

MACHINE LEARNING DETECTS TERMINAL SINGULARITIES

TOM COATES , ALEXANDER M. KASPRZYK , AND SARA VENEZIALE 

ABSTRACT. Algebraic varieties are the geometric shapes defined by systems of polynomial equations; they are ubiquitous across mathematics and science. Amongst these algebraic varieties are \mathbb{Q} -Fano varieties: positively curved shapes which have \mathbb{Q} -factorial terminal singularities. \mathbb{Q} -Fano varieties are of fundamental importance in geometry as they are ‘atomic pieces’ of more complex shapes – the process of breaking a shape into simpler pieces in this sense is called the Minimal Model Programme.

Despite their importance, the classification of \mathbb{Q} -Fano varieties remains unknown. In this paper we demonstrate that machine learning can be used to understand this classification. We focus on eight-dimensional positively-curved algebraic varieties that have toric symmetry and Picard rank two, and develop a neural network classifier that predicts with 95% accuracy whether or not such an algebraic variety is \mathbb{Q} -Fano. We use this to give a first sketch of the landscape of \mathbb{Q} -Fano varieties in dimension eight.

How the neural network is able to detect \mathbb{Q} -Fano varieties with such accuracy remains mysterious, and hints at some deep mathematical theory waiting to be uncovered. Furthermore, when visualised using the quantum period, an invariant that has played an important role in recent theoretical developments, we observe that the classification as revealed by ML appears to fall within a bounded region, and is stratified by the Fano index. This suggests that it may be possible to state and prove conjectures on completeness in the future.

Inspired by the ML analysis, we formulate and prove a new global combinatorial criterion for a positively curved toric variety of Picard rank two to have terminal singularities. Together with the first sketch of the landscape of \mathbb{Q} -Fano varieties in higher dimensions, this gives strong new evidence that machine learning can be an essential tool in developing mathematical conjectures and accelerating theoretical discovery.

1. INTRODUCTION

Systems of polynomial equations occur throughout mathematics and science; see e.g. [4, 23, 25, 43]. Solutions of these systems define shapes called *algebraic varieties*. Depending on the equations involved, algebraic varieties can be smooth (as in Figure 1(a)) or have singularities (as in Figures 1(b) and 1(c)). In this paper we show that machine learning methods can detect a class of singularities called *terminal singularities*.

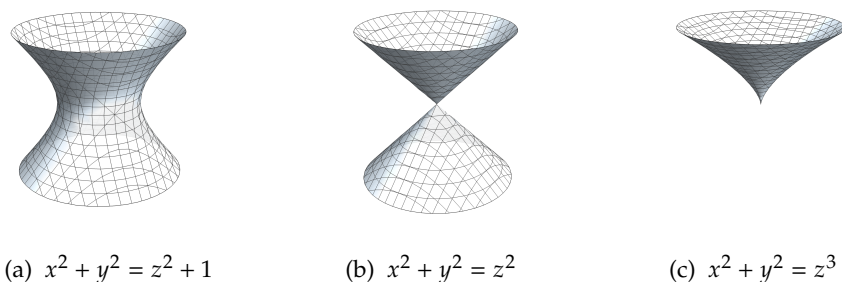


Figure 1. Algebraic varieties in \mathbb{R}^3 with different defining equations.

A key class of algebraic varieties are *Fano varieties*: positively curved shapes that are basic building blocks in algebraic geometry. Fano varieties are ‘atomic pieces’ of more complex shapes, in the sense of the Minimal Model Programme [11, 33, 35]. Running the Minimal Model Programme – that is, breaking an algebraic variety X into atomic pieces – involves making birational transformations of X . These are

2020 *Mathematics Subject Classification.* 14J45 (Primary); 68T07 (Secondary).

Key words and phrases. Fano varieties, terminal singularities, machine learning.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

modifications on subsets with zero volume (and codimension at least one), and can either introduce or remove singularities. The building blocks that emerge from this process are not necessarily smooth: they satisfy a weaker condition called \mathbb{Q} -factoriality,¹ and can have mild singularities called terminal singularities [48]. Fano varieties that are \mathbb{Q} -factorial and have terminal singularities are called \mathbb{Q} -Fano varieties.

The classification of \mathbb{Q} -Fano varieties is therefore a long-standing problem of great importance [6, 20, 34, 41, 42] – one can think of this as building a Periodic Table for geometry. But, despite more than a century of study, very little is known. In what follows we exploit the fact that machine learning can detect terminal singularities to give the first sketch of part of the classification of higher-dimensional \mathbb{Q} -Fano varieties.

We probe the classification of \mathbb{Q} -Fano varieties using a class of highly-symmetrical shapes called *toric varieties*. (For example, the algebraic varieties pictured in Figure 1 are toric varieties.) Toric varieties are particularly suitable for computation and machine learning, because their geometric properties are encoded by simple combinatorial objects. We consider Fano toric varieties of Picard rank two. These can be encoded using a $2 \times N$ matrix of non-negative integers called the *weight matrix*; here the dimension of the toric variety is $N - 2$.

To determine whether such a toric variety X is a \mathbb{Q} -Fano variety we need to check whether X is \mathbb{Q} -factorial, and whether the singularities of X are terminal. Checking \mathbb{Q} -factoriality from the weight matrix of X turns out to be straightforward (see §3) but checking terminality is extremely challenging. This is because there is no satisfactory theoretical understanding of the problem. We lack a global criterion for detecting terminality in terms of weight data (such as [32] in a simpler setting) and so have to fall back on first enumerating all the singularities to analyse, and then checking terminality for each singularity. Each step is a challenging problem in discrete geometry: the first step involves building a different combinatorial object associated to the n -dimensional toric variety X , which is a collection of cones in \mathbb{R}^n called the *fan* $\Sigma(X)$; the second step involves checking for various cones in the fan whether or not they contain lattice points on or below a certain hyperplane. To give a sense of the difficulty of the computations involved, generating and post-processing our dataset of 10 million toric varieties in dimension eight took around 30 CPU years.

To overcome this difficulty, and hence to begin to investigate the classification of \mathbb{Q} -Fano varieties in dimension eight, we used supervised machine learning. We trained a feed-forward neural network classifier on a balanced dataset of 5 million examples; these are eight-dimensional \mathbb{Q} -factorial Fano toric varieties of Picard rank two, of which 2.5 million are terminal and 2.5 million non-terminal. Testing on a further balanced dataset of 5 million examples showed that the neural network classifies such toric varieties as terminal or non-terminal with an accuracy of 95%. This high accuracy allowed us to rapidly generate many additional examples that are with high probability \mathbb{Q} -Fano varieties – that is, examples that the classifier predicts have terminal singularities. This ML-assisted generation step is much more efficient: generating 100 million examples in dimension eight took less than 120 CPU hours.

The fact that the ML classifier can detect terminal singularities with such high accuracy suggests that there is new mathematics waiting to be discovered here – there should be a simple criterion in terms of the weight matrix to determine whether or not a toric variety X has terminal singularities. In §5 we take the first steps in this direction, giving in Algorithm 1 a new method to check terminality directly from the weight matrix, for toric varieties of Picard rank two. A proof of correctness is given in §E. This new algorithm is fifteen times faster than the naïve approach that we used to generate our labelled dataset, but still several orders of magnitude slower than the neural network classifier. We believe that this is not the end of the story, and that the ML results suggest that a simpler criterion exists. Note that the neural network classifier cannot be doing anything analogous to Algorithm 1: the algorithm relies on divisibility relations between entries of the weight matrix (GCDs etc.) that are

¹An algebraic variety X is \mathbb{Q} -factorial if it is normal and, in addition, for each rank-one reflexive sheaf E on X , some tensor power of E is a line bundle. This implies that the dimension of the singular locus in X is at most $\dim X - 2$, and that some tensor power of the canonical sheaf (of top-degree differential forms) is a line bundle.

not visible to the neural network, as they are destroyed by the rescaling and standardisation that is applied to the weights before they are fed to the classifier.

In §6 we use the ML-assisted dataset of 100 million examples to begin to explore the classification of \mathbb{Q} -Fano varieties in dimension eight. We visualise the dataset using the *regularized quantum period*, an invariant that has played an important role in recent theoretical work on \mathbb{Q} -Fano classification, discovering that an appropriate projection of the data appears to fill out a wedge-shaped region bounded by two straight lines. This visualisation suggests some simple patterns in the classification: for example, the distance from one edge of the wedge appears to be determined by the Fano index of the variety.

Our work is further evidence that machine learning can be an indispensable tool for generating and guiding mathematical understanding. The neural network classifier led directly to Algorithm 1, a new theoretical result, by revealing that the classification problem was tractable and thus there was probably new mathematics waiting to be found. This is part of a new wave of application of artificial intelligence to pure mathematics [15, 19, 22, 27, 49–51], where machine learning methods drive theorem discovery.

A genuinely novel contribution here, though, is the use of machine learning for data generation and data exploration in pure mathematics. Sketching the landscape of higher-dimensional \mathbb{Q} -Fano varieties using traditional methods would be impossible with the current theoretical understanding, and prohibitively expensive using the current exact algorithms. Training a neural network classifier however, allows us to explore this landscape easily – a landscape that is unreachable with current mathematical tools.

Why dimension eight? We chose to work with eight-dimensional varieties for several reasons. It is important to distance ourselves from the surface case (dimension two), where terminality is a trivial condition. A two-dimensional algebraic variety has terminal singularities if and only if it is smooth. On the other hand, we should consider a dimension where we can generate a sufficient amount of data for machine learning (the analogue of our dataset in dimension three, for example, contains only 34 examples [31]) and where we can generate enough data to meaningfully probe the classification. Moreover, we work in Picard rank two because there already exists a fast combinatorial formula to check terminality in rank one [32]; Picard rank two is the next natural case to consider.

