

PAPER • OPEN ACCESS

Numerical metrics for complete intersection and Kreuzer–Skarke Calabi–Yau manifolds

To cite this article: Magdalena Larfors *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 035014

View the [article online](#) for updates and enhancements.

You may also like

- [F-theory duals of M-theory on \$G_2\$ manifolds from mirror symmetry](#)
Adil Belhaj
- [Worldsheet instantons and torsion curves. Part B: mirror symmetry](#)
Volker Braun, Burt A. Ovrut, Maximilian Kreuzer et al.
- [Complete free energy landscape and statistical thermodynamics of single poly\(ethylene glycol\) molecules](#)
Douglas Staple, Felix Hanke and Hans Jürgen Kreuzer



WORLD LEADING
MOLECULAR
SPECTROSCOPY SOLUTIONS



edinst.com



PAPER

OPEN ACCESS

RECEIVED
9 June 2022REVISED
8 August 2022ACCEPTED FOR PUBLICATION
31 August 2022PUBLISHED
20 September 2022

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Numerical metrics for complete intersection and Kreuzer–Skarke Calabi–Yau manifolds

Magdalena Larfors^{1,2,*} , Andre Lukas³, Fabian Ruehle^{4,5} and Robin Schneider² ¹ Department of Mathematical Sciences, Durham University, Upper Mountjoy Campus, Stockton Rd, Durham DH1 3LE, United Kingdom² Department of Physics and Astronomy, Uppsala University, SE-751 20 Uppsala, Sweden³ Rudolf Peierls Centre for Theoretical Physics, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom⁴ Department of Physics & Department of Mathematics, Northeastern University, 360 Huntington Avenue, Boston, MA 02115, United States of America⁵ The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Boston, MA, United States of America

* Author to whom any correspondence should be addressed.

E-mail: magdalena.larfors@physics.uu.se**Keywords:** metrics, Kreuzer–Skarke, Calabi–Yau, CICY, machine learning, string theory

Abstract

We introduce neural networks (NNs) to compute numerical Ricci-flat Calabi–Yau (CY) metrics for complete intersection and Kreuzer–Skarke (KS) CY manifolds at any point in Kähler and complex structure moduli space, and introduce the package `cymetric` which provides computation realizations of these techniques. In particular, we develop and computationally realize methods for point-sampling on these manifolds. The training for the NNs is carried out subject to a custom loss function. The Kähler class is fixed by adding to the loss a component which enforces the slopes of certain line bundles to match with topological computations. Our methods are applied to various manifolds, including the quintic manifold, the bi-cubic manifold and a KS manifold with Picard number two. We show that volumes and line bundle slopes can be reliably computed from the resulting Ricci-flat metrics. We also apply our results to compute an approximate Hermitian–Yang–Mills connection on a specific line bundle on the bi-cubic.

1. Introduction

Calabi–Yau (CY) manifolds arise naturally in the context of string theory [1]. Since the 80s mathematicians and physicists alike have invested a considerable amount of effort in constructing and studying these spaces. By now there are different popular constructions, such as complete intersection hypersurfaces in products of complex projective spaces [2, 3] (CICYs) or more generally in toric varieties [4]. CY three-folds constructed as hypersurfaces in toric four-folds are described by about half a billion reflexive polytopes (giving rise to many more topologically inequivalent manifolds) which we refer to as the Kreuzer–Skarke (KS) list. Much is known about their topological properties [5], such as their Chern classes, intersection numbers or Hodge numbers. However, objects related to their differential geometry are much less accessible. Crucially, the unique Ricci-flat metric on CY manifolds, established by theorems of Calabi [6] and Yau [7], is not known in analytic form on any compact CY three-folds, although there is recent progress for K3 [8, 9].

The Ricci-flat CY metric is of great importance to string theorists, as it plays a role in many computations connecting string theory with observable physics. For example, the CY metric is required to compute the physical Yukawa couplings from string theory [10] or it can be used to study the physics which arises along geodesics in moduli space, in order to probe the swampland distance conjecture [11]. There are also mathematical problems which benefit from knowledge of the CY metric, including the SYZ-conjecture [12], which aims to formalize mirror symmetry [13]. Given the importance of the Ricci-flat CY metric and the lack of analytic expressions it is only natural that several numerical approximation schemes have been advanced.

The first work in this direction relies on an algorithm set up by Donaldson [14]. This algorithm starts with an approximation for the Kähler potential underlying the Ricci-flat metric which is of the form

$K = \ln(s^i h_{ij} \bar{s}^j)$, where s^i is a basis of sections for the k th power, L^k , of a line bundle $L \rightarrow X$ on the CY manifold X and h_{ij} is a hermitian matrix. For each power k an optimal K for a balanced metric can be found by a fix point method which computes the matrix h_{ij} . In the limit $k \rightarrow \infty$ these balanced metrics converge to the Ricci-flat CY metric with Kähler class proportional to $c_1(L)$, the first Chern class of the line bundle. However, the scaling of this method with k makes it unsuitable for cases without a high degree of symmetry or with many Kähler parameters. In particular the algorithm requires a factorial increase in training data for a sub-linear increase in accuracy [15]. Another practical problem is the link between the Kähler class of the Ricci-flat metric and the lattice-valued quantity $c_1(L)$ which makes it difficult to determine the metric as a function of the Kähler parameters. Nevertheless this method has been used to construct metrics on the Fermat Quintic and other selected examples [15–19] and it has been generalized to compute the solution to the Hermitian Yang–Mills (HYM) equation for vector bundles [20, 21].

Scaling of the computation can be improved by a method which relies on energy functionals [22]. This method also uses a polynomial Ansatz for the (Kähler potential of the) metric and optimizes polynomial coefficients by minimizing the deviation from the Monge–Ampère equation. In this way solutions with exponential accuracy are found. The code⁶ published alongside [22] for CY hypersurfaces of Fermat type has been used in more recent studies [23] to investigate the curvature near conifold points and to accurately predict the Kähler potential [24].

The rise of neural networks (NNs) as universal function approximators has renewed the interest in finding numerical approximations of the Ricci-flat CY metric. They have been introduced into the string community in 2017 [25–28] (see [29] for a review) and by now several directions to use techniques from machine learning for CY metrics have emerged. Decision trees and curve fitting have been used to bootstrap from low order Donaldson algorithms to more accurate solutions [30]. Holomorphic NNs are shown to learn much better and more efficient approximations of the Kähler potential [31]. Finally, the metric has been learned directly with NNs in [32–34]. These new numerical methods allowed for direct application to test the swampland distance conjecture [11] and construct solutions to the HYM equation of line bundles [35].

While there has been progress on methods to compute the Ricci-flat CY metric, there are also still considerable gaps, particularly in view of providing systematic tools, for example in the context of string model building. Computations have been carried out for select manifolds but a tool which covers large classes of CY manifolds is still missing. This requires an understanding of some theoretical problems, including methods for point sampling on more general CY manifolds. A critical issue for applications is the ability to compute the CY metric at a given point in moduli space. This tends to be straightforward for the complex structure moduli which are usually fixed by the equations defining the CY manifold. For a single Kähler modulus, $h^{1,1}(X) = 1$, such as for the quintic, things are simple as every Kähler class can be obtained by a simple re-scaling of the metric. For $h^{1,1}(X) > 1$ this is no longer the case and care has to be taken in order to fix the desired Kähler class. While this is possible in principle by Donaldson’s algorithm, computations are limited by inefficiency and the limitation to integral Kähler classes. On the other hand, NN approaches which solely rely on minimizing the Ricci curvature or deviations from the Monge–Ampère equation produce a metric with an unspecified Kähler class.

The point of the present paper is to address and resolve these issues and provide both the theoretical background and practical methods for machine-learning Ricci-flat CY metrics at a prescribed point in moduli space, for any manifold from both the CICY and KS lists. This is extending and detailing recent work by the authors in [34]. In particular, we explain how the problem of point sampling on CY manifolds can be solved for both CICY and KS manifolds. We fix the Kähler class by matching line bundles slopes, which can be computed using topological data only, to their numeric evaluation using the approximate CY metric. For a given Kähler class J , we consider a basis L_i of line bundles and determine the slopes $\mu_J(L_i)$ using a topological formula. During training, a numerical version of these slopes, $\mu_J^{\text{num}}(L_i)$, is computed by numerical integration and a loss proportional to $\sum_i \|\mu_J(L_i) - \mu_J^{\text{num}}(L_i)\|_n$ is added to the loss function. As we will demonstrate, this does indeed lead to an approximately Ricci-flat metric with the prescribed Kähler class J .

These methods are realized in the `cymetric` package [36] which provides machine learning tools to find the Ricci-flat CY metric for both CICY and KS manifolds. The package is written in Python and Mathematica [37]. The point generation relies on routines from NumPy [38] and SciPy [39]. For KS manifolds we employ routines implemented in SageMath [40] and Mathematica [37]. The NNs are implemented and optimized with TensorFlow [41], which has efficient tracing for several nested gradient operations using automatic differentiation. The code is available at: <https://github.com/pythoncymetric/cymetric>.

⁶ <https://people.brandeis.edu/~headrick/Mathematica/index.html>.

Open source software has led to rapid progress in machine learning. Our goal is to achieve similarly remarkable results by open sourcing our code and we welcome contributions. There are several other open source packages developed by the string theory community. For example the toric geometry routines of SageMath [42] are an integral part of the workflow for many geometers. They rely on the PALP package [43] and are a necessary component in our toric point generator. The functionality of these routines have recently been expanded by the CYTools package [44]. Other open source packages include Stringvacua [45] for studying potentials from string theory and cohomcalc [46] and pyCICY [47] for studying line bundle cohomologies over CY manifolds.

The outline of this paper is as follows. In section 2 we will introduce the necessary mathematical background for the computation of the Ricci-flat metric, including the construction of the required Kähler forms and a discussion of the point sampling method. Section 3 reviews some basic mathematics related to HYM connections on line bundles and the numerical method for computing such connections. A brief review of NNs and an overview of our network architecture follows in section 4. Results for the quintic and bi-cubic CY manifolds as well as a Picard number two manifold from the KS list are presented in section 5 and we conclude in section 6. Some useful details on toric geometry are collected in [appendix](#).

2. Mathematical background

In this section we introduce the mathematical background necessary for sampling points and calculating the CY metric. For our procedure we require three pieces of data for the CY manifold: its holomorphic top form Ω , a Kähler form J , and the affine patches plus their transition functions. We collect here only the important formulas and provide an introduction to the necessary concepts in [appendix](#). We adopt a convention where we add subscripts FS and CY to quantities related to the ambient Fubini–Study metric (such as the metric itself or its associated Kähler potential) and the Ricci-flat CY metric, respectively. We do not distinguish between ambient space quantities and their restriction to not clutter the notation further and it should be clear from the context which one we are referring to. Our formulae will be written for CY three-folds, but they generalize straight-forwardly to CY n -folds⁷. We denote real CY coordinates on a patch by y^m , where $m = 1, \dots, 6$ and their complex counterparts by

$$z^a = \frac{1}{\sqrt{2}}(y^a + iy^{a+3}), \quad \bar{z}^{\bar{a}} = \frac{1}{\sqrt{2}}(y^{\bar{a}} - iy^{\bar{a}+3}), \quad (1)$$

where $a = 1, 2, 3$ and $\bar{a} = 1, 2, 3$.

2.1. Kähler forms and volumes

To start, let us recall that the CY theorem [6, 7] states that any Kähler n -fold X with $c_1(TX) = 0$ and given Kähler form J' admits a unique Ricci-flat metric g_{CY} (simple called CY metric, in the following) whose associated Kähler form J_{CY} (where $J_{CY, a\bar{b}} = ig_{CY, a\bar{b}}$ for $a, \bar{b} = 1, 2, 3$) is in the same cohomology class as J' . For three-folds, the Kähler form J_{CY} can be determined from the Monge–Ampère equation

$$J_{CY} \wedge J_{CY} \wedge J_{CY} = \kappa \Omega \wedge \bar{\Omega} \quad \text{with} \quad J_{CY} = J' + \partial\bar{\partial}\phi \quad (2)$$

where ϕ is a real function on X and κ is a complex constant (which may depend on moduli). The Kähler form J' can be any reference Kähler form in the given Kähler class but for our purposes we will take this to be the obvious Kähler form induced from the toric ambient space, that is the (generalized) Fubini–Study Kähler form denoted by $J' = J_{FS}$. It is usually written as a linear combination

$$J_{FS} = t^\alpha J_\alpha \quad (3)$$

relative to a basis J_α of $(1, 1)$ forms (where $\alpha = 1, \dots, h^{1,1}(X)$), also obtained by restriction from the ambient space. Here t^α are the Kähler parameters. Associated to these Kähler forms are volume forms

$$dVol_{CY} = \frac{1}{3!} J_{CY}^3, \quad dVol_{FS} = \frac{1}{3!} J_{FS}^3 \quad (4)$$

and a typical problem is to compute integrals of the type

$$\int_X dVol_{CY} f = \int_X d^6 y \sqrt{g_{CY}} f, \quad \int_X dVol_{FS} f = \int_X d^6 y \sqrt{g_{FS}} f, \quad (5)$$

⁷ The code supports CY manifolds up to complex dimension 6.

for some function f on X . We will see later how numerical results for J_{CY} (and, indeed, J_{FS}) can be used to evaluate such integrals numerically. For the constant function $f = 1$ this computes the CY volume

$$\text{Vol}_{\text{CY}} = \int_X \text{dVol}_{\text{CY}} = \int_X \text{dVol}_{\text{FS}} = \frac{1}{3!} d_{\alpha\beta\gamma} t^\alpha t^\beta t^\gamma \quad (6)$$

which coincides for the CY and FS measure (provided, as we assume, J_{CY} and J_{FS} are in the same cohomology class) and which can be computed from the topological formula on the RHS of equation (6). Here

$$d_{\alpha\beta\gamma} = \int_X J_\alpha \wedge J_\beta \wedge J_\gamma \quad (7)$$

are the intersection numbers which are known or can be computed for CICY and KS manifolds (our package does this automatically in the background).

