Machine learning of Calabi-Yau volumes

Daniel Krefl[1] and Rak-Kyeong Seong[2]

[1]*Theoretical Physics Department, CERN, Geneva 23, CH-1211 Switzerland*
[2]*Department of Physics and Astronomy, Uppsala University, SE-751 08 Uppsala, Sweden*

We employ machine learning techniques to investigate the volume minimum of Sasaki-Einstein base manifolds of noncompact toric Calabi-Yau three-folds. We find that the minimum volume can be approximated via a second-order multiple linear regression on standard topological quantities obtained from the corresponding toric diagram. The approximation improves further after invoking a convolutional neural network with the full toric diagram of the Calabi-Yau three-folds as the input. We are thereby able to circumvent any minimization procedure that was previously necessary and find an explicit mapping between the minimum volume and the topological quantities of the toric diagram. Under the AdS/CFT correspondence, the minimum volumes of Sasaki-Einstein manifolds correspond to central charges of a class of $4d$ $\mathcal{N} = 1$ superconformal field theories. We therefore find empirical evidence for a function that gives values of central charges without the usual extremization procedure.

I. INTRODUCTION

In recent years, machine learning has become a cornerstone for many fields of science, and it has been adopted more and more as a valuable toolbox. Machine learning has attracted much interest due to significant theoretical progress and due to increased availability of large amounts of data, computing power (GPUs) and easy to use software implementations of standard machine learning techniques.

Despite these developments, applications of machine learning techniques to mathematical physics have been limited to our knowledge. One of the reasons for this is that machine learning aims to empirically approximate the underlying probability density function of a given data set. Making use of machine learning to identify hidden structures in data sets, which teach us about new phenomena in string theory and mathematics, has not been systematically considered before.

This work aims to change the status quo and to provide evidence that machine learning can be used to discover hidden structures in large classes of gauge theories that are studied in theoretical physics as well as large classes of geometries that are studied in mathematics. Importantly, we illustrate that machine learning does not just provide an approximation of known functional relationships between physically and mathematically significant quantities, but also leads to discoveries of new functional relationships.

In particular, we concentrate on a class of $4d$ $\mathcal{N} = 1$ supersymmetric gauge theories that live on the worldvolume of a stack of D3-branes probing toric Calabi-Yau three-folds, characterized by convex lattice polygons known as toric diagrams [1–4]. These theories are expected to flow at low energies to a superconformal fixed point.

From a machine learning perspective, this work studies the minimum volumes of Sasaki-Einstein five-manifolds. These are the base manifolds of the probed toric Calabi-Yau three-folds. The minimum volume is of particular interest because under the AdS/CFT correspondence, it is expected to be related to the maximized a-function that gives the central charge of the $4d$ $\mathcal{N} = 1$ superconformal field theory [5–9].

Using a large data set of toric Calabi-Yau three-folds, our aim is to train a machine learning model in such a way that it approximates a functional relationship between topological quantities of the toric Calabi-Yau three-fold and the minimum volume of the Sasaki-Einstein base manifold. Such a functional relation would be of great use because it would circumvent the standard volume minimization procedure and highlight a direct relationship between topological quantities of the toric Calabi-Yau geometries and the central charges of the $4d$ superconformal field theories.

We ask whether for a given data set consisting of minimal volumes $V_{\min}$ and topological quantities $\mathcal{T}$ of the corresponding Calabi-Yau three-folds, we can approximate from the data a mapping F such that

$$V_{\min} \sim F(\mathcal{T}).$$

Here, we use machine learning to model F. Mainly, we will consider three different machine learning models. These are a modification of usual linear regression, a convolutional neural network (CNN) and also a combination of both. In

detail, a CNN model is a feed-forward neural network which includes additional convolutional layers.

We refer the reader to [10] for a basic introduction on CNN models and [11] for a comprehensive reference list.

II. BACKGROUND

We concentrate on noncompact Calabi-Yau three-folds $\mathcal{X}$ that are realized as affine cones over a complex base X. In particular, we focus on a special subclass of $\mathcal{X}$ where the base is a toric variety $X(\Delta)$, which is defined by a convex lattice polygon Δ known as the toric diagram. $\mathcal{X}$ can also be thought of as the real cone over a compact, smooth Sasaki-Einstein five-manifold Y, whose metrics have been studied extensively for various classes of toric Calabi-Yau three-folds. The Kähler metric of $\mathcal{X}$ has the form

$$ds^2(\mathcal{X}) = dr^2 + r^2 ds^2(Y), \tag{2.1}$$

where $Y = \mathcal{X}|_{r=1}$.

