$ supersymmetry has been found, which can be interpreted as a compactification of S theory on a Calabi-Yau fivefold. Despite their role in the construction of low-dimensional theories, examples of fivefolds have not been systematically constructed, until the recent effort in [40], which focuses on the CICY

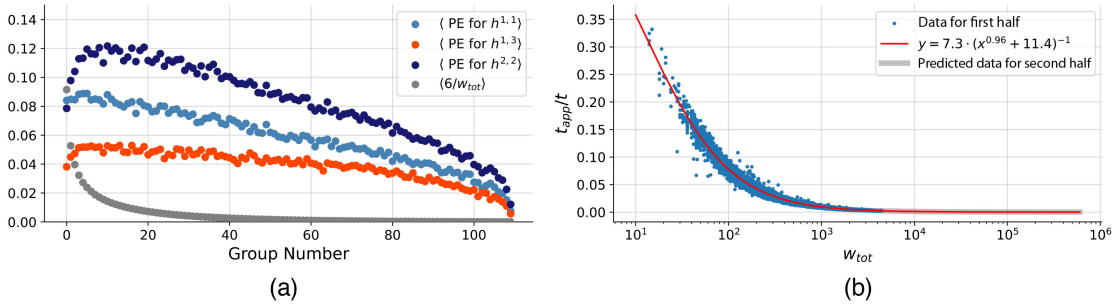


FIG. 12. These plots summarize the main features of the approximation (5.5), applied to the dataset of Calabi-Yau fourfolds as hypersurfaces in weighted projective spaces. (a) The mean accuracy for groups of 2000 weight systems ordered according to their sum of weights. We used 20% of the dataset for this plot, sampled uniformly. The computational efficiency is analyzed in (b), where the computational time of the approximation is compared to the one associated with the exact formula (5.4). We chose roughly 50000 samples randomly from the first half of the dataset (blue dots) and used these data to extrapolate the behavior for the second half. As discussed in the text, the best fit function shown predicts very accurately the ratios for systems with larger weights.

construction. Here, we present a second in that direction, i.e., we generate, for the first time, a subset of Calabi-Yau fivefolds obtained as hypersurfaces in $\mathbb{P}^6_w$. Specifically, we generate all seven-weight weight systems whose sum of weights $w_{\text{tot}} \leq 200$. To efficiently identify those that have the required property to describe a Calabi-Yau (for more details, see Sec. II), we employ a two-step approach. We first systematically search all partitions of each sum of weights up to 200 as weight systems, extracting those that are coprime, IP,[19] and intradivisible, performed with our code functionality. Then, we use the approximation as a tool for establishing the Calabi-Yau property and select all the weight systems that are well defined with respect to (5.5) or, equivalently, all those such that the polynomial division $[\prod_i \frac{(uv)^{q_i}-uv}{1-(uv)^{q_i}}]$ gives no reminder. The candidate weight systems identified this way were then all checked with respect to the exact formula (5.4), and all turned out to be well defined, yielding the full exact Hodge diamond. To provide confidence in the generated data, two nontrivial checks on the cohomological data were performed. First, computing the Euler number from the weights alone with (2.11) and then verifying it agrees with the identity

$$\chi = 2h^{1,1} - 4h^{1,2} + 4h^{1,3} + 2h^{2,2} - 2h^{1,4} - 2h^{2,3}, \quad (6.1)$$

in terms of the computed Hodge numbers. Moreover, we also checked that the Hodge numbers satisfy the constraint derived from the Atiyah-Singer index theorem,

$$11h^{1,1} - 10h^{1,2} - h^{2,2} + h^{2,3} + 10h^{1,3} - 11h^{1,4} = 0. \quad (6.2)$$

Both checks were passed by all the weight systems, which describe new Calabi-Yau geometries in complex dimension five. For reference, let us present five examples of such spaces, shown in Table XI. We point out a small difference in definitions between this section and the previous ones. In Secs. V B and V C, we compared our results with the existing datasets (which can be found at [121]), whose conventions are slightly different from ours. Namely, for Calabi-Yau threefolds, $h^{1,1}$ and $h^{1,2}$ determined using (5.4) have to be exchanged in order to match [121]. Analogously, $h^{1,1}$ and $h^{1,3}$ should be swapped for fourfolds to be consistent with the existing list. For the reminder of this paper, we present our results as they are obtained from (5.4).

TABLE XI. Examples of weight systems with seven weights, describing Calabi-Yau fivefolds, together with the associated invariants.