2.2. Line bundle slopes

Another interesting set of quantities which can be used for the same purpose (and we will, in fact, use it to enforce the correct Kähler class during training) is the slope of line bundles. The line bundle $L \rightarrow X$ with first Chern class $c_1(L) = [k^\alpha J_\alpha]$ is denoted by $\mathcal{O}_X(k)$ and its slope $\mu_t(\mathcal{O}_X(k))$ is defined by

$$\mu_t(\mathcal{O}_X(k)) = \int_X J_{\text{CY}}^2 \wedge c_1(\mathcal{O}_X(k)) = \int_X J_{\text{FS}}^2 \wedge c_1(\mathcal{O}_X(k)) = d_{\alpha\beta\gamma} t^\alpha t^\beta k^\gamma. \quad (8)$$

Note, since the intersection numbers are known, this can be computed from the Kähler parameters t^α and the line bundle integers k^α . On the other hand, introducing the curvature F_{FS} and corresponding sources ρ_{FS} and ρ_{CY} on $\mathcal{O}_X(k)$ by

$$F_{\text{FS}} = 2\pi i k^\alpha J_\alpha, \quad \rho_{\text{FS}} = \frac{1}{2} g_{\text{FS}}^{a\bar{b}} F_{\text{FS},a\bar{b}}, \quad \rho_{\text{CY}} = \frac{1}{2} g_{\text{CY}}^{a\bar{b}} F_{\text{FS},a\bar{b}} \quad (9)$$

the slope can also be computed from

$$\mu_t(\mathcal{O}_X(k)) = \frac{2}{\pi} \int_X \text{dVol}_{\text{CY}} \rho_{\text{CY}} = \frac{2}{\pi} \int_X \text{dVol}_{\text{FS}} \rho_{\text{FS}}. \quad (10)$$

The numerical version of this equation will be used to compute line bundle slopes during training and compute the slope loss, in order to enforce the correct Kähler class.

2.3. Constructing the holomorphic top form

For our purposes we need to construct the holomorphic top form Ω explicitly and we follow a method described, for example, in [48]. For a KS manifold, defined as a hypersurface in a toric four-fold $\mathcal{A}$ there are always four affine ambient space coordinates z^μ , where $\mu = 1, \dots, 4$, on a given patch, explicitly given in equation (A.5). Then we can write the holomorphic $(3, 0)$ form as

$$\Omega = \frac{dz_1 \wedge dz_2 \wedge dz_3}{\partial p / \partial z_4}, \quad (11)$$

where the coordinate z_4 in the denominator should be replaced by a solution $z_4 = z_4(z_a)$, where $a = 1, 2, 3$, of the defining equation $p(z_\mu) = 0$. More details on the underlying toric geometry can be found in [appendix](#).

For the case of CICYs, defined in an ambient space $\mathcal{A} = \mathbb{P}^{n_1} \times \dots \times \mathbb{P}^{n_h}$ with total dimension $d = \sum_{\alpha=1}^h n_\alpha$, we have affine coordinates z^μ , where $\mu = 1, \dots, d$ on each patch. From these we can choose three coordinates, z^a , where $a = 1, 2, 3$, to parameterize the CY patch while the remaining $K = d - 3$ coordinates z^q can be written as functions $z^q = z^q(z^a)$ by using the K defining polynomials p_r of the CICY. The holomorphic $(3, 0)$ -form can then be written in terms of the determinant of the $K \times K$ matrix $\partial p_r / \partial z^q$, and is explicitly given by

$$\Omega = \frac{dz_1 \wedge dz_2 \wedge dz_3}{\det(\partial p_r / \partial z^q)}, \quad (12)$$

where the coordinates z^q in the denominator have to be replaced by the solutions $z^q = z^q(z^a)$.

Given Ω , it is useful to introduce the associated volume form

$$\text{dVol}_\Omega = \Omega \wedge \bar{\Omega} \quad \Rightarrow \quad \text{dVol}_{\text{CY}} = \frac{\kappa}{3!} \text{dVol}_\Omega \quad (13)$$

which, by virtue of the Monge–Ampère equation (2) and (4), is proportional to the CY volume form. This means, while the Ricci-flat CY metric is not known in analytic form, its associated volume form can be constructed explicitly (up to a constant) from the expression for Ω .

2.4. Transition functions

The affine coordinates on each patch of the (toric) ambient space patch can be written as functions of the homogeneous coordinates x^i (see [appendix](#) ‘Affine patches’). More explicitly, for two ambient space patches $\mathcal{U}$ and $\mathcal{V}$ with affine coordinates u^μ and v^ν ,⁸ respectively, we can write $u^\mu = u^\mu(x)$ and $v^\nu = v^\nu(x)$, where the functional dependence on x^i is determined by equation (A.5). The transition function $v^\nu = v^\nu(u)$ can then be determined by matching the functional dependence on the homogeneous coordinates x^i . The transition matrix is then simply the Jacobian of this coordinate change,

$$(T_{\mathcal{UV}})^\nu_\mu = \frac{\partial v^\nu}{\partial u^\mu}. \quad (14)$$

2.5. Constructing Kähler forms

A further ingredient which we require explicitly is a basis J_α , where $\alpha = 1, 2, \dots, h^{1,1}(X)$, of the second cohomology. This basis is used to expand the Kähler form J_{FS} , as in equation (3), and the Kähler parameters t^α are defined relative to it. For simplicity, we focus on simplicial Kähler cones so that the J_α can be chosen as generators of the Kähler cone. Since these forms can be written in terms of Kähler potentials

$$J_\alpha = \frac{i}{2\pi} \partial \bar{\partial} \ln(\mathcal{K}_\alpha) \quad (15)$$

our task is to construct the $\mathcal{K}_\alpha$.

Recall that for our purpose of machine learning Ricci-flat metrics on CY manifold, we need to generate a random sample of points on the CY. We will discuss the method we employ for this task in section 2.7. A crucial point of this method is that it relies on properties of Fubini–Study Kähler metrics on projective spaces. Our objective in this section will therefore be to construct Kähler potentials $\mathcal{K}_\alpha$ that are ‘inherited’ from ambient space FS potentials on $\mathbb{P}^{r_\alpha}$, both for CICY and KS CY manifolds. For CICYs, this is a standard procedure. For KS CYs, the construction is less well known, so we will explain it in some detail.

As already mentioned above (and as explained in detail in [appendix](#)), toric ambient spaces have homogeneous coordinates x_i . The vanishing locus of each x_i defines a toric divisor, $D_i = \{x_i = 0\}$, and the Kähler cone generators J_α may be expanded in terms of these divisors: $J_\alpha = \sum_i c_i^\alpha D_i$. Denoting the space of sections of the (nef) line bundle dual to the Kähler cone divisors by $H^0(J_\alpha)$, we can find a monomial basis $\{s_j^{(\alpha)}\}$, $j = 0, 1, \dots, h^0(J_\alpha) - 1 := r_\alpha$, of this space following the methods outlined in [appendix](#) ‘Line bundles and sections’. In particular, we have

$$s_j^{(\alpha)} = \prod_{i=1}^n x_i^{\langle v_i, w_j \rangle + c_i^\alpha} \quad (16)$$

which expresses the monomial basis in terms of the toric coordinates. Here v_i are the vectors (1d cones) in the fan associated to the toric variety, and subsets of these span the top-dimensional cones of the lattice. The w_j are vectors in the dual top-dimensional cones of the dual lattice (see [appendix](#) for the precise definition). In general, the dimension of the space of sections will be larger than the number of homogeneous coordinates x_i , implying that (16) is a non-invertible relation.

By the theorem quoted in [appendix](#) ‘Line bundles and sections’ below (A.8), nef divisors are basepoint free, which means that the sections of the nef line bundles cannot all vanish at any given point in the CY. This means that the section monomials $s_j^{(\alpha)}$ can serve as homogeneous coordinates of a projective space $\mathbb{P}^{r_\alpha}$. The relations given in equation (16) thus correspond to maps Φ_α mapping the toric coordinates onto the coordinates of these projective spaces

$$\Phi_\alpha : [x_0 : x_1 : \dots] \rightarrow [s_0^{(\alpha)} : s_1^{(\alpha)} : \dots : s_{r_\alpha}^{(\alpha)}]. \quad (17)$$

Note that in the case where the ambient space is just a projective space $\mathbb{P}^n$ with homogeneous coordinates $[x_0 : x_1 : \dots : x_n]$, the sections of the Kähler cone generator are simply the monomials $s_i = x_i$, and the morphism Φ would be the identity map $\Phi : \mathbb{P}^n \rightarrow \mathbb{P}^n$. In the toric case, the map is non-trivial, and there are typically more sections than toric coordinates, leading to relations among the sections.

We can express the Kähler potentials $\mathcal{K}_\alpha$ via inner products of sections,

$$\mathcal{K}_\alpha = s^{(\alpha)\dagger} \cdot \mathbb{1} \cdot s^{(\alpha)}, \quad (18)$$

⁸ We caution the reader not to confuse these affine coordinates u^μ and v^ν with e.g. the toric vertices discussed in other sections of this paper.

where $s^{(\alpha)}$ is a vector containing the section monomials $s_i^{(\alpha)}$ and $\mathbb{1}$ is the identity matrix (or, as a straightforward generalization, any hermitian, non-degenerate matrix). Further, the inner product (indicated by $\cdot$) induces a notion of orthogonality (with respect to the Hermitian matrix), which will become important later when sampling points. Since not all s_α can vanish simultaneously, $\mathcal{K}$ is positive and is, hence, a sensible argument of the logarithm in equation (15). In the case of a projective ambient space $\mathbb{P}^n$, this Kähler potential is the standard FS Kähler potential

$$\mathcal{K}_{\text{FS}} = \sum_{i=0}^n |s_i|^2 = \sum_{i=0}^n |x_i|^2. \quad (19)$$

More generally, inserting the monomial basis (16) in (18), one obtains a Kähler potential for each Kähler cone generator of a toric variety

$$\mathcal{K}_\alpha = \sum_j |s_j^{(\alpha)}|^2 = \sum_{\substack{w_j \in M \\ \langle v_i, w_j \rangle + c_i^\alpha \geq 0 \forall i}} \left| \prod_i x_i^{\langle v_i, w_j \rangle + c_i^\alpha} \right|^2, \quad (20)$$

expressed in terms of homogeneous coordinates. Clearly, this is the standard Fubini–Study metric on the projectivized section space $\mathbb{P}H^0(J_\alpha) \cong \mathbb{P}^{r_\alpha}$, pulled back to the toric variety using the map (17).

The above results for $\mathcal{K}_\alpha$ are written in terms of the toric coordinates x_i . However, for subsequent applications we would like to re-write these in terms of affine coordinates on a patch. This can be accomplished by dividing $\mathcal{K}_\alpha$ in (20) by a monomial that is nowhere-vanishing in a chosen patch and that has the same toric scaling as $\mathcal{K}_\alpha$. It turns out, there is a unique choice $s_*^{(\alpha)}$ for this non-vanishing monomial, specified by a vector w_* in the dual lattice. Let us denote the affine patches by $\mathcal{U}_r$, with associated dual cones σ_r^* and affine coordinates z_μ . Then, as discussed in [appendix](#) ‘Affine patches’, the toric and affine coordinates are related by

$$z_\mu = \prod_{i=1}^n x_i^{\langle v_i, w_\mu \rangle}. \quad (21)$$

We introduce the index set $\mathcal{I}_r$ of indices whose associated coordinates generate the corresponding cone σ_r . Then, the affine version of $\mathcal{K}_\alpha$ in (20) in a patch $\mathcal{U}_r$ can be written as

$$\mathcal{K}_{\alpha, r} = 1 + \sum_{\substack{w_j \in \sigma_r^* \\ \langle v_i, w_j + w_* \rangle + c_i^\alpha \geq 0 \forall i \notin \mathcal{I}_r}} \left| \prod_{\mu \in \mathcal{I}_r} z_\mu^{\langle v_\mu, w_j \rangle} \right|^2, \quad (22)$$

where the dependence on the patch r has been made explicit. For the case where the ambient space is simply a projective space, this reduces to the standard (affine version of the) Fubini–Study Kähler potential, $\mathcal{K}_{\text{FS}} = 1 + \sum_\mu |z_\mu|^2$.

2.6. Numeric integration

Numeric integration can be performed using Monte-Carlo methods (for a method developed recently based on rejection sampling from tropical densities see [\[49\]](#)). The basic goal is to be able to numerically evaluate integrals of the form

$$\int_X \text{dVol}_{\text{CY}} f, \quad (23)$$

for a function f on a CY manifold X (of CICY or KS type). To do this, we require a sample of points $p_i \in X$, where $i = 1, \dots, N$, which are distributed according to a known measure, denoted by dA , which we discuss in detail below. In terms of the so-called weights (or masses), customarily defined as

$$w_i = \left. \frac{\text{dVol}_\Omega}{dA} \right|_{p_i}, \quad (24)$$

and the function values $f(p_i)$, the integral (23) can be approximated by

$$\int_X \text{dVol}_{\text{CY}} f = \int_X dA \frac{\text{dVol}_{\text{CY}}}{dA} f \simeq \frac{\kappa}{6N} \sum_{i=1}^N w_i f(p_i), \quad (25)$$

where (13) has been used. On the other hand, since the Ricci-flat CY metric is calculated numerically, there is no real need to rely on the measure $d\text{Vol}_\Omega$ and the integral (23) can alternatively be approximated by

$$\int_X d\text{Vol}_{\text{CY}} f \simeq \frac{1}{N} \sum_{i=1}^N \tilde{w}_i \det(g_{\text{CY}}(p_i)) f(p_i), \quad \tilde{w}_i = \left. \frac{d^6 y}{dA} \right|_{p_i}. \quad (26)$$

In the following we refer to the quantities $\tilde{w}_i$ as the auxiliary weights. The approximation (26) has a certain advantage in that it does not contain any unknown coefficient, such as κ in equation (25), which needs to be fixed by a reference calculation.

2.7. Sampling points on toric varieties

We now discuss a method for generating the sample points p_i with a known distribution dA on the CY X . In principle, this could be done by a Markov chain Monte-Carlo method using, for example, the Fubini–Study measure for dA , so $dA = d\text{Vol}_{\text{FS}}$. Here, we pursue a different approach based on a theorem by Shiffman and Zelditch [50], further generalized and explained in [16, 17]. This provides us with a viable method for ambient spaces which are products of projective spaces, and hence applies straightforwardly to CICYs. Importantly, using the techniques introduced in section 2.5, which allow us to write the toric Kähler potential in terms of a FS potential in the projectivized section space $\mathbb{P}H^0(J_\alpha) \cong \mathbb{P}^{r_\alpha}$ of the Kähler cone generators J_α , we can apply the same method to the toric case, as we will now explain.

To construct dA , we first compute the sections (16) and construct the associated FS Kähler potentials $\mathcal{K}_\alpha$ as in equation (18). Shiffman and Zelditch now state that the zeros of random sections are distributed w.r.t. to the (known) FS measure constructed from $\log(\mathcal{K}_{\text{FS}}) = \log \sum_{i=0}^n |s_i|^2$. Hence, we need to find zeros of random sections that are also lying on the CY and express these in terms of the homogeneous toric coordinates x_i .