When we talk of Calabi-Yau volumes, we actually refer to the volume function of the Sasaki-Einstein base Y, which takes the form

$$\text{vol}[Y] = \int_Y d\mu = \int_{r \leq 1} \omega^3, \tag{2.2}$$

where $d\mu$ is the Riemannian measure on the cone $\mathcal{X}$. ω is the Kähler form and is given by

$$\omega = -\frac{1}{2} d(r^2 \eta) = \frac{1}{2} i \partial \bar{\partial} r^2, \tag{2.3}$$

where η is a global one-form on Y. Normalized under the volume of S^5, we denote the volume function of Y as

$$V(b; Y) := \frac{\text{vol}[Y]}{\text{vol}[S^5]}. \tag{2.4}$$

Note that $V(b; Y)$ is an algebraic number and is expressed in terms of Reeb vector components $b_{i=1,\dots,3}$, where $b_3 = 3$ [12,13].

For a given projective variety X, realized as an affine variety in $\mathbb{C}^k$, the Hilbert series is the generating function for the dimension of the graded pieces of the coordinate ring $\mathbb{C}[x_1, \dots, x_k]/\langle f_i \rangle$, where f_i are the defining polynomials of X. The Hilbert series of X takes the form of a rational function with the expansion,

$$g(t; \mathcal{X}) = \sum_{i=0}^{\infty} \dim_{\mathbb{C}}(X_i) t^i, \tag{2.5}$$

where the ith graded piece X_i can be thought of as the number of algebraically independent degree i polynomials on the variety X, with t keeping track of the degree i.

When X is a toric variety defined by a convex lattice polygon Δ, the Hilbert series for $X(\Delta)$ and the corresponding Calabi-Yau cone $\mathcal{X}$ can be obtained from the ideal triangulation of the toric diagram Δ,

$$g(t_i; \mathcal{X}) = \sum_{i=1}^{r} \prod_{j=1}^{n} (1 - \vec{t}^{\vec{u}_{i,j}})^{-1}, \tag{2.6}$$

where the index $i = 1, \dots, r$ runs over the unit triangles in the ideal triangulation and $j = 1, 2, 3$ runs over the boundary edges of each such triangle [12,14]. $\vec{u}_{i,j}$ is a three-dimensional outer normal to the edge j of the associated unit triangle i, where $\vec{t}^{\vec{u}_{i,j}} = \prod_a^3 t_a^{u_{i,j}(a)}$. Note that we are dealing with 3-vectors because the two-dimensional toric diagram is on a plane at height 1. Using (2.6), the Hilbert series for $\mathbb{C}^3/\mathbb{Z}_3$, whose toric diagram is shown in Fig. 1, can be obtained as follows:

$$g(t_i; \mathbb{C}^3/\mathbb{Z}_3) = \frac{1}{(1 - t_2)(1 - t_1^{-1}t_2)(1 - t_1 t_2^{-2} t_3^{-1})}$$
$$+ \frac{1}{(1 - t_1)(1 - t_1 t_2^{-1})(1 - t_1^{-2} t_2 t_3)}$$
$$+ \frac{1}{(1 - t_1^{-1})(1 - t_2^{-1})(1 - t_1 t_2 t_3^{-1})}. \tag{2.7}$$

The volume function can be derived directly from the Hilbert series of $\mathcal{X}$ following the limit

$$V(b_i; Y) = \lim_{\mu \to 0} \mu^3 g(t_i = \exp[-\mu b_i]; \mathcal{X}). \tag{2.8}$$

The leading order in μ picked up by the above limit from the expansion of the Hilbert series was shown in [12,13] to be directly related to the volume of the Sasaki-Einstein base
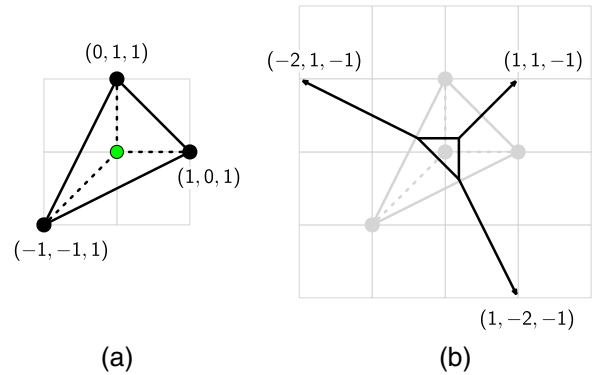

(a)

(b)

FIG. 1. (a) shows the toric diagram for $\mathbb{C}^3/\mathbb{Z}_3$ and the corresponding ideal triangulation into unit area triangles. (b) is the corresponding dual web-diagram with normal vectors to each boundary edges of the triangulation. Notice that we have 3-vectors, given that the original toric diagram is on a plane at height 1.

$Y = \mathcal{X}|_{r=1}$. For the $\mathbb{C}^3/\mathbb{Z}_3$ example, the volume function takes the form

$$V(b_i; \mathbb{C}^3/\mathbb{Z}_3)$$

$$= \frac{-9}{(b_1 - 2b_2 - 3)(-2b_1 + b_2 - 3)(b_1 + b_2 - 3)},$$

(2.9)

where $b_3 = 3$.