Weight system	$[h^{1,1}, h^{1,2}, h^{1,3}, h^{1,4}, h^{2,2}, h^{2,3}, \chi]$
[1, 1, 1, 1, 1, 1, 1]	[1667, 0, 0, 1, 18327, 1, 39984]
[1, 6, 8, 12, 14, 19, 60]	[3999, 0, 4, 3, 44022, 26, 96000]
[5, 9, 9, 18, 27, 34, 51]	[577, 201, 0, 12, 5430, 1225, 8736]
[4, 4, 5, 10, 27, 45, 85]	[5087, 8, 0, 12, 55938, 193, 121608]
[25, 25, 25, 25, 28, 32, 40]	[185, 350, 1, 30, 566, 2351, −4656]

Having mentioned this subtlety, we note that a quick consistency check of our results comes straightforwardly from considering the first entry in Table XI. This weighted projective space is trivial (i.e., is not actually weighted), so that it gives rise to the simplest Calabi-Yau fivefold defined by a degree-seven polynomial in $\mathbb{P}^6$. The associated cohomology matches the result reported in the appendix of [122].

The global properties of Calabi-Yau fivefolds as hypersurfaces in weighted projective spaces, with sum of weights up to 200, read

$$\langle h^{1,1}\rangle = 9004.6^{2314879}_{71}, \quad \langle h^{1,2}\rangle = 24.0^{15180}_{0}, \quad \langle h^{1,3}\rangle = 2.4^{703}_{0}, \quad \langle h^{1,4}\rangle = 8.8^{50}_{1}$$
$$\langle h^{2,2}\rangle = 98932.3^{25463560}_{566}, \quad \langle h^{2,3}\rangle = 194.9^{64930}_{1}, \quad \langle \chi\rangle = 215379.9^{55556832}_{-13248}. \quad (6.3)$$

The dataset of the 274730 weight systems with their topological properties is made available on GitHub, presented in the format $[[w_i], w_{\text{tot}}, [h^{p,q}], \chi]$, where $h^{p,q}$ are written in the same order as in (6.3) above. Some further analysis is given in Fig. 13.

As it can be guessed by looking at $\langle h^{1,1}\rangle$ and $\langle h^{1,4}\rangle$, the subset of spaces considered here does not show mirror symmetry. This is due to the fact that we only restricted ourselves to a small sum of weights, whose mirror-symmetric pairs lie in the large-weights regime. We expect the cohomological data in that regime to be practically inaccessible, due to the large computational times associated with (5.4). For this reason, we believe that the approximation presented in Sec. V, which proved to be extremely accurate for large weights in the fourfolds investigation, could be a key tool for attempting such a task. Moreover, we also expect the list of all possible Calabi-Yau seven-weight weight systems to be astronomical in size. Once again, the truncated formula (5.5) provides a quickly computable tight *lower* bound for the Hodge numbers of all those yet undiscovered manifolds.

B. Calabi-Yau sixfolds

While their role in physics is marginal (they could only be employed for compactifications of S theory), Calabi-Yau sixfolds have their own relevance directly within mathematics. These spaces could provide additional information

[19]Work in [104] showed that transverse weight systems are by necessity IP for any size weight system.

TABLE XII. Examples of weight systems with eight weights, describing Calabi-Yau sixfolds, together with the associated invariants.

Weight system	$[h_A^{1,1}, h_A^{1,2}, h_A^{1,3}, h_A^{1,4}, h_A^{1,5}, h_A^{2,2}, h_A^{2,3}, h_A^{2,4}, h_A^{3,3}, \chi]$
$[1,1,1,1,1,1,1,1]$	$[6371,0,0,0,1,154645,0,1,398568,720608]$
$[1,1,3,4,15,22,43,44]$	$[265283,0,0,0,6,6629968,0,27,16974104,30764676]$
$[1,1,3,8,15,16,40,84]$	$[484547,0,0,0,4,12217438,0,17,31218692,56622576]$
$[7,7,7,14,14,25,41,74]$	$[9905,0,0,0,22,109916,0,40,207950,-162552]$
$[18,20,20,25,25,26,30,36]$	$[344,0,0,0,6,7499,0,37,19303,34360]$

about the—still very mysterious to this day—landscape of Calabi-Yau geometries, as they are the second nontrivial family of Calabi-Yau manifolds in even complex dimensions. Their construction as hypersurfaces of weighted projective spaces involves using eight-weight weight systems, which are both more numerous and effectively infeasible to run bulk computation of exact topological parameters. For these reasons, we find the truncated approximation formula to be especially pertinent, allowing computation of approximated Hodge values for all the generated candidate transverse weight systems with $w_{\text{tot}} \leq 200$. Once again, we first identify the IP intradivisible weight systems and then select the ones that are well defined with respect to (5.5), numbering 1482022 candidate transverse weight systems of eight weights (accessible at this work's respective GitHub in the same format as for the fivefolds). From there the Euler number was also computed exactly with the less computationally intensive direct formula from weights (2.11). A few examples are reported in Table XII.

Similar to before, the first manifold in Table XI is nothing but the Calabi-Yau sixfold defined by a degree-eight polynomial in $\mathbb{P}^7$. We find that our results, despite coming from the truncated formula, exactly match the exact Hodge numbers, which can be found in [123].