First, we need to clarify what random section actually means in this context. As mentioned in section 2.5, the scalar product induces a notion of orthogonality (and hence independence) of sections. Since we chose the standard scalar product, the monomial sections are mutually orthogonal, and a random section S_α of the Kähler cone generator J_α is constructed from this orthogonal basis s_j^α with independent and identically distributed Gaussian coefficients $a_j^{(\alpha)} \sim \mathcal{N}(0, 1)$

$$S_\alpha = \sum_{j=0}^{r_\alpha} a_j^{(\alpha)} s_j^\alpha. \quad (27)$$

Since we want points on a CY n -fold, we need to construct n random sections $S_{\alpha,p}$, $p = 1, \dots, n$, and intersect their zero locus with the CY hypersurface, specified by its defining equation. We are relatively free in choosing the Kähler cone generators (that is, the indices α for each $p = 1, \dots, n$) for which we construct the random sections, as long as we are imposing not more than r_α equations for any given α .

We can use this to construct a point sample with known distribution and measure dA . Since we want to work with the toric coordinates x_i instead of the sections s_j^α , we need to express the sections in terms of the x_i . As mentioned around (17), the problem is that the maps Φ_α from x_i to s_j^α are not invertible. Indeed, there are non-linear relations (higher syzygies) among the s_j^α . Mathematically, this means that we have to construct a polynomial ring in x_i and s_j^α and add the ideals that describe the maps Φ_α , the hypersurface equations for the CY, and n random sections $S_{\alpha,p}$. Then, one finds the elimination ideal in the quotient ring modulo the relations among the sections to express the s_j^α in terms of the x_i . We carry this out in a two-step procedure. We solve the n equations $S_{\alpha,p} = 0$ subject to the relations among the sections (we will discuss how to find these relations in the next paragraph). We then express the remaining s_j^α in terms of the x_i as in (16) and impose the hypersurface equations (which are equations in the x_i). Solving this system of equations then gives a collection of homogeneous coordinates of points on the CY for each choice of n random sections. Moreover, these points are distributed with respect to the auxiliary measure dA . For a CY three-fold,

$$dA = \Phi_\alpha^*(J_\alpha) \wedge \Phi_\beta^*(J_\beta) \wedge \Phi_\gamma^*(J_\gamma), \quad (28)$$

where the indices α, β, γ correspond to the indices of the random sections $S_{\alpha,p}$, $p = 1, 2, 3$, that have been chosen above. Repeating this process for many random sections gives us the point sample we are after. While we appreciate that this is somewhat involved, it is absolutely crucial for our purposes. Any meaningful numerical integration relies on being able to generate points efficiently and distributed according to a known measure, dA . This is precisely what the above method allows us to do.

Finally, we need to explain how to derive the relations among the sections. One way to do this is to perform a primary decomposition of the defining ideal of the aforementioned quotient ring and use the elimination ideal to express the ideal generators, as obtained from the primary decomposition, in terms of

the sections. This method is general, and we implemented it in Sage (via Singular). However, it requires computing Groebner bases, which can be computationally costly.

Alternatively, we can do the following: We are interested in relations of the form⁹

$$\prod_I s_I^{f_I} = \prod_J s_J^{g_J} \Leftrightarrow \prod_I s_I^{h_I} = 1, \quad (29)$$

where we have introduced the multi-index $I = (i, \alpha)$, and likewise for J , and f_I, g_J are non-negative integers. At a generic point in section space, we can divide by the right-hand side and write the equation as done in the final step, where the h_I are integers (positive or negative) which encode the relations between the sections. Now recall from (A.8) or (16) that the sections are given in terms of the toric coordinates as

$$s_J = \prod_a x_a^{\langle v_a, w_J^{(\alpha)} \rangle + c_i^{(\alpha)}} =: \prod_a x_a^{M_{a,J}}, \quad (30)$$

where the last step defines the matrix of exponents $M_{a,J}$. Inserting this into (29), we get

$$\sum_I M_{a,I} h_I = 0. \quad (31)$$

This means the vectors h_I can be found as the (primitive integer) basis of the kernel of $M_{a,I}$. This approach based on lattice methods avoids potentially time-consuming Groebner basis calculations and is the default method used by the code.

Finally, we want to point out the following: At first, one might think that the system of equations comprising the CY hypersurface equation plus the equations that determine the sections in terms of the toric coordinates is simply a CICY in a product of projective ambient spaces $\otimes_{\alpha=1}^{h^{1,1}(X)} \mathbb{P}^{r_\alpha}$. However, if we naively count the dimensions of the resulting variety, we would find that it is in general smaller than n . Moreover, the degrees of the equations do not add up to the toric scalings plus one, so the hypersurface also seems to not be the anti-canonical one. However, the point is that the CY is *not* a complete intersection, precisely because of the relations of the sections (30). So another way of summarizing the procedure is that we have found an expression for the original CY n -fold X as a non-complete intersection CY in a product of projective ambient spaces $\otimes_{\alpha=1}^{h^{1,1}(X)} \mathbb{P}^{r_\alpha}$, which allows us to generate point samples with known distribution on X .

3. HYM equations

Line bundles $L \rightarrow X$ over CY manifolds X and their HYM connections provide an interesting application of numerical CY metrics. A vanishing slope is a necessary (and sufficient) condition for the existence of a HYM connection on a line bundle. Hence, meaningful numerical calculation of the HYM connection relies on being able to determine the Ricci-flat metric at prescribed points in Kähler moduli space—precisely one of the capabilities we are developing in the present work. In this section, we briefly review the underlying mathematics and set up the relevant formalism for our purposes. Numerical results for HYM connections will be presented in section 5.

3.1. Mathematical background

Line bundles $\mathcal{O}_X(k)$ over CY three-folds X and their slope have already been introduced in section 2.2. If H is a hermitian bundle metric on $\mathcal{O}_X(k)$ we have an associated (Chern) connection with gauge field and curvature given by

$$A = \partial \ln(H), \quad F = \bar{\partial} A = \bar{\partial} \partial \ln(H). \quad (32)$$

The above connection is said to be a HYM connection if its field strength satisfies the HYM equation

$$g_{\text{CY}}^{a\bar{b}} F_{a\bar{b}} = 0 \Leftrightarrow J_{\text{CY}} \wedge J_{\text{CY}} \wedge F = 0, \quad (33)$$

where g_{CY} is the (Ricci-flat) CY metric and J_{CY} the associated Kähler form. A necessary and sufficient condition for the existence for a HYM connection on $\mathcal{O}_X(k)$ is that its slope, as defined in equation (10), vanishes, that is, $\mu_t(\mathcal{O}_X(k)) = 0$. Checking this condition numerically is, therefore, the first step in a numerical calculation of the HYM connection.

⁹ In principle, we are free to use any coefficients, but a different choice would modify the parameterization of the complex structure as coefficients of the section monomials of the anti-canonical bundle that appear in the hypersurface that defines the CY.

3.2. Computing HYM connections

Provided the slope $\mu_t(\mathcal{O}_X(k))$ vanishes so that a HYM connection exists, how can this connection be computed explicitly? The first step is to start with a ‘Fubini–Study’ reference bundle metric on $\mathcal{O}_X(k)$ defined by

$$H_{\text{FS}} = \prod_{\alpha} \mathcal{K}_{\alpha}^{-k_{\alpha}}, \quad F_{\text{FS}} = \bar{\partial}\partial \ln(H_{\text{FS}}), \quad (34)$$

with the quantities $\mathcal{K}_{\alpha}$ constructed in section 2.5. This reference metric can be modified to

$$H = e^{\beta} H_{\text{FS}}, \quad F = \bar{\partial}\partial \ln(H) = F_{\text{FS}} + \bar{\partial}\partial \beta \quad (35)$$

where β is a function on X to be determined. Noting that the scalar Laplacian Δ_{CY} associated to the Ricci-flat metric g_{CY} on X can be written as $\Delta_{\text{CY}} = 2g_{\text{CY}}^{a\bar{b}} \partial_a \bar{\partial}_{\bar{b}}$, the HYM equation (33) for the field strength F from equation (35) turns into

$$\Delta_{\text{CY}} \beta = \rho_{\text{CY}} \quad (36)$$

where the source ρ_{CY} has been defined in equation (9). We note that the integrability condition for this equation amounts to ρ_{CY} integrating to zero over the CY manifold which, from equation (10), is equivalent to the slope vanishing. Provided this is the case, equation (36) has a solution β (unique up to an additive constant) which, inserted into equation (35), leads to the Hermitian bundle metric underlying the HYM connection.

In order to solve equation (36) numerically we introduce the usual scalar product

$$\langle f|h \rangle := \int_X \bar{f} \wedge \star_g h = \int_X d^6 y \sqrt{g} \bar{f} h \quad (37)$$

on the space $L^2(X)$ and consider a (not necessarily ortho-normal) basis of functions denoted by f_I or simply by $|I\rangle$. If we define the scalar products

$$G_{IJ} = \langle I|J \rangle, \quad \Delta_{IJ} = \langle I|\Delta_{\text{CY}}|J \rangle, \quad \tilde{\beta}_I = \langle I|\beta \rangle, \quad \rho_I = \langle I|\rho \rangle, \quad (38)$$

it follows that

$$|\beta\rangle = \sum_I \beta_I |I\rangle \quad \text{where} \quad \beta_I = \sum_J (G^{-1})_{IJ} \tilde{\beta}_J \quad (39)$$

and the Laplace equation (36) can be re-written as

$$\Delta \beta = \rho. \quad (40)$$

Here, Δ is the matrix with entries Δ_{IJ} and we have introduced the vectors $\beta = (\beta_I)$ and $\rho = (\rho_I)$. The kernel of Δ is non-trivial and consists of the vectors β which correspond to constant functions on X . This means Δ is not surjective and for a solution to exist it is necessary that $\rho \in \text{Im}(\Delta)$. This condition is of course nothing else but the integrability condition for the Laplace equation (36) which amounts to the vanishing of the slope. So as long as the slope vanishes, equation (40) has a solution.

In practice, we have to work with a suitably chosen finite set of functions f_I and things are not quite so straightforward. For a finite function set we should expect errors and we cannot expect that $\rho \in \text{Im}(\Delta)$. This means the finite-dimensional version of equation (40) typically does not have a solution and a naive attempt of solving the equation with standard methods of linear algebra fails. To remedy this problem we decompose ρ as $\rho = \rho^{\parallel} + \rho^{\perp}$ into the projection $\rho^{\parallel}$ onto $\text{Im}(\Delta)$ and the orthogonal complement $\rho^{\perp}$. Another way of stating the problem is that $\rho^{\perp}$ is typically non-zero in the finite-dimensional approximation. However, provided the slope vanishes and the finite-dimensional approximation is sufficiently accurate, we expect that $|\rho^{\perp}|/|\rho^{\parallel}| \ll 1$. Assuming this is indeed satisfied we can safely replace equation (40) by

$$\Delta \beta = \rho^{\parallel}, \quad (41)$$

which does have a solution.

In summary, our procedure involves computing the quantities Δ_{IJ} and ρ_I in equation (38) by numerical integration over X for a finite set of functions f_I and then solving the linear algebra problem (41). Given a solution $\beta = (\beta_I)$, the function β is then approximated by $\beta = \sum_I \beta_I f_I$.

A suitable set of functions f_I can be constructed by starting with a positive line bundle $\mathcal{O}_X(l)$. If we denote by s_p , where $p = 1, \dots, h^0(\mathcal{O}_X(l))$, a basis of its sections we can define the functions

$$f_{pq} = \frac{s_p \bar{s}_q}{\prod_{\alpha} \mathcal{S}_{\alpha}^{l_{\alpha}}}, \quad (42)$$

where $\mathcal{S}_{\alpha}$ is the homogeneous version of $\mathcal{K}_{\alpha}$. Of course, the higher the degree l , the larger the space spanned by the f_{pq} and the more accurate we expect the approximation to be. As a measure of the accuracy with which the HYM equation (36) is solved we define

$$\sigma_H = \frac{\int_X d^6 y \sqrt{g_{CY}} |\Delta_{CY} \beta - \rho_{CY}|}{\int_X d^6 y \sqrt{g_{CY}} |\rho_{CY}|}, \quad (43)$$

which we will refer to as the ‘HYM measure’ in the future.

4. Computational background

The open source library *cymetric* consists of point generators for CICY and KS CY manifolds and five metric models that learn the Ricci-flat CY metrics using the generated points as input. In this section, we describe these networks and their training in more detail. After a brief general introduction to NNs, we describe how the networks of *cymetric* are constructed and how their learning is controlled by custom loss functions that encode the mathematical constraints CY metrics must satisfy.

4.1. Introduction to NNs

This section provides a quick review of the building blocks of NNs. The literature on machine learning and NNs is vast, and includes many good introductory texts. For the reader with a background in (theoretical) high-energy physics, [51] provides a very good introduction to the topic, and [29] gives a topical discussion on the applications of modern data science techniques to string theory.

A NN is a collection of nodes, which are organized in several layers. Two of these layers, the input and output layers, are visible to the user, the others are hidden. NNs with many nodes in each layer are referred to as wide, and NNs with many (more than one) hidden layers are called deep. Given an input vector $v_0 \in \mathbb{R}^{n_0}$, the NN processes the data by transmitting it from one layer to the next, in a sequence of mappings $v_0 \mapsto v_1 \mapsto v_2 \mapsto \dots \mapsto v_N$, where $v_k \in \mathbb{R}^{n_k}$. In more detail, for a fully-connected NN, the output v_k of layer k is related to its input v_{k-1} by

$$v_k = \sigma_k(W_k v_{k-1} + b_k), \quad (44)$$

where the $n_k \times n_{k-1}$ matrix W_k contains the *weights* and the vector $b_k \in \mathbb{R}^{n_k}$ the *biases* of the k th layer. The functions σ_k act on each component of their vectorial arguments and are called the *activation functions*. Altogether, the NN represents a family $f_{\theta} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_N}$ of functions which is given by iterating equation (44). Here, θ collectively denotes the weights and biases in all layers.

Training of the NN is based on a set of input vectors $\{v_0\}$. The first step of any machine learning experiment is to randomly split this data into a training and a validation set (where the training-to-validation ratio depends on the size of the dataset). The training set is used to train the NN by adjusting the weights and biases θ , a process that is guided either by certain loss functions or a reward system provided by the user (see below for the loss functions relevant to our setting). The training data may also be further subdivided into (mini-) batches to improve training. The division of data into batches reduces the risk of the NN getting trapped in local minima of the loss function, thus increasing the chance of finding solutions that generalize well. This is also sometimes necessary, especially for networks that perform memory-intensive operations (such as differentiation with respect to the input data in our case) in order to compute loss functions. Networks operate in epochs, where each epoch constitutes one run over all batches.

In machine learning, one typically distinguishes different setups such as supervised, unsupervised, or reinforcement learning. We are dealing with a version of supervised machine learning. However, in classical supervised ML, the input *and* output is known for the training set. Lacking an analytical expression for the CY metric, while we have inputs (the points on the CY at which to compute the metric), we only have partial output information (the CY volume form and the line bundle slopes) together with a set of constraints that the NN output (that is, the metric) has to satisfy. Hence, the task is more akin to physics informed self-supervised learning or optimization than classical ‘supervision’ with given input-output pairs. We check

Table 1. Different Ansätze for the NN prediction of the Ricci-flat metric.