III. VOLUME MINIMIZATION AND THE ADS/CFT CORRESPONDENCE

The world volume theory on a stack of D3-branes probing Calabi-Yau three-folds $\mathcal{X}$ is a $4d$ $\mathcal{N} = 1$ supersymmetric gauge theory. It is expected that these theories flow at low energies to a superconformal fixed point. The superconformal R-charges of the theory are determined by a procedure known as a-maximization [7–9], which involves the maximization of the a-charge,

$$a(R; Y) = \frac{3}{32}(3\mathrm{Tr}R^3 - \mathrm{Tr}R).$$

(3.1)

a-maximization gives the value of the central charge at the conformal fixed point, which is by the AdS/CFT correspondence related to the volume minimum of the corresponding Sasaki-Einstein five-manifold [5,6] under

$$a(R; Y) = \frac{\pi^3 N^2}{4V(R; Y)},$$

(3.2)

where the R-charge R can be expressed in terms of Reeb vector components b_i. In other words, computing the minimum volume,

$$V_{\min} = \min_{b_i|b_3=3} V(b_i; Y),$$

(3.3)

is equivalent under (3.2) to computing the maximized value of $a(R; Y)$, which is the central charge of the $4d$ $\mathcal{N} = 1$ superconformal field theory.

IV. DATA

Our aim is to train a neural network to compute the volume minimum directly as a function of toric data, circumventing the minimization procedure that has been so far necessary. The available input data for the machine learning models for a given toric Calabi-Yau three-fold takes the following form,

$$(y, \mathcal{T}), \qquad \mathcal{T} = (f_1, f_2, f_3, \mathcal{D}),$$

(4.1)

where $y = 1/V_{\min}$ is the target inverse minimum volume, and f_i are the three features

$$f_1 = I, \qquad f_2 = E, \qquad f_3 = V,$$

(4.2)

with I being the number of internal lattice points, E being the number of perimeter points and V being the number of extremal corner points of the convex lattice polygon representing the toric diagram. Note that $2f_2 - 4$ is the Euler number of the corresponding toric variety [15]. In addition, we include the toric diagram itself as a square matrix $\mathcal{D}$, consisting of 0,1 entries, where an entry of 1 indicates the presence of an extremal vertex of the lattice polygon.

We generate a class of toric Calabi-Yau's whose toric diagrams originate from the toric diagram of the orbifold of the conifold of the form $\mathcal{C}/\mathbb{Z}_5 \times \mathbb{Z}_5$, which is a lattice square with side-length 5. By consecutively cutting corners of this toric diagram, we generate 187, 389 distinct toric diagrams. However, this set of toric diagrams exhibits a remaining $GL(2, \mathbb{Z})$ redundancy and, hence, certain toric diagrams from this set can be related to the same toric Calabi-Yau three-fold. We therefore remove the $GL(2, \mathbb{Z})$ redundancy and further reduce the number of toric diagrams down to 15,151, which now establishes a set of distinct toric Calabi-Yau three-folds. Using the integer rounded centroid of the convex lattice polygons, we recenter the toric diagrams. All the 15,151 recentered toric diagrams then fit into a 7×7 lattice square, which we further embed into a 9×9 lattice square. Accordingly, $\mathcal{D}$ for our data set is a 9×9 integer matrix with entries 0,1. In Fig. 2, we illustrate the distribution of the extremal vertices of all the 15,151 toric diagrams we use for our analysis.
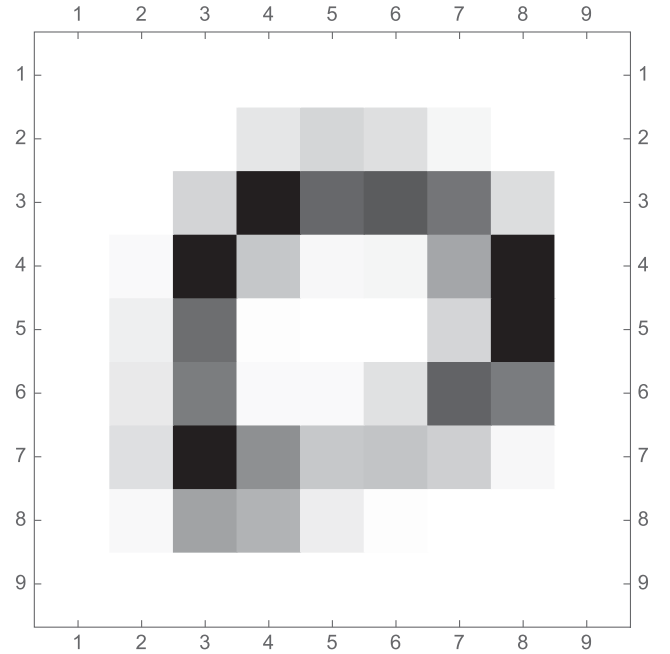

FIG. 2. The distribution of extremal vertices of toric diagrams for the set of 15,151 distinct toric Calabi-Yau three-folds that are used as train and test sets for our machine learning models.