The preliminary analysis is also shown in Fig. 13, where a comparison across different complex dimensions is shown; it illustrates that, for a low sum of weights, the fivefolds and sixfolds are appropriately skewed toward positive Euler numbers. The approximated Hodge numbers show a similar behavior to what one would expect based on the other distributions. Just for reference, we report here the main global features of the approximated invariants that we computed,

$$\begin{aligned}
&\langle h_A^{1,1}\rangle = 96686.7^{147270231}_{279}, \qquad h_A^{1,2} \equiv 0, \qquad h_A^{1,3} \equiv 0, \qquad h_A^{1,4} \equiv 0, \qquad \langle h_A^{1,5}\rangle = 4.3^{28}_{1},\\
&\langle h_A^{2,2}\rangle = 2424722.8^{3759686446}_{6366}, \qquad h_A^{2,3} \equiv 0, \qquad \langle h_A^{2,4}\rangle = 27.5^{65}_{1},\\
&\langle h_A^{3,3}\rangle = 6189235.0^{9581156426}_{15542}, \qquad \langle \chi\rangle = 11309730.8^{17395069848}_{-708480}.
\end{aligned} \tag{6.4}$$

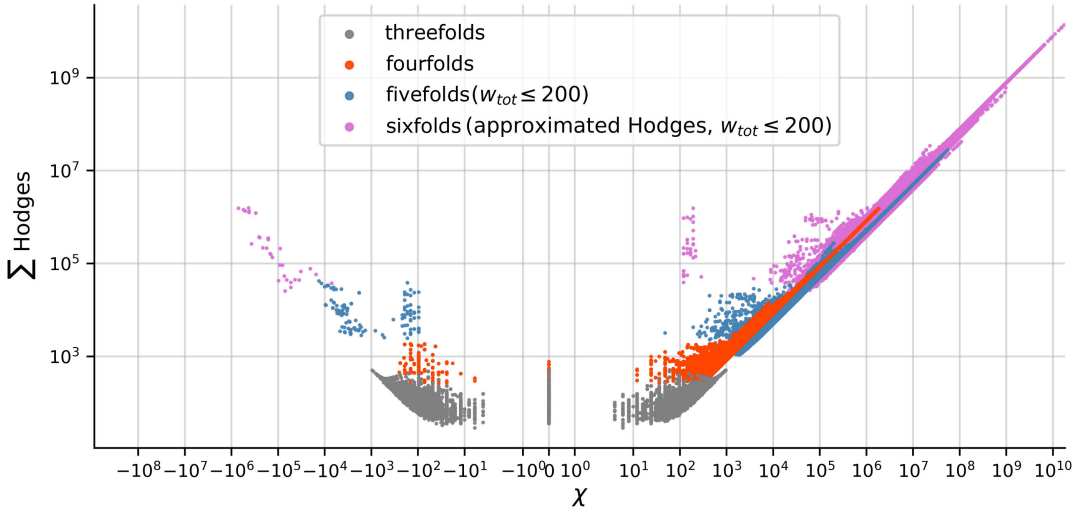


FIG. 13. This figure illustrates some features of the cohomological data and Euler numbers for Calabi-Yau manifolds constructed as hypersurfaces in weighted projective spaces. The threefolds' and fourfolds' data points exhaust all possible Calabi-Yau manifolds of that type. For fivefolds and sixfolds, we restricted ourselves to weight systems with $w_{\text{tot}} \leq 200$. These are all newly discovered geometries. The Hodge numbers for sixfolds were obtained through the approximation (5.5), hence they represent tight lower bounds of these cohomological properties.

VII. SUMMARY AND OUTLOOK

This work was focused on, but not limited to, the analysis of Calabi-Yau fourfolds obtained as hypersurfaces in weighted projective spaces. By restricting to systems with large weights, a linear clustering behavior analogous to the one found for threefolds in [28] was observed and quantitatively corroborated through the K-Means clustering normalized inertia. By gradually relaxing the conditions on the weights, we were able to produce a partition of coprime weight systems according to the most relevant properties: IP, reflexivity, intradivisibility, and transversality (Calabi-Yau), generating datasets for each subset in such a partition.

While all of the above was performed using concrete analytic algorithms, statistical machine learning techniques were also applied both to the dataset of Calabi-Yau fourfolds and to the partitioned set of more general weight systems. Regarding the former, a fully connected regressor network was shown to predict the cohomological Hodge data and the Euler number from the system weights. We found particularly good results, with $R^2 \sim 0.91$, for $h^{1,1}$, on the whole dataset. For the other invariants, we observed very different results for systems with small weights as opposed to systems with large weights. For instance, $h^{1,3}$ and $h^{2,2}$ showed results with $R^2 > 0.90$ for the half of the dataset containing lower weights, while the accuracy dropped significantly for the other half. These three numbers provide sufficient information to determine the full Hodge diamond; however, results were also reported associated with $h^{1,2}$, which had a poor performance since it is zero 48% of the time, and χ, which showed similar results to $h^{1,3}$ and $h^{2,2}$.