2. MATHEMATICAL BACKGROUND

The prototypical example of a Fano variety is projective space \mathbb{P}^{N-1} , which can be thought of as the quotient of $\mathbb{C}^N \setminus \{0\}$ by \mathbb{C}^\times acting as follows:

$$\lambda \cdot (z_1, \dots, z_N) = (\lambda z_1, \dots, \lambda z_N)$$

Fano toric varieties of Picard rank two arise similarly. They can be constructed as the quotient of $\mathbb{C}^N \setminus S$, where S is a union of subspaces, by an action of $(\mathbb{C}^\times)^2$. This action, and the union of subspaces S , is encoded by a weight matrix:

$$\begin{bmatrix} a_1 & \cdots & a_N \\ b_1 & \cdots & b_N \end{bmatrix} \quad (2.1)$$

Here we assume that all $(a_i, b_i) \in \mathbb{Z}^2 \setminus \{0\}$ lie in a strictly convex cone $C \subset \mathbb{R}^2$. The action is

$$(\lambda, \mu) \cdot (z_1, \dots, z_N) = (\lambda^{a_1} \mu^{b_1} z_1, \dots, \lambda^{a_N} \mu^{b_N} z_N)$$

and $S = S_+ \cup S_-$ is the union of subspaces S_+ and S_- , where

$$\begin{aligned} S_+ &= \{(z_1, \dots, z_N) \mid z_i = 0 \text{ if } b_i/a_i > b/a\} \\ S_- &= \{(z_1, \dots, z_N) \mid z_i = 0 \text{ if } b_i/a_i < b/a\} \end{aligned} \quad (2.2)$$

and $a = \sum_{i=1}^N a_i$, $b = \sum_{i=1}^N b_i$: see [8]. The quotient $X = (\mathbb{C}^N \setminus S)/(\mathbb{C}^\times)^2$ is an algebraic variety of dimension $N - 2$. We assume in addition that both S_+ and S_- have dimension at least two; this implies that the second Betti number of X is two, that is, X has Picard rank two.

Since we have insisted that all columns (a_i, b_i) lie in a strictly convex cone C , we can always permute columns and apply an $\mathrm{SL}_2(\mathbb{Z})$ transformation to the weight matrix to obtain a matrix in standard form:

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_N \\ 0 & b_2 & \cdots & b_N \end{bmatrix} \quad (2.3)$$

where all entries are non-negative, the columns are cyclically ordered anticlockwise, and $a_N < b_N$. This transformation corresponds to renumbering the co-ordinates of \mathbb{C}^N and reparametrising the torus $(\mathbb{C}^\times)^2$ that acts, and consequently leaves the quotient variety X that we construct unchanged.

We will consider weight matrices (2.1) that satisfy an additional condition called being *well-formed*. An $r \times N$ weight matrix is called standard if the greatest common divisor of its $r \times r$ minors is one, and is well-formed if every submatrix formed by deleting a column is standard [2]. Considering only well-formed weight matrices guarantees that a toric variety determines and is determined by its weight matrix, uniquely up to $\mathrm{SL}_r(\mathbb{Z})$ -transformation.

Testing terminality. As mentioned in the introduction, an n -dimensional toric variety X determines a collection $\Sigma(X)$ of cones in \mathbb{R}^n called the fan of X . A toric variety is completely determined by its fan. The process of determining the fan $\Sigma(X)$ from the weight matrix (2.1) is explained in §A; this is a challenging combinatorial calculation. In the fan $\Sigma(X)$, the one-dimensional cones are called rays. For a Fano toric variety X , taking the convex hull of the first lattice point on each ray defines a convex polytope P , and X has terminal singularities if and only if the only lattice points in P are the origin and the vertices. Verifying this is a conceptually straightforward but computationally challenging calculation in integer linear programming.

3. DATA GENERATION

We generated a balanced, labelled dataset of ten million \mathbb{Q} -factorial Fano toric varieties of Picard rank two and dimension eight. These varieties are encoded, as described above, by weight matrices. We generated 2×10 integer-valued matrices in standard form, as in (2.3), with entries chosen uniformly at random from the set $\{0, \dots, 7\}$. Minor exceptions to this were the values for a_1 and b_N , which were both chosen uniformly at random from the set $\{1, \dots, 7\}$, and the value for a_N , which was chosen uniformly at random from the set $\{0, \dots, b_N - 1\}$. Once a random weight matrix was generated, we retained it only if it satisfied:

- (1) None of the columns are the zero vector.
- (2) The sum of the columns is not a multiple of any of them.
- (3) The subspaces S_+ and S_- in (2.2) are both of dimension at least two.
- (4) The matrix is well-formed.

The first condition here was part of our definition of weight matrix; the second condition is equivalent to X being \mathbb{Q} -factorial; the third condition guarantees that X has Picard rank two; and the fourth condition was discussed above.

We used rejection sampling to ensure that the dataset contains an equal number of terminal and non-terminal examples. Before generating any weight matrix, a boolean value was set to True (terminal) or False (non-terminal). Once a random weight matrix that satisfied conditions (1)–(4) above was generated, we checked if the corresponding toric variety was terminal using the method discussed in §2. If the terminality check agreed with the chosen boolean, the weight matrix was added to our dataset; otherwise the generation step was repeated until a match was found.

As discussed, different weight matrices can give rise to the same toric variety. Up to isomorphism, however, a toric variety X is determined by the isomorphism class of its fan. We deduplicated our dataset by placing the corresponding fan $\Sigma(X)$, which we had already computed in order to test for terminality, in normal form [26, 37]. In practice, very few duplicates occurred.

4. BUILDING THE MACHINE LEARNING MODEL

We built a neural network classifier to determine whether a \mathbb{Q} -factorial Fano variety of Picard rank two and dimension eight is terminal. The network was trained on the features given by concatenating the two rows of a weight matrix, $[a_1, \dots, a_{10}, b_1, \dots, b_{10}]$. The features were standardised by translating

Hyperparameter	Value	Hyperparameter	Value
Layers	(512, 768, 512)	Momentum	0.99
Batch size	128	LeakyRelu slope	0.01
Initial learning rate	0.01		

Table 1. Final network architecture and configuration.

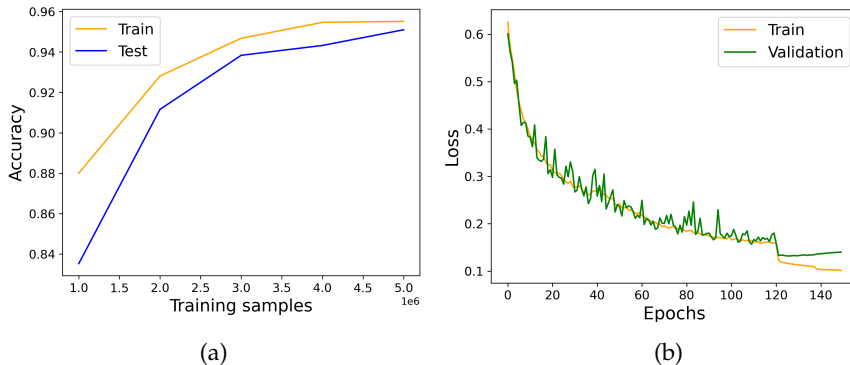


Figure 2. (a) Accuracy for different train-test splits; (b) epochs against loss for the network trained on 5M samples.

their mean to zero and scaling to variance one. The network, a multilayer perceptron, is a fully connected feedforward neural network with three hidden layers and leaky ReLU activation function. It was trained on the dataset described in §3 using binary cross-entropy as loss function, stochastic mini-batch gradient descent optimiser and using early-stopping, for a maximum of 150 epochs and with learning rate reduction on plateaux. We tested the model on a balanced subset of 50% of the data (5M); the remainder was used for training (40%; 4M balanced) and validation (10%; 1M).

Hyperparameter tuning was partly carried out using RayTune [39] on a small portion of the training data, via random grid search with Async Successive Halving Algorithm (ASHA) scheduler [38], for 100 experiments. Given the best configuration resulting from the random grid search, we then manually explored nearby configurations and took the best performing one. The final best network configuration is summarised in Table 1.

By trying different train-test splits, and using 20% of the training data for validation throughout, we obtained the learning curve in Figure 2(a). This shows that a train-validate-test split of 4M-1M-5M produced an accurate model that did not overfit. Training this model gave the loss learning curve in Figure 2(b), and a final accuracy (on the test split of size 5M) of 95%.

5. THEORETICAL RESULT

The high accuracy of the model in §4 was very surprising. As explained in the introduction, \mathbb{Q} -Fano varieties are of fundamental importance in algebraic geometry. However, asking whether a Fano variety has terminal singularities is, in general, an extremely challenging geometric question. In the case of a Fano toric variety one would typically proceed by constructing the fan, and then performing a cone-by-cone analysis of the combinatorics. This is computationally expensive and unsatisfying from a theoretical viewpoint. The success of the model suggested that a more direct characterisation is possible from the weight matrix alone. An analogous characterisation exists in the simpler case of weighted projective spaces [32], which have Picard rank one, however no such result in higher Picard rank was known prior to training this model.

Inspired by this we prove a theoretical result, Proposition 3, which leads to a new algorithm for checking terminality directly from the weight matrix, for \mathbb{Q} -factorial Fano toric varieties of Picard

rank two. Consider a weight matrix as in (2.1) that satisfies conditions (1)–(4) from §3, and the toric variety X that it determines. As discussed in §2, and explained in detail in §A, X determines a convex polytope P in \mathbb{R}^{N-2} , with N vertices given by the first lattice points on the N rays of the fan. Each of the vertices of P is a lattice point (i.e., lies in $\mathbb{Z}^{N-2} \subset \mathbb{R}^{N-2}$), and X has terminal singularities if and only if the only lattice points in P are the vertices e_1, \dots, e_N and the origin.

Definition 1. Let Δ_i denote the simplex in \mathbb{R}^{N-2} with vertices $e_1, \dots, \hat{e}_i, \dots, e_N$ where e_i is omitted. We say that Δ_i is *mostly empty* if each lattice point in Δ_i is either a vertex or the origin.

Notation 2. Let $\{x\}$ denote the fractional part $x - \lfloor x \rfloor$ of a rational number x .