Name	Ansatz
Free	$g_{\text{pr}} = g_{\text{NN}}$
Additive	$g_{\text{pr}} = g_{\text{FS}} + g_{\text{NN}}$
Multiplicative, element-wise	$g_{\text{pr}} = g_{\text{FS}} + g_{\text{FS}} \odot g_{\text{NN}}$
Multiplicative, matrix	$g_{\text{pr}} = g_{\text{FS}} + g_{\text{FS}} \cdot g_{\text{NN}}$
ϕ -model	$g_{\text{pr}} = g_{\text{FS}} + \partial\bar{\partial}\phi$

how well the output meets these constraints by constructing error measures that encode them. Importantly, given the split of the initial data, the quality of the output can be assessed on both the training and validation sets. For the validation set, this check is performed by applying the trained network to the validation data.

From this brief summary, it is clear that designing a NN to perform a certain task involves choosing a number of *hyperparameters*, that specify the network's properties. For fully connected NNs, this amounts to specifying the width and depth of the network, which activation functions to use, and the number of batches and epochs. More advanced network components, such as convolutional or dropout layers (see, for example, [29] for a recent discussion), require additional hyperparameters. For our task, previous studies [32–34] have shown that a simple fully connected NN exhibits good performance. Consequently, we focus on this setting, leaving studies involving more advanced NN architectures for the future. We build the networks using TensorFlow [41], an ML library with functionalities that suit our needs. In particular, TensorFlow allows to construct the NNs in a sequential manner, and has differentiation functions that allow us to compute derivatives with respect to the input data.

4.2. Network architectures

The basic idea underlying machine learning of CY metrics is to use a NN whose associated functions f_{θ} represent metrics on the manifold. In other words, the NN input consists of a point $p \in X$ on the CY manifold and its output represents a metric $g(p)$ at this point. There are a number of concrete realizations of this idea. Since CY three-folds X are complex manifolds, their metrics $g(p)$ at each point $p \in X$ can be written, relative to a local choice of complex coordinates, as a Hermitian 3×3 matrix. The first, and most obvious, approach is then to let the NN predict the nine independent entries (three real entries on the diagonal, and three complex entries on the off-diagonal) of this matrix. While this is possible, it does not take advantage of the mathematical knowledge we have about CY manifolds. For example, equation (2) shows that the CY metric is given by an exact correction to some reference Kähler metric g_{FS} . Moreover, by constructing CY manifolds as a hypersurfaces or complete intersection in an ambient space, one can construct the metric g_{FS} explicitly by pullback from the ambient space $\mathcal{A}$.

The *cymetric* package realizes five choices for how the metric g_{pr} predicted by the NN is related to the function g_{NN} that the NN actually represents. These possibilities are summarized in table 1. The first and most obvious choice, $g_{\text{pr}} = g_{\text{NN}}$, has been included for reference but is by no means the optimal one. A metric is required to be non-singular and this condition can easily be violated for a randomly initialized or stochastically trained NN. A NN which ‘accidentally’ represents a singular or near-singular metric can lead to numerical problems. Also, for this choice, the NN has to cope with the entire numerical variation of the metric g_{pr} . Both problems can be solved, or at least alleviated, by writing $g_{\text{pr}} = g_{\text{FS}} + \text{correction}$, and the four other possibilities in table 1 are of this form. Indeed, using the non-singular metric g_{FS} as a background makes accidentally generating singular metrics less likely (and, if there is an actual singularity in the space, the reference FS metric will already have this feature). Furthermore, under the plausible assumption that the entries of $g_{\text{CY}} - g_{\text{FS}}$ are typically smaller than the ones in g_{CY} , a NN dealing with a metric correction involves smaller numbers, likely leading to enhanced numerical stability. As table 1 shows, g_{pr} may be defined additively, $g_{\text{pr}} = g_{\text{FS}} + g_{\text{NN}}$, via element-wise multiplication, $g_{\text{pr}} = g_{\text{FS}} + g_{\text{FS}} \odot g_{\text{NN}}$, or via matrix multiplication, $g_{\text{pr}} = g_{\text{FS}} + g_{\text{FS}} \cdot g_{\text{NN}}$. Finally, using the full information contained in equation (2), we may set $g_{\text{pr}} = g_{\text{FS}} + \partial\bar{\partial}\phi$, where the NN represents the real function ϕ .

While we have already argued that the free NN is likely inefficient, there is no telling which of the other four Ansätze in table 1 will be most efficient in learning the Ricci-flat metric on a given CY manifold. For example, while the ϕ -model automatically gives a Kähler metric, it requires two additional derivatives on the input, which come at a computational cost. For this reason, the *cymetric* package makes all these options available for the user to explore. In the ensuing section, we will compare the performance of the different NNs on the quintic.

In each experiment the NN is trained on input data which consists of a set $\{p_i\}$ of points on the CY manifold. These points are generated using the method described in section 2.7. They are, by the theorem of Shiffman and Zelditch, distributed with respect to a measure dA , which is inherited from the ambient space.

This allows to compute any integrals used in the training and validation processes as weighted sums, following equations (25) and (26).

As discussed above, the learning of NNs is governed by the minimization of loss functions. A CY metric must satisfy a number of mathematical constraints, listed in section 2. We encode these restrictions in a custom loss function

$$\mathcal{L} = \alpha_1 \mathcal{L}_{\text{MA}} + \alpha_2 \mathcal{L}_{\text{dJ}} + \alpha_3 \mathcal{L}_{\text{transition}} + \alpha_4 \mathcal{L}_{\text{Ricci}} + \alpha_5 \mathcal{L}_{\text{Kclass}} \quad (45)$$

where α_i are hyperparameters of the NN with default value $\alpha_i = 1.0$, and the individual loss contributions are defined as follows:

- *Monge–Ampère loss* $\mathcal{L}_{\text{MA}}$

The Monge–Ampère loss is defined as

$$\mathcal{L}_{\text{MA}} = \left\| 1 - \frac{1}{\kappa} \frac{\det g_{\text{pr}}}{\Omega \wedge \bar{\Omega}} \right\|_n, \quad (46)$$

and it ensures that the metric satisfies the Monge–Ampère equation (2), as required by the CY theorem. This loss may be computed with any L_n norm, as indicated by the subscript n (default is $n = 1$)¹⁰.

- *Ricci loss* $\mathcal{L}_{\text{Ricci}}$

The Ricci loss is defined by

$$\mathcal{L}_{\text{Ricci}} = \|R\|_n = \|\partial \bar{\partial} \ln \det g_{\text{pr}}\|_n, \quad (47)$$

and it ensures that the metric has vanishing Ricci scalar¹¹. This is a necessary condition for the metric to satisfy the stronger CY requirement of having vanishing Ricci tensor. Again, this loss function may be computed with different L_n norms, with default $n = 1$.

- *Kähler loss* $\mathcal{L}_{\text{dJ}}$

The Kähler loss, defined by

$$\mathcal{L}_{\text{dJ}} = \sum_{ijk} \|\text{Re } c_{ijk}\|_n + \|\text{Im } c_{ijk}\|_n, \quad (48)$$

where $c_{ijk} = g_{\bar{j},k} - g_{k,\bar{j}}$ and $g_{\bar{j},k} = \partial_k g_{\bar{j}\bar{j}}$ ensures that $dJ = 0$, so that the metric is Kähler. The default for the L_n norm is $n = 2$.

- *Transition loss* $\mathcal{L}_{\text{transition}}$

The *transition loss function* ensures that $g_{\bar{j}\bar{j}}$ transforms as a complex tensor under coordinate transformations. For the additive and multiplicative networks, this is implemented as

$$\mathcal{L}_{\text{transition}} = \frac{1}{d} \sum_{\mathcal{U}, \mathcal{V}} \left\| g_{\text{pr}}^{\mathcal{V}} - T_{\mathcal{U}\mathcal{V}} \cdot g_{\text{pr}}^{\mathcal{U}} \cdot (T_{\mathcal{U}\mathcal{V}})^{\dagger} \right\|_n, \quad (49)$$

where $(T_{\mathcal{U}\mathcal{V}})_{\mu}^{\nu} = \partial v^{\nu} / \partial u^{\mu}$ are the transition matrices between patches $\mathcal{U}$ and $\mathcal{V}$, with local coordinates u and v , respectively. Moreover, d is the number of patch transitions, and the default for the norm is $n = 1$.

- *Kähler class loss* $\mathcal{L}_{\text{Kclass}}$

Finally, since we would like to compute the Ricci-flat metric for given Kähler parameters t_{α} , we require a loss contribution which enforces the correct Kähler class¹². It may appear this is not necessary for the ϕ -model—provided g_{FS} is set up with the correct Kähler class, the addition $\partial \bar{\partial} \phi$ is exact and, hence, does not change the class. While this is true mathematically, it depends on ϕ being a function (rather than a section), a condition which is difficult to enforce on a NN. For the other models in table 1 there is no mathematical reason for the Kähler class to remain unchanged. In order to fix the Kähler class we define the *Kähler class loss function* as

$$\mathcal{L}_{\text{Kclass}} = \frac{1}{h^{1,1}(X)} \sum_{\alpha=1}^{h^{1,1}(X)} \left\| \mu_t(\mathcal{O}_X(e_{\alpha})) - \int_X J_{\text{pr}}^2 \wedge F_{\text{FS},\alpha} \right\|_n. \quad (50)$$

¹⁰ The choice of L_n norm will affect the training of the network. A high n pushes the network to reduce outliers, that is, large loss function contributions that are localized at a few points. With a low n , the NN will instead strive to reduce the loss function of all points with equal weight. From our experiments, the default values chosen in *cymetric* lead to good performance on the manifolds we have studied.

¹¹ Recall that for Kähler metrics, the Ricci scalar simplifies to $R = g^{\bar{j}\bar{j}} \partial_i \bar{\partial}_{\bar{j}} \log \det(g)$.

¹² This is not essential for manifolds with $h^{1,1}(X) = 1$ (such as the quintic), since all Kähler classes are obtained by simple metric rescaling. However, it is critical in order to generate metrics with a well-defined Kähler class for $h^{1,1}(X) > 1$.

Here, $\mathcal{O}_X(e_\alpha)$ are the line bundles specified by integers $(e_\alpha)^\beta = \delta_\alpha^\beta$, $F_{\text{FS},\alpha} = -iJ_\alpha/(2\pi)$ are their associated field strengths and $\mu_t(\mathcal{O}_X(e_\alpha))$ their slopes for the given Kähler parameters t_α , as computed from the topological formula, equation (8). For the norm, the default is $n = 1$. Conversely, these slope values on the ‘line bundle basis’ $\mathcal{O}_X(e_\alpha)$, $\alpha = 1, \dots, h^{1,1}(X)$, fix the Kähler parameters uniquely and this is the basic idea behind equation (50): by enforcing the correct slopes we enforce the correct Kähler class.

The slope integral in equation (50) is evaluated numerically using MC integration, that is, integrals of the form (10) are computed using equation (26). Note that this integral requires a large batch size to provide a good approximation to the slope, while the other losses benefit from gradient descent with small mini-batches. Hence, we break each training epoch of the NN into two optimization steps with two different batch sizes. A small batch size (default is 64 points) is used for all above contributions to the loss function while a large one (defaults to 10 000 points or the entire data set, whichever is smaller) is used for the Kähler class loss. The learning rate is kept constant between the two batches, and during training. We use the same optimizer in order to share some of the information, like momentum, between the two batches, and gradients are accumulated over the two optimization steps. We have observed that also keeping the Monge–Ampère loss in the updates with the large batch size further improves stability of the code.

For all these loss functions, the derivatives with respect to the input coordinates are computed with TensorFlow’s automatic differentiation, which works reliably for all NNs.

A few comments are in order. By virtue of the CY theorem, the Monge–Ampère loss $\mathcal{L}_{\text{MA}}$ is sufficient to enforce Ricci-flatness of the metric. The Ricci-loss $\mathcal{L}_{\text{Ricci}}$ has been included as a potential additional check but, since its computation is costly (as it involves taking two further derivatives), it is disabled by default. Moreover, for the ϕ -model which is Kähler by construction, the Kähler loss $\mathcal{L}_{\text{dJ}}$ is disabled. Similarly, the transition loss is disabled for this network (this is guaranteed as long as ϕ is a function). Disabling these losses for the ϕ -model shortens training time and has had no ill effects on performance.

After the NN is trained, we require an error measure that tells us how well the NN approximates the quantity we are learning, and how well it compares with other approximation schemes. For our purposes, we consider error measures derived from the Monge–Ampère equation and the requirement of Ricci-flatness. There are two established benchmark measures in the literature [15, 20], the sigma and Ricci measures, $(\sigma, \mathcal{R})$. The σ measure is obtained by integrating, using the MC approximation as a weighted sum, the Monge–Ampère equation

$$\sigma = \frac{1}{\text{Vol}_\Omega} \int_X \left| 1 - \kappa \frac{\Omega \wedge \bar{\Omega}}{(J_{\text{pr}})^n} \right|, \quad (51)$$

where $\text{Vol}_\Omega = \int d\text{Vol}_\Omega$. The Ricci measure results from integrating the Ricci scalar

$$\mathcal{R} = \frac{\text{Vol}_{\text{CY}}^{1/3}}{\text{Vol}_\Omega} \int_X |R_{\text{pr}}|. \quad (52)$$

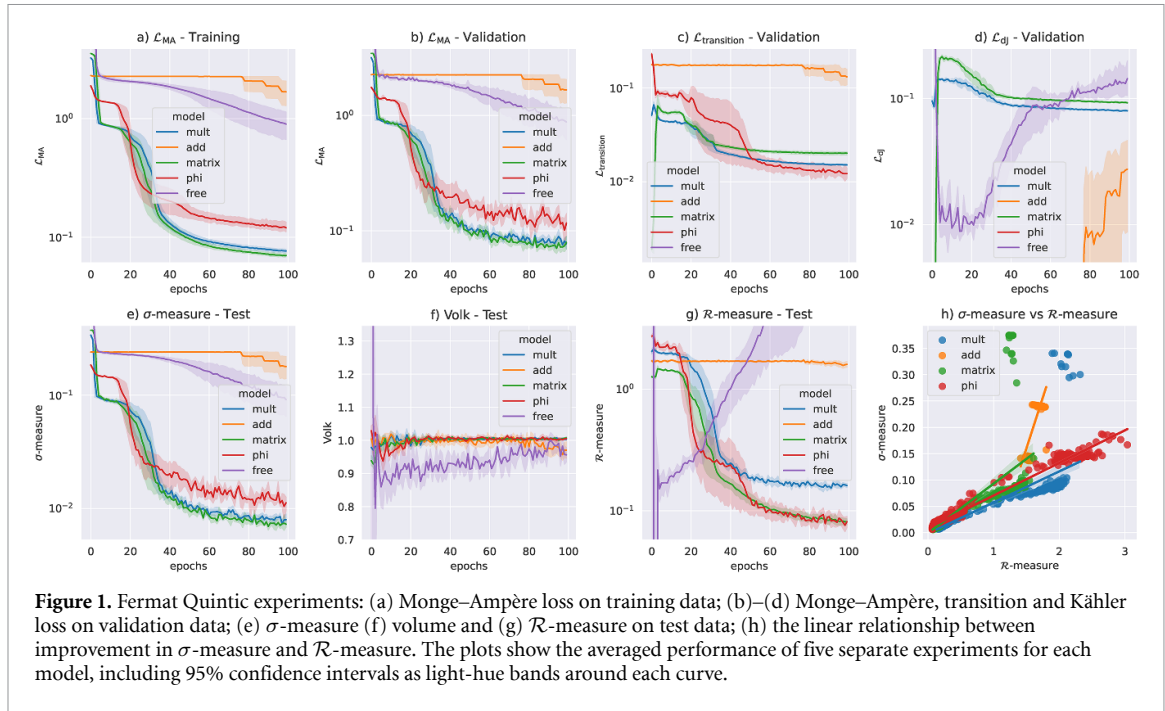
It should be noted that a small Ricci measure is necessary, but not sufficient, to guarantee that the entire Ricci tensor is small. One could explore other (integrated) curvature measures, formed from other contractions of the Ricci tensor, which, however, we have not done in this work. We also provide measures to monitor the quality of the Kähler, transition, and volume approximations.