The partition of weight systems according to their respective properties within $\{\text{IP}, \text{reflexive}, \text{intradivisible}, \text{transverse}\}$, where transversality implied the existence of a Calabi-Yau hypersurface, was classified with the respective fully connected classification architecture. Multiclassification results were surprisingly high between all parts of the partition, reaching MCC scores of 0.740. Separately, binary classification investigations managed to well identify each property respectively from unions of the partition subdatasets, struggling most with reflexivity.

Motivated by the strong performances of the neural networks and inspired by the interpretability of the gradient saliency analysis and symbolic regression, we explored a simpler truncated version of the formula coming from the Landau-Ginzburg model used for calculating the Calabi-Yau Hodge numbers from the ambient $\mathbb{P}^n_{\mathbf{w}}$'s weight system. This approximation drastically reduces the number of terms involved in the computation, making it easier to study analytically and substantially faster to compute numerically. Its main features are as follows: it provides a tight lower bound for the Hodge numbers; it is especially accurate for systems with large weights (average MAPE of $< 1\%$ for the 10000 systems with largest weights); it is dramatically faster than the exact formula (up to 10^4 times quicker); it reproduces the observed linear clustering behavior for large weights.

Finally, motivated by the speed improvements available from this approximation, transverse weight systems (satisfying the necessary intradivisible and IP properties and well defined with respect to both the approximation and exact Landau-Ginzburg formula) were generated for a sum of weights $w_{\text{tot}} \leq 200$, for systems of seven weights producing Calabi-Yau fivefolds. Additionally, where the exact Landau-Ginzburg formula computation time was infeasible for systems of eight weights, a complementary dataset of candidate transverse weight systems (satisfying the necessary intradivisible and IP properties and well-defined with respect to just the approximation) was generated, again for a sum of weights $w_{\text{tot}} \leq 200$, leading to candidate Calabi-Yau sixfolds. Some preliminary analysis of these data and the respectively computed topological properties is provided with a thorough analysis, and its full generation for $w_{\text{tot}} > 200$ is left for future work.

These datasets, the respective code for analysis and ML, and an example notebook illustrating functionality to check intradivisibility, compute Euler number, and compute exact and approximated Hodge numbers of an input weight system of any size are all available at this work's respective GitHub repository [52].

ACKNOWLEDGMENTS

E. H. acknowledges support from Pierre Andurand over the course of this research. T. S. G. was supported by the Science and Technology Facilities Council (STFC) Consolidated Grants No. ST/T000686/1 and No. ST/X00063X/1 "Amplitudes, Strings and Duality."

[1] E. Calabi, The space of Kähler metrics, *Proceedings of the International Congress Mathematicians Amsterdam* (1954), Vol. 2, pp. 206–207. Archived from the original on 07-17-2011.

[2] S.-T. Yau, On the Ricci curvature of a compact KÃ¤hler manifold and the complex Monge-ampere equation, I, Commun. Pure Appl. Math. **31**, 339 (1978).

[3] P. Candelas, G. T. Horowitz, A. Strominger, and E. Witten, Vacuum configurations for superstrings, Nucl. Phys. **B258**, 46 (1985).

[4] Y.-H. He, *The Calabi–Yau Landscape: From Geometry, to Physics, to Machine Learning*, Lecture Notes in Mathematics Vol. 5 (Springer, Cham, 2021).

[5] M. Kreuzer and H. Skarke, Complete classification of reflexive polyhedra in four-dimensions, Adv. Theor. Math. Phys. **4**, 1209 (2000).

[6] R. Altman, J. Gray, Y.-H. He, V. Jejjala, and B. D. Nelson, A Calabi-Yau database: Threefolds constructed from the Kreuzer-Skarke list, J. High Energy Phys. 02 (2015) 158.

[7] M. Demirtas, L. McAllister, and A. Rios-Tascon, Bounding the Kreuzer-Skarke landscape, Fortschr. Phys. **68**, 2000086 (2020).

[8] F. Ruehle, Data science applications to string theory, Phys. Rep. **839**, 1 (2020).

[9] J. Bao, Y.-H. He, E. Heyes, and E. Hirst, Machine learning algebraic geometry for physics, arXiv:2204.10334.

[10] Y.-H. He, E. Heyes, and E. Hirst, Machine learning in physics and geometry, arXiv:2303.12626.

[11] Y.-H. He, Deep-learning the landscape, arXiv:1706.02714.

[12] D. Krefl and R.-K. Seong, Machine learning of Calabi-Yau volumes, Phys. Rev. D **96**, 066014 (2017).

[13] F. Ruehle, Evolving neural networks with genetic algorithms to study the string landscape, J. High Energy Phys. 08 (2017) 038.

[14] J. Carifio, J. Halverson, D. Krioukov, and B. D. Nelson, Machine learning in the string landscape, J. High Energy Phys. 09 (2017) 157.