Proposition 3. Consider a weight matrix

$$\begin{bmatrix} a_1 & \cdots & a_N \\ b_1 & \cdots & b_N \end{bmatrix}$$

that satisfies conditions (1)–(4) from §3. Let $g_i = \gcd\{a_i, b_i\}$, and let A_i, B_i be integers such that $A_i a_i + B_i b_i = g_i$. Set

$$\begin{aligned} \alpha_i^j &= \frac{a_j b_i - b_j a_i}{g_i} & \alpha_i &= \sum_{j=1}^N \alpha_i^j \\ \beta_i^j &= -A_i a_j - B_i b_j & \beta_i &= \sum_{j=1}^N \beta_i^j & f_i &= \frac{\alpha_i g_i}{\gcd\{g_i, \beta_i\}} \end{aligned}$$

noting that all these quantities are integers. Then Δ_i is mostly empty if and only if for all $k \in \{0, \dots, f_i - 1\}$ and $l \in \{0, \dots, g_i - 1\}$ such that

$$\sum_{j=1}^N \left\{ k \frac{\alpha_i^j}{f_i} + l \frac{\beta_i^j}{g_i} \right\} = 1$$

we have that

$$\left\{ k \frac{\alpha_i^j}{f_i} + l \frac{\beta_i^j}{g_i} \right\} = \left\{ \frac{\alpha_i^j}{\alpha_i} \right\}$$

for all j .

Let $s_+ = \{i \mid a_i b - b_i a > 0\}$, $s_- = \{i \mid a_i b - b_i a < 0\}$, and let I be either s_+ or s_- . Then Δ_i , $i \in I$, forms a triangulation of P . Thus X has terminal singularities if and only if Δ_i is mostly empty for each $i \in I$. This leads to Algorithm 1.

Algorithm 1 Test terminality for weight matrix $W = [[a_1, \dots, a_N], [b_1, \dots, b_N]]$.

- 1: Set $a = \sum_{i=1}^N a_i$, $b = \sum_{i=1}^N b_i$.
 - 2: Set $s_+ = \{i \mid a_i b - b_i a > 0\}$ and $s_- = \{i \mid a_i b - b_i a < 0\}$.
 - 3: Set I to be the smaller of s_+ and s_- .
 - 4: **for** $i \in I$ **do**
 - 5: Test if Δ_i is mostly empty, using Proposition 3.
 - 6: **if** Δ_i is not mostly empty **then**
 - 7: return False.
 - 8: **end if**
 - 9: **end for**
 - 10: return True.
-

Comparisons. Testing on 100 000 randomly-chosen examples indicates that Algorithm 1 is approximately 15 times faster than the fan-based approach to checking terminality that we used when labelling our dataset (0.020s per weight matrix for Algorithm 1 versus 0.305s for the standard approach implemented in Magma). On single examples, the neural network classifier is approximately 30 times faster than Algorithm 1. The neural network also benefits greatly from batching, whereas the other two algorithms do not: for batches of size 10 000, the neural network is roughly 2000 times faster than Algorithm 1.

6. THE TERMINAL TORIC FANO LANDSCAPE

Having trained the terminality classifier, we used it to explore the landscape of \mathbb{Q} -Fano toric varieties with Picard rank two. To do so, we built a large dataset of examples and analysed their *regularized quantum period*, a numerical invariant of \mathbb{Q} -Fano varieties [12]. For smooth low-dimensional Fano varieties, it is known that the regularized quantum period is a complete invariant [13]. This is believed to be true in higher dimension, but is still conjectural. Given a \mathbb{Q} -Fano variety X , its regularized quantum period is a power series

$$\hat{G}_X(t) = \sum_{d=0}^{\infty} c_d t^d$$

where $c_0 = 1$, $c_1 = 0$, $c_d = d! r_d$, and r_d is the number of degree- d rational curves in X that satisfy certain geometric conditions. Formally speaking, r_d is a degree- d , genus-zero Gromov–Witten invariant [36]. The *period sequence* of X is the sequence (c_d) of coefficients of the regularized quantum period. This sequence grows rapidly. In the case where X is a \mathbb{Q} -Fano toric variety of Picard rank two, rigorous asymptotics for this growth are known.

Theorem 4 (Theorem 5.2, [15]). *Consider a weight matrix*

$$\begin{bmatrix} a_1 & \dots & a_N \\ b_1 & \dots & b_N \end{bmatrix}$$

for a \mathbb{Q} -factorial Fano toric variety X of Picard rank two. Let $a = \sum_{i=1}^N a_i$ and $b = \sum_{i=1}^N b_i$, and let $[\mu : \nu] \in \mathbb{P}^1$ be the unique real root of the homogeneous polynomial

$$\prod_{i=1}^N (a_i \mu + b_i \nu)^{a_i b} - \prod_{i=1}^N (a_i \mu + b_i \nu)^{b_i a} \quad (6.1)$$

such that $a_i \mu + b_i \nu \geq 0$ for all $i \in \{1, 2, \dots, N\}$. Let (c_d) be the corresponding period sequence. Then non-zero coefficients c_d satisfy

$$\log c_d \sim Ad - \frac{\dim X}{2} \log d + B$$

as $d \rightarrow \infty$, where

$$\begin{aligned} A &= - \sum_{i=1}^N p_i \log p_i \\ B &= - \frac{\dim X}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^N \log p_i - \frac{1}{2} \log \left(\sum_{i=1}^N \frac{(a_i b - b_i a)^2}{\ell^2 p_i} \right) \end{aligned} \quad (6.2)$$

Here $p_i = \frac{\mu a_i + \nu b_i}{\mu a + \nu b}$, so that $\sum_i p_i = 1$, and $\ell = \gcd\{a, b\}$ is the Fano index.

In Figure 3 we picture our dataset of \mathbb{Q} -Fano varieties by using the coefficients A and B to project it to \mathbb{R}^2 ; for the corresponding images for terminal Fano weighted projective spaces, see [15, Figure 7a]. Note the stratification by Fano index. Although many weight matrices can give rise to the same toric variety, in our context we are using well-formed weight matrices in standard form (2.3) and so at most two weight matrices can give rise to the same toric variety. We removed any such duplicates from our dataset, so the heatmap in Figure 3(b) reflects genuine variation in the distribution of \mathbb{Q} -Fano varieties, rather than simply the many-to-one correspondence between weight matrices and toric varieties.

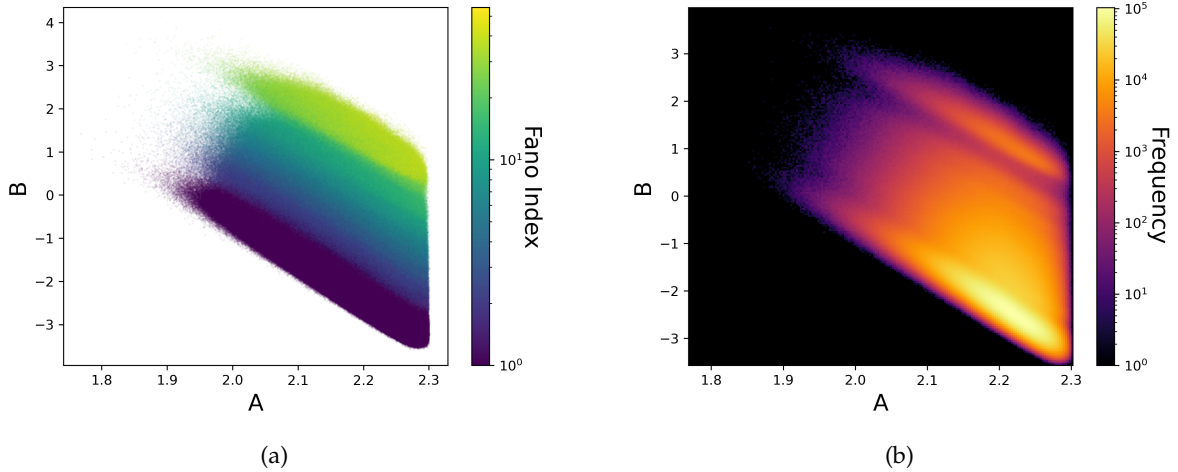


Figure 3. A dataset of 100M probably- \mathbb{Q} -Fano toric varieties of Picard rank two and dimension eight, projected to \mathbb{R}^2 using the growth coefficients A and B from (6.2). In (a) we colour by Fano index, while in (b) we colour a heatmap according to the frequency.

Data generation. The dataset pictured in Figure 3 was generated using an AI-assisted data generation workflow that combines algorithmic checks and our machine learning model, as follows.

- Generate a random 2×10 matrix with entries chosen uniformly from $\{0, 1, 2, 3, 4, 5, 6, 7\}$.
- Cyclically order the columns and only keep the matrix if it is in standard form, as in (2.3).
- Check conditions (1)–(4) from §3.
- Predict terminality using the neural network classifier from §4, only keeping examples that are classified as terminal and storing their probabilities.
- Set $\mu = 1$ in (6.1) and solve the univariate real polynomial in the correct domain to obtain the solution $(1, \nu)$.
- Calculate the coefficients A and B using the formulae in (6.2).

The final dataset is composed of 100M samples. Each of these represents a \mathbb{Q} -factorial toric Fano variety of dimension eight and Picard rank two that the classifier predicts is a \mathbb{Q} -Fano variety.

Data analysis. We note that the vertical boundary in Figure 3 is not a surprise. In fact, we can apply the log-sum inequality to the formula for A to obtain

$$A = - \sum_{i=1}^N p_i \log(p_i) \leq - \left(\sum_{i=1}^N p_i \right) \log \left(\frac{\sum_{i=1}^N p_i}{N} \right) = \log(N)$$

In our case $N = 10$, and the vertical boundary that we see in Figure 3(a) is the line $x = \log(10) \sim 2.3$. We also see what looks like a linear lower bound for the cluster; a similar bound was observed, and established rigorously, for weighted projective spaces in [15].

Closer analysis (see §B) reveals large overlapping clusters that correspond to Fano varieties of different Fano index. Furthermore the simplest toric varieties of Picard rank two – products of projective spaces, and products of weighted projective spaces – appear to lie in specific regions of the diagram.

7. LIMITATIONS AND FUTURE DIRECTIONS

The main message of this work is a new proposed AI-assisted workflow for data generation in pure mathematics. This allowed us to construct, for the first time, an approximate landscape of objects of mathematical interest (\mathbb{Q} -Fano varieties) which is inaccessible by traditional methods. We hope that this methodology will have broad application, especially to other large-scale classification questions in mathematics, of which there are many [1, 18, 28].