Using the Hilbert series, we compute the volume function $V(b_i; Y)$ for our data set and minimize them to obtain V_{min}. Given the entire data set with the minimized volumes, we identify four cases where the value of $y = 1/V_{min}$ is much larger than the remaining data set. In order to keep a nondistorted data set, we remove these four cases ending up with a data set of size 15,147. We also note that the minimum volumes are algebraic numbers, which can be irrational, and that we round the actual values for the machine learning model to four decimal points. This gives us 797 distinct values for the minimum volume under the chosen numerical precision, corresponding to the 15,147 distinct toric Calabi-Yau three-folds. Finally, for each toric diagram in our set of Calabi-Yau three-folds, we identify the features f_i that lead us to 15,147 input data of the form shown in (4.1). An example is illustrated in Fig. 3.

As neural networks easily overfit, meaning that they tend to memorize the specific training set instead of learning the underlying hidden structure, we have to be very careful in data preparation and usage in order not to fool ourselves. We here follow the common approach to split the data into an independent train (75%) and test (25%) set, where the machine learning models are only trained on the train set. We do not use an additional verification set, as we will not perform extensive hyper-parameter optimization. It is important to note that we have constructed our data set

of 15,147 toric Calabi-Yau three-folds in such a way that a split will give $GL(2, \mathbb{Z})$ independent train and test sets.

V. MULTIPLE LINEAR REGRESSION

Let us first see how well we can model the data of the three features given in (4.2) via a simple multiple linear regression, i.e.,

$$y^{(n)} \sim F(f^{(n)}) = \sum_{i=1}^{k_f} \omega_i f_i^{(n)} + \omega_0, \qquad (5.1)$$

where ω_i denotes the ith weight, and the data set is given by (y, f), with f being the features for target y. The weight ω_0 is usually referred to as the bias and can be viewed as the weight of an additional feature fixed to 1. Furthermore, we improve the modeling nonlinearly by taking order-two combinations of the $k_f = 3$ original features, which yields

$$\hat{k}_f = 2k_f + \frac{k_f(k_f - 1)}{2} \qquad (5.2)$$

new features of the form

$$\hat{f} = \left(f_1, ..., f_{k_f}, f_1 f_2, f_1 f_3, ..., f_1^2, ..., f_{k_f}^2\right). \qquad (5.3)$$

Figure 4 illustrates the setup of the linear regression model.

The optimization task is the usual mean least squares minimization

$$\underset{\omega}{\text{argmin}} \, \mathcal{L}, \qquad (5.4)$$

where

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \left(y^{(n)} - F\left(\hat{f}^{(n)}\right)\right)^2. \qquad (5.5)$$

$1/V_{min} = 28.6533$, $f_1 = I = 13$, $f_2 = E = 10$, $f_3 = V = 6$

$$\mathcal{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
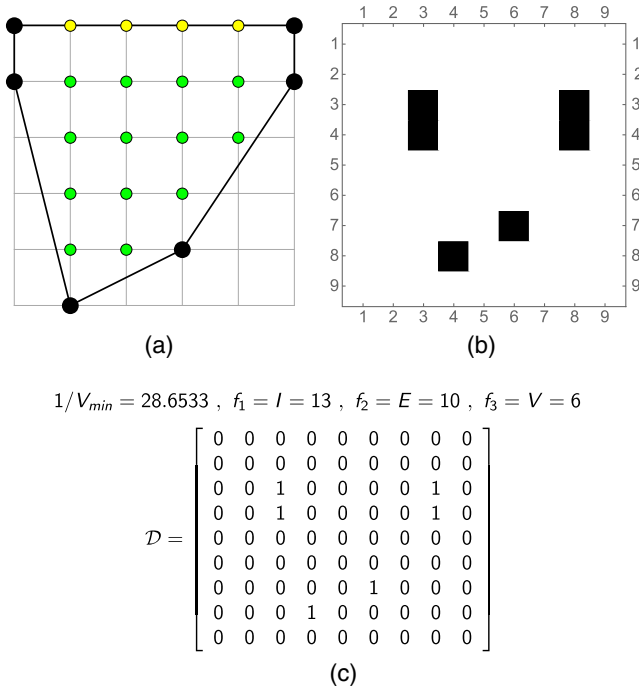
(c)