[15] M. Headrick and T. Wiseman, Numerical Ricci-flat metrics on K3, Classical Quantum Gravity **22**, 4931 (2005).

[16] M. R. Douglas, R. L. Karp, S. Lukic, and R. Reinbacher, Numerical Calabi-Yau metrics, J. Math. Phys. (N.Y.) **49**, 032302 (2008).

[17] A. Ashmore, Y.-H. He, and B. A. Ovrut, Machine learning Calabi-Yau metrics, Fortschr. Phys. **68**, 2000068 (2020).

[18] 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, J. High Energy Phys. 05 (2021) 013.

[19] M. R. Douglas, S. Lakshminarasimhan, and Y. Qi, Numerical Calabi-Yau metrics from holomorphic networks, arXiv:2012.04797.

[20] V. Jejjala, D. K. Mayorga Pena, and C. Mishra, Neural network approximations for Calabi-Yau metrics, J. High Energy Phys. 08 (2022) 105.

[21] M. Larfors, A. Lukas, F. Ruehle, and R. Schneider, Learning size and shape of Calabi-Yau spaces, arXiv:2111.01436.

[22] A. Ashmore, L. Calmon, Y.-H. He, and B. A. Ovrut, Calabi-Yau metrics, energy functionals and machine-learning, Int. J. Data Sci. Math. Sci. **1**, 49 (2023).

[23] M. Larfors, A. Lukas, F. Ruehle, and R. Schneider, Numerical metrics for complete intersection and Kreuzer-Skarke Calabi-Yau manifolds, Mach. Learn. Sci. Tech. **3**, 035014 (2022).

[24] P. Berglund, G. Butbaia, T. Hübsch, V. Jejjala, D. Mayorga Peña, C. Mishra, and J. Tan, Machine learned Calabi-Yau metrics and curvature, arXiv:2211.09801.

[25] M. Gerdes and S. Krippendorf, CYJAX: A package for Calabi-Yau metrics with JAX, Mach. Learn. Sci. Tech. **4**, 025031 (2023).

[26] A. Ashmore, Y.-H. He, E. Heyes, and B. A. Ovrut, Numerical spectra of the Laplacian for line bundles on Calabi-Yau hypersurfaces, J. High Energy Phys. 07 (2023) 164.

[27] H. Ahmed and F. Ruehle, Level crossings, attractor points and complex multiplication, J. High Energy Phys. 06 (2023) 164.

[28] D. S. Berman, Y.-H. He, and E. Hirst, Machine learning Calabi-Yau hypersurfaces, Phys. Rev. D **105**, 066002 (2022).

[29] K. Bull, Y.-H. He, V. Jejjala, and C. Mishra, Machine learning CICY threefolds, Phys. Lett. B **785**, 65 (2018).

[30] K. Bull, Y.-H. He, V. Jejjala, and C. Mishra, Getting CICY high, Phys. Lett. B **795**, 700 (2019).

[31] C. R. Brodie, A. Constantin, R. Deen, and A. Lukas, Machine learning line bundle cohomology, Fortschr. Phys. **68**, 1900087 (2020).

[32] H. Erbin and R. Finotello, Inception neural network for complete intersection Calabi-Yau 3-folds, Mach. Learn. Sci. Tech. **2**, 02LT03 (2021).

[33] B. Aslan, D. Platt, and D. Sheard, Group invariant machine learning by fundamental domain projections, in *Proceedings of the NeurIPS Workshop on Symmetry and Geometry in Neural Representations* (PMLR, 2023), pp. 181–218.

[34] H. Erbin and R. Finotello, *Deep Learning: Complete Intersection Calabi-Yau Manifolds* (World Scientific, Europe, 2023), pp. 151–181, 10.1142/9781800613706_0005.

[35] W. Cui, X. Gao, and J. Wang, Machine learning on generalized complete intersection Calabi-Yau manifolds, Phys. Rev. D **107**, 086004 (2023).

[36] D. Klaewer and L. Schlechter, Machine learning line bundle cohomologies of hypersurfaces in toric varieties, Phys. Lett. B **789**, 438 (2019).

[37] P. Berglund, B. Campbell, and V. Jejjala, Machine learning Kreuzer-Skarke Calabi-Yau threefolds, arXiv:2112.09117.

[38] T. S. Gherardini, Exotic spheres' metrics and solutions via Kaluza-Klein techniques, J. High Energy Phys. 12 (2023) 100.

[39] D. Aggarwal, Y.-H. He, E. Heyes, E. Hirst, H. N. S. Earp, and T. S. R. Silva, Machine-learning Sasakian and G_2 topology on contact Calabi-Yau 7-manifolds, Phys. Lett. B **850**, 138517 (2024).

[40] R. Alawadhi, D. Angella, A. Leonardo, and T. S. Gherardini, Constructing and machine learning Calabi-Yau five-folds, Fortschr. Phys. **72**, 2300262 (2024).