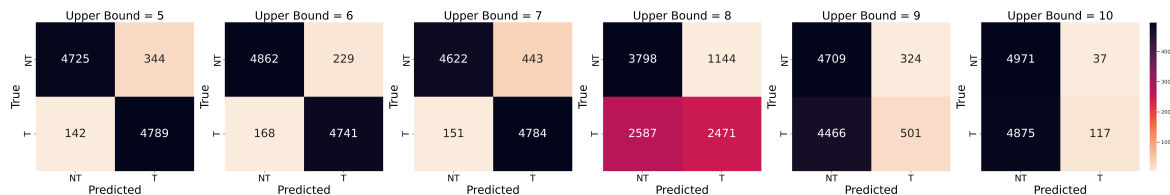


Figure 4. Confusion matrices for the neural network classifier on in-sample and out-of-sample data. In each case a balanced set of 10 000 random examples was tested.

Our approach has some limitations, however, which we enumerate here. Some of these limitations suggest directions for future research. A key drawback, common to most ML models, is that our classifier performs poorly on out-of-sample data. Recall from §3 that the dataset we generated bounded the entries of the matrices by seven. For weight matrices within this range the model is extremely accurate (95%), however this accuracy drops off rapidly for weight matrices that fall outside of this range: 62% for entries bounded by eight; 52% for entries bounded by nine; and 50% for entries bounded by ten. See Figure 4 for details. Note that the network quickly degenerates to always predicting non-terminal singularities.

Furthermore the training process seems to require more data than we would like, given how computationally expensive the training data is to generate. It is possible that a more sophisticated network architecture, that is better adapted to this specific problem, might require less data to train.

Mathematically, our work here was limited to toric varieties, and furthermore only to toric varieties of Picard rank two. Finding a meaningful vectorisation of an arbitrary algebraic variety looks like an impossible task. But if one is interested in the classification of algebraic varieties up to deformation, this might be less of a problem than it first appears. Any smooth Fano variety in low dimensions is, up to deformation, either a toric variety, a toric complete intersection, or a quiver flag zero locus [13, 30]; one might hope that this also covers a substantial fraction of the \mathbb{Q} -Fano landscape. Each of these classes of geometry is controlled by combinatorial structures, and it is possible to imagine a generalisation of our vectorisation by weight matrices to this broader context.

Generalising to \mathbb{Q} -factorial Fano toric varieties in higher Picard rank will require a more sophisticated approach to equivariant machine learning. In this paper, we could rely on the fact that there is a normal form (2.3) for rank-two weight matrices that gives an almost unique representative of each $SL_2(\mathbb{Z}) \times S_N$ -orbit of weight matrices. For higher Picard rank r we need to consider weight matrices up to the action of $G = SL_r(\mathbb{Z}) \times S_N$. Here no normal form is known, so to work G -equivariantly we will need to augment our dataset, to fill out the different G -orbits, or to use invariant functions of the weights as features. The latter option, geometrically speaking, is working directly with the quotient space.

The best possible path forward would be to train an explainable model that predicted terminality from the weight data. This would allow us to extract from the machine learning not only that the problem is tractable, but also a precise mathematical conjecture for the solution. At the moment, however, we are very far from this. The multilayer perceptron that we trained is a black-box model, and post-hoc explanatory methods such as SHAP analysis [40] yielded little insight: all features were used uniformly, as might be expected. We hope to return to this point elsewhere.

Data and code availability. The datasets underlying this work and the code used to generate them are available from Zenodo under a CC0 license [14]. Data generation and post-processing was carried out using the computational algebra system Magma V2.27-3 [7]. The machine learning model was built using PyTorch v1.13.1 [45] and scikit-learn v1.1.3 [46]. All code used and trained models are available from BitBucket under an MIT licence [16].

Acknowledgements. TC was partially supported by ERC Consolidator Grant 682603 and EPSRC Programme Grant EP/N03189X/1. AK was supported by EPSRC Fellowship EP/N022513/1. SV was supported by the Engineering and Physical Sciences Research Council [EP/S021590/1], the EPSRC Centre for Doctoral Training in Geometry and Number Theory (The London School of Geometry and

Number Theory), University College London. The authors would like to thank Hamid Abban, Alessio Corti, and Challenger Mishra for many useful conversations, and the anonymous referees for their insightful feedback and suggestions.

SUPPLEMENTARY MATERIAL A. MATHEMATICAL BACKGROUND

Toric varieties. The prototypical example of a toric Fano variety is two-dimensional projective space, \mathbb{P}^2 . As mentioned in §2, this is defined by taking the quotient of $\mathbb{C}^3 \setminus \{0\}$ by the following action of \mathbb{C}^\times :

$$\lambda \cdot (z_1, z_2, z_3) = (\lambda z_1, \lambda z_2, \lambda z_3)$$

The elements of \mathbb{P}^2 are equivalence classes that can be written as $[z_1 : z_2 : z_3]$ where at least one of the z_i is non-zero. The algebraic variety \mathbb{P}^2 is *smooth*, since we can cover it by three open subsets that are each isomorphic to the complex plane \mathbb{C}^2 . Namely,

$$U_1 = \{[z_1 : z_2 : z_3] \in \mathbb{P}^2 \mid z_1 \neq 0\}$$

$$U_2 = \{[z_1 : z_2 : z_3] \in \mathbb{P}^2 \mid z_2 \neq 0\}$$

$$U_3 = \{[z_1 : z_2 : z_3] \in \mathbb{P}^2 \mid z_3 \neq 0\}$$

To see that U_1 is isomorphic to \mathbb{C}^2 , we note that since $z_1 \neq 0$ it can be rescaled to one. Therefore, each point in U_1 can be identified with a (unique) point of the form $[1 : \bar{z}_2 : \bar{z}_3]$; this gives the isomorphism to \mathbb{C}^2 . Similar arguments show that U_2 and U_3 are each isomorphic to \mathbb{C}^2 .

More generally, $(N - 1)$ -dimensional projective space \mathbb{P}^{N-1} is smooth, since it can be covered by N open subsets each isomorphic to \mathbb{C}^{N-1} . By modifying the action of \mathbb{C}^\times on $\mathbb{C}^N \setminus \{0\}$ we can define more general examples of toric varieties, *weighted projective spaces*, which in general contain singular points.

For example, we can consider the action of \mathbb{C}^\times on $\mathbb{C}^3 \setminus \{0\}$ defined by

$$\lambda \cdot (z_1, z_2, z_3) = (\lambda z_1, \lambda z_2, \lambda^2 z_3)$$

which gives rise to the weighted projective space $\mathbb{P}(1, 1, 2)$. Here the entries of the vector $(1, 1, 2)$ are called the *weights* of the variety. In order to see that this variety is not smooth, we can consider the same open sets as above,

$$U_1 = \{[z_1 : z_2 : z_3] \in \mathbb{P}^2 \mid z_1 \neq 0\}$$

$$U_2 = \{[z_1 : z_2 : z_3] \in \mathbb{P}^2 \mid z_2 \neq 0\}$$

$$U_3 = \{[z_1 : z_2 : z_3] \in \mathbb{P}^2 \mid z_3 \neq 0\}$$

As before, U_1 and U_2 are each isomorphic to \mathbb{C}^2 . However, U_3 is not. In fact, since $z_3 \neq 0$ we can rescale the last entry to one, but the square in the definition of the action implies that there are two ways of doing so:

$$\pm z_3^{-1/2} \cdot (z_1, z_2, z_3) = (\pm z_3^{-1/2} z_1, \pm z_3^{-1/2} z_2, 1)$$

Therefore, $U_3 \cong \mathbb{C}^2/\mu_2$ where $\mu_2 = \{1, -1\}$ is the group of square roots of unity. Note that \mathbb{C}^2/μ_2 has a singular point at the origin, which corresponds to the singular point $[0 : 0 : 1]$ in U_3 . We say that $\mathbb{P}(1, 1, 2)$ has two smooth charts, U_1 and U_2 , and one singular chart U_3 .

This generalises to higher dimensions by considering \mathbb{C}^\times acting on $\mathbb{C}^N \setminus \{0\}$ by

$$\lambda \cdot (z_1, \dots, z_N) = (\lambda^{a_1} z_1, \dots, \lambda^{a_N} z_N)$$

for some choice of weights $(a_1, \dots, a_N) \in \mathbb{Z}_{>0}^N$. The algebraic variety $\mathbb{P}(a_1, a_2, \dots, a_N)$ is an $(N - 1)$ -dimensional \mathbb{Q} -factorial Fano toric variety of Picard rank one, called a *weighted projective space* [21, 29]. Setting the a_i equal to 1 recovers \mathbb{P}^{N-1} .

For any two weighted projective spaces $X = \mathbb{P}(a_1, \dots, a_N)$ and $Y = \mathbb{P}(b_1, \dots, b_M)$, we can consider their product $X \times Y$. This arises as a quotient of \mathbb{C}^{N+M} by an action of $\mathbb{C}^\times \times \mathbb{C}^\times$, where the first \mathbb{C}^\times acts on the first N co-ordinates of \mathbb{C}^{N+M} and the second \mathbb{C}^\times acts on the last M co-ordinates. The

two actions are specified by the weights of each weighted projective space. We can summarise this information in a *weight matrix*

$$\begin{bmatrix} a_1 & \cdots & a_N & 0 & \cdots & 0 \\ 0 & \cdots & 0 & b_1 & \cdots & b_M \end{bmatrix}$$

This type of construction can be generalised to any action of $\mathbb{C}^\times \times \mathbb{C}^\times$ on \mathbb{C}^N given defined as

$$(\lambda, \mu) \cdot (z_1, \dots, z_N) = (\lambda^{a_1} \mu^{b_1} z_1, \dots, \lambda^{a_N} \mu^{b_N} z_N)$$

and which can be encoded in a weight matrix of the form

$$\begin{bmatrix} a_1 & \cdots & a_N \\ b_1 & \cdots & b_N \end{bmatrix}$$

Note that in the case of projective spaces and weighted projective spaces we were considering $\mathbb{C}^N \setminus \{\mathbf{0}\}$, excluding the origin because it lies in the closure of every orbit. When generalising this concept, we need to exclude more points than just the origin for the quotient to be reasonable; explicitly we consider $\mathbb{C}^N \setminus S$, where $S = S_+ \cup S_-$ for linear subspaces

$$S_+ = \{(z_1, \dots, z_N) \mid z_i = 0 \text{ if } b_i/a_i > b/a\}$$

$$S_- = \{(z_1, \dots, z_N) \mid z_i = 0 \text{ if } b_i/a_i < b/a\}$$

and $a = \sum_{i=1}^N a_i, b = \sum_{i=1}^N b_i$: see [8]. The resulting quotient $X = (\mathbb{C}^N \setminus S)/(\mathbb{C}^\times)^2$ is an $(N-2)$ -dimensional toric variety. If the linear subspaces S_+ and S_- each have dimension at least two then X has Picard rank two.