FIG. 3. An example of a data entry vector for a particular toric Calabi-Yau three-fold in our data set. (a) shows the toric diagram obtained by cutting corners of a 5×5 lattice square and (b) shows the corresponding extremal lattice points of the toric diagram embedded in a 9×9 input matrix. The full data vector for this toric Calabi-Yau three-fold is summarized in (c).
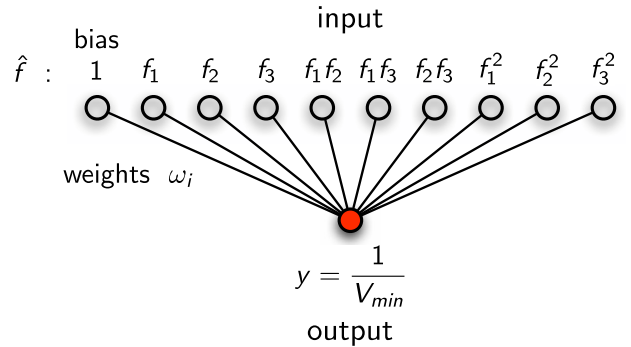

FIG. 4. Illustration of the linear regression model. The features are weighted by the respective weights and added to form the output. The bias weight w_0 is illustrated as an additional feature, which is fixed to 1.

Though one can solve for ω_i exactly due to the convexity of the optimization problem, we prefer here to solve iteratively via stochastic gradient descent using the Python package Keras [16] (with Theano [17] backend) and the Nadam optimizer in default settings running for 5000 epochs (with batch size 1000). Note that we will use the same software stacks and settings in the following section. The solution for the training set obtained via gradient descent and additional L_1 regularization of (5.5) $(+0.01 * |w|)$ reads (up to four digits)

$$\omega_1 = 1.9574, \quad \omega_2 = 0.8522, \quad \omega_3 = -0.7658,$$
$$\omega_4 = -0.0138, \quad \omega_5 = -0.0020, \quad \omega_6 = -0.0104,$$
$$\omega_7 = -0.0120, \quad \omega_8 = -0.0523, \quad \omega_9 = -0.0478,$$
$$\omega_0 = 1.3637. \tag{5.6}$$

One should note that the features $f^{(n)}$ are not unique in our data set. In fact, we have only 645 unique feature combinations for all the toric diagrams in our test data set. Hence, we expect that the solution Ansatz (5.2) built on the three features models the statistical expectation $E(y)$ (mean). After categorizing the test data set into 645 categories $\mathcal{C}_i$, we take the expectation $E(y^{\mathcal{C}_i})$ of the data sets in each category. Furthermore, we calculate the largest and smallest $y^{\mathcal{C}_i}$ in each category and order the categories according to their $E(y^{\mathcal{C}_i})$. Then we plot the minimum and maximum of each category as well as $F(f^{\mathcal{C}_i})$ against $E(y^{\mathcal{C}_i})$, leading to the plot in Fig. 5. Note that the diagonal of the figure corresponds to $E(y^{\mathcal{C}})$. We observe that the prediction of $F(f^{\mathcal{C}})$ (red curve) indeed seems to roughly approximate the mean of the y values in each of the categories.

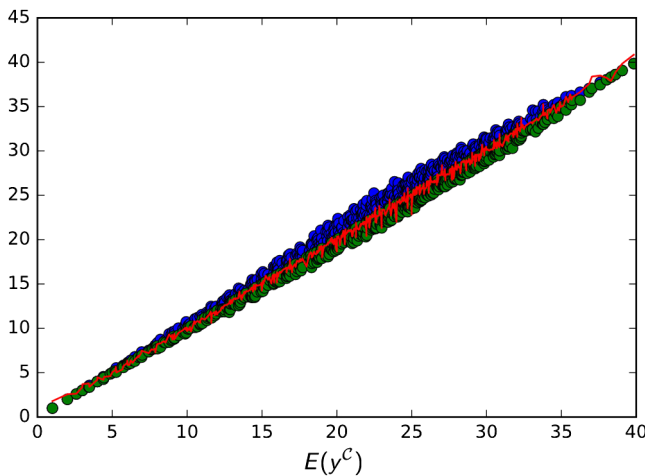How well does this simple Ansatz actually model the true y, or rather the minimum volume $1/y$ of interest? We

FIG. 5. The x axis corresponds to $E(y^{\mathcal{C}})$ of the 645 categories. The red curve plots the prediciton of y via linear regression for the classes. The blue dots indicate the maximum y taken for a class of values $\mathcal{C}$ and the green dots the minimum value.

run the test set through the predictor F and calculate the percentage errors (in $\times 100\%$)

$$\epsilon^{(n)} = 1 - \frac{y^{(n)}}{F(f^{(n)})}. \tag{5.7}$$

We observe a maximum error $|\epsilon_{\max}| \sim 0.132$ and

$$E(|\epsilon|) \sim 0.022, \qquad \sigma(|\epsilon|) \sim 0.017, \tag{5.8}$$

where $\sigma(|\epsilon|)$ denotes the standard deviation of the distribution of the absolute value of the percentage errors. We also averaged the values over three independent runs.