5. Results

In this section, we report the results from experiments where CY metrics are learned on CICY and KS manifolds. These runs illustrate that the ML routines that we have developed perform well on CY manifolds from these lists. As we will see, there is some variance in the performance of the different NNs. The ϕ network emerges as the most efficient learner, and the small values reported on the σ and $\mathcal{R}$ error measures corroborate the accuracy of the obtained metrics. We furthermore find that the point-generating routines are efficient with run-times ranging from seconds (for the quintic) to an hour (for the KS CY example) on a CPU of a standard laptop. We expect this efficiency will persist on CYs of similar complexity. For ease of notation, we will henceforth refer to the approximate CY metric obtained at the end of training as g_{CY} .

5.1. Quintic

The Quintic three-fold is specified as the vanishing locus of a quintic polynomial in the projective ambient space $\mathcal{A} = \mathbb{P}^4$, and appears in both the CICY and KS lists. It has one Kähler modulus, which measures the overall volume of the manifold, and 101 complex structure moduli. A particular, simple instance of this family of manifolds is the Fermat quintic, given by the vanishing locus of $Q = \sum_{i=0}^4 z_i^5$. The Fermat quintic is



arguably the simplest CY three-fold, and is a benchmark model for numerical computations of CY metrics, whose metric has been approximated using a variety of approaches [14–19, 22, 32, 33]. Given its simplicity, the Fermat quintic provides an excellent setting for comparing the performance of the different metric Ansätze in table 1, and this is the main purpose of this subsection.

5.1.1. Comparison of Ansätze for CY metric predictions

For these experiments, we have generated a random point sample of 198 000 points on the quintic using the CICY point generator of *cymetric*. For each metric Ansatz in table 1, we have constructed a fully connected feed-forward network with three hidden layers, each containing 64 nodes, and GELU activation functions¹³. These have been trained using an Adam optimizer on the point sample, with a 0.1 validation split of the data. The NNs were initialized with weights sampled from a Gaussian distribution. For the networks that learn a correction to the reference metric g_{FS} , we select a Gaussian distribution $\mathcal{N}(0, 0.01)$ with small variance, whereas the free NN is initialized with $\mathcal{N}(0, 1)$.

For each metric Ansatz, five experiments over 100 epochs were performed. The results of these experiments are shown in figure 1 (and were first published in the 2021 NeurIPS conference, [34]). Note that for these quintic runs, the volume of the manifold has been normalized to 1. As shown in subplot (f), all networks preserve this value of the volume.

It is evident from figure 1 that learning takes place for all network Ansätze. It is also clear that the performance of the NNs differ: the multiplication and ϕ models outperform the additive and free networks. In fact, the plots show that the training does not complete for the latter models: even after 100 epochs, most losses and error measures have not reached a stable value. In contrast, the two multiplication models, and the ϕ network, reduce the relevant losses and errors by approximately two orders of magnitude, and obtain stable values after the 100 epoch experiments are completed.

In more detail, we see that the Monge–Ampère loss, shown in subplots (a) and (b) for training and validation data, respectively, decreases by two orders of magnitude during training of the best performing networks. The similarity in performance between training and validation set implies that overfitting is not a problem. The transition loss on validation data is presented in subplot (c). Again, learning is evident for all networks but for the additive NN the loss function only starts decreasing after 80 epochs. The final plot on the first row of figure 1 shows the Kähler loss on validation data. The ϕ model does not appear in this plot, since it provides a Kähler metric by default (as long as ϕ is a function, which is guaranteed by the preservation of the Kähler class, which is automatic up to a scale factor for the Quintic). Interestingly, the multiplicative models perform worse than the additive model on this loss. This is to be expected: multiplying the metric with a non-constant matrix will generically result in an associated 2-form which fails to close. On

¹³ Before these experiments, we have performed hyperparameter tuning via box searches for some of the hyperparameters, including for the number of hidden layers and the type of activation function.

the other hand, for the additive network, while the addition to the metric will generically break the Kähler constraint, the smallness of the addition implies that this breaking will be small. One should expect the result to reflect this, and it is reassuring that it does.

On the second row of figure 1 it is checked that the volume of the manifold is not changed during learning. We also compute the σ and $\mathcal{R}$ error measures. These checks are performed on a separate test set of 22 000 points. The σ and $\mathcal{R}$ measures are benchmarks for the quality of CY metric approximations, and their low values testify to the performance of the NNs. The ϕ -models with lowest validation loss reach, within an hour on a single CPU of a standard laptop, a mean accuracy of $\sigma = 0.0086$ and $\mathcal{R} = 0.076$ and thus matches $k = 20$ in Donaldson algorithm [20] with a training time of 35 years on 4.6×10^8 points [30]. On the other hand, the energy functional methods obtain a similar accuracy already at $k = 4$ [22] and other machine learning methods report similar runtimes as ours¹⁴.

To the best of our knowledge, we are the first to systematically study $\mathcal{R}$, and to demonstrate, in subplot (h), a linear relation, $\sigma \approx 0.06\mathcal{R}$ for the ϕ -model, between optimization of the surrogate Monge–Ampère equation (2) and decrease in Ricci measure $\mathcal{R}$. This plot shows the linear fits as obtained using Seaborn’s linear regression model function `regplot`, together with 95% confidence intervals as light-hue bands¹⁵. While this is to be expected from the CY theorem, a numerical routine could in principle deviate from this result. It is reassuring that our implementation does not and that there is no tension between these two constraints.

Another conclusion from these experiments is that the ϕ -model performs better than the other models, at least for simple CY three-folds¹⁶. While this is a result that should be treated with some caution, we have reached similar conclusions on other manifolds, and will thus focus on this model in the following sections.

5.2. Bi-cubic

In this sub-section we describe our numerical results for the bi-cubic CY, a manifold with Picard number two. Specifically, we calculate the numerical Ricci-flat CY metrics at various specific points in the bi-cubic Kähler moduli space. For these metrics we perform two basic checks to confirm that the correct point in Kähler moduli space has indeed been reached: a numerical calculation of the CY volume and a numerical calculation of various line bundle slopes, each of which can be compared with the exact result obtained from a topological formula. As an example application, we also compute the HYM connection for a specific line bundle on the bi-cubic numerically.

5.2.1. Basic set-up

The bi-cubic CY is defined as a hypersurface in the ambient space $\mathcal{A} = \mathbb{P}^2 \times \mathbb{P}^2$ and it is contained in both the CICY and the KS lists. Specifically, it is given as the zero locus in $\mathcal{A}$ of a bi-degree (3, 3) polynomial. Its Hodge numbers are $(h^{1,1}(X), h^{2,1}(X)) = (2, 83)$ and its non-zero intersection numbers are $d_{112} = d_{122} = 3$. Throughout our calculations, we fix a specific point in complex structure moduli space by focusing on the defining polynomial

$$\begin{aligned} p = & 0.96x_0^3y_0^3 + 0.79x_0^3y_1^3 + 0.49x_0^3y_2^3 + 0.51x_0^3y_0y_1y_2 + 0.05x_1x_0^2y_0y_1^2 + 0.99x_2x_0^2y_0y_2^2 + 0.81x_1x_0^2y_1y_2^2 \\ & + 0.39x_2x_0^2y_0y_1^2 + 0.54x_1x_0^2y_0y_2^2 + 0.87x_2x_0^2y_1y_2^2 + 0.62x_1x_2x_0y_0^3 + 0.62x_1x_2x_0y_1^3 + 0.62x_1x_2x_0y_2^3 \\ & + 0.81x_2^2x_0y_0y_1^2 + 0.87x_1^2x_0y_0y_2^2 + 0.54x_2^2x_0y_1y_2^2 + 0.99x_1^2x_0y_0y_1^2 + 0.05x_2^2x_0y_0y_2^2 + 0.39x_1^2x_0y_1y_2^2 \\ & + 0.87x_1x_2x_0y_0y_1y_2 + 0.49x_1^3y_0^3 + 0.79x_2^3y_0^3 + 0.96x_1^3y_1^3 + 0.49x_2^3y_1^3 + 0.79x_1^3y_2^3 + 0.96x_2^3y_2^3 \\ & + 0.54x_1^2x_2y_0y_1^2 + 0.39x_1x_2^2y_0y_2^2 + 0.05x_1^2x_2y_1y_2^2 + 0.87x_1x_2^2y_0y_1^2 + 0.81x_1^2x_2y_0y_2^2 + 0.99x_1x_2^2y_1y_2^2 \\ & + 0.51x_1^3y_0y_1y_2 + 0.51x_2^3y_0y_1y_2, \end{aligned} \quad (53)$$

where (x_0, x_1, x_2) and (y_0, y_1, y_2) are homogeneous coordinates on the two $\mathbb{P}^2$ factors. We will calculate the Ricci-flat metric for seven points in Kähler moduli space, that are listed in table 2. These point were chosen so that seven simple line bundles (i.e. with the lowest integer charges) have zero slope relative to those points in Kähler moduli space. Thus, for each of these choices of t there is a corresponding slope-zero line bundle $\mathcal{O}_X(k)$ which is given in the last row of table 2. Further, while this relatively small sample of moduli space points does not cover the full Kähler cone, the points selected yield both symmetric and asymmetric members of the family of bicubic CYs. Moreover, as we will see below, this set-up provides 7 plus 49

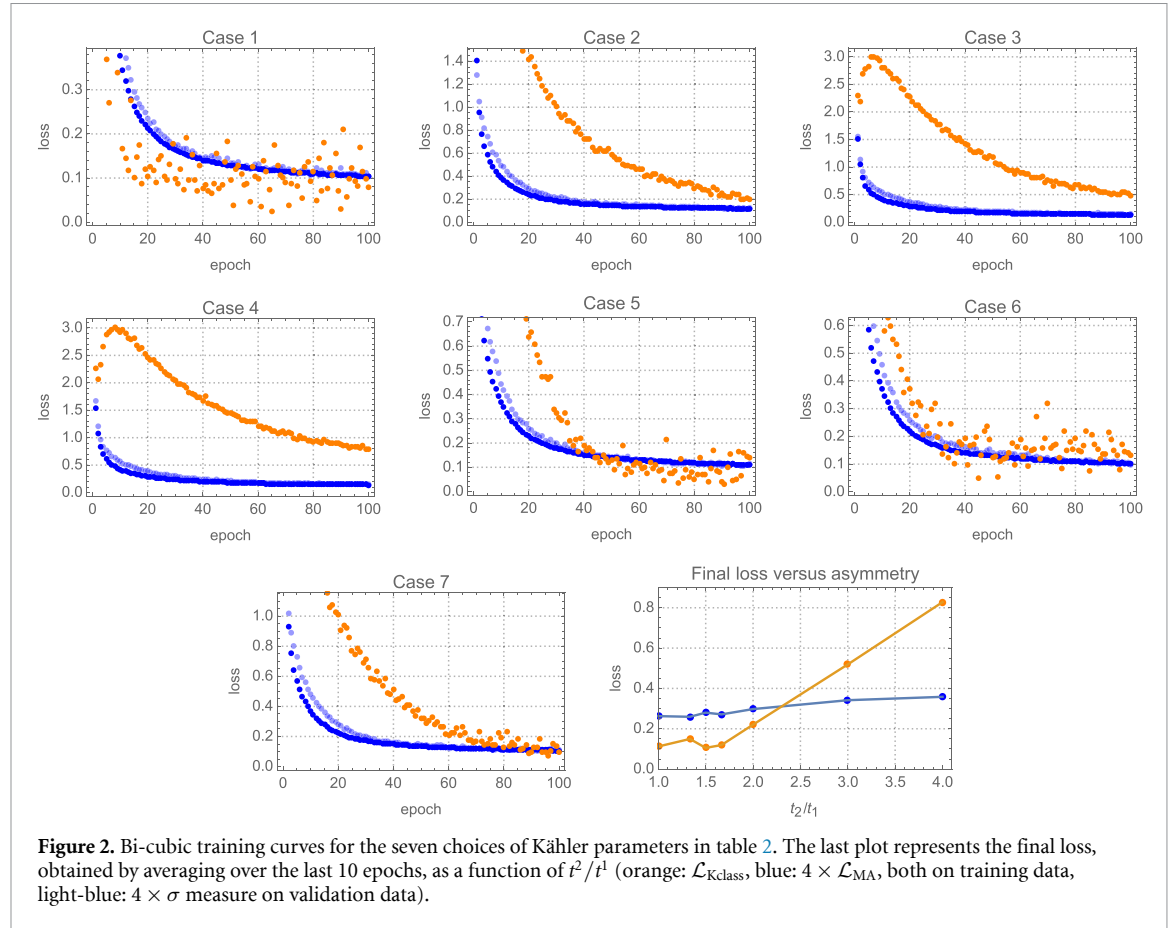
¹⁴ We recall that reducing runtime for ML experiments is not the primary goal of the *cymetric* package; rather the benefit of the new package is that it can be applied to a wider range of CY manifolds than previously possible and that it computes the Ricci-flat metric for a given Kähler class.

¹⁵ The R^2 coefficients (rounded to 2 digits) are, for the respective models, additive: 0.64 (0.64); matrix mult: 0.99 (0.51); element mult: 0.99 (0.66); ϕ : 0.98 (0.98); the values without (with) brackets exclude (include) outliers that come from initial fluctuations.

¹⁶ This conclusion is also corroborated by experiments reported on in [52].

Table 2. Choices $t_{(i)}$, where $i = 1, \dots, 7$, of the Kähler parameters for the bi-cubic and corresponding slope zero line bundles with line bundle integers $k_{(i)}$.