From weight matrices to fans. In §2, a toric variety X was determined by a matrix

$$W = \begin{bmatrix} a_1 & \cdots & a_N \\ b_1 & \cdots & b_N \end{bmatrix} \tag{A.1}$$

that, as recalled above, records the weights of an action of $(\mathbb{C}^\times)^2$ on \mathbb{C}^N . We will now explain how to recover the fan $\Sigma(X)$ for the toric variety from this data [17, 24].

Consider the right kernel of the matrix W , regarded as a \mathbb{Z} -linear map. The kernel is a free submodule of \mathbb{Z}^N , of rank $N - 2$, and choosing a basis for this submodule defines an $N \times (N - 2)$ matrix M such that $WM = 0$. The rows of M define distinct primitive vectors e_1, \dots, e_N in \mathbb{Z}^{N-2} such that

$$a_1 e_1 + \cdots + a_N e_N = 0$$

$$b_1 e_1 + \cdots + b_N e_N = 0$$

By construction, the vectors e_1, \dots, e_N span the kernel of W over \mathbb{Z} .

In general the construction of a toric variety (or equivalently a fan) from a weight matrix depends also on the choice of a *stability condition*, which is an element ω of the column space of W . In our case, however, because X is Fano there is a canonical choice for ω given by (a, b) , the sum of the columns of W . Let us denote the i th column of W by D_i . We set

$$\mathcal{A}_\omega = \{I \subset \{1, 2, \dots, N\} \mid \omega \in \angle_I\}$$

where

$$\angle_I = \left\{ \sum_{i \in I} \lambda_i D_i \mid \lambda_i \in \mathbb{R}_{>0} \right\}$$

The fan $\Sigma(X)$ is the collection of cones in \mathbb{R}^{N-2} given by

$$\{\sigma_I \mid \bar{I} \in \mathcal{A}_\omega\} \quad \text{where } \sigma_I = \text{cone}\{e_i \mid i \in I\}$$

Here \bar{I} is the complement of I in $\{1, 2, \dots, N\}$.

Recall our assumptions on the weight matrix W :

- (0) The columns of W span a strictly convex cone in \mathbb{R}^2 .
- (1) None of the columns are the zero vector.
- (2) The sum of the columns is not a multiple of any of them.

(3) The subspaces S_+ and S_- , defined in (2.2), are both of dimension at least two.

(We number from zero here to match the numbering of the conditions in §3.) Conditions (0) and (1) together guarantee that the fan $\Sigma(X)$ is complete; that is, its support covers \mathbb{R}^{N-2} . The toric variety X is therefore compact. Condition (2) ensures that each top-dimensional cone in the fan has $N - 2$ rays; that is, the fan is simplicial. This implies that the toric variety X is \mathbb{Q} -factorial. Condition (3) ensures that each of the vectors e_1, \dots, e_N generates a one-dimensional cone $\mathbb{R}_{\geq 0}e_i$ in the fan $\Sigma(X)$. Together with \mathbb{Q} -factoriality, this implies that the Picard rank of X is two.

Checking terminality. Each top-dimensional cone σ in $\Sigma(X)$ is generated over $\mathbb{R}_{\geq 0}$ by $N - 2$ of the vectors e_1, \dots, e_N . These generators are contained in a unique $(N - 3)$ -dimensional hyperplane H . The cone σ corresponds to a terminal singularity in X if and only if the only lattice points in σ that lie on or below H are the generators of σ and the origin [48]. X has terminal singularities if and only if each top-dimensional cone of $\Sigma(X)$ corresponds to a terminal singularity. This justifies the assertion, given in §2, that X has terminal singularities if and only if the convex polytope $P = \text{conv}\{e_1, \dots, e_N\}$ is mostly empty.

A subtlety with quotient gradings. In §1, in the paragraph ‘Why dimension eight?’, we noted that the analogue of our dataset in dimension three contains 34 examples. There are 35 \mathbb{Q} -Fano toric varieties of Picard rank two in dimension three [31], but precisely one of these has a quotient grading and so does not fit into the framework we consider here. The exception is $X = \mathbb{P}^1 \times \mathbb{P}^2 / \mu_3$, where μ_3 acts via $(u, v; x, y, z) \mapsto (u, \varepsilon v; x, \varepsilon y, \varepsilon^2 z)$ and ε is a primitive cube root of unity. The quotient grading arises here because the primitive generators for rays of the fan $\Sigma(X)$ fail to span the ambient lattice over \mathbb{Z} . If we instead regard the primitive generators as living inside the sublattice that they generate, then we recover one of the other 34 terminal examples: $\mathbb{P}^1 \times \mathbb{P}^2$. The analogue of this phenomenon happens in higher dimensions too, and so we ignore quotient gradings in our methodology.

Significance of \mathbb{Q} -Fano varieties. As mentioned in §1, \mathbb{Q} -Fano varieties are ‘atomic pieces’ from which more complicated algebraic varieties are made, and so one can think of the classification of \mathbb{Q} -Fano varieties as building a Periodic Table for geometry. Understanding this classification is a fundamental problem in algebraic geometry, and is the motivation behind a huge amount of research; see e.g. [9, 11, 33, 35] and the references therein.

\mathbb{Q} -Fano varieties also play an important role elsewhere in mathematics, for example in the study of K-stability and the existence of Kähler–Einstein metrics [5]. In theoretical physics, \mathbb{Q} -Fano varieties provide, through their ‘anticanonical sections’, the main construction of the Calabi-Yau manifolds which give geometric models of spacetime [10, 25, 47] in Type II string theory.

Moreover, terminal singularities – the focus of this paper – are the singularities that appear in the Minimal Model Program [33], and they also occur across mathematics. For example, in F-theory, terminal singularities reflect the presence of localized matter states from wrapped M2-branes which are not charged under any massless gauge potential [3]. Moreover, in the toric context, having only terminal singularities means that the corresponding polytope contains no lattice points other than the origin and the vertices. These are referred to in the combinatorics literature as one-point lattice polytopes, and are important in optimisation problems.

SUPPLEMENTARY MATERIAL B. FURTHER DATA ANALYSIS

The neural network classifier described in §4 is remarkably accurate at determining whether a \mathbb{Q} -factorial Fano toric variety of Picard rank two and dimension eight is terminal or not. Confusion matrices for the classifier are presented in Figure 5. Because of this high accuracy, we were able to use this classifier to generate a dataset of 100M probably- \mathbb{Q} -Fano toric varieties of Picard rank two and dimension eight; see §6. Creating this first glimpse of the \mathbb{Q} -Fano landscape would have been impractical using conventional methods. Based on the timing data outlined in §C below, we estimate that generating this dataset using conventional methods would have taken 160 days on our HPC cluster, equivalent to 600 CPU *years*. In contrast, by using the neural network classifier and batch processing we were able to generate this dataset in under 120 CPU *hours*.

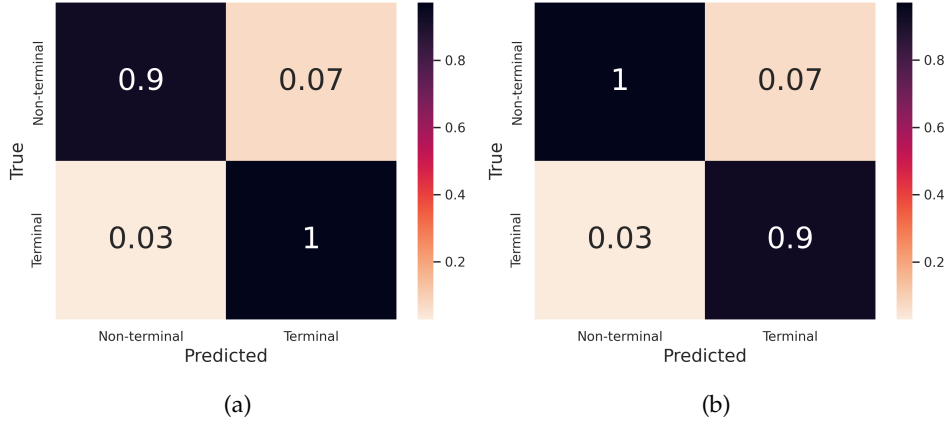


Figure 5. Confusion matrices for the classifier trained on 5M samples: (a) is normalised with respect to the true axis; (b) is normalised with respect to the predicted axis.

One striking feature of the landscape of 100M probably- \mathbb{Q} -Fano toric varieties, plotted in Figure 3, is the stratification by Fano index. Recall that the Fano index of X is equal to the greatest common divisor of a and b , where (a, b) is the sum of the columns of the matrix (A.1). For our dataset, the entries in the matrix (A.1) are bounded between zero and seven, and hence the range of possible Fano indices that can appear in the dataset is bounded. Figure 3 appears to show overlapping clusters of cases, with the Fano index increasing as we move from the bottom of the plot (Fano index one) to the top.

Products of weighted projective space. To better understand this clustering by Fano index, we consider the simplest \mathbb{Q} -factorial Fano toric varieties of Picard rank two: products of weighted projective spaces. Recall from §A that a product of weighted projective spaces $X = \mathbb{P}(a_1, \dots, a_N)$ and $Y = \mathbb{P}(b_1, \dots, b_M)$ is specified by a weight matrix

$$\begin{bmatrix} a_1 & \cdots & a_N & 0 & \cdots & 0 \\ 0 & \cdots & 0 & b_1 & \cdots & b_M \end{bmatrix}$$

This matrix determines a \mathbb{Q} -factorial Fano toric variety of Picard rank two and dimension $N + M - 2$, denoted $X \times Y$. The singular points of $X \times Y$ are determined by the singular points of X and Y . In particular, $X \times Y$ is terminal if and only if both X and Y are terminal.

In general a weighted projective space $X = \mathbb{P}(a_1, a_2, \dots, a_N)$ may have singular points; these are determined by the weights (a_1, a_2, \dots, a_N) . Proposition 2.3 of [32] characterises when the singular points of X are terminal. Namely, X is terminal if and only if

$$\sum_{i=1}^N \{ka_i/a\} \in \{2, \dots, N-2\}$$

for each $k \in \{2, \dots, a-2\}$. Here $a = a_1 + a_2 + \dots + a_N$, and $\{x\}$ denotes the fractional part $x - [x]$ of a rational number x . This is the Picard rank one analogue to Proposition 3.