Hence, the expected prediction error of the minimal volume is around 2.2%. This means that the linear regression Ansatz already yields a surprisingly good approximator to the minimal volume. Here we should make a remark regarding the order of feature combinations that was taken in (5.3). Taking order-two combinations gives a significant improvement in reducing the error in comparison to taking the plain three features, as the extended linear regression model seems to be able to learn better the more extreme volumes at the tails of the distribution of the minimum volumes. Including in addition order-three combinations does not seem to yield any further improvement but rather seems to lead to a worse result.

In general, in order to improve this method further, we need to introduce additional features which are able to distinguish between the individual members of a class $\mathcal{C}_i$. However, instead of hand-crafting new features, we try in the following section the more modern approach to let the approximator F learn the appropriate features *itself* from the raw data.

VI. WIDE AND CNN

The raw data in its purest form is the toric diagram itself, hence it behooves us to ask if we can learn additional information directly from the toric diagram, in order to minimize further the prediction error. Since we treat the toric diagram as a 9×9 matrix viewed as an image, the canonical computer science approach is to invoke a CNN–the main tool for image recognition.

We couple the linear regression, as described in the previous section, with a CNN as follows. The output of the linear regression is added via a single ReLu unit (rectified linear, $\max(x, 0)$) to the outputs of the CNN, i.e.,

$$y^{(n)} \sim \max\left(\omega_1^f F(f^{(n)}) + \sum_{i=1}^{o} \omega_{2i}^f m_i(\mathcal{D}^{(n)}) + \omega_0^f, 0\right), \tag{6.1}$$

where m_i denotes the ith output of the CNN and ω^f are the weights of the final layer. We use the ReLu unit for the final output because the minimum volume has to be positive.

Note that this kind of setup is also known as a wide and deep model.

The CNN is setup as follows. For the input layer we take a $2d$ convolutional layer consisting of 32 filters (size 3×3) and linear activation. The filters are convolved against the input and produce a $2d$ activation map of the filter. Hence, the layer learns spatially localized features of the inputs. For an illustration of the convolution layers see Fig. 9. This is followed by two relatively small fully connected layers (we use sizes 12 and 4) with tanh activation. Hence we have four outputs m_i in the CNN part. The precise network architecture is not of utmost relevance, as the results appear to be relatively stable against modifications in the number of layers, units in each layer, etc. However, generally smaller networks seem to be preferred with tanh activation functions in the dense layers. Figure 6 illustrates the combined setup.

The complete setup is trained on the train set as in the previous section via stochastic gradient descent minimizing the mean squared error.

Using the trained network, the prediction of the volume minima for the independent test set exhibits the following errors averaged over three independent test runs,

$$E(|\epsilon|) \sim 0.009, \qquad \sigma(|\epsilon|) \sim 0.009.$$

Hence, the expected prediction error is below 1%. The maximum observed error reads $|\epsilon_{\max}| \sim 0.20$. The distribution of errors for one test run is plotted in Fig. 7. We conclude that adding the CNN yields a significant improvement in predictive power. Note that the few larger errors
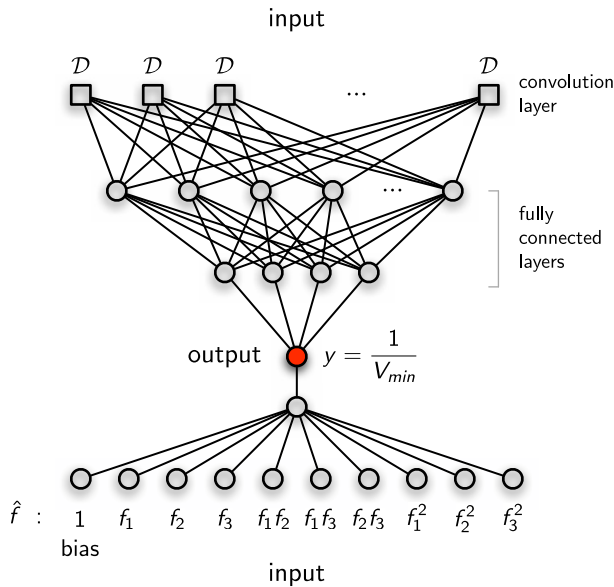

FIG. 7. The x axis corresponds to the minimum volume $V_{\min}$, and the y axis to the percentage error ε ($\times 100\%$). The blue dots correspond to the errors between the prediction and should be results for the coupled linear regression and CNN.

visible in the plot are due to the tails of the minimum volume distribution, which the model is not able to predict extremely well due to the lack of training data available at the extreme values.

Finally, let us consider the case of using just the CNN alone, without being coupled to a linear regression branch. The used CNN is identical to the one above.

We obtain on the test set

$$E(|\epsilon|) \sim 0.010, \qquad \sigma(|\epsilon|) \sim 0.014,$$

and $|\epsilon_{\max}| \sim 0.51$. Note that these values are averaged as well over three independent test runs. For illustration, the individual errors for one test run are plotted in Fig. 8.