[41] Y.-H. He and A. Lukas, Machine learning Calabi-Yau four-folds, Phys. Lett. B **815**, 136139 (2021).

[42] H. Erbin, R. Finotello, R. Schneider, and M. Tamaazousti, Deep multi-task mining Calabi-Yau four-folds, Mach. Learn. Sci. Tech. **3**, 015006 (2022).

[43] J. Gray, A. S. Haupt, and A. Lukas, All complete intersection Calabi-Yau four-folds, J. High Energy Phys. 07 (2013) 070.
[44] J. Gray, A. S. Haupt, and A. Lukas, Topological invariants and fibration structure of complete intersection Calabi-Yau four-folds, J. High Energy Phys. 09 (2014) 093.
[45] M. Kreuzer and H. Skarke, Calabi-Yau four folds and toric fibrations, J. Geom. Phys. **26**, 272 (1998).
[46] M. Lynker, R. Schimmrigk, and A. Wißkirchen, Landau-Ginzburg vacua of string, M- and F theory at $c = 12$, Nucl. Phys. **B550-2**, 123 (1999).
[47] G. Brown and A. Kasprzyk, Four-dimensional projective orbifold hypersurfaces, Exp. Math. **25**, 176 (2015).
[48] F. Schöller and H. Skarke, All weight systems for Calabi-Yau fourfolds from reflexive polyhedra, Commun. Math. Phys. **372**, 657 (2019).
[49] P. Berglund, Y.-H. He, E. Heyes, E. Hirst, V. Jejjala, and A. Lukas, New Calabi-Yau manifolds from genetic algorithms, Phys. Lett. B **850**, 138504 (2024).
[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, SCIKIT-LEARN: Machine learning in PYTHON, J. Mach. Learn. Res. **12**, 2825 (2011).
[51] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, https://www.tensorflow.org/.
[52] E. Hirst and T. S. Gherardini, P5CY4ML, Version 1, https://github.com/Tancredi-Schettini-Gherardini/P5CY4ML.
[53] J. Bao, Y.-H. He, and E. Hirst, Neurons on amoebae, J. Symb. Comput. **116**, 1 (2022).
[54] S. Chen, Y.-H. He, E. Hirst, A. Nestor, and A. Zahabi, Mahler measuring the genetic code of amoebae, arXiv:2212.06553.
[55] R.-K. Seong, Unsupervised machine learning techniques for exploring tropical coamoeba, brane tilings and Seiberg duality, Phys. Rev. D **108**, 106009 (2023).
[56] J. Halverson, B. Nelson, and F. Ruehle, Branes with brains: Exploring string vacua with deep reinforcement learning, J. High Energy Phys. 06 (2019) 003.
[57] G. J. Loges and G. Shiu, Breeding realistic D-brane models, Fortschr. Phys. **70**, 2200038 (2022).
[58] G. Arias-Tamargo, Y.-H. He, E. Heyes, E. Hirst, and D. Rodriguez-Gomez, Brain webs for brane webs, Phys. Lett. B **833**, 137376 (2022).
[59] G. J. Loges and G. Shiu, 134 billion intersecting brane models, J. High Energy Phys. 12 (2022) 097.
[60] H.-Y. Chen, Y.-H. He, S. Lal, and M. Z. Zaz, Machine learning etudes in conformal field theories, arXiv:2006.16114.
[61] G. Kántor, V. Niarchos, and C. Papageorgakis, Solving conformal field theories with artificial intelligence, Phys. Rev. Lett. **128**, 041601 (2022).
[62] G. Kántor, V. Niarchos, and C. Papageorgakis, Conformal bootstrap with reinforcement learning, Phys. Rev. D **105**, 025018 (2022).
[63] G. Kántor, V. Niarchos, C. Papageorgakis, and P. Richmond, 6D (2,0) bootstrap with the soft-actor-critic algorithm, Phys. Rev. D **107**, 025005 (2023).
[64] V. Niarchos, C. Papageorgakis, P. Richmond, A. G. Stapleton, and M. Woolley, Bootstrability in line-defect CFT with improved truncation methods, Phys. Rev. D **108**, 105027 (2023).
[65] J. Bao, S. Franco, Y.-H. He, E. Hirst, G. Musiker, and Y. Xiao, Quiver mutations, Seiberg duality and machine learning, Phys. Rev. D **102**, 086013 (2020).
[66] P.-P. Dechant, Y.-H. He, E. Heyes, and E. Hirst, Cluster algebras: Network science and machine learning, J. Comput. Algebra **8** (2023).
[67] M.-W. Cheung, P.-P. Dechant, Y.-H. He, E. Heyes, E. Hirst, and J.-R. Li, Clustering cluster algebras with clusters, arXiv:2212.09771.
[68] S. Chen, P.-P. Dechant, Y.-H. He, E. Heyes, E. Hirst, and D. Riabchenko, Machine learning clifford invariants of ADE coxeter elements, arXiv:2310.00041.
[69] S. Abel and J. Rizos, Genetic algorithms and the search for viable string vacua, J. High Energy Phys. 08 (2014) 010.
[70] M. Bies, M. Cvetič, R. Donagi, L. Lin, M. Liu, and F. Ruehle, Machine learning and algebraic approaches towards complete matter spectra in 4d F-theory, J. High Energy Phys. 01 (2021) 196.
[71] S. Krippendorf, R. Kroepsch, and M. Syvaeri, Revealing systematics in phenomenologically viable flux vacua with reinforcement learning, arXiv:2107.04039.
[72] A. Constantin, T. R. Harvey, and A. Lukas, Heterotic string model building with monad bundles and reinforcement learning, Fortschr. Phys. **70–3**, 2100186 (2022).
[73] S. Abel, A. Constantin, T. R. Harvey, and A. Lukas, Evolving heterotic gauge backgrounds: Genetic algorithms versus reinforcement learning, Fortschr. Phys. **70**, 2200034 (2022).
[74] D. Berman, T. Fischbacher, G. Inverso, B. Scellier, and B. Scellier, Vacua of ω-deformed SO(8) supergravity, J. High Energy Phys. 06 (2022) 133.
[75] S. A. Abel, A. Constantin, T. R. Harvey, A. Lukas, and L. A. Nutricati, Decoding nature with nature's tools: Heterotic line bundle models of particle physics with genetic algorithms and quantum annealing, Fortschr. Phys. **72**, 2300260 (2024).
[76] A. Dubey, S. Krippendorf, and A. Schachner, JAXVacua—A framework for sampling string vacua, J. High Energy Phys. 12 (2023) 146.
[77] Y.-H. He, E. Hirst, and T. Peterken, Machine-learning dessins d'enfants: Explorations via modular and Seiberg-Witten curves, J. Phys. A **54**, 075401 (2021).
[78] J. Bao, Y.-H. He, E. Hirst, J. Hofscheier, A. Kasprzyk, and S. Majumder, Hilbert series, machine learning, and applications to physics, Phys. Lett. B **827**, 136966 (2022).
[79] J. Bao, Y.-H. He, E. Hirst, J. Hofscheier, A. Kasprzyk, and S. Majumder, Polytopes and machine learning, Math. Sci. Hum. **01**, 181 (2023).
[80] X. Gao and H. Zou, Applying machine learning to the Calabi-Yau orientifolds with string vacua, Phys. Rev. D **105**, 046017 (2022).
[81] T. Coates, J. Hofscheier, and A. Kasprzyk, Machine learning the dimension of a polytope, arXiv:2207.07717.
[82] T. Coates, A. M. Kasprzyk, and S. Veneziale, Machine learning the dimension of a Fano variety, Nat. Commun. **14**, 5526 (2023).