We can enumerate all terminal weighted projective spaces in dimensions one to seven, with weights $1 \leq a_i \leq 7$, using the characterisation of terminal weighted projective space described above. The number in each dimension is given in Table 2. By taking products, we obtain 8792 distinct \mathbb{Q} -Fano toric varieties of Picard rank two in dimension eight; these examples are plotted in Figure 6. This supports our observation that the \mathbb{Q} -Fano varieties fall into large overlapping clusters that are determined by the Fano index. Note that the products of weighted projective space appear to fall within the upper region of each cluster.

Smooth Fano toric varieties. Projective space \mathbb{P}^{N-1} is smooth, and so products of projective spaces are also smooth. More generally, the smooth Fano toric varieties up to dimension eight have been classified [44]. There are 62 smooth Fano toric varieties in dimension eight and of Picard rank two, all

d	1	2	3	4	5	6	7
#	1	1	7	80	356	972	2088

Table 2. The number of terminal weighted projective spaces in dimension d , $1 \leq d \leq 7$, with weights a_i bounded by seven.

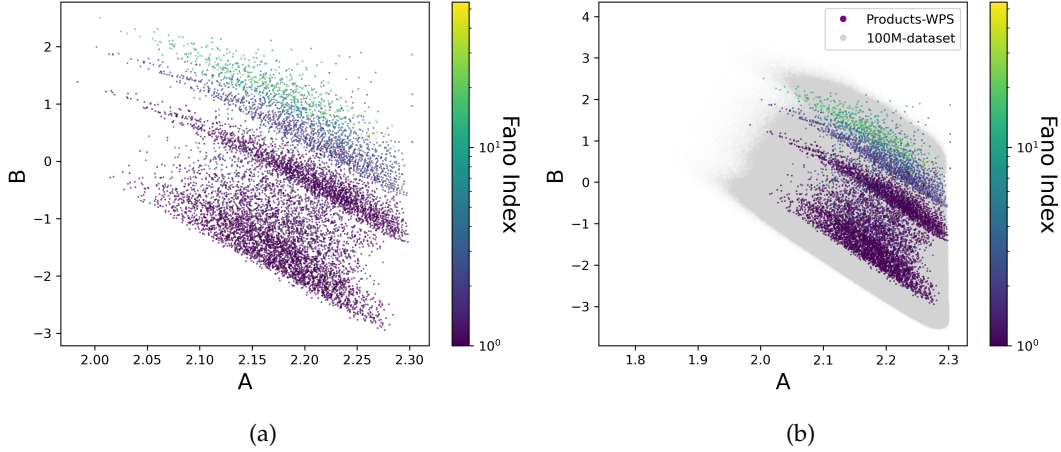


Figure 6. \mathbb{Q} -Fano products of weighted projective space in dimension eight, with weights bounded by seven. (a) Projection to \mathbb{R}^2 using the growth coefficients from (6.2). (b) The same as (a), but plotted on top of the dataset of 100M probably- \mathbb{Q} -Fano toric varieties, plotted in grey.

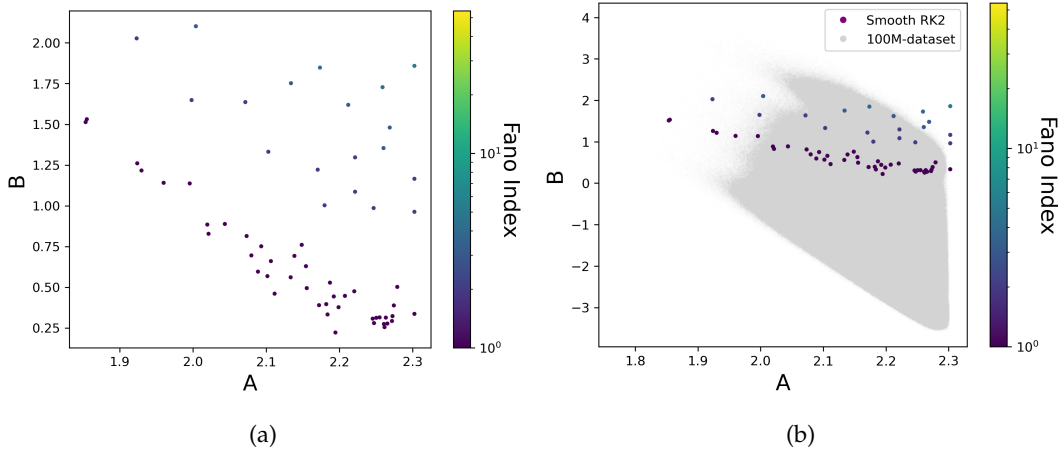


Figure 7. The smooth Fano toric varieties in dimension eight and of Picard rank two. (a) Projection to \mathbb{R}^2 using the growth coefficients from (6.2). (b) The same as (a), but plotted on top of the dataset of 100M probably- \mathbb{Q} -Fano toric varieties, plotted in grey.

of which have weights bounded by seven when expressed in standard form (2.3). These are plotted in Figure 7, and appear to fall in the upper extreme region within each cluster.

A cluster of high-Fano index examples. Figure 3 appears to show a cluster of high-Fano-index cases (at the top of the plot) standing apart from the remainder of the data. We now give an explanation for this high-Fano-index cluster. Figure 8 shows the frequency distribution of Fano indices in the dataset. The uptick in frequencies in the histogram in Figure 8 can be explained as follows. Consider how many ways we can write N as a sum of ten numbers between zero and seven (inclusive, and with possible repeats). This resembles a normal distribution with $N = 35$ the most frequent case. This higher probability is due to our sampling constraints on the entries of the weight matrix: amongst

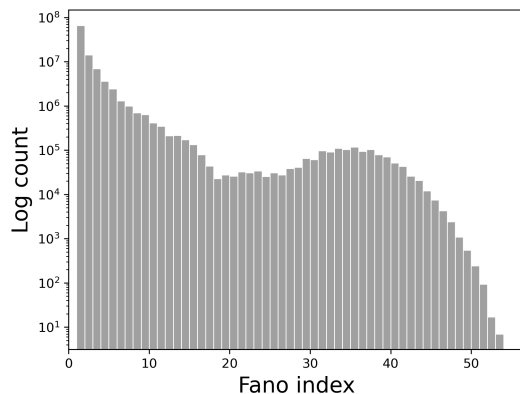


Figure 8. Distribution of the Fano index $\gcd\{a, b\}$ in the dataset of 100M probably- \mathbb{Q} -Fano toric varieties (note that the vertical axis scale is logged).

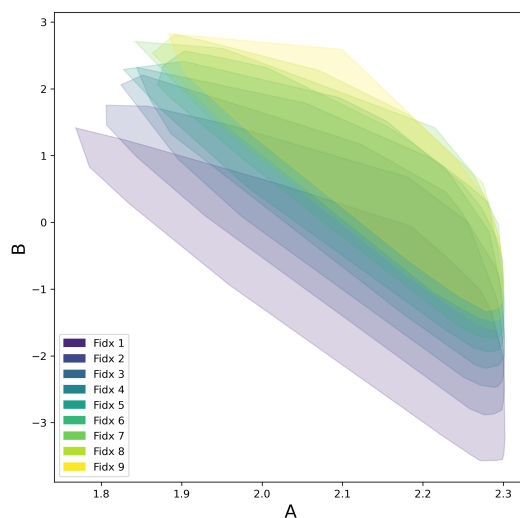


Figure 9. Convex hulls obtained from the point clouds for probably- \mathbb{Q} -Fano toric varieties with Fano indices between one and nine, obtained by projecting to \mathbb{R}^2 using the growth coefficients from (6.2).

those matrices that have $a = b$ we have the highest probability of selecting one that has $a = b = 35$. Therefore, we see a misleading accumulation around those Fano indices.

In Figure 9 we restrict the dataset to low Fano indices. For each Fano index in the range one through to nine, we plot the convex hull of the resulting point cloud. The overlap between these clusters is clear.

SUPPLEMENTARY MATERIAL C. COMPUTATIONAL RESOURCES

In this section we describe the computational resources required by different steps of our analysis. We will refer to a *desktop PC* and an *HPC cluster*. The desktop PC has an Intel Xeon 5222 quad-core processor, 64GB RAM, and an NVIDIA RTX A2000 12 GB GPU; note however that all CPU jobs on the desktop PC ran single-core. The HPC cluster has Intel Xeon E5-2650 processors with a total of 1400 cores.

Data generation. The datasets `bound_7_terminal` and `bound_7_non_terminal` were generated using scripts for the computational algebra system Magma [7], running on the HPC cluster in parallel over 1400 cores for eight days, with 2GB of memory per core. Deduplication of the dataset was performed on the desktop PC and took approximately eight hours.

Hyperparameter tuning. This was carried out on the desktop PC, using the GPU. Each experiment ran on average for two minutes, for a total run time of 200 minutes for 100 experiments.

Model training. This was carried out using the desktop PC, using the GPU. Training on 5M balanced samples for 150 epochs took four hours.

Model evaluation. The model evaluation was carried out using the desktop PC, using the GPU. Evaluation took approximately ten minutes.

Further data generation. The dataset `terminal_dim8_probable` was generated by running Python scripts on the HPC cluster in parallel over 120 cores for one hour, with 16GB of memory per core. Deduplication of the dataset was performed on the desktop PC and took approximately one hour.

SUPPLEMENTARY MATERIAL D. TRAINING FOR WEIGHTS WITH A LARGER BOUND

In §7 we highlighted that the trained neural network does not perform well out of sample. Therefore, it is natural to ask whether the neural network is approximating an actual general mathematical statement, or if its performance is the result of some ‘finite size effect’ due to the choice of a particular weight bound (in our case seven). Our intuition here is as follows. Given that the testing and training data are free of noise (they are created through exact mathematical calculation) and the neural network classifier is so accurate, we believe that the classifier is indeed approximating a precise, general mathematical statement. However, the poor out-of-sample performance makes it unclear *what kind of mathematical statement* the network is picking up. The statement could be about weight matrices with entries of arbitrary size, or could be about weight matrices with small entries (mathematically, this would be a statement about Fano varieties with terminal singularities of bounded index). In the first case the out-of-sample performance drop-off would happen because the network is approximating the true statement in a way that does not generalise to higher weight bounds; this is a common phenomenon when developing and using neural network models. In the second case the out-of-sample performance drop-off would happen because of the underlying mathematical statement that the classifier approximates.