In general, the machine learning models have greater difficulty in learning the tails of the minimum volume distribution. This might be due to lack of data in this regime

FIG. 6. The wide and deep model. The toric diagram data $\mathcal{D}$ is fed into a convolutional layer and further processed in two fully connected layers. The outputs are linearly combined with the output of a linear regression on the features $\hat{f}$.
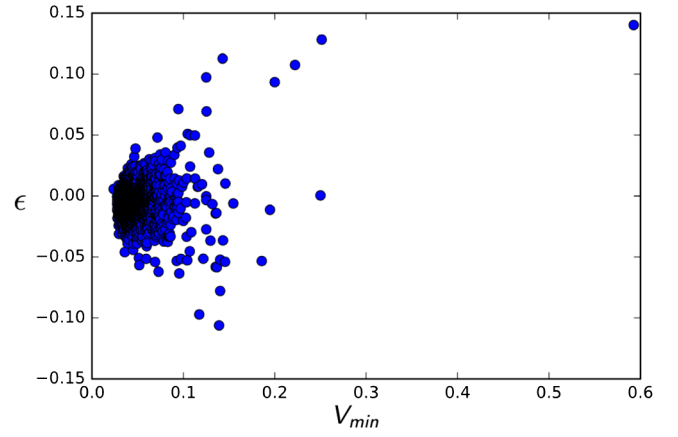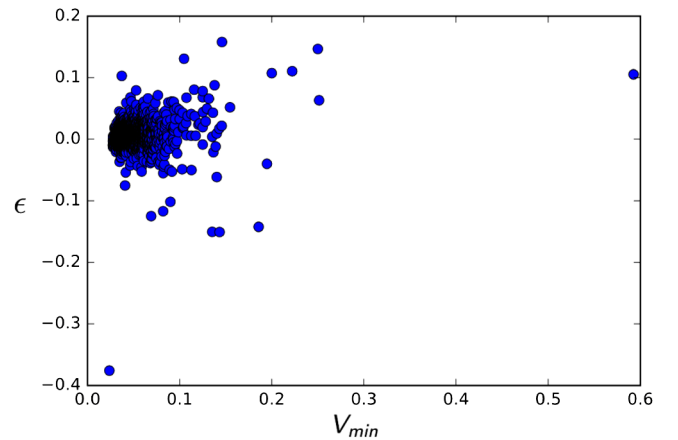

FIG. 8. The x axis corresponds to the minimum volume $V_{\min}$ and the y axis to the percentage error ε ($\times 100\%$). The blue dots correspond to the errors between the prediction and should be results for the pure CNN.
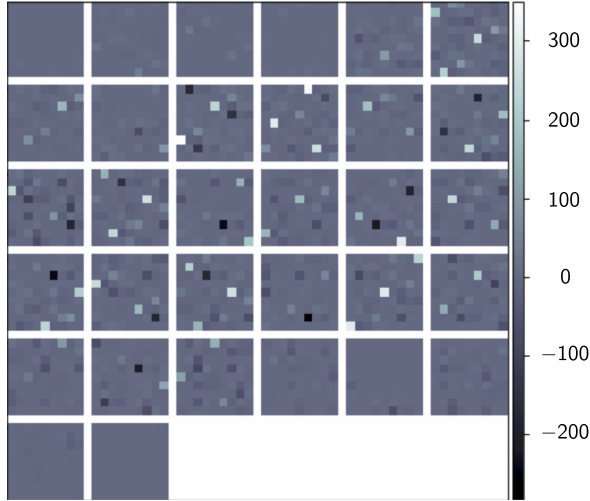
FIG. 9. An illustration of the 32 convolutional filters acting on the average input in the CNN. The color shading refers to the output of the convolution layer.

(the data distribution restricted to our frame is not uniform). We observe that the combined setup of both linear regression and CNN performs better. This is because linear regression seems to stabilize the prediction of the tail sections of the minimum volume distribution due to its knowledge about the properties of the feature vector classes, which we discussed in the previous section.

VII. SUMMARY AND OUTLOOK

In this work, we have demonstrated that machine learning techniques, in particular neural networks, can be a useful addition to the toolbox of researchers tackling formal questions in mathematics and physics. A necessary condition for using machine learning techniques is that at least some aspect of the question can be translated to a data science problem.

This work studied whether the minimum volume of Sasaki-Einstein base manifolds for toric Calabi-Yau three-folds can be directly computed from topological quantities originating from toric geometry, replacing the usual minimization procedure that is necessary to identify the minimum volume. This question has aspects of a data science problem, i.e., out of known topological data and the corresponding minimal volumes, can we (or rather the machine) learn a mapping between these quantities (that is, find an approximate functional relation)?

The answer seems to be affirmative. Even taking for the machine learning model just a linear combination of order-two combinations of numbers for different characteristic vertices in the toric diagrams of the Calabi-Yau three-folds, yields already a good universal approximation to the minimal volume of the corresponding Sasaki-Einstein manifolds. In addition, a CNN model, which learns new

kinds of features of the toric diagram, further improves predictive power for the volume minimum.