Case	1	2	3	4	5	6	7
$t_{(i)}$	$\begin{pmatrix} 1.414 \\ 1.414 \end{pmatrix}$	$\begin{pmatrix} 0.687 \\ 1.878 \end{pmatrix}$	$\begin{pmatrix} 0.421 \\ 1.955 \end{pmatrix}$	$\begin{pmatrix} 0.299 \\ 1.977 \end{pmatrix}$	$\begin{pmatrix} 0.962 \\ 1.753 \end{pmatrix}$	$\begin{pmatrix} 1.092 \\ 1.676 \end{pmatrix}$	$\begin{pmatrix} 0.853 \\ 1.809 \end{pmatrix}$
$\mathcal{O}_X(k_{(i)})$	$\mathcal{O}_X(1, -1)$	$\mathcal{O}_X(1, -2)$	$\mathcal{O}_X(1, -3)$	$\mathcal{O}_X(1, -4)$	$\mathcal{O}_X(2, -3)$	$\mathcal{O}_X(3, -4)$	$\mathcal{O}_X(3, -5)$



independent checks of our numerical routines through the computation of CY volumes and line bundle slopes, respectively.

5.2.2. Point sampling and training with ϕ -model

For each of the seven choices of Kähler parameters in table 2, we have generated 100 000 points (with 0.1 validation split) on the bi-cubic, using the Mathematica point generator of the *cymetric* package. For the ϕ -model and a NN with width 64, depth 3, GELU activation functions and initialization with $\mathcal{N}(0, 0.01)$, training has been carried out for 100 epochs, using the Adam optimizer with a batch size of 64 and a learning rate of $1/1000$. Training has been completed on a CPU of a standard laptop in about three hours, for each of the seven choices of Kähler parameters.

The only two relevant contributions to the loss are the Monge–Ampère loss and the Kähler class loss which are shown, together with the σ error measure in figure 2 (where, in order to clearly display all functions in the same plot, the Monge–Ampère loss and σ error measure are rescaled by a factor 4). Evidently, training is efficient and successful for all seven cases. The last plot in figure 2 shows the final MA and Kclass loss, obtained by averaging over the last 10 epochs, as a function of the modulus ratio t^2/t^1 , which can be seen as a measure of the asymmetry of the manifold. There is a clear tendency for the final loss to increase with increasing asymmetry, a behavior which is intuitively expected.

Table 3. Exact volume from intersection form (row 2), and volume from numerical integration with g_{FS} (row 3) and g_{CY} (row 4), for the seven choices of Kähler parameters in table 2.

Case	1	2	3	4	5	6	7
Vol_{int}	8.49	4.97	2.93	2.02	6.87	7.59	6.16
Vol_{FS}	8.49	4.50	2.94	2.03	6.91	7.58	6.26
Error	<1%	<1%	<1%	<1%	<1%	<1%	~2%
Vol_{CY}	8.56	5.03	2.96	2.03	6.86	7.58	6.28
Error	<1%	~1%	<1%	<1%	<1%	<1%	~2%

5.2.3. Volume computations

From the runs described above we have obtained numerical results for the Ricci-flat CY metric $g_{CY}(p_i)$ at 100 000 points p_i on the bi-cubic and for seven choices of Kähler parameters. In addition, the `cymetric` package provides the weights w_i , the auxiliary weights $\tilde{w}_i$ and the Fubini–Study metric $g_{FS}(p_i)$ at those points.

As a first check, we would like to compute the CY volume for all seven cases, based on equation (6). The exact results are obtained from the intersection formula on the RHS of equation (6). Alternatively, these volumes can be computed by integrating over the CY or the FS measure, as in the middle of equation (6), and we will evaluate these integrals numerically, as explained in section 2.7. Explicitly, we compute for all seven cases

$$\text{Vol}_{\text{int}} = \frac{1}{6} d_{\alpha\beta\gamma} t^\alpha t^\beta t^\gamma, \text{Vol}_{FS} = \frac{1}{N} \sum_{i=1}^N \tilde{w}_i \det(g_{FS}(p_i)), \text{Vol}_{CY} = \frac{1}{N} \sum_{i=1}^N \tilde{w}_i \det(g_{CY}(p_i)), \quad (54)$$

where we recall that g_{CY} refers to the network’s prediction for the CY metric after the training is completed. The results are given in table 3.

The volumes computed with the Fubini–Study metric are in good agreement with the exact results, with most errors at the level of 1% or below. In this case, the only uncertainty comes from point sampling (as the Fubini–Study metric is known exactly) so these results confirm that our method for sampling points works and that 100 000 points are sufficient.

The more important results in table 3 are the volumes computed with the Ricci-flat CY metric, given in the last row. Again, the accuracy is impressive at 1% or less for most cases. Overall, these results confirm that the Ricci-flat CY metrics obtained are indeed in the prescribed Kähler class—only then can the volume be expected to come out correctly. We emphasize that obtaining the correct volume has *not* been built into the loss function. Rather, the correct Kähler class has been enforced during training by imposing the Kähler class loss (50). Table 3 provides strong evidence that this method does indeed work.

5.2.4. Slope computations

While the correct slope has been imposed during training it is still worth checking that our numerical metrics can be used for accurate slope computations. For the seven values of Kähler parameters $t_{(i)}$ and the seven line bundles $\mathcal{O}_X(k_{(j)})$ in table 2 we can compute a 7×7 slope matrix with entries $\mu_{t_{(i)}}(\mathcal{O}_X(k_{(j)}))$. The accurate entries of this matrix are obtained from the topological formula (8) and are given by

$$\begin{bmatrix} 0.00 & -18.00 & -36.00 & -54.00 & -18.00 & -18.00 & -36.00 \\ 9.16 & 0.00 & -9.16 & -18.33 & 9.16 & 18.33 & 9.16 \\ 10.94 & 5.47 & 0.00 & -5.47 & 16.41 & 27.34 & 21.87 \\ 11.46 & 7.64 & 3.82 & 0.00 & 19.1 & 30.57 & 26.75 \\ 6.45 & -6.45 & -19.34 & -32.24 & 0.00 & 6.45 & -6.45 \\ 4.85 & -9.7 & -24.25 & -38.8 & -4.85 & 0.00 & -14.55 \\ 7.63 & -3.82 & -15.26 & -26.71 & 3.82 & 11.45 & 0.00 \end{bmatrix} \quad (55)$$

where the rows are labeled by the values of the Kähler parameters and the columns by the line bundles. The zeros along the diagonal indicate the existence of HYM connections for these cases, that is, for the line bundles $\mathcal{O}_X(k_{(i)})$ and their associated Kähler parameters $t_{(i)}$.

Alternatively, we can work out the slope by carrying out the integrals in equation (10) numerically. Doing this first for the FS metrics, that is, evaluating $\frac{2}{N\pi} \sum_{i=1}^N \tilde{w}_i \det(g_{FS}(p_i)) \rho_{FS}(p_i)$ gives

$$\begin{bmatrix} 0.11 & -17.84 & -35.78 & -53.73 & -17.73 & -17.62 & -35.57 \\ 9.12 & -0.13 & -9.37 & -18.62 & 8.99 & 18.11 & 8.86 \\ 10.95 & 5.46 & -0.03 & -5.52 & 16.41 & 27.35 & 21.86 \\ 11.47 & 7.65 & 3.82 & -0.01 & 19.12 & 30.6 & 26.77 \\ 6.27 & -6.79 & -19.84 & -32.9 & -0.52 & 5.75 & -7.31 \\ 4.95 & -9.53 & -24.01 & -38.49 & -4.58 & 0.37 & -14.11 \\ 7.34 & -4.43 & -16.19 & -27.95 & 2.91 & 10.25 & -1.51 \end{bmatrix}. \quad (56)$$

Agreement with the exact results in (55) is impressive, with an average error of the non-zero slope of $\sim 3\%$, suggesting that the point sample size is sufficient, at least for the purpose of computing slopes. Using the Ricci-flat CY metrics and evaluating $\frac{2}{N\pi} \sum_{i=1}^N \tilde{w}_i \det(g_{CY}(p_i)) \rho_{CY}(p_i)$, we find

$$\begin{bmatrix} 0.03 & -17.97 & -35.98 & -53.98 & -17.95 & -17.92 & -35.92 \\ 8.86 & -0.42 & -9.70 & -18.98 & 8.43 & 17.29 & 8.01 \\ 10.11 & 4.52 & -1.07 & -6.65 & 14.63 & 24.74 & 19.15 \\ 9.96 & 5.96 & 1.97 & -2.02 & 15.92 & 25.87 & 21.88 \\ 6.38 & -6.45 & -19.29 & -32.12 & -0.07 & 6.31 & -6.53 \\ 4.96 & -9.41 & -23.77 & -38.13 & -4.45 & 0.51 & -13.86 \\ 7.53 & -4.12 & -15.76 & -27.4 & 3.41 & 10.93 & -0.71 \end{bmatrix}. \quad (57)$$

Again, this is in good agreement with the exact results in (55), with an average error of the non-zero slopes of $\sim 7\%$, and it confirms that we have obtained Ricci-flat CY metrics with the correct, intended Kähler class. We also note that, in most cases, the results are accurate enough to distinguish zero slope cases (along the diagonal) which allow for a HYM connection from cases with non-zero slope. For highly asymmetric manifolds, this distinction is less clear, which is related to the fact that training takes longer in these cases.

5.2.5. HYM connection

As a final application of our results we would like to compute an approximation to the HYM connection on a line bundle over the bi-cubic, following the procedure described in section 3. To do this we focus on the line bundle $\mathcal{O}_X(1, -1)$ and Kähler parameters $t_1 = t_2 \simeq 1.414$, corresponding to case 1 in table 2. As the result in (55) (specifically, the (11) entry of this matrix) shows, the slope of $\mathcal{O}_X(1, -1)$ vanishes for this choice of Kähler parameters, so that a HYM connection does indeed exist.

Our method requires choosing a set of functions (42) in which to expand the various quantities. We do this using the sections $\Gamma(\mathcal{O}_X(1, 1))$, a nine-dimensional space with basis $(x_a y_b)$, where $a, b = 0, 1, 2$. This leads to 81 functions of the form (42), explicitly given by

$$f_{(ab)(cd)} = \frac{x_a y_b \bar{x}_c \bar{y}_d}{(|x_0|^2 + |x_1|^2 + |x_2|^2)(|y_0|^2 + |y_1|^2 + |y_2|^2)}. \quad (58)$$

Note, these expressions are indeed of homogeneity degree zero in each set of $\mathbb{P}^2$ coordinates and, therefore, constitute functions on the ambient space and, by restriction, on the bi-cubic CY.

With these functions we first compute the 81×81 matrix Δ , which represents the Laplacian. Second, the 81-dimensional vector ρ , representing the inhomogeneity in Laplace's equations, is obtained by evaluating the matrix elements in equation (38) by numerical integration, using the point sample and numerical Ricci-flat metric for case 1. We can check that the matrix Δ obtained in this way has a one-dimensional kernel which corresponds to constant functions on the bi-cubic, as can be verified by taking the linear combination of the functions (58) with a vector in the kernel. As expected, the source vector ρ is not contained in the image $\text{Im}(\Delta)$, but we can verify that $|\rho^\perp|/|\rho| \sim 0.03$, so the component of ρ orthogonal to $\text{Im}(\Delta)$ is small. Then we solve equation (41) to find an 81-dimensional solution vector β and the linear combination of the functions (58) formed with this vector provides our approximation of the function β . Via equation (35), this function β then determines the approximation to the HYM bundle metric. For this result we compute the quantity (43) as a measure of the accuracy of the approximation. This comes out at $\sim 3\%$, indicating a reasonably accurate HYM connection.

5.3. Toric hypersurface CY with Picard rank 2

To test the `cymetric` package's ability to predict CY metrics on KS CY manifolds, we select a CY manifold with Picard rank two from the KS list¹⁷. After specifying the Kähler moduli as $(t_1, t_2) = (1, 1)$ and the complex structure moduli by a random assignment of the coefficient of the defining polynomial, we generate points on the manifold as specified in section 2.7. These points are then used in training the ϕ -model to predict the CY metric.

5.3.1. Geometric set-up and point sampling

The selected manifold has a toric ambient space with two Kähler moduli. The vertices¹⁸

$$v_0 = \begin{pmatrix} -1 \\ -1 \\ -1 \\ 0 \end{pmatrix}, \quad v_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad v_4 = \begin{pmatrix} 2 \\ 0 \\ 0 \\ -1 \end{pmatrix}, \quad v_5 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (59)$$

span a fan in the lattice N which defines a $\mathbb{P}^1$ -fibered toric ambient space over $\mathbb{P}^3$, $\mathcal{A} = \mathbb{P}^1 \rightarrow \mathbb{P}^3$, of Picard rank two. Each vertex corresponds to a homogeneous coordinate $x^i \sim v_i$, and a divisor $D_i = \{x^i = 0\}$; indeed, the vertices v_0, v_2, v_3, v_5 can be seen to correspond to $\mathbb{P}^3$ by projection onto the first three entries, and vertices v_1, v_4 give rise to a $\mathbb{P}^1$ by projection onto the last entry. There is a unique fine regular star triangulation of the corresponding polytope, giving eight top-dimensional cones spanned by four out of the six v_i . Thus, $\mathcal{A}$ can be covered by eight patches, and in each such patch there are four affine coordinates.

We can use the methods detailed in [appendix](#) to compute the sections of the anti-canonical bundle, whose linear combinations give a homogeneous polynomial that specifies a CY hypersurface X in $\mathcal{A}$. For this example, there are 80 sections, and we select random coefficients, corresponding to a generic choice for the complex structure moduli of the CY. Rather than writing out this lengthy expression, we note that a different random choice leads to comparable performance for the metric model.

Following the method described in section 2.7, we sample points on X using the toric point generator of `cymetric`. For this we note that the GLSM charges are

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & -2 \end{pmatrix}. \quad (60)$$

The Stanley–Reisner ideal is given by

$$\text{SRI} = \langle x^1 x^4, x^0 x^2 x^3 x^5 \rangle, \quad (61)$$

from which we can of course also infer the fibration structure of $\mathbb{P}^1$ over $\mathbb{P}^3$.