[83] T. Coates, A. M. Kasprzyk, and S. Veneziale, Machine learning detects terminal singularities, arXiv:2310.20458.

[84] M. Manko, An upper bound on the critical volume in a class of toric Sasaki-Einstein manifolds, arXiv:2209.14029.

[85] E. Choi and R.-K. Seong, Machine learning regularization for the minimum volume formula of toric Calabi-Yau 3-folds, Phys. Rev. D **109**, 046015 (2024).

[86] C. Vafa, Evidence for F-theory, Nucl. Phys. **B469**, 403 (1996).

[87] A. Klemm, B. Lian, S.-S. Roan, and S.-T. Yau, Calabi-Yau four-folds for M- and F-theory compactifications, Nucl. Phys. **B518**, 515 (1998).

[88] S. Gukov, C. Vafa, and E. Witten, CFT's from Calabi-Yau four folds, Nucl. Phys. **B584**, 69 (2000); **B608**, 477(E) (2001).

[89] R. Donagi and M. Wijnholt, Model building with F-theory, arXiv:0802.2969.

[90] P. Candelas, M. Lynker, and R. Schimmrigk, Calabi-Yau manifolds in weighted P4, Nucl. Phys. **B341**, 383 (1990).

[91] A. Klemm and R. Schimmrigk, Landau-Ginzburg string vacua, Nucl. Phys. **B411**, 559 (1994).

[92] M. Kreuzer and H. Skarke, No mirror symmetry in Landau-Ginzburg spectra!, Nucl. Phys. **B388**, 113 (1992).

[93] M. Kreuzer and H. Skarke, On the classification of quasihomogeneous functions, Commun. Math. Phys. **150**, 137 (1992).

[94] C. Vafa and N. P. Warner, Catastrophes and the classification of conformal theories, Phys. Lett. B **218**, 51 (1989).

[95] E. Witten, Phases of $N = 2$ theories in two-dimensions, Nucl. Phys. **B403**, 159 (1993).