It is surprising to see that the simple setups we are considering are able to predict the minimum volumes relatively well, effectively showing that the minimum volume is encoded in the toric diagrams of the Calabi-Yau three-folds. This is an indication that the procedure of volume minimization can be avoided. In fact, we show that the volume minimum can explicitly be computed from the toric data, by using an underlying functional relation, which we have approximated in this work.

With this analysis, we have shown a working example of how a machine learning model can identify functional relationships between mathematically and physically interesting quantities in cases where such functional relationships were not known before.

Note that it would be interesting to refine our analysis by increasing the data set of toric Calabi-Yau's in such a way that the minimum volume distribution is more uniform. Furthermore, averaging over more test runs as well as increasing the rounding precision for irrational values for minimum volumes would be important improvements that we leave for future work.

We believe that there are other suitable problems which can be approached from a data science perspective, similar to what we have done in this work for the minimum volume of toric Calabi-Yau three-folds. Approaching problems in this way might yield some novel insights, as well as hints to hidden and unexpected relations between physically and mathematically relevant quantities that have not been observed before.

For example, large data sets of both physical and mathematical significance exist in the context of $4d$ $\mathcal{N} = 1$ theories related to toric Calabi-Yau three-folds [1–4], as well as to a recently discovered new class of $2d$ $(0,2)$ theories related to toric Calabi-Yau four-folds [18]. Furthermore, interesting rich data sets exist in relation to so called complete intersection Calabi-Yaus (CICYs) [19] characterized by configuration matrices that can be taken as inputs for machine learning models. Finally, large data sets exist in relation to hyperbolic three-manifolds related to knots [20], which may exhibit hidden structures that could be discovered using again machine learning techniques. In a future work [21], we hope to shed light on these interesting problems.

ACKNOWLEDGMENTS

[1] I. R. Klebanov and E. Witten, Superconformal field theory on three-branes at a Calabi-Yau singularity, Nucl. Phys. **B536**, 199 (1998).

[2] A. Hanany and A. Zaffaroni, On the realization of chiral four-dimensional gauge theories using branes, J. High Energy Phys. 05 (1998) 001.

[3] A. Hanany and A. M. Uranga, Brane boxes and branes on singularities, J. High Energy Phys. 05 (1998) 013.

[4] S. Franco, A. Hanany, K. D. Kennaway, D. Vegh, and B. Wecht, Brane dimers and quiver gauge theories, J. High Energy Phys. 01 (2006) 096.

[5] S. S. Gubser, Einstein manifolds and conformal field theories, Phys. Rev. D **59**, 025006 (1998).

[6] M. Henningson and K. Skenderis, The holographic Weyl anomaly, J. High Energy Phys. 07 (1998) 023.

[7] K. A. Intriligator and B. Wecht, The exact superconformal R symmetry maximizes a, Nucl. Phys. **B667**, 183 (2003).

[8] A. Butti and A. Zaffaroni, R-charges from toric diagrams and the equivalence of a-maximization and Z-minimization, J. High Energy Phys. 11 (2005) 019.

[9] A. Butti and A. Zaffaroni, From toric geometry to quiver gauge theory: The Equivalence of a-maximization and Z-minimization, Fortschr. Phys. **54**, 309 (2006).

[10] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[11] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Netw. **61**, 85 (2015).

[12] D. Martelli, J. Sparks, and S.-T. Yau, The geometric dual of a-maximisation for Toric Sasaki-Einstein manifolds, Commun. Math. Phys. **268**, 39 (2006).

[13] D. Martelli, J. Sparks, and S.-T. Yau, Sasaki-Einstein manifolds and volume minimisation, Commun. Math. Phys. **280**, 611 (2008).

[14] S. Benvenuti, B. Feng, A. Hanany, and Y.-H. He, Counting BPS operators in gauge theories: Quivers, syzygies and plethystics, J. High Energy Phys. 11 (2007) 050.

[15] Y.-H. He, R.-K. Seong, and S.-T. Yau, Calabi-Yau volumes and reflexive polytopes, arXiv:1704.03462.

[16] F. Chollet et al., Keras, GitHub, 2015, https://github.com/fchollet/keras.

[17] Theano Development Team, Theano: A Python framework for fast computation of mathematical expressions, arXiv:1605.02688.

[18] S. Franco, S. Lee, and R.-K. Seong, Brane brick models, toric Calabi-Yau 4-folds and 2d (0,2) quivers, J. High Energy Phys. 02 (2016) 047.

[19] P. Candelas, A. M. Dale, C. A. Lutken, and R. Schimmrigk, Complete intersection Calabi-Yau manifolds, Nucl. Phys. **B298**, 493 (1988).

[20] V. F. Jones, A polynomial invariant for knots via von Neumann algebras, Bull. Am. Phys. Soc. **12**, 103 (1985).

[21] R.-K. Seong and S.-T. Yau (to be published).