Using the GLSM charges we may read off the homogeneous scalings of the toric coordinates, and deduce the relations between the divisors:

$$D_0 \sim D_2 \sim D_3, \quad D_1 \sim D_4, \quad D_5 \sim D_0 - 2D_1. \quad (62)$$

We may thus choose $J_1 = D_1$ and $J_2 = D_3$ as generators for the Kähler cone, with non-vanishing triple intersection numbers given by

$$d_{122} = d_{212} = d_{221} = 4, \quad d_{222} = 8. \quad (63)$$

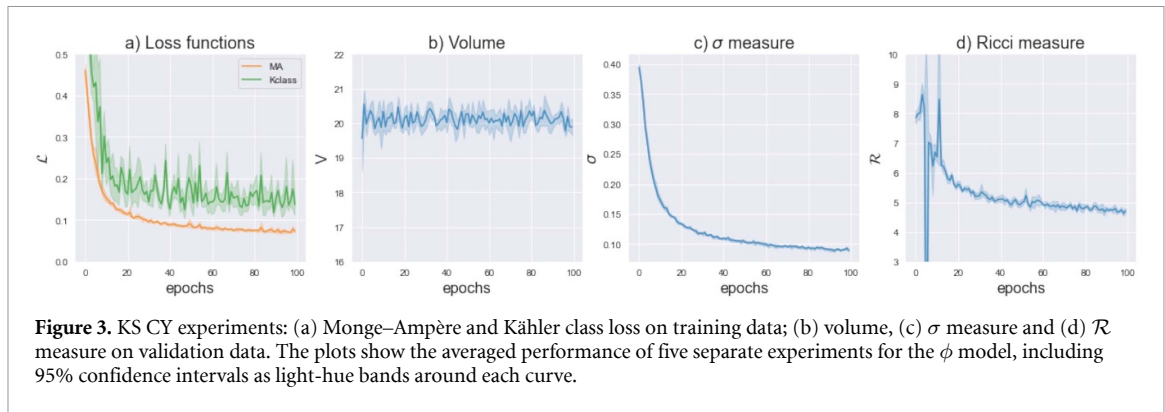
Finally, the sections of the Kähler cone generators are, for this example,

$$H^0(J_1) = (x_1, x_4), \quad H^0(J_2) = (x_0, x_2, x_3, x_1^2 x_5, x_4^2 x_5, x_1 x_4 x_5), \quad (64)$$

(which by (60) all have the appropriate scaling). This defines the morphisms $\Phi_{1,2}$ into $\mathbb{P}^1$ and $\mathbb{P}^5$, respectively, and allow us to construct points on the CY as discussed in section 2.7.

¹⁷ Picard rank 2 suffices to provide a non-trivial example of a KS CY manifold, while retaining a minimal computational complexity. We defer a study of higher Picard ranks to future work.

¹⁸ The reader unfamiliar with toric geometry may wish to consult [appendix](#).



5.3.2. Training with ϕ -model

We have performed five experiments using the ϕ -model and a NN with width 256, three hidden layers, GELU activation functions and initialization with $\mathcal{N}(0, 0.01)$. The network is trained for 100 epochs with a training set of 200 000 points, and error measures are computed on a validation set of 20 000 points. We chose default values for α using a SGD optimizer with learning rate 5/10 000 and momentum 0.95, and the batch sizes for the optimization steps were (128, 10 000). It was observed in the design phase that the two updates with different batch sizes required to minimize the Monge–Ampère loss while staying in the reference Kähler class, sometimes lead to the determinant of the predicted metric turning negative in isolated points during training. To ameliorate the long term effects of this behavior, training was conducted with lower learning rate and a larger momentum compared to the bicubic runs. The training took a couple of hours on a single GPU.

The results of the five experiments are shown in figure 3. We observe that the Monge–Ampère loss, $\mathcal{L}_{MA}$, decreases steadily over training, totaling to a factor 4 decrease over 100 epochs. While this is certainly less impressive than the improvement on the quintic, learning clearly takes place. For the ϕ model, the transition and Kähler losses are automatically zero, and hence are disabled. The decrease in the Kähler class loss, $\mathcal{L}_{Kclass}$, is also a factor 4, and the volume stays approximately constant around 20, which is the true value. The σ measure, which is calculated on validation data, also decreases steadily over training, showing that the NN provides a better and better solution to the Monge–Ampère equation. The most interesting part of this experiment is the evolution of the Ricci measure. During the initial 20 epochs, this goes through sharp spikes, corresponding to occurrences of points where the determinant of the predicted metric $g_{pr} = g_{FS} + \partial\bar{\partial}\phi$ becomes negative. This unphysical behavior is not lasting, and the training converges to a Ricci measure of around 5. While this is clearly non-zero, training decreases $\mathcal{R}$ by roughly 37%.

As for the bicubic, we can now compute the volume of the CY using the intersection numbers, FS and CY metric. We find, for $t_1 = t_2 = 1$,

$$\text{Vol}_{\text{int}} = 20, \quad \text{Vol}_{FS} = 19.98, \quad \text{Vol}_{CY} = 19.89, \quad (65)$$

using equation (54) and the intersection numbers given in (63), and Vol_{CY} refers to the network’s prediction for the CY volume after the training is completed, and is here computed as the mean of the five runs¹⁹. We find that the volumes agree within 1%, a result that testifies to the accuracy of the point generating routine and metric prediction.

6. Conclusions

In this paper, we have presented methods and results for numerical CY metrics in the context of machine learning algorithms. Broadly, we have made progress in four directions. (1) The library *cymetric* presented here contains functions which can be applied to the numerical computation of CY metrics on a wide range of manifolds, namely to complete intersection CYs in products of projective spaces (CICYs) and to CY manifolds obtained from the KS list. (2) In particular, we have realized point sampling for CY hypersurfaces in general toric ambient spaces. (3) We have shown how line bundle slopes can be used to obtain the numerical CY metric for given points in Kähler moduli space. (4) We have used the *cymetric* library to

¹⁹ For the five experiments, we find $\text{Vol}_{CY} = \{19.92, 19.73, 20.34, 19.49, 19.96\}$, respectively.

obtain new results on the bi-cubic CY, including the computation of volumes, slopes and HYM connections on line bundles, and on a Picard number two CY from the KS list.

The *cymetric* library consists of two main parts: the point generator and NNs which are used to predict the CY metric. For CICYs it is known for some time how to generate point samples with a defined distribution, following an algorithm explained in [16, 17] which is based on a theorem by Shiffman and Zelditch [50]. For CY hypersurfaces in toric varieties we have generalized this algorithm, making use of projective embeddings and the Shiffman-Zelditch theorem. Both versions of this algorithm, covering CICYs and KS manifolds, have been realized in the *cymetric* library.

The CY metric can be represented by a NN in a number of ways and the *cymetric* library contains five possible realizations, as summarized in table 1. To ensure that all NNs converge to the Ricci-flat CY metric for the specified values of the moduli, we use a custom loss function with five contributions. They encode the Monge–Ampère equation, the Ricci scalar, the Kähler property of the metric, the correct transition functions between coordinate patches, and the correct Kähler class. For the CY manifolds considered in this papers, the so called ϕ -network (the last entry in table 1), built from an exact correction to a reference Kähler metric, performs the best. This is largely explained by the fact that this Ansatz guarantees the Kähler and transition loss conditions are automatically satisfied. However, other network Ansätze might well turn out to be useful for different CY manifolds. They can also be adapted to non-Kähler situations, for example in the context of special structure manifolds with torsion [32].

In addition to testing our library on the quintic CY we have also performed runs on the bi-cubic CY, a manifold with Picard number two. On this manifold, we have computed the numerical metric at seven specific points in Kähler moduli space. The results have been used to compute, via numerical integration with the CY metric, the CY volumes and some line bundle slopes at those seven points in moduli space. These quantities can also be computed exactly by intersection formulae and the good agreement of our numerical results, typically at the level of $\sim 1\%$ or smaller for the volume and a few percent for the slope, confirms that the numerical metrics have indeed been computed at the correct points in moduli space. We have also considered a Picard number two CY from the KS list and have carried out a computation of the CY metric at a specific point in its Kähler moduli. Again, we were able to reproduce the CY volume from numerical integration, indicating that the toric point generator works correctly and that the metric is indeed computed at the desired point in Kähler moduli space.

In a final step we have used the numerical CY metric to find numerical solutions of the HYM equation for line bundles on the bicubic. A necessary and sufficient condition for the existence of such solutions is that the line bundle has vanishing slope, a condition which can, depending on the choice of the bundle, be satisfied at some particular locus in Kähler moduli space. Using the *cymetric* package, we have computed the numerical CY metric at such a point in moduli space. At this point we have then numerically solved the HYM equation for the line bundle metric, which can be cast in the form of an inhomogeneous Laplace equation. In this way, we find a numerical HYM line bundle metric at percent-level accuracy.

The goal is for *cymetric* to become an active open source community project similar to popular projects in the machine learning community. To this end we encourage researchers to include their own models and extensions. In the future, we plan to expand its capabilities by adding new models and NN architectures based on more systematic hyperparameter searches. It is important to emphasize that algebraic metrics such as in [22] can be implemented as special cases, by creating a custom network that just learns the coefficients in the polynomial approximating the Kähler potential. The same holds true for learning metrics, akin to the h -balanced metric from Donaldson’s algorithm, on holomorphic sections of line bundles. The package then allows the full power of the TensorFlow optimizer to be applied in this context. We hope that the generality and flexibility of the package will help to solve important physics problems, such as, for example, the calculation of the physical Yukawa couplings from string theory.

There are a number of interesting additions, modifications and new directions which should be mentioned. It may be interesting to try out different methods for point sampling, for example a Markov Chain Monte Carlo sampling based on the measure dV_Ω , in order to complement (and check) the methods based on the Shiffman and Zelditch theorem. The *cymetric* library allows computing the CY metric at given, specified points in moduli space. It would be interesting to use this feature and study the dependence of the CY metric on the moduli in more detail. We should also point out that, while the present paper has focused on CY three-folds, the *cymetric* library can be applied to CY n -folds more generally. For example, it would be interesting to compute CY metrics on K3 and compare with recent analytic solutions found in [8, 9], or compute a CY metric on a four-fold. Another direction is to investigate whether the present methods generalize to related settings, such as to manifolds with SU(3)-structure [53] or manifolds with G_2 holonomy. Work in this direction is in progress.

Data availability statement

All data have been generated, and can be reproduced, with the *cymetric* package. The data that support the findings of this study are available upon request from the authors.

Acknowledgments

We thank Callum Brodie for useful discussions. The work of F R is supported by startup funding from Northeastern University. The work of M L and R S is funded by Vetenskapsrådet, under Grant 2020-03230. Computations were in part enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at the HPC cluster *Tetralith*, partially funded by the Vetenskapsrådet through Grant Agreement No. 2018-05973, the Hydra cluster of the Physics Department at the University of Oxford, and the computing cluster at CERN.

Appendix. Some toric geometry

We collect some results from toric geometry that are used in the main part of the text. Toric varieties serve as ambient spaces for the CY manifolds we want to study. We will restrict the discussion to the case where the CY manifold is given as a hypersurface in a toric ambient space or as a complete intersection in a direct product of projective ambient spaces. We will focus on the parts relevant to this setup only, and use succinct and simple exposition, somewhat at the expense of precise terminology (for example, we will be referring to finitely generated strongly convex rational polyhedral cones simply as cones). For an introduction to toric geometry see the excellent textbooks [48, 54]. For the benefit of the ML-interested reader, we note that topological properties of toric varieties and CY hypersurfaces therein have been studied using ML tools in [55–59].

Throughout the discussion, we focus on the more general toric case, and point out how this simplifies for (products of) projective spaces along the way.

Fan and cones

A d -dimensional toric variety is described in terms of a fan, which is a collection of k -dimensional cones, where $k = 0, 1, \dots, d$. The one-dimensional cones v_i are called rays, and they generate a lattice $N \subseteq \mathbb{Z}^d$. Each vertex $v_i \in N$, $i = 1, \dots, n$ corresponds to one homogeneous coordinate x_i of the toric ambient space. Their toric scalings q_α are given by the kernel of the vertex matrix:

$$q_\alpha^i v_i = 0 \quad \Rightarrow \quad (x_1, x_2, \dots, x_n) \rightarrow (\lambda_\alpha^{q_\alpha^1} x_1, \lambda_\alpha^{q_\alpha^2} x_2, \dots, \lambda_\alpha^{q_\alpha^n} x_n). \quad (\text{A.1})$$

The matrix q_α^i is called the GLSM charge matrix or the Mori cone matrix. Note that for a given set of charge vectors $\vec{q}_\alpha$, we can form a new set $\vec{q}_\beta = R_\beta^\alpha \vec{q}_\alpha$ by rotating the charges. One typically chooses the basis such that the GLSM D-terms

$$\sum_i q_\alpha^i |x_i|^2 = t_\alpha \quad (\text{A.2})$$

have a solution for Kähler parameters $t_\alpha \geq 0$.

For a projective ambient space $\mathbb{P}^{n_\alpha}$, the GLSM charge matrix is simply $q_\alpha^i = (1, 1, \dots, 1)$ with $i = 1, \dots, n_\alpha + 1$. Hence, for products of projective ambient spaces, we obtain a block-diagonal charge matrix. In contrast to the toric case, this does not encode any information beyond what is specified by just giving the dimensions n_α of the projective ambient space factors.

The top-dimensional, that is, d -dimensional, cones are called facets. All other cones are typically just referred to as faces. A k -dimensional cone $\sigma_{i_1, i_2, \dots, i_k}$ in $N_\mathbb{R} = N \times \mathbb{R}$ is generated by vertices $v_{i_1}, v_{i_2}, \dots, v_{i_k}$,

$$\sigma_{i_1, i_2, \dots, i_k} = \left\{ \sum_{i=1}^k \lambda_i v_i \in N_\mathbb{R} \mid \lambda_i \geq 0 \quad \forall k \right\}. \quad (\text{A.3})$$

Associated to this, we can define a dual cone σ^* in the dual lattice $M_\mathbb{R} \subseteq \mathbb{Z}^d$,

$$\sigma_{i_1, i_2, \dots, i_k}^* = \{ w \in M \mid \langle v, w \rangle \geq 0 \quad \forall v \in \sigma_{i_1, i_2, \dots, i_k} \}. \quad (\text{A.4})$$

Note that the dual of a top-dimensional cone $\sigma_{i_1, i_2, \dots, i_d}$ is again d -dimensional.

Affine patches

Each top-dimensional cone σ of a fan corresponds to an affine patch $\mathbb{C}^d$. The coordinates that are allowed to vanish in this patch are the ones given by the generators of σ , while all others are non-vanishing. We can construct the affine coordinates z_μ from the dual cone σ^* by taking the product of all coordinates x_i with exponents given by the inner product of the corresponding vertex v_i with the generators w_μ of the dual cone σ^* , so that

$$z_\mu = \prod_{i=1}^n x_i^{\langle v_i, w_\mu \rangle}. \quad (\text{A.5})$$

Note that in particular σ^* is d -dimensional. Hence a toric variety of complex dimension d always has d affine coordinates in any given patch. For a projective ambient space $\mathbb{P}^{n_\alpha}$, this construction reduces to the standard patches $\mathcal{U}_a = \{z_a \neq 0\}$ with affine coordinates $z_\mu = x_i/x_a$ for $i \neq a$.