[96] A. M. Kasprzyk, Bounds on fake weighted projective space, Kodai Math. J. **32**, 197 (2009).

[97] A. Hanany and K. D. Kennaway, Dimer models and toric diagrams, arXiv:hep-th/0503149.

[98] A. M. Kasprzyk, Canonical toric Fano threefolds, Can. J. Math. **62**, 1293 (2010).

[99] D. Cox, J. Little, and H. Schenck, *Toric Varieties*, Graduate Studies in Mathematics (American Mathematical Society, Providence, 2011).

[100] V. V. Batyrev, Dual polyhedra and mirror symmetry for Calabi-Yau hypersurfaces in toric varieties, J. Alg. Geom. **3**, 493 (1994).

[101] P. Candelas, X. de la Ossa, and S. H. Katz, Mirror symmetry for Calabi-Yau hypersurfaces in weighted P^4 and extensions of Landau-Ginzburg theory, Nucl. Phys. **B450**, 267 (1995).

[102] V. Bouchard, Lectures on complex geometry, Calabi-Yau manifolds and toric geometry, arXiv:hep-th/0702063.

[103] V. Batyrev and K. Schaller, Mirror symmetry for quasi-smooth Calabi-Yau hypersurfaces in weighted projective spaces, J. Geom. Phys. **164**, 104198 (2021).

[104] H. Skarke, Weight systems for toric Calabi-Yau varieties and reflexivity of Newton polyhedra, Mod. Phys. Lett. A **11**, 1637 (1996).

[105] J. Bao, Y.-H. He, E. Hirst, and S. Pietromonaco, Lectures on the Calabi-Yau landscape, arXiv:2001.01212.

[106] T. Hübsch, *Calabi-Yau Manifolds: A Bestiary for Physicists*, G—Reference, Information and Interdisciplinary Subjects Series (World Scientific, Singapore, 1994), https://books.google.co.uk/books?id=bTRqDQAAQBAJ.

[107] S. Sethi, C. Vafa, and E. Witten, Constraints on low-dimensional string compactifications, Nucl. Phys. **B480-2**, 213 (1996).

[108] A. Chandra, A. Constantin, C. S. Fraser-Taliente, T. R. Harvey, and A. Lukas, Enumerating Calabi-Yau manifolds: Placing bounds on the number of diffeomorphism classes in the Kreuzer-Skarke list, arXiv:2310.05909.

[109] N. Gendler, N. MacFadden, L. McAllister, J. Moritz, R. Nally, A. Schachner, and M. Stillman, Counting Calabi-Yau threefolds, arXiv:2310.06820.

[110] C. Vafa, String vacua and orbifoldized L-G models, Mod. Phys. Lett. A **04**, 1169 (1989).

[111] V. V. Batyrev, On the stringy Hodge numbers of mirrors of quasi-smooth Calabi-Yau hypersurfaces, arXiv:2006.15825.

[112] Y.-H. He, V. Jejjala, and L. Pontiggia, Patterns in Calabi-Yau distributions, Commun. Math. Phys. **354**, 477 (2017).

[113] A. Ashmore and Y.-H. He, Calabi-Yau three-folds: Poincaré polynomials and fractals, in *Strings, Gauge Fields, and the Geometry Behind* (World Scientific, Singapore, 2012), pp. 173–186. 10.1142%2F9789814412551_0007.

[114] M. Kreuzer and H. Skarke, On the classification of reflexive polyhedra, Commun. Math. Phys. **185**, 495 (1997).

[115] M. Kreuzer and H. Skarke, PALP: A package for analysing lattice polytopes with applications to toric geometry, Comput. Phys. Commun. **157**, 87 (2004).

[116] A. Testolin, Can neural networks do arithmetic? A survey on the elementary numerical skills of state-of-the-art deep learning models, Appl. Sci. **14**, 744 (2024).

[117] A. S. Haupt, A. Lukas, and K. S. Stelle, M-theory on Calabi-Yau five-folds, J. High Energy Phys. 05 (2009) 069.

[118] S. Schäfer-Nameki and T. Weigand, F-theory and 2d (0,2) theories, J. High Energy Phys. 05 (2016) 059.

[119] J. Tian and Y.-N. Wang, Elliptic Calabi-Yau fivefolds and 2d (0,2) F-theory landscape, J. High Energy Phys. 03 (2021) 069.

[120] G. Curio and D. Lust, New $N = 1$ supersymmetric three-dimensional superstring vacua from U manifolds, Phys. Lett. B **428**, 95 (1998).

[121] M. Kreuzer and H. Skarke, Calabi-Yau data, http://hep.itp.tuwien.ac.at/kreuzer/CY/.

[122] A. S. Haupt, A. Lukas, and K. Stelle, M-theory on Calabi-Yau five-folds, J. High Energy Phys. 05 (2009) 069.

[123] V. Dumachev, Complete intersection Calabi-Yau six-folds, Appl. Math. Sci. **9**, 7121 (2015).