To probe this further, we repeated the same experiments as in the main text on a dataset of weight matrices with weights bounded by a larger constant, ten. We generated a new dataset of size 20 million, balanced between terminal and non-terminal examples, where the entries of each weight matrix are bounded by ten. The data generation steps were the same as described in §3, except that the terminality check was now carried out using the new algorithm discussed in §5 (and proved correct in §E). We remark that the increased speed of the new algorithm allowed us to generate double the amount of data of the original dataset.

We used a fully-connected feed-forward neural network with the same architecture as the original neural network from the paper. This architecture is recalled in Table 3. Again, the network was trained on the features given by flattening the weight matrices, which were standardised by translating the mean to zero and rescaling the variance to one. It was trained using binary cross-entropy as loss function, stochastic mini-batch gradient descent optimiser and using early-stopping, for a maximum of 150 epochs and with learning rate reduction on plateaux.

Training on 5M samples (using 80% for training and 10% for validation) and testing on the remaining data (15M samples) produced an accuracy of 90% – see Figure 10(b) for the loss learning curve. This performance is worse than that achieved for the same training sample size for weight bound seven, potentially indicating that the condition approximated by the network is harder to capture. Training on a larger sample of size 10M (again using 80% for training and 10% for validation) and testing on the remaining data (10M samples) improves the accuracy to 94% – see Figure 10(c) for the loss learning curve. The training and validation accuracies for intermediate training sizes are shown in Figure 10(a).

We were able to recover a high accuracy for this new dataset. However, this was only possible by using a larger training sample size, which hints at the increased difficulty of the task. Moreover, Figure 10(a) suggests that increasing the size of the training set further is unlikely to improve the accuracy. Being able to train a high-accuracy neural network for a larger weights bound supports the hypothesis that the neural network is approximating a general mathematical statement but in a way

Hyperparameter	Value	Hyperparameter	Value
Layers	(512, 768, 512)	Momentum	0.99
Batch size	128	LeakyRelu slope	0.01
Initial learning rate	0.01		

Table 3. Final network architecture and configuration.

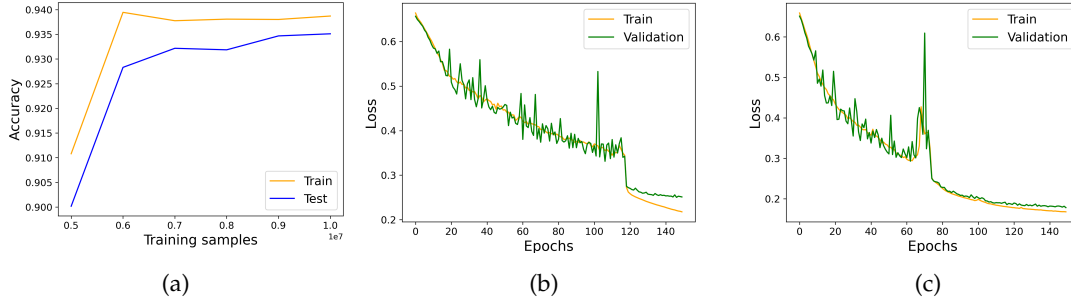


Figure 10. (a) Accuracy for different train-test splits; (b) epochs against loss for the network trained on 5M samples; (c) epochs against loss for the network trained on 10M samples.

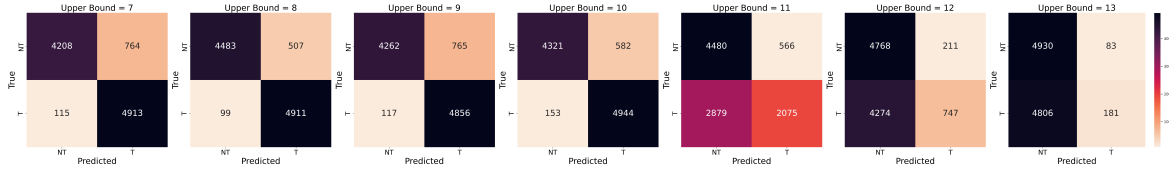


Figure 11. Confusion matrices for the neural network classifier on in-sample and out-of-sample data. In each case a balanced set of 10 000 random examples was tested.

that does not generalise well to higher bounds. However, it is too early to exclude the hypothesis that the network might be capturing a mathematical statement that needs weight matrices with small entries. Similar studies with even higher bounds would add confidence here and, if the network is indeed approximating a statement about weight matrices with small weights, experiments of this type should also be able to deduce what the cut-off bound for the weights is.

SUPPLEMENTARY MATERIAL E. PROOF OF PROPOSITION 3

In this section we prove Proposition 3. This is the main ingredient in the new algorithm to check terminality. Recall from the discussion above that X determines a convex polytope P with vertices $e_1, \dots, e_N \in \mathbb{Z}^{N-2}$, and that

$$\begin{aligned} a_1 e_1 + \dots + a_N e_N &= 0 \\ b_1 e_1 + \dots + b_N e_N &= 0 \end{aligned}$$

where the a_i and b_j are entries in the weight matrix (A.1). The same argument applied to the equivalent weight matrix

$$\begin{bmatrix} b_i/g_i & -a_i/g_i \\ A_i & B_i \end{bmatrix} \begin{bmatrix} a_1 & \dots & a_N \\ b_1 & \dots & b_N \end{bmatrix}$$

gives barycentric co-ordinates for the origin and e_i in terms of the remaining vertices of Δ_i :

$$\begin{aligned} \alpha_i^1 e_1 + \dots + \alpha_i^{i-1} e_{i-1} + \alpha_i^{i+1} e_{i+1} + \dots + \alpha_i^N e_N &= 0 \\ \beta_i^1 e_1 + \dots + \beta_i^{i-1} e_{i-1} + \beta_i^{i+1} e_{i+1} + \dots + \beta_i^N e_N &= g_i e_i \end{aligned}$$

Fix $i \in \{1, 2, \dots, N\}$. Define $u: \mathbb{Q}^{N-1} \rightarrow \mathbb{Q}$ by $u(x_1, \dots, x_{N-1}) = x_1 + \dots + x_{N-1}$, and let Ψ denote the lattice

$$\{v \in \mathcal{Z} \mid u(v) = 1\}$$

where \mathcal{Z} is the span over \mathbb{Z} of the standard basis E_1, \dots, E_{N-1} for \mathbb{Q}^{N-1} together with

$$\frac{1}{f_i}(\alpha_1^2, \dots, \hat{\alpha}_i^i, \dots, \alpha_1^N) \quad \text{and} \quad \frac{1}{g_i}(\beta_1^2, \dots, \hat{\beta}_i^i, \dots, \beta_1^N)$$

Here the $\hat{}$ indicates that the i th entry in each vector is omitted. We define $\phi: \Psi \rightarrow \mathbb{Z}^{N-2}$ to be the \mathbb{Z} -linear map that sends E_1, \dots, E_{N-1} to $e_1, \dots, \hat{e}_i, \dots, e_N$ and

$$\phi\left(\frac{1}{f_i}(\alpha_1^2, \dots, \hat{\alpha}_i^i, \dots, \alpha_1^N)\right) = 0 \quad \phi\left(\frac{1}{g_i}(\beta_1^2, \dots, \hat{\beta}_i^i, \dots, \beta_1^N)\right) = e_i$$

It is easy to see that ϕ is well-defined and bijective.

Consider the higher-dimensional parallelepiped Γ in \mathcal{Z} generated by the standard basis of \mathbb{Z}^{N-1} . We note that each lattice point of \mathcal{Z} in Γ can be represented as a linear combination

$$\frac{k}{f_i}(\alpha_1^2, \dots, \hat{\alpha}_i^i, \dots, \alpha_1^N) + \frac{l}{g_i}(\beta_1^2, \dots, \hat{\beta}_i^i, \dots, \beta_1^N) \quad (\text{E.1})$$

for some $k \in \{0, 1, \dots, f_i - 1\}$ and $l \in \{0, 1, \dots, g_i - 1\}$; this representation is unique if and only if the vertices of Δ_i span \mathbb{Z}^{N-2} . Hence, Δ_i is almost empty if and only if whenever

$$\sum_{j \neq i} \left\{ k \frac{\alpha_i^j}{f_i} + l \frac{\beta_i^j}{g_i} \right\} = 1 \quad (\text{E.2})$$

we have that the linear combination in (E.1) represents the origin. But this is the case if and only if

$$\left\{ k \frac{\alpha_i^j}{f_i} + l \frac{\beta_i^j}{g_i} \right\} = \left\{ \frac{\alpha_i^j}{\alpha_i} \right\}$$

for all j , since $(k, l) = (\frac{f_i}{\alpha_i}, 0)$ represents the origin by construction. Note that the sum (E.2) could include $j = i$, since that term is an integer and its fractional part will not contribute to the sum. \square