Divisors and Stanley–Reisner ideal

Each vertex $v_i \in N$ corresponds to one homogeneous coordinate x_i of the toric ambient space. Hence, we can introduce a set of (in general linearly dependent) divisors $D_i := \{x_i = 0\}$. Two ambient space divisors $D = \sum_i c_i D_i$ and $D' = \sum_i c'_i D_i$ for some $c_i, c'_i \in \mathbb{Z}$ are linearly equivalent if all their GLSM charges agree,

$$D \sim D' \quad \Leftrightarrow \quad \sum_i c_i q_\alpha^i = \sum_j c'_j q_\alpha^j \quad \forall \alpha. \quad (\text{A.6})$$

In particular, this means that favorable CYs (that is, CYs which inherit the entire second cohomology from the ambient space) lead to the index range $\alpha = 1, 2, \dots, h^{1,1}(X)$.

If certain divisors are not contained in a common cone of the fan, they cannot vanish simultaneously. We associate a monomial $s = \prod x_i$ to each such combination. The collection of these monomials generates an ideal,

$$\text{SRI} = \langle s_i \rangle, \quad (\text{A.7})$$

called the Stanley–Reisner ideal. A toric variety is specified by the GLSM charges and the SRI, or, equivalently, by its vertices plus a chosen triangulation.

For a projective space $\mathbb{P}^{n_\alpha}$, the Stanley–Reisner ideal is generated by a single monomial, $\text{SRI} = \langle x_0 x_1 \dots x_{n_\alpha} \rangle$, which corresponds to the familiar statement that not all coordinates of $\mathbb{P}^{n_\alpha}$ can vanish simultaneously.

Line bundles and sections

The sections $\Gamma(D)$ of a line bundle L associated to a divisor D can be readily computed from the vertices v_i of N and their dual vertices w in the dual lattice M :

$$D = \sum_{i=1}^n c_i D_i \quad \rightarrow \quad \Gamma(D) \sim \left\{ \prod_{i=1}^n x_i^{\langle v_i, w \rangle + c_i} \mid \langle v_i, w \rangle + c_i \geq 0, \quad w \in M \right\}. \quad (\text{A.8})$$

Note that for non-negative c_i , only finitely many $w \in M$ satisfy the condition $\langle v_i, w \rangle + c_i \geq 0$. For a divisor D inside a product of n projective ambient spaces, this reduces to the statement that a section basis for D is given by the monomials of multi-degree $(c_1, c_2, \dots, c_n)$.

One can prove a theorem which will be important for constructing Kähler forms later. It states that for any nef divisor (or rather the sections of the associated nef line bundle), precisely one of the monomials that form a basis of the sections vanishes nowhere in a given patch. This means that the base-point free divisors are given by sections of nef line bundles.

Since the anti-canonical divisor of a toric variety is given by the sum over all divisors, $-K = \sum_{i=1}^n D_i$ where n is the number of vertices, this motivates defining a lattice polytope $P \in N$ given by the convex hull of all vertices v_i , and its dual lattice polytope $P^* \in M$ in the dual M -lattice via

$$P^* := \{w_i \in N \mid \langle v_i, w_j \rangle \geq -1 \quad \forall v_i \in P\}. \quad (\text{A.9})$$

For the purpose of CYs given as hypersurfaces in toric varieties, we only need to look at reflexive polytopes, which are lattice polytopes whose dual are again lattice polytopes. All reflexive polytopes have the origin as their unique interior point. Their dual is also a reflexive polytope and satisfies $(P^*)^* = P$. With this, it is easy

to find a basis of sections of the anti-canonical bundle and which then allows writing down the most general CY defining equation

$$p = \sum_{w \in P^*} c_w \prod_{i=1}^n x_i^{\langle v_i, w \rangle}. \quad (\text{A.10})$$

Here, the coefficients $c_w \in \mathbb{C}$ parameterize the complex structure of the CY.

In the case of ambient spaces $\mathcal{A}$ that are a product of projective spaces, $\mathcal{A} = \otimes_{\alpha=1}^N \mathbb{P}^{n_\alpha}$, we are considering complete intersections of k hypersurfaces H_i , $i = 1, \dots, k$ rather than just a single hypersurface. This requires specifying the multi-degrees, c_i^α , of the normal bundle sections whose complete intersection defines the CY manifold. The CY condition amounts to the constraints $\sum_i c_i^\alpha = n_\alpha + 1$, where $\alpha = 1, \dots, k$, on these multi-degrees. (In the toric case, this gives rise to nef partitions of the reflexive polytope, but we will not consider such cases here, even though our methods still apply.) Moreover, for a CY d -fold, we require that $\sum_\alpha n_\alpha - k = d$. This means that we can specify a complete intersection CY manifold inside an ambient space that is given as a product of projective spaces purely by specifying the degrees c_i^α , which form a matrix referred to as the ‘configuration matrix’ of the CICY.

ORCID iDs

Magdalena Larfors  <https://orcid.org/0000-0003-1011-8777>

Fabian Ruehle  <https://orcid.org/0000-0002-8409-9823>

Robin Schneider  <https://orcid.org/0000-0002-4438-0901>

References

- [1] Candelas P, Horowitz G T, Strominger A and Witten E 1985 Vacuum configurations for superstrings *Nucl. Phys. B* **258** 46–74
- [2] Green P S, Hubsch T and Lutken C A 1989 All Hodge numbers of all complete intersection Calabi–Yau manifolds *Class. Quant. Grav.* **6** 105–24
- [3] Candelas P, Dale A M, Lutken C A and Schimmrigk R 1988 Complete intersection Calabi–Yau manifolds *Nucl. Phys. B* **298** 493
- [4] Kreuzer M and Skarke H 2000 Complete classification of reflexive polyhedra in four-dimensions *Adv. Theor. Math. Phys.* **4** 1209–30
- [5] Hubsch T 1994 *Calabi–Yau Manifolds: A Bestiary for Physicists* (Singapore: World Scientific)
- [6] Calabi E 2015 *On Kähler Manifolds with Vanishing Canonical Class* (Princeton, NJ: Princeton University Press) pp 78–89
- [7] Yau S-T 1978 On the Ricci curvature of a compact Kähler manifold and the complex Monge–Ampère equation, I *Commun. Pure Appl. Math.* **31** 339–411
- [8] Kachru S, Tripathy A and Zimet M 2020 K3 metrics (arXiv:2006.02435)
- [9] Kachru S, Tripathy A and Zimet M 2018 K3 metrics from little string theory (arXiv:1810.10540)
- [10] Green M B, Schwarz J H and Witten E 2012 *Superstring Theory: Volume 2, Loop Amplitudes, Anomalies and Phenomenology* (Cambridge: Cambridge University Press)
- [11] Ashmore A and Ruehle F 2021 Moduli-dependent KK towers and the swampland distance conjecture on the quintic Calabi–Yau manifold *Phys. Rev. D* **103** 106028
- [12] Strominger A, Yau S-T and Zaslow E 1996 Mirror symmetry is T duality *Nucl. Phys. B* **479** 243–59
- [13] Hori K, Katz S, Pandharipande R, Vakil R and Zaslow E 2003 *Mirror Symmetry* vol 1 (Providence, RI: American Mathematical Society)
- [14] Donaldson S K 2005 Some numerical results in complex differential geometry (arXiv:math/0512625)
- [15] Douglas M R, Karp R L, Lukic S and Reinbacher R 2007 Numerical solution to the Hermitian Yang–Mills equation on the Fermat quintic *J. High Energy Phys.* **2007** 083
- [16] Douglas M R, Karp R L, Lukic S and Reinbacher R 2008 Numerical Calabi–Yau metrics *J. Math. Phys.* **49** 032302
- [17] Braun V, Brelidze T, Douglas M R and Ovrut B A 2008 Calabi–Yau metrics for quotients and complete intersections *J. High Energy Phys.* **2008** 080
- [18] Braun V, Brelidze T, Douglas M R and Ovrut B A 2008 Eigenvalues and eigenfunctions of the scalar Laplace operator on Calabi–Yau manifolds *J. High Energy Phys.* **2008** 120
- [19] Afkhami-Jeddi N, Ashmore A and Cordova C 2022 Calabi–Yau CFTs and random matrices *J. High Energy Phys.* **2022** 021
- [20] Anderson L B, Braun V, Karp R L and Ovrut B A 2010 Numerical hermitian Yang–Mills connections and vector bundle stability in heterotic theories *J. High Energy Phys.* **2010** 107
- [21] Anderson L B, Braun V and Ovrut B A 2012 Numerical hermitian Yang–Mills connections and Kahler cone substructure *J. High Energy Phys.* **2012** 014
- [22] Headrick M and Nassar A 2013 Energy functionals for Calabi–Yau metrics *Adv. Theor. Math. Phys.* **17** 867–902
- [23] Cui W and Gray J 2020 Numerical metrics, curvature expansions and Calabi–Yau manifolds *J. High Energy Phys.* **2020** 044
- [24] Ashmore A, Calmon L, He Y-H and Ovrut B A 2021 Calabi–Yau metrics, energy functionals and machine-learning (arXiv:2112.10872)
- [25] He Y-H 2017 Deep-learning the landscape (arXiv:1706.02714)
- [26] Ruehle F 2017 Evolving neural networks with genetic algorithms to study the string landscape *J. High Energy Phys.* **2017** 038
- [27] Krefl D and Seong R-K 2017 Machine learning of Calabi–Yau volumes *Phys. Rev. D* **96** 066014
- [28] Carifio J, Halverson J, Krioukov D and Nelson B D 2017 Machine learning in the string landscape *J. High Energy Phys.* **2017** 157
- [29] Ruehle F 2020 Data science applications to string theory *Phys. Rep.* **839** 1–117
- [30] Ashmore A, He Y-H and Ovrut B A 2020 Machine learning Calabi–Yau metrics *Fortsch. Phys.* **68** 2000068
- [31] Douglas M R, Lakshminarasimhan S and Qi Y 2020 Numerical Calabi–Yau metrics from holomorphic networks (arXiv:2012.04797)

- [32] Anderson L B, Gerdes M, Gray J, Krippendorf S, Raghuram N and Ruehle F 2021 Moduli-dependent Calabi–Yau and $SU(3)$ -structure metrics from machine learning *J. High Energy Phys.* **2021** 013
- [33] Jejjala V, Mayorga Pena D K and Mishra C 2020 Neural network approximations for Calabi–Yau metrics (arXiv:2012.15821)
- [34] Larfors M, Lukas A, Ruehle F and Schneider R 2021 Learning size and shape of Calabi–Yau spaces *4th Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021)* (arXiv:2111.01436)
- [35] Ashmore A, Deen R, He Y-H and Ovrut B A 2021 Machine learning line bundle connections (arXiv:2110.12483)
- [36] Ruehle F and Schneider R 2021 Cymetric <https://github.com/pythoncymetric/cymetric> (accessed 1 June 2022)
- [37] Wolfram Research, Inc. 2021 *Mathematica, Version 12.3.1* (Champaign, IL) (available at: www.wolfram.com/mathematica)
- [38] Harris C R et al 2020 Array programming with NumPy *Nature* **585** 357–62
- [39] McKinney W 2010 Data structures for statistical computing in Python *Proc. 9th Python in Science Conf.* ed S van der Walt and J Millman pp 56–61
- [40] The Sage Developers 2021 *SageMath, the Sage Mathematics Software System (Version 9.4.0)* (available at: www.sagemath.org)
- [41] Abadi M et al 2015 TensorFlow: large-scale machine learning on heterogeneous systems (available at: www.tensorflow.org/)
- [42] Novoseltsev A and Braun V 2021 *SageMath, Toric Varieties (Version 9.4.0)* (available at: <https://doc.sagemath.org/html/en/reference/schemes/sage/schemes/toric/variety.html>)
- [43] Kreuzer M and Skarke H 2004 PALP: a package for analyzing lattice polytopes with applications to toric geometry *Comput. Phys. Commun.* **157** 87–106
- [44] Demirtas M and Rios-Tascon A 2021 CY tools—a software package for analyzing Calabi–Yau hypersurfaces in toric varieties (available at: <https://github.com/LiamMcAllisterGroup/cytools>)
- [45] Gray J, He Y-H, Ilderton A and Lukas A 2009 STRINGVACUA: a mathematica package for studying vacuum configurations in string phenomenology *Comput. Phys. Commun.* **180** 107–19
- [46] Blumenhagen R, Jurke B, Rahn T and Roschy H 2010 cohomcalg package high-performance line bundle cohomology computation based on [60] (available at: <https://github.com/BenjaminJurke/cohomCalg>)
- [47] Larfors M and Schneider R 2019 pyCICY—a Python CICY toolkit implemented for [61] (available at: <https://github.com/robin-schneider/CICY>)
- [48] Cox D, Little J and Schenck H 2011 *Toric Varieties (Graduate Studies in Mathematics)* (Providence, RI: American Mathematical Society)
- [49] Borinsky M, Sattelberger A-L, Sturmfels B and Telen S 2022 Bayesian integrals on toric varieties (arXiv:2204.06414)
- [50] Shiffman B and Zelditch S 1999 Distribution of zeros of random and quantum chaotic sections of positive line bundles *Commun. Math. Phys.* **200** 661–83
- [51] Mehta P, Bukov M, Wang C-H, Day A G R, Richardson C, Fisher C K and Schwab D J 2019 A high-bias, low-variance introduction to machine learning for physicists *Phys. Rep.* **810** 1–124
- [52] Schneider R 2022 Heterotic compactifications in the era of data science *PhD Thesis* Uppsala University
- [53] Larfors M, Lukas A and Ruehle F 2019 Calabi–Yau manifolds and $SU(3)$ structure *J. High Energy Phys.* **2019** 171
- [54] Fulton W 1993 *Introduction to Toric Varieties (Annals of Mathematics Studies)* (Princeton, NJ: Princeton University Press)
- [55] Altman R, Carifio J, Halverson J and Nelson B D 2019 Estimating Calabi–Yau hypersurface and triangulation counts with equation learners *J. High Energy Phys.* **2019** 186
- [56] Demirtas M, McAllister L and Rios-Tascon A 2020 Bounding the Kreuzer–Skarke landscape *Fortsch. Phys.* **68** 2000086
- [57] Bao J, He Y-H, Hirst E, Hofscheier J, Kasprzyk A and Majumder S 2021 Polytopes and machine learning (arXiv:2109.09602)
- [58] Berglund P, Campbell B and Jejjala V 2021 Machine learning Kreuzer–Skarke Calabi–Yau threefolds (arXiv:2112.09117)
- [59] Berman D S, He Y-H and Hirst E 2022 Machine learning Calabi–Yau hypersurfaces *Phys. Rev. D* **105** 066002
- [60] Roschy H and Rahn T 2010 Cohomology of line bundles: proof of the algorithm *J. Math. Phys.* **51** 103520
- [61] Larfors M and Schneider R 2019 Line bundle cohomologies on CICYs with Picard number two *Fortsch. Phys.* **67** 1900083