REFERENCES

- [1] Jeffrey Adams, Annegret Paul, Ran Cui, Susana Salamanca-Riba, Peter Trapa, Marc van Leeuwen, and David Vogan. Atlas of Lie groups and representations. Online, 2016. <http://www.liegroups.org>.
- [2] Hamid Ahmadinezhad. On pliability of del Pezzo fibrations and Cox rings. *J. Reine Angew. Math.*, 723:101–125, 2017. [doi:10.1515/crelle-2014-0095](https://doi.org/10.1515/crelle-2014-0095).
- [3] Philipp Arras, Antonella Grassi, and Timo Weigand. Terminal singularities, Milnor numbers, and matter in F-theory. *J. Geom. Phys.*, 123:71–97, 2018. [doi:10.1016/j.geomphys.2017.09.001](https://doi.org/10.1016/j.geomphys.2017.09.001).
- [4] Michael F. Atiyah, Nigel J. Hitchin, Vladimir G. Drinfeld, and Yuri I. Manin. Construction of instantons. *Phys. Lett. A*, 65(3):185–187, 1978. [doi:10.1016/0375-9601\(78\)90141-X](https://doi.org/10.1016/0375-9601(78)90141-X).
- [5] Robert J. Berman. K-polystability of \mathbb{Q} -Fano varieties admitting Kähler-Einstein metrics. *Invent. Math.*, 203(3):973–1025, 2016. [doi:10.1007/s00222-015-0607-7](https://doi.org/10.1007/s00222-015-0607-7).
- [6] Caucher Birkar. Singularities of linear systems and boundedness of Fano varieties. *Ann. of Math. (2)*, 193(2):347–405, 2021. [doi:10.4007/annals.2021.193.2.1](https://doi.org/10.4007/annals.2021.193.2.1).
- [7] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. [doi:10.1006/jsco.1996.0125](https://doi.org/10.1006/jsco.1996.0125).
- [8] Gavin Brown, Alessio Corti, and Francesco Zucconi. Birational geometry of 3-fold Mori fibre spaces. In *The Fano Conference*, pages 235–275. Univ. Torino, Turin, 2004.
- [9] Gavin Brown and Alexander M. Kasprzyk. The graded ring database. Online, 2007–present. <http://www.grdb.co.uk>.
- [10] Philip Candelas, Gary T. Horowitz, Andrew Strominger, and Edward Witten. Vacuum configurations for superstrings. *Nuclear Phys. B*, 258(1):46–74, 1985. [doi:10.1016/0550-3213\(85\)90602-9](https://doi.org/10.1016/0550-3213(85)90602-9).
- [11] Paolo Cascini. New directions in the minimal model program. *Boll. Unione Mat. Ital.*, 14(1):179–190, 2021. [doi:10.1007/s40574-020-00250-9](https://doi.org/10.1007/s40574-020-00250-9).
- [12] Tom Coates, Alessio Corti, Sergey Galkin, Vasily Golyshev, and Alexander M. Kasprzyk. Mirror symmetry and Fano manifolds. In *European Congress of Mathematics*, pages 285–300. Eur. Math. Soc., Zürich, 2013. [doi:10.4171/120](https://doi.org/10.4171/120).
- [13] Tom Coates, Alessio Corti, Sergey Galkin, and Alexander M. Kasprzyk. Quantum periods for 3-dimensional Fano manifolds. *Geom. Topol.*, 20(1):103–256, 2016. [doi:10.2140/gt.2016.20.103](https://doi.org/10.2140/gt.2016.20.103).

- [14] Tom Coates, Alexander M. Kasprzyk, and Sara Veneziale. A dataset of 8-dimensional \mathbb{Q} -factorial Fano toric varieties of Picard rank 2. *Zenodo*, 2023. doi:10.5281/zenodo.10046893.
- [15] Tom Coates, Alexander M. Kasprzyk, and Sara Veneziale. Machine learning the dimension of a Fano variety. *Nat. Commun.*, 14:5526, 2023. doi:10.1038/s41467-023-41157-1.
- [16] Tom Coates, Alexander M. Kasprzyk, and Sara Veneziale. Supporting code. https://bitbucket.org/fanosearch/ml_terminality, 2023.
- [17] David A. Cox, John B. Little, and Henry K. Schenck. *Toric varieties*, volume 124 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2011. doi:10.1090/gsm/124.
- [18] John Cremona. The L-functions and modular forms database project. *Found. Comput. Math.*, 16(6):1541–1553, 2016. doi:10.1007/s10208-016-9306-z.
- [19] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600:70–74, 2021. doi:10.1038/s41586-021-04086-x.
- [20] Pasquale Del Pezzo. Sulle superficie dell' n^{mo} ordine immerse nello spazio ad n dimensioni. *Rend. del Circolo Mat. di Palermo*, 1:241–255, 1887.
- [21] Igor Dolgachev. Weighted projective varieties. In *Group actions and vector fields (Vancouver, B.C., 1981)*, volume 956 of *Lecture Notes in Math.*, pages 34–71. Springer, Berlin, 1982. doi:10.1007/BFb0101508.
- [22] Harold Erbin and Riccardo Finotello. Machine learning for complete intersection Calabi–Yau manifolds: a methodological study. *Phys. Rev. D*, 103(12):Paper No. 126014, 40, 2021. doi:10.1103/physrevd.103.126014.
- [23] Nicholas Eriksson, Kristian Ranestad, Bernd Sturmfels, and Seth Sullivant. Phylogenetic algebraic geometry. In *Projective varieties with unexpected properties*, pages 237–255. Walter de Gruyter, Berlin, 2005.
- [24] William Fulton. *Introduction to toric varieties*, volume 131 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 1993. doi:10.1515/9781400882526.
- [25] Brian R. Greene. String theory on Calabi–Yau manifolds. In *Fields, strings and duality (Boulder, CO, 1996)*, pages 543–726. World Sci. Publ., River Edge, NJ, 1997.
- [26] Roland Grinis and Alexander M. Kasprzyk. Normal forms of convex lattice polytopes. [arXiv:1301.6641 \[math.CO\]](https://arxiv.org/abs/1301.6641), 2013.
- [27] Yang-Hui He. Machine-learning mathematical structures. *International Journal of Data Science in the Mathematical Sciences*, 1:23–47, 2023.
- [28] Yang-Hui He, Kyu-Hwan Lee, Thomas Oliver, and Alexey Pozdnyakov. Murmurations of elliptic curves. [arXiv:2204.10140 \[math.NT\]](https://arxiv.org/abs/2204.10140), 2022.
- [29] A. R. Iano-Fletcher. Working with weighted complete intersections. In *Explicit birational geometry of 3-folds*, volume 281 of *London Math. Soc. Lecture Note Ser.*, pages 101–173. Cambridge Univ. Press, Cambridge, 2000.
- [30] Elana Kalashnikov. Four-dimensional Fano quiver flag zero loci. *Proc. Royal Society A.*, 475(2225):20180791, 23, 2019. doi:10.1098/rspa.2018.0791.
- [31] Alexander M. Kasprzyk. Toric Fano three-folds with terminal singularities. *Tohoku Math. J. (2)*, 58(1):101–121, 2006. doi:10.2748/tmj/1145390208.
- [32] Alexander M. Kasprzyk. Classifying terminal weighted projective space. [arXiv:1304.3029 \[math.AG\]](https://arxiv.org/abs/1304.3029), 2013.
- [33] János Kollár. The structure of algebraic threefolds: an introduction to Mori’s program. *Bull. Amer. Math. Soc. (N.S.)*, 17(2):211–273, 1987. doi:10.1090/S0273-0979-1987-15548-0.
- [34] János Kollár, Yoichi Miyaoka, Shigefumi Mori, and Hiromichi Takagi. Boundedness of canonical \mathbb{Q} -Fano 3-folds. *Proc. Japan Acad. Ser. A Math. Sci.*, 76(5):73–77, 2000. doi:10.3792/pjaa.76.73.
- [35] János Kollár and Shigefumi Mori. *Birational geometry of algebraic varieties*, volume 134 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 1998. doi:10.1017/CBO9780511662560.
- [36] Maxim Kontsevich and Yuri Manin. Gromov–Witten classes, quantum cohomology, and enumerative geometry. In *Mirror symmetry, II*, volume 1 of *AMS/IP Stud. Adv. Math.*, pages 607–653. Amer. Math. Soc., Providence, RI, 1997. doi:10.1090/amsip/001/23.
- [37] Maximilian Kreuzer and Harald Skarke. PALP: a package for analysing lattice polytopes with applications to toric geometry. *Comput. Phys. Comm.*, 157(1):87–106, 2004. doi:10.1016/S0010-4655(03)00491-0.
- [38] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-Tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246, 2020.
- [39] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. [arXiv:1807.05118 \[cs.LG\]](https://arxiv.org/abs/1807.05118), 2018.
- [40] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.
- [41] Shigefumi Mori and Shigeru Mukai. Classification of Fano 3-folds with $B_2 \geq 2$. *Manuscripta Math.*, 36(2):147–162, 1981/82. doi:10.1007/BF01170131.
- [42] Shigefumi Mori and Shigeru Mukai. Erratum: “Classification of Fano 3-folds with $B_2 \geq 2$ ”. *Manuscripta Math.*, 110(3):407, 2003. doi:10.1007/s00229-002-0336-2.
- [43] Harald Niederreiter and Chaoping Xing. *Algebraic geometry in coding theory and cryptography*. Princeton University Press, Princeton, NJ, 2009.
- [44] Mikkel Øbro. An algorithm for the classification of smooth Fano polytopes. [arXiv:0704.0049 \[math.CO\]](https://arxiv.org/abs/0704.0049), 2007.

- [45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library, 2019.
- [46] Fabina Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. doi:10.5555/1953048.2078195.
- [47] Joseph Polchinski. *String theory. Vol. II*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, Cambridge, 2005. Superstring theory and beyond, Reprint of 2003 edition.
- [48] Miles Reid. Young person’s guide to canonical singularities. In *Algebraic geometry, Bowdoin, 1985 (Brunswick, Maine, 1985)*, volume 46 of *Proc. Sympos. Pure Math.*, pages 345–414. Amer. Math. Soc., Providence, RI, 1987.
- [49] Adam Zsolt Wagner. Constructions in combinatorics via neural networks. [arXiv:2104.14516 \[math.CO\]](https://arxiv.org/abs/2104.14516), 2021.
- [50] Geordie Williamson. Is deep learning a useful tool for the pure mathematician? [arXiv:2304.12602 \[math.RT\]](https://arxiv.org/abs/2304.12602), 2023.
- [51] Yue Wu and Jesús A De Loera. Turning mathematics problems into games: Reinforcement learning and Gröbner bases together solve integer feasibility problems. [arXiv:2208.12191 \[cs.LG\]](https://arxiv.org/abs/2208.12191), 2022.

DEPARTMENT OF MATHEMATICS, IMPERIAL COLLEGE LONDON, 180 QUEEN’S GATE, LONDON, SW7 2AZ, UK
Email address: t.coates@imperial.ac.uk

SCHOOL OF MATHEMATICAL SCIENCES, UNIVERSITY OF NOTTINGHAM, NOTTINGHAM, NG7 2RD, UK
Email address: a.m.kasprzyk@nottingham.ac.uk

DEPARTMENT OF MATHEMATICS, IMPERIAL COLLEGE LONDON, 180 QUEEN’S GATE, LONDON, SW7 2AZ, UK
Email address: s.veneziale21@imperial.ac.uk